BLOCKCHAIN TECHNOLOGY THEORY & SOLIDITY FOR SMART CONTRACTS

A CERTIFICATE COURSE CONDUCTED

BY

THE SURE TRUST

Skill Upgradation for Rural-youth Empowerment – TRUST

(www.suretrustforruralyouth.com)

COURSE TRAINING ATTENDED

BY

Dubbaka Srikanth

JULY - December 2020

Declaration

This is to certify that Mr. Dubbaka Srikanth has successfully completed the four months training in “Python & Machine Learning – Basic Applications” course conducted by SURE Trust during the period July 2022 - December 2022.

By

Mr. Jagadeesh Gajula
Senior Engineer at Atsuya Technologies
Trainer in SURE Trust

Prof. Radhakumari
Vandana Nagesh
Executive Director & Founder
SURE Trust

Mrs.

Director & Founder
SURE Trust

TABLE OF CONTENTS

1. Introduction to the SURE TRUST

Introduction to The SURE TRUST:

The SURE TRUST is born to enhance the employability of educated unemployed rural youth. It is observed that there is a wide gap between the skills acquired by students from the academic institutions and the skills required by the industry to employ them. Employability enhancement is done through giving one on one training in emerging technologies, completely through online mode. The mission of the SURE TRUST is to bridge the gap between the skills acquired and the skills required by training them in the most emerging technologies such as Artificial Intelligence (AI), Python Program, Machine Learning (ML), Deep Learning (DL), Data Science & Data Analytics, Blockchain Technology, Robotic Process Automation (RPA), Project Management, Excel for Business Application, Statistical tools & Applications, Spoken English and Business Communication etc., that will enhance their employability. After completion of four months training in the course, the trainees will get live projects from industries as internship activity to get experience in applying to real time situation what they have learnt during the course. These projects will give them hands on experience which is much sought after by the prospective industry employing them. Currently students from all over India are enrolling for various courses offered by the SURE TRUST. The SURE TRUST offers every course free of cost with no financial burden of any kind to students. This initiative is purely a service-oriented one aiming to guide the rural youth who are educated but unemployed due to lack of upgradation in their skill sets. The birth of SURE TRUST is a God given boon to rural youth who could reach great heights either in employment or in entrepreneurship once they receive the training offered followed by the company internship. Many companies are coming forward to join their hands with us by offering internship projects to hand hold and lead the rural youth in their career settlement.

Vision of the SURE TRUST:

The vision of the SURE TRUST is to enhance the employability of educated unemployed youth, particularly living in rural areas, through skill upgradation, with no cost to the students.

Mission of the SURE TRUST:

The mission is to bridge the gap between the skills acquired in the academic institutions and the skills required in industries as a pre-condition for employment.

Functioning of the SURE TRUST:

There are three dedicated, committed, and hard-working women on the board of management of the SURE TRUST who will look into the various administrative and other matters relating to the enrolment of students, organizing trainers, entering into agreements with companies for getting live projects to students as internship programs, and so on. All the three women on the board are all the alumni from Sri Sathya Sai Institute of Higher Learning, Anantapur Campus, deemed to be a University. The women board is supported by five eminent advisories who are from different walks of life and have made outstanding mark in career in their respective fields. For more details about SURE TRUST please visit the website www.suretrustforruralyouth.com

2. Course Content:

The SURE TRUST conducts a four months training for every course on a uniform basis. A session spanning across one to one & half hour is taken by the trainers for every major course. Sessions are conducted to complete the predesigned course structure within the fixed time period. Course content is designed to suit the current requirement of the Industry and validated by the industry experts. The course content of all these courses is so dynamic that any changed condition noticed in the industry will automatically get reflected in the content of the respective course. As the course content is dynamic, the Following is the course content of the current course in Python and machine learning:

Python and machine learning

Objective**:**

- To train engineering, MBA; and other professional course graduates to become full-fledged smart contract developers and blockchain creators for direct recruitment by the industry.

Course Content:

1. Python for data science and machine learning • Python basics • Python OOPs
2. Data analysis • Pandas library • Numpy library
3. Data visualization • Matplotlib • Seaborn

4. Exploratory data analysis • Univariate • Bivariate • Derived metrics • Introduction to databases (MySQL for Data science students
5. Math for machine learning • Statistics • Linear algebra • Calculus • Probability theory
6. Supervised learning • Feature selection • Regression • Linear regression • Polynomial regression • Advanced regression introduction • Classification • Logistic regression • Naive Bayes • Support vector machine • Tree models
7. Advanced regression • Regularized regression • Ridge and Lasso regression • Model selection and Grid Search methods
8. Unsupervised learning • K means clustering • Hierarchical clustering • Principal component analysis
9. Theory introduction to deep learning and reinforcement learning • Theory about ANN, CNN, RNN, LSTM • Theory about Markov Decision process
10. Writing assignments, tests and preparing the course report to take along, what is learnt during the course.

****

3.Conduct of the course:

a) Modalities for the conduct of all the courses are fixed by the SURE TRUST which are u uniformly followed across the courses.

- Mode of Training --- Online
- Period of Training --- Four months
- Sessions per week --- 3 to 6
- Length of the session --- 1 to 2 hours
- Tests to be taken --- 2 per month
- Assignments --- 2 per month
- Last 15 days --- Final practice and preparing the course report

b) Student byelaws:

Students enrolling for the courses under SURE TRUST are strictly required to follow the following byelaws set for them:

Byelaws for students to become eligible for certificate at the end of the course:

I. Minimum Attendance:

Every student must put in a minimum of 85% attendance in attending the classes for getting the eligibility to receive the certificates.

II.. Two written tests are to be taken in each month:

Since the objective of the certification program is to turn out well qualified students from the respective courses, minimum two written tests are to be taken in each month for each course to ensure that the students are pulled along the expected line of standard.

III. Assignment submissions:

Ten exercises constituting one assignment for every two to three new functions/topics taught, resulting in minimum seven such assignments are to be submitted during the four months period.

IV. Preparing the final course report in the prescribed format:

During the last fifteen days in the fourth month, students many be asked to consolidate and compile all the assignments submitted in a word document along with the other chapters which will constitute a course report for each student. This report will be the unique contribution a student carries from the trust to show case the rigorous training he/she received during the four months period. Besides the report will stand as a testimony for the detailed learning a student has acquired in the chosen area. Thiswill facilitate the industry in handpicking the required student for the job.

V. External Viva-voce:

Every student has to successfully clear the external viva-voce arranged in their respective course.VI.

KYC norms:

Each student wishing to enrol for the course must submit a written letter saying that he/she will not dropfrom the course until its completion, which will also be signed by father / mother besides the student himself / herself.

VII. Attend the full class:

All the students are expected to attend each class for full duration. Some students are observed movingout of classes after logging in which does not go well with the learning objective of students.

VIII. Ensure discipline in the group:

All the students are advised strictly to follow group etiquette and restrain from posting in the group anyunethical messages or teasing messages or personal interactive messages. This group is purely createdfor academic purpose and hence only academic interactions should go on

Python and Machine Learning:

The course is taught to the students as per the syllabus prescribed and as per the course modalities keeping in mind the bylaws set for them. Periodical tests are conducted to assess the understanding of the students in the course. Assignments are given to ensure that students gain versality in the solidity program and development of smart contracts. The assignments of the student are given below. These assignments which are done with high creativity and out of box thinking not only reflect the student's innovativeness but also constitute solved exercises in solidity for the fresh learners to practice and gainconfidence. The assignments are listed below in the order of their conducting during the course:

Assignment-1

write a program to take input from the user and print prime if its a primenumber else print non-prime number?

```
primenumber=int(input("enter your number: "))

if primenumber==2 :
    print(primenumber, "is a primenumber")
elif primenumber==3:
    print(primenumber, "is a primenumber")
elif primenumber==5:
    print(primenumber, "is a primenumber")
elif primenumber==7:
    print(primenumber, "is a primenumber")
elif primenumber%2==0:
    print(primenumber, "is not a primenumber")
elif primenumber%3==0:
    print(primenumber, "is not a primenumber")
elif primenumber%5==0:
    print(primenumber, "is not a primenumber")
elif primenumber%7==0:
    print(primenumber, "is not a primenumber")
else:
    print(primenumber, "is a primenumber")
```

or

```
 primenumber=int(input("enter your number: "))
flag=False
if primenumber > 1:
    for i in range (2,primenumber):
        if (primenumber%i)==0:
            flag=True
            break
if flag:
    print(primenumber,"is not a prime number")
else:
    print(primenumber,"is a prime number")
```

1) write a program to remove odd number from a input list? (create a list of numbers)

```
numlist=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
for i in numlist:
    if i%2!=0:
        numlist.remove(i)

print (numlist)
```

3 write a program to print first 100 Fibonacci series?

```
num=int(input("enter your positive number: "))
n1 , n2 = 0 , 1
x=0
for i in range(0,100):
```

```
    n1=n2
    n2=x
    x=n1+n2
    print(x)
```

4 write a function which takes input and returns True if input is integer else returns False?

```
num = 0
while True:
    try:
        num = int ( input ("Enter your number: "))
    except ValueError:
        print(num,"is not an integer")
        continue
    else:
        print(num, "is an integer")
        break
```

5 write a function to reverse given list?

```
num=int(input("enter your positive number: "))
n1 , n2 = 0 , 1
x=0
for i in range(0,100):
    n1=n2
    n2=x
    x=n1+n2
    print(x)
```

6 write a program to take 5 numerical inputs from the user and print the largest number

```
numlist=[1,2,3,4,5,6,7,8,9,10]
print(max(numlist))
```

7 write a program to take input and write that input into a txt file

```
txtfile=open("srikanth.txt","w")
txtfile.write("Hi, Im srikanth.Thank you for opening my file ")
txtfile.close()
```

8 write a program to take radius from the user and print area of circle

```
r = float(input ("enter the radius of circle : "))

calculateArea = str(3.14159 * r**2);

print (calculateArea, "is the area of circle " )
```

9 write a program to demonstrate exception handling (any example)

```
def func(x,y):
    try:
        return x/2+y/2
    except Exception as error:
        print(error)

x=int(input("please enter your first number: "))
y=int(input("please enter your second number: "))

print(func(x,y))
```

10 write a program to take input from the user and print only vowels in the input

```
word = input("Enter your word: ")

for char in word:
    if char.lower() in 'aeiou' and char.upper() in 'AEIOU':
        print(char)
```

What is your Name and E-mail

NAME: DUBBAKA SRIKANTH
EMAIL: srikanthdubbaka123@gmail.com

```
# create numpy in 5 builtin methods.
#import numpy
import numpy as np
#created list
List=[0,1,2,3,4,5,6,7,8,9,10]
print("1.The list is:")
print(List)

#array of list
array= np.array(List)
print("2.array of list is:")
print(array)

#dtype=float
arrayis = np.array(List,dtype="float")
print("3.dtype =float--array is:")
print(arrayis)

#array-dtype
print("4.data type of array is:")
print(array.dtype)

#list is in dtype=int
arrayis = np.array(List,dtype="int")
print("5.The array of dtype=int is:")
print(arrayis)

#find array of alpha
a=["a","b","c"]
b = np.array(a)
print("6.The array of alphabets(a) is:")
print(b)
print("7.data type of array of alphabets is:")
print(b.dtype)
```

```
1. The list is:
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2. array of list is:
[ 0  1  2  3  4  5  6  7  8  9 10]
3.dtype =float--array is:
[ 0.  1.  2.  3.  4.  5.  6.  7.  8.  9. 10.]
4. data type of array is:
```

```
int32
5. The array of dtype=int is:
[ 0  1  2  3  4  5  6  7  8  9 10]
6. The array of alphabets(a) is:
['a' 'b' 'c']
7. data type of array of alphabets is:
<U1
```

```
# perform 4 operations on matrices
import numpy as np
# define a Python matrix with data elements in it
list = [[ 'sri', 1, 2, 3],
        ["raj", 4, 5, 6],
        [7, "raghu", 8, 9]]
print("1.Python matrix with data elements in list is:")
print(list)
# define the length of the list

print("2.length of list:")
print(len(list))

print("3.add two matrices using Numpy")
a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])
c = np.add(a, b)
print(c)
print("4.subtract two matrices using Numpy")
a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])
c = np.subtract(a, b)
print(c)
```

```
1. Python matrix with data elements in list is:
[['sri', 1, 2, 3], ['raj', 4, 5, 6], [7, 'raghu', 8, 9]]
2. length of list:
3
3. add two matrices using Numpy
[[ 6  8]
 [10 12]]
4. subtract two matrices using Numpy
[[-4 -4]
 [-4 -4]]
```

```
#Inverse of matrix
import numpy as np

a = np.array([[1, 2, 3], [1, 3, 3], [1, 2, 4]])
```

```
invofa= np.linalg.inv(a)

print(invofa)
```

```
[[ 6. -2. -3.]
 [-1.  1.  0.]
 [-1.  0.  1.]]
```

```
# write a custom operation function and verctorize it.
# be careful while intializing applying on big matrices
import numpy as np
def func1(a, b):
    if a > b:
        return a - b
    else:
        return a + b
vfun = np.vectorize(func1)
x=vfun([1, 2, 3, 4], 5)
print(x)
```

```
[6 7 8 9]
```

```
# import numpy and pandas
import pandas as pd
import numpy as np

# import data into dataframes
company = pd.read_csv(r"G:\company.csv",delimiter="\t")
rounds = pd.read_csv(r"G:\rounds2.csv",encoding="iso-8859-1")
# we imported two files rounds and company. we have 2 dataframes

# convert rounds dataframe into numpy array
company.to_numpy()
```

```
array([["/Organization/-Fame", "#fame", "http://livfame.com", ...,
        "Mumbai", "Mumbai", nan],
       ["/Organization/-Qounter", ":Qounter",
"http://www.qounter.com",
        ..., "DE - Other", "Delaware City", "04-09-2014"],
       ["/Organization/-The-One-Of-Them-Inc-", "(THE) ONE of
THEM,Inc.",
        "http://oneofthem.jp", ..., nan, nan, nan],
       ...,
       ["/Organization/Ã\x81Eron", "Ã\x81ERON",
"http://www.aeron.hu/",
        ..., nan, nan, "01-01-2011"],
       ["/Organization/Ã”Asys-2", "Ã”asys",
"http://www.oasys.io/", ...,
        "SF Bay Area", "San Francisco", "01-01-2014"],
       ["/Organization/Ä°Novatiff-Reklam-Ve-Tanä±Tä±M-Hizmetleri-Tic",
        "Ä°novatiff Reklam ve TanÄ±tÄ±m Hizmetleri Tic",
        "http://inovatiff.com", ..., nan, nan, nan]], dtype=object)
```

```
# convert rounds dataframe into numpy array
rounds.to_numpy()
```

```
array([["/organization/-fame",
        "/funding-round/9a01d05418af9f794eebff7ace91f638", "venture",
        "B", "05-01-2015", 10000000.0],
       ["/ORGANIZATION/-QOUNTER",
        "/funding-round/22dacff496eb7acb2b901dec1dfe5633", "venture",
        "A", "14-10-2014", nan],
       ["/organization/-qounter",
        "/funding-round/b44fbb94153f6cdef13083530bb48030", "seed",
nan,
        "01-03-2014", 700000.0],
       ...,
       ["/organization/ã\x81eron",
        "/funding-round/59f4dce44723b794f21ded3daed6e4fe", "venture",
        "A", "01-08-2014", nan],
       ["/ORGANIZATION/Ã\x94ASYS-2",
        "/funding-round/35f09d0794651719b02bbfd859ba9ff5", "seed",
nan,
        "01-01-2015", 18192.0],
       ["/organization/ä°novatiff-reklam-ve-tanä±tä±m-hizmetleri-tic",
        "/funding-round/af942869878d2cd788ef5189b435ebc4", "grant",
nan,
        "01-10-2013", 14851.0]], dtype=object)
```

print basic imformation like shape,info(company)

```
company
```

```
                                              permalink  \
0                                    /Organization/-Fame
1                                  /Organization/-Qounter
2                       /Organization/-The-One-Of-Them-Inc-
3                                  /Organization/0-6-Com
4                            /Organization/004-Technologies
...                                                   ...
66363   /Organization/Zznode-Science-And-Technology-Co...
66364                            /Organization/Zzzzapp-Com
66365                                 /Organization/ÃEron
66366                               /Organization/Ã”Asys-2
66367  /Organization/Ä°Novatiff-Reklam-Ve-Tanä±Tä±M-H...

                                                    name  \
0                                                  #fame
1                                               :Qounter
2                                  (THE) ONE of THEM,Inc.
3                                                0-6.com
4                                     004 Technologies
...                                                  ...
66363                        ZZNode Science and Technology
```

```
66364                          Zzzzapp Wireless ltd.
66365                                          ÃERON
66366                                          Ã”asys
66367  Ä°novatiff Reklam ve TanÄ±tÄ±m Hizmetleri Tic

                            homepage_url  \
0                      http://livfame.com
1                  http://www.qounter.com
2                     http://oneofthem.jp
3                      http://www.0-6.com
4       http://004gmbh.de/en/004-interact
...                                   ...
66363               http://www.zznode.com
66364              http://www.zzzzapp.com
66365                http://www.aeron.hu/
66366                http://www.oasys.io/
66367                http://inovatiff.com

                                        category_list     status  \
0                                               Media  operating
1      Application Platforms|Real Time|Social Network...  operating
2                                 Apps|Games|Mobile  operating
3                                       Curated Web  operating
4                                          Software  operating
...                                             ...        ...
66363                          Enterprise Software  operating
66364        Advertising|Mobile|Web Development|Wireless  operating
66365                                           NaN  operating
66366  Consumer Electronics|Internet of Things|Teleco...  operating
66367                  Consumer Goods|E-Commerce|Internet  operating

      country_code state_code                region           city  \
0              IND         16                Mumbai         Mumbai
1              USA         DE            DE - Other  Delaware City
2              NaN        NaN                   NaN            NaN
3              CHN         22               Beijing        Beijing
4              USA         IL  Springfield, Illinois      Champaign
...            ...        ...                   ...            ...
66363          CHN         22               Beijing        Beijing
66364          HRV         15                 Split          Split
66365          NaN        NaN                   NaN            NaN
66366          USA         CA          SF Bay Area  San Francisco
66367          NaN        NaN                   NaN            NaN

      founded_at
0            NaN
1     04-09-2014
2            NaN
3     01-01-2007
4     01-01-2010
```

```
...            ...
66363          NaN
66364   13-05-2012
66365   01-01-2011
66366   01-01-2014
66367          NaN

[66368 rows x 10 columns]
```

```
#print basic imformation like shape,info
company.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 66368 entries, 0 to 66367
Data columns (total 10 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   permalink      66368 non-null  object
 1   name           66367 non-null  object
 2   homepage_url   61310 non-null  object
 3   category_list  63220 non-null  object
 4   status         66368 non-null  object
 5   country_code   59410 non-null  object
 6   state_code     57821 non-null  object
 7   region         58338 non-null  object
 8   city           58340 non-null  object
 9   founded_at     51147 non-null  object
dtypes: object(10)
memory usage: 5.1+ MB
```

```
# print basic imformation like shape,info
company.describe
```

```
<bound method NDFrame.describe of
permalink  \
0                                        /Organization/-Fame
1                                   /Organization/-Qounter
2                      /Organization/-The-One-Of-Them-Inc-
3                                    /Organization/0-6-Com
4                          /Organization/004-Technologies
...                                                   ...
66363  /Organization/Zznode-Science-And-Technology-Co...
66364                        /Organization/Zzzzapp-Com
66365                             /Organization/ÃEron
66366                           /Organization/Ã”Asys-2
66367  /Organization/Ä°Novatiff-Reklam-Ve-Tanä±Tä±M-H...

                                                  name  \
0                                                #fame
1                                             :Qounter
2                                (THE) ONE of THEM,Inc.
```

```
3                                        0-6.com
4                                  004 Technologies
...                                              ...
66363                  ZZNode Science and Technology
66364                          Zzzzapp Wireless ltd.
66365                                        ÃERON
66366                                        Ã”asys
66367  Ä°novatiff Reklam ve TanÄ±tÄ±m Hizmetleri Tic

                             homepage_url  \
0                     http://livfame.com
1                  http://www.qounter.com
2                     http://oneofthem.jp
3                      http://www.0-6.com
4       http://004gmbh.de/en/004-interact
...                                   ...
66363                http://www.zznode.com
66364               http://www.zzzzapp.com
66365                 http://www.aeron.hu/
66366                 http://www.oasys.io/
66367                 http://inovatiff.com

                                        category_list     status  \
0                                               Media  operating
1      Application Platforms|Real Time|Social Network...  operating
2                                  Apps|Games|Mobile  operating
3                                        Curated Web  operating
4                                           Software  operating
...                                              ...        ...
66363                          Enterprise Software  operating
66364       Advertising|Mobile|Web Development|Wireless  operating
66365                                             NaN  operating
66366  Consumer Electronics|Internet of Things|Teleco...  operating
66367                Consumer Goods|E-Commerce|Internet  operating

      country_code state_code              region          city  \
0              IND         16              Mumbai        Mumbai
1              USA         DE          DE - Other  Delaware City
2              NaN        NaN                 NaN           NaN
3              CHN         22             Beijing       Beijing
4              USA         IL  Springfield, Illinois      Champaign
...            ...        ...                 ...           ...
66363          CHN         22             Beijing       Beijing
66364          HRV         15               Split         Split
66365          NaN        NaN                 NaN           NaN
66366          USA         CA         SF Bay Area  San Francisco
66367          NaN        NaN                 NaN           NaN

      founded_at
0            NaN
```

```
1       04-09-2014
2              NaN
3       01-01-2007
4       01-01-2010
...            ...
66363          NaN
66364   13-05-2012
66365   01-01-2011
66366   01-01-2014
66367          NaN

[66368 rows x 10 columns]>
```

```
# print basic imformation like shape,info
company.shape
```

```
(66368, 10)
```

print basic imformation like shape,info(Rounds)

```
rounds
```

```
                                                  permalink  \
0                                      /organization/-fame
1                                    /ORGAN▯ZAT▯ON/-QOUNTER
2                                   /organization/-qounter
3                         /ORGAN▯ZAT▯ON/-THE-ONE-OF-THEM-▯NC-
4                                      /organization/0-6-com
...                                                      ...
114944                          /organization/zzzzapp-com
114945                          /ORGAN▯ZAT▯ON/ZZZZAPP-COM
114946                               /organization/ã▯eron
114947                             /ORGAN▯ZAT▯ON/Ã”ASYS-2
114948   /organization/ä°novatiff-reklam-ve-tanä±tä±m-h...

                                    funding_round_permalink
funding_round_type  \
0       /funding-round/9a01d05418af9f794eebff7ace91f638
venture
1       /funding-round/22dacff496eb7acb2b901dec1dfe5633
venture
2       /funding-round/b44fbb94153f6cdef13083530bb48030
seed
3       /funding-round/650b8f704416801069bb178a1418776b
venture
4       /funding-round/5727accaeaa57461bd22a9bdd945382d
venture
...                                                    ...
...
114944  /funding-round/8f6d25b8ee4199e586484d817bceda05
```

```
convertible_note
114945  /funding-round/ff1aa06ed5da186c84f101549035d4ae
seed
114946  /funding-round/59f4dce44723b794f21ded3daed6e4fe
venture
114947  /funding-round/35f09d0794651719b02bbfd859ba9ff5
seed
114948  /funding-round/af942869878d2cd788ef5189b435ebc4
grant

        funding_round_code   funded_at  raised_amount_usd
0                        B  05-01-2015         10000000.0
1                        A  14-10-2014                NaN
2                      NaN  01-03-2014           700000.0
3                        B  30-01-2014          3406878.0
4                        A  19-03-2008          2000000.0
...                    ...         ...                ...
114944                 NaN  01-03-2014            41313.0
114945                 NaN  01-05-2013            32842.0
114946                   A  01-08-2014                NaN
114947                 NaN  01-01-2015            18192.0
114948                 NaN  01-10-2013            14851.0

[114949 rows x 6 columns]

rounds.describe

<bound method NDFrame.describe of
permalink  \
0                                      /organization/-fame
1                                   /ORGANIZATION/-QOUNTER
2                                   /organization/-qounter
3                        /ORGANIZATION/-THE-ONE-OF-THEM-INC-
4                                    /organization/0-6-com
...                                                    ...
114944                            /organization/zzzzapp-com
114945                            /ORGANIZATION/ZZZZAPP-COM
114946                                 /organization/ãeron
114947                               /ORGANIZATION/Ã”ASYS-2
114948  /organization/ä°novatiff-reklam-ve-tanä±tä±m-h...

                                  funding_round_permalink
funding_round_type  \
0       /funding-round/9a01d05418af9f794eebff7ace91f638
venture
1       /funding-round/22dacff496eb7acb2b901dec1dfe5633
venture
2       /funding-round/b44fbb94153f6cdef13083530bb48030
seed
3       /funding-round/650b8f704416801069bb178a1418776b
```

```
venture
4        /funding-round/5727accaeaa57461bd22a9bdd945382d
venture
...                                                    ...
...
114944   /funding-round/8f6d25b8ee4199e586484d817bceda05
convertible_note
114945   /funding-round/ff1aa06ed5da186c84f101549035d4ae
seed
114946   /funding-round/59f4dce44723b794f21ded3daed6e4fe
venture
114947   /funding-round/35f09d0794651719b02bbfd859ba9ff5
seed
114948   /funding-round/af942869878d2cd788ef5189b435ebc4
grant

       funding_round_code   funded_at  raised_amount_usd
0                       B  05-01-2015         10000000.0
1                       A  14-10-2014                NaN
2                     NaN  01-03-2014           700000.0
3                       B  30-01-2014          3406878.0
4                       A  19-03-2008          2000000.0
...                   ...         ...                ...
114944                NaN  01-03-2014            41313.0
114945                NaN  01-05-2013            32842.0
114946                  A  01-08-2014                NaN
114947                NaN  01-01-2015            18192.0
114948                NaN  01-10-2013            14851.0

[114949 rows x 6 columns]>

rounds.info

<bound method DataFrame.info of
permalink  \
0                                      /organization/-fame
1                                  /ORGANIZATION/-QOUNTER
2                                  /organization/-qounter
3                       /ORGANIZATION/-THE-ONE-OF-THEM-INC-
4                                   /organization/0-6-com
...                                                    ...
114944                           /organization/zzzzapp-com
114945                           /ORGANIZATION/ZZZZAPP-COM
114946                                /organization/ãeron
114947                             /ORGANIZATION/Ã"ASYS-2
114948   /organization/ä°novatiff-reklam-ve-tanä±tä±m-h...

                                  funding_round_permalink
funding_round_type  \
0        /funding-round/9a01d05418af9f794eebff7ace91f638
```

```
venture
1         /funding-round/22dacff496eb7acb2b901dec1dfe5633
venture
2         /funding-round/b44fbb94153f6cdef13083530bb48030
seed
3         /funding-round/650b8f704416801069bb178a1418776b
venture
4         /funding-round/5727accaeaa57461bd22a9bdd945382d
venture
...                                                     ...
...
114944    /funding-round/8f6d25b8ee4199e586484d817bceda05
convertible_note
114945    /funding-round/ff1aa06ed5da186c84f101549035d4ae
seed
114946    /funding-round/59f4dce44723b794f21ded3daed6e4fe
venture
114947    /funding-round/35f09d0794651719b02bbfd859ba9ff5
seed
114948    /funding-round/af942869878d2cd788ef5189b435ebc4
grant

        funding_round_code   funded_at  raised_amount_usd
0                        B  05-01-2015         10000000.0
1                        A  14-10-2014                NaN
2                      NaN  01-03-2014           700000.0
3                        B  30-01-2014          3406878.0
4                        A  19-03-2008          2000000.0
...                    ...         ...                ...
114944                 NaN  01-03-2014            41313.0
114945                 NaN  01-05-2013            32842.0
114946                   A  01-08-2014                NaN
114947                 NaN  01-01-2015            18192.0
114948                 NaN  01-10-2013            14851.0

[114949 rows x 6 columns]>

rounds.shape

(114949, 6)

#convert all the permalink column values into lowercase in both
dataframes
x=company.permalink.str.lower()
x

0                                        /organization/-fame
1                                   /organization/-qounter
2                        /organization/-the-one-of-them-inc-
3                                    /organization/0-6-com
4                          /organization/004-technologies
```

```
                                    ...
66363    /organization/zznode-science-and-technology-co...
66364                                  /organization/zzzzapp-com
66365                                       /organization/ãeron
66366                                    /organization/ã”asys-2
66367    /organization/ä°novatiff-reklam-ve-tanä±tä±m-h...
Name: permalink, Length: 66368, dtype: object
```

```
#convert all the permalink column values into lowercase in both
dataframes
y=rounds.permalink.str.lower()
y
```

```
0                                          /organization/-fame
1                                        /organization/-qounter
2                                        /organization/-qounter
3                          /organization/-the-one-of-them-inc-
4                                         /organization/0-6-com
                                    ...
114944                                   /organization/zzzzapp-com
114945                                   /organization/zzzzapp-com
114946                                        /organization/ãeron
114947                                     /organization/ã”asys-2
114948     /organization/ä°novatiff-reklam-ve-tanä±tä±m-h...
Name: permalink, Length: 114949, dtype: object
```

```
#check how many unique values in permalink columns in both the
dataframes
unique_values_company= company.nunique()
unique_values_company
```

```
permalink         66368
name              66102
homepage_url      61191
category_list     27296
status                4
country_code        137
state_code          311
region             1092
city               5111
founded_at         3978
dtype: int64
```

```
#check how many unique values in permalink columns in both the
dataframes
unique_values_rounds= company.nunique()
unique_values_rounds
```

```
permalink         66368
name              66102
homepage_url      61191
```

```
category_list     27296
status                4
country_code        137
state_code          311
region             1092
city               5111
founded_at         3978
dtype: int64
```

```
# merge two dataframes into one master frame (on permalink)
add_company_rounds=[company,rounds]
add_company_rounds
```

```
[                                                  permalink  \
 0                                        /Organization/-Fame
 1                                    /Organization/-Qounter
 2                        /Organization/-The-One-Of-Them-Inc-
 3                                     /Organization/0-6-Com
 4                           /Organization/004-Technologies
 ...                                                      ...
 66363  /Organization/Zznode-Science-And-Technology-Co...
 66364                          /Organization/Zzzzapp-Com
 66365                               /Organization/ÃEron
 66366                             /Organization/Ã”Asys-2
 66367  /Organization/Ä°Novatiff-Reklam-Ve-Tanä±Tä±M-H...

                                                    name  \
 0                                                 #fame
 1                                             :Qounter
 2                                (THE) ONE of THEM,Inc.
 3                                                0-6.com
 4                                      004 Technologies
 ...                                                 ...
 66363                    ZZNode Science and Technology
 66364                           Zzzzapp Wireless ltd.
 66365                                           ÃERON
 66366                                           Ã”asys
 66367  Ä°novatiff Reklam ve TanÄ±tÄ±m Hizmetleri Tic

                             homepage_url  \
 0                     http://livfame.com
 1                  http://www.qounter.com
 2                     http://oneofthem.jp
 3                      http://www.0-6.com
 4       http://004gmbh.de/en/004-interact
 ...                                   ...
 66363               http://www.zznode.com
 66364              http://www.zzzzapp.com
 66365                http://www.aeron.hu/
 66366                http://www.oasys.io/
 66367                http://inovatiff.com
```

			category_list			status \
0			Media			operating
1	Application Platforms|Real Time|Social Network...					operating
2	Apps|Games|Mobile					operating
3	Curated Web					operating
4	Software					operating
...	...					...
66363	Enterprise Software					operating
66364	Advertising|Mobile|Web Development|Wireless					operating
66365	NaN					operating
66366	Consumer Electronics|Internet of Things|Teleco...					operating
66367	Consumer Goods|E-Commerce|Internet					operating

	country_code	state_code	region	city \
0	IND	16	Mumbai	Mumbai
1	USA	DE	DE - Other	Delaware City
2	NaN	NaN	NaN	NaN
3	CHN	22	Beijing	Beijing
4	USA	IL	Springfield, Illinois	Champaign
...	...	...	...	...
66363	CHN	22	Beijing	Beijing
66364	HRV	15	Split	Split
66365	NaN	NaN	NaN	NaN
66366	USA	CA	SF Bay Area	San Francisco

```
66367            NaN          NaN                        NaN              NaN


        founded_at
0              NaN
1       04-09-2014
2              NaN
3       01-01-2007
4       01-01-2010
...            ...
66363          NaN
66364   13-05-2012
66365   01-01-2011
66366   01-01-2014
66367          NaN

[66368 rows x 10 columns],
                                                    permalink  \
0                                          /organization/-fame
1                                        /ORGANIZATION/-QOUNTER
2                                       /organization/-qounter
3                          /ORGANIZATION/-THE-ONE-OF-THEM-INC-
4                                        /organization/0-6-com
...                                                        ...
114944                                /organization/zzzzapp-com
114945                               /ORGANIZATION/ZZZZAPP-COM
114946                                    /organization/ãeron
114947                                  /ORGANIZATION/Ã"ASYS-2
114948   /organization/ä°novatiff-reklam-ve-tanä±tä±m-h...

                                    funding_round_permalink
funding_round_type  \
 0        /funding-round/9a01d05418af9f794eebff7ace91f638
venture
 1        /funding-round/22dacff496eb7acb2b901dec1dfe5633
venture
 2        /funding-round/b44fbb94153f6cdef13083530bb48030
seed
 3        /funding-round/650b8f704416801069bb178a1418776b
venture
 4        /funding-round/5727accaeaa57461bd22a9bdd945382d
venture
 ...                                                   ...
...
 114944   /funding-round/8f6d25b8ee4199e586484d817bceda05
convertible_note
 114945   /funding-round/ff1aa06ed5da186c84f101549035d4ae
seed
 114946   /funding-round/59f4dce44723b794f21ded3daed6e4fe
venture
```

```
 114947  /funding-round/35f09d0794651719b02bbfd859ba9ff5
seed
 114948  /funding-round/af942869878d2cd788ef5189b435ebc4
grant

       funding_round_code   funded_at  raised_amount_usd
 0                      B  05-01-2015         10000000.0
 1                      A  14-10-2014                NaN
 2                    NaN  01-03-2014          700000.0
 3                      B  30-01-2014         3406878.0
 4                      A  19-03-2008         2000000.0
 ...                  ...         ...                ...
 114944               NaN  01-03-2014           41313.0
 114945               NaN  01-05-2013           32842.0
 114946                 A  01-08-2014                NaN
 114947               NaN  01-01-2015           18192.0
 114948               NaN  01-10-2013           14851.0

 [114949 rows x 6 columns]]
```

```
from functools import reduce
data_merge=reduce(lambda left, right:
                  pd.merge(left,right,how="outer" ),add_company_rounds)
print(data_merge)
```

```
                                                permalink  \
0                                      /Organization/-Fame
1                                    /Organization/-Qounter
2                         /Organization/-The-One-Of-Them-Inc-
3                                      /Organization/0-6-Com
4                           /Organization/004-Technologies
...                                                    ...
181312                           /ORGANIZATION/ZZZZAPP-COM
181313                           /ORGANIZATION/ZZZZAPP-COM
181314                                /organization/ãeron
181315                             /ORGANIZATION/Ã”ASYS-2
181316  /organization/ä°novatiff-reklam-ve-tanä±tä±m-h...

                          name                          homepage_url  \
0                        #fame                    http://livfame.com
1                    :Qounter                http://www.qounter.com
2       (THE) ONE of THEM,Inc.                   http://oneofthem.jp
3                       0-6.com                    http://www.0-6.com
4             004 Technologies  http://004gmbh.de/en/004-interact
...                        ...                                  ...
181312                    NaN                                  NaN
181313                    NaN                                  NaN
181314                    NaN                                  NaN
181315                    NaN                                  NaN
181316                    NaN                                  NaN
```

	category_list	status
0	Media	operating
1	Application Platforms|Real Time|Social Network...	operating
2	Apps|Games|Mobile	operating
3	Curated Web	operating
4	Software	operating
...	...	...
181312	NaN	NaN
181313	NaN	NaN
181314	NaN	NaN
181315	NaN	NaN
181316	NaN	NaN

	country_code	state_code	region	city
0	IND	16	Mumbai	Mumbai
1	USA	DE	DE - Other	Delaware City
2	NaN	NaN	NaN	NaN
3	CHN	22	Beijing	Beijing
4	USA	IL	Springfield, Illinois	Champaign
...	...	...	...	...
181312	NaN	NaN	NaN	NaN
181313	NaN	NaN	NaN	NaN
181314	NaN	NaN	NaN	NaN
181315	NaN	NaN	NaN	NaN

```
181316          NaN          NaN                      NaN              NaN


        founded_at                                funding_round_permalink  \
0              NaN                                                    NaN
1       04-09-2014                                                    NaN
2              NaN                                                    NaN
3       01-01-2007                                                    NaN
4       01-01-2010                                                    NaN
...            ...                                                    ...
181312         NaN  /funding-round/6ba41360588bc6e3f77e9b50a0ebfafa
181313         NaN  /funding-round/ff1aa06ed5da186c84f101549035d4ae
181314         NaN  /funding-round/59f4dce44723b794f21ded3daed6e4fe
181315         NaN  /funding-round/35f09d0794651719b02bbfd859ba9ff5
181316         NaN  /funding-round/af942869878d2cd788ef5189b435ebc4

       funding_round_type funding_round_code    funded_at
raised_amount_usd
0                     NaN                 NaN          NaN
NaN
1                     NaN                 NaN          NaN
NaN
2                     NaN                 NaN          NaN
NaN
3                     NaN                 NaN          NaN
NaN
4                     NaN                 NaN          NaN
NaN
...                   ...                 ...          ...
...
181312               seed                 NaN  01-08-2013
33149.0
181313               seed                 NaN  01-05-2013
32842.0
181314            venture                   A  01-08-2014
NaN
181315               seed                 NaN  01-01-2015
18192.0
181316              grant                 NaN  01-10-2013
14851.0

[181317 rows x 15 columns]
```

masterframe will have around 114949 rows, If not please recheck the code.

