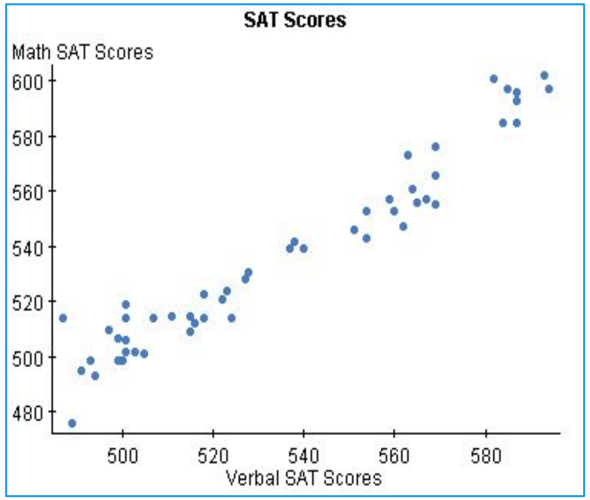
Descriptive Statistics

Bivariate Relationships in Python

Contents

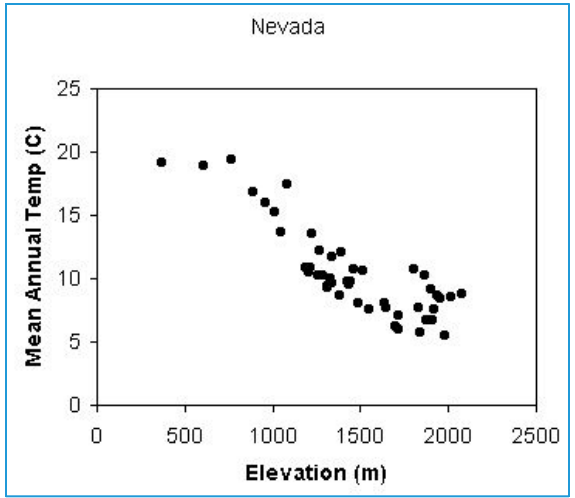
Interpreting a Scatterplot

Positive Correlation



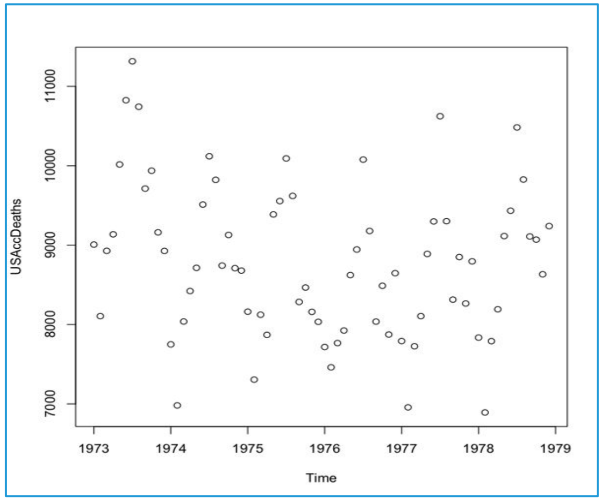
This is a positive sloping (upward) graph.
As the value of one variable increases, the value of other variable also increases.

Negative Correlation



This is a negative sloping (downward) graph.
As the value of one variable increases, the value of other variable tends to decrease.

No Correlation



This is a graph with random pattern.
There is no connection between the two variables. If value of one variable increases, other might increase/decrease.

Pearson's Coefficient of Correlation

The Pearson's correlation coefficient numerically measures the strength of a linear relation between two variables

$$r = \frac{\sum(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum(X_i-\bar{X})^2}\sqrt{\sum(Y_i-\bar{Y})^2}} = \frac{cov\,(X,Y)}{sd(x)sd(y)}$$

RANGE	
Positive Correlation	r > 0
Negative Correlation	r < 0
No Correlation	r = 0

- The two variables can be measured in entirely different units.
- Example, you could correlate a person's age with their blood sugar levels. Here, the units are completely different.
- It is not affected by change of Origin and Scale

* Both Covariance and Pearson's correlation coefficient can be used only for continuous Numeric variables

Simple Linear Regression

The equation of line of best fit is used to describe relationship between two variables

Mathematical form of simple linear regression : $Y = aX + b + e$

Where,

a : Intercept (The value at which the fitted line crosses the y-axis i.e. X=0)
b : Slope of the Line
e : error which is assumed to be a random variable

NOTE : a and b are population parameters which are estimated using sample

Here, variable Y is known as a ‘Dependent' variable, that ‘depends on’ X which is known as the ‘Independent' variable.

