
Forecasting Supply Needs: A Regression-Based Approach

Problem Statement: An FMCG company in the instant noodles business has observed a mismatch between demand and supply across its warehouses which lead to inventory cost losses. The management aims to optimize supply distribution by reducing excess supply in low-demand areas and shortages in high-demand areas.

Objectives:

- Develop a model that determines the optimum weight of products to be shipped to the warehouse, based on historical data and relevant operational factors.
- Provide insights for logistics and supply chain teams to improve overall supply chain performance and product availability.

Exploratory Data Analysis

```
> str(df)
'data.frame':   25000 obs. of  24 variables:
 $ Ware_house_ID              : chr  "WH_100000" "WH_100001" "WH_100002" "WH_100003" ...
 $ WH_Manager_ID              : chr  "EID_50000" "EID_50001" "EID_50002" "EID_50003" ...
 $ Location_type              : chr  "Urban" "Rural" "Rural" "Rural" ...
 $ WH_capacity_size           : chr  "Small" "Large" "Mid" "Mid" ...
 $ zone                       : chr  "West" "North" "South" "North" ...
 $ WH_regional_zone           : chr  "Zone 6" "Zone 5" "Zone 2" "Zone 3" ...
 $ num_refill_req_l3m         : int  3 0 1 7 3 8 8 1 8 4 ...
 $ transport_issue_l1y        : int  1 0 0 4 1 0 0 0 1 3 ...
 $ Competitor_in_mkt          : int  2 4 4 2 2 2 4 4 4 3 ...
 $ retail_shop_num            : int  4651 6217 4306 6000 4740 5053 4449 7183 5381 3869 ...
 $ wh_owner_type              : chr  "Rented" "Company Owned" "Company Owned" "Rented" ...
 $ distributor_num            : int  24 47 64 50 42 37 38 45 42 35 ...
 $ flood_impacted             : int  0 0 0 0 1 0 0 0 0 0 ...
 $ flood_proof                : int  1 0 0 0 0 0 0 0 0 0 ...
 $ electric_supply            : int  1 1 0 0 1 1 1 0 1 0 ...
 $ dist_from_hub              : int  91 210 161 103 112 152 77 241 124 78 ...
 $ workers_num                : num  29 31 37 21 25 35 27 23 22 43 ...
 $ wh_est_year                : num  NA NA NA NA 2009 ...
 $ storage_issue_reported_l3m : int  13 4 17 10 18 17 32 19 15 7 ...
 $ temp_reg_mach              : int  0 0 0 1 0 1 0 0 1 0 ...
 $ approved_wh_govt_certificate: chr  "A" "A" "A" "A+" ...
 $ wh_breakdown_l3m           : int  5 3 6 3 6 3 3 6 5 6 ...
 $ govt_check_l3m             : int  15 17 22 27 24 3 6 24 2 2 ...
 $ product_wg_ton             : int  17115 5074 23137 22115 24071 32134 30142 24093 18082 7130 ...
>
```

- The first version of the dataset contained 24 variables.
- The column names were not meaningful.
- There were null values in multiple columns like wh_est_year and workers_num.

```
> colSums(is.na(df))
             Ware_house_ID               WH_Manager_ID               Location_type
                         0                           0                           0
          WH_capacity_size                        zone              WH_regional_zone
                         0                           0                           0
        num_refill_req_l3m         transport_issue_l1y             Competitor_in_mkt
                         0                           0                           0
           retail_shop_num              wh_owner_type                distributor_num
                         0                           0                           0
            flood_impacted                 flood_proof              electric_supply
                         0                           0                           0
             dist_from_hub                 workers_num                wh_est_year
                         0                         990                       11881
storage_issue_reported_l3m           temp_reg_mach approved_wh_govt_certificate
                         0                           0                           0
          wh_breakdown_l3m               govt_check_l3m               product_wg_ton
                         0                           0                           0
>
```

```
> colSums(is.na(df))
             Location_type             WH_capacity_size                         zone
                         0                            0                            0
          WH_regional_zone          num_refill_req_l3m       transport_issue_l1y
                         0                            0                            0
          Competitor_in_mkt            retail_shop_num             wh_owner_type
                         0                            0                            0
           distributor_num             flood_impacted               flood_proof
                         0                            0                            0
           electric_supply              dist_from_hub               workers_num
                         0                            0                            0
              wh_est_year storage_issue_reported_l3m             temp_reg_mach
                         0                            0                            0
approved_wh_govt_certificate          wh_breakdown_l3m             govt_check_l3m
                         0                            0                            0
            product_wg_ton
                         0
>
```

```
> str(df)
'data.frame':   12127 obs. of  22 variables:
 $ location              : chr  "Rural" "Rural" "Rural" "Rural" ...
 $ capacity              : chr  "Large" "Small" "Large" "Small" ...
 $ zone                  : chr  "North" "West" "West" "South" ...
 $ reg_zone              : chr  "Zone 5" "Zone 1" "Zone 6" "Zone 6" ...
 $ refill                : int  3 8 8 8 7 7 4 6 4 8 ...
 $ transport_issue       : int  1 0 0 1 1 0 0 1 1 0 ...
 $ competitor            : int  2 2 4 4 3 5 3 2 4 2 ...
 $ retail_shop           : int  4740 5053 4449 5381 4623 4627 5012 6858 4598 5678 ...
 $ warehouse_Owner       : chr  "Company Owned" "Rented" "Company Owned" "Rented" ...
 $ distributors          : int  42 37 38 42 31 40 48 26 58 31 ...
 $ flood_impacted        : int  1 0 0 0 0 0 0 0 0 0 ...
 $ flood_proof           : int  0 0 0 0 0 0 0 0 0 0 ...
 $ electric_supply       : int  1 1 1 1 1 0 0 1 1 1 ...
 $ distance_hub          : int  112 152 77 124 150 225 95 242 159 65 ...
 $ workers_num           : num  25 35 27 22 37 16 28 36 22 41 ...
 $ w_est_year            : num  2009 2009 2010 2013 1999 ...
 $ storage_issues        : int  18 17 32 15 17 11 4 22 36 11 ...
 $ temperature_regulation: int  0 1 0 1 0 0 0 1 1 0 ...
 $ govt_cert             : chr  "C" "A+" "B" "A+" ...
 $ warehouse_breakdown   : int  6 3 3 5 4 2 1 5 5 4 ...
 $ govt_check            : int  24 3 6 2 6 28 1 11 27 1 ...
 $ product_weight        : int  24071 32134 30142 18082 21125 14115 5124 30063 38082 24062 ...
>
```

```
> cat("Number of numerical variables:", num_numerical, "\n")
Number of numerical variables: 16
> cat("Number of categorical variables:", num_categorical, "\n")
Number of categorical variables: 6
>
```