ROWS: 181317 ; COLUMNS: 15

```
# check percentage of missing values in each column
company["permalink"].isnull().sum()/len(company["permalink"])
```

0.0

```
# check percentage of missing values in each column
company["name"].isnull().sum()/len(company["name"])
```

1.5067502410800386e-05

```
# check percentage of missing values in each column
company["homepage_url"].isnull().sum()/len(company["homepage_url"])
```

0.07621142719382835

```
# check percentage of missing values in each column
company["category_list"].isnull().sum()/len(company["category_list"])
```

0.04743249758919962

```
# check percentage of missing values in each column
company["status"].isnull().sum()/len(company["status"])
```

0.0

```
# check percentage of missing values in each column
company["country_code"].isnull().sum()/len(company["country_code"])
```

0.10483968177434909

```
# check percentage of missing values in each column
company["state_code"].isnull().sum()/len(company["state_code"])
```

0.1287819431051109

```
# check percentage of missing values in each column
company["region"].isnull().sum()/len(company["region"])
```

0.1209920443587271

```
# check percentage of missing values in each column
company["city"].isnull().sum()/len(company["city"])
```

0.1209619093539055

```
# check percentage of missing values in each column
company["founded_at"].isnull().sum()/len(company["founded_at"])
```

0.22934245419479268

```
# check percentage of missing values in each column
rounds["permalink"].isnull().sum()/len(rounds["permalink"])
```

0.0

```
# check percentage of missing values in each column
rounds["funding_round_permalink"].isnull().sum()/len(rounds["funding_round_permalink"])
```

0.0

```
# check percentage of missing values in each column
rounds["funding_round_type"].isnull().sum()/len(rounds["funding_round_type"])
```

0.0

```
# check percentage of missing values in each column
rounds["funding_round_code"].isnull().sum()/len(rounds["funding_round_code"])
```

0.7290972518247223

```
# check percentage of missing values in each column
rounds["funded_at"].isnull().sum()/len(rounds["funded_at"])
```

0.0

```
# check percentage of missing values in each column
rounds["raised_amount_usd"].isnull().sum()/len(rounds["raised_amount_usd"])
```

0.17390320924931926

```
# drop all the rows where 'raised_amount_usd' value is zero.
rounds = rounds[rounds.raised_amount_usd.isin(rounds) == False]
rounds
```

```
                                                  permalink  \
0                                        /organization/-fame
1                                    /ORGANIZATION/-QOUNTER
2                                     /organization/-qounter
3                          /ORGANIZATION/-THE-ONE-OF-THEM-INC-
4                                        /organization/0-6-com
...                                                       ...
114944                                /organization/zzzzapp-com
114945                                /ORGANIZATION/ZZZZAPP-COM
114946                                     /organization/ãeron
114947                                   /ORGANIZATION/Ã”ASYS-2
114948   /organization/ä°novatiff-reklam-ve-tanä±tä±m-h...

                                    funding_round_permalink
funding_round_type  \
0         /funding-round/9a01d05418af9f794eebff7ace91f638
venture
1         /funding-round/22dacff496eb7acb2b901dec1dfe5633
venture
```

```
2         /funding-round/b44fbb94153f6cdef13083530bb48030
seed
3         /funding-round/650b8f704416801069bb178a1418776b
venture
4         /funding-round/5727accaeaa57461bd22a9bdd945382d
venture
...                                                     ...
...
114944    /funding-round/8f6d25b8ee4199e586484d817bceda05
convertible_note
114945    /funding-round/ff1aa06ed5da186c84f101549035d4ae
seed
114946    /funding-round/59f4dce44723b794f21ded3daed6e4fe
venture
114947    /funding-round/35f09d0794651719b02bbfd859ba9ff5
seed
114948    /funding-round/af942869878d2cd788ef5189b435ebc4
grant

        funding_round_code   funded_at  raised_amount_usd
0                        B  05-01-2015         10000000.0
1                        A  14-10-2014                NaN
2                      NaN  01-03-2014           700000.0
3                        B  30-01-2014          3406878.0
4                        A  19-03-2008          2000000.0
...                    ...         ...                ...
114944                 NaN  01-03-2014            41313.0
114945                 NaN  01-05-2013            32842.0
114946                   A  01-08-2014                NaN
114947                 NaN  01-01-2015            18192.0
114948                 NaN  01-10-2013            14851.0

[114949 rows x 6 columns]
```

```
# drop funding_round_code,funded_at columns
```

```
rounds

                                                  permalink  \
0                                        /organization/-fame
1                                     /ORGANIZATION/-QOUNTER
2                                     /organization/-qounter
3                          /ORGANIZATION/-THE-ONE-OF-THEM-INC-
4                                       /organization/0-6-com
...                                                       ...
114944                             /organization/zzzzapp-com
114945                            /ORGANIZATION/ZZZZAPP-COM
114946                                 /organization/ãeron
114947                                /ORGANIZATION/Ã”ASYS-2
114948  /organization/ä°novatiff-reklam-ve-tanä±tä±m-h...
```

```
                                  funding_round_permalink
funding_round_type  \
0         /funding-round/9a01d05418af9f794eebff7ace91f638
venture
1         /funding-round/22dacff496eb7acb2b901dec1dfe5633
venture
2         /funding-round/b44fbb94153f6cdef13083530bb48030
seed
3         /funding-round/650b8f704416801069bb178a1418776b
venture
4         /funding-round/5727accaeaa57461bd22a9bdd945382d
venture
...                                                    ...
...
114944    /funding-round/8f6d25b8ee4199e586484d817bceda05
convertible_note
114945    /funding-round/ff1aa06ed5da186c84f101549035d4ae
seed
114946    /funding-round/59f4dce44723b794f21ded3daed6e4fe
venture
114947    /funding-round/35f09d0794651719b02bbfd859ba9ff5
seed
114948    /funding-round/af942869878d2cd788ef5189b435ebc4
grant

        funding_round_code   funded_at  raised_amount_usd
0                        B  05-01-2015         10000000.0
1                        A  14-10-2014                NaN
2                      NaN  01-03-2014           700000.0
3                        B  30-01-2014          3406878.0
4                        A  19-03-2008          2000000.0
...                    ...         ...                ...
114944                 NaN  01-03-2014            41313.0
114945                 NaN  01-05-2013            32842.0
114946                   A  01-08-2014                NaN
114947                 NaN  01-01-2015            18192.0
114948                 NaN  01-10-2013            14851.0

[114949 rows x 6 columns]
```

```
# aggregate rows with funding_type and print mean investment for each
type
# you can use grouping
rounds.set_index(["funding_round_type"]).mean
```

```
<bound method NDFrame._add_numeric_operations.<locals>.mean of
permalink  \
funding_round_type
```

venture	/organization/-fame
venture	/ORGANIZATION/-QOUNTER
seed	/organization/-qounter
venture	/ORGANIZATION/-THE-ONE-OF-THEM-INC-
venture	/organization/0-6-com
...	...
convertible_note	/organization/zzzzapp-com
seed	/ORGANIZATION/ZZZZAPP-COM
venture	/organization/ãeron
seed	/ORGANIZATION/Ã"ASYS-2
grant	/organization/ä°novatiff-reklam-ve-tanä±tä±m-h...

funding_round_type	funding_round_permalink
venture	/funding-round/9a01d05418af9f794eebff7ace91f638
venture	/funding-round/22dacff496eb7acb2b901dec1dfe5633
seed	/funding-round/b44fbb94153f6cdef13083530bb48030
venture	/funding-round/650b8f704416801069bb178a1418776b
venture	/funding-round/5727accaeaa57461bd22a9bdd945382d
...	...
convertible_note	/funding-round/8f6d25b8ee4199e586484d817bceda05
seed	/funding-round/ff1aa06ed5da186c84f101549035d4ae
venture	/funding-round/59f4dce44723b794f21ded3daed6e4fe
seed	/funding-round/35f09d0794651719b02bbfd859ba9ff5
grant	/funding-round/af942869878d2cd788ef5189b435ebc4

funding_round_type	funding_round_code	funded_at	raised_amount_usd
venture	B	05-01-2015	10000000.0
venture	A	14-10-2014	NaN
seed	NaN	01-03-2014	700000.0
venture	B	30-01-2014	3406878.0
venture	A	19-03-2008	2000000.0
...	...	...	...
convertible_note	NaN	01-03-2014	41313.0
seed	NaN	01-05-2013	32842.0
venture	A	01-08-2014	NaN
seed	NaN	01-01-2015	18192.0

```
grant                                NaN  01-10-2013              14851.0

[114949 rows x 5 columns]>
```

```
# find out which funding_type have more funding. the retain the rows
of
# that paticular type and delete other rows.
rounds["raised_amount_usd"].value_counts()

#rounds["funding_round_type"].value_counts()
```

```
1000000.0     2860
2000000.0     2225
500000.0      2145
5000000.0     2057
10000000.0    2052
              ...
9259078.0        1
2665127.0        1
353102.0         1
782660.0         1
14851.0          1
Name: raised_amount_usd, Length: 22095, dtype: int64
```

```
#rounds.drop(["raised_amount_usd"!=1000000.0],axis=1)
drop_raised_amount_usd=rounds[rounds.raised_amount_usd =="10000000.0"]
#print(drop_raised_amount_usd)
print([rounds.raised_amount_usd==10000000.0])
```

```
[0          True
1         False
2         False
3         False
4         False
          ...
114944    False
114945    False
114946    False
114947    False
114948    False
Name: raised_amount_usd, Length: 114949, dtype: bool]
```

```
result_df=  rounds.loc[rounds["raised_amount_usd"]==10000000.0]
result_df
```

```
                                  permalink  \
0                      /organization/-fame
52                    /organization/1006-tv
133                  /ORGAN▮ZAT▮ON/17-MED▮A
141                   /ORGAN▮ZAT▮ON/17ZUOYE
190     /organization/1st-merchant-funding
```

```
...                                        ...
114711                  /ORGAN▮ZAT▮ON/ZPOWER
114742                   /organization/zubie
114799         /ORGAN▮ZAT▮ON/ZUN▮VERS▮TY-COM
114861         /ORGAN▮ZAT▮ON/ZYGA-TECHNOLOGY
114908                   /organization/zynga

                                  funding_round_permalink
funding_round_type  \
0         /funding-round/9a01d05418af9f794eebff7ace91f638
venture
52        /funding-round/b6aeb7401ec6993f92a16cbca153b600
venture
133       /funding-round/f8ffde9fa822843ac6ea9b66cc615df1
venture
141       /funding-round/8d87f771e938e0f31641bd600abbafca
venture
190       /funding-round/ed21cc0e56104c9e20611730a5ebefc1
debt_financing
...                                                  ...
...
114711    /funding-round/a8d010b9de7c04c98dad65033d381a40
venture
114742    /funding-round/c834914cc13644c86b33542a0622ee8e
venture
114799    /funding-round/a058abef15e17654e754acd8d4303af8
venture
114861    /funding-round/00d2d8d0c604892594b4bd25cf7e18b8
venture
114908    /funding-round/66d64809546fe2031d3479161e3c88f9
venture

        funding_round_code   funded_at  raised_amount_usd
0                        B  05-01-2015         10000000.0
52                       B  31-07-2014         10000000.0
133                      A  05-11-2015         10000000.0
141                      B  01-09-2013         10000000.0
190                    NaN  07-05-2014         10000000.0
...                    ...         ...                ...
114711                 NaN  28-05-2014         10000000.0
114742                   A  12-12-2013         10000000.0
114799                   B  10-01-2001         10000000.0
114861                   C  16-12-2013         10000000.0
114908                   A  01-01-2008         10000000.0

[2052 rows x 6 columns]
```

URL = "https://en.wikipedia.org/wiki/List_of_territorial_entities_where_English_is_an_official_language"

```
# use above link and extract all english( as primary language)
speaking countries.
'''
#-------------------this code will help you to get countries table
tables = pd.read_html(URL)
table=tables[1]
countries=pd.DataFrame(table)

you manually need to clean table for better understanding
'''
english_language_speaking=pd.read_html("https://en.wikipedia.org/wiki/
List_of_territorial_entities_where_English_is_an_official_language")
english_language_speaking
```

```
[                          Country ISO code      Geographic
region \
 0                Antigua and Barbuda      ATG
Caribbean
 1                     The Bahamas[1]      BHS
Caribbean
 2                        Barbados[2]      BRB
Caribbean
 3                         Belize[3]      BLZ        Central
America
 4                       Botswana[3]      BWA
Africa
 5                        Burundi[4]      BDI
Africa
 6                       Cameroon[1]      CMR
Africa
 7                            Canada      CAN          North
America
 8                       Dominica[1]      DMA
Caribbean
 9                       Eswatini[1]      SWZ
Africa
 10                          Fiji[1]      FJI
Oceania
 11                    The Gambia[1]      GMB
Africa
 12                         Ghana[1]      GHA
Africa
 13                        Grenada[1]      GRD
Caribbean
 14                         Guyana[5]      GUY          South
America
 15                        India[3][6]      IND
Asia
 16                      Ireland[7][8]      IRL
Europe
```

17	Jamaica[9]	JAM	Caribbean
18	Kenya[1]	KEN	Africa
19	Kiribati[1]	KIR	Oceania
20	Lesotho[1]	LSO	Africa
21	Liberia[1]	LBR	Africa
22	Malawi[10]	MWI	Africa
23	Malta[1]	MLT	Europe
24	Marshall Islands[1]	MHL	Oceania
25	Mauritius[1]	MUS	Africa / Indian Ocean
26	Micronesia[1]	FSM	Oceania
27	Namibia[1]	NAM	Africa
28	Nauru[11]	NRU	Oceania
29	Nigeria[1][12]	NGA	Africa
30	Pakistan[1]	PAK	Asia
31	Palau[3]	PLW	Oceania
32	Papua New Guinea[13][14]	PNG	Oceania
33	The Philippines	PHL	Asia
34	Rwanda[15]	RWA	Africa
35	Saint Kitts and Nevis[16]	KNA	Caribbean
36	Saint Lucia[1]	LCA	Caribbean
37	Saint Vincent and the Grenadines[17]	VCT	Caribbean
38	Samoa[18]	WSM	Oceania
39	Seychelles[1]	SYC	Africa / Indian Ocean
40	Sierra Leone[1]	SLE	Africa
41	Singapore	SGP	Asia

42	Solomon Islands[1]	SLB	Oceania
43	South Africa[19]	ZAF	Africa
44	South Sudan[20]	SSD	Africa
45	Sudan[1]	SDN	Africa
46	Tanzania[1]	TZA	Africa
47	Tonga[21]	TON	Oceania
48	Trinidad and Tobago[1]	TTO	Caribbean
49	Tuvalu[3]	TUV	Oceania
50	Uganda[22]	UGA	Africa
51	Vanuatu[23]	VUT	Oceania
52	Zambia[1]	ZMB	Africa
53	Zimbabwe[1]	ZWE	Africa

	Population1	Primary language?
0	85000	Yes (English-based creole language)
1	331000	Yes (English-based creole language)
2	294000	Yes (English-based creole language)
3	288000	Yes (English-based creole language)
4	1882000	Yes
5	10114505	No
6	22534532	No (co-official with French, but only spoken p...
7	38048738	Yes (Co-official with French, and a predominan...
8	73000	Yes (French-based creole language)
9	1141000	No
10	828000	Yes (used as lingua franca, mostly and widely ...
11	1709000	Yes
12	27000000	Yes (used as lingua franca)
13	111000	Yes (English-based creole language)
14	738000	Yes (English-based creole language)
15	1247540000	No (but official and educational)
16	4900000	Yes (Irish is co-official)
17	2714000	Yes (English-based creole language)
18	45010056	Yes (used in business and education)
19	95000	No
20	2008000	No
21	3750000	Yes
22	16407000	Yes (used as lingua franca)
23	430000	No (but official and in business / education)

```
 24        59000                                                  No
 25      1262000                                                 Yes
 26       110000                                                 Yes
 27      2074000                        No (used as lingua franca)
 28        10000                            No (but widely spoken)
 29    182202000                  Yes (used as official language)
 30    212742631                 No (but official and educational)
 31        20000                                                  No
 32      7059653                Yes (but official and educational)
 33    110864327                Yes (but official and educational)
 34     13240439                                                  No
 35        50000               Yes (English-based creole language)
 36       165000                Yes (French-based creole language)
 37       120000               Yes (English-based creole language)
 38       188000                                                  No
 39        87000                                                  No
 40      6190280               Yes (English-based creole language)
 41      5469700  Yes (official language, lingua franca, mostly ...
 42       507000                                                 Yes
 43     54956900  Yes (official, educational and lingua franca i...
 44     12340000                                                 Yes
 45     40235000                                                  No
 46     51820000                                                  No
 47       100000                                                  No
 48      1333000               Yes (English-based creole language)
 49        11000                                                  No
 50     47053690                       No (used as lingua franca)
 51       226000                                                  No
 52     16212000                     Yes (used as lingua franca)
 53     13061239                      No (used as lingua franca)  ,
            Country ISO code Geographic region  Population1 Primary
language?
 0        Australia      AUS            Oceania     25795700
Yes
 1  New Zealand[24]      NZL            Oceania      4893830
Yes
 2   United Kingdom      GBR             Europe     66040229
Yes
 3    United States      USA     North America    328239523
Yes,
                    Country ISO code            Geographic region  \
 0            Bahrain[25][26]      BHR           Asia / Middle East
 1             Bangladesh[27]      BGD                         Asia
 2                Bhutan[28]      BTN                         Asia
 3              Cambodia[29]      KHM                         Asia
 4                Cyprus[30]      CYP  Europe / Asia / Middle East
 5                Eritrea[1]      ERI                       Africa
 6               Ethiopia[1]      ETH                       Africa
 7         Israel[31][32][33]      ISR           Asia / Middle East
 8                Jordan[34]      JOR           Asia / Middle East
```

9	Kuwait[35]	KWT	Asia / Middle East	1378000
10	Malaysia	MYS	Asia	150039000
11	Maldives[36]	MDV	Asia	727145
12	Myanmar[37]	MMR	Asia	15288489
13	Oman[38]	OMN	Asia / Middle East	1141166
14	Qatar[39]	QAT	Asia / Middle East	6234000
15	Sri Lanka[40][41]	LKA	Asia	85000000
16	United Arab Emirates[42]	ARE	Asia / Middle East	8051200

,

	Entity	Region	Population1
0	Akrotiri and Dhekelia	Europe	15700
1	American Samoa11	Oceania	67700
2	Anguilla[1]	Caribbean	18090
3	Bermuda9[1]	North America	65000
4	British Virgin Islands[1]	Caribbean	23000
5	Cayman Islands[3]	Caribbean	47000
6	Cook Islands[1]14	Oceania	20000
7	Curaçao[43]	Caribbean	150563
8	Falkland Islands	South America	3000
9	Gibraltar[1]	Europe	33000
10	Guam4	Oceania	173000
11	Hong Kong2[1]	Asia	7097600
12	Isle of Man8	Europe	80058
13	Jersey6[1]	Europe	89300
14	Niue[1]14	Oceania	1600
15	Norfolk Island[1]	Oceania	1828
16	Northern Mariana Islands7	Oceania	53883
17	Pitcairn Islands13[1]	Oceania	50
18	Puerto Rico3	Caribbean	3991000
19	Rotuma	Oceania	1594
20	Sint Maarten[45]	Caribbean	40900
21	Turks and Caicos Islands[1]	Caribbean	26000

```
22         U.S. Virgin Islands5      Caribbean       111000

                                  Primary language?
0                                                No
1                                No (official language)
2                    No (English-based creole language)
3                                               Yes
4                    No (English-based creole language)
5                    No (English-based creole language)
6                                                No
7                                                No
8                                               Yes
9                                               Yes
10                     Yes (Co-official with Chamorro)
11  No (but de jure and de facto co-official with ...
12                                              Yes
13                                              Yes
14                                               No
15                   No (English-based creole language)
16                     Yes (Co-official with Chamorro)
17                                              Yes
18  No (co-official with Spanish as the primary la...
19                                               No
20                   No (English-based creole language)
21                   No (English-based creole language)
22                   No (English-based creole language)  ,
                                        Entity          Region  \
0                                     Barbuda       Caribbean
1                   British Indian Ocean Territory    Indian Ocean
2                                    Guernsey10          Europe
3                                  Montserrat[1]       Caribbean
4  Saint Helena, Ascension and Tristan da Cunha[3]  South Atlantic

   Population1
0         1300
1         3000
2        61811
3         5900
4         5660  ,
                          Entity     Region  Population1
0         Christmas Island12[1]  Australia         1508
1  Cocos (Keeling) Islands16[1]  Australia          596
2                    Tokelau[46]    Oceania         1400,
                                      Subdivision         Country
\
0                                     Alabama[47]   United States

1                                      Alaska[48]   United States

2                                     Arizona[49]   United States
```

3	Arkansas[47]	United States
4	California[47]	United States
5	Colorado[47]	United States
6	Florida[47]	United States
7	Georgia[47]	United States
8	Hawaii[47]	United States
9	Idaho[47]	United States
10	Illinois[47]	United States
11	Indiana[47]	United States
12	Iowa[47]	United States
13	Kansas[47]	United States
14	Kentucky[47]	United States
15	Massachusetts[50]	United States
16	Mississippi[47]	United States
17	Missouri[47]	United States
18	Montana[47]	United States
19	Nebraska[47]	United States
20	New Hampshire[47]	United States
21	North Carolina[47]	United States
22	North Dakota[47]	United States
23	Northern Ireland[51]	United Kingdom
24	Oklahoma[52]	United States
25	North Caribbean Coast Autonomous Region[53][ci...	Nicaragua
26	South Caribbean Coast Autonomous Region[54][ci...	Nicaragua

27	Saba[55]	Netherlands
28	San Andrés y Providencia[56]	Colombia
29	Sarawak[57][58][59]	Malaysia
30	Scotland[60]	United Kingdom
31	Sint Eustatius[55]	Netherlands
32	South Carolina[47]	United States
33	South Dakota[47]	United States
34	Tennessee[47]	United States
35	Utah[47]	United States
36	Virginia[47]	United States
37	Wales[61]	United Kingdom
38	West Virginia[62]	United States
39	Wyoming[47]	United States

	Region	Population
0	North America	4833722
1	North America	735132
2	North America	6626624
3	North America	2959373
4	North America	38332521
5	North America	5268367
6	North America	21299325
7	North America	10519475
8	Oceania	1404054
9	North America	1612136
10	North America	12882135
11	North America	6570902
12	North America	3090416
13	North America	2893957
14	North America	4395295
15	North America	6794422
16	North America	2991207
17	North America	6083672
18	North America	1015165
19	North America	1868516

20	North America	1323459
21	North America	9848060
22	North America	723393
23	Europe	1876695
24	North America	3850568
25	Central America	480874
26	Central America	385102
27	Caribbean	1991
28	South America	75167
29	Asia	2471140
30	Europe	5424800
31	Caribbean	3897
32	North America	4774839
33	North America	844877
34	North America	6495978
35	North America	2900872
36	North America	8260405
37	Europe	3125000
38	North America	1844128
39	North America	582658

,
.mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"["}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:"]"}.mw-parser-output .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;margin:0 4em}vteCountries and languages lists \

0	CountriesBy languagesSpoken Countries by spoke...
1	Countries
2	By languagesSpoken Countries by spoken languag...
3	By languages
4	Spoken
5	Official
6	Endonyms
7	Exonyms

8 LanguagesBy continent Africa Americas North So...

9 Languages

10 By continent Africa Americas North South Asia ...

11 By continent

12 By country

13 By population

14 By family

15 Language-basedgeopoliticalorganizations

16 See also

.mw-parser-output .navbar{display:inline;font-size:88%;font-weight:normal}.mw-parser-output .navbar-collapse{float:left;text-align:left}.mw-parser-output .navbar-boxtext{word-spacing:0}.mw-parser-output .navbar ul{display:inline-block;white-space:nowrap;line-height:inherit}.mw-parser-output .navbar-brackets::before{margin-right:-0.125em;content:"["}.mw-parser-output .navbar-brackets::after{margin-left:-0.125em;content:"]"}.mw-parser-output .navbar li{word-spacing:-0.125em}.mw-parser-output .navbar a>span,.mw-parser-output .navbar a>abbr{text-decoration:inherit}.mw-parser-output .navbar-mini abbr{font-variant:small-caps;border-bottom:none;text-decoration:none;cursor:inherit}.mw-parser-output .navbar-ct-full{font-size:114%;margin:0 7em}.mw-parser-output .navbar-ct-mini{font-size:114%;margin:0 4em}vteCountries and languages lists.1

0 CountriesBy languagesSpoken Countries by spoke...

1 Countries

2 By languagesSpoken Countries by spoken languag...

3 Spoken Countries by spoken languages Official ...

4 Countries by spoken languages

5 Countries by the number of recognized official...

6 Countries and capitals in native languages

```
7   Country names in various languages A–C D–I J–P...

8   LanguagesBy continent Africa Americas North So...

9                                                Languages

10  By continent Africa Americas North South Asia ...

11  Africa Americas North South Asia East South Eu...

12  Official languages by country and territory Li...

13  By number of native speakers By number of tota...

14  Language families List of Afro-Asiatic languag...

15  Arab League (Arabic) Dutch Language Union (Dut...

16               Lists of languages Category:Languages
,
                                                Countries  \
0  By languagesSpoken Countries by spoken languag...
1                                            By languages
2                                                  Spoken
3                                                Official
4                                                Endonyms
5                                                 Exonyms

                                               Countries.1
0  By languagesSpoken Countries by spoken languag...
1  Spoken Countries by spoken languages Official ...
2                     Countries by spoken languages
3  Countries by the number of recognized official...
4         Countries and capitals in native languages
5  Country names in various languages A–C D–I J–P...   ,
              0                                                    1
0  By languages  Spoken Countries by spoken languages Official ...
1        Spoken                        Countries by spoken languages
2      Official  Countries by the number of recognized official...
3     Endonyms          Countries and capitals in native languages
4       Exonyms  Country names in various languages A–C D–I J–P...,
          0                                                      1
0    Spoken                        Countries by spoken languages
1  Official  Countries by the number of recognized official ..... ,
                                                Languages  \
0  By continent Africa Americas North South Asia ...
1                                            By continent
2                                              By country
3                                           By population
```

4	By family
5	Language-basedgeopoliticalorganizations
6	See also

	Languages.1
0	By continent Africa Americas North South Asia ...
1	Africa Americas North South Asia East South Eu...
2	Official languages by country and territory Li...
3	By number of native speakers By number of tota...
4	Language families List of Afro-Asiatic languag...
5	Arab League (Arabic) Dutch Language Union (Dut...
6	Lists of languages Category:Languages ,

	0 \
0	By continent
1	By country
2	By population
3	By family
4	Language-basedgeopoliticalorganizations
5	See also

	1
0	Africa Americas North South Asia East South Eu...
1	Official languages by country and territory Li...
2	By number of native speakers By number of tota...
3	Language families List of Afro-Asiatic languag...
4	Arab League (Arabic) Dutch Language Union (Dut...
5	Lists of languages Category:Languages ,

	vteEnglish-speaking world \
0	NaN
1	Further linksArticles English-speaking world H...
2	Further links
3	Articles English-speaking world History of the...
4	Articles
5	Lists
6	Countries and territories where English is the...
7	NaN
8	Countries and territories where English is the...
9	Africa
10	Americas
11	Europe
12	Oceania
13	NaN
14	Countries and territories where English is an ...
15	Africa
16	Americas
17	Asia
18	Europe
19	Oceania
20	Dependencies shown in italics.

	vteEnglish-speaking world.1
0	NaN
1	Further linksArticles English-speaking world H...
2	Further links
3	Articles English-speaking world History of the...
4	English-speaking world History of the English ...
5	List of countries by English-speaking populati...
6	Countries and territories where English is the...
7	Countries and territories where English is the...
8	Countries and territories where English is the...
9	Saint Helena, Ascension and Tristan da Cunha
10	Anguilla Antigua and Barbuda The Bahamas Barba...
11	Guernsey Ireland Isle of Man Jersey United Kin...
12	Australia New Zealand Norfolk Island Pitcairn ...
13	Countries and territories where English is an ...
14	Countries and territories where English is an ...
15	Botswana Cameroon Eswatini The Gambia Ghana Ke...
16	Puerto Rico
17	Christmas Island Cocos (Keeling) Islands Hong ...
18	Gibraltar Akrotiri and Dhekelia Malta
19	American Samoa Cook Islands Fiji Guam Kiribati...
20	Dependencies shown in italics.

	Further links	\
0	Articles English-speaking world History of the...	
1	Articles	
2	Lists	
3	Countries and territories where English is the...	
4	NaN	
5	Countries and territories where English is the...	
6	Africa	
7	Americas	
8	Europe	
9	Oceania	
10	NaN	
11	Countries and territories where English is an ...	
12	Africa	
13	Americas	
14	Asia	
15	Europe	
16	Oceania	
17	Dependencies shown in italics.	

	Further links.1
0	Articles English-speaking world History of the...
1	English-speaking world History of the English ...
2	List of countries by English-speaking populati...
3	Countries and territories where English is the...
4	Countries and territories where English is the...
5	Countries and territories where English is the...
6	Saint Helena, Ascension and Tristan da Cunha

```
7   Anguilla Antigua and Barbuda The Bahamas Barba...
8   Guernsey Ireland Isle of Man Jersey United Kin...
9   Australia New Zealand Norfolk Island Pitcairn ...
10  Countries and territories where English is an ...
11  Countries and territories where English is an ...
12  Botswana Cameroon Eswatini The Gambia Ghana Ke...
13                                        Puerto Rico
14  Christmas Island Cocos (Keeling) Islands Hong ...
15             Gibraltar Akrotiri and Dhekelia Malta
16  American Samoa Cook Islands Fiji Guam Kiribati...
17                    Dependencies shown in italics.  ,
                                                    0  \
0                                             Articles
1                                                Lists
2   Countries and territories where English is the...
3                                                  NaN
4   Countries and territories where English is the...
5                                               Africa
6                                             Americas
7                                               Europe
8                                              Oceania
9                                                  NaN
10  Countries and territories where English is an ...
11                                              Africa
12                                            Americas
13                                                Asia
14                                              Europe
15                                             Oceania
16                    Dependencies shown in italics.

                                                    1
0   English-speaking world History of the English ...
1   List of countries by English-speaking populati...
2   Countries and territories where English is the...
3   Countries and territories where English is the...
4   Countries and territories where English is the...
5        Saint Helena, Ascension and Tristan da Cunha
6   Anguilla Antigua and Barbuda The Bahamas Barba...
7   Guernsey Ireland Isle of Man Jersey United Kin...
8   Australia New Zealand Norfolk Island Pitcairn ...
9   Countries and territories where English is an ...
10  Countries and territories where English is an ...
11  Botswana Cameroon Eswatini The Gambia Ghana Ke...
12                                        Puerto Rico
13  Christmas Island Cocos (Keeling) Islands Hong ...
14             Gibraltar Akrotiri and Dhekelia Malta
15  American Samoa Cook Islands Fiji Guam Kiribati...
16                    Dependencies shown in italics.  ,
                                                    0  \
0                                                  NaN
```

```
1   Countries and territories where English is the...
2                                              Africa
3                                            Americas
4                                              Europe
5                                             Oceania
6                                                 NaN
7   Countries and territories where English is an ...
8                                              Africa
9                                            Americas
10                                               Asia
11                                             Europe
12                                            Oceania

                                                    1
0   Countries and territories where English is the...
1   Countries and territories where English is the...
2        Saint Helena, Ascension and Tristan da Cunha
3   Anguilla Antigua and Barbuda The Bahamas Barba...
4   Guernsey Ireland Isle of Man Jersey United Kin...
5   Australia New Zealand Norfolk Island Pitcairn ...
6   Countries and territories where English is an ...
7   Countries and territories where English is an ...
8   Botswana Cameroon Eswatini The Gambia Ghana Ke...
9                                         Puerto Rico
10  Christmas Island Cocos (Keeling) Islands Hong ...
11              Gibraltar Akrotiri and Dhekelia Malta
12  American Samoa Cook Islands Fiji Guam Kiribati...   ,
                                                    0  \
0   Countries and territories where English is the...
1                                              Africa
2                                            Americas
3                                              Europe
4                                             Oceania

                                                    1
0   Countries and territories where English is the...
1        Saint Helena, Ascension and Tristan da Cunha
2   Anguilla Antigua and Barbuda The Bahamas Barba...
3   Guernsey Ireland Isle of Man Jersey United Kin...
4   Australia New Zealand Norfolk Island Pitcairn ...   ,
                                                    0  \
0   Countries and territories where English is an ...
1                                              Africa
2                                            Americas
3                                                Asia
4                                              Europe
5                                             Oceania

                                                    1
0   Countries and territories where English is an ...
```

```
 1  Botswana Cameroon Eswatini The Gambia Ghana Ke...
 2                                          Puerto Rico
 3  Christmas Island Cocos (Keeling) Islands Hong ...
 4             Gibraltar Akrotiri and Dhekelia Malta
 5  American Samoa Cook Islands Fiji Guam Kiribati...   ,
                                                  0                    1
 0  Authority control: National libraries  Israel United States]
```

```
#print(drop_raised_amount_usd)
#print([english_language_speaking.Country.head(3)])
import matplotlib.pyplot as plt
import pandas as pd
english_language_speaking.Country.value_counts()
```

```
Australia           1
New Zealand[24]     1
United Kingdom      1
United States       1
Name: Country, dtype: int64
```

```
'''
#-------------------this code will help you to get countries table
tables = pd.read_html(URL)
table=tables[1]
countries=pd.DataFrame(table)

you manually need to clean table for better understanding
'''
english_language_speaking=english_language_speaking[1]
english_language_speaking_countries=pd.DataFrame(english_language_spea
king)
english_language_speaking_countries
```

```
           Country ISO code Geographic region  Population1 Primary
language?
0        Australia      AUS            Oceania     25795700
Yes
1  New Zealand[24]      NZL            Oceania      4893830
Yes
2   United Kingdom      GBR             Europe     66040229
Yes
3    United States      USA      North America    328239523
Yes
```

```
# now group data by country code (as per country code is given
dataframe)
# find out top 10 countries with highest funding. (raised_amount_usd
is the column to aggregate)
"""result_df =
english_language_speaking.loc[english_language_speaking['Country']==en
glish]
```

```
result_df  """
from pandas.api.types import is_numeric_dtype

english_language_speaking_countries = []

for raised_amount_usd in english_language_speaking.columns:
    top_values = []
    if is_numeric_dtype(english_language_speaking[raised_amount_usd]):
        top_values =
english_language_speaking[raised_amount_usd].nlargest(n=10)

english_language_speaking_countries.append(pd.DataFrame({raised_amount
_usd: top_values}).reset_index(drop=True))
pd.concat(english_language_speaking_countries, axis=1)
```

```
   Population1
0    328239523
1     66040229
2     25795700
3      4893830
```

```
# find out top 3 english speaking countries to invest.(based on raised
amount of investment)
# you should get. [United States of America , India, United Kingdom]
english_language_speaking.value_counts()
```

```
Country          ISO code  Geographic region  Population1  Primary
language?
Australia        AUS       Oceania            25795700     Yes
1
New Zealand[24]  NZL       Oceania            4893830      Yes
1
United Kingdom   GBR       Europe             66040229     Yes
1
United States    USA       North America      328239523    Yes
1
dtype: int64
```

```
abc=english_language_speaking[english_language_speaking.Country
=="Population1"]
#print(drop_raised_amount_usd)
#print([english_language_speaking.Country.head(3)])
import matplotlib.pyplot as plt
import pandas as pd
temp = english_language_speaking.groupby(["Country"]).sum()
["Population1"]

plt.pie(temp.values,labels=temp.index,colors=["red","green","blue","or
ange","black","brown"])

plt.show()
```

```
---------------------------------------------------------------------------
------
NameError                                 Traceback (most recent call
last)
Input In [1], in <cell line: 6>()
      4 import matplotlib.pyplot as plt
      5 import pandas as pd
----> 6 temp = english_language_speaking.groupby(["Country"]).sum()
["Population1"]
      8
plt.pie(temp.values,labels=temp.index,colors=["red","green","blue","or
ange","black","brown"])
     10 plt.show()

NameError: name "english_language_speaking" is not defined
```

Plese explain how would you split investment percentage amoung these countries and why

this question is understand your business prespective.

"""sir, Im srikanth sri G-17 python. here we are having manyways to find how to split investment percentage amoung these countries. I thought to do by visualizing the data. so i used the above link and extracted all english speaking countries. Now i tried to clean some data with that i will able to understand the data properly.after that we can observe heighest funding countries.As you asked top 3 english speaking countries to invest i thought to do By showing in pie chart. """

how do you decide, What value should be imputed in missing place.

this is subjective question. write your own answer based on what you understood.

"""Its upon the data what we have recieved. some times we can fill some null values for understanding purpose. but it willnot work every time. as we observe some bank data sets, data will be different for differnet persons. banker wants to know what kind of people he should approach so that he can make more people sign up for personal loans using the data he collected. so data is very important but the way someone visualise the data is the important thing. so according to data we have to imputed in missing places"""

Assignment 3 Data Visualization, EDA and Statistics

Exploratory data analysis on COVID-19 across the globe. Problem is divided into 3 sections.

Data Understaing, EDA and Visualization Statistics.

Please pledge for honour code.

I,Dubbaka Srikanth, promise that during the course of this assignment I shall not use unethical and nefarious means in an attempt to defraud the sanctity of the assignment and gain an unfair advantage. I will adhere to the virtues of truth and honesty. I will honour the SURE Trust honour code.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
```

```
data = pd.read_csv(r"F:\covid_19.csv")
```

Data Understanding

```
data
```

```
      Province/State          Country/Region        Lat        Long
Date  \
0                NaN             Afghanistan  33.939110  67.709953
2020-01-22
1                NaN                 Albania  41.153300  20.168300
2020-01-22
2                NaN                 Algeria  28.033900   1.659600
2020-01-22
3                NaN                 Andorra  42.506300   1.521800
2020-01-22
4                NaN                  Angola -11.202700  17.873900
2020-01-22
...              ...                     ...        ...        ...
...
49063            NaN  Sao Tome and Principe    0.186400   6.613100
2020-07-27
49064            NaN                   Yemen  15.552727  48.516388
2020-07-27
49065            NaN                 Comoros -11.645500  43.333300
2020-07-27
49066            NaN              Tajikistan  38.861000  71.276100
2020-07-27
49067            NaN                 Lesotho -29.610000  28.233600
2020-07-27

       Confirmed  Deaths  Recovered  Active              WHO Region
```

```
0              0       0          0       0  Eastern Mediterranean
1              0       0          0       0                 Europe
2              0       0          0       0                 Africa
3              0       0          0       0                 Europe
4              0       0          0       0                 Africa
...          ...     ...        ...     ...                    ...
49063        865      14        734     117                 Africa
49064       1691     483        833     375  Eastern Mediterranean
49065        354       7        328      19                 Africa
49066       7235      60       6028    1147                 Europe
49067        505      12        128     365                 Africa

[49068 rows x 10 columns]
```

```
#1 properties of data. shape, columns
data.shape
```

```
(49068, 10)
```

```
#1 properties of data. shape, columns
data.columns.tolist()
```

```
['Province/State',
 'Country/Region',
 'Lat',
 'Long',
 'Date',
 'Confirmed',
 'Deaths',
 'Recovered',
 'Active',
 'WHO Region']
```

```
#2 data glipse with info and describe
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49068 entries, 0 to 49067
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Province/State  14664 non-null  object 
 1   Country/Region  49068 non-null  object 
 2   Lat             49068 non-null  float64
 3   Long            49068 non-null  float64
 4   Date            49068 non-null  object 
 5   Confirmed       49068 non-null  int64  
 6   Deaths          49068 non-null  int64  
 7   Recovered       49068 non-null  int64  
 8   Active          49068 non-null  int64  
 9   WHO Region      49068 non-null  object 
```

```
dtypes: float64(2), int64(4), object(4)
memory usage: 3.7+ MB
```

```
#2 data glipse with info and describe
data.describe()
```

```
                Lat           Long      Confirmed          Deaths
Recovered  \
count  49068.000000  49068.000000  4.906800e+04   49068.000000
4.906800e+04
mean      21.433730     23.528236  1.688490e+04     884.179160
7.915713e+03
std       24.950320     70.442740  1.273002e+05    6313.584411
5.480092e+04
min      -51.796300   -135.000000  0.000000e+00       0.000000
0.000000e+00
25%        7.873054    -15.310100  4.000000e+00       0.000000
0.000000e+00
50%       23.634500     21.745300  1.680000e+02       2.000000
2.900000e+01
75%       41.204380     80.771797  1.518250e+03      30.000000
6.660000e+02
max       71.706900    178.065000  4.290259e+06  148011.000000
1.846641e+06

             Active
count  4.906800e+04
mean   8.085012e+03
std 7.625890e+04min
-1.400000e+01   25%
0.000000e+00
50%     2.600000e+01
75%     6.060000e+02
max     2.816444e+06
```

```
#(Optional step) Data cleaning if required.
pd.set_option("display.max_rows", 180)
(data.filter(["Country/Region" ,"Date" ,"Confirmed",
"Deaths","Recovered", "Active", "WHO Region" ]).drop_duplicates())
```

```
              Country/Region        Date  Confirmed  Deaths  Recovered
\
0                Afghanistan  2020-01-22          0       0          0

1                    Albania  2020-01-22          0       0          0

2                    Algeria  2020-01-22          0       0          0

3                    Andorra  2020-01-22          0       0          0

4                     Angola  2020-01-22          0       0          0
```

```
...                      ...         ...        ...     ...        ...
49063  Sao Tome and Principe  2020-07-27        865      14        734
49064                  Yemen  2020-07-27       1691     483        833
49065                Comoros  2020-07-27        354       7        328
49066             Tajikistan  2020-07-27       7235      60       6028
49067                Lesotho  2020-07-27        505      12        128
```

```
       Active             WHO Region
0           0  Eastern Mediterranean
1           0                 Europe
2           0                 Africa
3           0                 Europe
4           0                 Africa
...       ...                    ...
49063     117                 Africa
49064     375  Eastern Mediterranean
49065      19                 Africa
49066    1147                 Europe
49067     365                 Africa

[47172 rows x 7 columns]
```

```
(data
 .isnull()
 .sum()
)
```

```
Province/State    34404
Country/Region        0
Lat                   0
Long                  0
Date                  0
Confirmed             0
Deaths                0
Recovered             0
Active                0
WHO Region            0
dtype: int64
```

```
#3 which country had highest covid DEATHS in january 2020.
df=data[(data.Date>="2020-01-01") & (data.Date<="2020-01-31")]
df[df.Deaths==df.Deaths.max()][["Country/Region"]]
```

```
     Country/Region
2410          China
```

```
#4 which WHO region had highest ACTIVE covid cases in March 2020.
dx=data[(data.Date>="2020-03-01") & (data.Date<="2020-03-31")]
dx[dx.Active==dx.Active.max()][["WHO Region"]]
```

```
      WHO Region
18232   Americas
```

```
#5 print top 5 countries with highest CONFIRMED covid cases. Print
month wise.
"""dp=data.loc[data["Date"]=="2020-03-
22"].sort_values(by='Confirmed',ascending=False)
dp.head(10)"""
for i in range(1,6):
    i=str(i)
    m=data[(data.Date>="2020-0"+i+"-01") & (data.Date<="2020-0"+i+"-
31")]
    m.sort_values("Confirmed")
    print(m[-5:][["Country/Region"]])
```

