Quantyl® Discovery user guide

This document gives fundamentals to use Quantyl Discovery software.

James Dugalais
7/2/2019

DISCLAIMER

Quantyl technologies corp provides this documentation without warranty, term, or condition of any kind, either implied or expressed, including, but not limited to, the implied warranties, terms or conditions of merchantability, satisfactory quality, and fitness for a particular purpose. Quantyl technologies corp, its employees and agents will not be responsible for any loss, however arising, from the use of, or reliance on this information.

COPYRIGHT

Table of Contents

1 DOCUMENT INTRODUCTION

1.1 TARGETED AUDIENCE

This document is dedicated to Quantyl Software users. This document assumes that the reader has some minimal notions of **data visualization** (data tables, scatterplots, bar charts, pie charts, histograms knowledge).

This document is not exhaustive; it gives key features and details on how they are working. Obvious commands are self-described by "contextual help" (tooptip info. displayed when mouse is moved over an icon).

"Quantyl" will mostly refer to "Quantyl Discovery" product in the document.

1.2 HOW THIS DOCUMENT SHOULD BE USED?

We recommend **making a first read** of this User Guide document sequentially. Indeed, some concepts presented in earlier sections are required to understand following sections.

Please, refer to the naming conventions section to get explanations on some terms and acronyms used in this document.

1.3 SOFTWARE GUIDELINES AND BEST PRACTICES

Quantyl has been designed to **maximize the number of capabilities** with a minimal feature set that can be easily filtered, enriched and extended (APIs, Add-in mechanism). Keep in mind that Quantyl gives to the user a **maximum of freedom** in features usage and combinations. The user experience is not driven by "Wizard tools" but organized around independent features and commands that can be **combined** in a non-sequential manner.

So, this document **cannot sweep all possible usages** of the product but **gives you the keys to understand** its underlying paradigms and logic.

Screen captures

This document is based on a lot of screen captures to illustrate features. Some graphics in the UI may differ a little bit from the definitive release but should not be source of confusion.

2 SOFTWARE REQUIREMENTS

2.1 APPLICATION

Requirements	
Must have	• OS: **Windows 10 (v1607) Anniversary Edition (version10.0; build 14393) 64-bits or higher.** • Memory: **2GB** min • Disk: **1GB** for install • Mouse with **left, right buttons** and **scroll wheel** • **OpenGL** compatible graphic card (>= OGL 1,5) ⚠ *Quantyl graphic engine is OpenGL based, but NO HIGH-END graphic card is needed here. Even an integrated Graphics card is ok. Prefer to invest in fast Multicore CPU with big memory cache to boost Quantyl performances.* • Minimum Color Depth: **24-bit or higher** • Minimum screen resolution: **1024x768 or higher**
Recommended	• Memory: **8GB** (or >) • Large and high speed **SSD disk (>= 256GB) is highly recommended** to speed up I/O. The Amount of cache on the disk is important. By default, Quantyl user data are installed in [my documents]\Quantyl folder. This is this folder that needs to be installed on your highest speed disk • **Four cores** CPU or better, Quantyl performances will scale with the number of cores • Screen Resolution: 1920x1080 or higher (2560 * 1440 is perfect)

Notes:

- *High GPU speed and CUDA compatible graphic board are **unnecessary***
- *Multi-touch screen is supported*

Memory management is critical in any high-performance computing process as it directly impacts on computing speed & amount of data processed. Memory has always been a critical resource when we aim the best performances.

64-bit systems became cheap enough to consider in-memory technology where it was impossible with 4GB barrier on 32bits machine. **Quantyl works exclusively on x64 windows system**.

2.2 INSTALLATION

Quantyl installation is straightforward. It has few dependencies already packed with Quantyl app.

- **Certified Microsoft application, all in one install**
- **No** external database pre-requisites
- **No** permanent internet connection required, except for licensing activation

You will be prompted during the first launch to download and install demopack containing multiple Workspaces and data examples.

2.3 GRAPHICS CARD(S)

Important

Professional laptops are generally equipped with a specific graphics card to complete integrated graphics. If you have a 3D / OpenGL graphic card on your machine, be sure that Quantyl application benefits from it by adding the program in your graphic card option.

2.4 HIGH DPI DEVICES (HDPI) AND WINDOWS SCALING

Though Full HD (1,920 x 1,080) and higher displays are much more common, and 4K UHD (3,840 x, 2,160) displays are arriving on an increasing number of machines.

High resolutions are great, but when they're packed into smaller displays, on-screen items can become very small and thus hard to see and use. This is measured by looking at PPI (often named DPI).

By default, Quantyl Discovery application is "system DPI aware", it means it will adapt its user interface (font, icons, graphs) depending on the current Windows Scaling ratio you are using. This default mode lets you benefit of your full resolution without bitmap scaling the full application.

If you want to **bitmap scale the full application**, an option exists in the application (see screenshot below). It may be handy in some rare scenarios to optimize graphical performances or power consumption (4K to 2K). Texts will be blurry in this mode if you are using a fractional Windows scaling value, and you won't benefit of the native resolution of your device.

In both modes avoid fractional scaling if you can, prefer 100% (no scaling) or 200%.

In both modes, you can adjust texts and glyphs size in the graphical area by changing manually PPI settings inside the app.

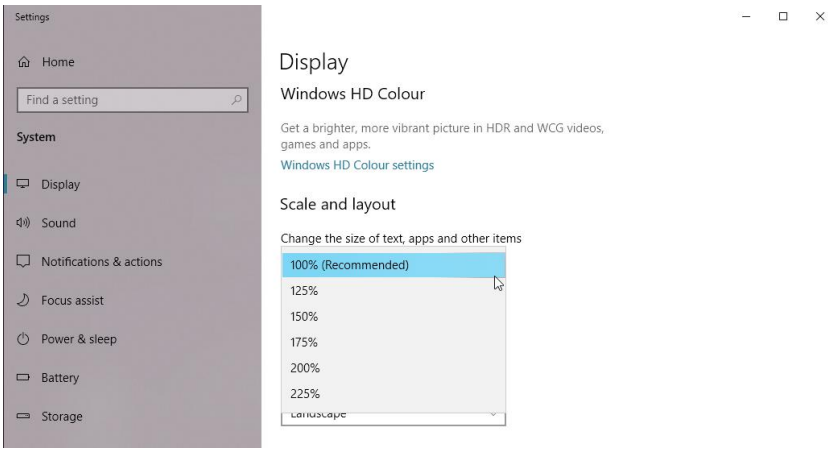


Figure 1 Windows scaling, system settings

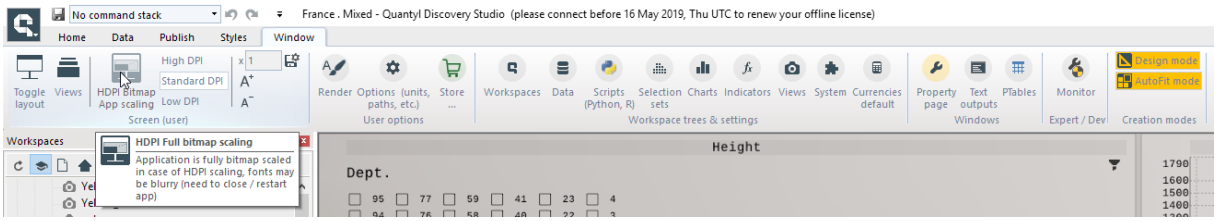


Figure 2 Full bitmap scaling option in Quantyl Discovery

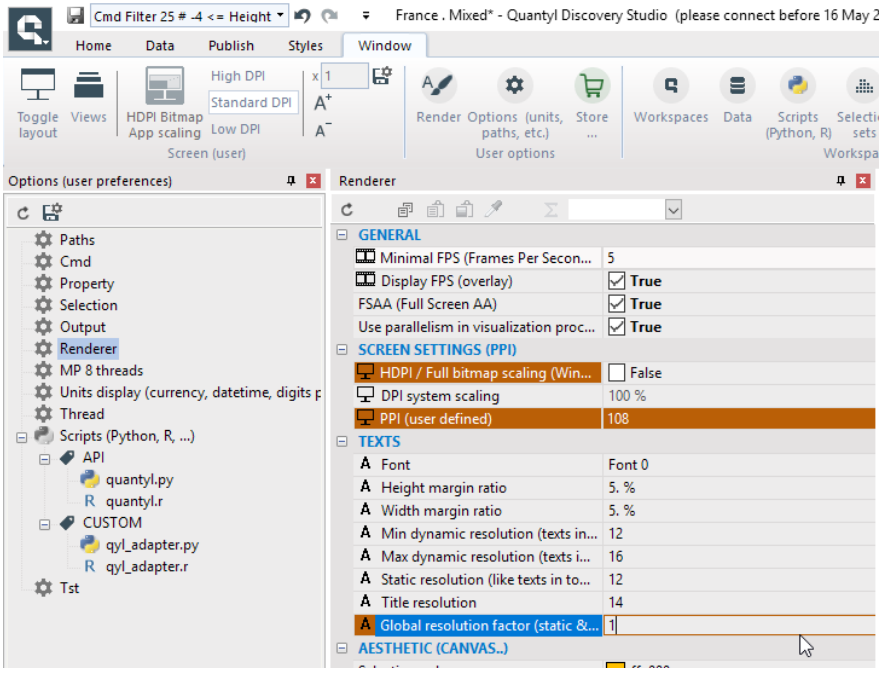


Figure 3 Custom PPI adjustment in Quantyl Discovery

3 GETTING STARTED

3.1 WHAT IS QUANTYL DISCOVERY?

Data have insights to deliver, stories to tell. The mission of Quantyl Discovery is to assist you to explore them.

More pragmatically, Quantyl Discovery is a standalone application that extracts dimensions and measures from tabulated data to provide a rich set of **interactive** tools to filter, explore, aggregate, visualize and animate those data. It's a self-service exploration tool to get valuable data insights where visualization-authoring and consumption are merged. Quantyl Discovery is designed to be **agile, ease to use, non-intrusive and serverless**.

Data preparation / aggregation process is not required, everything is performed on the fly, leveraging memory cache and multicore CPU architecture on commodity Windows machine. Quantyl is designed to scale up on multi-core CPUs. You open/import your data directly in the application and start to work.

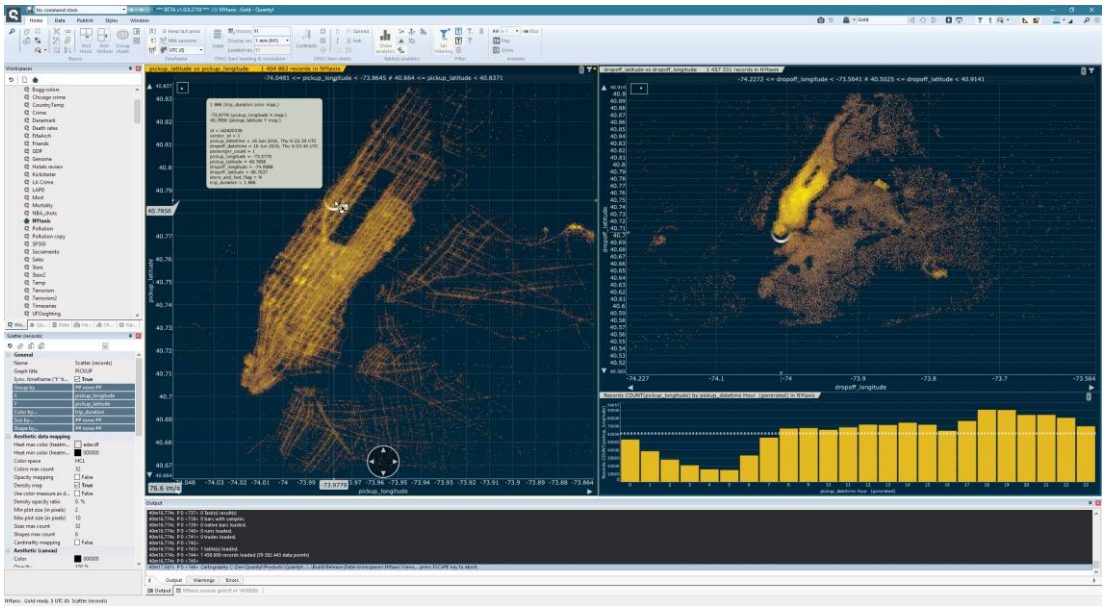
Quantyl is non-intrusive (non-destructive workflow), it extracts needed data in memory and saves in its own format (a mix of binaries and .xml). **Quantyl does NOT modify your data,** they are read-only. The content (your study) created in Quantyl (workspaces and views) may be updated when input data has changed.

A workspace captures all needed data (except contracts) and content, without external reference. You may archive your workspace in a compact zip-archive to share it or for later use. **It is self-contained**.

Unlike BI tools, it is not focused on data Aggregation, it is not focused on dashboard usage. You can still work **on fine grain records** without performances drop, visualizing several millions of data points sorted in different layers, on a standard 64-bits OS-Window machine like a tablet, a laptop or a desktop PC. Of course, you still can filter and **aggregate** data in any dimension you need, like you would do in a BI tool.

From a technology point of view, Quantyl has more to share with game engines than D3.js or any web technology which are mainstream in data visualization. Quantyl Discovery is designed to support many charts with ease, all charts interactively connected with cross selection, dynamic filtering, animation. This emphasis on multiple charts and performances offers new way to explore data.

Its user experience and flexibility make it unique on the market for exploring data in the most efficient ways.



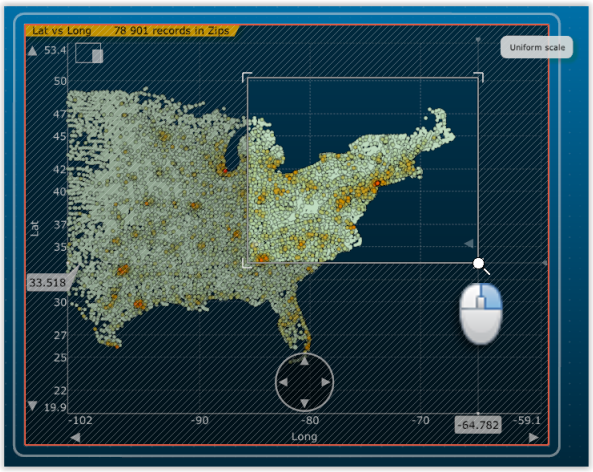
3.2 NAVIGATION & SELECTION COMMANDS

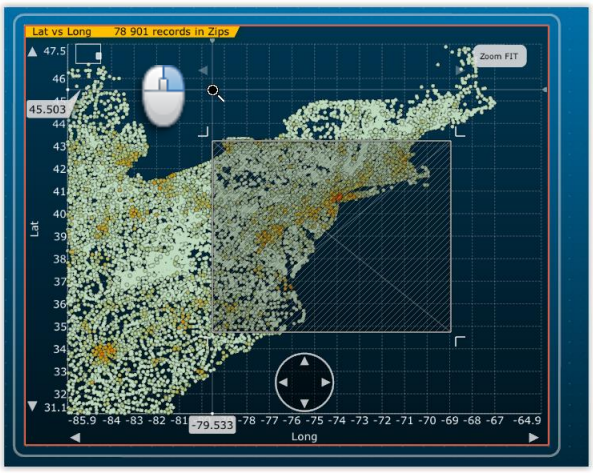
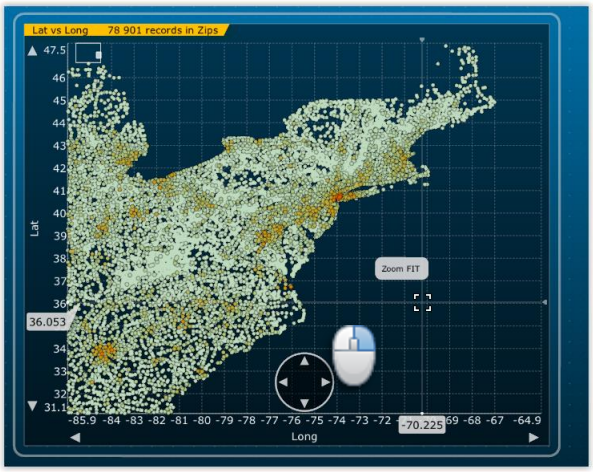
Depending on the context defined by the object under the mouse when action is started, navigation is performed on:

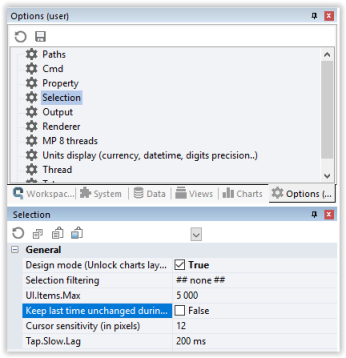
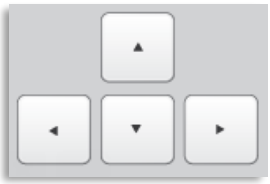
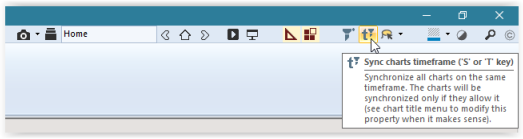
1. The chart itself
2. The complete chart area (press **Ctrl key** to force this mode)
3. Axis of the chart (Timeframe, X, Y…)
4. Graph area

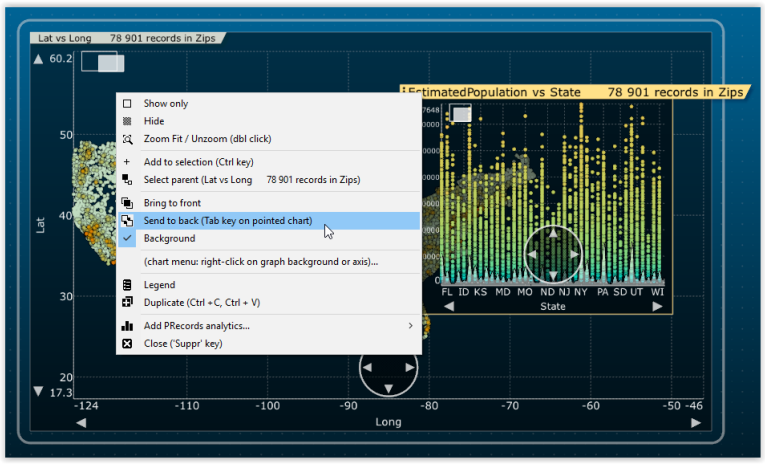
Mouse button and Ctrl key combinations are quite standard ones (Similar to Microsoft© Office ones)

Command	Description
 Mouse move	Objects are highlighted under mouse cursor
 Left click Touch: **tap**	Unselect previous selection, and select object under mouse
 Left click + **Ctrl key**	Add object to the current selection
 Left click + **Shift key**	Add object to the current selection if it is not yet selected, else remove it from the current selection
 **Double left click** Touch: **double tap**	**Zoom / Unzoom** object under mouse, it may be: • Chart itself • Time frame axis (fit time frame) • Price Axis (fit prices) • Bar (fit around bar timeframe)
 +  Left pressed and move Touch: **drag**	**Drag** object, depending on the drag zone identified when button has been pressed, it may be: • Chart move • Chart resize • Timeframe pan • Prices rescale if price axis under the mouse • 3D rotation (3D heatmaps) If no specific drag zone on charts is identified, then pan the view
Ctrl key +  +  Left pressed and move	Pan the view
Ctrl key +  +  Left pressed and move	**Drag a Copy** of the current chart(s), if dragging has been started on title or drag chart zone

CTRL + C	**Copy chart(s) selection** (in the Quantyl clipboard)
CTRL + V	**Paste chart(s) selection** (from the Quantyl clipboard)
Right click Touch: **slow tap**	**Popup a menu** depending on the object zone: • Title menu • Axis menu (X and Y axis may be independent) • Graph menu • Selection menu
+ Right pressed and move Touch: **pinch / spread with 2 fingers**	**Zoom window** depending on timeframe identified under the mouse: • Timeframe if identified, • Else zoom all charts area **Important:** • **Drag from left to right to zoom IN** • **Drag from right to left to zoom OUT** Zoom in (left to right):  Zoom out (right to left):

	 Zoom fit all: If you zoom out a little zone on the left, a specific cursor indicates you that the whole timeframe or view will be fitted: 
 +  Right pressed and move + **Ctrl key**	Zoom window on all charts **Important:** • **Drag from left to right to zoom IN** • **Drag from right to left to zoom OUT**
 Mouse wheel Touch: **pinch / spread with 2 fingers**	**Zoom / Unzoom** depending on timeframe identified under the mouse: • Timeframe if identified • Else zoom / unzoom all charts (the virtual charts desktop) Important: by default, zoom is directional, the position of the cursor is considered as center of the zoom. **If you want to keep the last time displayed unchanged, you have to enable this option in selection manager:**

	
 Mouse wheel + **Ctrl key**	**Zoom / Unzoom** all charts area (the virtual charts desktop)
 Double right click on the viewport (no charts under the cursor)	**Refit all charts to the view**
 **Arrow** keys	If charts are not fitted to their parent, they will be fitted. If charts are already fitted, they will be "**Rolled**" in a tiling pattern depending on the key pressed: • **Roll** left • **Roll** right • **Roll** up • **Roll** down This command takes care of the charts currently selected to be able to tile and roll those charts inside groups. If no charts are selected, then root charts are tiled and rolled into the view
 **Space** bar	Fit global timeframe
 **S** key Or  **T** key	Synchronize / Desynchronize charts timeframes. 
 **Tab** key	When multiple charts are under the mouse, pressing tab key **roll the them front to back** changing they **layer** property



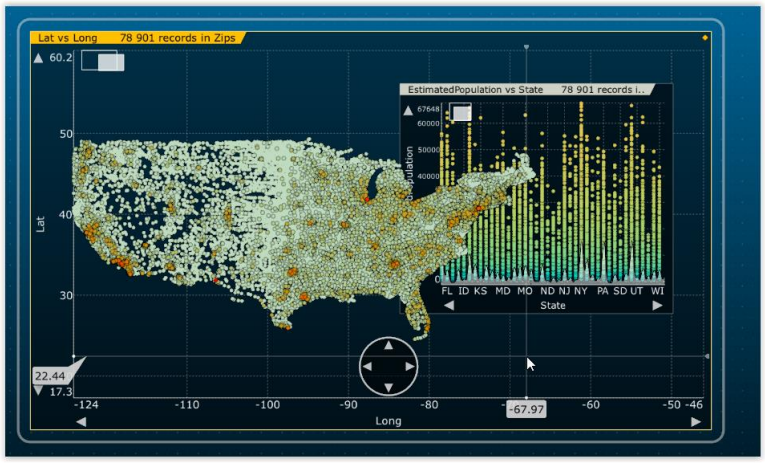
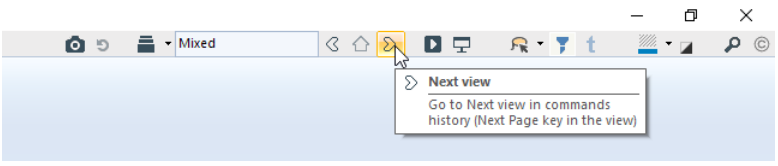


Figure 4 Layering, Bring to front / Send to back commands

When you drag an object on top of another it is automatically drawn in front of others. If it not the desired result you want, it is possible to draw object in its parent background, using the “Send to back” command.

 **Suppr** key Or **Delete** key	Remove selected charts
	Abort current task if any Unselect all objects
	Go to next or previous view 

3.3 SAMPLES

Quantyl is installed with multiples samples, that you may explore, copy and modify to experiment some of the capabilities of the product on different use cases. A large part of the screenshots in this document comes from those samples. Do not forget to explore the different views when you open a Workspace!

Important

Most samples are geo-localized data oriented to facilitate data / features understanding.
Quantyl is **not** geo-data specific.

Those samples are standard workspaces in read-only. Double-click on drag & drop the workspace in the viewport to load it:

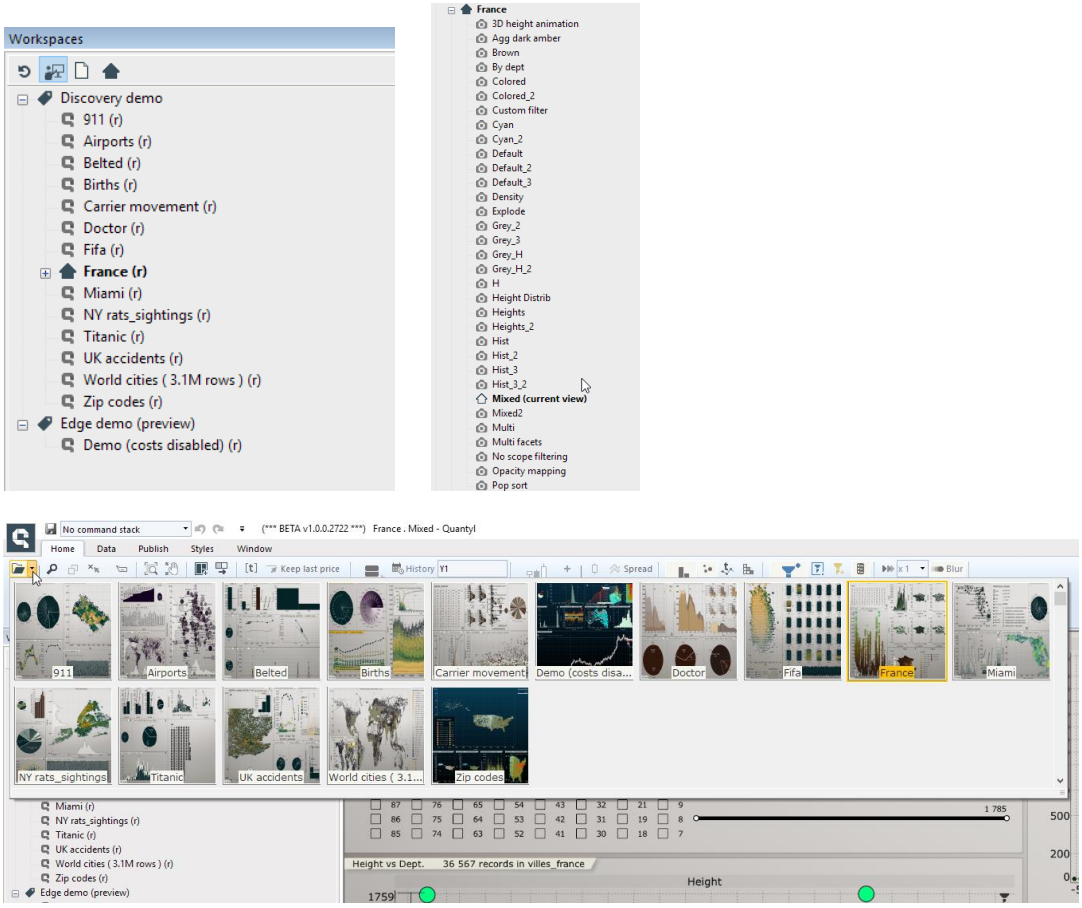
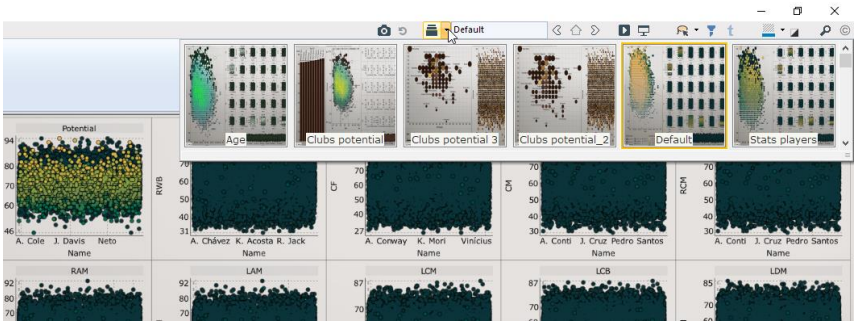


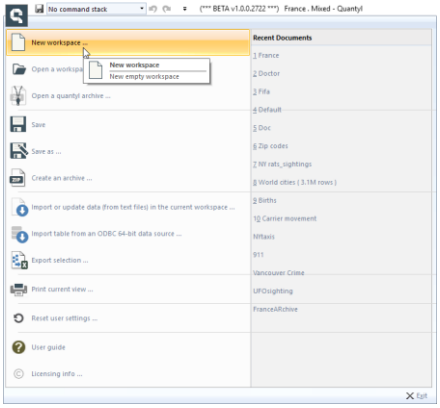
Figure 5 Demo workspaces and workspace´s views

Explore the different views of the current workspace:

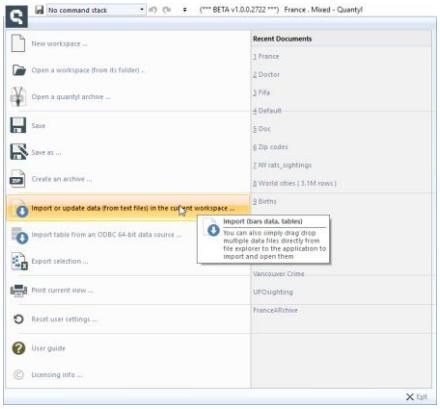


3.4 QUICK START SCENARIO

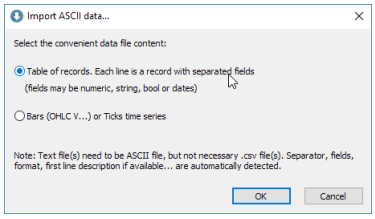
Create a new Workspace (by default, Quantyl opens the last one)

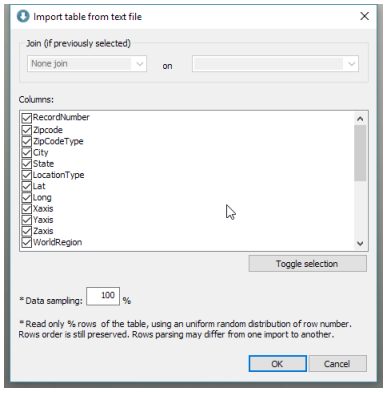


Drag & drop a .csv file(s) into the application or use File/Import command

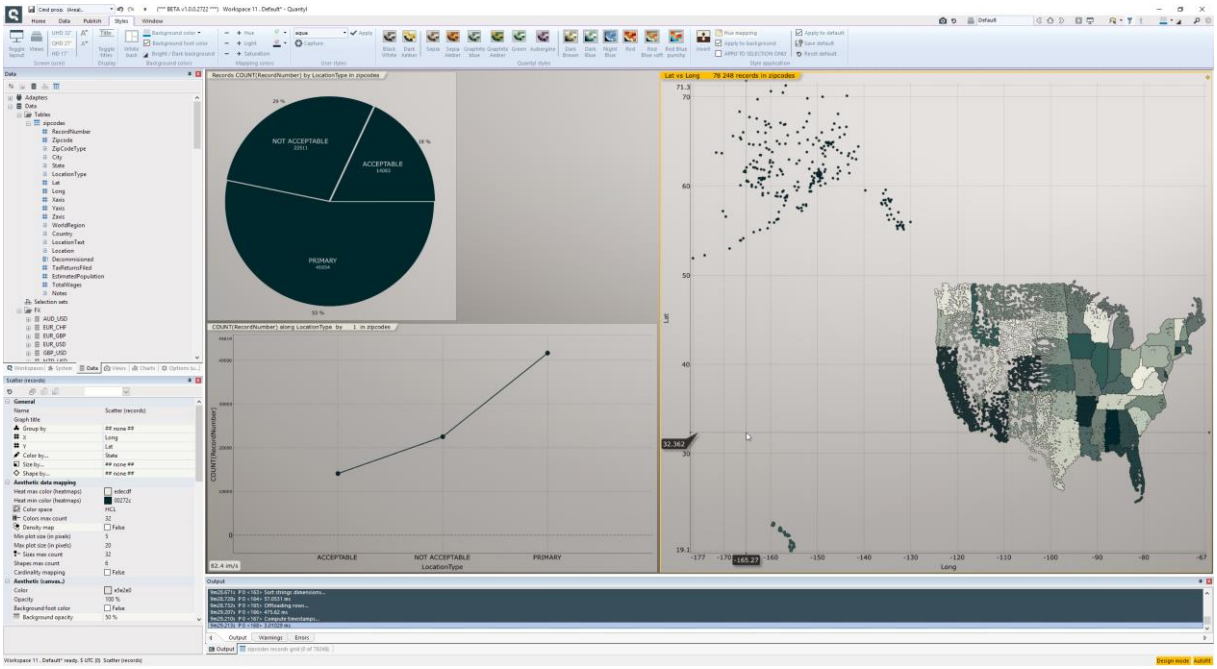


Choose data columns you want to import

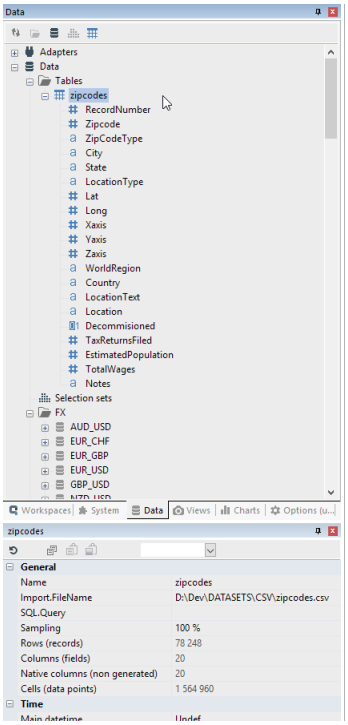


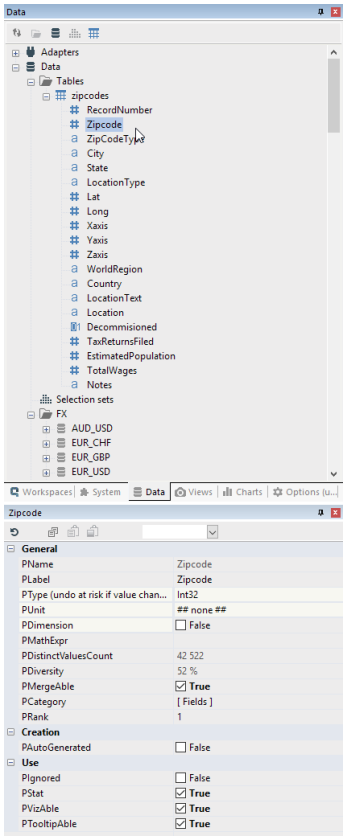


Ready for your data discovery journey? Some pertinent charts are created by default

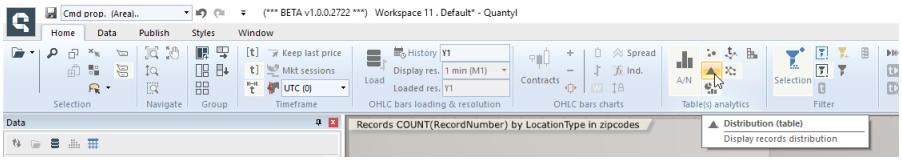


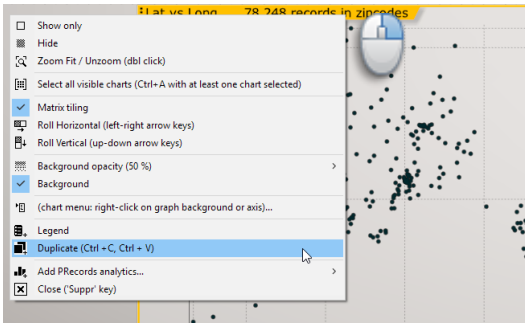
Check table and fields



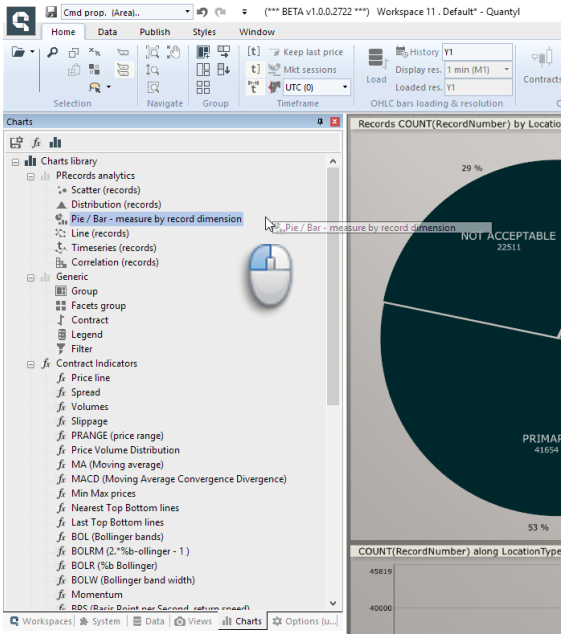


Add new chart or duplicate existing ones with chart contextual menu

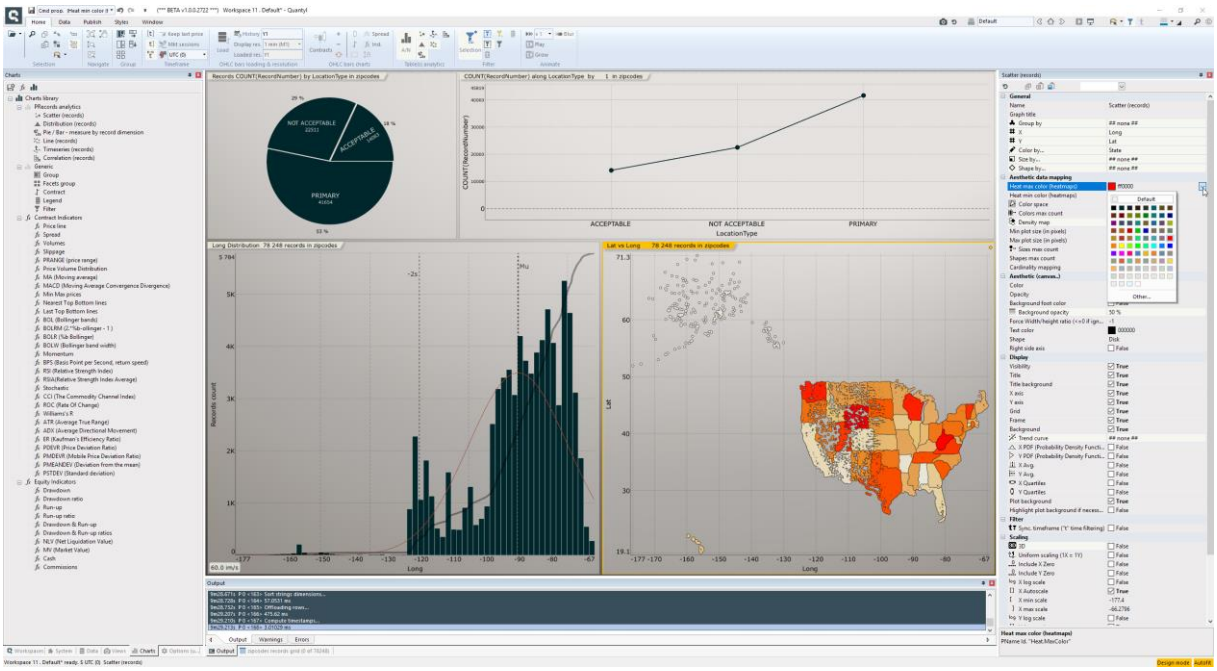


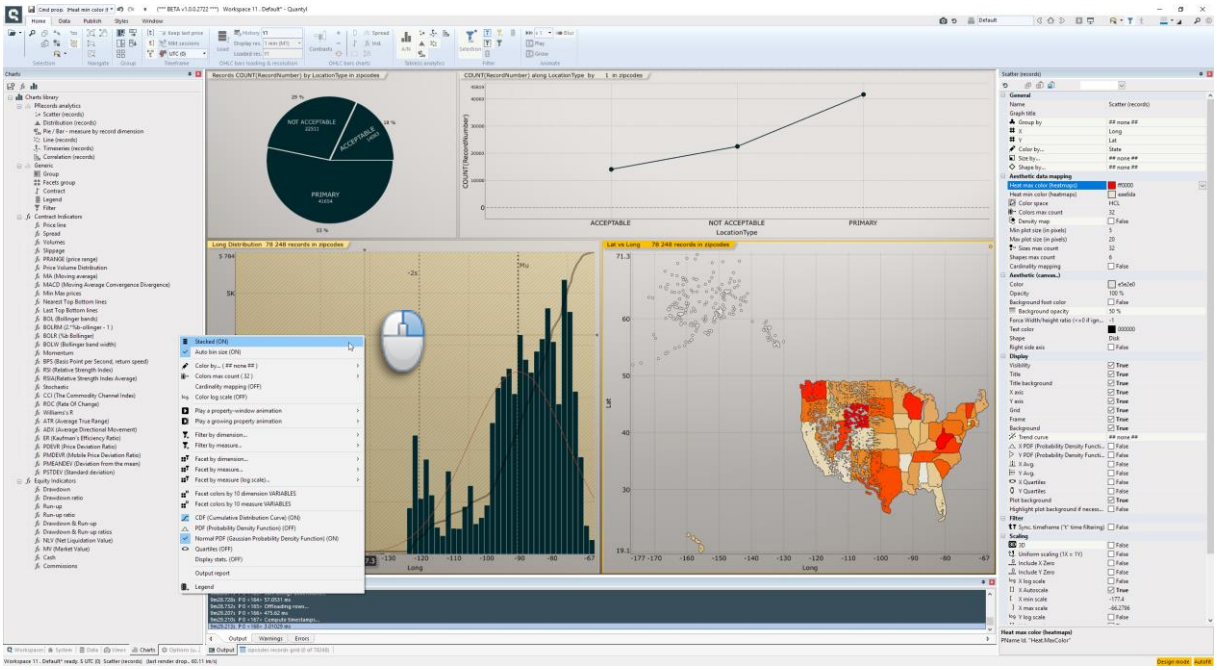


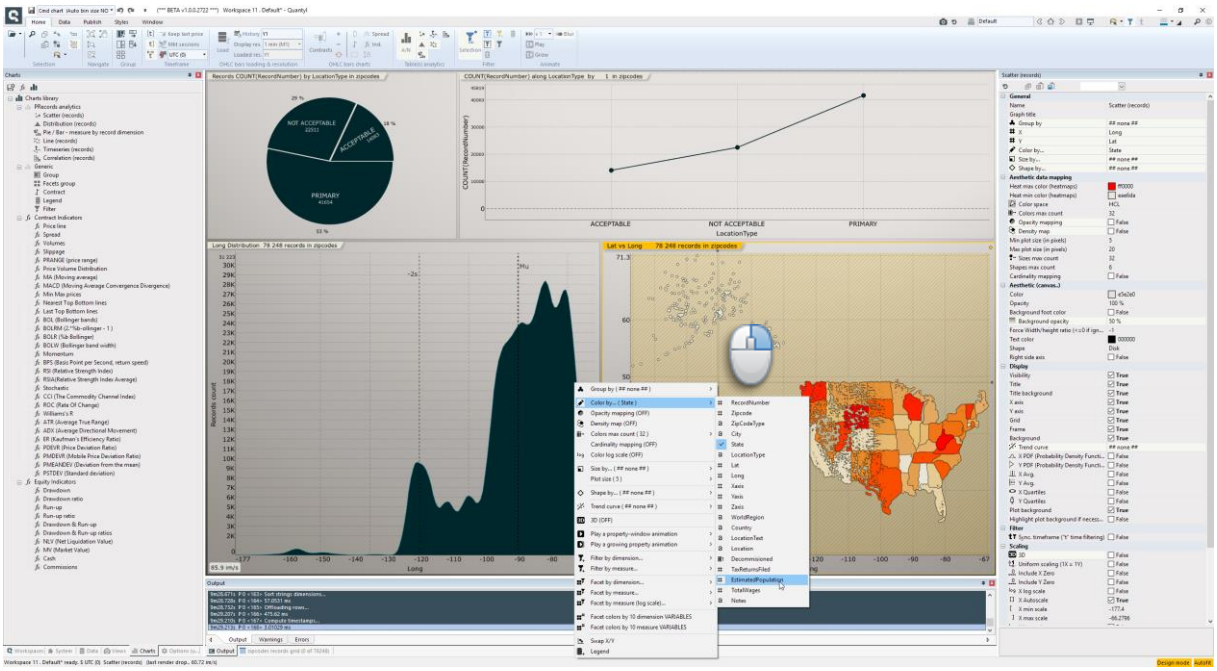
Drag charts from chart library into the viewport

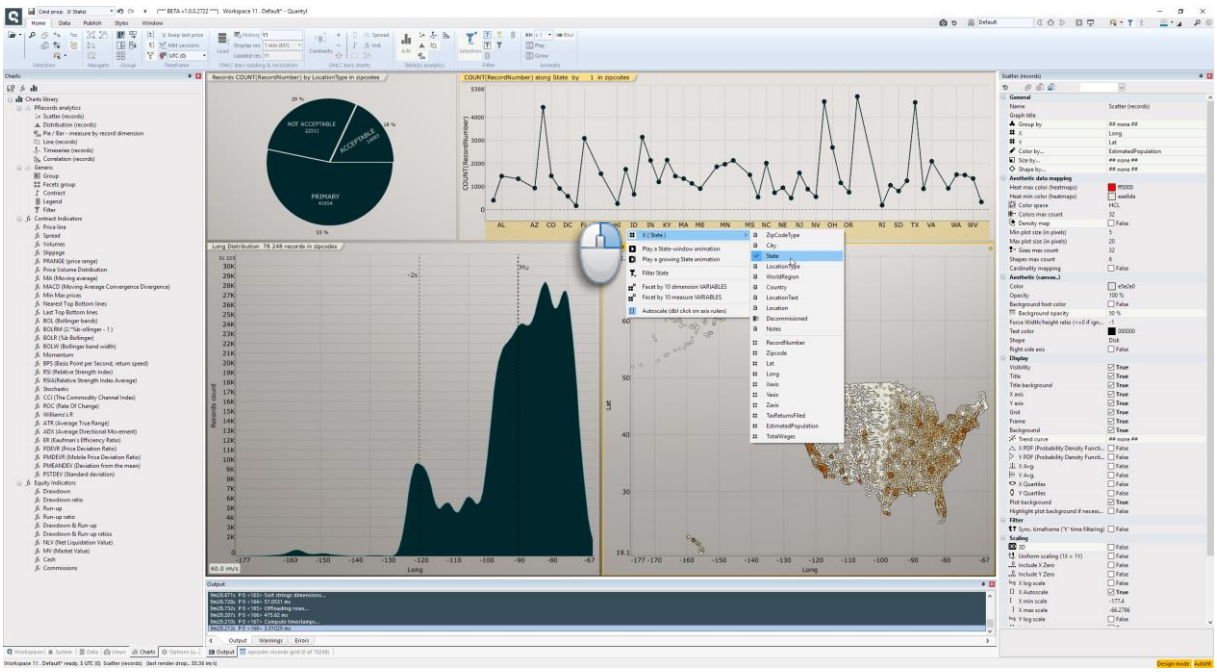


Modify charts properties from the property page or from the different chart menus (title, axis rulers, graph area...)

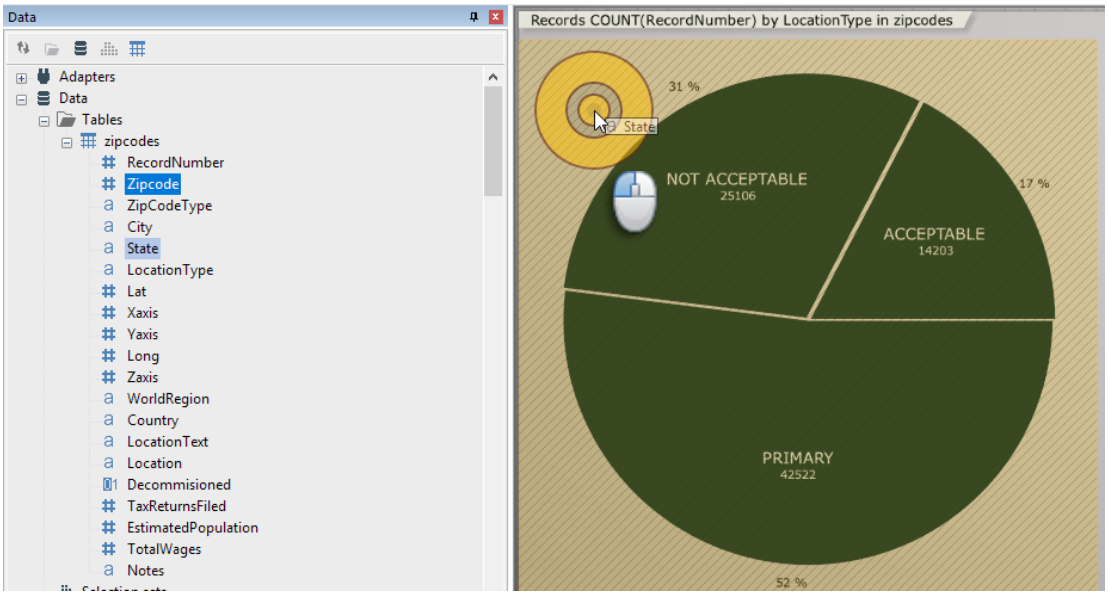




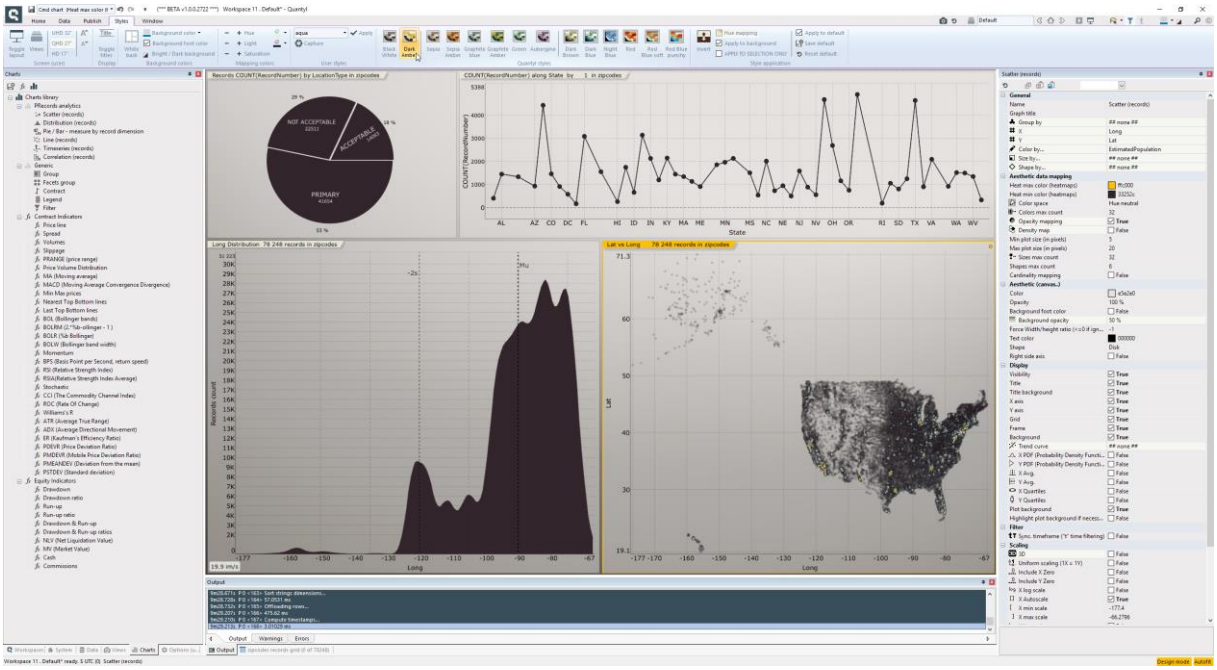




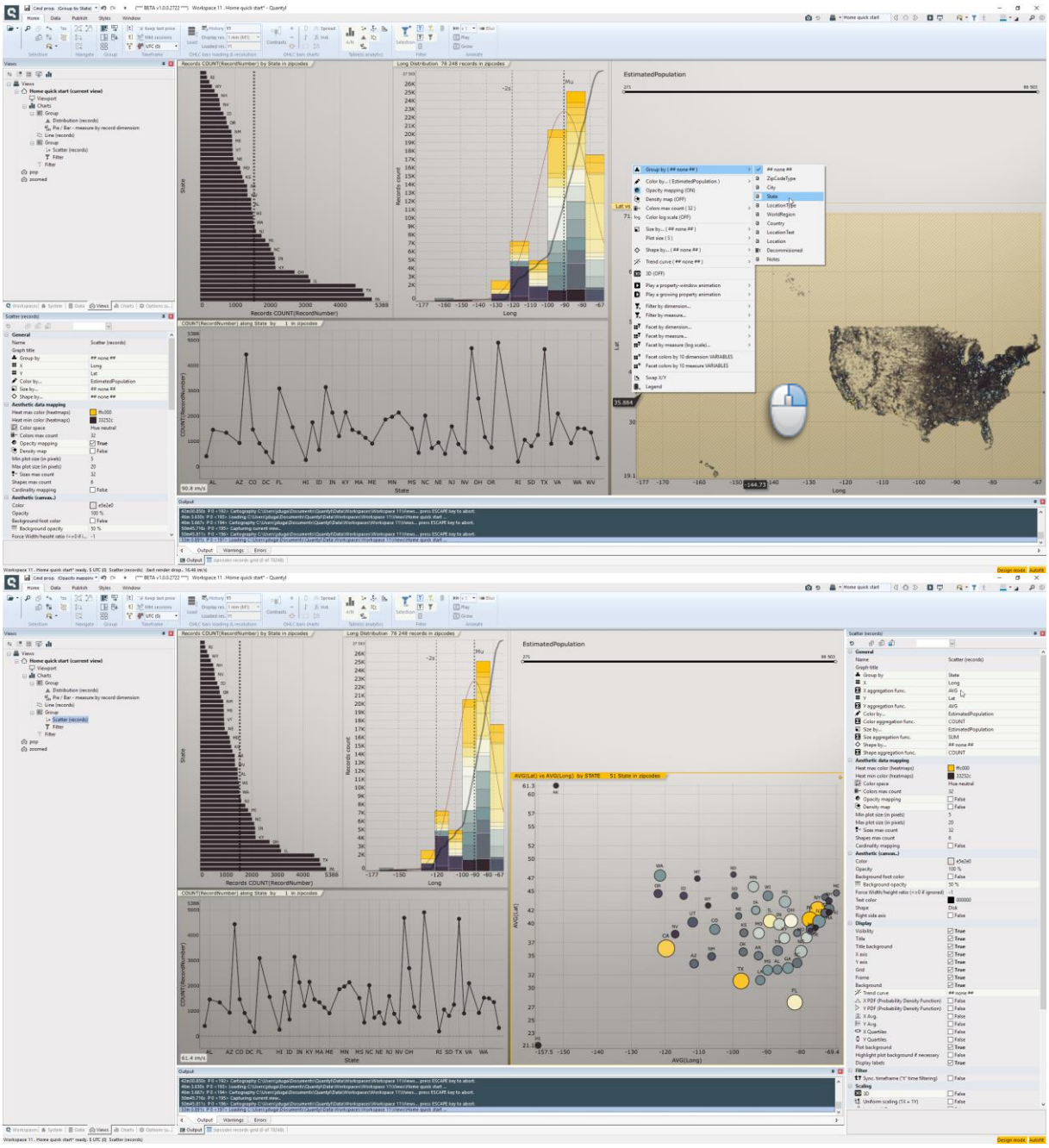
You may also drag directly dimensions and measures from table tree to charts and axis