- Data cleaning was done by removing irrelevant columns and null values, and changing the column names. The modified dataset has 24 variables.
- There are 6 categorical variables: location, capacity, zone, reg_zone, warehouse_owner, and govt_cert. All other variables are numerical.

location	capacity	zone	reg_zone	refill	transport_issue	competitor	retail_shop	warehouse_Owner	distributors	flood_impacted	flood_proof	electric_supply
Rural	Large	North	Zone 5	3	1	2	4740	Company Owned	42	1	0	1
Rural	Small	West	Zone 1	8	0	2	5053	Rented	37	0	0	1
Rural	Large	West	Zone 6	8	0	4	4449	Company Owned	38	0	0	1
Rural	Small	South	Zone 6	8	1	4	5381	Rented	42	0	0	1
Rural	Large	North	Zone 6	7	1	3	4623	Company Owned	31	0	0	1
Rural	Large	North	Zone 6	7	0	5	4627	Rented	40	0	0	0
Urban	Mid	North	Zone 2	4	0	3	5012	Rented	48	0	0	0
Rural	Mid	South	Zone 4	6	1	2	6858	Company Owned	26	0	0	1
Rural	Mid	North	Zone 3	4	1	4	4598	Rented	58	0	0	1
Rural	Mid	South	Zone 2	8	0	2	5678	Company Owned	31	0	0	1

distance_hub	workers_num	w_est_year	storage_issues	temperature_regulation	govt_cert	warehouse_breakdown	govt_check	product_weight
112	25	2009	18	0	C	6	24	24071
152	35	2009	17	1	A+	3	3	32134
77	27	2010	32	0	B	3	6	30142
124	22	2013	15	1	A+	5	2	18082
150	37	1999	17	0	B+	4	6	21125
225	16	2017	11	0	B	2	28	14115
95	28	2022	4	0	B+	1	1	5124
242	36	2008	22	1	A	5	11	30063
159	22	2001	36	1	A+	5	27	38082
65	41	2016	11	0	B+	4	1	24062

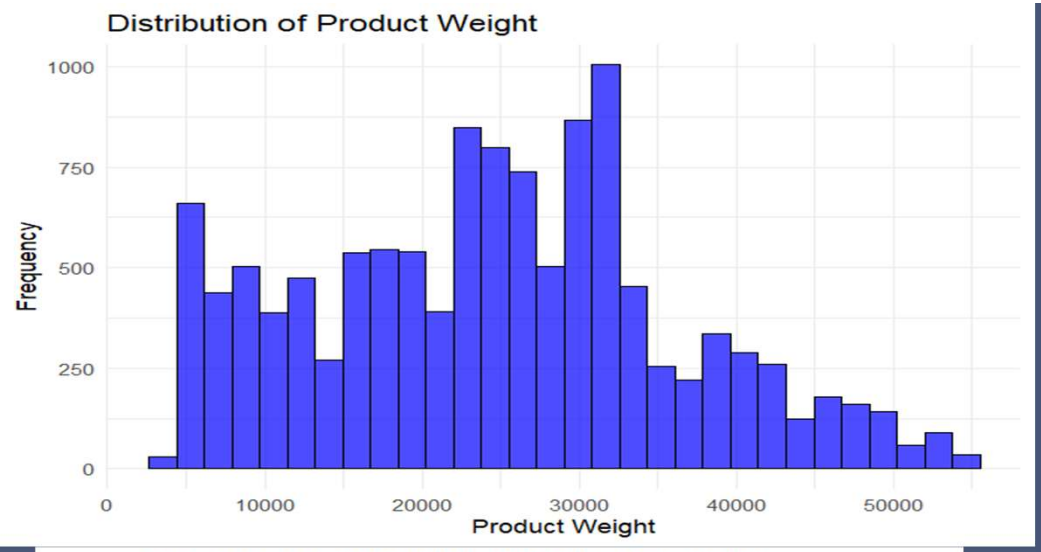
capacity: Storage capacity of the warehouse