```
             Country/Region
2605  Sao Tome and Principe
2606                 Yemen
2607               Comoros
2608            Tajikistan
2609               Lesotho
             Country/Region
10174  Sao Tome and Principe
10175                  Yemen
10176                Comoros
10177             Tajikistan
10178                Lesotho
             Country/Region
18265  Sao Tome and Principe
18266                  Yemen
18267                Comoros
18268             Tajikistan
18269                Lesotho
             Country/Region
26095  Sao Tome and Principe
26096                  Yemen
26097                Comoros
26098             Tajikistan
26099                Lesotho
             Country/Region
34186  Sao Tome and Principe
34187                  Yemen
34188                Comoros
```

```
34189              Tajikistan
34190                 Lesotho
```

```
#6 top 3 countries with highest DEATH covid cases print month wise.
"""dp=data.loc[data["Date"]=="2020-02-
22"].sort_values(by='Deaths',ascending=False)
dp.head(3)"""
for i in range(1,12):
    i=str(i)
    o=data[(data.Date>="2020-0"+i+"-01") & (data.Date<="2020-0"+i+"-
31")]
    o.sort_values("Deaths")
    print(o[-3:][["Country/Region"]])
```

```
     Country/Region
2607         Comoros
2608      Tajikistan
2609         Lesotho
      Country/Region
10176        Comoros
10177     Tajikistan
10178        Lesotho
      Country/Region
18267        Comoros
18268     Tajikistan
18269        Lesotho
      Country/Region
26097        Comoros
26098     Tajikistan
26099        Lesotho
      Country/Region
34188        Comoros
34189     Tajikistan
34190        Lesotho
      Country/Region
42018        Comoros
42019     Tajikistan
42020        Lesotho
      Country/Region
49065        Comoros
49066     Tajikistan
49067        Lesotho
Empty DataFrame
Columns: [Country/Region]
Index: []
Empty DataFrame
Columns: [Country/Region]
Index: []
Empty DataFrame
Columns: [Country/Region]
Index: []
```

```
Empty DataFrame
Columns: [Country/Region]
Index: []
```

```
#7 print countries which are amongst top 10 with highest CONFIRMED
cases
# for 3 months consecutively.
"""p=[]
for i in range(1,8):
    i=str(i)
    q=data[(data.Date>='2022-0'+i+'-01')&(data.Date>='2022-0'+i+'-
31')]
    q.sort_values('Confirmed')
    p.append(q[-10:][['Country/Region']].values)
for i in range(5):
    print(p[i][np.in1d(p[i],p[i+1],p[i+2])])"""
p=[]
for i in range(1,8):
 i=str(i)
 q=data[(data.Date>="2020-0"+i+"-01") & (data.Date<="2020-0"+i+"-31")]
 q.sort_values("Confirmed")
 p.append(q[-10:][["Country/Region"]].values)
for i in range(5):
 print( p[i][np.in1d(p[i],p[i+1],p[i+2])] )
```

```
[["Malawi"]
 ["United Kingdom"]
 ["France"]
 ["South Sudan"]
 ["Western Sahara"]
 ["Sao Tome and Principe"]
 ["Yemen"]
 ["Comoros"]
 ["Tajikistan"]
 ["Lesotho"]]
[["Malawi"]
 ["United Kingdom"]
 ["France"]
 ["South Sudan"]
 ["Western Sahara"]
 ["Sao Tome and Principe"]
 ["Yemen"]
 ["Comoros"]
 ["Tajikistan"]
 ["Lesotho"]]
[["Malawi"]
 ["United Kingdom"]
 ["France"]
 ["South Sudan"]
 ["Western Sahara"]
 ["Sao Tome and Principe"]
```

```
 ["Yemen"]
 ["Comoros"]
 ["Tajikistan"]
 ["Lesotho"]]
[["Malawi"]
 ["United Kingdom"]
 ["France"]
 ["South Sudan"]
 ["Western Sahara"]
 ["Sao Tome and Principe"]
 ["Yemen"]
 ["Comoros"]
 ["Tajikistan"]
 ["Lesotho"]]
[["Malawi"]
 ["United Kingdom"]
 ["France"]
 ["South Sudan"]
 ["Western Sahara"]
 ["Sao Tome and Principe"]
 ["Yemen"]
 ["Comoros"]
 ["Tajikistan"]
 ["Lesotho"]]
```

```
#8 how many total CONFIRMED cases in Asia as per dataset
a=data[data["WHO Region"]=="South-East Asia"].Confirmed.sum()
a
```

```
55118365
```

```
#9 Which WHO region have highest DEATHS?
p=data.sort_values(by="Deaths",ascending=False)
p.head(1)
```

```
      Province/State Country/Region    Lat    Long        Date
Confirmed  \
49030            NaN             US  40.0 -100.0  2020-07-27
4290259

       Deaths  Recovered    Active WHO Region
49030  148011    1325804   2816444   Americas
```

```
#10 print top 10 contries with LOWEST CONFIRMED:DEATH ratio print
month wise.
p=data.sort_values(by="Deaths",ascending=True)
p.head(10)
```

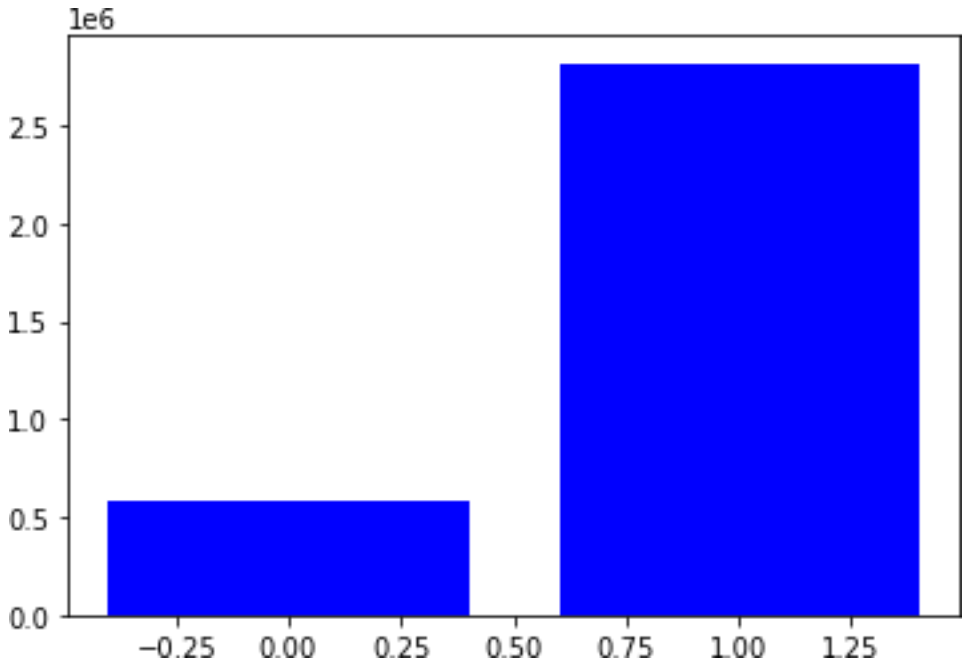
```
         Province/State Country/Region        Lat         Long
Date  \
0                   NaN    Afghanistan  33.939110   67.709953  2020-
```

```
01-22
14989          St Martin          France   18.070800  -63.050100  2020-
03-19
14988  Saint Barthelemy          France   17.900000  -62.833300  2020-
03-19
14987           Reunion          France  -21.115100   55.536400  2020-
03-19
14986     New Caledonia          France  -20.904305  165.618042  2020-
03-19
14985           Mayotte          France  -12.827500   45.166244  2020-
03-19
14984        Guadeloupe          France   16.265000  -61.551000  2020-
03-19
14983  French Polynesia          France  -17.679700  149.406800  2020-
03-19
14992               NaN           Gabon   -0.803700   11.609400  2020-
03-19
14982     French Guiana          France    3.933900  -53.125800  2020-
03-19

       Confirmed  Deaths  Recovered  Active              WHO Region
0              0       0          0       0  Eastern Mediterranean
14989          4       0          0       4                  Europe
14988          3       0          0       3                  Europe
14987         14       0          0      14                  Europe
14986          2       0          0       2                  Europe
14985          3       0          0       3                  Europe
14984         33       0          0      33                  Europe
14983          6       0          0       6                  Europe
14992          1       0          0       1                  Africa
14982         11       0          0      11                  Europe
```

```
#11 print India's CONFIRMED:RECOVERED cases ratio in the month of
March
ab=data[(data.Date>="2020-03-01")& (data.Date>="2020-03-31")]
ab.Recovered.sum()/ab.Confirmed.sum()
```

```
0.4709724860053174
```

```
#12 which month have highest active cases in USA.
plt.bar(data["Country/Region"]=="US", data["Active"], color="blue")
plt.show()
```

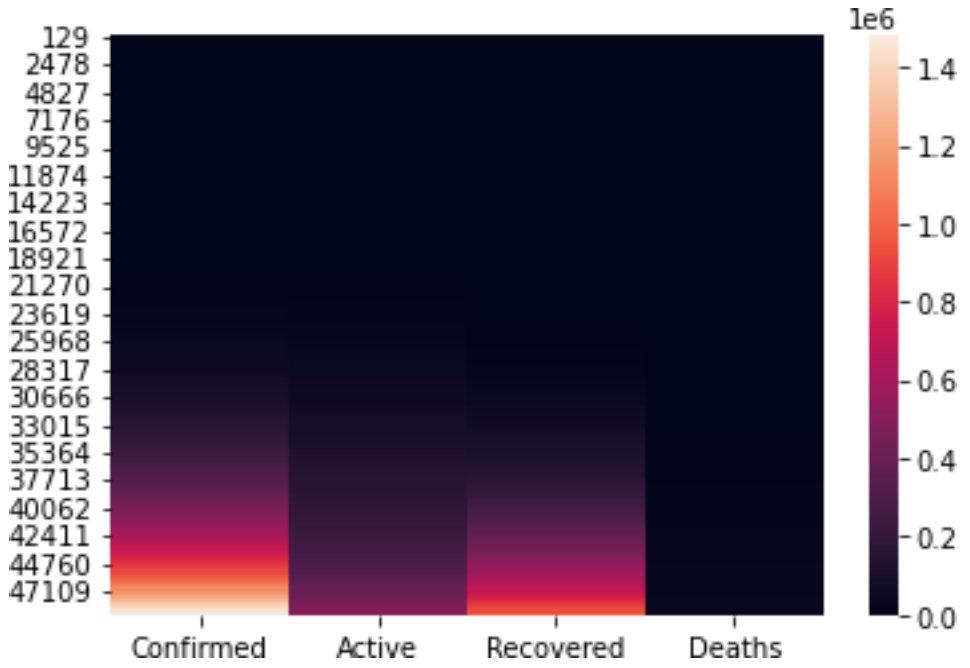



```
#12 which month have highest active cases in USA.
highest_active_cases=data.groupby(["Country/Region","Date"]).sum("Acti
ve")
print(highest_active_cases.sort_values(by="Active",ascending=False).he
ad(1))
```

```
                              Lat    Long  Confirmed  Deaths  Recovered
Active
Country/Region Date

US             2020-07-27   40.0 -100.0    4290259  148011    1325804
2816444
```

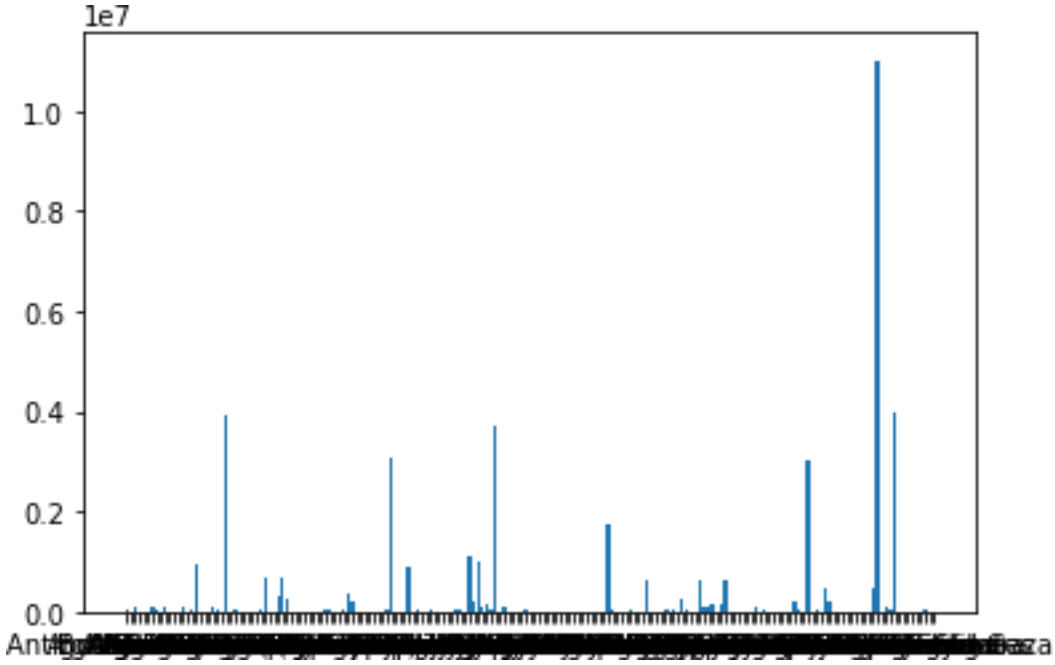
```
sns.heatmap(data[data["Country/Region"]=="India"]
[["Confirmed","Active","Recovered","Deaths"]])
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



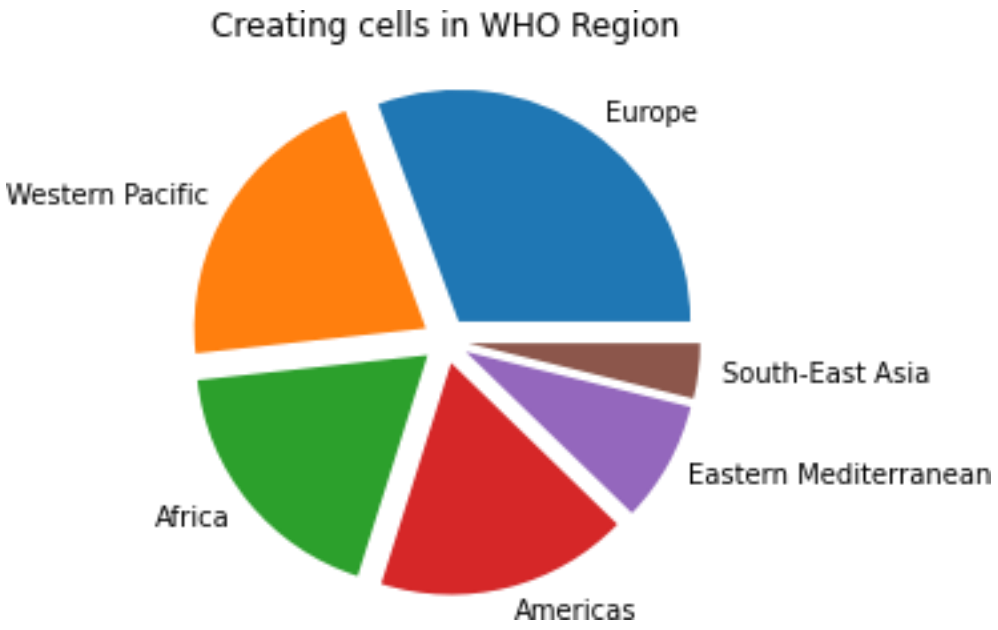

```
dx ={i:j.Deaths.sum() for i,j in data.groupby("Country/Region")}
plt.bar(dx.keys(),dx.values())
```

```
<BarContainer object of 187 artists>
```



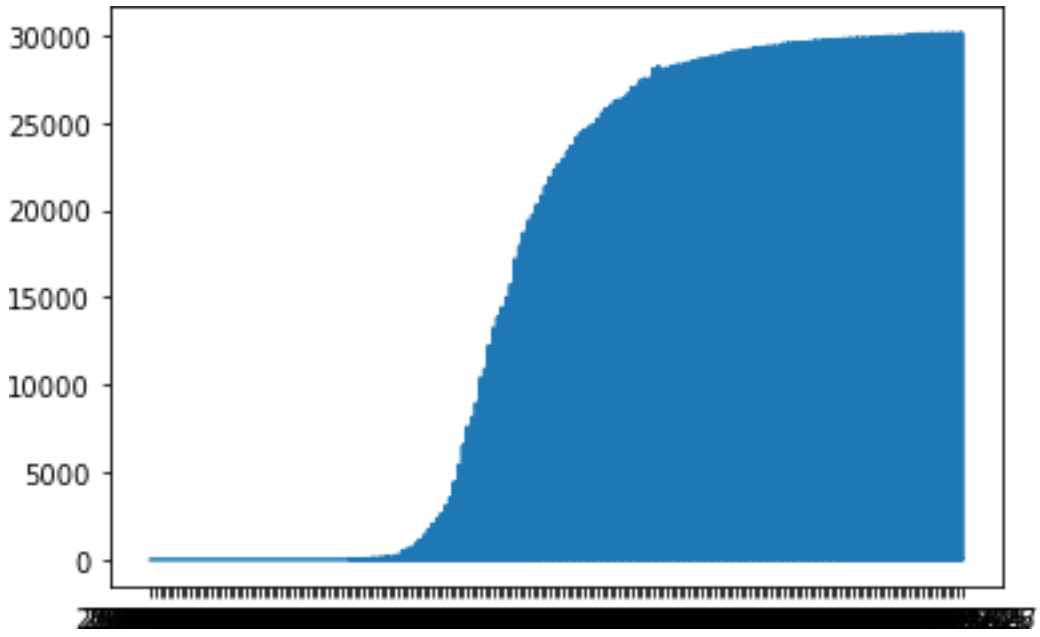

```
# create as many cells as required to do EDA and visulization. show
your creativity.
c=data["WHO Region"].value_counts().to_dict()
#plt.title('WHO Region')
```

```
plt.title("Creating cells in WHO Region")
plt.pie(c.values(),labels=c.keys(),explode=[0.1]*6)
plt.show()
```



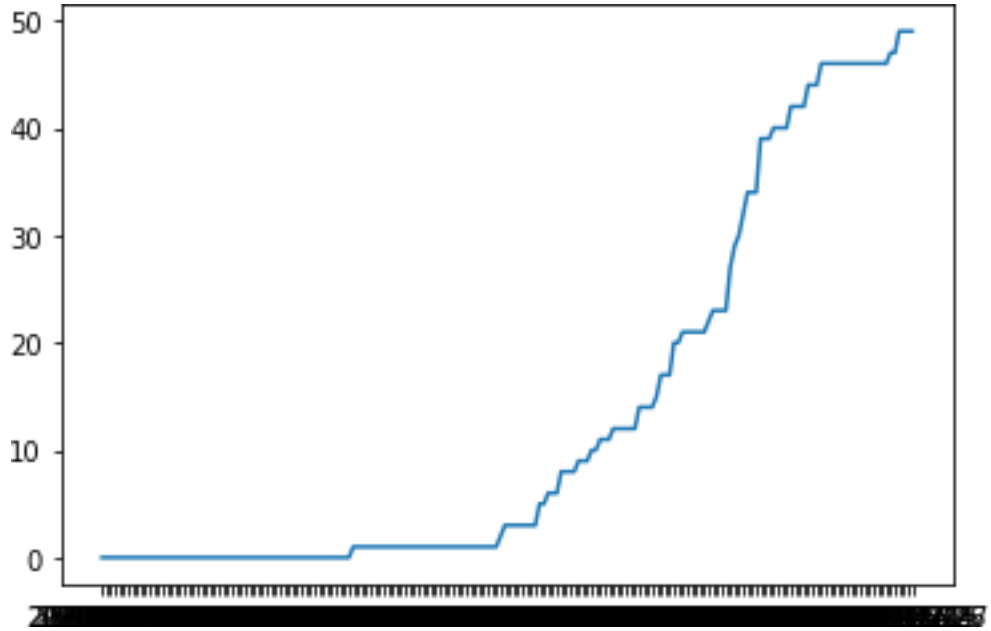

```
pt=data[data["Country/Region"]=="France"]
plt.plot(pt.Date, pt.Deaths)
```

```
[<matplotlib.lines.Line2D at 0x20f3b393d30>]
```



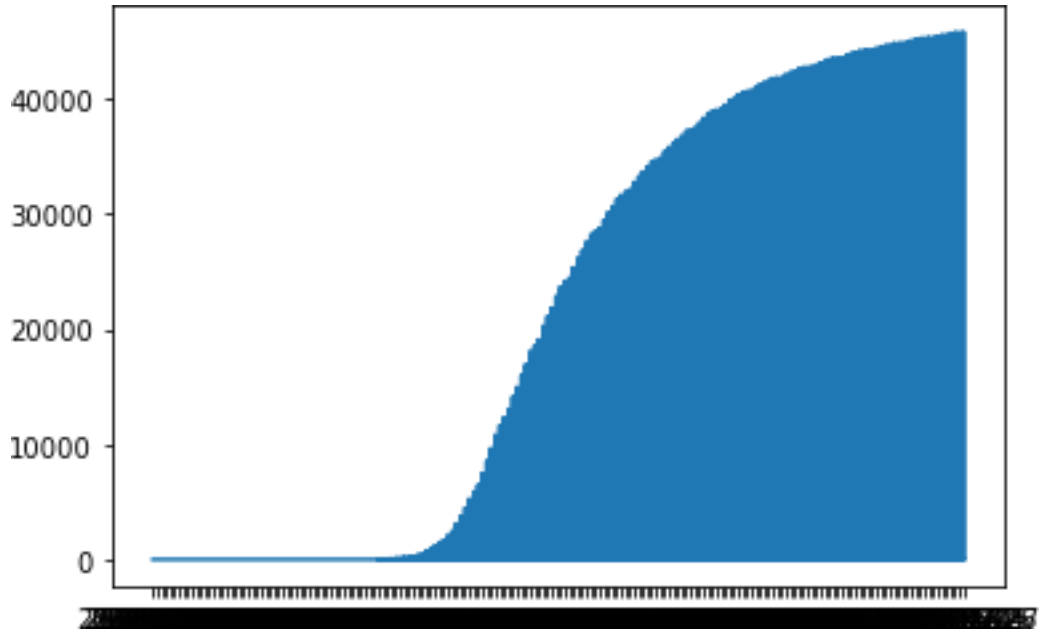

```
pt=data[data["Country/Region"]=="Gabon"]
plt.plot(pt.Date, pt.Deaths)
```

```
[<matplotlib.lines.Line2D at 0x20f3b5e4fa0>]
```



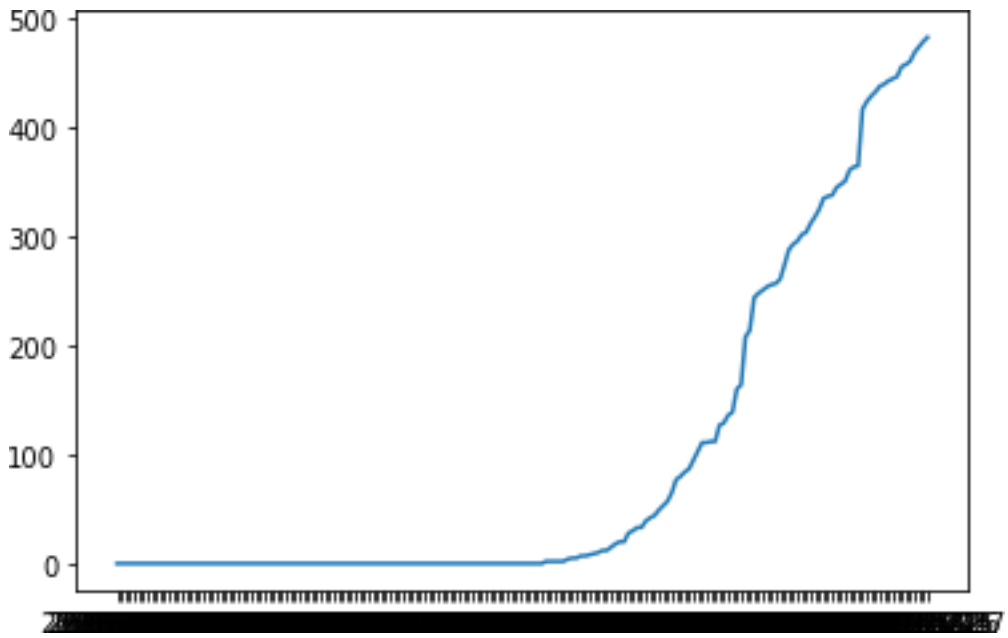

```
pt=data[data["Country/Region"]=="United Kingdom"]
plt.plot(pt.Date, pt.Deaths)
```

```
[<matplotlib.lines.Line2D at 0x20f39224490>]
```



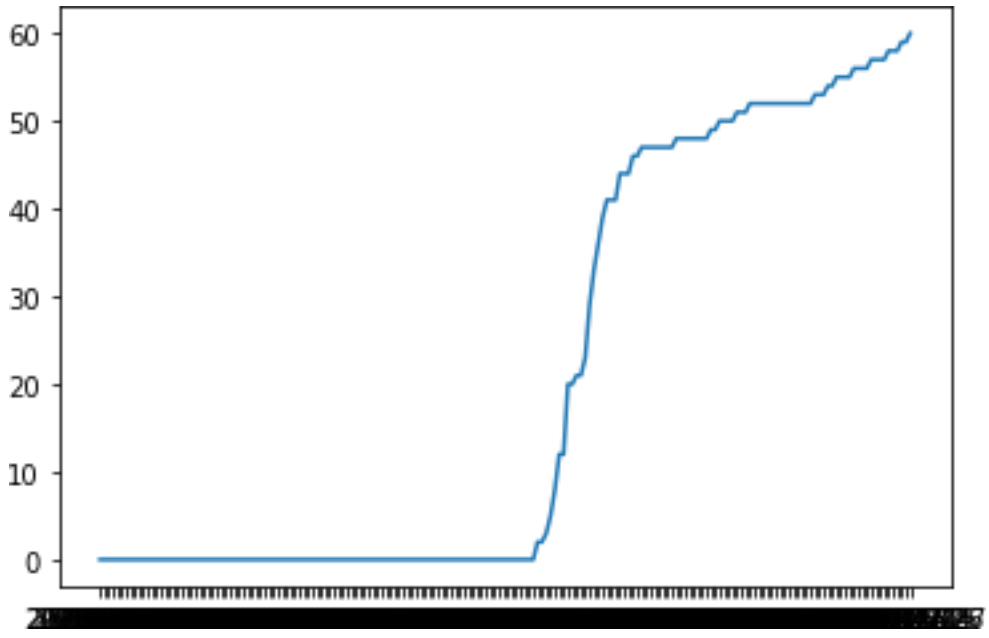

```
pt=data[data["Country/Region"]=="Yemen"]
plt.plot(pt.Date, pt.Deaths)
```

```
[<matplotlib.lines.Line2D at 0x20f390c4eb0>]
```



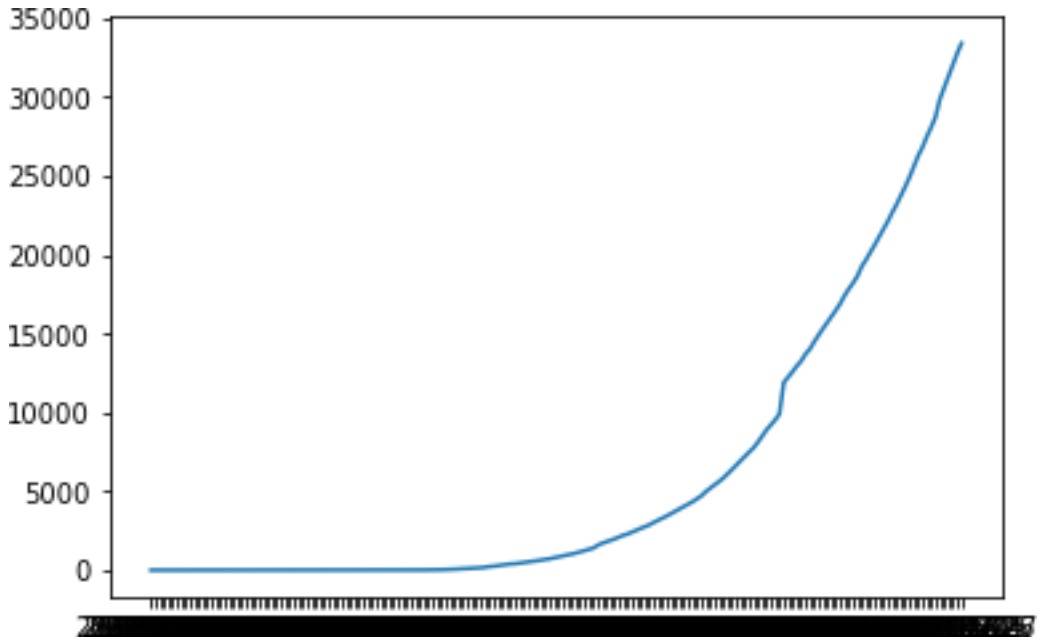

```
pt=data[data["Country/Region"]=="Tajikistan"]
plt.plot(pt.Date, pt.Deaths)
```

```
[<matplotlib.lines.Line2D at 0x20f3b8d79d0>]
```



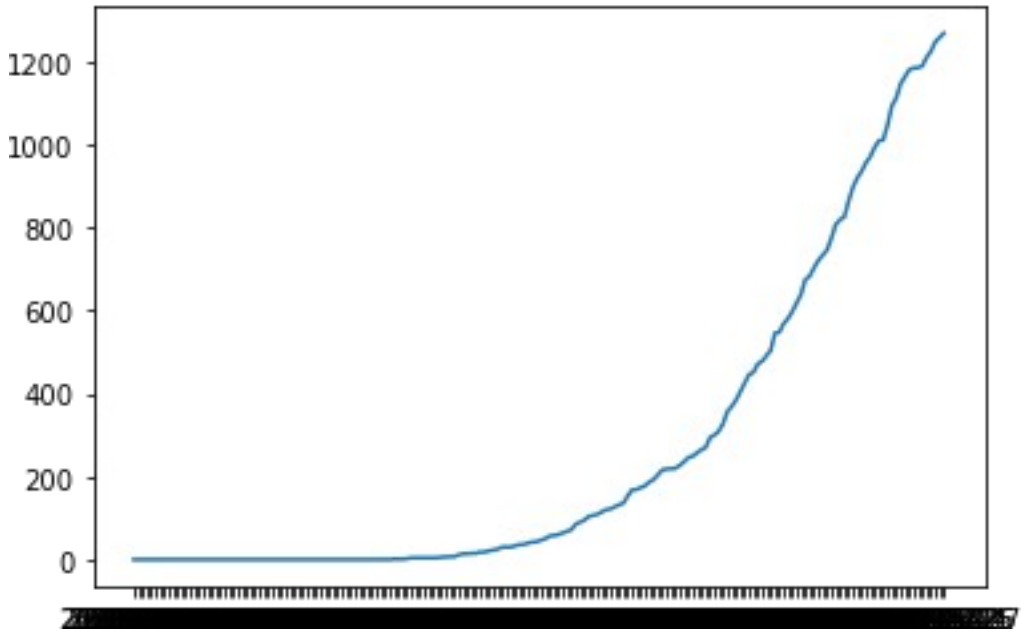

```
pt=data[data["Country/Region"]=="India"]
plt.plot(pt.Date, pt.Deaths)
```

```
[<matplotlib.lines.Line2D at 0x20f39c1ca60>]
```



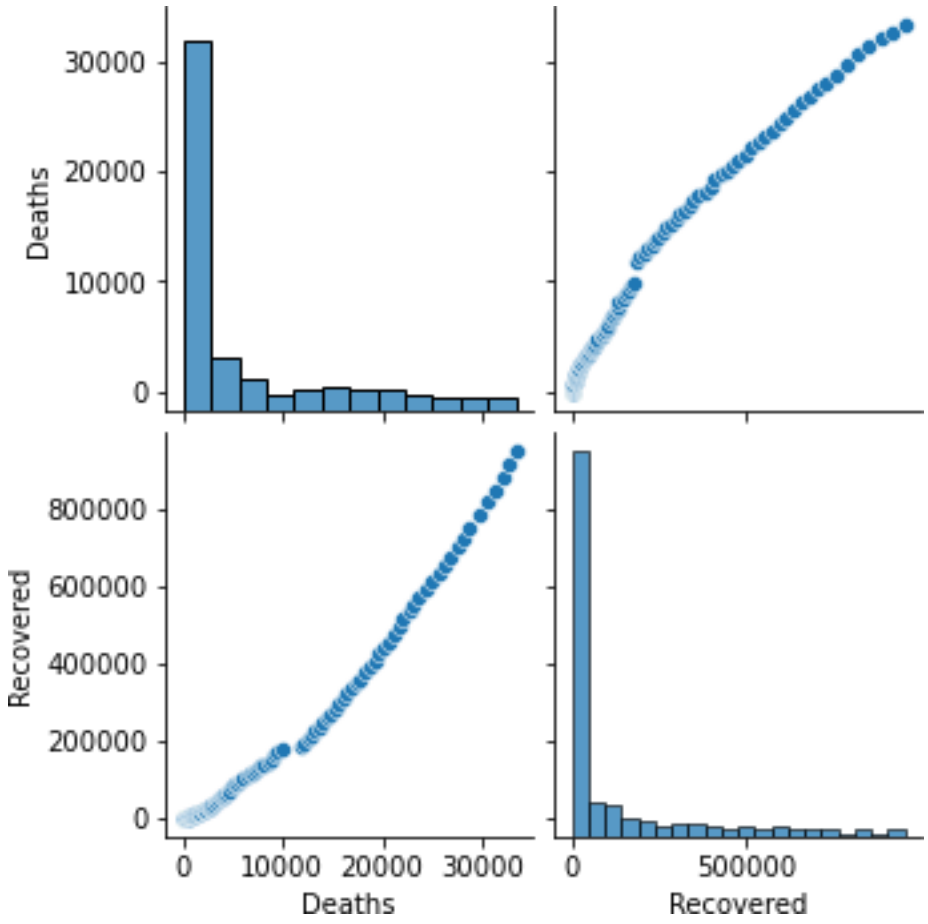

```
pt=data[data["Country/Region"]=="Afghanistan"]
plt.plot(pt.Date, pt.Deaths)
```

```
[<matplotlib.lines.Line2D at 0x20f3a1206a0>]
```



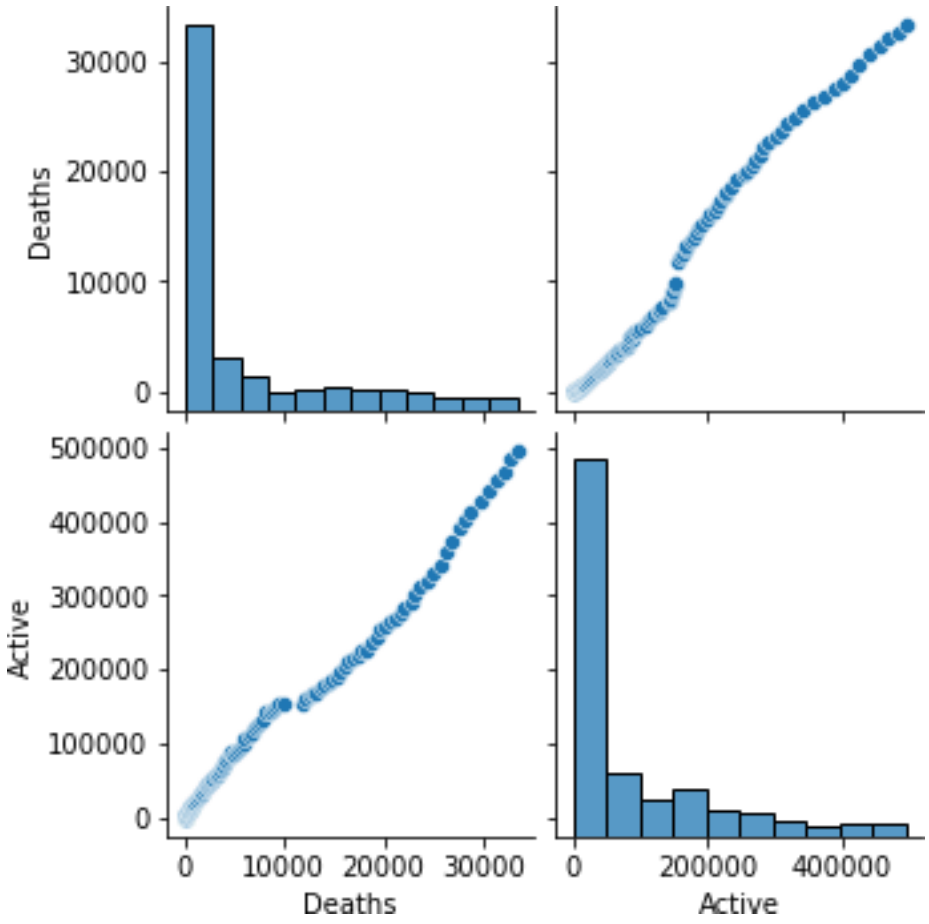

```
sns.pairplot(data[data["Country/Region"]=="India"][["Deaths",
"Recovered"]])
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



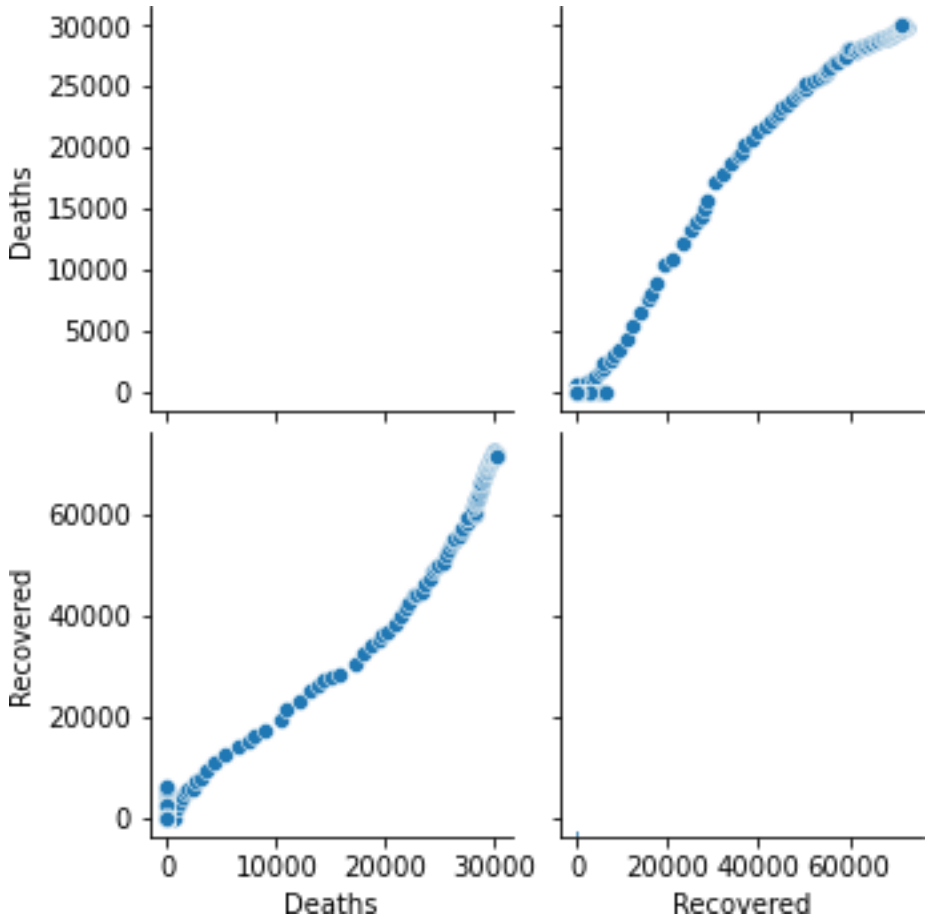

```
sns.pairplot(data[data["Country/Region"]=="India"][["Deaths",
"Active"]])
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



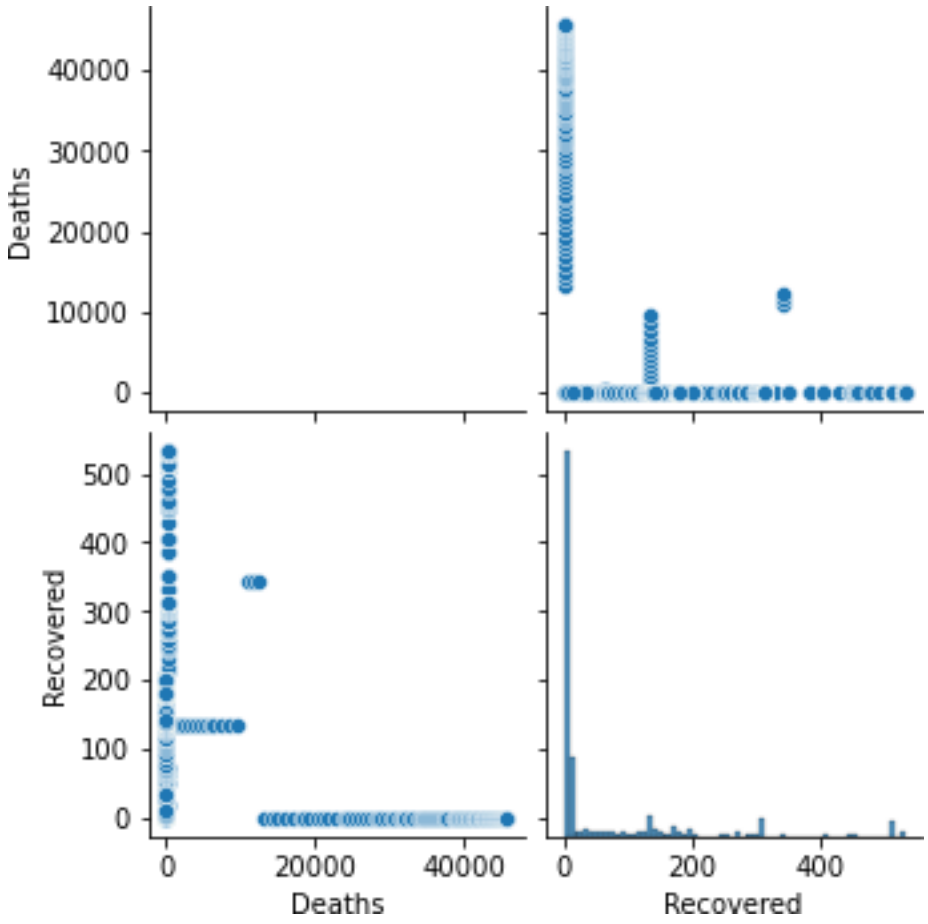

```
sns.pairplot(data[data["Country/Region"]=="France"][["Deaths",
"Recovered"]])
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```




```
sns.pairplot(data[data["Country/Region"]=="United Kingdom"][["Deaths",
"Recovered"]])
plt.show
```

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Statistics

```
#13 write a function to calculate standard deviation. (don't use
builtin methods)
dx=data[data["Country/Region"]=="India"].Deaths
np.sqrt(sum((dx-dx.mean())**2/len(dx)))
```

8976.169694119606

```
#14 write a function to calculate pearson's r for two numerical
columns.
dm=pd.DataFrame({"A":range(4), "B": [2*i for i in range(4)]})
dm["A"].corr(dm["B"])
dm.corr()
```

	A	B
A	1.0	1.0
B	1.0	1.0

```
#15 what is central limit theorem.
"""The mean of all the given samples of a population is the same as
the
mean of the population(approx) if the sample size is sufficiently
```

```
large
enough with a finite variation."""
```

```
"The mean of all the given samples of a population is the same as
the \nmean of the population(approx) if the sample size is
sufficiently large\nenough with a finite variation."
```

Hypothesis testing Critical Value Method

Consider this problem — H_0: $\mu \leq 350$ and H_1: $\mu > 350$

In case of a two-tailed test, you find the z-score of 0.975 in the z-table, since 0.975 was cumulative probability of UCV in that case. In this problem, what would be the cumulative probability of critical point in this example for the same significance level of 5%? Type your answer below cell

```
#16 Type your answer here.
""" The cumulative probability of the critical point (the total area
till that point) would be 0.950
&  significance level of 5% is 0.0475"""
```

```
" The cumulative probability of the critical point (the total area
till that point) would be 0.950\n&  significance level of 5% is
0.0475"
```

Using Critical Value Method

Consider this problem, H_0: $\mu \leq 350$ and H_1: $\mu > 350$

So, the Zc comes out to be 1.645. Now, find the critical value for the given Zc and make the decision to accept or reject the null hypothesis.

$\mu = 350$ $\sigma = 90$ N (Sample size) = 36 $\bar{x} = 370.16$

```
#17 type your answer here.
"""The critical value for the given Zc is 374.67 and
The decision to accept or reject the null hypothesis is Rejected"""
```

```
"The critical value for the given Zc is 374.67 and \nThe decision to
accept or reject the null hypothesis is Rejected"
```

Let's solve the following problem on how to make a decision about any hypothesis using the p-value method.

You are working as a data analyst at an auditing firm. A manufacturer claims that the average life of its product is 36 months. An auditor selects a sample of 49 units of the product, and calculates the average life to be 34.5 months. The population standard deviation is 4 months. Test the manufacturer's claim at 3% significance level using the p-value method.

First, formulate the hypotheses for this two-tailed test, which would be:

H_0: μ = 36 months and H_1: $\mu \neq$ 36 months

Now, you need to follow the three steps to find the p-value and make a decision.

STEP 1 - Calculate the value of z-score for the sample mean point on the distribution. Calculate z-score for sample mean ($\bar{x}$) = 34.5 months.

STEP 2 - Calculate the p-value from the cumulative probability for the calculated z-score using the z-table.(Hint z-score of -2.62)

Hint: The sample mean is on the left side of the distribution and it is a two-tailed test.