Application Areas

Scatter Plot	It is useful in visualising the relationship between any two variables as an initial step. - Life expectancy and the number of cigarettes smoked per day - Literacy rate and life expectancy in a particular region
Correlation Coefficient	It gives the exact numeric measure of the extent of bivariate relationship. - Distance between home & office and the time taken to get there - Size of car engine and cost of car insurance
Simple Linear Regression	It is very useful in predicting the value of one variable given the value of another in a bivariate scenario. - Number of bedrooms and cost of home insurance - Scores in the final exam given the scores in mock test

Case Study - 1

Background

- A company conducts different written tests before recruiting employees. The company wishes to see if the scores of these tests have any relation with post-recruitment performance of those employees.

Objective

- To study the correlation between Aptitude and Job Proficiency.
- Predict the Job proficiency for a given Aptitude score.

Available Information

- Sample size is 33
- Independent Variables: Scores of tests conducted before recruitment on the basis of four criteria – Aptitude, Test of English, Technical Knowledge, General Knowledge
- Dependent Variable: Job Performance Index calculated after an employee finishes probationary period (6 months)

Data Snapshot

Job_Proficiency

Variables

Observations

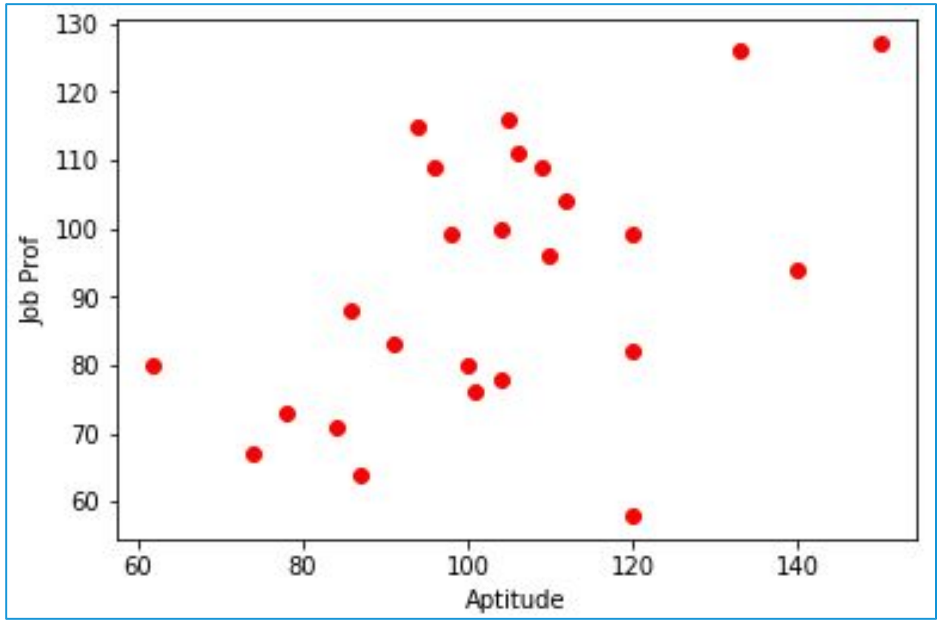
empno	aptitude	testofen	tech_	g_k_	job_prof
1	86	110	100	87	88
2	62	62	99	100	80
3	110	107	103	103	96
4	101	117	93	95	76
5	100	101	95	88	80
6	78	85	95	84	73
7	120	77	80	74	58
8	105	122	116	102	116

Columns	Description	Type	Measurement	Possible values
Empno	Employee Number	numeric	-	positive values
aptitude	Aptitude Score of the Employee	numeric	-	positive values
Testofen	Test of English	numeric	-	positive values
tech_	Technical Score	numeric	-	positive values
g_k	General Knowledge Score	numeric	-	positive values
Job_prof	Job Proficiency Score	numeric	-	positive values

Scatter Plot in Python

```
# Importing Data and necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
job= pd.read_csv("Job_Proficiency.csv")
```

```
# Scatterplot
plt.scatter(job.aptitude,job.job_prof, color='red');
plt.xlabel('Aptitude'); plt.ylabel('Job Prof')
```



- **plt.scatter()** gives a scatterplot of the two variables mentioned.
- **color=** provides color to the points.

Pearson Correlation Coefficient in Python

```
# Correlation
import numpy as np
np.corrcoef(job.aptitude,job.job_prof)
```

corrcoef() calculates Pearson Correlation Coefficient for the two variables mentioned.

```
array([[1.        , 0.51441069],
       [0.51441069, 1.        ]])
```

Pearson Correlation Coefficient	0.5144

There is positive relation between aptitude and job proficiency but the relation is of moderate degree.

