# Python Chilla Pandas Assignment

> **Title= "Mr"\ Name= "Ali Nawaz"\ email = "nawazktk99@gmail.com"\ whatsapp = "03358043653"\ Artificial Intelligence Engineer at NUST\ Education : Master in Software Engineering**

In [ ]:
```
# import all the lib
import pandas as pd
import seaborn as sns
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import linear_model
import sklearn.metrics as sm
import matplotlib.pyplot as plt

# Importing Linear Regression model from scikit learn
from sklearn.linear_model import LinearRegression
# Importing metrics for the evaluation of the model
from sklearn.metrics import r2_score,mean_squared_error

# read the dataset using pandas
data = pd.read_csv('D:/Python ka Chilla/python_chilla/data/Salary_Data.csv')
```

## In this Notebook we are going to know about Machine Learning in python

> Choosing right statictical method\ do's and don'ts of statistics\ reliable results\ paper ewvision with proof of statistical test\ making data visualization\ interpreting results\

## Test and Teir types

# ParametericTest

> more reliable results\ first we have meet the assumption\ e.g how much male and female in this group

# non-Parameteric Test

> less reliable results\ calculate the rank of data\ e.g how much male and female have age in 22, 30 etc in this group

**Before stating we should start with Normality Test**

Test to be used:

- Shapiro wilk test specific (reliable)
- Kolmogorve-smirnov Test General (less reliable)

**2nd Step is check the Homogeneity Test**

- variance of the variable is data are equal
- Test be used: **Levene's Test**

**3rd PurposeTest**

- Know the pupose of your resaerch question

# Two types of purposes

1. comparsion

> - Differece
> - Compare two group not single
> - e.g male vs female
> - control group vs short group reearch wise

2. Relationship

> - find a connection
> - can food predict weight of a group of individuals
> - do fertilizer applicatiion increases crop growth?
> - we will see connection correlation, causation, prediction

## Data type Step 4

> know the type of data you are working with

# Two Types of Data

> catergorical (Qualitative, non numerical meaning, e.g character, factors) Continuous (Quantitative, numerical , e.g number, int or float represent)

## Statistical Test

> Choose a statistical test from three main families

1. Chi-Squared (Purpose: comparision, Data: Categorical Only)
2. t-Test / ANOVA (purpose: Comparison, Data: Categorical and continous)
3. Correlation (purpose: Relationship, Data: Continous only )

## 1- Chi-Squared

types:

> chi-squred test of homogeneity chi-squred test of independence

when to use?\ nothing effrects this.\ can be use with any number of levels or groups

## 2- t-Test/ANOVA

types:

> One sample t-test: for one sample group with a known mean\ Two-sample:
>
> - un-paired t-test (two dif group)\
> - paired t-test (same group twice)
>
> ANOVA: (analysis of variance [3+ level or group are involved])
>
> - one way ANOVA (even one of group is significant you will get significant result but does'nt tell you which one)\
> - Two Way ANOVA\
> - Repeated measured of ANOVA (3+ paired group, scale up of paired test)

when to use?\ nothing effrects this.\ can be use with any number of levels or groups

# Correlation

### Wen and where to use?

**Types:**\

> Pearson Correlation (one-independent and one dependent variable)\ Regression (one-independent and one dependent variable)\

**Correlation:**

tell us how closely connected two variables are?

is food a predictor of weight gain?

Regeression:

Tells us a specfic mathematical equation that descubes the relationship. e.g missing values can be predicited like this

# Important thing

**Assumption about your data**\ Your data will be nomal distributed\ Your data will be nomal gaussion disstribution

if you not follow the assumption than ur results will be worst

## Types of ANOVA Test

links is here for moer info: URL

## Other Test

> Reliablity test (its have diff type) \ Inter rater reliablity test (its have diff type)\ Validity test (its have diff type)\ Sample Size compuatuib (its have diff type)

