Introduction to Linear Regression

Agenda

- Business Problem

- Visiting Basics

    - Covariance
    - Correlation

- Regression Analysis

    - Simple Linear Regression
    - Error Calculation

- Ordinary Least Squares Method

    - Best Fit Line
    - Math Behind OLS
    - Interpretation of β coefficients

Agenda

- Measures of Variation
  - Sum of squares total
  - Sum of squares regression
  - Sum of squares of error
  - Standard error of estimate
  - R-Squared

- Inferences about slope
  - t-test for significance of slope and intercept
  - Interval estimation slope and intercept
  - ANOVA for regression
  - t-test for correlation coefficient

Agenda

- **Multiple Linear Regression**
  - Matrix Equation
  - Parameter Estimation - OLS Method
  - Interpretation of β coefficients
  - ANOVA for Multiple Linear Regression

- **Assumptions of Linear Regression**

- **Model Evaluation Metrics**
  - R-Squared
  - Adjusted R-Squared
  - F Test

- **Model Performance Evaluation**
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Error (MAE)
  - Mean Percentage Absolute Error (MAPE)

Simple Linear Regression

Business problem: predict vehicle insurance premium

It is important for insurers to develop models that accurately forecast premium for car insurance

These model estimates can be used to create premium tables that can assist to set the price of the premiums, depending on the expected treatment costs.

Dependent variable

- The variable we wish to explain or predict

- Usually denoted by Y

- Dependent Variable = Response Variable = Target Variable

- Here 'Insurance Premium' is our target variable

Independent variable

- The variables used to explain the dependent variable

- Usually denoted by X

- Independent Variable = Predictor Variable

- In our example, Age, Mileage and Condition of the car are the independent variables

Variables that may contribute to insurance premium

Manufacturer
(Independent Variable)

Mileage (Independent Variable)

Engine capacity
(Independent Variable)

Age
(Independent Variable)

Condition
(Independent Variable)

Insurance Premium
(Target Variable)
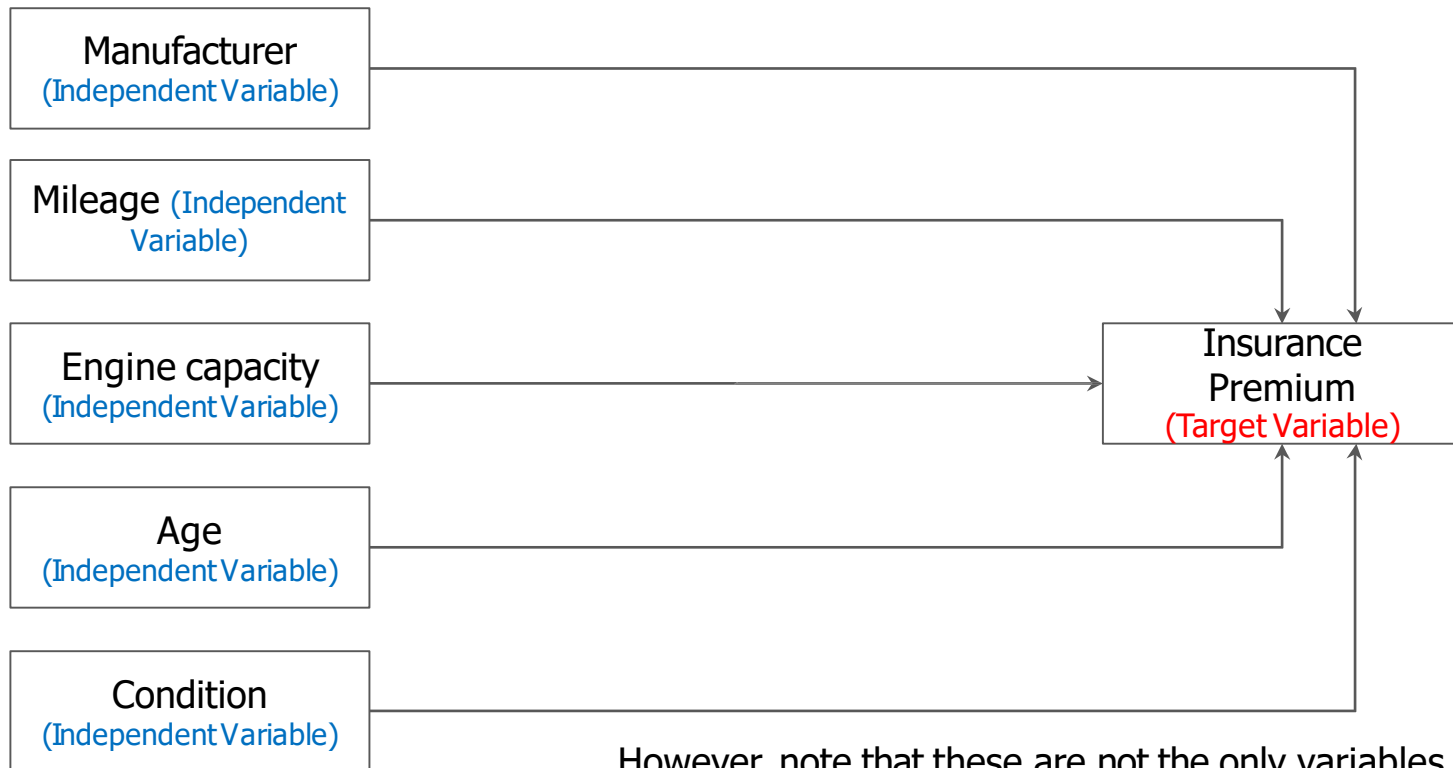
However, note that these are not the only variables considered. You may have some more in mind.

Visiting Basics

Covariance

Covariance is a measure of how changes in one variable are associated with changes in

another variable.

$$COV(X,Y) = \frac{\sum_{i=1}^{n}\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{n-1}$$

Xi = values taken by variable X, $\forall$ X $\in$ [1, n]

Yi = values taken by variable Y, $\forall$ Y $\in$ [1, n]

$\overline{X}$ = mean of Xi

$\overline{Y}$ = mean of Yi

Pearson's correlation coefficient

Correlation is a measure for linear association between two numeric variables.

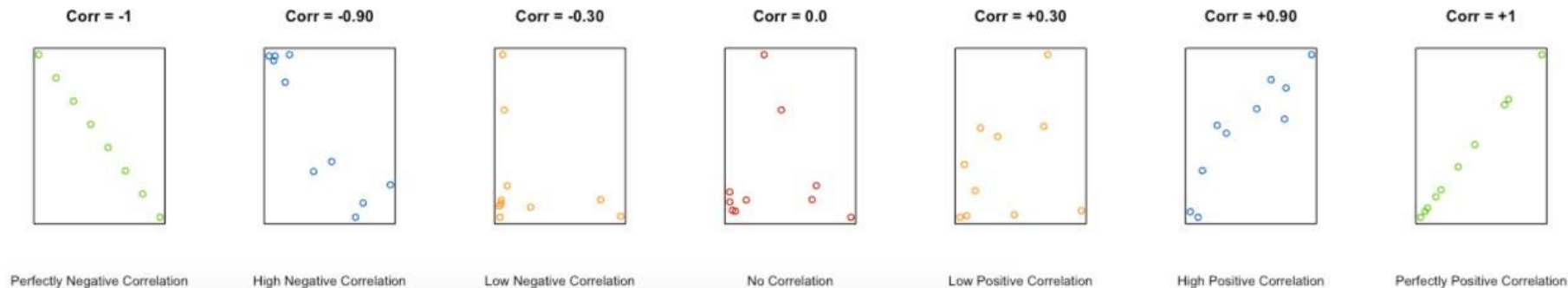$$R = \frac{Cov(x,y)}{\sigma_x \cdot \sigma_y}$$

Cov(x, y) = covariance of variables x and y

σ_x = standard deviation of x

σ_y = standard deviation of y

Value of correlation

Correlation is a scaled version of covariance that takes on values in [−1,1] with a correlation of ±1 indicating perfect linear association and 0 indicating no linear relationship.
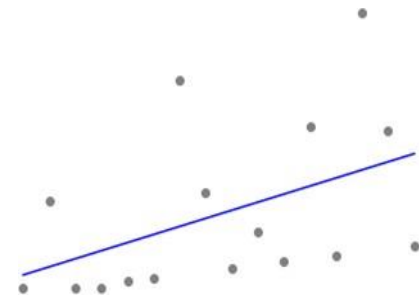
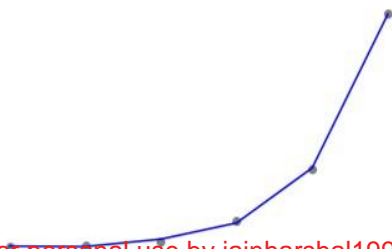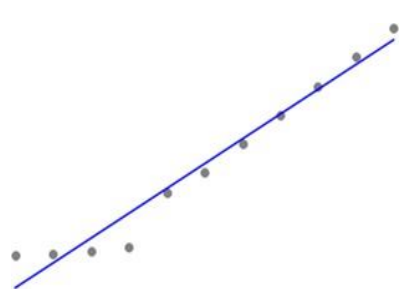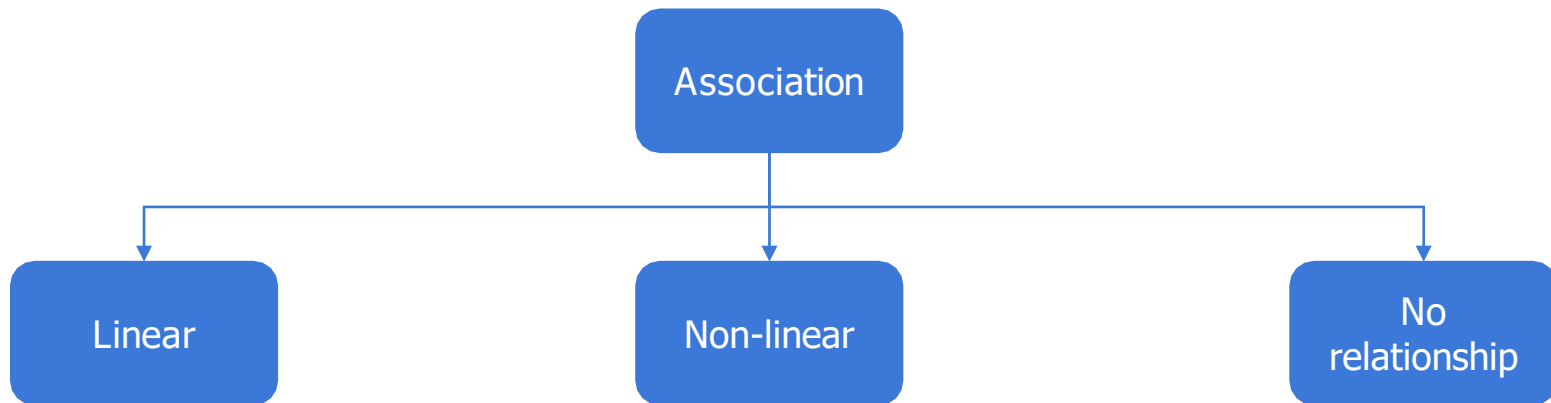| Corr = -1	Corr = -0.90	Corr = -0.30	Corr = 0.0	Corr = +0.30	Corr = +0.90	Corr = +1
Perfectly Negative Correlation	High Negative Correlation	Low Negative Correlation	No Correlation	Low Positive Correlation	High Positive Correlation	Perfectly Positive Correlation

Regression Analysis

What is regression analysis?

- Regression analysis allows us to examine which independent variables have an

  impact on the dependent variable

- Regression analysis investigates and **models the relationship between variables**

- Determine which independent variables can be ignored, which ones are most

  important and how they influence each other

- We shall first see simple linear regression and then multiple linear regression

Types of associations

```
                    ┌─────────────┐
                    │ Association │
                    └──────┬──────┘
          ┌────────────────┼────────────────┐
    ┌──────────┐     ┌────────────┐   ┌──────────────┐
    │  Linear  │     │ Non-linear │   │      No      │
    │          │     │            │   │ relationship │
    └──────────┘     └────────────┘   └──────────────┘
```

Simple linear regression

A simple linear regression model (also called **bivariate regression**) has one independent

variable X that has a linear relationship with the dependent variable Y

$$y = \beta_0 + \beta_1 x + \varepsilon$$

β_0 and β_1 are the parameters of the linear regression model.

Variable that contributes to insurance premium

Let us consider impact of a single variable for now.

```
┌─────────────────┐                    ┌─────────────────┐
│     Mileage     │                    │    Insurance    │
│  (Independent   │ ─────────────────► │     Premium     │
│    Variable)    │                    │ (Target Variable)│
└─────────────────┘                    └─────────────────┘
```

We say, that only mileage decides what the insurance premium should be.

Data

Let us consider the following data.

Mileage	Premium (in dollars)
15	392.5
14	46.2
17	15.7
7	422.2
10	119.4
7	170.9
20	56.9
21	77.5
18	214
11	65.3
7.9	250
8.6	220
12.3	217.5
17.1	140.88
19.4	97.25

Linear regression line

$$y = \beta_0 + \beta_1 x + \varepsilon$$

y = set of values taken by dependent variable Y

x = set of values taken by independent variable X

β_0 = y intercept

β_1 = slope

ε = random error component

Linear regression line using example

In context with our example,

$$\text{Premium} = \beta_0 + \beta_1 \text{ Mileage} + \varepsilon$$

y = set of values taken by dependent variable, Premium

x = set of values taken by independent variable, Mileage

β_0 = premium value where the best fit line cuts the Y - axis (Premium)

β_1 = beta coefficient for Mileage

ε = random error component

Mileage	Premium (in dollars)
15	392.5
14	46.2
17	15.7
7	422.2
10	119.4
7	170.9
20	56.9
21	77.5
18	214
11	65.3
7.9	250
8.6	220
12.3	217.5
17.1	140.88
19.4	97.25

What is the error term?