```
#18 type your answer
"""STEP 1 --> the value of z-score for the sample mean point on the
distribution.
z-score for sample mean (¯x) = 34.5 months is (34.5 - 36)/(4/√49) = (-
1.5) * 7/4 = -2.62.
STEP 2 --> The value in the z-table corresponding to -2.6 on the
vertical
axis and 0.02 on the horizontal axis is 0.0044.
The p-value would be 2 * 0.0044 = 0.0088."""
```

```
"STEP 1 --> the value of z-score for the sample mean point on the
distribution.\nz-score for sample mean (¯x) = 34.5 months is (34.5 -
36)/(4/√49) = (-1.5) * 7/4 = -2.62.\nSTEP 2 --> The value in the z-
table corresponding to -2.6 on the vertical\naxis and 0.02 on the
horizontal axis is 0.0044.\nThe p-value would be 2 * 0.0044 = 0.0088."
```

Linear Regression Assignment

In this assignment will do end to end linear regression project to predict car prices.Data and Data Dictionary is given along with the Jupyter notebook

Steps Involved

""" Data Importing and Understanding Data Inspection

```
Data Cleaning
    Delete useless columns
    Delete all Unique columns
    Impute missing values

Data Visualization
    Plot distribution graphs for numerical data
    plot categorical data

Data analysis
    Univariate Analysis
    Bivariate Analysis
    Create heatmaps and pairplots

Understanding Business insights
    find some business related insights

Data preprocessing
    Labelencoding,One hot encoding if needed
    Standardization, normalization if needed

Creating Model and training model
    implement from scratch
    Use sklearn library

Validate your model with right metrics
    R2 score is used for final grading"""
```

Target Variable is car price in the data

Rubrics: Each smection will fetch 10 marks 10 Marks is for commenting code properly more than 0.80 R2score will fetch good marks (Test set) Submit assignment in time

import libraries

Data Importing and Understanding

Data Cleaning

Data Visualization

Data analysis

Understanding Business insights

Data preprocessing

Feature Engineering

Creating Model and training model

Validating model with right metrics

```
#importing the libraries
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import os

cars_data = pd.read_csv(r"F:\car_price.csv")
cars_data #Read Cars_data
```

```
     car_ID  symboling                   CarName fueltype
aspiration  \
0         1          3        alfa-romero giulia      gas
std
1         2          3       alfa-romero stelvio      gas
std
2         3          1  alfa-romero Quadrifoglio      gas
std
3         4          2               audi 100 ls      gas
std
4         5          2                audi 100ls      gas
std
..      ...        ...                       ...      ...
...
200     201         -1           volvo 145e (sw)      gas
std
201     202         -1               volvo 144ea      gas
turbo
202     203         -1               volvo 244dl      gas
std
```

```
203      204          -1                  volvo 246    diesel      turbo

204      205          -1                volvo 264gl       gas      turbo


    doornumber      carbody drivewheel enginelocation  wheelbase  ...
\
0          two  convertible        rwd           front       88.6  ...

1          two  convertible        rwd           front       88.6  ...

2          two    hatchback        rwd           front       94.5  ...

3         four        sedan        fwd           front       99.8  ...

4         four        sedan        4wd           front       99.4  ...

..         ...          ...        ...             ...        ...  ...

200       four        sedan        rwd           front      109.1  ...

201       four        sedan        rwd           front      109.1  ...

202       four        sedan        rwd           front      109.1  ...

203       four        sedan        rwd           front      109.1  ...

204       four        sedan        rwd           front      109.1  ...


     enginesize  fuelsystem  boreratio  stroke compressionratio
horsepower \
0           130        mpfi       3.47    2.68              9.0
111
1           130        mpfi       3.47    2.68              9.0
111
2           152        mpfi       2.68    3.47              9.0
154
3           109        mpfi       3.19    3.40             10.0
102
4           136        mpfi       3.19    3.40              8.0
115
..          ...         ...        ...     ...              ...
...
200         141        mpfi       3.78    3.15              9.5
114
201         141        mpfi       3.78    3.15              8.7
160
```

```
202          173         mpfi        3.58     2.87              8.8
134
203          145          idi        3.01     3.40             23.0
106
204          141         mpfi        3.78     3.15              9.5
114

     peakrpm citympg  highwaympg    price
0       5000       21          27  13495.0
1       5000       21          27  16500.0
2       5000       19          26  16500.0
3       5500       24          30  13950.0
4       5500       18          22  17450.0
..       ...      ...         ...      ...
200     5400       23          28  16845.0
201     5300       19          25  19045.0
202     5500       18          23  21485.0
203     4800       26          27  22470.0
204     5400       19          25  22625.0

[205 rows x 26 columns]
```

```
cars_data.head() #Lets see top 5 in cars_data
```

```
   car_ID  symboling                   CarName fueltype aspiration
doornumber  \
0       1          3        alfa-romero giulia      gas        std
two
1       2          3       alfa-romero stelvio      gas        std
two
2       3          1  alfa-romero Quadrifoglio      gas        std
two
3       4          2               audi 100 ls      gas        std
four
4       5          2                audi 100ls      gas        std
four

       carbody drivewheel enginelocation  wheelbase  ...
enginesize  \
0  convertible        rwd          front       88.6  ...          130

1  convertible        rwd          front       88.6  ...          130

2    hatchback        rwd          front       94.5  ...          152

3        sedan        fwd          front       99.8  ...          109

4        sedan        4wd          front       99.4  ...          136
```

```
   fuelsystem  boreratio  stroke  compressionratio horsepower  peakrpm
citympg  \
0        mpfi       3.47    2.68               9.0        111     5000
21
1        mpfi       3.47    2.68               9.0        111     5000
21
2        mpfi       2.68    3.47               9.0        154     5000
19
3        mpfi       3.19    3.40              10.0        102     5500
24
4        mpfi       3.19    3.40               8.0        115     5500
18

   highwaympg    price
0          27  13495.0
1          27  16500.0
2          26  16500.0
3          30  13950.0
4          22  17450.0

[5 rows x 26 columns]
```

```
cars_data.describe() #Lets Describe
```

```
           car_ID   symboling   wheelbase   carlength    carwidth
carheight  \
count  205.000000  205.000000  205.000000  205.000000  205.000000
205.000000
mean   103.000000    0.834146   98.756585  174.049268   65.907805
53.724878
std     59.322565    1.245307    6.021776   12.337289    2.145204
2.443522
min      1.000000   -2.000000   86.600000  141.100000   60.300000
47.800000
25%     52.000000    0.000000   94.500000  166.300000   64.100000
52.000000
50%    103.000000    1.000000   97.000000  173.200000   65.500000
54.100000
75%    154.000000    2.000000  102.400000  183.100000   66.900000
55.500000
max    205.000000    3.000000  120.900000  208.100000   72.300000
59.800000

        curbweight  enginesize   boreratio      stroke
compressionratio  \
count   205.000000  205.000000  205.000000  205.000000
205.000000
mean   2555.565854  126.907317    3.329756    3.255415
10.142537
std     520.680204   41.642693    0.270844    0.313597
```

```
3.972040
min      1488.000000    61.000000     2.540000     2.070000
7.000000
25%      2145.000000    97.000000     3.150000     3.110000
8.600000
50%      2414.000000   120.000000     3.310000     3.290000
9.000000
75%      2935.000000   141.000000     3.580000     3.410000
9.400000
max      4066.000000   326.000000     3.940000     4.170000
23.000000
```

	horsepower	peakrpm	citympg	highwaympg	price
count	205.000000	205.000000	205.000000	205.000000	205.000000
mean	104.117073	5125.121951	25.219512	30.751220	13276.710571
std	39.544167	476.985643	6.542142	6.886443	7988.852332
min	48.000000	4150.000000	13.000000	16.000000	5118.000000
25%	70.000000	4800.000000	19.000000	25.000000	7788.000000
50%	95.000000	5200.000000	24.000000	30.000000	10295.000000
75%	116.000000	5500.000000	30.000000	34.000000	16503.000000
max	288.000000	6600.000000	49.000000	54.000000	45400.000000

```
cars_data.info() #Cars_data info

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 205 entries, 0 to 204
Data columns (total 26 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   car_ID             205 non-null    int64
 1   symboling          205 non-null    int64
 2   CarName            205 non-null    object
 3   fueltype           205 non-null    object
 4   aspiration         205 non-null    object
 5   doornumber         205 non-null    object
 6   carbody            205 non-null    object
 7   drivewheel         205 non-null    object
 8   enginelocation     205 non-null    object
 9   wheelbase          205 non-null    float64
 10  carlength          205 non-null    float64
 11  carwidth           205 non-null    float64
 12  carheight          205 non-null    float64
 13  curbweight         205 non-null    int64
 14  enginetype         205 non-null    object
 15  cylindernumber     205 non-null    object
 16  enginesize         205 non-null    int64
 17  fuelsystem         205 non-null    object
 18  boreratio          205 non-null    float64
 19  stroke             205 non-null    float64
 20  compressionratio   205 non-null    float64
```

```
 21   horsepower        205 non-null    int64
 22   peakrpm           205 non-null    int64
 23   citympg           205 non-null    int64
 24   highwaympg        205 non-null    int64
 25   price             205 non-null    float64
dtypes: float64(8), int64(8), object(10)
memory usage: 41.8+ KB
```

Data Cleaning and Preparation

```
#Splitting company name from CarName column
CompanyName = cars_data["CarName"].apply(lambda x : x.split(" ")[0])
cars_data.insert(3,"CompanyName",CompanyName)
cars_data.drop(["CarName"],axis=1,inplace=True)
cars_data.head()
```

```
   car_ID  symboling  CompanyName fueltype aspiration doornumber
carbody  \
0       1          3  alfa-romero      gas        std        two
convertible
1       2          3  alfa-romero      gas        std        two
convertible
2       3          1  alfa-romero      gas        std        two
hatchback
3       4          2         audi      gas        std       four
sedan
4       5          2         audi      gas        std       four
sedan

  drivewheel enginelocation  wheelbase  ...  enginesize  fuelsystem  \
0        rwd          front       88.6  ...         130        mpfi
1        rwd          front       88.6  ...         130        mpfi
2        rwd          front       94.5  ...         152        mpfi
3        fwd          front       99.8  ...         109        mpfi
4        4wd          front       99.4  ...         136        mpfi

   boreratio  stroke compressionratio horsepower  peakrpm citympg
highwaympg  \
0       3.47    2.68              9.0        111     5000       21
27
1       3.47    2.68              9.0        111     5000       21
27
2       2.68    3.47              9.0        154     5000       19
26
3       3.19    3.40             10.0        102     5500       24
30
4       3.19    3.40              8.0        115     5500       18
22

     price
```

```
0  13495.0
1  16500.0
2  16500.0
3  13950.0
4  17450.0

[5 rows x 26 columns]
```

```
cars_data.CompanyName.unique() #Lets see Unique in CompanyName using
unique
```

```
array(["alfa-romero", "audi", "bmw", "chevrolet", "dodge", "honda",
       "isuzu", "jaguar", "maxda", "mazda", "buick", "mercury",
       "mitsubishi", "Nissan", "nissan", "peugeot", "plymouth",
"porsche",
       "porcshce", "renault", "saab", "subaru", "toyota", "toyouta",
       "vokswagen", "volkswagen", "vw", "volvo"], dtype=object)
```

```
#replacing names
cars_data.CompanyName = cars_data.CompanyName.str.lower()
def replace_name(a,b):
    cars_data.CompanyName.replace(a,b,inplace=True)

replace_name("maxda","mazda")
replace_name("porcshce","porsche")
replace_name("toyouta","toyota")
replace_name("vokswagen","volkswagen")
replace_name("vw","volkswagen")

cars_data.CompanyName.unique()
```

```
array(["alfa-romero", "audi", "bmw", "chevrolet", "dodge", "honda",
       "isuzu", "jaguar", "mazda", "buick", "mercury", "mitsubishi",
       "nissan", "peugeot", "plymouth", "porsche", "renault", "saab",
       "subaru", "toyota", "volkswagen", "volvo"], dtype=object)
```

```
#Checking for duplicates
cars_data.loc[cars_data.duplicated()]
```

```
Empty DataFrame
Columns: [car_ID, symboling, CompanyName, fueltype, aspiration,
doornumber, carbody, drivewheel, enginelocation, wheelbase, carlength,
carwidth, carheight, curbweight, enginetype, cylindernumber,
enginesize, fuelsystem, boreratio, stroke, compressionratio,
horsepower, peakrpm, citympg, highwaympg, price]
Index: []

[0 rows x 26 columns]
```

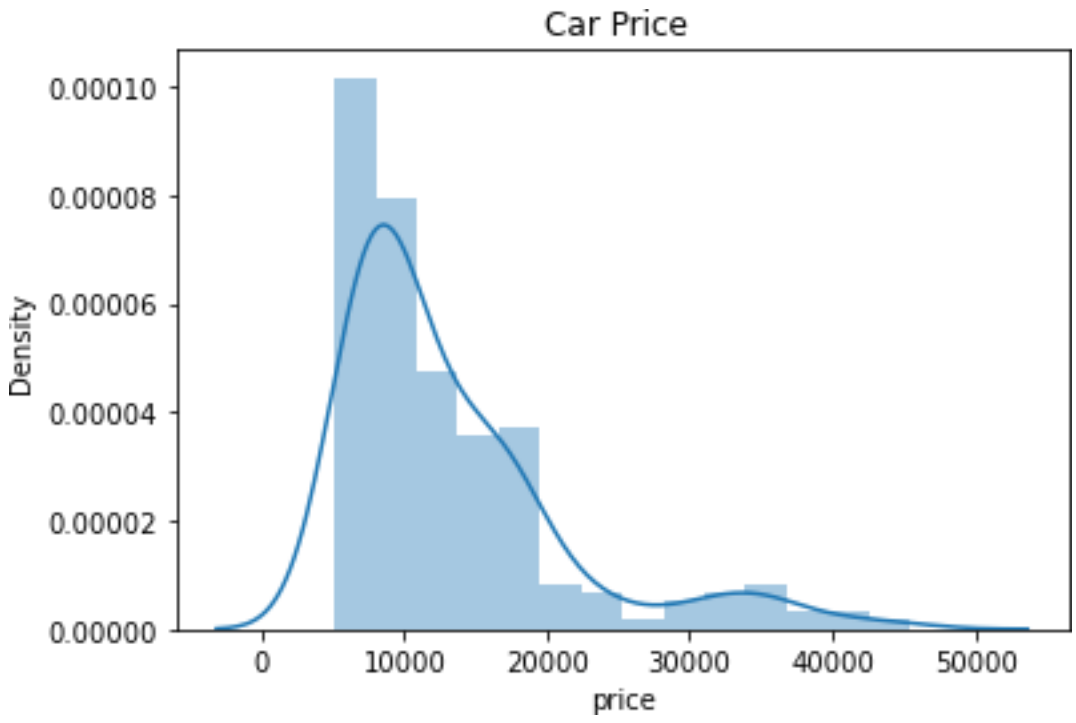
```
cars_data.columns
```

```
Index(["car_ID", "symboling", "CompanyName", "fueltype", "aspiration",
       "doornumber", "carbody", "drivewheel", "enginelocation",
"wheelbase",
       "carlength", "carwidth", "carheight", "curbweight",
"enginetype",
       "cylindernumber", "enginesize", "fuelsystem", "boreratio",
"stroke",
       "compressionratio", "horsepower", "peakrpm", "citympg",
"highwaympg",
       "price"],
      dtype="object")
```

Data Visualization

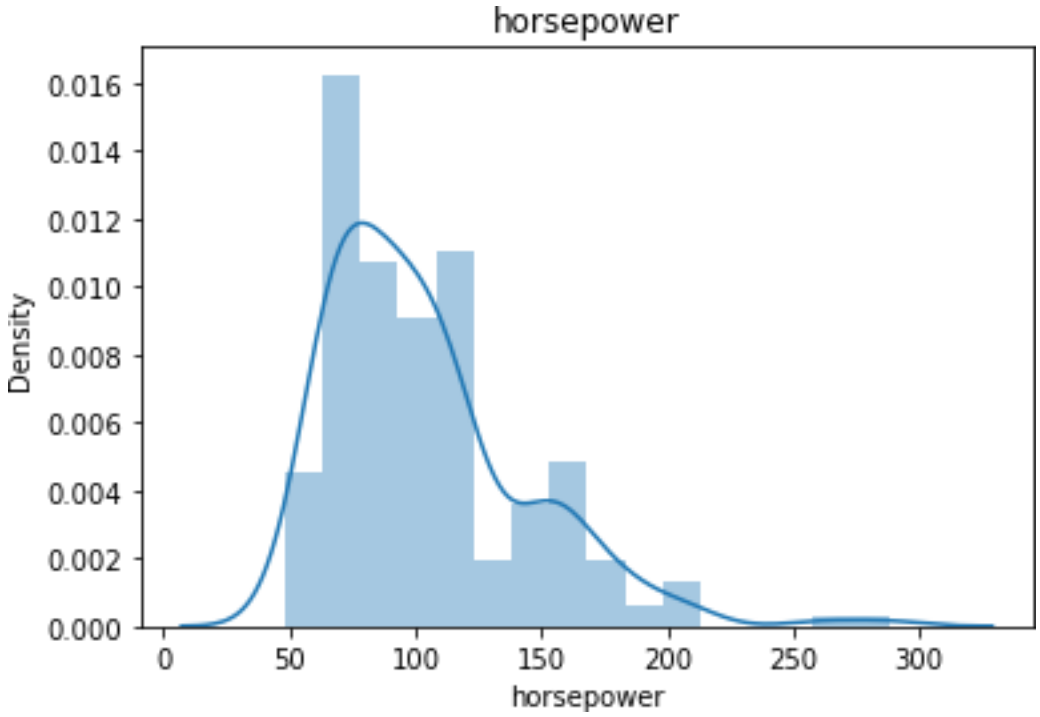
#Plot distribution graphs for numerical data

```
#"""plt.figure(figsize=(25,10))
#plt.subplot(1,1,1)"""

#Distribution plot - car price
plt.title("Car Price ")
sns.distplot(cars_data.price)
plt.show()
```

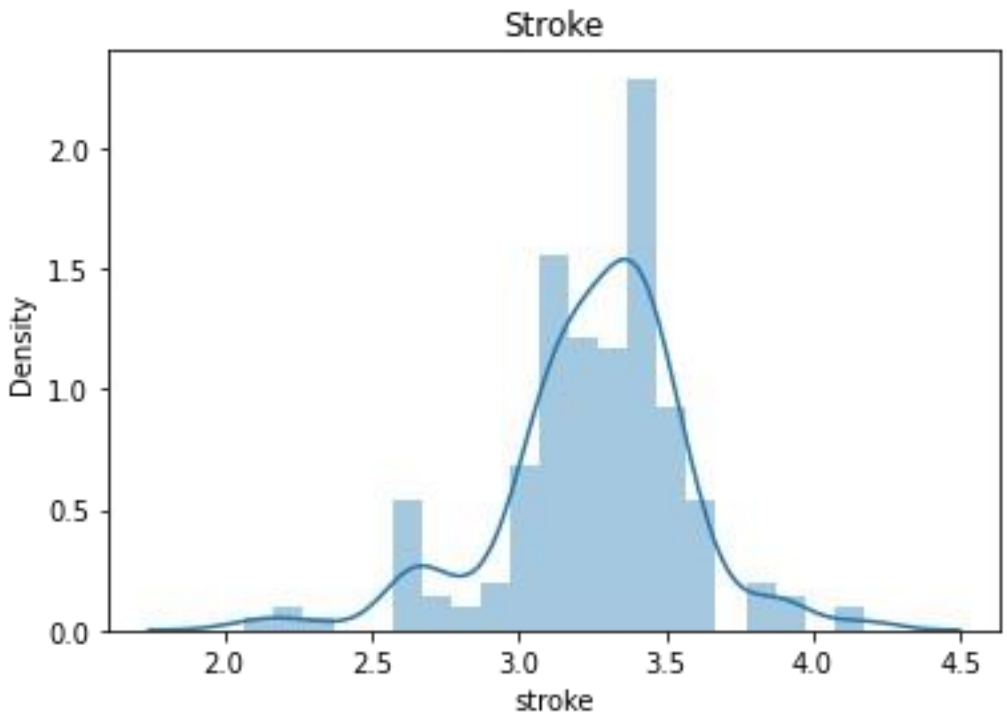



```
#Distribution plot - Horse Power
plt.title("horsepower")
```

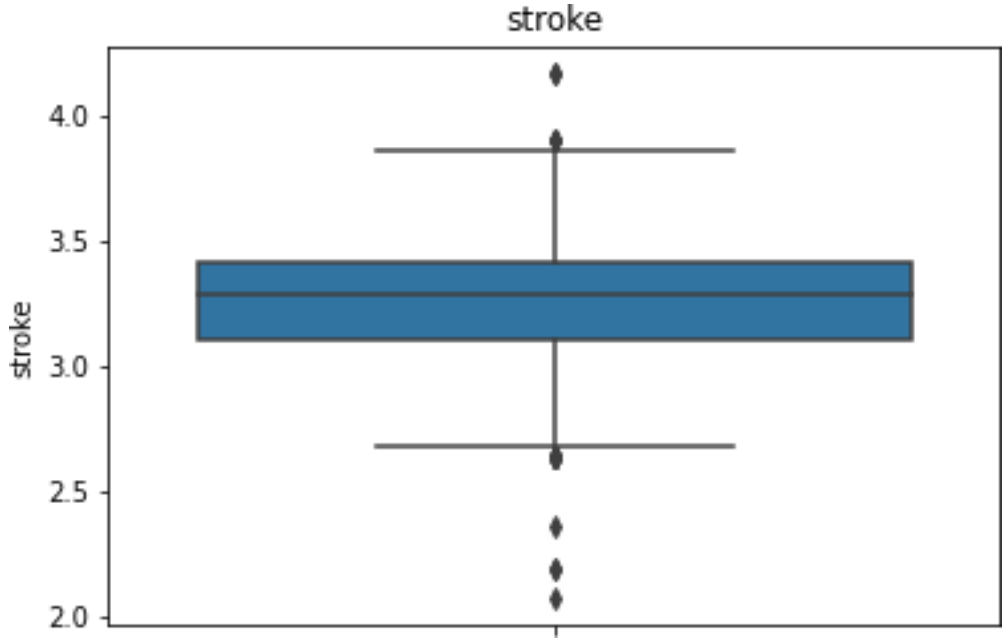
```
sns.distplot(cars_data.horsepower)
plt.show()
```




```
#Distribution plot - Stroke
plt.title("Stroke")
sns.distplot(cars_data.stroke)
plt.show()
```

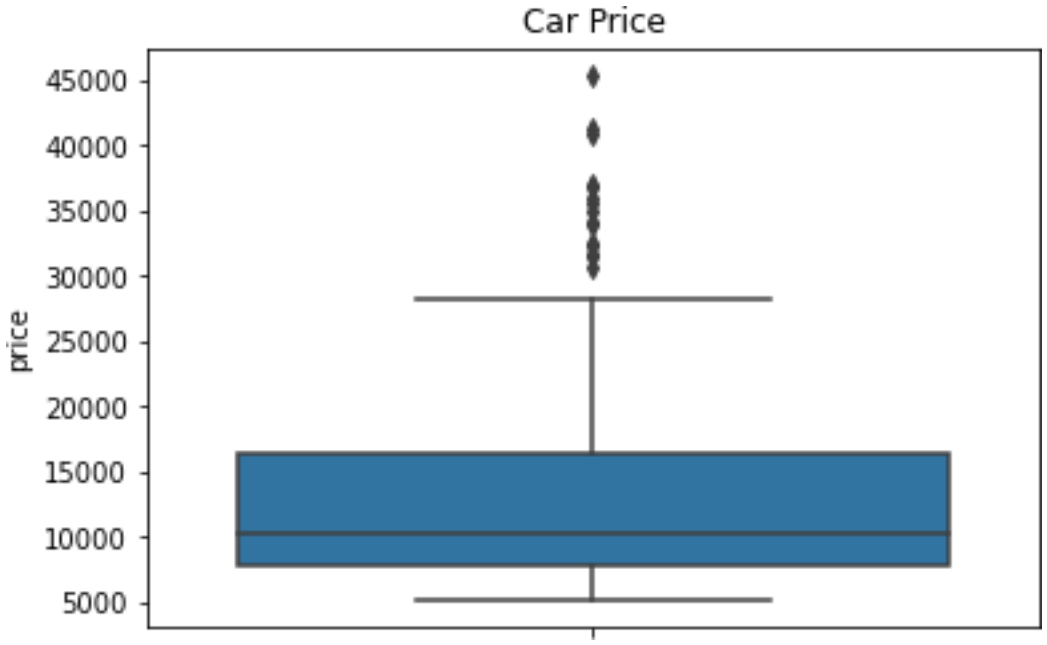



```
#BOX plot - Stroke
plt.title("stroke")
sns.boxplot(y=cars_data.stroke)
plt.show()
```




```
#Box plot - Car price
plt.title("Car Price")
```

```
sns.boxplot(y=cars_data.price)
plt.show()
```

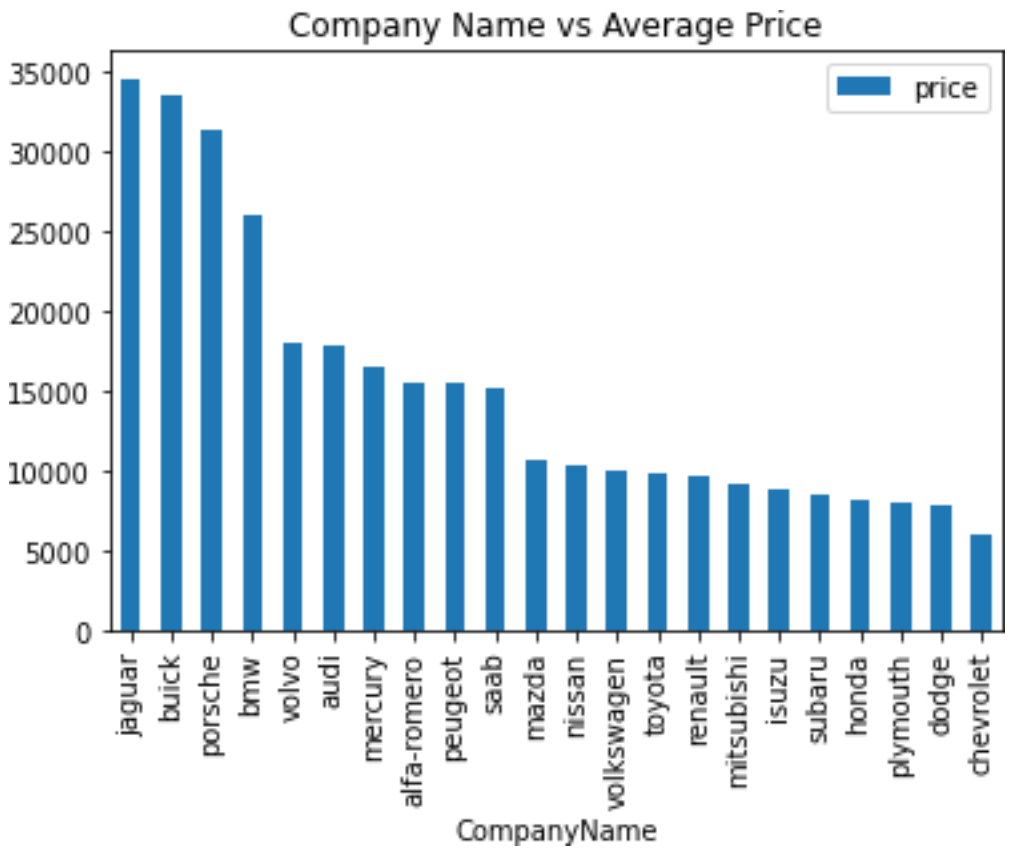



```
#lets describe cars_data
cars_data["price"].describe()
```

```
count      205.000000
mean     13276.710571
std       7988.852332
min       5118.000000
25%       7788.000000
50%      10295.000000
75%      16503.000000
max      45400.000000
Name: price, dtype: float64
```

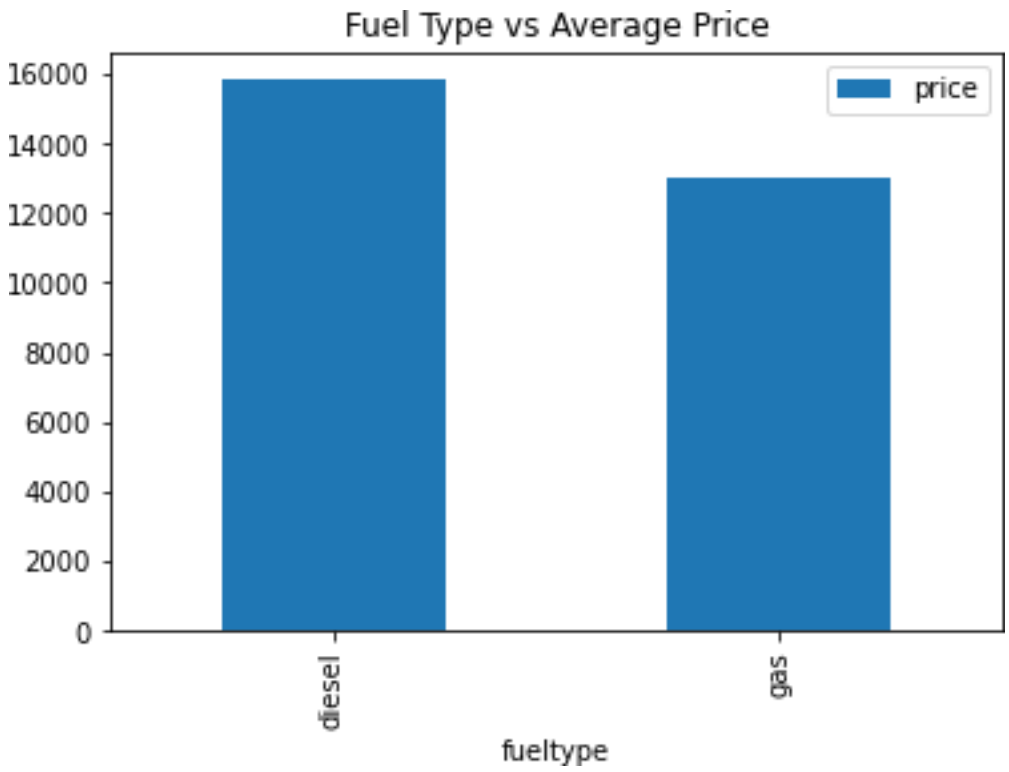
plot categorical data

```
x = pd.DataFrame(cars_data.groupby(["CompanyName"])
["price"].mean().sort_values(ascending = False))
x.plot.bar()
plt.title("Company Name vs Average Price")
plt.show()
```



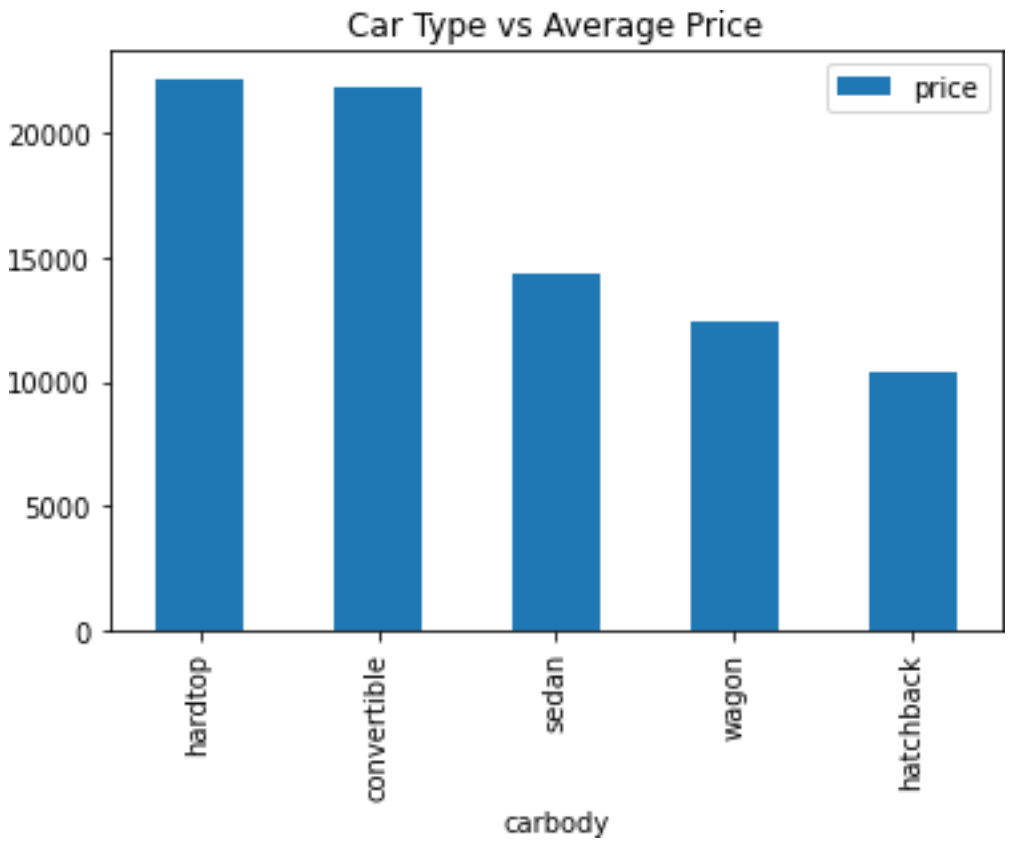
as we observe bar plot: Jaguar,Buick and porsche are having highest average price.

```
x = pd.DataFrame(cars_data.groupby(["fueltype"])
["price"].mean().sort_values(ascending = False))
x.plot.bar()
plt.title("Fuel Type vs Average Price")
plt.show()
```



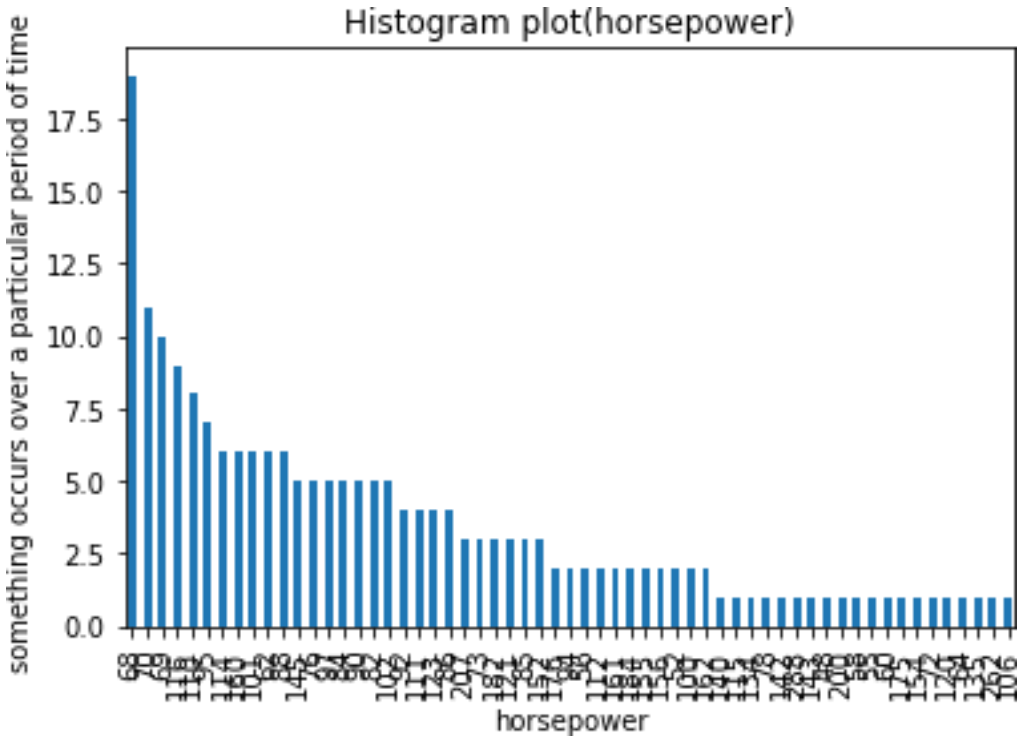
we can clearly say diesel has higher average price than gas.