Simple Linear Regression in Python

```
# Simple Linear Regression
```

```
import statsmodels.formula.api as smf
model1= smf.ols("job_prof ~ aptitude", data = job).fit()
model1.summary()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:               job_prof   R-squared:                       0.265
Model:                            OLS   Adj. R-squared:                  0.233
Method:                 Least Squares   F-statistic:                     8.276
Date:                Fri, 18 Oct 2019   Prob (F-statistic):            0.00852
Time:                        10:46:39   Log-Likelihood:                -105.28
No. Observations:                  25   AIC:                             214.6
Df Residuals:                      23   BIC:                             217.0
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     41.3216     18.010      2.294      0.031       4.065      78.578
aptitude       0.4922      0.171      2.877      0.009       0.138       0.846
==============================================================================
Omnibus:                        1.110   Durbin-Watson:                   2.409
Prob(Omnibus):                  0.574   Jarque-Bera (JB):                0.746
Skew:                          -0.416   Prob(JB):                        0.689
Kurtosis:                       2.845   Cond. No.                         557.
==============================================================================
```

- **ols()** gives us the linear regression model.
- **summary()** gives us the summary statistics

Inferences : Simple Linear Regression

Dependent Variable : Job Proficiency

Independent Variable : Aptitude

Intercept	Aptitude
41. 3216	0.4922

Equation : Job Proficiency = 41. 3216 + 0.4922 * Aptitude

Here Job Proficiency changes by 0.4992 units with a unit change in aptitude.

Case Study - 2

To learn more Descriptive Statistics in Python, we shall consider the below case as an example.

Background

Data of 100 retailers in platinum segment of an FMCG company.

Objective

To describe the variables present in the data

Sample Size

Sample size: 100
Variables: Retailer, Zone, Retailer_Age, Perindex, Growth, NPS_Category

Data Snapshot

Retail Data

Variables

Retailer	Zone	Retailer_Age	Perindex	Growth	NPS_Category
1	North	<=2	81.84	3.04	Promoter

Observations

Columns	Description	Type	Measurement	Possible values
Retailer	Retailer ID	numeric	-	-
Zone	Location of the retailer	character	North, East, West, South	4
Retailer_Age	Number of years doing business with the company	character	<=2, 2 to 5, >5	3
Perindex	Index of performance based on sales, buying frequency and buying recency	numeric	-	positive values
Growth	Annual sales growth	numeric	-	positive values
NPS_Category	Category indicating loyalty with the company	character	Detractor, Passive, Promoter	3

Summarizing Two Categorical Variables

Using Frequency/Cross Tables describing the counts, percentages, etc. is a very basic and most useful way in summarizing two categorical variables.

```
#Importing Data
retail_data = pd.read_csv('Retail_Data.csv')
```

```
# Frequency Tables
Freq = pd.crosstab(index=retail_data["Zone"],
columns=retail_data["NPS_Category"])
Freq
```

crosstab() in Python, gives the frequency of counts of the two variables mentioned.

```
NPS_Category  Detractor  Passive  Promoter
Zone
East                  5        9         1
North                 5       13         7
South                 7        9        16
West                  6       10        12
```

Summarizing Two Categorical Variables

```
# Percentage Frequency Tables
```

```
Freq = pd.crosstab(index=retail_data["Zone"],
columns=retail_data["NPS_Category"], normalize=True)
Freq
```

NPS_Category Zone	Detractor	Passive	Promoter
East	0.05	0.09	0.01
North	0.05	0.13	0.07
South	0.07	0.09	0.16
West	0.06	0.10	0.12

By specifying **normalize=True** we can get percentage frequency

```
Freq = pd.crosstab(index=retail_data["Zone"],
columns=retail_data["NPS_Category"], normalize='index')
Freq
```

NPS_Category Zone	Detractor	Passive	Promoter
East	0.333333	0.600000	0.066667
North	0.200000	0.520000	0.280000
South	0.218750	0.281250	0.500000
West	0.214286	0.357143	0.428571

- By using **normalize = 'index'** we can get row wise distribution.
- Similarly for columns use **normalize = 'columns'**

Summarizing Three Categorical Variables

```
# Three Way Frequency Table
```

```
table1 = pd.crosstab([retail_data.Zone, retail_data.NPS_Category],
                     retail_data.Retailer_Age, margins = False)
table1
```

```
Retailer_Age          2 to 5  <=2  >5
Zone  NPS_Category
East  Detractor            2    2   1
      Passive              3    3   3
      Promoter             0    0   1
North Detractor            2    2   1
      Passive              6    1   6
      Promoter             0    1   6
South Detractor            2    1   4
      Passive              4    2   3
      Promoter             3    3  10
West  Detractor            3    1   2
      Passive              1    1   8
      Promoter             1    0  11
```

crosstab() in Python, gives the frequency of counts of the three variables in one table itself.

Quick Recap

In this session, we covered bivariate data analysis using Python.

Scatter Plot	• Each dot on the scatterplot is one observation from a data set representing the corresponding variable value on X and Y axis respectively. Here X & Y are continuous variables.
Pearson's Correlation Coefficient	• Numerically measures the strength of a linear relation between two variables
Simple Linear Regression	• The equation of the line of best fit used to describe relationship between two variables