Simple and Multiple Linear Regression Model

Installing and loading required packages

```
install.packages("tidyverse")
install.packages("car")
library(car)
library(tidyverse)
library(dplyr)
```

Reading the file Food_Texture_Data.csv using read.csv()function

```
food_df_01 <- read.csv("Food_Texture_Data.csv")
```

Previewing the data frame using View()function

```
View(food_df_01)
```

Checking first few rows of the data set

```
head(food_df_01,5)
```

	X <chr>	Oil <dbl>	Density <int>	Crispy <int>	Fracture <int>
1	B110	16.5	2955	10	23
2	B136	17.7	2660	14	9
3	B171	16.2	2870	12	17
4	B192	16.7	2920	10	31
5	B225	16.3	2975	11	26

5 rows

Adding a new column called Price and storing it in a data frame

Hide

```
food_df_02 <- food_df_01 %>% mutate(Price=80*(Density/(Crispy+Oil+Fracture+Ha
rdness)))

head(food_df_02,5)
```

	X	Oil	Density	Crispy	Fracture	Hard
	<chr>	<dbl>	<int>	<int>	<int>	<
1	B110	16.5	2955	10	23	
2	B136	17.7	2660	14	9	
3	B171	16.2	2870	12	17	
4	B192	16.7	2920	10	31	
5	B225	16.3	2975	11	26	

5 rows

Renaming some variables for ease of using them and storing them in a new data frame

Hide

```
food_df_03 <- food_df_02 %>% rename(Oil_Percentage=Oil)

head(food_df_03)
```

	X	Oil_Percentage	Density	Crispy	Fracture	Harc
	<chr>	<dbl>	<int>	<int>	<int>	
1	B110	16.5	2955	10	23	
2	B136	17.7	2660	14	9	
3	B171	16.2	2870	12	17	
4	B192	16.7	2920	10	31	
5	B225	16.3	2975	11	26	
6	B237	19.1	2790	13	16	

6 rows

Creating the final data frame to work on it further by deselecting the ‘X’ variable

Hide

```
food_data <- food_df_03 %>% select(-X)
head(food_data)
```

	Oil_Percentage	Density	Crispy	Fracture	Hardn
	<dbl>	<int>	<int>	<int>	<i
1	16.5	2955	10	23	
2	17.7	2660	14	9	1
3	16.2	2870	12	17	1
4	16.7	2920	10	31	
5	16.3	2975	11	26	1
6	19.1	2790	13	16	1

6 rows

Previewing the food_data data frame

Hide

```
View(food_data)
```

Building a simple linear regression model

Price as response and Oil_Percentage as predictor variable. Here we use the lm() function to build the model

Hide

```
model1 <- lm(data=food_data, Price ~ Oil_Percentage)
summary(model1)
```

```

Call:
lm(formula = Price ~ Oil_Percentage, data = food_data)


Residuals:
    Min      1Q  Median      3Q     Max
-379.39 -131.15  -23.08   70.96  680.23
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     1580.41     370.82   4.262 9.42e-05 ***
Oil_Percentage   -14.97      21.47  -0.697    0.489    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 239.2 on 48 degrees of freedom
Multiple R-squared:  0.01002,	Adjusted R-squared:  -0.0106 
F-statistic: 0.4861 on 1 and 48 DF,  p-value: 0.489
```

Price as response and Density as predictor variable. Here we use the lm() function to build the model

Hide

```
model2 <- lm(data=food_data,Price ~ Density)
summary(model2)
```

```
Call:
lm(formula = Price ~ Density, data = food_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-368.65 -144.44   -8.28   78.51  686.29 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 733.7651   784.5012   0.935    0.354
Density       0.2062     0.2743   0.752    0.456

Residual standard error: 239 on 48 degrees of freedom
Multiple R-squared:  0.01164,	Adjusted R-squared:  -0.008954 
F-statistic: 0.5651 on 1 and 48 DF,  p-value: 0.4559
```

Price as response and Crispy as predictor variable. Here we use the lm() function to build the model

Hide

```
model3 <- lm(data=food_data,Price ~ Crispy)
summary(model3)
```

```

Call:
lm(formula = Price ~ Crispy, data = food_data)

Residuals:
    Min      1Q  Median      3Q     Max 
-377.37 -112.15  -31.55  116.63  521.46 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2144.72     190.85  11.238 4.84e-15 ***
Crispy        -71.33      16.38  -4.356 6.94e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 203.6 on 48 degrees of freedom
Multiple R-squared:  0.2833,    Adjusted R-squared:  0.2683 
F-statistic: 18.97 on 1 and 48 DF,  p-value: 6.943e-05
```

