CA\_2 = Teacher Note:

1. Visualization: types , rational, justification, serie
2. EAD: Visualization , rationalization ( explain for use the colours
3. Sentimental Analysis : web scraping
   * 1. Spider
     2. Any web scrapping ( not only tweet )
4. Python Libraries.:

**Prob\_distributions\_calculations**

import numpy as np

import pandas as pd

import statistics as stats

import numpy as np

import seaborn as sns

from scipy import stats

import scipy as scipy

from scipy.stats import poisson

from scipy.stats import binom

from scipy.stats import levene

import matplotlib.pyplot as plt

import statsmodels.api as sm

from statsmodels.formula.api import ols

import pingouin as pg

[*https://learnpython.com/blog/print-table-in-python/*](https://learnpython.com/blog/print-table-in-python/) *(shows tables)*

*stats.probplot*

*stats.shapiro*

*aov = sm.stats.anova\_lm*

1. Machine Learning (tutorials)

%matplotlib inline

from preamble import \*

***# Import plot\_acf***

from statsmodels.graphics.tsaplots import plot\_acf

***# Plot ACF of data\_monthly***

plot\_acf(air\_passengers, lags = 40);

***#import warnings***

warnings.filterwarnings()

import nltk

import mglearn

import re

import sklearn

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import MinMaxScaler

from sklearn.preprocessing import PolynomialFeatures

from sklearn.preprocessing import scale

from sklearn.feature\_selection import RFE

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import cross\_val\_score

from sklearn.model\_selection import KFold

from sklearn.model\_selection import GridSearchCV

from sklearn.pipeline import make\_pipeline

*import openpyxl*

from cycler import cycler

# fit multiple polynomial features

degrees = [1, 2, 3, 6, 10, 20]

# visualise train and test predictions

# note that the y axis is on a log scale

plt.figure(figsize=(16, 8))

# train data

plt.subplot(121)

plt.scatter(X\_train, y\_train)

plt.yscale('log')

plt.title("Train data")

for i, degree in enumerate(degrees):

plt.scatter(X\_train, y\_train\_pred[:, i], s=15, label=str(degree))

plt.legend(loc='upper left')

# test data

plt.subplot(122)

plt.scatter(X\_test, y\_test)

plt.yscale('log')

plt.title("Test data")

for i, degree in enumerate(degrees):

plt.scatter(X\_test, y\_test\_pred[:, i], label=str(degree))

plt.legend(loc='upper left')

1. Data visualization (tutorials)

pip install altair vega\_datasets

import altair as alt

import pandas as pd

lambda

import plotly.express as px

import plotly.graph\_objects as go

import plotly.express as px

import tkinter as tk

from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg

from matplotlib.figure import Figure

import tkinter as tk

from tkinter.ttk import