In [ ]:
```
import plotly.express as px
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

# statistics outline

> - Descriptive statisctics
> - Data Visuliation
> - Probablity Distribution
> - Hypothesis testing
> - Regression Analysis

**Pakacges we will cover and covered**

> pandas (Data Structures and 2D Dataframe)\ numpu (Arrays and matrices)\ scipy (Optimization and solving diffrential equation)\ Matplotlib (Plots and graphs and figures)\ Seaborn (Heat maps and time series and oter plots)

We will discuss it later\

## Scikit Learn

> ML, Regression, classification, clustring analysis etc

## Stats models

> Exploredata and estimation of statistical modelsand perform statistical analysis

# Statistics

Statisctis is a collection of methods and collecting, displuing, analyzing and drawing conclusion from data.

# Statistics is everywhere

Incom of Pak avg (**Average**)

Highest score in PSL (**MAX**)

**Fastest** Bowler

**Lowest** runs

Female **Percentage** of teacher in pakistan

Rain forcast **(Likelihood)**

Dollar range **Variance**

Hostel male are more exepensive than female in (**t_test**)

Best from this place in terms of jokes, culture etc is **ANOVA**

# Type of Data

## Type-1 Data

1. Cross Sectional (Collect at one point aj kitny log video dek rahy hy? kind of)
2. Time Series (Data Collected over different TIme points e.g covid)

## Type-2 Data

1. Univariate (Data contain a single variable to measure entity e.g plnat hight in time stamp e.g kitna khana katy hy jis say wazan bhar raha hy)
2. Multi Variate (Data contain more variable to measure something e.g plant hight, fertilizer amount irrigation)

## Variable Types-1

### categorical (Nominal)

> - Binominal (True/False) no quantitative relationship is given
> - Multinomonal (Travel Choices) e.g hue col in sns

## Variable Types-2

### categorical (Nominal)

> - Ordinal Variable Data ranked or ordered e.g mery pas kitnyu phone hy? no fix limit size etc ranking in simple word and you have to search your own

## Variable Types-3

### Ratio Data

> - Data have a natral zero e.g aj zaida mall bika hy kal ki nisbat and economic data
> - or measurement in unit and ratios are continous note: You will not mix with catogorical varible e.g kal sale nae hoe aj hoe hy so don't mix it with continous ratio data is meri hieht tm say zaida hy

## Variable Types-3

### Interval data or Variables Data

> - Ordered and charaterized data e.g is june may grami zaida hy? 2020 ki nisbat

Rarios are meaning less (50 degree is not double hot of 25 degere diffrence are meaningful)

we can not say about diffrence between double or triple you can google it for more ifo

## Meaure of Central Tendency

### Mean, Meadian and Mode

> Population vs Sample
>
> - Population research has more power (less error chances covid vaccine e.g)
> - Samples are used to reduced the cost of data collection (less accurate less powerfull )

## Notion and Terms in Statistics

> N = size of population n = size of sample E = sum

## Notion and Terms in Statistics

> measurement (chaces of servival)\ sample\ parameter (no and summarize of the poplation)\ statistic (mean modian and mode)\ Desciptive Statistics (describing analysis of data)\ Inferential Statistics (darwing conclusion about a population based on info contained in a sample taken from that \poplation) Qualititve Data (measurement for which there is no natural numerical scale but which conssit of attribute that arise from a natural numerical scale)\ Quantitive Data (numerical measurement that arise from a natural numerical scale\ Mean (is the sim divied by the no of observation average)\ Meaninful (for inteval and ratio data) Outliers (Change the means of a data therefore median is usefull)\ Median (is middl eno of any sorted ascending or descending order of list)\ (describing analysis of data)\ Mode (The value that occures most frewuently e.g 18 year age most common in a class) (describing analysis of data)\

In [ ]:

```
df = sns.load_dataset('iris')
df.describe()
```

| | sepal_length | sepal_width | petal_length | petal_width |
|---|---|---|---|---|
| **count** | 150.000000 | 150.000000 | 150.000000 | 150.000000 |
| **mean** | 5.843333 | 3.057333 | 3.758000 | 1.199333 |
| **std** | 0.828066 | 0.435866 | 1.765298 | 0.762238 |
| **min** | 4.300000 | 2.000000 | 1.000000 | 0.100000 |
| **25%** | 5.100000 | 2.800000 | 1.600000 | 0.300000 |
| **50%** | 5.800000 | 3.000000 | 4.350000 | 1.300000 |
| **75%** | 6.400000 | 3.300000 | 5.100000 | 1.800000 |
| **max** | 7.900000 | 4.400000 | 6.900000 | 2.500000 |

## Measure of Despersion

How much data spread around mean of the data. (the dispurstion is called standard deviation or standard error or varinace or bell curve)

e.g range = min -to- max 123456789123456789 (1 is min last 9 is max)

## SD and Mean

Data from 100 shopes is ramdan first calculate mean of 2 karhye abd biryani mean 1= 30.1 mean 2= 30.2

sd = 4.47 sd = 10.99

the central tendency will be mean and disperse SD checking in a data

# SD and Mean Reliablity

Mean give us small picture

Means are incomplete withour dispersion (SD)

Mean with a SD is more usefull than only mean by itself

## Fundamental of Visuliztion

### Variable Type Matters

Type of visulalization depends on the variable type

1. Catergorical Var
    - Count plot type

- Qualitative variable
- Male vs female
- T/F
- 0 vs 1
- Yes vs No

2. Contiunous Variable
    - Scater plot
    - Quantitative Var
    - Statistical Proportion
    - Means and their comparison
    - e.g amount no, age, plant height

### Chart Suggestion A thought starter from the exterme Presentation Method (Created by DR Andrew ALbela)

# Data Wrangling Notebook

> **Steps**
>
> - Data collection
> - handling missing val
> - data formating
> - data normalization (scaling, centring)
> - Data binnin (for group of data)
> - making dummies of catagorical data nurmerical data
> - Clean the Data
> - Find a Relationship between data
> - analayize data
> -

In [ ]:
```
import plotly.express as px
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
df = sns.load_dataset('titanic')
df.head()
```

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck | e |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 3 | male | 22.0 | 1 | 0 | 7.2500 | S | Third | man | True | NaN | S |
| **1** | 1 | 1 | female | 38.0 | 1 | 0 | 71.2833 | C | First | woman | False | C | |
| **2** | 1 | 3 | female | 26.0 | 0 | 0 | 7.9250 | S | Third | woman | False | NaN | S |

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck | e |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **3** | 1 | 1 | female | 35.0 | 1 | 0 | 53.1000 | S | First | woman | False | C | S |
| **4** | 0 | 3 | male | 35.0 | 0 | 0 | 8.0500 | S | Third | man | True | NaN | S |

# Dealing with missing valuse

**steps:**\ if missing or nan or no value than recollect the data\ remove missing col if more is nan\ replance with mean, mode, or based on some algorithms etc\ leave that col

In [ ]:
```
df.isnull().sum()
```

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

In [ ]:
```
# drop the missing val col
df.shape
```

```
(891, 15)
```

In [ ]:
```
dff = df
```

In [ ]:
```
dff.dropna(subset = ['deck'], axis=0, inplace=True)
```

In [ ]:
```
dff.head()
```

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 1 | female | 38.0 | 1 | 0 | 71.2833 | C | First | woman | False | C |
| **3** | 1 | 1 | female | 35.0 | 1 | 0 | 53.1000 | S | First | woman | False | C |
| **6** | 0 | 1 | male | 54.0 | 0 | 0 | 51.8625 | S | First | man | True | E |
| **10** | 1 | 3 | female | 4.0 | 1 | 1 | 16.7000 | S | Third | child | False | G |
| **11** | 1 | 1 | female | 58.0 | 0 | 0 | 26.5500 | S | First | woman | False | C |

In [ ]:
```
dff.shape
```

```
(203, 15)
```

In [ ]:
```
dff.dropna()
dff.reset_index()
dff.head()
```

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 1 | female | 38.0 | 1 | 0 | 71.2833 | C | First | woman | False | C |
| **3** | 1 | 1 | female | 35.0 | 1 | 0 | 53.1000 | S | First | woman | False | C |
| **6** | 0 | 1 | male | 54.0 | 0 | 0 | 51.8625 | S | First | man | True | E |
| **10** | 1 | 3 | female | 4.0 | 1 | 1 | 16.7000 | S | Third | child | False | G |
| **11** | 1 | 1 | female | 58.0 | 0 | 0 | 26.5500 | S | First | woman | False | C |

In [ ]:
```
dff.reset_index(False)
```

| | index | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1 | 1 | female | 38.0 | 1 | 0 | 71.2833 | C | First | woman | False |
| **1** | 3 | 1 | 1 | female | 35.0 | 1 | 0 | 53.1000 | S | First | woman | False |
| **2** | 6 | 0 | 1 | male | 54.0 | 0 | 0 | 51.8625 | S | First | man | True |
| **3** | 10 | 1 | 3 | female | 4.0 | 1 | 1 | 16.7000 | S | Third | child | False |
| **4** | 11 | 1 | 1 | female | 58.0 | 0 | 0 | 26.5500 | S | First | woman | False |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **198** | 871 | 1 | 1 | female | 47.0 | 1 | 1 | 52.5542 | S | First | woman | False |
| **199** | 872 | 0 | 1 | male | 33.0 | 0 | 0 | 5.0000 | S | First | man | True |
| **200** | 879 | 1 | 1 | female | 56.0 | 0 | 1 | 83.1583 | C | First | woman | False |
| **201** | 887 | 1 | 1 | female | 19.0 | 0 | 0 | 30.0000 | S | First | woman | False |
| **202** | 889 | 1 | 1 | male | 26.0 | 0 | 0 | 30.0000 | C | First | man | True |

203 rows × 16 columns

In [ ]:
```
#  finding mean and replace with it
mean = df['age'].mean()
df['age']  = df['age'].replace(np.nan, mean)
df.head()
```

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 1 | female | 38.0 | 1 | 0 | 71.2833 | C | First | woman | False | C |
| **3** | 1 | 1 | female | 35.0 | 1 | 0 | 53.1000 | S | First | woman | False | C |
| **6** | 0 | 1 | male | 54.0 | 0 | 0 | 51.8625 | S | First | man | True | E |
| **10** | 1 | 3 | female | 4.0 | 1 | 1 | 16.7000 | S | Third | child | False | G |
| **11** | 1 | 1 | female | 58.0 | 0 | 0 | 26.5500 | S | First | woman | False | C |

In [ ]:
```
# assignment code
# deck value replace with mean