Price as response and Hardness as predictor variable. Here we use the lm() function to build the model

Hide

```
model4 <- lm(data=food_data, Price ~ Hardness)
summary(model4)
```

```

Call:
lm(formula = Price ~ Hardness, data = food_data)

Residuals:
     Min       1Q   Median       3Q      Max 
```

```
-163.727  -40.152   -1.015   52.931  244.707

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2249.1325    47.5212   47.33   <2e-16 ***
Hardness      -7.2255     0.3605  -20.05   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 78.54 on 48 degrees of freedom
Multiple R-squared:  0.8933,    Adjusted R-squared:  0.8911
F-statistic: 401.8 on 1 and 48 DF,  p-value: < 2.2e-16
```

Price as response and Fracture as predictor variable. Here we use the lm() function to build the model

Hide

```
model5 <- lm(data=food_data, Price ~ Fracture)
summary(model5)
```

```
Call:
lm(formula = Price ~ Fracture, data = food_data)

Residuals:
    Min      1Q  Median      3Q     Max
-385.09 -138.82  -26.05  100.80  603.99

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   928.39     121.97   7.612 8.55e-10 ***
Fracture       18.91       5.66   3.342  0.00162 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 216.6 on 48 degrees of freedom
Multiple R-squared:  0.1888,    Adjusted R-squared:  0.1719
```

```
F-statistic: 11.17 on 1 and 48 DF,  p-value: 0.001617
```

Building a Multiple Linear regression Model

price as response variable and the other variables as predictor variables

Hide

```
model6 <- lm(data=food_data, Price ~ Oil_Percentage+Density+Hardness+Crispy+F
racture)
summary(model6)
```

```

Call:
lm(formula = Price ~ Oil_Percentage + Density + Hardness + Crispy +
    Fracture, data = food_data)

Residuals:
   Min     1Q Median     3Q    Max
-65.18 -35.08 -20.29  13.95 180.62

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    1811.3143   476.7959   3.799 0.000442 ***
Oil_Percentage  -15.3751     8.0573  -1.908 0.062899 .
Density           0.3539     0.1198   2.954 0.005027 **
Hardness         -7.6961     0.3568 -21.570  < 2e-16 ***
Crispy           -9.0464    10.7384  -0.842 0.404101
Fracture         -6.9299     2.7996  -2.475 0.017233 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 56.99 on 44 degrees of freedom
Multiple R-squared:  0.9485,    Adjusted R-squared:  0.9426
F-statistic: 162.1 on 5 and 44 DF,  p-value: < 2.2e-16
```

All the interpretations of models that are build are given in detail in a separate Markdown File. It contains basic notations, basic terminologies and other things. The concept of being confounding and collinear is also covered in that discussion.

Checking the coefficient correlation matrix to know which factor effects the most in predicting the Price. We will discuss this matrix later in detail in the markdown document.

Hide

```
round(cor(food_data[c("Oil_Percentage","Density","Hardness","Fracture","Crispy")]),2)
```

```
               Oil_Percentage Density Hardness Fracture Crispy
Oil_Percentage           1.00   -0.75    -0.10    -0.53   0.59
Density                 -0.75    1.00     0.11     0.57  -0.67
Hardness                -0.10    0.11     1.00    -0.37   0.41
Fracture                -0.53    0.57    -0.37     1.00  -0.84
Crispy                   0.59   -0.67     0.41    -0.84   1.00
```

Now here VIF (Variance Inflation Factor) is calculated for the predictor variables. Here we observe that the VIF value is the highest for the variable Crispy (>5). So we eliminate it and build the model with the other features.