refill: Number of times refilling has been done in last 3 months

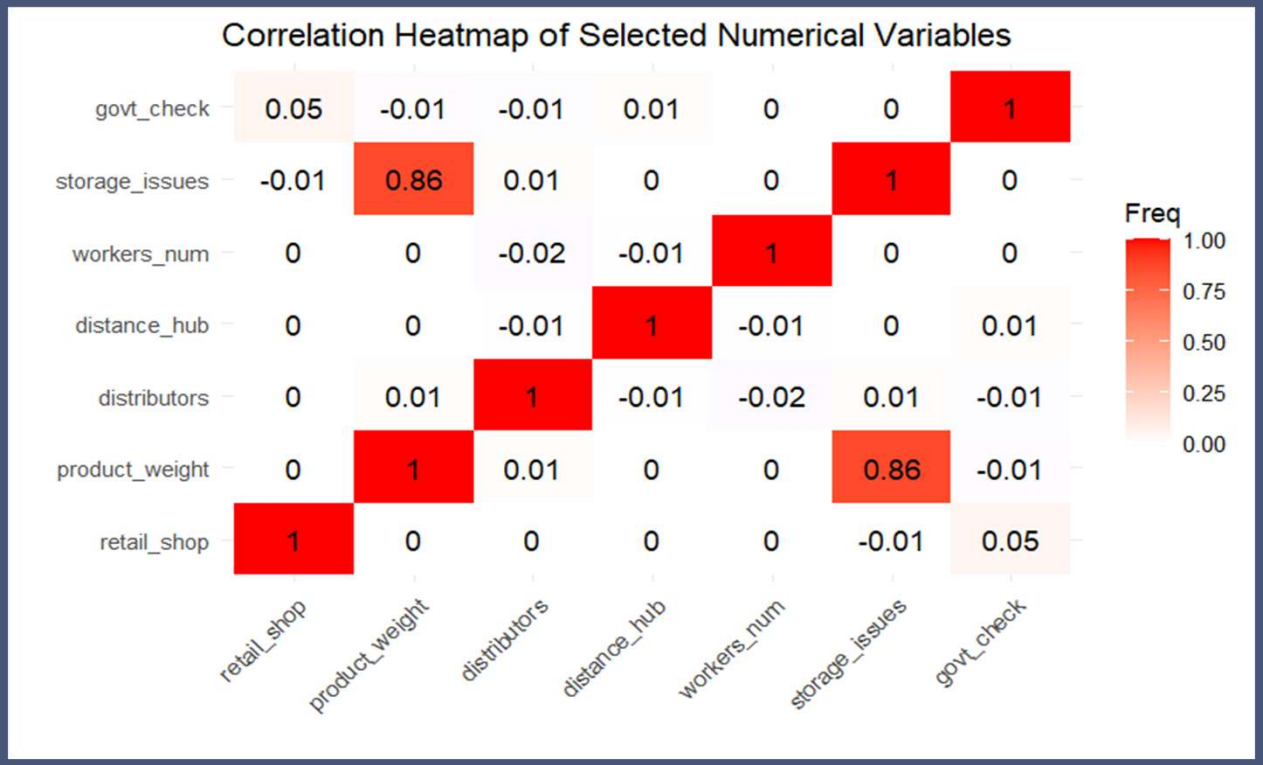
transport_issue: Any transport issue like accident or goods stolen reported

competitor: Number of instant noodles competitor in the market
retail_shop: Number of retails shop who sell the product under the warehouse area
distance_hub: Distance between warehouse to the production hub in kms
storage_issues: Warehouse reported storage issues like rat or fungus infestation to corporate office in last 3 months

warehouse_breakdown: Breakdowns such as strike from worker, flood, or electrical failure
govt_check: Number of time government officers have visited the warehouse
product_weight: Product that has been shipped in last 3 months. Weight is in tons



- A histogram of product weight shows that most of the product weights fall between 20,000–30,000 tons.
- Overall product weight is higher in urban areas than rural suggesting that demand is likely greater in urban regions.
- The East zone has the highest median product weight, followed by the West zone. The North and South zones have almost identical median weights, indicating similar demand levels.



- There is a strong positive correlation (0.86) between product weight and storage issues, suggesting that warehouses with higher product weight are more likely to report storage problems.
- There is a slight positive correlation (0.05) between retail shops and government checks, indicating that locations with more retail shops are subjected to more government checks which could impact logistics and operations.

Data Pre-Processing

Log Transformation

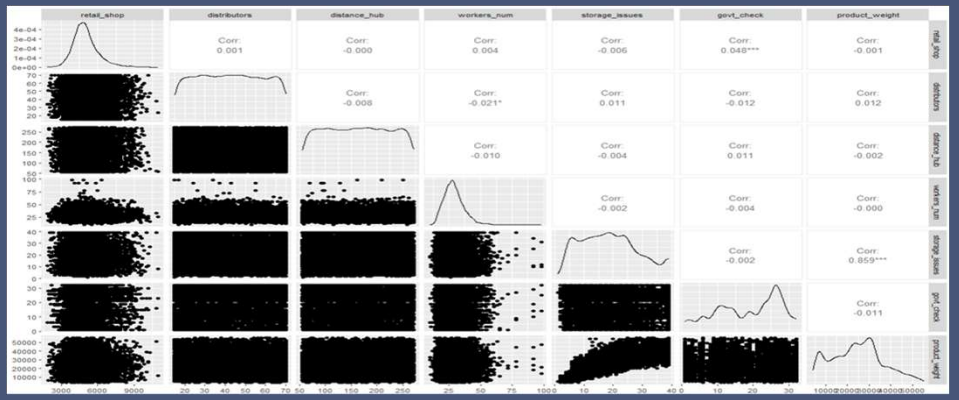
Before Log Transformation

```
'data.frame':   12127 obs. of  6 variables:
 $ retail_shop   : int  4740 5053 4449 5381 4623 4627 5012 6858 4598 5678 ...
 $ distributors  : int  42 37 38 42 31 40 48 26 58 31 ...
 $ distance_hub  : int  112 152 77 124 150 225 95 242 159 65 ...
 $ workers_num   : num  25 35 27 22 37 16 28 36 22 41 ...
 $ storage_issues: int  18 17 32 15 17 11 4 22 36 11 ...
 $ govt_check    : int  24 3 6 2 6 28 1 11 27 1 ...
```