```
x = pd.DataFrame(cars_data.groupby(["carbody"])
["price"].mean().sort_values(ascending = False))
x.plot.bar()
plt.title("Car Type vs Average Price")
plt.show()
```



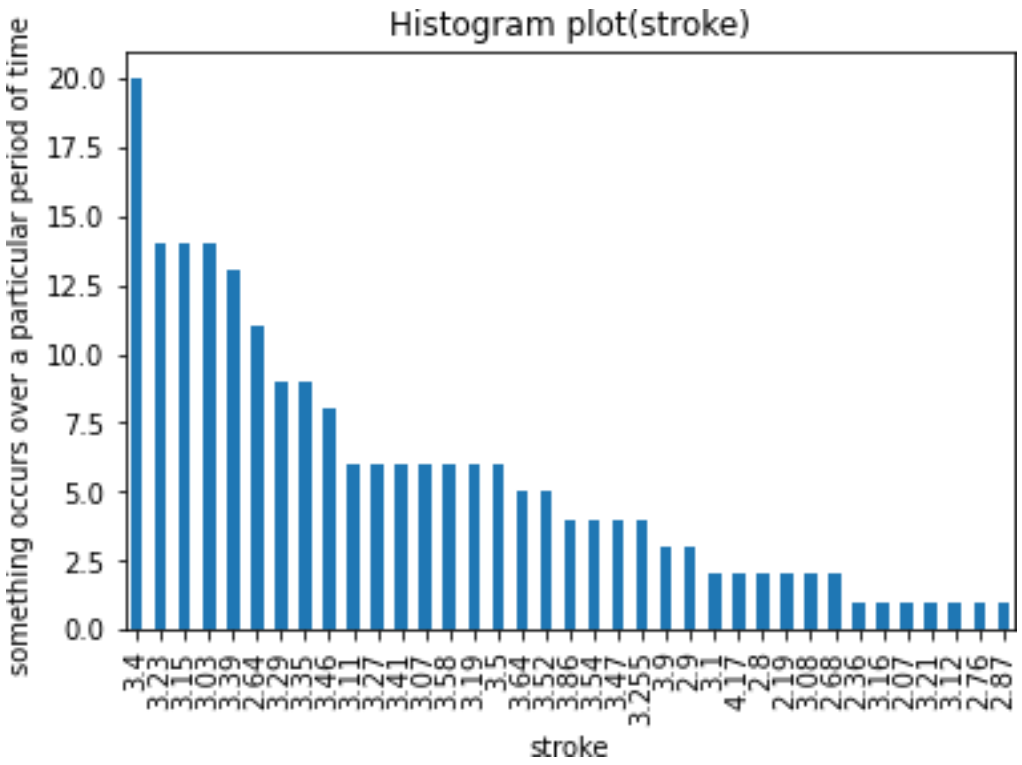
#hardtop and convertible have higher average price.

```
x = cars_data.horsepower.value_counts().plot(kind="bar")
plt.title("Histogram plot(horsepower)")
x.set(xlabel ="horsepower", ylabel="something occurs over a particular
period of time ")

plt.show()
```



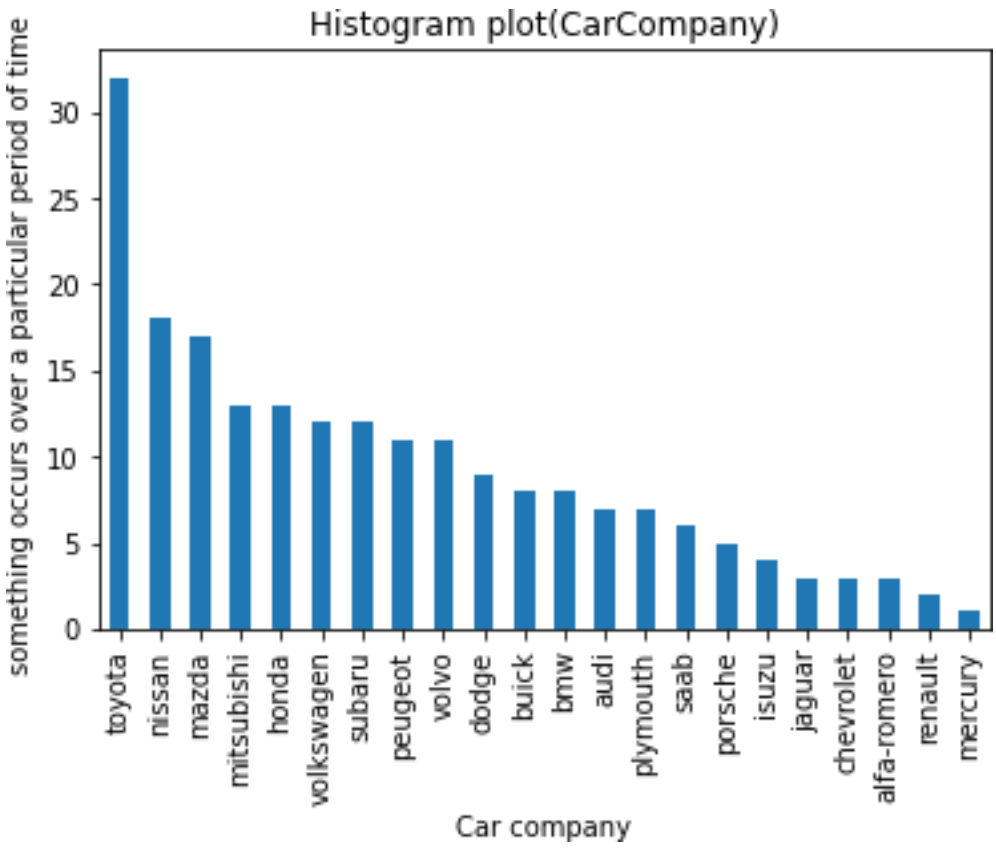
#68 is more favorable to everyone as we observe above histogram plot

```
x = cars_data.stroke.value_counts().plot(kind="bar")
plt.title("Histogram plot(stroke)")
x.set(xlabel ="stroke", ylabel="something occurs over a particular period of time ")

plt.show()
```



#3.4 is more favorable stroke

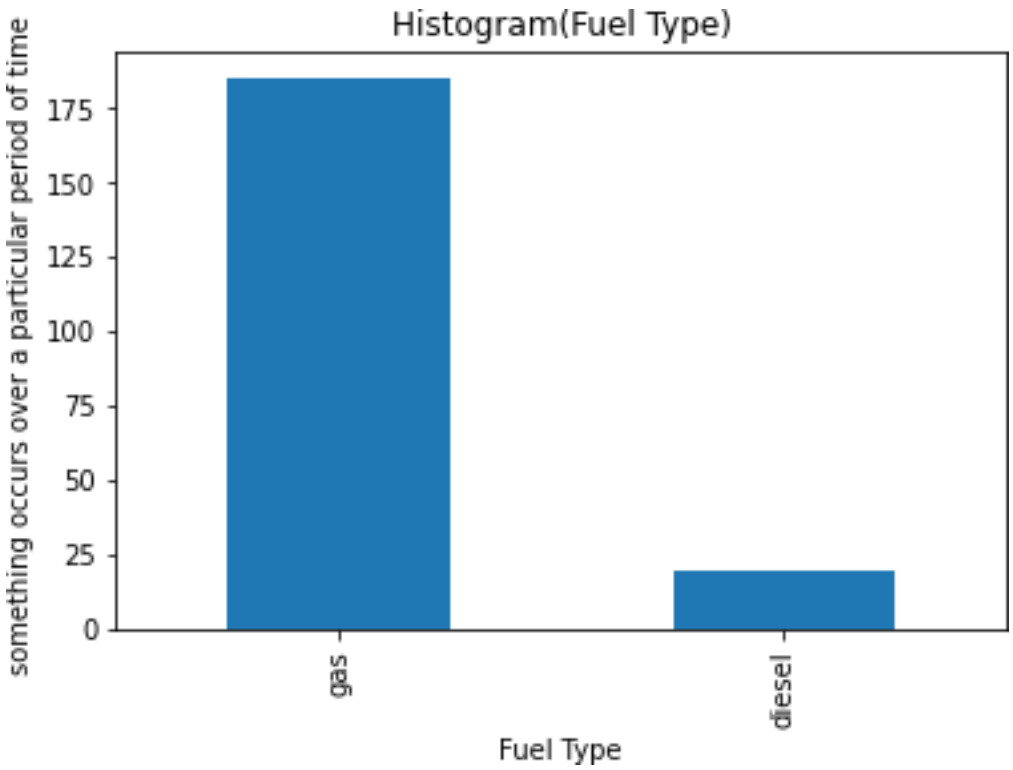
```
x = cars_data.CompanyName.value_counts().plot(kind="bar")
plt.title("Histogram plot(CarCompany)")
x.set(xlabel ="Car company", ylabel="something occurs over a
particular period of time ")

plt.show()
```




```
#toyota is more favorable in car company
```

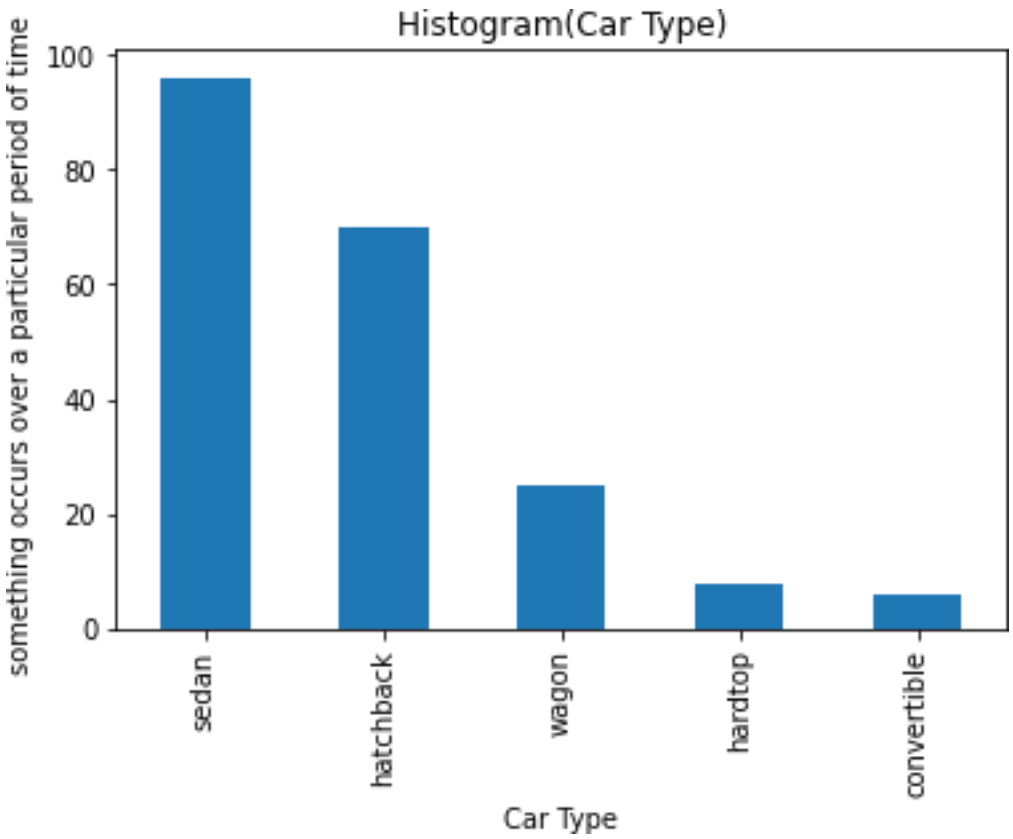
```
x = cars_data.fueltype.value_counts().plot(kind="bar")
plt.title("Histogram(Fuel Type)")
x.set(xlabel ="Fuel Type", ylabel="something occurs over a particular
period of time ")
```

```
[Text(0.5, 0, "Fuel Type"),
 Text(0, 0.5, "something occurs over a particular period of time ")]
```



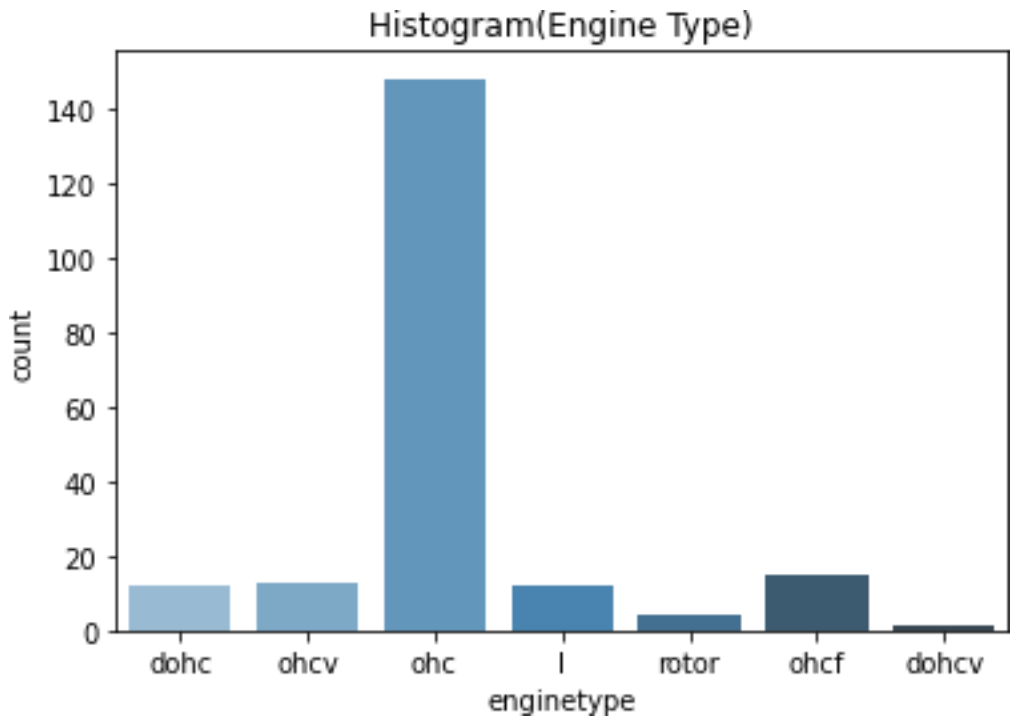

```
x = cars_data.carbody.value_counts().plot(kind="bar")
plt.title("Histogram(Car Type)")
x.set(xlabel = "Car Type", ylabel="something occurs over a particular
period of time ")
```

```
[Text(0.5, 0, "Car Type"),
 Text(0, 0.5, "something occurs over a particular period of time ")]
```

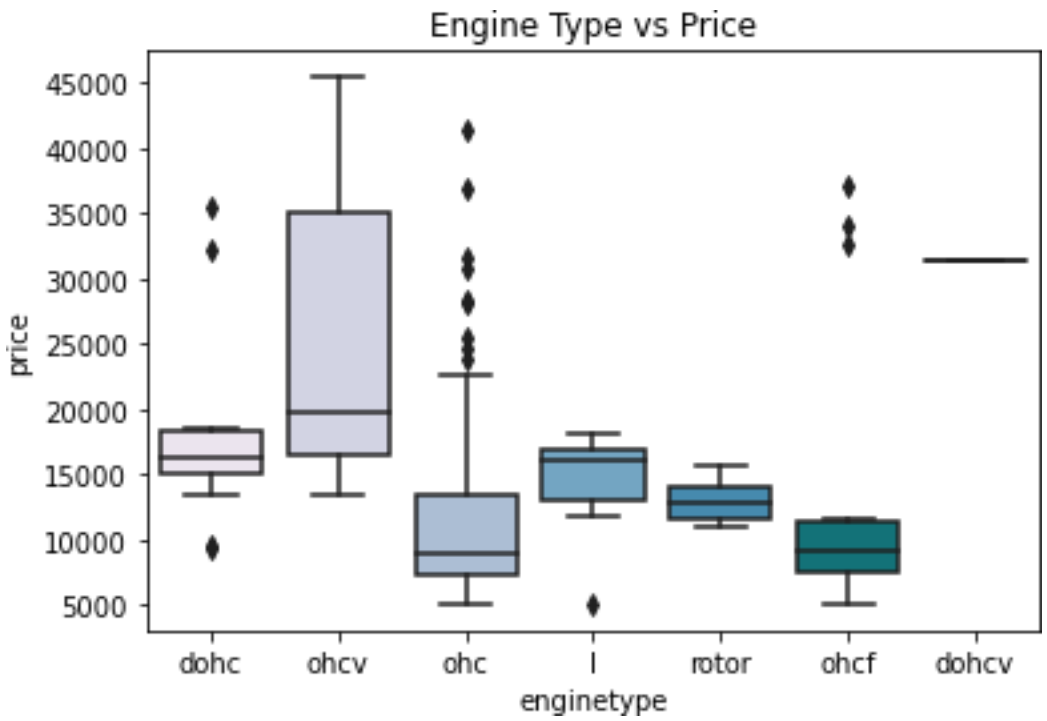


#as we observe above car type histogram plot we can conclude that sedan is better when comparewith all types of cars

```
plt.title("Histogram(Engine Type)")
sns.countplot(cars_data.enginetype, palette=("Blues_d"))
plt.show()
```

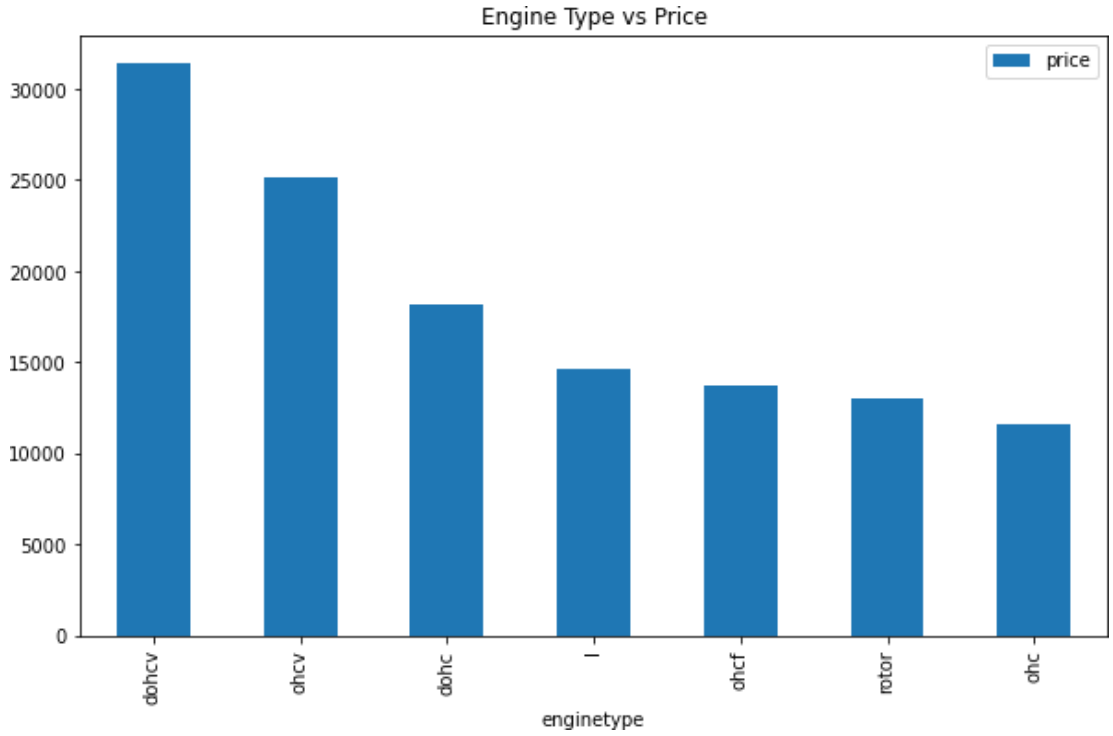



```
#ohc engine type is more favour to everyone(ohc engine seems like more favourable)

plt.title("Engine Type vs Price")
sns.boxplot(x=cars_data.enginetype, y=cars_data.price,
palette=("PuBuGn"))
plt.show()
```



#Engine type vs Price

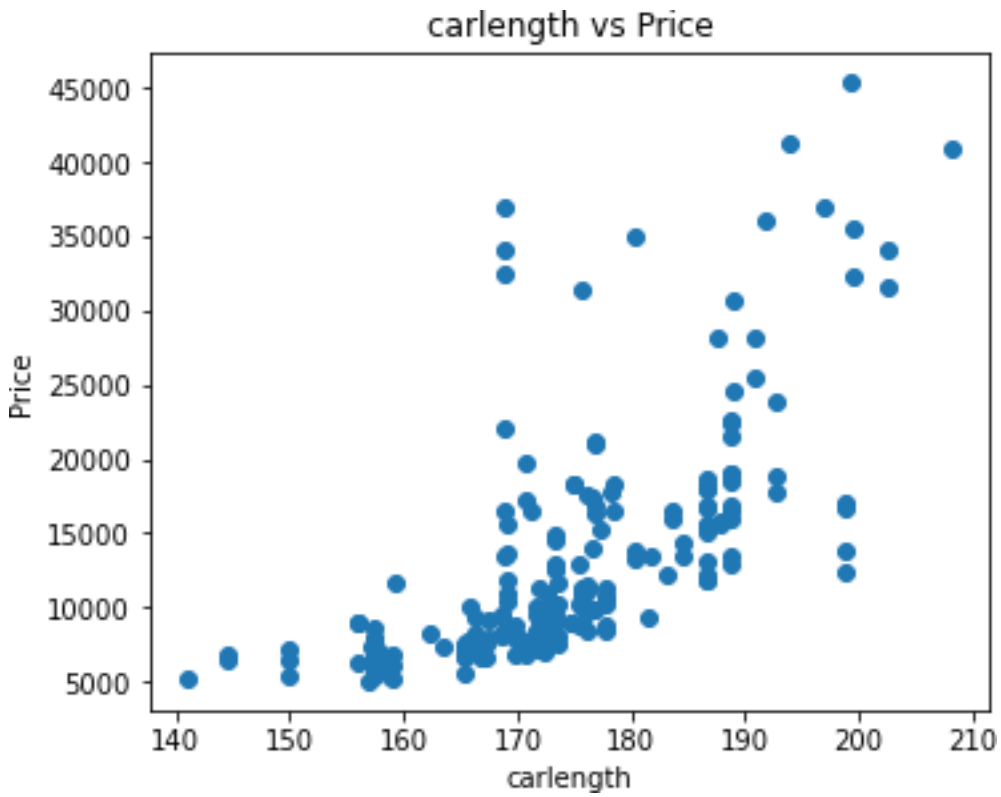
```
df = pd.DataFrame(cars_data.groupby(["enginetype"])
["price"].mean().sort_values(ascending = False))
df.plot.bar(figsize=(10,6))
plt.title("Engine Type vs Price")

Text(0.5, 1.0, "Engine Type vs Price")
```




```
#comparing engine type vs price(0hcv is having height price range)
```

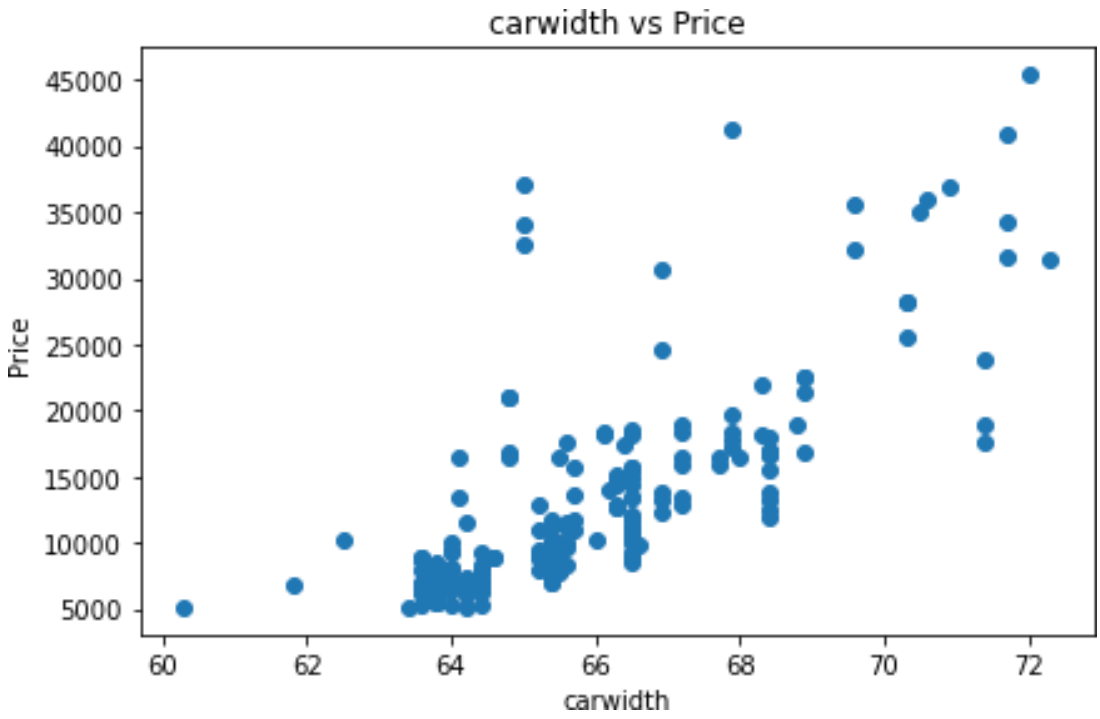
Visualising numerical data

```
#Plottting Scatter Plot for Car_length
def scatter(x,fig):
    plt.subplot(5,2,fig)
    plt.scatter(cars_data[x],cars_data["price"])
    plt.title(x+" vs Price")
    plt.ylabel("Price")
    plt.xlabel(x)

plt.figure(figsize=(10,20))

scatter("carlength", 1)

plt.tight_layout()
```



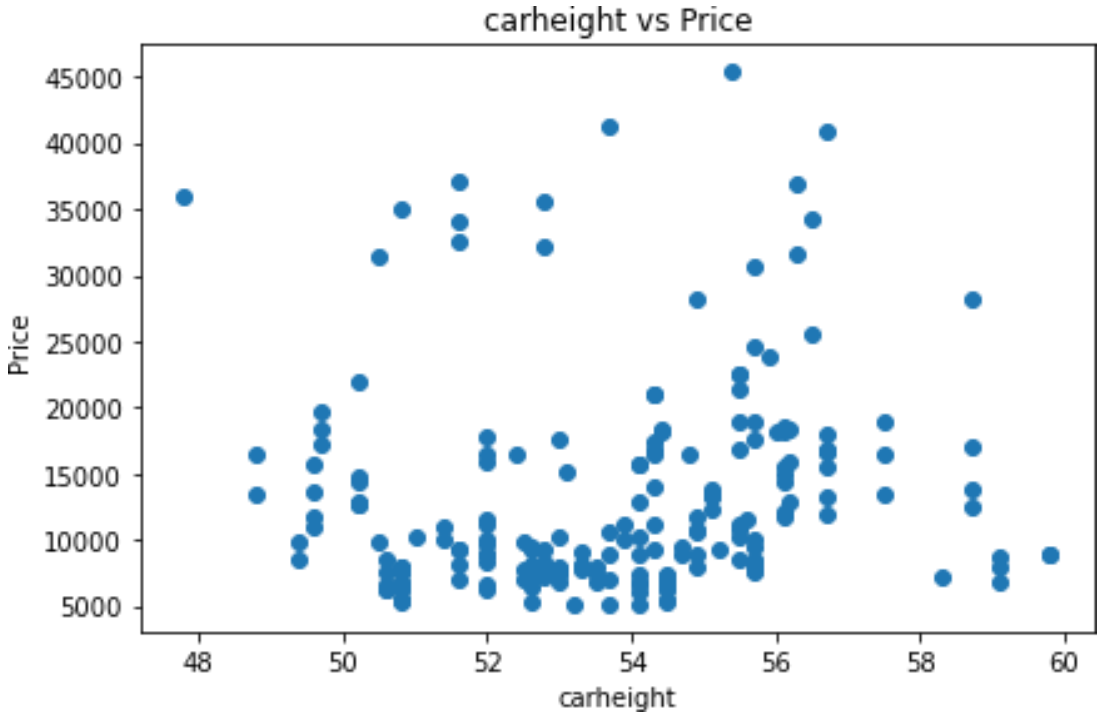

```
#as we observe car length is having somewhat relation with price
```

```
#Plotting Scatter Plot For Car_Width
plt.figure(figsize=(15,25))
scatter("carwidth", 2)
```



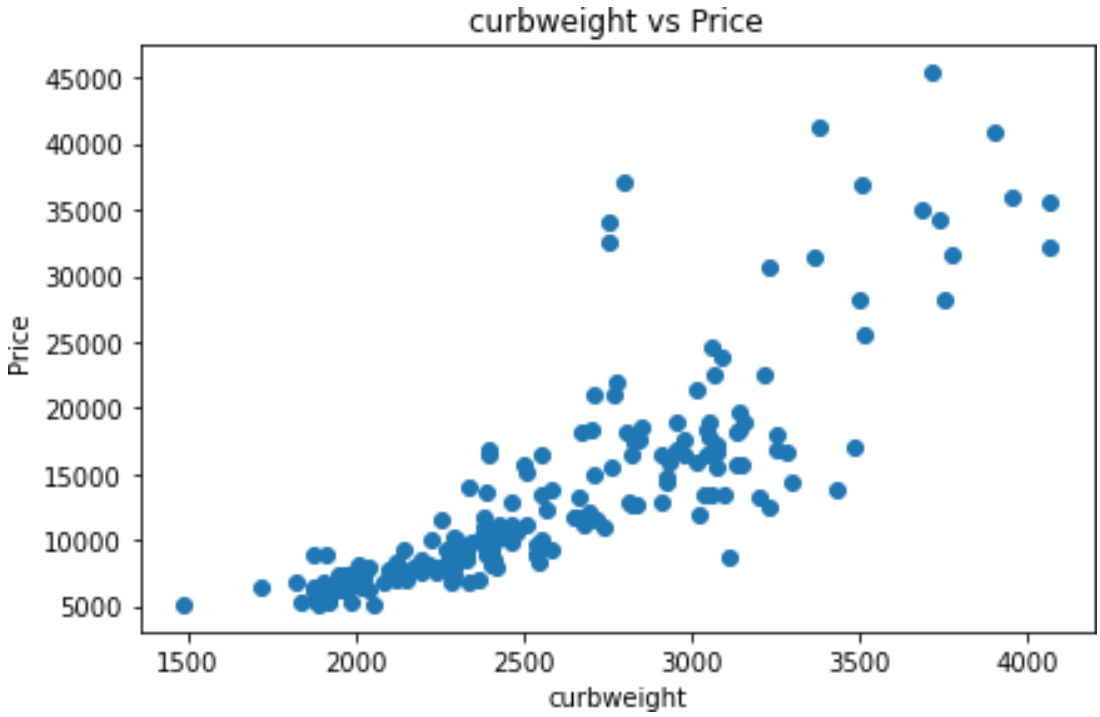
#as we observe car width is having somewhat relation with price

```
#Plotting Scatter Plot For Car_Height
plt.figure(figsize=(15,25))
scatter("carheight", 3)
```



#as we observe car height, it is not having relation with price because of irrelivent in nature

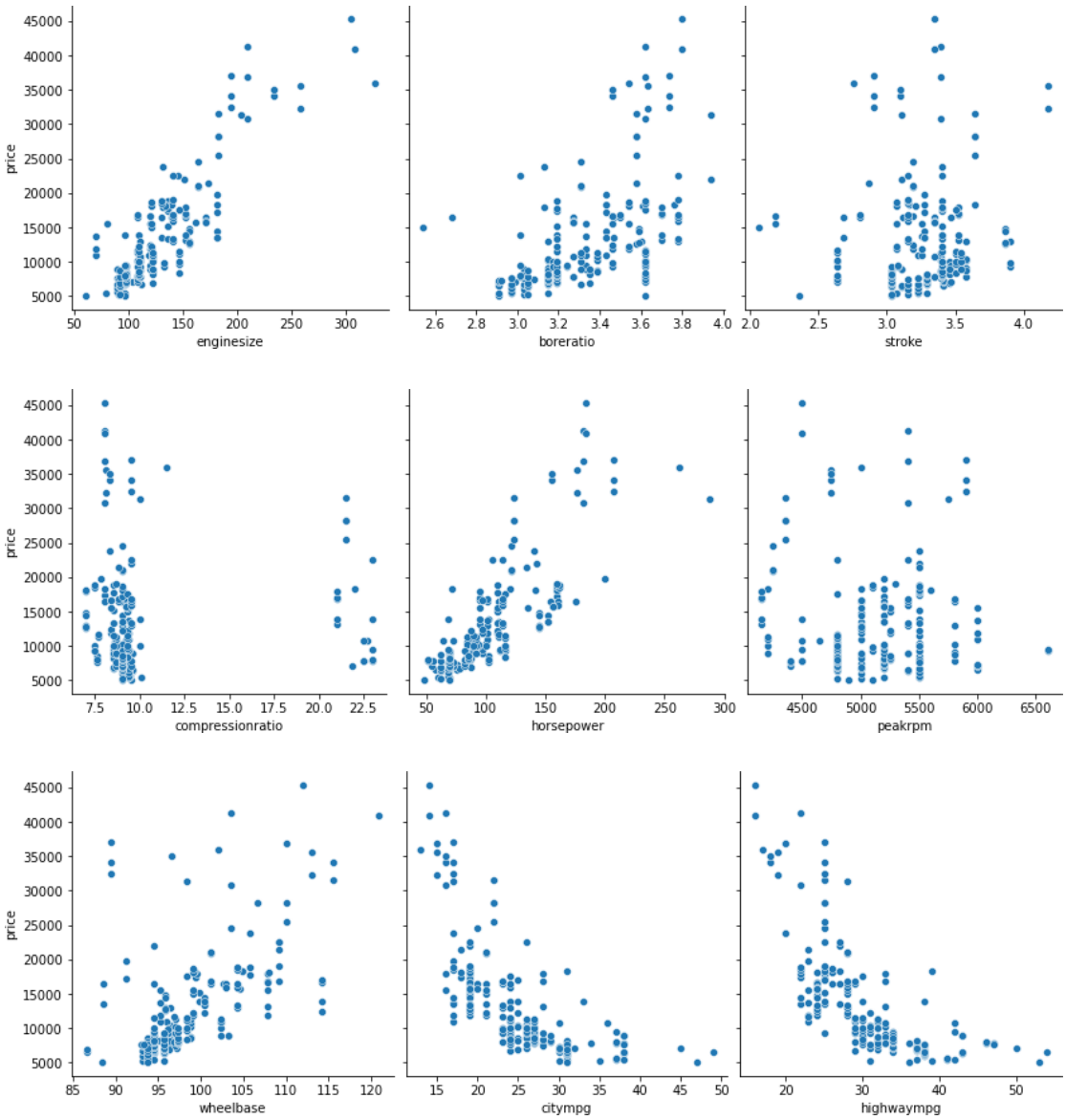
```
#Plotting scatterplot for curbweight
plt.figure(figsize=(15,25))
scatter("curbweight", 4)
```




```
#as we observe curbweight is having some what relation with price
```

#now we can say clearly that car length , car width, curbweight are moreliked when compare with car height

```
#Ploting Pair Plots

def a(x,y,z):
    sns.pairplot(cars_data, x_vars=[x,y,z], y_vars="price",size=4,
aspect=1, kind="scatter")
    plt.show()

a("enginesize", "boreratio", "stroke")
a("compressionratio", "horsepower", "peakrpm")
a("wheelbase", "citympg", "highwaympg")
```



#enginesize, boreratio, horsepower, wheelbase are having positive relation with price whereas citympg, highwaympg are not hvaing relation with price.

lets observe the correlation coefficient matrix between components correlation

```
np.corrcoef(cars_data["carlength"], cars_data["carwidth"])[0, 1]
```

0.841118268481846

```
np.corrcoef(cars_data["carwidth"], cars_data["stroke"])[0, 1]
```

0.18294169251421802

```
np.corrcoef(cars_data["carlength"], cars_data["stroke"])[0, 1]
```

```
0.1295326111279122
```

```
np.corrcoef(cars_data["carheight"], cars_data["carwidth"])[0, 1]
```

```
0.2792103211097495
```

```
np.corrcoef(cars_data["carlength"], cars_data["carheight"])[0, 1]
```

```
0.4910294575042146
```

```
np.corrcoef(cars_data["carheight"], cars_data["horsepower"])[0, 1]
```

```
-0.10880205898413571
```

```
np.corrcoef(cars_data["horsepower"], cars_data["stroke"])[0, 1]
```

```
0.0809395357449534
```

```
np.corrcoef(cars_data["carlength"], cars_data["price"])[0, 1]
```

```
0.6829200156779629
```

```
np.corrcoef(cars_data["carheight"], cars_data["price"])[0, 1]
```

```
0.11933622657049436
```

```
np.corrcoef(cars_data["carwidth"], cars_data["price"])[0, 1]
```

```
0.7593252997415118
```

```
np.corrcoef(cars_data["horsepower"], cars_data["price"])[0, 1]
```

```
0.8081388225362212
```

```
np.corrcoef(cars_data["carlength"], cars_data["carwidth"])[0, 1]
```

```
0.841118268481846
```

```
cars_data.columns
```

```
Index(["car_ID", "symboling", "CompanyName", "fueltype", "aspiration",
       "doornumber", "carbody", "drivewheel", "enginelocation",
"wheelbase",
       "carlength", "carwidth", "carheight", "curbweight",
"enginetype",
       "cylindernumber", "enginesize", "fuelsystem", "boreratio",
"stroke",
       "compressionratio", "horsepower", "peakrpm", "citympg",
"highwaympg",
       "price"],
      dtype="object")
```