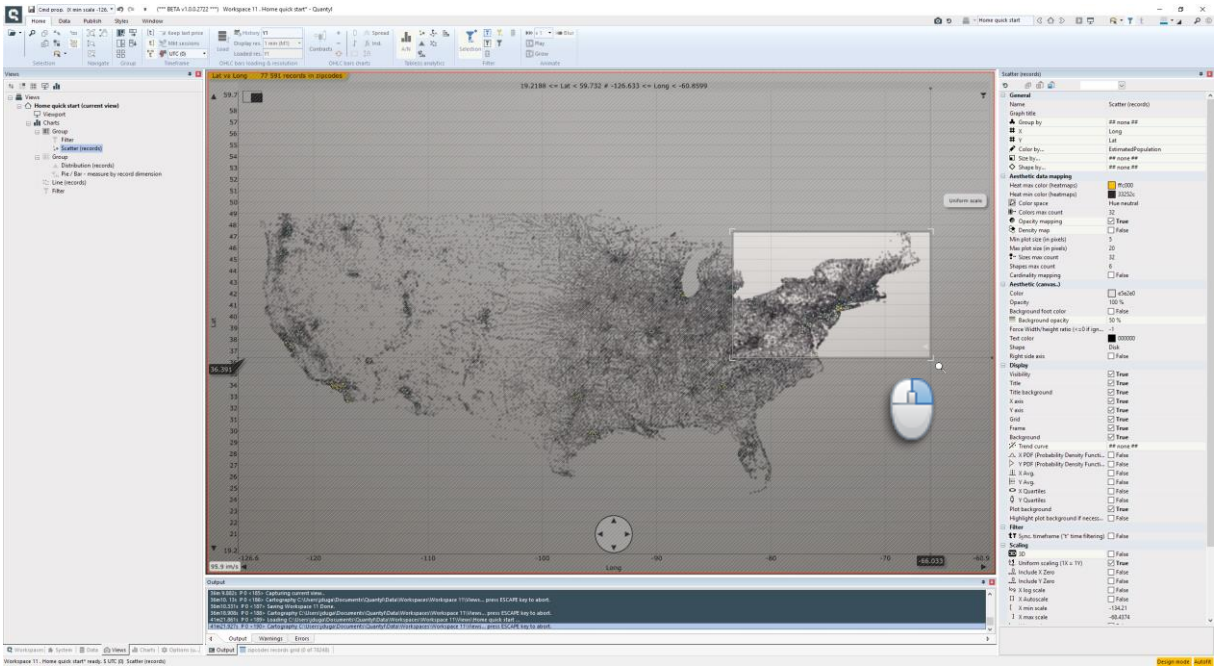
Try different pre-defined styles through the style ribbon category or build your own one

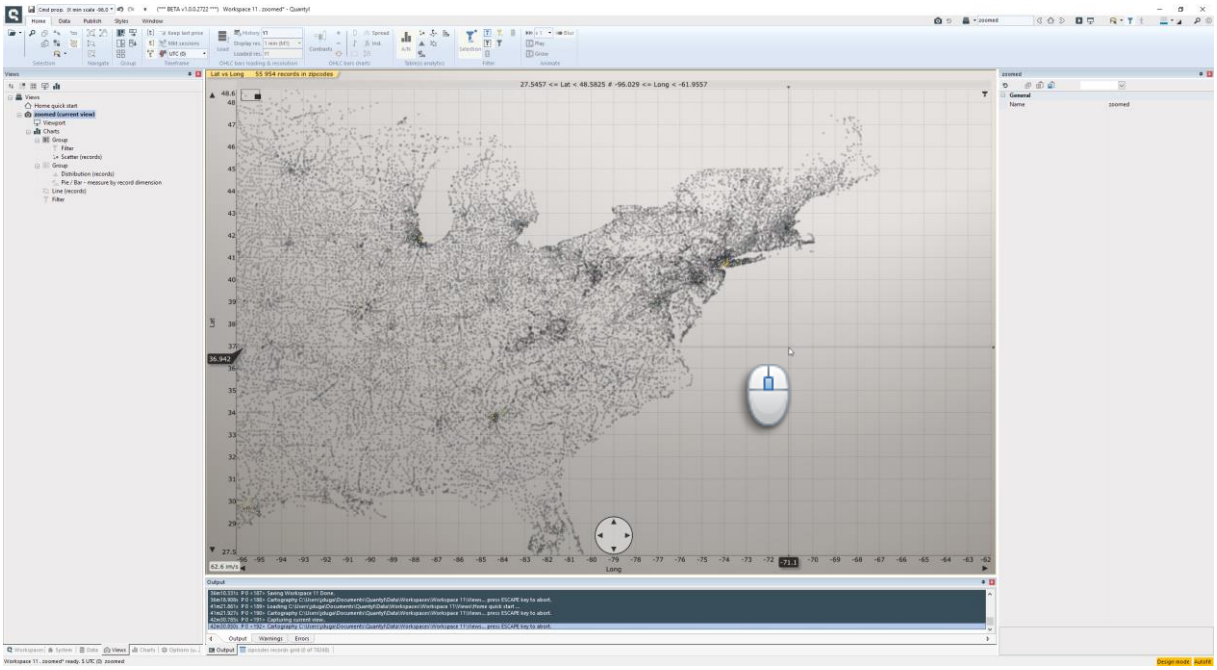


Aggregate data if you need

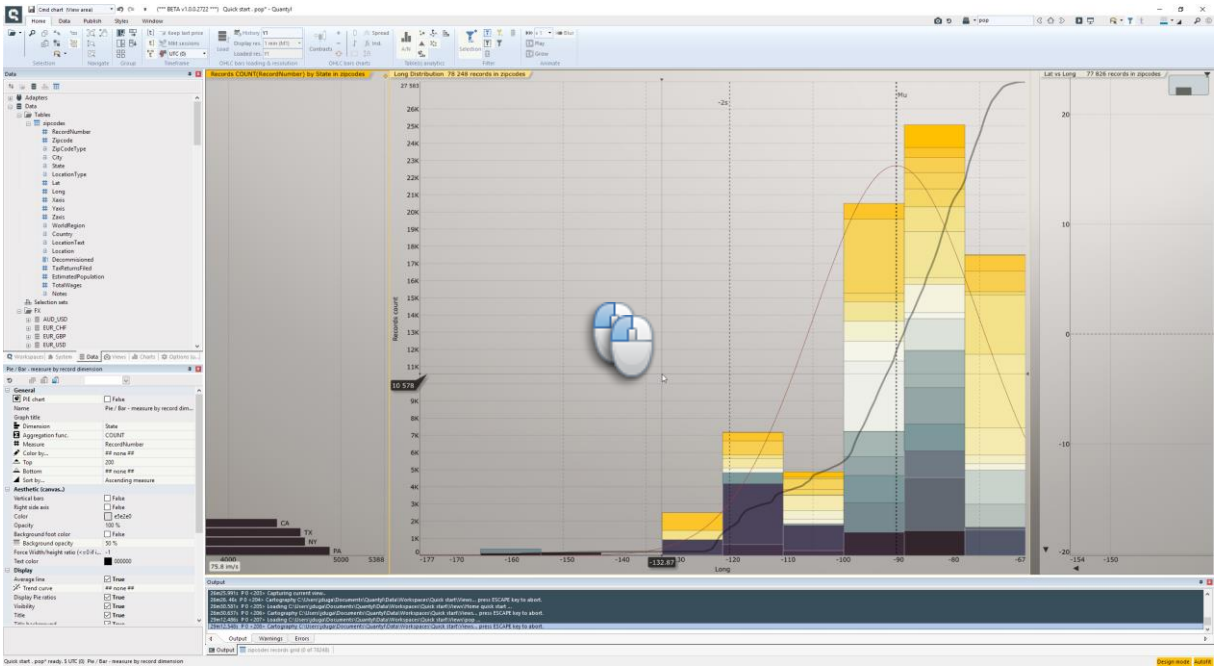


Scale graph as you need by using mouse wheel and mouse right drag

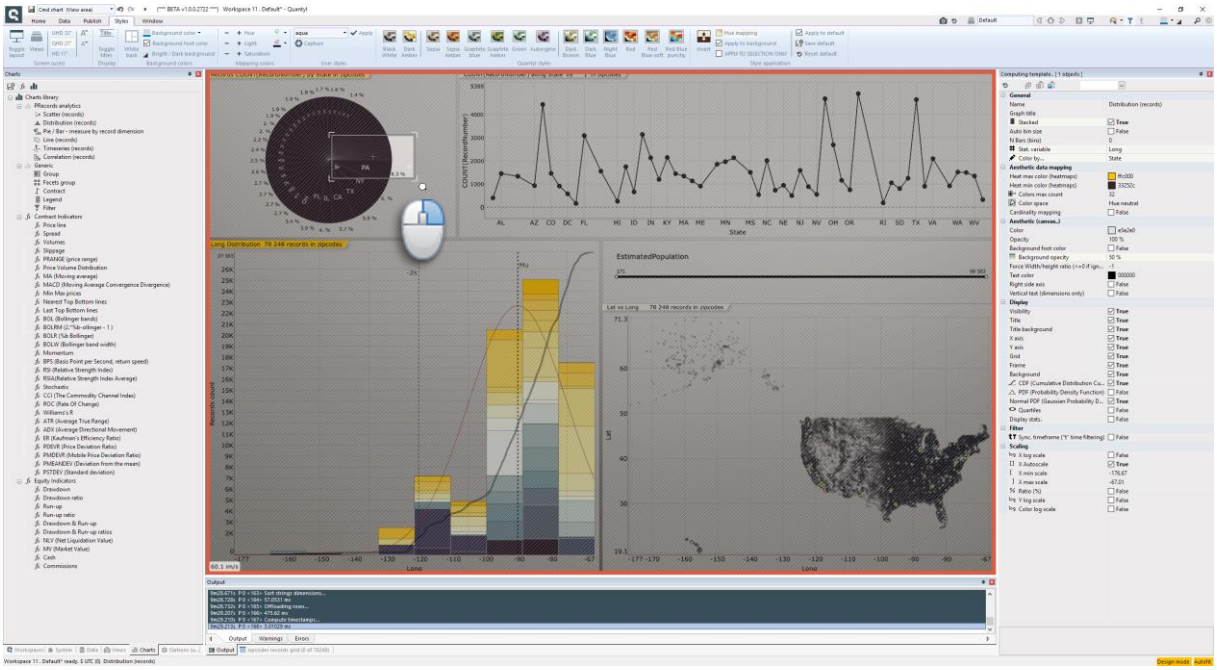


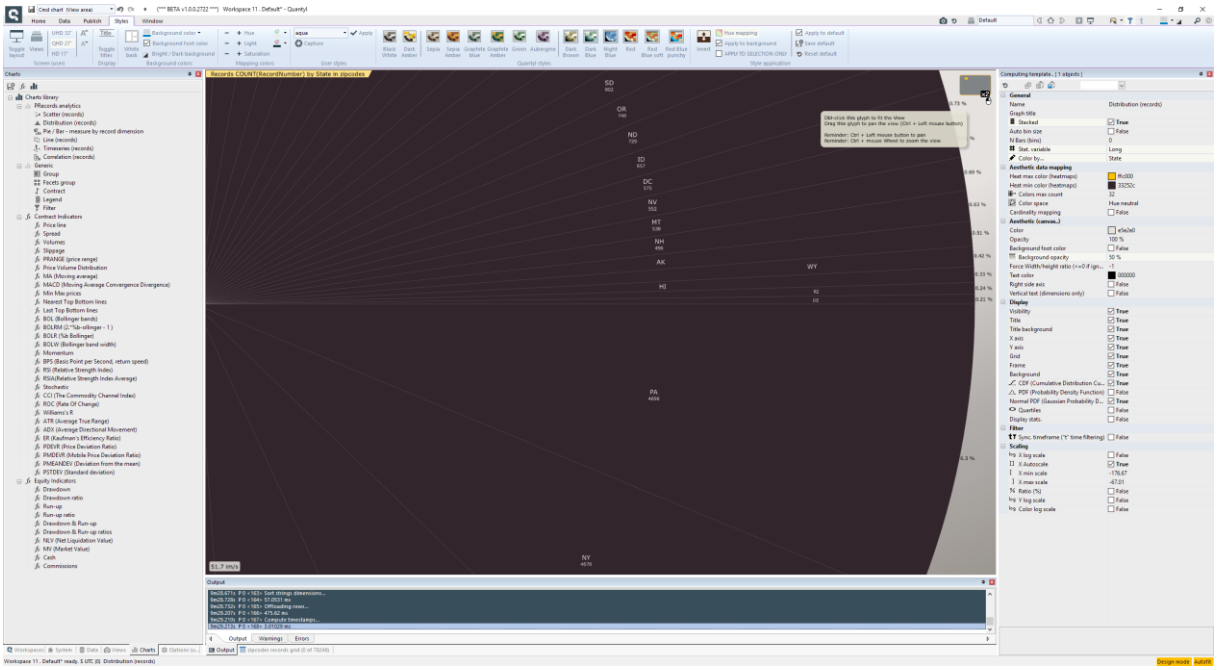


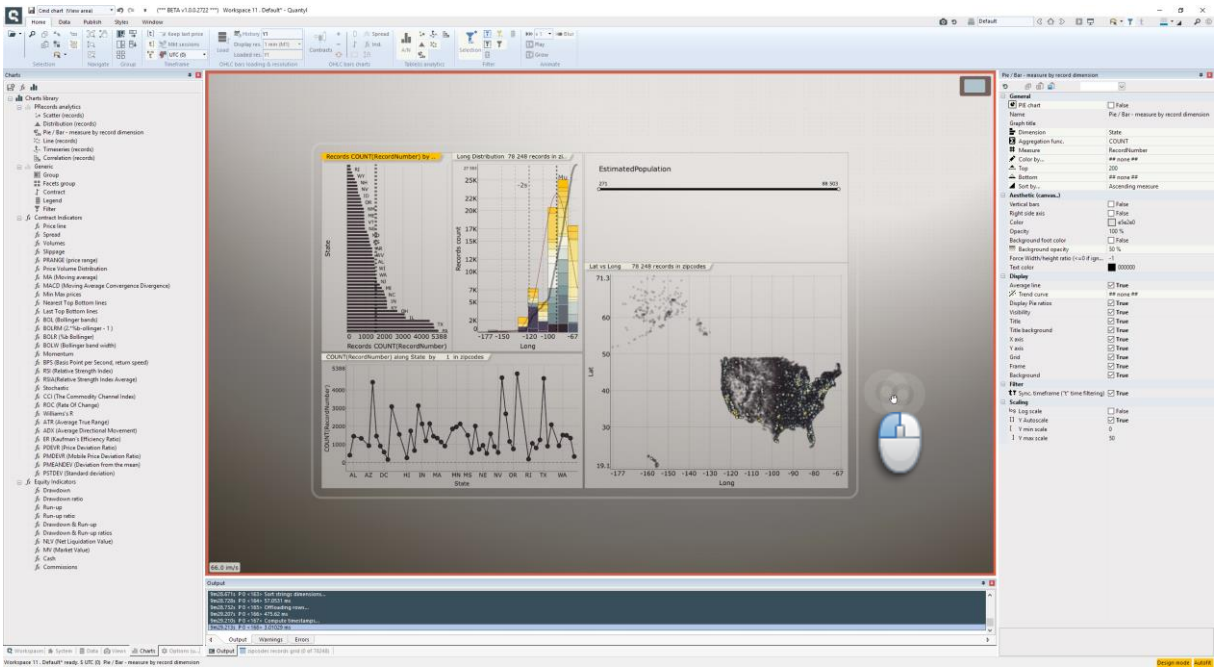
Double click on any chart to zoom in / out



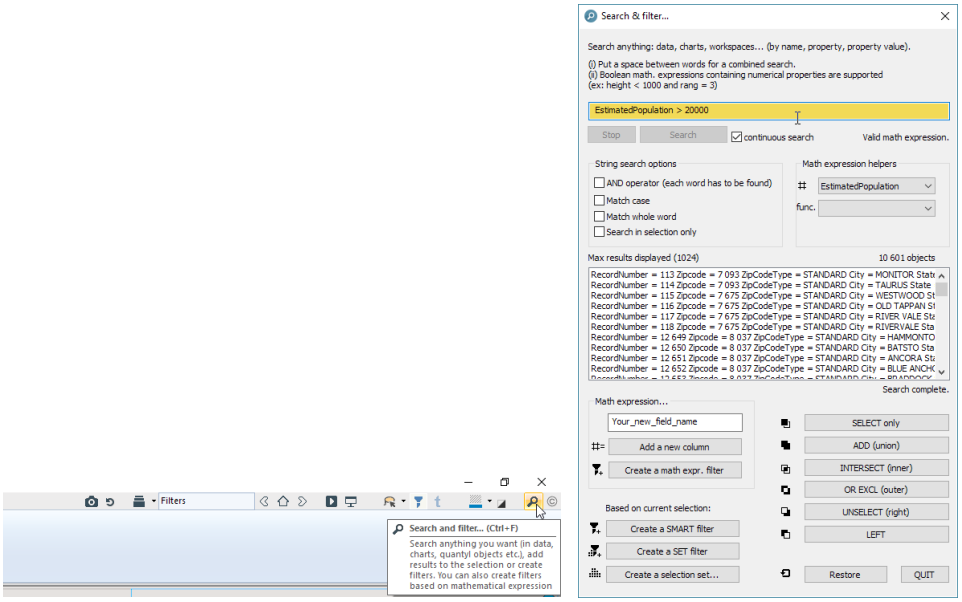
Navigate: use Ctrl + mouse wheel and right mouse click window to zoom-in / zoom-out inside charts



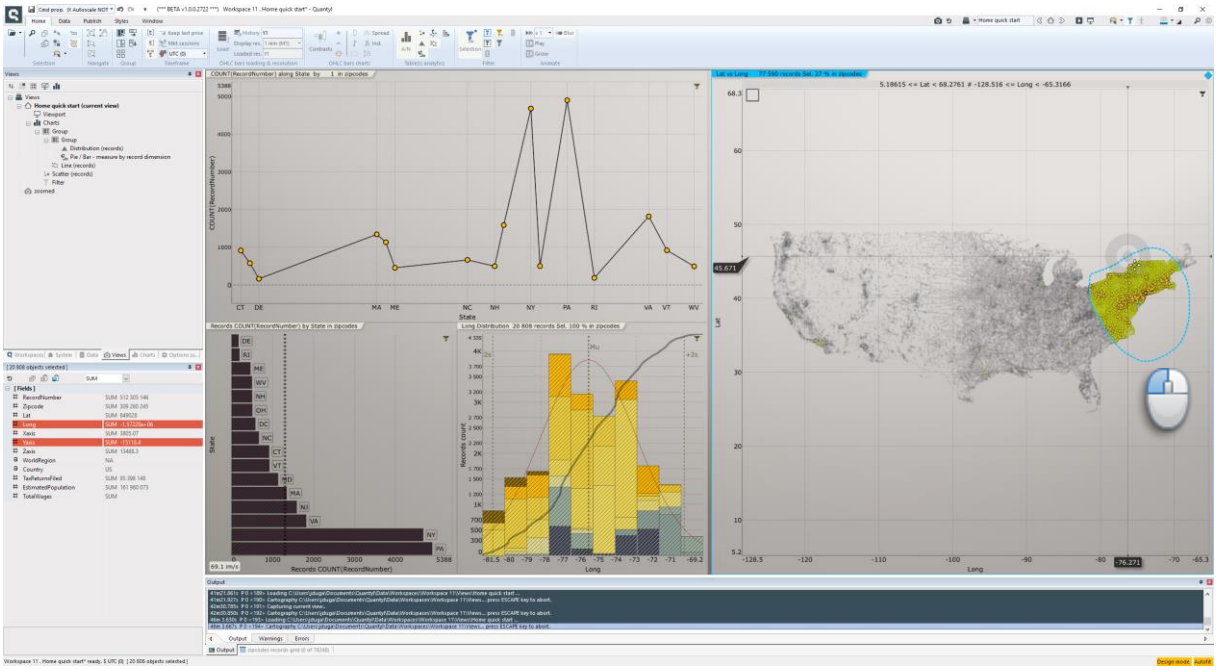




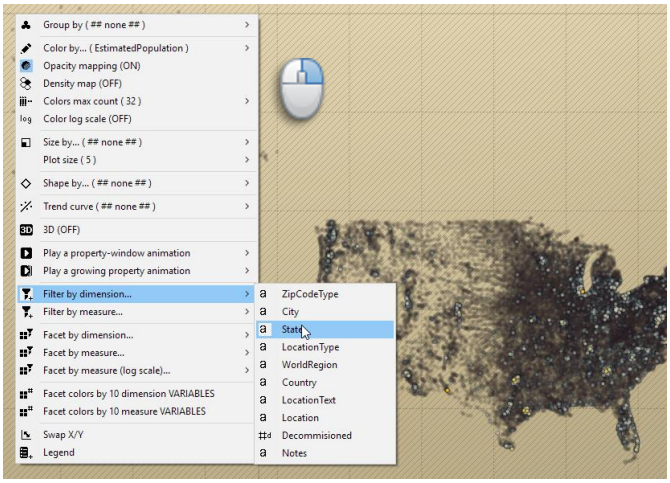
Search, select and filter specific data through powerful search and formulas

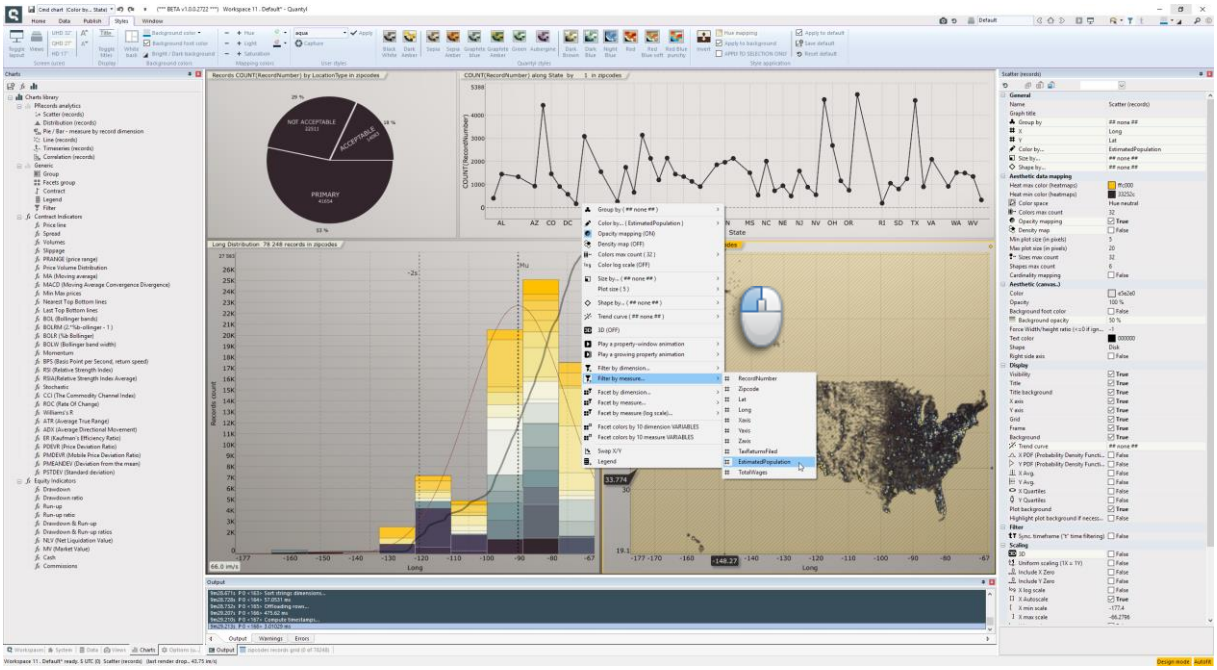


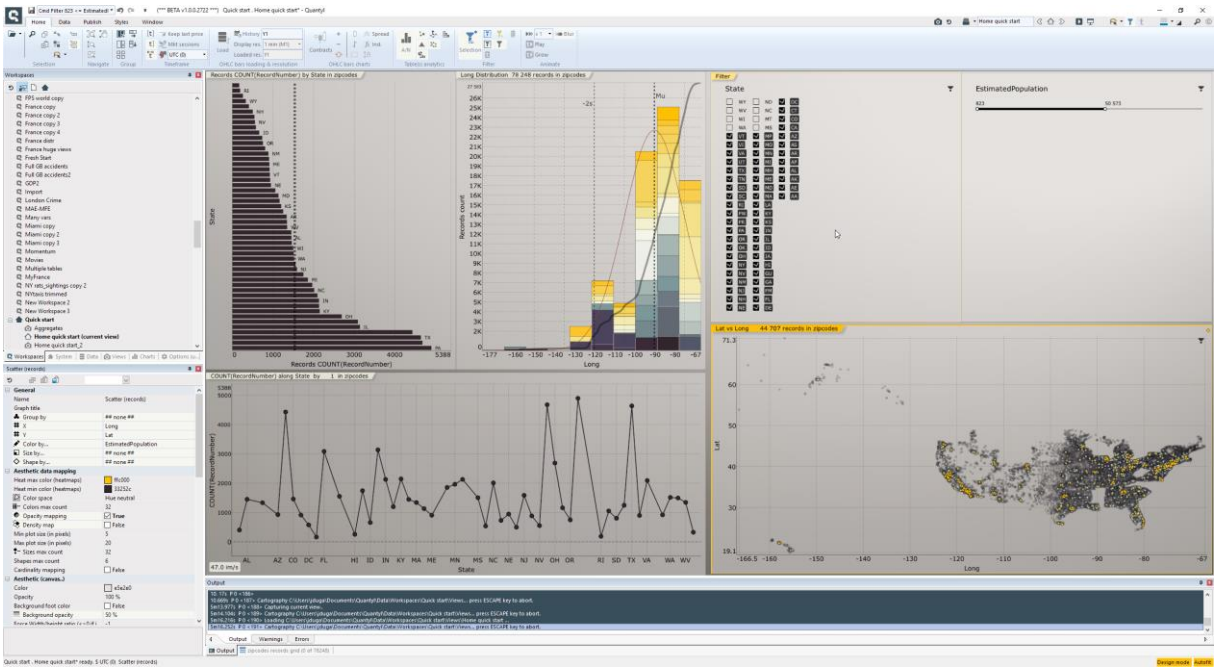
Filter dynamically through selection, try to drag the selection contour (selection brushing)



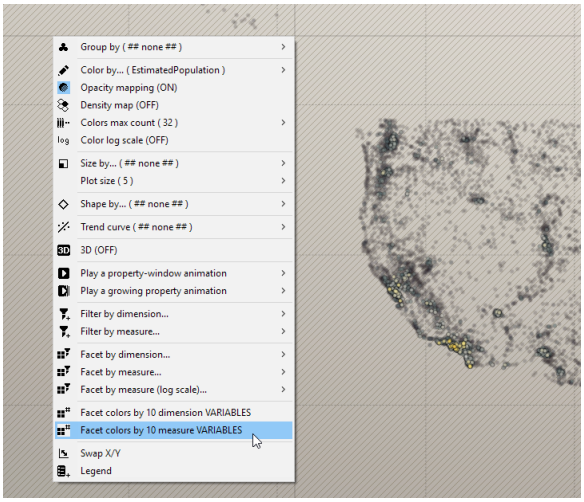
Create static filters, directly from the charts

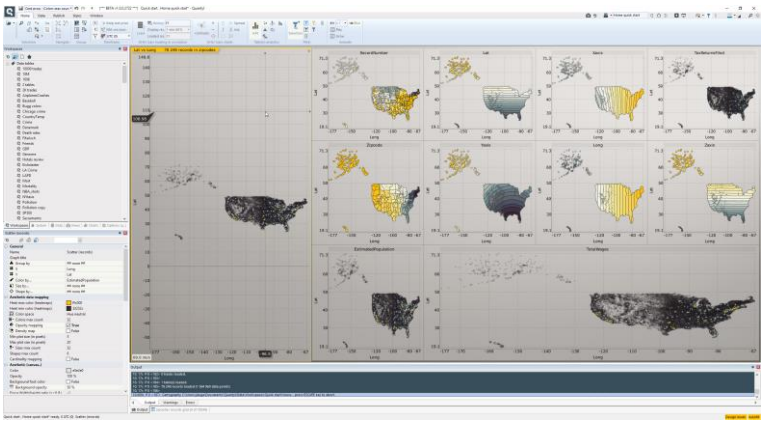




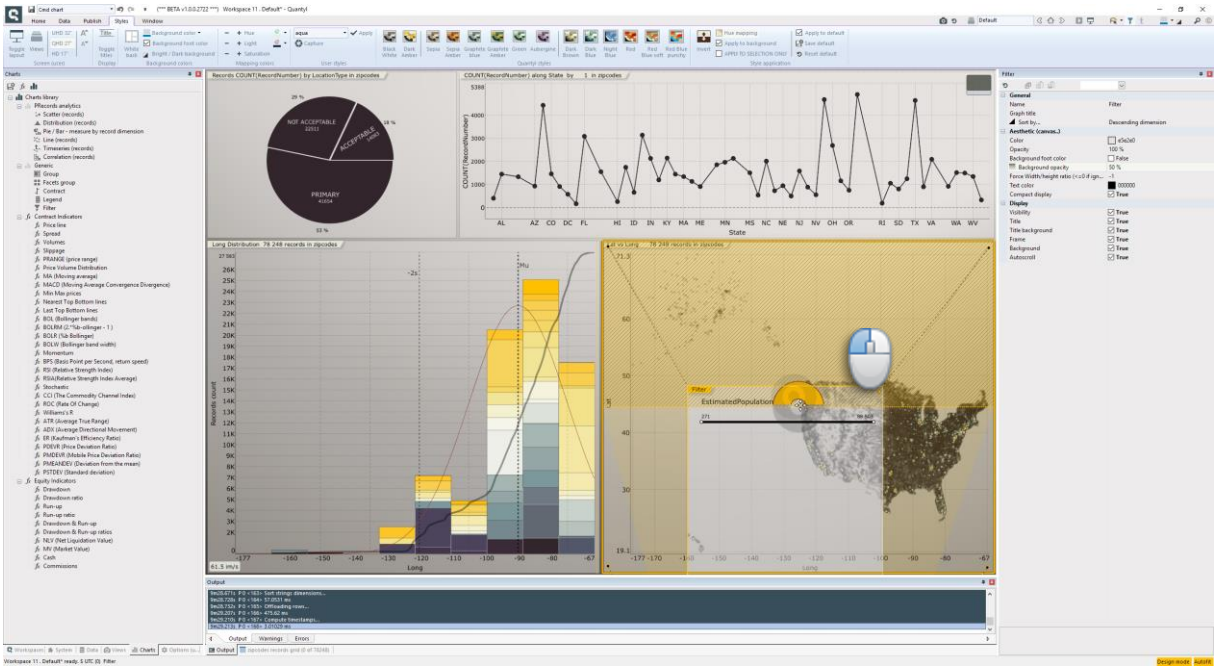


Use faceting command to display variable(s) facets

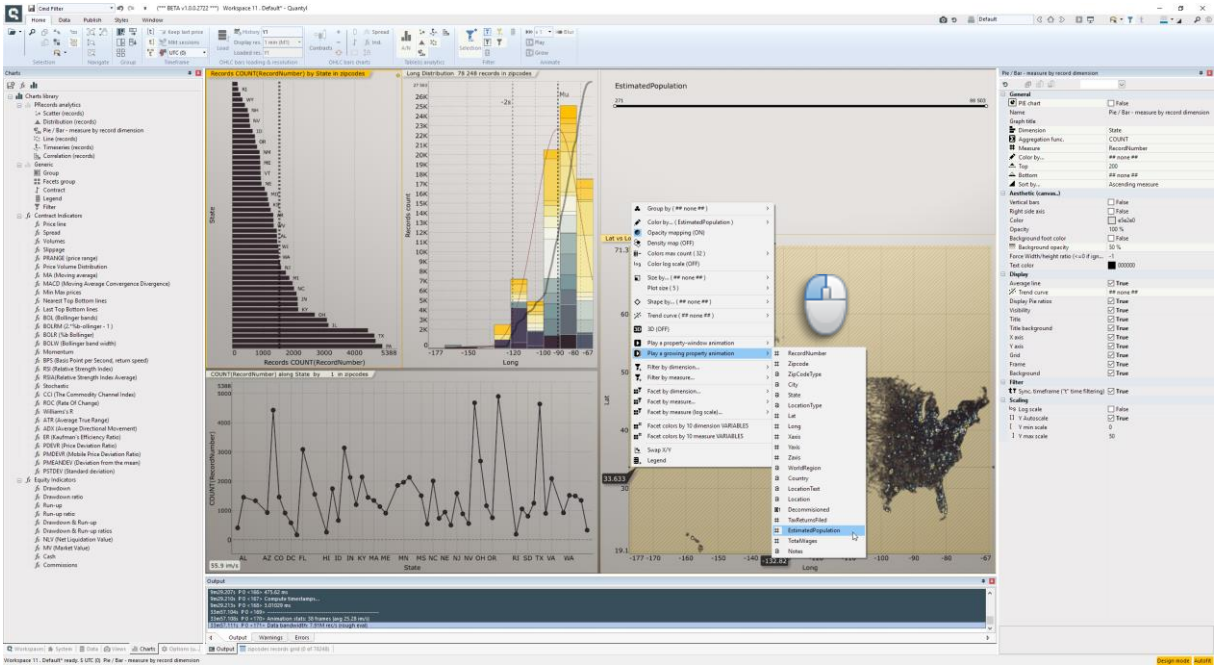




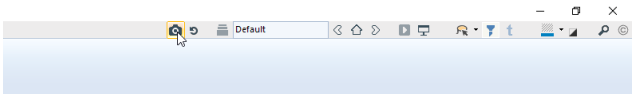
Group/ungroup charts using their handles

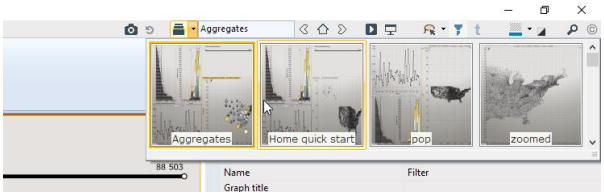


Animate

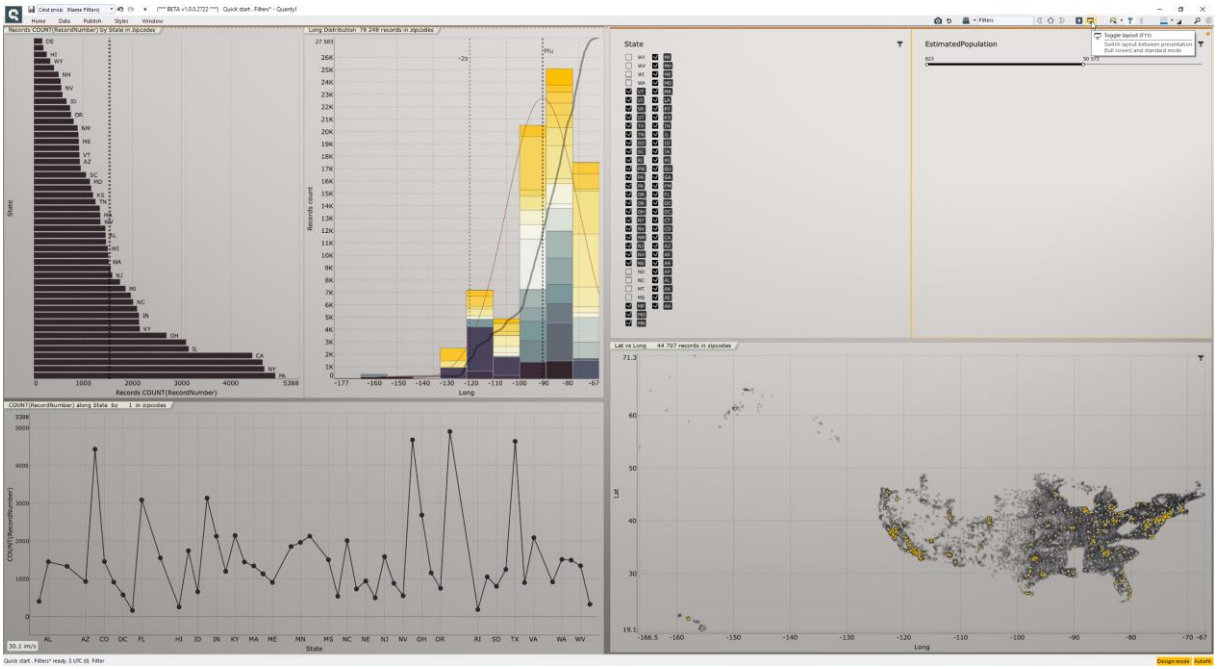


Create as many views as you wish, each view captures your current chart configuration

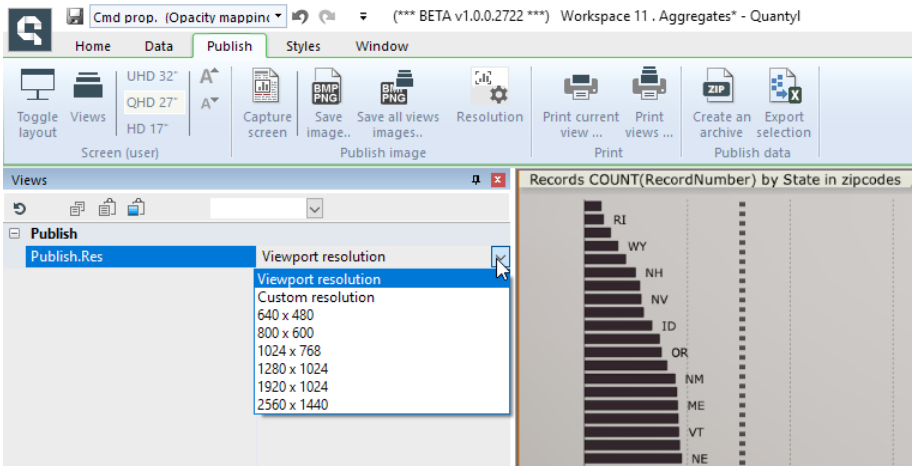




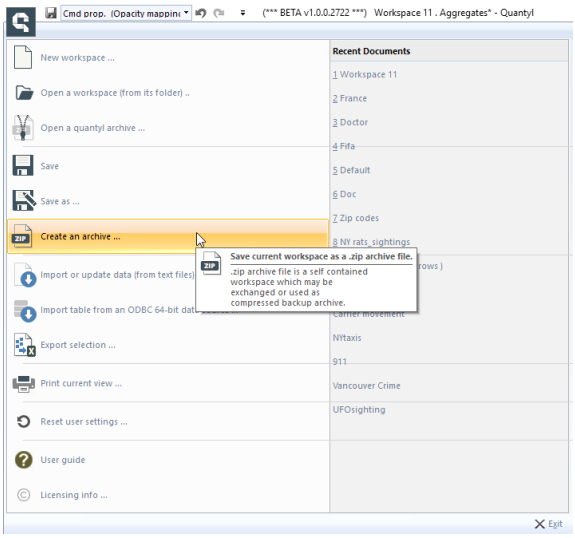
Switch to presentation mode and navigate or play the views



Publish some images in .png format



Save the workspace or create an archive to share your workspace



4 FUNDAMENTALS

User should be familiar with the following key data visualization concepts (most of them are not Quantyl specific).

4.1 SELECTION

Powerful selection mechanisms are central to any data manipulation and data transformation process. Quantyl user interface is based upon selections: selection of charts, data, managers etc. Selection may be simple or multiple, implicit or explicit.

4.2 PROPERTIES

A property is a value associated to a string identifier. Property also owns different attributes (is it a dimension or a measure? what is the unit? etc.). Properties of the selection may be displayed and modified, depending on the context.

A chart owns properties, a table owns properties, a table field owns properties, a record (row) owns properties etc.… In Quantyl, almost any selectable object exposes properties. There are more than a thousand available properties, only on Quantyl objects (charts, options etc.).

When you select multiple objects, their properties are **aggregated** (when possible), to be displayed as a whole.

Properties are at the core of the Quantyl data model, and they can be extended. The common types of property are:

- `BOOL`
- `STRING`
- `FLOAT64`
- `FLOAT32`
- `INT32`
- `INT64 //!< Datetime`

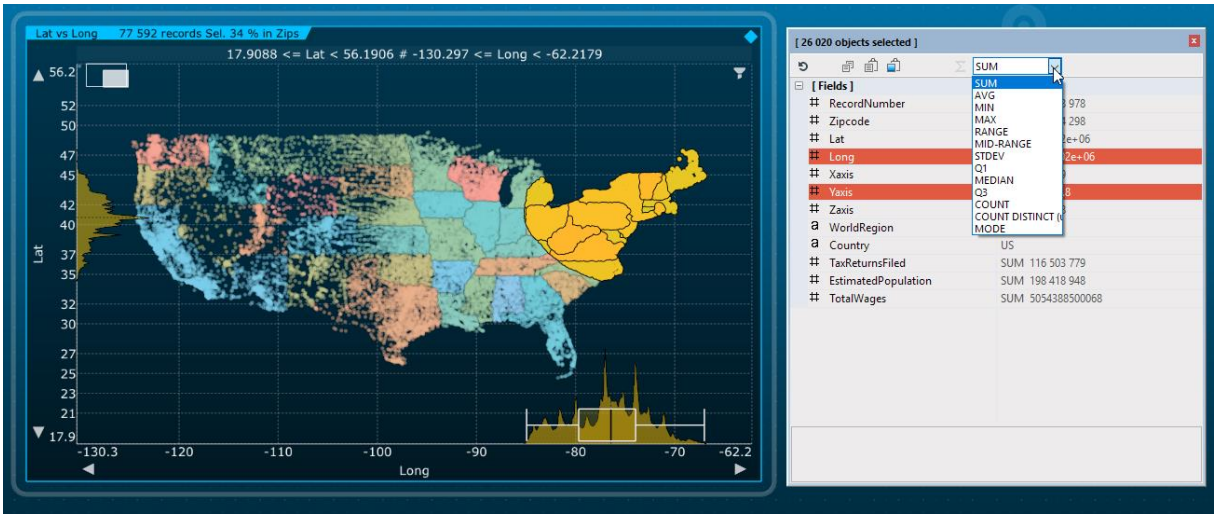


Figure 6 Selection properties

4.3 DIMENSIONS AND MEASURES

Any property is defined as a Measure or a Dimension.

- **Dimensions** values cannot be aggregated (like a sum): symbol name, date, month, DOW, any tag category etc. Dimensions are generally presented as categories (string), but numerical values may also be dimensions like rank number
- **Measure** are numerical values that can be aggregated; they may also be aggregated with a function (sum, avg, min, max, stdev..)

Dimension and measure choices are generally available in the contextual menu of the different analysis chart axis. Quantyl is very flexible, and generally you can choose either dimensions or measures as parameters of a chart, and the graph of the chart will adapt automatically to the nature of the choices you have made.

4.4 TABLE, RECORDS AND FIELDS

A table is a collection of data rows where each row has the same number of fields (aka columns).

Each field is represented by a property value, which can be a dimension or a measure.

In Quantyl, each row is called a “record”. Records are objects that you can select, visualize and aggregate. When you select or highlight a record, its properties are displayed (in tooltip, property page, grids...). If multiple records are selected, you can display the sum, the average etc. of their different measures for example.

At the start of any study, you have to acquire data. Data may be collected as Tables coming from text files or data bases.

Those records and their relative properties are at the core of data discovery: Filter, aggregate, visualize, interact, animate… all those processes are working on records and properties.

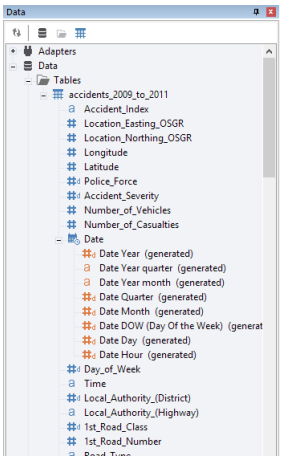


Figure 7 Table: dimensions and measures

4.5 AGGREGATING (GROUP BY...)

When data are aggregated, groups of records are replaced with summary statistics based on measures that are combined with an aggregation function. Aggregation is often performed on a dimension criterion: Group by state, Group by year etc. Some charts (pie, bar) are working on aggregates only, they need at least a dimension to be able to work.

Aggregating commands are contextual. Scatter plot chart and bubble chart are the same chart, the difference is that you have used the "**group by**" command to get a "bubble" chart display, like in screen capture below. When records are grouped, new commands appear to choose an **aggregation function** (for each axis, color, size etc.). You can group by multiple dimensions (by Year / by Type for example).

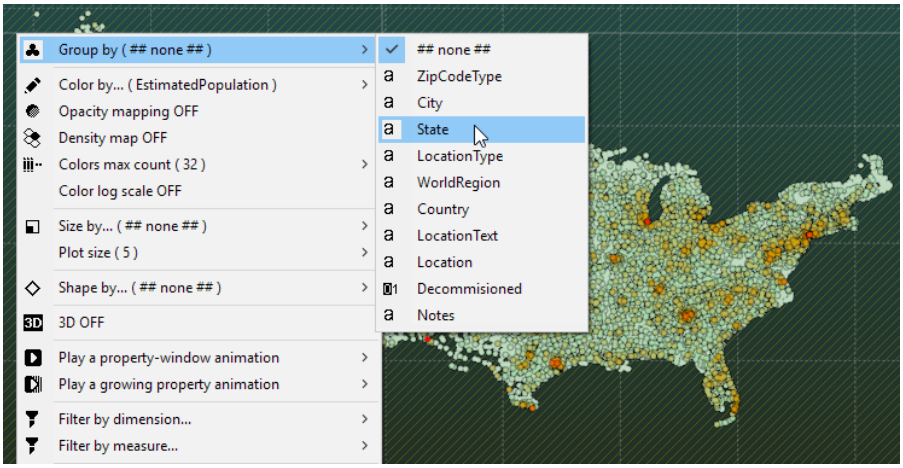


Figure 8 Group by command

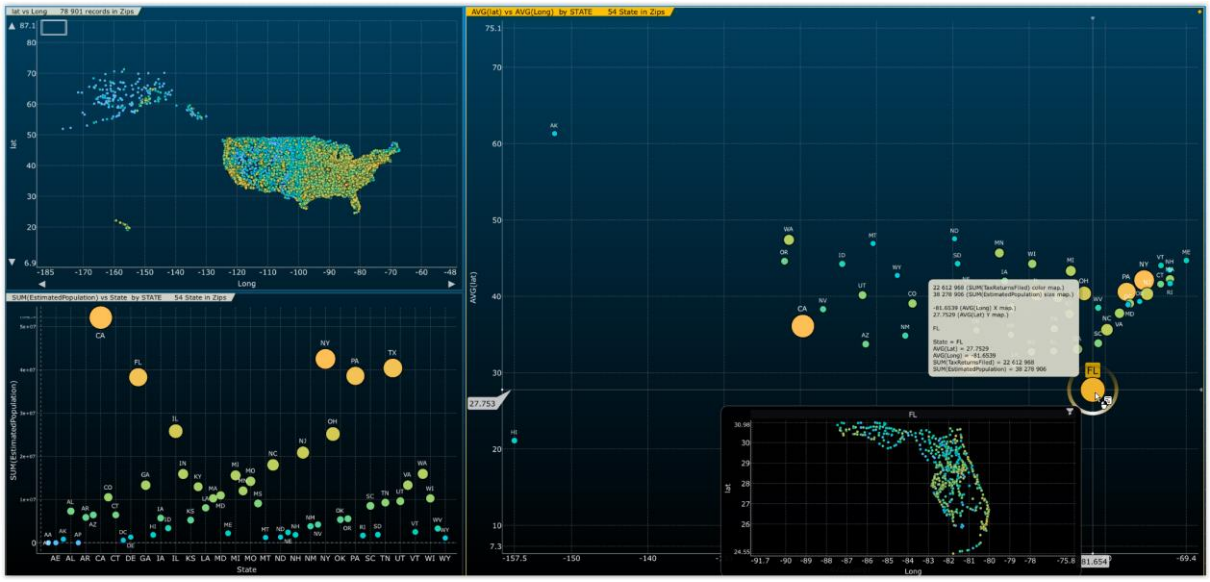


Figure 9 Example of aggregation result: a bubble chart (Group by State)

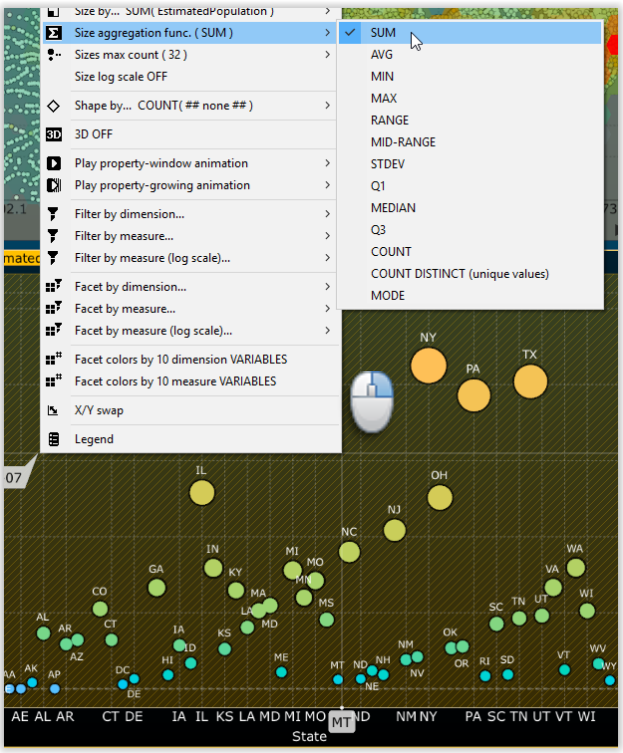


Figure 10 Aggregation function

4.6 FILTERING

Filtering is any process that generates a subset of a records collection. You can filter records by measure or dimension values, by interactive selection in a chart (**cross filtering**) or define a mathematical expression to combine different properties in a Boolean result (to keep or not the record).

By default, a filter is based on an **include** criteria. But in some cases, it may be more convenient to define it as an **exclude** criteria.

4.7 CHARTS AND GRAPHS

In Quantyl, a chart is a rectangular window which may contain different graphical information. Depending of the properties of the chart, the graph(s) may change.

We make a difference between chart and graph:

- A **graph** is always associated to a chart, it is graphically represented in a sub-area of the chart: the graph area
- A **chart** may have multiple graphs

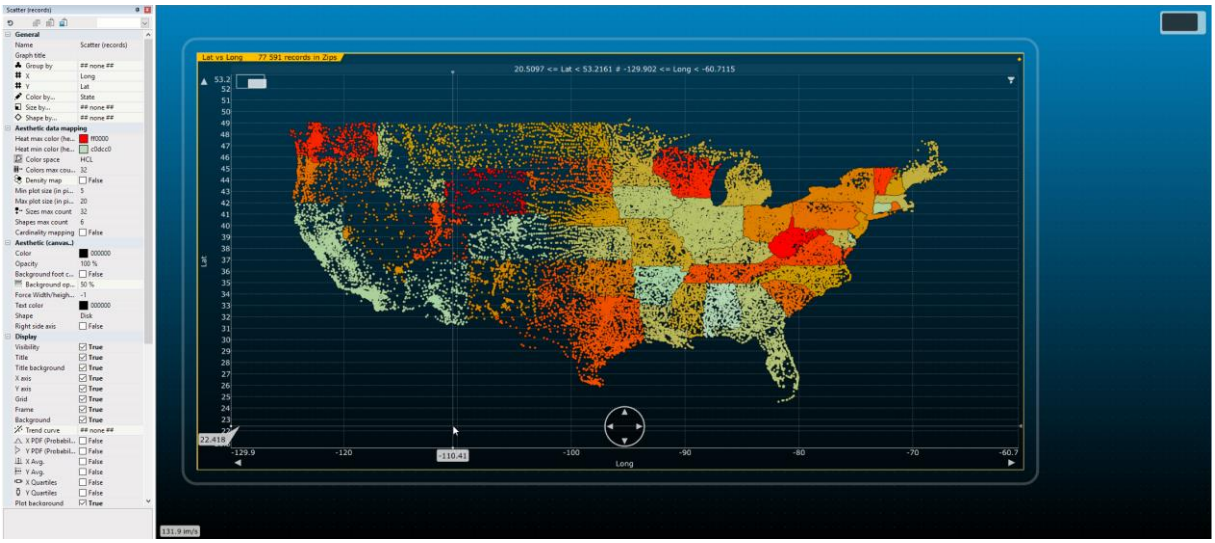


Figure 11 Chart properties of the selected chart

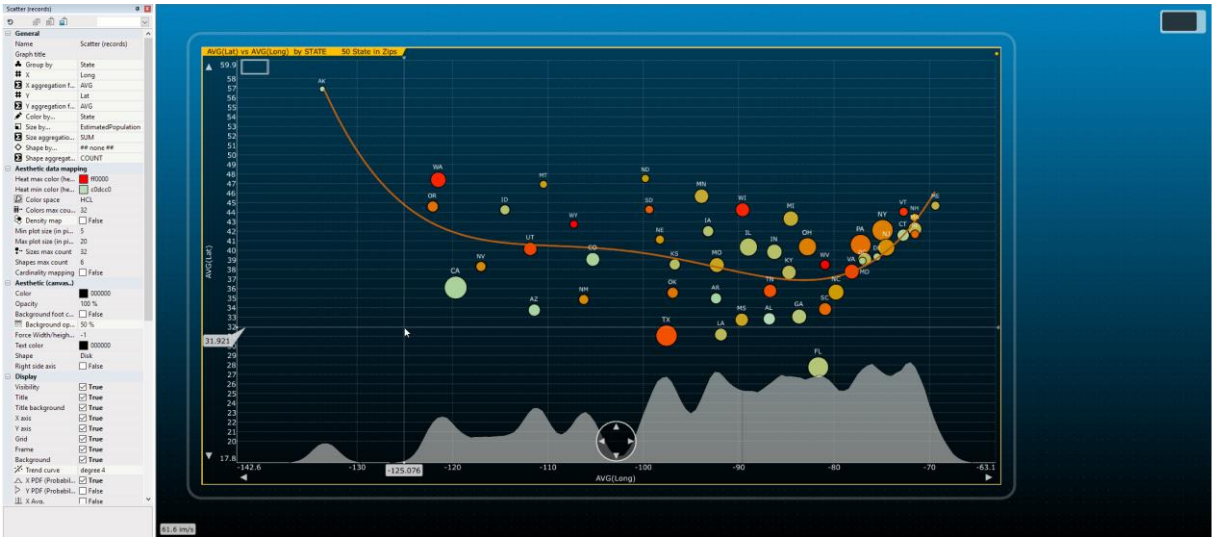


Figure 12 The same chart with different properties and graphs

4.8 VIEWS

A view is the composition of multiple charts on the same display, it's basically what you see. A single current view is valid at a time. A view captures each visible chart property. You can create as many views as you wish in a workspace and navigate between them.

Selection is kept when you navigate between views, and the time to switch from a view to another is designed to be very quick. Views may be even played to tell a story if you need to.