After Log Transformation

```
'data.frame':   12127 obs. of  13 variables:
 $ retail_shop                : num  8.46 8.53 8.4 8.59 8.44 ...
 $ distributors               : num  3.76 3.64 3.66 3.76 3.47 ...
 $ distance_hub               : num  4.73 5.03 4.36 4.83 5.02 ...
 $ workers_num                : num  3.26 3.58 3.33 3.14 3.64 ...
 $ storage_issues             : num  2.94 2.89 3.5 2.77 2.89 ...
 $ govt_check                 : num  3.22 1.39 1.95 1.1 1.95 ...
 $ product_weight             : num  10.09 10.38 10.31 9.8 9.96 ...
```

Pairplot Before Pre-Processing



- Skewness for variables like storage issues, distributors, distance hub, govt check, product weight
- Applied log(x + 1) to the numerical columns
- Reduced skewness in the distribution of numerical variables

Standardization

- Process of re-scaling features so they have
- Mean = 0 and Standard Deviation = 1
- Ensures all the features contribute equally to the mode

Encoding

- Converted categorical data into numerical data i.e. 1,0
- Created binary columns for each category by taking a reference column

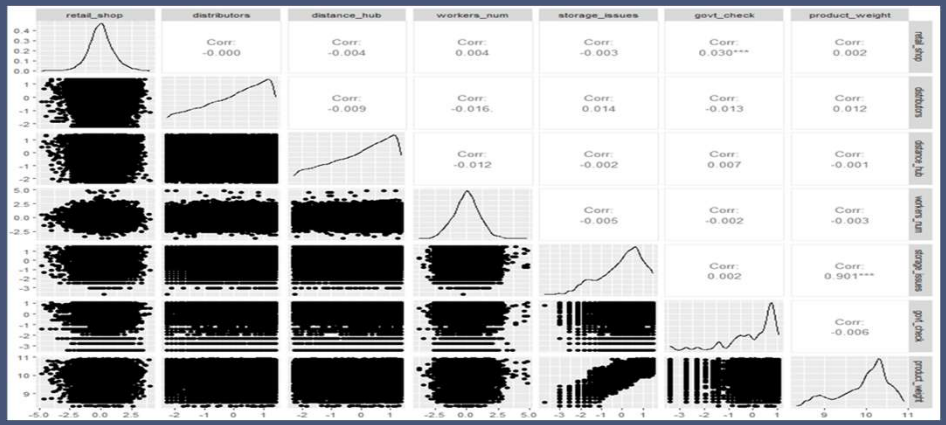
Quadratic & Interaction

- Quadratic terms represents the squared effect of a variable on the target variable (e.x.) `retail_shop^2`
- Interaction terms represents the combined effect of two variables on the target variables `retail_shop * retail_shop`

DataFrame Structure After Pre-processing

```
'data.frame':   12127 obs. of  28 variables:
 $ location                  : Factor w/ 2 levels "Rural","Urban": 1 1 1 1 1 1 2 1 1 1 ...
 $ capacity                  : Factor w/ 3 levels "Large","Mid",..: 1 3 1 3 1 1 2 2 2 2 ...
 $ zone                      : Factor w/ 4 levels "East","North",..: 2 4 4 3 2 2 2 3 2 3 ...
 $ reg_zone                  : Factor w/ 6 levels "Zone 1","Zone 2",..: 5 1 6 6 6 6 2 4 3 2 ...
 $ refill                    : Factor w/ 6 levels "3","4","5","6",..: 1 6 6 6 5 5 2 4 2 6 ...
 $ transport_issue           : Factor w/ 5 levels "0","1","2","3",..: 2 1 1 2 2 1 1 2 2 1 ...
 $ competitor                : Factor w/ 12 levels "0","1","2","3",..: 3 3 5 5 4 6 4 3 5 3 ...
 $ retail_shop               : num  -0.144 0.168 -0.453 0.474 -0.266 ...
 $ warehouse_Owner           : Factor w/ 2 levels "Company Owned",..: 1 2 1 2 1 2 2 1 2 1 ...
 $ distributors              : num  0.1635 -0.135 -0.0722 0.1635 -0.5499 ...
 $ flood_impacted            : Factor w/ 2 levels "0","1": 2 1 1 1 1 1 1 1 1 1 ...
 $ flood_proof               : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ electric_supply           : Factor w/ 2 levels "0","1": 2 2 2 2 2 1 1 2 2 2 ...
 $ distance_hub              : num  -0.6732 0.0317 -1.5354 -0.4384 0.0011 ...
 $ workers_num               : num  -0.406 0.856 -0.119 -0.882 1.066 ...
 $ w_est_year                : Factor w/ 27 levels "1996","1997",..: 14 14 15 18 4 22 27 13 6 21 ...
 $ storage_issues            : num  0.1819 0.0878 1.1423 -0.1171 0.0878 ...
 $ temperature_regulation    : Factor w/ 2 levels "0","1": 1 2 1 2 1 1 1 2 2 1 ...
 $ govt_cert                 : Factor w/ 5 levels "A","A+","B","B+",..: 5 2 3 2 2 4 3 4 1 2 4 ...
 $ warehouse_breakdown       : Factor w/ 6 levels "1","2","3","4",..: 6 3 3 5 4 2 1 5 5 4 ...
 $ govt_check                : num  0.603 -2.297 -1.411 -2.752 -1.411 ...
 $ product_weight            : num  10.09 10.38 10.31 9.8 9.96 ...
 $ retail_shop_retail_shop   : num  -0.144 0.168 -0.453 0.474 -0.266 ...
 $ distributors_distributors  : num  0.1682 -0.129 -0.0665 0.1682 -0.5437 ...
 $ distance_hub_distance_hub  : num  -0.67144 0.03324 -1.536 -0.43657 0.00268 ...
 $ workers_num_workers_num   : num  -0.402 0.857 -0.114 -0.88 1.065 ...
 $ storage_issues_storage_issues: num  0.1957 0.1037 1.1229 -0.0975 0.1037 ...
 $ govt_check_govt_check     : num  0.596 -2.327 -1.384 -2.827 -1.384 ...
```

