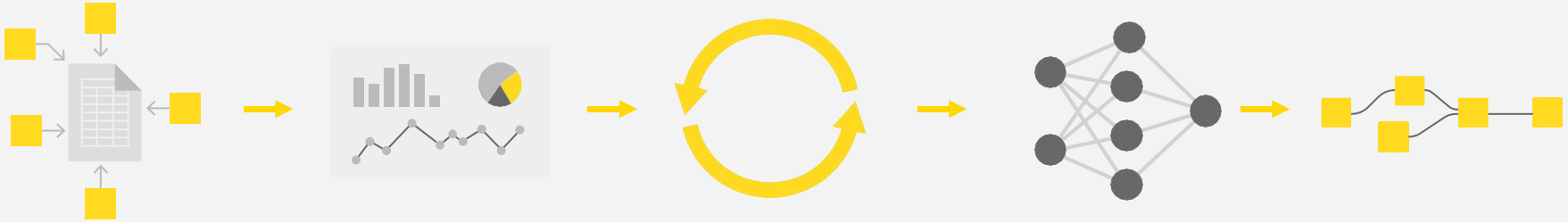
# Data Visualizations

# Who is this course for? Data and business analysts

| Data Analyst | Data Scientist | Data Engineer |
|---|---|---|
| Data acquisition, cleaning, analysis, visualizations, descriptive statistics, reporting, dashboards. | Data pre-processing, training machine learning and statistics algorithms, modeling, predicting. | Integrating various data sources, building data pipelines (ETL, ELT), databases, data lakes, data warehouses, file systems, and/or data mart maintenance, monitoring and testing. |

# Data analytics within an organization



**Data Collection**

Gathering data from different sources

**Data Visualization**

Data visualization and statistics

**Data Preprocessing**

Cleaning the data; checking quality

**Machine Learning**

Modeling and evaluation

**Deployment for Users**

Production model

# Data analytics for this course

**Data Collection**

Gathering data from different sources

**Data Visualization**

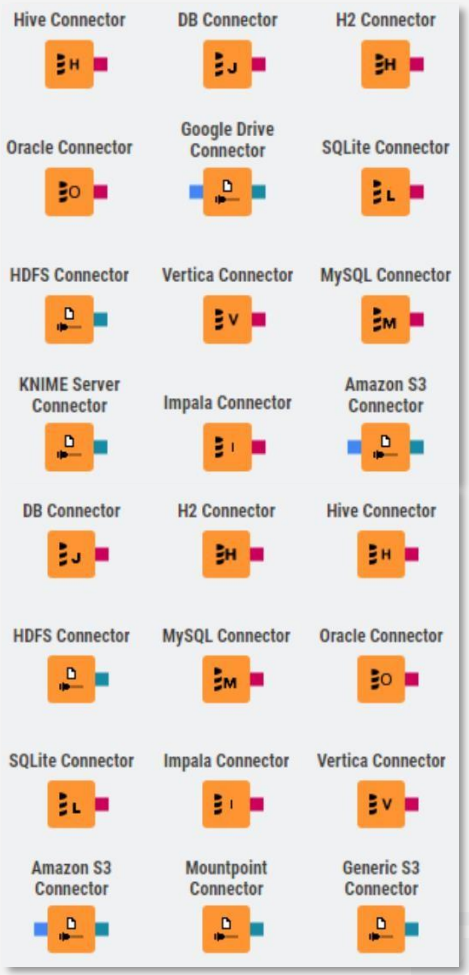
Data visualization and statistics

**Data Preprocessing**

Cleaning the data; checking quality

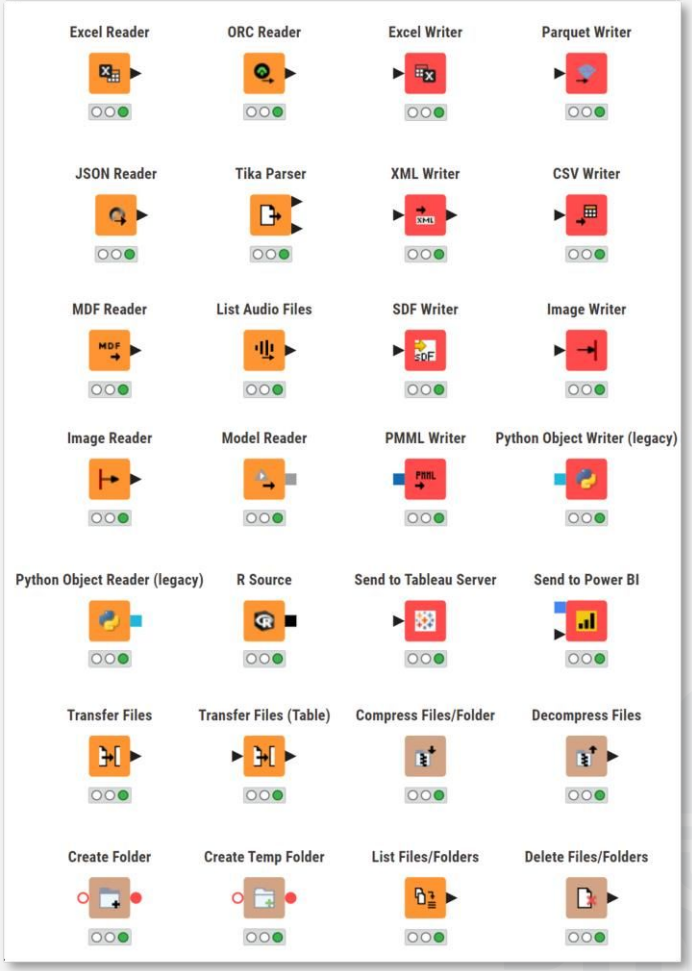
## Database nodes

- Dedicated nodes to connect to specific Databases
- Hive and Impala connector part of the KNIME Big Data Connectors extension
- General Database Connector
  - Register new JDBC driver via File -> Preferences -> KNIME -> Databases



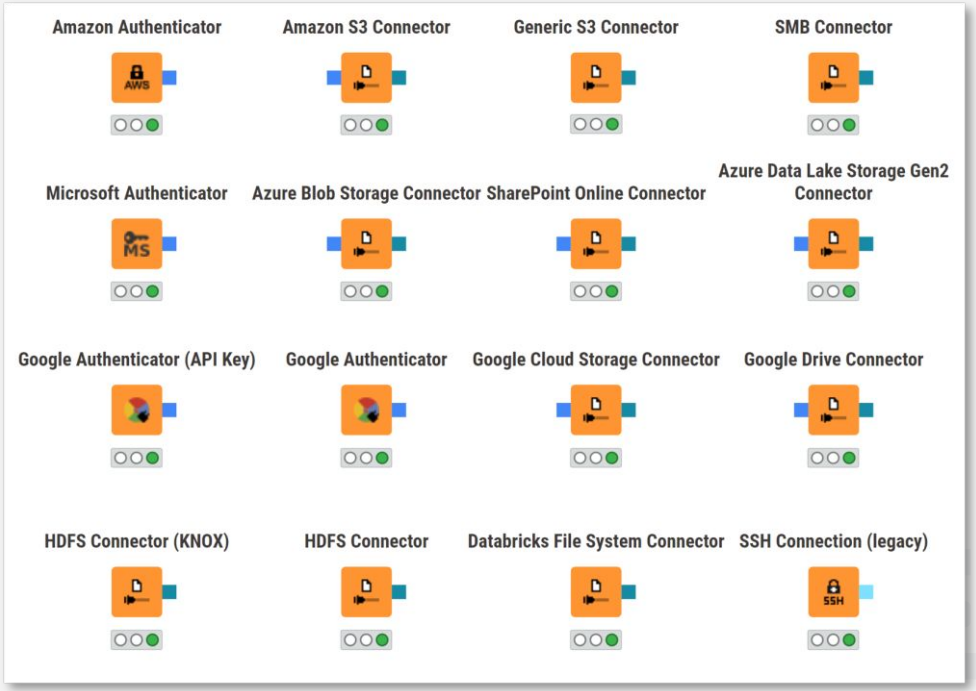
# Reader/Writer/Utility nodes

- Reading/writing **tabular, structured, textual**, chemical data, audio, image, and model files
- Reading one or **multiple** files
- Support of integrations: **Python**, R,
  H2O, **PowerBI, Tableau**



# Authentication/Connector nodes

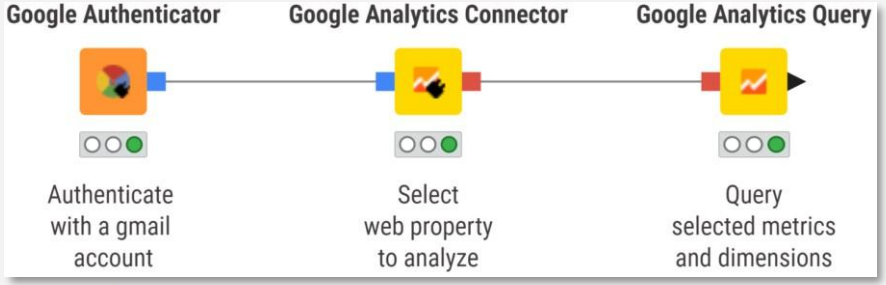
- Connected file systems
  - File systems with external authentication
    - Amazon
    - Microsoft
    - **Google**
  - File systems without external authentication
    - Databricks
    - HDFS, httpFS
    - SSH, HTTP(S), KNIME Business Hub



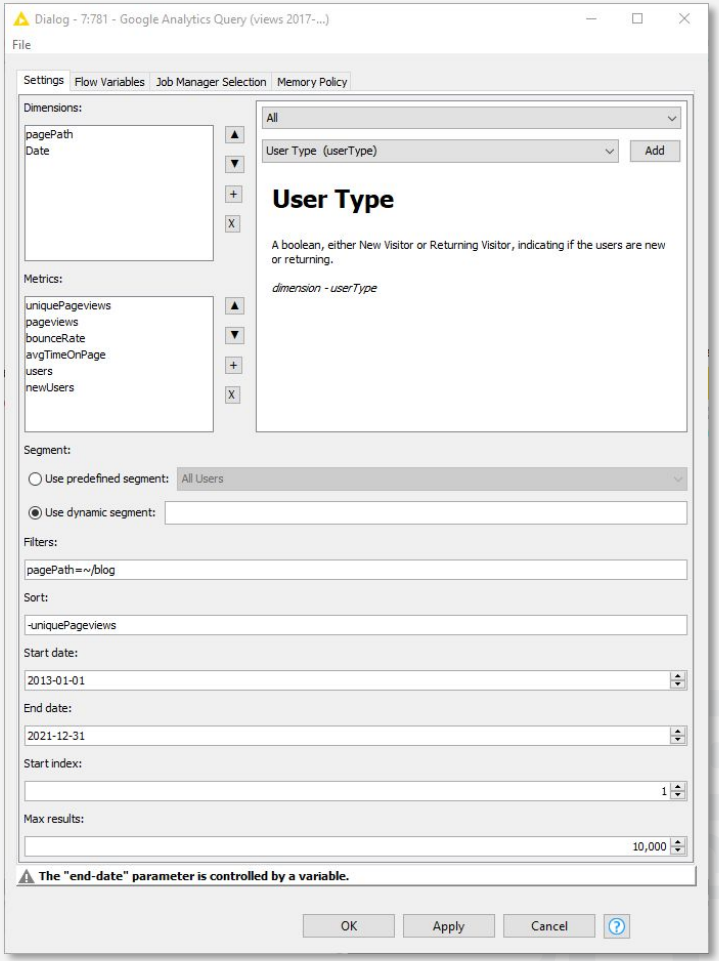
Google analytics extension

- Access Google Analytics data
  - Assemble the queries to extract metrics and dimensions of interest
- No need to write API requests

Select metrics and dimensions, filter, restrict period, sort, etc.




Authenticate via pop- up window (Oauth2)



# Google sheets

- Access data stored in Google Services:
  - Read data from Google Sheets
  - Transform in KNIME
  - Modify existing sheets
- No need to write API requests

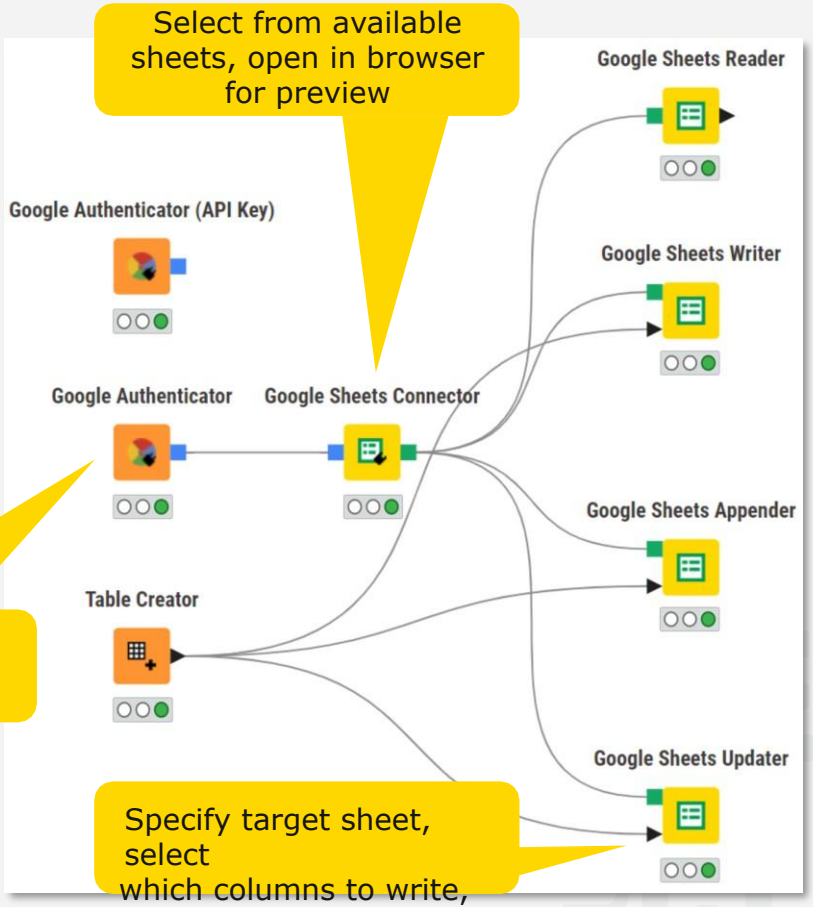
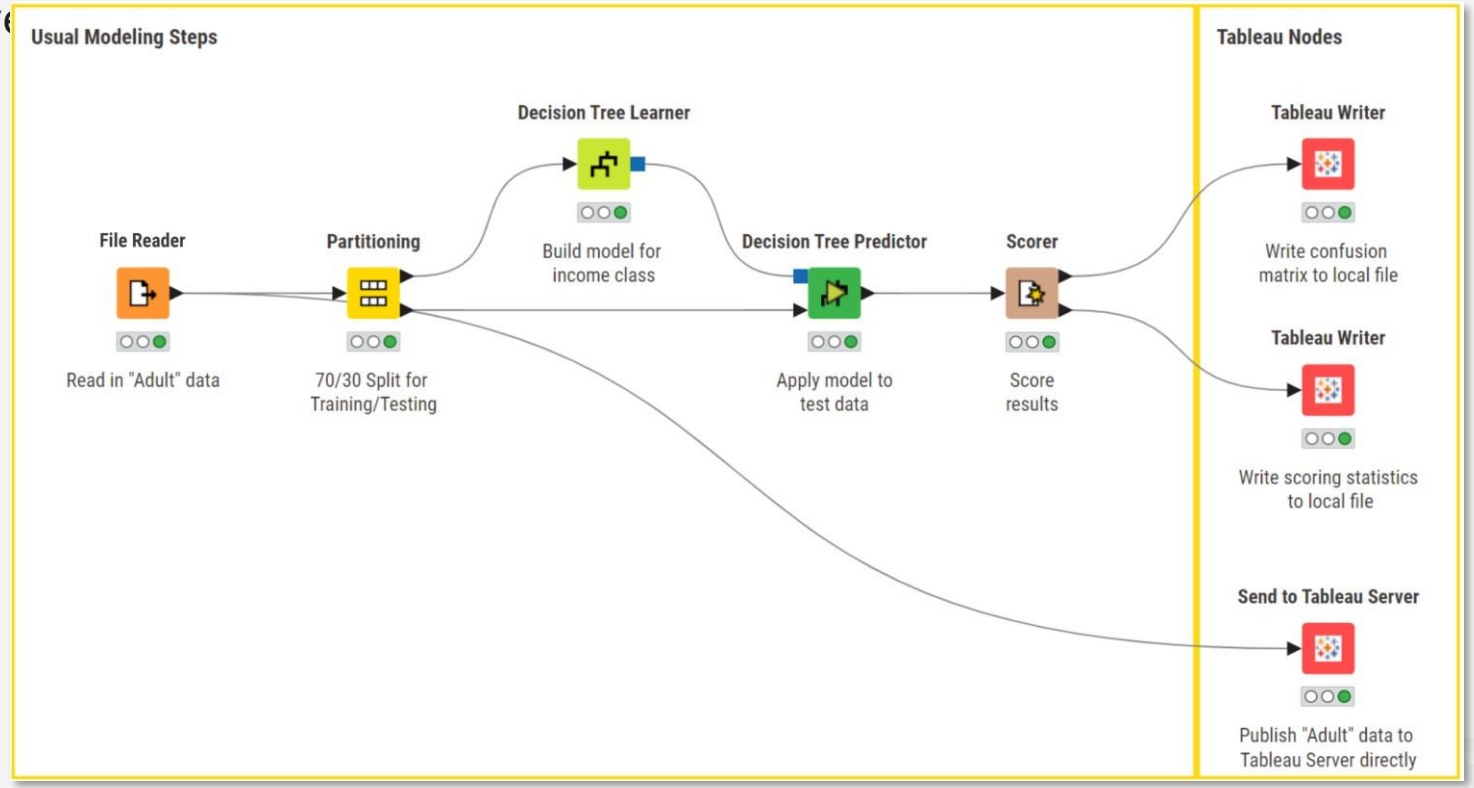


