**Exam Notes**

**Class Notes**

1. [Class 5. Introduction to Python: data structures, control flow with NOTES URL](https://nbviewer.jupyter.org/github/Parandzem-S/MGMT300-Summer2021/blob/main/Class%205.%20Introduction%20to%20Python/Class%205.%20Python%20Basics.ipynb)
2. [Class 6: Functions, While loop, Pandas, Matplotlib URL](https://github.com/Parandzem-S/MGMT300-Summer2021/blob/main/Class%206.%20Functions%2C%20While%20loop%2C%20Pandas%2C%20Matplotlib/Class%206.%20Functions%2C%20While%20loop%2C%20Pandas%2C%20Matplotlib.ipynb)
3. [Class 7. Data Visualization, Reading Data from Web, Interactive Plotting with Notes](https://nbviewer.jupyter.org/github/Parandzem-S/MGMT300-Summer2021/blob/main/Class%207.%20Data%20Visulization%2C%20Reading%20Data%20from%20Web%2C%20Interactive%20Plotting/Class%207.%20Data%20Visualization%2C%20Reading%20Data%20from%20Web%2C%20Interactive%20Plotting%20Notes.ipynb)

[URL](https://nbviewer.jupyter.org/github/Parandzem-S/MGMT300-Summer2021/blob/main/Class%207.%20Data%20Visulization%2C%20Reading%20Data%20from%20Web%2C%20Interactive%20Plotting/Class%207.%20Data%20Visualization%2C%20Reading%20Data%20from%20Web%2C%20Interactive%20Plotting%20Notes.ipynb)

1. [Class 8. Statistical Tests from Scipy and Overview of Section 2 with NOTES URL](https://github.com/Parandzem-S/MGMT300-Summer2021/blob/main/Class%208%20Statistical%20Tests%20and%20Overview/Class%208.%20Statistical%20Tests%20from%20Scipy%20and%20Overview.ipynb)
2. [Basic Charts](https://nbviewer.jupyter.org/github/Parandzem-S/Blog_metric.am/blob/master/blog1_ps_matplotlib.ipynb)

**Libraries to Import**

!pip install quandl

!pip install plotly

!pip install pandas\_datareader

import pandas as pd

from pandas\_datareader import data as dt #will be used to download data from yahoo finance

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import quandl

from plotly.offline import iplot

from plotly.offline import init\_notebook\_mode

import plotly.graph\_objects as go

from scipy.stats import pearsonr

from scipy.stats import ttest\_ind

from scipy.stats import f\_oneway

init\_notebook\_mode(connected=True)

**ADD DATA + SIMPLE STUFF**

wine\_df.info()

data=pd.read\_excel("Data.xlsx",sheet\_name="Data")

data.head(10)

data.shape

data\_salary['race'].describe()

data\_salary['race'].value\_counts()

*#extracting unique values*

wine\_df.Varietal.unique()

*#number of unique values*

wine\_df.Varietal.nunique()

Q: How many columns have **missing values**? data.isna().any() + data.isna().sum()

Q: **Drop** missing values. data.dropna(inplace=True) Check with: data.isna().any()

Q: Dropping 2 variables in one line - data.drop(["Rent\_Amount","floor"],axis=1,inplace=True)

**Pivot table** example - round(pd.pivot\_table(data,index='furniture',values='area', aggfunc=['mean','count']),2)

**Defining simple IF function**

def house\_type(data):

if data["furniture"]=="not furnished" and data["Animal\_Acceptance"]=="not acept":

return "not interested in offer"

else:

return "interested"

**Finding specific data** - print(house\_type(data[data["House\_ID"] == 344].iloc[0]))

**For loop+interactive histogram**

for i in data[["area","rooms","Floor\_Cleaned","Rent\_Amount\_Cleaned"]].columns:

fig=go.Figure(data=[go.Histogram(x=data[i],

marker\_color='#bfb2b9',

nbinsx=5)])

fig.update\_layout(

title=i,

height=500,

width=500)

fig.show()

**Data from Yahoo**

from pandas\_datareader import data

FB\_df=data.DataReader(name="FB",

data\_source='yahoo',

start='2019-08-03', end='2020-08-03')

**Av and median** - print("Average: ", round(FB\_df.Close.mean(),2))

print("Median: ", round(FB\_df.Close.median(),2))