In context with our example,

$$Premium = \beta_0 + \beta_1\, Mileage + \varepsilon$$

y = set of values taken by dependent variable, Premium

x = set of values taken by independent variable, Mileage

β_0 = premium value where the best fit line cuts the Y - axis (Premium)

β_1 = beta coefficient for Mileage

ε = random error component

- Error term also called residual represents the distance of the observed value from the value predicted by regression line

- In our example,

  Error term = Actual Premium - Predicted Premium

  for each observation

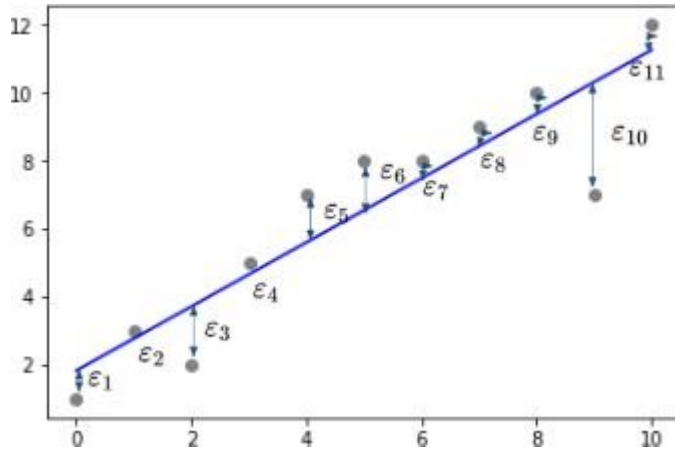# Calculating the error term
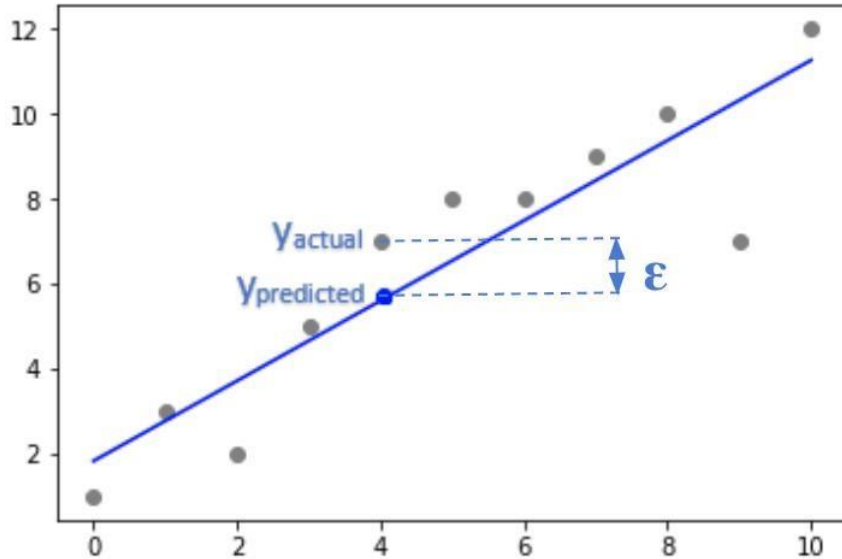
Equation of regression line is given by,

$$y = \beta_0 + \beta_1 x + \varepsilon$$

$$\therefore \varepsilon = y - (\beta_0 + \beta_1 x)$$

$$\therefore \varepsilon = y_{actual} - y_{predicted}$$

Error calculation

We have an error term for every observation in the data.

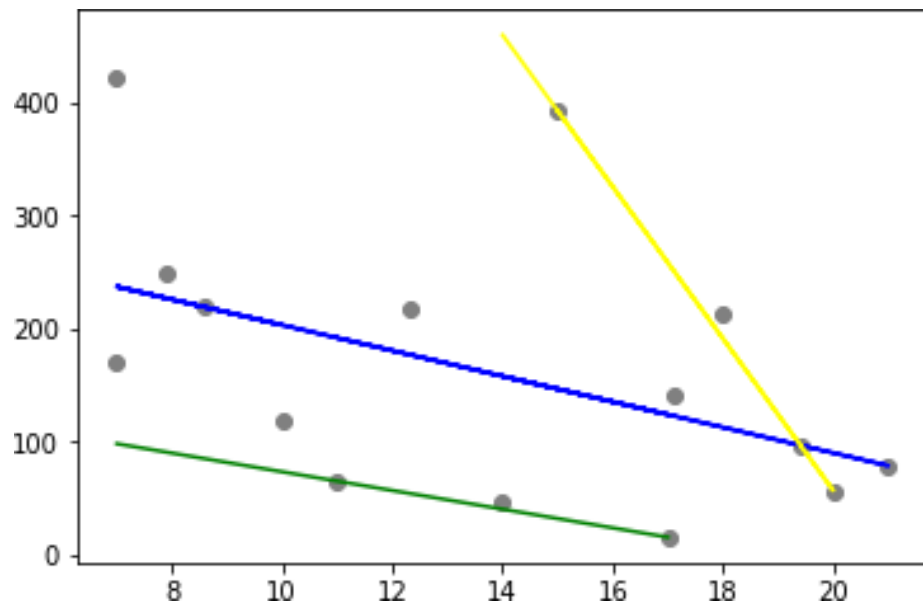**We have**

$$\varepsilon_i = y_{actual} - y_{predicted}$$

Squared error :

$$\varepsilon_i^2 = (y_{actual} - y_{predicted})^2$$

Sum of squared errors = $\sum \varepsilon_i^2$

Ordinary Least Squares Method

Which line best fits our data?

- The regression line which best explains the trend in the data is the best fit line

- It may pass through all of the points, some of the points or none of the points

How to obtain the best fit line?

- The ordinary least square method is used to find the best fit line for given data

- This method aims at minimizing the sum of squares of the error terms, that is, it determines those values of **β_0 and β_1 at which the error terms are minimum**

$$min \sum_{i=1}^{n} (y_i - \beta_i x_i)^2$$

Simple linear regression model

- We have seen that the error term $\varepsilon = y - (\beta_0 + \beta_1 x)$

- The OLS method minimizes $E = \sum \varepsilon^2 = \sum (y - (\beta_0 + \beta_1 x))^2$

- To minimize the error we take partial derivatives with respect to β_0 and β_1 and equate them to zero

$$\delta E / \delta \beta_0 = 0$$

$$\delta E / \delta \beta_1 = 0$$

So we get two equations with two unknowns, β_0 and β_1

Simple linear regression model

- So we get:

$$\delta E/\delta \boldsymbol{\beta_0} = \sum 2 (\mathbf{y} - \boldsymbol{\beta_0} - \boldsymbol{\beta_1}\mathbf{x}) (-1) = 0$$

$$\delta E/\delta \boldsymbol{\beta}_1 = \sum 2 (\mathbf{y} - \boldsymbol{\beta_0} - \boldsymbol{\beta_1}\mathbf{x})) (-x_1) = 0$$

- Expanding these equations, we get $\boldsymbol{\beta_0}$ and $\boldsymbol{\beta_1}$ as:

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \qquad\qquad \beta_1 = \frac{Cov(X,Y)}{Var(X)}$$

Simple linear regression model

Based on the data and the formulae obtained, the **β** parameters are:

$β_0$ =327.0860 and $β_1$ = -11.6905.

Thus the model is

> Y = 327.0860 - 11.6905 X

That is,

> Premium = 327.0860 - 11.6905 Mileage

Mileage	Premium (in dollars)
15	392.5
14	46.2
17	15.7
7	422.2
10	119.4
7	170.9
20	56.9
21	77.5
18	214
11	65.3
7.9	250
8.6	220
12.3	217.5
17.1	140.88
19.4	97.25

Interpretation of β coefficients

- β_1 gives the amount of change in response variable per unit change in predictor variable

- β_0 is the y intercept which means when X=0, Y is β_0

- β's have an associated p value, which is used to assess its significance in prediction of response variable

- Depending on whether β's take a positive value k or - k the response variable increases or decreases respectively by k units for every one unit increment in a predictor variable, keeping all other predictor variables constant

Interpretation of B0 & B1

![g1 logo]

- Dependent Variable is **Premium** and the predictor variable is **Mileage** that is acting as an input for the model.

- β_0 **= 327.0860**: represents the premium of a car immediately after manufacture (i.e. Mileage = 0).

- β_1 **= - 11.6905**: the average decrease in the premium of the cars due to the mileage.

- In a nutshell, when the Mileage is 0 then the premium would be equal to **$ 327.0860.**

OLS Regression Results

Dep. Variable:	Premium	R-squared:	0.226
Model:	OLS	Adj. R-squared:	0.166
Method:	Least Squares	F-statistic:	3.789
Date:	Tue, 29 Dec 2020	Prob (F-statistic):	0.0735
Time:	13:23:38	Log-Likelihood:	-90.831
No. Observations:	15	AIC:	185.7
Df Residuals:	13	BIC:	187.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	327.0860	87.035	3.758	0.002	139.057	515.115
Mileage	-11.6905	6.006	-1.947	0.074	-24.665	1.284

Omnibus:	3.225	Durbin-Watson:	2.347
Prob(Omnibus):	0.199	Jarque-Bera (JB):	1.770
Skew:	0.841	Prob(JB):	0.413
Kurtosis:	3.057	Cond. No.	44.3

B0 & B1 – some additional points

- The correlation direction is based on the sign of β_1.

- **If** the sign is positive, then the correlation is positive. Here the **sign of Mileage Coefficient** is **negative**, hence the correlation is **negative.**

- Taking the Square root of R Squared, we get the Correlation Value

- *In a Nutshell, if the Mileage Increases, the Premium goes down and vice-versa. This is legit because year on year, the insurance premium goes down as the car becomes older.*

- The correlation can be calculated using R Squared value. The direction of correlation can be found by looking at the sign of

		coef	std err	t	P>|t|	[0.025	0.975]
Intercept		327.0860	87.035	3.758	0.002	139.057	515.115
Mileage		-11.6905	6.006	-1.947	0.074	-24.665	1.284

Dep. Variable:	Premium	R-squared:	0.226
Model:	OLS	Adj. R-squared:	0.166
Method:	Least Squares	F-statistic:	3.789
Date:	Tue, 29 Dec 2020	Prob (F-statistic):	0.0735
Time:	14:27:05	Log-Likelihood:	-90.831
No. Observations:	15	AIC:	185.7
Df Residuals:	13	BIC:	187.1
Df Model:	1		
Covariance Type:	nonrobust		

Omnibus:	3.225	Durbin-Watson:	2.347
Prob(Omnibus):	0.199	Jarque-Bera (JB):	1.770
Skew:	0.841	Prob(JB):	0.413
Kurtosis:	3.057	Cond. No.	44.3

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

```
# Calculating the Correlation from R Squared
np.sqrt(0.226)

0.4753945729601885

# Cross-checking the Correlation using Correlation Coefficent from Numpy
np.corrcoef(premium["Mileage"], premium["Premium"])
array([[ 1.        , -0.47507329],
       [              , ]])
```

How is the $y_{predicted}$ obtained?

Using the Equation yhat = -11.6905*x + 327.0860, we will predict the Premium using Mileage as Input.

```
# Predicting the Premium based on the Linear Regression Equation Obtained
premium["Predicted_Premium"] = -11.6905*premium["Mileage"]+327.0860
premium
```

	Mileage	Premium	Predicted_Premium
0	15.0	392.50	151.72850
1	14.0	46.20	163.41900
2	17.0	15.70	128.34750

Substitute the values for X in the model:

Premium (predicted) = 327.0860 – 11.6905*Mileage

E.g.: For mileage **(x) = 15**, the predicted premium, ($y_{predicted}$) is obtained as:

$y_{predicted}$ = 327.0860 - 11.6905 *15 = $ 151.72850

Here:

- β_0 = 327.0860 &
- β_1 = - 11.6905

Simple regression - best fit line
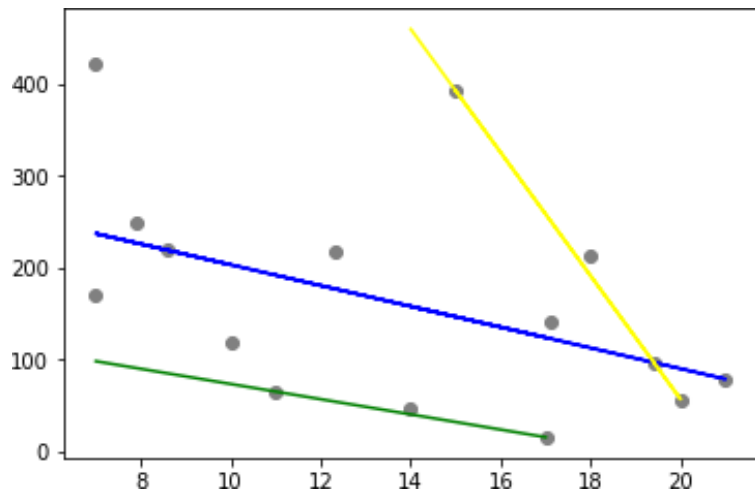
Three Different models are giving different errors

$\sum \varepsilon^2$	$\sum \varepsilon^2$	$\sum \varepsilon^2$
3.94×10^5	1.6×10^5 (Least Error)	26.8×10^5

Since the blue line has **least error** it is the **best fit line**

Measures of Variation

Sum of squares total

Sum Squared Total
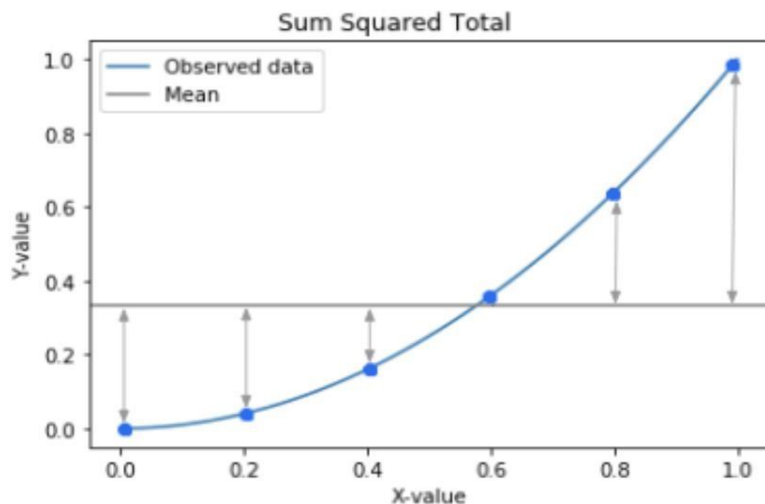
```
Sum of Squares Total

# Calculate Sum of Square Total => sum(yi-ybar)**2
yi  = premium["Premium"]

ybar = premium["Premium"].mean() # Remember we have to take ybar not xbar

print("The Sum of Squares Total is:",np.sum((yi - ybar)**2))

The Sum of Squares Total is: 206222.20604
```

- The sum of squares total (SST) is the sum of squared differences between the observed response variable and its mean

- It can be seen as the total variation of the response variable about its mean value

- SST is the measure of variability in the response variable without considering the effect of predictor variables

- Also known as Total Sum of Square (TSS)