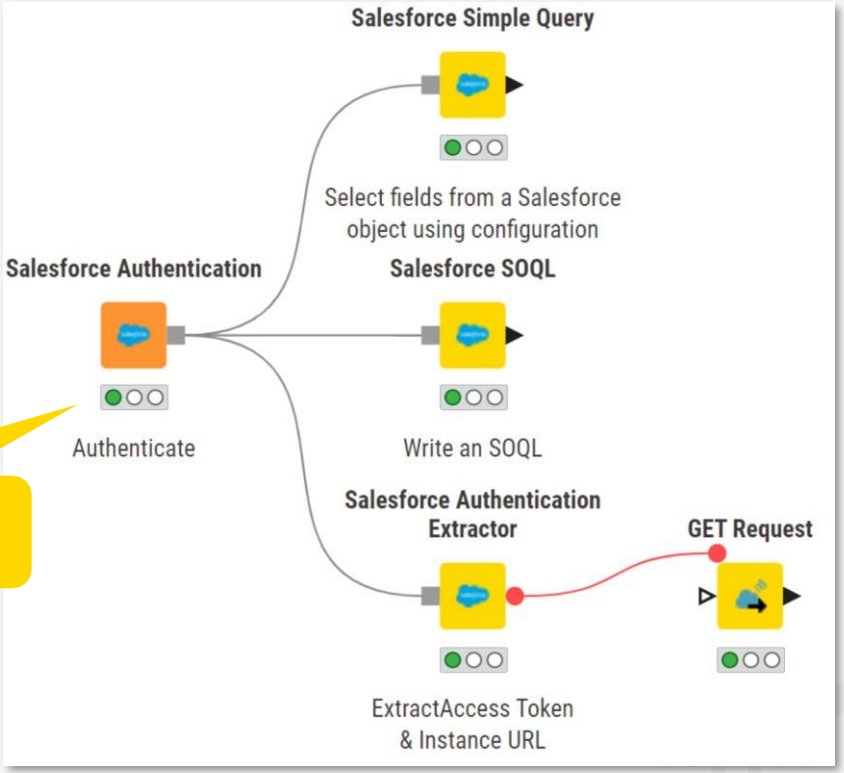
Tableau [extension]

- Write tableau files/send to Tableau serve



Salesforce extension

- Access Salesforce data
- Write queries and build further REST requests

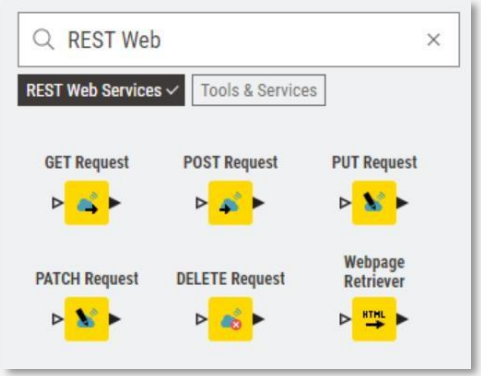


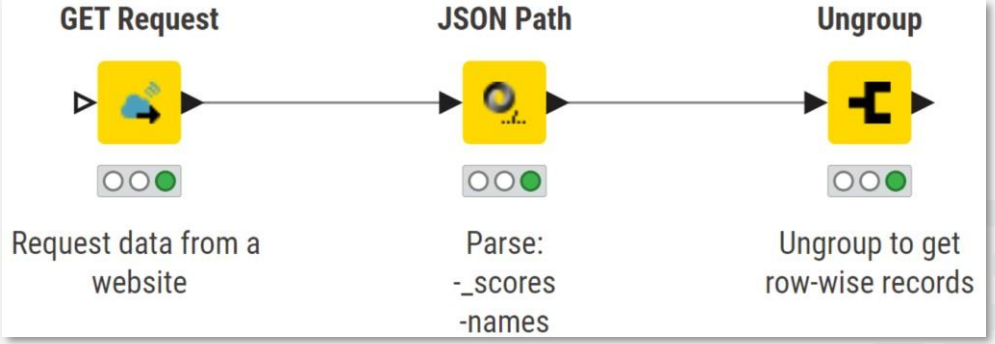

Authenticate via Oauth2
or username & password

# RESTful API extension

- Execute **Re**presentational **S**tate **T**ransfer commands
  - RESTful APIs are Web Service APIs that adhere to the REST constraints
  - One the most predominant **architectures for obtaining and managing data** across applications

- Existing KNIME nodes



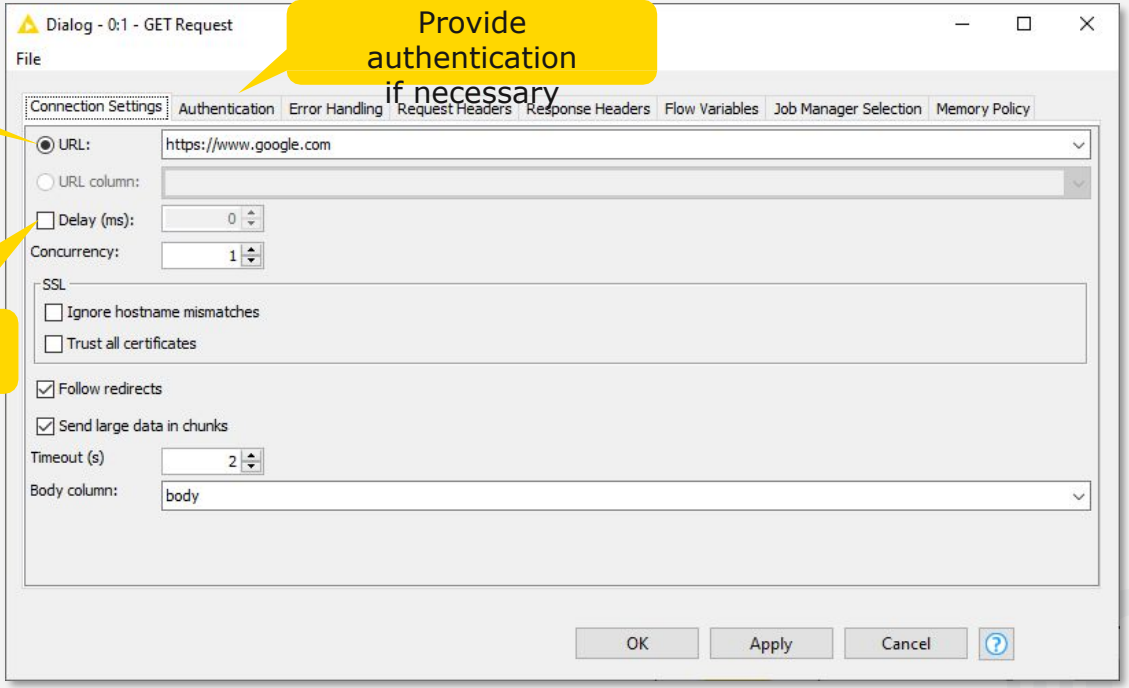



# RESTful web services / API


Enter URL, or use from column

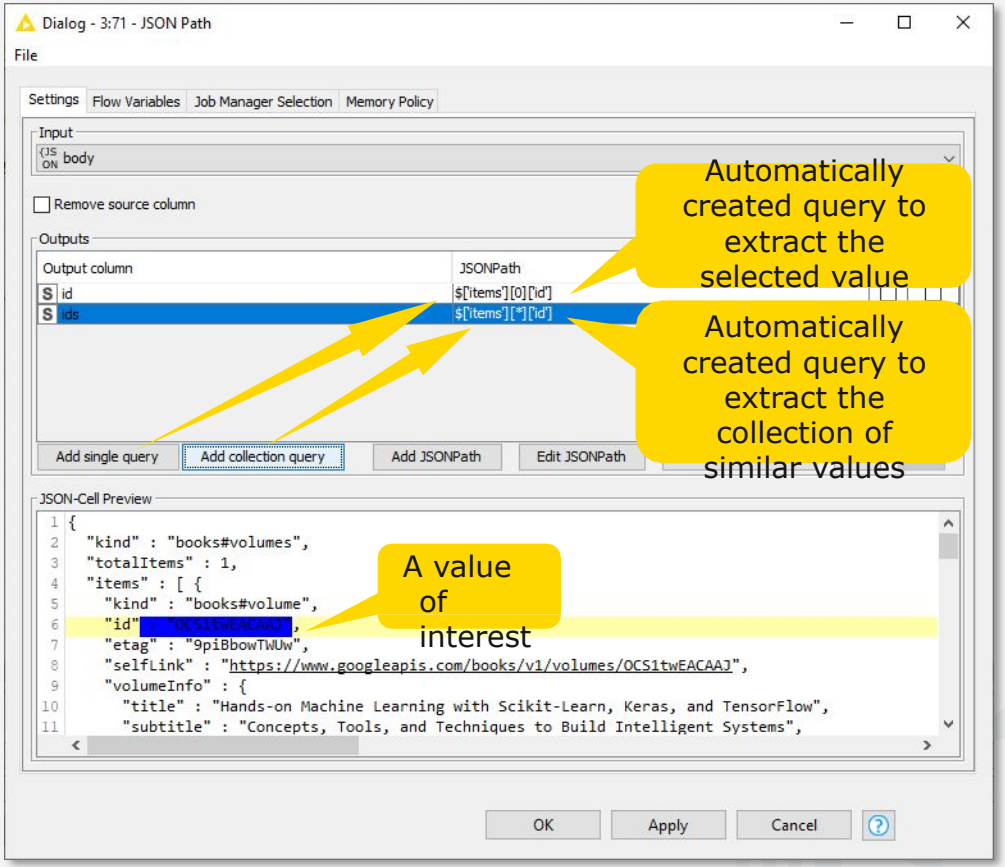
Add delay between individual requests

Provide authentication if necessary



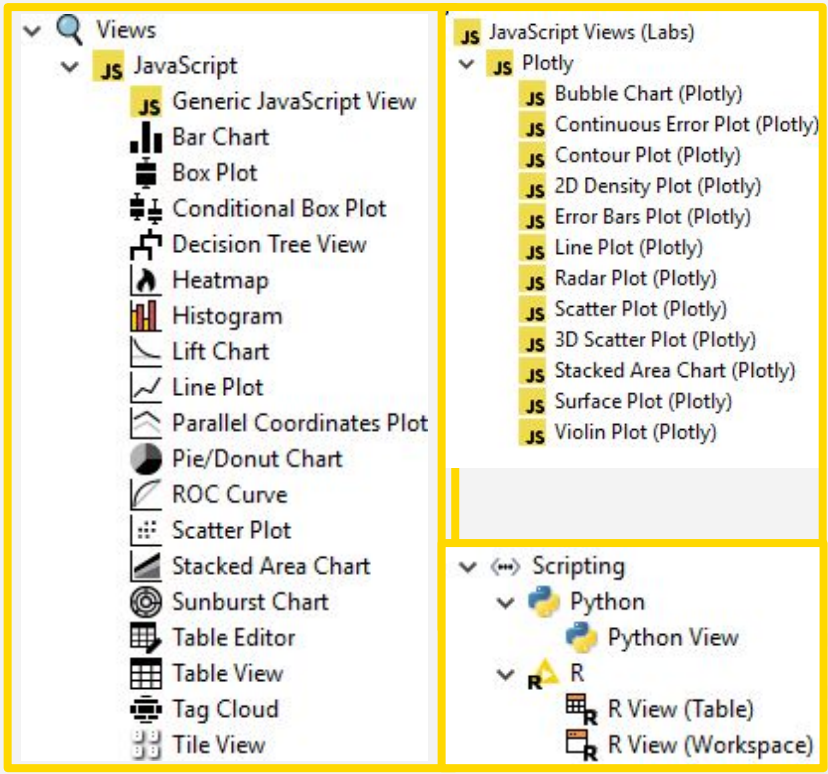
# JSON and XML parsing tips

- Use the JSON Path node to query the JSON file and extract parameters
- Editor window simplifies construction of JSON queries by auto-generating them
  - Select the value of interest in the JSON- Cell Preview and use the buttons to automatically add a query to extract this single value or a collection of similar values
  - OR write a JSONPath query manually
- Analogously with Xpath node for XML



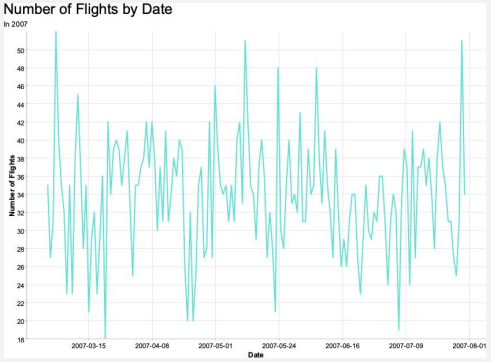
# Data Visualization

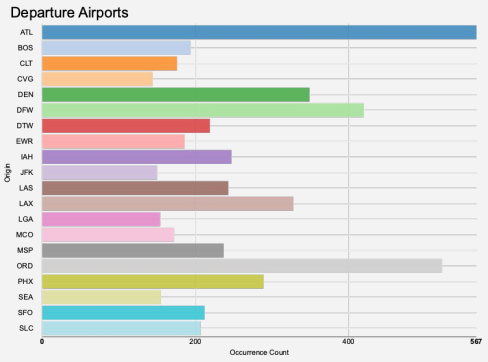
- Large selection of easy to use visualization nodes
  - Web-based and interactive
  - Dedicated nodes,
  - no scripting required
- Plotly nodes
  - Similar but integrated from an external library
- R and Python View nodes for highly customizable graphics
  - Require scripting