**2 line chart (2 variables)**

plt.figure(figsize=(9,8))

sns.lineplot(data=FB\_df[["Open","Close"]],

palette=['#1d63f0','#eb213c'])

plt.show()

CLASS 3

1. plt.figure(figsize=(8,8)) *#changing the size of figure*

**plt.subplot(2,1,1)** \* *#first plot when plotting to figures side by side, or under each other*

plt.scatter(data\_numeric.horsepow, *#X-axis values*

data\_numeric.price *#Y-axis values*

)

plt.title("Price VS Horsepower") *#title of the first chart*

plt.subplot(2,1,2) *#second plot*

plt.scatter(data\_numeric.horsepow, *#X-axis values*

data\_numeric.sales *#Y-axis values*

)

plt.title("Sales VS Horsepower") *#title of second chart*

plt.show()

* Part means: (x,y,z)

x=number of rows or columns

y=change to see the charts next to each other

z=the rank of the chart in the output

WITHOUT subplot python automatically puts all charts together in one.

1. Box plots are good for finding **outliers** in the data.

To remove - Problem 8 - plt.figure(figsize=(9,7),facecolor="#f2f4f9")

plt.bar(data\_salary.age, my\_data.income,color="#5886a5", label='<=50K')

plt.bar(data\_salary.age, my\_data.income, color="#004c6d", label='>=50K') #label will allow to have names of variables on graph

plt.ylim(0,800) #define the range of y axis

#giving titles

plt.title("Field Goals Made and Field Goals Attempts")

plt.xlabel("Positions")

plt.legend() #to create legend so that we can differenciate variables

plt.show() #show graph

TRYING TO BUILD THE CHART

1. Important

*#calculating mean heights by position*

mean\_height=my\_data.Height.groupby(my\_data.Pos).mean() *#returns the mean value by each position*

posistion=my\_data.Pos.unique() *#returns array with names of positions*

plt.figure(figsize=(9,7),facecolor="#f9f5f2") *#making plot bigger*

*#draw plot*

plt.plot(posistion,

mean\_height,

color="#ff8019",

marker='o')

*#giving titles*

plt.title("WNBA Players Mean Height by Position")

plt.ylabel("Height (cm)")

plt.xlabel("Positions")

plt.show()

## LAST CLASS (8) - 30.07

**Excel Overview**

1. Wine data

=VLOOKUP([@[Offer '#]],OfferInformation!$C$2:$G$33,5,FALSE)

VLOOKUP

* If we’re bringing data to “X” sheet, lookup value should be in “X” sheet
* Values should be unique in the sheet from where you’re bringing the data
* ID should be on the **left** of the column values that you’re trying to bring

=INDEX(OfferInformation!$A:$A,MATCH([@[Offer '#]],OfferInformation!$C:$C,0)) - the last value here before exact match is the variable (in a different sheet, that has unique values)

* The first argument of INDEX is what we want to bring (e.g Campaign)
* The first argument of MATCH (lookup value) should be unique in the sheet from where we’re getting the data. (it can have duplicates in sheet where you’re righting the formula)

=VLOOKUP([@[Offer '#]],OfferInformation!$C$2:$F$33,4,FALSE)

**Python**

**Libraries needed for data analyzation in Python**

#Reading libs

import pandas as pd

from pandas\_datareader import data as dt #will be used to download data from yahoo finance

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import quandl

from plotly.offline import iplot

from plotly.offline import init\_notebook\_mode

import plotly.graph\_objects as go

from scipy.stats import pearsonr

from scipy.stats import ttest\_ind #Independent Sample T-test

from scipy.stats import f\_oneway #One-Way ANOVA

init\_notebook\_mode(connected=True)

**Creating a Pie Chart**

col=["#003f5c","#a05195"]

plt.figure(figsize=(12,8),facecolor="#f2f4f9")

plt.pie(data\_salary.Geder.value\_counts(),

labels=data\_salary.income.unique(),

colors=col)

plt.title("Relationship between Income and Gender")

plt.legend()

plt.show()

8c

plt.figure(figsize=(20,15))

pd.crosstab(data\_salary.income,data\_salary.race).plot(kind="bar",figsize=(15,6),color=['#FFC300','#581845'])

plt.title('Relationship between Income and Race')

plt.xticks(rotation = 0)

plt.legend(["Female", "Male"])

plt.show()