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2$$

y_i = observed values of y $\quad$ $\bar{y}$ = mean value of variable y

Sum of Squares Regression

Sum of Squares Regression

$y = x - 0.15$
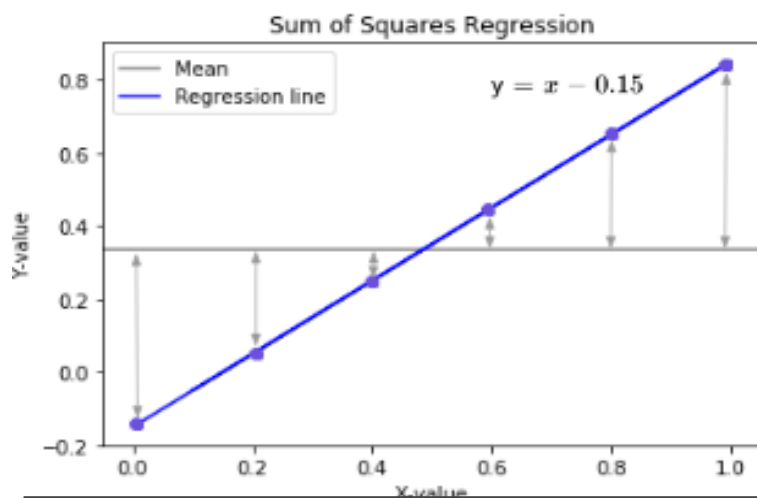
Sum of Squares Regression (SSR)

```python
# Calculate Sum of Square Regression => sum(yhat-ybar)**2

yhat  = premium["Predicted_Premium"]

ybar = premium["Premium"].mean() # Remember we have to take ybar not xbar

print("The Sum of Squares Regression is:",np.sum((yhat - ybar)**2))

The Sum of Squares Regression is: 46543.21820010749
```

- The sum of squares regression (SSR) is the sum of squared differences between the predicted value and the mean of the response variable

- SSR is the measure of variability in the response variable considering the effect of predictor variable . It is the explained variation

- It is the explained variation

- Also known as Regression Sum of Square (RSS)

$$SSR = \sum_{i=1}^{n} (\hat{y} - \bar{y})^2$$

Sum of squares of error
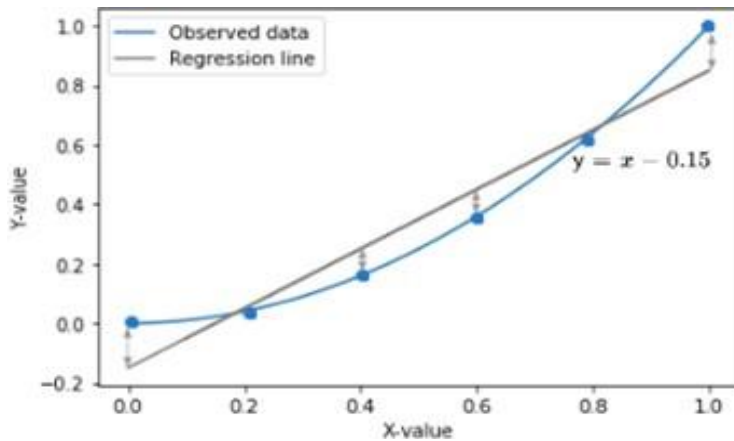
**Sum of Squares Error (SSE)**

```
# Calculate Sum of Square Error => sum(yi-yhat)**2 | Actual - Predicted

yi   = premium["Premium"]

yhat  = premium["Predicted_Premium"]

print("The Sum of Squares Error is:",np.sum((yi - yhat)**2))

The Sum of Squares Error is: 159678.9622455075
```

- The sum of squares of error (SSE) is the sum of squared differences between observed response variable and its predicted value

- SSE is the measure of variability in the response variable remaining after considering the effect of predictor variables

- It is the unexplained variation

- Also known as Error Sum of Square (ESS)

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y})^2$$

Variation in response variable

$$SST = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$
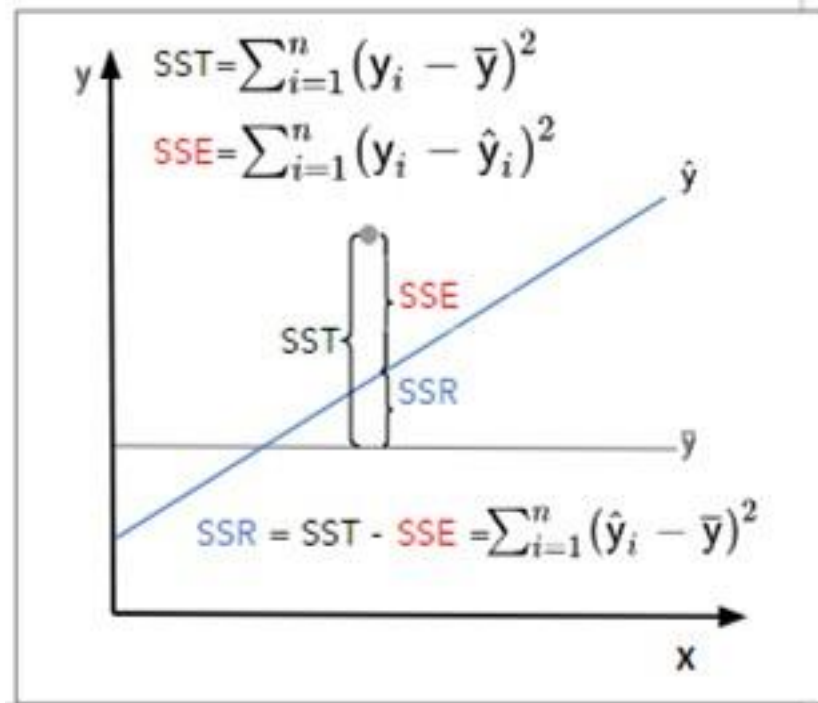
$$SSR = SST - SSE = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

y_i = observed values of y

$\hat{y}_i$ = predicted values of y

$\bar{y}$ = mean value of variable y

Total variation

Total variation   =   Explained variation + Unexplained variation

$$SST = SSR + SSE$$

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\hat{y} - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \hat{y})^2$$

Measure of unexplained variation

- Standard error of estimate is a measure of the unexplained variance

- Smaller value of standard error of estimate indicates a better model

$$Sxy = \sqrt{\frac{\sum(y_i - \hat{y}_i)^2}{n - k}}$$

n = sample size

k = number of parameter estimates (β_0, β_1)

Measure of explained variation

R^2 also called the coefficient of determination gives total percentage of variation in Y that is

explained by predictor variable.

$$R^2 = \frac{\text{Explained variation}}{\text{Total variation}} = \frac{\text{SSR}}{\text{SST}}$$

$$0 \leq R^2 \leq 1$$

$$R^2 = 1 - \frac{SSE}{SST}$$

Calculation of R Squared

- It is given by 1 - (SSE/SST)

```
# Calculation of R Squared

SSE = np.sum((yi - yhat)**2) # Sum of Squared Error

SST = np.sum((yi - ybar)**2) # Sum of Squared Total

# R Squared
print("RSquared: ", 1-(SSE)/SST)

RSquared:  0.2256946266274774
```

R-squared

- Since $0 \leq SSE \leq SST$, mathematically we have $0 \leq R^2 \leq 1$

- R^2 assumes that all the independent variables explain the variation in the dependent variable

- For simple linear regression, the squared correlation between the response variable Y and independent variable X is the R^2 value

- For our model, $R^2 = 0.226$. It implies **that 22.6% variation in premium amount is explained by the mileage of a car.**

Drawing Inference on R Squared

g1

- R Squared = 0.226
- Independent Variable - Mileage

```python
from statsmodels.formula.api import ols
import warnings
warnings.filterwarnings("ignore")

ols("Premium~Mileage", data = premium).fit().summary()
```

OLS Regression Results

Dep. Variable:	Premium	R-squared:	0.226
Model:	OLS	Adj. R-squared:	0.166
Method:	Least Squares	F-statistic:	3.789
Date:	Tue, 29 Dec 2020	Prob (F-statistic):	0.0735
Time:	14:27:05	Log-Likelihood:	-90.831
No. Observations:	15	AIC:	185.7
Df Residuals:	13	BIC:	187.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	327.0860	87.035	3.758	0.002	139.057	515.115
Mileage	-11.6905	6.006	-1.947	0.074	-24.665	1.284

Omnibus:	3.225	Durbin-Watson:	2.347
Prob(Omnibus):	0.199	Jarque-Bera (JB):	1.770
Skew:	0.841	Prob(JB):	0.413
Kurtosis:	3.057	Cond. No.	44.3

Drawing Inference on R Squared

- Since the R Squared value is quite low, it suggests that Mileage is not the correct variable that is contributing to the model. There may be some other predictors that have a significant contribution in explaining the model.

- We will perform a statistical test on β_1 to verify if mileage is really a contributor towards the model or not.

- Remember higher the R Squared, better the model is. However, we need to check **Adjusted R Squared** to be sure about the model performance.

```python
from statsmodels.formula.api import ols
import warnings
warnings.filterwarnings("ignore")

ols("Premium~Mileage", data = premium).fit().summary()
```

OLS Regression Results

Dep. Variable:	Premium	R-squared:	0.226
Model:	OLS	Adj. R-squared:	0.166
Method:	Least Squares	F-statistic:	3.789
Date:	Tue, 29 Dec 2020	Prob (F-statistic):	0.0735
Time:	14:27:05	Log-Likelihood:	-90.831
No. Observations:	15	AIC:	185.7
Df Residuals:	13	BIC:	187.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	327.0860	87.035	3.758	0.002	139.057	515.115
Mileage	-11.6905	6.006	-1.947	0.074	-24.665	1.284

Omnibus:	3.225	Durbin-Watson:	2.347
Prob(Omnibus):	0.199	Jarque-Bera (JB):	1.770
Skew:	0.841	Prob(JB):	0.413
Kurtosis:	3.057	Cond. No.	44.3

Inferences about Slope

Understanding role of T-Test in Regression

```python
from statsmodels.formula.api import ols
import warnings
warnings.filterwarnings("ignore")

ols("Premium~Mileage", data = premium).fit().summary()
```

- Thus, we want to know if the Population Slope is 0 or not.

- If the Population slope is 0, there is no relation between Mileage & Premium Paid. We perform a T-Test to identify the relationship

OLS Regression Results

Dep. Variable:	Premium	R-squared:	0.226
Model:	OLS	Adj. R-squared:	0.166
Method:	Least Squares	F-statistic:	3.789
Date:	Tue, 29 Dec 2020	Prob (F-statistic):	0.0735
Time:	14:27:05	Log-Likelihood:	-90.831
No. Observations:	15	AIC:	185.7
Df Residuals:	13	BIC:	187.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	327.0860	87.035	3.758	0.002	139.057	515.115
Mileage	-11.6905	6.006	-1.947	0.074	-24.665	1.284

Omnibus:	3.225	Durbin-Watson:	2.347
Prob(Omnibus):	0.199	Jarque-Bera (JB):	1.770
Skew:	0.841	Prob(JB):	0.413
Kurtosis:	3.057	Cond. No.	44.3

The t test for significance

- For β to be significant, $\beta > 0$.

$$H_0 : \beta = 0 \quad \text{against} \quad H_1 : \beta \neq 0$$

- It implies

$$H_0: \text{The parameter } \beta \text{ is not significant}$$

against $\quad H_1: \text{The parameter } \beta \text{ is significant}$

- Failing to reject H_0 implies that the parameter β is not significant

The t test for significance

- The test statistic is t given by

$$t = \frac{\hat{\beta}}{SE(\hat{\beta})}$$ where $\hat{\beta}$ is the estimated value of β.

- The t-statistic follows the $t_{(n-2)}$ distribution

- Decision Rule: Reject H_0 if $|t| > t_{(n-2), \alpha/2}$ or if the p-value is less than the α (level of significance)

The t test for slope

- For a existence of a linear relationship $\beta_1 > 0$, to test

$$H_0 : \beta_1 = 0 \quad \text{against} \quad H_1 : \beta_1 \neq 0$$

- It implies

H_0: There is no relationship between variables X and Y

against H_1: There is relationship between variables X and Y

- Failing to reject H_0 implies that there is no relationship between X and Y

T-Test of Slope in Regression

Defining the Null & Alternate Hypothesis

- H0: $\beta_1 = 0$

- H1: $\beta_1 \neq 0$

- If there is significant relationship between the Independent variable(Mileage) and the Dependent Variable(Premium), then the slope won't be 0.

- Test Used – We will use t-test to determine whether the slope of the regression line differs significantly from zero.

```python
from statsmodels.formula.api import ols
import warnings
warnings.filterwarnings("ignore")

ols("Premium~Mileage", data = premium).fit().summary()
```

OLS Regression Results

Dep. Variable:	Premium	R-squared:	0.226
Model:	OLS	Adj. R-squared:	0.166
Method:	Least Squares	F-statistic:	3.789
Date:	Tue, 29 Dec 2020	Prob (F-statistic):	0.0735
Time:	14:27:05	Log-Likelihood:	-90.831
No. Observations:	15	AIC:	185.7
Df Residuals:	13	BIC:	187.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	327.0860	87.035	3.758	0.002	139.057	515.115
Mileage	-11.6905	6.006	-1.947	0.074	-24.665	1.284

Omnibus:	3.225	Durbin-Watson:	2.347
Prob(Omnibus):	0.199	Jarque-Bera (JB):	1.770
Skew:	0.841	Prob(JB):	0.413
Kurtosis:	3.057	Cond. No.	44.3

T-Test of Slope in Regression

Thus, the value of test statistic is:

$$\text{Test statistic} = \frac{\beta_1 \ (-11.6908)}{\text{S.E.}(6.006)} = -1.9464$$

Standard Error is given by:

$$\text{S.E.} = \sqrt{\frac{\Sigma(y_i - \hat{y})^2}{n-2}} \div \sqrt{\Sigma(x_i - \bar{x})^2}$$

```
# Standard Error:  sqrt [ Σ(ŷi-yi)2 / (n - 2) ] / sqrt [ Σ(xi - xbar)2 ]
se = np.round(np.sqrt(np.sum((yi-yhat)**2)/(len(premium)-2))/np.sqrt(np.sum((xi - xbar)**2)),3)
print(se)
```

```
# Finding PValue on the basis of Test Statistic
print("P-Value:",np.round((1.0 - stats.t.cdf(abs(tstats), 13)) * 2.0,3))

6.006
T-Test Statistic: -1.9464701964701965
T-Test Statistic Rounded: -1.946
P-Value: 0.074
```

Conclusion: Since P-Value > 0.05, we fail to reject Ho meaning that there is no significant relationship between Mileage and Premium or in other words, there is not enough evidence to conclude that there is a linear relationship between Mileage and Premium

Note: Multiply by 2 as t-test is a 02-tail test in python. Refer to t-test in Statistics for more

The t test for intercept

- For a existence of a linear relationship $\beta_1 > 0$, to test

$$H_0 : \beta_0 = 0 \qquad \text{against} \qquad H_1 : \beta_0 \neq 0$$

- It implies

$$H_0: \text{The parameter } \beta_0 \text{ is not significant}$$

against $\quad H_1: \text{The parameter } \beta_0 \text{ is significant}$

- Failing to reject H_0 implies that the parameter β_0 is not significant

T-Test of Intercept in Regression

The intercept parameter is the mean of target variable at x = 0. In order to conduct the test, the Hypothesis are:

- Ho:$\beta_0 = 0$

- H1:$\beta_0 > 0$

- We are testing here if the Intercept is greater than 0. However, it can be both ways ($\beta_0 < 0$ & $\beta_0 > 0$)

- Test Used – We will use t-test to determine whether the intercept of the regression line differs significantly from zero.

```python
from statsmodels.formula.api import ols
import warnings
warnings.filterwarnings("ignore")

ols("Premium~Mileage", data = premium).fit().summary()
```

OLS Regression Results

Dep. Variable:	Premium	R-squared:	0.226
Model:	OLS	Adj. R-squared:	0.166
Method:	Least Squares	F-statistic:	3.789
Date:	Tue, 29 Dec 2020	Prob (F-statistic):	0.0735
Time:	14:27:05	Log-Likelihood:	-90.831
No. Observations:	15	AIC:	185.7
Df Residuals:	13	BIC:	187.1
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	327.0860	87.035	3.758	0.002	139.057	515.115
Mileage	-11.6905	6.006	-1.947	0.074	-24.665	1.284

Omnibus:	3.225	Durbin-Watson:	2.347
Prob(Omnibus):	0.199	Jarque-Bera (JB):	1.770
Skew:	0.841	Prob(JB):	0.413
Kurtosis:	3.057	Cond. No.	44.3

T-Test of Intercept in Regression

Lets calculate the test statistic and p-value here.

