1. Delivery\_time -> Predict delivery time using sorting time

Ans:

import pandas as pd

import numpy as np

df1=pd.read\_csv('C:/Users/ashiq/Desktop/csv/assignment\_4/delivery\_time.csv')

Delivery\_Time Sorting\_Time

count 21.000000 21.000000

mean 16.790952 6.190476

std 5.074901 2.542028

min 8.000000 2.000000

25% 13.500000 4.000000

50% 17.830000 6.000000

75% 19.750000 8.000000

max 29.000000 10.000000

* Null values are not present
* And from box plot ,we got outliers are not present
* From pair plot also ,we will get the correlation plot

(note: all plots are given in python code)

t=df1.corr()

Delivery\_Time Sorting\_Time

Delivery\_Time 1.0 0.8259972607955329

Sorting\_Time 0.8259972607955329 1.0

Correlation value is 0.8259972607955329

Model building:

import statsmodels.formula.api as smf

model=smf.ols('df1.Delivery\_Time~df1.Sorting\_Time', data= df1).fit()

p=model.params

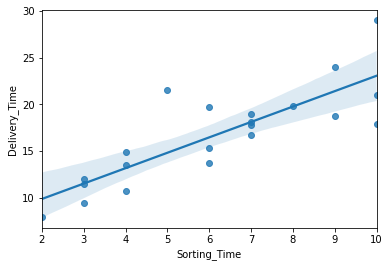
Intercept (BETA\_0) 6.582733971997052

df1.Sorting\_Time (BETA\_1) 1.6490198968312455

Equation is Y= BETA\_0 + BETA\_1\*X +ERROR

sns.regplot(x=df1.Sorting\_Time,y=df1.Delivery\_Time, data=df1)

Regression plot for best fit model



(model.tvalues, model.pvalues)

(6.387447, 0.000004) – can be used in hypothesis testing

(model.rsquared,model.rsquared\_adj)

(0.6822714748417231, 0.6655489208860244) – rsquared value should be close to 1, then it is a good model

Best linear model is : Y= 6.582733971997052+ 1.6490198968312455\*X + ERROR

1. Salary\_hike -> Build a prediction model for Salary\_hike

Ans:

df2=pd.read\_csv('C:/Users/ashiq/Desktop/csv/assignment\_4/Salary\_Data.csv')

t1=df2.corr()

correlation=0.978242

YearsExperience Salary

count 30.000000 30.000000

mean 5.313333 76003.000000

std 2.837888 27414.429785

min 1.100000 37731.000000

25% 3.200000 56720.750000

50% 4.700000 65237.000000

75% 7.700000 100544.750000

max 10.500000 122391.000000

4 business moments are calculated and different plots are taken.

From correlate plot we identify the type of correlation

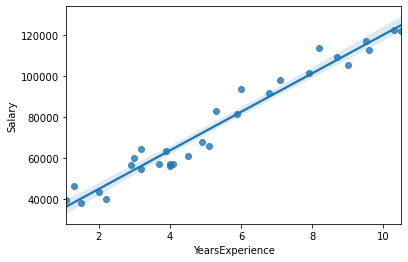
df2.isnull().sum() ## to identify any missing value is there—not present any null value

dist plot can also be used to find the type of distribution

building model:

model1=smf.ols('df2.Salary~df2.YearsExperience', data= df2).fit()

sns.regplot(x=df2.YearsExperience,y=df2.Salary, data=df2)



p1=model1.params

Intercept (BETA\_0) -0.018236

df2.YearsExperience (BETA\_1) 1.049252

(model1.tvalues, model1.pvalues)

(-0.806598, 4.266967e-01) – can be used in hypothesis testing

(model1.rsquared,model1.rsquared\_adj)

(0.9569566641435086, 0.9554194021486339)– rsquared value should be close to 1, then it is a good model

Best linear model is : Y= -0.018236+ 1.049252\*X + ERROR