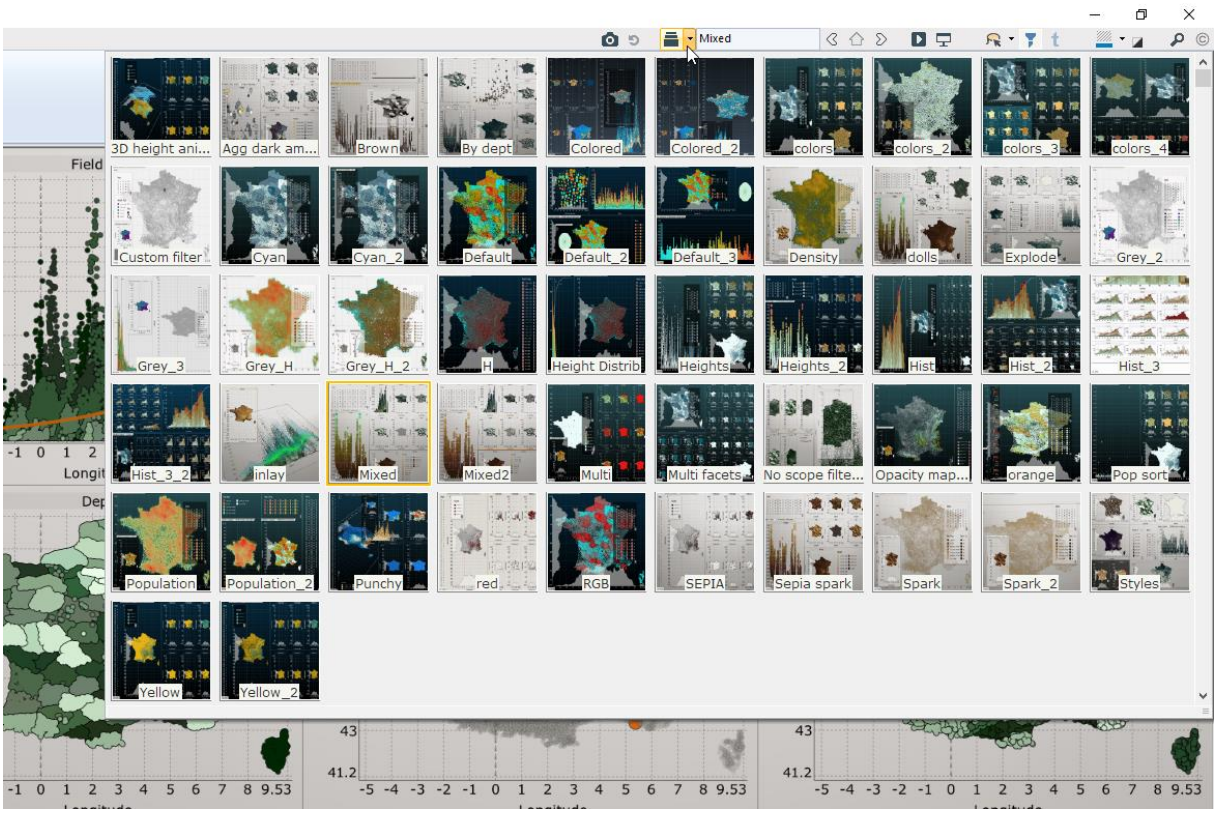


Figure 13 Navigation between views

4.9 BRUSHING

When you select records (chart, grid etc.), **data brushing** mechanism highlights the corresponding data values in the other charts. Depending on the chart, it may be displayed differently, but the goal is always the same: give insight on selection vs all records.

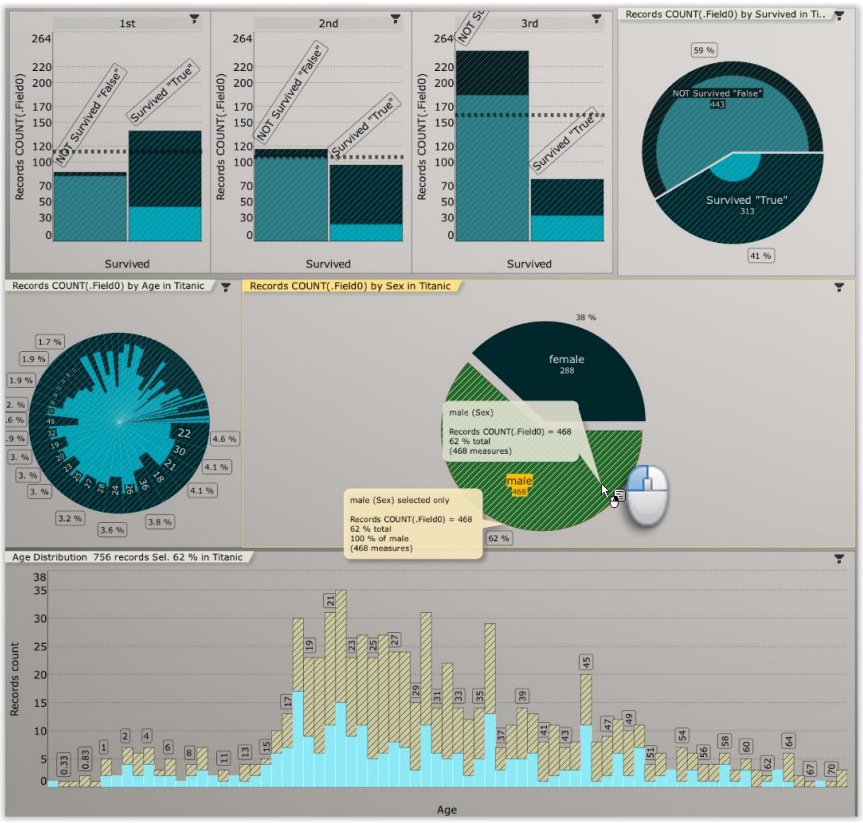


Figure 14 Data brushing (in light blue)

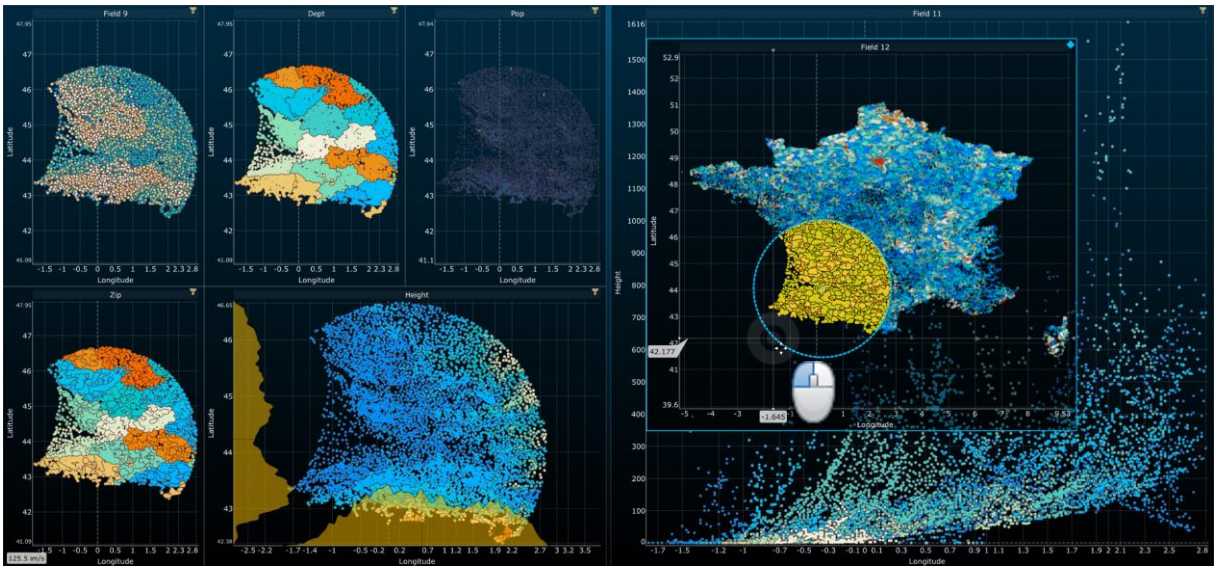
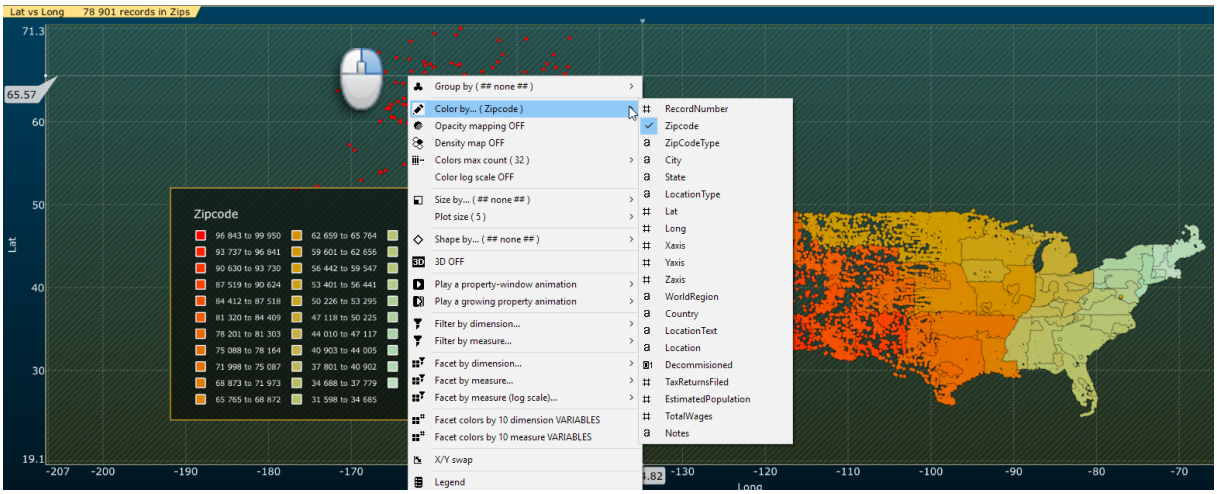


Figure 15 Selection brushing combined with filtering

4.10 AESTHETICAL DATA MAPPING

Aesthetical properties in graphs (color by, size by, shape by) may be defined by dimension(s) or measure(s) values.



4.11 SCALING

Scaling properties generally define a property range: min, max and limits. For measures, it also defines if semi-**log scale** is applied when a property is visualized.

Uniform scaling implies that 2 different measures will be represented by the same length on screen. It is useful when representing (longitude, latitude) data-points in scatter plots.

Mouse wheel and right drag zoom window in the graphical area of a chart automatically disable auto-scale of the chart. When auto-scaling is disabled, some new glyphs are displayed in the chart to allow navigation inside the graph.

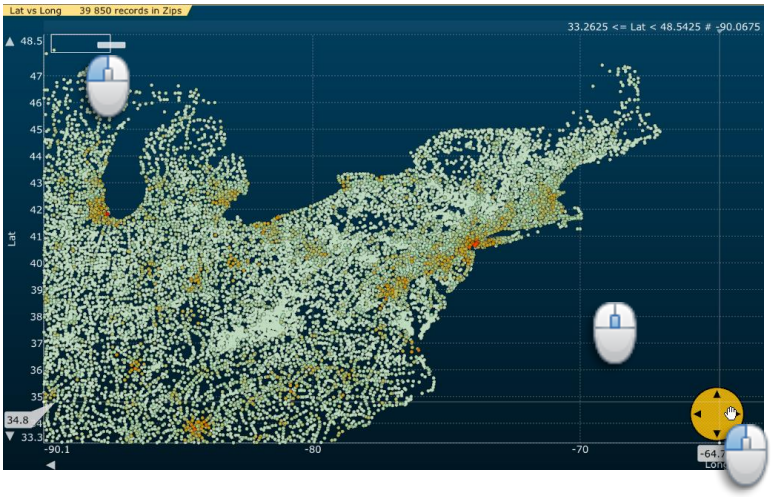


Figure 16 Scaling glyphs

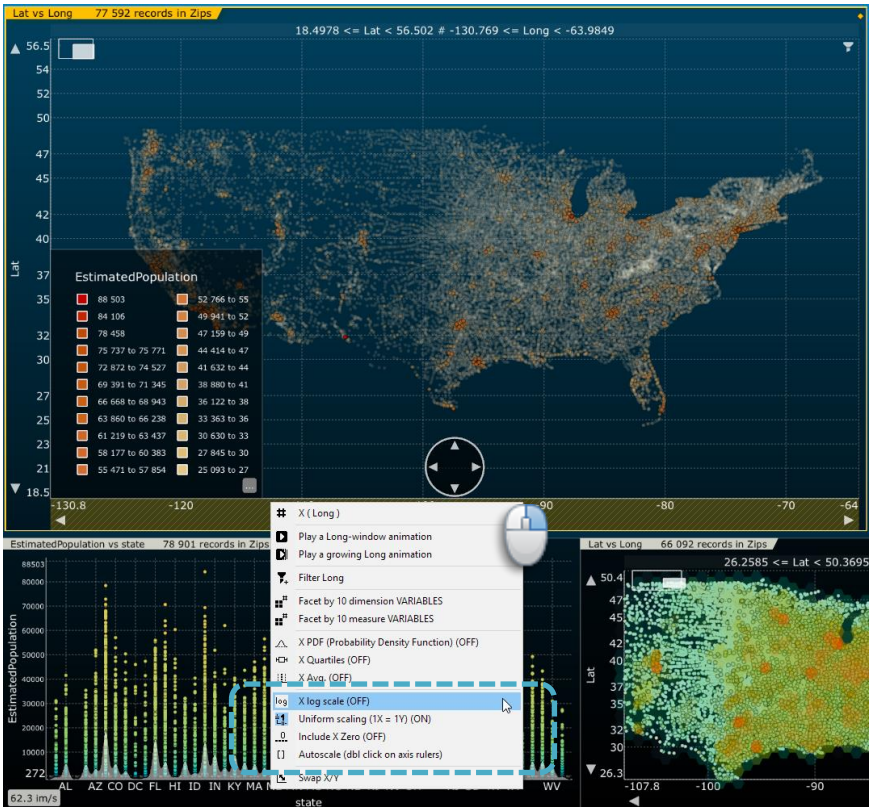


Figure 17 Scaling commands in axis menu

4.12 Scoping <>

Scoping allows you to limit filtering and legend to a subset of charts. In Quantyl, scoping is strongly coupled to grouping (chart structure).

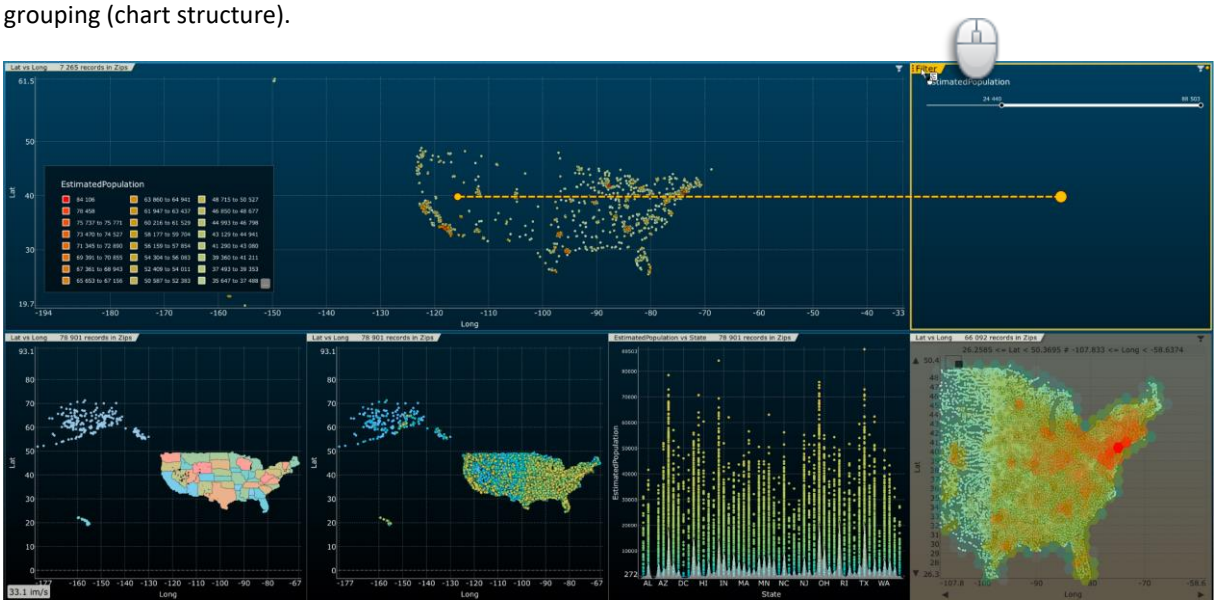


Figure 18 Filter grouped with a single chart

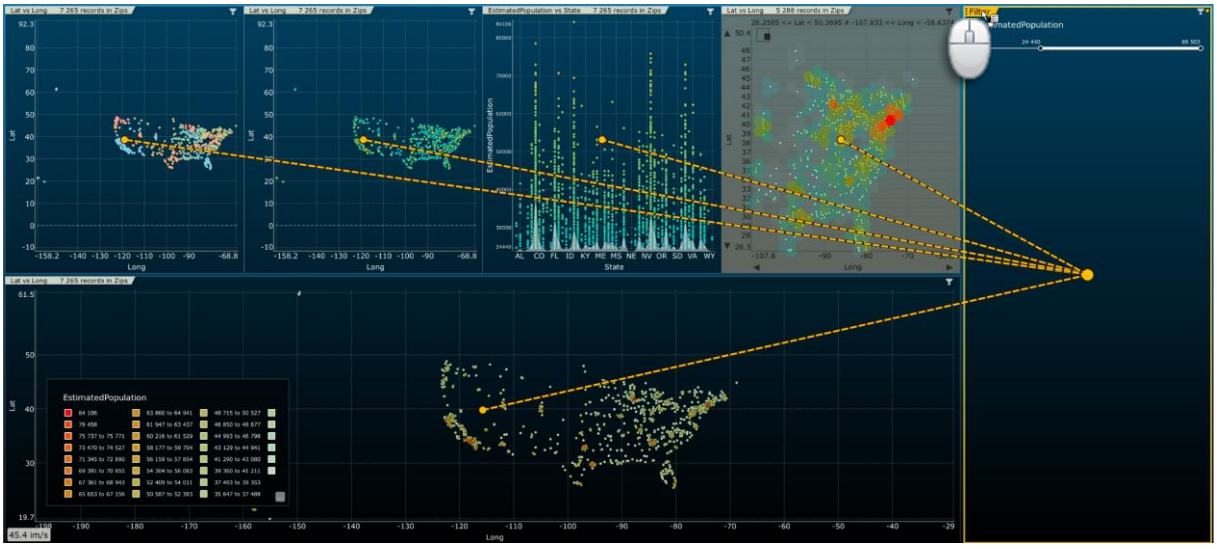


Figure 19 Detached filter applied to each chart

Important

Put the mouse over a filter or a legend title to display scoped chart.

4.13 DATA DISCOVERY

In your data discovery process, you will create and organize multiple charts in views to filter and visualize records. You will aggregate them through different dimensions. You will perform **Aesthetic data mapping** by assigning dimension or a measure to some graphical properties like **color**, **size** or **shape.** It will be also possible to animate properties to see data in motion (like time progression etc.)

5 USER INTERFACE PARADIGMS

The Microsoft Windows® ribbon style interface organizes elements improving the accessibility, usability, and discoverability of its functionality. The Ribbon is designed to help you quickly find the commands that you need to complete a task. Commands are organized in logical groups, which are collected together under tabs (categories).

Visualization, navigation and interactions are based on 5 pillars:

1. **Morphic User Interface**: Group/dock charts (including indicators) in a natural manner with drag & drop. A hierarchical structure of charts allows to perform scoping (filters, legends) and adaptive view layout (Tile & Roll + Autofit mode)
2. **Zoomable User Interface:** Thanks to **ZUI** capability, chart display may be adapted to any device resolution, and you won't miss any detail in a view
3. **Selection / properties** core mechanism (data brushing: cross selection, cross highlight etc.). Property change are **undoable**. Properties may depend on the current context.
4. **Contextual menus:** menus may differ depending on what is selected, chart state, chart area etc.
5. **Persistency and replay**: Workspace and Views capture everything

To avoid losing context, many commands are **animated** (zoom, property change, views etc.). Interpolation needed for **animation is performed on data, not graphics**. If you change aesthetical data mapping like the number of colors used, or number of bins in histograms, you will experiment the true interpolated states, with anything related updated (like legend). This seems like a detail, but this demonstrates coherency and capabilities of the architecture.

There are not a lot of specialized dialog boxes in the application. Property page, contextual menus on charts (axes properties), and some combo boxes in ribbon are completely property-driven (not blocking, not modal); they are preferred to static UI components like Wizard dialogs. Depending on the objects & properties exposed to the system, you get different UI.

When you put the mouse over a command and wait a second, a **tooltip** is displayed to describe it.

WYSIWYG (What You **See** Is What You Get): You get highlighting and tooltips on almost every displayed graphical representation with a cross feedback in the different charts (cross selection, cross highlight, scope links, etc.).

UI may be customized as de-facto standard ribbon user interfaces.

5.1 Morphic User Interface

5.1.1 Charts transformations

Quantyl implements some of the powerful and visionary concepts of Morphic UI to group and transform charts in a natural manner (see Working with charts section).

Morphic is a graphics system which uses graphical objects for simplified GUI-building which allow for a great degree of flexibility and dynamism.

Morphic UI are **responsive** by design; they can adapt to any device resolution.

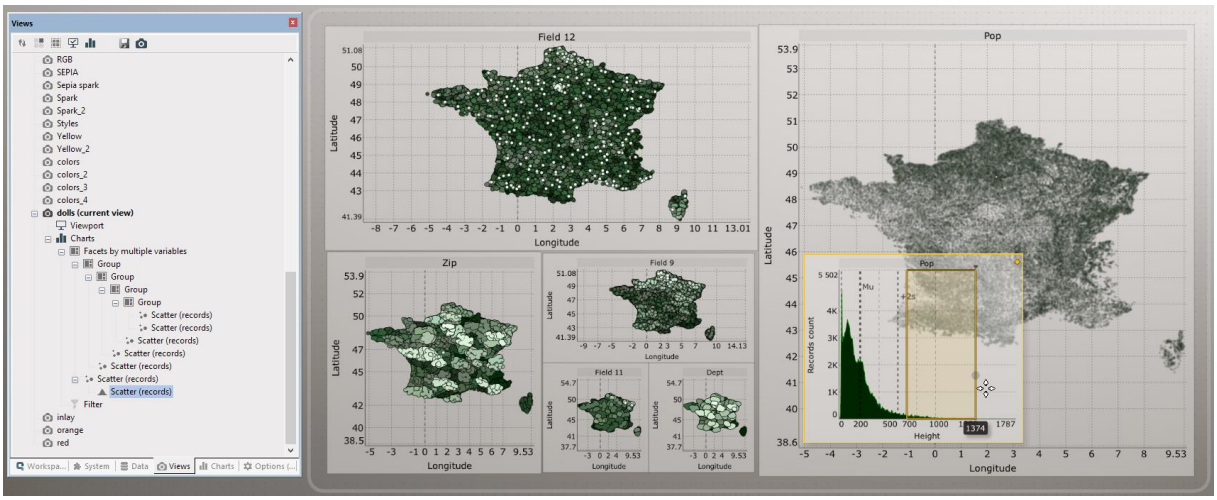


Figure 20 Morphic UI

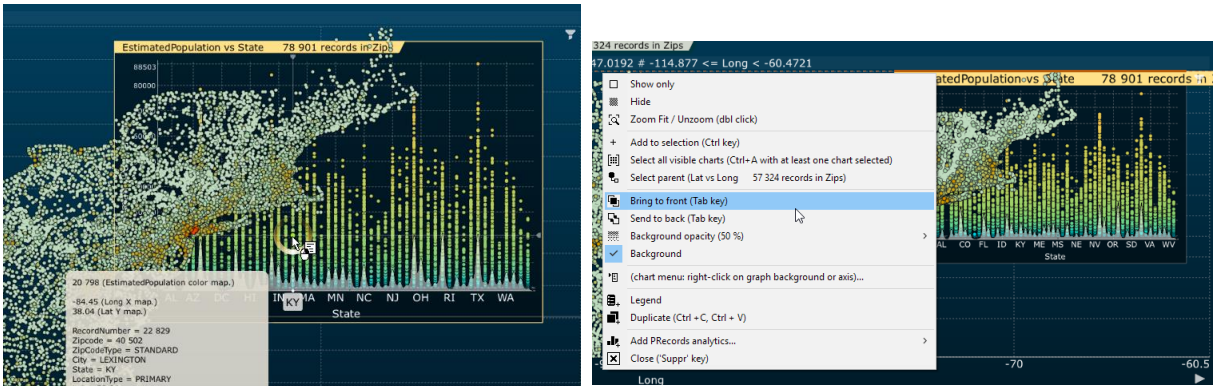
*Morphic UI has been introduced by **John Maloney** and **Randy Smith** for the Self programming language, starting around 1993 (Published under **"Directness and Liveness in the Morphic User Interface Construction Environment"** in 1995).*

5.1.2 Adaptive layout

Through drag & drop grouping, and autofit modes that applied both on **groups** and **viewport**, multiple chart display may adapt automatically without losing any space in the viewport, with desired consistency.

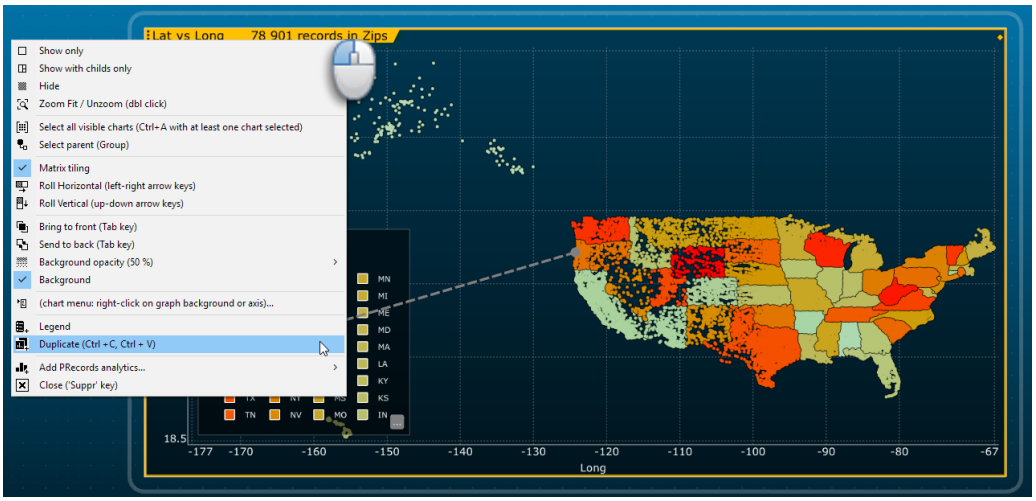
5.1.3 Transparency and charts interactions

When charts are stacked and have no background or are drawn in the back of the parent, it is possible to pick through them, this is really "true" transparency that goes **beyond boundaries of classical windows**: the "sensitivity" of content has priority on window background. You can force child charts to be drawn in the parent background if you wish ("Send to back" or "Tab" key).



5.1.4 DUPLICATING

Copy is a very simple and powerful tool; you can copy charts from libraries or copy existing running objects by simple drag & drop. Then, by modifying properties, you can achieve what you want to display.



5.2 ZOOMABLE USER INTERFACE (ZUI)

A Zooming user interface or zoomable user interface (ZUI, pronounced zoo-ee) is a graphical environment where users can change the scale of the viewed area in order to see more detail or less. A ZUI is a type of graphical user interface (GUI). Information elements appear directly on an infinite virtual desktop (usually created using vector graphics), instead of in windows. Users can pan across the virtual surface in two dimensions and zoom into objects of interest.

Chart area (AKA viewport) is fully ZUI:

- **Ctrl + mouse Wheel** to **zoom**
- **Ctrl + Left** button to **pan**
- **Ctrl + Right** click to **zoom window**
- **Double click (double tap) as universal way to Zoom or Unzoom (if already fitted). Top-right glyph display the current position of the camera in the view, you can double-click it too to fit all.**

Note: Zooming on a chart or a part of chart is NOT scaling. Scaling is generally interactive in the charts: if you perform a zoom in the graphical area of a scatter plot, scaling will change on X and Y axis. Uniform and log scaling are accessible through axis contextual menus or property page of the chart.

During zooming, you will see a bird eye view of your position in the view in the top right corner. Double-click on it to re-fit your position.

Figure 21 ZUI glyph

Reflexibility

One of the well-known drawbacks of ZUI is that at some points, user may be lost: accidental pan, zoom-in or zoom-out, double click to fit a chart etc. It's why Quantyl puts Zoom fit all as a reflexive command when double click is performed. It means that a series of double click that starts to zoom will always end-up by a zoom fit all when it zoom-in makes no sense at all.

Dragging a zoom-out window from right to left is also a reflective command.

5.3 SELECTION AND PROPERTIES

Property mechanism is at the core of the user interface and data model.

Every setting, parameter, variable, output result is managed through properties. This mechanism is open and extensible, so custom charts may expose dedicated properties that are manipulated, saved as any default property of the system. Several hundreds of different properties are available by default.

Property mechanism is closely linked to the selection mechanism. When you select multiple objects, the properties of the selection are aggregated. **You can modify property of multiple objects with the same command**. For example, you can select 10 charts and change background color or aesthetic mapping with a single change in the convenient property page field. Selecting records, charts, managers, viewport etc. follows the same process and UI.

Important

You can modify property of multiple objects with the same command.
When you right-click on a chart to open its menu, commands will happen on the selection of the charts

5.3.1 SELECTION-FIRST PARADIGM

When a command works on a selection (like grouping), you have to set the selection **before** invoking the command. If nothing is selected, or selection is not compatible with the command, the command will not be accessible (disabled in the UI). You can generally combine new selection to the previous one by pressing SHIFT key or performing selection from right to left (there is a feedback).

5.3.2 TEMPLATES (DEFAULT PROPERTIES)

Charts are based on dynamic libraries that can be extended. Default properties for any chart type (class) may be modified and saved as a template. Any new chart created from the same class will get those properties by default.

Those default properties are shared by all workspaces and stored at user level by default.

5.3.3 SEARCH

There is a single search command in Quantyl, but it works on any object that exposes properties. You can search and find workspaces, charts, records and almost anything that handle a property. Making a choice in the dynamic list selects the object and displays the corresponding tree if needed.

It is a powerful search as everything is based on properties (name or value). Obviously, this mechanism will work with all custom objects you introduce through the framework, at 0 cost.

5.3.4 UNDO / REDO

Almost all actions and change in Quantyl are based on commands objects that are undoable. Undo stack is a powerful tool as it allows user to perform exploration and “what if” scenarios and then undo those non-permanent modifications.

5.4 CONTEXTUAL MENUS

Contextual menus are displaying commands when you mouse right click an object. This object may be a chart, a graph ruler, a specific area of a graph, a selection, a node in a tree window etc.

More often, menu commands are contextual, depending on charts properties, dimensions or measures, single or multiple selection etc.

When multiple objects are selected, the command will be applied to the complete selection if possible.

5.5 PERSISTENCY

Workspace and views are capturing every state of your current study.

If you close and re-open a Workspace, you should be exactly in the same state (except selection). Everything is saved in the workspace folder which can be archived, copied or shared.

6 USER INTERFACE, WINDOWS PRESENTATION

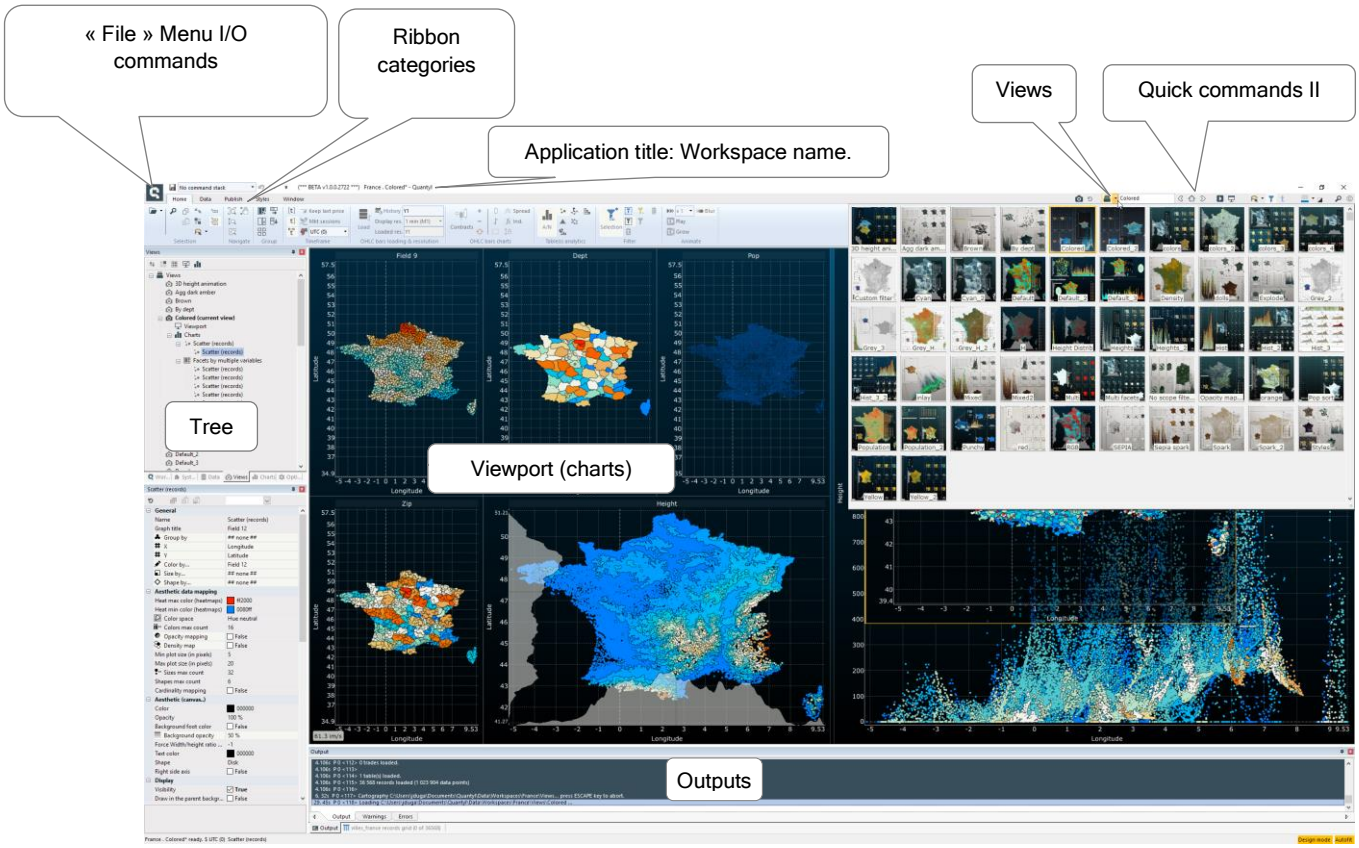
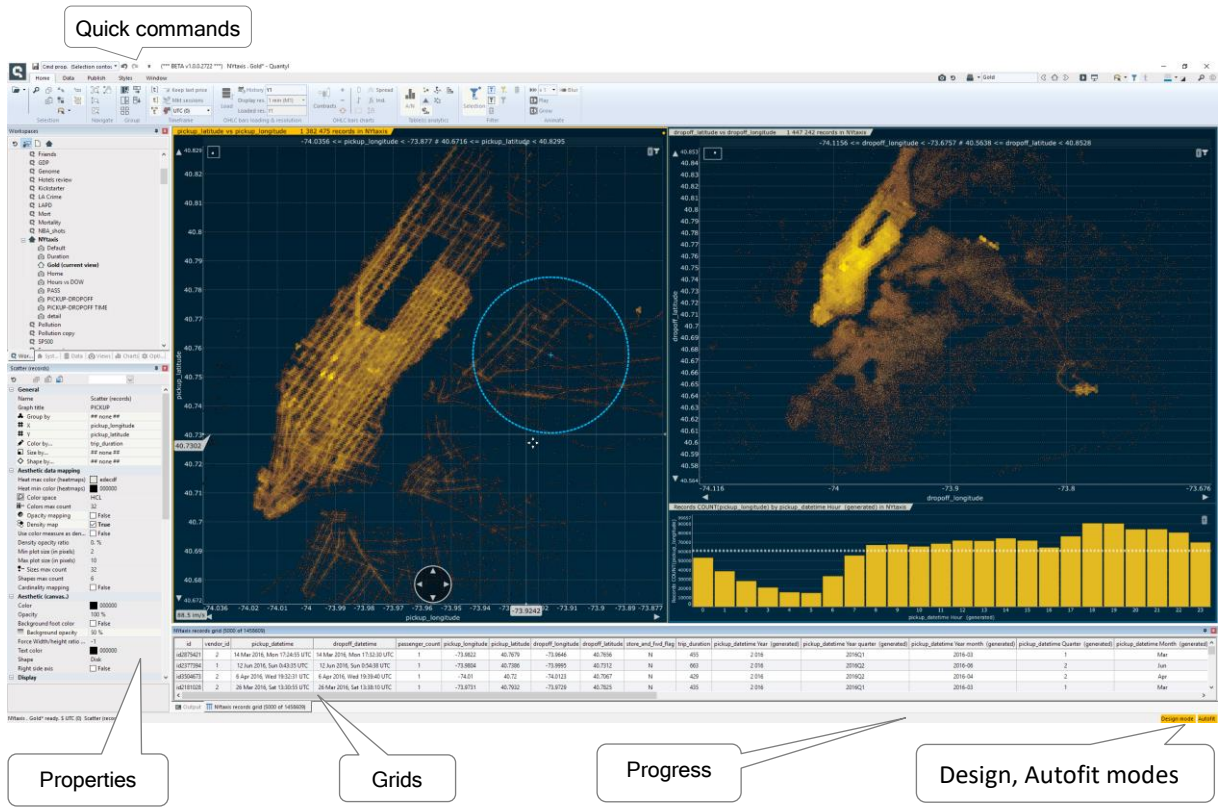


Figure 22 Discovery UI



If some UI window is not accessible, you can display it with the dedicated commands in View category:

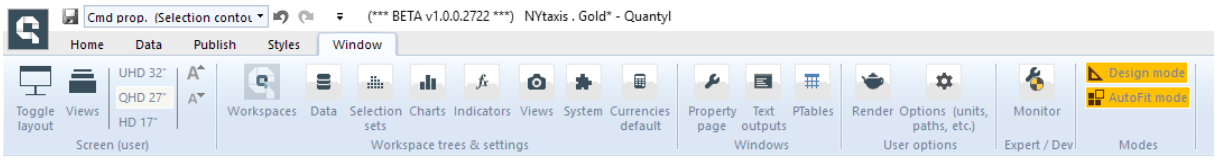


Figure 23 UI Display

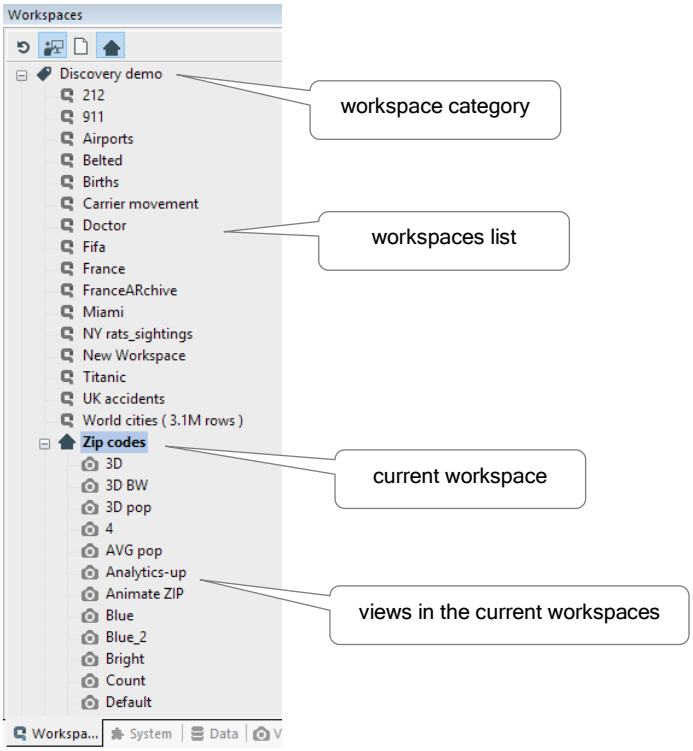
6.1 WORKSPACES AND VIEWS

When you change properties, launch commands in the UI, change occurs in memory.

A single Workspace is opened in memory. Loading a new workspace unloads current one. But you can run multiple Quantyl.exe sessions with different loaded workspaces.

Saving is a manual action that needs to be performed or validated by the user. Fortunately, if you attempt to quit or load a new workspace or a view, you are prompted to save.

For further details on views, see Views section.



6.2 RIBBON AND CATEGORIES

UI is Microsoft Ribbon style. Categories are self-explained, and tooltip on each command helps you to understand what their purpose is.

Category	Description
Home	Chats navigation and display
Data	Data import / export / update, calculated column adding
Publish	Image generation / print
Styles	Charts and viewport styles change as a whole
View	UI windows display
Monitor (dev)	*Debug/dev mode. Log options (verbose levels)*

Right click in a ribbon to customize user interface.

6.3 TREES

Trees are supporting node right click, double click, drag & drop inside the same tree, between trees, and drag & drop to the viewport.

Tree	Description
Workspaces	Workspaces management
System	System management, loaded contract and default currency rates
Data	Tables and fields Historical data management
Views	Chart and view management Views and chart hierarchical structure of the current view. It reflects what is currently displayed in the viewport
Charts	Chart library Simply drag & drop them into charts or viewport
Options	User options Set your preferences (units etc.)

Tree nodes are objects that you can select and modify through the property page. Some nodes **seem** to be only tree structure nodes but are managers' nodes. Some properties, like bar color preferences, have no commands in the Ribbon UI. If you select the corresponding manager in the charts tree, you discover new properties you can interact with. Hundreds of properties are available, that allow high level of customization.

6.4 PROPERTY PAGE

Property page is **essential** to modify and consult object(s) properties. Even if some properties have different ways to be modified (through commands in ribbon for example), properties are always handled by object(s) you select.

Property page supports **multiple objects selection and modification**; it's very helpful when you want to assign the same property to multiple objects in a single shot (like changing background color of multiple charts etc.).

When multiple objects are selected, measure properties are aggregated, and you can choose what kind of aggregation function you want to apply.

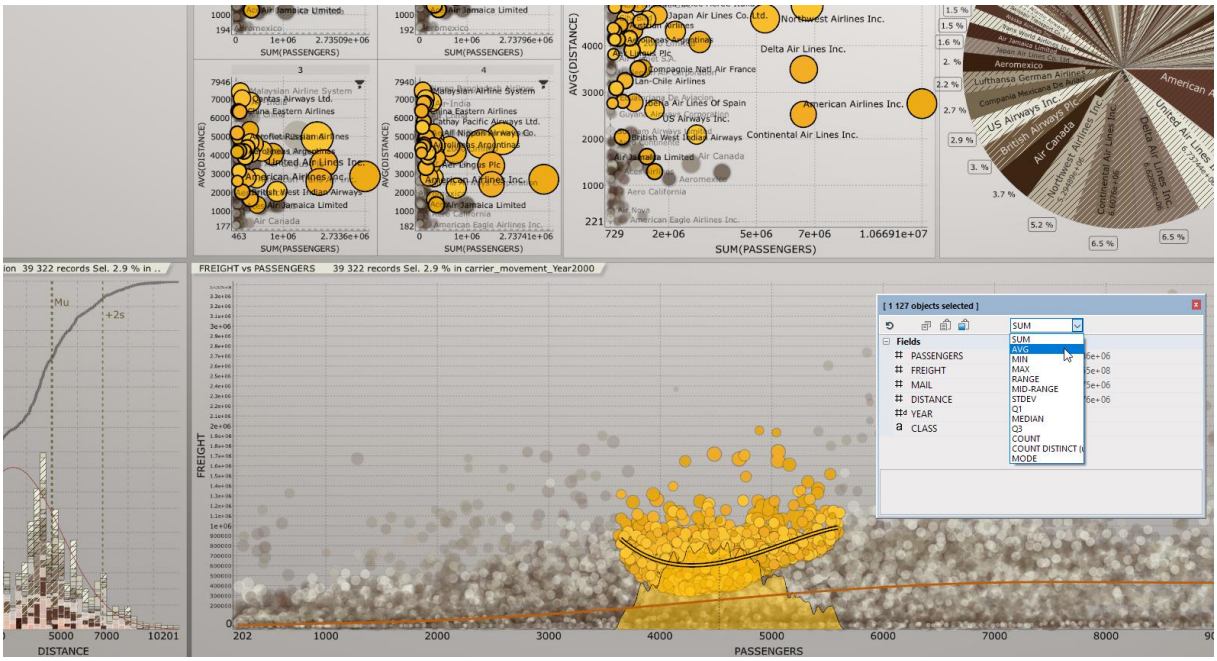


Figure 24 Properties aggregation function

Last property change may be re-called on a new selection through **F4** command.

Important

Property page is always synchronized with the selection.

6.5 VIEWPORT

Viewport is where charts are drawn, where the current view is displayed. You can put viewport full screen if needed.

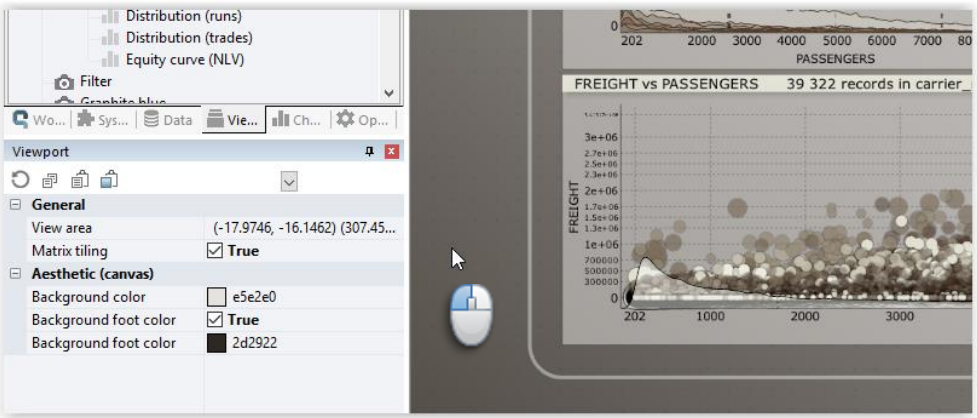


Figure 25 Viewport background properties

6.6 GRID

Grid displays records and properties. You can select records directly in the grid. You can also sort them by property. The grid window is unique, and it is dynamically linked to the chart-table selected.

Grid supports unlimited number of rows.

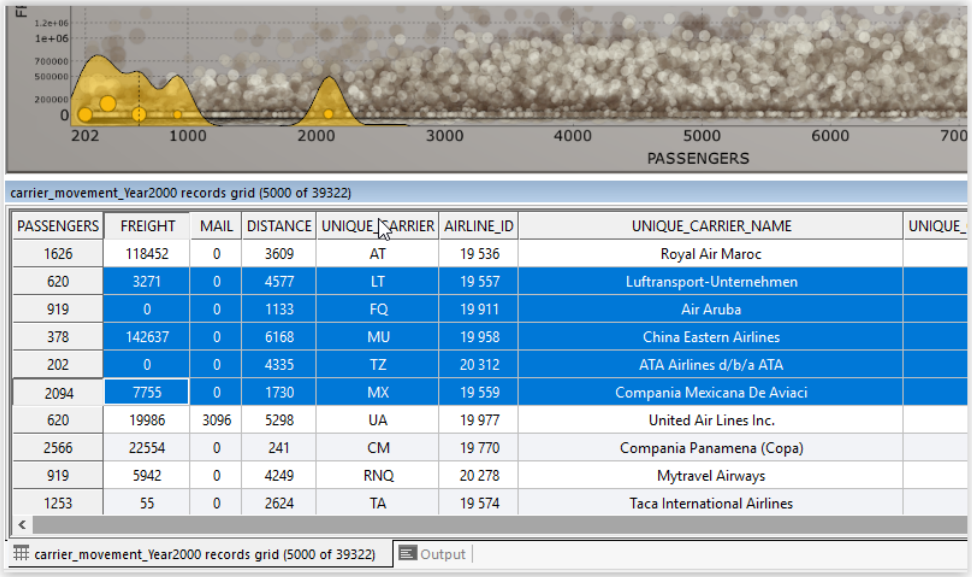


Figure 26 Grids

Important

Dynamic selection filtering is also applied on charts and grids.

6.7 OUTPUT

Output is complete log of your session; you can increase or decrease information displayed with verbose levels available in the **monitor** category.

7 WORKING WITH SELECTION AND PROPERTIES

7.1 PROPERTY PAGE SYNCHRONIZATION

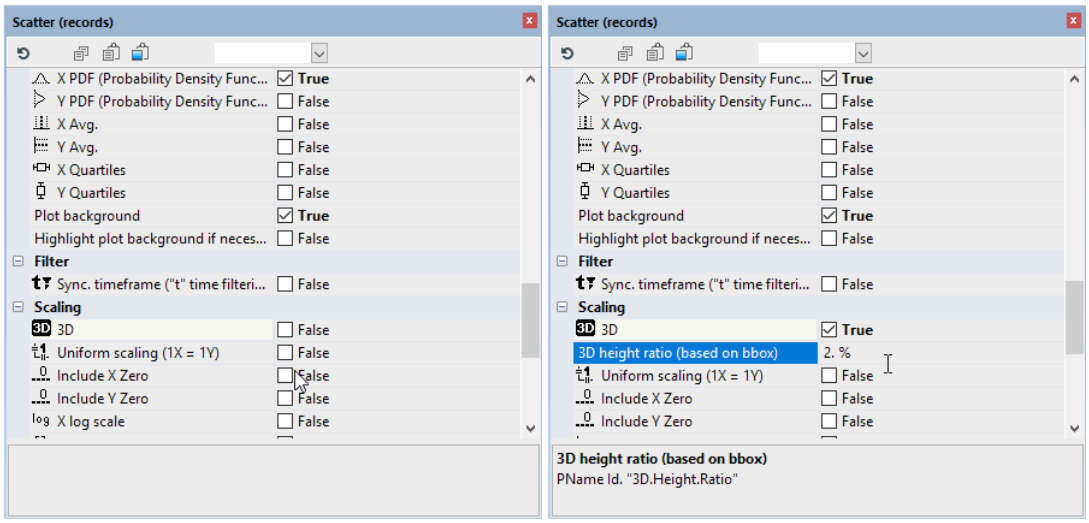
Property page always reflects the properties of the selected objects. If multiple objects are selected, and the different values of the same property are different, the field appears blank.

Important

Property page aggregates properties of the different objects in case of multiple selections. Through the property page, you can modify properties of the entire selection in a single step.

Some properties are conditional; it means that properties appear only in some states.

In this example, enabling “3D” will display “3D height ratio…”:



7.2 MULTIPLE AND ADDITIVE SELECTIONS

Quantyl support standard Windows mechanism to perform multiple selections with **Ctrl** and **Shift** key pressed. It works in trees, grids, charts and records.

When you perform a window selection by dragging a selection window from **right to left**, it is generally an **additive selection** (scatter, distribution, “by” charts).

Figure 27 Additive selection (right to left)

7.3 MULTIPLE REPRESENTATIONS

An object may have several representations: it may appear in multiple trees, grids, and charting area. Object can be selected through all those representations and feedback of the selection is updated in each of its representation occurrence. And of course, property page is updated consequently.

7.4 DRAG & DROP

You can experiment drag & drop of symbols, charts, fields inside the same tree, between trees, or from trees to charts area, or axis. Let's try it!

7.5 CONTEXTUAL MENU (RIGHT CLICK / TAP AND HOLD)

Contextual menu is enabled through **right click** (or touch maintained) on a selection. Contextual menu works on the complete selection if it makes sense; for example, if multiple charts are selected, the right click menu commands displayed on title of the charts is applied to multiple selected charts.

The menu may also be contextual to a zone in a chart, as when you "right click" on X or Y axis of scatter chart, or it graph area.

7.6 SELECTION BRUSHING (CONTOURS)

Scatter, distribution and lines charts offer a specific feature to drag previous selection through the entire area. You have no specific command to invoke, just drag the mouse over the contour of the last selected area (it will be highlighted) and move the selected region by drag & drop. You may combine this powerful feature with selection filtering to dynamically see how values area continuously (or not) change in different charts.

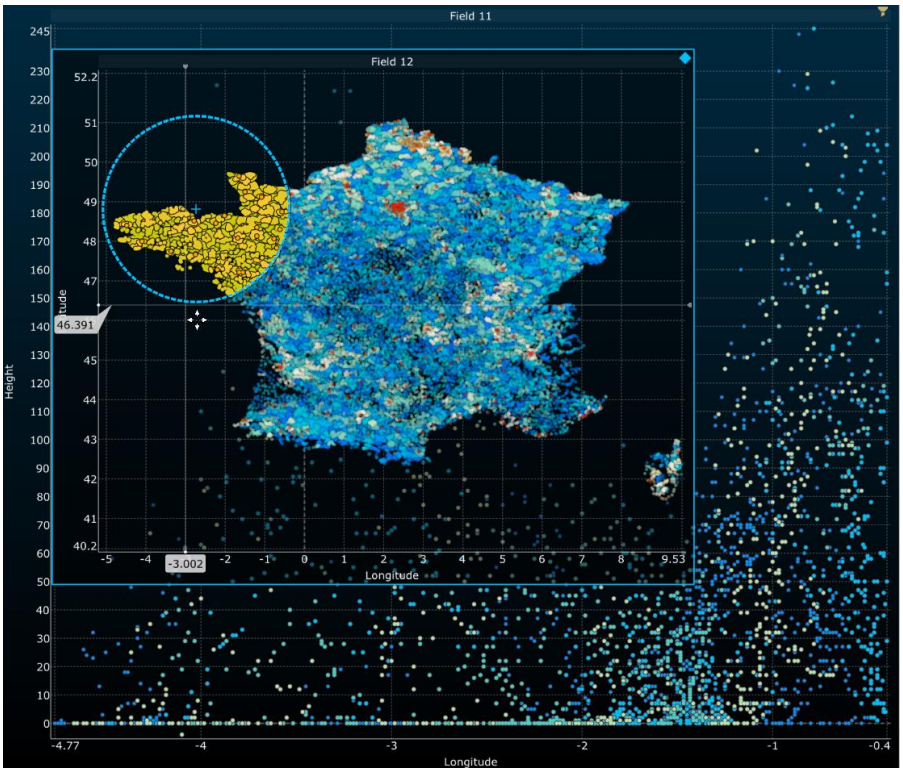


Figure 28 Selection brushing (scatter chart)

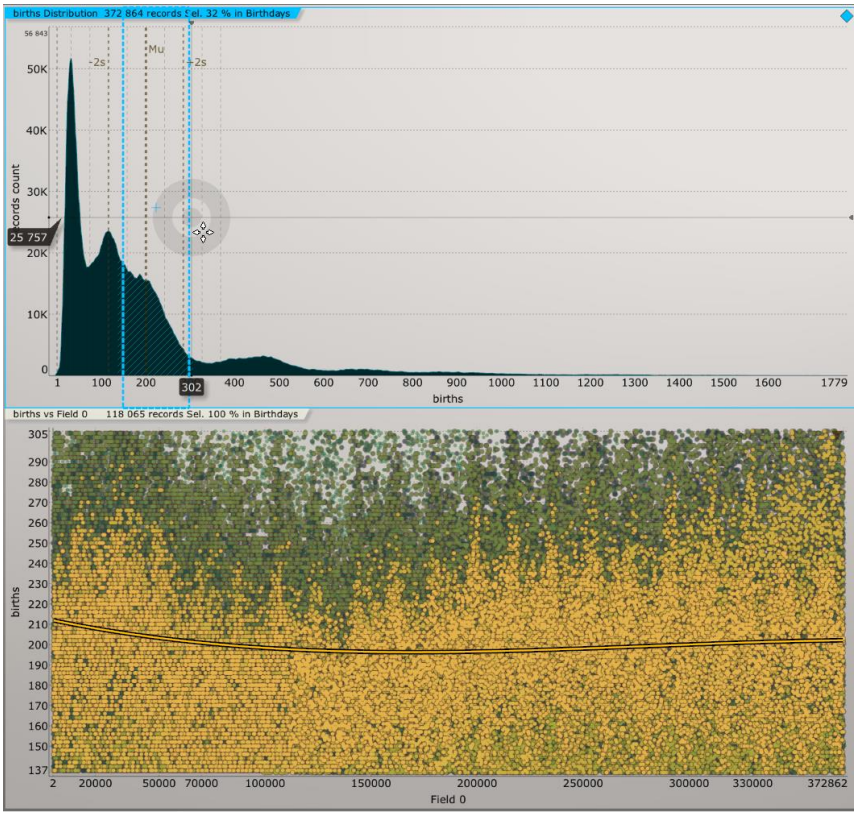


Figure 29 Selection brushing (distribution chart)

7.7 SELECTION SETS

You can capture a selection in a selection set to reuse it latter, or perform operation between selections (intersection, union, unselect etc.).