Since test statistic is given by:

$$t = \frac{\hat{\beta}}{SE(\hat{\beta})}$$

Thus, the value of test statistic is:

Test statistic = $\dfrac{\beta_0 \ (327.0860)}{S.E.(87.035)} = 3.758$
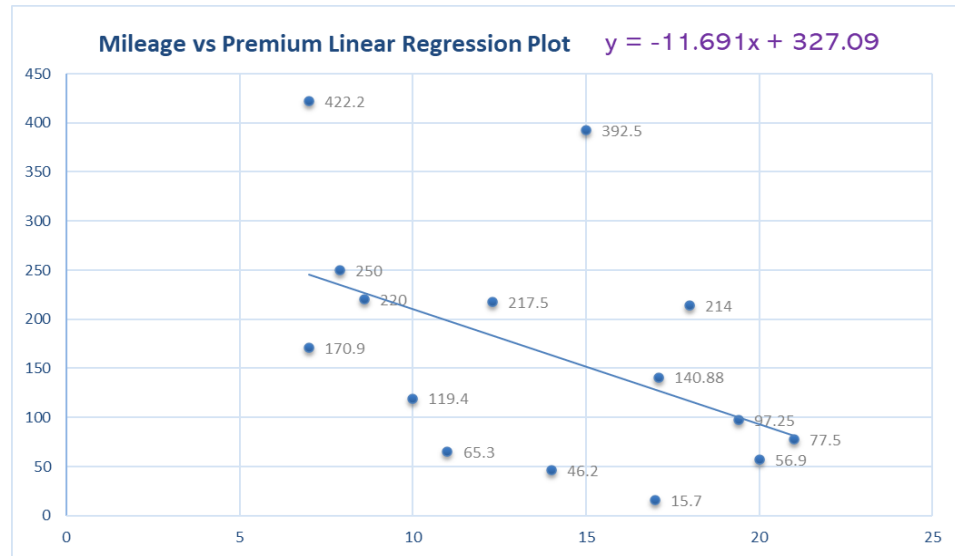
Mileage vs Premium Linear Regression Plot   y = -11.691x + 327.09


```
# Finding PValue on the basis of Test Statistic
print("P-Value:",np.round((1.0 - stats.t.cdf(abs(tstats), 13)) * 2.0,3))

T-Test Statistic: 3.758097317171253
T-Test Statistic Rounded: 3.758
P-Value: 0.002
```

Note: The P-Value is calculated assuming the alternate hypothesis is "two-tailed not-equal-to-0" hypothesis.

Conclusion: Since P-Value > 0.05, we fail to reject Ho meaning that the intercept is not 0. In simple words, mileage is not 0 in the sample selected.

Note: Multiply by 2 as t-test is a 02-tail test in python. Refer to t-test in Statistics for more

The interval estimation of β

- The interval estimate of a parameter gives the 100(1-α)% confidence interval

    (Say α = 0.05, 100(1-α)% = 95%)

- In other words, for an experiment conducted 100 times, the estimate would lie within the confidence interval 95 times. This would give the 95% confidence interval

Interval estimation for slope

- The test statistic for slope is

$$t_1 = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)} \qquad \text{where } t_1 \sim t_{(n-2)}$$

- The 100(1-α)% confidence interval for slope is given by

$$\beta_1 - t_{(n-2),\alpha/2} * \text{S.E.} \quad \& \quad \beta_1 + t_{(n-2),\alpha/2} * \text{S.E.}$$

β_1 is the slope of the Linear Regression Model & and n are the number of observations.

Note: Degrees of Freedom would be calculated using (n-2)

Interval estimation for intercept

- The test statistic for intercept is

$$t_0 = \frac{\hat{\beta}_0}{SE(\hat{\beta}_0)} \qquad \text{where } t_0 \sim t_{(n-2)}$$

- The 100(1-α)% confidence interval for intercept is given by

$$\beta_0 - t_{(n-2),\alpha/2} * \text{S.E.} \quad \& \quad \beta_0 + t_{(n-2),\alpha/2} * \text{S.E.}$$

β_0 is the intercept of the Linear Regression Model & and n are the number of observations.

Note: Degrees of Freedom would be calculated using (n-2)

Confidence intervals

- We have α = 0.05, thus α/2 = 0.025

- For the lower bound of CI, 0 + α/2 = 0.025

- For the upper bound of CI, 1 - α/2 = 0.975

Parameter	0.025	0.975
β_1	-24.665	1.284
β_0	139.057	515.115

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	327.0860	87.035	3.758	0.002	139.057	515.115
Mileage	-11.6905	6.006	-1.947	0.074	-24.665	1.284

```
# Confidence Interval Estimation - Slope

alpha_by_two = (0.05/2)
deg_freedom = len(premium) - 2

beta1 = -11.6905
se = 6.006
tcrit = np.abs(stats.t.ppf(alpha_by_two,deg_freedom) )

print("Confidence Interval 0.025: ", np.round(beta1-tcrit*se, 4))
print("Confidence Interval 0.975: ", np.round(beta1+tcrit*se,4))

Confidence Interval 0.025: -24.6657
Confidence Interval 0.975: 1.2847
```

$$\beta_1 \text{ -/+ } t_{(n-2),\alpha/2} * \text{S.E.}$$

Note: stats.t.ppf takes two parameters – alpha & df

ANOVA for regression

- The hypothesis for ANOVA in regression framework are

$$H_0: \beta_1 = 0 \quad \text{against} \quad H_1: \beta_1 \neq 0$$

- It implies

H_0: The regression model is not significant

against H_1: The regression model is significant

ANOVA table for bivariate regression

Source of variation	Sum of Squares	Degrees of Freedom	Mean Sum of Squares	F ratio
Regression	RSS	$k = 1$	MRSS = RSS/1	
Residual	ESS	$n - k - 1 = n - 1 - 1 = n - 2$	MESS = ESS/(n-2)	$F_0 = $ MRSS/MESS
Total	TSS	$n - 1$	-	

- Decision rule: Reject H_0, if $F_0 > F_{(1, n-2), \alpha}$ or if the p-value is less than the α (level of significance)

- Failure to reject H_0 implies that the model is not significant

The t test for correlation coefficient

- For a existence of a correlation ρ, i.e. to test

$$H_0 : \rho = 0 \quad \text{against} \quad H_1 : \rho \neq 0$$

- It implies

$$H_0: \text{There is no correlation}$$

against $H_1: \text{The correlation is significant}$

- Failing to reject H0 implies that there is no significant correlation

The t test for correlation coefficient

- The test statistic is t_{xy} given by

$$t_{xy} = \frac{\rho\sqrt{n-2}}{\sqrt{1-\rho^2}}$$

ρ: correlation coefficient
n: number of observations

- The t-statistic follows the $t_{(n-2)}$ distribution

- Decision Rule: Reject H_0 if $|t_{xy}| > t_{(n-2),\alpha/2}$ or the p-value is less than the α (level of significance)

Multiple Linear Regression

Multiple linear regression

Multiple regression model is used when multiple predictor variables [X_1, X_2, X_3, …,

X_n] are used to predict the response variable Y

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_n x_n + \varepsilon$$

β_0, β_1, β_2, β_3, …, β_n are the parameters of the linear regression model with n

independent variables

Variable that contributes to Insurance Premium

Let us consider impact of a multiple variables on the Insurance Premium
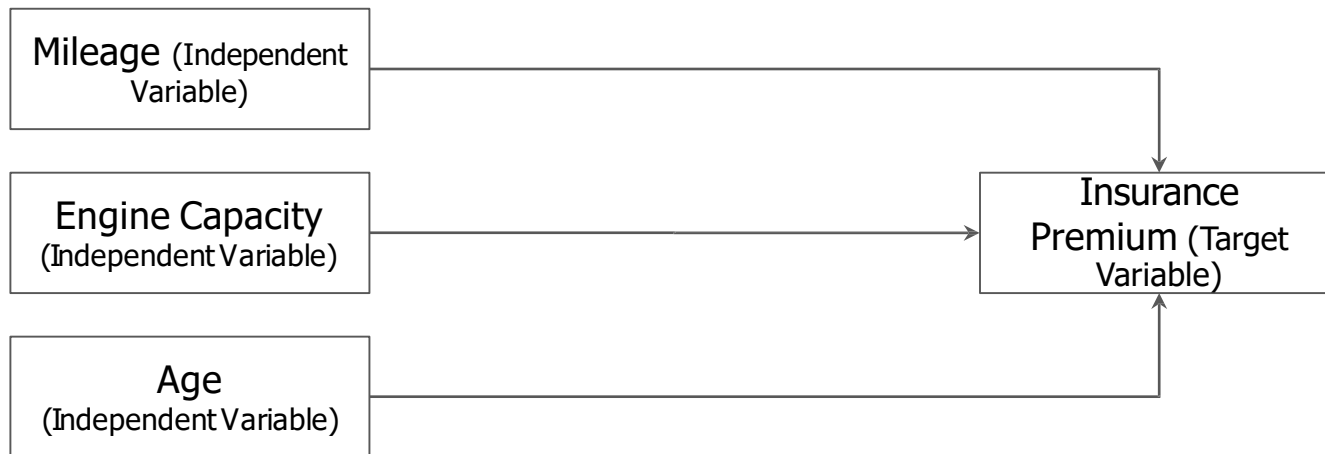
We say that only Mileage, Engine Capacity and Age decide what the insurance premium should be.

Data

Let us consider the following data.

Mileage	Engine_Capacity	Age	Premium (in dollars)
15	1.8	2	392.5
14	1.2	10	46.2
17	1.2	8	15.7
7	1.8	3	422.2
10	1.6	4	119.4
7	1.4	3	170.9
20	1.2	7	56.9
21	1.6	6	77.5
18	1.2	2	214
11	1.6	5	65.3
7.9	1.4	3	250
8.6	1.6	3	220
12.3	1.2	2	217.5
17.1	1.6	1	140.88
19.4	1.2	6	97.25

Multiple Linear Regression Line Equation

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots + \beta_n x_n + \varepsilon$$

y = set of values taken by dependent variable Y

x_i = set of values taken by independent variable X_i , $i \in [1,n]$

β_0 = y intercept

β_i = beta coefficient for the ith independent variable X_i, $i \in [1,n]$

ε = random error component

Linear regression for our example

$$Premium = \beta_0 + \beta_1 \, Mileage + \beta_2 \, Engine\_Capacity + \beta_3 Age + \varepsilon$$

	Description
Premium	Set of values taken by the variable Premium
β_0	Premium value where the best fit line cuts the Y-axis (Premium)
β_1	Regression coefficient of variable Mileage
Mileage	Set of values taken by the variable Mileage
β_2	Regression coefficient of variable Engine_Capacity
Engine_Capacity	Set of values taken by the variable Engine_Capacity
β_3	Regression coefficient of variable Age
Age	Set of values taken by the variable Age
ε	Error component

Multiple linear regression model

Based on the data and the formulae obtained, the β parameters a

$\beta_0 = 138.398$, $\beta_1 = -4.876$,

$\beta_2 = 137.633$ and $\beta_3 = -23.718$.

Mileage	Engine_Capacity	Age	Premium (in dollars)
15	1.8	5	392.5
14	1.2	5	46.2
17	1.2	5	15.7
7	1.8	10	422.2
10	1.6	4	119.4
7	1.4	5	170.9
20	1.2	3	56.9
21	1.6	4	77.5
18	1.2	4	214
11	1.6	5	65.3
7.9	1.4	3	250
8.6	1.6	5	220
12.3	1.2	2	217.5
17.1	1.6	6	140.88
19.4	1.2	2	97.25

Thus the model is

$$Y = 138.398 - 4.876\, x_1 + 137.633\, x_2 - 23.71$$

That is,

Premium = 138.398 - 4.876*Mileage + 137.633*Engine_Capacity - 23.718*Age

Interpreting the β coefficients

In context with our example,

- $\beta_0 = $ 138.398: the value of premium when the mileage, engine capacity, and age are all equal to 0 (which is absurd)

- $\beta_1 = -$ 4.876: is the average decrease in the premium of cars due to unit increase in mileage, all else held constant.

- $\beta_2 = $ 137.633: the average increase in the premium of the cars due to engine capacity, all else held constant.

- $\beta_3 = -$ 23.718: the average decrease in the premium of the cars due to age, all else held constant.

```
# Mulitple Linear Model
ols("Premium~Mileage+Engine+Age", data = premium).fit().summary()
```

OLS Regression Results

Dep. Variable:	Premium	R-squared:	0.591
Model:	OLS	Adj. R-squared:	0.480
Method:	Least Squares	F-statistic:	5.309
Date:	Fri, 01 Jan 2021	Prob (F-statistic):	0.0166
Time:	22:01:04	Log-Likelihood:	-86.035
No. Observations:	15	AIC:	180.1
Df Residuals:	11	BIC:	182.9
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	138.3977	218.865	0.632	0.540	-343.321	620.116
Mileage	-4.8756	5.252	-0.928	0.373	-16.436	6.684
Engine	137.6328	118.936	1.157	0.272	-124.143	399.409
Age	-23.7176	10.444	-2.271	0.044	-46.704	-0.731

Omnibus:	0.101	Durbin-Watson:	2.374
Prob(Omnibus):	0.951	Jarque-Bera (JB):	0.285
Skew:	0.143	Prob(JB):	0.867
Kurtosis:	2.388	Cond. No.	167.