Hide

```
car::vif(model6)
```

```
Oil_Percentage        Density       Hardness         Crispy       Fracture
      2.482121       3.357679       1.860826       5.484192       3.532598
```

Building model considering all but expect Crispy as predictor variable

Hide

```
model7 <- lm(data=food_data, Price ~ Oil_Percentage+Density+Hardness+Fracture
)
summary(model7)
```

```

Call:
lm(formula = Price ~ Oil_Percentage + Density + Hardness + Fracture,
    data = food_data)

Residuals:
   Min     1Q Median     3Q    Max
-59.67 -35.57 -18.70  13.77 182.44

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    1584.3118   392.0806   4.041 0.000205 ***
Oil_Percentage  -16.3574     7.9467  -2.058 0.045370 *
Density           0.3986     0.1071   3.722 0.000548 ***
Hardness         -7.8376     0.3137 -24.983  < 2e-16 ***
Fracture         -5.4860     2.2064  -2.486 0.016684 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 56.81 on 45 degrees of freedom
Multiple R-squared:  0.9477,    Adjusted R-squared:  0.943
F-statistic: 203.7 on 4 and 45 DF,  p-value: < 2.2e-16
```

Here to check collinearity we crate a new column called Density_in_hundred and store it in a new data frame

Hide

```
food_data_1 <- food_data %>% mutate(Density_in_hundred = Density/100)
head(food_data_1,5)
```

	Oil_Percentage	Density	Crispy	Fracture	Hardness	Price
	<dbl>	<int>	<int>	<int>	<int>	<dbl>
1	16.5	2955	10	23	97	1613.652
2	17.7	2660	14	9	139	1184.196
3	16.2	2870	12	17	143	1219.979
4	16.7	2920	10	31	95	1529.797
5	16.3	2975	11	26	143	1212.430

5 rows

Now we build a Multiple Linear Regression Model with Price as response and Density and Density_in_hundred and Hardness as predictor

Hide

```
model8 <- lm(data=food_data_1, Price ~ Density+Hardness+Density_in_hundred)
summary(model8)
```

```

Call:
lm(formula = Price ~ Density + Hardness + Density_in_hundred, 
    data = food_data_1)

Residuals:
    Min      1Q  Median      3Q     Max 
-78.135 -42.325  -6.265  19.954 204.248 

Coefficients: (1 not defined because of singularities)
                     Estimate Std. Error t value Pr(>|t|)    
(Intercept)        1111.67246  199.30589   5.578 1.17e-06 ***
Density               0.40591    0.06991   5.806 5.28e-07 ***
Hardness             -7.40077    0.27962 -26.467  < 2e-16 ***
Density_in_hundred         NA         NA      NA       NA    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 60.57 on 47 degrees of freedom
Multiple R-squared:  0.9379,    Adjusted R-squared:  0.9352 
```

```
F-statistic: 354.7 on 2 and 47 DF,  p-value: < 2.2e-16
```

Getting the mean of Price

Hide

```
mean(food_data$Price)
```

```
[1] 1322.964
```

Now to know what happens in the result if there are categorical predictors, we create a column called Costly(Yes and No) Depending on the price and store it in a new data frame

Hide

```
food_data_final <- food_data_1 %>% mutate(Costly=ifelse(Price>mean(Price),"Ye
s","No"))

head(food_data_final)
```

	Oil_Percentage <dbl>	Density <int>	Crispy <int>	Fracture <int>	Hardness <int>	Price <dbl>	Density_in_hundred <dbl>	Costly <chr>
1	16.5	2955	10	23	97	1613.6519	29.55	Yes
2	17.7	2660	14	9	139	1184.1959	26.60	No
3	16.2	2870	12	17	143	1219.9787	28.70	No
4	16.7	2920	10	31	95	1529.7970	29.20	Yes
5	16.3	2975	11	26	143	1212.4300	29.75	No
6	19.1	2790	13	16	189	941.3749	27.90	No

6 rows

Previewing the latest data frame

Hide

```
View(food_data_final)
```

Now make a model with Price as response variable and Density, Hardness and Costly as predictor variables

Hide

```
model9 <- lm(data=food_data_final, Price ~ Density+Hardness+Costly)
summary(model9)
```

```
Call:
lm(formula = Price ~ Density + Hardness + Costly, data = food_data_final)

Residuals:
    Min      1Q  Median      3Q     Max
-80.915 -41.165  -6.406  19.057 206.500

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 1110.05916  201.33747   5.513 1.54e-06 ***
Density        0.40080    0.07264   5.518 1.52e-06 ***
Hardness      -7.30285    0.43242 -16.888  < 2e-16 ***
CostlyYes      7.97020   26.65785   0.299    0.766
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 61.17 on 46 degrees of freedom
Multiple R-squared:  0.938, Adjusted R-squared:  0.9339
F-statistic: 231.9 on 3 and 46 DF,  p-value: < 2.2e-16
```