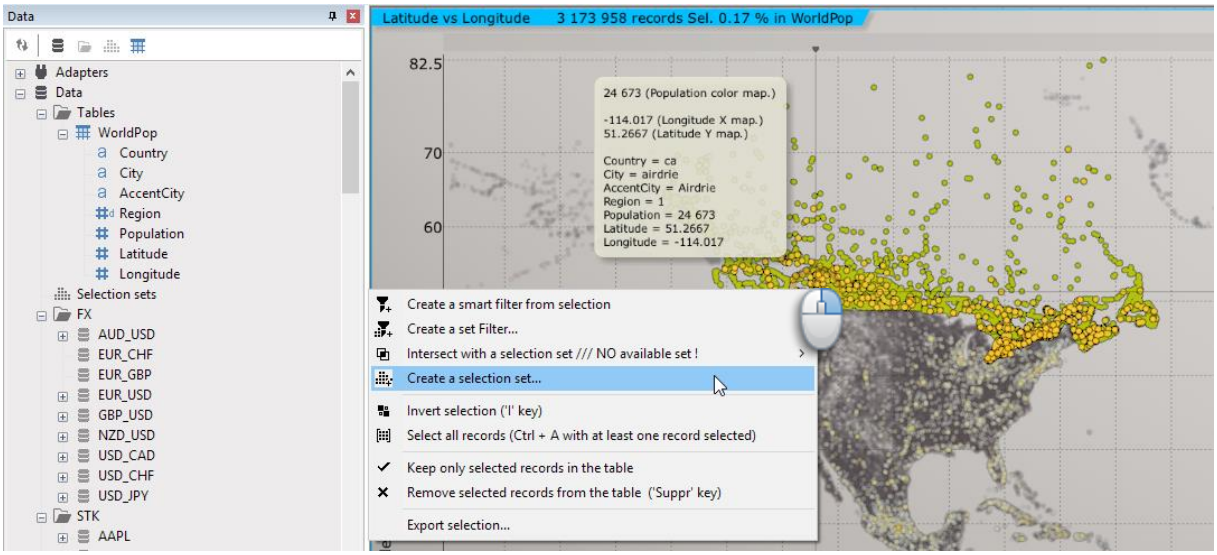
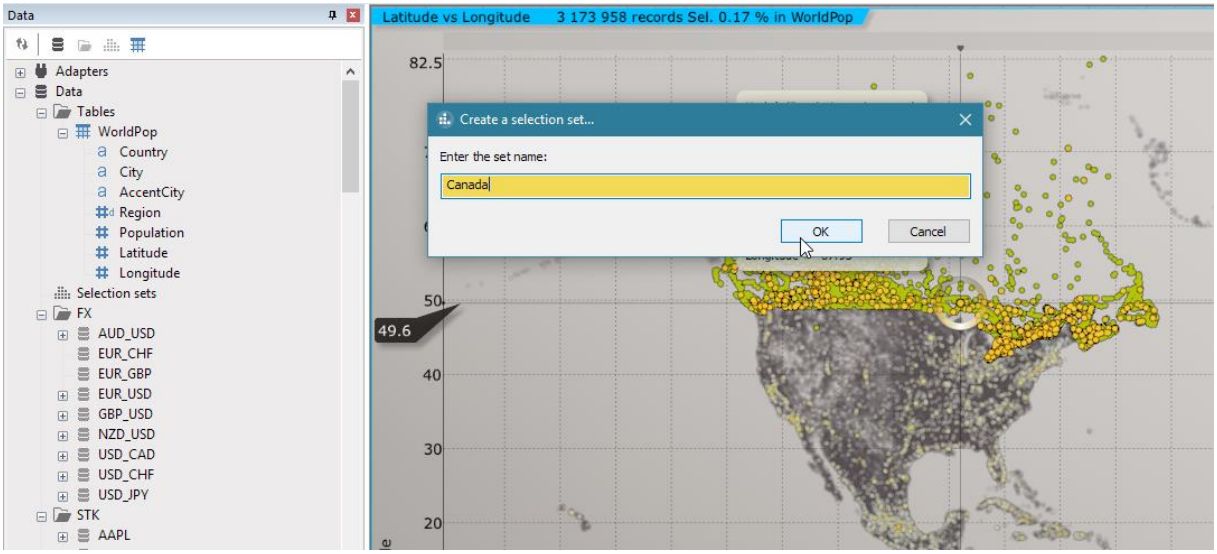


Figure 30 Creating a selection set with a right menu click on the selection



Selection sets is a working tool that you may use at any moment to perform operations with the current selection:

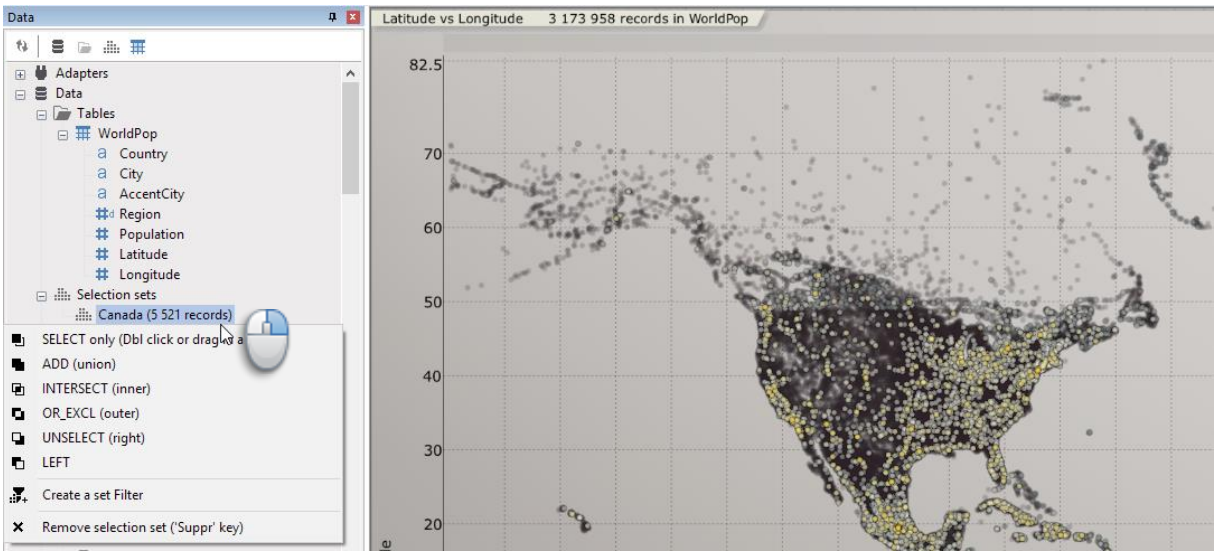


Figure 31 Selection set available in the data tree

7.8 Copy/Paste

You can copy/paste charts in Quantyl inside the same view or from view to view.

Important

You can copy selection (Ctrl + C) of any records collection from Quantyl and paste them in a text editor or any application that accepts text from the windows clipboard.

7.9 Undo/Redo

Undo/redo mechanism is implemented on any property change and almost all commands. Commands are stored in an undo stack, you can undo multiple commands in your history by a simple selection in the undo stack combo box, this will undo last commands including the selected one

To save some memory, you can modify the number of maximum undoable commands through command manager property (user options).

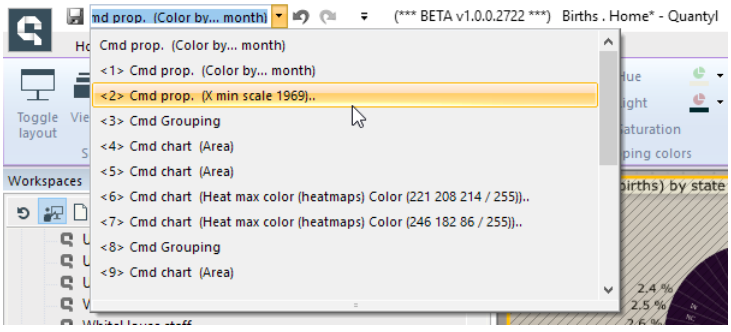


Figure 32 Undo stack

8 DATA

You may import data from different data sources:

- Text (comma separated values)
- Any database which offers **ODBC 64-bits connector**
- Custom adapter provided by a plugin
- Scripts (Python, R, etc.)

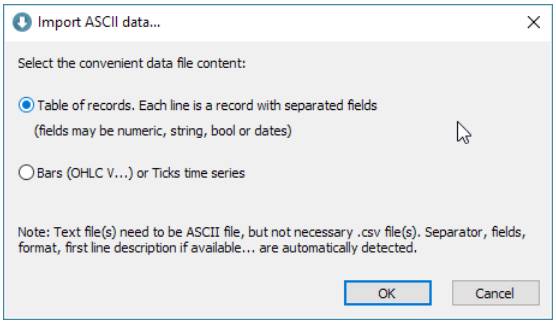
OHLC timeseries used in finance benefit of a specific data management in Quantyl. The reason is that they may be very data intensive and specific in their manipulation and representation (indicators etc., time resolution); they need specific data management to be efficient.

- Tables are attached and embedded in the workspace
- Market data (OHLC bars) are shared by all workspace

You can import data in different manners:

1. Simple drag & drop of the file(s) on Quantyl app
2. From file menu Import…
3. From data ribbon category
4. Command line

When you import data from a file, you are first prompted to know what kind of data you are importing:



8.1 IMPORTING TABLE(S) FROM FILE(S)

Separator, data types, datetime format etc. are **automatically** detected, the import process is very straightforward. Filter what columns you want to import and that 'it.

If there are too much data to import, you can decide to import a subset of those data. It is called **data sampling**. Simply enter a ratio in [0% 100%] of data you want to keep for your study. Rows data will be picked from the file through a uniform random function that matches your ratio. A row will be imported a single time or not. Data sampling is an interesting tool to get a quick overview on huge data.

At the end of the import, some charts associated to the new table are displayed, depending on the type and cardinality of dimensions and measures in the table.

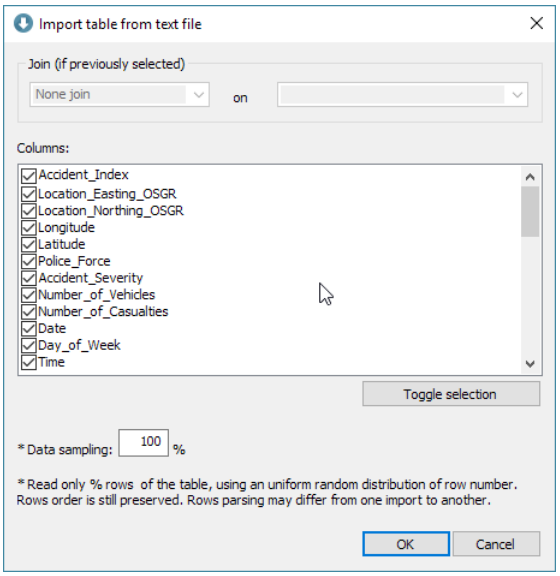


Figure 33 Column import and data sampling

Quantyl supports UTF8 code texts.

Limitation

UNICODE symbols are not yet correctly displayed in the viewport (charts, notes in tooltips etc.)

8.2 Installing Quantyl add 'in exporter for Microsoft Excel

It is possible to install a Quantyl macro to export MS Excel current working sheet directly in Quantyl through excel add 'in mechanism. The add 'in file is **quantyl.xlam**, it is copied in Quantyl document folder.

The macro automatically generates a temporary .csv file which is then opened in Quantyl.

Quick summary of the installation procedure:

1. Check that you have privileges to install macros on your computer
2. Copy [my documents]\Quantyl**quantyl.xlam** into [user]\AppData\Roaming\Microsoft\Add-ins
3. Enable Quantyl Add 'in in Excel/File/Options/Add 'in/Manage-Add 'in-Go
4. Finally, add a custom button or shortcut associated to the **ExportToQuantyl** macro: Excel/File/Options/Customize ribbon/Add commands – macro category

Please see the documentation of your specific version of MS Excel about installing macro for precise instructions.

8.3 IMPORTING AND TRANSFORMING DATA WITH SCRIPTS

Scripts integration is an important feature to customize data support and tailor the solution to your specific needs.

See Scripts chapter for more information.

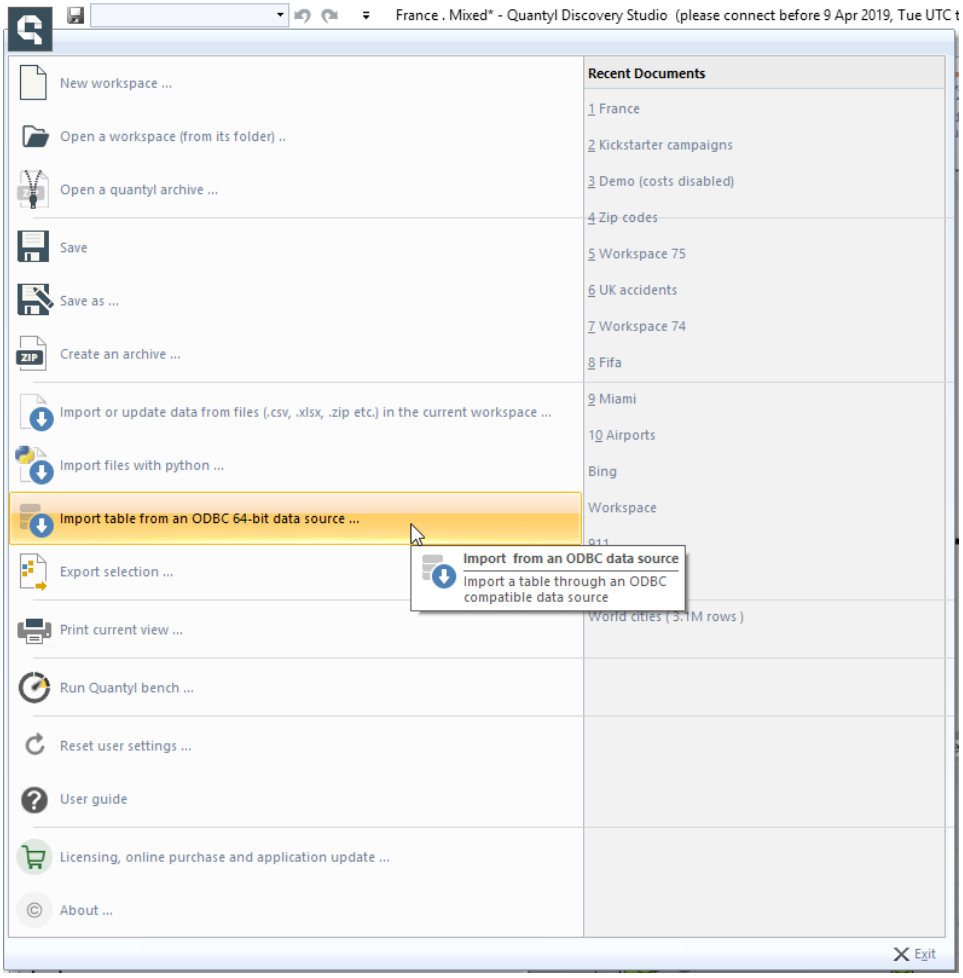
8.4 IMPORTING TABLE(S) FROM DATABASE WITH ODBC 64-BITS CONNECTOR

Database import is based on ODBC standard connectors installed on your machine.

Use Microsoft Windows ODBC 64-bits data source tools to create your desired data source.

Important

Quantyl is a 64-bits application only, a **64-bits** version of the needed ODBC connector has to be installed on your machine.



Retrieve the data source you created with ODBC 64-bits tool:

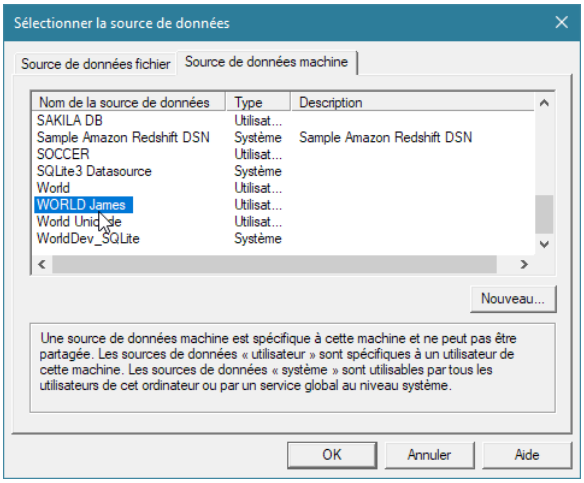


Figure 34 ODBC 64-bits data source

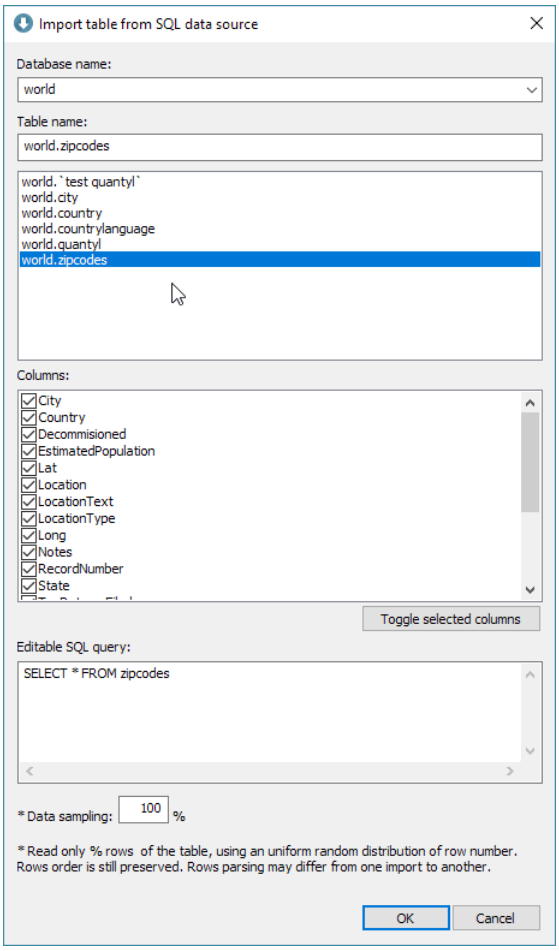
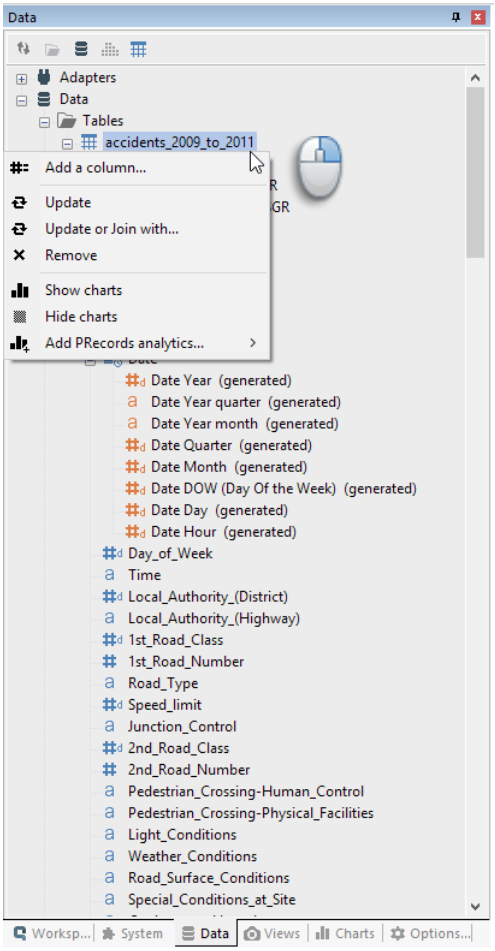


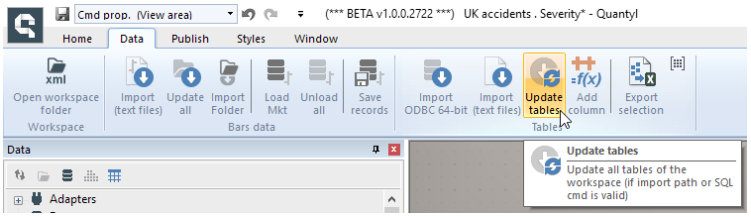
Figure 35 Data extraction

8.5 UPDATING AND JOINING TABLE(S)

You may:

1. Import data again, if new table seems to match an existing one, you will be prompted
2. Launch update command from table menu
3. Launch table(s) update from data ribbon category
4. Launch update command line on the desired workspace to update





You are then prompted to choose an update action:

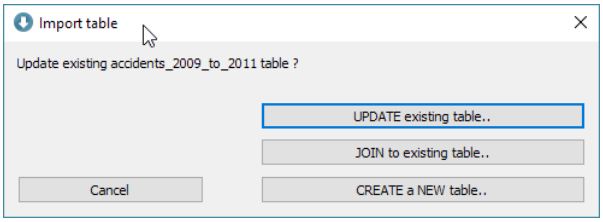


Figure 36 Table update prompt

Then the process continues as a classical table import, except if you have to choose a join command:

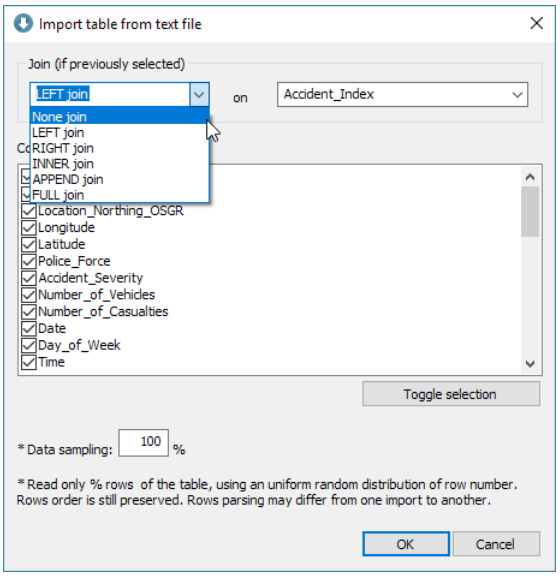
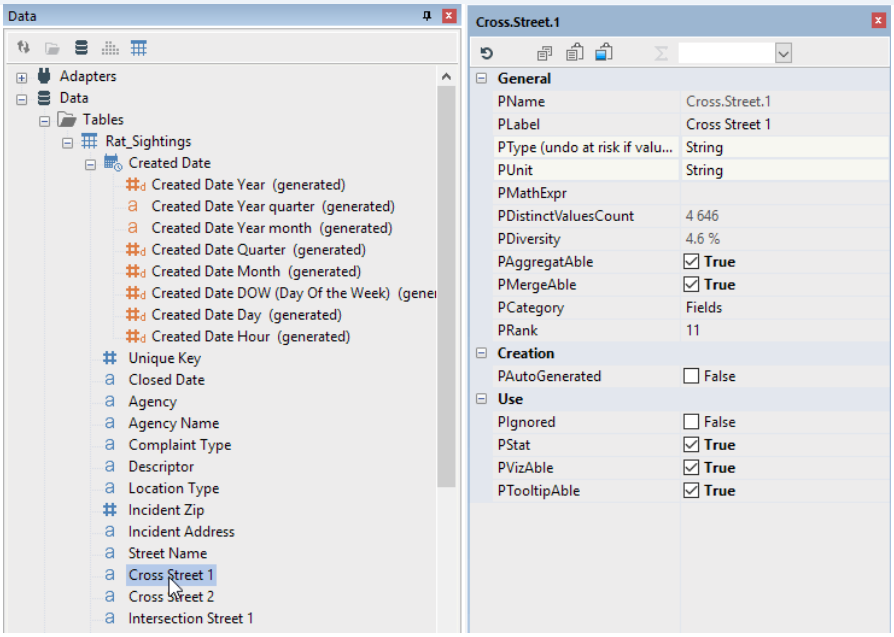


Figure 37 Table Join

8.6 WORKING WITH TABLES

You can browse table and columns in the data tree, modify table structure and their column properties. Some columns are autogenerated to facilitate data mapping. Datetime fields are decomposed in sub-dimensions to isolate year, month, quarter etc. You have also the capability to add new calculated columns based on mathematical expression containing fields as variables.

8.6.1 TABLE FIELDS



	Field type
Native table data	
a	String (dimension)
01	Boolean (dimension)
#	Numerical measure
#d	Numerical dimension
	Date (dimension)
Automatically generated (from date)	
a	String generated
#d	Numerical dimension generated
#	Numerical generated
	Date generated
User defined	
#:	Math expression (numerical measure)

In the table tree, you can still modify some properties of the fields (post-import).

8.6.2 TABLE COLUMNS, PROPERTIES AND COMMANDS

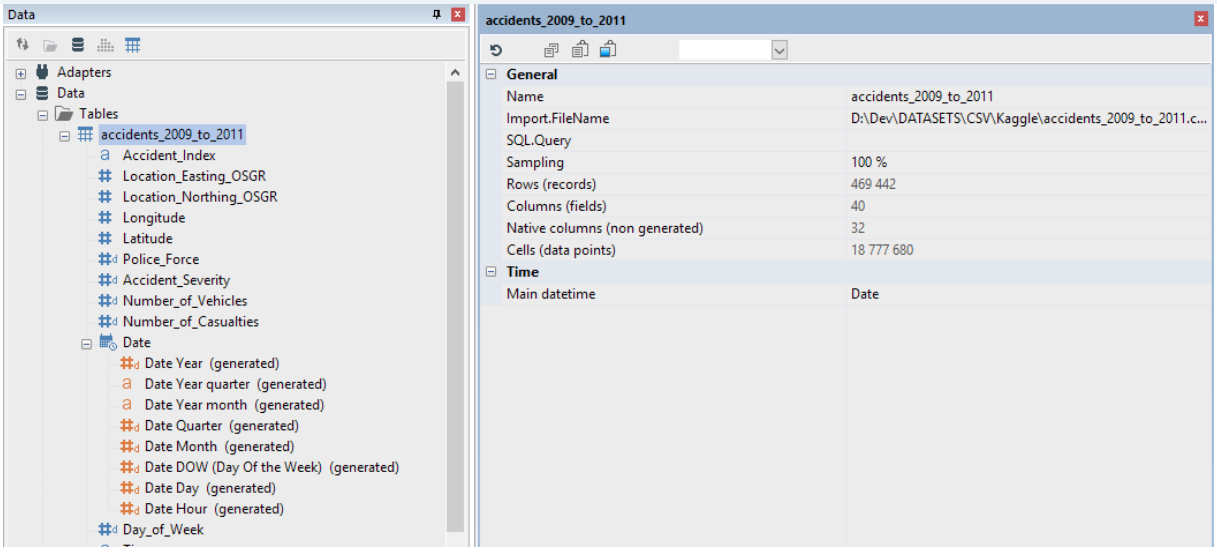


Figure 38 Table tree with fields

You can launch many commands from the table and fields popup menus:

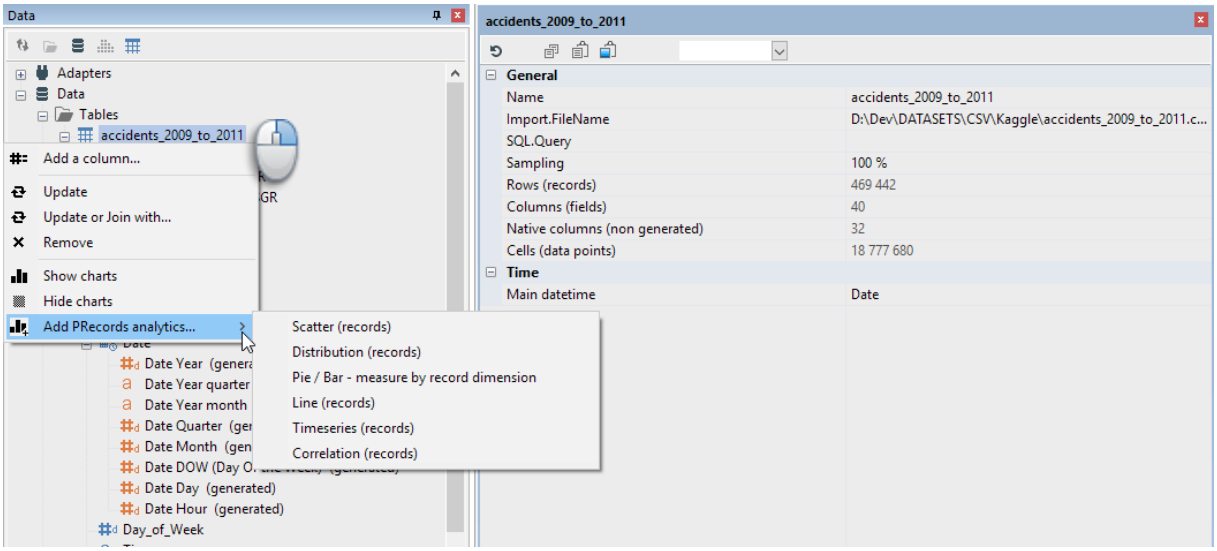


Figure 39 Table popup menu

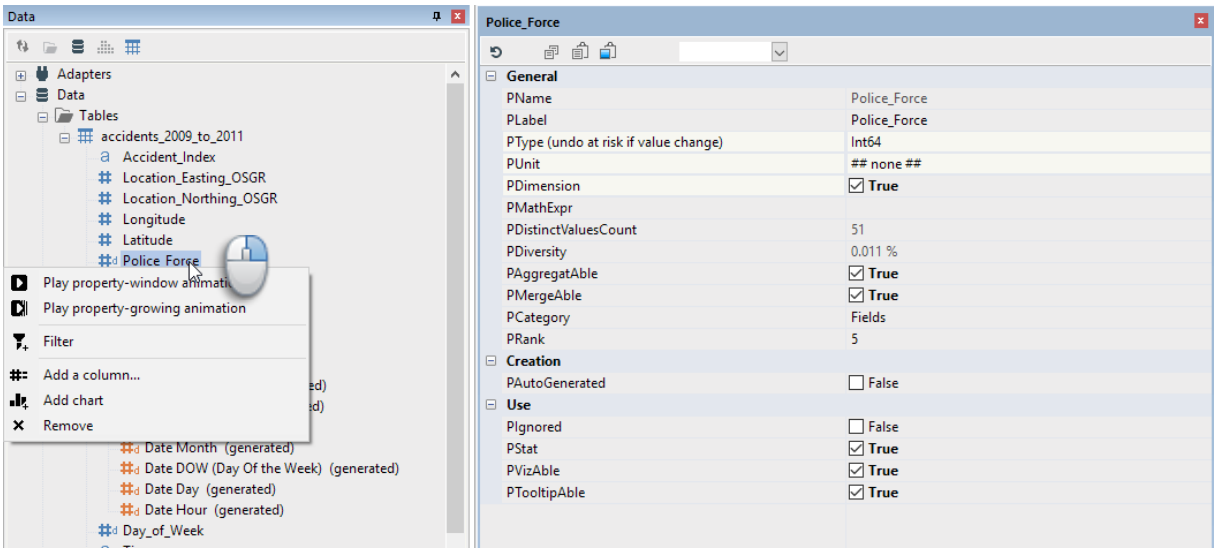


Figure 40 Filed popup menu

You can drag & drop columns directly on specific chart area, to modify X, Y or color mapping for example:

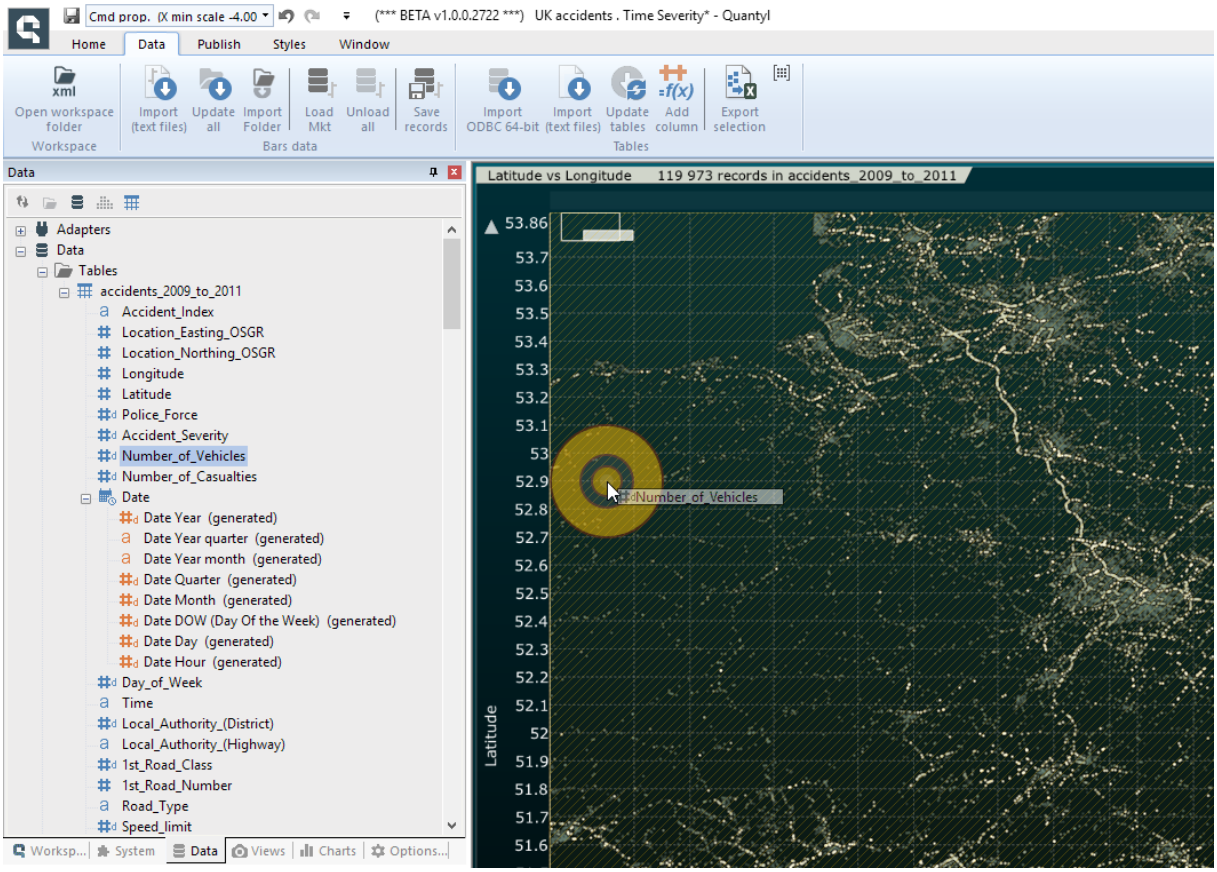


Figure 41 Table column drag & drop

8.6.3 ADDING COLUMNS (CALCULATED MATHEMATICAL EXPRESSION)

You are not limited to the columns of the table; you can also add calculated ones with mathematical expressions.

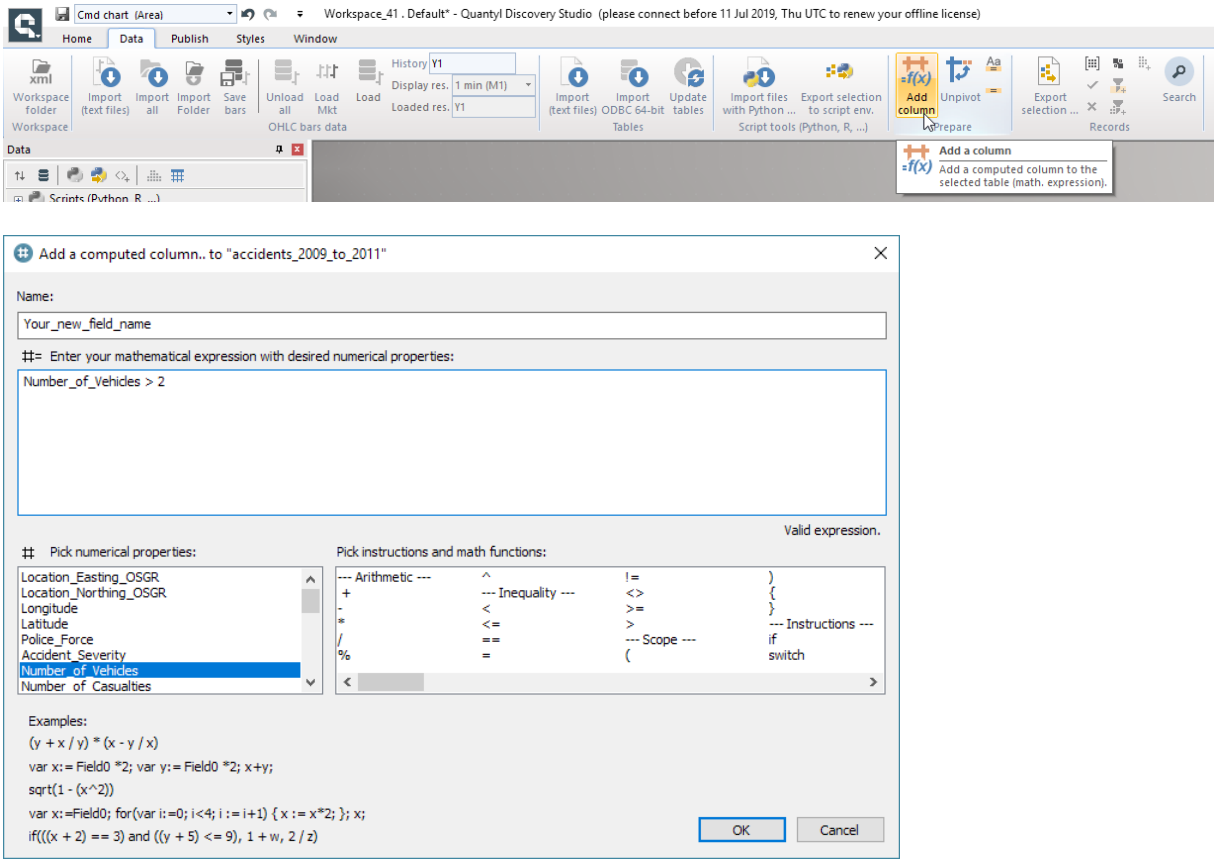
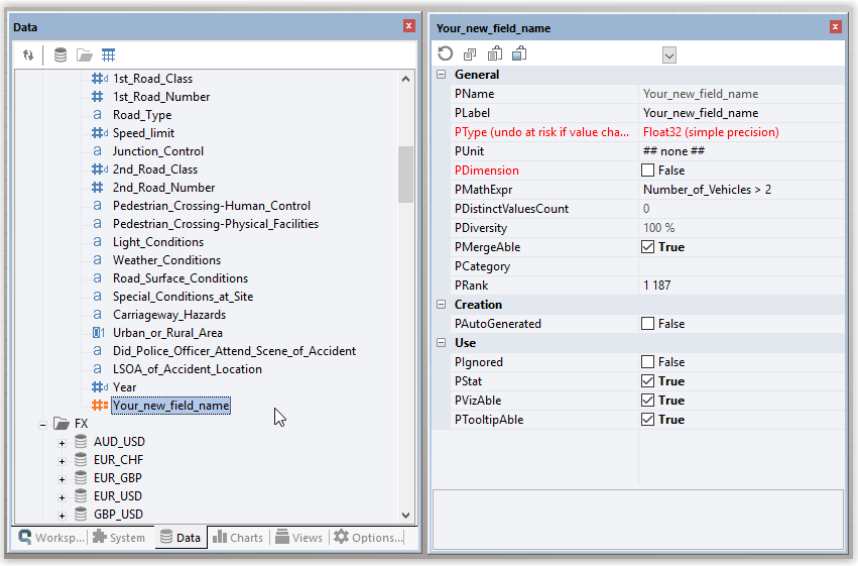


Figure 42 Add a column

Now, you can use the new column as any other measure or dimension.

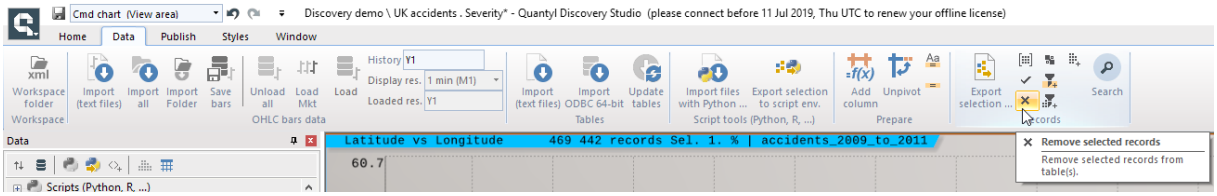


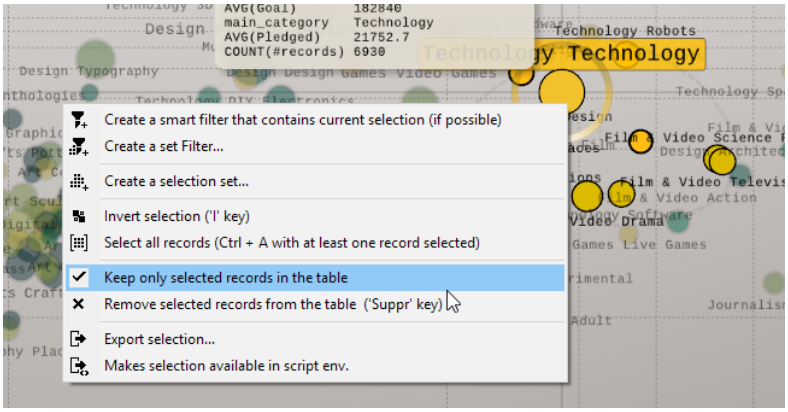
8.6.4 CLEANSING DATA

From the current selection menu (or the ribbon data category), you may **keep only** or **remove** rows from their relative tables. When you remove rows, those are physically and definitively removed from the table extract stored in the Workspace.

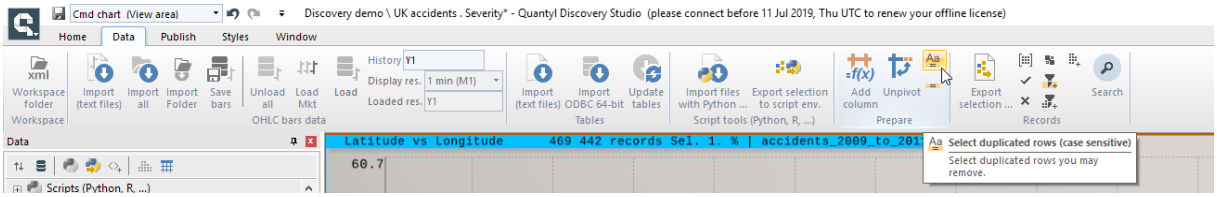
1- Use the large panel of tools to select data you want to remove
2- Keep only or Remove records (aka rows)

You can remove rows from interactive selection, search, selection sets etc.





Specific tools are provided to **identify duplicated rows** in the data ribbon category, as it is a common purpose of cleansing data. It may be case sensitive or not.

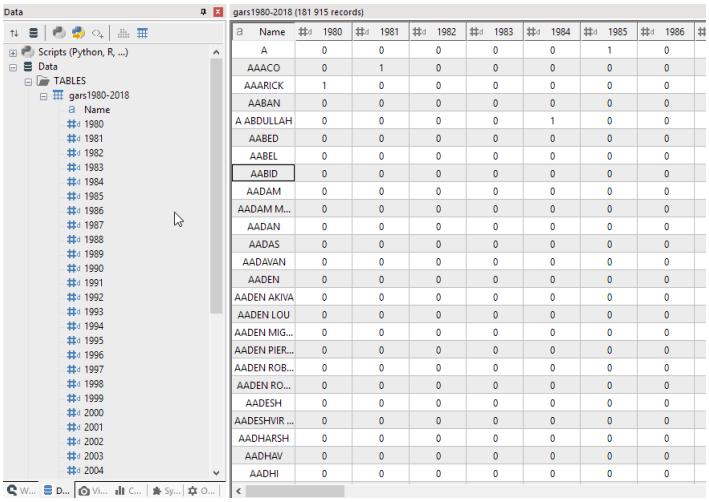


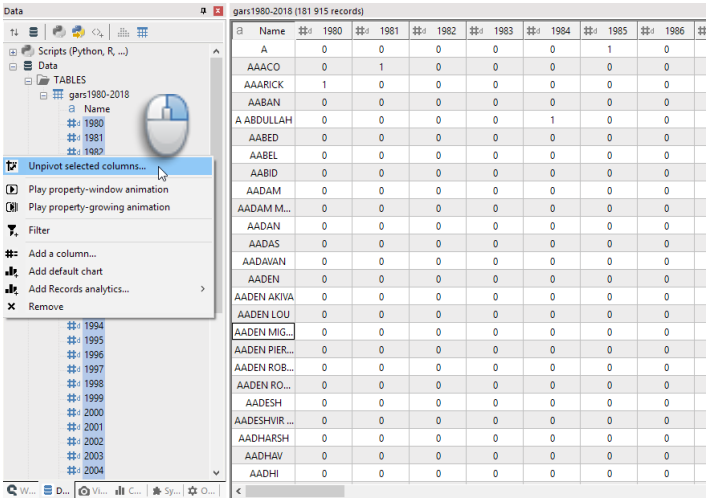
8.6.5 UNPIVOTING COLUMNS

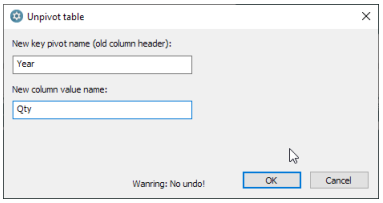
You can transform selected columns into attribute-value pairs where columns become rows.

It may be necessary to perform this transformation in scenarios where data values (dimensions or measures) have been placed as column header. In this case, field name IS the data value.

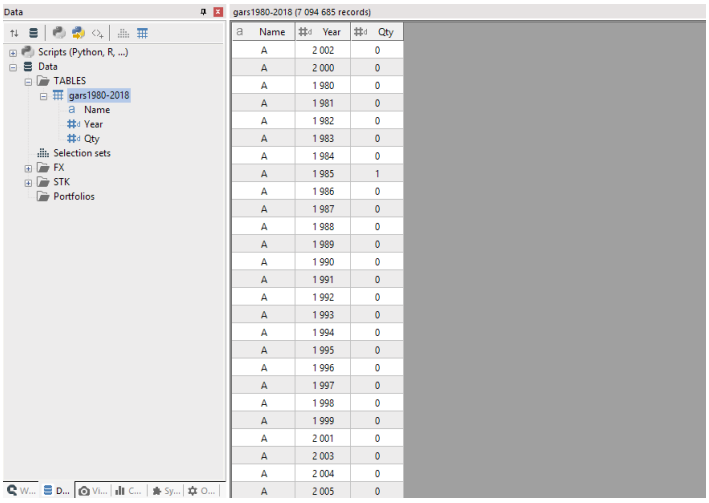
See the example below to understand how it works:



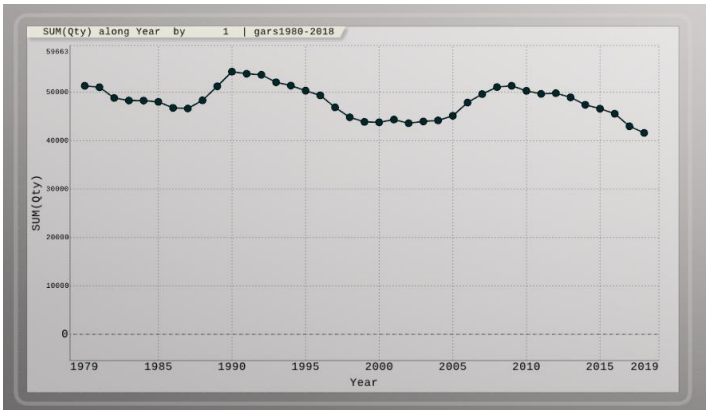




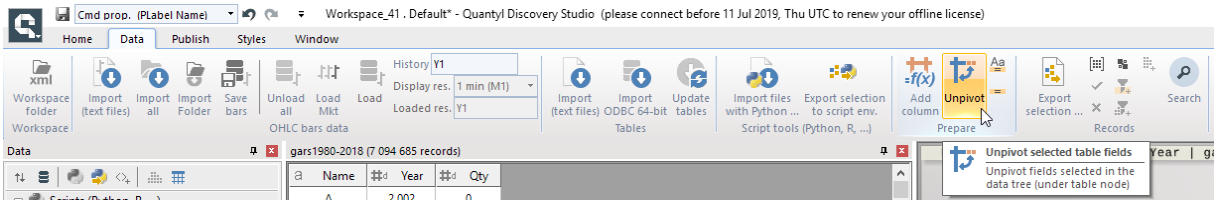
Result of the command :



Now, it is possible to build this chart (not possible before unpivoting):



Instead of using the right-click menu, you can use the command in the data ribbon:



8.7 OHLC TIMESERIES

There is no limitation in the way you navigate in your loaded OHLC bars. You can zoom and unzoom inside charts as you wish, there is no timeframe limitation linked to the OHLC bar resolution. The bar resolution of data displayed may be static or dynamic; you are never limited in your navigation by display constraints. You can either display different charts at different resolution or synchronize all charts on the same timeframe.

8.7.1 OHLC BARS IMPORT

Quantyl can work with any contracts that match **currency pair trading model** or **stock trading model**.

Contract	Description
Forex (FX)	Foreign exchange currency pairs. Symbol name contains both currencies separated with '_'
Stocks (STK)	Stocks

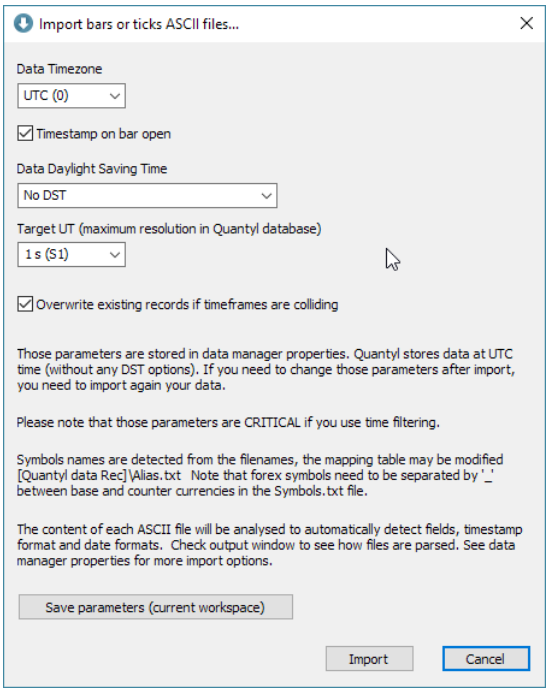
Quantyl organizes historical data in flat files **.qylrec** (records):

- Binary files loaded quickly in memory are not compressed
- Single file per contract, per month (continuous 1s data file 24h/24h for a 1-month file size is ~123MB)

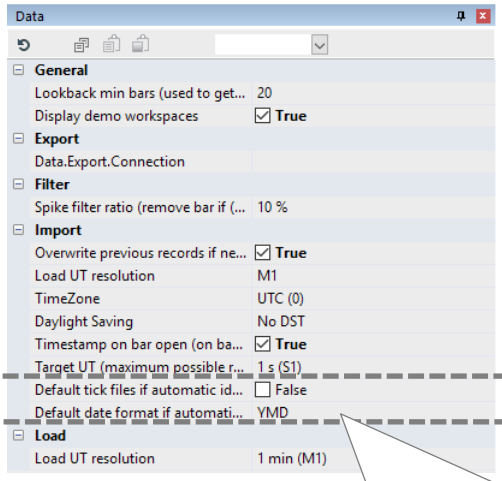
- Filename identify symbol, contract, month and year
- Single resolution chosen by the user (usually the finest: second bars), upper resolution are computed by "on the fly" down sampling. Loading is fast enough to avoid multiple resolution files on the same data.
- Different resolutions can be mixed for the same contract

Importing large ASCII files containing OHLC or ticks values is generally a painful process. Date-time formats, separators, field identification etc. are many sources of error. And if data are compiled from different sources, they may even change by contract.

Quantyl analyses files to detect automatically fields and how those fields are formatted. It is based on the study of different ASCII formats that you can find on the market: Metatrader, Metastock, NinjaTrader, Dukascopy, Pepperstone, Disktrading, Histdata, Kibot, Yahoo etc.



Data manager properties



If there is not enough data in the file to identify fields, those default settings are used

For each file, a first scan is performed to identify:

- Skip useless line (first generally)
- Separator(s)
- Fields:
  - Date or datetime
  - Time
  - OHLC prices or any LHOC, HLOC.. combination
  - Bid/Ask or Ask/Bid
  - Volumes or Bid/Ask size
  - Sort order in the file
- Date format
- Time format
- Datetime format

One assumption is that Volumes are following prices.

You can either import large tick data (in GB) or OHLC-V bars data.

The result of identification is displayed in the output window for each file.

Imported fields are the one that match unified bar structure: OHLC, spread and volumes (V1 & V2).

Simple rules are the following ones:

1 Name of the contract is extracted from the filename
2 **Unknown currency pairs need to be separated by a '_' in the filename.**
3 Automatic fields detection simplifies a lot the process, and let user to set some minimal parameters that cannot be automatically deducted from the file

Tick size is deducted from loaded data.

When you import files, symbol name is deducted from the filename with the following process:

1. File extension is ignored
2. Numerical characters are ignored
3. Filename is translated in uppercase
4. **Alias.txt** file is used to match name with or without '_' string and replace pieces of strings in the filename
5. Known symbols (coming from **Contracts.xml dictionary**) are tested against filtered filename to find potential match

8.7.2 DATA MODEL

Quantyl works on a “unified” bar model, used for ticks and bars:

- Bar finest resolution is **1µs.** A bar can represent a single tick (O=H=L=C)
- Bar time open (**TO**) and bar time close (**TC**) are stored in the bar
- **OHLC prices**
- **Maximum spread** during the bar timeframe is stored into the bar
- **V1 & V2 Volumes.** They may represent bid and ask size, or down and up ticks, as well as volume traded (V2 = 0)

Quantyl supports non-uniform and non-continuous bar timeframe in data series (but APIs like SMA are not time weighted). It saves a lot of memory for non-traded periods.

8.7.3 FLOATING PRECISION & MEMORY FOOTPRINT

Prices, volumes and spreads are stored in **32bits simple floating** format (7 digits precision).

Bar size in memory is **56 Bytes:**

- **2 * 8 Bytes for timestamps**
- **8 * 4 Bytes for prices, spread and volumes**
- **8 Bytes to store a pointer on data bar extension**

Theoretical memory footprint is **~19 Million bars / GBytes** (without samples).