Revisiting R-squared

R² also called the coefficient of determination gives total percentage of variation in Y that is

explained by predictor variable.

$$R^2 = \frac{\text{Explained variation}}{\text{Total variation}} = \frac{\text{SSR}}{\text{SST}}$$

$$R^2 = 1 - \frac{SSE}{SST}$$

```
# Mulitple Linear Model
ols("Premium~Mileage+Engine+Age", data = premium).fit().summary()
```

OLS Regression Results

Dep. Variable:	Premium	R-squared:	0.591
Model:	OLS	Adj. R-squared:	0.480
Method:	Least Squares	F-statistic:	5.309
Date:	Fri, 01 Jan 2021	Prob (F-statistic):	0.0166
Time:	22:01:04	Log-Likelihood:	-86.035
No. Observations:	15	AIC:	180.1
Df Residuals:	11	BIC:	182.9
Df Model:	3		
Covariance Type:	nonrobust		

$$0 \leq R^2 \leq 1$$

Demerits of R-squared

- The value of R^2 increases as new numeric predictors are added to the model, it may appear that it is a better model, which can be misleading

- Also, if the model has too many variables, the model is feared to be overfitted. Overfitted data generally has a high R^2 value.

Overfitting

- *Overfitting is a concept that is used in Machine Learning and it suggests that the model is not only capturing the data but the noise too. Hence, it will do very good on training data (because it is learning the exact patterns from it) but it will perform poorly on test set (because test set will not exhibit those patterns which training data did. In other words, the model has not generalised well on Unseen Data.*

Adjusted R-squared

Adjusted R^2 gives the percentage of variation explained by independent variables that actually affect the dependent variable

$$R^2_{adj} = 1 - \frac{(1-R^2)(n-1)}{n-k-1}$$

R^2 = R squared value for model

n = sample size

k = number of features excluding Dep Var.

```
# Mulitple Linear Model
ols("Premium~Mileage+Engine+Age", data = premium).fit().summary()
```

OLS Regression Results

Dep. Variable:	Premium	R-squared:	0.591
Model:	OLS	Adj. R-squared:	0.480
Method:	Least Squares	F-statistic:	5.309
Date:	Fri, 01 Jan 2021	Prob (F-statistic):	0.0166
Time:	22:01:04	Log-Likelihood:	-86.035
No. Observations:	15	AIC:	180.1
Df Residuals:	11	BIC:	182.9
Df Model:	3		
Covariance Type:	nonrobust		

Adjusted R Squared Value

```
# Calculating Adjusted R Squared
rsq = 0.591
n = premium.shape[0]
k = premium.shape[1]-1 # Need Independent Variables i.e. exclude Premium thus n-1

print("Adj R Squared:",np.round(1-((1-rsq)*(n-1))/(n-k-1),2))

Adj R Squared: 0.48
```

Adjusted R-squared

- $R^2_{adj} \le R^2$ (always)

- As the number of independent variables in the model increase, the adjusted R^2 will decrease unless the model significantly increases the R^2

- So to know whether addition of a variable explains the variation of the response variable, compare the R^2_{adj} values along with R^2

$$R^2_{adj} = 1 - \frac{(1-R^2)(n-1)}{n-k-1}$$

As k (no. of independent variables) increases, value of (n-k-1) decreases

R^2 vs. Adjusted R^2

- The value of R Squared never decreases. If we add new independent variables, then the value of R Squared increases. It cannot show the effect of adding a bad or insignificant variable

- As compared to the R-Squared value, Adj. R-Squared has an ability to decrease whenever a bad or an insignificant variables are added to the model. Thus we get an accurate evaluation.

Which is Preferred: R^2 vs. Adjusted R^2

- Since R Squared will always increase as we add variables/features in the dataset, this will create false impression of model performance getting improved. On the contrary the feature might not be adding any value to the model.

- On the other hand, Adjusted R Squared will only increase, if the new feature has really added value to the model and thereby improving the model as a whole. Thus, Adj R Squared is always preferred over R Squared.

ANOVA for regression with 'k' predictors

- The hypothesis for ANOVA in regression framework are

$$H_0: \beta_1 = \beta_2 = \beta_3 = 0 \quad \text{against} \quad H_1: \text{At least one } \beta_k \neq 0 \quad (k = 1,2,3)$$

- It implies

H_0: the regression model is not significant

against H_1: the regression model is significant

ANOVA table for regression with 'k' predictors

Source of variation	Sum of Squares	Degrees of Freedom	Mean Sum of Squares	F ratio
Regression	RSS	k	MRSS = RSS/1	F_0 = MRSS/MESS
Residual	ESS	n - k - 1	MESS = ESS/(n-k-1)	
Total	TSS	n-1	-	

- Decision rule: Reject H_0, if $F_0 > F_{(k,n-k-1),\alpha}$ or if the p-values is less than the α (level of significance)

- Failure to reject H_0 implies that the model is not significant

Model Performance Evaluation

Mean absolute error (MAE)

MAE is robust to outliers

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\mathbf{y}_i - \hat{\mathbf{y}}_i|$$

Sample
size

Observed
value

Predicte
d
value

Mean Absolute Error & Mean Squared Error

- Mean Absolute Error is the AVG(abs(Actual Value - Predicted Value))
- Mean Squared Error is the AVG[(Actual Value - Predicted Value)2]

```
# Predicted Premium is calculated using 327.086-11.6905 * Mileage
premium.head()
```

	Mileage	Premium	Predicted_Premium
0	15.0	392.5	151.7285
1	14.0	46.2	163.4190
2	17.0	15.7	128.3475
3	7.0	422.2	245.2525
4	10.0	119.4	210.1810

```
# Mean Absolute Error - (Actual - Predicted)

from sklearn.metrics import mean_absolute_error, mean_squared_error
print("Mean Absolute Error of Mileage & Premium: ", mean_absolute_error(premium["Premium"],
                                                        premium["Predicted_Premium"]))
```

```
Mean Absolute Error of Mileage & Premium:  77.09864333333333
```

Mean square error (MSE)

Squaring of error terms handles the negative values of error and also emphasizes larger errors.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(\mathbf{y}_i - \hat{\mathbf{y}}_i \right)^2$$

Sample size

Observed value

Predicted value

Mean Squared Error

Remember that the Mean Squared Error is the AVG[(Actual Value - Predicted Value)2]

```
# Predicted Premium is calculated using 327.086-11.6905 * Mileage
premium.head()
```

	Mileage	Premium	Predicted_Premium
0	15.0	392.5	151.7285
1	14.0	46.2	163.4190
2	17.0	15.7	128.3475
3	7.0	422.2	245.2525
4	10.0	119.4	210.1810

```
# Mean Squared Error - (Actual - Predicted)
print("Mean Squared Error of Mileage & Premium: ", mean_squared_error(premium["Premium"],
                                                                       premium["Predicted_Premium"]))

Mean Squared Error of Mileage & Premium:  10645.264149700499
```

Root mean square error (RMSE)

Lower the value of RMSE, better is the fit of regression line

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}$$

Sample size

Observed value

Predicted value

Mean Squared Error

Remember that the Mean Squared Error is the AVG[(Actual Value - Predicted Value)2]

```
# Predicted Premium is calculated using 327.086-11.6905 * Mileage
premium.head()
```

	Mileage	Premium	Predicted_Premium
0	15.0	392.5	151.7285
1	14.0	46.2	163.4190
2	17.0	15.7	128.3475
3	7.0	422.2	245.2525
4	10.0	119.4	210.1810

```
# Root Mean Squared Error - (Actual - Predicted)
print("Root Mean Squared Error of Mileage & Premium: ", np.sqrt(mean_squared_error(premium["Premium"],
                                                                                    premium["Predicted_Premium"])))
```

Root Mean Squared Error of Mileage & Premium: 103.17588938167918

Mean absolute percentage error (MAPE)

MAPE is robust to outliers

Observed value

Predicted value

$$MAPE = 100\left(\frac{1}{n}\right)\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|$$

Multiplication with 100 gives percentage

Sample size

Mean Absolute Percentage Error

- Most Common Methods to Forecast Error and works best when there are no outliers.
- Not Included in Python due to Divison by 0 error is possible.
- We will create a MAPE Function

```python
# define a function to calculate MAPE
# pass the actual and predicted values as input to the function
# return the calculated MAPE
def mape(actual, predicted):
    return (np.mean(np.abs((actual - predicted) / actual)) * 100)

# Mean Absolute Percentage Error (Actual-Predicted)/Actual %age

print("Mean Absolute Percentage Error of Mileage & Premium: ", mape(premium["Premium"],
                                                                    premium["Predicted_Premium"]))
```

Mean Absolute Percentage Error of Mileage & Premium: 103.35540343460177

Assumptions of Linear Regression

Assumptions of linear regression

- The dependent variable must be numeric

- Linear relationship between dependent and independent variables

- Predictors must not show multicollinearity

- Independence of observations should exist (Absence of Autocorrelation)

- The error terms should be homoscedastic

- The error terms must follow normal distribution

Assumptions of linear regression

- Numeric dependent variable
- Absence of Multicollinearity

Assumption to be checked before building a model

Build a model

- Linear relationship
- Absence of Autocorrelation
- Error terms should be homoscedastic
- Error terms must follow $N(0,\sigma^2)$

Assumption to be checked after building a model

Tests before model building

- The dependent variable must be numeric

- Predictors must not show multicollinearity

Is the dependent variable numeric?

- Regression Analysis requires the target variable to be numeric in nature

- For example: returns, sales of a product, yield of a crop, risk in financial services

- In context with our example, we see that Premium is numeric

Mileage	Premium (in dollars)
15	392.5
14	46.2
17	15.7
7	422.2
10	119.4
7	170.9
20	56.9
21	77.5
18	214
11	65.3
7.9	250
8.6	220
12.3	217.5
17.1	140.88
19.4	97.25

If our target variable is: 0, 1, 1, 0, 0, where 0 indicates presence of a disease and 1 indicates absence.

Is it appropriate to use regression to find the whether the person has a disease?

Question:

If our target variable is: 0, 1, 1, 0, 0, where 0 indicates presence of a disease and 1 indicates absence. Is it appropriate to use regression to find the whether the person has a disease?

Answer:

No. Because the target variable is a categorical variable. Thus, it is a classification problem.

Tests before model building

- The dependent variable must be numeric

- Predictors must not show multicollinearity

What is multicollinearity?

- Multicollinearity arises when the **independent variables have high correlation** among each other

- Multicollinearity may be introduced if there exists empirical relationship among variables such as income = expenditure + saving

- In presence of the multicollinearity, the best fit line obtained from OLS method is no more "best"

- Also, the confidence interval obtained for β's is wider since the **SE(β) becomes large**

Multicollinearity detection

- Determinant of correlation matrix

- Condition Number (CN)

Is there multicollinearity present?

- Correlation matrix

- Variance Inflation Number (VIF)

Which variables are involved in multicollinearity?

Is there multicollinearity?

Determinant of the correlation matrix:

Let D be the determinant of correlation matrix. Then $0 < D < 1$

D=0	High multicollinearity
D=1	No multicollinearity

Condition Number

(CN):

CN > 1000	Severe multicollinearity
100 < CN < 1000	Moderate multicollinearity
100 < CN	No multicollinearity

Which variables are involved in multicollinearity?

Correlation matrix:

If the off-diagonal values tend to ± 1 then it indicates high correlation between the variable pair. However this inspection is not enough

```
# Heatmap
sns.heatmap(premium.corr(), annot = True, annot_kws = {"size": 11}, cmap = "YlGnBu")
plt.show()
```
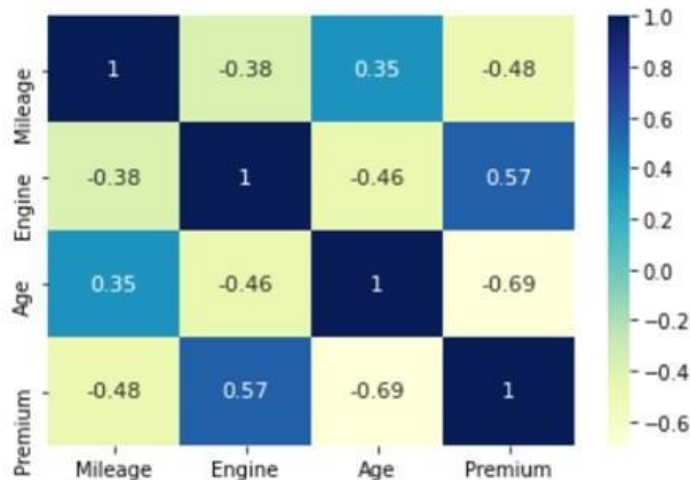
Condition No – Checking Multicollinearity

Condition Number (CN)

- A condition number(sometimes called a condition index) shows the degree of multicollinearity in a regression design matrix.

- It is an alternative to other methods like variance inflation factors.

CN > 1000	Severe multicollinearity
100 < CN < 1000	Moderate multicollinearity
100 < CN	No multicollinearity

Note: CN will tell us whether Multicollinearity exists or not. However, in order to find out which columns are suffering from Multicollinearity, we need to apply VIF.

```
# Mulitple Linear Model
ols("Premium~Mileage+Engine+Age", data = premium).fit().summary()
```

OLS Regression Results

Dep. Variable:	Premium	R-squared:	0.591
Model:	OLS	Adj. R-squared:	0.480
Method:	Least Squares	F-statistic:	5.309
Date:	Fri, 01 Jan 2021	Prob (F-statistic):	0.0166
Time:	22:01:04	Log-Likelihood:	-86.035
No. Observations:	15	AIC:	180.1
Df Residuals:	11	BIC:	182.9
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	138.3977	218.865	0.632	0.540	-343.321	620.116
Mileage	-4.8756	5.252	-0.928	0.373	-16.436	6.684
Engine	137.6328	118.936	1.157	0.272	-124.143	399.409
Age	-23.7176	10.444	-2.271	0.044	-46.704	-0.731

Omnibus:	0.101	Durbin-Watson:	2.374
Prob(Omnibus):	0.951	Jarque-Bera (JB):	0.285
Skew:	0.143	Prob(JB):	0.867
Kurtosis:	2.398	Cond. No.	167.

Which variables are involved in multicollinearity?