Pairplot After Pre-Processing



Model Validation Approach

Train-Test Split :

Train Data (9701 rows)

Test Data (2426 rows)

K-Cross Validation:

- It ensures that the model generalizes well to unseen data by repeatedly training and testing it on different subsets of the dataset
- Split dataset into K equal-sized subsets
- Perform K iterations of training and testing

Dataset Assumptions

In the regression models, We have some assumptions for data sets such as

- The relation is linear
- The random errors εi are normally distributed $\sim N(0,\sigma 2)$
- The variance $\sigma 2$ is constant
- The expectation value is zero
- The random errors εi are independent
- No Multicollinearity between independent variables

Model Selection

Simple Linear Regression (SLR)

-> $E(Yi) = \beta 0 + \beta 1 Xi + \varepsilon i$

Multiple Linear Regression w/o interactions and Polynomial Terms (MLR w/o I&Q)

-> $E(Yi) = \beta 0 + \beta 1 Xi1 + \beta 2 Xi2 + \beta 3 Xi3 + \beta 4 Xi4 + \varepsilon i$

Multiple Linear Regression with interactions and Polynomial Terms (MLR with I&Q)

-> $E(Yi) = \beta 0 + \beta 1 Xi1 + \beta 2 Xi2 + \beta 3 Xi3 + \beta 4 Xi4 + \beta 5 Xi1\text{^}2 + \beta 6 Xi2\text{^}2 + \beta 7 (Xi3 * Xi4) + \varepsilon i$

Model Selection - Simple Linear Regression

Formula:

```
#SLR Model
slr_model <- lm(product_weight ~  storage_issues , data =train_df)
summary(slr_model)
anova(slr_model)
```

Summary:

```
Call:
lm(formula = product_weight ~ storage_issues, data = train_df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.06926 -0.15811 -0.01239  0.13320  1.13113

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     9.966757   0.002559    3895   <2e-16 ***
storage_issues 0.522261   0.002560     204   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.252 on 9699 degrees of freedom
Multiple R-squared:  0.811,     Adjusted R-squared:  0.811
F-statistic: 4.163e+04 on 1 and 9699 DF,  p-value: < 2.2e-16
```

Anova:

```
Analysis of Variance Table

Response: product_weight
                 Df  Sum Sq Mean Sq F value    Pr(>F)
storage_issues    1 2643.62 2643.62   41625 < 2.2e-16 ***
Residuals      9699  615.98    0.06
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- SLR Model explains 81.1% of the variance in product weight
- Residual standard error (0.252) is very low which represents average deviation of residuals
- Model fits the data well with a extra sum of squares of 2643.62 units of variance contributed by storage issues feature

Simple Linear Regression - Hypothesis Testing

t-test for slope (β_1) of SLR:

Hypothesis:

Null Hypothesis(Ho): $\beta_1 = 0$

No linear relationship between independent and dependant variable

Alternate Hypothesis (Ha): $\beta_1 \neq 0$

linear relationship between independent and dependant variable

T-statistic (t*) : b1 / s(b1)

Significance Level (alpha) : 5%

Decision Rule:

- **If |t*| > tcritical** or **p-value < α**: Reject Null Hypothesis (Ho)
- **If |t*| <= tcritical** or **p-value >= α**: Fail to reject Null Hypothesis (Ho)

Results:

Since **p-values < significance level**, we reject the null hypothesis and conclude there's a linear association between **Product weight** and **storage issues** features

```
Call:
lm(formula = product_weight ~ storage_issues, data = train_df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.06926 -0.15811 -0.01239  0.13320  1.13113