32 bits float values are only used in bars for obvious memory reasons; Computing (indicators etc.) is perform in 64-bits.

All **API** are **64-bits double floating precisions (even bar interfaces).**

9 WORKING WITH CHARTS

This section describes how to work with any kind of chart in a general manner. This section does not provide deep details on the different charts and their specific features.

9.1 DESIGN MODE

Design mode allows you to transform charts as you desire: move, scale, align borders etc. In design mode, handles and alignment lines are displayed when mouse is moved over it.

Important

Design mode is a key mode to know when you work with Quantyl user Interface. Design mode state is saved at user level (not attached to the workspace).

Design mode does not forbid you to launch simulation and work as you want; but to prevent any transformation or handle display, or boring highlighting in some cases, this mode can be disabled.

Design mode is enabled by default (**bottom right of the app**).

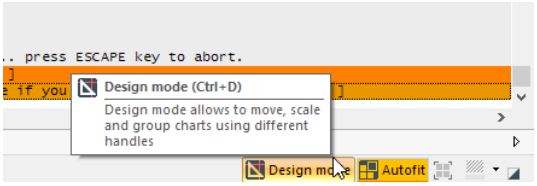


Figure 43 Design mode

In **Design mode,** charts display different handles or lines when mouse is moved over them; they allow different behaviors through simple dragging:

- Chart move
- Chart resize
- Multiple borders dragging (single or multiple charts)

When a chart is moved on a top of another one, its layer property is automatically updated to be displayed on top.

9.2 CREATING AND DUPLICATING CHARTS

Charts may be created implicitly or explicitly:

- When you ask to show a specific chart type from ribbon “home”
- When you ask to add a specific chart type from ribbon “home”
- When you drag & drop charts from the chart library or a chart tree
- When you duplicate charts (**Ctrl-C/Ctrl-V** or Duplicate menu, drag a chart with Shift pressed)
- When you create charts from a menu (table, existing chart, viewport, see below)
- When you group charts (group creations)
- When you launch a Faceting command

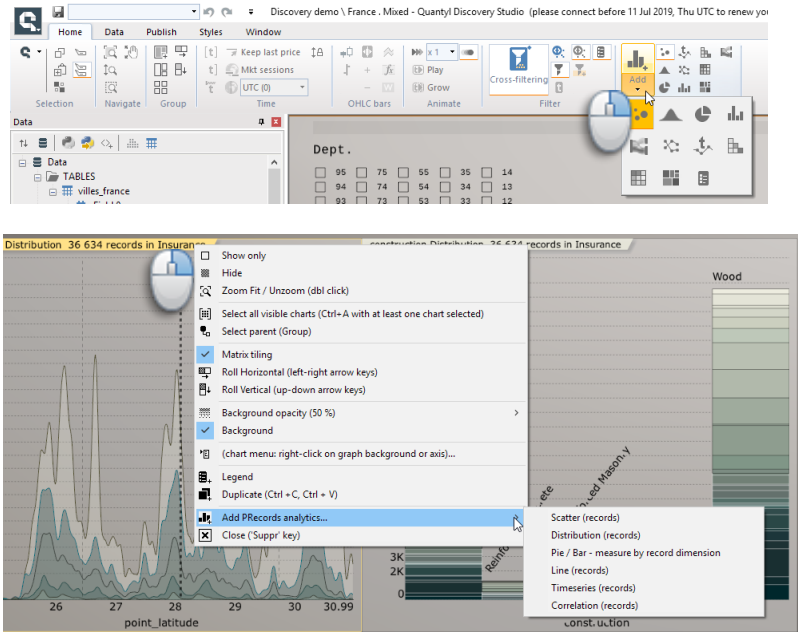


Figure 44 Create chart from a chart menu (properties of the current chart are used)

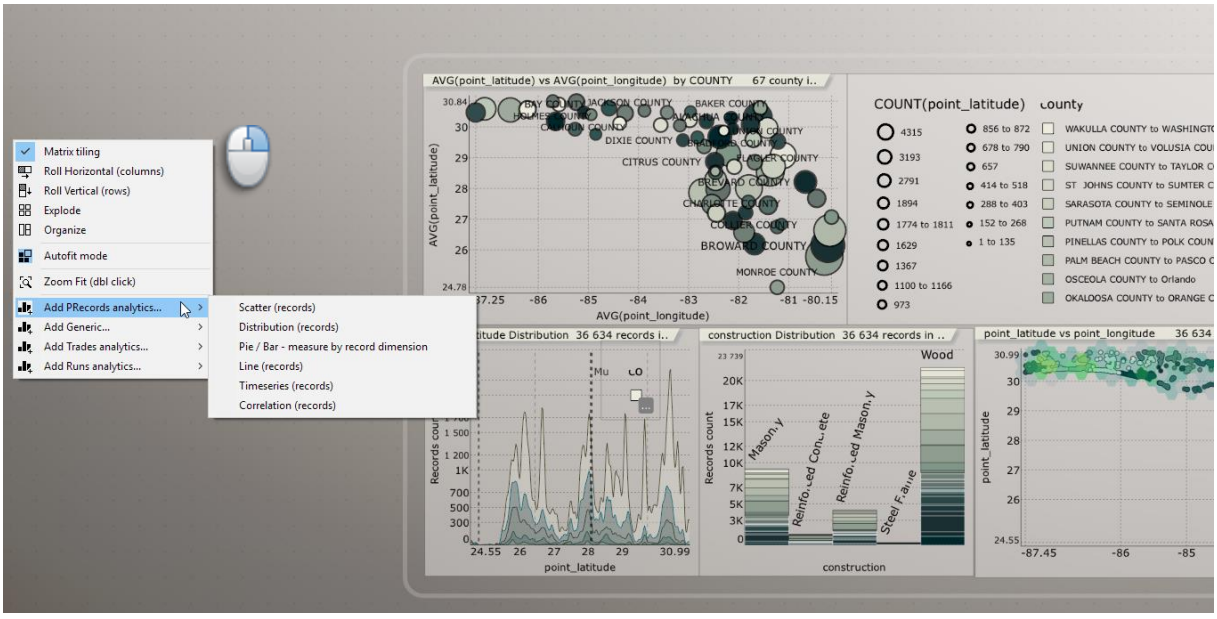


Figure 45 Create charts from viewport menu

Charts are **automatically** structured in the charts tree depending on how you are organizing them in the viewport (see Charts grouping section).

Select the charts and press **Delete**/**Suppr** key or use “close” in the title menu to remove a chart. You can also directly close charts in the tree with same right-click menu, or pressing **delete** or **suppr** key.

There are at least 4 ways to copy a selection of charts. If multiple charts are selected, the complete charts selection is copied:

1. Press Ctrl key and drag an existing one from its title handle or move handle
2. Drag & drop it from the tree to the view
3. Ctrl + C, Ctrl V (only valid in the same workspace)
4. Use “Duplicate” command from chart menu.

Important

You can **copy and paste** charts **between different views** in the **same workspace**.

9.3 CHARTS LIBRARY AND CHARTS TEMPLATES

When a chart is created, it is always initialized from an existing one. This chart may be an existing chart in the current view (duplication) or a chart template in the library. A chart template, or chart default, is simply an instance of the chart type in the charts library.

Context	Used template (default properties)
Duplicate	Chart source
From scratch (ex nihilo)	Instance of the chart type in the library

The properties of the different chart types stored in the **chart library** are shared by all workspaces. You can define default properties (aka templates) for any chart type: contract charts, analytics charts, indicators etc. When you apply style through style ribbon category, you may choose if you want to modify charts default.

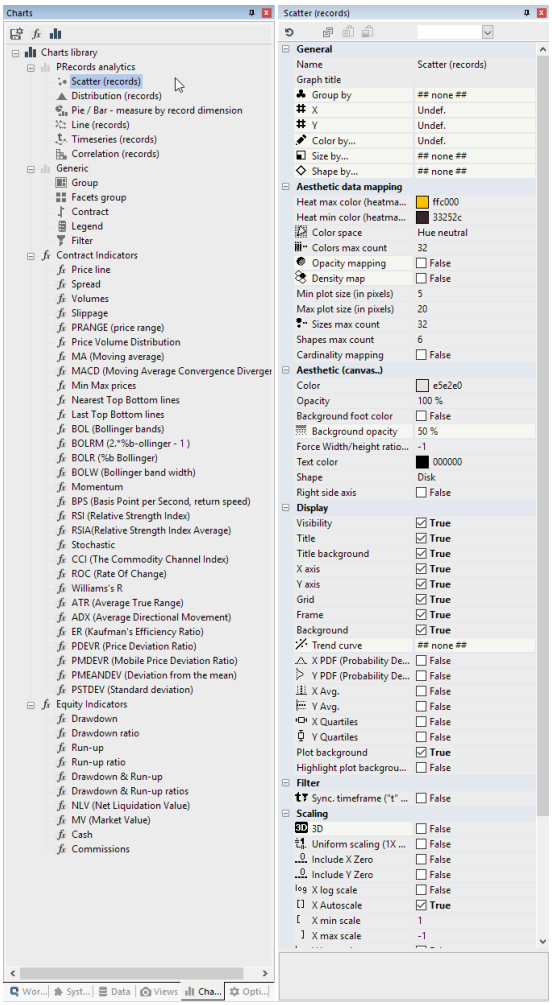


Figure 46 Chart library (default)

Important

When you drag & drop a chart from the tree library on a chart in the viewport, the new chart copy properties of the existing chart (color mapping etc.).

9.4 CHART AREAS AND POPUP MENUS

Chart has multiple sensitive areas dedicated to interactive behaviors. You can drag or right click the highlighted zones to perform specific behaviors. We make a distinction between the chart (the whole object) and the graphical representations (in the graph area of the chart).

Important

If multiple charts are selected, the commands of the contextual menu are applied to the selection.

Visibility and some other basics commands are available from the right click menu in the chart title bar as in the capture below:

Figure 47 Title area

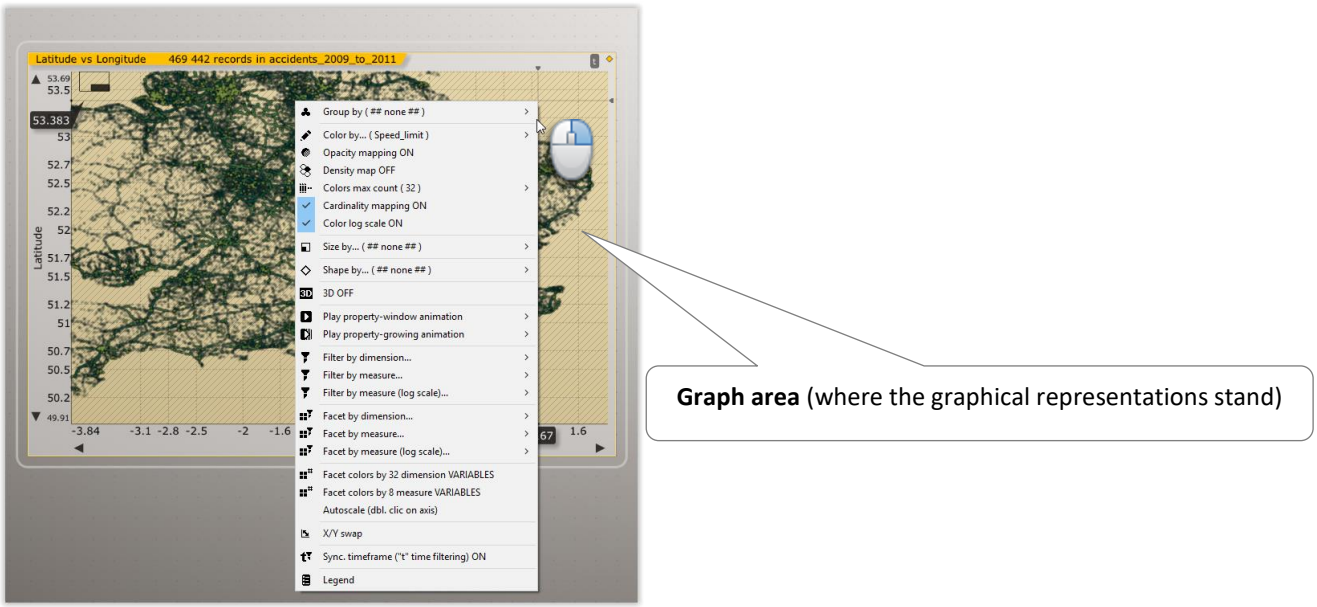


Figure 48 Graph area with contextual menu

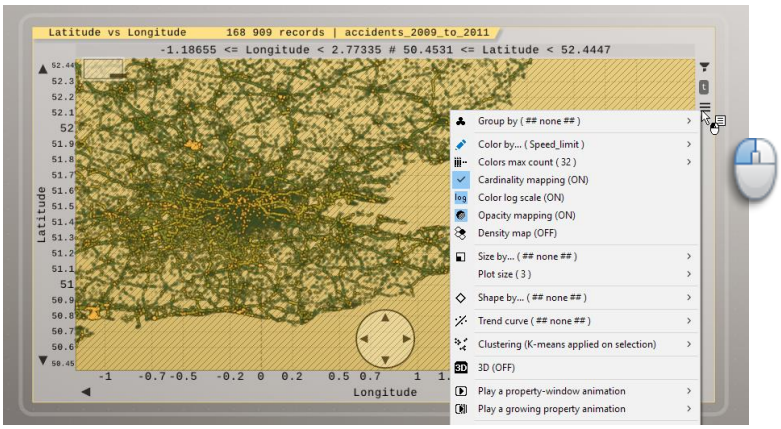


Figure 49 Graph area direct menu

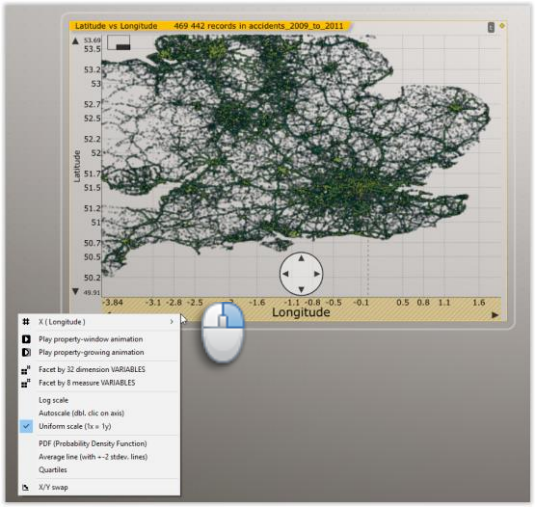


Figure 50 X axis area with contextual menu

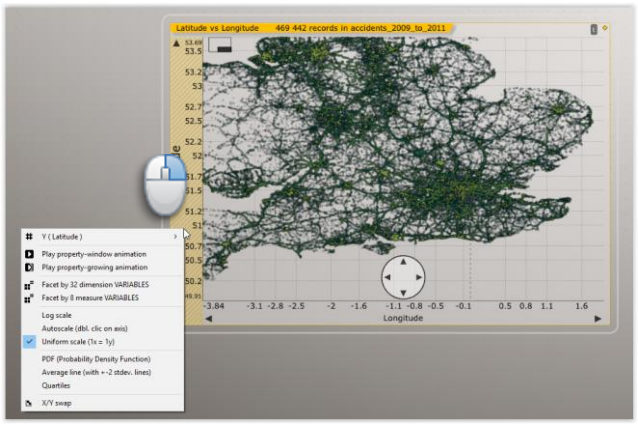


Figure 51 Y axis area with contextual menu

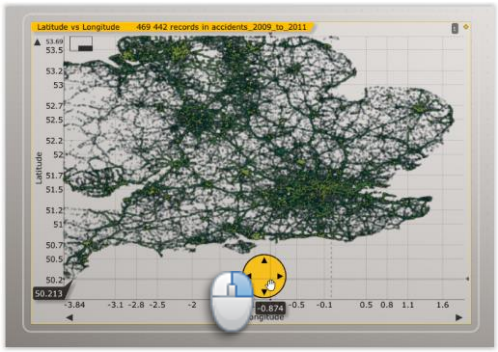
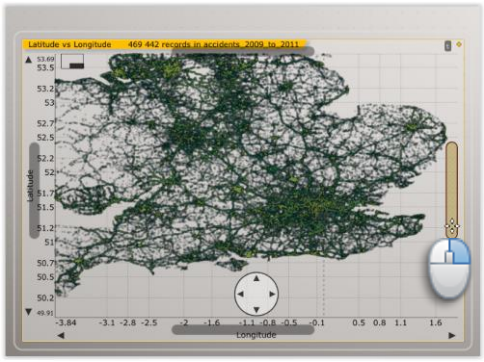


Figure 52 Scaling area (pan inside the graph) Double-click to zoom fit

In **design mode only**, some handles appear when you mouse moves over them:



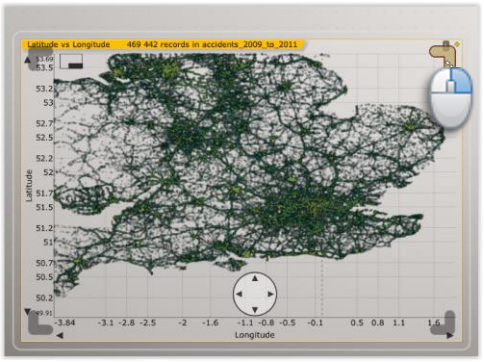


Figure 53 Resizing handles (design mode) **Figure 54 translation handles (design mode)**

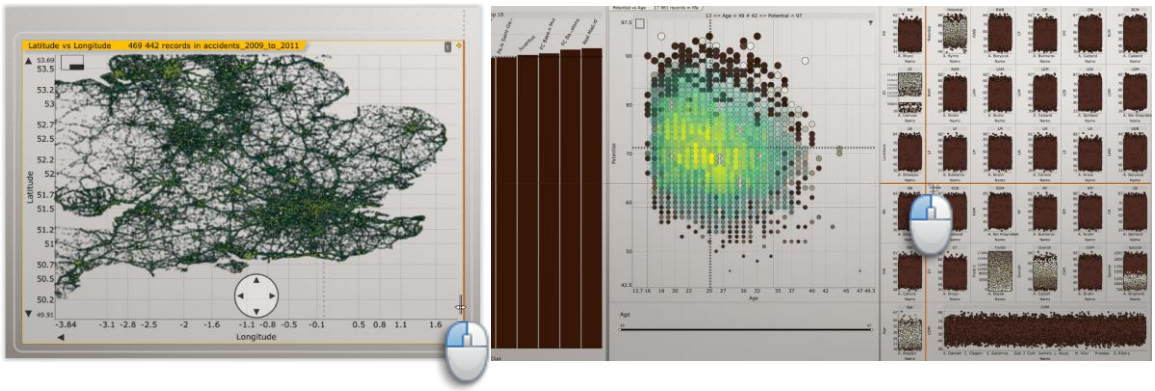


Figure 55 Borders drag lines (design mode)

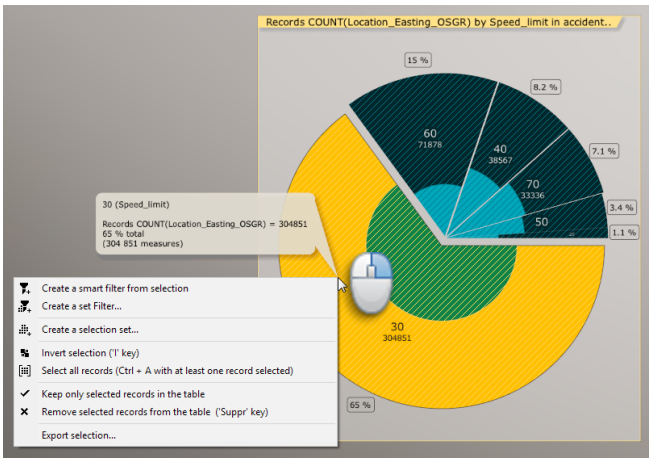
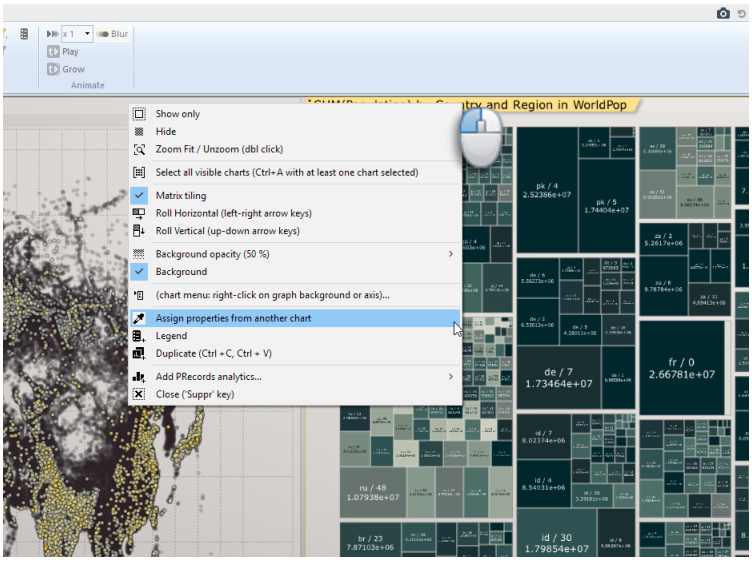
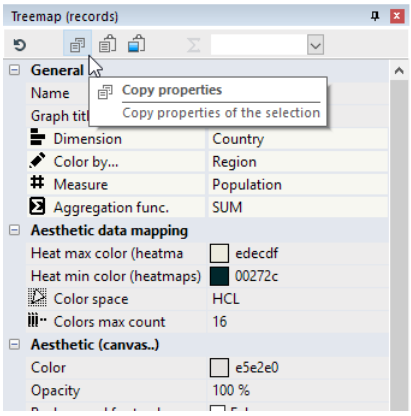


Figure 56 Selection area and selection menu

9.5 COPYING PROPERTIES BETWEEN CHARTS

You can use the chart title menu to copy properties from another chart, or use copy / paste command in the property page toolbar:





9.6 NAVIGATION AND SELECTION

Please, see Navigation & selection commands at the beginning of the document.

9.7 MULTI-TOUCH SUPPORT

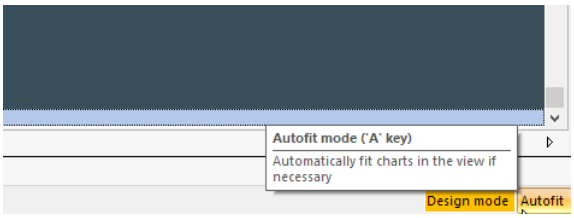
Multi-touch (1 or 2 fingers) screen is fully supported in charts area (viewport). It is not a simple emulation of mouse behaviors; specific support of multi-touch has been implemented:

Specific touch feature	Command
Single tap (slow)	Open contextual menu on object selected
Zoom 2 fingers (pinch / spread)	Zoom in the selected chart

9.8 AUTOFIT MODE

In autofit mode, charts are automatically tiled when a new chart is added (new contract loaded) or charts become visible (after a simulation, equity curve may be displayed).

View captures autofit property as well as other chart manager properties. If you want to prevent a refit at load of the view, you have to disable this mode in the view.

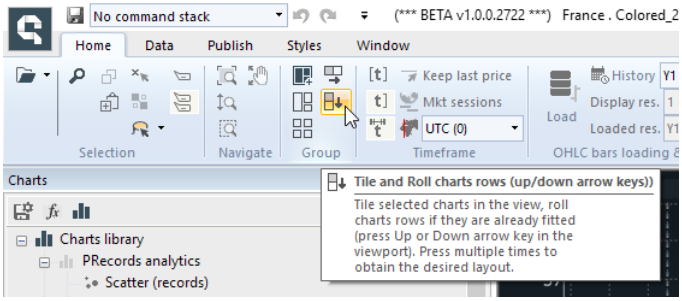


Note: group charts have also their own autofit property (see group section).

9.9 TILING AND ROLLING

It is possible to tile and roll selected charts in their respective groups and in the view with simple commands. If no charts are selected, only “root” charts are tiled and rolled. Root charts are charts without parents (see Charts grouping section).

Charts may be tiled and rolled on X and Y with those 2 simple commands, that you can use multiple times to obtain the desired layout:



You can also use arrows keys to tile & roll your selection in the different directions.

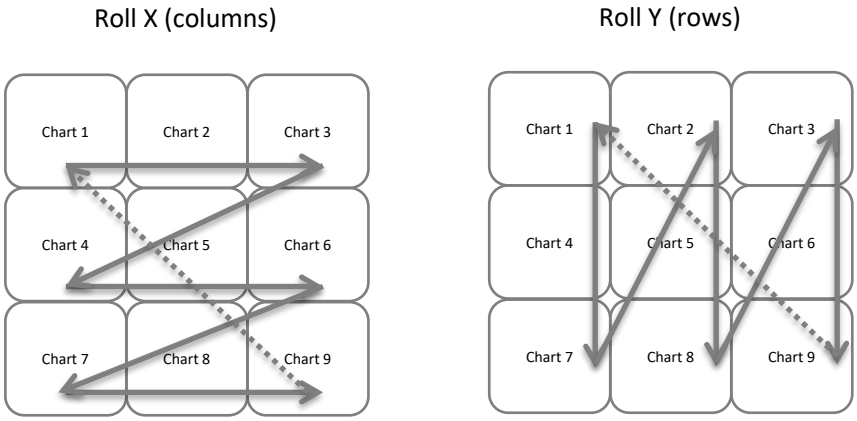


Figure 57 Tiling and rolling

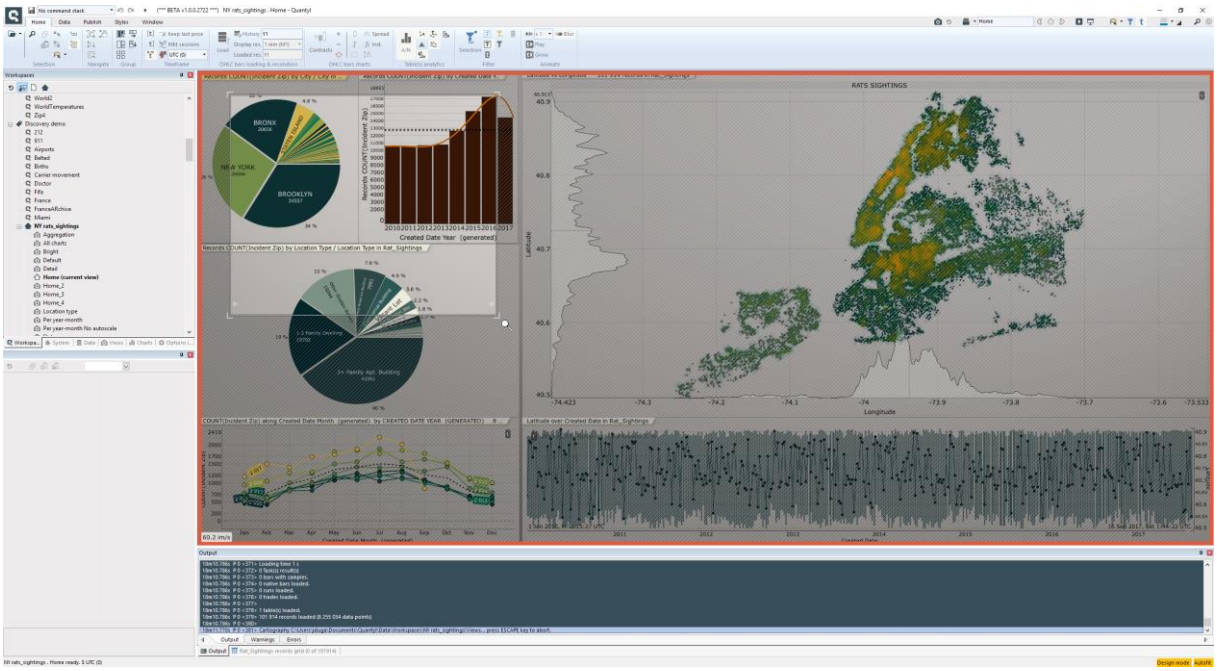
9.10 ZUI: ZOOM IN / OUT IN THE VIEW

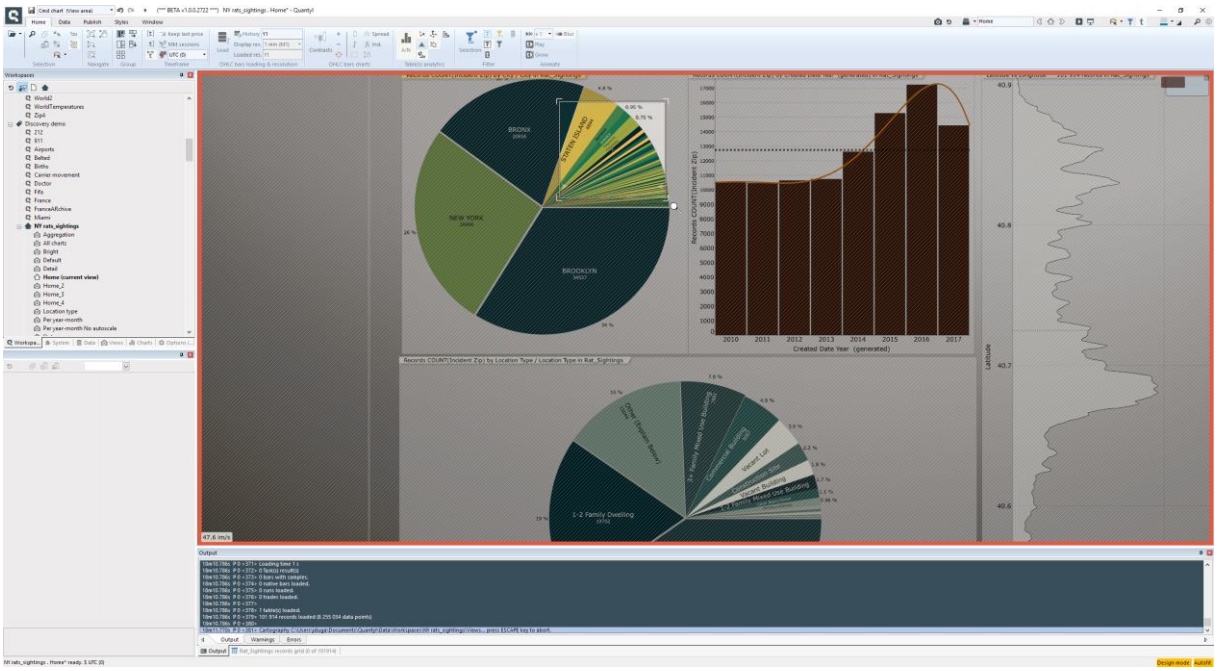
The viewport, the area where charts are displayed, is a **true virtual window** on the charts. It means that you can navigate (pan and zoom) in the viewport itself without decreasing quality of the charts. It is a fully **zoomable user interface** (ZUI).

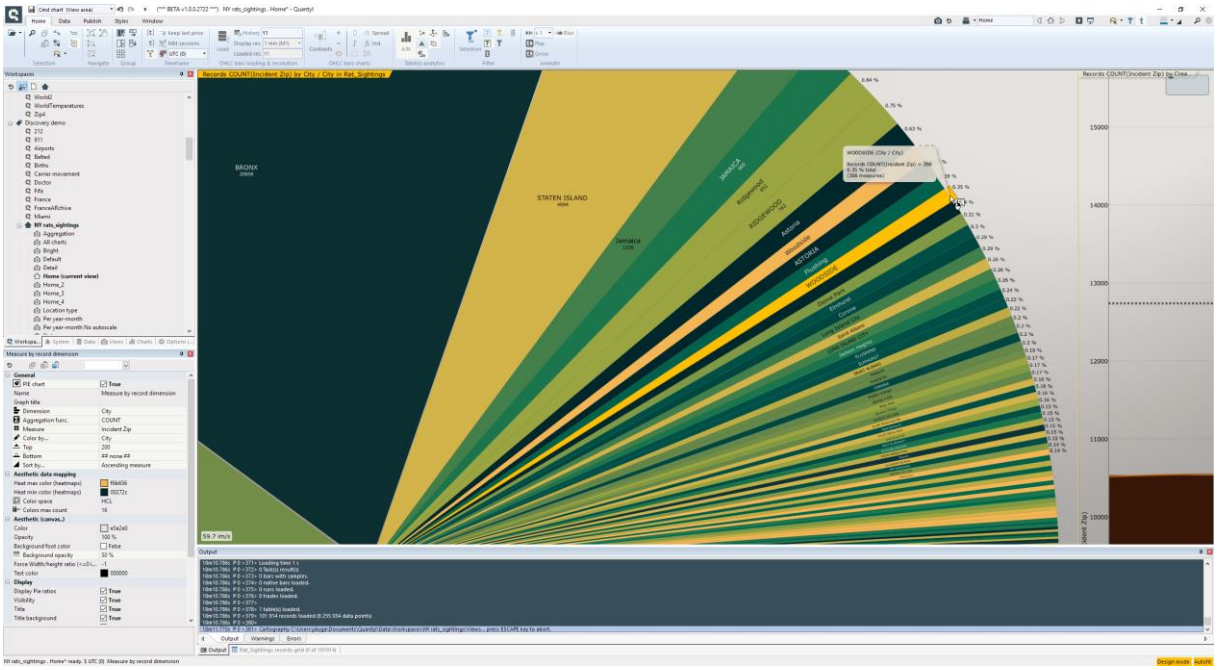
Imagine a board with all charts pinned on it, you can zoom on a portion of this board, pan your position, refit all your charts etc. It seems complex to explain but very intuitive to experiment:

- **Double click** on the chart fits the view to chart if it is not already fitted. Double click again on the same chart and view fits all visible charts
- **Press Ctrl key during mouse dragging** to “pan” the view
- **Press Ctrl key during mouse wheel scrolling** to “zoom” in or “zoom” out the view
- **Press arrows keys** at any time to refit and roll from the selected charts

It is a true virtual chart environment that you can customize, save and invoke as you wish with **views**.







Double click to zoom fit all.

9.11 CHART TRANSFORMATIONS

Enable Design mode, then:

- Drag & drop a “move” handle to translate chart(s)
- Drag & drop a “resizing” handle to resize chart
- Drag line(s) to align border or move multiple borders together.

9.12 GROUPING, DOCKING AND TILING CHARTS

Each chart (except indicators) has the capability to accept other charts as child. When a chart is a child, it follows all transformations made to its parent (move, resize). Parent-child link relationship creates a branch. There are no limits to the number of charts in a branch. **Root charts** are charts without parent.

Charts can be grouped **implicitly** or **explicitly**.

Implicit grouping: Simply drag a chart on another one; it must be in the entire area of the targeted parent to create the link. To break the relation, child chart may be dragged outside its parent (on another parent or on the background).

When you are dragging a chart, you may also see “target glyph” display at the center of underlying chart or groups. Use them to dock chart on a specific place. It creates automatically groups. There is no limit on the deep of the hierarchy you are creating. You can put groups into groups etc., creating multiple level hierarchy of charts.

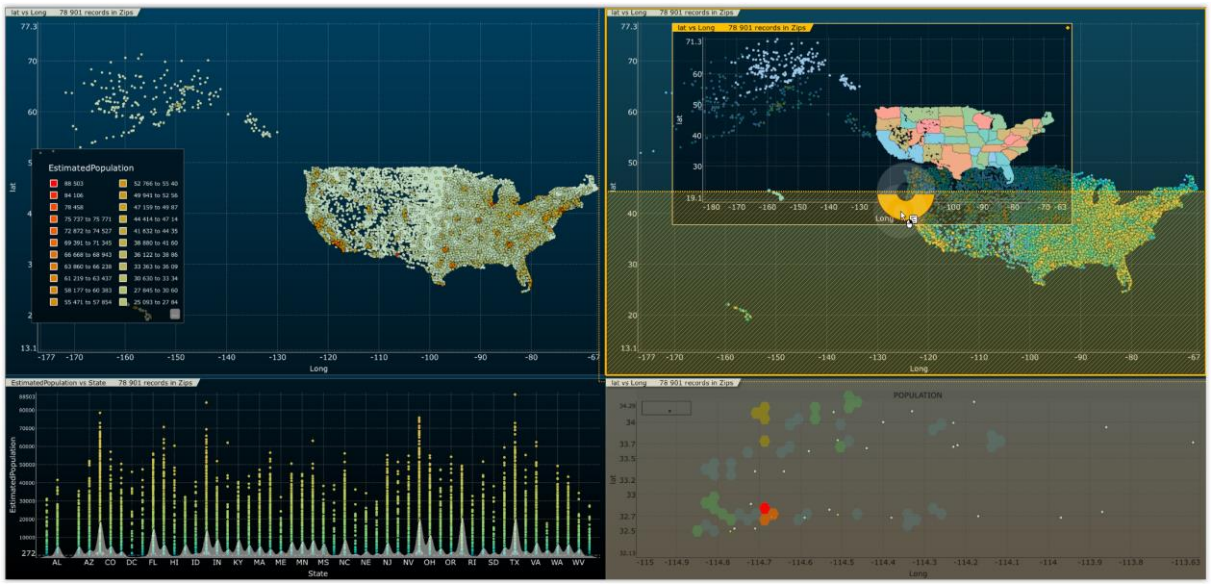


Figure 58 Group target display when dragging

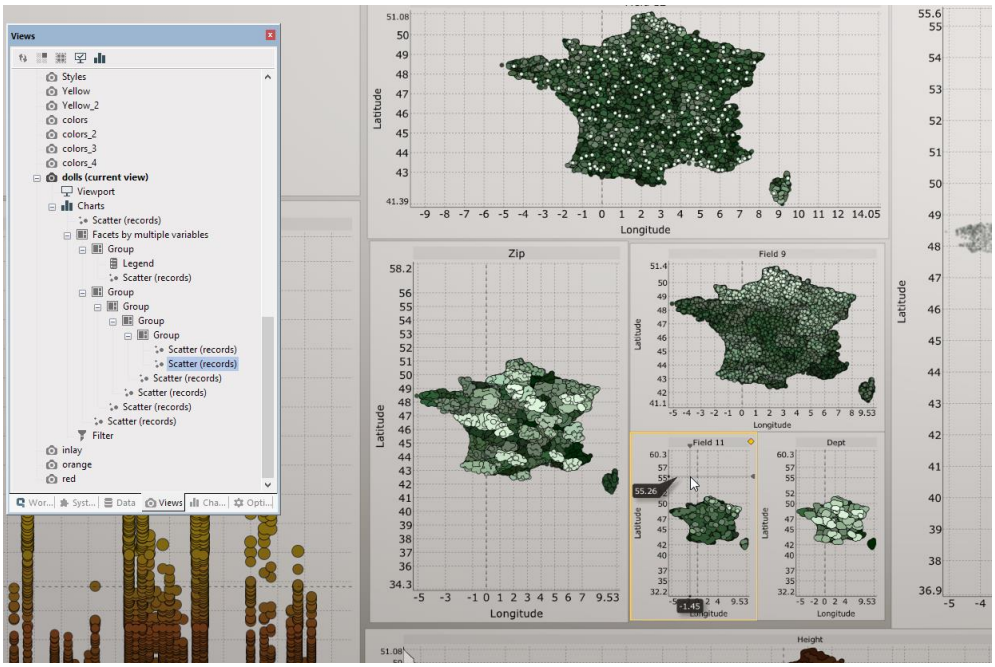


Figure 59 Multiple level hierarchy

Explicit grouping :

- Create an empty group and drag charts inside it
- or select charts you want to group and launch the group chart command
- or use Group command(s) from the right click menu on a chart title (multiple charts need to be selected first):

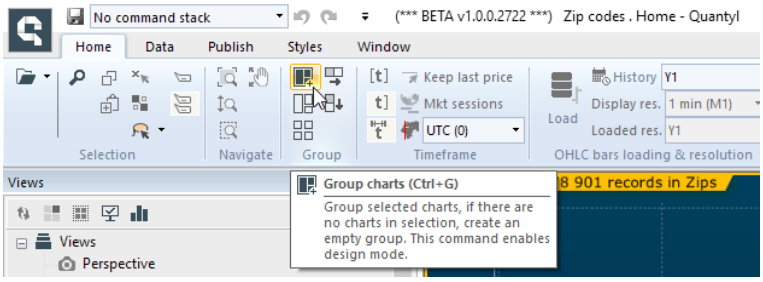


Figure 60 Explicit group command

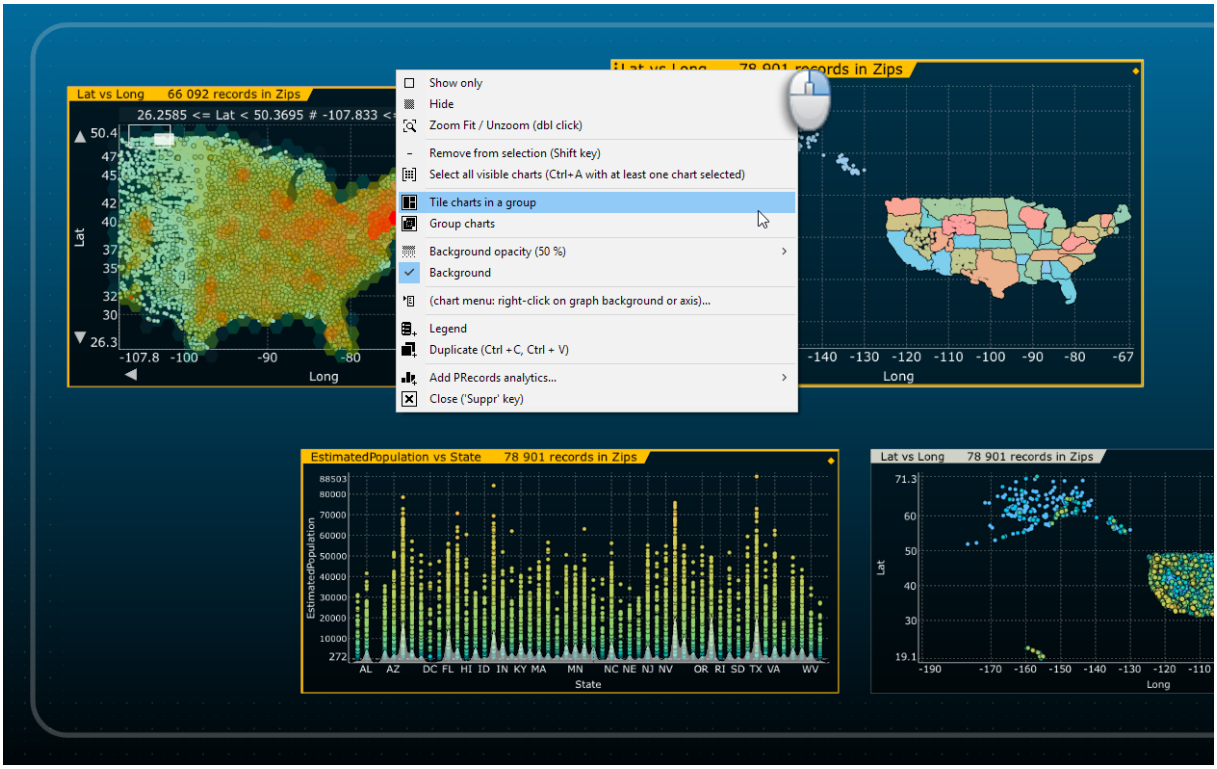


Figure 61 Multiple chart selection, grouping command

When a group becomes useless (childs deleted or removed, single child etc.), group is **automatically** deleted. Of course, you can select and remove group manually as any other chart.

Group charts have their handles closer the border to facilitate selection of the group. Groups are the key to organize charts exactly as you wish.

Press mouse right button to open the roll menu that allows you to tile and roll charts inside the group.

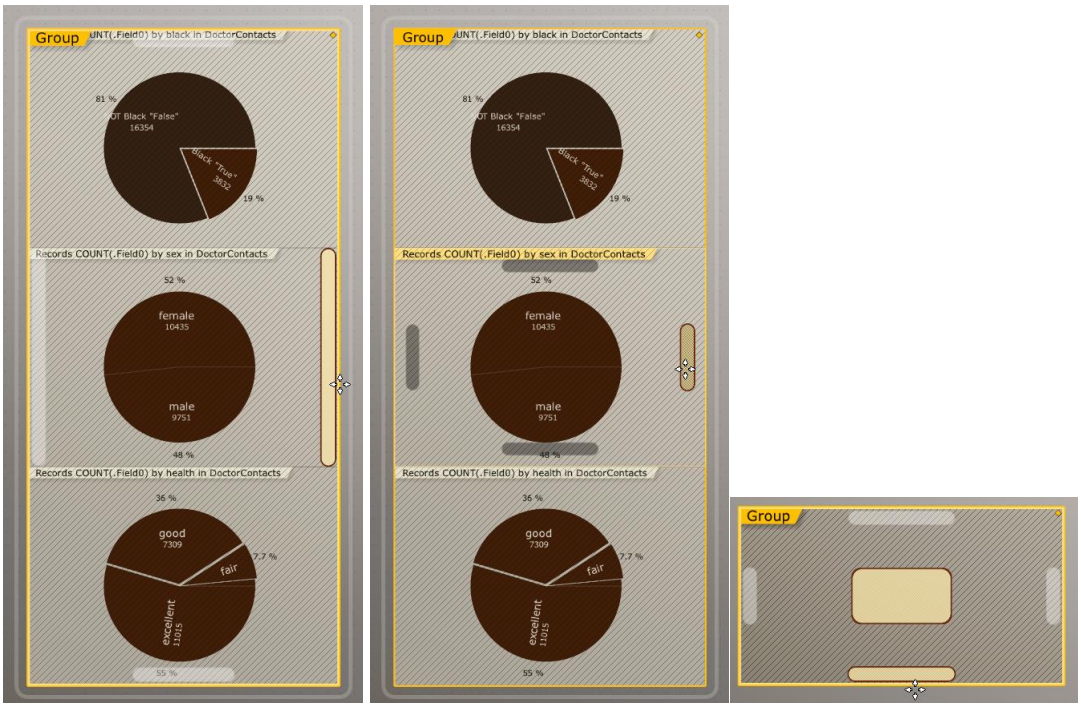


Figure 62 Group handles are closer the borders

Important

Group handles are closer the border.

By default, groups are in **autofit mode**. The way charts are automatically organized in this mode is called **tiling**. You may decide to have rows and columns, a single row, or a single column.

Rolling command "roll" charts in the desired direction. Experiment the different commands to find the desired result. Create sub-group (group inside groups) to achieve any structure you like.

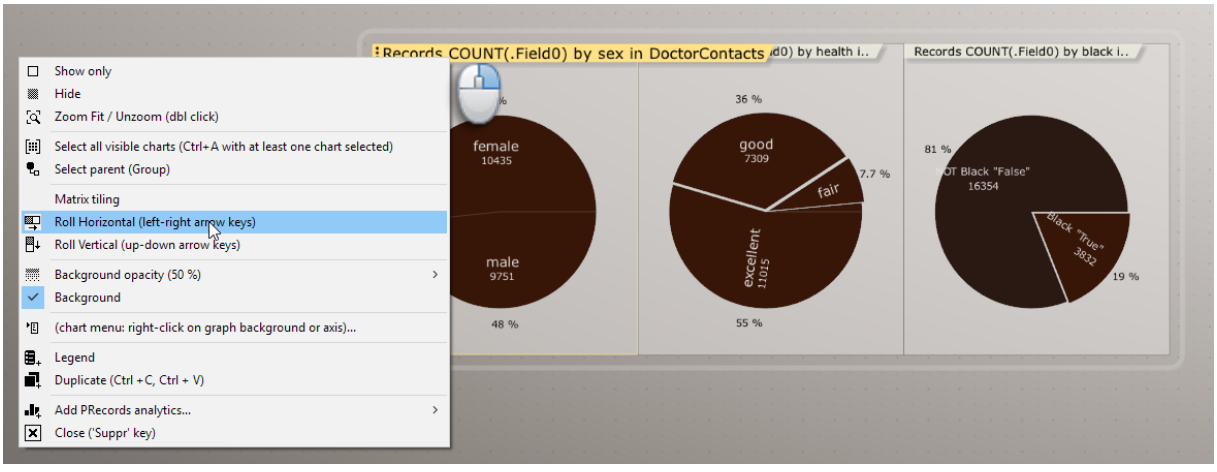
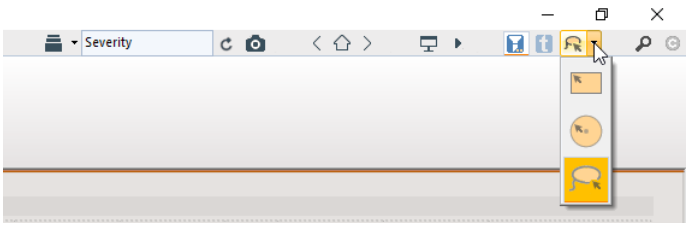


Figure 63 Tiling and rolling

9.13 SELECTION MODES (CONTOURS)

Scatter charts have multiple type of selection, the selection type is saved with the view. Those 3 selections support right to left dragging additive selection, and support **+Ctrl** and **+Shift** key too.



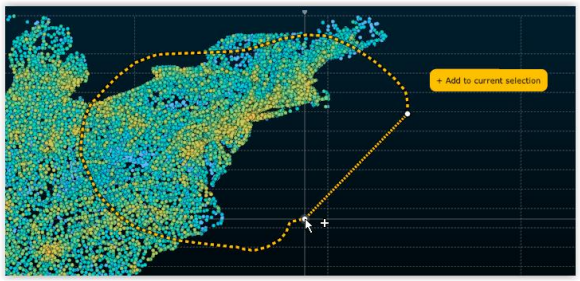


Figure 64 Lasso selection

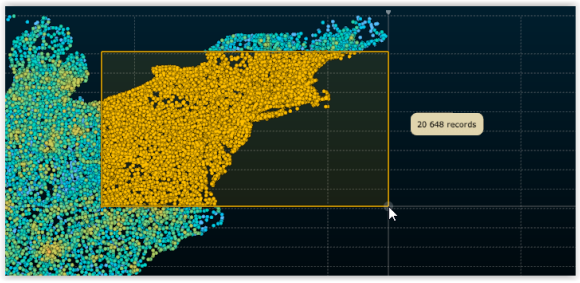


Figure 65 Box selection

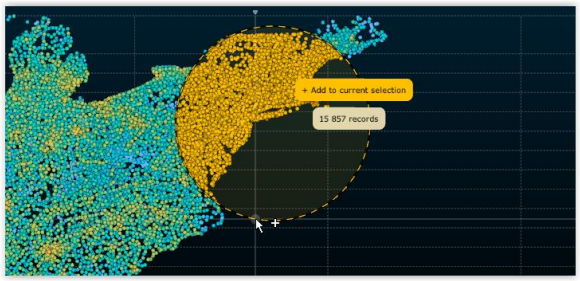


Figure 66 Circle selection

Important

Right to Left and Counter Clock Wise lasso are additive selection (previous selection is kept in the new one)

Important

Last selection contour is draggable in most of the charts

Important

When an aggregate is selected, underlying records are selected.

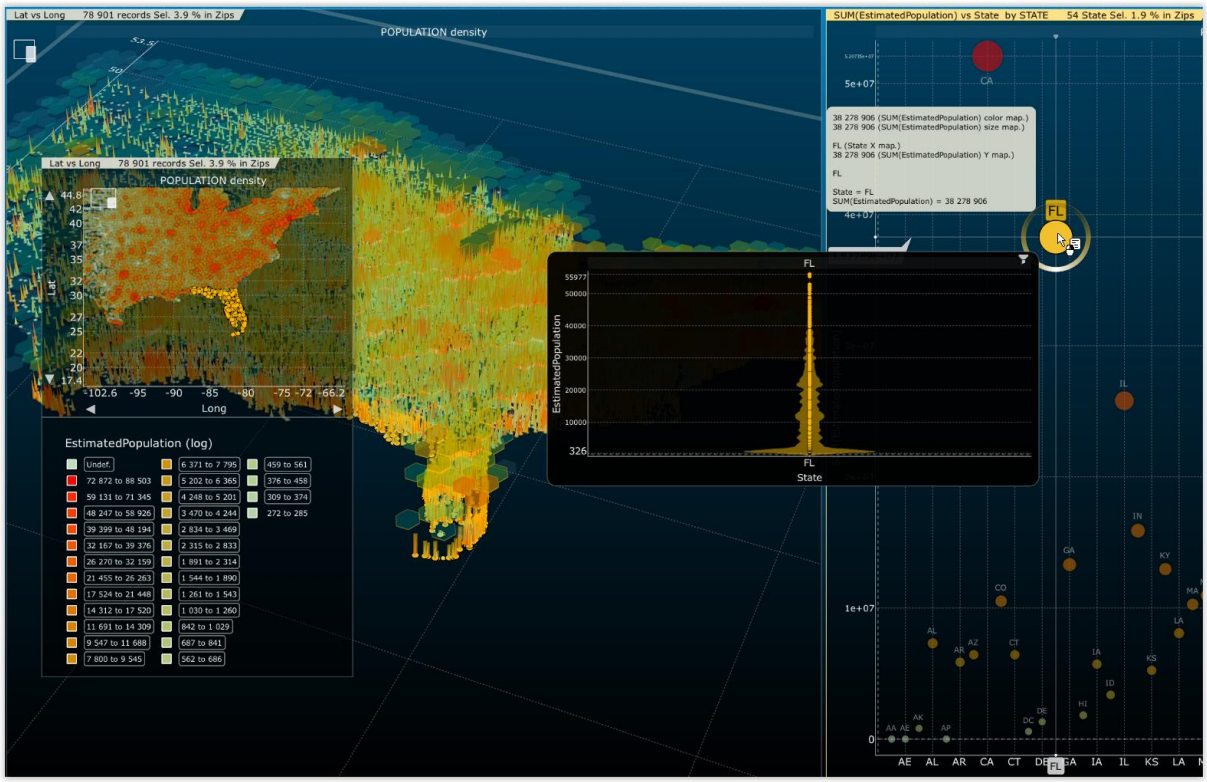


Figure 67 Aggregate selection

In 3D scatter plot chart, default behavior of the mouse left button is camera rotation. To temporary switch to selection mode, you need to select one of the selection available (rectangle, disk or lasso).

9.14 SELECTION BRUSHING

When you have performed a selection rectangle, lasso or disk in a scatter chart, the contour of this selection remains active to be dragged after the command. This feature is available for scatter, distribution and line charts. This is interesting to combine selection dragging with dynamic selection filtering and use the selection contour as a brush tool.

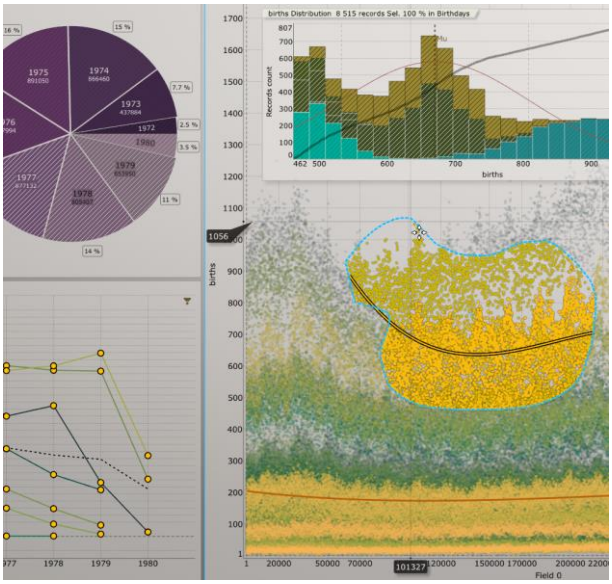


Figure 68 Selection brush dragging

9.15 SELECTION SETS: CAPTURING SELECTIONS

Selection set capture the current selection (right mouse menu on selection). It is basically a set of records.