Variance Inflation Factor (VIF):

$$VIF = \frac{1}{1-R^2}$$

Where R^2 is obtained by regressing a predictor variable over all the other predictors in the model

Value	Interpretation
VIF > 5	High correlation
5 > VIF >1	Moderate correlation
VIF = 1	No correlation

Analyzing VIF on Premium Data

Value	Interpretation
VIF > 5	High correlation
5 > VIF >1	Moderate correlation
VIF = 1	No correlation

Inference of VIF:

- The VIF test shows that the Mileage and Engine are highly correlated.
- Only **Age** is the variable that is **contributing significantly** to the model

Difference between VIF & Condition Number (CN)

CN tells us whether the data is suffering from Multicollinearity or not whereas VIF will show us what all features are multicollinear in Nature

Checking MultiCollinearity - VIF

- Variance Inflation Factor (VIF) is used to find multicollinearity amongst predictor variables.
- VIF estimates how much the Regression Coefficients are inflated(increase) due to Multicollinearity.
- It is calculated by taking a predictor and regressing a model against the remaining predictors.
- This gives R Squared Value which is plugged in VIF formula to return VIF model for each predictor.
- Rule of Thumb: 1 = No Correlation, 1 to 5 - Moderately Correlated, >5 - High Correlation

```python
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Drop Dependent Variable a.k.a Target Variable
x = premium.drop("Premium", axis = 1)

# create an empty dataframe to store the VIF for each variable
vif = pd.DataFrame()

# calculate VIF using list comprehension
# use for loop to access each variable
# calculate VIF for each variable and create a column 'VIF_Factor' to store the values
vif["VIF_Factor"] = [variance_inflation_factor(x.values, i) for i in range(x.shape[1])]

# create a column of variable names
vif["Features"] = x.columns

# sort the dataframe based on the values of VIF_Factor in descending order
# 'ascending = False' sorts the data in descending order
# 'reset_index' resets the index of the dataframe
# 'drop = True' drops the previous index
vif.sort_values('VIF_Factor', ascending = False).reset_index(drop = True)
```

	VIF_Factor	Features
0	8.641017	Mileage
1	6.172812	Engine
2	4.359966	Age

Tests after model building

- Linear relationship between dependent and independent variables

- Independence of observations should exist (i.e. Absence of Autocorrelation)

- The error terms should be homoscedastic (Constant Variance)

- The error terms must follow normal distribution

Tests after model building

- Linear relationship between dependent and independent variables

- Independence of observations should exist (i.e. Absence of Autocorrelation)

- The error terms should be homoscedastic

- The error terms must follow normal distribution

Assumption of linearity

- An assumption of linear regression is that it should be linear in the parameter

- The independent variables must have a linear relationship with the dependent variable

- The residuals and the fitted values should be independent

Existence of linear relationship

- The independent variables must have a linear relationship with the dependent variable

- This can be checked by plotting a scatter plot of residuals vs predictors

- A scatter plot depicting no pattern indicates that the variable has a linear relationship with the response variable

Existence of linear relationship

In context with our data, we see a random pattern in all the three plots. Hence, we may say that, the predictors are linearly related with the response variable.
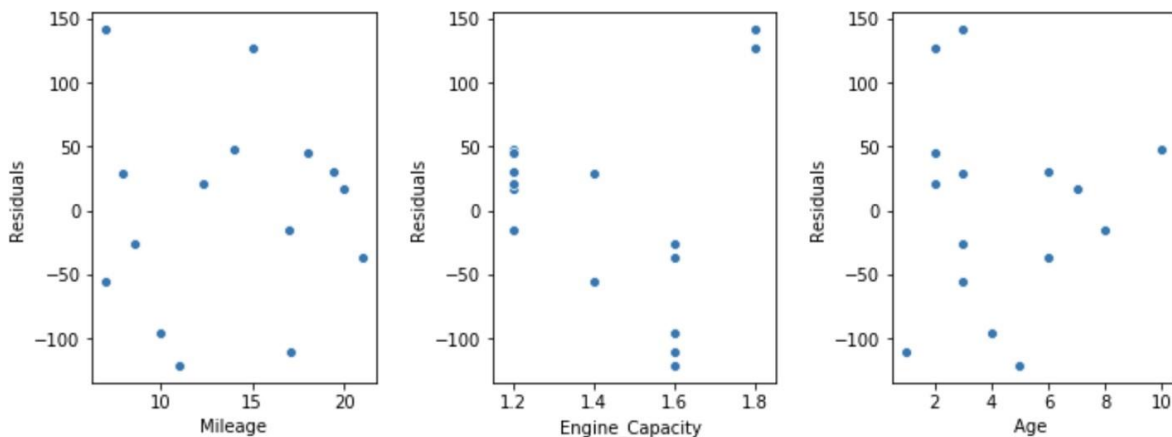
Existence of linear relationship

- **The plot of residuals against the fitted tells the presence of linear relationship**

- For linear relationship, the points must be at random, i.e., it should not exhibit much distinctive pattern, no non-linear trends or changes in variability

Tests after model building

- Linear relationship between dependent and independent variables

- Independence of observations should exist (i.e. Absence of Autocorrelation)

- The error terms should be homoscedastic

- The error terms must follow normal distribution
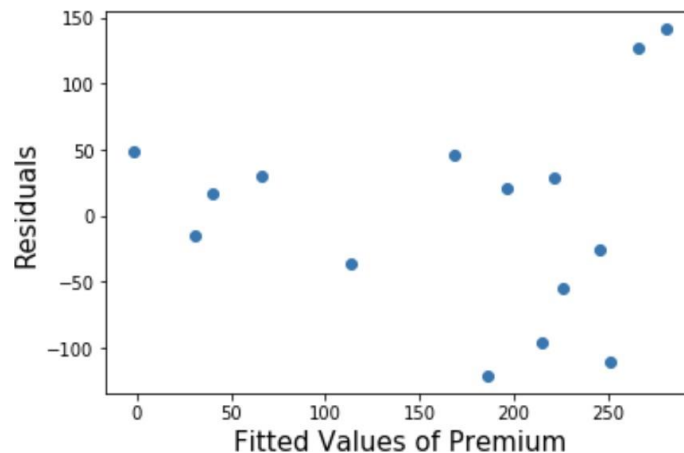
Assumption of autocorrelation

- Assumption of autocorrelation is violated **when residuals are correlated within themselves,** ie they are serially correlated

- Autocorrelation does not impact the regression coefficients but the associated standard errors are reduced

- This reduction in standard error leads to a reduction in associated p-value

- It incorrectly concludes that a predictor is statistically significant

Causes of autocorrelation

- Some important variables are not considered in the data

- If the relationship between the target and predictor variables is non-linear and is incorrectly considered linear

- Presence of carry over effect

  Example: The additional expenses from the budget for last month are carried over in creating the budget for next month

Durbin - Watson Test

- To test whether the **error terms are autocorrelated,** we use Durbin-Watson test

- We test whether autocorrelation is present or not

- The hypothesis is given by

H_0: The error terms are not autocorrelated

against H_1: The error terms are autocorrelated

- Failing to reject H_0, implies that the error terms are not autocorrelated

Durbin - Watson test

The test statistic is given by

Residual of tth observation

Residual of t-1th observation

$$d = \frac{\sum \hat{e}_t - \hat{e}_{t-1}}{\sum \hat{e}_t^2}$$

Value	Interpretation
0 < d < 2	Positive autocorrelation
d = 2	No autocorrelation
2 < d < 4	Negative autocorrelation

Checking AutoCorrelation in Errors - Durbin Watson Test

- The DW test is a measure of **Autocorrelation in Residuals** in Regression Analysis.
- It can lead to underestimation of Standard Errors which will indirectly make believe that Features are Significant wherein they aren't.
- Ho: No AutoCorrelation Exists & H1: AutoCorrelation Exists
- The Test Statistic throws a value between 0 to 4.
- Parameter 2 = No AutoCorrelation, 0 to 2 - Positive AutoCorrelated, >2 to 4 - Negative AutoCorrelation
- Rule of Thumb - test statistic values in the range of 1.5 to 2.5 are relatively normal. Values outside of this range could be cause for concern.

```
from statsmodels.stats.stattools import durbin_watson

# Fitting the Model
model = ols("Premium~Mileage+Engine+Age", data = premium).fit()

#Generating DW Test Statistic
print("DW Test Statistic:",durbin_watson(model.resid))

# Inference: Since it is in range of 1.5 to 2.5, the model is not suffering from Autocorrelation

DW Test Statistic: 2.3735258408206343
```

Model Summary

|				
Omnibus:	0.101	Durbin-Watson:	2.374
Prob(Omnibus):	0.951	Jarque-Bera (JB):	0.285
Skew:	0.143	Prob(JB):	0.867
Kurtosis:	2.388	Cond. No.	167.

Tests after model building

- Linear relationship between dependent and independent variables

- Independence of observations should exist (i.e. Absence of Autocorrelation)

- The error terms should be homoscedastic

- The error terms must follow normal distribution

Homoscedasticity assumption

Homoscedasticity

Same Variance

- Variance of the residual is assumed to be independent of the explanatory variables

- Heteroscedasticity: non-constant variance of residuals

- It happens due to the presence of extreme values

Heteroscedasticity
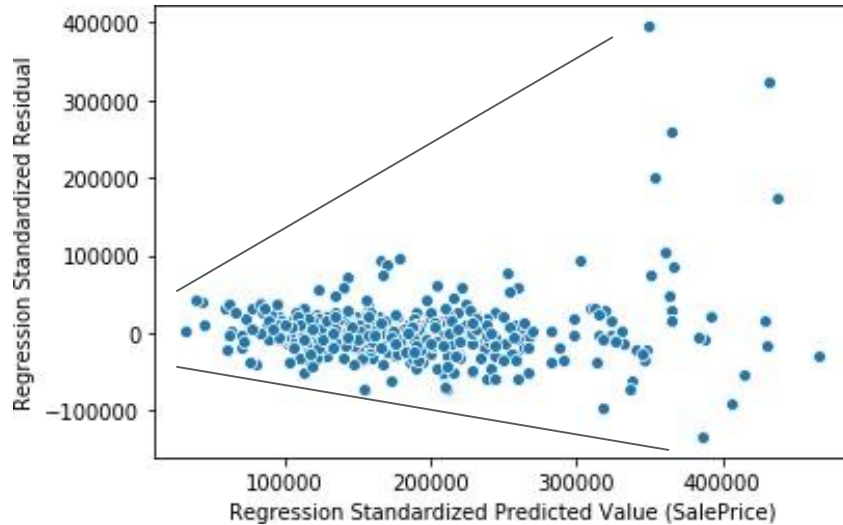
- **Funnel type shape** is seen in the graph

- Hence we can say that there is a presence of "Heteroscedasticity"

Homoscedasticity
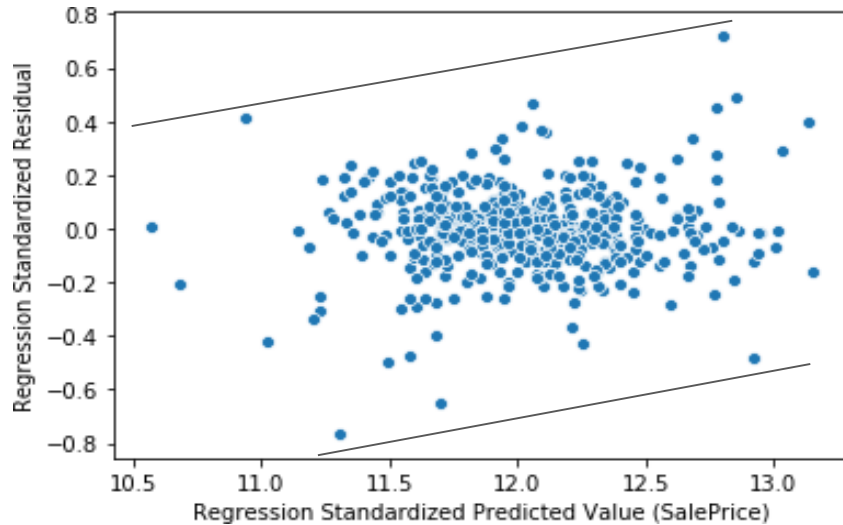
- There is no visible funnel or bow type pattern in the plot

- We can see presence of "Homoscedasticity"

Residual vs Fitted plot

- The plot of residuals against the fitted values tells whether the error terms have equal variance

- It should look random, i.e., it should not exhibit much distinctive pattern, no non-linear trends or changes in variability
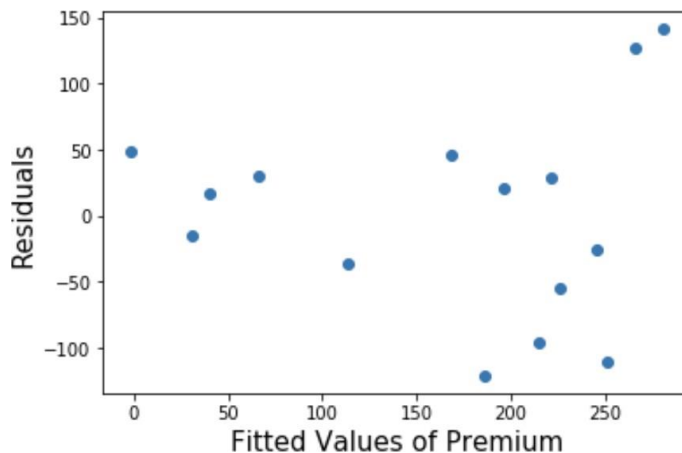
Homoscedasticity

The statistical test to test for the homoskedasticity of the errors are

- Goldfeld Quandt test

- Breusch Pagan test

Goldfeld-Quandt test

- For presence of a constant variance of error terms, i.e. to test

H_0: The errors terms are homoskedastic

against H_1:   The errors terms are heteroskedastic

- Decision rule: Reject H_0, if the p-value associated with test statistic is less than α (level of significance), which implies that there is heteroskedastic, i.e. the error terms have do not equal variance

Breusch Pagan test

- For presence of a constant variance of error terms, i.e. to test

$$H_0: \text{The errors terms are homoskedastic}$$

against $$H_1: \text{The errors terms are heteroskedastic}$$

- Decision rule: Reject H_0, if the p-value associated with test statistic is less than α (level of significance), which implies that there is heteroskedastic, i.e. the error terms have do not equal variance

GoldFeld Quandt Test

Checking Homoscedasticity – Residuals have uniform variance

Checking Heteroscedasticity - GoldFeldQuandt Test

- The Goldfeld Test is a measure of **Homoskedasticity** in Regression Analysis.
- It tells us whether the sample pulled out randomly has a uniform variance or not.
- Ho: Residuals are Homoskedastic & H1: Residuals are not Homoskedastic
- **Inference: If the PValue > 0.05, then we fail to Reject the Ho meaning that the Residuals have uniform Variance.**

```python
import statsmodels.stats.api as gq

predictors = premium.drop("Premium", axis = 1)

# The Test generates Test Statistic & P-Value
gq.het_goldfeldquandt(model.resid, predictors)

# Inference: The Residuals are Homoskedastic in Nature.

(0.4630657017401425, 0.7621813259514729, 'increasing')
```

Tests after model building

- Linear relationship between dependent and independent variables

- Independence of observations should exist (i.e. Absence of Autocorrelation)

- The error terms should be homoscedastic

- The error terms must follow normal distribution

Normality test

- Parametric statistical methods assume that the underlying data has a normal distribution

- Normality tests are used to determine if a data set is well-modeled by a normal distribution

Normality testing techniques

- Quantile-Quantile Plot

- Jarque-Bera (JB) Test

- Shapiro-Wilk Test
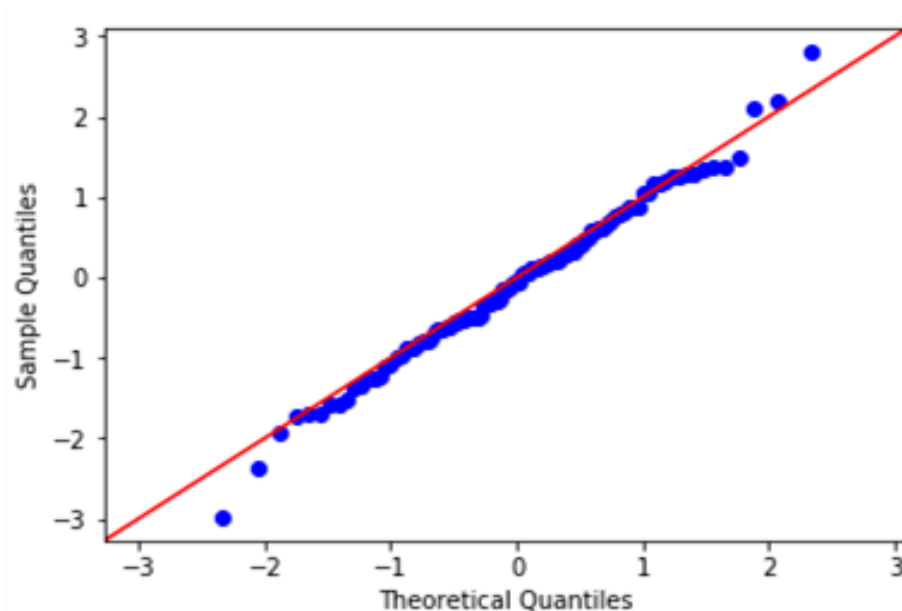
Quantile-Quantile Plot (QQ plot)

- Used to determine whether two datasets follow the same distribution

- The quantiles of two datasets are plotted against each other

- A reference line is plotted at 45^0

- If the points lie on the reference line we conclude that they follow the same

  distribution

Normal QQ plot
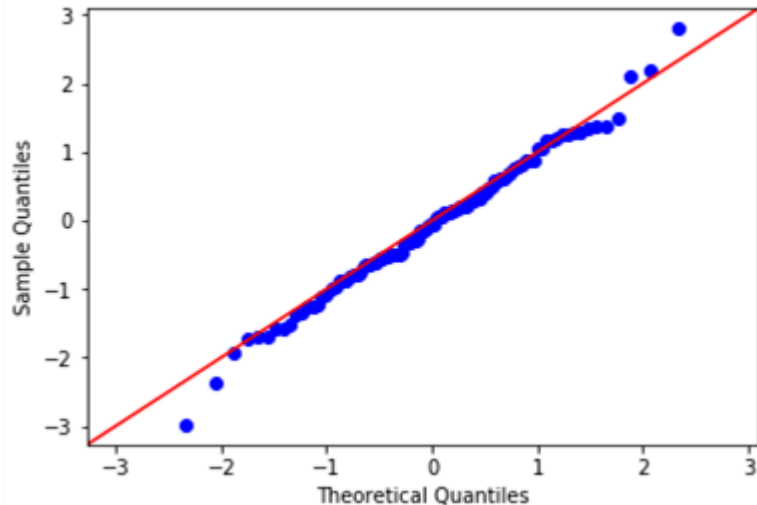
- The x axis has points from a theoretically calculated normal distribution

- They are compared with sample data on the y axis

- If the sample data has a normal distribution the points lie on the reference line

Shapiro-Wilk test

- To test whether the data follows normal distribution, i.e. to test

H_0: The data is normally distributed

against H_1: The data is not normally distributed

- Failing to reject H0, implies that the data follows normal distribution

Shapiro-Wilk test statistic

The test statistic is given by

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_i\right)^2}{\sum_{i=1}^{n} \left(x_i - \overline{x}\right)^2}$$

n = sample size

a_i = values computed from n samples (of size n each) from normal distribution based on their means, covariance matrix

x_i = i^{th} ordered sample values

$\overline{x}$ = sample mean

Shapiro Wilk Test of Normality

Applying Shapiro Wilk on Premium Dataset

Checking Normality - Shapiro Wilk Test & Probability Plot

- The Shapiro Wilk Test is a measure of **Normality** in Regression Analysis.
- It tells us whether the sample pulled out randomly is normally distributed or not.
- Ho: Data is Normal & H1: Data is not Normal
- Inference: If the PValue > 0.05, then we fail to Reject the Ho meaning that the Data is Normal else the data is not Normal.

```
# Applying Shapiro Wilk test on Dependent Variable
stats.shapiro(premium["Premium"])

#Inference: Since the P-Value is more than 0.05, We conclude that the Data is Normal.
# Note: We can create Prob Plot to visualize the Same.
# Remember that the Data Points should be close to the red line to ensure Normality.

ShapiroResult(statistic=0.9073696732521057, pvalue=0.12340115755796432)
```

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_i\right)^2}{\sum_{i=1}^{n} (x_i - \overline{x})^2}$$

```
# Import Probplot
from scipy.stats import probplot

# Generating the Prob Plot
probplot(premium["Premium"], plot = plt)
plt.show()
```


Probability Plot

- Notice that most of the data points are lying on the redline or lying very close to the red line. This shows that the data is Normal.
- The same can be verified by the Shapiro Wilk Test of Normality
- Note: The test is ideally performed on Residuals to check the Normality
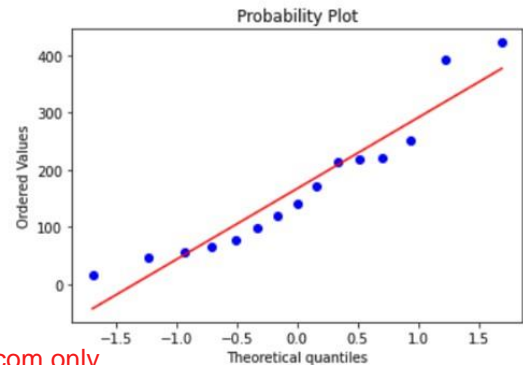
Shapiro Wilk Test of Normality - Residuals

Applying Shapiro Wilk on Residuals of Premium Dataset

Checking Normality - Shapiro Wilk Test & Probability Plot

- The Shapiro Wilk Test is a measure of **Normality** in Regression Analysis.
- It tells us whether the sample pulled out randomly is normally distributed or not.
- Ho: Data is Normal & H1: Data is not Normal
- **Inference: If the PValue > 0.05, then we fail to Reject the Ho meaning that the Data is Normal else the data is not Normal.**

```
# Applying Shapiro Wilk test on Residuals

stats.shapiro(model.resid)

#Inference: Since the P-Value is more than 0.05, We conclude that the Data is Normal.
# Note: We can create Prob Plot to visualize the Same.
# Remember that the Data Points should be close to the red line to ensure Normality.

ShapiroResult(statistic=0.9564691185951233, pvalue=0.6313204169273376)
```

```
from scipy.stats import probplot

probplot(model.resid, plot = plt)
plt.show()
```


Probability Plot

- Notice that most of the data points are lying on the redline or lying very close to the red line. This shows that the residuals are Normally Distributed.
- The same can be verified by the Shapiro Wilk Test of Normality
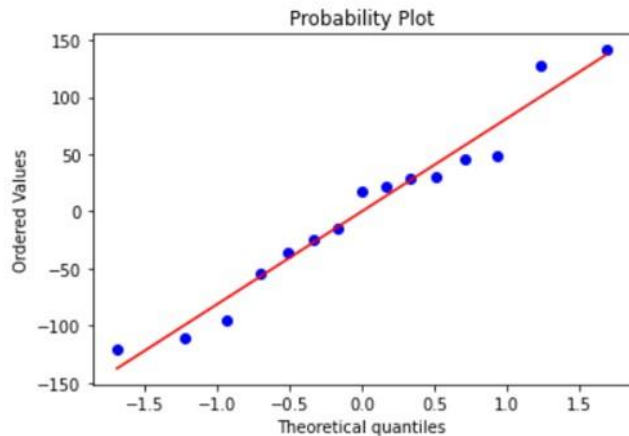
JB test

- To test whether the data follows normal distribution, we test whether the skewness and kurtosis of the data are same as that of the normal distribution, i.e. to test

H_0: Skewness (S) = 0 and Kurtosis (K) = 0

against H_1: Skewness (S) ≠ 0 and Kurtosis (K) ≠ 0

- Failing to reject H_0, implies that the data follow normal distribution

$$JB = \frac{n}{6}\left(S^2 + \frac{1}{4}(K-3)^2\right)$$

JB test

The test statistics for n observations is given by

$$JB = \frac{n}{6}\left(S^2 + \frac{1}{4}(K-3)^2\right)$$

Sample skewness

Kurtosis coefficient

Inference: JB test confirms that the Data is normal.

Checking Normality - Jarque Bera Test

- The JB Test is a measure of **Normality** in Regression Analysis.
- It tells us whether the sample pulled out randomly is normally distributed or not.
- The JB test is usually meant for **Large Datasets** as Shapiro Wilk supports till 5000 rows of data.
- The Test uses Skewness and Kurtosis to find out if the Data is Normal or not.
- The Skewness is 0 and Kurtosis is 3 for a Normally Distributed Data. Any Deviation from this is considered that the Data is not Normal.
- Ho: Data is Normal & H1: Data is not Normal
- **Inference: If the PValue > 0.05, then we fail to Reject the Ho meaning that the Data is Normal else the data is not Normal.**

```python
# Import Jarque Bera
from scipy.stats import jarque_bera

# Applying JB test on Dependent Variable
print("Dependent Variable Pvalue:", jarque_bera(premium["Premium"])[1])

# Applying JB test on Model Residuals
print("Model Residuals Test Statistic:", jarque_bera(model.resid)[0])
print("Model Residuals P Value:", jarque_bera(model.resid)[1])
```

```
Dependent Variable Pvalue: 0.4135353808086716
Model Residuals Test Statistic: 0.2852794190012313
Model Residuals P Value: 0.8670664084149167
```

Omnibus:	0.101	**Durbin-Watson:**	2.374
Prob(Omnibus):	0.951	**Jarque-Bera (JB):**	0.285
Skew:	0.143	**Prob(JB):**	0.867
Kurtosis:	2.388	**Cond. No.**	167.

Summary of Assumptions

- Data is Normally Distributed

- Data is Homoscedastic in Nature.

- There is **no Autocorrelation** of Errors in the Model

- The Multicollinear Variables appear to be Mileage & Engine. Thus, it is advisable to remove these columns and build a new model.

Interaction Effect

Interaction effect

Interaction

- An interaction effect occurs when the effect of one variable depends on another variable. This combined effect may or may not improve the performance of the model

- Note: It does not imply that the predictor variables are collinear

Example: Salary of an employee increases with experience, but this may vary based whether the person has completed additional courses like MBA

Interaction Effect

- In context with our example, we shall consider the interaction effect of variables Engine_Capacity and Mileage

- We obtained Int_EC_Mil by taking the product of Mileage and Engine_Capacity

- Let us check whether the interaction term is adding value to our model

Mileage	Engine_Capacity	Int_EC_Mil	Age	Premium (in dollars)
15	1.8	27	2	392.5
14	1.2	16.8	10	46.2
17	1.2	20.4	8	15.7
7	1.8	12.6	3	422.2
10	1.6	16	4	119.4
7	1.4	9.8	3	170.9
20	1.2	24	7	56.9
21	1.6	33.6	6	77.5
18	1.2	21.6	2	214
11	1.6	17.6	5	65.3
7.9	1.4	11.06	3	250
8.6	1.6	13.76	3	220
12.3	1.2	14.76	2	217.5
17.1	1.6	27.36	1	140.88
19.4	1.2	23.28	6	97.25

Interaction Effect

Now our model is

$$\text{Premium} = \beta_0 + \beta_1 \text{ Mileage} + \beta_2 \text{ Engine\_Capacity} + \beta_3 \text{Age} + \beta_4 \text{ Int\_EC\_Mil} + \varepsilon$$

Parameter	Description
β_0	Premium value where the best fit line cuts the Y-axis (Premium)
β_1	Regression coefficient of the variable Mileage
β_2	Regression coefficient of the variable Engine_Capacity
β_3	Regression coefficient of the variable Age
β_4	Regression coefficient of the variable Int_EC_Mil

Mileage	Engine_Capacity	Int_EC_Mil	Age	Premium (in dollars)
15	1.8	27	2	392.5
14	1.2	16.8	10	46.2
17	1.2	20.4	8	15.7
7	1.8	12.6	3	422.2
10	1.6	16	4	119.4
7	1.4	9.8	3	170.9
20	1.2	24	7	56.9
21	1.6	33.6	6	77.5
18	1.2	21.6	2	214
11	1.6	17.6	5	65.3
7.9	1.4	11.06	3	250
8.6	1.6	13.76	3	220
12.3	1.2	14.76	2	217.5
17.1	1.6	27.36	1	140.88
19.4	1.2	23.28	6	97.25

Linear regression model (interaction effect)

Based on the data, the β parameters are:

$\beta_0 = -502.011$, $\beta_1 = 40.306$, $\beta_2 = 568.723$,

$\beta_3 = -25.781$ and $\beta_4 = -30.547$

Thus the model is

$$Y = 502.011 + 40.306\ x_1 + 68.723\ x_2 - 25.781 x_3 - 30.547\ x_4$$

That is,

Premium = 502.011 + 40.306 Mileage + 68.723 Engine_Capacity - 30.54 Age - 25.781 Int_EC_Mil

Mileage	Engine_Capacity	Int_EC_Mil	Age	Premium (in dollars)
15	1.8	27	2	392.5
14	1.2	16.8	10	46.2
17	1.2	20.4	8	15.7
7	1.8	12.6	3	422.2
10	1.6	16	4	119.4
7	1.4	9.8	3	170.9
20	1.2	24	7	56.9
21	1.6	33.6	6	77.5
18	1.2	21.6	2	214
11	1.6	17.6	5	65.3
7.9	1.4	11.06	3	250
8.6	1.6	13.76	3	220
12.3	1.2	14.76	2	217.5
17.1	1.6	27.36	1	140.88
19.4	1.2	23.28	6	97.25

Presence of categorical variable

Linear regression of categorical variable

- The regression method fails in presence of categorical variable

- Thus we need to convert the categorical variable to numeric variable

- In order to so, we use N - 1 dummy encoding

N-1 dummy encoding

- Dummy variables are binary variables used to represent categorical data

- For a categorical variable that can take k values k-1 dummy variables need to be created

- Dummy variable is assigned 1 if it takes a particular value else it is assigned 0

Dummy variable example

Consider a variable, Gender, used to represent the gender of a citizen during the census

Gender: Male, Female

Since Gender takes 2 values it can be represented with 1 dummy variable D_1 as:

Value	D_1
Male	0
Female	1

Data

Let us consider a categorical variable Manufacturer in the data and find out how it behaves.

Mileage	Manufacturer	Premium (in dollars)
15	Ford	392.5
14	Honda	46.2
17	Tata	15.7
7	Ford	422.2
10	Ford	119.4
7	Tata	170.9
20	Tata	56.9
21	Honda	77.5
18	Honda	214
11	Tata	65.3
7.9	Ford	250
8.6	Tata	220
12.3	Tata	217.5
17.1	Ford	140.88
19.4	Honda	97.25

Example

- In context with our example, the categorical variable Manufacturer takes values Ford, Honda and Tata

- Since Manufacturer takes 3 values, two dummy variables Mfr_Honda and Mfr_Tata are created

Value	Mfr_Honda	Mfr_Tata
Ford	0	0
Honda	1	0
Tata	0	1

Model with categorical variable

Now our model is

$$Premium = \beta_0 + \beta_1\,Mileage + \beta_2\,Mfr\_Honda + \beta_3\,Mfr\_Tata + \varepsilon$$

Parameter	Description
β_0	Premium value where the best fit line cuts the Y-axis (Premium)
β_1	Regression coefficient of the variable Mileage
β_2	Regression coefficient of the dummy variable Mfr_Honda
β_3	Regression coefficient of the dummy variable Mfr_Tata

Linear regression model (dummy variable)

Mileage	Manufacturer	Premium (in dollars)
15	Ford	392.5
14	Honda	46.2
17	Tata	15.7
7	Ford	422.2
10	Ford	119.4
7	Tata	170.9
20	Tata	56.9
21	Honda	77.5
18	Honda	214
11	Tata	65.3
7.9	Ford	250
8.6	Tata	220
12.3	Tata	217.5
17.1	Ford	140.88
19.4	Honda	97.25

Based on the data, the β parameters are:

$\beta_0 = 368.93$, $\beta_1 = -9.117$,

$\beta_2 = -95.174$ and $\beta_3 = -129.216$

Thus the model is

$Y = 368.93 - 9.117\, x_1 - 95.174\, x_2 - 129.216\, x_3$

That is,

Premium = 368.93 - 9.117 Mileage - 95.174 Mfr_Honda - 129.216 Mfr_Tata

Regression line (dummy variable)

The regression line:

$$\text{Premium} = \beta_0 + \beta_1 \text{ Mileage} + \beta_2 \text{ Mfr\_Honda} + \beta_3 \text{ Mfr\_Tata} + \varepsilon$$

If the manufacturer is Honda, the regression line becomes:

$$\text{Premium} = \beta_0 + \beta_1 \text{ Mileage} + \beta_2 \text{ Mfr\_Honda} + \beta_3 \text{ Mfr\_Tata}$$
$$= \beta_0 + \beta_1 \text{ Mileage} + \beta_2 \text{ (1)} + \beta_3 \text{ (0)}$$
$$= \beta_0 + \beta_1 \text{ Mileage} + \beta_2 + 0$$
$$= (\beta_0 + \beta_2) + \beta_1 \text{ Mileage}$$

Value	Mfr_Honda	Mfr_Tata
Ford	0	0
Honda	1	0
Tata	0	1

Note the change in the intercept value.

Regression line (dummy variable)

The regression line:

Premium $= \beta_0 + \beta_1$ Mileage $+ \beta_2$ Mfr_Honda $+ \beta_3$ Mfr_Tata $+ \varepsilon$

Value	Mfr_Honda	Mfr_Tata
Ford	0	0
Honda	1	0
Tata	0	1

For manufacturer = Ford,

Premium $= \beta_0 + \beta_1$ Mileage

Actual intercept

For manufacturer = Honda,

Premium $= (\beta_0 + \beta_2) + \beta_1$ Mileage

Change in intercept

For manufacturer = Tata,

Premium $= (\beta_0 + \beta_3) + \beta_1$ Mileage

Change in intercept

Appendix

Maths behind OLS

- We have seen that the error term $\varepsilon = y - (\beta_0 + \beta_1 x)$

- The OLS method minimizes $E = \sum \varepsilon^2 = \sum(y - (\beta_0 + \beta_1 x))^2$

- To minimize the error we take partial derivatives with respect to β_0 and β_1 and equate them to zero

$$\delta E/\delta\beta_0 = 0$$

$$\delta E/\delta\beta_1 = 0$$

- So we get two equations with two unknowns, β_0 and β_1

Maths behind OLS

- So we get:

$$\delta E / \delta \boldsymbol{\beta_0} = \Sigma \ 2 \ (\mathbf{y} - \boldsymbol{\beta_0} - \boldsymbol{\beta_1}\mathbf{x}) \ (-1) = 0$$

$$\delta E / \delta \boldsymbol{\beta_1} = \Sigma \ 2 \ (\mathbf{y} - \boldsymbol{\beta_0} - \boldsymbol{\beta_1}\mathbf{x})) \ (-x_1) = 0$$

- Expanding these equations, we get $\boldsymbol{\beta_0}$ and $\boldsymbol{\beta_1}$ as:

$$\beta_0 = \bar{y} - \beta_1 \bar{x} \qquad\qquad \beta_1 = \frac{Cov(X,Y)}{Var(X)}$$

Parameter estimation - OLS method

- We obtain the estimates of β_0, β_1, β_2 and β_3 to minimize the term

$$E = \sum \mathbf{\varepsilon}^2 = y - \sum(\mathbf{y} - (\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3))^2$$

- To minimize the error we take partial derivatives with respect to β_0, β_1, β_2 and β_3 and equate them to zero

$$\delta E / \delta \beta_0 = 0 \qquad\qquad\qquad \delta E / \delta \beta_1 = 0$$
$$\delta E / \delta \beta_2 = 0 \qquad\qquad\qquad \delta E / \delta \beta_3 = 0$$

- So we get four equations with four unknowns, β_0, β_1, β_2 and β_3

Parameter estimation - OLS method

- Solving those equations gets tough

- So, we make use of matrix form, in order to get OLS estimates

- We will first see matrix notation for simple linear regression and then for multiple linear regression

Equations for simple linear regression

Using (x_1, y_1), (x_2, y_2), (x_3, y_3) ...... (x_n, y_n) we would have the equations:

$$y_1 = (\beta_0 + \beta_1 x_{11}) + \varepsilon_1$$

$$y_2 = (\beta_0 + \beta_1 x_{12}) + \varepsilon_2$$

$$y_3 = (\beta_0 + \beta_1 x_{13}) + \varepsilon_3$$

...

$$y_n = (\beta_0 + \beta_1 x_{1n}) + \varepsilon_n$$

Matrix equation for simple linear regression

Expressing the equations from previous slide in matrix form:

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} \quad X = \begin{bmatrix} 1 & x_{11} \\ 1 & x_{12} \\ 1 & x_{13} \\ \vdots & \vdots \\ 1 & x_{1n} \end{bmatrix} \quad \hat{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

$$\text{n x 1} \qquad\qquad \text{n x 2} \qquad\qquad \text{2 x 1} \qquad\qquad \text{n x 1}$$

This gives us the Matrix equation: $Y = \beta X + \varepsilon$

Using Linear regression technique, we solve for β's

Equations for multiple linear regression

For 3 predictor variable and n observations, we would have the following equations:

$$y_1 = (\beta_0 + \beta_1 x_{11} + \beta_2 x_{21} + \beta_3 x_{31}) + \varepsilon_1$$
$$y_2 = (\beta_0 + \beta_1 x_{12} + \beta_2 x_{22} + \beta_3 x_{32}) + \varepsilon_2$$
$$y_3 = (\beta_0 + \beta_1 x_{13} + \beta_2 x_{23} + \beta_3 x_{33}) + \varepsilon_3$$
$$\ldots$$
$$y_n = (\beta_0 + \beta_1 x_{1n} + \beta_2 x_{2n} + \beta_3 x_{3n}) + \varepsilon_n$$

Matrix equation for multiple linear regression

In Matrix form, it would look as follows:

$$
Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}
\quad
X = \begin{bmatrix} 1 & x_{11} & x_{21} & x_{31} \\ 1 & x_{12} & x_{22} & x_{32} \\ 1 & x_{13} & x_{23} & x_{33} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} & x_{3n} \end{bmatrix}
\quad
\hat{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix}
\quad
E = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}
$$

$$
n \times 1 \qquad\qquad n \times (3+1) \qquad\qquad (3+1) \times 1 \qquad\qquad n \times 1
$$

Here n is the number of observations.

The OLS estimates

For multiple linear regression, the OLS estimates which give the best fit are obtained as

$$\hat{\beta} = [X'X]^{-1}X'Y$$

$$\hat{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}$$

X' denotes the transpose of matrix X.

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & x_{31} \\ 1 & x_{12} & x_{22} & x_{32} \\ 1 & x_{13} & x_{23} & x_{33} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1n} & x_{2n} & x_{3n} \end{bmatrix}$$

Existence of linear relationship

An assumption of linear regression is that it should be linear in the parameter

Linear Relationship

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2^2 + \varepsilon$$

$$y = \beta_0 - \beta_1 \log(x_1) + \beta_2 x_2 + \varepsilon$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 - \beta_3 x_1 x_2 + \varepsilon$$

Nonlinear Relationship

$$y = \beta_0 - e^{\beta_1 x_1} + \varepsilon$$

$$y = \beta_0 x_1 / \beta_1 x_1 + \varepsilon$$

$$y = \beta_0 + x_1^{\beta_1} . x_2^{\beta_2} + \varepsilon$$

Model Evaluation Metrics

Model evaluation metrics

The model evaluation metrics are

- R^2

- Adjusted R^2

- The F test for overall significance

R-squared

- The R^2 value gives the percentage of variation in the response variable explained by the predictor variables

- If the values of $R^2 = 0.87$, it implies that 87% of variation in the response variable is explained by the predictor variables

$$R^2 = \frac{\text{Explained variation}}{\text{Total variation}} = \frac{\text{SSR}}{\text{SST}}$$

Adjusted R-squared

- Adjusted R^2 gives the percentage of variation explained by independent variables that actually affect the dependent variable

- If the values of $R^2 = 0.87$, it implies that 87% of variation in the response variable is explained by the predictor variables

$$R^2_{adj} = 1 - \frac{(1-R^2)(n-1)}{n-k-1}$$

F test

- To check the significance of the regression model we use the F test

- It is similar to ANOVA for regression

- The test hypothesis is given by

-
    $$H_0: \beta_1 = \beta_2 = \beta_2 = \dots = \beta_n = 0$$

    against $\quad$ $H_1 : \beta_i > 0 \ \text{ or } \ \beta_i < 0 \quad$ for at least one of the i values

- Failing to reject H_0, implies that the model is not significant

F statistic

- The test statistics is given by

$$Fstat = \frac{(SST-SSE)/k}{SSE/(n-k-1)}$$

$$n = \text{sample size}$$

$$k = \text{number of predictor variables}$$

- Decision rule: Reject H_0, if $F_0 > F_{(k,n-k-1),\alpha}$ or if the p-value is less than the α (level of significance)

Optimization Algorithm

The gradient descent

- Using the OLS method, we get the estimates of parameters of the linear regression model by minimizing the sum of the square of errors

- The gradient descent is an optimization technique which finds the β parameters such that the error term is minimum

- Computation speed for higher data dimension is more if parameter were to be obtained using the OLS method whereas the gradient descent does it faster

Gradient descent
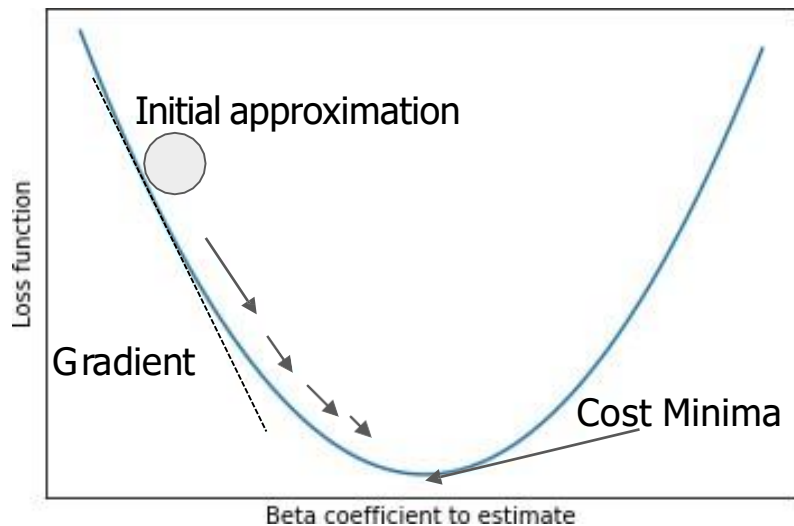
- An error function, also known as a loss function is used to calculate the cost associated with the deviation of observed data from predicted data

- It is an iterative method which converges to the optimum solution

- The estimates of the parameter are updated at every iteration

Example: Gradient descent
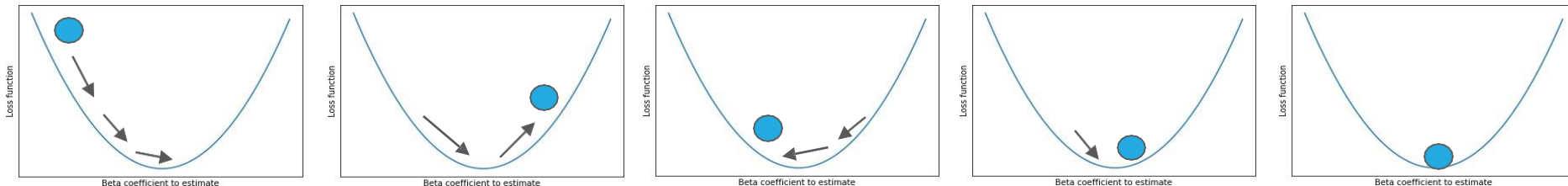
- Consider a ball rolling down the slope as shown above

- Any position on the slope is the loss of the current values of the coefficients (cost)

- The bottom of the slope where the cost function is minimum

- The objective is to find lowest point in the cost function by continuously trying different values of the parameters

- Repeating this process numerous times, the best parameters are such that the cost is minimum

Thank You