df['deck']  = df['deck'].replace(np.nan, mean)
```

In [ ]:
```
df.head()
```

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 1 | female | 38.0 | 1 | 0 | 71.2833 | C | First | woman | False | C |
| **3** | 1 | 1 | female | 35.0 | 1 | 0 | 53.1000 | S | First | woman | False | C |
| **6** | 0 | 1 | male | 54.0 | 0 | 0 | 51.8625 | S | First | man | True | E |
| **10** | 1 | 3 | female | 4.0 | 1 | 1 | 16.7000 | S | Third | child | False | G |
| **11** | 1 | 1 | female | 58.0 | 0 | 0 | 26.5500 | S | First | woman | False | C |

# Data Formatting

> Data ko aik common standard pr lana Ensures data is consitent and undersatanble
>
> - easy to gather
> - easy to work

In [ ]:
```
df.dtypes
```

```
survived          int64
pclass            int64
sex              object
age             float64
sibsp             int64
parch             int64
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
```

```
alive            object
alone              bool
dtype: object
```

In [ ]:
```
df['survived'] = df['survived'].astype('float64')
```

In [ ]:
```
df.dtypes
```

```
survived        float64
pclass            int64
sex              object
age             float64
sibsp             int64
parch             int64
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
alive            object
alone              bool
dtype: object
```

In [ ]:
```
# ere we will convert the age into days instrad of year
df['age']= df['age']*365
# assignment to remove the zeros
df['age'] = df['age'].astype('int64')
df.dtypes
```

```
survived        float64
pclass            int64
sex              object
age               int64
sibsp             int64
parch             int64
fare            float64
embarked         object
class          category
who              object
adult_male         bool
deck           category
embark_town      object
alive            object
alone              bool
dtype: object
```

In [ ]:
```
df.head()
```

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1.0 | 1 | female | 13870 | 1 | 0 | 71.2833 | C | First | woman | False | C |
| **3** | 1.0 | 1 | female | 12775 | 1 | 0 | 53.1000 | S | First | woman | False | C |
| **6** | 0.0 | 1 | male | 19710 | 0 | 0 | 51.8625 | S | First | man | True | E |
| **10** | 1.0 | 3 | female | 1460 | 1 | 1 | 16.7000 | S | Third | child | False | G |

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **11** | 1.0 | 1 | female | 21170 | 0 | 0 | 26.5500 | S | First | woman | False | C |

In [ ]:
```
df.rename(columns={'age': "age_days"}, inplace=True)
```

# Data Normalization

> Uniform the data\ making use they have same impact\ also good for computation

In [ ]:
```
df2 = df[['age_days', 'fare']]
df2.head()
```

| | age_days | fare |
|---|---|---|
| **1** | 13870 | 71.2833 |
| **3** | 12775 | 53.1000 |
| **6** | 19710 | 51.8625 |
| **10** | 1460 | 16.7000 |
| **11** | 21170 | 26.5500 |

In [ ]:
```
# normalization
# to sacle the features like new = old/max
# min max method
# z-score
# etc
```

In [ ]:
```
df2.fare = df2.fare/df.fare.max() #simple feature scaling
```