```
np.corrcoef(cars_data["peakrpm"], cars_data["price"])[0, 1]
```

```
-0.08526715027785685
```

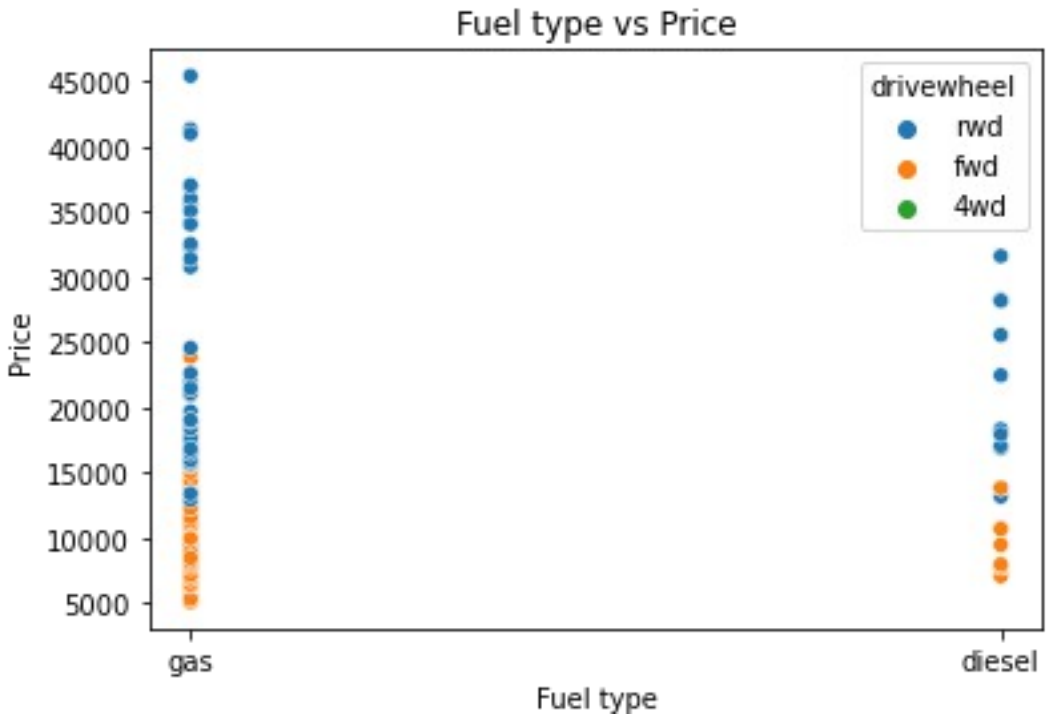
```
np.corrcoef(cars_data["compressionratio"], cars_data["price"])[0, 1]
```

0.06798350579944265

so, we can observe above the co-relation between what people want to observe in cars_data

Bivariate Analysis

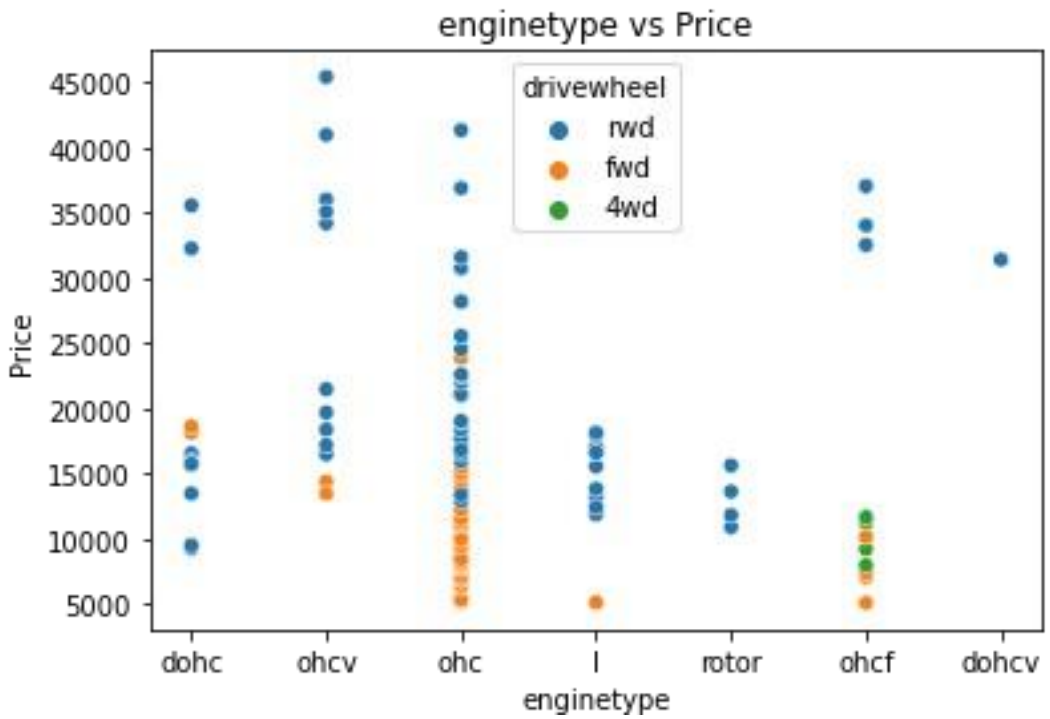
```
#plotting scatter plot between Fuel type vs Price
plt.title("Fuel type vs Price")
sns.scatterplot(x=cars_data["fueltype"],y=cars_data["price"],hue=cars_
data["drivewheel"])
plt.xlabel("Fuel type")
plt.ylabel("Price")

plt.show()
```



#we can observe co-relation between fuel economy and price of car

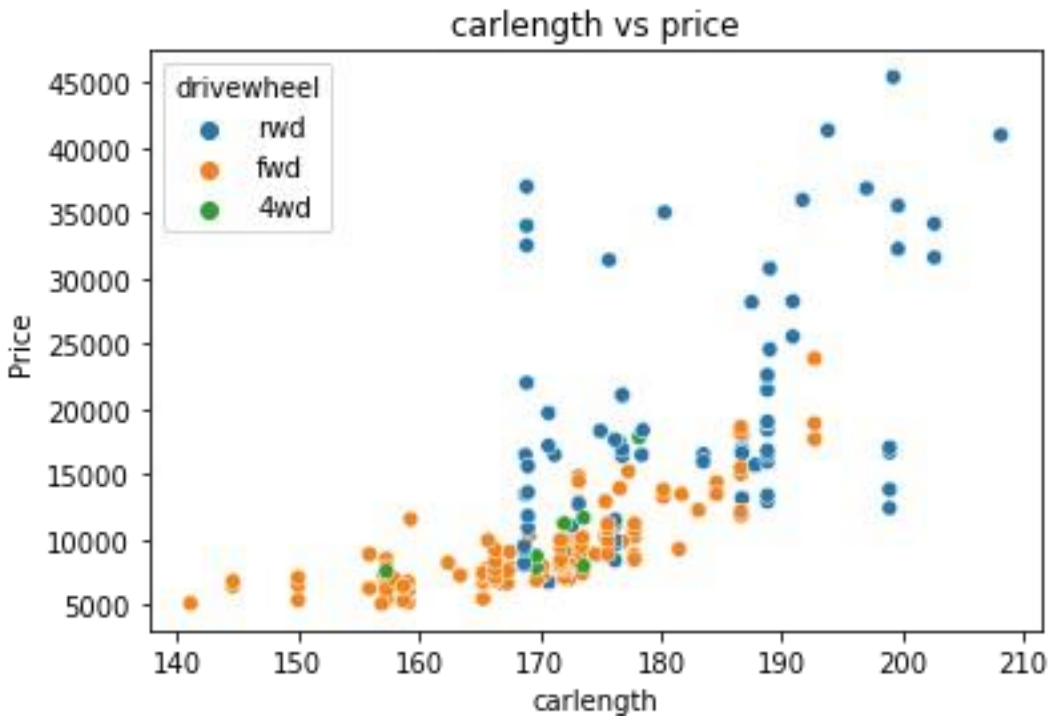
```
#plotting scatter plot between enginetype vs Price
plt.title("enginetype vs Price")
sns.scatterplot(x=cars_data["enginetype"],y=cars_data["price"],hue=car
s_data["drivewheel"])
plt.xlabel("enginetype")
plt.ylabel("Price")
```

```
plt.show()
```



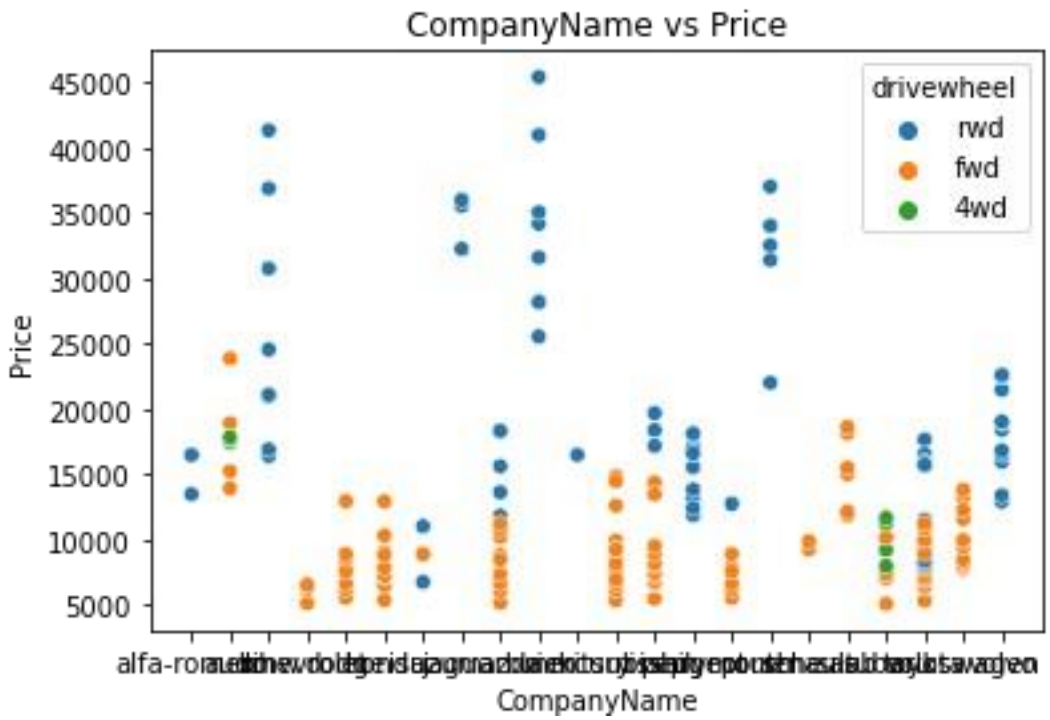
#we can observe co-relation between enginetype and price of car

```
#plotting scatter plot between carlength vs price
plt.title("carlength vs price")
sns.scatterplot(x=cars_data["carlength"],y=cars_data["price"],hue=cars
_data["drivewheel"])
plt.xlabel("carlength")
plt.ylabel("Price")

plt.show()
```



#we can observe co-relation between carlength and price of car

```
#plotting scatter plot between companyname vs price
plt.title("CompanyName vs Price")
sns.scatterplot(x=cars_data["CompanyName"],y=cars_data["price"],hue=ca
rs_data["drivewheel"])
plt.xlabel("CompanyName")
plt.ylabel("Price")
plt.figure(figsize=(20,50))
plt.show()
```




```
<Figure size 1440x3600 with 0 Axes>
```

```
   #we can observe co-relation between  carname and price of car
```

```
#After analysing the data decided to delete some columns and
finaalised some columns
new_cars_data = cars_data[["price", "fueltype",
"aspiration","carbody", "drivewheel","wheelbase",
                   "curbweight", "enginetype", "cylindernumber",
"enginesize", "boreratio","horsepower",
                    "fueltype", "carlength","carwidth"]]

new_cars_data.head()
```

```
     price fueltype aspiration      carbody drivewheel  wheelbase
curbweight  \
0  13495.0      gas        std  convertible        rwd       88.6
2548
1  16500.0      gas        std  convertible        rwd       88.6
2548
2  16500.0      gas        std    hatchback        rwd       94.5
2823
3  13950.0      gas        std        sedan        fwd       99.8
2337
4  17450.0      gas        std        sedan        4wd       99.4
2824

  enginetype cylindernumber  enginesize  boreratio  horsepower
```

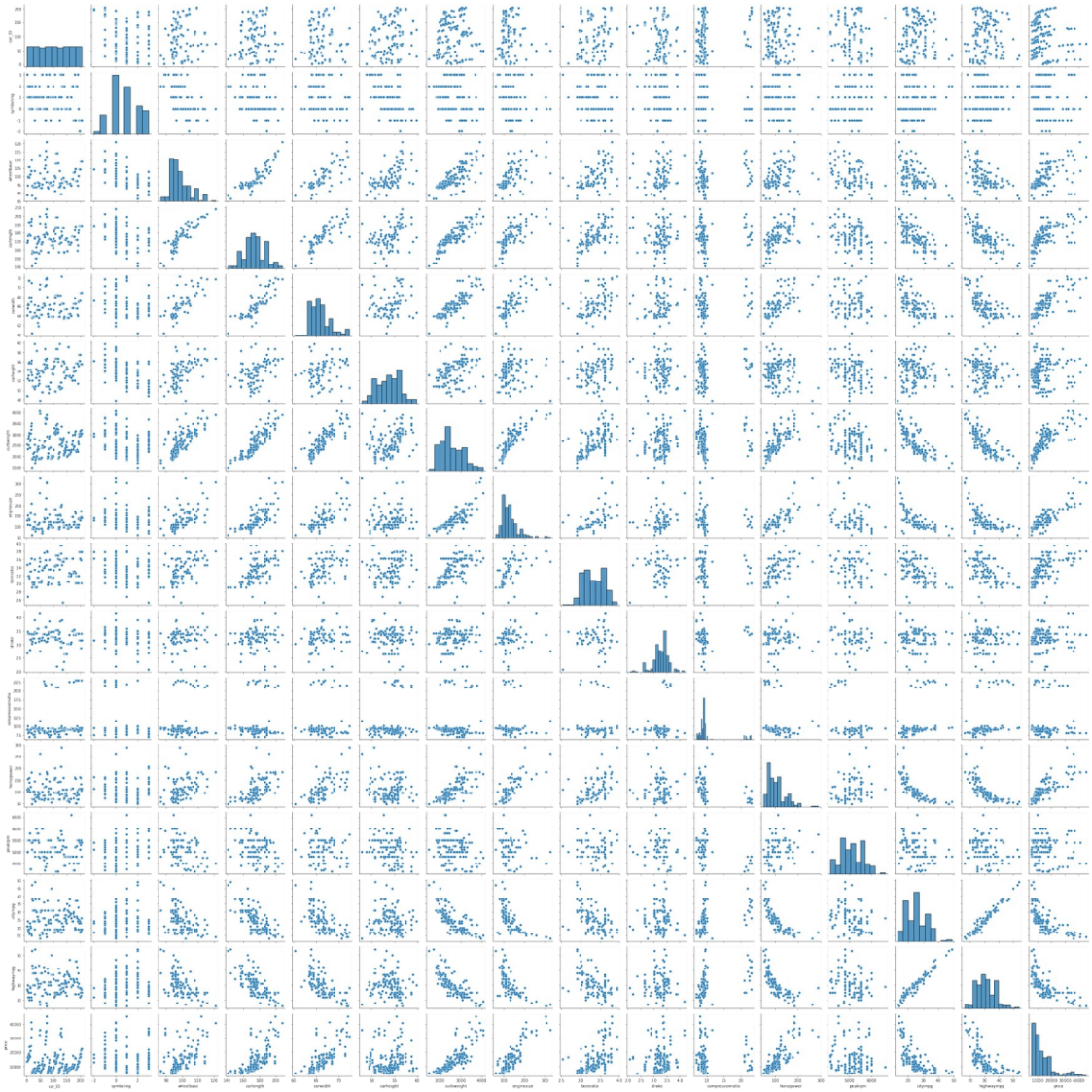
```
fueltype  \
0        dohc          four          130        3.47          111
gas
1        dohc          four          130        3.47          111
gas
2        ohcv           six          152        2.68          154
gas
3         ohc          four          109        3.19          102
gas
4         ohc          five          136        3.19          115
gas

   carlength  carwidth
0      168.8      64.1
1      168.8      64.1
2      171.2      65.5
3      176.6      66.2
4      176.6      66.4
```

```
#Now we can observe shape of the new_cars_data after finalised the
data
new_cars_data.shape
```

```
(205, 15)
```

```
#Lets see pairplot of cars_data
import seaborn as sns
sns.pairplot(cars_data)
```

```
<seaborn.axisgrid.PairGrid at 0x207b63a9580>
```



```
#lets import train_test_split from sklearn.model_selection & create
new_cars_data is in df_train using train_test_split
from sklearn.model_selection import train_test_split

np.random.seed(0)
df_train, df_test = train_test_split(new_cars_data, train_size = 0.7,
test_size = 0.3, random_state = 100)

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
num_vars = ["wheelbase", "curbweight", "enginesize", "boreratio",
"horsepower","carlength","carwidth","price"]
df_train[num_vars] = scaler.fit_transform(df_train[num_vars])

#print head in df_train
df_train.head()
```

```
        price fueltype aspiration      carbody drivewheel
wheelbase  \
122  0.068818      gas        std        sedan        fwd    0.244828

125  0.466890      gas        std    hatchback        rwd    0.272414

166  0.122110      gas        std    hatchback        rwd    0.272414

1    0.314446      gas        std  convertible        rwd    0.068966

199  0.382131      gas      turbo        wagon        rwd    0.610345


     curbweight enginetype cylindernumber  enginesize  boreratio
horsepower  \
122     0.272692         ohc           four     0.139623   0.230159
0.083333
125     0.500388         ohc           four     0.339623   1.000000
0.395833
166     0.314973        dohc           four     0.139623   0.444444
0.266667
1       0.411171        dohc           four     0.260377   0.626984
0.262500
199     0.647401         ohc           four     0.260377   0.746032
0.475000

    fueltype  carlength  carwidth
122      gas   0.426016  0.291667
125      gas   0.452033  0.666667
166      gas   0.448780  0.308333
1        gas   0.450407  0.316667
199      gas   0.775610  0.575000
```

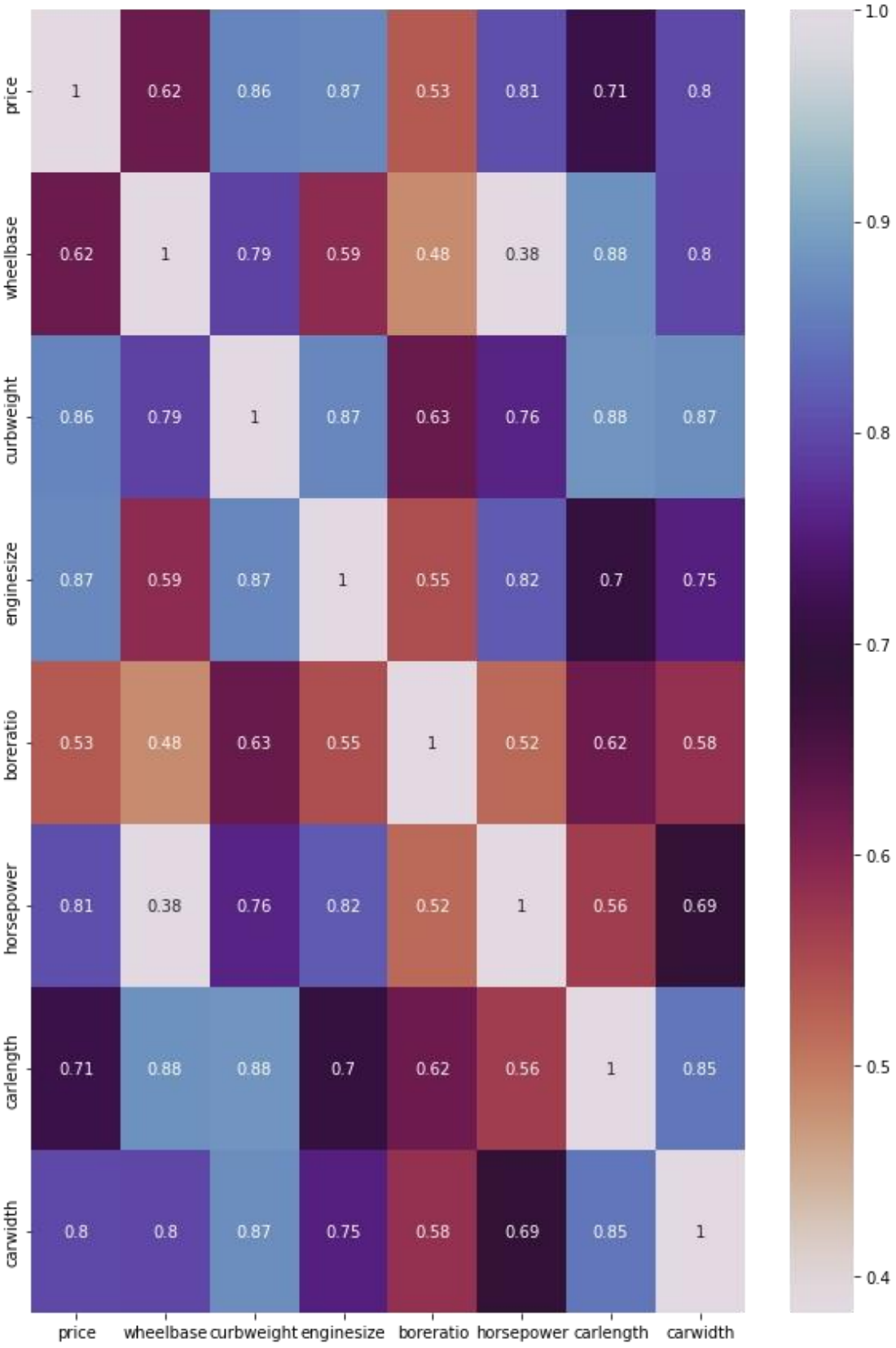
```
#Because of str to float error ...removing fueltype using pop option
df_train.pop("fueltype")
```

```
    fueltype fueltype
122      gas      gas
125      gas      gas
166      gas      gas
1        gas      gas
199      gas      gas
..       ...      ...
87       gas      gas
103      gas      gas
67    diesel   diesel
24       gas      gas
8        gas      gas

[143 rows x 2 columns]
```

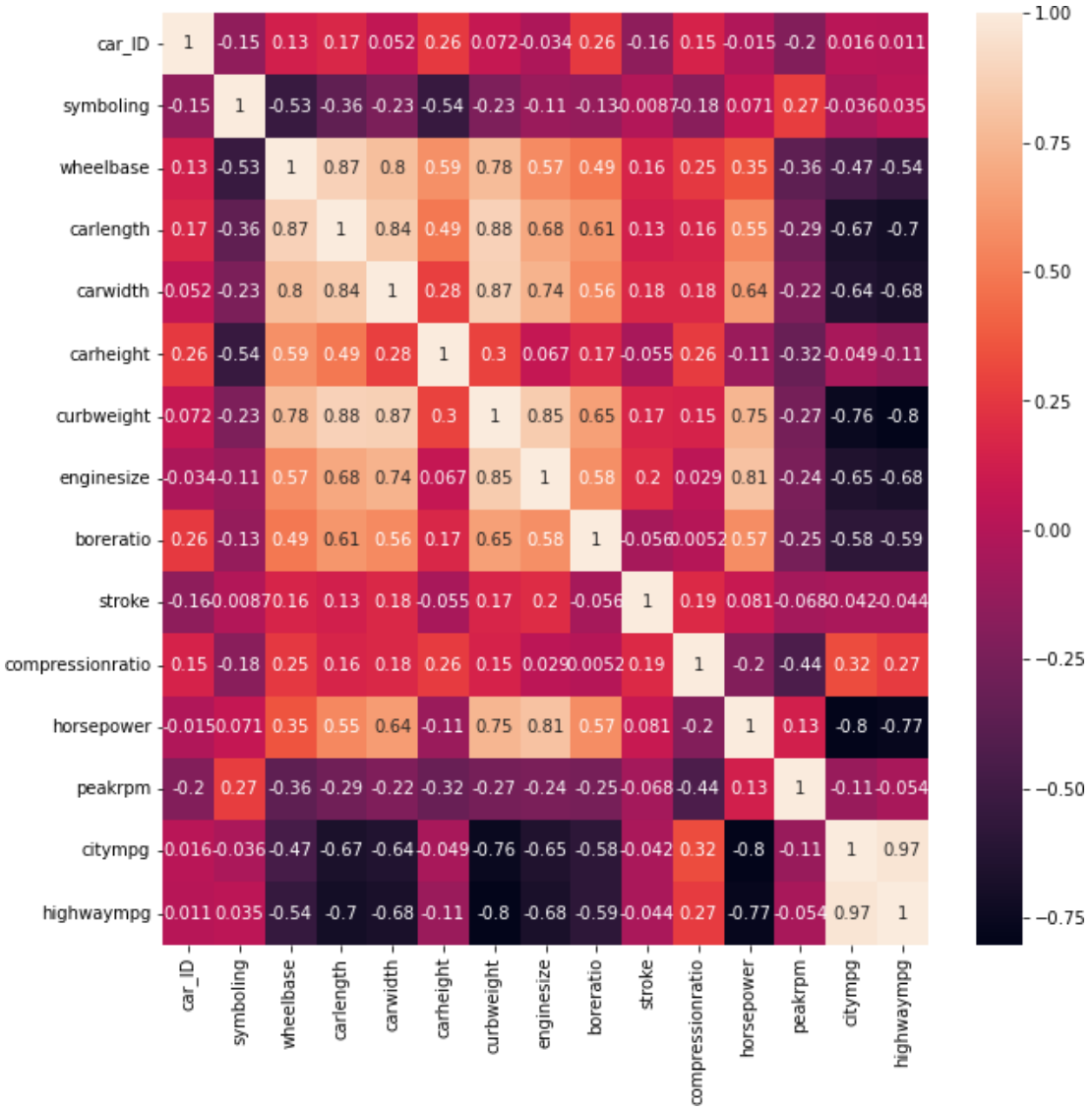
```
#Lets describe df_train
df_train.describe()
```

```
              price   wheelbase   curbweight   enginesize   boreratio
horsepower  \
count   143.000000  143.000000  143.000000  143.000000  143.000000
143.000000
mean      0.219310    0.411141    0.407878    0.241351    0.497946
0.227302
std       0.215682    0.205581    0.211269    0.154619    0.207140
0.165511
min       0.000000    0.000000    0.000000    0.000000    0.000000
0.000000
25%       0.067298    0.272414    0.245539    0.135849    0.305556
0.091667
50%       0.140343    0.341379    0.355702    0.184906    0.500000
0.191667
75%       0.313479    0.503448    0.559542    0.301887    0.682540
0.283333
max       1.000000    1.000000    1.000000    1.000000    1.000000
1.000000

         carlength    carwidth
count   143.000000  143.000000
mean      0.525476    0.461655
std       0.204848    0.184517
min       0.000000    0.000000
25%       0.399187    0.304167
50%       0.502439    0.425000
75%       0.669919    0.550000
max       1.000000    1.000000
```

```
#Correlation using heatmap
plt.figure(figsize = (10, 15))
sns.heatmap(df_train.corr(), annot = True, cmap="twilight_r")
plt.show()
```


price
wheelbase
curbweight
enginesize
boreratio
horsepower
carlength
carwidth
1 0.62 0.86 0.87 0.53 0.81 0.71 0.8
0.62 1 0.79 0.59 0.48 0.38 0.88 0.8
0.86 0.79 1 0.87 0.63 0.76 0.88 0.87
0.87 0.59 0.87 1 0.55 0.82 0.7 0.75
0.53 0.48 0.63 0.55 1 0.52 0.62 0.58
0.81 0.38 0.76 0.82 0.52 1 0.56 0.69
0.71 0.88 0.88 0.7 0.62 0.56 1 0.85
0.8 0.8 0.87 0.75 0.58 0.69 0.85 1
price
wheelbase
curbweight
enginesize
boreratio
horsepower
carlength
carwidth
1.0
0.9
0.8
0.7
0.6
0.5
0.4

```
#Corelation using Heatmap
plt.figure(figsize=[10,10])
sns.heatmap(cars_data.drop("price",axis=1).corr(),annot=True)
plt.show()
```



ModelBuilding and Training

```
from sklearn.preprocessing import LabelEncoder

cars_data = cars_data.apply(lambda x:LabelEncoder().fit_transform(x))

y = cars_data.price
x = cars_data.drop("price",axis=1)

y = y.values.reshape((205,1))
```

```
m  =  np.zeros((x.shape[1],1))

lr = 0.00001
cost_list = []
for i in range(10):
    pred = np.dot(x,m)
    cost = sum((pred-y)**2)
    cost_list.append(cost)
    error = (pred-y)**2
    gradient  = 2 * np.dot(x.T,error)
    m = m - lr * gradient
```

Import Linear regression, Lasso,Ridge Regression model from sklearn.linear_model

```
from sklearn.linear_model import LinearRegression, Lasso, Ridge

regression_model = Ridge()

regression_model.fit(x,y)
```

```
Ridge()
```

```
#Lets predict x
predictions = regression_model.predict(x)

regression_model.intercept_
```

```
array([40.77068588])
```

```
#lets see r2_score(predictions,y)
from sklearn.metrics import r2_score
r2_score(predictions,y)
```

```
0.9117296884272439
```

```
#Print columns list to observe the columns
cars_data.columns.tolist()
```

```
["car_ID",
 "symboling",
 "CompanyName",
 "fueltype",
 "aspiration",
 "doornumber",
 "carbody",
 "drivewheel",
 "enginelocation",
 "wheelbase",
 "carlength",
```

```
 "carwidth",
 "carheight",
 "curbweight",
 "enginetype",
 "cylindernumber",
 "enginesize",
 "fuelsystem",
 "boreratio",
 "stroke",
 "compressionratio",
 "horsepower",
 "peakrpm",
 "citympg",
 "highwaympg",
 "price"]
```

Import statsmodels.api and observe Ordinary Least Squares regression (OLS)

because it is a common technique for estimating coefficients of linear regression equations which describe the relationship between one or more independent quantitative variables and a dependent variable (simple or multiple linear regression).

```
predictors=cars_data.drop("price",axis=1)
target=cars_data.price
predictors1=predictors["enginesize"]
import statsmodels.api as sm
predictors1= sm.add_constant(predictors1)
lm_1 = sm.OLS(target,predictors1).fit()
print(lm_1.summary())
```

```
                            OLS Regression Results

==============================================================================
========
Dep. Variable:                  price   R-squared:
0.670
Model:                            OLS   Adj. R-squared:
0.668
Method:                 Least Squares   F-statistic:
411.7
Date:                Mon, 21 Nov 2022   Prob (F-statistic):
9.87e-51
Time:                        17:18:34   Log-Likelihood:
-996.17
No. Observations:                 205   AIC:
1996.
Df Residuals:                     203   BIC:
2003.
```

```
Df Model:                           1

Covariance Type:            nonrobust

==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         23.2121      4.016      5.780      0.000      15.294      31.130
enginesize     4.1048      0.202     20.291      0.000       3.706       4.504
==============================================================================
Omnibus:                       13.796   Durbin-Watson:                   0.935
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               14.833
Skew:                           0.653   Prob(JB):                     0.000601
Kurtosis:                       3.175   Cond. No.                         36.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```
predictors2=predictors[["enginesize","fueltype"]]
import statsmodels.api as sm
predictors2= sm.add_constant(predictors2)
lm_2 = sm.OLS(target,predictors2).fit()
print(lm_2.summary())
```

```
                            OLS Regression Results

==============================================================================
Dep. Variable:                  price   R-squared:                       0.671
Model:                            OLS   Adj. R-squared:                  0.668
Method:                 Least Squares   F-statistic:                     206.1
Date:                Mon, 21 Nov 2022   Prob (F-statistic):           1.66e-49
Time:                        17:18:34   Log-Likelihood:
```

```
-995.75
No. Observations:                  205   AIC:
1997.
Df Residuals:                      202   BIC:
2007.
Df Model:                            2

Covariance Type:            nonrobust

==============================================================================
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
const          29.7614      8.198      3.631      0.000      13.598
45.925
enginesize      4.0812      0.204     20.004      0.000       3.679
4.483
fueltype       -6.8216      7.443     -0.917      0.360     -21.497
7.854
==============================================================================
Omnibus:                       14.547   Durbin-Watson:
0.952
Prob(Omnibus):                  0.001   Jarque-Bera (JB):
15.762
Skew:                           0.672   Prob(JB):
0.000378
Kurtosis:                       3.192   Cond. No.
97.2
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

```
#adding both models
predictors4=predictors[["enginesize","fueltype","carlength"]]
import statsmodels.api as sm
predictors4= sm.add_constant(predictors4)
lm_4 = sm.OLS(target,predictors4).fit()
print(lm_4.summary())
```

```
                            OLS Regression Results

==============================================================================
Dep. Variable:                  price   R-squared:
```

```
0.759
Model:                            OLS   Adj. R-squared:
0.755
Method:                 Least Squares   F-statistic:
211.1
Date:                Mon, 21 Nov 2022   Prob (F-statistic):
7.44e-62
Time:                        17:18:34   Log-Likelihood:
-963.84
No. Observations:                 205   AIC:
1936.
Df Residuals:                     201   BIC:
1949.
Df Model:                           3

Covariance Type:            nonrobust

==============================================================================
========
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------------------
const          7.9506      7.480      1.063      0.289      -6.799
22.700
enginesize     2.4939      0.255      9.784      0.000       1.991
2.997
fueltype       1.6783      6.463      0.260      0.795     -11.065
14.421
carlength      1.1339      0.132      8.567      0.000       0.873
1.395
==============================================================================
========
Omnibus:                       10.660   Durbin-Watson:
1.092
Prob(Omnibus):                  0.005   Jarque-Bera (JB):
10.847
Skew:                           0.547   Prob(JB):
0.00441
Kurtosis:                       3.274   Cond. No.
231.
==============================================================================
========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

```
predictors5=predictors[["enginesize","fueltype","stroke"]]
import statsmodels.api as sm
```

```
predictors5= sm.add_constant(predictors5)
lm_5 = sm.OLS(target,predictors5).fit()
print(lm_5.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.687
Model:                            OLS   Adj. R-squared:                  0.683
Method:                 Least Squares   F-statistic:                     147.2
Date:                Mon, 21 Nov 2022   Prob (F-statistic):           1.76e-50
Time:                        17:18:34   Log-Likelihood:                -990.61
No. Observations:                 205   AIC:                             1989.
Df Residuals:                     201   BIC:                             2003.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         49.7785     10.149      4.905      0.000      29.767      69.790
enginesize     4.2338      0.205     20.650      0.000       3.830       4.638
fueltype     -13.6987      7.584     -1.806      0.072     -28.654       1.257
stroke        -0.8268      0.257     -3.215      0.002      -1.334      -0.320
==============================================================================
Omnibus:                       15.316   Durbin-Watson:                   0.941
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               16.478
Skew:                           0.661   Prob(JB):                     0.000264
Kurtosis:                       3.425   Cond. No.                         160.
```

```
==============================================================================
========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

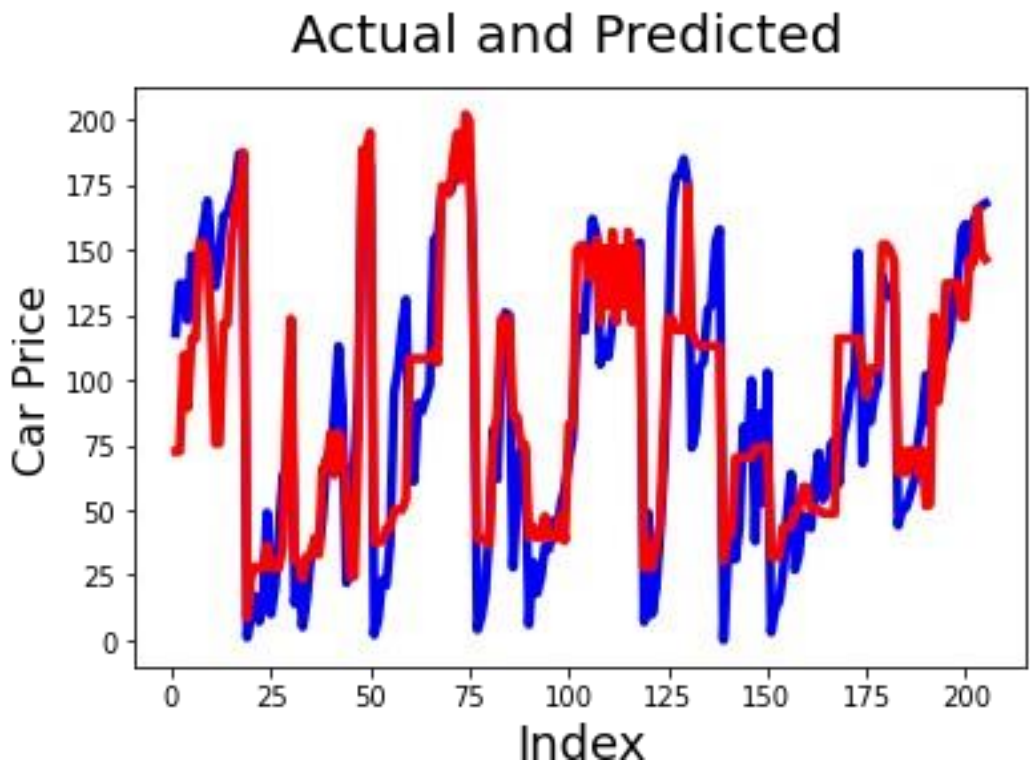
```
predictors6=predictors[["enginesize","carwidth","carlength"]]
import statsmodels.api as sm
predictors6= sm.add_constant(predictors6)
lm_6 = sm.OLS(target,predictors6).fit()
print(lm_6.summary())
```

```
                            OLS Regression Results

==============================================================================
========
Dep. Variable:                  price   R-squared:
0.772
Model:                            OLS   Adj. R-squared:
0.769
Method:                 Least Squares   F-statistic:
227.5
Date:                Mon, 21 Nov 2022   Prob (F-statistic):
2.42e-64
Time:                        17:18:34   Log-Likelihood:
-957.99
No. Observations:                 205   AIC:
1924.
Df Residuals:                     201   BIC:
1937.
Df Model:                           3

Covariance Type:            nonrobust

==============================================================================
========
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------------------
const          8.5213      3.697      2.305      0.022       1.232
15.811
enginesize     2.1860      0.263      8.298      0.000       1.667
2.705
carwidth       1.3376      0.388      3.447      0.001       0.572
2.103
carlength      0.6356      0.191      3.322      0.001       0.258
1.013
==============================================================================
```

```
========
Omnibus:                       13.835   Durbin-Watson:
1.030
Prob(Omnibus):                  0.001   Jarque-Bera (JB):
14.894
Skew:                           0.655   Prob(JB):
0.000583
Kurtosis:                       3.169   Cond. No.
101.
================================================================================
========

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

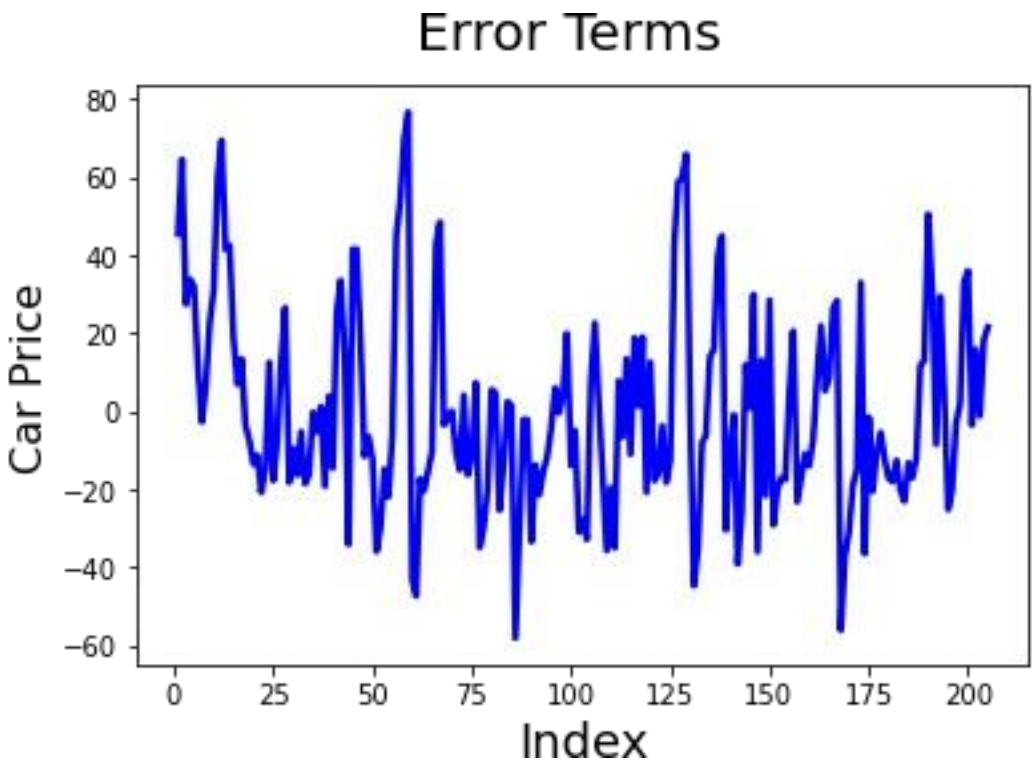
```
#lets plot Actual vs predicted plots by giving proper labeling
pred=lm_6.predict(predictors6)
# Actual vs Predicted
c = [i for i in range(1,206,1)]
fig = plt.figure()
plt.plot(c,target, color="blue", linewidth=3.5, linestyle="-")
#Plotting Actual
plt.plot(c,pred, color="red", linewidth=3.5, linestyle="-")
#Plotting predicted
fig.suptitle("Actual and Predicted", fontsize=20)
# Plot heading
plt.xlabel("Index", fontsize=18)
# X-label
plt.ylabel("Car Price", fontsize=16)
(0, 0.5, "Car Price")
```