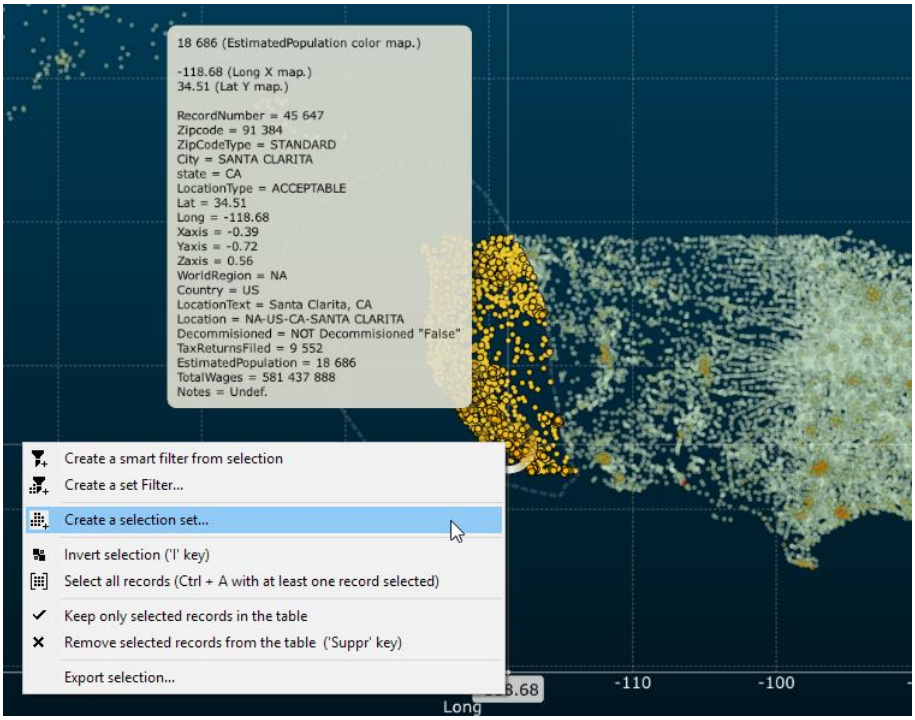
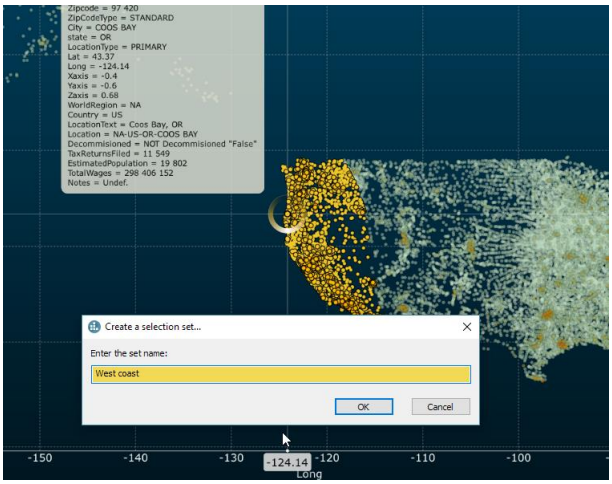
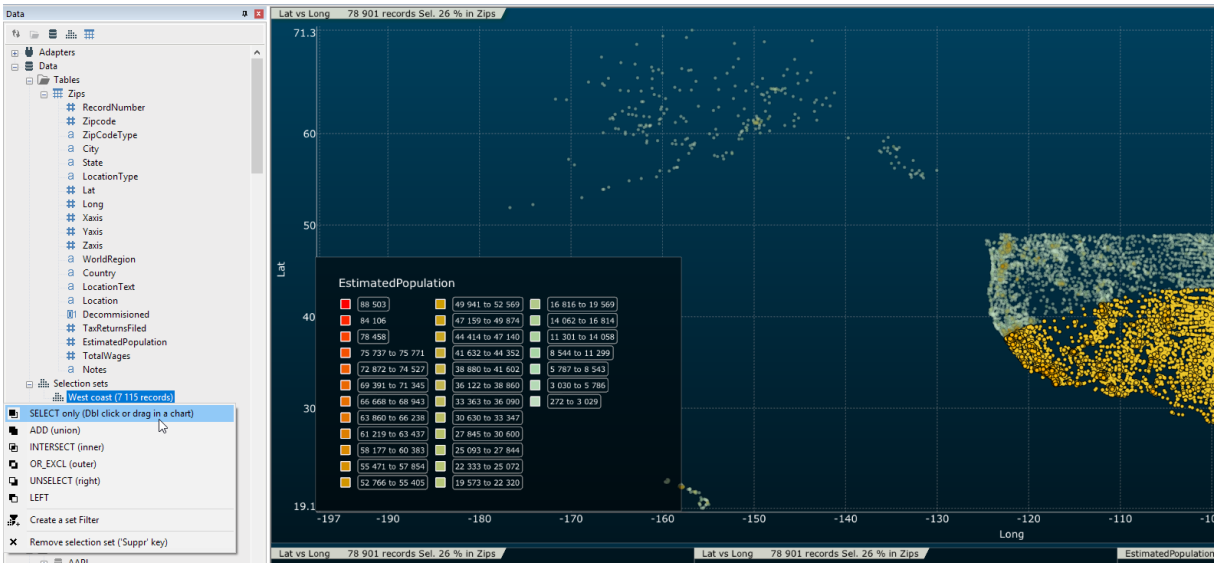


Figure 69 Creating a selection set





You can save as many selections as you wish, they are saved with the workspace. You can then invoke them or perform Boolean operation against your current selection.

Important

Selection sets reference each record, they do not store the command that performed the selection. If you need to capture the selection command, use filters or smart filters instead.

9.16 CHARTS LIST

As we have seen in fundamentals, aggregation is generally performed through dedicated contextual commands. This reduces the number of charts, because each chart is more flexible and polyvalent, thanks to their aggregating capabilities and both dimensions and measure acceptance.

For example, scatter charts accept dimension or measure in both X or Y axis, and it's possible to group data through any dimension. At any time, you can switch to a 3D view, or add density map. Line charts accept both dimensions and measure as X values etc.

Depending of their different properties and their nature (dimension, measure), charts graphs will adapt to something that makes sense to you, grouping by dimension if necessary, displaying violin histogram or box plots along the same dimension if possible (scatter chart). Combinate the different possibilities as you wish to obtain the desired insights.

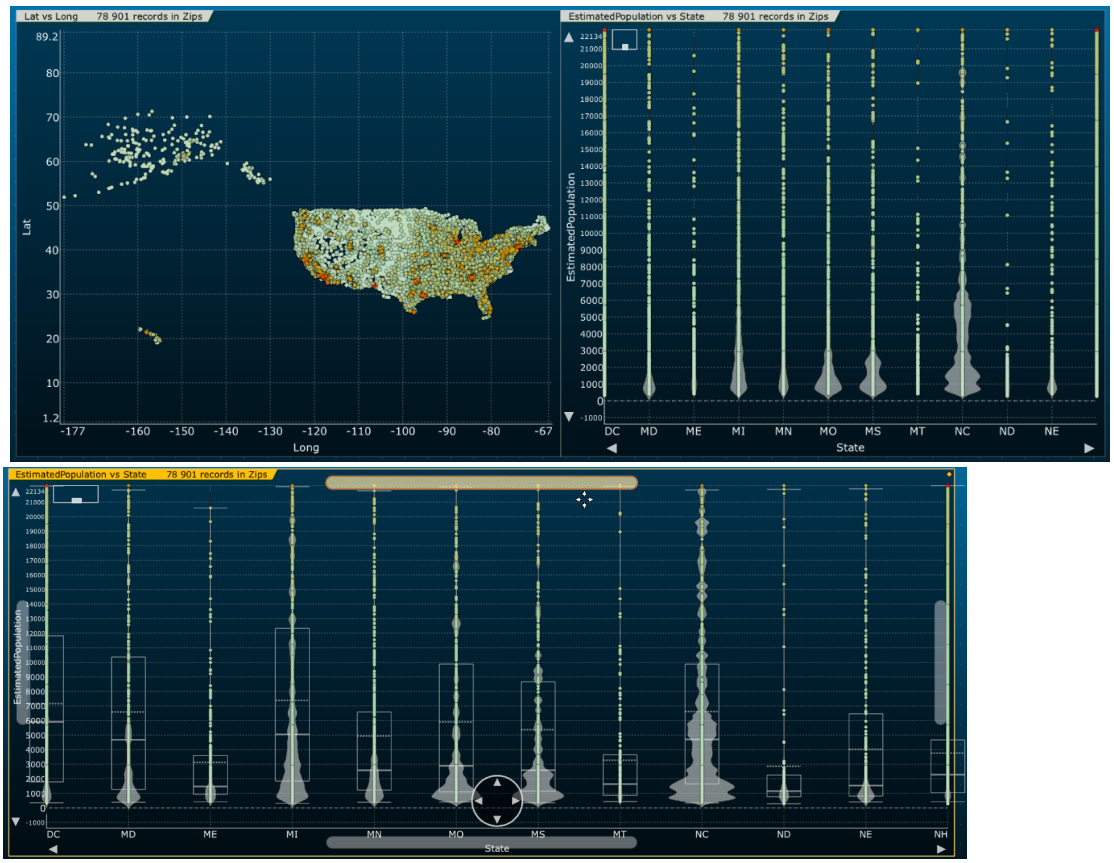
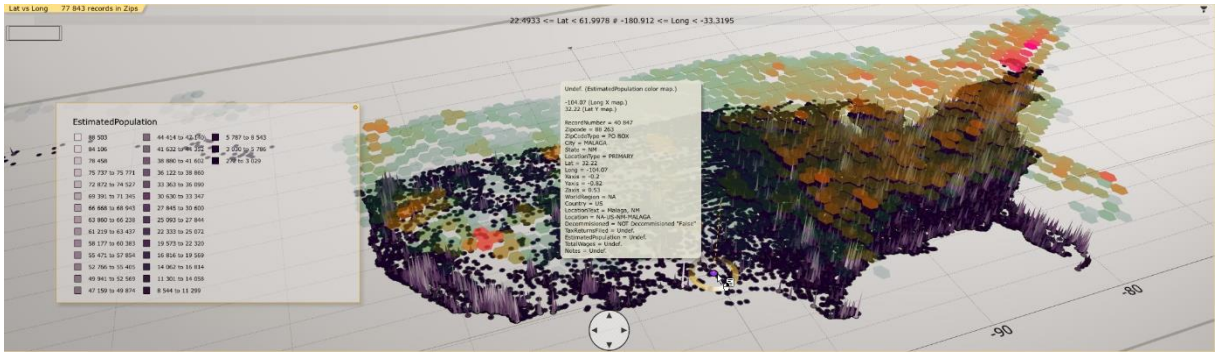


Figure 70 Same scatter plot chart with different X / Y parameters



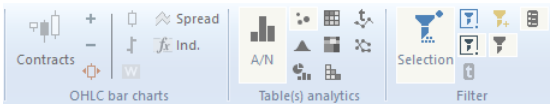
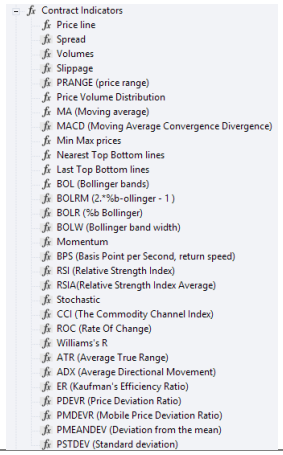


Figure 71 Charts

Figure 72 Chart library (that can be extended through plug'ins)

Charts table	
Chart type	*Main Features (sub-graphs declination)*
Scatter	• Scatter plots • Bubbles (group by) • Density map • 3D (2D + height based on color mapping) • Violins (dimension as axis) • Boxplots (dimension as axis) • PDF histograms (on axis) • Averages lines (on axis) • Quartiles (on axis) • Trend line (polynomial)
Distribution	• Stacked or not • Dimension or measure as variable • PDF and normal • CDF • Quartiles
By (Pie or Bars)	• Pie chart • Bar chart single dimension • Bar chart with main and secondary dimension (color by) • Grouped bar charts • Compare relative % selection between different slices of a pie • Sorting by measure or dimensions • Tops and Bottoms filtering
Line	• Multi-lines (color by) • Measure / measure • Measure / dimensions with aggregation • Trend line (polynomial)
Timeline	• Multiple time series (color by)
Heatmap	• Heatmap between any measures or dimensions • 2D aggregation bins (2D histogram) • 3D heatmap (height as aggregation measure)
Treemap	• Treemap with 1 or 2 dimensions and measure aggregation
Sankey	• Sankey diagram (up to 7 dimensions)
Correlation	• Measures correlation table
Contract	• Bars • Candlesticks • Spread • Market hours

	• Multiple indicators available
Indicators	• Multiple contracts indicators available 
Group	• Rolling and tiling • Matrix, row or column structure
Legend	• Legend any chart(s) • Scoping
Filter	• Filter dimension(s) or measure(s) on any chart • Scoping
Group	• Group charts and organize them in matrix, row or column
Facet Group	• Charts placed in a facet group are sharing the same scaling

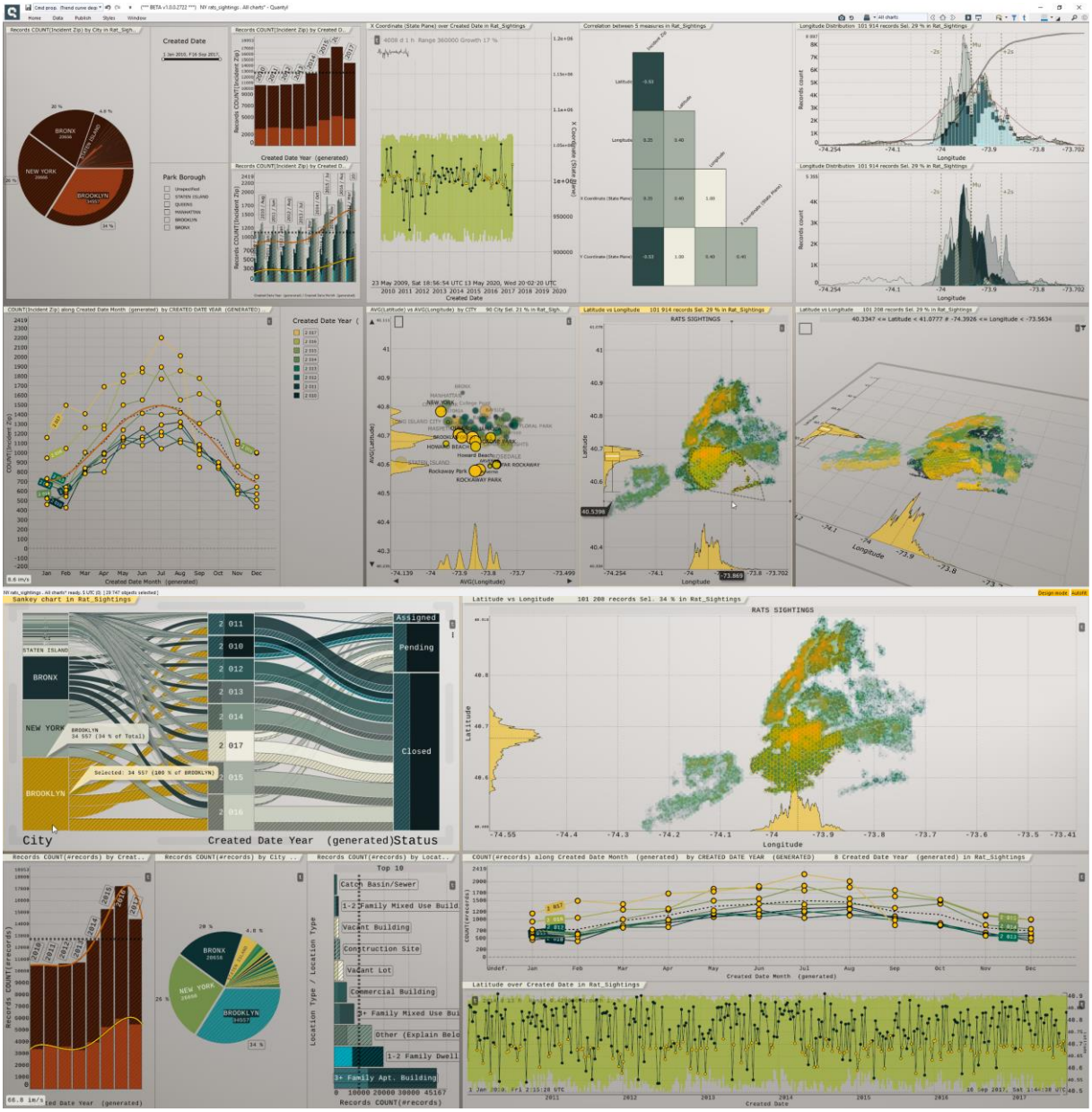


Figure 73 Different type of charts and parameters based on the same data table

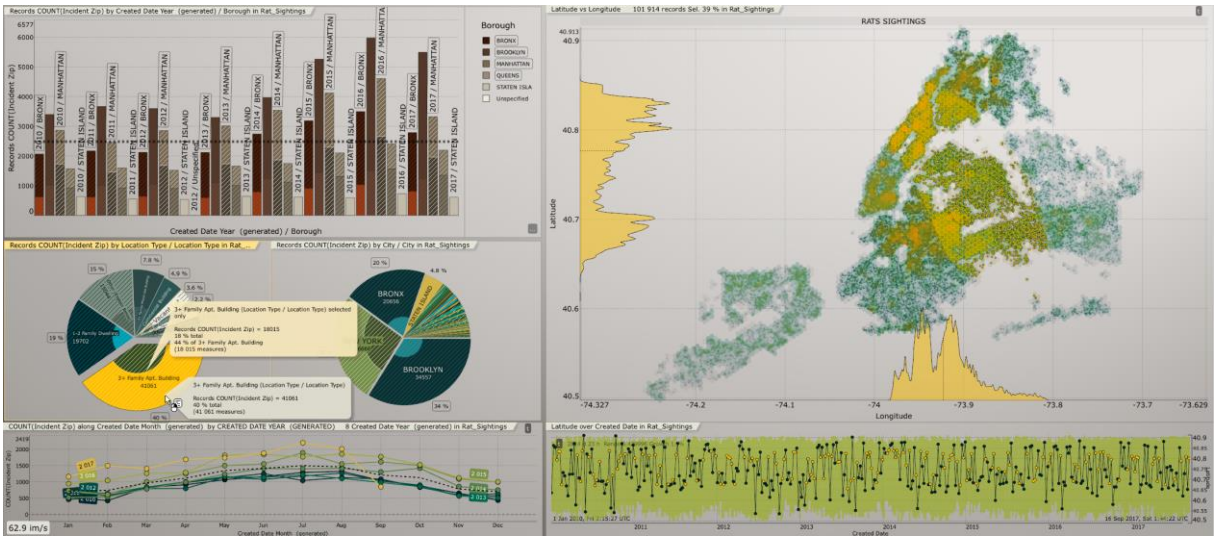


Figure 74 Relative % of the selection

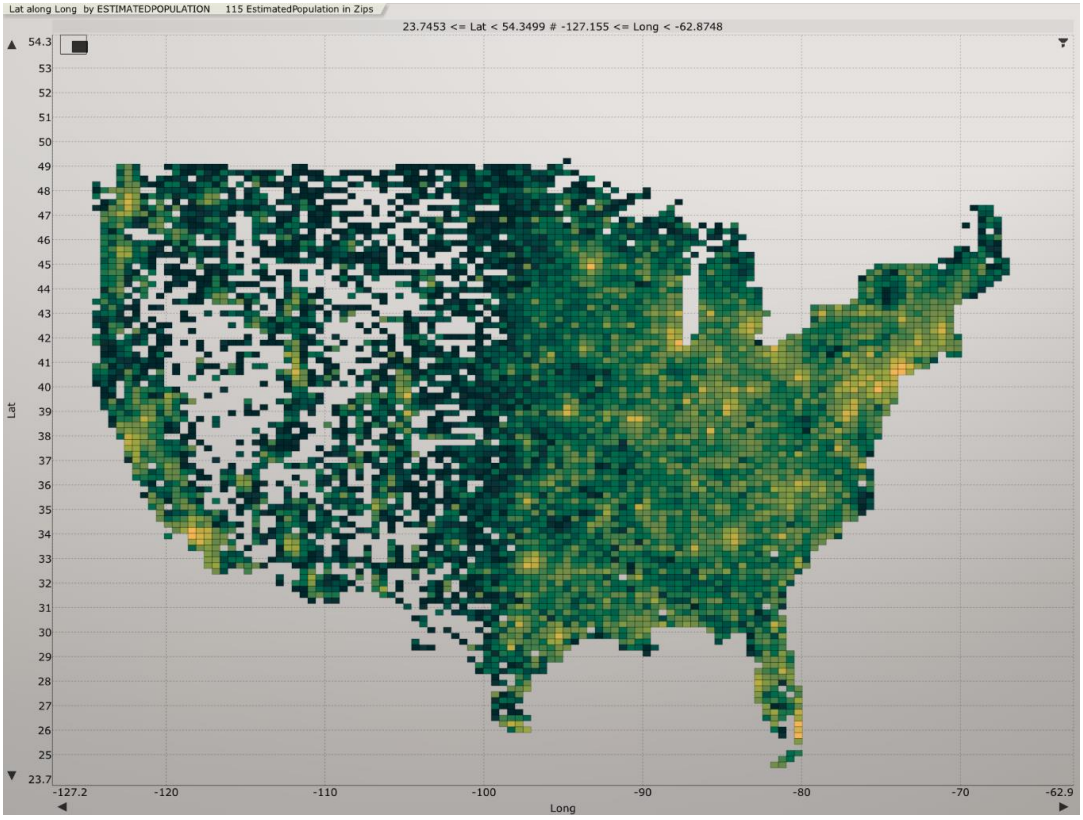


Figure 75 Heatmap between measures (bins)

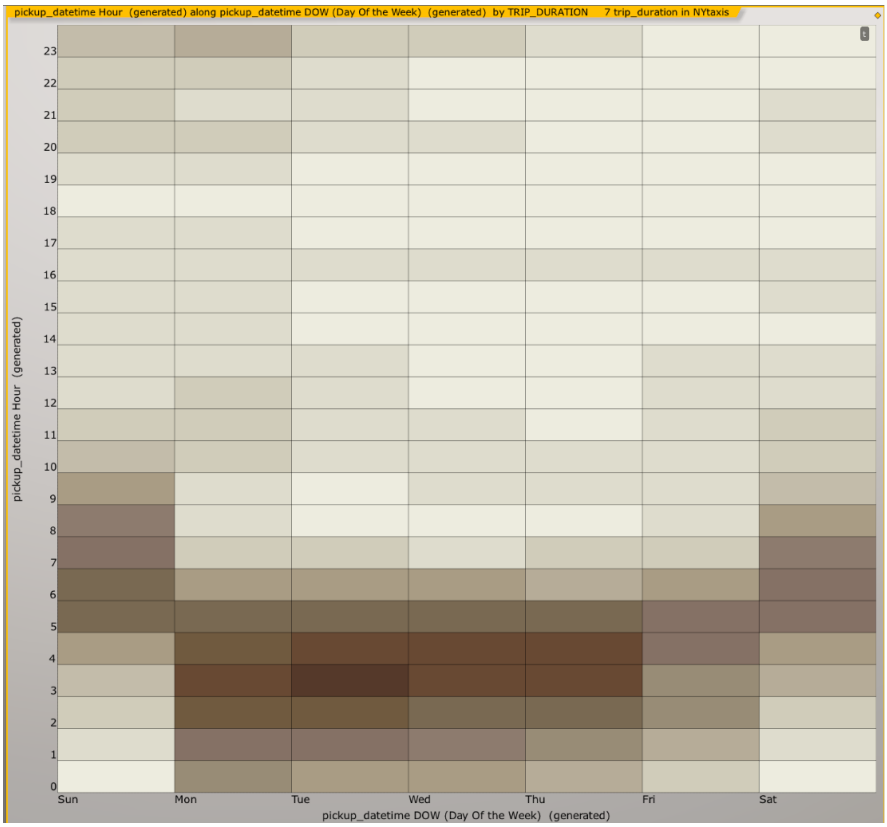


Figure 76 Heatmap between dimensions

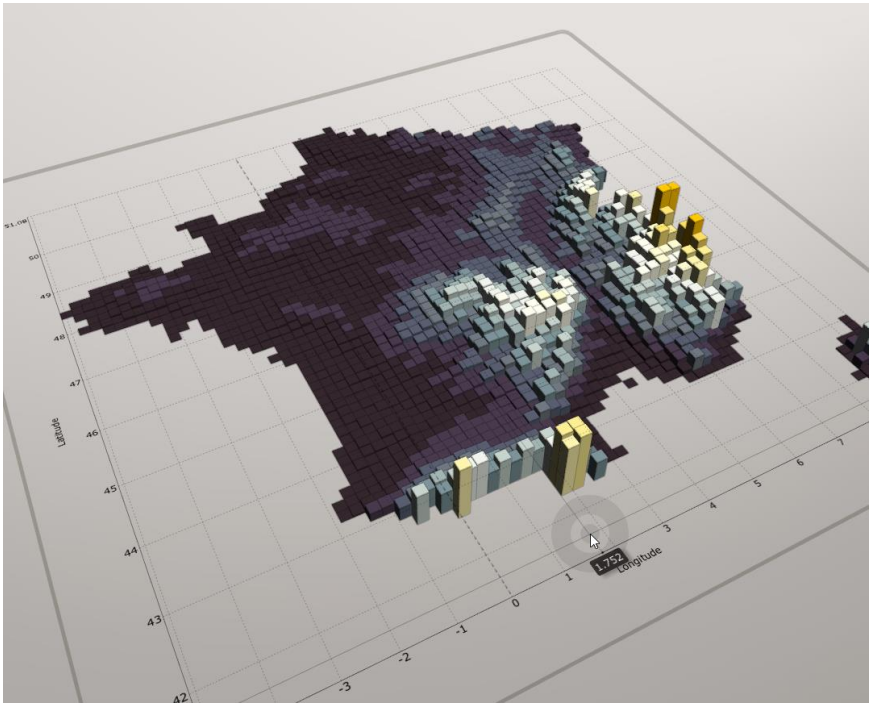


Figure 77 Heatmap measures (3D)

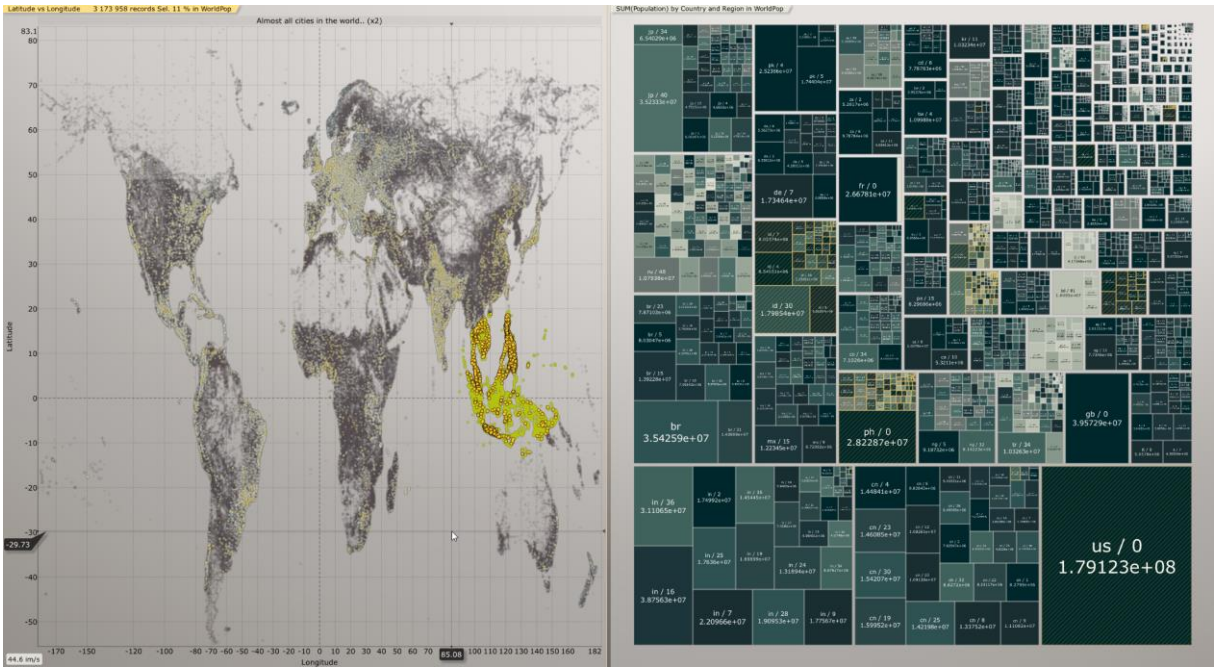


Figure 78 Treemap with 2 dimensions

Figure 79 OHLC bars (candlesticks) and some indicators

10 VIEWS

A view captures all properties needed to display any custom charts configuration in the viewport. Views are useful to save and invoke custom charts configuration. They capture displayed charts with hundreds of properties and restore them in a snap.

A workspace owns multiple views that you can create, modify and remove. A current view is always active. What you see in the viewport is saved in the current view when you save the workspace. The current view during workspace saving will be the view loaded at workspace loading (home view).

Views are not shared between workspaces.

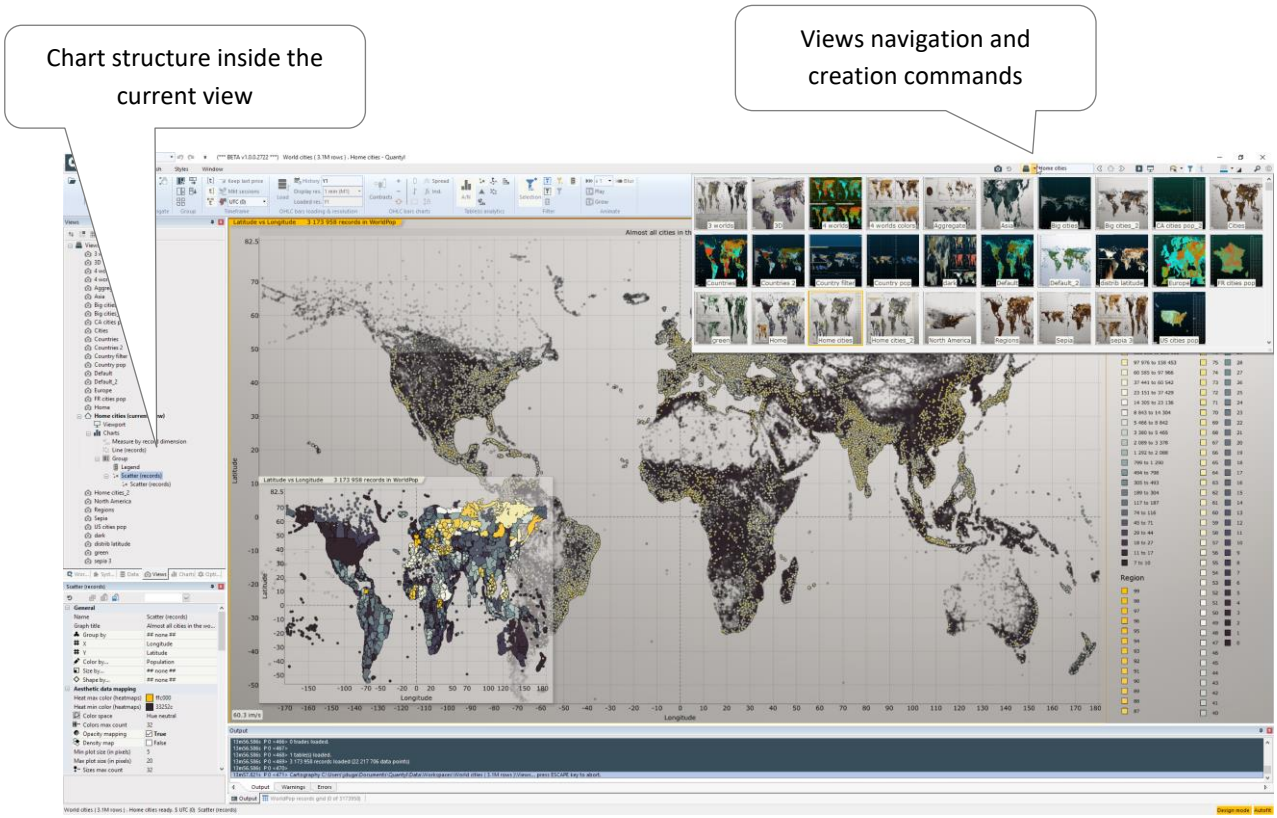
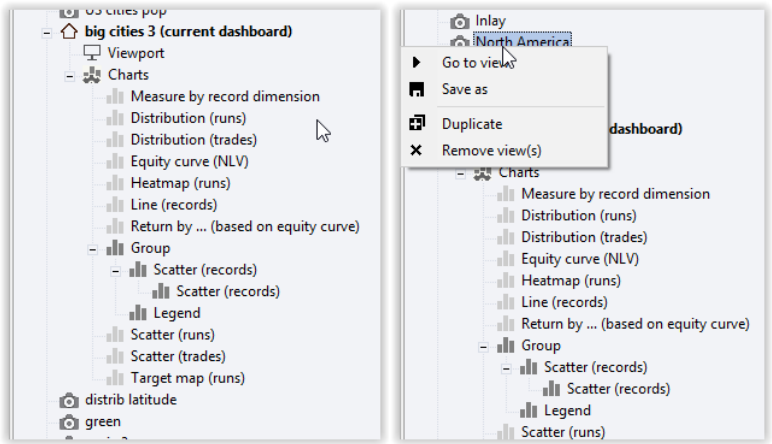


Figure 80 Views



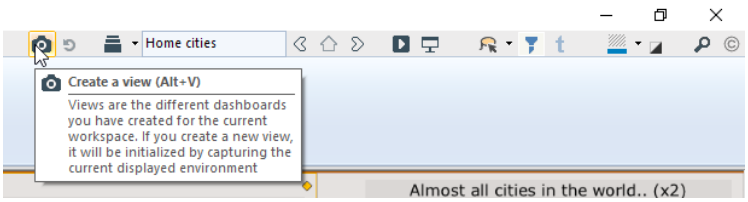
10.1 What is captured in a view?

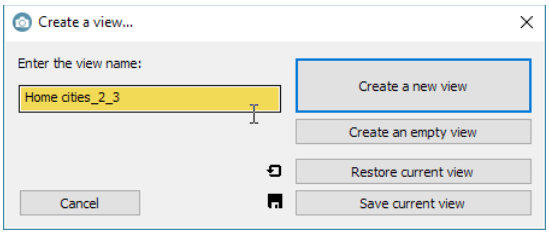
A view captures:

- Visible and non-visible charts **with all their properties** (for each chart: visibility, position, AES mapping settings etc.)
- Viewport properties (background colors etc.)
- Chart manager properties

Views are capturing different states of data visualization; you can see them as “dashboards” if you like.

Note that selection is not captured in the view. When you switch from a view to another, selection is preserved (and so dynamic selection filtering).





10.2 Navigation between the views

You can easily invoke a view through the gallery, or simply navigate by using previous/next commands (page down/page up keys) or drag them from the tree view into the viewport. Those commands take care of the previous view that you opened, like an undo stack. And you can also navigate beyond views already visited. The views are loaded very quickly, and it is very easy to go back and forth in the view stack.

You can also play them automatically, in a story telling fashion (play button in the view commands).

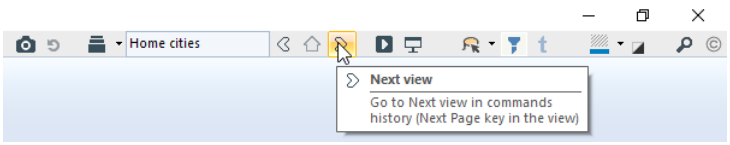


Figure 81 Views navigation

You can define the order of the views in the view tree, simply drag & drop views in the tree to organize them as you wish.

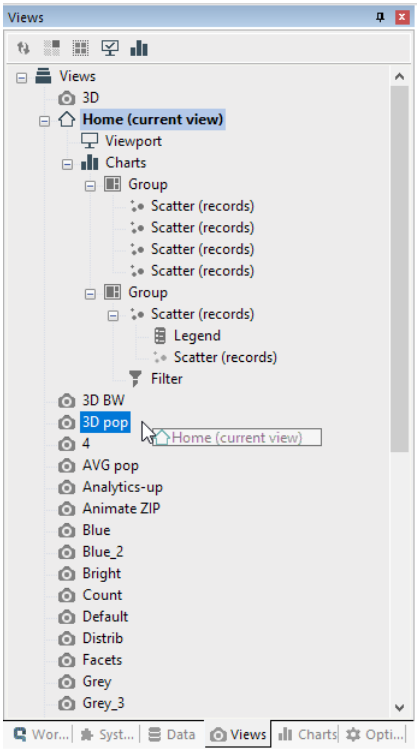


Figure 82 Drag the views in the tree to set the desired order

10.3 STYLES

You can tweak each chart color property if you like, they are independent. You can modify background color, color mapping, viewport color of each chart as desired. You can copy and paste aesthetical properties from chart to chart if needed.

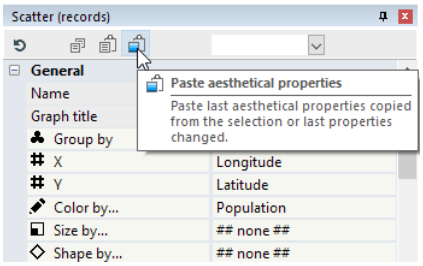


Figure 83 Paste aesthetical properties (property page)

Do you need some quick and simple good-looking pre-defined styles? you can use tools and commands of the **style ribbon**. You also have commands that allows to modify all colors properties as a whole in the current view. You may decide to apply style to default values, new charts created will adopt the style.

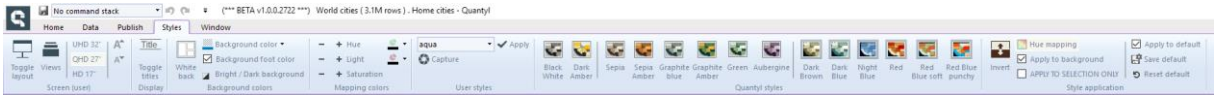


Figure 84 Styles ribbon category

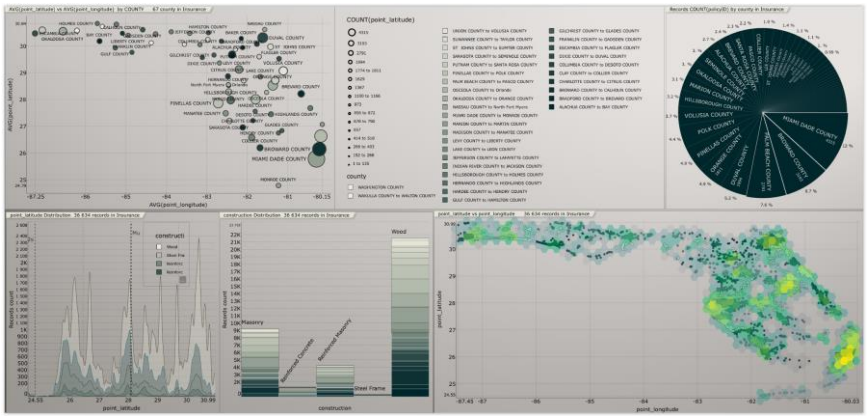


Figure 85 Blue graphite style

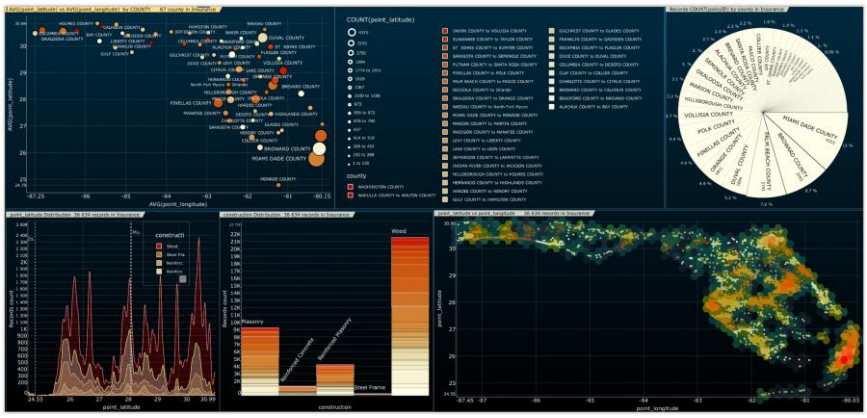


Figure 86 Dark brown style

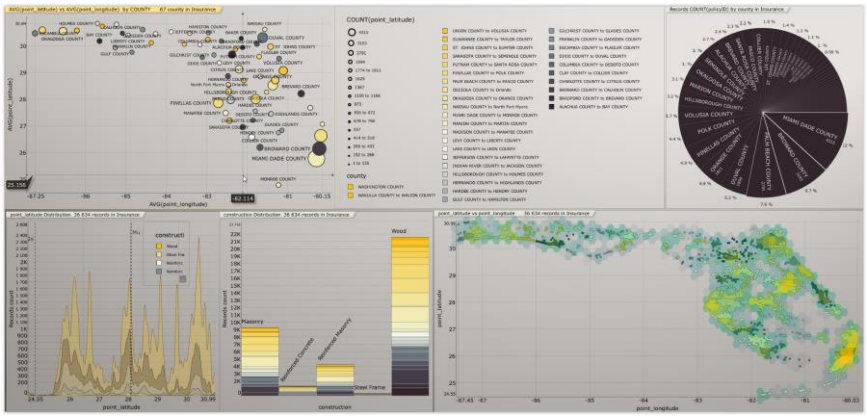


Figure 87 Dark amber style

You can also create your own custom style by simple capture of the charts. Modify charts color as you wish, then capture and rename it as a custom style (10 available slots).

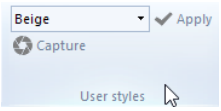


Figure 88 Custom styles creation

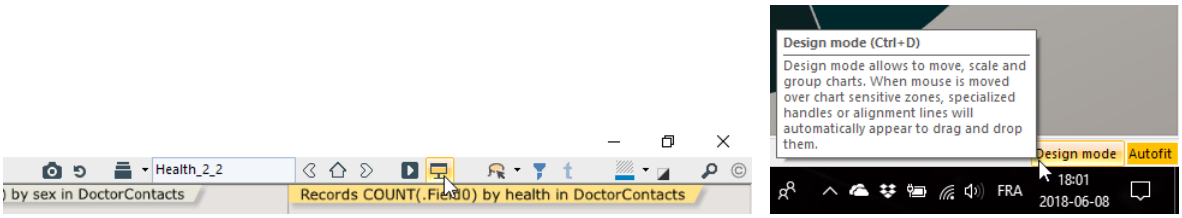
10.4 Persistency

Only current view is resident in memory. When you save a view, it is really “hard” saved. Use and abuse of views, they are cheap.

10.5 Presentation mode and story telling

“Presentation mode” is the combination of 2 features:

1. Switch to full-screen mode
2. Disable design mode



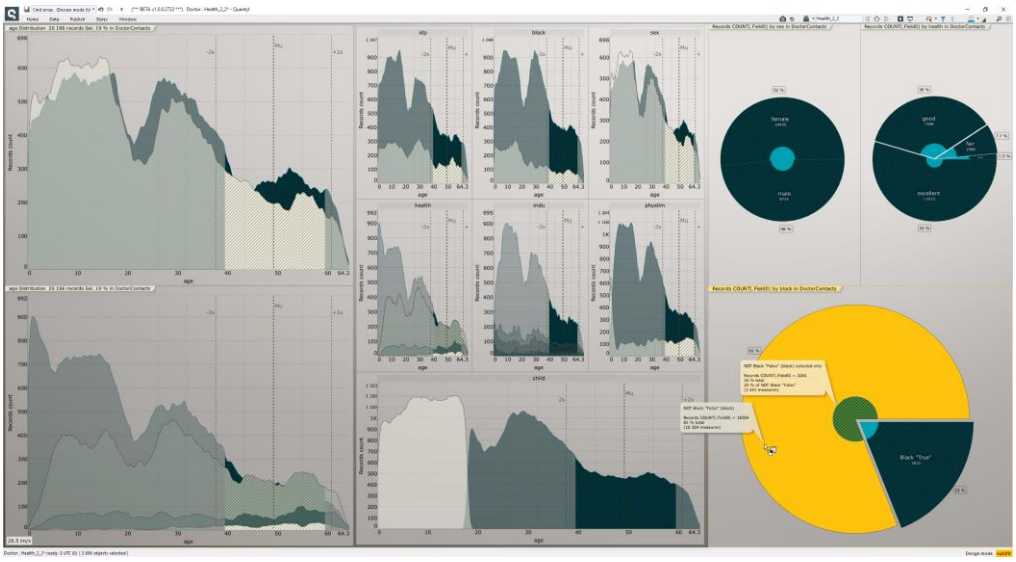


Figure 89 Presentation mode

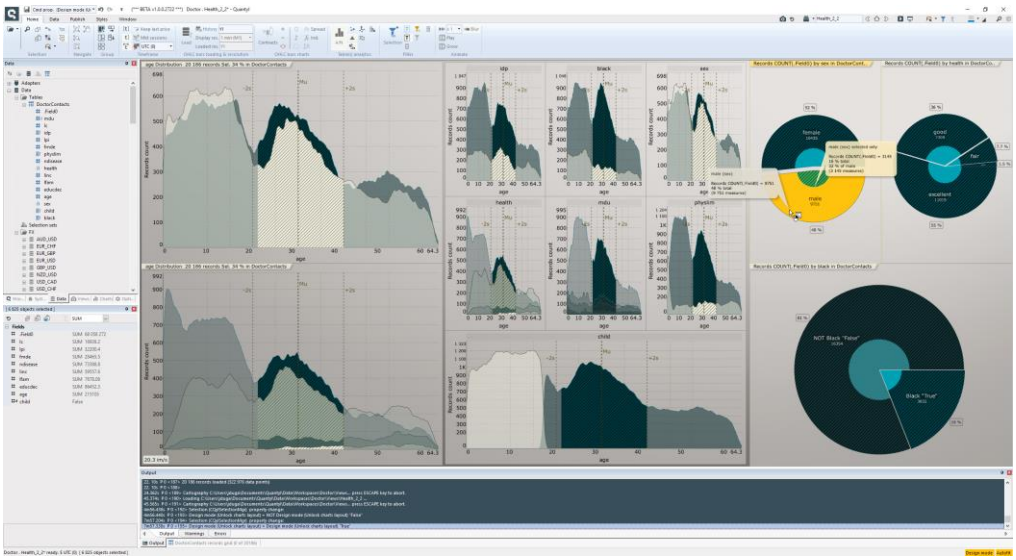
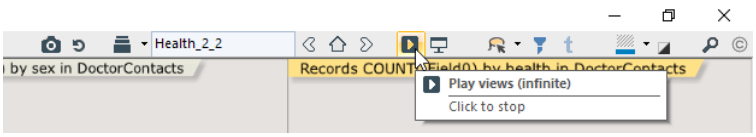


Figure 90 Standard mode (windows and ribbon are not hidden)

In presentation mode, it is often interesting to navigate between the views or play them automatically. See the view section know how to organize them.



11 SEARCH AND FILTER

Search is a very powerful tool that allows you to select and create custom filters.

You have many operators available to combine the search results to the current selection. With successive search you may combine different results. If cross filtering is enabled, the selected chart is used as the master.

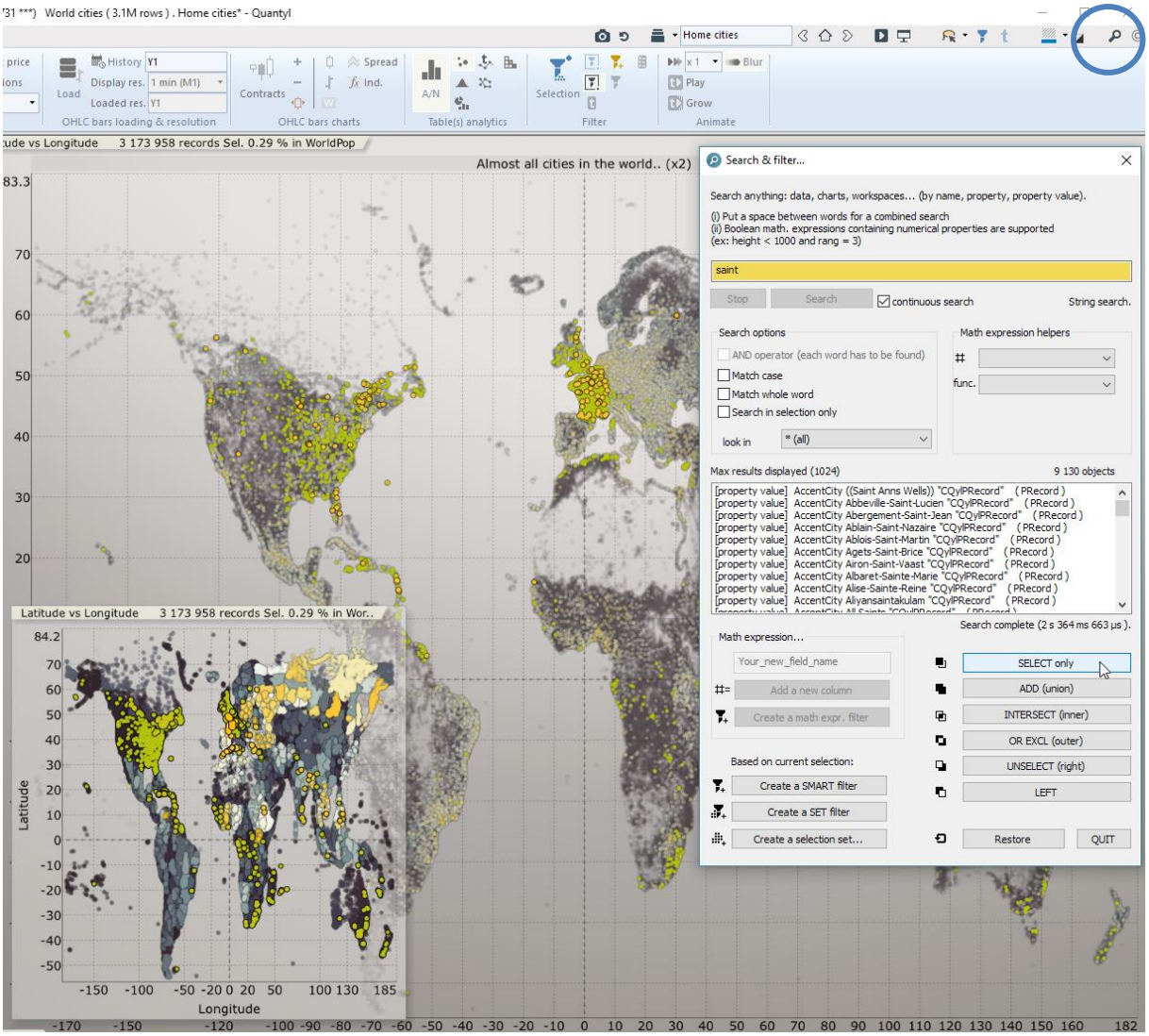


Figure 91 Search & Filter dialog box (CTRL + F)

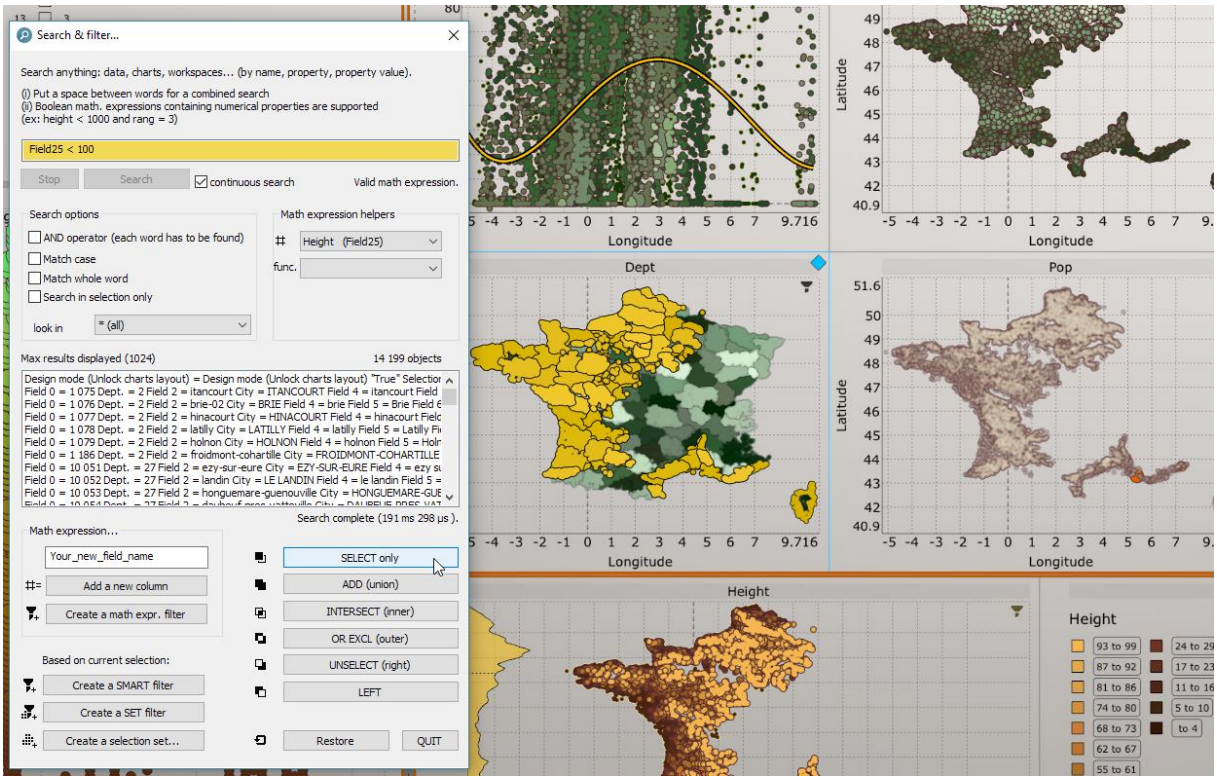


Figure 92 Search with mathematical expression based on dimension or measure

12 FILTER

Filtering is a core capability in data discovery. Quantyl offers multiple ways to filter your data.

12.1 STATIC FILTERS

You have to create filtering chart, aka “filter”. Those filters are applied to all charts in the filtering scope. Filter applied to its sibling charts in the view hierarchy. With this simple mechanism, you can apply filter to a single chart, a group of charts, or to all chart by simple drag & drop of the filter on the desired target. You can detach filters from a chart and apply it to another.

The filter is applied only when the filter chart is visible. Filtering charts inherit of chart’s common behavior: you can move, duplicate, remove them as you wish.

You can create filter from different area of a chart through right mouse menu:

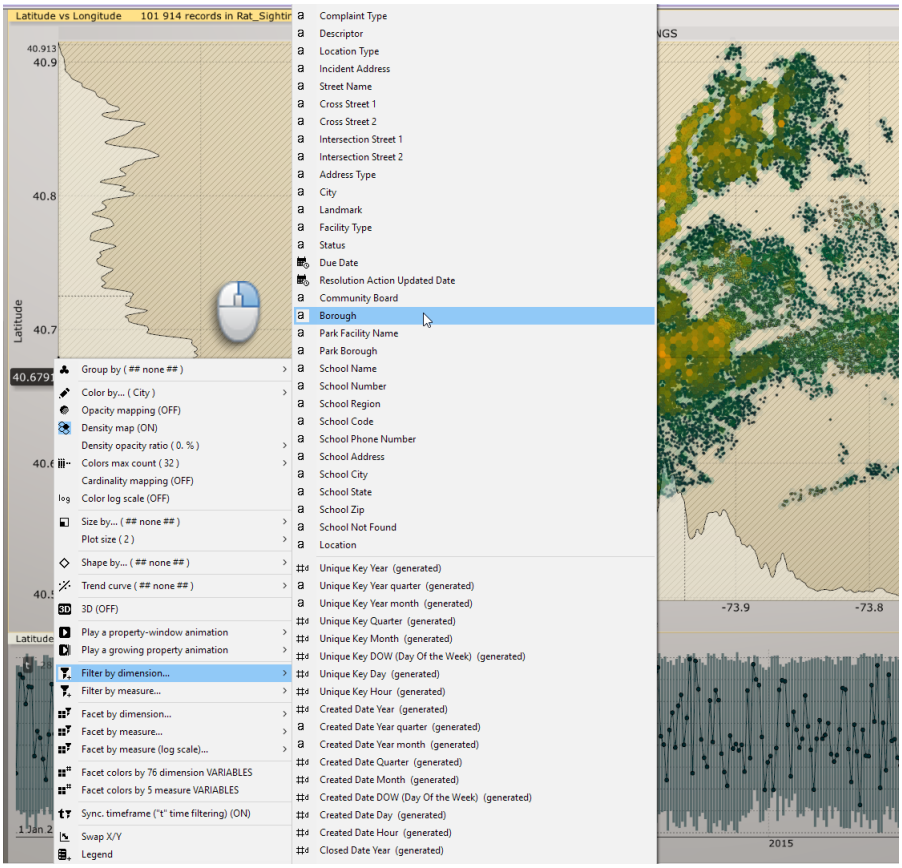


Figure 93 Filter by dimension from graph area menu

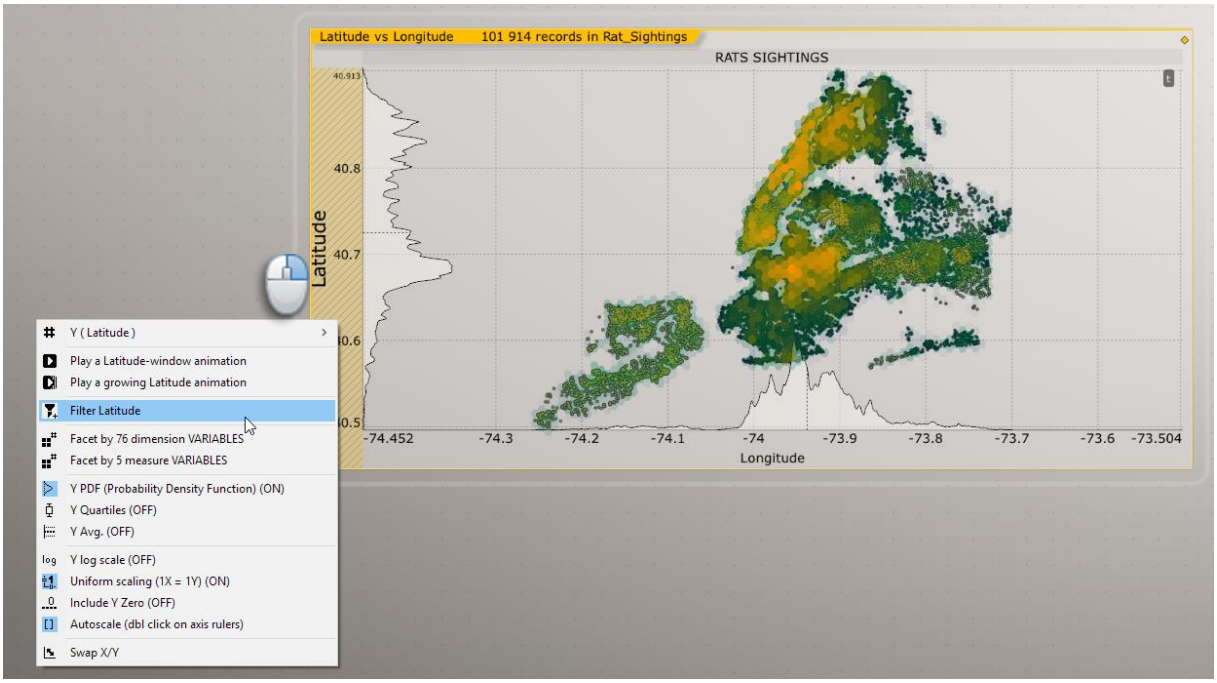


Figure 94 Create a filter from an axis menu

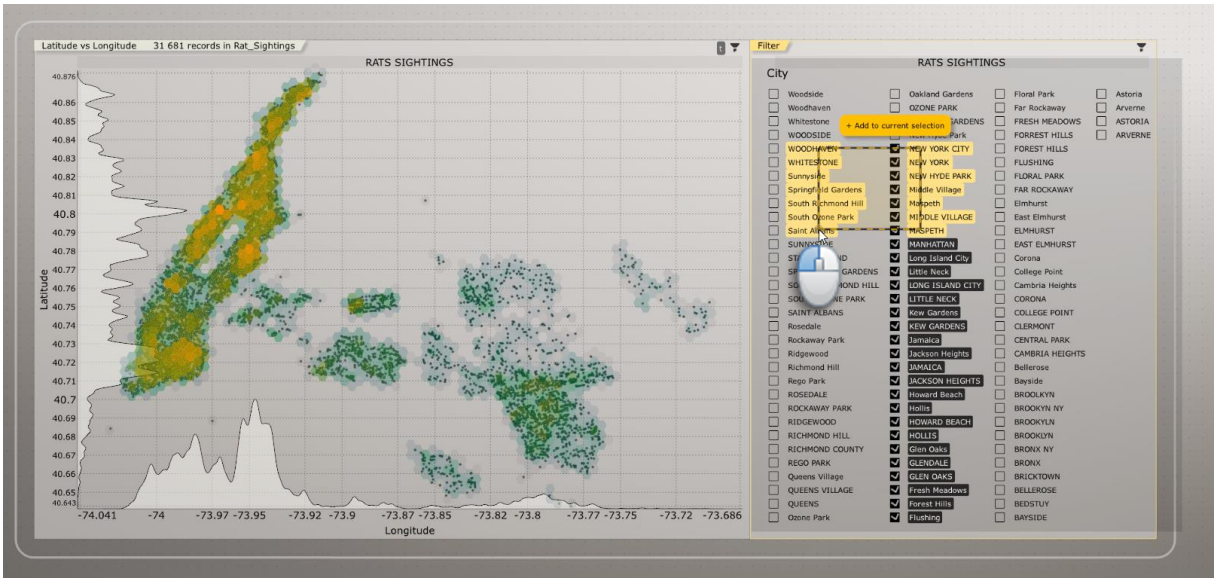


Figure 95 Dragging a rectangle to select or deselect multiple values

Important

As for standard selection, rectangle selection from right to left is additive when applied on check boxes.

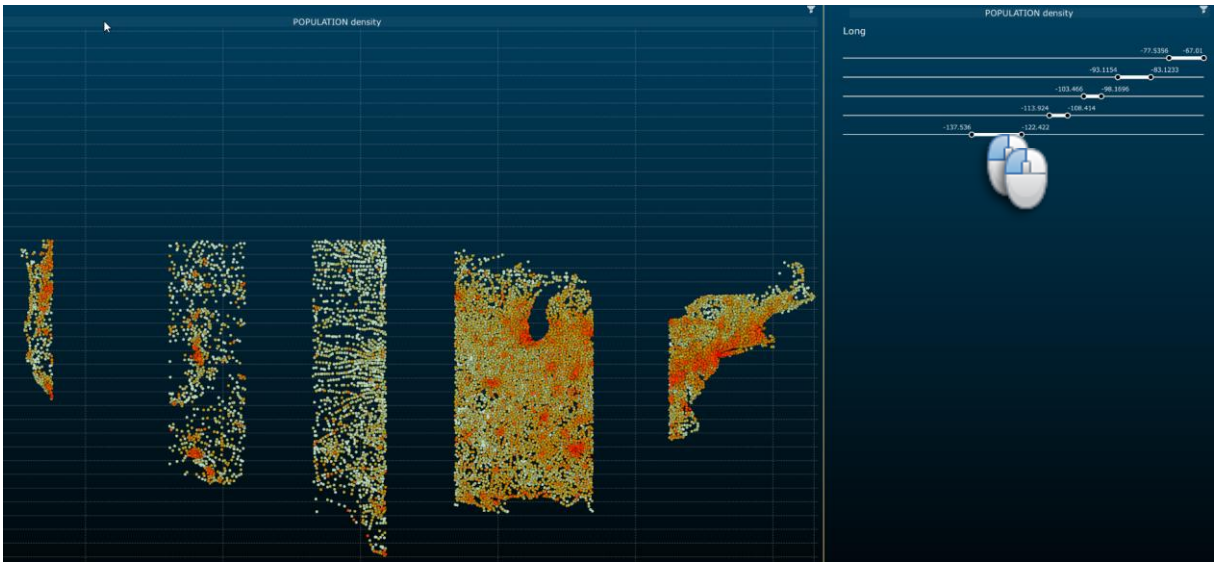
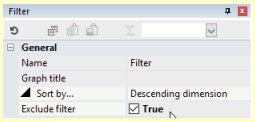


Figure 96 Double click on measure range filter to create a new range

Important

To switch to an **exclude** filter, you have just to check the exclude filter property of the filter chart:



12.2 Dynamic selection filtering (cross filtering / drilldown)

Dynamic selection filtering (also known as **cross filtering**) is a way to filter your data **"on the fly"** when you select records in your charts. As selection is volatile, the created filter is in a "temporary" state. The power of the tool is that you can drill down in your data by performing successive selections in different charts to affine your target (**drilldown**). As long as the selection is maintained, dynamic filtering is applied.

Cross filtering is a mode that is saved with the view.

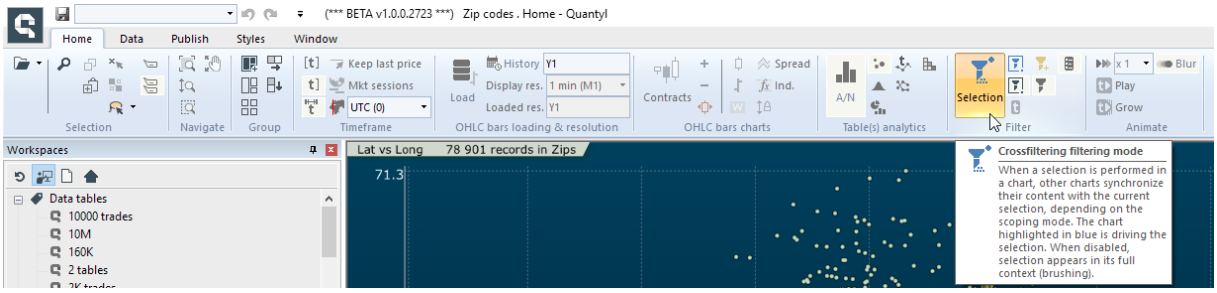


Figure 97 Cross filtering mode

If you want to capture the current filter as a permanent filter, you may create a **smart filter** based on this selection.

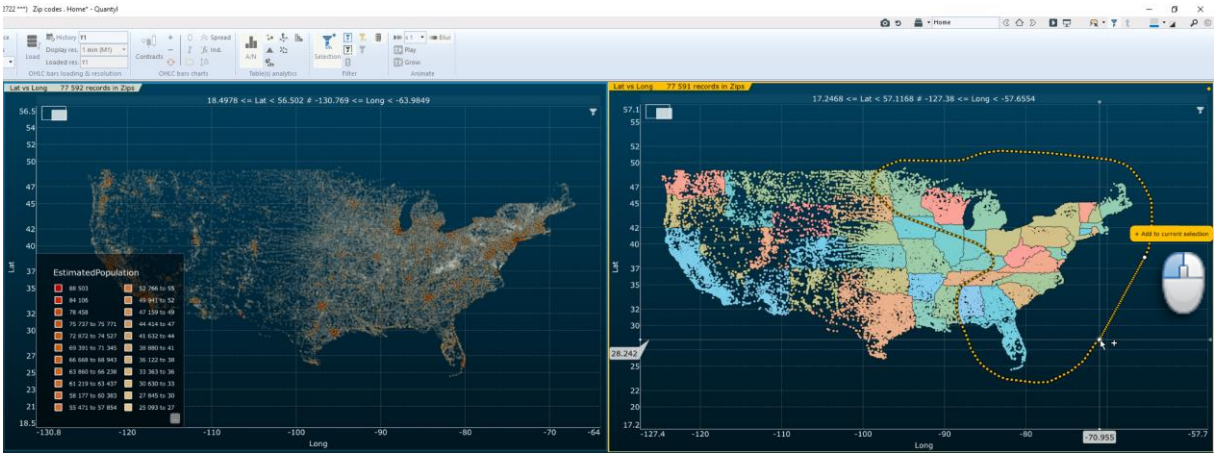


Figure 98 Lasso selection

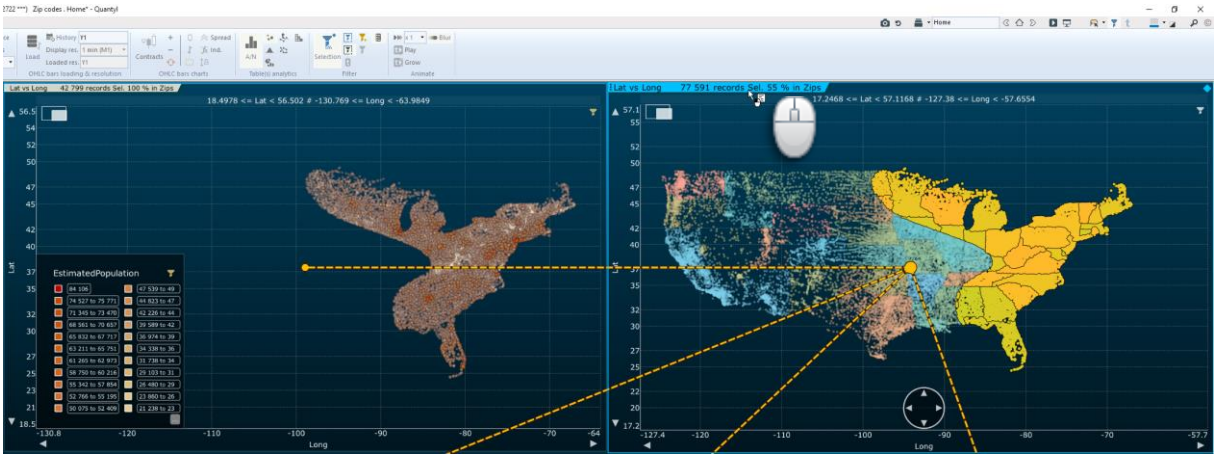
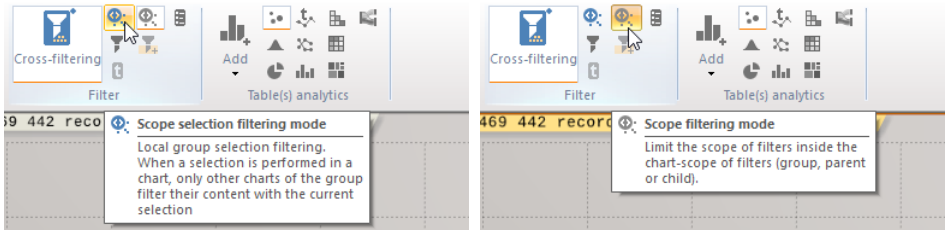


Figure 99 Lass selection result with selection filtering

12.3 SCOPING MODES

There two scoping modes:

1. one for static filters context
2. and one for dynamic section filtering



Scoping mode	Filtering
enabled	Scoping is defined by the charts grouping structure
disabled	There is no scoping, the complete view is filtered

Important

Put the mouse over the chart title to display scope context.

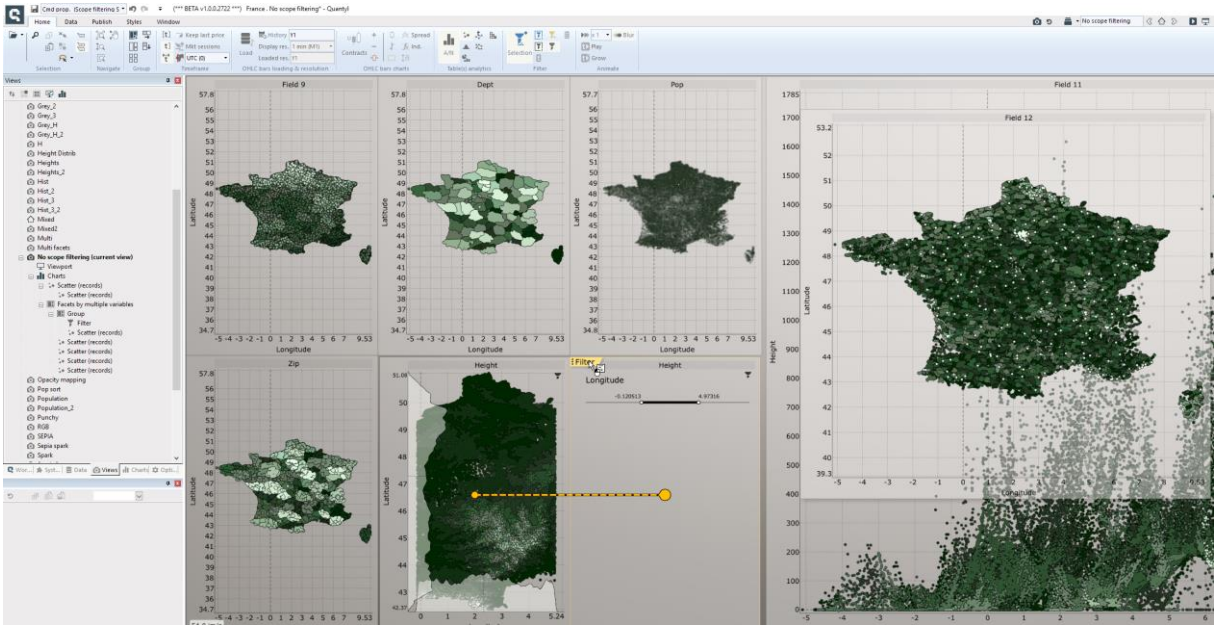


Figure 100 Scoping enabled

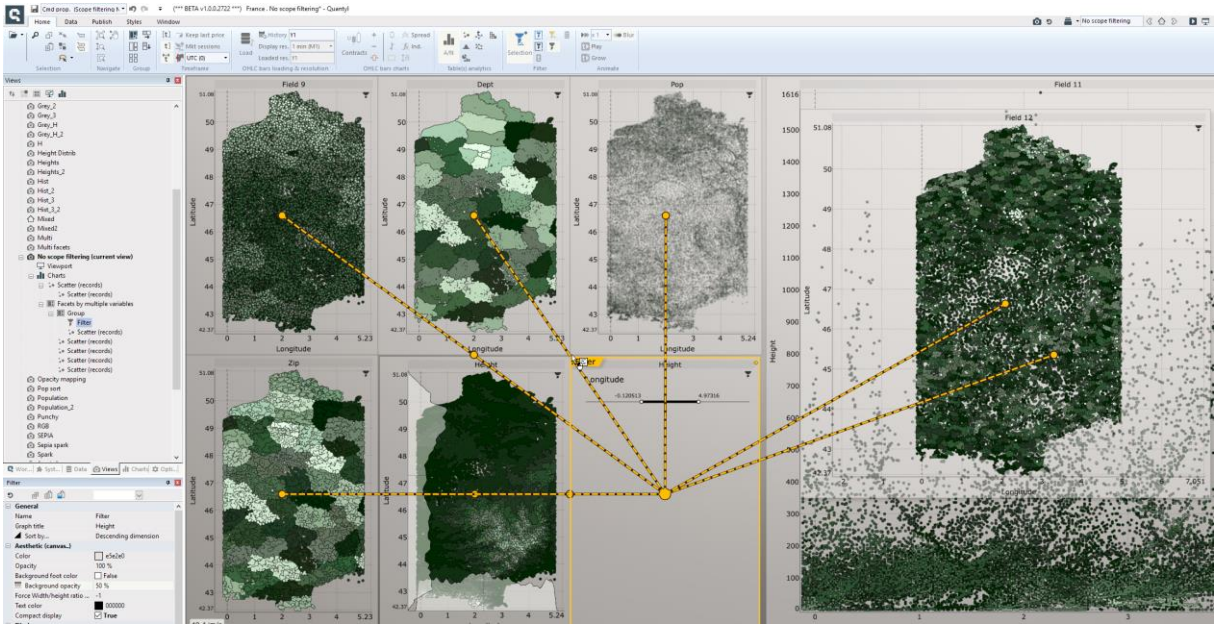


Figure 101 Scoping mode disabled

12.4 SCOPING DISPLAY: WHICH IS FILTERED BY WHICH?

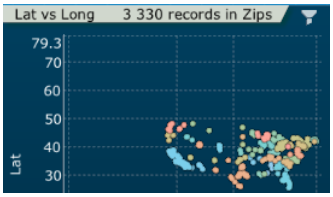


Figure 102 Filtered graph have a filter glyph in top right corner

As it will follow, there are different type of filters, with different scope. Some indicators help you to understand relationship between filters and filtered objects.

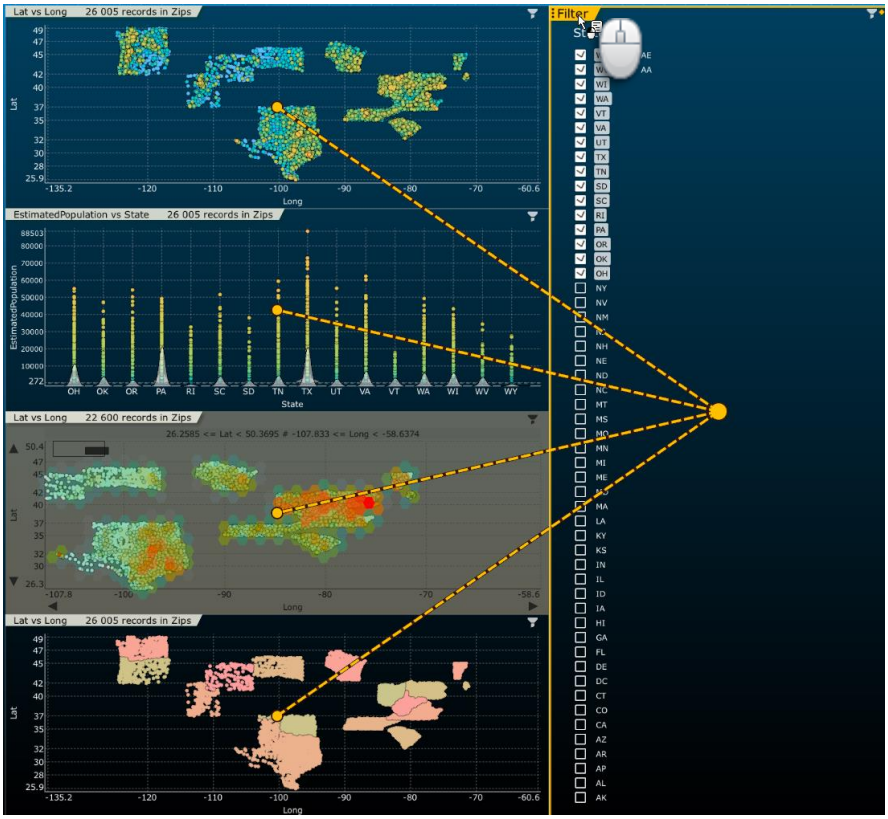


Figure 103 Mouse over a filter title to see scoped chart(s)

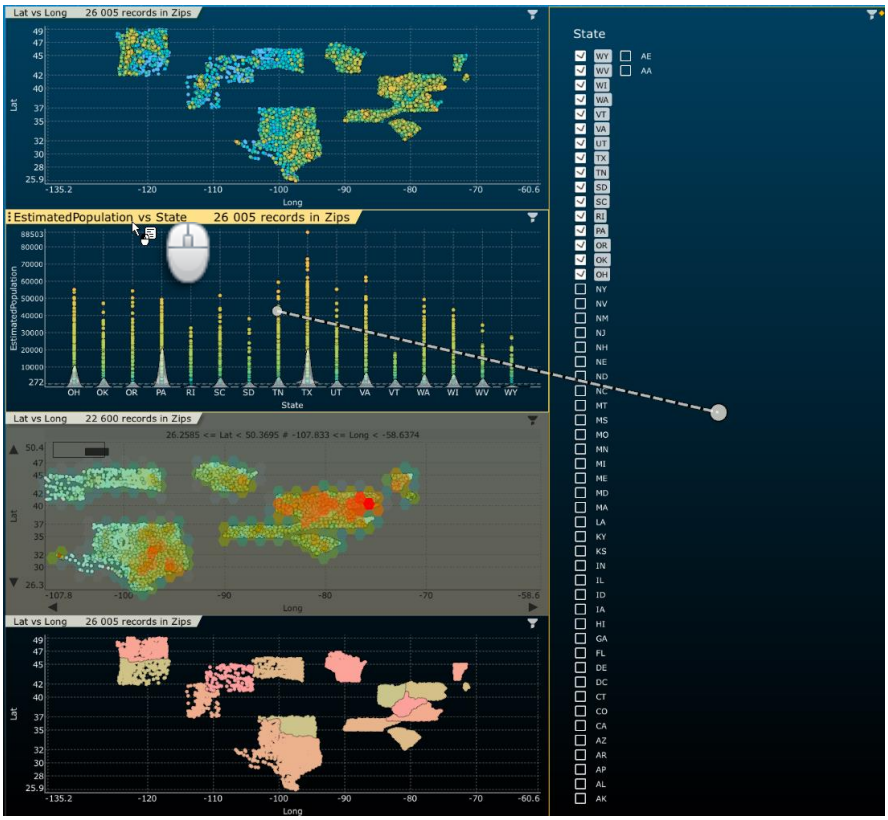
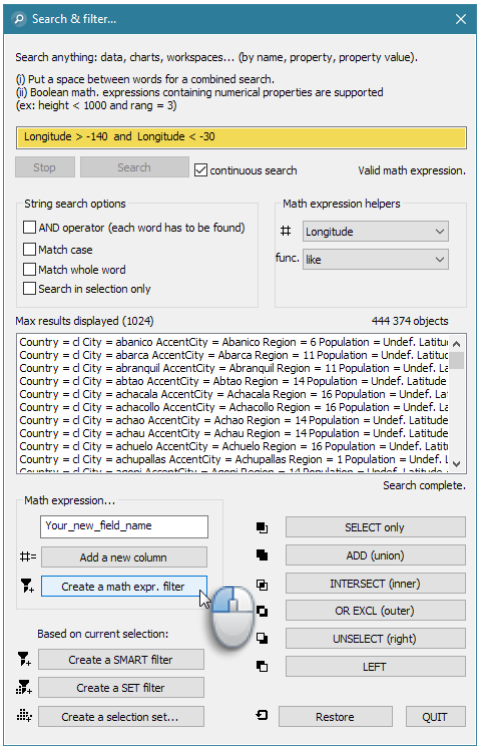


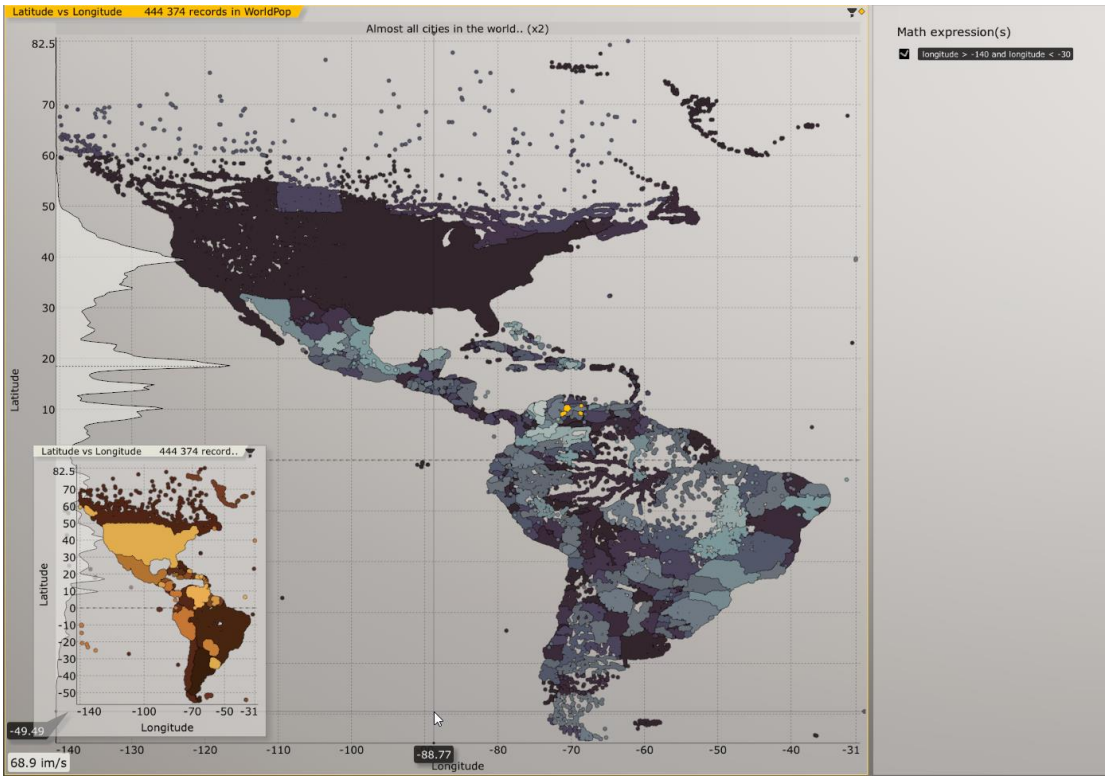
Figure 104 Mouse over a chart title to see its filter(s)

12.5 MATH EXPRESSION FILTERS

You can create filters based on mathematical formulas containing numerical fields.

You can add a static filter based on columns that you have added to the table (with math expression) or you can directly add a filter based on a math expression through the Search & filter window.





12.6 SMART FILTERS

Smart filter is more a command than a filter. The purpose of the smart filter creation is to analyze current selection to create a filter based on a minimal set of dimensions that contains the current selection. The result of a smart filter is always larger or equal to the current selection it is built upon.

After the creation of the filter, you can still affine your filter by removing dimensions values that you want to exclude.

The huge benefit of smart filter is to simplify the process of filter creation. It allows to capture dynamic selection filtering in a snap.

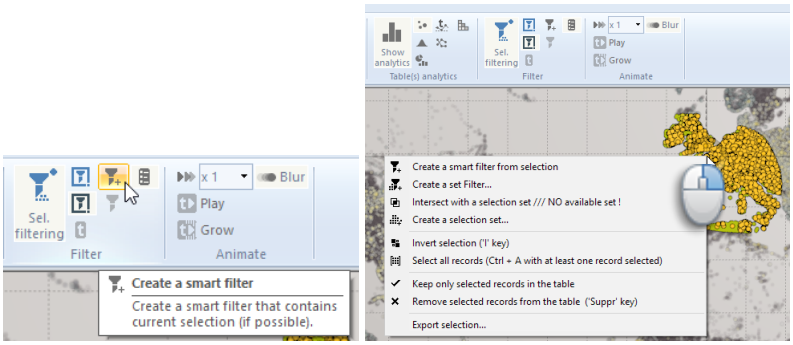


Figure 105 Create a smart filter from selection menu or from the command in the ribbon

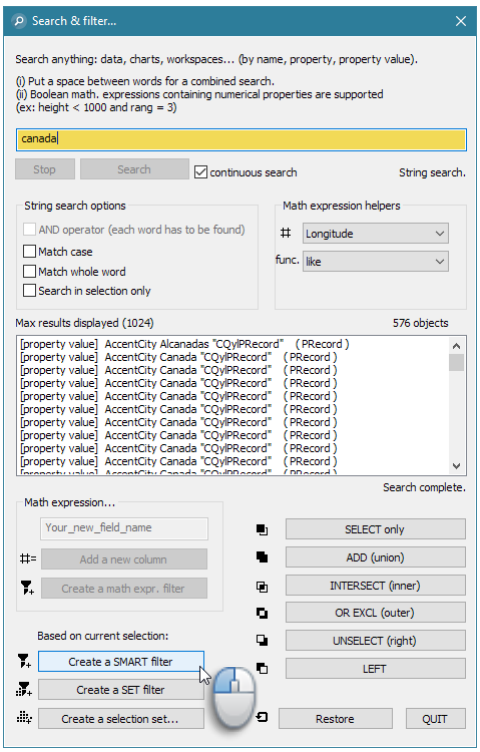


Figure 106 Create a smart filter from Search and filter window

12.7 SELECTION SET FILTERS

Those filters are based on selection set. This is the finest grain of filter you can apply because they are referencing row numbers in tables. This is a double-edged sword; those filters are not robust to table rows change.

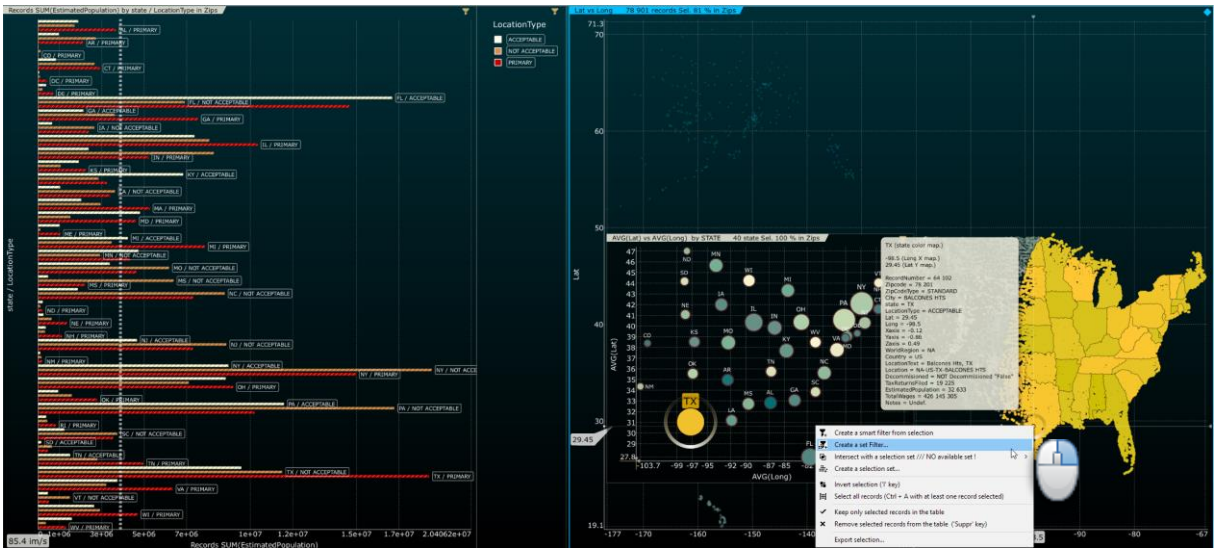
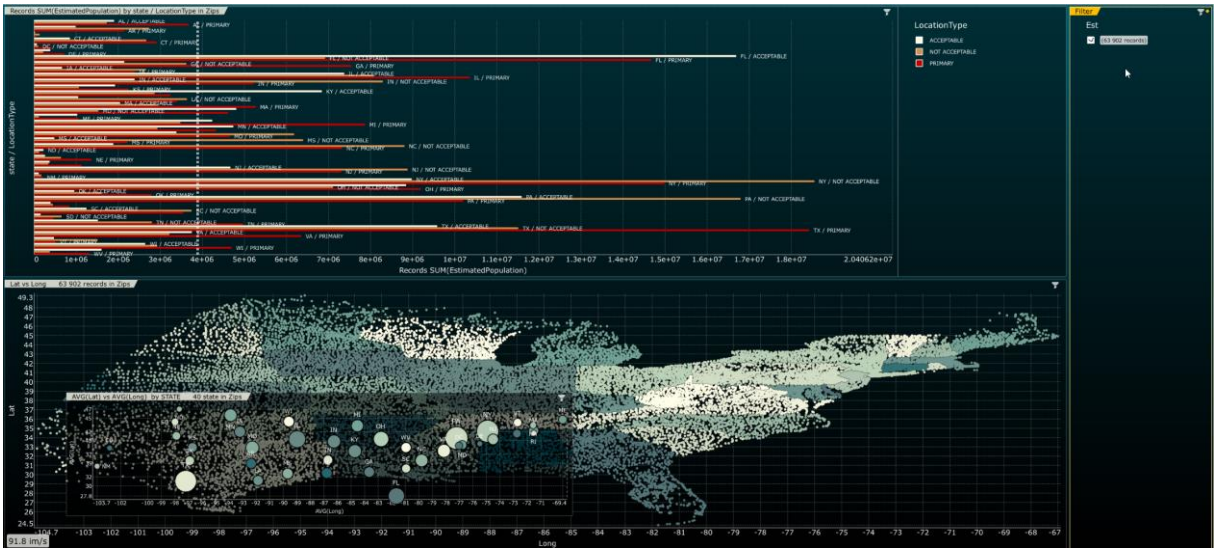


Figure 107 Creating a selection set filter (right click menu on selection)



12.8 TIME

Time dimension benefits of specific features. Time dimension is automatically decomposed in sub-dimensions to facilitate its use in charts and filters:

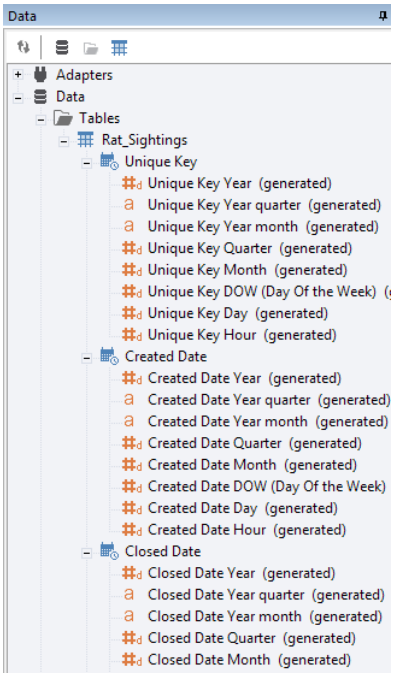


Figure 108 Time decomposition (orange fields are automatically generated)

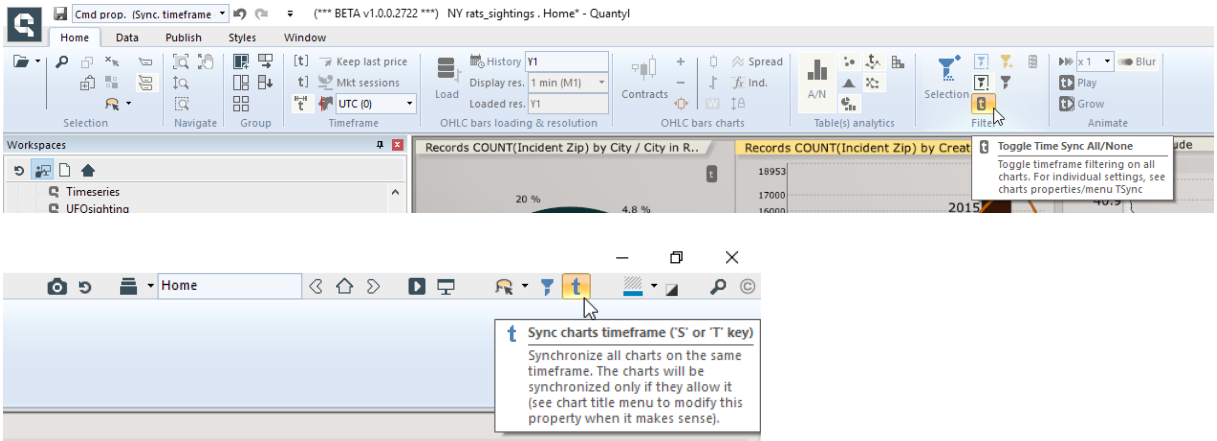
You can **filter** by time as any measure or dimension, but you can also decide to synchronize charts to the same global time range in a view.

Some charts like contracts, are timeline based, they need a time dimension to display something. When you navigate along time in those charts, you may desire to **synchronize** all or some of the other charts to the time range you modify. **It is not exactly a "time filter"** because it may scale other time windows instead of simply filter the display and manipulate all time dimensions as a single time reference.

Time synchronization is both a global and a local property.

When time is not synchronized, you can see time range prints in different charts and modify them.

When a time property is available, it unlocks chart time filtering and time synchronization capabilities.



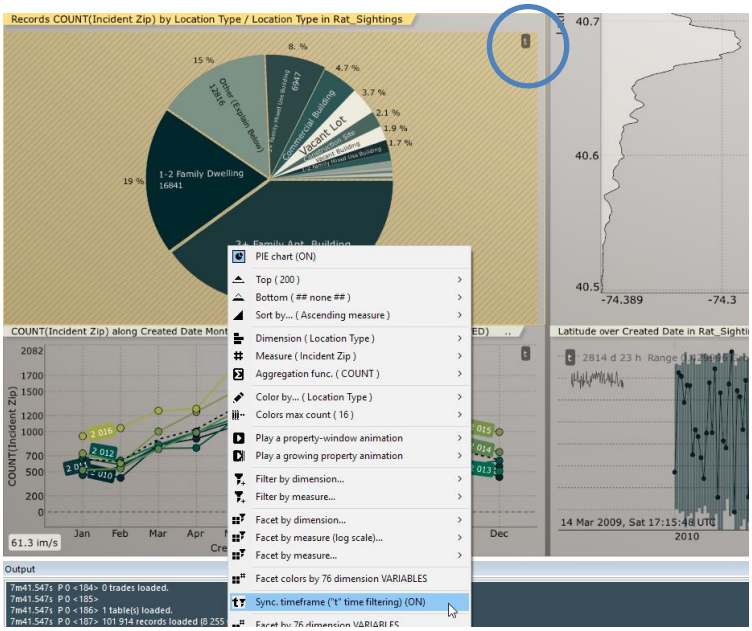


Figure 109 Time synchronization

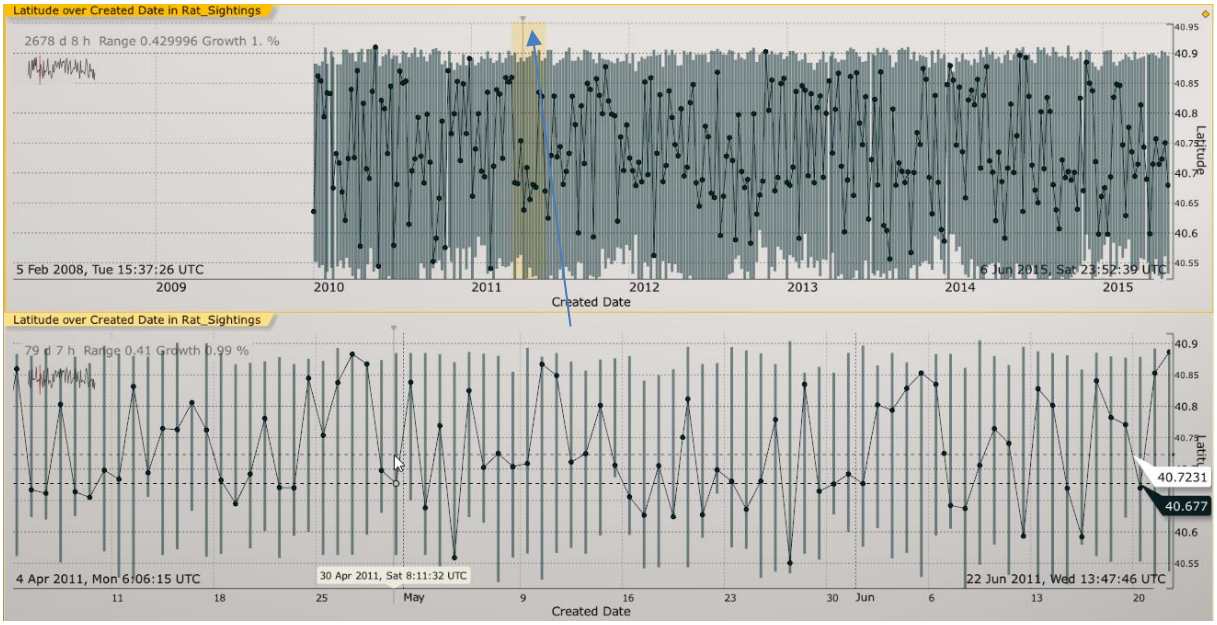


Figure 110 Interactive shadow timeframe in case of multiple time charts

12.9 FACETING

Faceting commands allow to create multiple charts to compare dimensions or measures values. The faceting process may automatically compute a different filter per chart, or map color to a different property.

For example, it may be interesting to create the same chart for each dimension value (the year, the month etc.). You may achieve successive faceting command to combine filters: facet by year, then facet the charts created by month.

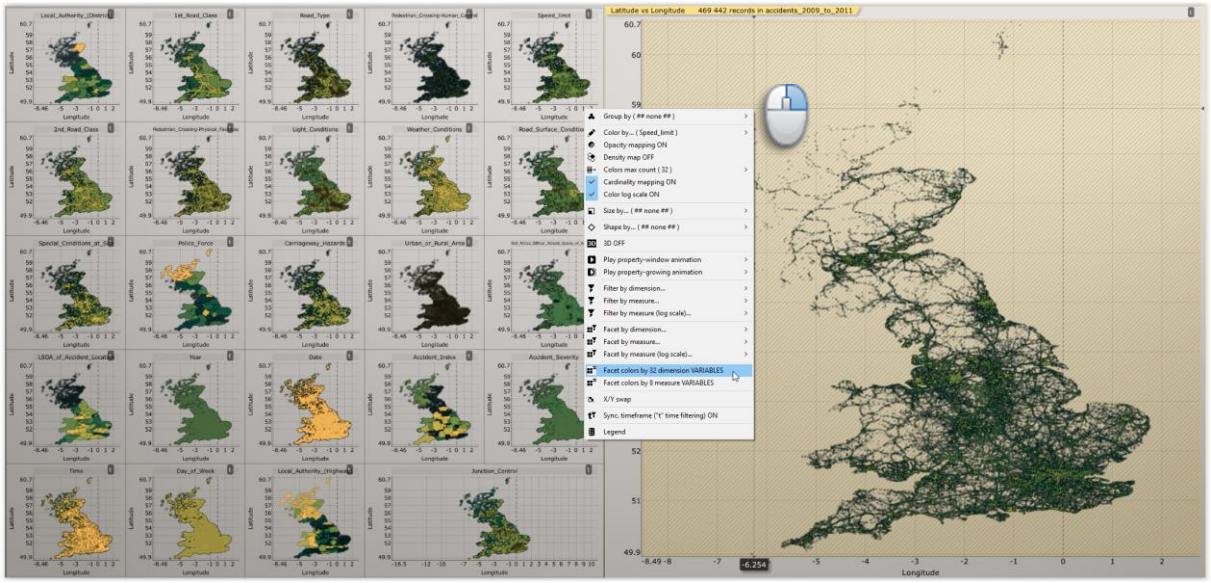


Figure 111 Faceting by colors

Important

Some faceting commands may create filter(s) that are encapsulated in the chart. A text describing the filter is automatically displayed in the graph area. It happens when you facet a single variable to display all its different categories.

12.10 ANIMATING

You can animate any dimension or measure. It is very interesting with datetime. The animation process creates temporary filters for each visible chart, those filters are update at each animation frame.

This filter may be a window or a growing range of values.

- Window play translate both limit of the range, the range is still the same
- Growing play translates only superior value of the range, range is growing

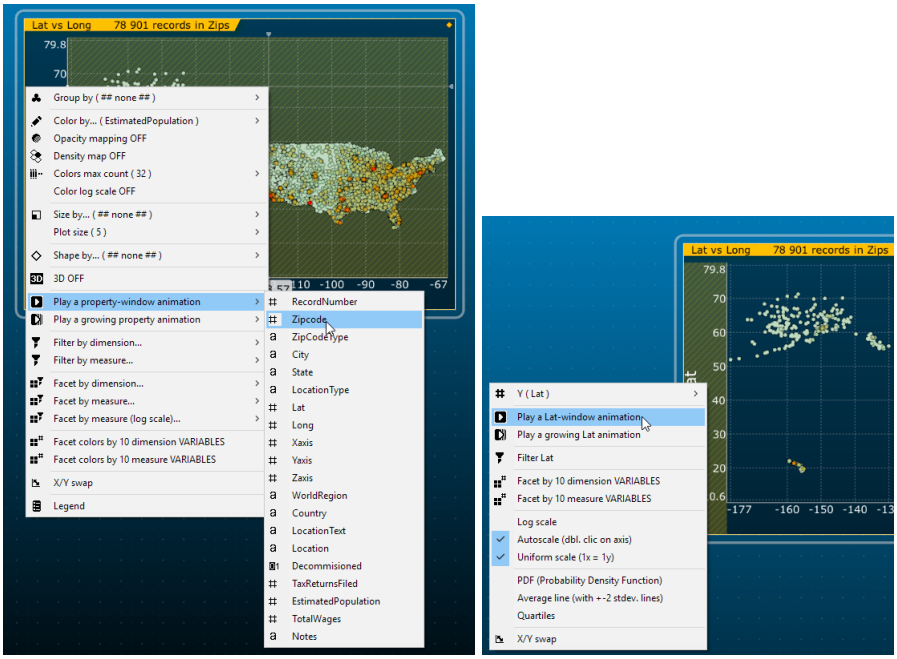


Figure 112 Animate menu from the graph area or an axis

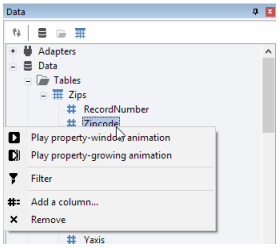


Figure 113 Animate from table fields

Important

Animation filter overrides existing filters that target the same property during the animation.

13 AESTHETICAL AND AXIS FACETING

It is possible to automatically create a collection of charts to explore multiple dimensions or measures. This faceting does not create filters but declines the same chart with the different available properties.

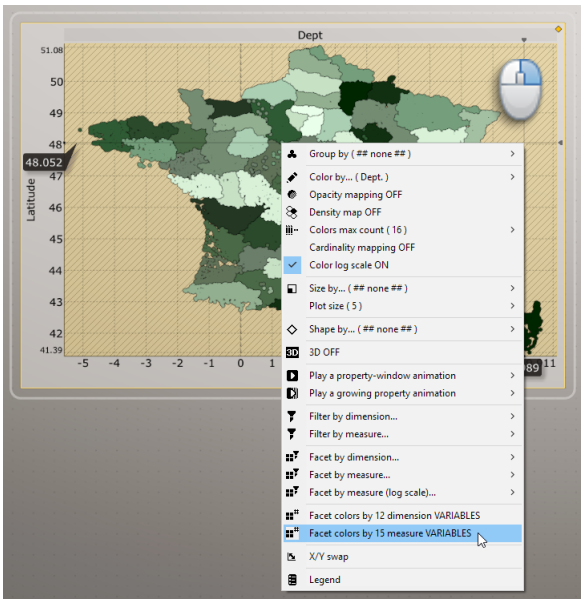


Figure 114 Color faceting command

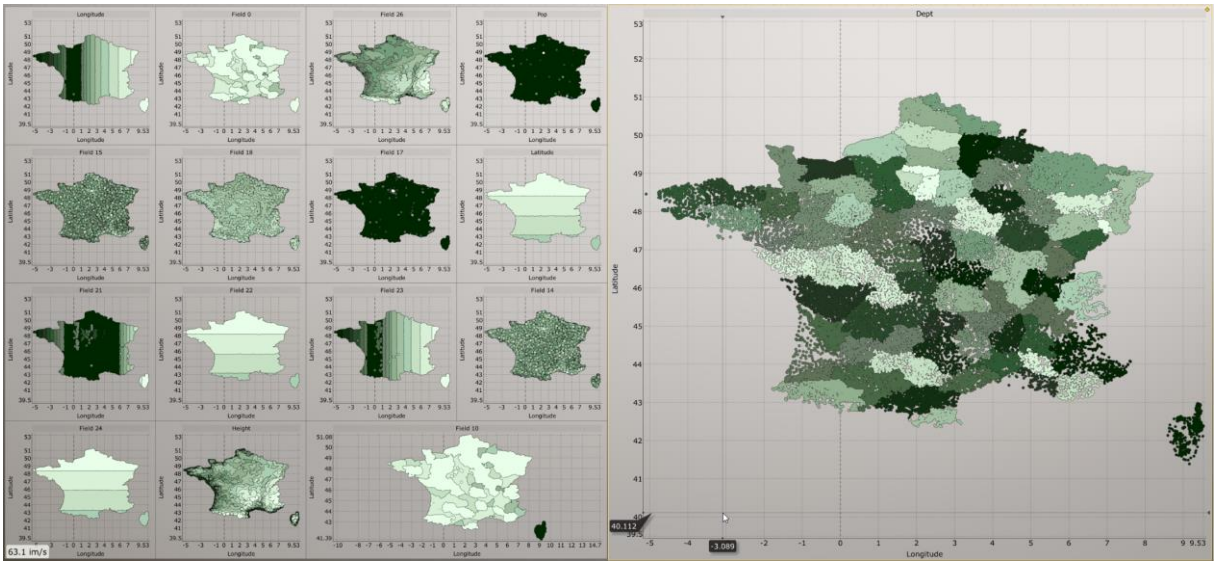


Figure 115 Color faceting applied on available measures

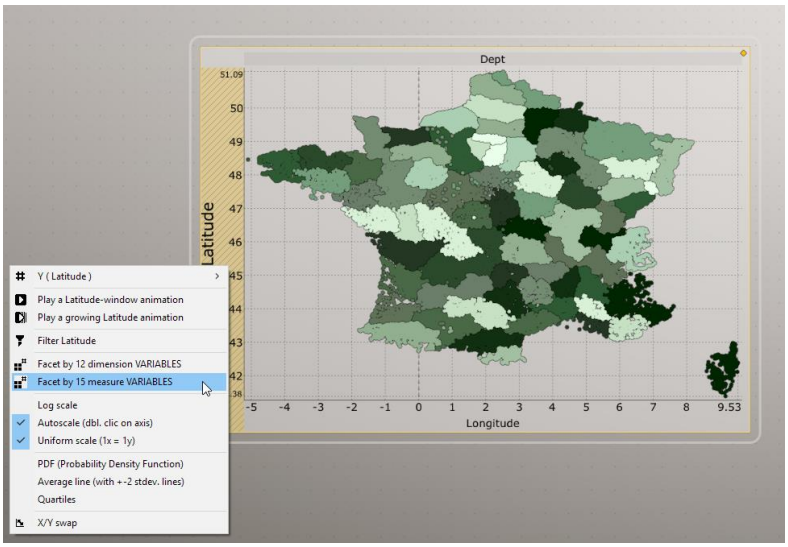


Figure 116 Axis faceting

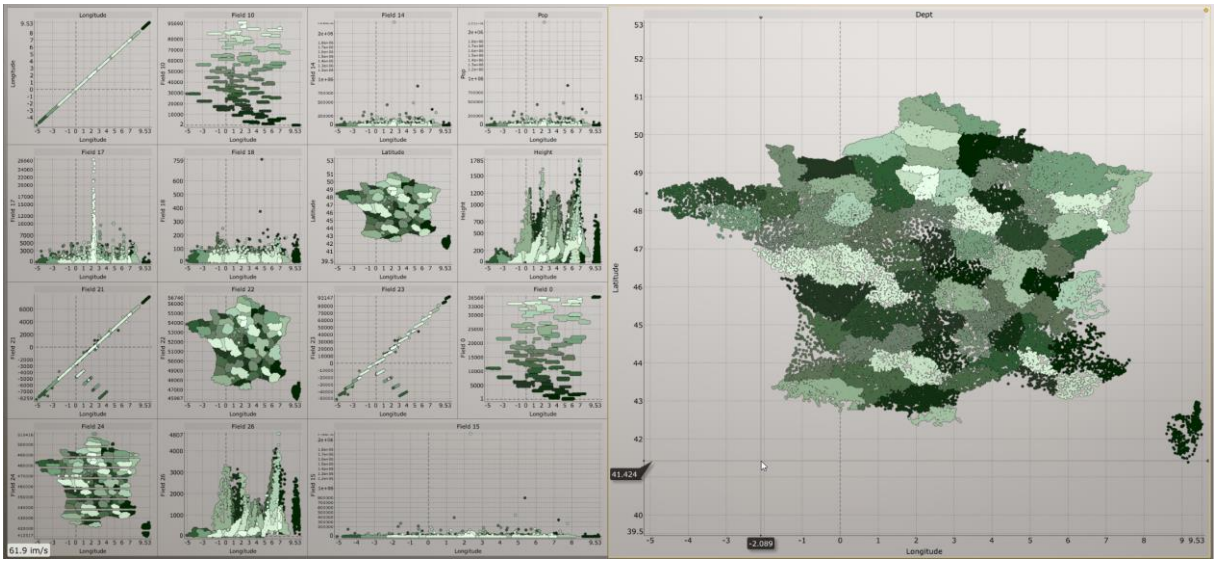
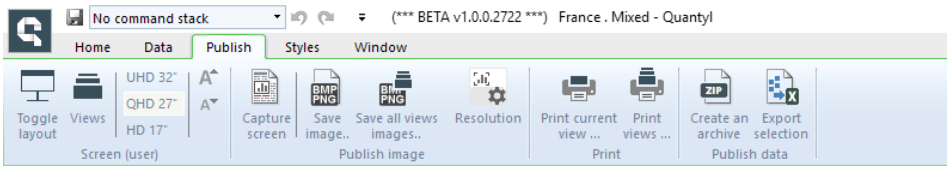


Figure 117 Y-axis faceting on all available measures

14 PUBLISH



You can print or generate images for the current or each view of the workspace. Note that **.png** will be generated with a **transparency background** which may be quite helpful in image integration:

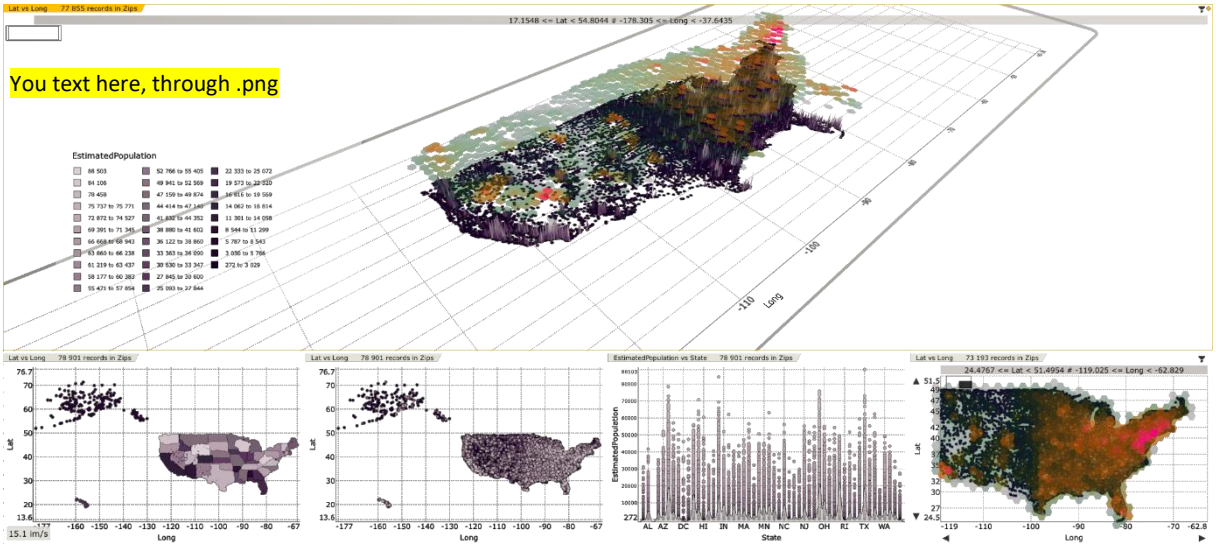


Figure 118 .png transparency

15 SCRIPTS INTEGRATION (PYTHON, R, UNKNOWN)

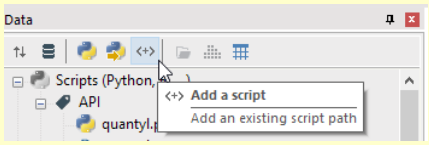
15.1 IMPORTING AND TRANSFORMING DATA WITH SCRIPTS

Quantyl provides a simple but efficient mechanism to run scripts from the application. Those scripts may be written in **Python**, **R** or any script language with a command line interpreter that accepts multiple arguments. It is important to note that the implementation as separate processes allows the use of different interpreter versions of your choice.

Those scripts have 3 different goals:

1- Importing and updating data in the Quantyl session
2- Transforming data or part of the data loaded in the Quantyl session
3- Exchanging data between a Quantyl session and a decorrelated script session

Python / R scripts are the way to go to extend data support. Scripts may connect to any data base you need through standard or specific Python or R connectors for example.

	Important
⚠	To be displayed in the Quantyl user interface, your scripts have to be placed in: **[my documents]\Quantyl\Scripts** folder. You still may add different scripts paths, one by one, in the user interface (data tree): 

Helpers-API modules are provided for both **Python and R**, they are automatically copied in

[my documents]\Quantyl\Scripts You may copy those files and derived them for your own purpose.

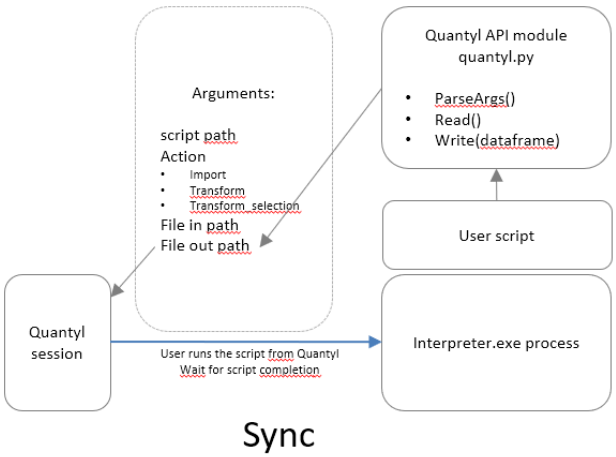
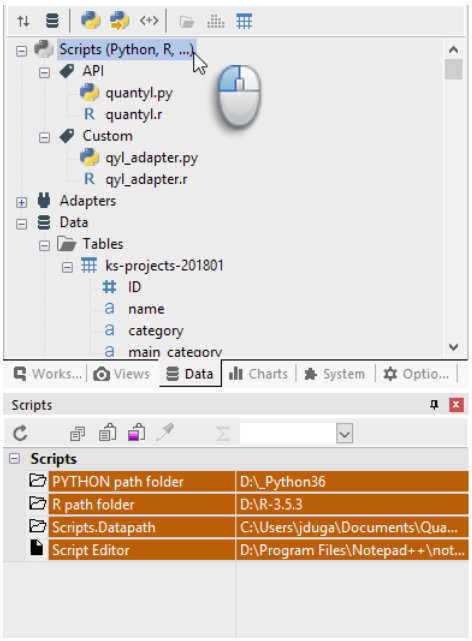


Figure 119 How script works in Synchronous mode from Quantyl

15.2 PYTHON INSTALL

You can point the path of your version of Python or use a stripped version downloaded by Quantyl. If Python is not found, you will be prompted to choose which version you want to use or install at script execution.

- Quantyl API module is copied in [my documents]\Quantyl\Scripts\quantyl.py
- Example of python script use is copied [my documents]\Quantyl\Scripts\qyl_adapter.py



15.3 R INSTALL

R install is **not** part of the Quantyl installation process. R has to be installed on your own.

Then you have to point the path where your R version is installed.

- Example of R script use is copied [my documents]\Quantyl\Scripts\qyl_adapter.r

15.4 UNKNOWN SCRIPTS TYPES

Quantyl support unknown scripts type as long as they parse arguments as:

interpreter.exe script_path action_type in_path out_path

In case of unknown scripts, the interpreter.exe path has to be specified in the script property page.

15.5 Working with scripts from Quantyl

By default, scripts and API module(s) are located in [my documents]\Quantyl\Scripts

The integration of Python and R is not tight to allow you to work with **your version** of Python or R. Quantyl runs the interpreter with specific arguments depending on what kind of task you want to achieve. That's the reason why you have to specify what action you are making when invoking the scripts.

The interpreter is launched from Quantyl as **a separate process** each time you are invoking a script.

Based on their file extension, Python (.py) and R (.r) scripts will be identified as such, and the convenient interpreter (python.exe or rscript.exe) will be searched on the specified path. I you want to work with unknown script types, you can specify the interpreter .exe in the script properties as in the screen captures below.

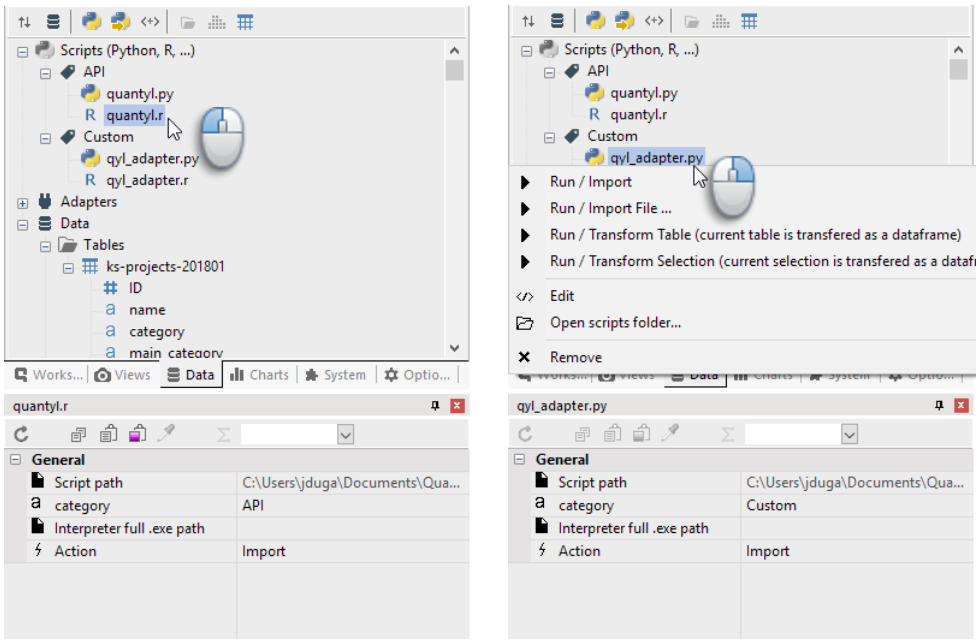


Figure 120 Running a script directly from the tree

The interpreter must understand the command line as:

interpreter.exe script_path action_type in_path out_path

argument	
interpreter.exe	Script interpreter full exe path
script_path	Script full path
action_type	It may be one of those choices: • import • transform • transform_selection
in_path	Path of the file to import if any
out_path	Path of the .csv file to export from the script

15.6 WORKING WITH SCRIPTS FROM EXTERNAL ENVIRONMENTS

Running script from Quantyl launches the interpreter process each time you invoke a script.

When you are working in a Python environment like Spyder or R environment with R Studio for example, you are in different **scenarios**:

- From your current process you want to inject data into a new Quantyl session or into the current Quantyl Workspace
- Or you should be able to retrieve data from the current Quantyl session into your current script environment

Solution	
Command line	You may launch a new Quantyl session with the input file as parameter, using the Quantyl command line
quantyl.write() API quantyl.py quantyl.r	You may simply export data (as .csv) in a specific folder monitored by Quantyl. Data will be automatically imported in the current workspace of the running session. You can use quantyl.write() API provided for Python and R users. There is no specific action required to import data in Quantyl when it is placed in this folder, the parsing process is automatic ([my documents]\Quantyl\TmpExchange\Import\ )
quantyl.read() API quantyl.py quantyl.r	Within an external script, you can read the temporary specific folder where Quantyl may have exported data.

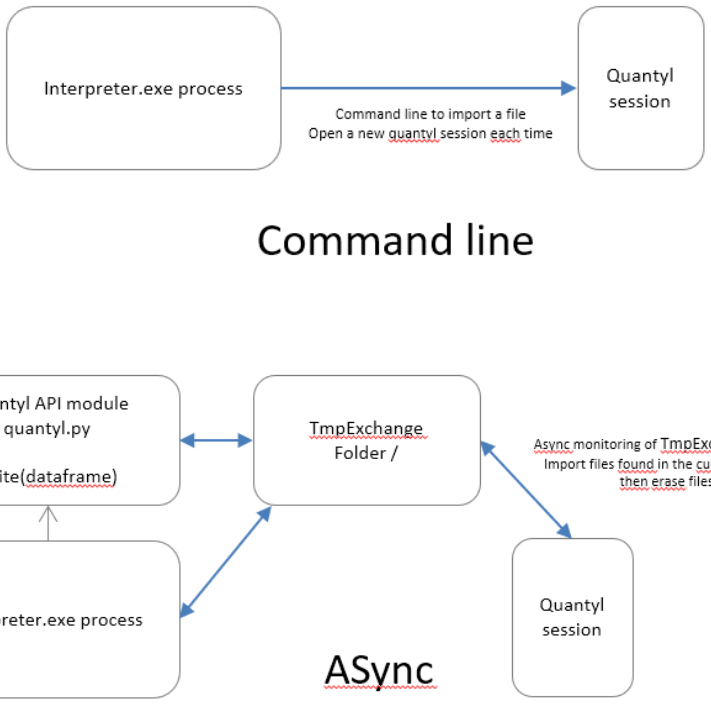


Figure 121 Use of [my documents]\Quantyl\TmpExchange\Import\ folder (asynchronous mode)

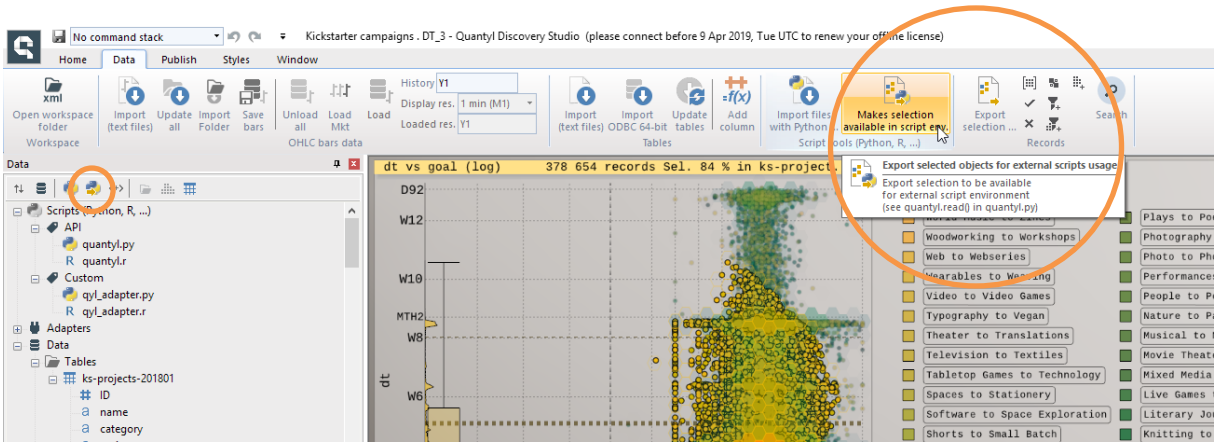
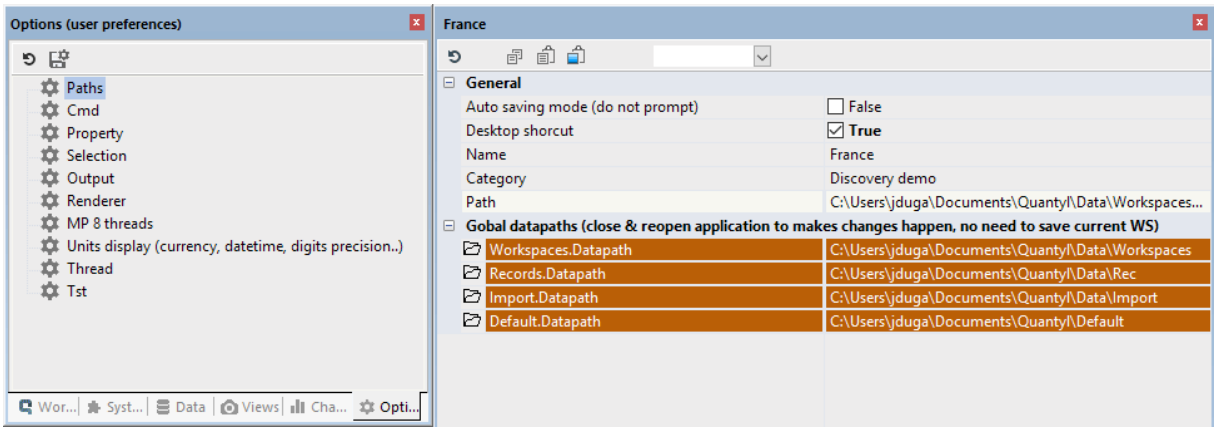


Figure 122 Export current data in a temporary file to be read by an external script asynchronously

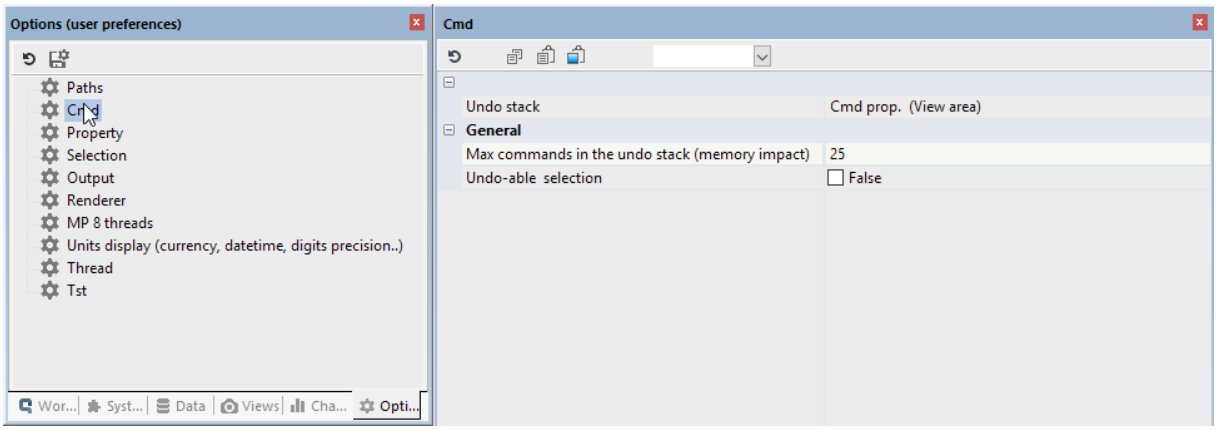
16 User options

User options are not saved with the workspace.

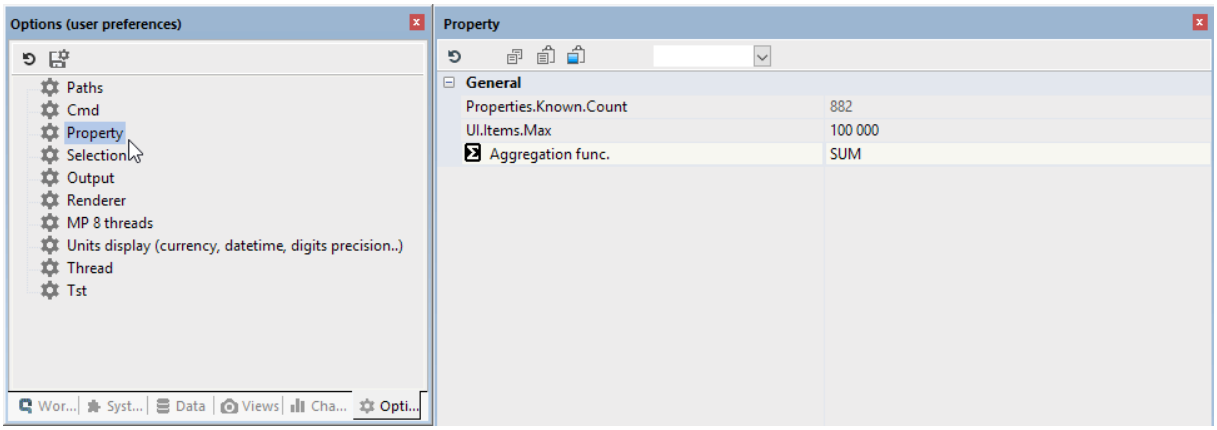
16.1 Paths



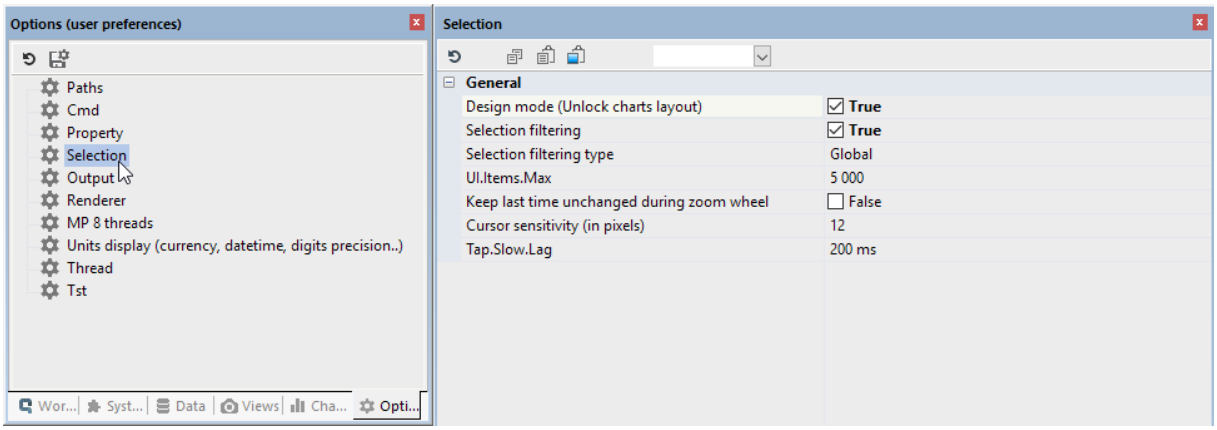
16.2 Commands



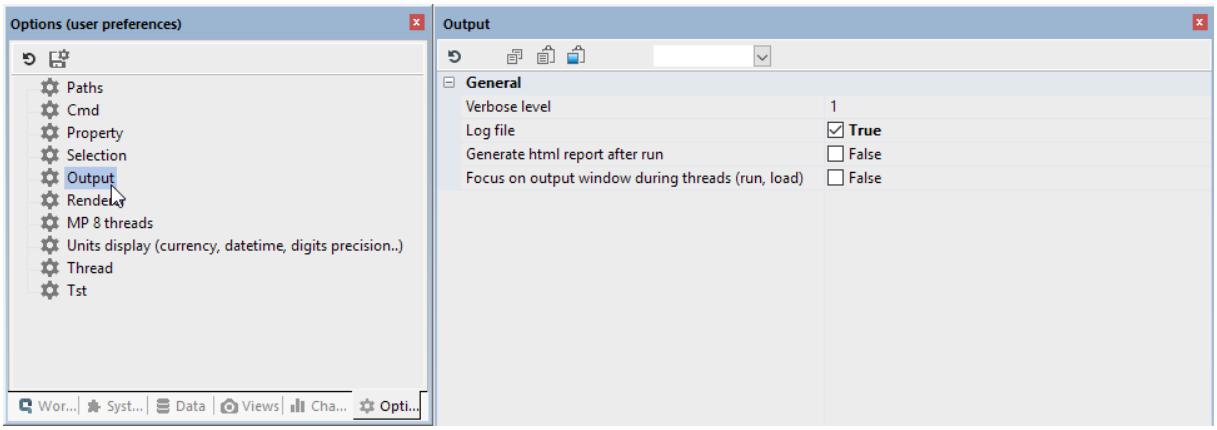
16.3 Properties



16.4 SELECTION

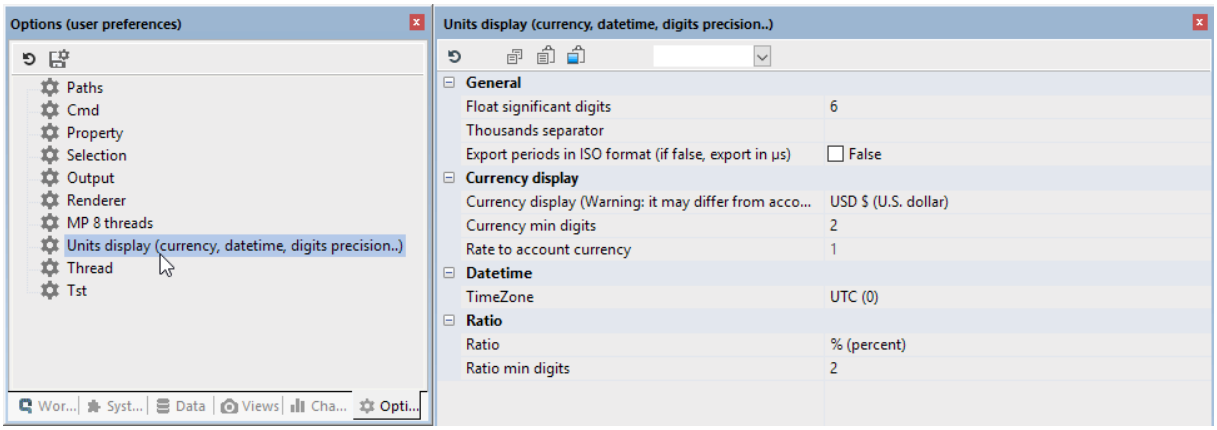


16.5 OUTPUT



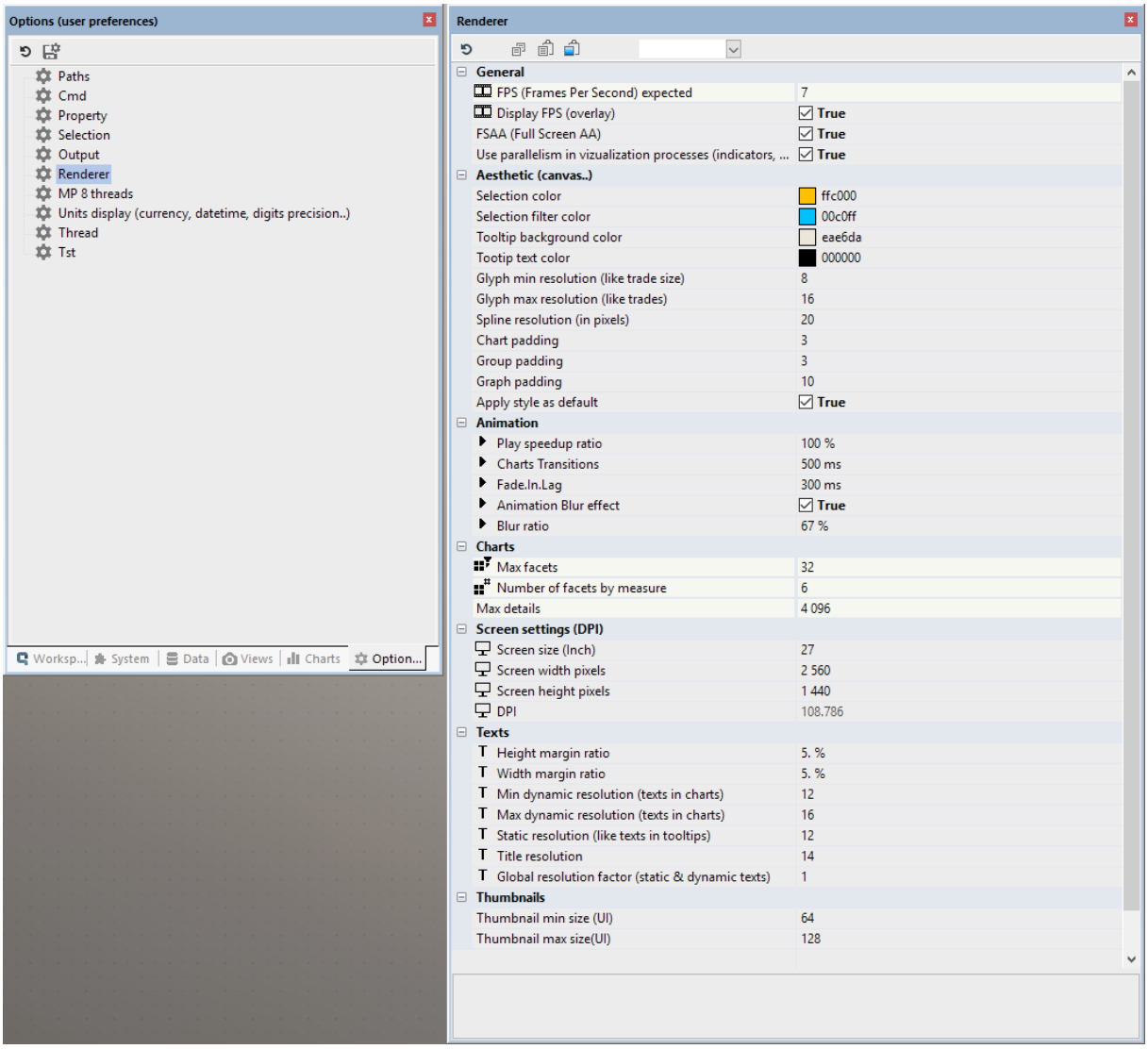
16.6 UNITS

Unit manager defines how property values are displayed when unit is known. It is just a formatting display mechanism; it doesn't modify the way values are internally managed and persisted.



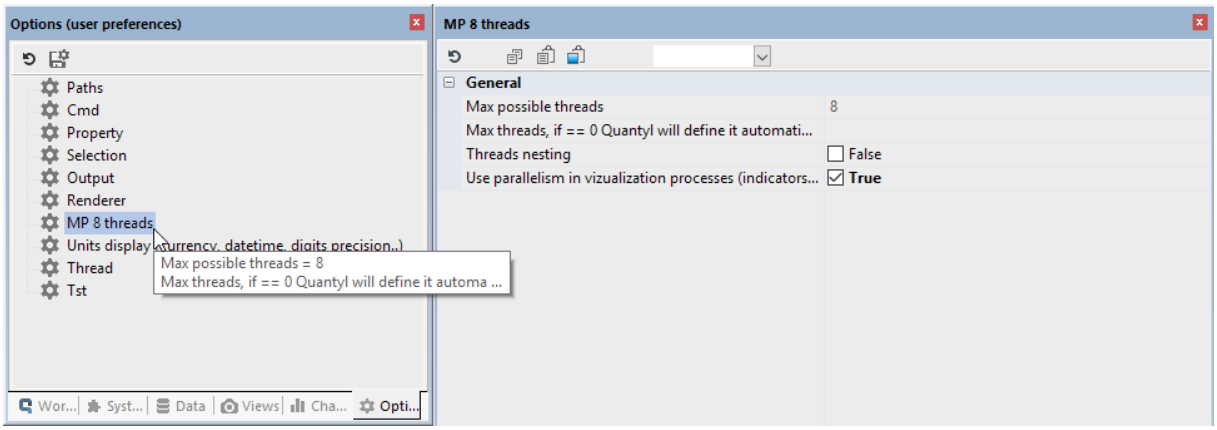
16.7 Rendering options (Texts size, selection color etc.)

Some graphical parameters are not saved with the view, they are part of user preferences.

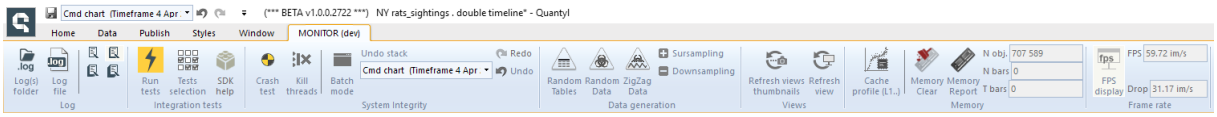


16.8 Multi-Processing

You can define here how much processing power (cores) you are allocating to Quantyl.



17 MONITOR



This ribbon category provides debugging tools to mainly monitor memory usage, frame rate, logs, and integration tests.

The **verbose level** defines how much information you can get in Output Window. Please note that verbose level may degrade performances a lot if too much information is generated.

18 Open architecture

Architecture of Quantyl is largely based on XML files (Input / Output) that can be easily handled by third party applications. Data model based on generic property mechanism is a guaranty of its simplicity.

18.1 Database / Filesystem

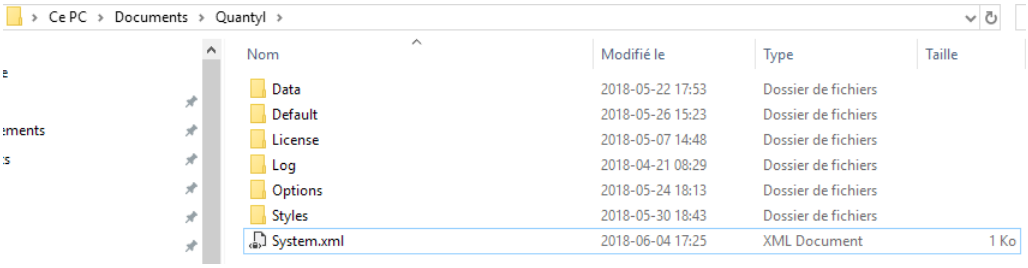
Quantyl data are structured in a standard file system (not a database) in **[my documents]** folder.

If you look in the file structure, you will see that many objects have their own dedicated xml file (each manager, each chart etc.). This simple modular approach facilitates partial loading, sharing, and reuse. Through dedicated APIs, data I/O may be replaced by custom database access to store and request those pieces of data.

18.2 Files system

A workspace is a **non-compressed folder**; it has its own files structure under the root.

Copying files and views between workspaces is possible, if you want to spread some view configurations in multiple workspaces.



Default structure:

Folder or file	Description
[My documents] \Quantyl\system.xml	Data paths and last workspace used.
[My documents] \Quantyl\Log\	Logs, you will find here.
[My documents] \Quantyl\Options\	User settings (cross workspaces).
[My documents] \Quantyl\Default\	User default (charts default).
[My documents] \Quantyl\Data\Rec	Contract data (Forex, Stock quotes) are shared between workspaces.
[My documents] \Quantyl\Data\Workspaces	Workspaces data. 1 sub-folder per workspace. The name of the folder is the name of the workspace.
[My documents] \Quantyl\Data\Scripts	Scripts and API modules (quantyl.py)

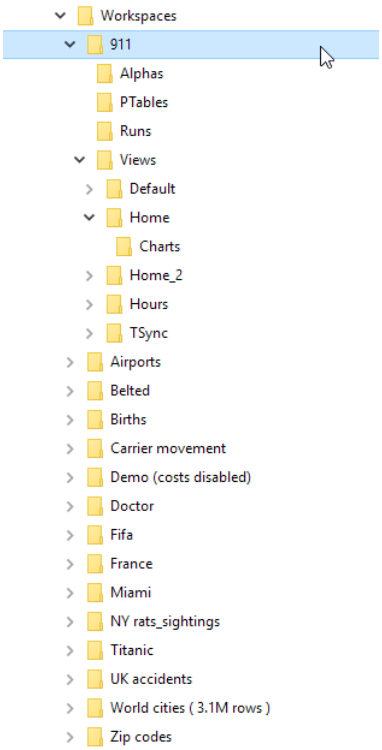


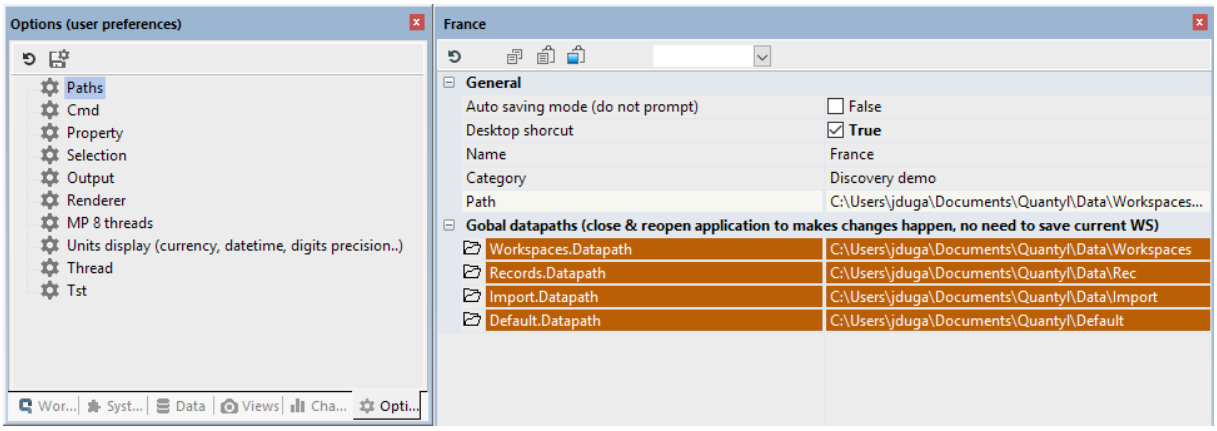
Figure 123 Workspaces structure

18.3 DATAPATH

Some datapath are used to retrieve default data without specifying the complete path. Those paths are exposed as properties of the current workspace. In fact, they are common to all workspaces. They can be changed to redirect on different sources.

Those properties are stored in **[my documents]\Quantyl\System.xml** file:

Property	Description
Workspaces.Datapath	Default workspaces folder
Records.Datapath	Default data records folder
Import.Datapath	Default import folder
Default.Datapath	Default templates folder [Default]\Charts



18.4 Environment variables

Variable	Description
QUANTYL_WORKSPACES_PATH	Paths directories for workspaces. It may be empty or not defined.
QUANTYL_PLUGINS_PATH	Paths directories for *Plugin*.dll. It may be empty or not defined.

18.5 Files types

File extension	Type	Description
.xml	**UTF8 XML** plain text	XML files are used to save system objects which hold properties. As all inputs, managers, alpha parameters, simulation settings, trades results, reports, charts etc. are managed through properties, it made all those data as open formats. Those files may be parsed or generated by third party applications with no license restriction. They are themselves “Open” format.
.qylrec	**Binary**	OHLC Record data (no compression)
.qyldf	**Binary**	Dataframe (table records)
.qyleq	**Binary**	Equity curve data (no compression)

18.6 DICTIONARY AND ALIAS FILES

Files	Description
[Rec]**Contracts.xml**	This file contains contract description, contract currency (ies) etc. This is a kind of **dictionary**.
[Rec]**Alias.txt**	This file contains alias used when you import data files. Example: *SampleStringToReplace=EUR_USD,* *USD_CAD,* *USD_CHF,* *USD_JPY,* *EUR_CHF*

18.7 UNICODE SUPPORT

Quantyl is UNICODE (UTF8) string based. Filenames and path containing UNICODE symbols are supported.

Limitation
UNICODE symbols are not yet correctly displayed in the viewport (charts, notes in tooltips etc.)

18.8 VERSIONS NUMBER DESCRIPTION

Version number of Quantyl application (exe) and components (dll) is decomposed as:

Major . Minor . Build Number [Beta]

Number	Description
Major	Major release Features change, Features adding, New APIs, UI organization may change Potentially API change
Minor	Minor release Features adding, Potentially New APIs
Build Number	Change at each Build. Possible minor feature(s) adding (completion of existing ones). If it's the only number to change: no API change Fix are complete refresh of the applications. As they are new installs, they do not require installing previous fixes (already included in the complete refresh).
[Beta] optional	Beta versions have "Beta" in their version name.

18.9 BATCH COMMAND LINE

Launch Quantyl.exe UI	
Quantyl.exe [worskpace_name | worskpace_path | Delimited text file(s) to import as table(s) (.csv .txt etc.)]	Launch Quantyl UI. If no WorkspacesPath is provided, the most recent workspace is opened by default. *Several Quantyl.exe sessions may coexist in memory. Nevertheless, there is no control on potential mismatched read/write operations that may occur (2 Quantyl working on the same workspace, same import etc.).*

As Quantyl.exe is a windows application with user interface, it returns immediately prompt if you launch it through a batch. It is why **QuantylCmd.exe** exists: it returns only after tasks performed.

Command line option []	Description (available results)
QuantylCmd.exe <WorkspacePath> -update	Workspace data tables are updated and workspace is saved.
QuantylCmd.exe <WorkspacePath used for import parameters> -importtickbars <file to import>	Import OHLC bars or tick data

Limitation

If you have not access to QuantylCmd.exe, you can still use same command line options with **Quantyl.exe**, it will perform the task but returns immediately.

A log file [My documents]\Quantyl\log.txt is generated during the batch.

18.10 LOG

Please see monitor section.

19 NAMING CONVENTIONS

You find below a sum-up of terms and objects used both in Quantyl user interface and in this document. Some terms are commonly used in trading but are **described here in the context of Quantyl usage** to avoid possible confusion.

<table>
<tr><th>Object
(by close domains)</th><th>Description</th></tr>
<tr><td>AKA</td><td>also known as</td></tr>
<tr><td>Add-in
AKA plug-in</td><td>Add-in are external software components that can modify, add or remove features in Quantyl application(s). Add-in mechanism allow customers and third-parties to extend Quantyl capabilities.</td></tr>
<tr><td>Table
Tabular data</td><td>A table is an arrangement of data in rows and columns</td></tr>
<tr><td>Contract</td><td>Any instrument (Forex or Stock)
Each contract is identified by a unique symbol (like “EUR_USD”, “AAPL”)</td></tr>
<tr><td>Analytics</td><td>Data Analytics is a multi-dimensional discipline. There is extensive use of mathematics and statistics, the use of descriptive techniques and predictive models to gain valuable knowledge from data—data analysis.</td></tr>
<tr><td>Manager</td><td>Root objects to manage specific tasks.
Each manager has its own properties and manages its children (Chart manager children are charts, data manager children are tables, contracts etc.)</td></tr>
<tr><td>STK</td><td>Stock contracts</td></tr>
<tr><td>FX</td><td>Foreign Exchange currency pair</td></tr>
<tr><td>Portfolio</td><td>List of contracts identified with a single name identifier</td></tr>
<tr><td>Symbol</td><td>Compact name to identify either contract or portfolio</td></tr>
<tr><td>Market</td><td>List of symbols names, which can be either contracts or portfolio</td></tr>
<tr><td>Time
Timestamp</td><td>A timestamp is way to reference date and time in an absolute manner. Timestamp is generally a large number which counts the number of seconds since 1.1.1970 at UTC time (posix).

Quantyl uses extended timestamp: number of µs since 1.1.1970 UTC.

We talk about time in a very large manner in Quantyl and in this document. It is often date and time.</td></tr>
<tr><td>Prices</td><td>Prices design STK prices as well as FX exchange rate.

<table><tr><td>Bid</td><td>Bid price</td></tr><tr><td>Ask</td><td>Ask price (Bid + spread)</td></tr><tr><td>Mid</td><td>(Bid+Ask)*0.5</td></tr></table>
We also combine OHLC bar prices with Bid, Ask and Mid. H ask is highest ask price of the bar, L bid, the lowest bid, etc.</td></tr>
<tr><td>OHLC bar
OHLCV bar</td><td>Bar with the following information:

<table><tr><td>O</td><td>Open price</td></tr></table></td></tr>
</table>

<table>
<tr><th>Object
(by close domains)</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>H</td><td>High price</td></tr>
<tr><td>L</td><td>Low price</td></tr>
<tr><td>C</td><td>Close price</td></tr>
<tr><td>TO</td><td>Time open in µs (posix extended timestamp)</td></tr>
<tr><td>TC</td><td>Time close in µs (posix extended timestamp)</td></tr>
<tr><td>V</td><td>Volume</td></tr>
<tr><td>S</td><td>Spread max between TO & TC</td></tr>
</table>
</td></tr>
<tr><td>Library</td><td>List of objects templates (that can be copied), that are used to extend Quantyl capability:

• Library of charts to extend or customize existing charts
Etc.</td></tr>
<tr><td>Plug-in</td><td>Plug-in are dll loaded at the launch of Quantyl. This is through Add-in mechanism that custom objects can be provided to Quantyl.
QylPlugin.dll is the main dll that provides system and managers. Other *Plugin.dll found are used to extend libraries.
Factories are used to create objects.</td></tr>
<tr><td>Property</td><td>Any distinct property exposed by an object.
Each property has:

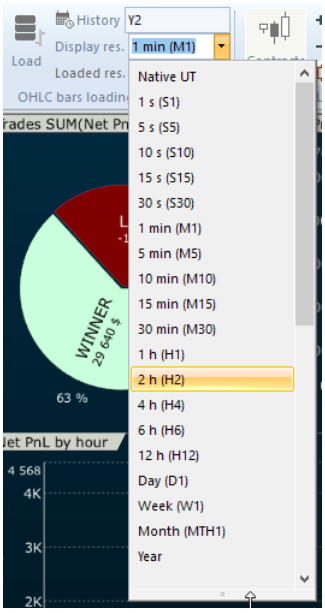
• A single name identifier (like “Color”)
• A value (not necessary numerical, it may be anything)
• An optional unit (Currency, duration etc.).

Property has other attributes, like the label displayed, that can be different than the property name, etc.
Properties are displayed in the property page when objects are selected.
Any objects settings or variable parameters are managed through properties.
Several hundreds of properties are available by default.

Property may be used as a specification or a result.</td></tr>
<tr><td>PStat</td><td>Property that can be used as a result statistic</td></tr>
<tr><td>Timeframe</td><td>Date-time window with a start date and an end date.
Max resolution of a timeframe is 1µs.</td></tr>
<tr><td>UT
AKA time period
AKA time interval</td><td>Units of time (measured in µseconds)

Generally manipulated and displayed with those compact naming:

• Yn for n years
• MTHn or Mon for n months (as 30.5 days)
• Wn for n weeks
• Dn for n days
• Hn for n hours
• Mn for n minutes</td></tr>
</table>

Object (by close domains)	Description
	• **Sn** for n seconds • **msn** for n milliseconds UT are generally used to define bar's period, displayed resolution. 
Resolution	Highest resolution has finest precision. Lowest resolution has rough resolution. Resolution on data time series is defined by its UT. • **Native data resolution** is the resolution of data files. • **Loaded data resolution** is the data resolution in memory. **Displayed resolution** is the resolution used to display data.
Sample (data sample)	Data at a given UT resolution in memory.
Down sampling	Process that reduces resolution of data (down sample UT < Initial UT).
DOW	Day Of the Week: • SUNDAY = 1, • MONDAY = 2, • TUESDAY = 3, • WEDNESDAY = 4, • THURSDAY = 5, • FRIDAY = 6, • SATURDAY = 7.
DST	Daylight Saving Time (DST) is a way of making better use of the natural daylight by setting your clock forward one hour during the summer months, and back again in the fall.
Viewport	Drawing window where charts are visualized

Object (by close domains)	Description
View	A view stores the properties of: • each chart • charts manager • viewport Views are not cross workspaces, they are owned by the current workspace. Views can be created, removed, invoked in the same workspace.
To pan	Make a panoramic view, by moving the mouse
Highlighting AKA "Mouse over"	Object under mouse is drawn in a distinctive manner, with a lighter selection color. A tooltip may be displayed.
Tooltip	Contextual information displayed during highlighting.
Contextual menu	Mouse Right click menu (or press and hold on a touchscreen)
Renderer	Manager that orchestrates drawing refresh.
UI	User Interface.
UX	User eXperience.
Morphic UI	Morphic user Interface.
ZUI	Zoomable User Interface.
OGL	OpenGL (2D & 3D graphical UI).
Command	A command is usually launched when user click on a button of the user interface to perform an action.
Mode	A mode is an option that is applied to some commands
I/O	Intput/Output
FS	Files System Refers generally to Quantyl files I/O instead of database I/O.
Callback	In computer programming, a callback is a piece of executable code that is passed as an argument to other code, which is expected to call back (execute) the argument at some convenient time. In Quantyl callbacks are managed through C++ inheritance model.
Horizontal scaling AKA scaling out	Horizontal scalability is the ability to increase capacity by connecting multiple hardware or software entities so that they work as a single logical unit.
Vertical scaling AKA scaling up	Vertical scalability increases capacity by adding more resources, such as more memory or an additional CPU, to a machine.
MapReduce	MapReduce is a programming model and an associated implementation. It is used for processing and for generating large data sets with a parallel, distributed algorithm (usually in a cluster).
[Quantyl]	Quantyl installation path.
[my Quantyl documents]	Documents Quantyl path.
GPU	Graphics Processing Unit. GPU technologies are also used in high performance parallel computing.
GB	Giga Bytes.
MB	Mega Bytes.

Object (by close domains)	Description
API	Application Programming Interface.
Framework	A software framework, in computer programming, is an abstraction in which common code providing generic functionality can be selectively overridden or specialized by user code providing specific functionality. Frameworks are similar to software libraries in that they are reusable abstractions of code wrapped in a well-defined API. Unlike libraries, however, the overall program's flow of control is not dictated by the caller, but by the framework. This inversion of control is the distinguishing feature of software frameworks.
SDK	A software development kit (SDK or "devkit") is typically a set of development tools that allows a software engineer to create applications for a certain software package, software framework.
OLAP	OLAP is an acronym for online analytical processing.
AES	Aesthetical properties, generally used for color, size or shape data mapping.
PDF	Probability density function
CDF	Cumulative density function
Stdev	Standard deviation

20 FAQ (Frequently Asked Questions)

The complete FAQ is here: https://www.quantyl.com/faq/

How to benefit from GPU power?	Quantyl current release is **not** GPU accelerated. GPU parallelism is an awesome tool when you want to speed up intensive parallel computation with low branching. It's not good for all but really shines in this area. GPU acceleration is one of the main R&D axes; so future release will certainly benefit of GPU speed-up.
How many rows are supported?	It is not possible to answer without specific data and hardware spec (CPUs, memory, cache...). The number and the nature of columns matter a lot, and number of charts and their types must be taken in account too. Instead of answering this question, we offer free trial on your own data and a bunch of interesting data samples. We think that the nature and size of those samples (up to 3.1M rows) will give you a good taste on how Quantyl is performing in those kinds of context, in your environment.
How to deal with huge data?	Avoid importing irrelevant/unnecessary columns you don't care of. String type columns are far more computing intensive than numerical ones; **especially when text strings cardinality is high**. Use sampling if possible, when you import data.
Is it possible to import large tick files data (several GBytes)?	Yes, Quantyl performs import per chunk of data file. The complete file is not completely loaded in memory to avoid memory peak.
I got a crash at launch, my workspace is corrupted?	Delete [my document\Quantyl\system.xml] file that point on the corrupted workspace and launch Quantyl again.
Display is slow, what happens?	You can modify the responsiveness in renderer manager properties (increase FPS "Frame per second"), it will display less info when you move the mouse.

How to avoid "animations" or change their speed?	This is a property of renderer manager that you set to 0 (no animated transitions)
"Cold start": First load in Quantyl is sometimes slower?	It is always fastest to re-load data previously loaded because you benefit from the disk cache.
Quantyl is frozen during a long time when launched?	Quantyl cartographies Import\ folder and Rec\ folders recursively at each start. Depending on the number of files and folders to scan, it can spend time. Move or erase non-necessary data if needed, or reference other import folder as below: You can also modify directly [my documents]\Quantyl\System.xml
If a Quantyl.exe is already launched, new Quantyl.exe are opening "blank"?	This is a normal behavior and a security to avoid opening 2 occurrences of the same workspace in memory. It is not forbidden but avoided by default. If your workspace crashes at launch for any reason, it also avoids locking and repeating the systemic crash at launch.
Is there a limit of number of multi-cores supported?	Yes, there is a hard-coded limit for the number of threads (256 `QYL_MAX_THREADS`). It is used to cache some data per thread in vectors. It can be stretched easily in the next releases.
Is this document exhaustive?	No, this document gives key features and details on how they are working. Some obvious commands are self-described themselves by their tooltip.

21 Release note(s)

https://www.quantyl.com/support/

22 Credits

* This product includes GAlib, a library of genetic algorithm components. Copyright Massachusetts Institute of Technology and Matthew Wall. (please check [Install]\Doc\ThirdPartyComponents\GALIB_COPYRIGHT.txt).

* This product includes QuantLIB library (please check [Install]\Doc\ThirdPartyComponents\QUANTLIB_LICENSE.txt).

* This product includes FANN library, the Fast Artificial Neural Network Library (fann), under the terms of the GNU Lesser General Public License (LGPL)

* This product includes RapidXMLlibrary under MIT license (please check [Install]\Doc\ThirdPartyComponents\RapidXML_LICENSE.txt)

* This product includes C++ Mathematical Expression Toolkit Library (Arash Partow (1999-2017)) under MIT license (please check [Install]\Doc\ThirdPartyComponents\MIT_LICENSE.txt)

* This product includes fastapprox library Copyright (C) 2011 Paul Mineiro (please check [Install]\Doc\ThirdPartyComponents\fastapprox_LICENSE.txt)

* This product includes some templates of SGL library, a Scene Graph Library from Scott McMillan, under the terms of the GNU Library General Public License v2

* This product includes 7-Zip 7z.exe under the terms of GNU Lesser General Public License v2.1 (please check [Install]\Doc\ThirdPartyComponents\7-ZIP_LICENSE.txt)

* This product includes csvtk (Wei Shen) under MIT license (please check [Install]\Doc\ThirdPartyComponents\MIT_LICENSE.txt)

* This product includes Intel® TBB (Threading Building Blocks) under the terms of Apache 2.0 license

* This product may install a stripped version of Python® (please check [Install]\Doc\ThirdPartyComponents\PYTHON_LICENSE.txt)

"Python®" is a registered trademark of the Python Software Foundation

/// Included data (as data demo samples): ///

* [UK accidents demo...] This product includes some data coming from public sector information licensed under the Open Government Licence v3.0.

* [World cities demo...] This product includes some data created by MaxMind, available from http://www.maxmind.com/

* [Rats sighting...] This product includes some data under CC0 1.0 Universal (CC0 1.0) Public Domain Dedication

* [Airports demo...] This product contains some data under Database Contents License (DbCL) v1.0

* [Fifa demo...] This product contains some data under Attribution-ShareAlike 4.0 International (CC BY-SA 4.0), Creative Commons, license (https://creativecommons.org/licenses/by-sa/4.0/)

* [911 demo...] This product contains some data under Open Database License (ODbL) v1.0 (https://opendatacommons.org/licenses/odbl/1.0/)