```
C:\Users\Ali\anaconda3\envs\python-chilla\lib\site-packages\pandas\core\generic.py:5170:
SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
```

In [ ]:
```
df2
```

| | age_days | fare |
|---|---|---|
| **1** | 13870 | 0.139136 |
| **3** | 12775 | 0.103644 |

| | age_days | fare |
|---|---|---|
| **6** | 19710 | 0.101229 |
| **10** | 1460 | 0.032596 |
| **11** | 21170 | 0.051822 |
| **...** | ... | ... |
| **871** | 17155 | 0.102579 |
| **872** | 12045 | 0.009759 |
| **879** | 20440 | 0.162314 |
| **887** | 6935 | 0.058556 |
| **889** | 9490 | 0.058556 |

203 rows × 2 columns

In [ ]:
```
# min max method
df2.fare = (df2.fare-df.fare.min())/ (df.fare.max()- df.fare.min())
df2.head()
```

```
C:\Users\Ali\anaconda3\envs\python-chilla\lib\site-packages\pandas\core\generic.py:5170:
SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
```

| | age_days | fare |
|---|---|---|
| **1** | 13870 | 0.000272 |
| **3** | 12775 | 0.000202 |
| **6** | 19710 | 0.000198 |
| **10** | 1460 | 0.000064 |
| **11** | 21170 | 0.000101 |

In [ ]:
```
# Log trasnformation
df.fare = np.log(df.fare)
df.head()
```

```
C:\Users\Ali\anaconda3\envs\python-chilla\lib\site-packages\pandas\core\series.py:726: R
untimeWarning:

divide by zero encountered in log
```

| | survived | pclass | sex | age_days | sibsp | parch | fare | embarked | class | who | adult_male | c |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

| | survived | pclass | sex | age_days | sibsp | parch | fare | embarked | class | who | adult_male | c |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1.0 | 1 | female | 13870 | 1 | 0 | 4.266662 | C | First | woman | False | |
| **3** | 1.0 | 1 | female | 12775 | 1 | 0 | 3.972177 | S | First | woman | False | |
| **6** | 0.0 | 1 | male | 19710 | 0 | 0 | 3.948596 | S | First | man | True | |
| **10** | 1.0 | 3 | female | 1460 | 1 | 1 | 2.815409 | S | Third | child | False | |
| **11** | 1.0 | 1 | female | 21170 | 0 | 0 | 3.279030 | S | First | woman | False | |

# Binning

> grouping of value into smaller no of val\ convert numeric into categories (1-15)(15-30) etc\ to have better understaing\

In [ ]:

## convert data into 0 and 1

> e.g male and female into 1 and 0\ e.g yes or no into 1 and 0

In [ ]:
```
pd.get_dummies(df['sex'])
```

| | female | male |
|---|---|---|
| **1** | 1 | 0 |
| **3** | 1 | 0 |
| **6** | 0 | 1 |
| **10** | 1 | 0 |
| **11** | 1 | 0 |
| **...** | ... | ... |
| **871** | 1 | 0 |
| **872** | 0 | 1 |
| **879** | 1 | 0 |
| **887** | 1 | 0 |
| **889** | 0 | 1 |

203 rows × 2 columns

In [ ]:
```
# two ways
# df_gender = pd.get_dummies(df['sex'])
```

```
# df_new = pd.concat([df, df_gender], axis=1)
df = sns.load_dataset('titanic')
df['sex'] = df['sex'].map({'male': 1, 'female': 0})

df.head()
```

| | survived | pclass | sex | age | sibsp | parch | fare | embarked | class | who | adult_male | deck | emb |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 3 | 1 | 22.0 | 1 | 0 | 7.2500 | S | Third | man | True | NaN | Sou |
| **1** | 1 | 1 | 0 | 38.0 | 1 | 0 | 71.2833 | C | First | woman | False | C | C |
| **2** | 1 | 3 | 0 | 26.0 | 0 | 0 | 7.9250 | S | Third | woman | False | NaN | Sou |
| **3** | 1 | 1 | 0 | 35.0 | 1 | 0 | 53.1000 | S | First | woman | False | C | Sou |
| **4** | 0 | 3 | 1 | 35.0 | 0 | 0 | 8.0500 | S | Third | man | True | NaN | Sou |

In [ ]:

In [ ]: