**CHEMBUR TROMABY EDUCATION SOCIETY**

**N. G. ACHARYA & D. K. MARATHE COLLEGE**

**OF ARTS, SCIENCE & COMMERCE**

**N. G. Acharya Marg, Chembur, Mumbai – 400 071.**

**NAAC ACCREDITED COLLEGE**

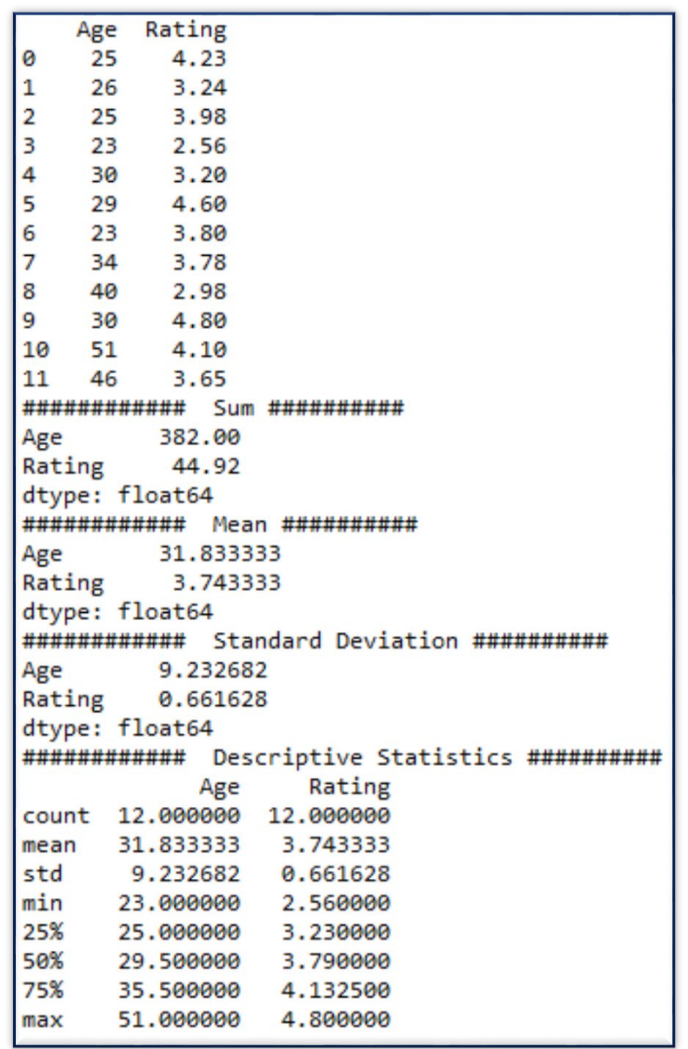
**M.Sc ( Computer Science /Information Technology)**



Certificate

Certified that the work entered in this journal was done in the Computer laboratory by the student Mr. / Miss Korde Suraj Navanath Roll No. 7 Of Class M.Sc. IT & A Semester I During the Year 2021-2022 in a Satisfactory manner.

Lecture Incharge External Examiner Head of Dept



# Practical 1:

## Write a program for obtaining descriptive statistics of data.

################################################################

#Practical 1A: Write a python program on descriptive statistics analysis. ################################################################

import pandas as pd

#Create a Dictionary of series

d = {'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]), 'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65])}

#Create a DataFrame df = pd.DataFrame(d) print(df)

print('############ Sum ########## ')

print (df.sum())

print('############ Mean ########## ')

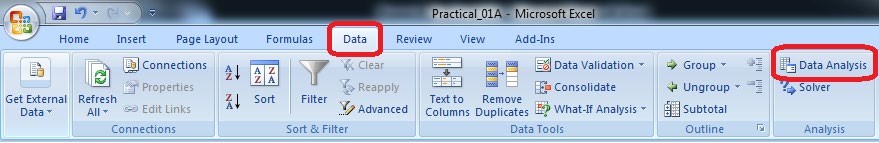
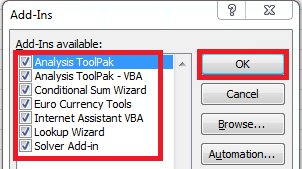
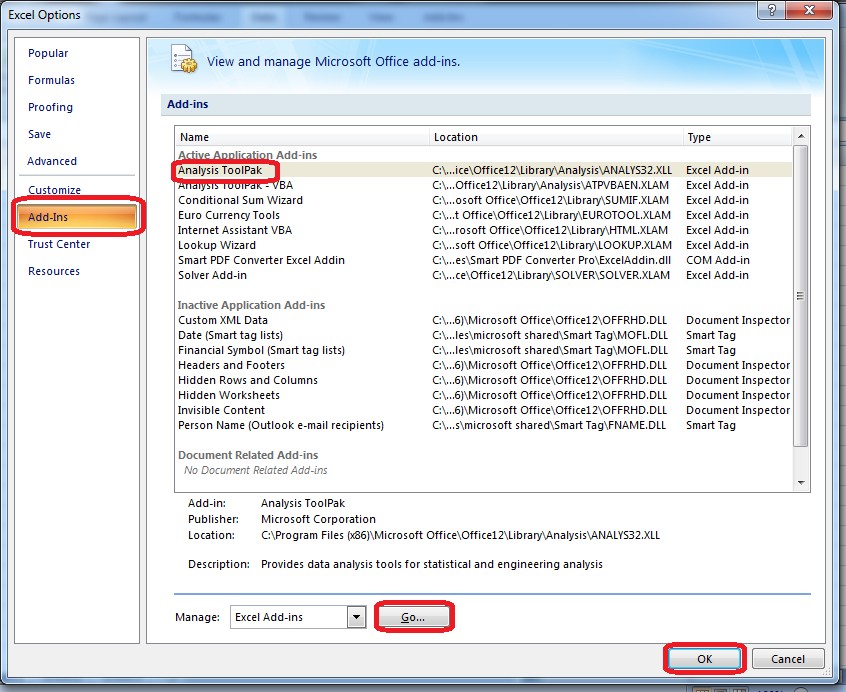
print (df.mean())

print('############ Standard Deviation ########## ') print (df.std())

print('############ Descriptive Statistics ########## ') print (df.describe())

**Output:**

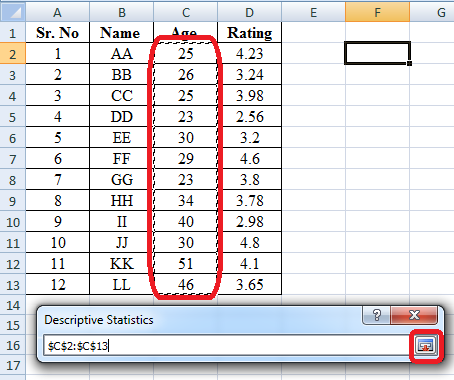
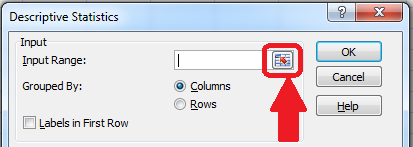
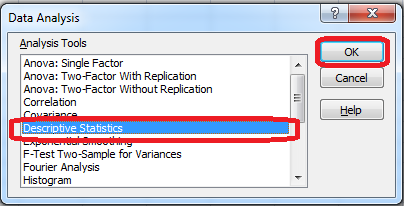
2



#### Using Excel

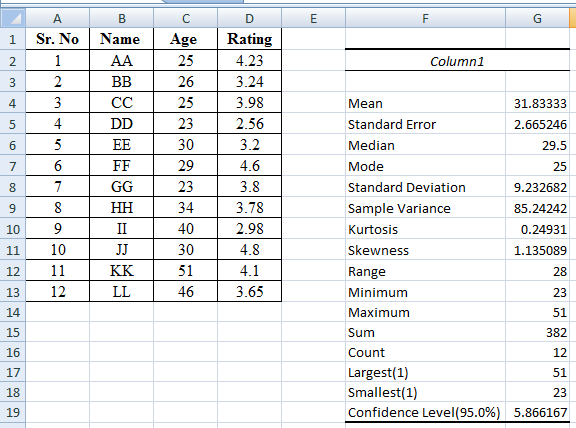
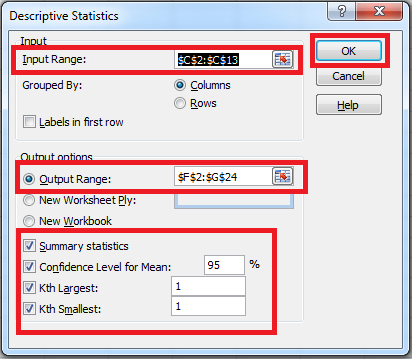
Go to File Menu  Options  Add-Ins Select Analysis ToolPak Press OK

3



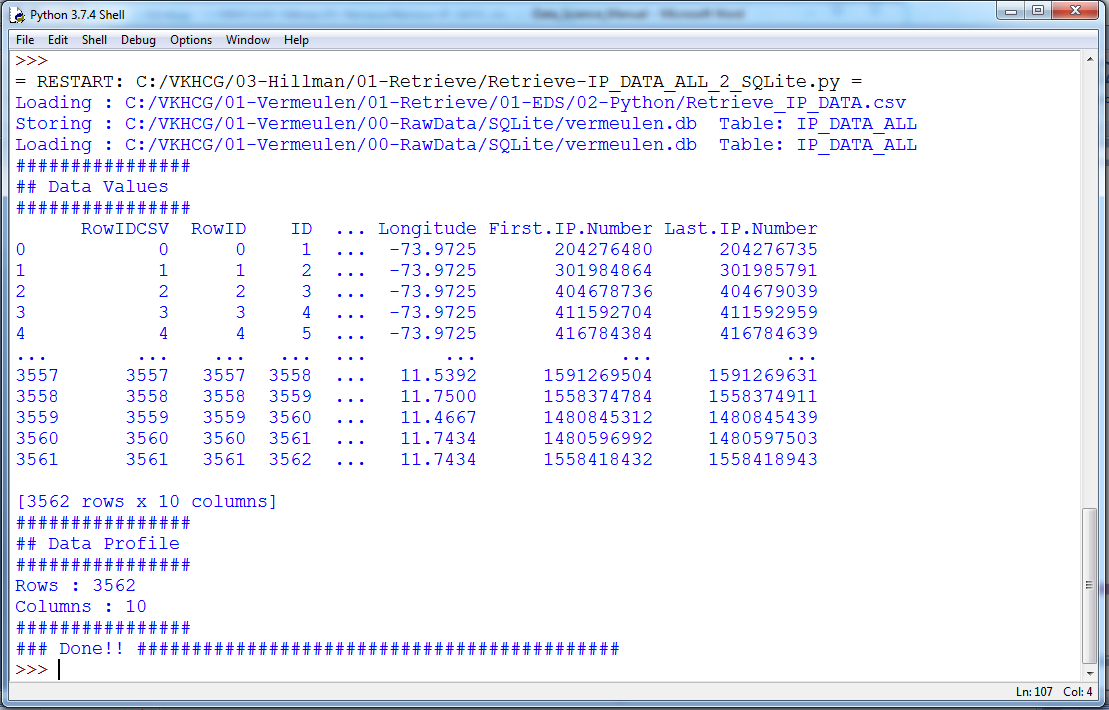
Select the data range from the excel worksheet.

4



#### Output:

5



## Import data from different data sources (from Excel, csv, mysql, sql server, oracle to R/Python/Excel)

SQLite:

######################################### #######################

# -\*- coding: utf-8 -\*- ################################################################

import sqlite3 as sq import pandas as pd

################################################################ Base='C:/VKHCG'

sDatabaseName=Base + '/01-Vermeulen/00-RawData/SQLite/vermeulen.db' conn = sq.connect(sDatabaseName) ################################################################

sFileName='C:/VKHCG/01-Vermeulen/01-Retrieve/01-EDS/02-Python/Retrieve\_IP\_DATA.csv' print('Loading :',sFileName) IP\_DATA\_ALL\_FIX=pd.read\_csv(sFileName,header=0,low\_memory=False) IP\_DATA\_ALL\_FIX.index.names = ['RowIDCSV']

sTable='IP\_DATA\_ALL'

print('Storing :',sDatabaseName,' Table:',sTable) IP\_DATA\_ALL\_FIX.to\_sql(sTable, conn, if\_exists="replace") print('Loading :',sDatabaseName,' Table:',sTable) TestData=pd.read\_sql\_query("select \* from IP\_DATA\_ALL;", conn) print('################')

print('## Data Values') print('################')

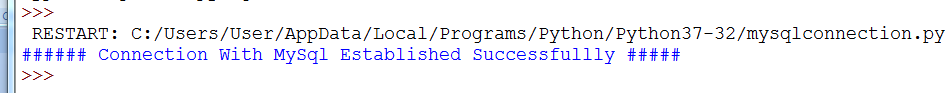
print(TestData) print('################')

print('## Data Profile') print('################')

print('Rows :',TestData.shape[0]) print('Columns :',TestData.shape[1]) print('################')

print('### Done!! ############################################')

6



**MySQL:**

Open MySql

Create a database “DataScience”

Create a python file and add the following code:

################ Connection With MySQL ######################

importmysql.connector

conn = mysql.connector.connect(host='localhost', database='DataScience',

user='root', password='root') conn.connect if(conn.is\_connected):

print('###### Connection With MySql Established Successfullly ##### ') else:

print('Not Connected -- Check Connection Properites')

Microsoft Excel

##################Retrieve-Country-Currency.py ################################################################

# -\*- coding: utf-8 -\*- ################################################################

importos

import pandas as pd ################################################################ Base='C:/VKHCG' ################################################################

sFileDir=Base + '/01-Vermeulen/01-Retrieve/01-EDS/02-Python' #if not os.path.exists(sFileDir):

#os.makedirs(sFileDir) ################################################################

CurrencyRawData = pd.read\_excel('C:/VKHCG/01-Vermeulen/00-RawData/Country\_Currency.xlsx') sColumns = ['Country or territory', 'Currency', 'ISO-4217']

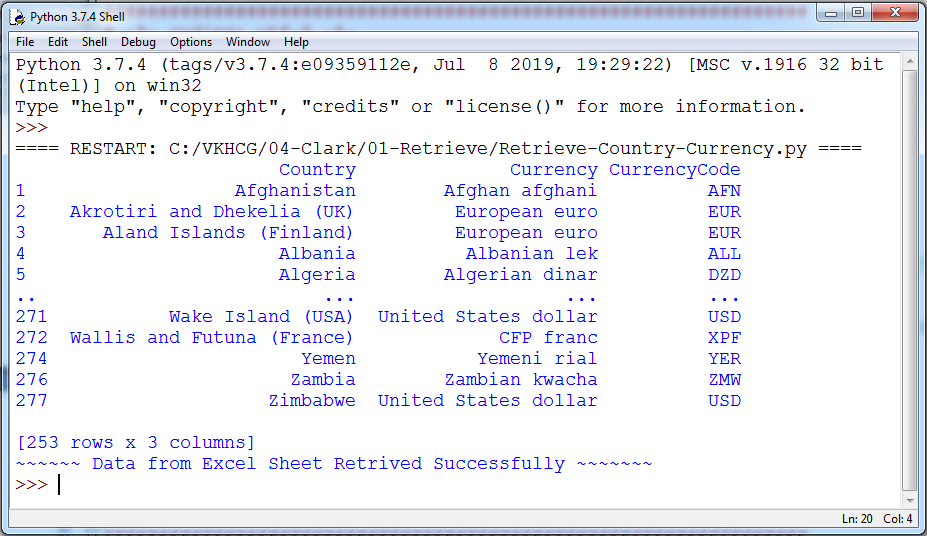
CurrencyData = CurrencyRawData[sColumns] CurrencyData.rename(columns={'Country or territory': 'Country', 'ISO-4217': 'CurrencyCode'}, inplace=True) CurrencyData.dropna(subset=['Currency'],inplace=True) CurrencyData['Country'] = CurrencyData['Country'].map(lambda x: x.strip()) CurrencyData['Currency'] = CurrencyData['Currency'].map(lambda x: x.strip())

CurrencyData['CurrencyCode'] = CurrencyData['CurrencyCode'].map(lambda x: x.strip())

print(CurrencyData)

print('~~~~~~ Data from Excel Sheet Retrived Successfully ~~~~~~~ ') ################################################################

7



sFileName=sFileDir + '/Retrieve-Country-Currency.csv' CurrencyData.to\_csv(sFileName, index = False) ################################################################

**OUTPUT:**

8

# Practical 2:

## Design a survey form for a given case study, collect the primary data and analyse it

#### Case 1:

A researcher wants to conduct a Survey in colleges on Use of ICT in higher education from Mumbai, Thane and Navi Mumbai. The survey focuses on access to and use of ICT in teaching and learning, as well as on attitudes towards the use of ICT in teaching and learning.

Design questionnaire addressed to teachers seeks information about the target class, his experience using ICT for teaching, access to ICT infrastructure, support available, ICT based activities and material used, obstacles to the use of ICT in teaching, learning activities with the target class, your skills and attitudes to ICT, and some personal background information.

Arrange question in following groups:

* 1. Information about the target class you teach
  2. Experience with ICT for teaching
  3. ICT access for teaching
  4. Support to teachers for ICT use
  5. ICT based activities and material used for teaching
  6. Obstacles to using ICT in teaching and learning
  7. Learning activities with the target class
  8. Teacher skills
  9. Teacher opinions and attitudes
  10. Personal background information

#### Case 2:

A research agency wants to study the perception about App based taxi service in Mumbai, Thane and Navi Mumbai. The survey focuses on customers attitude towards app base taxi service as well as on attitudes towards regular taxi cab.

Design questionnaire seeks information about the target taxi service, his experience using taxi services, access, support available, obstacles and some personal background information, with the following objectives:

1. To find out the customer satisfaction towards the App based-taxi services.
2. To find the level of convenience and comfort with App based -taxi services.
3. To know their opinion about the tariff system and promptness of service.

9

1. To ascertain the customer view towards the driver behaviour and courtesy.
2. To provide inputs to enhance the services to delight the customers.
3. To examine relationship between service quality factors and taxi passenger satisfaction.
4. To suggest better regulations for transportation authorities regarding customer protection and effective monitoring of taxi services.

## Case 3:

A popular electronic store want to conduct a survey to develop awareness of branded laptop baseline estimates and determine popularity of different company’s laptop. It suggests steps to be initiated or strengthened in the field of demand in a region. The key indicators are among the general population, demand branded laptop and the problem users.

The objectives of this particular study are:-

1. To know the preferences of different types of branded laptops by students and professionals.
2. To study which factor influence for choosing different types of branded laptops.
3. To know about the level of satisfaction towards different types of branded laptops.
4. To identify the perception of consumers towards the laptop positioning strategy.
5. To know the consumer preference towards laptop in the present era.

## Use the collected data for analysis.

10



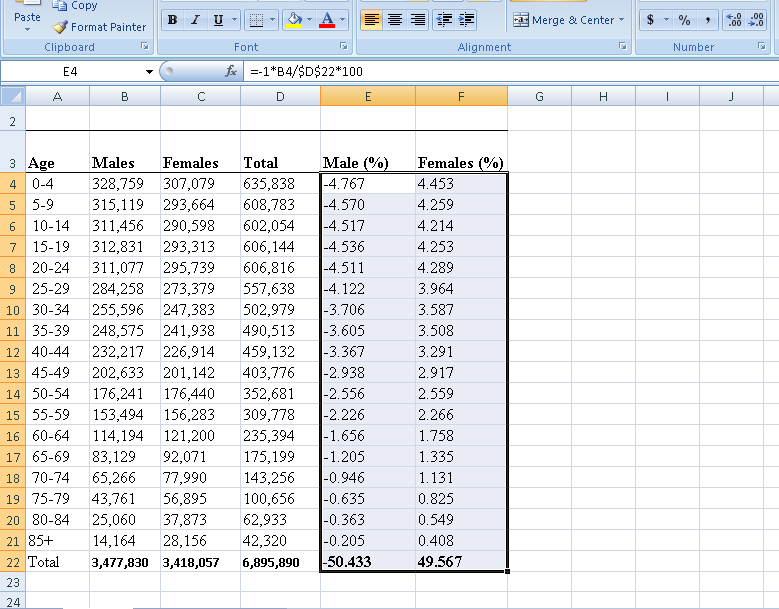
## Perform analysis of given secondary data.

### Steps in Secondary Data Analysis

* 1. **Determine your research question** – Knowing exactly what you are looking for.
  2. **Locating data**– Knowing what is out there and whether you can gain access to it. A quick Internet search, possibly with the help of a librarian, will reveal a wealth of options.
  3. **Evaluating relevance of the data** – Considering things like the data’s original purpose, when it was collected, population, sampling strategy/sample, data collection protocols, operationalization of concepts, questions asked, and form/shape of the data.
  4. **Assessing credibility of the data** – Establishing the credentials of the original researchers, searching for full explication of methods including any problems encountered, determining how consistent the data is with data from other sources, and discovering whether the data has been used in any credible published research.
  5. **Analysis –** This will generally involve a range of statistical processes.

**Example:** Analyze the given Population Census Data for Planning and Decision Making by using the size and composition of populations.

11



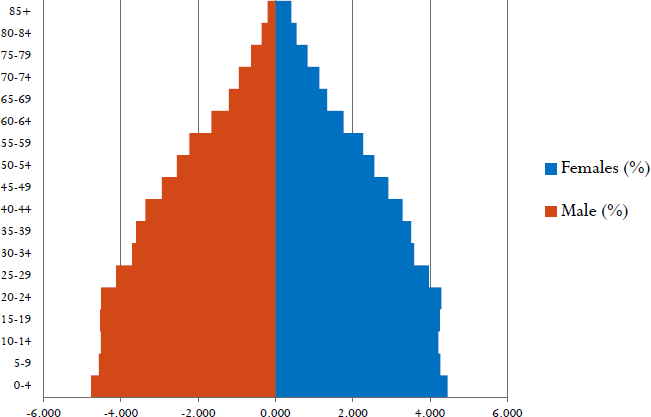
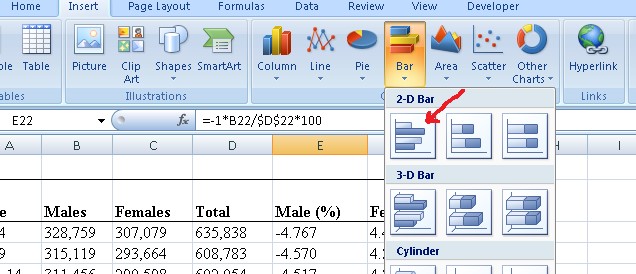
Put the cursor in cell **B22** and click on the **AutoSum** and then click **Enter**. This will calculate the total population. Then copy the formula in cell **D22** across the row **22.**

To calculate the percent of males in cell **E4**, enter the formula **=-1\*100\*B4/$D$22** . And copy the formula in cell **E4** down to cell **E21.**

To calculate the percent of females in cell **F4**, enter the formula =**100\*C4/$D$22**. Copy the formula in cell **F4** down to cell **F21.**

To build the population pyramid, we need to choose a horizontal bar chart with two series of data (% male and % female) and the age labels in column A as the **Category X-axis** labels. Highlight the range **A3:A21**, hold down the CTRL key and highlight the range **E3:F21**

12



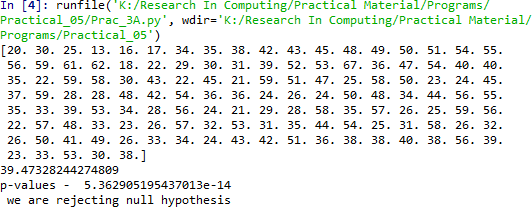
Under **inset** tab, under horizontal bar charts select **clustered bar chart**

Put the tip of your mouse arrow on the **Y-axis** (vertical axis) so it says “Category Axis”, right click and chose **Format Axis**

Choose **Axis options** tab and set the major and minor tick mark type to **None**, Axis labels to **Low**, and click **OK**.

Click on any of the bars in your pyramid, click right and select “format data series”. Set the **Overlap** to **100** and **Gap Width** to **0**. Click **OK**.

13



# Practical 3:

## Perform testing of hypothesis using one sample t-test.

**One sample t-test** : The One Sample *t* Test determines whether the sample mean is statistically different from a known or hypothesised population mean. The One Sample *t* Test is a parametric test.

#### Program Code:

# -\*- coding: utf-8 -\*- """

Created on Mon Dec 16 18:01:46 2019 @author: Ahtesham Shaikh

"""

fromscipy.stats import ttest\_1samp importnumpy as np

ages = np.genfromtxt('ages.csv') print(ages)

ages\_mean = np.mean(ages) print(ages\_mean)

tset, pval = ttest\_1samp(ages, 30) print('p-values - ',pval)

if pval< 0.05: # alpha value is 0.05

print(" we are rejecting null hypothesis")

else:

print("we are accepting null hypothesis")

#### Output:

14

## Write a program for t-test comparing two means for independent samples.

The *t* distribution provides a good way to perform one sample tests on the mean when the population variance is not known provided the population is normal or the sample is sufficiently large so that the Central Limit Theorem applies.

### Two Sample t Test

Example: A college Princiapal informed classroom teachers that some of their students showedunusual potential for intellectual gains. One months later the students identified to teachers ashaving potentional for unusual intellectual gains showed significiantly greater gains performanceon a test said to measure IQ than did students who were not so identified. Below are the data forthe students:

|  |  |  |
| --- | --- | --- |
| **Experimental** | **Comparison** |  |
| 35 | 2 |
| 40 | 27 |
| 12 | 38 |
| 15 | 31 |
| 21 | 1 |
| 14 | 19 |
| 46 | 1 |
| 10 | 34 |
| 28 | 3 |
| 48 | 1 |
| 16 | 2 |
| 30 | 3 |
| 32 | 2 |
| 48 | 1 |
| 31 | 2 |
| 22 | 1 |
| 12 | 3 |
| 39 | 29 |
| 19 | 37 |
| 25 | 2 |
| 27.15 | 11.95 | Mean |
| 12.51 | 14.61 | Sd |

Experimental Data

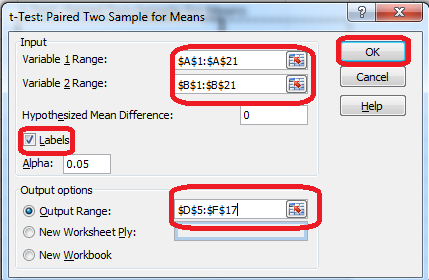
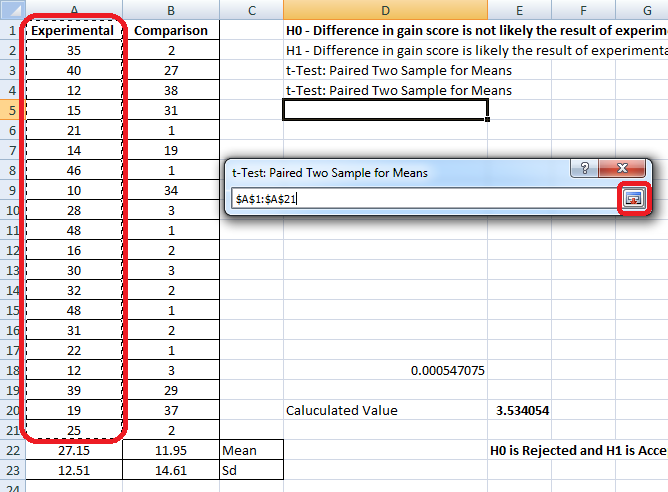
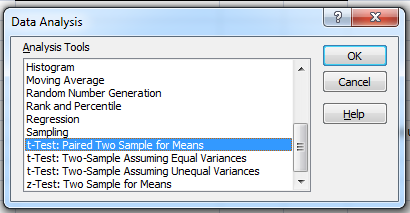
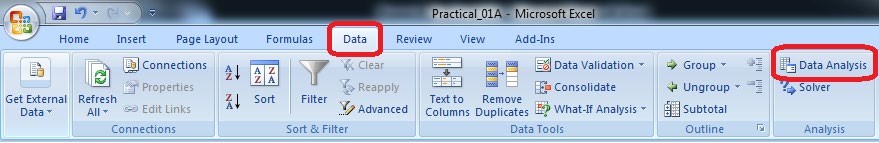
To calculate Standard Mean go to cell A22 and type =SUM(A2:A21)/20

To calculate Standard Deviation go to cell A23 and type =STDEV(A2:A21)

Comparison Data

To calculate Standard Mean go to cell B22 and type =SUM(B2:B21)/20

15



To calculate Standard Deviation go to cell B23 and type =STDEV(B2:B21) To find T-Test Statistics go to data Data Analysis

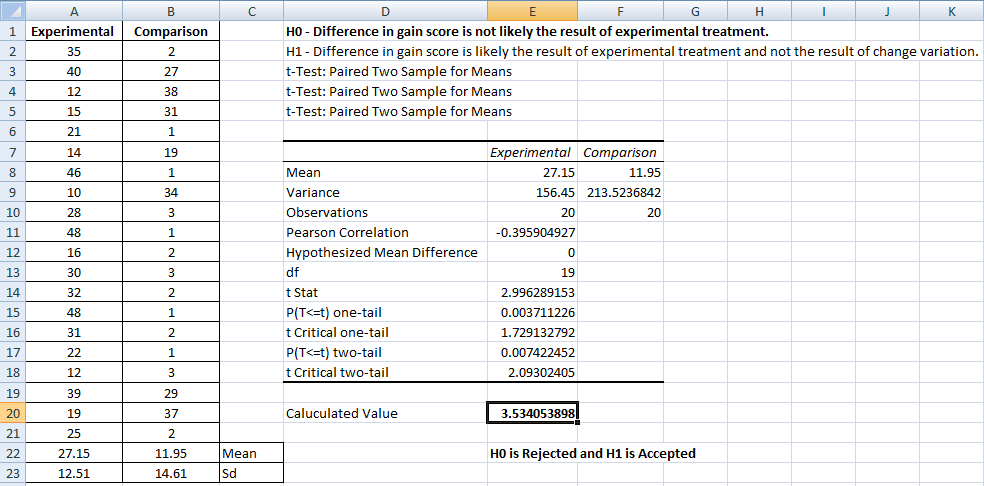
To caluculate the T-Test square value go to cell E20 and type

=(A22-B22)/SQRT((A23\*A23)/COUNT(A2:A21)+(B23\*B23)/COUNT(A2:A21))

Now go to cell E20 and type

=IF(E20<E12,"H0 is Accepted", "H0 is Rejected and H1 is Accepted")

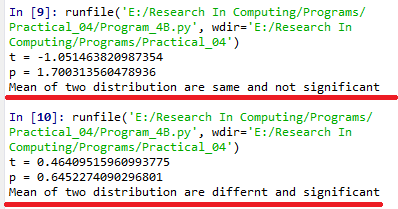
Our calculated value is larger than the tabled value at alpha = .01, so we reject the null hypothesisand accept the alternative hypothesis, namely, that the difference in gain scores is likely the resultof the experimental treatment and not the result of chance variation.



#### Output:

16

17



#### Using Python

importnumpy as np fromscipy import stats

fromnumpy.random import randn N = 20

#a = [35,40,12,15,21,14,46,10,28,48,16,30, 32,48,31,22,12,39,19,25]

#b = [2,27,31,38,1,19,1,34,3,1,2,1,3,1,2,1,3,29,37,2]

a = 5 \* randn(100) + 50 b = 5 \* randn(100) + 51 var\_a = a.var(ddof=1) var\_b = b.var(ddof=1)

s = np.sqrt((var\_a + var\_b)/2)

t = (a.mean() - b.mean())/(s\*np.sqrt(2/N))

df = 2\*N - 2

#p-value after comparison with the t p = 1 - stats.t.cdf(t,df=df)

print("t = " + str(t)) print("p = " + str(2\*p)) if t> p :

print('Mean of two distribution are differnt and significant') else:

print('Mean of two distribution are same and not significant')

#### Output:

18

#### Practice Questions:

**Example 1:** we have to test whether the height of men in the population is different from height of women in general. So we take a sample from the population and use the t-test to see if the result is significant.

H0 – Height of men and women are same

H1 – Height of men and women are the different

|  |  |
| --- | --- |
| **Men** | **Women** |
| 181 | 160 |
| 169 | 150 |
| 160 | 160 |
| 170 | 175 |
| 175 | 160 |
| 158 | 170 |
| 152 | 160 |
| 172 | 150 |
| 160 | 155 |
| 175 | 162 |
| 180 | 165 |
| 170 | 148 |
| 165 | 159 |
| 180 | 163 |
| 155 | 170 |
| 159 | 178 |
| 163 | 180 |
| 171 | 156 |
| 182 | 164 |
| 150 | 167 |

**Example 2:** Design a survey form to get grade of students who have passed B. Sc. IT and B. Sc. CS from the same University. Perform T-Test to test the given hypothsis: H0 – Scores of students in two courses are same.

H1 – Scores of students are the different.

**Example 2:** Collect a sample data know that use of Online Food Ordering app to compare whether the usage is equal or different.

19

## Perform testing of hypothesis using paired t-test.

The paired sample t-test is also called dependent sample t-test. It’s an univariate test that tests for a significant difference between 2 related variables. An example of this is if you where to collect the blood pressure for an individual before and after some treatment, condition, or time point. The data set contains blood pressure readings before and after an intervention. These are variables “bp\_before” and “bp\_after”.

The hypothesis being test is:

* + **H0** - The mean difference between sample 1 and sample 2 is equal to 0.
  + **H0** - The mean difference between sample 1 and sample 2 is not equal to 0

#### Program Code:

# -\*- coding: utf-8 -\*- """

Created on Mon Dec 16 19:49:23 2019

@author: MyHome """

from scipy import stats

import matplotlib.pyplot as plt import pandas as pd

df = pd.read\_csv("blood\_pressure.csv")

print(df[['bp\_before','bp\_after']].describe())

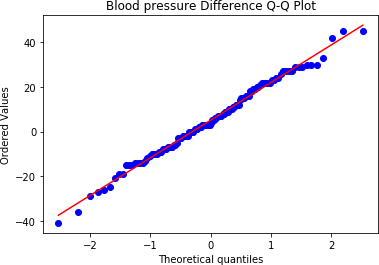
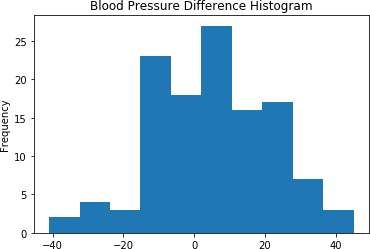
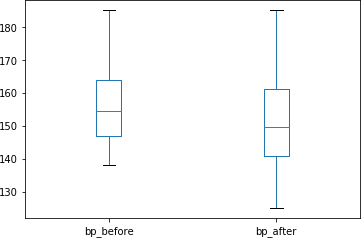
#First let’s check for any significant outliers in #each of the variables.

df[['bp\_before', 'bp\_after']].plot(kind='box') # This saves the plot as a png file plt.savefig('boxplot\_outliers.png')

# make a histogram to differences between the two scores. df['bp\_difference'] = df['bp\_before'] - df['bp\_after']

df['bp\_difference'].plot(kind='hist', title= 'Blood Pressure Difference Histogram') #Again, this saves the plot as a png file

20

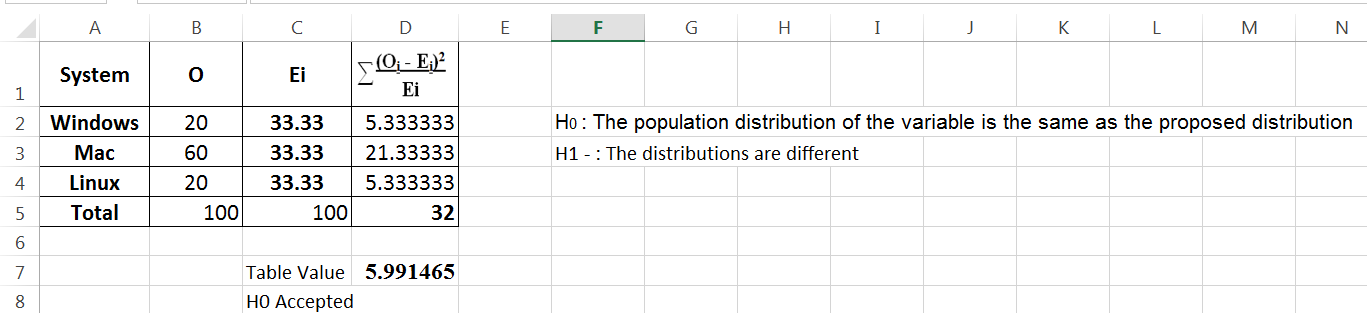


plt.savefig('blood pressure difference histogram.png') stats.probplot(df['bp\_difference'], plot= plt) plt.title('Blood pressure Difference Q-Q Plot') plt.savefig('blood pressure difference qq plot.png') stats.shapiro(df['bp\_difference']) stats.ttest\_rel(df['bp\_before'], df['bp\_after'])

#### Output:

A paired sample t-test was used to analyze the blood pressure before and after the intervention to test if the intervention had a significant affect on the blood pressure. The blood pressure before the intervention was higher (156.45 ± 11.39 units) compared to the blood pressure post intervention (151.36 ± 14.18 units); there was a statistically significant decrease in blood pressure (t(119)=3.34, p= 0.0011) of 5.09 units.

21



# Practical 4:

## Perform testing of hypothesis using chi-squared goodness- of-fit test.

#### Problem

Ansystem administrator needs to upgrade the computers for his division. He wants to know what sort of computer system his workers prefer. He gives three choices: Windows, Mac, or Linux. Test the hypothesis or theory that an equal percentage of the population prefers each type of computer system .

|  |  |  |  |
| --- | --- | --- | --- |
| **System** | **O** | **Ei** |  |
| **Windows** | 20 | **33.33%** |  |
| **Mac** | 60 | **33.33%** |  |
| **Linux** | 20 | **33.33%** |  |

H0 : The population distribution of the variable is the same as the proposed distribution HA : The distributions are different

To calculate the Chi –Squred value for Windows go to cell D2 and type =((B2- C2)\*(B2-C2))/C2

To calculate the Chi –Squred value for Mac go to cell D3 and type =((B3-C3)\*(B3- C3))/C3

To calculate the Chi –Squred value for Mac go to cell D3 and type =((B4-C4)\*(B4- C4))/C4

Go to Cell D5 for and type=SUM(D2:D4)

To get the table value for Chi-Square for α = 0.05 and dof = 2, go to cell D7 and type

=CHIINV(0.05,2)

At cell D8 type =IF(D5>D7, "H0 Accepted","H0 Rejected")

#### Output:

22

#### Practice Questions:

* 1. The Mobile Association of Mumbai conducted a survey in 2019 and determined that 60% of users have only one SIM card, 28% have two SIM cards and 12% have three or more. Supposing that you have decided to conduct yourown survey and have collected the data that out of 129 Mobile Users, 73 had one SIM and 38 had two SIM, determine whether your data supports the results of the association’s study.**(Use a significance level of 0.05.)**
  2. In a debate, Geeta told Pankaj that the reason her car insurance is less expensive is that metro city drivers get in more accidents thanRuralarea drivers. According to her study,in metro cities drivers are held responsible in 65% of accidents. If Pankaj does some research of his own and discovers that 46 out of the 85 accidents he investigates involve rural areadrivers, does his data support or refute Geeta’s hypothesis?

Ho –Metro citiesdrivers are more responsible for accidents than rural area drivers are.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **O** | **Ei** |  |
| **Rural Drivers** | 46 | **65%** |  |
| **Metro Drivers** | 39 | **35%** |  |
| **Total** | 85 | **100** |  |

23

## Perform testing of hypothesis using chi-squared test of independence.

In a study to understatnd the permormacne of M. Sc. IT Part -1 class, a college selects a random sample of 100 students. Each student was asked his grade obtained in B. Sc. IT. The sample is as given below

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.**  **No** | **Roll No** | **Student's Name** | **Gen** | **Grade** |  | **Sr. No** | **Roll No** | **Student's Name** | **Gen** | **Grade** |
| **1** | 1 | Gaborone | m | O | **62** | 3 | Maun | f | O |
| **2** | 2 | Francistown | m | O | **63** | 7 | Tete | f | O |
| **3** | 5 | Niamey | m | O | **64** | 9 | Chimoio | f | O |
| **4** | 13 | Maxixe | m | O | **65** | 11 | Pemba | f | O |
| **5** | 16 | Tema | m | O | **66** | 14 | Chibuto | f | O |
| **6** | 17 | Kumasi | m | O | **67** | 25 | Mampong | f | O |
| **7** | 34 | Blida | m | O | **68** | 36 | Tlemcen | f | O |
| **8** | 35 | Oran | m | O | **69** | 40 | Adrar | f | O |
| **9** | 38 | Saefda | m | O | **70** | 41 | Tindouf | f | O |
| **10** | 42 | Constantine | m | O | **71** | 46 | Skikda | f | O |
| **11** | 43 | Annaba | m | O | **72** | 47 | Ouargla | f | O |
| **12** | 45 | Bejaefa | m | O | **73** | 10 | Matola | f | D |
| **13** | 48 | Medea | m | O | **74** | 20 | Legon | f | D |
| **14** | 49 | Djelfa | m | O | **75** | 21 | Sunyani | f | D |
| **15** | 50 | Tipaza | m | O | **76** | 72 | Teenas | f | D |
| **16** | 51 | Bechar | m | O | **77** | 73 | Kouba | f | D |
| **17** | 54 | Mostaganem | m | O | **78** | 75 | HussenDey | f | D |
| **18** | 55 | Tiaret | m | O | **79** | 77 | Khenchela | f | D |
| **19** | 56 | Bouira | m | O | **80** | 82 | HassiBahbah | f | D |
| **20** | 59 | Tebessa | m | O | **81** | 84 | Baraki | f | D |
| **21** | 61 | El Harrach | m | O | **82** | 91 | Boudouaou | f | D |
| **22** | 62 | Mila | m | O | **83** | 95 | Tadjenanet | f | D |
| **23** | 65 | Fouka | m | O | **84** | 4 | Molepolole | f | C |
| **24** | 66 | El Eulma | m | O | **85** | 8 | Quelimane | f | C |
| **25** | 68 | SidiBel Abbes | m | O | **86** | 23 | Bolgatanga | f | C |
| **26** | 69 | Jijel | m | O | **87** | 58 | Mohammadia | f | C |
| **27** | 70 | Guelma | m | O | **88** | 83 | Merouana | f | C |
| **28** | 85 | Khemis El Khechna | m | O | **89** | 24 | Ashaiman | f | B |
| **29** | 87 | Bordj El Kiffan | m | O | **90** | 76 | N'gaous | f | B |
| **30** | 88 | Lakhdaria | m | O | **91** | 90 | Bab El Oued | f | B |
| **31** | 6 | Maputo | m | D | **92** | 92 | BordjMenael | f | B |
| **32** | 12 | Lichinga | m | D | **93** | 93 | Ksar El Boukhari | f | B |
| **33** | 15 | Ressano Garcia | m | D | **94** | 74 | Reghaa | f | A |
| **34** | 19 | Accra | m | D | **95** | 78 | Cheria | f | A |
| **35** | 27 | Wa | m | D | **96** | 79 | Mouzaa | f | A |
| **36** | 28 | Navrongo | m | D | **97** | 80 | Meskiana | f | A |
| **37** | 37 | Mascara | m | D | **98** | 81 | Miliana | f | A |
| **38** | 44 | Batna | m | D | **99** | 94 | Sig | f | A |
| **39** | 57 | El Biar | m | D | **100** | 99 | Kadiria | f | A |
| **40** | 60 | Boufarik | m | D |  | | | | | |
| **41** | 63 | OuedRhiou | m | D |
| **42** | 64 | Souk Ahras | m | D |
| **43** | 71 | Dar El Befda | m | D |
| **44** | 86 | Birtouta | m | D |
| **45** | 18 | Takoradi | m | C |
| **46** | 22 | Cape Coast | m | C |
| **47** | 29 | Kwabeng | m | C |
| **48** | 30 | Algiers | m | C |
| **49** | 31 | Laghouat | m | C |
| **50** | 39 | Relizane | m | C |
| **51** | 52 | Setif | m | C |
| **52** | 53 | Biskra | m | C |
| **53** | 67 | Kolea | m | C |
| **54** | 100 | AefnFakroun | m | C |
| **55** | 26 | Nima | m | B |
| **56** | 32 | TiziOuzou | m | B |
| **57** | 33 | Chlef | m | B |

24

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **58** | 89 | M'sila | m | A |
| **59** | 96 | Heliopolis | m | A |
| **60** | 97 | Berrouaghia | m | A |
| **61** | 98 | Sougueur | m | A |

**Null Hypothesis - H0 :** The performance of girls students is same as boys students. **Alternate Hypothesis - H1 :** The performance of boys and girls students are different. Open Excel Workbook

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **O** | **A** | **B** | **C** | **D** | **Total** |  |
| **Girls** | 11 | 7 | 5 | 5 | 11 | **39** | 6.075 |
| **Boys** | 30 | 4 | 3 | 10 | 14 | **61** | 6.075 |
| **Total** | 41 | 11 | 8 | 15 | 25 | **100** | **12.150** |
| **Ei** | **20.5** | **5.5** | **4** | **7.5** | **12.5** | **50** |  |

Prepare a contingency table as shown above. To calculate Girls Students with ‘O’ Grade

Go to Cell N6 and type =COUNTIF($J$2:$K$40,"O")

To calculate Girls Students with ‘A’ Grade

Go to Cell O6 and type =COUNTIF($J$2:$K$40,"A")

To calculate Girls Students with ‘B’ Grade

Go to Cell P6 and type =COUNTIF($J$2:$K$40,"B")

To calculate Girls Students with ‘C’ Grade

Go to Cell Q6 and type =COUNTIF($J$2:$K$40,"C")

To calculate Girls Students with ‘D’ Grade

Go to Cell R6 and type =COUNTIF($J$2:$K$40,"D")

To calculate Boys Students with ‘O’ Grade

Go to Cell N7 and type =COUNTIF($D$2:$E$62,"O")

To calculate Boys Students with ‘A’ Grade

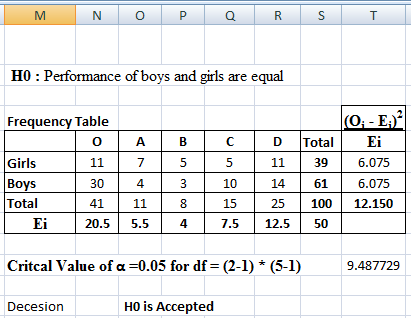
Go to Cell O7 and type =COUNTIF($D$2:$E$62,"A")

To calculate Boys Students with ‘B’ Grade

Go to Cell P7 and type =COUNTIF($D$2:$E$62,"B") To calculate Boys Students with ‘C’ Grade

Go to Cell Q7 and type =COUNTIF($D$2:$E$62,"C")

25



To calculate Boys Students with ‘D’ Grade

Go to Cell R7 and type =COUNTIF($D$2:$E$62,"D")

#### To calculated the expected value Ei

Go to Cell N9 and type =N8/2 Go to Cell O9 and type =O8/2 Go to Cell P9 and type =P8/2 Go to Cell Q9 and type =Q8/2 Go to Cell R9 and type =R8/2

Go to Cell S6 and calculate total girl students = SUM(N6:R6) Go to Cell S7 and calculate total girl students = SUM(N7:R7)

#### Now Calculate

Go to cell **T6** and type

=SUM((N6-$N$9)^2/$N$9,(O6-$O$9)^2/$O$9,(P6-$P$9)^2/$P$9,(Q6-Q$9)^2/$Q$9, (R6-$R$9)^2/$R$9)

Go to cell **T7** and type

=SUM((N7-$N$9)^2/$N$9,(O7-$O$9)^2/$O$9,(P7-$P$9)^2/$P$9,(Q7-Q$9)^2/$Q$9, (R7-$R$9)^2/$R$9)

To get the table value go to cell T11 and type **=CHIINV(0.05,4)**

Go to cell O13 and type =IF(T8>=T11," H0 is Accepted", "H0 is Rejected")

26

#### Using Python

importnumpy as np import pandas as pd importscipy.stats as stats

np.random.seed(10)

stud\_grade = np.random.choice(a=["O","A","B","C","D"],

p=[0.20, 0.20 ,0.20, 0.20, 0.20], size=100)

stud\_gen = np.random.choice(a=["Male","Female"], p=[0.5, 0.5], size=100) mscpart1 = pd.DataFrame({"Grades":stud\_grade, "Gender":stud\_gen}) print(mscpart1)

stud\_tab = pd.crosstab(mscpart1.Grades, mscpart1.Gender, margins=True) stud\_tab.columns = ["Male", "Female", "row\_totals"]

stud\_tab.index = ["O", "A", "B", "C", "D", "col\_totals"] observed = stud\_tab.iloc[0:5, 0:2 ]

print(observed)

expected = np.outer(stud\_tab["row\_totals"][0:5], stud\_tab.loc["col\_totals"][0:2]) / 100 print(expected)

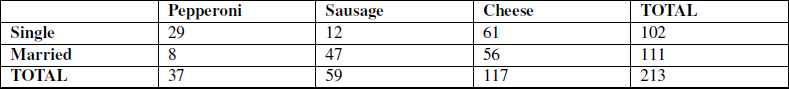
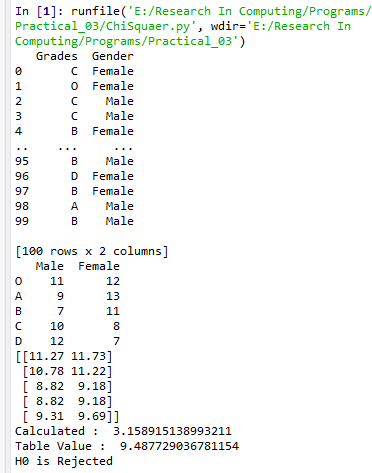
chi\_squared\_stat = (((observed-expected)\*\*2)/expected).sum().sum() print('Calculated : ',chi\_squared\_stat)

crit = stats.chi2.ppf(q=0.95, df=4) print('Table Value : ',crit)

ifchi\_squared\_stat>= crit: print('H0 is Accepted ') else:

print('H0 is Rejected ')

27



#### Output :

**Practice Questions**

* 1. Anita claims that girls take more normal and filter appliedselfies than boys, but Karan does not agree with her, so he conducts a survey collects the following data, would it be correct to say that he should reject Anita’s claim that gender affects tendency to take selfies?

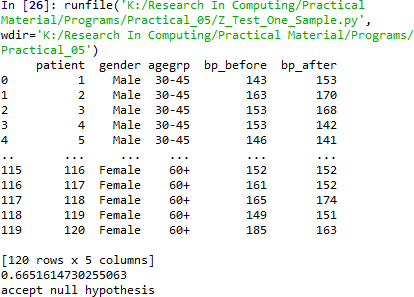
H0 - Gender affects tendency to take more photographs

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Normal Selfie** | **Apply Filter** | **Total** |
| **Female** | 72 | 489 | 561 |
| **Male** | 48 | 530 | 578 |
| **TOTAL** | 120 | 1019 | 1139 |

* 1. Ketan claims that single people prefer different pizzas than married people do. Kato’s brother Anand doesn’t think that is true, so he conducts some research of his own, and collects the data below.

H0: Marital status and pizza type are not associated. H1: Marital type and pizza type are associated.

28



# Practical 5:

## Perform testing of hypothesis using Z-test.

Use a Z test if:

* Your sample size is greater than 30. Otherwise, use a t test.
* Data points should be independent from each other. In other words, one data point isn’t related or doesn’t affect another data point.
* Your data should be normally distributed. However, for large sample sizes (over 30) this doesn’t always matter.
* Your data should be randomly selected from a population, where each item has an equal chance of being selected.
* Sample sizes should be equal if at all possible.

**Ho -** Blood pressure has a mean of 156 units

### Program Code for one-sample Z test.

from statsmodels.stats import weightstats as stests import pandas as pd

from scipy import stats

df = pd.read\_csv("blood\_pressure.csv") df[['bp\_before','bp\_after']].describe() print(df)

ztest ,pval = stests.ztest(df['bp\_before'], x2=None, value=156) print(float(pval))

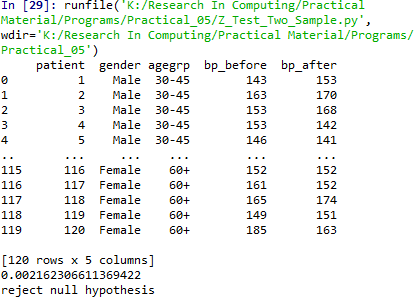
if pval<0.05:

print("reject null hypothesis") else:

print("accept null hypothesis")

#### Output:

29



**Two-sample Z test-** In two sample z-test , similar to t-test here we are checking two independent data groups and deciding whether sample mean of two group is equal or not.

#### H0 : mean of two group is 0

**H1 : mean of two group is not 0**

# -\*- coding: utf-8 -\*- """

Created on Mon Dec 16 20:42:17 2019 @author: MyHome

"""

import pandas as pd

from statsmodels.stats import weightstats as stests df = pd.read\_csv("blood\_pressure.csv") df[['bp\_before','bp\_after']].describe()

print(df)

ztest ,pval = stests.ztest(df['bp\_before'], x2=df['bp\_after'], value=0,alternative='two- sided')

print(float(pval))

if pval<0.05:

print("reject null hypothesis") else:

print("accept null hypothesis")

30

# Practical 6:

## Perform testing of hypothesis using One-way ANOVA.

#### ANOVA Assumptions

* + The dependent variable (SAT scores in our example) should be continuous.
  + The independent variables (districts in our example) should be two or more categorical groups.
  + There must be different participants in each group with no participant being in more than one group. In our case, each school cannot be in more than one district.
  + The dependent variable should be approximately normally distributed for each category.
  + Variances of each group are approximately equal.

From our data exploration, we can see that the average SAT scores are quite different for each district. Since we have five different groups, we cannot use the t-test, use the 1-way ANOVA test anyway just to understand the concepts.

#### H0 - There are no significant differences between the groups' mean SAT scores.

**µ1 = µ2 = µ3 = µ4 = µ5**

#### H1 - There is a significant difference between the groups' mean SAT scores.

If there is at least one group with a significant difference with another group, the null hypothesis will be rejected.

import pandas as pd importnumpy as np importmatplotlib.pyplot as plt importseaborn as sns fromscipy import stats

data = pd.read\_csv("scores.csv") data.head() data['Borough'].value\_counts()

#### ############### There is no total score column, have to create it.

**########In addition, find the mean score of the each district across all schools.**

data['total\_score'] = data['Average Score (SAT Reading)'] + \ data['Average Score (SAT Math)'] + \

data['Average Score (SAT Writing)']

31

data = data[['Borough', 'total\_score']].dropna()

x = ['Brooklyn', 'Bronx', 'Manhattan', 'Queens', 'Staten Island'] district\_dict = {}

#Assigns each test score series to a dictionary key for district in x:

district\_dict[district] = data[data['Borough'] == district]['total\_score']

y = []

yerror = []

#Assigns the mean score and 95% confidence limit to each district for district in x:

y.append(district\_dict[district].mean()) yerror.append(1.96\*district\_dict[district].std()/np.sqrt(district\_dict[district].shape[0])) print(district + '\_std : {}'.format(district\_dict[district].std()))

sns.set(font\_scale=1.8)

fig = plt.figure(figsize=(10,5))

ax = sns.barplot(x, y, yerr=yerror) ax.set\_ylabel('Average Total SAT Score') plt.show()

#### ###################### Perform 1-way ANOVA

print(stats.f\_oneway(

district\_dict['Brooklyn'], district\_dict['Bronx'], \ district\_dict['Manhattan'], district\_dict['Queens'], \ district\_dict['Staten Island']

))

districts = ['Brooklyn', 'Bronx', 'Manhattan', 'Queens', 'Staten Island'] ss\_b = 0

for d in districts:

ss\_b += district\_dict[d].shape[0] \* \ np.sum((district\_dict[d].mean() - data['total\_score'].mean())\*\*2)

ss\_w = 0

for d in districts:

ss\_w += np.sum((district\_dict[d] - district\_dict[d].mean())\*\*2)

32



msb = ss\_b/4

msw = ss\_w/(len(data)-5) f=msb/msw

print('F\_statistic: {}'.format(f))

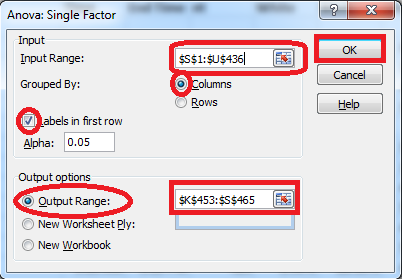
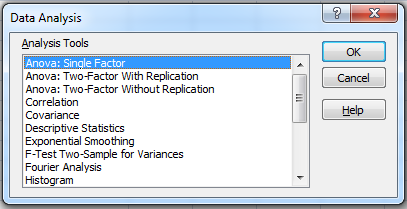
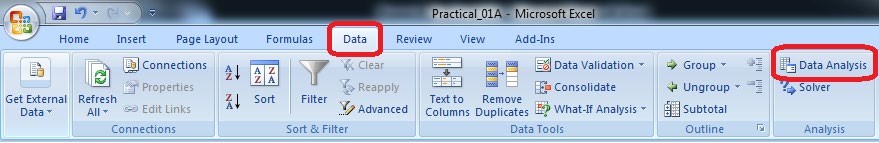
ss\_t = np.sum((data['total\_score']-data['total\_score'].mean())\*\*2) eta\_squared = ss\_b/ss\_t

print('eta\_squared: {}'.format(eta\_squared))

#### Output:

Since theresulting pvalueis less than 0.05. The null hypothesis is rejected and conclude that there is a significant difference between the SAT scores for each district.

33



# Using Excel

#### H0 - There are no significant differences between the Subject’s mean SAT scores. µ1 = µ2 = µ3 = µ4 = µ5

**H1 - There is a significant difference between the Subject's mean SAT scores.**

To perform ANOVA go to data Data Analysis

**Input Range** : $S$1:$U$436*( Select columns to be analyzed in group)*

**Output Range** :$K$453:$S$465*( Can be any Range)*

34

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Anova: Single Factor |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| SUMMARY |  |  |  |  |  |  |
| *Groups* | *Count* | *Sum* | *Average* | *Variance* |  |  |
| Average Score (SAT Math) | 375 | 162354 | 432.944 | 5177.144 |  |  |
| Average Score (SAT Reading) | 375 | 159189 | 424.504 | 3829.267 |  |  |
| Average Score (SAT Writing) | 375 | 156922 | 418.4587 | 4166.522 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| ANOVA |  |  |  |  |  |  |
| *Source of Variation* | *SS* | *df* | *MS* | *F* | *P-value* | *F crit* |
| Between Groups | 39700.57 | 2 | 19850.28 | 4.520698 | 0.01108 | 3.003745 |
| Within Groups | 4926677 | 1122 | 4390.977 |  |  |  |
|  |  |  |  |  |  |  |
| Total | 4966377 | 1124 |  |  |  |  |

Since theresulting pvalueis less than 0.05. The null hypothesis (H0) is rejected and conclude that there is a significant difference between the SAT scores for each subject.

35

1. **Perform testing of hypothesis using Two-way ANOVA.**

#### Program Code:

import pandas as pd

import statsmodels.api as sm

from statsmodels.formula.api import ols

from statsmodels.stats.anova import anova\_lm

from statsmodels.graphics.factorplots import interaction\_plot import matplotlib.pyplot as plt

from scipy import stats

def eta\_squared(aov): aov['eta\_sq'] = 'NaN'

aov['eta\_sq'] = aov[:-1]['sum\_sq']/sum(aov['sum\_sq']) return aov

def omega\_squared(aov):

mse = aov['sum\_sq'][-1]/aov['df'][-1] aov['omega\_sq'] = 'NaN'

aov['omega\_sq'] = (aov[:-1]['sum\_sq']-(aov[:-

1]['df']\*mse))/(sum(aov['sum\_sq'])+mse) return aov

datafile = "ToothGrowth.csv" data = pd.read\_csv(datafile)

fig = interaction\_plot(data.dose, data.supp, data.len, colors=['red','blue'], markers=['D','^'], ms=10)

N = len(data.len)

df\_a = len(data.supp.unique()) - 1 df\_b = len(data.dose.unique()) - 1 df\_axb = df\_a\*df\_b

df\_w = N - (len(data.supp.unique())\*len(data.dose.unique())) grand\_mean = data['len'].mean()

#Sum of Squares A – supp

ssq\_a = sum([(data[data.supp ==l].len.mean()-grand\_mean)\*\*2 for l in data.supp]) #Sum of Squares B – supp

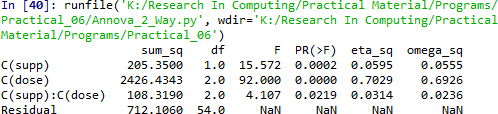
ssq\_b = sum([(data[data.dose ==l].len.mean()-grand\_mean)\*\*2 for l in data.dose]) #Sum of Squares Total

ssq\_t = sum((data.len - grand\_mean)\*\*2) vc = data[data.supp == 'VC']

oj = data[data.supp == 'OJ']

vc\_dose\_means = [vc[vc.dose == d].len.mean() for d in vc.dose]

36



oj\_dose\_means = [oj[oj.dose == d].len.mean() for d in oj.dose]

ssq\_w = sum((oj.len - oj\_dose\_means)\*\*2) +sum((vc.len - vc\_dose\_means)\*\*2) ssq\_axb = ssq\_t-ssq\_a-ssq\_b-ssq\_w

ms\_a = ssq\_a/df\_a #Mean Square A ms\_b = ssq\_b/df\_b #Mean Square B

ms\_axb = ssq\_axb/df\_axb #Mean Square AXB ms\_w = ssq\_w/df\_w

f\_a = ms\_a/ms\_w f\_b = ms\_b/ms\_w

f\_axb = ms\_axb/ms\_w

p\_a = stats.f.sf(f\_a, df\_a, df\_w) p\_b = stats.f.sf(f\_b, df\_b, df\_w)

p\_axb = stats.f.sf(f\_axb, df\_axb, df\_w)

results = {'sum\_sq':[ssq\_a, ssq\_b, ssq\_axb, ssq\_w], 'df':[df\_a, df\_b, df\_axb, df\_w],

'F':[f\_a, f\_b, f\_axb, 'NaN'],

'PR(>F)':[p\_a, p\_b, p\_axb, 'NaN']}

columns=['sum\_sq', 'df', 'F', 'PR(>F)']

aov\_table1 = pd.DataFrame(results, columns=columns, index=['supp', 'dose',

'supp:dose', 'Residual'])

formula = 'len ~ C(supp) + C(dose) + C(supp):C(dose)' model = ols(formula, data).fit()

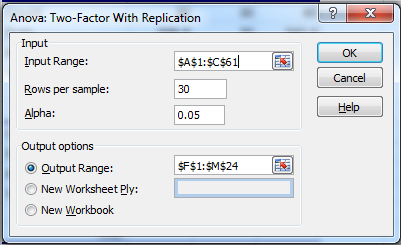
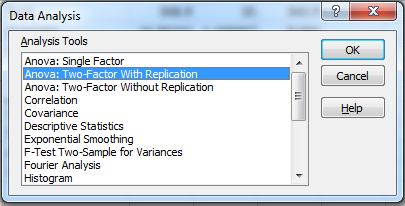
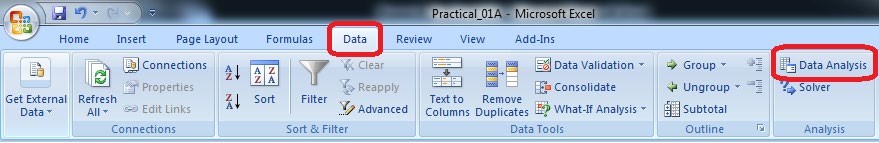
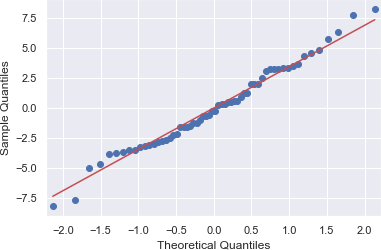
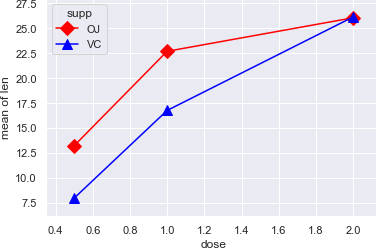
aov\_table = anova\_lm(model, typ=2) eta\_squared(aov\_table) omega\_squared(aov\_table) print(aov\_table.round(4))

res = model.resid

fig = sm.qqplot(res, line='s') plt.show()

## Output:

37



## Using Excel:

Go to Data tab  Data Analysis

Input Range - $A$1:$C$61

38

Rows Per Sample – 30 (Beacause 30 Patients are given each dose) Alpha – 0.05

Output Range - $F$1:$M$24

## Output:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Anova: Two-Factor With Replication** | | |  |  |  |  |
|  |  |  |  |  |  |  |
| SUMMARY | len | dose | Total |  |  |  |
| *1* |  |  |  |  |  |  |
| Count | 30 | 30 | 60 |  |  |  |
| Sum | 508.9 | 35 | 543.9 |  |  |  |
| Average | 16.96333 | 1.166667 | 9.065 |  |  |  |
| Variance | 68.32723 | 0.402299 | 97.22333 |  |  |  |
|  |  |  |  |  |  |  |
| *31* |  |  |  |  |  |  |
| Count | 30 | 30 | 60 |  |  |  |
| Sum | 619.9 | 35 | 654.9 |  |  |  |
| Average | 20.66333 | 1.166667 | 10.915 |  |  |  |
| Variance | 43.63344 | 0.402299 | 118.2854 |  |  |  |
|  |  |  |  |  |  |  |
| *Total* |  |  |  |  |  |  |
| Count | 60 | 60 |  |  |  |  |
| Sum | 1128.8 | 70 |  |  |  |  |
| Average | 18.81333 | 1.166667 |  |  |  |  |
| Variance | 58.51202 | 0.39548 |  |  |  |  |
| **ANOVA** |  |  |  |  |  |  |
| *Source of*  *Variation* | *SS* | *df* | *MS* | *F* | *P-value* | *F crit* |
| Sample | 102.675 | 1 | 102.675 | 3.642079 | 0.058808 | 3.922879 |
| Columns | 9342.145 | 1 | 9342.145 | 331.3838 | 8.55E-36 | 3.922879 |
| Interaction | 102.675 | 1 | 102.675 | 3.642079 | 0.058808 | 3.922879 |
| Within | 3270.193 | 116 | 28.19132 |  |  |  |
| Total | 12817.69 | 119 |  |  |  |  |

P-value = 0.0588079 column in the ANOVA Source of Variation table at the bottom of the output. Because the p-values for both medicin dose and interaction are less than our significance level, these factors are statistically significant. On the other hand, the interaction effect is not significant because its p-value (0.0588) is greater than our significance level. Because the interaction effect is not significant, we can focus on only the main effects and not consider the interaction effect of the dose.

39



## Perform testing of hypothesis using MANOVA.

MANOVA is the acronym for Multivariate Analysis of Variance. When analyzing data, we may encounter situations where we have there multiple response variables (dependent variables). In MANOVA there also some assumptions, like ANOVA. Before performing MANOVA we have to check the following assumptions are satisfied or not.

* + The samples, while drawing, should be independent of each other.
  + The dependent variables are continuous in nature and the independent variables are categorical.
  + The dependent variables should follow a multivariate normal distribution.
  + The population variance-covariance matrices of each group are same, i.e. groups are homogeneous.

#### Code:

import pandas as pd fromstatsmodels.multivariate.manova import MANOVA df = pd.read\_csv('iris.csv', index\_col=0)

df.columns = df.columns.str.replace(".", "\_") df.head()

print('~~~~~~~~ Data Set ~~~~~~~~') print(df)

maov = MANOVA.from\_formula('Sepal\_Length + Sepal\_Width + \ Petal\_Length + Petal\_Width ~ Species', data=df)

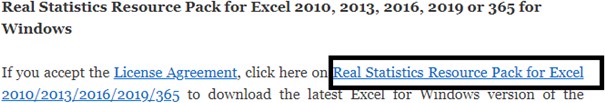
print('~~~~~~~~ MANOVA Test Result ~~~~~~~~') print(maov.mv\_test())

**Output:**

# Excel:

40

Go to <http://www.real-statistics.com/free-download/>



1. Download Real Statistics Resource Pack

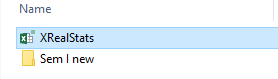
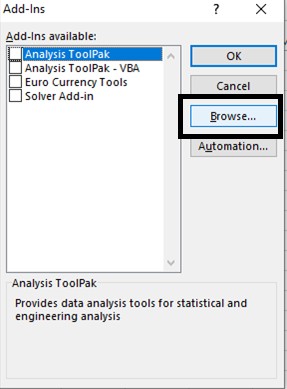
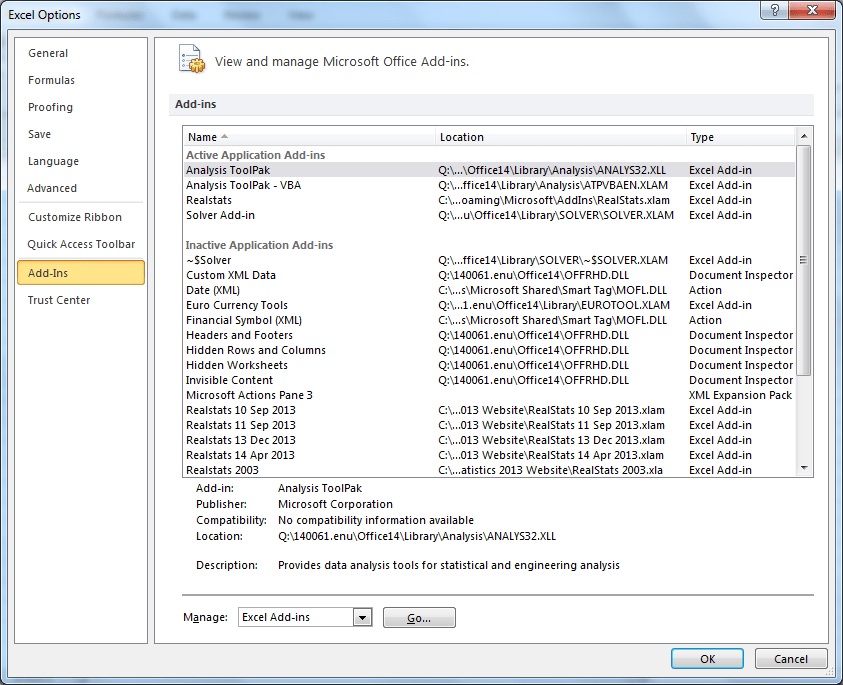
Or

<http://www.real-statistics.com/wp-content/uploads/2019/11/XRealStats.xlam>

Install Add-in in excel. Select **File > Help|Options > Add-Ins** and click on the **Go** button at the bottom of the window (see Figure 1).

Add-ins -> Analysis Pack -> Go

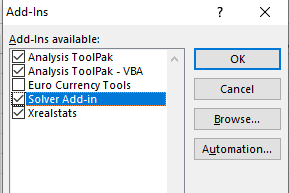
41



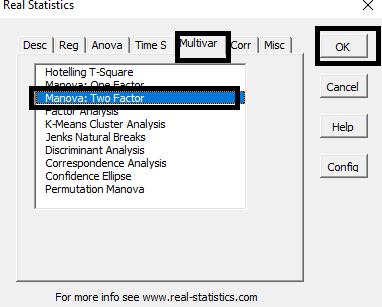
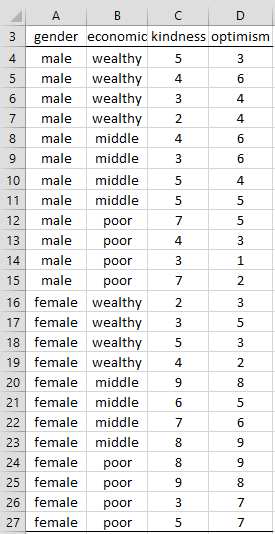
Click on browse and select XrealStats file (previously downloaded).

Select the following Add-Ins. Click OK.

42



43

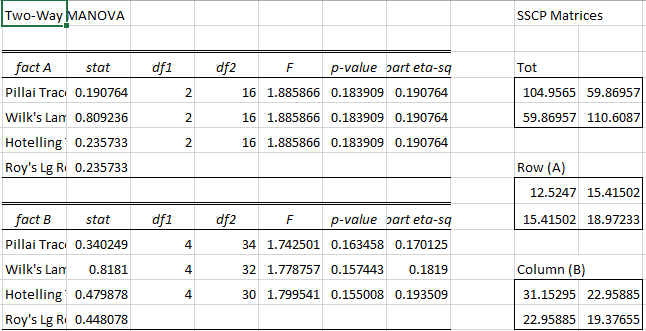
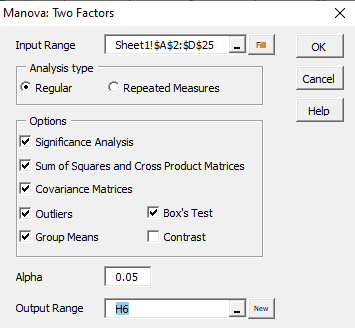


Now create an excel sheet with following data.

A study was conducted to see the impact of social-economic class (rich, middle, poor) and gender (male, female) on kindness and optimism using on a sample of 24 people based on the data in Figure 1.

Press ctrl-m to open Real Statistics menu.

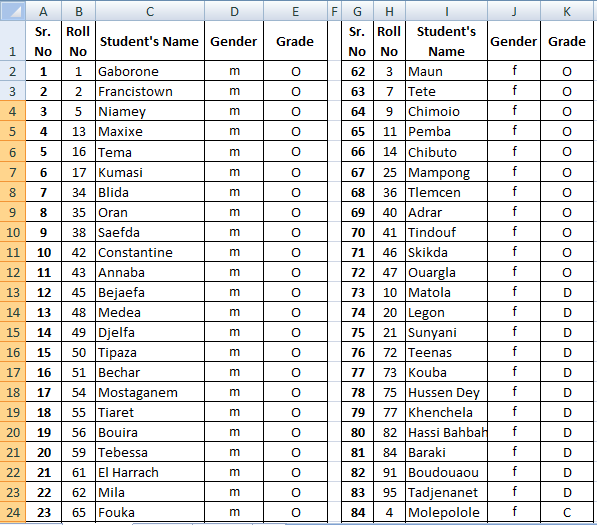
44



Select the data excluding column names. Select a cell for output.

Output:

45



# Practical 7:

## Perform the Random sampling for the given data and analyse it.

**Example 1**: From a population of 10 women and 10 men as given in the table in Figure 1 on the left below, create a random sample of 6 people for Group 1 and a periodic sample consisting of every 3rd woman for Group 2.

You need to run the sampling data analysis tool twice, once to create Group 1 and again to create Group 2. For Group 1 you select all 20 population cells as the Input Range and Random as the Sampling Method with 6 for the Random Number of Samples. For Group 2 you select the 10 cells in the Women column as Input Range and Periodic with Period 3.

Open existing excel sheet with population data Sample Sheet looks as given below:

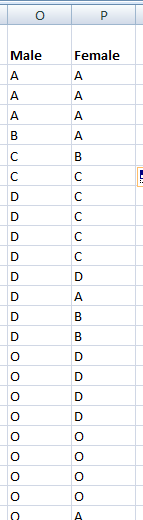
Set Cell O1 = Male and Cell O2 = Female

To generate a random sample for male students from given population go to Cell O1 and type

=INDEX(E$2:E$62,RANK(B2,B$2:B$62))

Drag teh formula to the desired no of cell to select random sample.

46



Now, to generate a random sample for female students go to cell P1 and type

=INDEX(K$2:K$40,RANK(H2,H$2:H$40))

Drag teh formula to the desired no of cell to select random sample.

#### Output:

47

## Perform the Stratified sampling for the given data and analyse it.

we are to carry out a **hypothetical** housing quality survey across Lagos state, Nigeria. And we looking at a total of 5000 houses (hypothetically). We don’t just go to one local government and select 5000 houses, rather we ensure that the 5000 houses are a representative of the whole 20 local government areas Lagos state is comprised of. This is called stratified sampling. The population is divided into homogenous strata and the right number of instances is sampled from each stratum to guarantee that the test-set (which in this case is the 5000 houses) is a representative of the overall population. If we used random sampling, there would be a significant chance of having bias in the survey results.

#### Program Code:

import pandas as pd importnumpy as np

importmatplotlib importmatplotlib.pyplot as plt

plt.rcParams['axes.labelsize'] = 14

plt.rcParams['xtick.labelsize'] = 12

plt.rcParams['ytick.labelsize'] = 12

importseaborn as sns color = sns.color\_palette() sns.set\_style('darkgrid')

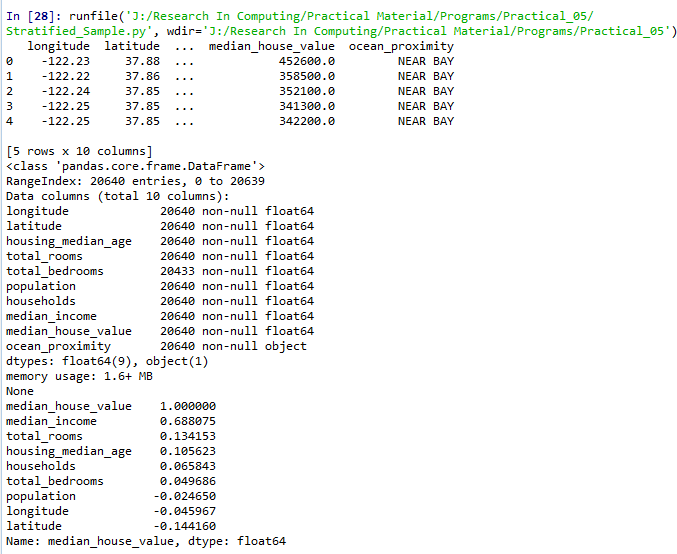
importsklearn

fromsklearn.model\_selection import train\_test\_split

housing =pd.read\_csv('housing.csv') print(housing.head()) print(housing.info())

#creating a heatmap of the attributes in the dataset

48



correlation\_matrix = housing.corr() plt.subplots(figsize=(8,6))

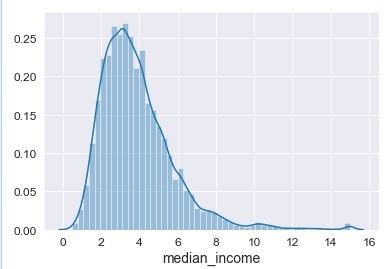
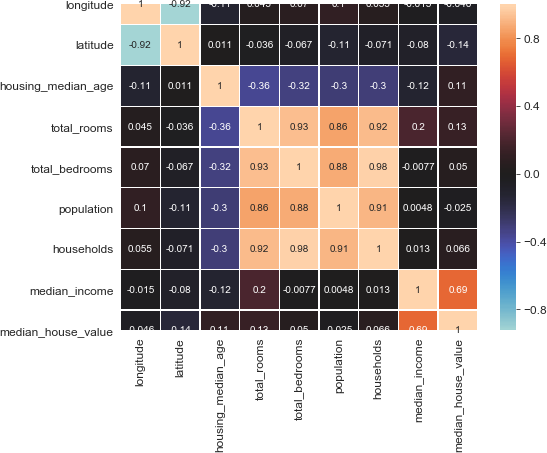
sns.heatmap(correlation\_matrix, center=0, annot=True, linewidths=.3)

corr =housing.corr() print(corr['median\_house\_value'].sort\_values(ascending=False))

sns.distplot(housing.median\_income) plt.show()

#### output:

49

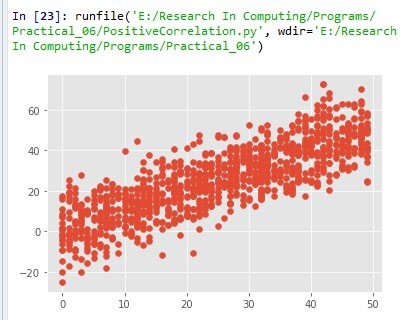


There’s a ton of information we can mine from the heatmap above, a couple of strongly positively correlated features and a couple of negatively correlated features. Take a look at the small bright box right in the middle of the heatmap from total\_rooms on the left ’y-axis’ till households and note how bright the box is as well as the highly positively correlated attributes, also note that median\_income is the most correlated feature to the target which is median\_house\_value.

From the image above, we can see that most median incomes are clustered between

$20,000 and $50,000 with some outliers going far beyond $60,000 making the distribution skew to the right.

50



# Practical 8:

## Write a program for computing different correlation.

#### Positive Correlation:

Let’s take a look at a positive correlation. Numpy implements a corrcoef() function that returns a matrix of correlations of x with x, x with y, y with x and y with y. We’re interested in the values of correlation of x with y (so position (1, 0) or (0, 1)).

#### Code:

importnumpy as np importmatplotlib.pyplot as plt np.random.seed(1)

# 1000 random integers between 0 and 50

x = np.random.randint(0, 50, 1000)

# Positive Correlation with some noise y = x + np.random.normal(0, 10, 1000)

np.corrcoef(x, y) matplotlib.style.use('ggplot')

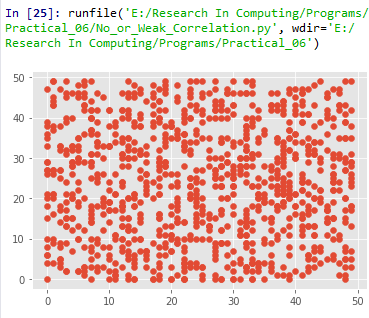
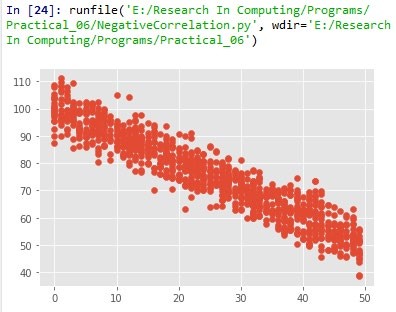
plt.scatter(x, y) plt.show()

#### Output:

**Negative Correlation:** importnumpy as np importmatplotlib.pyplot as plt np.random.seed(1)

# 1000 random integers between 0 and 50

51



x = np.random.randint(0, 50, 1000)

# Negative Correlation with some noise

y = 100 - x + np.random.normal(0, 5, 1000)

np.corrcoef(x, y) plt.scatter(x, y) plt.show()

#### Output:

**No/Weak Correlation:** importnumpy as np importmatplotlib.pyplot as plt np.random.seed(1)

x = np.random.randint(0, 50, 1000)

y = np.random.randint(0, 50, 1000) np.corrcoef(x, y)

plt.scatter(x, y) plt.show() **Output:**

52

# Practical 9:

## Write a program to Perform linear regression for prediction.

# -\*- coding: utf-8 -\*- """

Created on Mon Dec 16 21:56:32 2019 @author: MyHome

"""

import Quandl, math import numpy as np import pandas as pd

from sklearn import preprocessing, cross\_validation, svm from sklearn.linear\_model import LinearRegression import matplotlib.pyplot as plt

from matplotlib import style import datetime

style.use('ggplot')

df = Quandl.get("WIKI/GOOGL")

df = df[['Adj. Open', 'Adj. High', 'Adj. Low', 'Adj. Close', 'Adj. Volume']]

df['HL\_PCT'] = (df['Adj. High'] - df['Adj. Low']) / df['Adj. Close'] \* 100.0

df['PCT\_change'] = (df['Adj. Close'] - df['Adj. Open']) / df['Adj. Open'] \* 100.0

df = df[['Adj. Close', 'HL\_PCT', 'PCT\_change', 'Adj. Volume']] forecast\_col = 'Adj. Close'

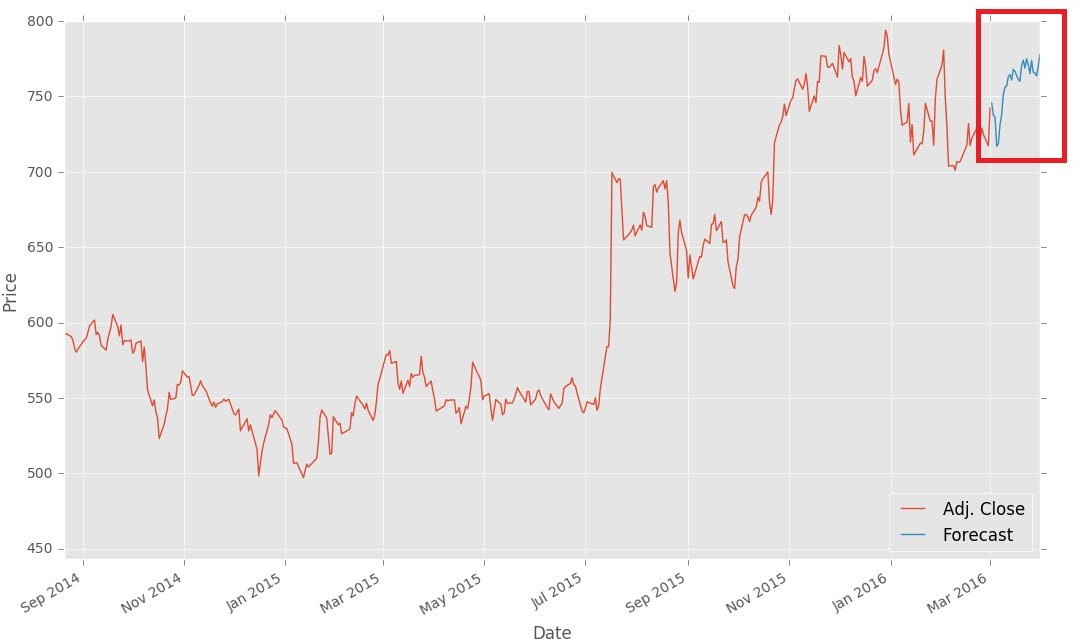
df.fillna(value=-99999, inplace=True) forecast\_out = int(math.ceil(0.01 \* len(df))) df['label'] = df[forecast\_col].shift(-forecast\_out)

X = np.array(df.drop(['label'], 1)) X = preprocessing.scale(X) X\_lately = X[-forecast\_out:]

X = X[:-forecast\_out]

df.dropna(inplace=True) y = np.array(df['label'])

53



X\_train, X\_test, y\_train, y\_test = cross\_validation.train\_test\_split(X, y, test\_size=0.2) clf = LinearRegression(n\_jobs=-1)

clf.fit(X\_train, y\_train)

confidence = clf.score(X\_test, y\_test)

forecast\_set = clf.predict(X\_lately) df['Forecast'] = np.nan

last\_date = df.iloc[-1].name last\_unix = last\_date.timestamp() one\_day = 86400

next\_unix = last\_unix + one\_day

for i in forecast\_set:

next\_date = datetime.datetime.fromtimestamp(next\_unix) next\_unix += 86400

df.loc[next\_date] = [np.nan for \_ in range(len(df.columns)-1)]+[i]

df['Adj. Close'].plot()

df['Forecast'].plot() plt.legend(loc=4) plt.xlabel('Date') plt.ylabel('Price') plt.show()

### Output:

54

## Perform polynomial regression for prediction.

importnumpy as np importmatplotlib.pyplot as plt

defestimate\_coef(x, y):

# number of observations/points n = np.size(x)

# mean of x and y vector

m\_x, m\_y = np.mean(x), np.mean(y)

# calculating cross-deviation and deviation about x SS\_xy = np.sum(y\*x) - n\*m\_y\*m\_x

SS\_xx = np.sum(x\*x) - n\*m\_x\*m\_x

# calculating regression coefficients b\_1 = SS\_xy / SS\_xx

b\_0 = m\_y - b\_1\*m\_x return(b\_0, b\_1)

defplot\_regression\_line(x, y, b):

# plotting the actual points as scatter plot plt.scatter(x, y, color = "m",

marker = "o", s = 30)

# predicted response vector y\_pred = b[0] + b[1]\*x

# plotting the regression line plt.plot(x, y\_pred, color = "g")

# putting labels plt.xlabel('x')

plt.ylabel('y')

# function to show plot plt.show()

def main():

# observations

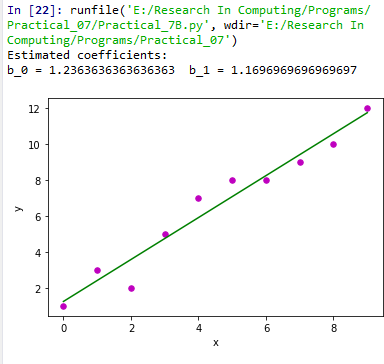
x = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

y = np.array([1, 3, 2, 5, 7, 8, 8, 9, 10, 12])

# estimating coefficients b = estimate\_coef(x, y)

print("Estimated coefficients:\nb\_0 = {} b\_1 = {}".format(b[0], b[1]))

55



# plotting regression line plot\_regression\_line(x, y, b)

if name == " main ": main()

# Output:

56

# Practical 10:

## Write a program for multiple linear regression analysis.

**Step #1:** Data Pre Processing

* 1. Importing The Libraries.
  2. Importing the Data Set.
  3. Encoding the Categorical Data.
  4. Avoiding the Dummy Variable Trap.
  5. Splitting the Data set into Training Set and Test Set.

**Step #2:** Fitting Multiple Linear Regression to the Training set

**Step #3:** Predicting the Test set results.

importnumpy as np importmatplotlib as mpl

from mpl\_toolkits.mplot3d import Axes3D importmatplotlib.pyplot as plt defgenerate\_dataset(n):

x = []

y = []

random\_x1 = np.random.rand() random\_x2 = np.random.rand() fori in range(n):

x1 = i

x2 = i/2 + np.random.rand()\*n x.append([1, x1, x2])

y.append(random\_x1 \* x1 + random\_x2 \* x2 + 1) returnnp.array(x), np.array(y)

x, y = generate\_dataset(200) mpl.rcParams['legend.fontsize'] = 12 fig = plt.figure()

ax = fig.gca(projection ='3d')

ax.scatter(x[:, 1], x[:, 2], y, label ='y', s = 5) ax.legend()

ax.view\_init(45, 0) plt.show() defmse(coef, x, y):

57

returnnp.mean((np.dot(x, coef) - y)\*\*2)/2 def gradients(coef, x, y):

returnnp.mean(x.transpose()\*(np.dot(x, coef) - y), axis = 1) defmultilinear\_regression(coef, x, y, lr, b1 = 0.9, b2 = 0.999, epsilon = 1e-8):

prev\_error = 0

m\_coef = np.zeros(coef.shape) v\_coef = np.zeros(coef.shape)

moment\_m\_coef = np.zeros(coef.shape) moment\_v\_coef = np.zeros(coef.shape) t = 0

while True:

error = mse(coef, x, y)

if abs(error - prev\_error) <= epsilon: break

prev\_error = error

grad = gradients(coef, x, y) t += 1

m\_coef = b1 \* m\_coef + (1-b1)\*grad v\_coef = b2 \* v\_coef + (1-b2)\*grad\*\*2 moment\_m\_coef = m\_coef / (1-b1\*\*t) moment\_v\_coef = v\_coef / (1-b2\*\*t)

delta = ((lr / moment\_v\_coef\*\*0.5 + 1e-8) \*

(b1 \* moment\_m\_coef + (1-b1)\*grad/(1-b1\*\*t))) coef = np.subtract(coef, delta)

returncoef

coef = np.array([0, 0, 0])

c = multilinear\_regression(coef, x, y, 1e-1) fig = plt.figure()

ax = fig.gca(projection ='3d') ax.scatter(x[:, 1], x[:, 2], y, label ='y',

s = 5, color ="dodgerblue") ax.scatter(x[:, 1], x[:, 2], c[0] + c[1]\*x[:, 1] + c[2]\*x[:, 2],

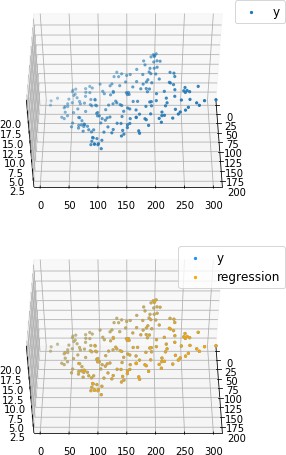
label ='regression', s = 5, color ="orange")

ax.view\_init(45, 0) ax.legend()

plt.show()

58

# Output:



59

## Perform logistic regression analysis.

Logistic regression is a classification method built on the same concept as linear

regression. With linear regression, we take linear combination of explanatory variables plus an intercept term to arrive at a prediction.

In this example we will use a logistic regression model to predict survival.

### Program Code:

import os

import numpy as np import pandas as pd import matplotlib

import matplotlib.pyplot as plt import scipy.stats as stats

from sklearn import linear\_model from sklearn import preprocessing from sklearn import metrics

matplotlib.style.use('ggplot') plt.figure(figsize=(9,9))

def sigmoid(t): # Define the sigmoid function return (1/(1 + np.e\*\*(-t)))

plot\_range = np.arange(-6, 6, 0.1) y\_values = sigmoid(plot\_range) # Plot curve

plt.plot(plot\_range, # X-axis range

y\_values, # Predicted values color="red")

titanic\_train = pd.read\_csv("titanic\_train.csv") # Read the data char\_cabin = titanic\_train["Cabin"].astype(str) # Convert cabin to str new\_Cabin = np.array([cabin[0] for cabin in char\_cabin]) # Take first letter

titanic\_train["Cabin"] = pd.Categorical(new\_Cabin) # Save the new cabin var # Impute median Age for NA Age values

new\_age\_var = np.where(titanic\_train["Age"].isnull(), # Logical check 28, # Value if check is true

60

titanic\_train["Age"]) # Value if check is false titanic\_train["Age"] = new\_age\_var

label\_encoder = preprocessing.LabelEncoder()

# Convert Sex variable to numeric

encoded\_sex = label\_encoder.fit\_transform(titanic\_train["Sex"])

# Initialize logistic regression model

log\_model = linear\_model.LogisticRegression()

# Train the model

log\_model.fit(X = pd.DataFrame(encoded\_sex), y = titanic\_train["Survived"])

# Check trained model intercept print(log\_model.intercept\_)

# Check trained model coefficients print(log\_model.coef\_)

# Make predictions

preds = log\_model.predict\_proba(X= pd.DataFrame(encoded\_sex)) preds = pd.DataFrame(preds)

preds.columns = ["Death\_prob", "Survival\_prob"]

# Generate table of predictions vs Sex pd.crosstab(titanic\_train["Sex"], preds.ix[:, "Survival\_prob"])

# Convert more variables to numeric

encoded\_class = label\_encoder.fit\_transform(titanic\_train["Pclass"]) encoded\_cabin = label\_encoder.fit\_transform(titanic\_train["Cabin"])

train\_features = pd.DataFrame([encoded\_class,

encoded\_cabin, encoded\_sex, titanic\_train["Age"]]).T

# Initialize logistic regression model

log\_model = linear\_model.LogisticRegression()

61

# Train the model log\_model.fit(X = train\_features ,

y = titanic\_train["Survived"])

# Check trained model intercept print(log\_model.intercept\_)

# Check trained model coefficients print(log\_model.coef\_)

# Make predictions

preds = log\_model.predict(X= train\_features)

# Generate table of predictions vs actual pd.crosstab(preds,titanic\_train["Survived"])

log\_model.score(X = train\_features ,

y = titanic\_train["Survived"])

metrics.confusion\_matrix(y\_true=titanic\_train["Survived"], # True labels y\_pred=preds) # Predicted labels

# View summary of common classification metrics print(metrics.classification\_report(y\_true=titanic\_train["Survived"],

y\_pred=preds) )

# Read and prepare test data

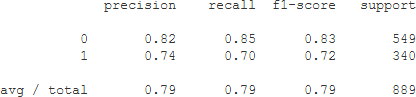
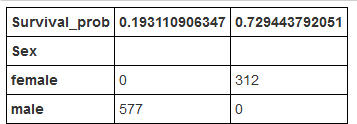
titanic\_test = pd.read\_csv("titanic\_test.csv") # Read the data char\_cabin = titanic\_test["Cabin"].astype(str) # Convert cabin to str

new\_Cabin = np.array([cabin[0] for cabin in char\_cabin]) # Take first letter titanic\_test["Cabin"] = pd.Categorical(new\_Cabin) # Save the new cabin var

# Impute median Age for NA Age values

new\_age\_var = np.where(titanic\_test["Age"].isnull(), # Logical check 28, # Value if check is true titanic\_test["Age"]) # Value if check is false

62



titanic\_test["Age"] = new\_age\_var

# Convert test variables to match model features

encoded\_sex = label\_encoder.fit\_transform(titanic\_test["Sex"]) encoded\_class = label\_encoder.fit\_transform(titanic\_test["Pclass"]) encoded\_cabin = label\_encoder.fit\_transform(titanic\_test["Cabin"])

test\_features = pd.DataFrame([encoded\_class,

encoded\_cabin,encoded\_sex,titanic\_test["Age"]]).T

# Make test set predictions

test\_preds = log\_model.predict(X=test\_features)

# Create a submission for Kaggle

submission = pd.DataFrame({"PassengerId":titanic\_test["PassengerId"], "Survived":test\_preds})

# Save submission to CSV submission.to\_csv("tutorial\_logreg\_submission.csv",

index=False) # Do not save index values

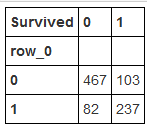
print(pd)

### Output:

The table shows that the model predicted a survival chance of roughly 19% for males and 73% for females.

For the Titanic competition, accuracy is the scoring metric used to judge the competition, so we don't have to worry too much about other metrics.

63



The table above shows the classes our model predicted vs. true values of the Survived variable.

This logistic regression model has an accuracy score of 0.75598 which is actually

worse than the accuracy of the simplistic women survive, men die model (0.76555).

# Example 2:

The dataset is related to direct marketing campaigns (phone calls) of a Portuguese banking institution. The classification goal is to predict whether the client will subscribe (1/0) to a term deposit (variable y). The dataset provides the bank customers’ information. It includes 41,188 records and 21 fields.

#### Input variables

1. **age** (numeric)
2. **job :** type of job (categorical: “admin”, “blue-collar”, “entrepreneur”, “housemaid”, “management”, “retired”, “self-employed”, “services”, “student”, “technician”, “unemployed”, “unknown”)
3. **marital :** marital status (categorical: “divorced”, “married”, “single”, “unknown”)
4. **education** (categorical: “basic.4y”, “basic.6y”, “basic.9y”, “high.school”, “illiterate”, “professional.course”, “university.degree”, “unknown”)
5. **default:** has credit in default? (categorical: “no”, “yes”, “unknown”)
6. **housing:** has housing loan? (categorical: “no”, “yes”, “unknown”)
7. **loan:** has personal loan? (categorical: “no”, “yes”, “unknown”)
8. **contact:** contact communication type (categorical: “cellular”, “telephone”)
9. **month:** last contact month of year (categorical: “jan”, “feb”, “mar”, …, “nov”, “dec”)
10. **day\_of\_week:** last contact day of the week (categorical: “mon”, “tue”, “wed”, “thu”, “fri”)
11. **duration:** last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y=’no’). The duration is not known before a call is performed, also, after the end of the call, y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model

64

1. **campaign:** number of contacts performed during this campaign and for this client (numeric, includes last contact)
2. **pdays:** number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
3. **previous:** number of contacts performed before this campaign and for this client (numeric)
4. **poutcome:** outcome of the previous marketing campaign (categorical: “failure”, “nonexistent”, “success”)
5. **emp.var.rate:** employment variation rate — (numeric)
6. **cons.price.idx:** consumer price index — (numeric)
7. **cons.conf.idx**: consumer confidence index — (numeric)
8. **euribor3m:** euribor 3 month rate — (numeric)
9. **nr.employed:** number of employees — (numeric)

#### Predict variable (desired target):

**y — has the client subscribed a term deposit? (binary: “1”, means “Yes”, “0” means “No”)**

#### Program Code:

# -\*- coding: utf-8 -\*- """

Created on Mon Dec 16 22:24:44 2019 @author: MyHome

"""

import pandas as pd import numpy as np

from sklearn import preprocessing import matplotlib.pyplot as plt plt.rc("font", size=14)

from sklearn.linear\_model import LogisticRegression from sklearn.model\_selection import train\_test\_split import seaborn as sns

sns.set(style="white") sns.set(style="whitegrid", color\_codes=True) data = pd.read\_csv('bank.csv', header=0) data = data.dropna()

print(data.shape) print(list(data.columns)) data['education'].unique()

data['education']=np.where(data['education'] =='basic.9y', 'Basic', data['education']) data['education']=np.where(data['education'] =='basic.6y', 'Basic', data['education'])

65

data['education']=np.where(data['education'] =='basic.4y', 'Basic', data['education']) data['education'].unique()

data['y'].value\_counts()

sns.countplot(x='y', data=data, palette='hls') plt.show();

plt.savefig('Practical10B-plot.jpeg')

count\_no\_sub = len(data[data['y']==0]) count\_sub = len(data[data['y']==1])

pct\_of\_no\_sub = count\_no\_sub/(count\_no\_sub+count\_sub) print("percentage of no subscription is", pct\_of\_no\_sub\*100) pct\_of\_sub = count\_sub/(count\_no\_sub+count\_sub) print("percentage of subscription", pct\_of\_sub\*100)

data.groupby('y').mean()

data.groupby('job').mean() data.groupby('marital').mean() data.groupby('education').mean()

########### Purchase Frequency for Job Title pd.crosstab(data.job,data.y).plot(kind='bar') plt.title('Purchase Frequency for Job Title') plt.xlabel('Job')

plt.ylabel('Frequency of Purchase') plt.savefig('purchase\_fre\_job')

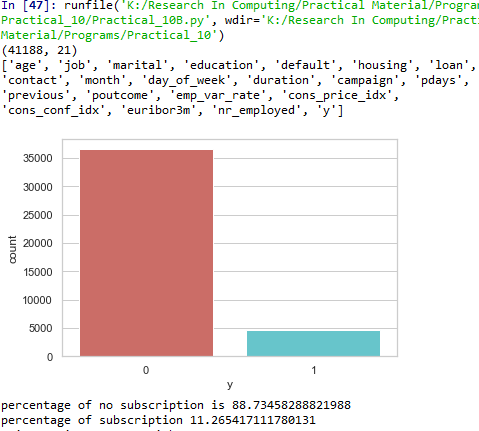
###################### Marital Status vs Purchase table=pd.crosstab(data.marital,data.y)

table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True) plt.title('Stacked Bar Chart of Marital Status vs Purchase') plt.xlabel('Marital Status')

plt.ylabel('Proportion of Customers') plt.savefig('mariral\_vs\_pur\_stack')

############ Education vs Purchase table=pd.crosstab(data.education,data.y) table.div(table.sum(1).astype(float), axis=0).plot(kind='bar', stacked=True) plt.title('Stacked Bar Chart of Education vs Purchase')

66



plt.xlabel('Education') plt.ylabel('Proportion of Customers') plt.savefig('edu\_vs\_pur\_stack')

pd.crosstab(data.day\_of\_week,data.y).plot(kind='bar') plt.title('Purchase Frequency for Day of Week') plt.xlabel('Day of Week')

plt.ylabel('Frequency of Purchase') plt.savefig('pur\_dayofweek\_bar')

############ Purchase Frequency for Month pd.crosstab(data.month,data.y).plot(kind='bar') plt.title('Purchase Frequency for Month') plt.xlabel('Month')

plt.ylabel('Frequency of Purchase') plt.savefig('pur\_fre\_month\_bar')

########### Age Purchase frequency pattern data.age.hist()

plt.title('Histogram of Age') plt.xlabel('Age') plt.ylabel('Frequency') plt.savefig('hist\_age')

**Output: -**

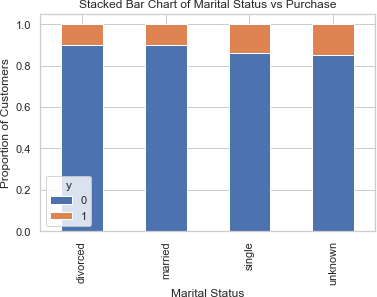
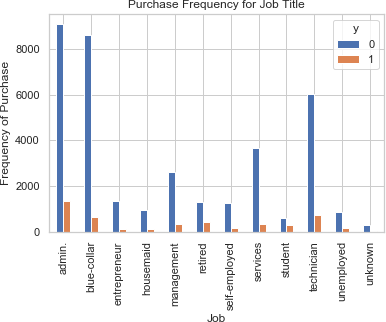
***percentage of no subscription is 88.73458288821988***

***percentage of subscription 11.265417111780131***

Our classes are imbalanced, and the ratio of no- subscription to subscription instances is 89:11.

* The average age of customers who bought the term deposit is higher than that of the customers who didn’t.
* The pdays (days since the customer was last contacted) is understandably lower for the customers who bought it. The lower the pdays, the better the memory of the last call and hence the better chances of a sale.
* Surprisingly, campaigns (number of contacts or calls made during the current campaign)

67

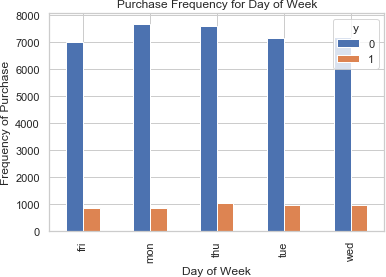
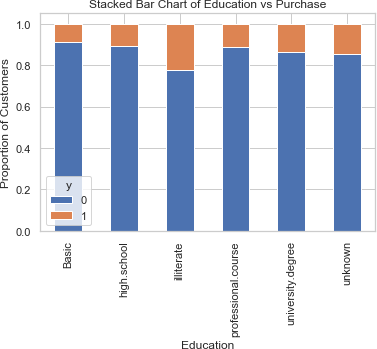


are lower for customers who bought the term deposit.

The frequency of purchase of the deposit depends a great deal on the job title. Thus, the job title can be a good predictor of the outcome variable.

The marital status does not seem a strong predictor for the outcome variable.

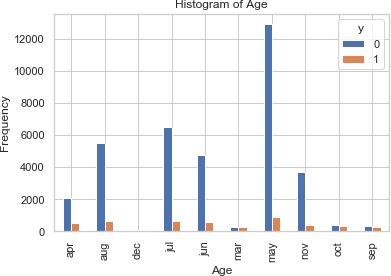
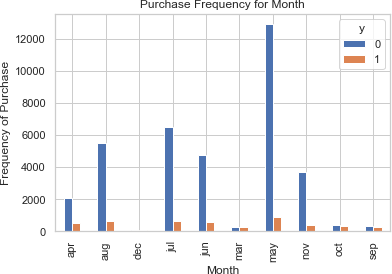
68



Education seems a good predictor of the outcome variable.

Day of week may not be a good predictor of the outcome.

69



Month might be a good predictor of the outcome variable.

Most of the customers of the bank in this dataset are in the age range of 30–40.

70

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Practical No** | | **Name of the Practical** | **File Names** |
| **1)** | **1** | **A** | Write a program for obtaining descriptive  statistics of data. |  |
| **2)** | **B** | Import data from different data sources (from Excel, csv, mysql, sql server, oracle to  R/Python/Excel) | As required in Data Science Pracitcal |
| **3)** | **2** | **A** | Design a survey form for a given case study, collect the primary data and analyse it |  |
| **4)** | **B** | Perform suitable analysis of given secondary  data. |  |
| **5)** | **3** | **A** | Perform testing of hypothesis using one sample  t-test. | ages.csv |
| **6)** | **B** | Perform testing of hypothesis using two sample  t-test. |  |
| **7)** | **C** | Perform testing of hypothesis using paired t-test. | blood\_pressure.csv |
| **8)** | **4** | **A** | Perform testing of hypothesis using chi-squared  goodness-of-fit test. | Students\_Score.xlsx |
| **9)** | **B** | Perform testing of hypothesis using chi-squared  Test of Independence |  |
| **10)** | **5** |  | Perform testing of hypothesis using Z-test. | blood\_pressure.csv |
| **11)** | **6** | **A** | Perform testing of hypothesis using one-way  ANOVA. | scores.csv  scores.xlsx |
| **12)** | **B** | Perform testing of hypothesis using two-way  ANOVA. | ToothGrowth.csv |
| **13)** | **C** | Perform testing of hypothesis using multivariate  ANOVA (MANOVA). | iris.csv |
| **14)** | **7** | **A** | Perform the Random sampling for the given  data and analyse it. | Students\_Score.xlsx |
| **15)** | **B** | Perform the Stratified sampling for the given  data and analyse it. | housing.csv |
| **16)** | **8** |  | Compute different types of correlation. |  |
| **17)** | **9** | **A** | Perform linear regression for prediction. |  |
| **18)** | **B** | Perform polynomial regression for prediction. |  |
| **19)** | **10** | **A** | Perform multiple linear regression. |  |
| **20)** | **B** | Perform Logistic regression. | titanic\_train.csv  bank.csv |

71

**Dear Teacher,**

**Please send your valuable feedback and contribution to make this manual more effective.**

**Feel Free to connect us on …….**

[dandhiren@yahoo.co.in](mailto:dandhiren@yahoo.co.in) [patilrajendrab@gmail.com](mailto:patilrajendrab@gmail.com) [ahtesham.shaikh@apcollege.edu.in](mailto:ahtesham.shaikh@apcollege.edu.in)