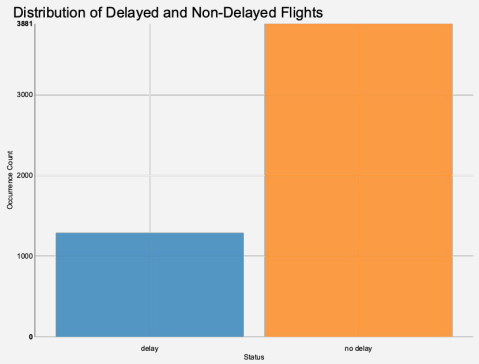


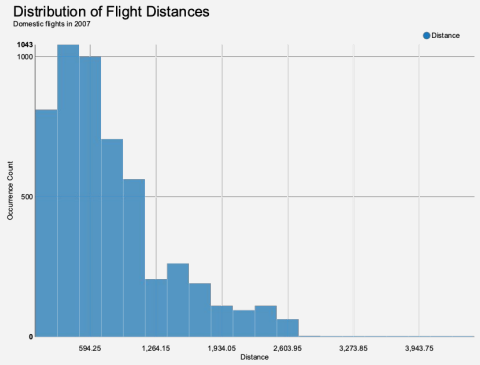
# Data Visualization Charts and Tables

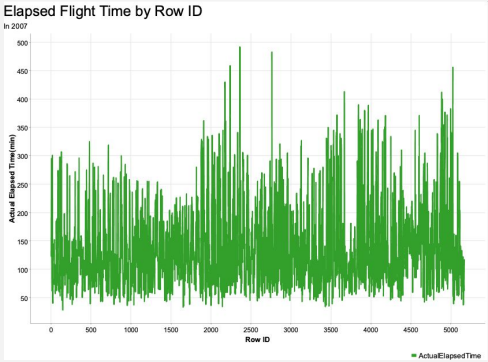
# Visualizations Using One Column

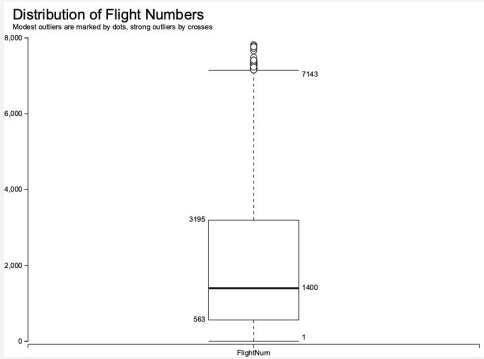


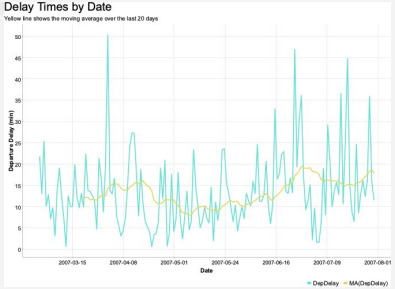



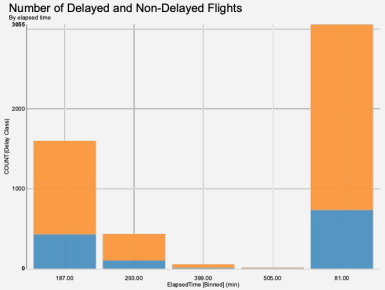



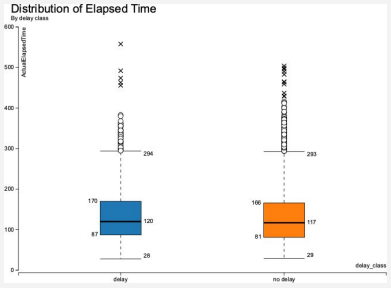



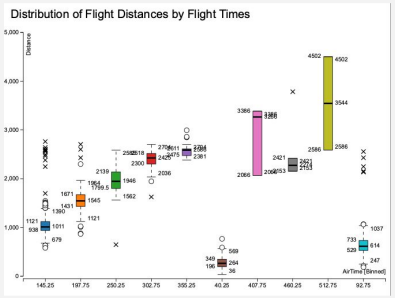



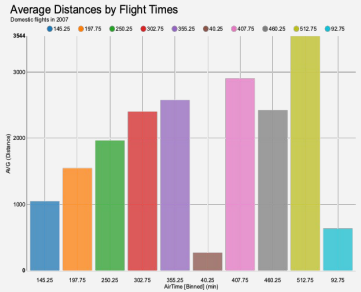
# Visualizations Using Two Columns

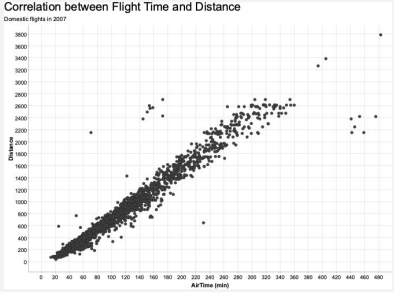


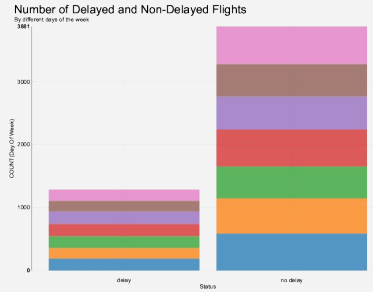



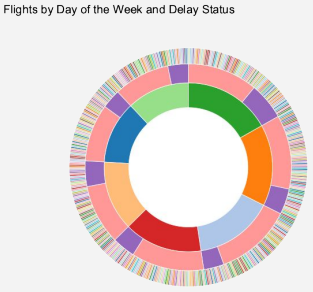



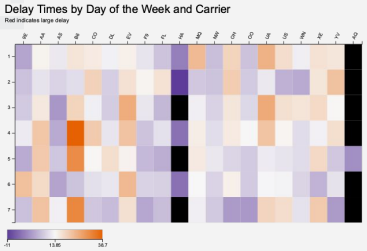



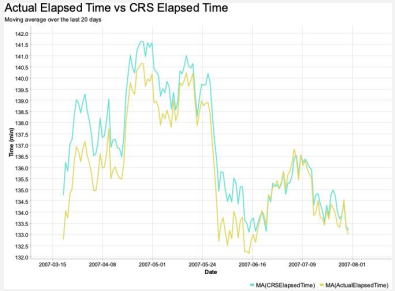



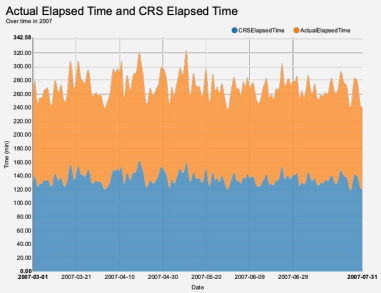



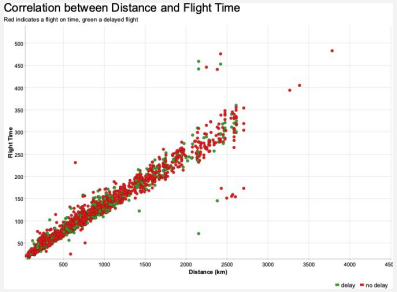



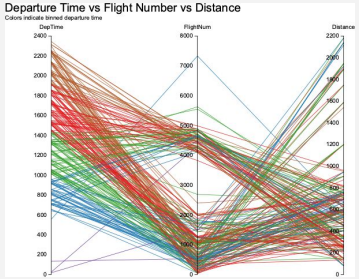
# Visualizations Using Three Columns

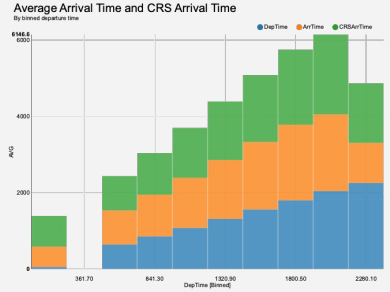


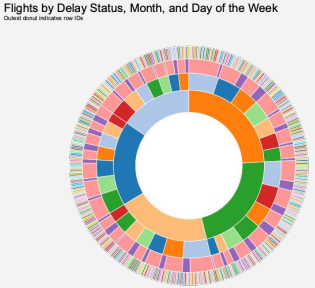



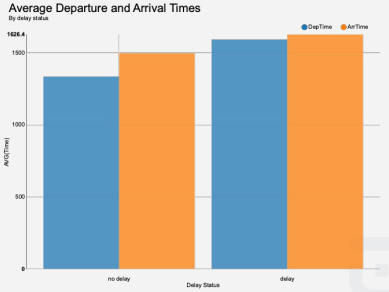



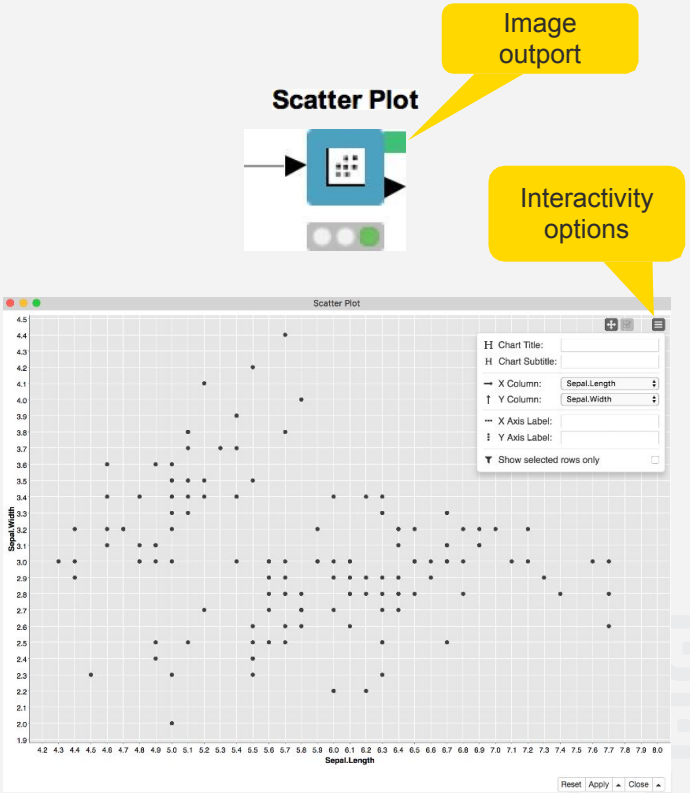



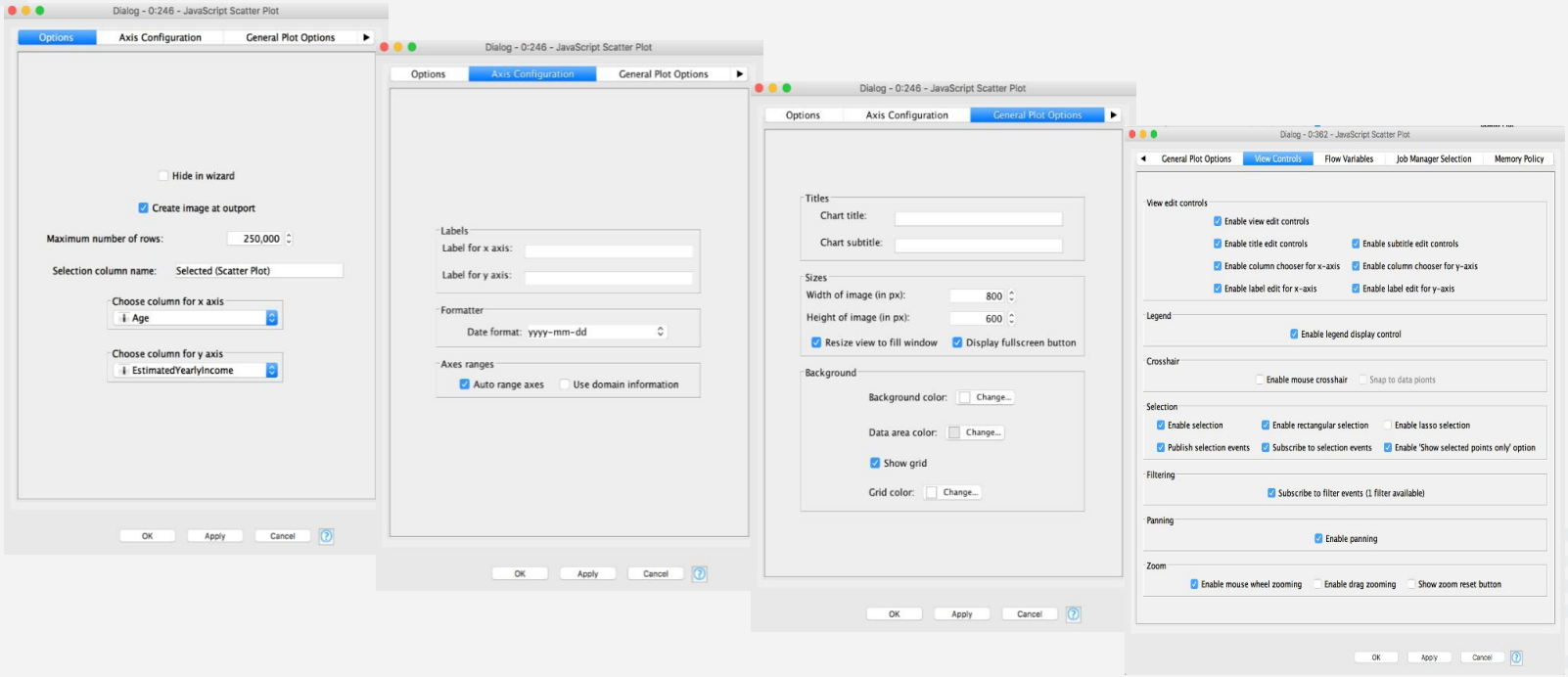
# New Node: Scatter Plot

- Plots different columns on X and Y
- Displays data including color information
- Produces an interactive view and an image
- Select data points and publish selection to other views



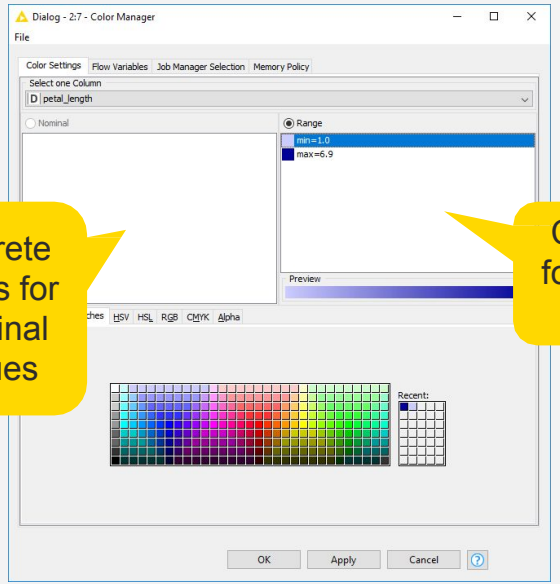
# New Node: Scatter Plot

- Four configuration tabs



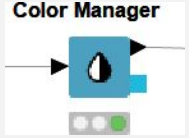
# New Node: Color Manager

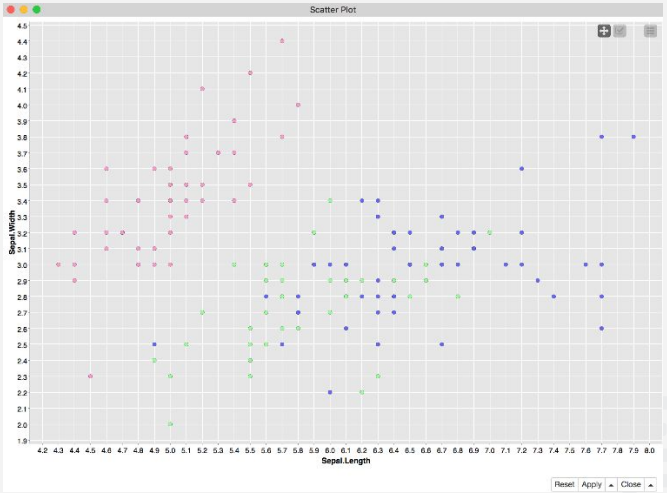
- Color by nominal or continuous values
- Sync colors between views using the color model port and Color Appender node




Discrete colors for nominal values

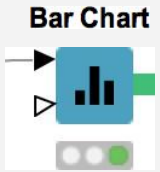
Color range for numerical values

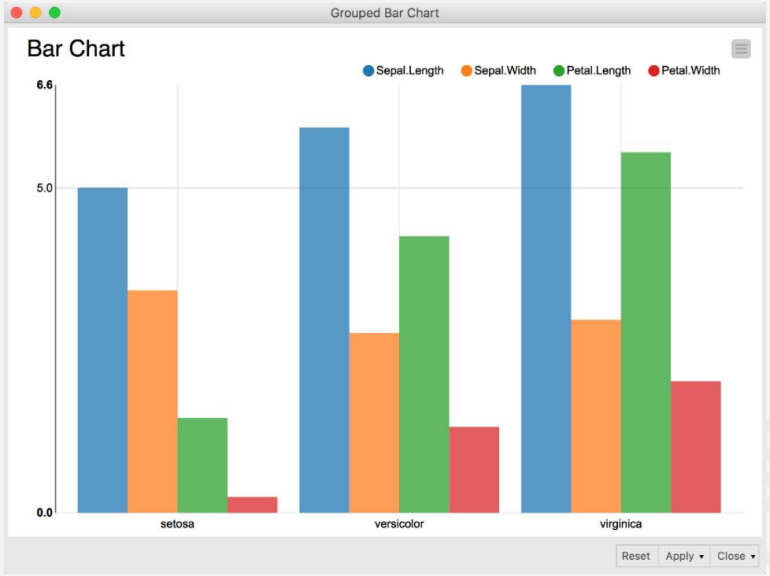


# New Node: Bar Chart

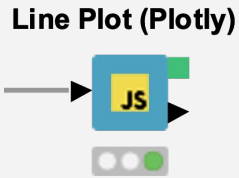
- Show numerical values across categories
- Vertical or horizontal bars
- Bars can be grouped or stacked

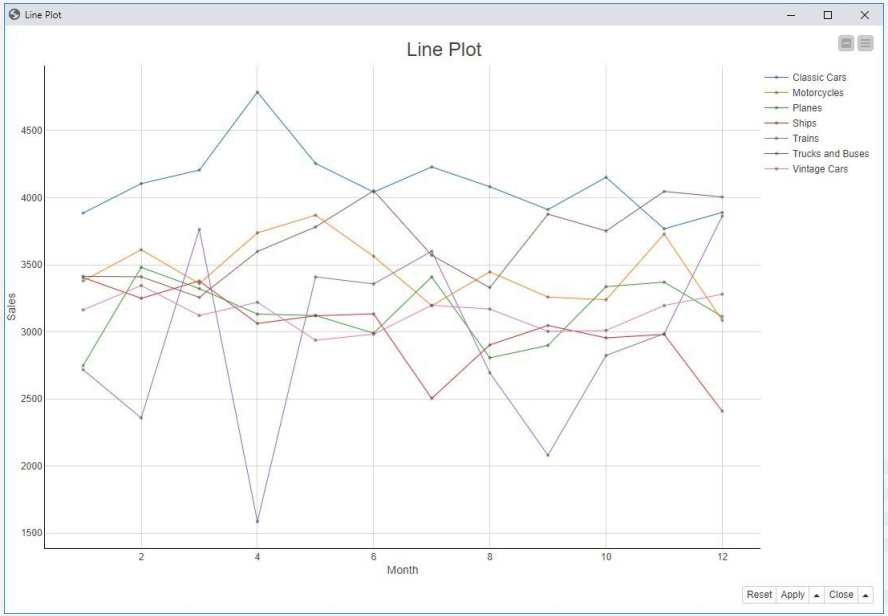


# New Node: Line Plot

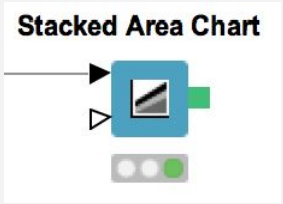
- Plot sequence of values, e.g. over time
- Useful to identify trends, also between groups

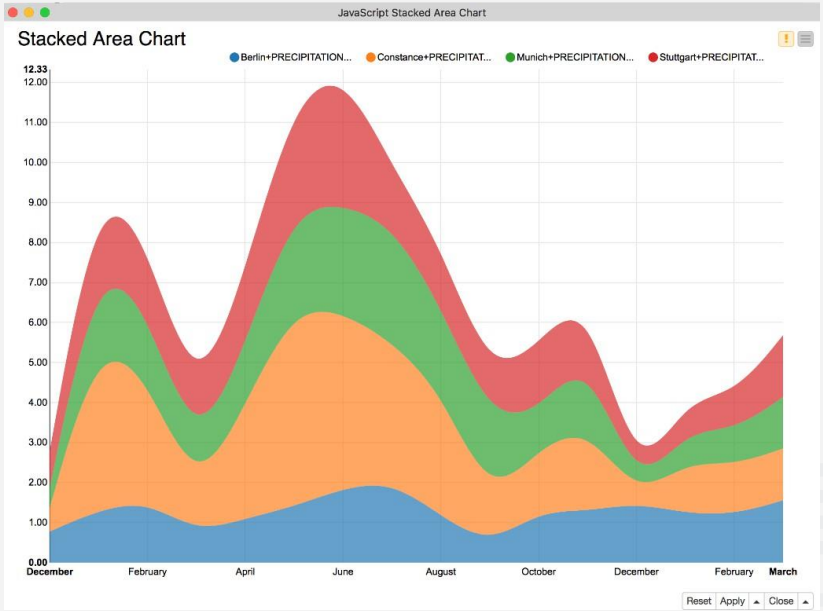


# New Node: Stacked Area Chart

- Visualizes numerical values from multiple columns as stacked areas
- Great for plotting distributions over time

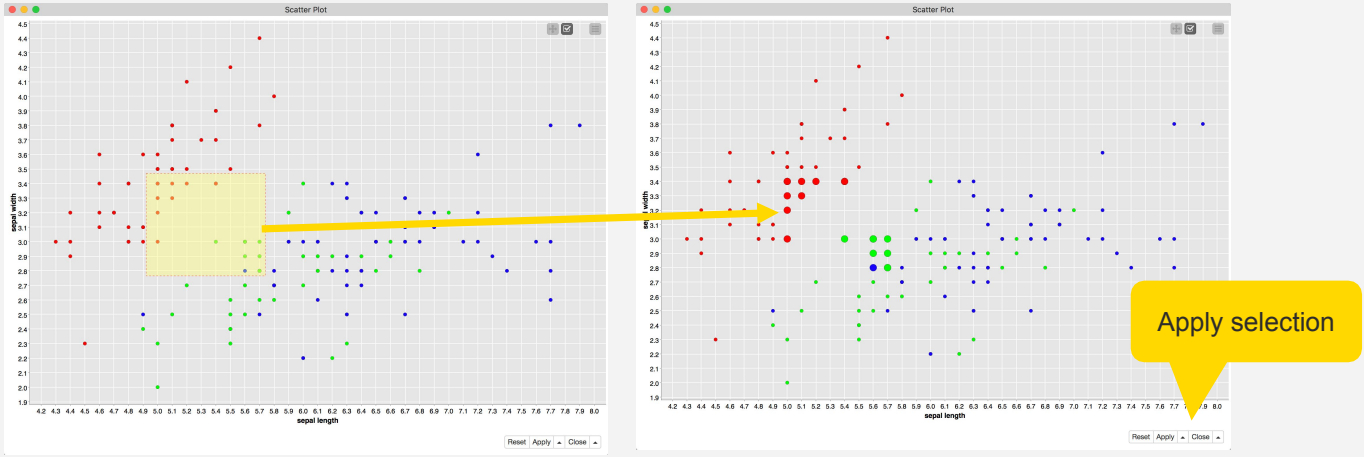


# Selection & Filtering in JavaScript Views

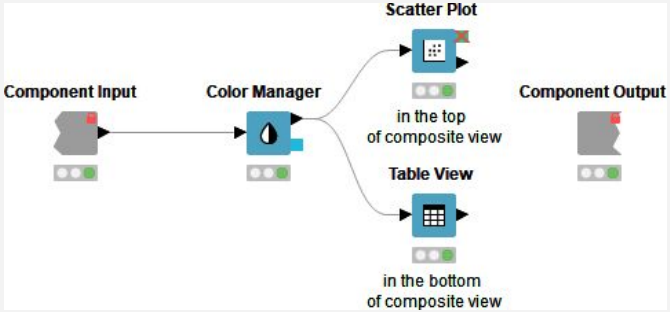
Interactivity allows you to select data points in views

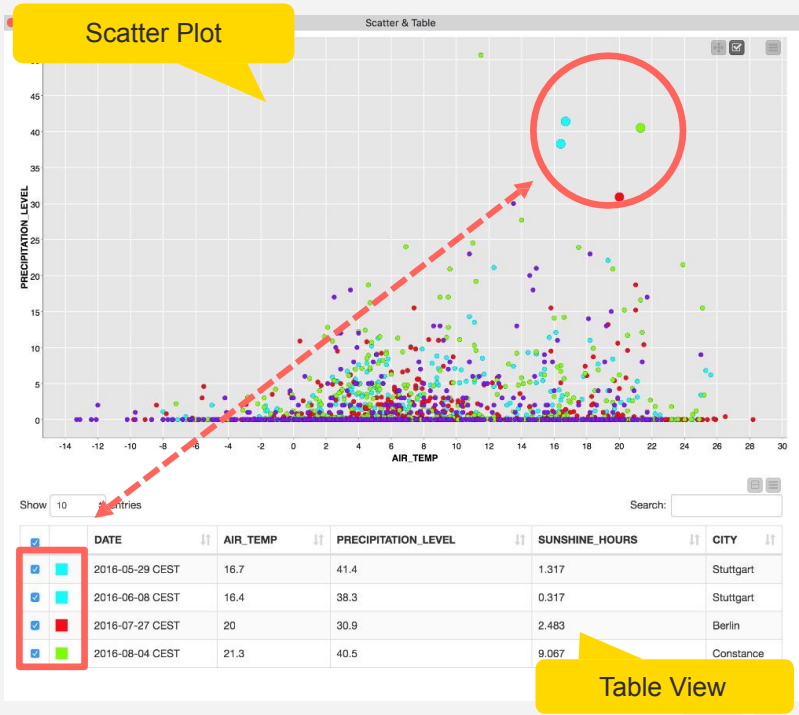
- Selection is propagated to other views
- Highlight selected rows or filter them
- Click “Apply” to add column to data that indicates selection (true/false) for use in downstream nodes



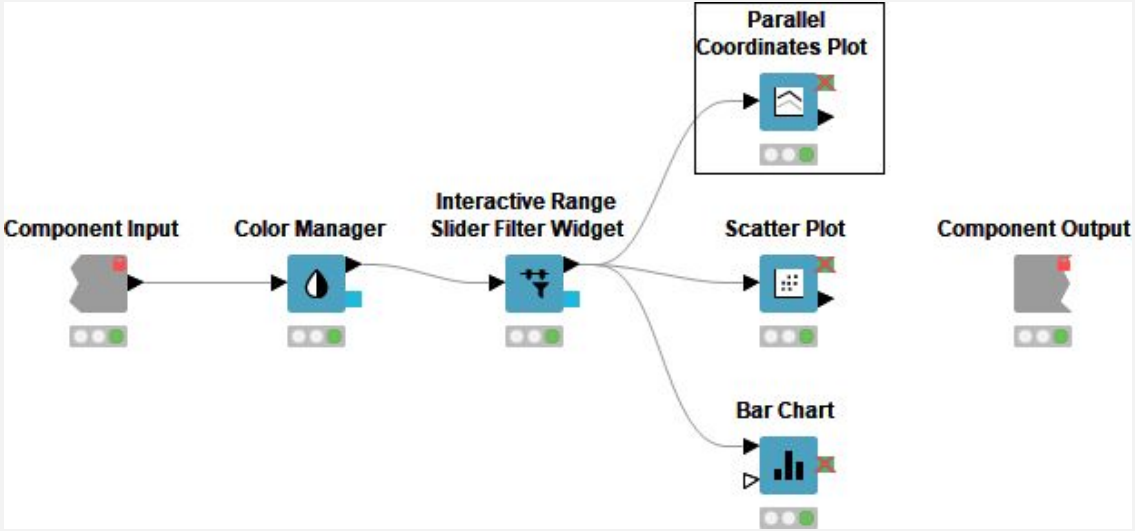
# Components – Combined Views

- Multiple JavaScript View nodes can be combined in Components
- Selections are transmitted to all other views
- Also for use on the KNIME

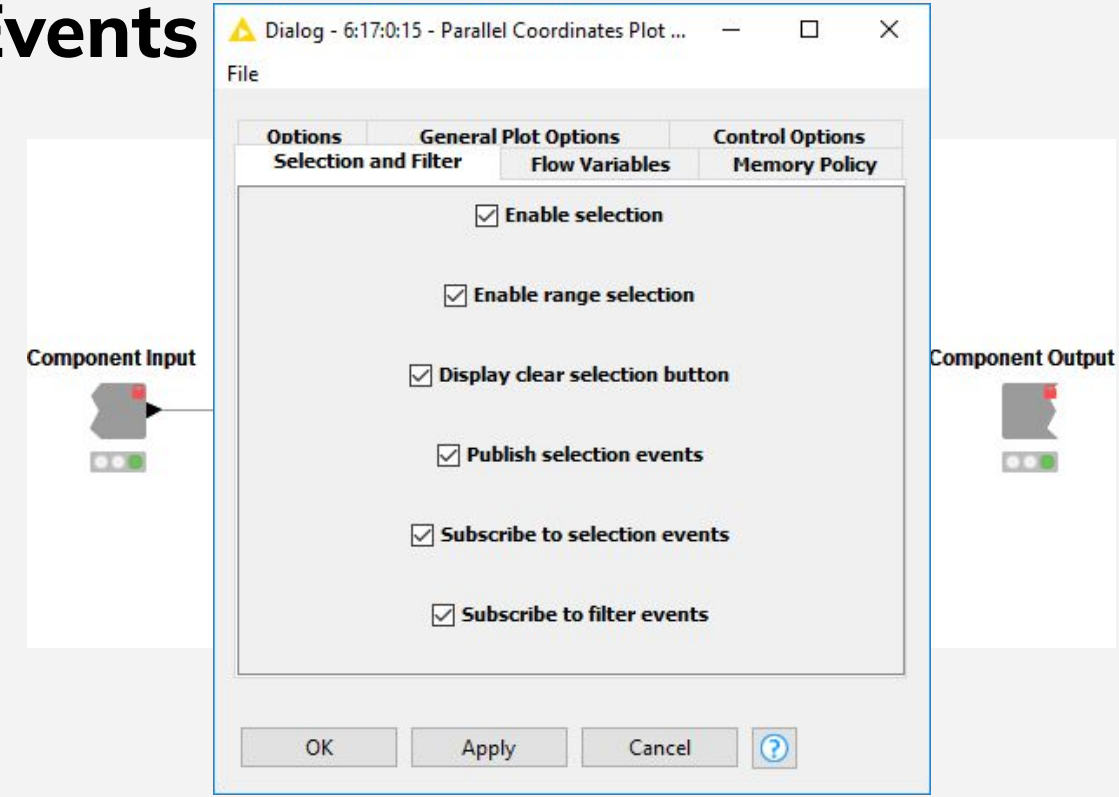


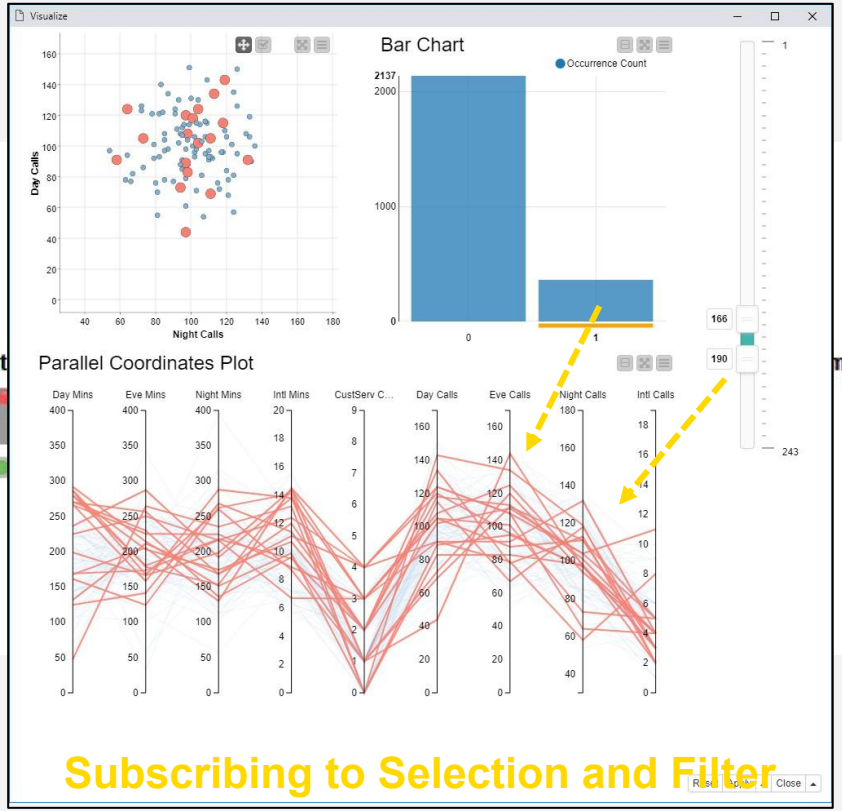
# Interactivity across Charts: Selection and Filter Events



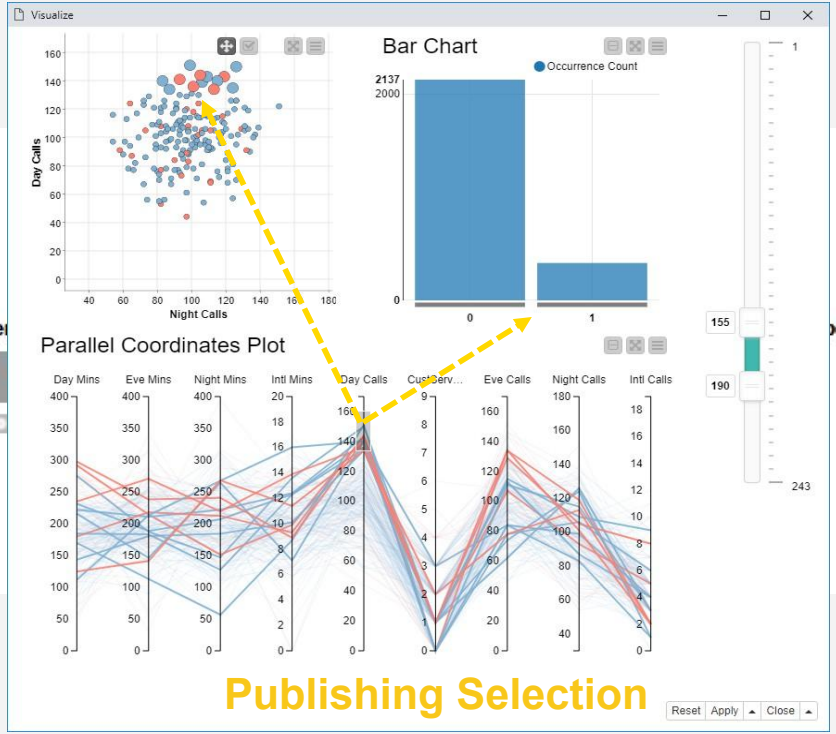
# Interactivity across Charts: Selection and Filter Events



# Interactivity across Charts: Selection and Filter Events

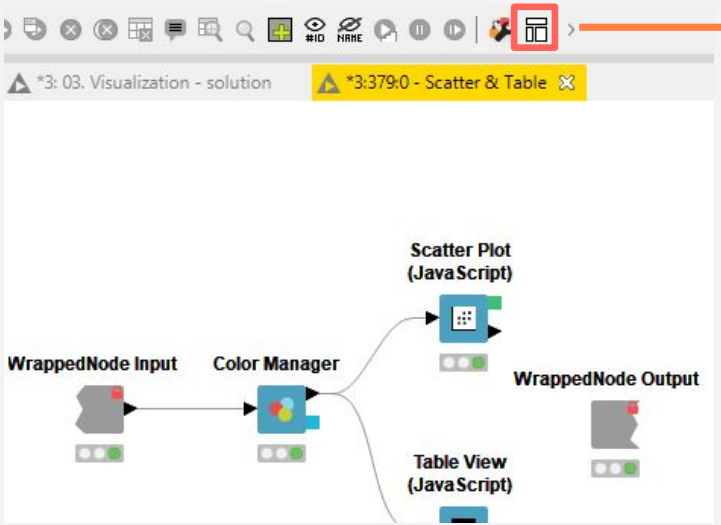


# Interactivity across Charts: Selection and Filter Events

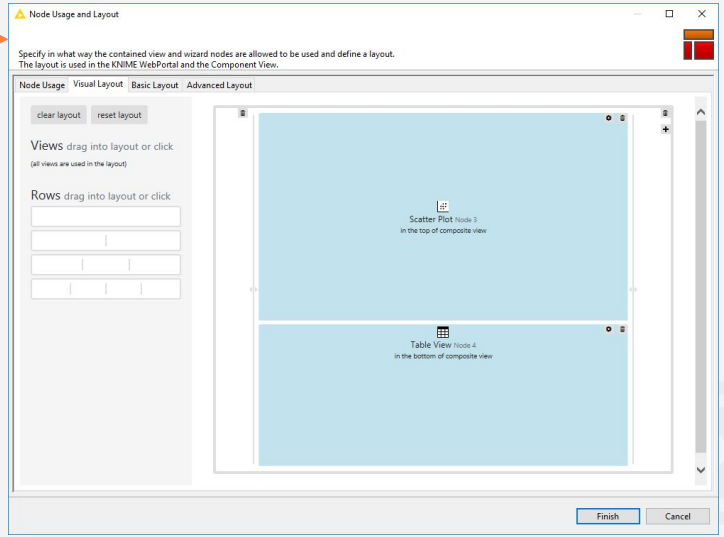


# Configure Content and Views Layout

- Click layout button when inside Component to assign views to rows and columns




- Add views and rows via *drag&drop*
- Add columns using **+** buttons



# Data Aggregation

| Product ID | Store | Category | # Ordered Items |
|---|---|---|---|
| P 1 | Online | Clothing | 2 |
| P 2 | Onsite | Home | 3 |
| P 3 | Onsite | Clothing | 1 |
| P 4 | Online | Clothing | 5 |
| P 5 | Online | Electronics | 7 |
| P 6 | Online | Electronics | 5 |



Aggregation: Count

| Category | Online | Onsite |
|---|---|---|
| Clothing | 2 | 1 |
| Home | 0 | 1 |
| Electronics | 2 | 0 |

Aggregation: Sum (# Ordered Items)

| Category | Online | Onsite |
|---|---|---|
| Clothing | 7 | 1 |
| Home | 0 | 3 |
| Electronics | 12 | 0 |

**Solution: Pivoting Node**

# Data Aggregation

| Product ID | Store | Category | # Ordered Items |
|---|---|---|---|
| P 1 | Online | Clothing | 2 |
| P 2 | Onsite | Home | 3 |
| P 3 | Onsite | Clothing | 1 |
| P 4 | Online | Clothing | 5 |
| P 5 | Online | Electronics | 7 |
| P 6 | Online | Electronics | 5 |



Aggregation: Sum (# Ordered Items)

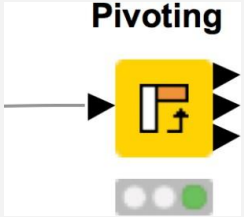
| Category | Online | Onsite |
|---|---|---|
| Clothing | 7 | 1 |
| Home | 0 | 3 |
| Electronics | 12 | 0 |

**Pivoting Node: Group - Pivot - Aggregate**

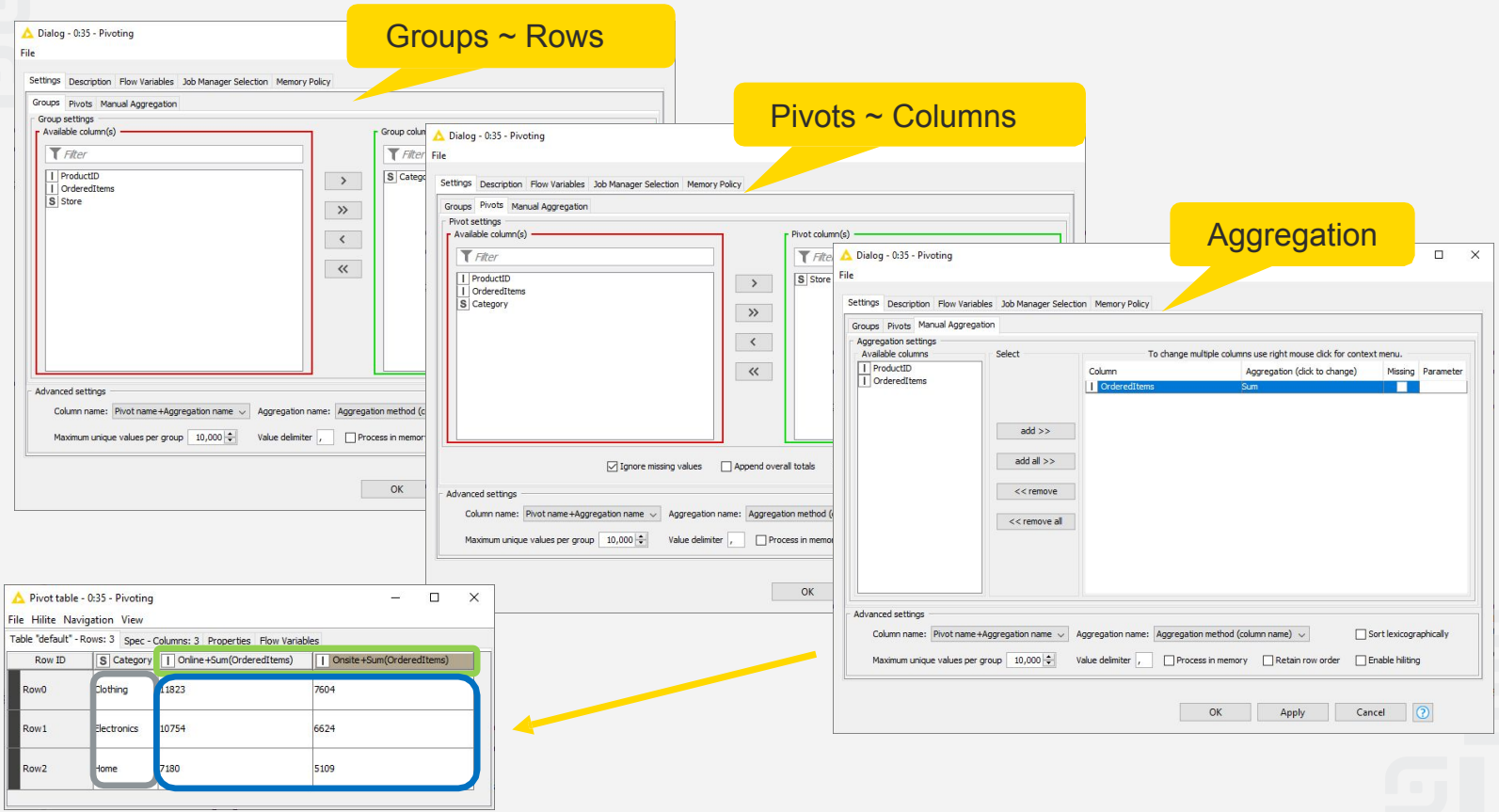
# New Node: Pivoting

Performs pivoting on selected columns for grouping and pivoting

- Values of group columns become unique rows
- Values of the pivot columns become unique columns for each set of column combination together with each aggregation
- Many aggregation methods are provided (similar to GroupBy)

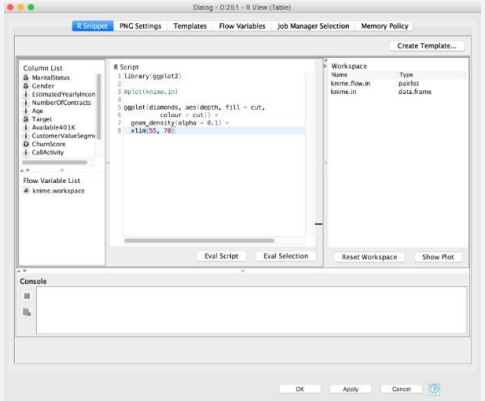


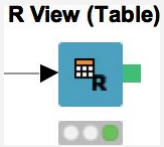
# New Node: Pivoting

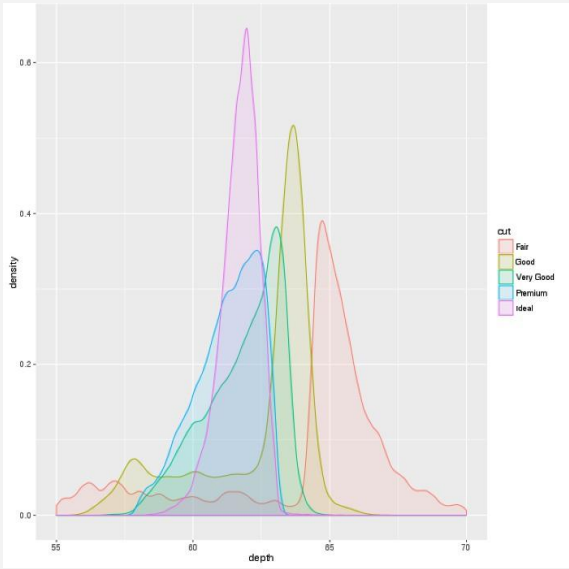


# Script-based View Nodes

- R View nodes for greater customizability
  - Use your favorite libraries, e.g. ggplot2
- If you prefer Python: Python View node
- For JS developers: Generic JavaScript View

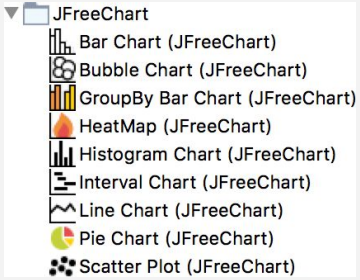


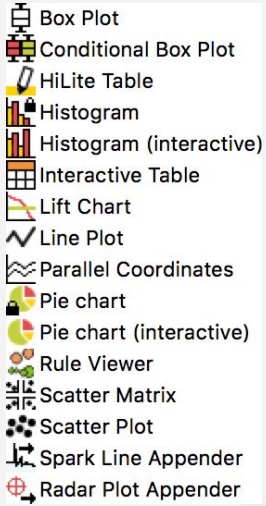



# Legacy View Nodes: JFreeChart & KNIME Views

- KNIME provides three types of visualizations
  - **JavaScript Views**
  - JFreeChart Views
  - Local Views
- Active development only for JavaScript Views -> use those!
- JFreeChart and Local Views still useful when visualizing locally




JFreeChart Views

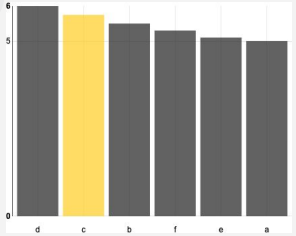



Local Views

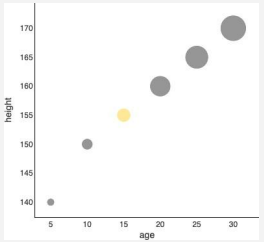
# Match correct visualization to a specific task

# Match correct visualization for a task

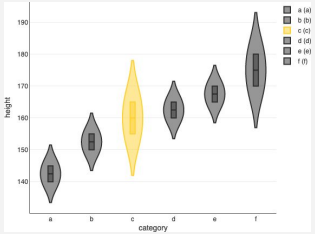
## Comparison



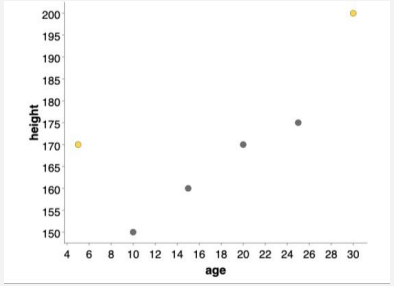

## Correlation



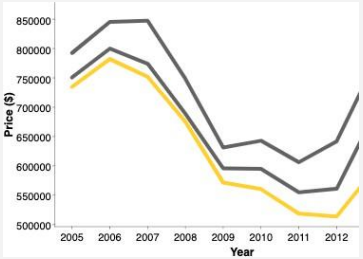

## Distribution




## Outliers

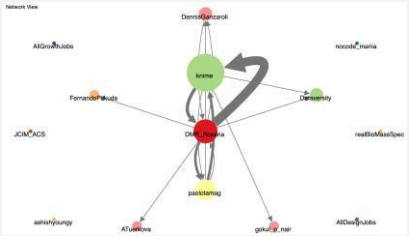



## Time

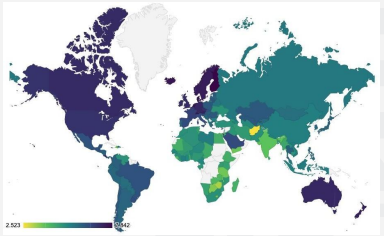



## Text


## Networks

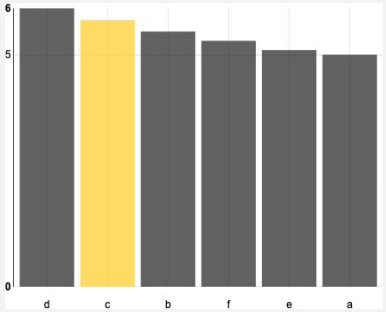



## Geography



# Comparison: Pie chart vs bar chart

- Bar chart
  - Differences between **2 or more categories**



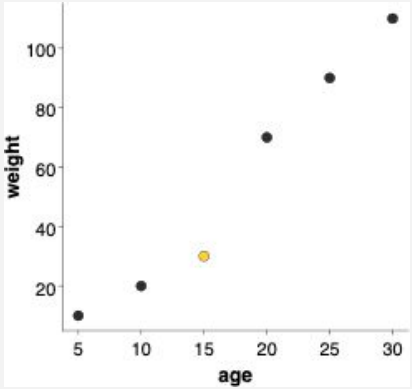

Example: compare growth

- Pie chart
  - differences between **2 categories**


Example: compare revenue

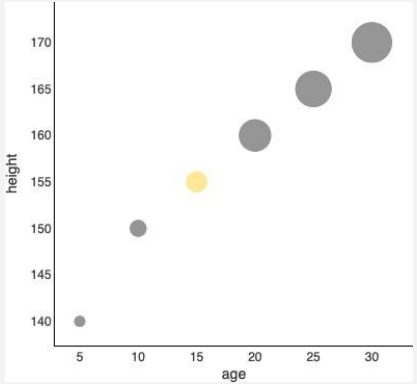
# Correlation: Scatter plot vs bubble chart

- Scatter plot
  - Correlation among **2 numeric columns**




Example:
comparing age vs weight

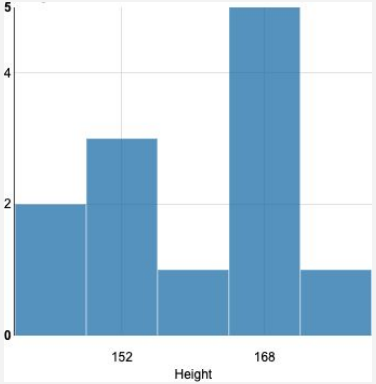
- Bubble chart
  - Correlation among **3 numeric columns**




Example:
comparing height vs age vs weight

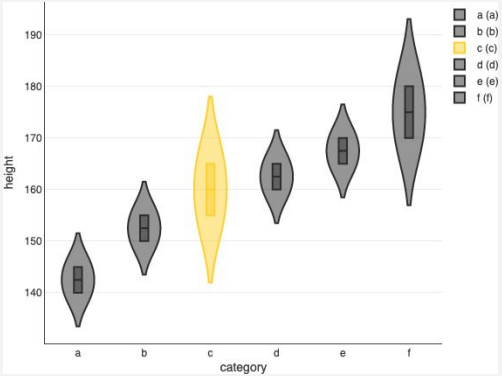
# Distribution: Histogram vs violin plot

- Histogram
  - The distribution of a **single numeric column**




Example: distribution of height

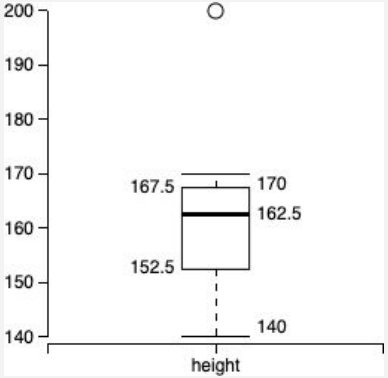
- Violin plot
  - The distribution of a **numeric column per category**




Example: distribution of height per category

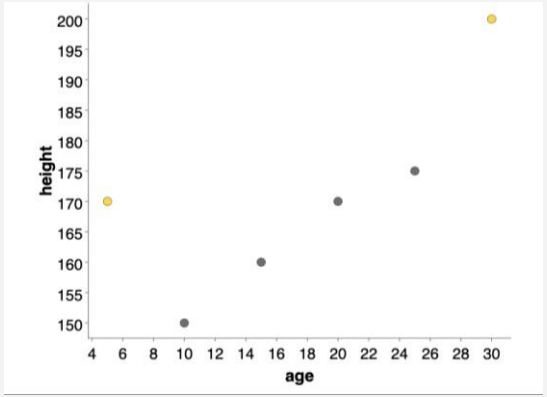
# Outliers: Box plot vs scatter plot

- Box plot
  - Spot outliers for **1 numeric column**




Example: spot unusually tall or short people

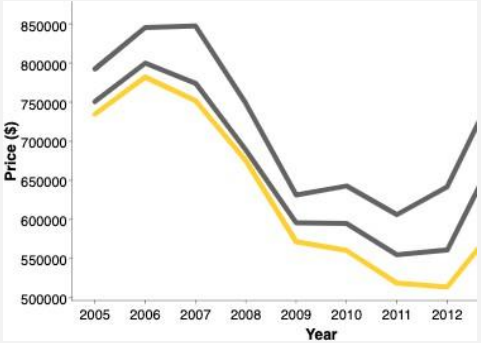
- Scatter plot
  - Spot outliers among **2 numeric columns**




Example: spot unusually tall or short people for their age

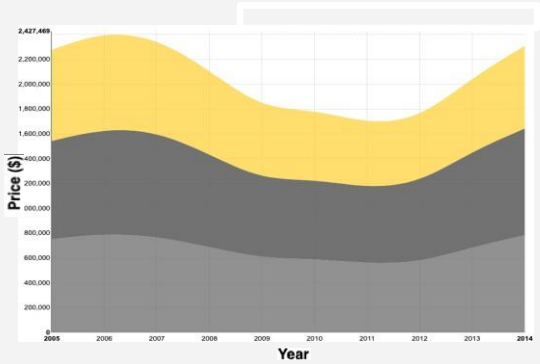
# Time: Line plot vs stacked area chart

- Line plot
  - 1 or more **numeric columns** with a **time dependency**




Example: prices of American houses in the early 2000s

- Stacked area chart
  - 2 or more **aggregated numeric columns** with a **time dependency**




Example: prices of American houses in the early 2000s

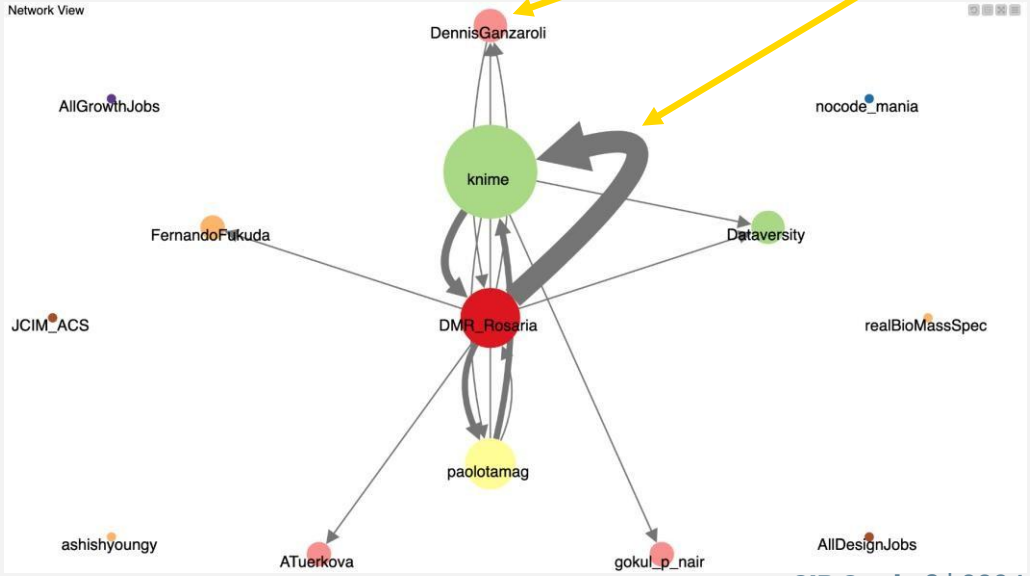
# Text: Tag cloud

- Tag cloud
  - Analyze word usage and frequency in data


- Key features:
  - Size of word corresponds to frequency
- Use cases:
  - Tweet analysis, product review analysis, customer topic analysis

# Networks: Network viewer

- Networks
  - Analyze relationship strength among entities

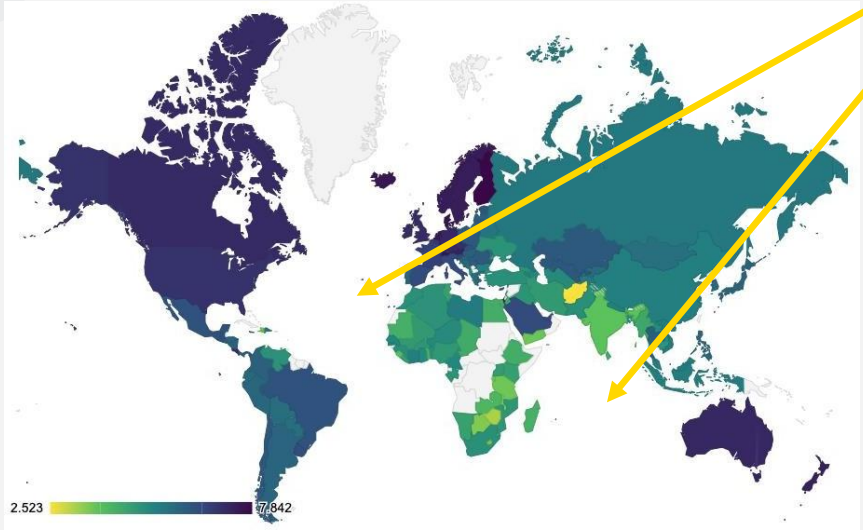



- Key features:
  - Bubble size represents one column
  - Arrow size indicates strength of relationship
  - Pointing arrows indicate a one-way relationship

- Use case:
  - Locating trend setters on Twitter, marketing groups for TikTok

# Geography: Choropleth map

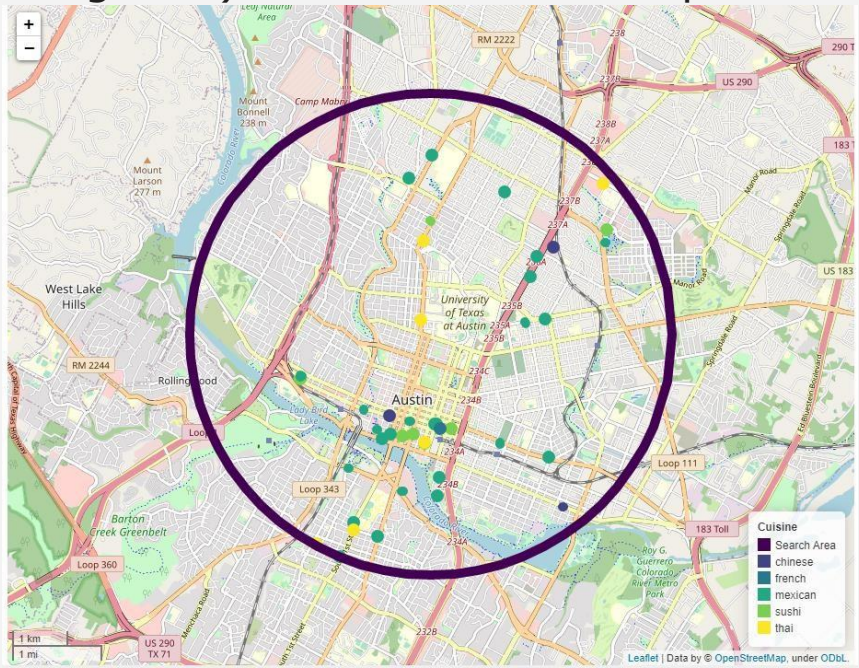
- Choropleth map
  - Analyze numeric metrics by regions




- Key features:
  - Gradient colors indicate the progression of the numeric values
- Use cases:
  - Regional sales report, customer satisfaction, hot-spot tracking

# Geography: Geospatial view

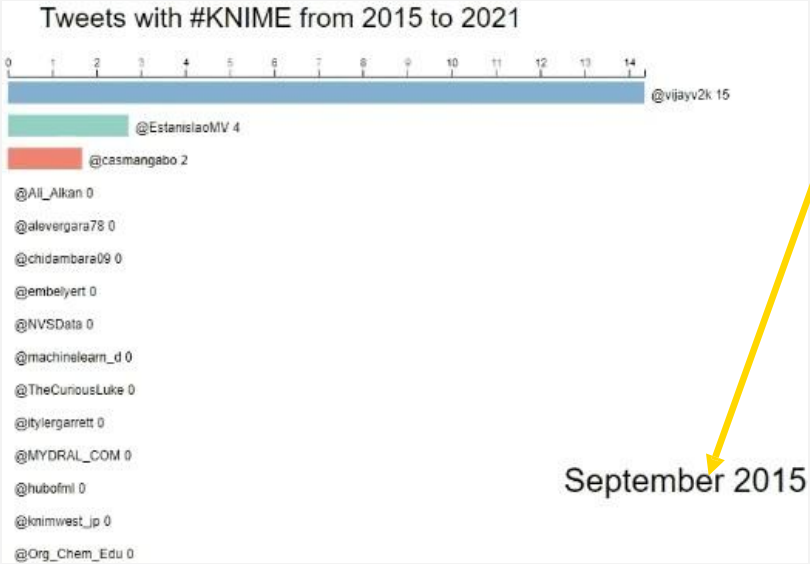
- Analyze data with location information (e.g., latitude and longitude) in an interactive map




- Key features
  - A part of the Geospatial Analytics Extension for KNIME by the Center for Geographic Analysis from Harvard and KNIME
- Use cases
  - Impact analysis
  - Points of interest search
  - Location optimization

Bonus: Animated bar chart

- Animated bar chart
  - Visualize different entities competing or changing over time

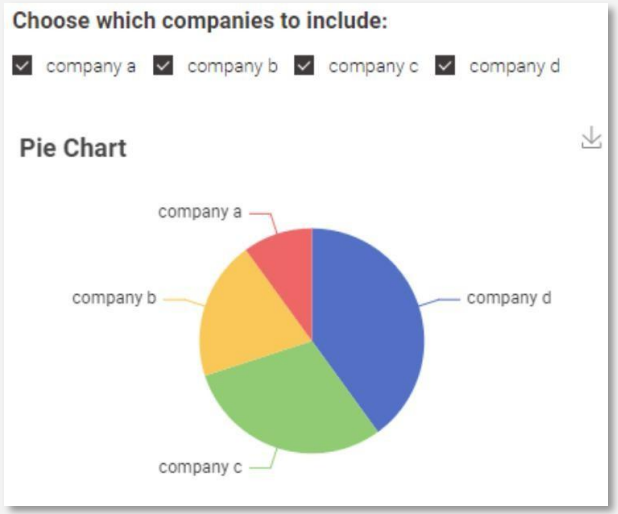



- Key features:
  - Time change
- Use case:
  - Attention-grabbing, trend development analysis

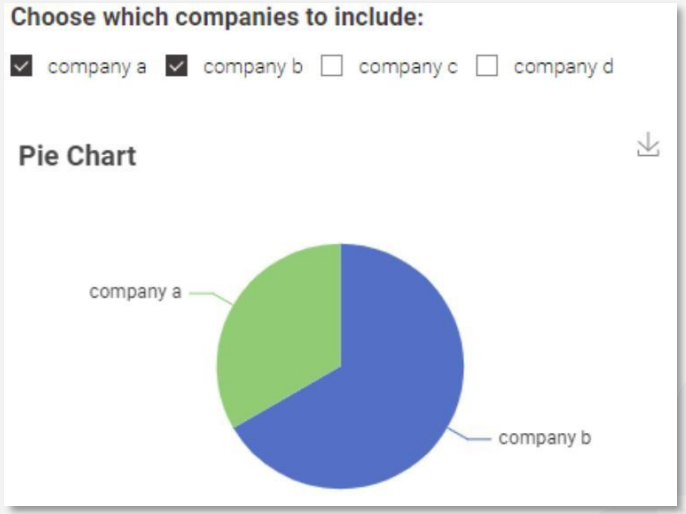
# Adding filters on plots

# Adding filters on plots

- How do you add filtering in the interactive view to update the visuals?

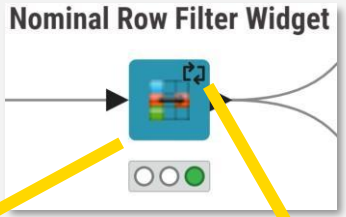


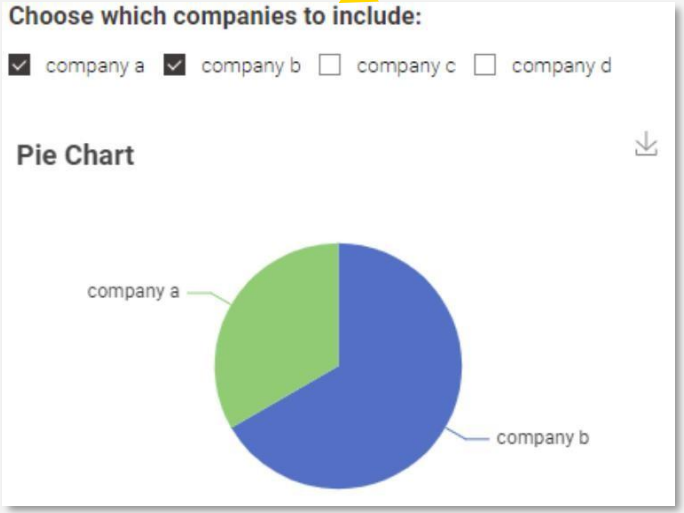

?

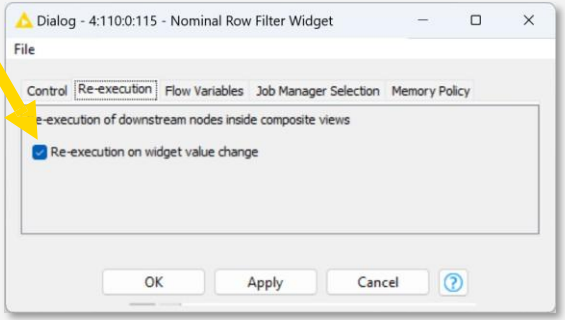


# Filtering plots

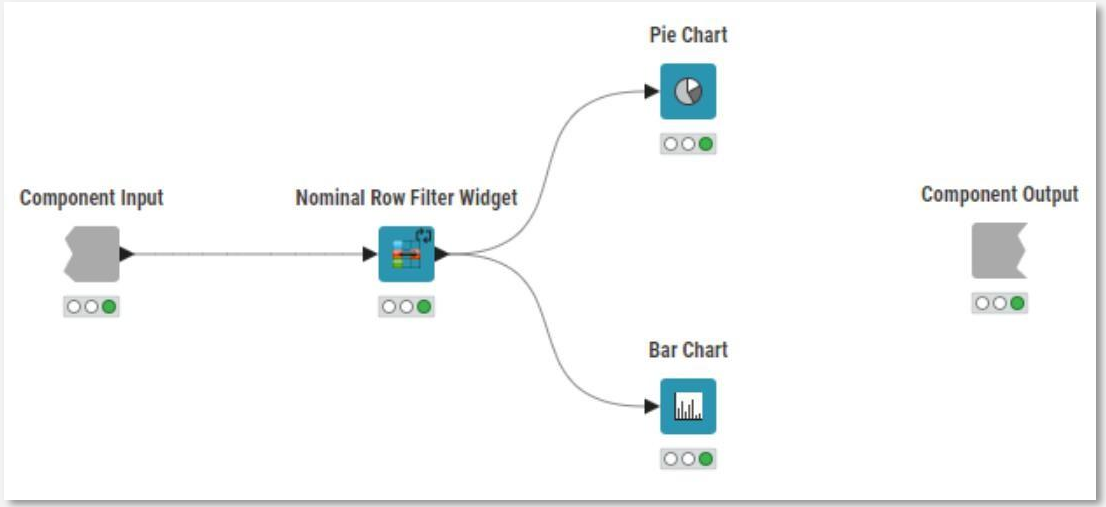
- Filter data with widgets
- Enable re-execution



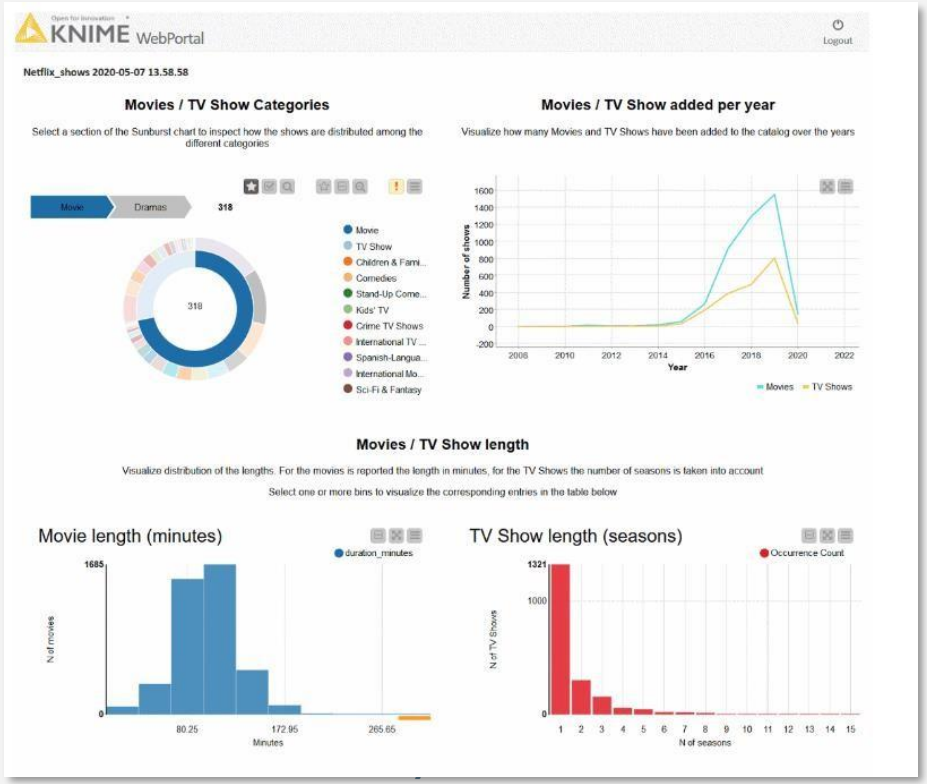



# 03: Adding interactive filtering demo



# More resources for interactive dashboards

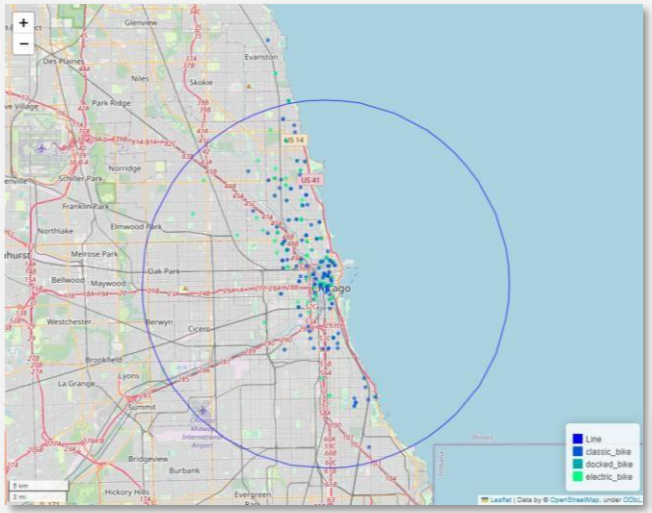
- https://www.knime.com/blog/how-to-create-an-interactive-dashboard-in-three-steps-with-knime



# Accessing and visualizing geospatial data

- Enter and access location information and visualize it in an interactive map


| lat<br>*Number (double)* | lon<br>*Number (double)* |
|---|---|
| 41.878 | -87.63 |

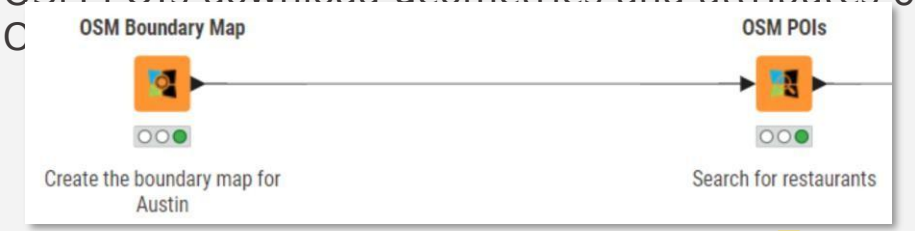


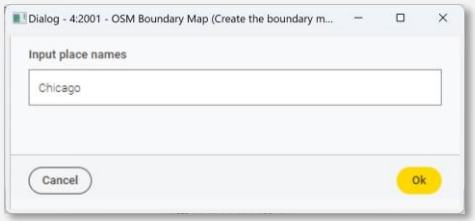
# Geospatial Analytics Extension for KNIME

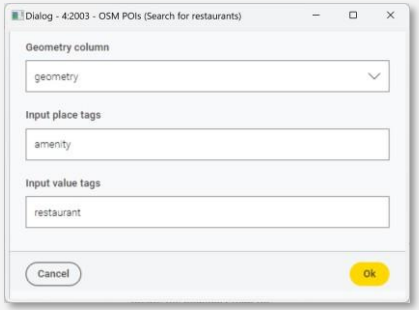
- KNIME Community Extension (Trusted) jointly developed by the Center for Geographic Analysis from Harvard and KNIME for
  - Reading and writing geospatial files (e.g., shapefiles, GeoJSON)
  - Performing spatial calculations (e.g., computing distances, joining)
  - Viewing data on an interactive map
- Access a collection of example workflows on the KNIME Community Hub
- View the KNIME Geospatial Analytics playlist on YouTube

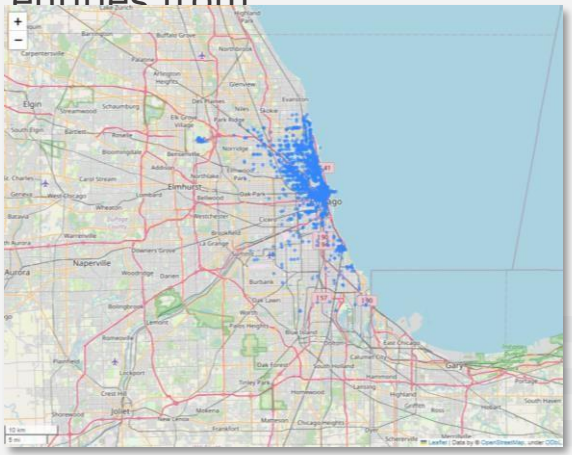
# Open Datasets nodes to get basic geospatial shapes

- The OSM node uses the OpenStreetMap, an open geographical database, to obtain
  boundary information.
  - The OSM Boundary Map takes a place name (e.g., country, city, or village) as input.
  - OSM POIs download geometries and attributes of geospatial entities from
    C

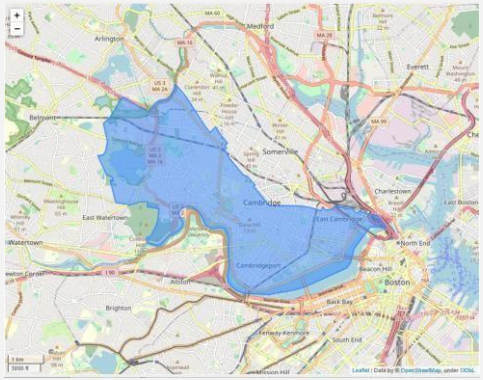


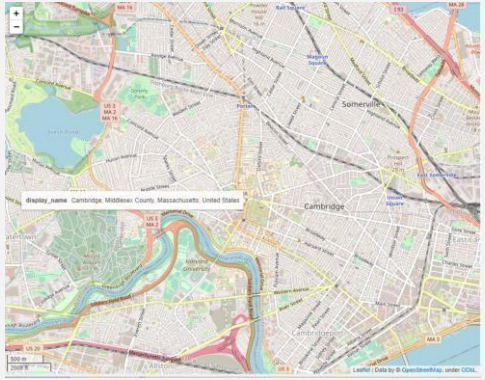



- Geospatial analytics relies on three basic shapes:
  - **(Multi)polygons** enclose a city or a country within its geographical boundaries.
  - **Points** indicate just the location without any information on boundaries.
  - **Linestrings** are lines connecting the coordinates of two points.





- Points, linestrings and polygons are stored in a new data type: the **Geometry** type.

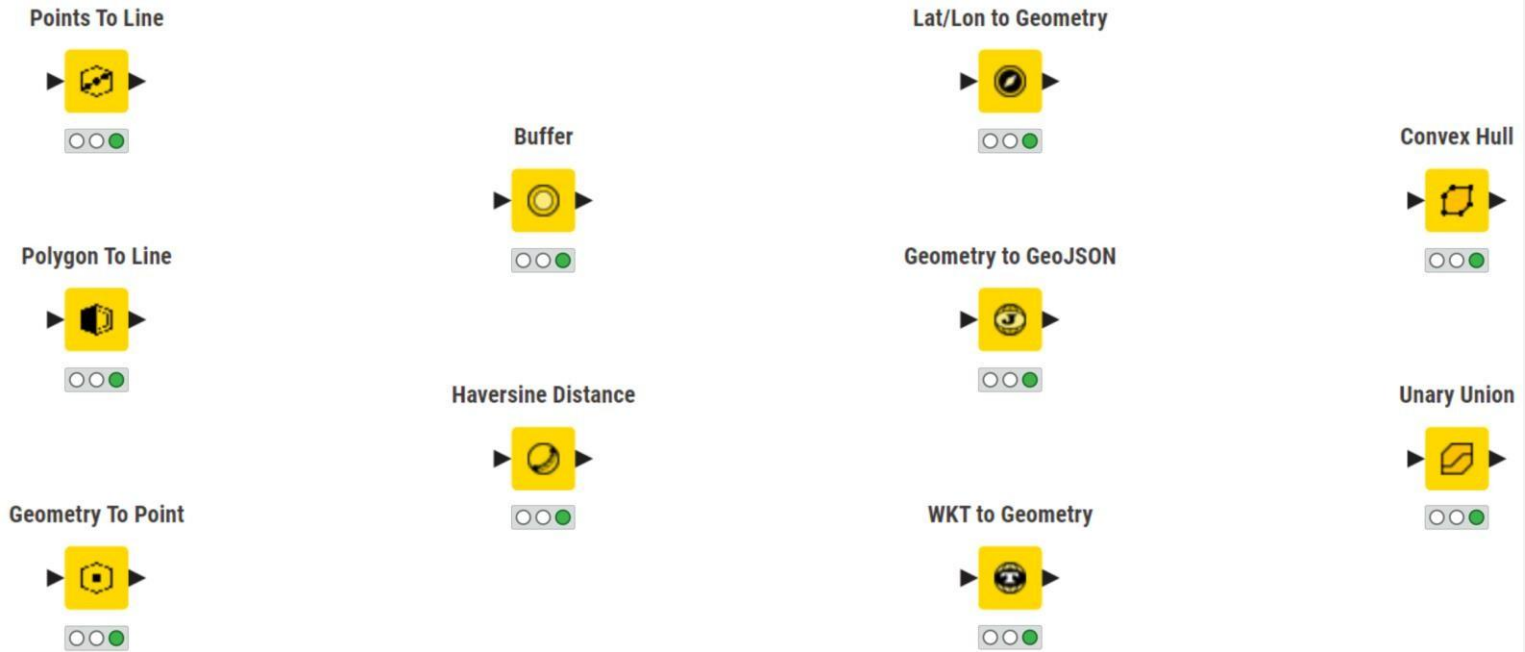
# Geospatial Nodes

TRANSFORMATION

MANIPULATION

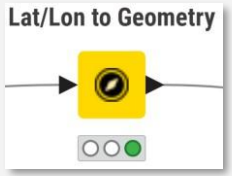
CONVERSION

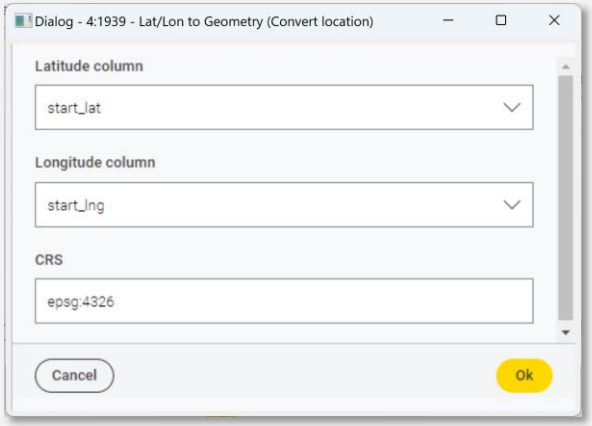
CALCULATION



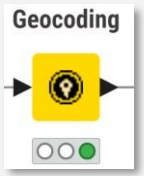
# Spatial Conversion nodes

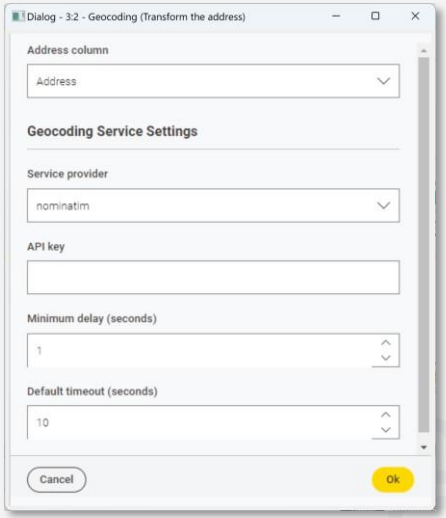
- Use the **Lat/Lon to Geometry** node to obtain geometry from latitude and longitude



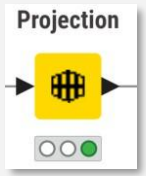

- **Geocoding** node produces a geometry from a given location

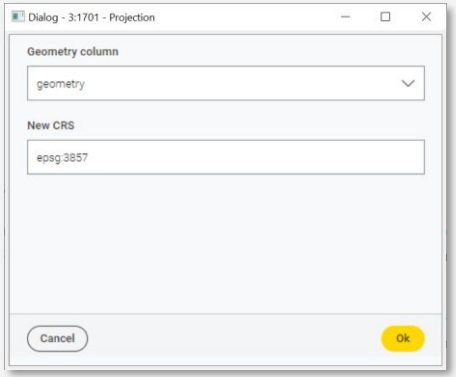


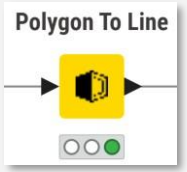
# Spatial Transformation nodes

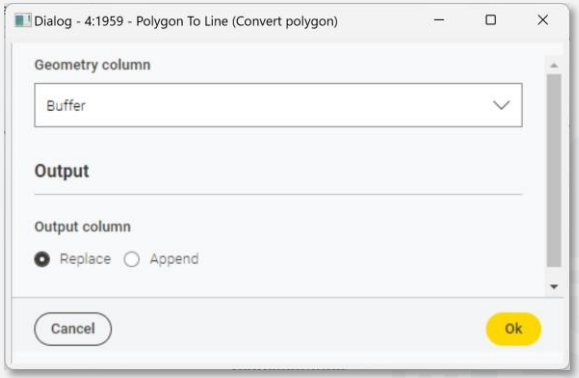
- The **Projection** node
  - Coordinates in the Geometry type column are expressed by default in degrees.
  - Transforms the Coordinate Reference System of a Geometry into a new system for mapping coordinates (and the related units of measurement, i.e., degrees or meters) using predefined EPSG codes.




- The **Polygon To Line** node retrieves the polygon boundaries, yielding lower-dimensional objects that depict the set-theoretic boundaries of each geometry.

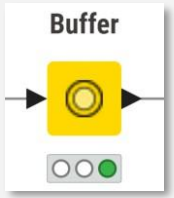


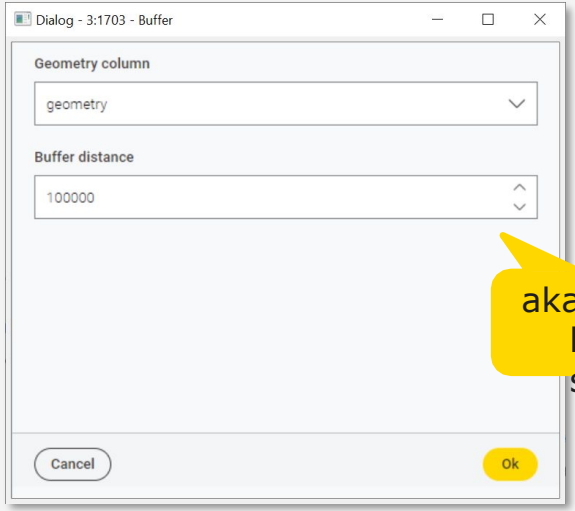



# Spatial Manipulation nodes

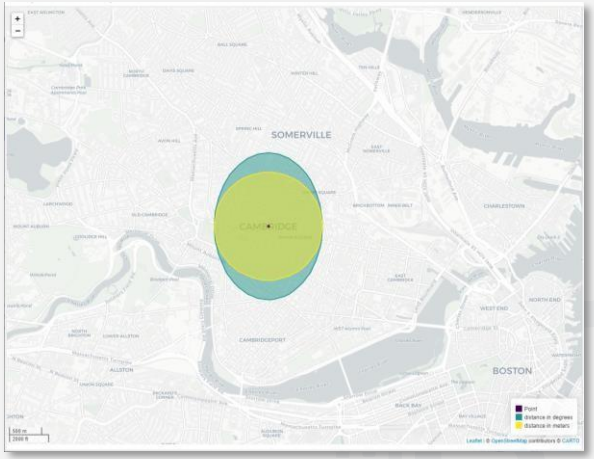
- The visualization of a point is usually very tiny and hard to distinguish especially if compared to the size of the country.

- The **Buffer** node
  - Transforms the point into a polygon containing the padding space.



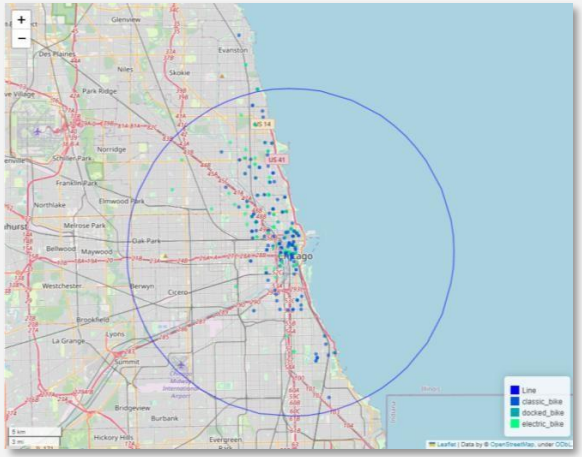




aka buffer size

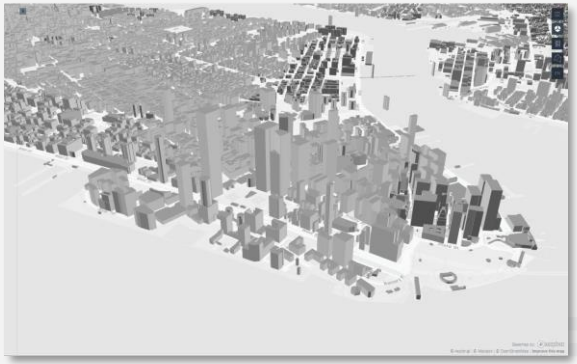


# Spatial Visualization nodes

- The **Geospatial View** node
  - Visualizes a polygon or a point object of Geometry type on a world map.
  - Allows to configure marker tooltip, size and color, marker classification method, base map, and legend.

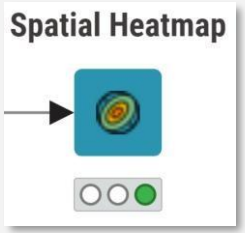



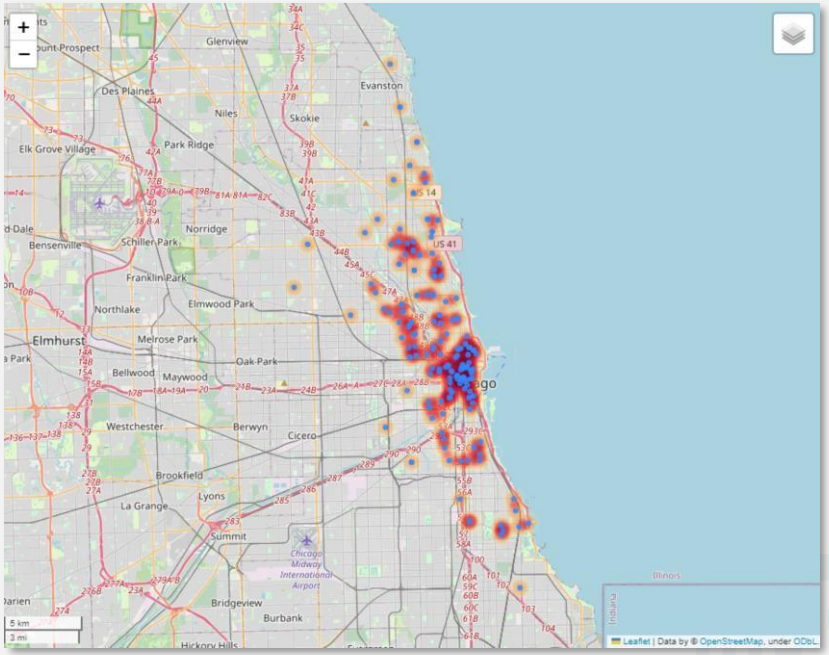
- The **Kepler.gl Geoview** node

# Spatial Heatmap node

- This node displays the provided data as an interactive heatmap. A chosen weight column is represented as color intensity of each data point.



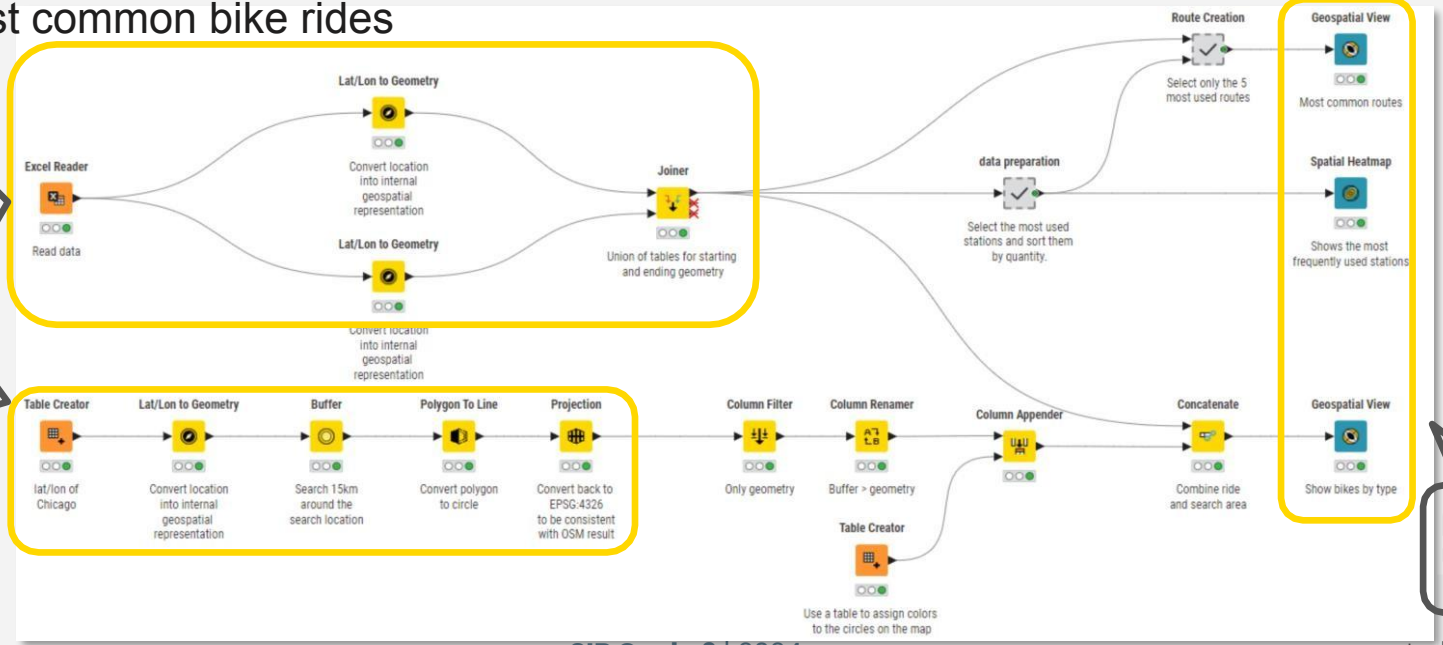



# Demo: Finding the most frequently used areas and routes

- Example of a workflow for pre-processing geospatial data with the goal of identifying bikes, specifically focusing on whether they are used outside a central area within a 15km radius, tracking multiple locations where they are taken, and ultimately analyzing the most common bike rides

1. Enter and transform your location

2. Create a circle around a point




4. Show the results

Now you should be able to:

1. Match correct visualization for a task.

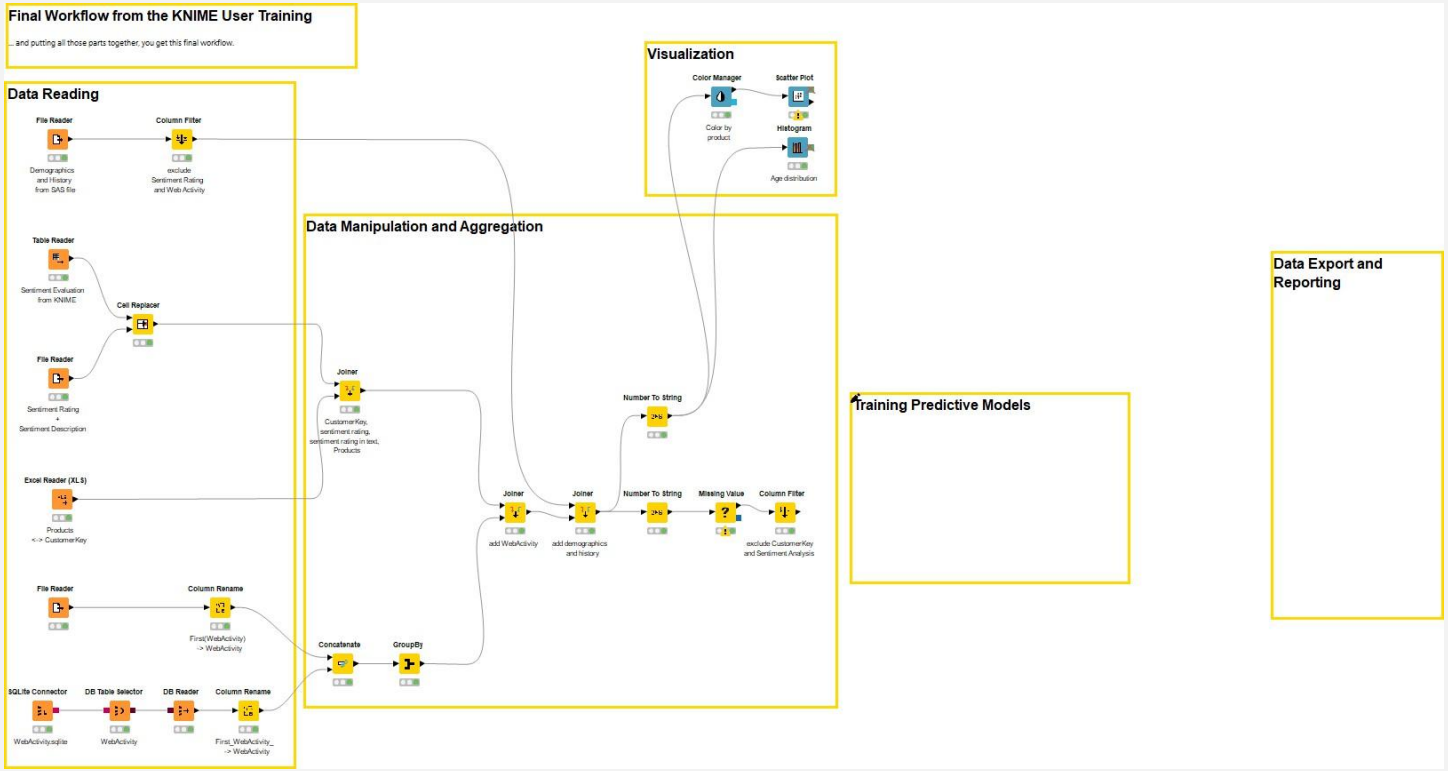
2. Apply visualizations to common tasks.

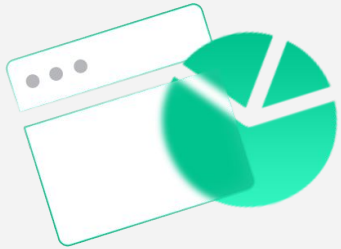
# Visualization Exercise

Start with exercise: *Visualization*

- Read *sales.csv* data
- Assign a different color to each product
- Plot BasketValue against BasketSize using the Scatter Plot node
- Show the total BasketValue by time and product in a Line Plot and a Stacked Area Chart (Use the Pivoting node to get the sum of sales by Quarter and Product!)
- Execute the *Fully Joined Data* metanode
- Show the number of customers in the different web activity categories in a Bar Chart
- Show the age distribution of the customers in a Histogram
- Create a composite view by combining the Bar Chart and Histogram
- Select one web activity class in the Bar Chart. Which age classes are represented in the selected web activity class?

# Today's Example





# Terima Kasih