# What python package will we use?

|  |  |
| --- | --- |
| datetime | optional |
| pandas | Go to dataframe and data wrangling package |
| seaborn | For vizualisation |
| Numpy | For numerical calculation |

# Where to find the data

<https://github.com/Emmanuel75/Data>

# BestRunCorp

1. Open "BestRunCorp\_Retail.csv" in a DataFrame. Print it shape, have a look at it  
     
   df = pd.read\_csv("BestRunCorp\_Retail.csv")

print(df.shape)

df.head()

1. Look at the date columns, convert it to datetime (hint use pandas function to\_datetime)
2. Replace it by new columns Day, Month, Year
3. Replace column “Year-Quarter” by a new column Quarter
4. Check columns data type and convert to numeric what needed
5. Display the sales per year and then the sales per quarter and year using pandas groupby
6. Visualize quantity sold by year in a bar plot using seaborn
7. Visualize both quantity sold and sale revenue by year in a bar plot using seaborn
8. Display the aggregation of each numerical value per year in a pie chart (tip: iterate on relevant columns)
9. Display the number of issues reported by year in a line chart
10. Sum of customer satisfaction does not mean much. Mean would make more sense. Aggregate with mean. Display customer satisfaction by month in a bar chart
11. Display a data frame with aggregation by year and month. Customer satisfaction being aggregated by mean and nulber of issues reported being aggregated by mean and sum
12. Dispay in a chart the mean number of issues reported by months (use pandas DataFrame plot)
13. Display in a chart the average sales by cities
14. Iterates on plt.style.available to see difference (use matplotlib.style.use(style) to change chart style and pandas plot for charting)
15. About Null behavior  
    df\_null = pd.DataFrame({"Person":

                   ["John", "Myla", "Lewis", "John", "Myla"],

                   "Age": [24., np.nan, 21., 33, 26],

                   "Single": [False, True, True, True, False]})

* 1. Try df\_null.count() and df\_null.nunique() and compare results

1. Plot Average sales by cities sorted by values
2. Add to the dataframe the rank of sales revenue for each city. Experiment with the different methods
3. Sort the dataframe according to rank
4. Check how the NaN are ranked. Experiment with the different value of na\_option
5. So what are the top best countries?
6. Lookup. get data about country population for example from: https://data.worldbank.org/indicator/SP.POP.TOTL and add a column to previous dataset (df8) with country population
7. Get the list of country sorted by the 2020 population
8. “merge” with the original dataset (left outer join on country)
9. Did it work as expected for every country ? Why ?

# Jenkins

1. Open the data and have look at them.  
   df = pd.read\_csv("jenkins.csv")
2. Using groupby and nunique, display the number of issues by status and the number of issues by type
3. For each type of issues display the number of issues by status
4. Convert the date columns to datetime. Extract in a new columns, year, month, day and iso day/iso week / iso year
5. Display the number of issues created by month
6. Display the number of issues resolution by month
7. Display the issues created after 2019
8. Display the on the same chart the the number of issue created and resolved by month