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    9.966757   0.002559    3895   <2e-16 ***
storage_issues 0.522261   0.002560     204   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.252 on 9699 degrees of freedom
Multiple R-squared:  0.811,     Adjusted R-squared:  0.811
F-statistic: 4.163e+04 on 1 and 9699 DF,  p-value: < 2.2e-16
```

Simple Linear Regression - Hypothesis Testing

Anova test for SLR:

Hypothesis:

Null Hypothesis(Ho): $\beta 1 = 0$
Storage issue feature has no effect on product weight feature

Alternate Hypothesis (Ha): $\beta 1 \neq 0$
Storage issue feature does have effect on product weight feature

F-statistic (F*) : MSR/MSE

Significance Level (alpha) : 5%

Decision Rule:

- **If |F*| > Fcritical** or **p-value < α**: Reject Null Hypothesis (Ho)
- **If |F*| <= Fcritical** or **p-value >= α**: Fail to reject Null Hypothesis (Ho)

Results:

Since **p-values < significance level**, we reject the null hypothesis and conclude storage issues significantly affects product weight

```
Call:
lm(formula = product_weight ~ storage_issues, data = train_df)

Residuals:
     Min       1Q   Median       3Q      Max
-1.06926 -0.15811 -0.01239  0.13320  1.13113

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.966757   0.002559    3895   <2e-16 ***
storage_issues 0.522261   0.002560     204   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.252 on 9699 degrees of freedom
Multiple R-squared:  0.811,     Adjusted R-squared:  0.811
F-statistic: 4.163e+04 on 1 and 9699 DF,  p-value: < 2.2e-16
```

```
Analysis of Variance Table

Response: product_weight
                Df  Sum Sq Mean Sq F value    Pr(>F)
storage_issues   1 2643.62 2643.62   41625 < 2.2e-16 ***
Residuals     9699  615.98    0.06
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simple Linear Regression - Hypothesis Testing

Correlation Coefficient (ρ) Test:

Hypothesis:

Null Hypothesis(Ho): $\rho = 0$
Storage issue and product weight feature are independant

Alternate Hypothesis (Ha): $\rho \neq 0$
Storage issue and product weight feature are dependant

t-statistic (t*) : sqrt(n-2) (r_xy / sqrt(1-(r_xy) ^2))

Significance Level (alpha) : 5%

Decision Rule:

- **If |t*| > tcritical** or **p-value < α**: Reject Null Hypothesis (Ho)
- **If |t*| <= tcritical** or **p-value >= α**: Fail to reject Null Hypothesis (Ho)

Results:
Since **|t*| > tcritical**, we reject the null hypothesis and conclude there is a significant relationship between storage issues and product weight

Estimation of Pearson product moment correlation coefficient (ρ):

r_xy : 0.9005695
t_statistic: 204.023
t_critical: 1.9602

Confidence Interval for rho(ρ)

Fisher Z transformation (Z') 1.475225

The 95% confidence interval for the true correlation coefficient is:
0.6216 <= rho <= 0.6337

MLR w/o Quadratic & Interaction terms

Formula with all predictors:

```
#First Order MLR Model
fo_model <- lm(product_weight ~ location + capacity + zone + reg_zone + refill + transport_issue + competitor +
                                retail_shop + warehouse_Owner + distributors +
                                flood_impacted + flood_proof + electric_supply + distance_hub +
                                workers_num + w_est_year + storage_issues +
                                temperature_regulation + govt_cert +
                                warehouse_breakdown + govt_check , data = train_df)
```

Stepwise Model Selection (Both Direction):

- Combines forward selection and backward elimination search procedures to simply it to reduce overfitting

```
reg.null <- lm(product_weight ~ 1,data=train_df)
best_fo_model <- step(reg.null,direction="both",scope=list(upper= fo_model,lower=reg.null))
```

Summary:

```
Residual standard error: 0.1939 on 9662 degrees of freedom
Multiple R-squared:  0.8885,    Adjusted R-squared:  0.8881
F-statistic:  2027 on 38 and 9662 DF,  p-value: < 2.2e-16
```

Anova:

```
Analysis of Variance Table

Response: product_weight
                       Df  Sum Sq Mean Sq  F value    Pr(>F)
storage_issues          1 2643.62 2643.62 70300.4734 < 2.2e-16 ***
w_est_year             26  211.37    8.13   216.1888 < 2.2e-16 ***
govt_cert               4   31.00    7.75   206.0952 < 2.2e-16 ***
transport_issue         4    9.24    2.31    61.4514 < 2.2e-16 ***
temperature_regulation  1    0.84    0.84    22.2567 2.419e-06 ***
govt_check              1    0.11    0.11     3.0299   0.08177 .
location                1    0.08    0.08     2.1171   0.14570
Residuals            9662  363.33    0.04
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- Above Model explains 88.5% of the variance in product weight
- Adjusted R-square is approx. equal to Multiple R-square i.e. no irrelevant predictors in the model
- Residual standard error (0.1939) is very low and high degree of freedom (n-p) indicates lesser number of predictors fitted for the model

Extra Sum of Squares: Test for Regression Some Coefficients βk:

Hypothesis
Null Hypothesis(Ho): βgovt_check = βlocation = 0
Govt check and Location do not significantly improve the model

Alternate Hypothesis (Ha): βgovt_check ≠ βlocation ≠ 0
Govt check and Location significantly improve the model

F-statistic (F*) : $MSR(Xq,\ldots,Xp-1|X1,X2,\ldots,Xq-1)$/ MSE

Significance Level (alpha) : 5%

Decision Rule:

- **If |F*| > Fcritical** or **p-value < α**: Reject Null Hypothesis (Ho)
- **If |F*| <= Fcritical** or **p-value >= α**: Fail to reject Null Hypothesis (Ho)

Results:
Since p-value >= α, we **failed to reject the null hypothesis** and conclude govt check and location feature do not significantly improve the model

