Laurentian University
Université Laurentienne

OPER-5151EL-01: Business Analytics

LAB 1: INTRODUCTION TO

tableau

Activity # 1: Navigating Tableau

1. **Exploring Tableau Desktop and Tableau Prep**
   - 1.1 Launch Tableau Desktop
   - 1.2 Launch Tableau Prep
2. **Connecting to data sources**
   - 2.1 Connect to the file Global "Superstore Orders 2016.xlsx"
   - 2.2 Types of connection
     - 2.2.1 Live
     - 2.2.2 Extract
   - 2.3 Bring the sheet Orders to the view data pane
3. **Data types**
   - 3.1 Dimensions
   - 3.2 Measures
4. **Main components**
   - 4.1 Analysis tool
   - 4.2 Analytics tool
   - 4.3 "Show Me" pane
   - 4.4 Sheets
   - 4.5 Dashboards
   - 4.6 Stories
5. **Basic graphs in Tableau**
   - 5.1. Click on the Show me Menu (on the right upper corner)
6. **Creating connection**
   - 6.1 Loading new data sources
     - 6.1.1 Click on Data/New Data Source
     - 6.1.2 Load the file
   - 6.2 Adding new data sources
     - 6.2.1 Click on Add
     - 6.2.2 Load the file "Global Superstore Returns 2016.csv"

Question 1. What is the difference between loading and adding new databases?

   - 6.3 Join the two datasets using a join
   - 6.4 Join the two datasets using a relationship

Question 2: What is a join, union, and relationship in database language?

Activity # 2: Exploring the Datasets

1. Open the two datasets
2. Explore the data

Question 3. What are the variables? What are their types? Their scales of measurement?

Question 4. What types of visuals can we build with them?

Question 5. How many observations are there in the Orders dataset?

Activity # 3: Creating your first visuals with Tableau

1. Create a new worksheet (at the very bottom of your screen)
2. Under Tables, click on Category
3. Drag it to the Rows pane
4. At the lower-left corner, click on Orders (Count), drag it to the table on the main canvas
5. Click on the Show Me menu, click on the pie chart icon
6. Click on Category and drag it to Label (under marks)
    a. If the labels are not visible, press Ctrl + Shift + B (press ⇧ + ⌘ + B on a Mac)
7. Click on the Orders (Count) field and drag it to Label (under marks)
8. Change the frequencies to relative frequencies (percentages)
    a. Click on the analysis menu (top ribbon)
    b. Click on Percentage of/Table
9. Change the name of your worksheet

Activity # 4: Explore the other types of graphs with other variables

LaurentianUniversity
UniversitéLaurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

LAB 2: DATA VISUALIZATION WITH

+ a b | e a u

Objective of this lab: At the end of the lab, students will be able to

- Use Tableau to implement the following visualization techniques:
  - Pie chart
  - Bar chart
  - Stacked bar chart
  - Side-by-side bar chart
  - Histogram
  - Box plot
  - Line graph
  - Area graph

- Efficiently exploit Table Mark Cards
  - Color
  - Size
  - Label
  - Detail
  - Tooltip

1. Connect to the file Global "Superstore Orders 2016.xlsx"

Activity # 1: Creating a pie chart

2. Create a pie chart to illustrate the contribution (in percentage) of each product to the total sales.
   2.2. Color the chart by product category
   2.3. Add the labels of the product categories
   2.4. Show the relative frequencies on the chart
   2.5. Increase the size of the chart as needed
   2.6. Rename your sheet "Product category"

Activity # 2: Creating bar charts

3. Now, let's analyze the sales distribution per Market and Product category
   3.1. Add a new sheet
   3.2. Drag Market the **Rows shelf**
   3.3. Drag Sales to the **Text Mark Card**
   3.4. Click on the **Show Me menu** to change the view to a bar chart
   3.5. Use the **Swap function** to change the chart to a **vertical bar chart**
   3.6. Use the **Sort function** to sort the bars from highest to lowest
   3.7. Drag Category onto the **Color Mark Card** to transform the vertical bar chart to a **vertical stacked bar chart**
   3.8. Drag Category onto the **Size Mark Card** to illustrate the contribution of each product category
   3.9. Rename your sheet Sales-market-category

Activity # 3: Creating a side-by-side chart

4. Let's compare the different markets based upon the sales
   4.1. Repeat the first three steps of the previous exercise
   4.2. Drag Segment to the **Columns Shelf**
   4.3. Change the view to a **side-by-side chart** (use Show Me)
   4.4. Move Segment to the right of Market
   4.5. Color the chart bar segment
   4.6. Rename your sheet "sales-market-segment"

Activity # 4: Creating a histogram

5. Let's analyze the frequency distribution of the sales through a histogram
   5.1. Remember that as sales are ratio data, we need to transform them into ordinal data
   5.2. Right-click on Sales
      5.2.1. Click on **Create**
         5.2.1.1. Click on **Bins** (keep the default values)
      5.2.2. Drag Sales (bin) (under Tables) to the Columns shelf
      5.2.3. Display the frequencies (now, you know how to do it)
      5.2.4. Drag Category to the color Mark Card to illustrate the contribution of each product category for each class
   5.3. Let's format the Sales axis
         5.3.1.1. Right-click on the y-axis
         5.3.1.2. Click on format
         5.3.1.3. Under the Scale menu
            5.3.1.3.1. Click on the Numbers menu
            5.3.1.3.2. Choose Currency (Custom)
               5.3.1.3.2.1. Unit Display
                  5.3.1.3.2.1.1. Millions (M)
   5.4. Let's increase the size of the bins (intervals)
         5.4.1.1. Click on the Sales (bin) field (variable)
            5.4.1.1.1. Click on Edit
               5.4.1.1.1.1. Change the bin size to 1,000
   5.5. Rename your sheet "sales-histogram"

Activity # 5: Creating a box plot

6. Now let's take a look at the variation of the discounts across the different product categories
   6.1. Drag the Discount field to the Rows shelf and Category to the Columns
   6.2. Under Show Me, click on the box plot icon
   6.3. Click on the Analysis menu
      6.3.1. Deactivate the **Aggregate function**
   6.4. **Let's change the whisker upper limit to the maximum values**
      6.4.1. Right-click the y-axis
         6.4.1.1. Click on **Edit Reference Line**
            6.4.1.1.1. Under Whiskers extended to

6.4.1.1.1.1. Maximum extent of the data
6.4.1.1.1.2. Change the Formatting style to your liking

6.5. Rename the sheet “box plot – category”

Activity # 6: Creating Line and Area charts

7. Let’s look at the distribution of the annual sales across the quarters
   7.1. Drag the Sales field to the Rows shelf
   7.2. Drag the Order Date field to the Columns Shelf
   7.3. Drag the Order Date field to the Columns Shelf (to the right)
   7.4. (See what happened)
   7.5. Repeat the exercise now to analyze the evolution of the annual sales over the months
   7.6. Add the Category field to the Color Mark Card
   7.7. Rename your sheet to your liking
   7.8. Create another sheet
   7.9. Repeat the same analysis now using an area chart

Laurentian University
Université Laurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

LAB 3: DATA VISUALIZATION WITH tableau II

Objective of this lab: At the end of this lab, students will be able to

- Use Tableau to create:
  - Bubble plots
  - Tree maps
  - Heat maps
  - Geographic maps
- Efficiently exploit the following Tableau Marks Cards
  - Color
  - Size
  - Tooltip

1. Connect to the file Global "Superstore Orders 2016.xlsx"

Activity # 1: Creating a Bubble Plot

2. Let's explore the relationship between Profit, Quantity, and Sales across the different markets

2.1. Drag Profit to Rows and Quantity to Columns
2.2. Drag Market to the Color Marks Card
2.3. Choose Circle as the marker type
2.4. Drag Sales to Size to define the size of the bubbles
2.5. Increase the size of the bubbles
2.6. Drag Market and Sales to the Label mark
2.7. Format the Profit axis to $K and the Sales to $M
2.8. Remove the grid
  2.8.1. Click on the **Format** menu
  2.8.2. Click on the **Line** icon
  2.8.3. Choose "None" for **Grid Lines**
2.9. Rename your worksheet

Activity # 2: Creating a Tree Map

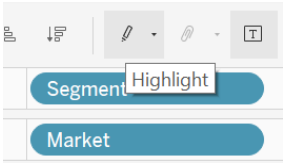
3. Let's compare the Sales of the different product subcategories across the product categories