```
(0, 0.5, "Car Price")
```



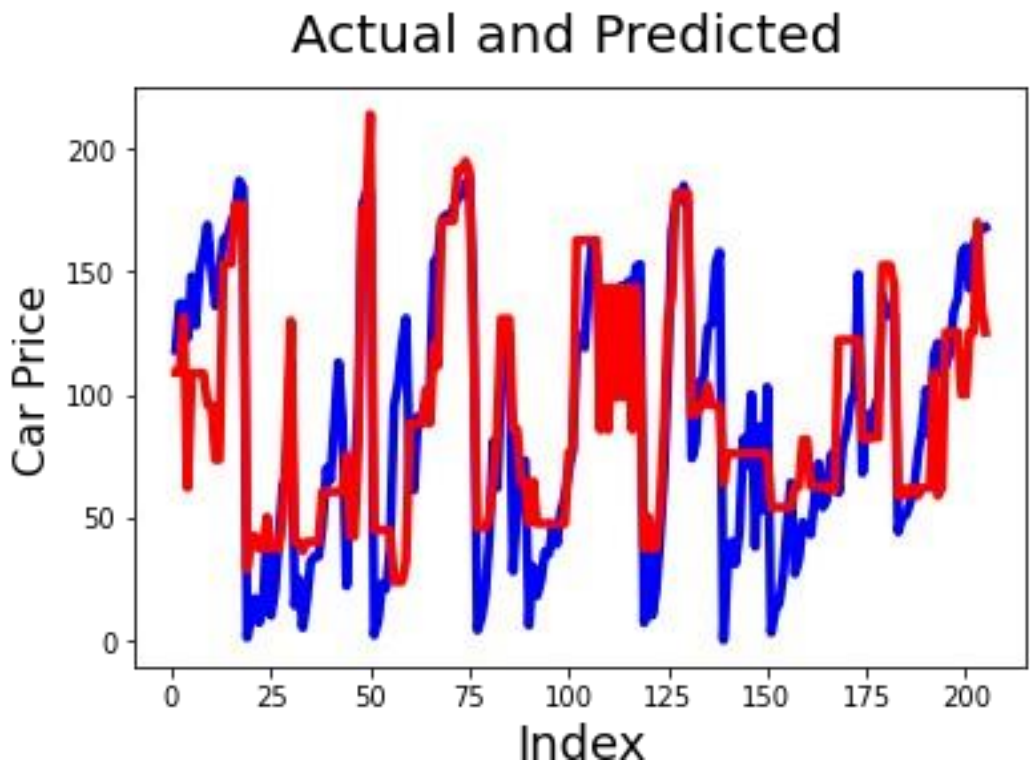

```
# lets see Error terms like mean_Square_error and r_square_value
c = [i for i in range(1,206,1)]
fig = plt.figure()
plt.plot(c,target-pred, color="blue", linewidth=2.5, linestyle="-")
fig.suptitle("Error Terms", fontsize=20)
# Plot heading
plt.xlabel("Index", fontsize=18)
# X-label
plt.ylabel("Car Price", fontsize=16)
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(target, pred)
r_squared = r2_score(target, pred)
print("Mean_Squared_Error :" ,mse)
print("r_square_value :",r_squared)
```

```
Mean_Squared_Error : 670.7493342021307
r_square_value : 0.7724622527736109
```



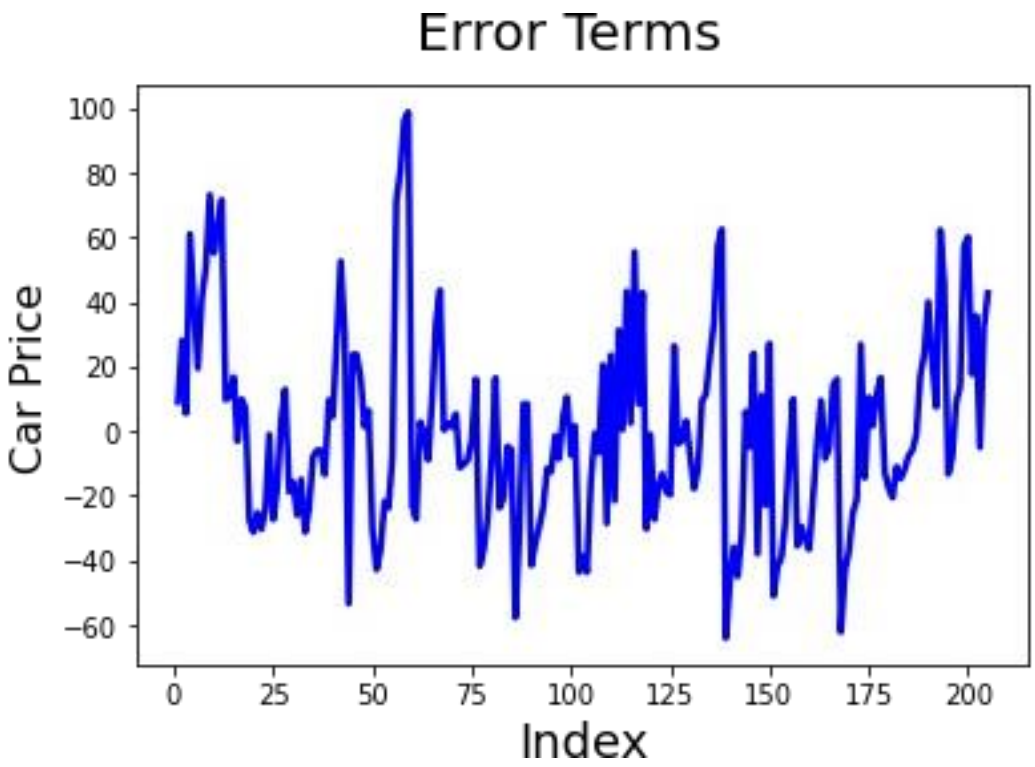

```
#ploting car_price between actual vs predicted
pred=lm_5.predict(predictors5)
# Actual vs Predicted
c = [i for i in range(1,206,1)]
fig = plt.figure()
plt.plot(c,target, color="blue", linewidth=3.5, linestyle="-")
#Plotting Actual
plt.plot(c,pred, color="red", linewidth=3.5, linestyle="-")
#Plotting predicted
fig.suptitle("Actual and Predicted", fontsize=20)
# Plot heading
plt.xlabel("Index", fontsize=18) # X-label
plt.ylabel("Car Price", fontsize=16)
(0, 0.5, "Car Price")
```

```
(0, 0.5, "Car Price")
```



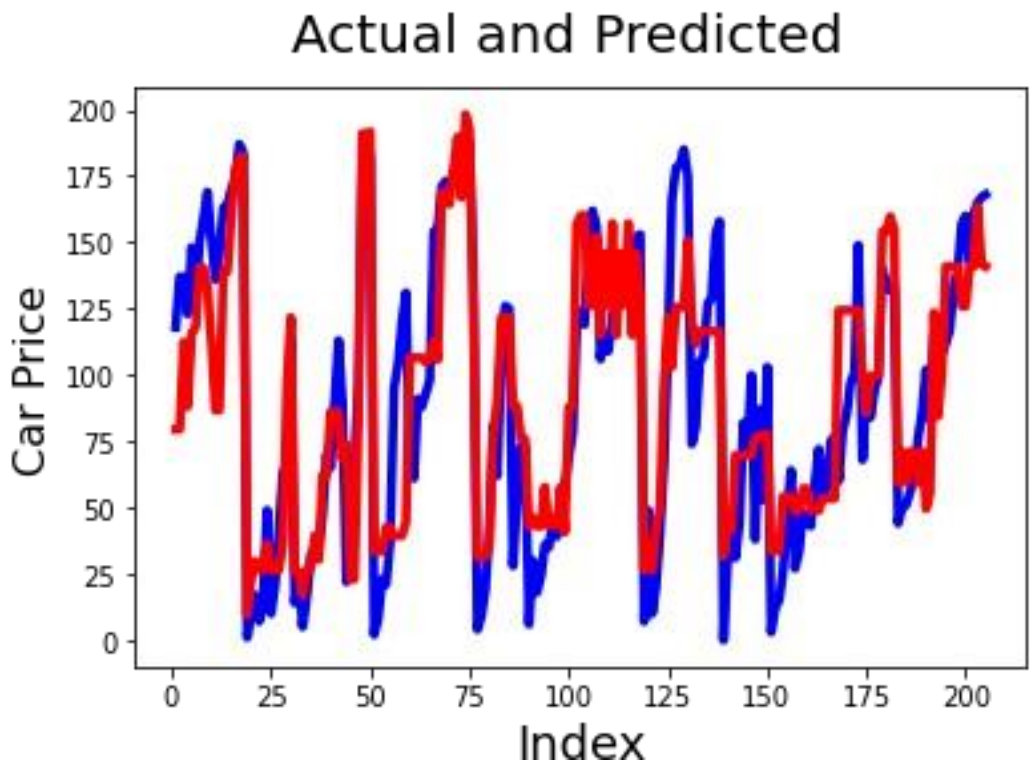

```
# Error terms
c = [i for i in range(1,206,1)]
fig = plt.figure()
plt.plot(c,target-pred, color="blue", linewidth=2.5, linestyle="-")
fig.suptitle("Error Terms", fontsize=20)
# Plot heading
plt.xlabel("Index", fontsize=18)
# X-label
plt.ylabel("Car Price", fontsize=16)
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(target, pred)
r_squared = r2_score(target, pred)
print("Mean_Squared_Error :" ,mse)
print("r_square_value :",r_squared)
```

```
Mean_Squared_Error : 922.0408783230008
r_square_value : 0.687216828095973
```



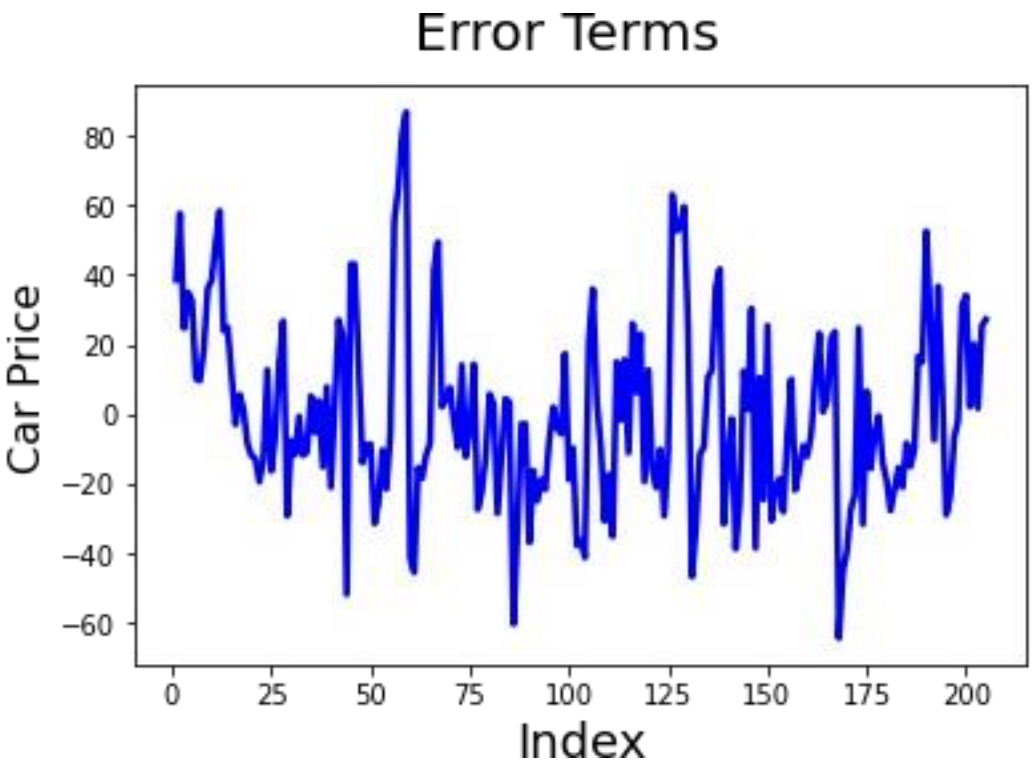

```
pred=lm_4.predict(predictors4)
# Actual vs Predicted
c = [i for i in range(1,206,1)]
fig = plt.figure()
plt.plot(c,target, color="blue", linewidth=3.5, linestyle="-")
#Plotting Actual
plt.plot(c,pred, color="red", linewidth=3.5, linestyle="-")
#Plotting predicted
fig.suptitle("Actual and Predicted", fontsize=20)
# Plot heading
plt.xlabel("Index", fontsize=18) # X-label
plt.ylabel("Car Price", fontsize=16)
(0, 0.5, "Car Price")
```

```
(0, 0.5, "Car Price")
```




```
# Error terms
c = [i for i in range(1,206,1)]
fig = plt.figure()
plt.plot(c,target-pred, color="blue", linewidth=2.5, linestyle="-")
fig.suptitle("Error Terms", fontsize=20)
# Plot heading
plt.xlabel("Index", fontsize=18)
# X-label
plt.ylabel("Car Price", fontsize=16)
from sklearn.metrics import mean_squared_error, r2_score
mse = mean_squared_error(target, pred)
r_squared = r2_score(target, pred)
print("Mean_Squared_Error :" ,mse)
print("r_square_value :",r_squared)
```

```
Mean_Squared_Error : 710.1583508115406
r_square_value : 0.7590935643493743
```



SO my point is that i have concluded from the above plots:-
'enginesize','Horsepower','drivewheel','carwidth','enginetype','price' plays a significant role in price prediction. So everyone are thinking mainly about these things. As we observe above plots, In cars_price_prediction ,
'enginesize','drivewheel','carwidth','enginetype','price','Horsepower' plays a significant role.

Thank you sir

Dubbaka Srikanth

G-17 Python & ML

Input is all columns and output(Target) is y column

you need to implement

Logisitic regression

Naive bayes

SVC classifier

Decision Tree classifier

Random Forest Classifier

Compare all the R2 Scores and execution speed to decide the best one

Logisitic regression

```
import pandas as pd
import numpy as np
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.decomposition import PCA
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Read Train Data and Test Data

```
#read data train
data_train = pd.read_csv(r"F:\my projects\Users\datasets\bank-full.csv",sep=";")
data_train
```

```
       age           job   marital  education default  balance housing loan  \
0       58    management   married   tertiary      no     2143     yes   no
1       44    technician    single  secondary      no       29     yes
```

```
no
2        33  entrepreneur   married  secondary      no         2      yes
yes
3        47   blue-collar   married    unknown      no      1506      yes
no
4        33       unknown    single    unknown      no         1       no
no
...     ...           ...       ...        ...     ...       ...      ...
...
45206    51    technician   married   tertiary      no       825       no
no
45207    71       retired  divorced    primary      no      1729       no
no
45208    72       retired   married  secondary      no      5715       no
no
45209    57   blue-collar   married  secondary      no       668       no
no
45210    37  entrepreneur   married  secondary      no      2971       no
no

          contact  day month  duration  campaign  pdays  previous
poutcome      y
0         unknown    5   may       261         1     -1         0
unknown    no
1         unknown    5   may       151         1     -1         0
unknown    no
2         unknown    5   may        76         1     -1         0
unknown    no
3         unknown    5   may        92         1     -1         0
unknown    no
4         unknown    5   may       198         1     -1         0
unknown    no
...           ...  ...   ...       ...       ...    ...       ...
...  ...
45206    cellular   17   nov       977         3     -1         0
unknown   yes
45207    cellular   17   nov       456         2     -1         0
unknown   yes
45208    cellular   17   nov      1127         5    184         3
success   yes
45209   telephone   17   nov       508         4     -1         0
unknown    no
45210    cellular   17   nov       361         2    188        11
other    no

[45211 rows x 17 columns]
```

```
#read data_test
data_test = pd.read_csv(r"F:\my projects\Users\datasets\bank-
```

```
full.csv", sep=";")
data_test
```

```
       age           job   marital  education default  balance housing
loan  \
0       58    management   married   tertiary      no     2143     yes
no
1       44    technician    single  secondary      no       29     yes
no
2       33  entrepreneur   married  secondary      no        2     yes
yes
3       47   blue-collar   married    unknown      no     1506     yes
no
4       33       unknown    single    unknown      no        1      no
no
...    ...           ...       ...        ...     ...      ...     ...
...
45206   51    technician   married   tertiary      no      825      no
no
45207   71       retired  divorced    primary      no     1729      no
no
45208   72       retired   married  secondary      no     5715      no
no
45209   57   blue-collar   married  secondary      no      668      no
no
45210   37  entrepreneur   married  secondary      no     2971      no
no

         contact  day month  duration  campaign  pdays  previous
poutcome     y
0        unknown    5   may       261         1     -1         0
unknown    no
1        unknown    5   may       151         1     -1         0
unknown    no
2        unknown    5   may        76         1     -1         0
unknown    no
3        unknown    5   may        92         1     -1         0
unknown    no
4        unknown    5   may       198         1     -1         0
unknown    no
...          ...  ...   ...       ...       ...    ...       ...
...  ...
45206   cellular   17   nov       977         3     -1         0
unknown   yes
45207   cellular   17   nov       456         2     -1         0
unknown   yes
45208   cellular   17   nov      1127         5    184         3
success   yes
45209  telephone   17   nov       508         4     -1         0
unknown    no
```

```
45210   cellular    17   nov        361          2     188         11
other    no

[45211 rows x 17 columns]
```

Preprocessing the data

```
#Encoder categorial features
def categorize(df):
    new_df = df.copy()
    le = preprocessing.LabelEncoder()

    new_df["job"] = le.fit_transform(new_df["job"])
    new_df["marital"] = le.fit_transform(new_df["marital"])
    new_df["education"] = le.fit_transform(new_df["education"])
    new_df["default"] = le.fit_transform(new_df["default"])
    new_df["housing"] = le.fit_transform(new_df["housing"])
    new_df["month"] = le.fit_transform(new_df["month"])
    new_df["loan"] = le.fit_transform(new_df["loan"])
    new_df["contact"] = le.fit_transform(new_df["contact"])
    new_df["day_of_week"] = le.fit_transform(new_df["day_of_week"])
    new_df["poutcome"] = le.fit_transform(new_df["poutcome"])
    new_df["y"] = le.fit_transform(new_df["y"])
    return new_df

# concat and replace column with basic
data = pd.concat([data_train, data_test])
data.replace(["basic.6y","basic.4y", "basic.9y"], "basic",
inplace=True)
```

Checking for null values

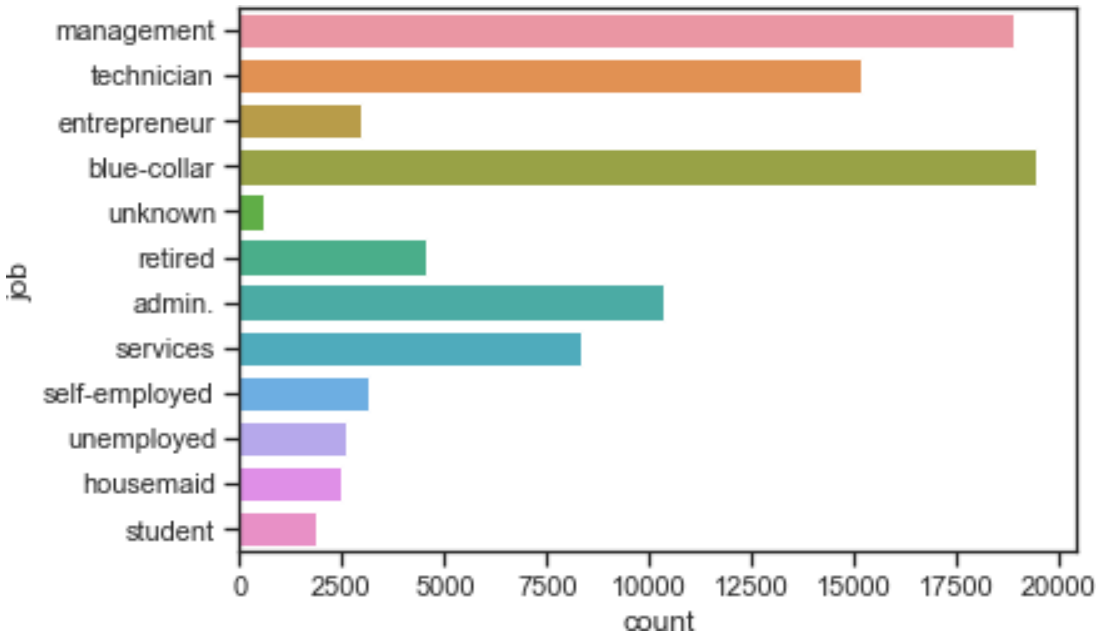
```
# check missing values
data.isnull().sum()
```

```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

Data Visualization

```
# Visualize job column
sns.set(style="ticks", color_codes=True)
sns.countplot(y="job", data=data)
```

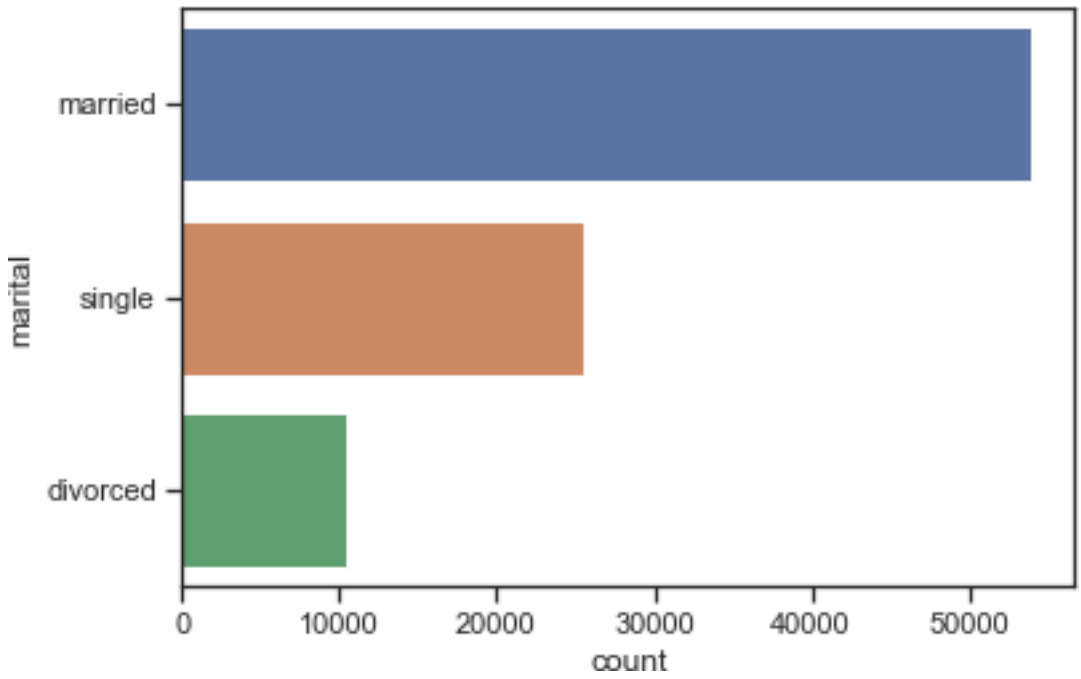
```
<AxesSubplot:xlabel="count", ylabel="job">
```




```
# select job is unknown
data = data[data.job != "unknown"]
```

```
# cisualize marital feature
sns.countplot(y="marital", data=data)
```

```
<AxesSubplot:xlabel="count", ylabel="marital">
```



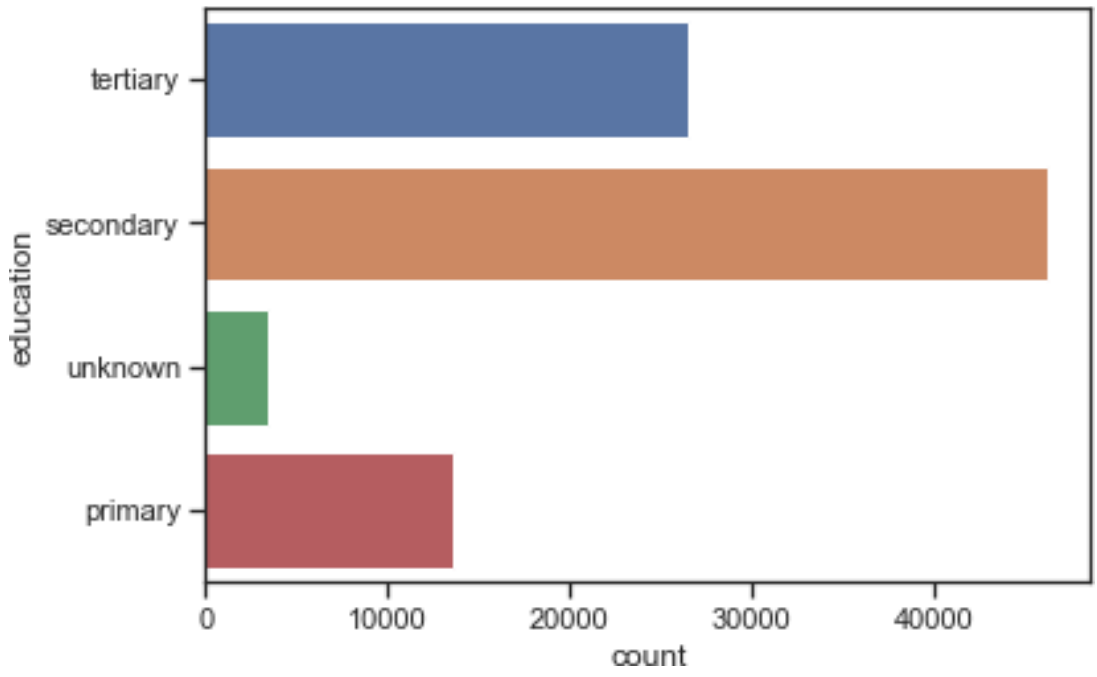

```
# count marital column values
data.marital.value_counts()
```

```
married     54022
single      25444
divorced    10380
Name: marital, dtype: int64
```

```
#select unknow values in marital and loan features
data = data[data.marital != "unknown"]
data = data[data.loan != "unknown"]
```

```
#visualize education column
sns.countplot(y="education", data=data)
```

```
<AxesSubplot:xlabel="count", ylabel="education">
```



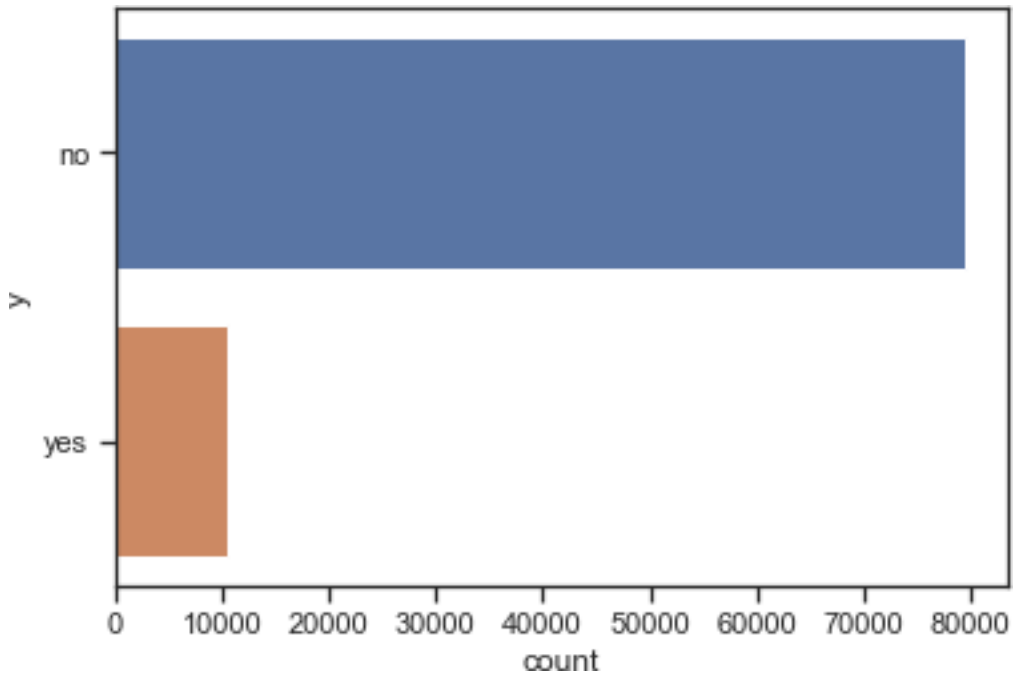

```
# select illiterate value from education
data = data[data.education != "illiterate"]

# describe data
data.describe()
```

```
                age        balance            day       duration
campaign  \
count   89846.000000   89846.000000   89846.000000   89846.000000
89846.000000
mean       40.893529    1359.643011      15.813882     258.294838
2.760345
std        10.604340    3045.074573       8.319994     257.712336
3.092820
min        18.000000   -8019.000000       1.000000       0.000000
1.000000
25%        33.000000      72.000000       8.000000     103.000000
1.000000
50%        39.000000     447.000000      16.000000     180.000000
2.000000
75%        48.000000    1421.000000      21.000000     319.000000
3.000000
max        95.000000  102127.000000      31.000000    4918.000000
63.000000

              pdays      previous
count  89846.000000  89846.000000
mean      40.321016      0.581996
std      100.254588      2.309064
```

```
min        -1.000000      0.000000
25%        -1.000000      0.000000
50%        -1.000000      0.000000
75%        -1.000000      0.000000
max       871.000000    275.000000
```

```
# visualize label column
sns.countplot(y="y", data=data)
```

```
<AxesSubplot:xlabel="count", ylabel="y">
```



Checking for outliers using boxplots

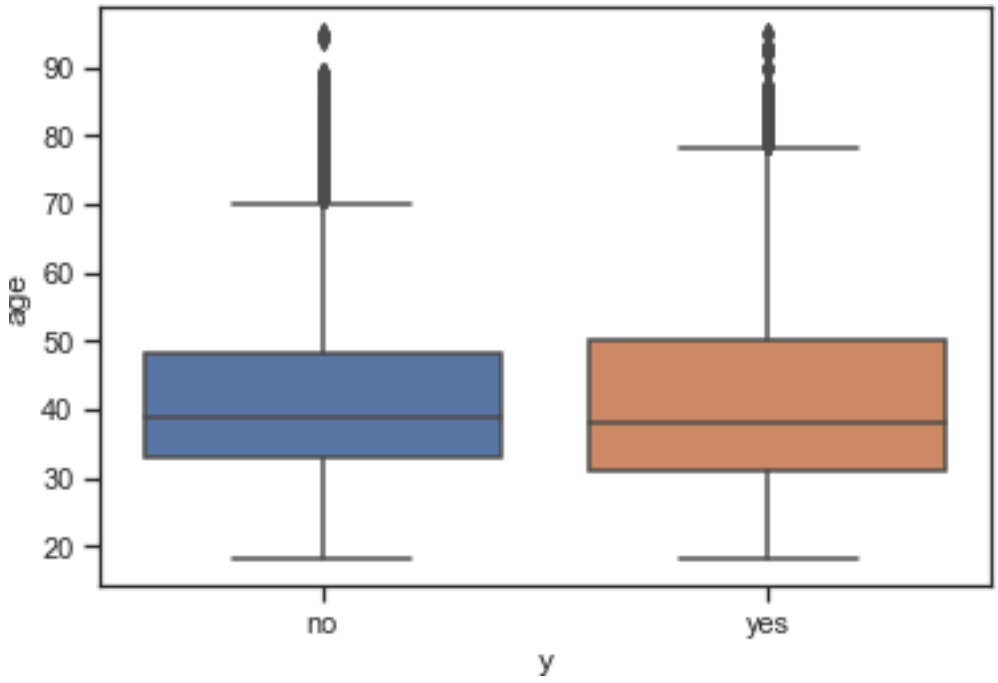
```
data.head()
```

```
   age           job  marital  education default  balance housing loan
\
0   58    management  married   tertiary      no     2143     yes   no
1   44    technician   single  secondary      no       29     yes   no
2   33  entrepreneur  married  secondary      no        2     yes  yes
3   47   blue-collar  married    unknown      no     1506     yes   no
5   35    management  married   tertiary      no      231     yes   no

   contact  day month  duration  campaign  pdays  previous poutcome
y
```

0	unknown	5	may	261	1	-1	0	unknown	no
1	unknown	5	may	151	1	-1	0	unknown	no
2	unknown	5	may	76	1	-1	0	unknown	no
3	unknown	5	may	92	1	-1	0	unknown	no
5	unknown	5	may	139	1	-1	0	unknown	no

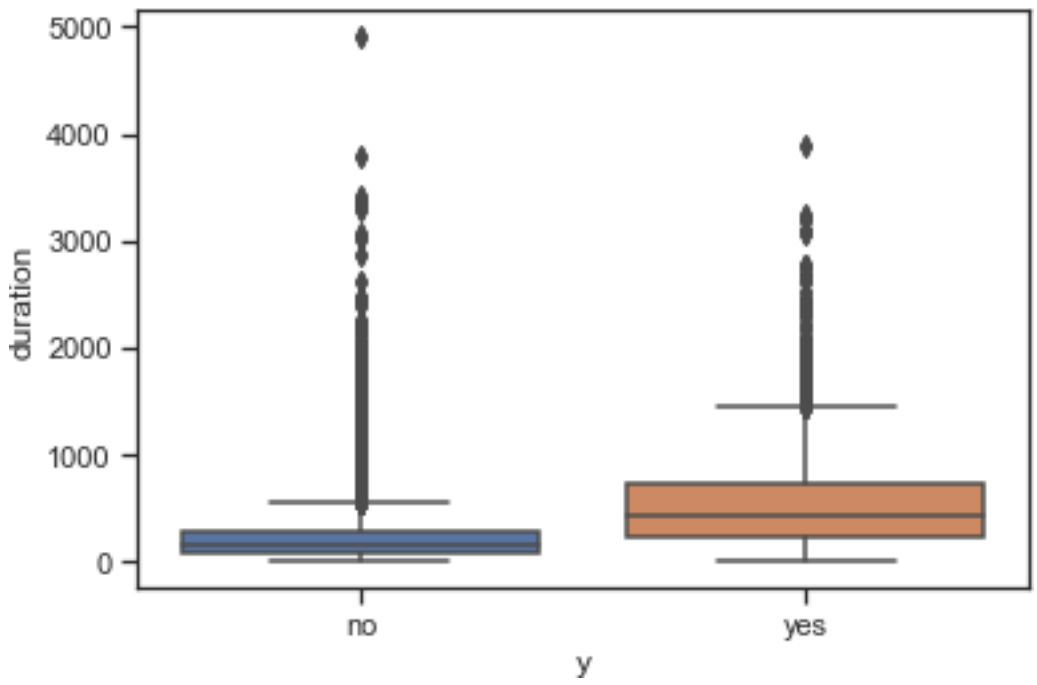
```
# visualize education column
sns.boxplot(x="y", y="age", data=data)
```

```
<AxesSubplot:xlabel="y", ylabel="age">
```




```
#visualize housing and label
sns.boxplot(x="y", y="duration", data=data)
```

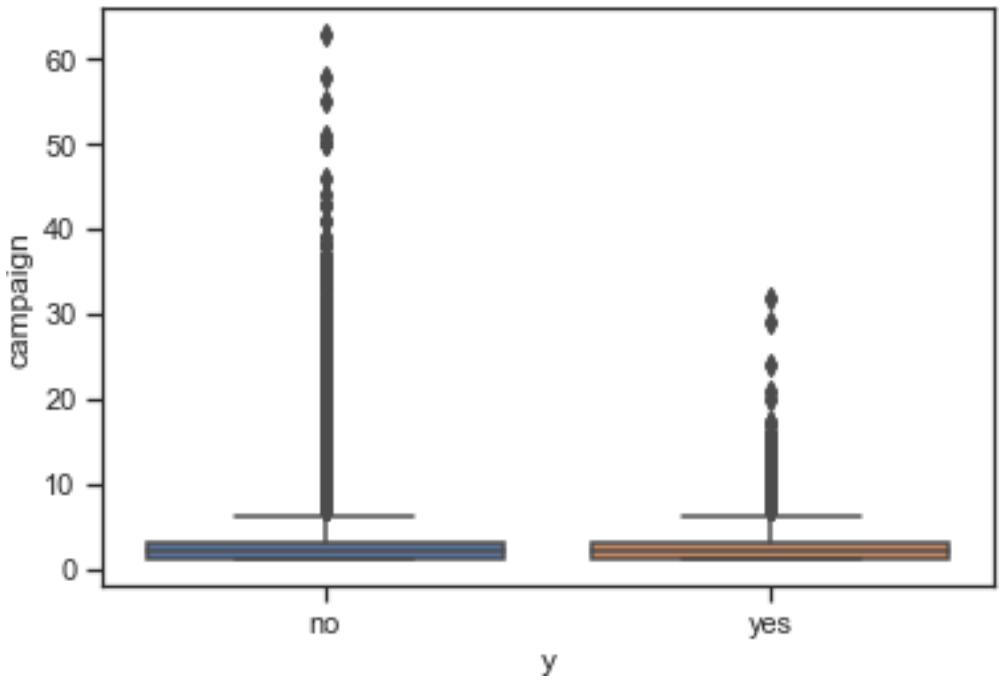
```
<AxesSubplot:xlabel="y", ylabel="duration">
```




```
# visualize label and job
sns.boxplot(data["y"],data["campaign"])
```

```
E:\Anaconda\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

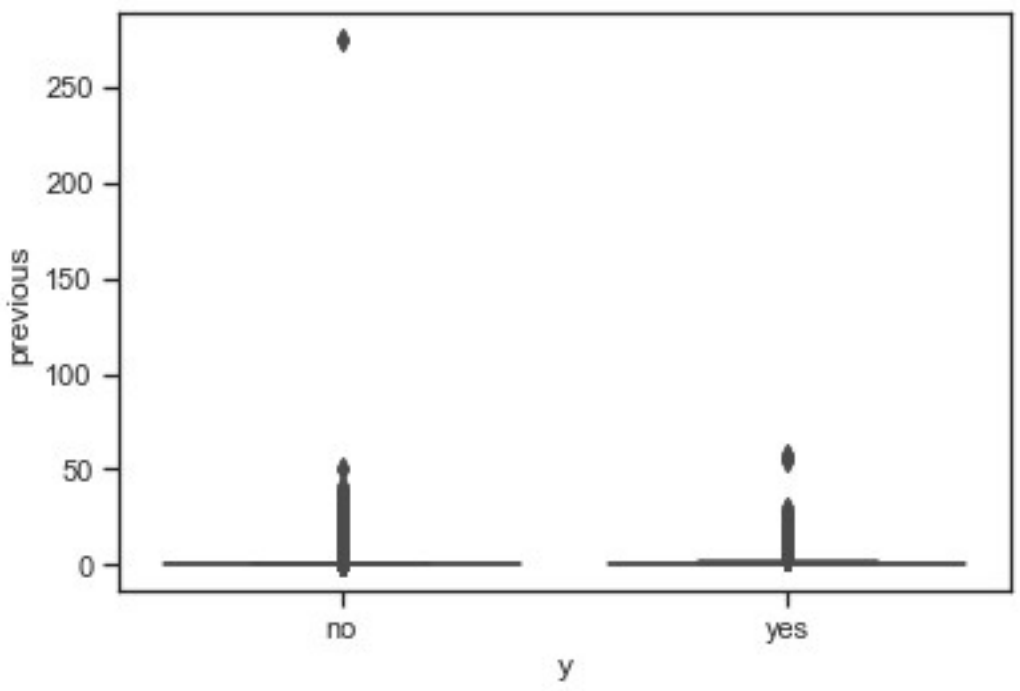
```
<AxesSubplot:xlabel="y", ylabel="campaign">
```




```
# visualize label and job
sns.boxplot(data["y"],data["previous"])
```

```
E:\Anaconda\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y.
From version 0.12, the only valid positional argument will be `data`,
and passing other arguments without an explicit keyword will result in
an error or misinterpretation.
  warnings.warn(
```

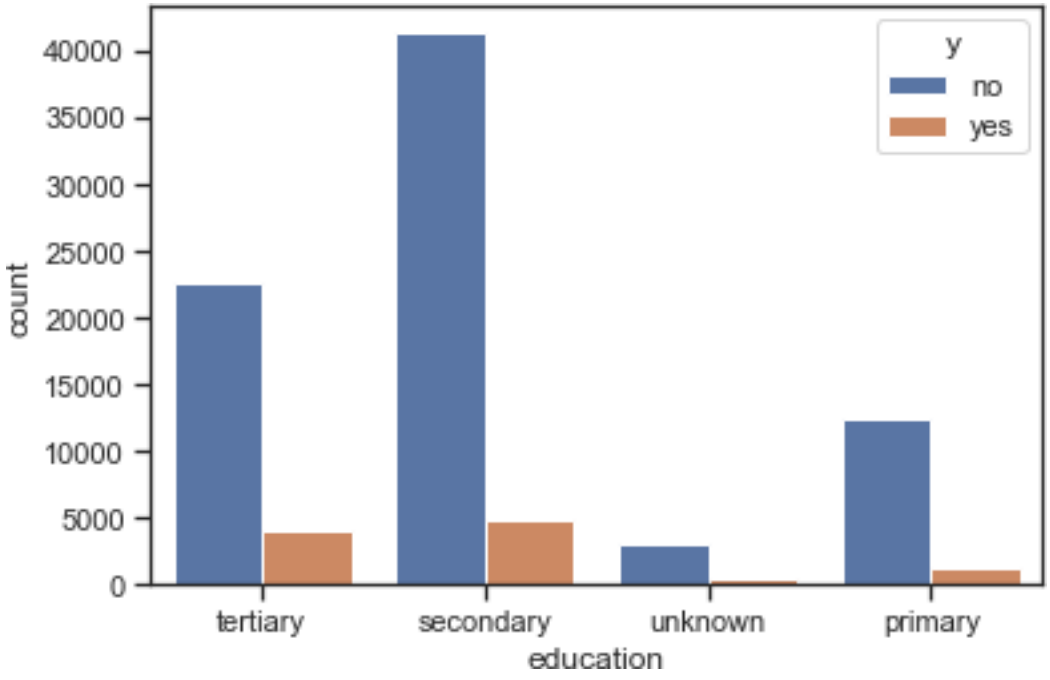
```
<AxesSubplot:xlabel="y", ylabel="previous">
```



Dropping less meaningful columns

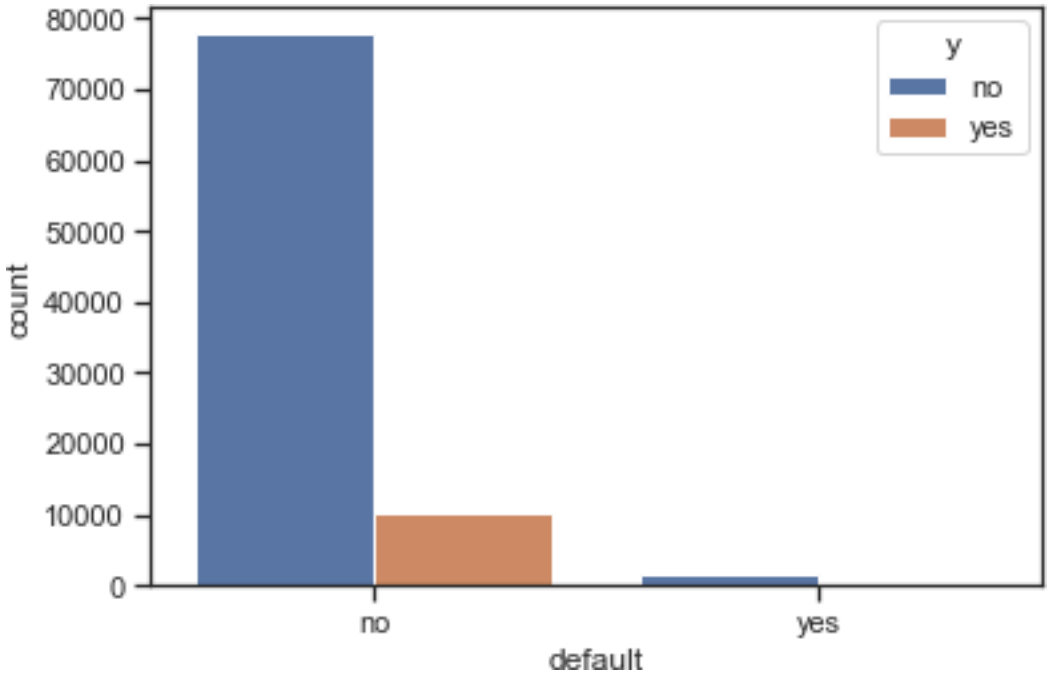
```
# visualize education and label
sns.countplot(x="education",hue="y",data=data)
```

```
<AxesSubplot:xlabel="education", ylabel="count">
```




```
# visualize default and label
sns.countplot(x="default",hue="y",data=data)
```

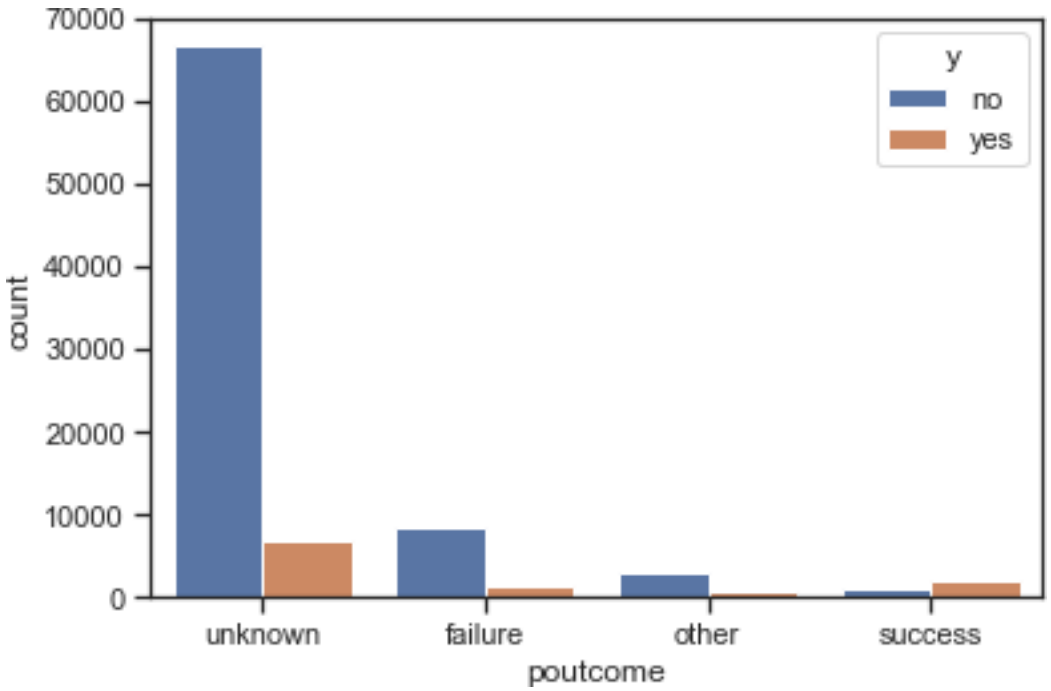
```
<AxesSubplot:xlabel="default", ylabel="count">
```



It is skewed to 0. So We can drop this.