3.1. Drag Sales to Rows and Subcategory to Columns
3.2. Choose a Tree map as the visual tool (under Show Me)
3.3. Color the Viz by the product category
3.4. Add the labels and the Sales values to the Viz
3.5. Format the Sales values to $M
3.6. Rename your worksheet

Activity # 3: Creating a Heat Map

4. Let's explore the Profit across the different markets and segment and illustrate the magnitude of the profit using a heat map

4.1. Bring Market to Rows and Segment to Columns
4.2. Drag Profit to the Text Marks Card
4.3. Under Show Me, choose the **highlight tables** viz
4.4. Add the Grand Totals
    4.4.1. Click on Analysis
        4.4.1.1. Total
            4.4.1.1.1. Show Row Grand Total
            4.4.1.1.2. Show Column Grand Total
    4.4.2. Let's change the color scale
            4.4.2.1.1. Click on the Color Marks Card
                4.4.2.1.1.1. Edit Colors
                4.4.2.1.1.2. Under Palette choose "Green blue diverging"
        4.4.2.2. Let's add a **highlighter** for the Market
            4.4.2.2.1. Click on the Highlight tool



            4.4.2.2.2. Click on Market
            4.4.2.2.3. The Highlighter is on the upper right corner of your screen
            4.4.2.2.4. Rename your worksheet

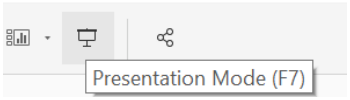
Activity # 4: Creating a Geographic Plot

5. Now, we want to explore the Sales across different geographic characteristics
5.1. Create a new worksheet
5.2. First we need to make sure we have geographic variables
5.3. For instance, click on Country
    5.3.1. Geographic Role
5.4. Double click on Country
5.5. Choose Map as the **Marker type**
5.6. Add Country to the Color **Marks Card**
5.7. Add Sales to the Label **Marks Card**
5.8. Format the Sales to $K (1 decimal)

Activity # 5: Exploring the Tooltip Marks Card

6. Now, for each country, we want to interactively explore the Sales for each product category
6.1. Click on the Tooltip Marks Card
    6.1.1. In the upper right corner, click on Insert
    6.1.2. Click on Sheets
        6.1.2.1. Select Category (the worksheet with the Sales distribution per product category)
        6.1.2.2. Hover over each country and observe what happens

6.1.2.3. Now, display the workbook in **Presentation Mode**



6.1.2.4. Rename your workbook

Challenge

1. Create a profit map
2. Color the map by Region
3. Interactively illustrate:
    a. The evolution of the annual sales for each product category using an area chart
    b. The sales across the segments using a bar char (color the chart by segment) (Format all the dollar figures in $K)

Laurentian University
Université Laurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

LAB 3: DATA VISUALIZATION WITH tableau III

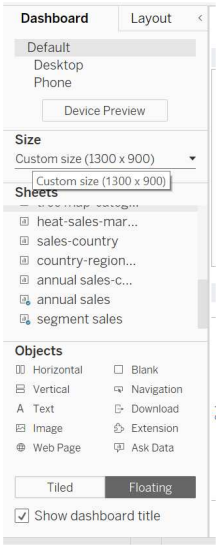
Objective of this lab: At the end of this lab, students will be able to create simple **dashboards** in Tableau

1. **Connect to the file Global "Superstore Orders 2016.xlsx"**

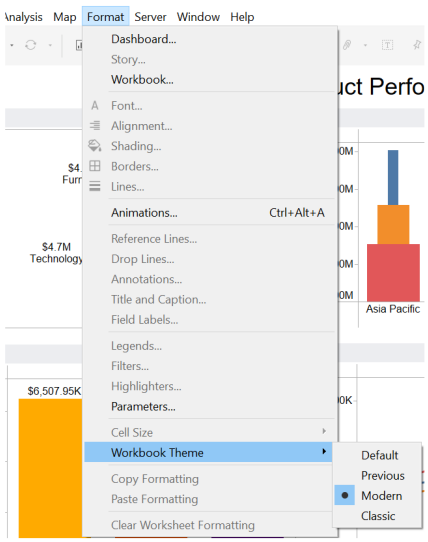
Activity # 1: Creating a dashboard

2. **Let's create a dashboard to illustrate the performance of the products based on the following criteria:**
    - Overall Sales
    - Sales across the Markets
    - Sales across the Segments
    - Annual Sales

2.1. Click on the Dashboard tool (Tool menu or the lower right corner)
2.2. Name the dashboard "Product Performance)
2.3. Let's format the dashboard
    2.3.1. Click on Format (toolbar)
    2.3.2. Change whichever parameter you want (see the big rectangle on the left side of your screen)
    2.3.3. Change the size to your liking



2.4. Let's add a title
    2.4.1. On the lower-left corner of your screen, active the **Show dashboard** title option
2.5. Now, move the relevant worksheets to the main canvas
2.6. Let's add a little makeup
    2.6.1. Click on format to change the theme to **Modern**

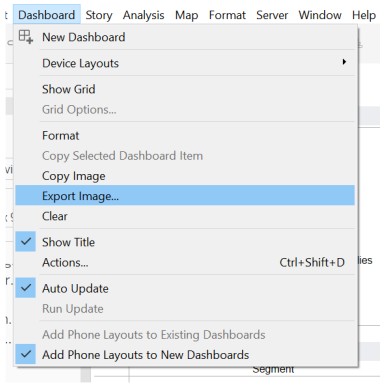


2.7. Let's export the dashboard as an image (for instance to be inserted in a report)

2.7.1. Click on Dashboard

2.7.1.1. Click on Export image

2.7.1.1.1. Save the image to any location you want (Choose any format/file extension you want)



Laurentian University
Université Laurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

LAB 3: DATA VISUALIZATION WITH tableau IV

Objective of this lab: At the end of this lab, students will be able to

- Create interactive **dashboards** and **stories** in Tableau
- Publish worksheets on **Tableau Public Server**

1. Connect to the file Global "Superstore Orders 2016.xlsx"

2. Activity # 1: Creating a sales analysis dashboard

2.1. Let's create a dashboard entitled Sales analysis, that combines the following visuals

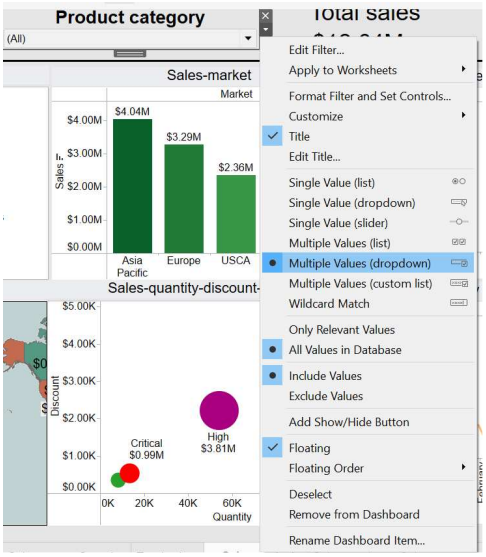
- A pie chart with the sales per product category
- A bar chart of sales across the markets. The bars are colored based on the magnitude of the sales
- A bar chart of the sales across the segments. The bars are colored based on the magnitude of the sales
- Map chart of the sales across the countries
- A plot to illustrate the relationship between the Quantity (x axis), the Discount (y axis), and the Order Priority
- The size of the bubbles is based on the Sales amount
- The bubbles are colored based on the Order Priority (choose intuitive colors)
- A plot of the monthly sales over the years
- The sum of all the Sales (at the upper right corner of the dashboard)
  - Create a new worksheet
  - Drag Sales to text (format to $M)
  - Remove all grid lines
- (Format everything efficiently)

2.2. To get full control over the size and the positioning of the **objects**, the latter will be **floating** instead of **tiled**.

2.3. Let's add a **filter** on the product category

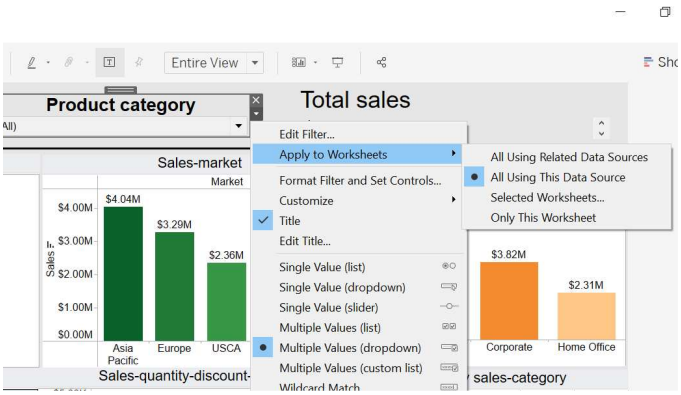
2.3.1. Click on product Category object

2.3.2. Let's change it to a dropdown menu



2.3.3. Let's apply the filter to all the dashboard elements

2.3.3.1. Click on the filter object

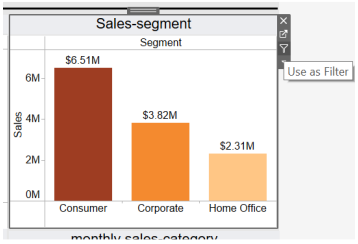


2.3.3.2. Filter the product categories and observe the changes across the dashboard elements

2.4. Let's create more interactive filters (based on any components of the charts on the dashboard)

2.4.1. Click on each object on the dashboard, but the filter and the total sales

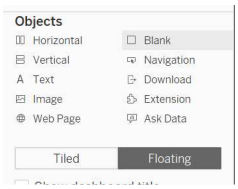
2.4.1.1. Select **Use as Filter**



2.4.1.2. Select each component of each object and observe the changes across the dashboard

2.5. Let's add a divider (line) between the titles and the charts

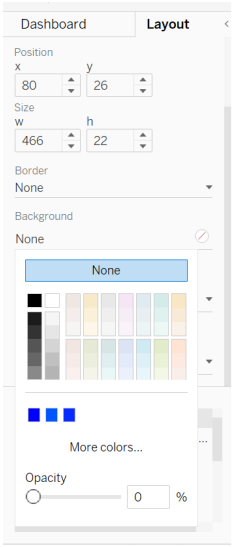
2.5.1. Double click on a blank **container**/object



2.5.2. Click on Layout in the upper left corner

2.5.3. Choose the same width as the dashboard's and decrease the height

2.5.4. Change the background color



3. Activity # 2: Creating a profit analysis dashboard

3.1. Let's use a different approach to create the objects (utilization of containers)

3.1.1. First let's create the following visuals

3.1.1.1. A bar chart of the profit across the region

3.1.1.2. Lines plots of the yearly quarterly profits

3.1.1.3. A bar chart of the profit per shipping mode

3.1.2. Let's create a new dashboard (Profit analysis)

3.1.2.1. As the bar chart of the profit across the region is big, let's put it in a horizontal **container**

3.1.2.2. Let's put the second chart in another horizon container below the first one

3.1.2.3. Let's put the last in a vertical container by the second chart

3.1.2.4. Let's use each object as an interactive filter

3.1.2.5. Let's create two other filters:

- Product category
- Segment

(Remark: you may need to create these filters on your worksheets)

3.1.3. Let's create a presentation page using a dashboard

3.1.3.1. Create a new dashboard (Presentation)

3.1.3.1.1. Let's add Laurentian logo

3.1.3.1.1.1. Go to the university website

3.1.3.1.1.1.1. Right-click the logo and copy the address

3.1.3.1.1.1.2. On the dashboard, under **Objects,** double-click the **Image** object

3.1.3.1.1.1.2.1. Click on **Lick to image** and paste the link

3.1.3.1.1.1.2.2. Click on **Ok**

3.1.3.1.1.2. Change the background color

3.1.3.1.1.3. Add a horizontal container

3.1.3.1.1.3.1. Double the **Text** object to provide a short summary or anything you like of your work.

3.1.3.1.2. Now, let's create a story combination the presentation, and the two dashboards

3.1.3.1.2.1. Create a new story

3.1.3.1.2.1.1. Name it anything you want

3.1.3.1.2.1.1.1. Drag you presentation page to the canvas and add a caption (like Presentation)

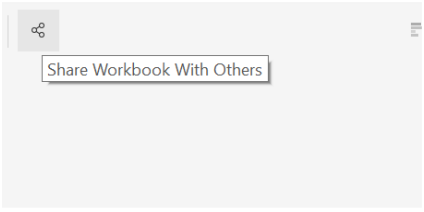
3.1.3.1.2.1.1.2. On the left, under **new story point**, click on **blank**

3.1.3.1.2.1.1.2.1. Add the Sales Analysis Dashboard

3.1.3.1.2.1.1.2.2. Do the same for the Profit Analysis

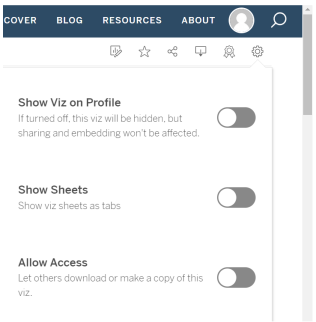
4. Activity # 3: Publishing your work

4.1. On the upper right corner of the screen, click on the **share** icon (make sure that your connexion is on the **Extract mode** (Data source))



4.2. Use your Tableau Public credentials to connect to the server

4.3. To control the level of visibility :



Laurentian University
Université Laurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

LAB 6: Numerical Descriptive Analytics With

+ a b | e a u

Objective of this lab: At the end of this lab, students will be able to

- Create frequency, relative frequency, and cumulative distributions with Tableau
- Calculate basic descriptive statistics with Tableau

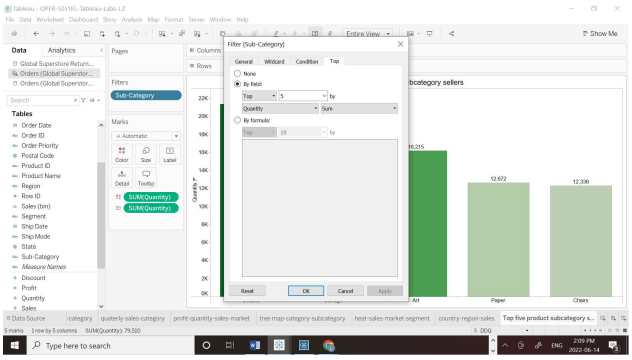
1. Connect to the file Global "Superstore Orders 2016.xlsx"

Activity # 1: Create frequency, relative frequency, and cumulative frequency distributions in Tableau.

Let's imagine we want to find the top five product subcategory sellers (quantity sold).

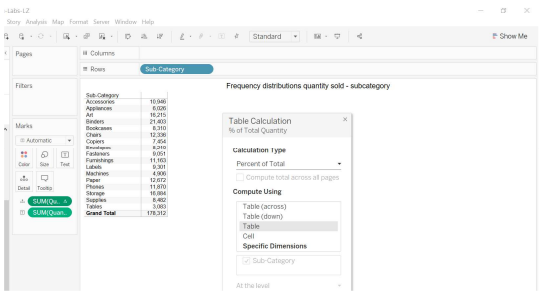
Strategy # 1: Let's use a bar graph

- Create a bar graph for the quantity sold for each product subcategory
    - Add the quantity sold on the bars
    - Color the bars based on the quantity sold
- We will answer the question using a filter:
    - Add the Subcategory to the filters and click on top to set the relevant parameters

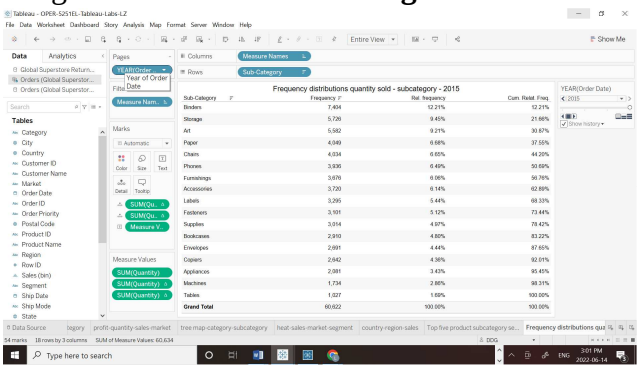


Strategy # 2: Let's conduct a deeper analysis through frequency distributions

- Let's create a table with the quantity sold for each product subcategory
    - Let's add the total (Analysis/Totals/Show Column Grand Total)
- Let's add a second column to calculate the relative frequencies
- Drag the Quantity field to the **Detail** Marks Cards
    - Right-Click the Quantity field
        - Add Table Calculation
            - Percent of
                - Table