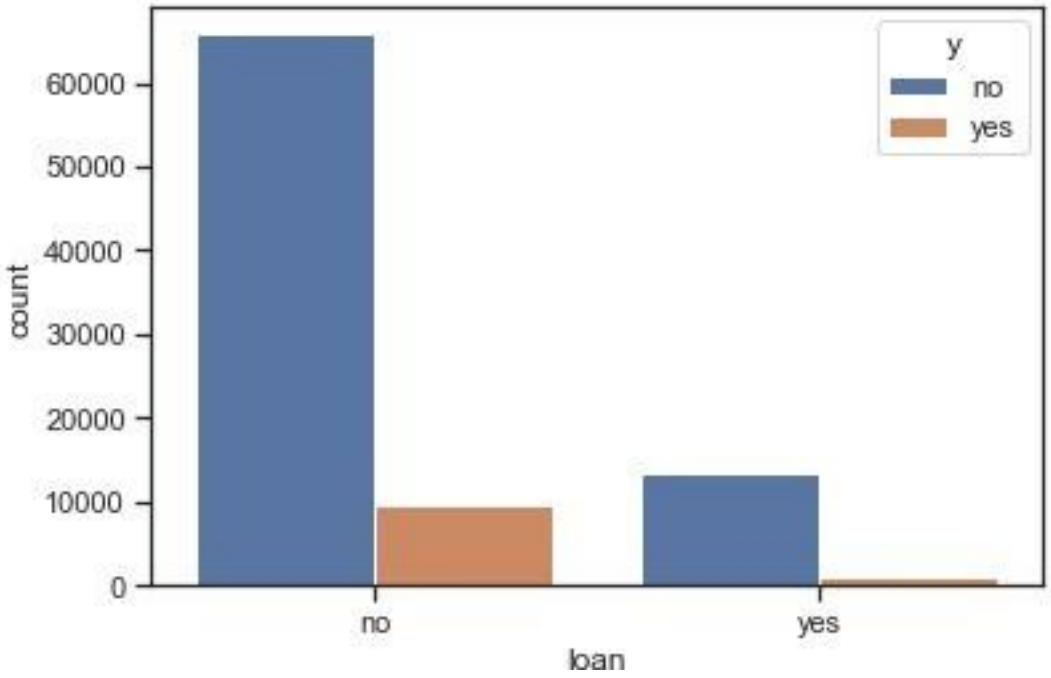
```
# drop default because of skewed
data = data.drop("default",axis=1)

# visualize poutcome and label
sns.countplot(x="poutcome",hue="y",data=data)
```

```
<AxesSubplot:xlabel="poutcome", ylabel="count">
```



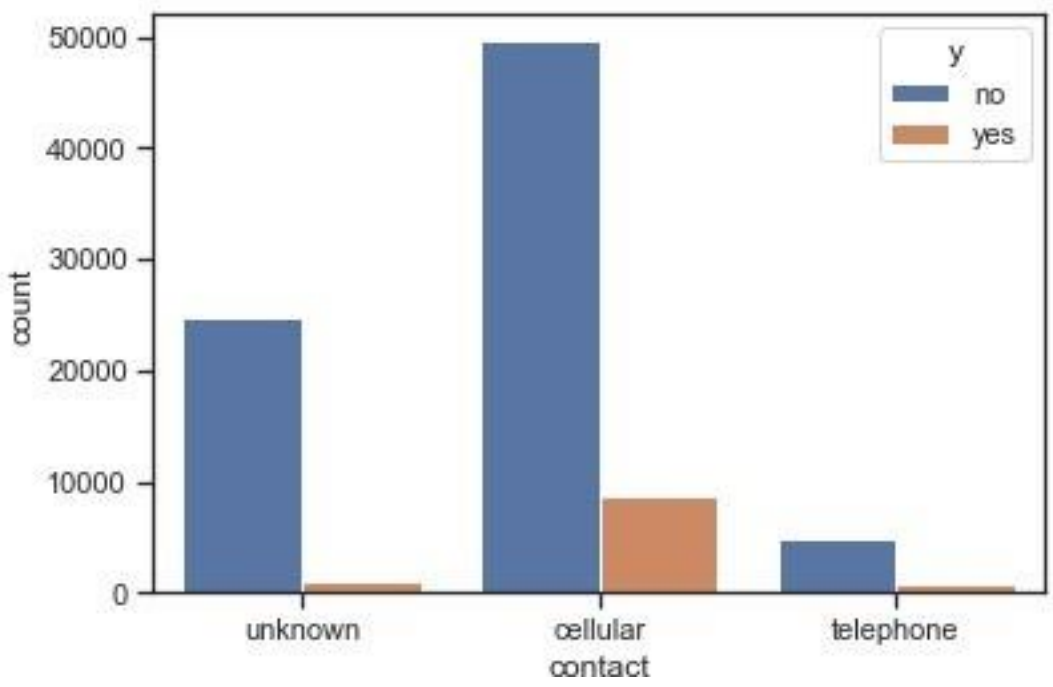

```
# drop poutcome
data = data.drop("poutcome",axis=1)

#visualize loan and label
sns.countplot(x="loan",hue="y",data=data)
```

```
<AxesSubplot:xlabel="loan", ylabel="count">
```



```
#visualize contact and label
sns.countplot(x="contact",hue="y",data=data)
```

```
<AxesSubplot:xlabel="contact", ylabel="count">
```



data

	age	job	marital	education	balance	housing	loan	\
0	58	management	married	tertiary	2143	yes	no	
1	44	technician	single	secondary	29	yes	no	
2	33	entrepreneur	married	secondary	2	yes	yes	
3	47	blue-collar	married	unknown	1506	yes	no	
5	35	management	married	tertiary	231	yes	no	
...	...	...	...	...	...	...	...	
45206	51	technician	married	tertiary	825	no	no	
45207	71	retired	divorced	primary	1729	no	no	
45208	72	retired	married	secondary	5715	no	no	
45209	57	blue-collar	married	secondary	668	no	no	
45210	37	entrepreneur	married	secondary	2971	no	no	

	contact	day	month	duration	campaign	pdays	previous	y
0	unknown	5	may	261	1	-1	0	no
1	unknown	5	may	151	1	-1	0	no
2	unknown	5	may	76	1	-1	0	no
3	unknown	5	may	92	1	-1	0	no

```
5          unknown    5    may        139          1     -1          0    no
...            ...  ...    ...        ...        ...    ...        ...   ...
45206     cellular   17    nov        977          3     -1          0   yes
45207     cellular   17    nov        456          2     -1          0   yes
45208     cellular   17    nov       1127          5    184          3   yes
45209    telephone   17    nov        508          4     -1          0    no
45210     cellular   17    nov        361          2    188         11    no

[89846 rows x 15 columns]
```

```
#drop contact
data = data.drop("contact",axis=1)

#drop unnecessary column
data =
data.drop(["housing","loan","job","duration","month","day","marital","
education"],axis=1)

data
```

```
       age  balance  campaign  pdays  previous    y
0       58     2143         1     -1         0   no
1       44       29         1     -1         0   no
2       33        2         1     -1         0   no
3       47     1506         1     -1         0   no
5       35      231         1     -1         0   no
...    ...      ...       ...    ...       ...  ...
45206   51      825         3     -1         0  yes
45207   71     1729         2     -1         0  yes
45208   72     5715         5    184         3  yes
45209   57      668         4     -1         0   no
45210   37     2971         2    188        11   no

[89846 rows x 6 columns]
```

Splitting into train and test data

```
#split data into train and test data
X = data.drop("y",axis = 1).values
y = data["y"].values
X_train, X_test, Y_train, Y_test = train_test_split(X, y,
test_size=0.20, random_state=42)
```

```
#Scaler data train
scaler = StandardScaler()
scaler.fit(X_train)
X_train = scaler.fit_transform(X_train)

#Feature selection with PCA
pca = PCA(n_components=5)
pca.fit(X_train)
X_train = pca.fit_transform(X_train)

X_train.shape
```

```
(71876, 5)
```

Building different Models and validating using kfold cross validation

```
# append all model into an array
models = []
models.append(("LogReg", LogisticRegression()))
models.append(("Decison-Tree", DecisionTreeClassifier()))
models.append(("Naive Bayes", GaussianNB()))
models.append(("KNN", KNeighborsClassifier()))
models.append(("RandForest",RandomForestClassifier()))
models.append(("SVM", SVC()))

# do kfold cross validation into modelling
results = []
names = []
for name, model in models:
    kfold = model_selection.KFold(n_splits=5,
random_state=42,shuffle=True)
    cv_results = model_selection.cross_val_score(model, X_train,
Y_train, cv=kfold, scoring="accuracy")
    results.append(cv_results)
    names.append(name)
    msg = "{}: {}".format(name, cv_results.mean())
    print(msg)
```

```
LogReg: 0.8825338075536522
Decison-Tree: 0.9270131697757131
Naive Bayes: 0.859953155742663
KNN: 0.8814764675425225
RandForest: 0.9370721666545693
SVM: 0.8837164114103215
```

Testing with the test data

```
scaler.fit(X_test)
X_test = scaler.fit_transform(X_test)

pca.fit(X_test)
X_test = pca.fit_transform(X_test)
```

```
RandFor = RandomForestClassifier()
RandFor.fit(X_train, Y_train)
predictions = RandFor.predict(X_test)
print("Accuracy : ", accuracy_score(Y_test, predictions))
print("Confusion Matrix : \n",confusion_matrix(Y_test, predictions))
print("Classification Report: \n",classification_report(Y_test,
predictions))
```

```
Accuracy :  0.847968836950473
Confusion Matrix :
 [[14894   959]
 [ 1773   344]]
Classification Report:
              precision    recall  f1-score   support

          no       0.89      0.94      0.92     15853
         yes       0.26      0.16      0.20      2117

    accuracy                           0.85     17970
   macro avg       0.58      0.55      0.56     17970
weighted avg       0.82      0.85      0.83     17970
```

```
LogisticRegression = LogisticRegression()
LogisticRegression.fit(X_train, Y_train)
predictions = LogisticRegression.predict(X_test)
print("Accuracy : ", accuracy_score(Y_test, predictions))
print("Confusion Matrix : \n",confusion_matrix(Y_test, predictions))
print("Classification Report: \n",classification_report(Y_test,
predictions))
```

```
Accuracy :  0.8809682804674458
Confusion Matrix :
 [[15827    26]
 [ 2113     4]]
Classification Report:
              precision    recall  f1-score   support

          no       0.88      1.00      0.94     15853
         yes       0.13      0.00      0.00      2117

    accuracy                           0.88     17970
   macro avg       0.51      0.50      0.47     17970
weighted avg       0.79      0.88      0.83     17970
```

```
DecisionTreeClassifier = DecisionTreeClassifier()
DecisionTreeClassifier.fit(X_train, Y_train)
predictions = DecisionTreeClassifier.predict(X_test)
```

```
print("Accuracy : ", accuracy_score(Y_test, predictions))
print("Confusion Matrix : \n",confusion_matrix(Y_test, predictions))
print("Classification Report: \n",classification_report(Y_test,
predictions))
```

```
Accuracy :  0.7208124652198108
Confusion Matrix :
 [[12263  3590]
 [ 1427   690]]
Classification Report:
              precision    recall  f1-score   support

          no       0.90      0.77      0.83     15853
         yes       0.16      0.33      0.22      2117

    accuracy                           0.72     17970
   macro avg       0.53      0.55      0.52     17970
weighted avg       0.81      0.72      0.76     17970
```

```
GaussianNB = GaussianNB()
GaussianNB.fit(X_train, Y_train)
predictions = GaussianNB.predict(X_test)
print("Accuracy : ", accuracy_score(Y_test, predictions))
print("Confusion Matrix : \n",confusion_matrix(Y_test, predictions))
print("Classification Report: \n",classification_report(Y_test,
predictions))
```

```
Accuracy :  0.8583750695603785
Confusion Matrix :
 [[15205   648]
 [ 1897   220]]
Classification Report:
              precision    recall  f1-score   support

          no       0.89      0.96      0.92     15853
         yes       0.25      0.10      0.15      2117

    accuracy                           0.86     17970
   macro avg       0.57      0.53      0.54     17970
weighted avg       0.81      0.86      0.83     17970
```

```
KNeighborsClassifier = KNeighborsClassifier()
KNeighborsClassifier.fit(X_train, Y_train)
predictions = KNeighborsClassifier.predict(X_test)
print("Accuracy : ", accuracy_score(Y_test, predictions))
print("Confusion Matrix : \n",confusion_matrix(Y_test, predictions))
print("Classification Report: \n",classification_report(Y_test,
predictions))
```

```
Accuracy :  0.8730662214802448
Confusion Matrix :
 [[15264   589]
 [ 1692   425]]
Classification Report:
              precision    recall  f1-score   support

          no       0.90      0.96      0.93     15853
         yes       0.42      0.20      0.27      2117

    accuracy                           0.87     17970
   macro avg       0.66      0.58      0.60     17970
weighted avg       0.84      0.87      0.85     17970
```

```
svm = SVC()
svm.fit(X_train, Y_train)
predictions = svm.predict(X_test)
print("Accuracy : ", accuracy_score(Y_test, predictions))
print("Confusion Matrix : \n",confusion_matrix(Y_test, predictions))
print("Classification Report: \n",classification_report(Y_test,
predictions))
```

```
Accuracy :  0.8824707846410684
Confusion Matrix :
 [[15823    30]
 [ 2082    35]]
Classification Report:
              precision    recall  f1-score   support

          no       0.88      1.00      0.94     15853
         yes       0.54      0.02      0.03      2117

    accuracy                           0.88     17970
   macro avg       0.71      0.51      0.48     17970
weighted avg       0.84      0.88      0.83     17970
```

```
print("Score:  \n", svm.score(X,y))
print("Score:  \n", KNeighborsClassifier.score(X,y))
print("Score:  \n", GaussianNB.score(X,y))
print("Score:  \n", DecisionTreeClassifier.score(X,y))
print("Score:  \n", LogisticRegression.score(X,y))
print("Score: \n", RandFor.score(X,y))
#Y_test, predictions
```

```
Score:
 0.8830220599692807
Score:
 0.7342786545867374
```

Score:
0.11960465685728915
Score:
0.672417247289807
Score:
0.1815105847783986
Score:
0.6728624535315985

Compare all the R2 Scores and execution speed to decide the best one

As we compare all the scores and execution speed- SVC and RandomForestClassifier are taking much more time to execute and LogisticRegression, DecisionTreeClasifier, GaussianNB, KNeighborsClassifier are taking less time execution speed.SVC and RandomForestClassifier are having better accuracy but these are taking much more time to execute. So when compare with LogisticRegression, DecisionTreeClasifier, GaussianNB, KNeighborsClassifier; LogisticRegression & KNeighborsClassifier is having better accuracy when compare with DecisionTreeClasifier, GaussianNB. So, I conclude that LogisticRegression & KNeighborsClassifier are best models for this dataset with accuracy and execution speed.

Thank you

D.Srikanth

SURE Trust

(G-17 Python & ML)

```
#Library required for K means Clustering model
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from pylab import rcParams
#sklearn
import sklearn
from sklearn.cluster import KMeans
from sklearn.preprocessing import scale # for scaling the data
import sklearn.metrics as sm # for evaluating the model
from sklearn import datasets
from sklearn.metrics import confusion_matrix,classification_report

#Define the Parameters for the Visualization
%matplotlib inline
rcParams["figure.figsize"] =20,10

#Read & Load Dataset
data = pd.read_csv(r"F:\my projects\Users\datasets\bank-
full.csv",delimiter=";")
data
```

```
       age           job   marital  education default  balance housing
loan  \
0       58    management   married   tertiary      no     2143     yes
no
1       44    technician    single  secondary      no       29     yes
no
2       33  entrepreneur   married  secondary      no        2     yes
yes
3       47   blue-collar   married    unknown      no     1506     yes
no
4       33       unknown    single    unknown      no        1      no
no
...    ...           ...       ...        ...     ...      ...     ...
...
45206   51    technician   married   tertiary      no      825      no
no
45207   71       retired  divorced    primary      no     1729      no
no
45208   72       retired   married  secondary      no     5715      no
no
45209   57   blue-collar   married  secondary      no      668      no
no
45210   37  entrepreneur   married  secondary      no     2971      no
no

         contact  day month  duration  campaign  pdays  previous
poutcome    y
0        unknown    5   may       261         1     -1         0
```

```
unknown    no
1          unknown    5    may        151          1     -1          0
unknown    no
2          unknown    5    may         76          1     -1          0
unknown    no
3          unknown    5    may         92          1     -1          0
unknown    no
4          unknown    5    may        198          1     -1          0
unknown    no
...            ...  ...    ...        ...        ...    ...        ...
...  ...
45206   cellular   17    nov        977          3     -1          0
unknown  yes
45207   cellular   17    nov        456          2     -1          0
unknown  yes
45208   cellular   17    nov       1127          5    184          3
success  yes
45209  telephone   17    nov        508          4     -1          0
unknown    no
45210   cellular   17    nov        361          2    188         11
other    no

[45211 rows x 17 columns]
```

```
#Load and scale the Dataset
iris = datasets.load_iris()
#scale the data
data = scale(iris.data) # scale the iris data
target = pd.DataFrame(iris.target) # define the target
variable_names = iris.feature_names
data[0:10]
```

```
array([[-0.90068117,  1.01900435, -1.34022653, -1.3154443 ],
       [-1.14301691, -0.13197948, -1.34022653, -1.3154443 ],
       [-1.38535265,  0.32841405, -1.39706395, -1.3154443 ],
       [-1.50652052,  0.09821729, -1.2833891 , -1.3154443 ],
       [-1.02184904,  1.24920112, -1.34022653, -1.3154443 ],
       [-0.53717756,  1.93979142, -1.16971425, -1.05217993],
       [-1.50652052,  0.78880759, -1.34022653, -1.18381211],
       [-1.02184904,  0.78880759, -1.2833891 , -1.3154443 ],
       [-1.74885626, -0.36217625, -1.34022653, -1.3154443 ],
       [-1.14301691,  0.09821729, -1.2833891 , -1.44707648]])
```

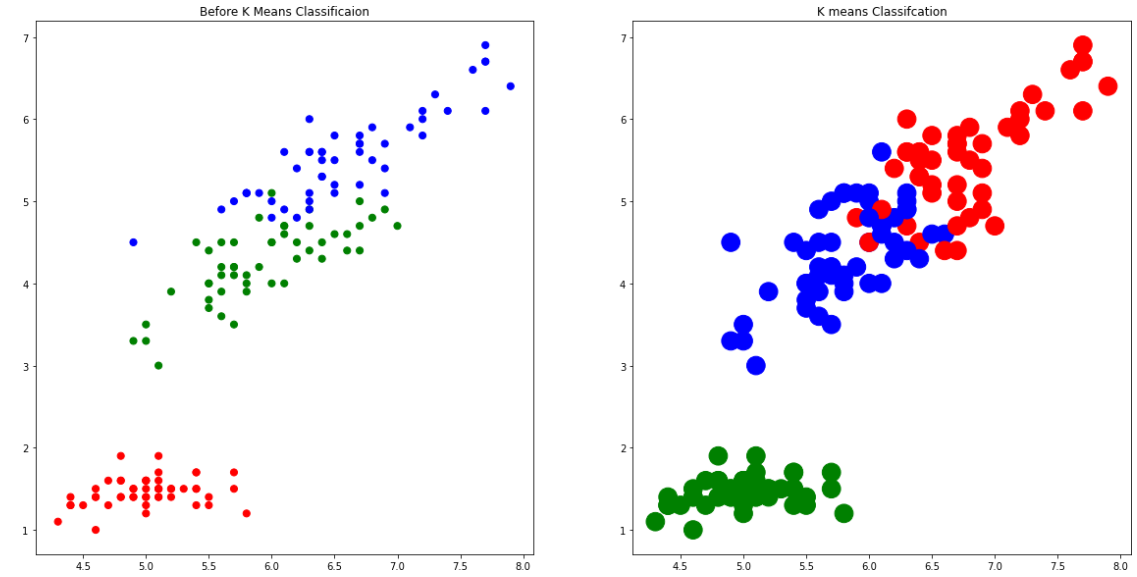
```
clustering = KMeans(n_clusters=3,random_state=5)
#fit the dataset
clustering.fit(data)
```

```
E:\Anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:870:
FutureWarning: The default value of `n_init` will change from 10 to
"auto" in 1.4. Set the value of `n_init` explicitly to suppress the
warning
```

```
  warnings.warn(
E:\Anaconda\lib\site-packages\sklearn\cluster\_kmeans.py:1382:
UserWarning: KMeans is known to have a memory leak on Windows with
MKL, when there are less chunks than available threads. You can avoid
it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(

KMeans(n_clusters=3, random_state=5)

#Build the Cluster Model and model the output
iris_df = pd.DataFrame(iris.data)
iris_df.columns = ["age","balance","duration","pdays" ]
target.columns =["y"]
```

Plot the Model Output using Matplotlib

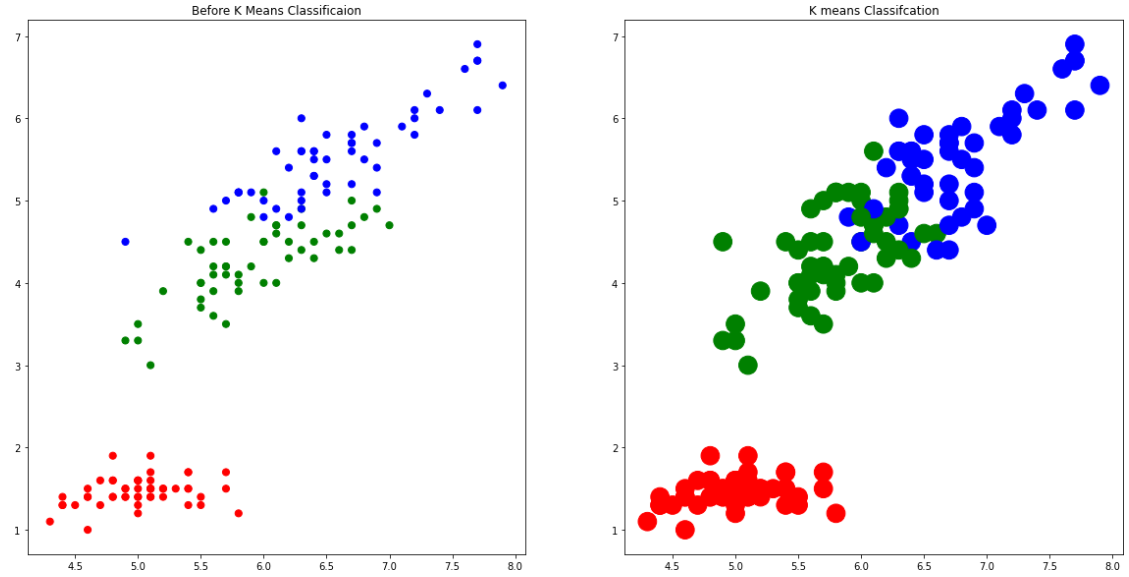
```
# Plotting between age and duration
colors = np.array(["Red","Green","Blue"])
plt.subplot(1,2,1)
plt.scatter(x=iris_df["age"] ,y= iris_df["duration"],c =
colors[iris.target],s=50)
plt.title("Before K Means Classificaion")

plt.subplot(1,2,2)
plt.scatter(x=iris_df["age"] ,y= iris_df["duration"],c =
colors[clustering.labels_],s=350)
plt.title("K means Classifcation")

Text(0.5, 1.0, "K means Classifcation")
```




```
#ploting between age and duration
relabel = np.choose(clustering.labels_,[2,0,1]).astype(np.int64)
colors = np.array(["Red","Green","Blue"])
plt.subplot(1,2,1)
```

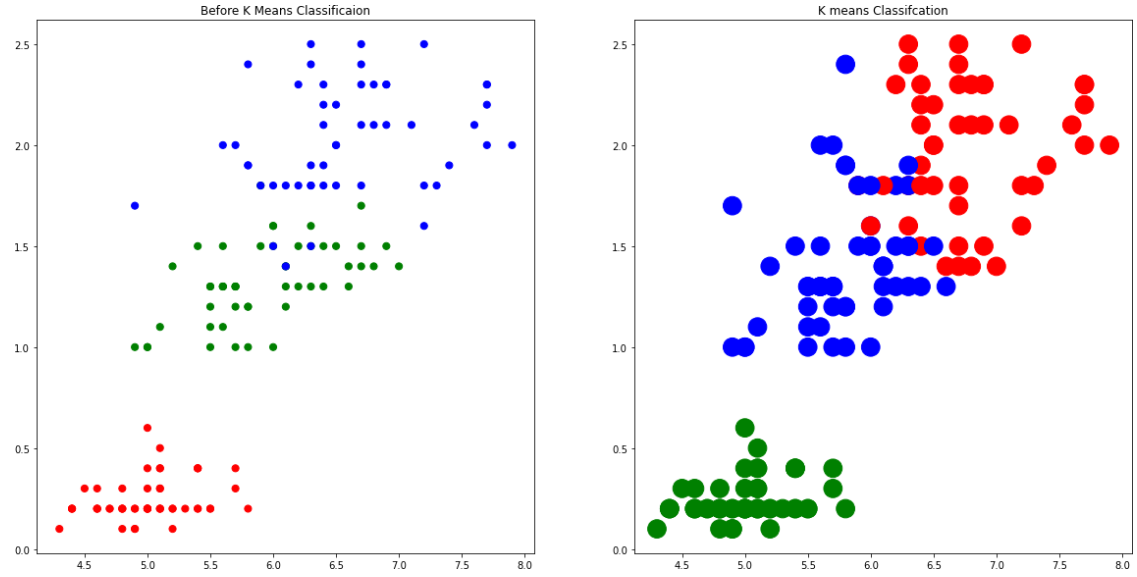
```
plt.scatter(x=iris_df.age ,y= iris_df.duration,c =
colors[iris.target],s=50)
plt.title("Before K Means Classificaion")

plt.subplot(1,2,2)
plt.scatter(x=iris_df.age ,y= iris_df.duration,c =
colors[relabel],s=350)
plt.title("K means Classifcation")
```

```
Text(0.5, 1.0, "K means Classifcation")
```

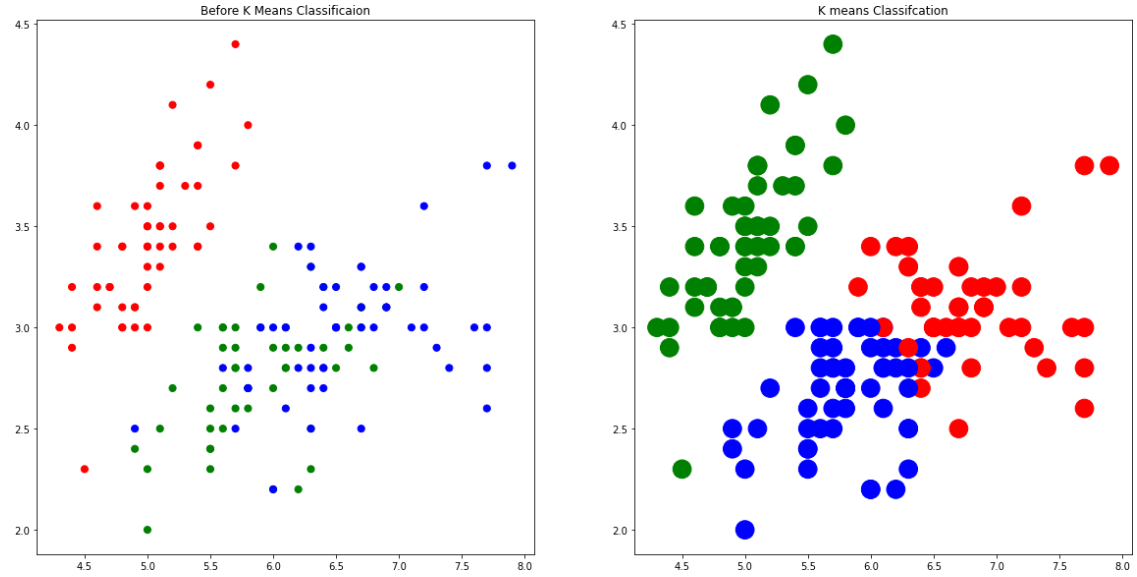



```
#ploting between age and pdays
plt.subplot(1,2,1)
plt.scatter(x=iris_df["age"] ,y= iris_df["pdays"],c =
colors[iris.target],s=50)
plt.title("Before K Means Classificaion")

plt.subplot(1,2,2)
plt.scatter(x=iris_df["age"] ,y= iris_df["pdays"],c =
colors[clustering.labels_],s=350)
plt.title("K means Classifcation")
```

```
Text(0.5, 1.0, "K means Classifcation")
```



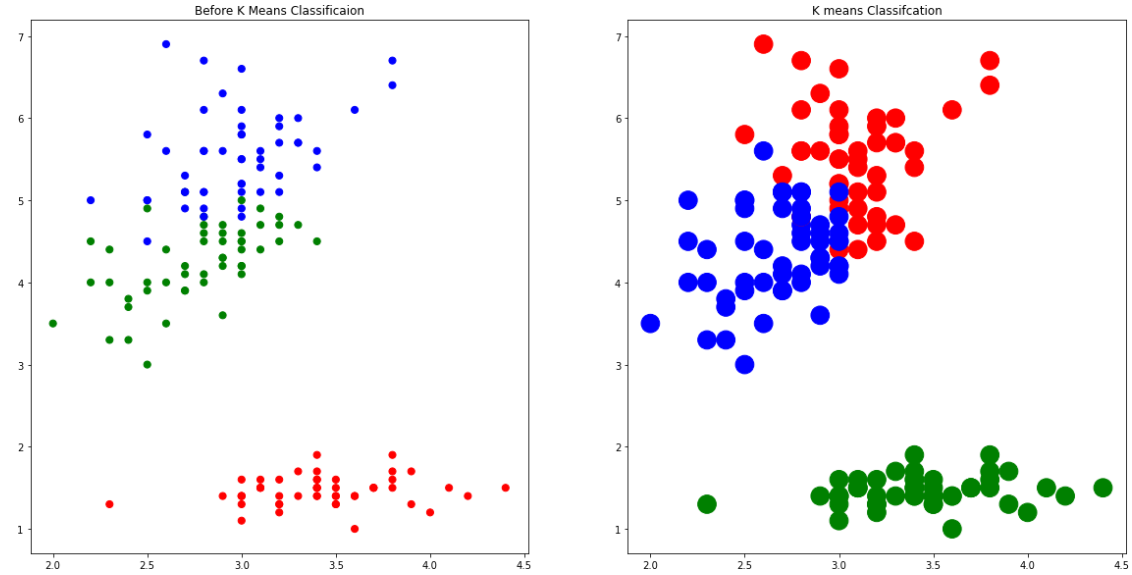

```
#ploting betweeen age and balance
plt.subplot(1,2,1)
plt.scatter(x=iris_df["age"] ,y= iris_df["balance"],c =
colors[iris.target],s=50)
plt.title("Before K Means Classificaion")

plt.subplot(1,2,2)
plt.scatter(x=iris_df["age"] ,y= iris_df["balance"],c =
colors[clustering.labels_],s=350)
plt.title("K means Classifcation")

Text(0.5, 1.0, "K means Classifcation")
```




```
#ploting between balance and duration
plt.subplot(1,2,1)
```

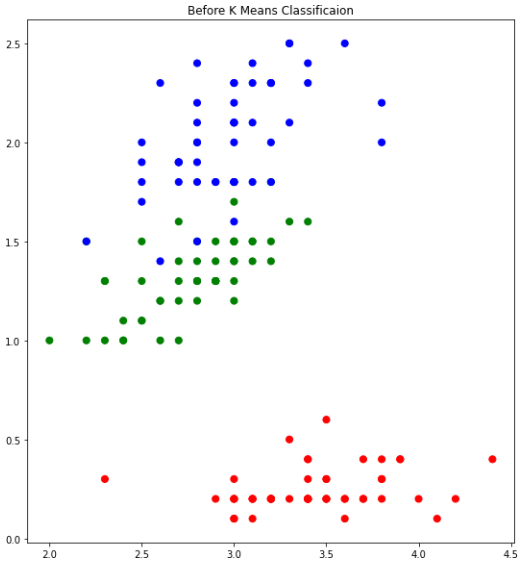
```
plt.scatter(x=iris_df["balance"] ,y= iris_df["duration"],c =
colors[iris.target],s=50)
plt.title("Before K Means Classificaion")

plt.subplot(1,2,2)
plt.scatter(x=iris_df["balance"] ,y= iris_df["duration"],c =
colors[clustering.labels_],s=350)
plt.title("K means Classifcation")
```

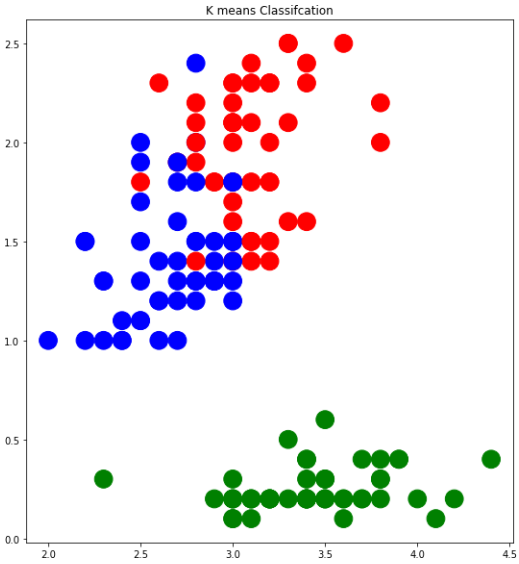
```
Text(0.5, 1.0, "K means Classifcation")
```




```
#ploting between balance and pdays
plt.subplot(1,2,1)
plt.scatter(x=iris_df["balance"] ,y= iris_df["pdays"],c =
colors[iris.target],s=50)
plt.title("Before K Means Classificaion")

plt.subplot(1,2,2)
plt.scatter(x=iris_df["balance"] ,y= iris_df["pdays"],c =
colors[clustering.labels_],s=350)
plt.title("K means Classifcation")
```

```
Text(0.5, 1.0, "K means Classifcation")
```


Before K Means Classificaion
2.5
2.0
1.5
1.0
0.5
0.0
2.0
2.5
3.0
3.5
4.0
4.5


K means Classifcation
2.5
2.0
1.5
1.0
0.5
0.0
2.0
2.5
3.0
3.5
4.0
4.5

Feedback:
First of all, I am Dubbaka Srikanth, I am very much thankful to Prof. Ch. Radhakumari madam for the wonderful opportunity she gave to upgrade my skills. SURE Trust is doing a great service by providing free courses to students. During the course we not only learned the skills but also discipline, dedication and attentiveness as well. SURE Trust is also conducting LST Sessions on Sundays which is more useful for us in our day today life.
Our Trainer Mr. Ashwani Ahlawat is one of the best we got as he teaches the course in a very understandable manner. Mr. Ashwani Ahlawat is always on time for the session and he never skipped any classes. Mr. Ashwani Ahlawat probes us with questions during the session and clears our doubts when we are unable to understand on the topic. Mr. Ashwani Ahlawat gave us assignments on every topic and helps us to finish it if we are stuck or unable to complete it. Thank you so much Mr. Ashwani Ahlawat
Heartly wishing SURE TRUST & Mr. Ashwani Ahlawat for their future endeavors
4. Uniqueness of the Course
* Courses are thought practically and explained in detail.
*Machine Learning is the goal of our course and it was thought in an easy understandable manner.
* Assignments are given on every topic which made us to learn.
* Concentrated on every student and thought us discipline along with the course.
* Made us to prepare the student's report which will be helpful for our future reference.
5. Concluding Remarks
I am very much happy that I have learnt the course successfully. I am blessed to learn the course in SURE TRUST. I would like to thank each and every person of SURE TRUST for providing me an opportunity to learn the course. SURE TRUST is helping the students to gain the knowledge on latest technology and helps them to get good jobs. I am proud to be a student of SURE TRUST.