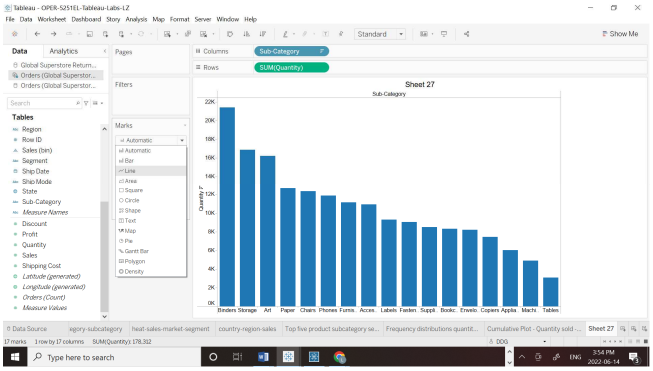


- Still on the Detail Marks Card, drag the Quantity field to the canvas
- To swap the columns (to have the relative frequencies as the second column)
    - Under **Measure Values** (very bottom of the Marks Cards), drag the second field to the top
- Now, let's create a third column to calculate the cumulative relative frequencies
    - Again, drag the Quantity to the Detail Marks Card
    - Right-click the field (make sure it's the last one added)
        - Add Table Calculation
            - Running Total
                - Click add secondary calculation (at the bottom of the menu)
                    - Percent of/Table (as we don't want the absolute numbers, but the percentages)
    - Drag the same field to the canvas
- Now, to order the product subcategories from top sellers to the least
    - Click on the frequency or relative frequency column
        - Click on the appropriate Sort function
- Compare the final output with the bar chart
  (What is the total contribution of the top five sellers?)
- Let's do some formatting
    - Right-click each column header
        - Edit aliases (to change the labels)
- Lastly, let's breakdown the view into a series of pages, each page representing a year
    - Drag the Order date field to **Pages card control**

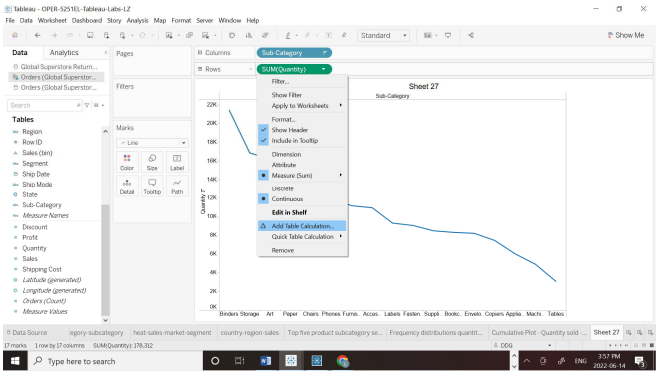


Strategy # 3: Let's create a **cumulative plot**

- This is a combination of the first two strategies
    - Determine the frequency distribution (quantity sold per product subcategory)
    - Transform the view to a bar chart
    - Sort the product sub-categories from largest to smallest quantity sold
    - Change the bars to a line



    - Calculate the cumulative relative frequencies (add Tableau calculation/Running total/ Percent of total)

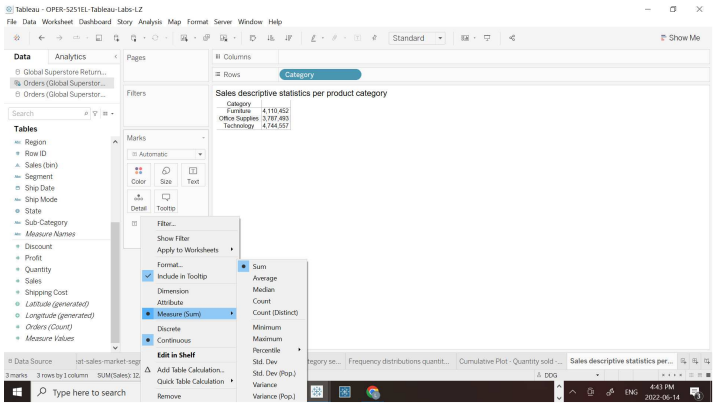


    - Let's add a **Drop Line** and display the corresponding cumulative frequency each time we click on the line
        - Right-click the view
            - Drop Lines
                - Show Drop Lines
        - Hold the Ctrl button (Command for mac) and drag the Quantity field from the Rows shelf to the Label Marks card (it's important to hold the Ctrl button)
        - Click the Label Marks card
            - Marks to Label
                - Change All to Selected
    - Lastly, let's do a little formatting
        - Double-click the y-axis and change the label to "Cumulative percentage"
        - Let's clean the information displayed on the view (when we click on the graph)
        - Click the Tooltip Marks card

- Delete everything and write “Cumulative % of the total quantity sold”

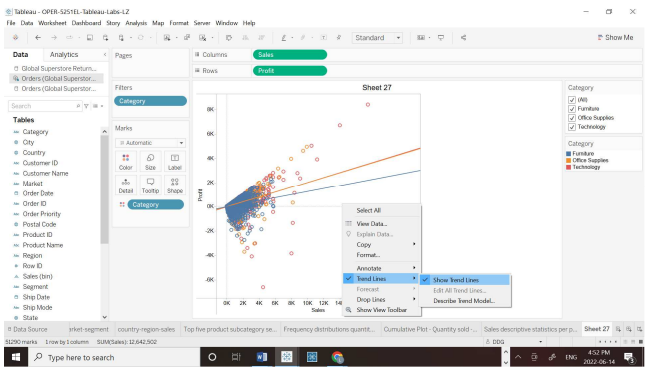
Activity # 2: Summarize the sales with a few numerical descriptive statistics

- We will calculate the min, max, average, standard deviation, and a few percentiles (including the median) sales for each product category
    - Actually you know how to do that ☺
        - Double-click the Category field
            - Drag the Sales field to the canvas and change the default Sum measure to the measure you want
            - Keep doing the same until you have all the measures you want



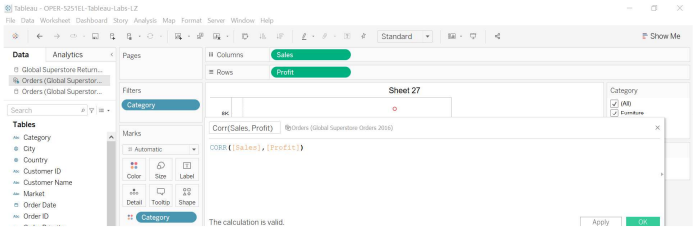
Activity # 3: Analyze the relationship between profit and sales

- Double click both fields
- To see the scatter plot
    - Deactivate the Aggregate function (under the analysis menu)
    - Color the dots by product category
    - To add a trend line
        - Right-click the view

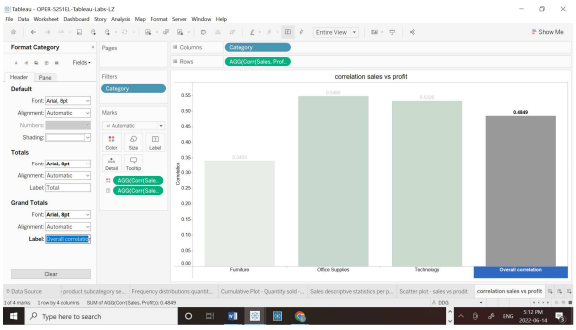


- Add a filter to interactively change the view based on the product category
- Let’s quantify the strength of the linear relationship (correlation)

- Create a new worksheet
- Click on Analysis
  - Create Calculated Field (Called Corr(Sales, Profit) or whatever name you want)
    - And type the following formula (using the Tableau built-in function **Corr**)
      (Note that as you type the field names, Tableau will provide you with suggestions, you can just double click on the field name or function you want.)



    - Let's display the correlation for each product category and the overall correlation as well
      - Double-click the Category field and your newly created field (correlation)
      - Show the Column Grand Total (overall correlation)
      - Display the correlations on a bar chart
      - Let's change the Grand Total label to Overall Correlation
        - Right-click the Label
          - Format
            - Change the label at the very bottom of the menu



    - To conclude, let's combine both views
      - You know how, right?
        - Dashboard!!!!!

Laurentian University
Université Laurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

LAB 8: Simple Linear Regression in

Learning outcome: At the end of this lab, students will be able to

- Conduct simple linear regression with python

Activity # 1: Estimate a simple linear regression model in python

1. Read the data file and store the content in my data
    a. Let's read the global superstore data
        i. Let's import the pandas library to store the data
        import pandas as pd
        ii. Let's provide the address of the file we want to read
        f_name = r'location of the file/name of the file.xlsx'
        iii. Let's read the data and store them in a variable called my_data
        my_data = pd.read_excel(fname, 'Orders')
        iv. As python is case-sensitive, let's read the field names to make sure we have them correctly later

```
var_names = list(my_data)
```

    b. We want to create a scatter for sales and profit
        i. Let's import a library to create a scatter plot later
        import matplotlib.pyplot as plt
        ii. Let's define our x and y variables

```
x = my_data['Sales'].values

y = my_data['Profit'].values
```

        iii. Let's create a scatter plot
        plt.scatter(x,y)
    c. Let's fit a simple linear regression to explain profit using sales
        i. Import the required library, create a regression model and define x and y variables

```
from sklearn.linear_model import LinearRegression

reg = LinearRegression()

x = x.reshape(-1,1)

y = y.reshape(-1,1)
```

        ii. To fit the regression line

```
reg.fit(x,y)
```

iii. To retrieve the intercept

reg.intercept_

iv. To retrieve the slope
reg.coef_

v. To retrieve the R squared

reg.score(x,y)

LaurentianUniversity
UniversitéLaurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

LAB 9: Multiple Linear Regression in

Learning Outcome: At the end of this lab, students will be able to perform multiple linear regression analysis with Python using the sklearn and statmodels packages

Description:

We have data of different houses, and we want to predict the selling price per unit area using:

- The transaction date
- The house age in years
- The distance from the nearest metro station
- The number of convenience stores in the area, and
- The location of the house given the geographic coordinates

Source: https://www.kaggle.com/datasets/quantbruce/real-estate-price-prediction

Activity # 1

We will import the following packages as follows:

import pandas as pd

import seaborn as sn

from sklearn.linear_model import LinearRegression

import statsmodels.api as sm

- Write a script
    - Use the pandas built-in function read_cvs to read the dataset (as it is a cvs file)
    - Let's assume you store the dataset in a variable called data
- Let's define our x and y data
    - We know that we have eight fields in the dataset; however, the first one is just an index field, so we won't use it.
    - We also observe that our x variable data span columns 2 to 6. We can read the x data in one shot as follows:
        - x = data.iloc[:,[1,2,3,4,5,6]] or x=data.iloc[:,1:7](The first colon means that we want to retrieve all the rows)
        - (Also observe that it seems we start reading the data from column 1 to column 7; this is not the case: in a list, Python assigns the index 0 to the first element, and the index n-1 to the last one, where n is the size of the list)
    - Similarly, we'll read our y data as follows:
        - y=data.iloc[:,7]
- Let's do a quick data exploration by creating a set of histograms and looking at all possible pairwise scatter plots between the x and y data
    - To do so, we'll use an amazing seaborn built-in function, called *pairplot*
        - For this, we need to create single dataframe for the x and y data
            - We'll use the pandas built-in function *concat* to concatenate the two dataframes.
                - In one shot:
                    - sn.pairplot(pd.concat([x,y],axis=1)) (the instruction option axis =1 means we add y add a new column to x)

- Now, let's use the sklearn package to fit the multiple linear regression
    - Let's fit our regression model
        - reg = LinearRegression()
        - reg.fit(x,y)
        - Let's print the regression parameters, and the R squared:
            - print("b0 = ", reg.intercept_)
            - print("b1 = ", reg.coef_)
            - print("R2 = ", reg.score(x,y))
            - (Please, write these values somewhere)
    - Now, let's use the statsmodels package
        - Let's fit the model
            - model = sm.OLS(y,x).fit() (Observe that in contrast with sklearn, we need to provide the y data first; and you need to CAPITALIZE OLS)
            - Run the following code to see how much information statsmodels provide you with:
                - model.summary()
            - What do you observe? (It looks like the two packages provide different estimates ☺)
            - No, by default, in contrast to sklearn, statsmodel doesn't include a constant in the model.
            - Now, let's include the constant as follows:
                - x = sm.add_constant(x) (I'll explain that in class)
            - Let's fit the model again:
                - model = sm.OLS(y,x).fit()
            - And retrieve the results:
                - model.summary()

Laurentian University
Université Laurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

LAB 10: Connecting Tableau with Python

Learning Outcome: At the end of this lab, students will be able to perform a multiple linear regression in Tableau by connecting with Python.

Activity # 1: Refresher

First, let's start by rerunning the multiple linear regression we performed in Lab 9 for the real estate dataset. We'll do it using the statsmodels package. We'll run two regression models, with and without constant.

Remember we want to predict the selling price per unit area using:

- The transaction date
- The house age in years
- The distance from the nearest metro station
- The number of convenience stores in the area, and
- The location of the house given the geographic coordinates

Activity # 2: Connect to the real estate dataset in Tableau

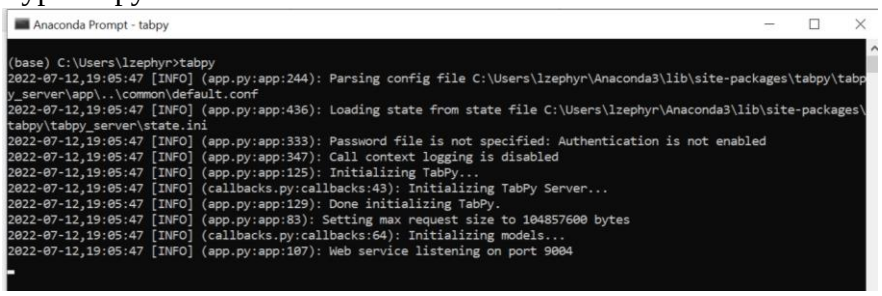
Activity # 3: Preliminary data exploration

- Let's built six different scatterplots (each predictor against the response variable)
    - Drag the "House Price of Unit Area" to the Rows shelf
    - Drag the six predictors to the Columns shelf

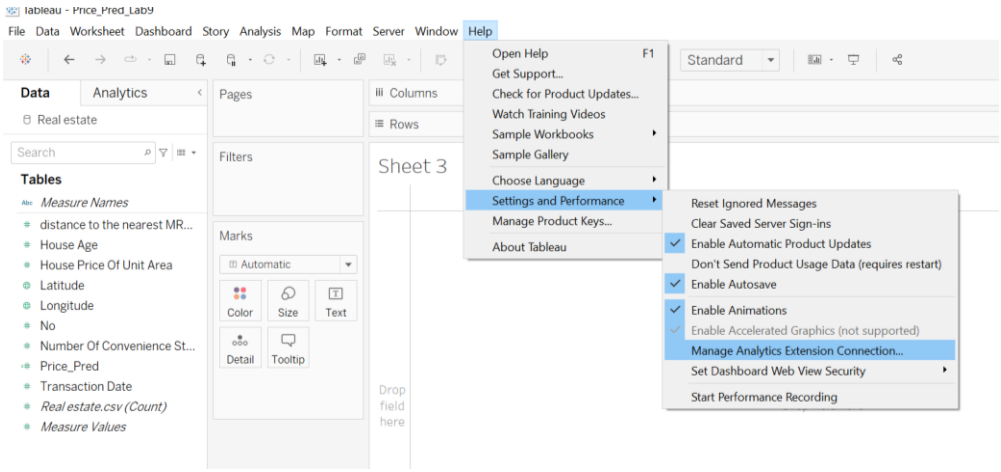
Activity # 4: Launch tabpy

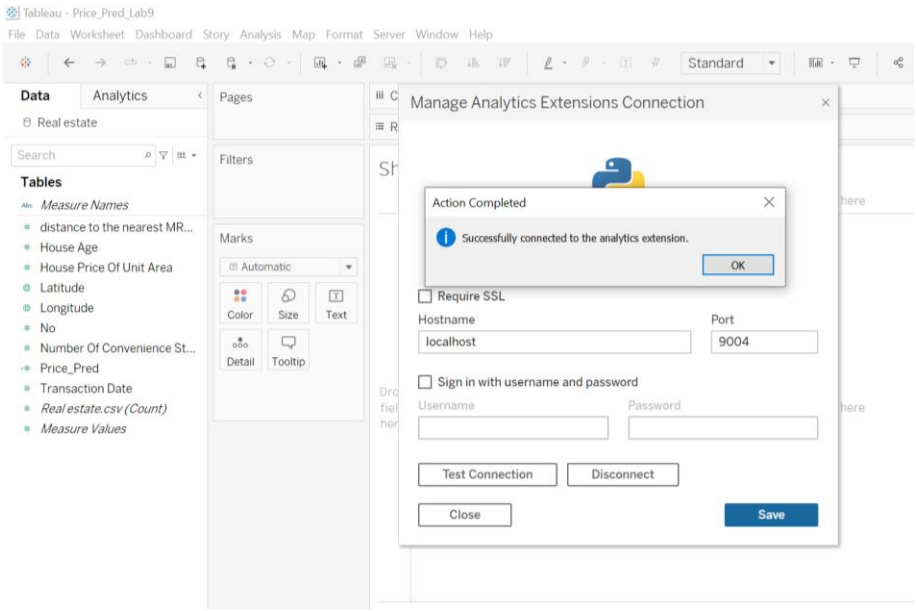
(I assume you already installed tabpy on your computer)

- To launch tabpy, follow either of the following two steps
    - At the Windows start menu, type anaconda prompt and click on it
    - Or find Anaconda Navigator and click on anaconda prompt
        - Type tabpy



- Test the connection





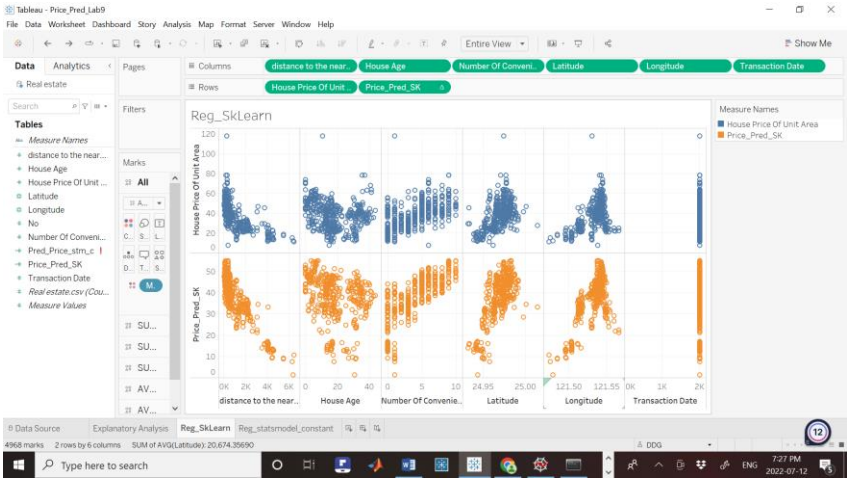
Activity 5: Doing the regression with sklearn

- First, we need to create a calculated field (analysis,…). Let's call it Price_Pred_SK
- We will need to write a Python script using the SCRIPT_REAL Tableau function (we use the real script because the output will be the predicted prices, which are real numbers) as follows

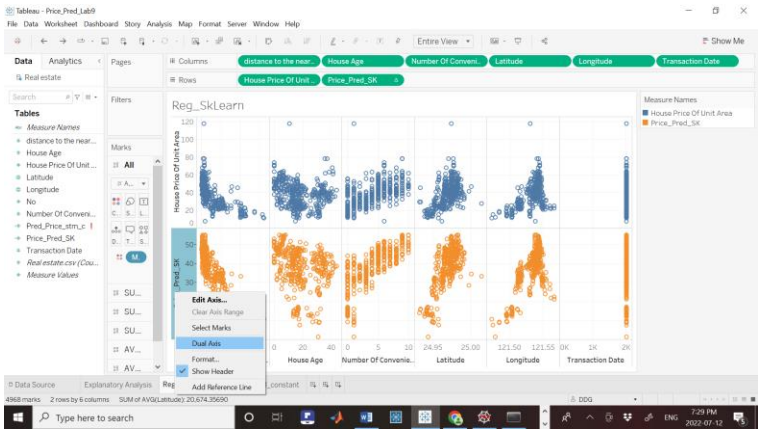
```
SCRIPT_REAL("

import numpy as np # we import the numpy package to reshape the x and y data (to make them matrix)

from sklearn.linear_model import LinearRegression

x=np.transpose(np.array([_arg2,_arg3,_arg4,_arg5,_arg6,_arg7])) #_arg2, means second variable, etc. Notice that the variables are listed at the end of the script

y=np.array(_arg1)

reg = LinearRegression()

reg.fit(x,y)

return reg.predict(x).tolist() # We return the fitted prices as a list of numbers for the x data in the dataset

",SUM([House Price Of Unit Area]),SUM([distance to the nearest MRT station]),SUM([Transaction Date]),SUM([House Age]),

SUM([Number Of Convenience Stores]),SUM([Latitude]),SUM([Longitude]))
```

Remark: Anything after an # is called a comment in Python. As such is not a piece of code

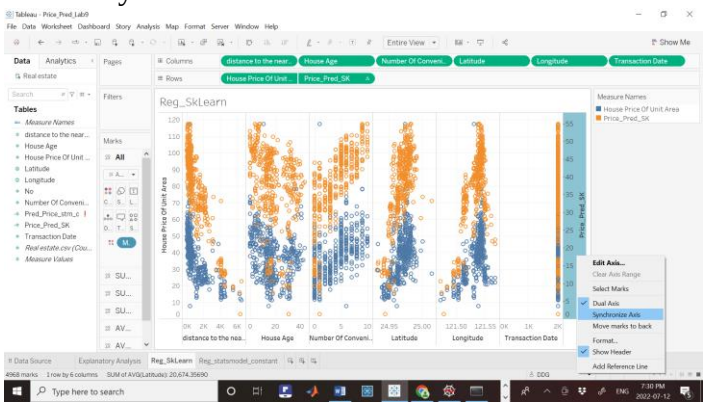
- Now, let's plot all the six predictors against the true prices and the fitted ones.
- Drag the six predictors to the Columns shelf
- Drag the Price field to the Rows shelf
- Drag the Price_Pred_SK to the Rows shelf as well



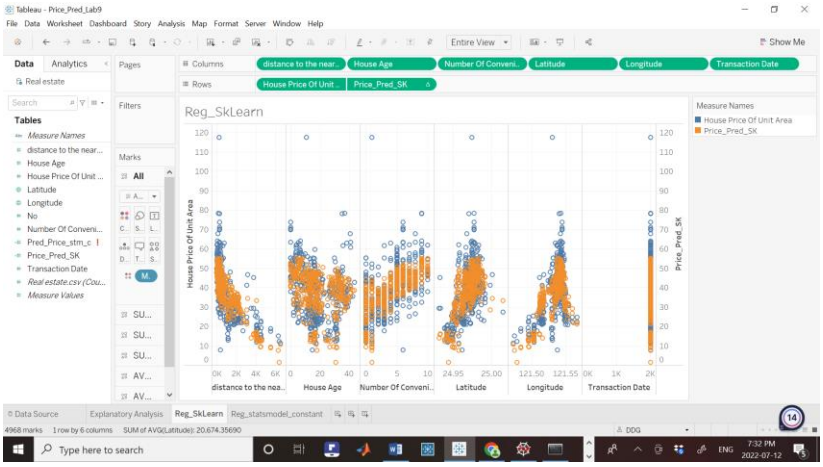
- But we want the exact prices (y) and the fitted ones to be one the same graph
  - Right-click the Price_Pred_SK axis
    - Dual axis



- Right-click the dual y axis
  - Synchronize



- Now, we can visually inspect the quality of the regression



Activity 6: Doing the regression with statsmodels

- Let's create a new calculated field, and call it Pred_Price_stm
- Again, we need to create a Python script a SCRIPT_REAL, as follows:

```
SCRIPT_REAL("

import numpy as np # we import the numpy package to reshape the x and y data (to make them matrix)

import statsmodels.api as sm


x=np.transpose(np.array([_arg2,_arg3,_arg4,_arg5,_arg6,_arg7])) #_arg2, means second variable, etc. Notice that the variables are listed at the end of the script

y=np.array(_arg1)

x = sm.add_constant(x) # adding the constant

model = sm.OLS(y,x).fit()

print(model.summary())

return model.predict(x).tolist() # We return the fitted prices as a list of numbers

",SUM([House Price Of Unit Area]),SUM([distance to the nearest MRT station]),SUM([Transaction Date]),SUM([House Age]),

SUM([Number Of Convenience
```

- Observe that we print the model summary
    - Go back to the Anaconda prompt for the output
- Again, let's do a visual inspection of the quality of the regression, as we did with the sklearn fit.

Laurentian University
Université Laurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

LAB 11: Regression Forecasting in

Learning Outcome: At the end of this lab, students will be able to perform regression forecasting, and assess the quality of the forecasts, with Python using the sklearn package.

Description:

We'll use the real estate datasets and a simple machine learning technique called dataset splitting to (i) build the regression model, and (ii) assess its quality.

Dataset Splitting

The dataset splitting technique consists in randomly splitting the dataset into two parts, the first part, called training dataset, will be used to build the regression, and the other part, called testing dataset, will be used to assess the quality of the regression. We'll use 70% of the data to train the regression model and the remaining 30% for testing.

The sklearn package allows doing the splitting, by importing the train_test_split module as follows:

```
from sklearn.model_selection import train_test_split
```

We'll do the random splitting, assuming the dataset is loaded into the pandas data frame data, as follows:

```
x = data.iloc[:,1:7]

y = data.iloc[:,7]

x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3)
```

Assessment

We'll assess the quality of the regression by comparing the R^2 on both the training and testing data.

Simulations

As the splitting is done randomly, if we repeated the process, we should expect the value of the R^2 to change each time. As a result, we'll conduct simulations (repeating the process a fixed number of times).

For each simulation run, we'll calculate and add the in-sample R^2's (calculated on the training data) and the out-of-sample R^2's (calculated on the testing data) to two **lists**, respectively, using the **append** Python built-in function. Toward this end, before starting the simulation, we'll create two empty lists, as follows:

```
R_squared_train = [] #(to store the training R^2's)

R_squared_test = [] #(to store the test R^2's)
```

To add an element to the first list, for instance, we'll call the append function as follows:

```
R_squared_train.append(the element we want to add)
```

Descriptive Statistics

We'll compare the in-sample R^2's and the out-of-sample R^2's on descriptive statistics. Toward this end, we'll add the two lists to a pandas data frame as follows:

```
R_squared = pd.DataFrame({'Training':R_squared_train,'Testing':R_squared_test})
```

We'll use the pandas **describe** built-in function to calculate the descriptive statistics:

stats = R_squared.describe()

Distributions

We could build the distributions of the two R^2's using the matplotlib package. Remember that we first need to import it:

import matplotlib.pyplot as plt

We can create the two histograms as follows:

plt.hist(R_squared_train)

plt.hist(R_squared_test)

Full Script

Below is the full script (without the distributions).

Remark: please, create a new script to run this.

```
import numpy as np

import pandas as pd

from sklearn.linear_model import LinearRegression

from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt


#############################################################################

fname = r'C:\\Users\\lzephyr\\Dropbox\\LU\\Courses\\Spring 2022\\Datasets\\Real estate.csv' (please change this to the location of the dataset on your computer)

data = pd.read_csv(fname)

#############################################################################

x = data.iloc[:,1:7]# we're using the 2nd, 3rd and 4th fields as predictors

y = data.iloc[:,7] # we're using the last field (sales) are our dependent variable

#############################################################################


n_simul = 1000 # number of simulations runs I want to perform

reg = LinearRegression()

R_squared_train = []

R_squared_test = []
```

```
################################################################################

for i in range(n_simul):
    x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3) # random splitting of the dataset
    reg.fit(x_train,y_train) # fitting the regression model
    R_squared_train.append(reg.score(x_train,y_train)) # retrieving the R squared
    y_pred = reg.predict(x_test) # predicting the y_test using the test predictors

    # Next we calculate the out-of-sample R squared using the formula R^2 = 1 - SSE/SST
    SST_pred = sum((y_test - np.mean(y_test))**2)
    SSE_pred = sum((y_test-y_pred)**2)
    R_squared_pred = 1-SSE_pred/SST_pred
    R_squared_test.append(R_squared_pred)
################################################################################

R_squared = pd.DataFrame({'Training':R_squared_train,'Testing':R_squared_test})
stats = R_squared.describe()
print(stats)
################################################################################
```

Laurentian University
Université Laurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

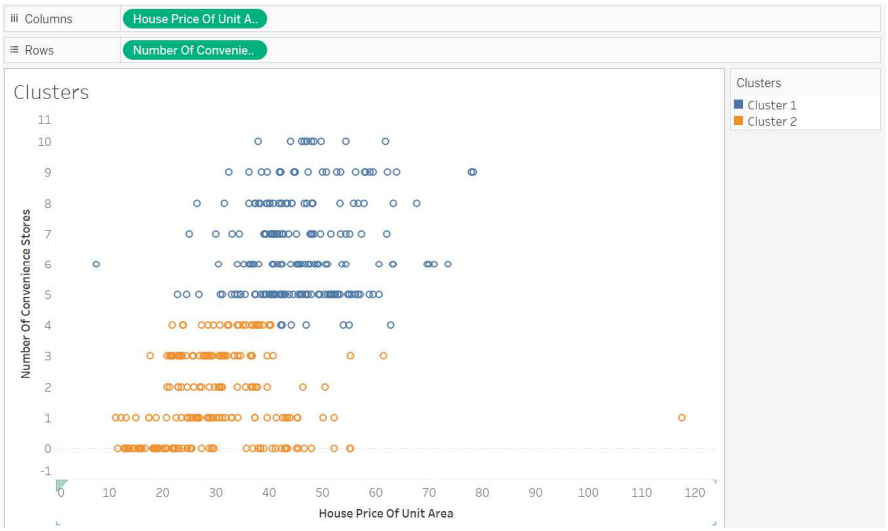
LAB 12: Cluster Analysis with

tableau

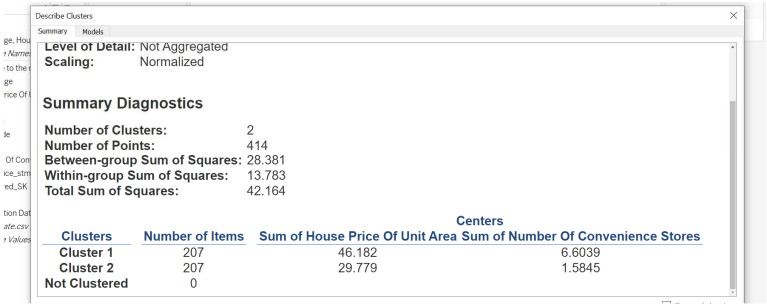
Learning outcome: At the end of this lab, students will be able to

- Conduct cluster analysis with Tableau
- Assess the quality of the clustering performed by tableau

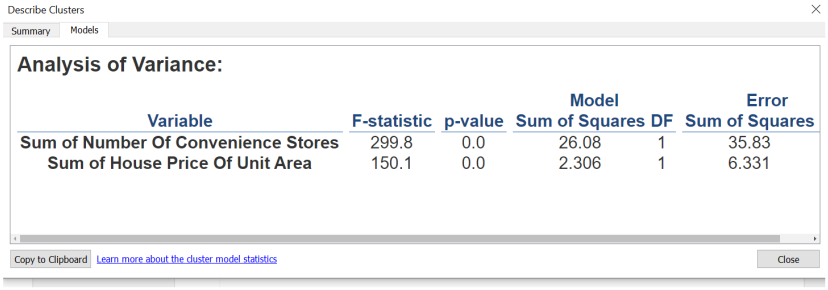
1. **Connect to the real estate dataset**
2. **Objective of the exercise:** Find out whether we can discover hidden similarities between the houses. Toward this end, we'll try to group/cluster them based on the following variables/features:
   - Their unit
   - Their age
   - Their distance to the nearest metro station; and
   - The number of convenience stores nearby.
3. Let's start by creating a scatter plot between the unit price and the number of convenience stores.
   - Drag house price to the Columns shelf and Number of convenience stores to the Rows, or vice versa.
4. Click on the Analytics pane, double-click "cluster" under the Model section, or click on cluster and drag it to the canvas.
5. You see that Tableau automatically creates two clusters.



6. Now, let's explore a few things.
   - Click on **Clusters** on the Color Marks Cards and **Edit clusters**
     - Notice that Tableau automatically determines the number of clusters
     - Change the number of clusters and observe what happens
     - Let's keep the automatic option
   - Click on Clusters again and **Describe clusters**
     - Look at the **results summary** and try to understand whether the clustering is sensical



- Now, let's take a look at the statistical results under the model menu (top left corner)



7. Let's add another feature to the **clustering**

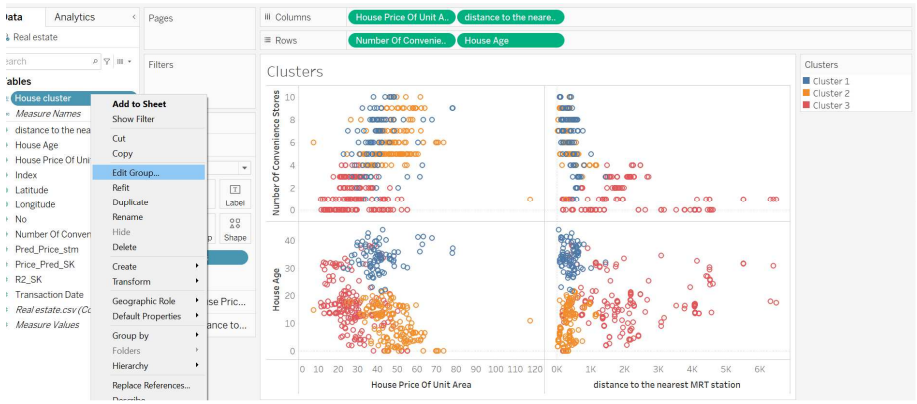
- Click on Clusters again
  - Edit clusters
    - Drag House Age in the box containing the two other features

- Let's inspect the results again
  - Describe clusters
    - Summary
    - Model
- Let's add the distance to the nearest metro station

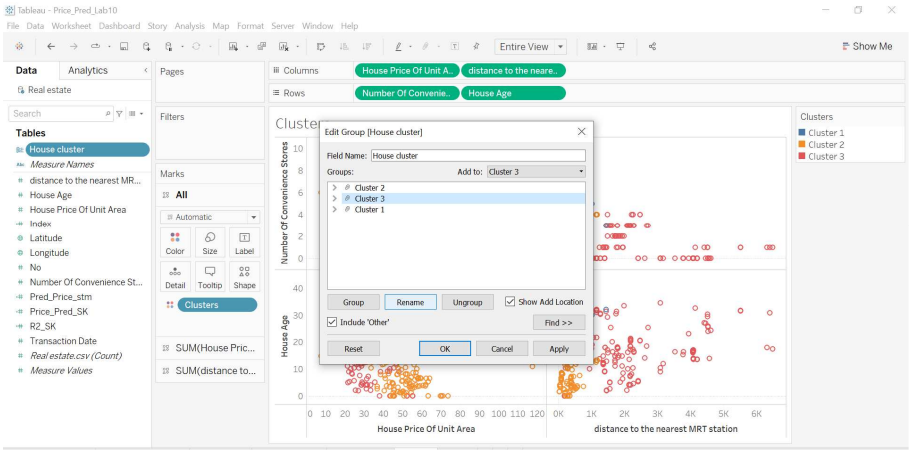
8. Let's plot our scatter plot matrix

9. Let's use the cluster as a new field now

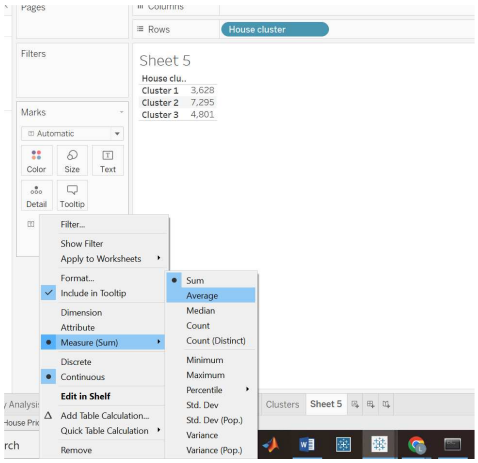
- Click on Clusters
  - Hold it
    - Drag it to the data pane (the pane with the field names)
- Let's change the name and call the new field House cluster
  - Click or right-click the newly created field
  - Try to find a label for each cluster based on the results and change their labels accordingly
    - Right-click the newly created field
      - Edit Group



      - Click on each label and then rename (at the bottom of the box)



- Create a new worksheet
- Calculate the average price of the houses in each cluster, and compare the averages with those in clustering results section.



Remark: Clustering analysis is relatively easy with Python. You'll find a lot of good tutorials on YouTube and Google.

LaurentianUniversity
UniversitéLaurentienne

SCHOOL OF BUSINESS ADMINISTRATION

OPER-5151EL-01: Business Analytics

LAB 13: Implementing a Simple Linear Discriminant Analysis in

Learning Outcome: At the end of this lab, students will be able to:

- perform a simple Linear Discriminant analysis in Python via a simple linear regression model
- calculate and visualize a **confusion matrix** to assess the quality of the model

Description:

We have a dataset of 284,807 European credit card transactions, of which 492 were fraudulent. The dataset contains 32 features, including a field "Class", with two values: 1: the transaction was fraudulent and 0, otherwise. For because of the sensitivity of the information, most of the data were transformed through a mathematical technique (data reduction) called Principal Component Analysis, except for the time and the amount of the transaction. As a result, we won't be able to interpret of the variables.

We'll use the following packages:

```
import pandas as pd

from sklearn.linear_model import LinearRegression

import matplotlib.pyplot as plt

from sklearn.metrics import confusion_matrix

import numpy as np

import seaborn as sn
```

Source: https://www.kaggle.com/code/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets

Step # 1 : Exploratory Analysis

- Normally, as usual, we'd need to do an exploratory analysis

Step # 2 : Reading the Variables

- We'll read the data the CVS file using the pandas library
- As we have a lot of fields, we'll read the x and y data a bit more clever
    - dfile=r'C:\Users\lzephyr\Dropbox\LU\Courses\Spring 2022\Datasets\creditcard.csv' # Change the adress, please.
    - credit_data = pd.read_csv(dfile)
    - y=credit_data.loc[:,['Class']] # this variable defines the category of each transaction
    - x=credit_data.loc[:,~credit_data.columns.isin(['Class'])] # all variables, but the category of the transaction

Step # 3 : Fitting a simple linear regression model

- reg = LinearRegression()
- reg.fit(x,y)
- predic_class=reg.predict(x) # we use the model to predict the variable y

Step # 4 : Calculation of the discriminant scores

- credit_data['Pred_reg']=predic_class # first, we add the predictions as a new column in the dataset
- dis_score_fraud = credit_data.loc[credit_data['Class']==1,'Pred_reg'].mean() # we calculate the mean discriminant score of the fraudulent transactions
- dis_score_good = credit_data.loc[credit_data['Class']==0,'Pred_reg'].mean() # we calculate the mean discriminant score of the normal transactions
- dis_score = 0.5*(dis_score_good + dis_score_fraud)# we calculate the average of the two scores (**cut-off score**).

Step # 5 : We classify the transactions based on the cut-off score

Pred_class = [] # we'll put the predicted categories in a list

for i in range(len(y)):

if credit_data['Pred_reg'][i]>=dis_score :

Pred_class.append(1) # if the discriminant score of the transaction is above or equal to the cut-off score, the transaction is classified as fraudulent

else:

Pred_class.append(0), otherwise the transacted is classified as normal

Step # 6 : Assessing the quality of the classifications

We'll use a simple device called **confusion matrix**. For each transaction category, we'll compare the true nature of the transaction and the classification using a simple cross table (with relative frequencies). The idea is to determine for each transaction category, how many were correctly and incorrectly classified. We'll use the sklearn package to create the confusion matrix as follows:

- from sklearn.metrics import confusion_matrix
- conf_matrix = confusion_matrix(y,Pred_class)

I want to transform the absolute values into relative frequencies.

- n,m = conf_matrix.shape
- for i in range(n):
    - for j in range(m):
    - conf_mat_rel[i][j]=np.round(conf_matrix[i][j]/sum(conf_matrix[i]),5)

Step # 7 : Plotting the confusion matrix (relative frequencies) as a heatmap and saving it

- conf_cm = pd.DataFrame(conf_mat_rel, range(2), range(2))
- fig = plt.figure()
- sn.heatmap(conf_cm,annot=True,fmt=".2%")
- plt.show()
- fig.savefig('confusion heatmap.pdf')