```
> anova(best_fo_model, best_fo_model_1)
Analysis of Variance Table

Model 1: product_weight ~ storage_issues + w_est_year + govt_cert + transport_issue +
    temperature_regulation + govt_check + location
Model 2: product_weight ~ storage_issues + w_est_year + govt_cert + transport_issue +
    temperature_regulation
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1   9662 363.33
2   9664 363.53 -2  -0.19355 2.5735 0.07632 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Best Subset MLR w/o Quadratic & Interaction terms

Formula:

```
best_fo_model_1 <- lm(product_weight ~ storage_issues + w_est_year + govt_cert + transport_issue +
  temperature_regulation , data = train_df)
```

Summary:

```
Residual standard error: 0.194 on 9664 degrees of freedom
Multiple R-squared:  0.8885,    Adjusted R-squared:  0.8881
```

Multicollinearity (VIF Values):

	GVIF	Df	GVIF^(1/(2*Df))
storage_issues	3.102658	1	1.761436
w_est_year	3.270324	26	1.023048
govt_cert	1.555713	4	1.056796
transport_issue	1.026658	4	1.003294
temperature_regulation	1.456053	1	1.206670

Anova:

```
Analysis of Variance Table

Response: product_weight
                       Df  Sum Sq Mean Sq   F value    Pr(>F)
storage_issues          1 2643.62 2643.62 70277.588 < 2.2e-16 ***
w_est_year             26  211.37    8.13   216.119 < 2.2e-16 ***
govt_cert               4   31.00    7.75   206.028 < 2.2e-16 ***
transport_issue         4    9.24    2.31    61.431 < 2.2e-16 ***
temperature_regulation  1    0.84    0.84    22.250 2.428e-06 ***
Residuals            9664  363.53    0.04
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

- All predictors significantly contribute to explaining the variation
- All predictors have low Variance Inflation Factor (VIF < 2), indicating no significant multicollinearity concerns

MLR with Quadratic & Interaction terms

Full Model:

```
full_model <- lm(product_weight ~ (location + capacity + zone + reg_zone + refill + transport_issue + competitor +
                                   retail_shop + warehouse_Owner + distributors +
                                   flood_impacted + flood_proof + electric_supply + distance_hub +
                                   workers_num + w_est_year + storage_issues +
                                   temperature_regulation + govt_cert +
                                   warehouse_breakdown + govt_check + retail_shop_retail_shop + distributors_distributors +
                                   distance_hub_distance_hub + workers_num_workers_num + storage_issues_storage_issues + govt_check_govt_check )^2 , data = train_df)
```

Stepwise Model Selection (Both Direction):

```
reg.null <- lm(product_weight ~ 1,data=train_df)
both_model <- step(reg.null,direction="both",scope=list(upper= full_model,lower=reg.null))
```

Anova:

Response: product_weight

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
storage_issues_storage_issues	1	2650.91	2650.91	79574.1698	< 2.2e-16	***
w_est_year	26	206.18	7.93	238.0435	< 2.2e-16	***
govt_cert	4	31.09	7.77	233.2781	< 2.2e-16	***
storage_issues	1	2.39	2.39	71.6751	< 2.2e-16	***
transport_issue	4	9.11	2.28	68.3745	< 2.2e-16	***
temperature_regulation	1	0.83	0.83	24.7662	6.585e-07	***
govt_check_govt_check	1	0.10	0.10	3.0268	0.081934	.
capacity	2	0.14	0.07	2.1353	0.118271	
location	1	0.08	0.08	2.5095	0.113194	
electric_supply	1	0.05	0.05	1.4428	0.229720	
storage_issues_storage_issues:w_est_year	26	33.68	1.30	38.8843	< 2.2e-16	***
w_est_year:storage_issues	26	4.04	0.16	4.6666	3.704e-14	***
govt_cert:storage_issues	4	0.49	0.12	3.6475	0.005655	**
storage_issues_storage_issues:storage_issues	1	0.18	0.18	5.5135	0.018890	*
temperature_regulation:govt_check_govt_check	1	0.10	0.10	3.0359	0.081474	.
storage_issues:temperature_regulation	1	0.10	0.10	2.9897	0.083828	.
storage_issues_storage_issues:temperature_regulation	1	0.10	0.10	2.9501	0.085903	.
storage_issues_storage_issues:capacity	2	0.18	0.09	2.7013	0.067170	.
govt_check_govt_check:electric_supply	1	0.09	0.09	2.8292	0.092598	.
capacity:electric_supply	2	0.18	0.09	2.6592	0.070056	.
Residuals	9593	319.58	0.03			

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

F-test for Interaction Terms

Interaction model:

```
interaction_model <- lm(product_weight ~ storage_issues_storage_issues + w_est_year +
                        govt_cert + storage_issues + transport_issue + temperature_regulation +
                        govt_check_govt_check + capacity + location + electric_supply +
                        storage_issues_storage_issues:w_est_year + w_est_year:storage_issues +
                        govt_cert:storage_issues + storage_issues_storage_issues:storage_issues +
                        temperature_regulation:govt_check_govt_check + storage_issues:temperature_regulation +
                        storage_issues_storage_issues:temperature_regulation + storage_issues_storage_issues:capacity +
                        govt_check_govt_check:electric_supply + capacity:electric_supply, train_df)
```

Additive Model:

```
additive_model <- lm(product_weight ~ storage_issues_storage_issues + w_est_year +
  govt_cert + storage_issues + transport_issue + temperature_regulation +
  govt_check_govt_check + capacity + location + electric_supply, data =train_df)
```

Hypothesis

Null Hypothesis(Ho): interaction terms coefficient = 0
Interaction terms does not add variation i.e. not improve the model

Alternate Hypothesis (Ha): interaction terms coefficient = 0≠ 0
Interaction terms does add variation i.e. improve the model

Significance Level (alpha) : 5%

Decision Rule:

- **If |F*| > Fcritical** or **p-value < α**: Reject Null Hypothesis (Ho)
- **If |F*| <= Fcritical** or **p-value >= α**: Fail to reject Null Hypothesis (Ho)

Results:

Since p-value < α, we **reject the null hypothesis** and conclude Interaction terms does add variation i.e. significantly improve the model

```
> anova(additive_model, both_model)
Analysis of Variance Table

Model 1: product_weight ~ storage_issues_storage_issues + w_est_year +
    govt_cert + storage_issues + transport_issue + temperature_regulation +
    govt_check_govt_check + capacity + location + electric_supply
Model 2: product_weight ~ storage_issues_storage_issues + w_est_year +
    govt_cert + storage_issues + transport_issue + temperature_regulation +
    govt_check_govt_check + capacity + location + electric_supply +
    storage_issues_storage_issues:w_est_year + w_est_year:storage_issues +
    govt_cert:storage_issues + storage_issues_storage_issues:storage_issues +
    temperature_regulation:govt_check_govt_check + storage_issues:temperature_regulation +
    storage_issues_storage_issues:temperature_regulation + storage_issues_storage_issues:capacity +
    govt_check_govt_check:electric_supply + capacity:electric_supply
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1   9658 358.72
2   9593 319.58 65    39.142 18.076 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Extra Sum of Squares: Test for Regression individual Coefficients βk:

We'll take Electric Supply feature and Test if it significant or not

Hypothesis

Null Hypothesis(Ho): βelectric_supply= 0

Electric supply do not significantly improve the model

Alternate Hypothesis (Ha): βelectric_supply ≠ 0

Electric supply significantly improve the model

F-statistic (F*) : $MSR(Xq|X1,X2,....,Xq-1)$/ MSE

Significance Level (alpha) : 5%

```
Analysis of Variance Table

Model 1: product_weight ~ storage_issues_storage_issues + w_est_year +
    govt_cert + storage_issues + transport_issue + temperature_regulation +
    govt_check_govt_check + capacity + location + electric_supply +
    storage_issues_storage_issues:w_est_year + w_est_year:storage_issues +
    govt_cert:storage_issues + storage_issues_storage_issues:storage_issues +
    temperature_regulation:govt_check_govt_check + storage_issues:temperature_regulation +
    storage_issues_storage_issues:temperature_regulation + storage_issues_storage_issues:capacity +
    govt_check_govt_check:electric_supply + capacity:electric_supply
Model 2: product_weight ~ storage_issues_storage_issues + w_est_year +
    govt_cert + storage_issues + transport_issue + temperature_regulation +
    govt_check_govt_check + capacity + location + storage_issues_storage_issues:w_est_year +
    w_est_year:storage_issues + govt_cert:storage_issues + storage_issues_storage_issues:storage_issues +
    temperature_regulation:govt_check_govt_check + storage_issues:temperature_regulation +
    storage_issues_storage_issues:temperature_regulation + storage_issues_storage_issues:capacity +
    govt_check_govt_check:electric_supply + capacity:electric_supply
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1   9593 319.58
2   9594 319.63 -1 -0.052248 1.5684 0.2105
```

Decision Rule:

- **If |F*| > Fcritical** or **p-value < α**: Reject Null Hypothesis (Ho)
- **If |F*| <= Fcritical** or **p-value >= α**: Fail to reject Null Hypothesis (Ho)

Results:

Since p-value >= α, we **failed to reject the null hypothesis** and conclude electric supply feature do not significantly improve the model

Best Subset MLR with Quadratic & Interaction terms

Formula:

```
updated_both_model <- lm(product_weight ~ storage_issues_storage_issues + w_est_year +
                         govt_cert + storage_issues + transport_issue + temperature_regulation +
                         storage_issues_storage_issues:w_est_year + w_est_year:storage_issues +
                         govt_cert:storage_issues + storage_issues_storage_issues:storage_issues , data =train_df)
```

Summary:

```
Residual standard error: 0.1827 on 9606 degrees of freedom
Multiple R-squared:  0.9016,    Adjusted R-squared:  0.9006
F-statistic: 936.2 on 94 and 9606 DF,  p-value: < 2.2e-16
```

Multicollinearity (VIF Values):

	GVIF	Df	GVIF^(1/(2*Df))
storage_issues_storage_issues	5.317517e+06	1	2305.974292
w_est_year	7.211596e+27	26	3.433455
govt_cert	1.725969e+00	4	1.070605
storage_issues	5.030673e+06	1	2242.916126
transport_issue	1.054377e+00	4	1.006641
temperature_regulation	1.480145e+00	1	1.216612
storage_issues_storage_issues:w_est_year	1.188174e+111	26	136.790326
w_est_year:storage_issues	1.216337e+111	26	136.851965
govt_cert:storage_issues	7.471416e+00	4	1.285806
storage_issues_storage_issues:storage_issues	4.319322e+02	1	20.782979

Anova:

```
> anova(updated_both_model)
Analysis of Variance Table

Response: product_weight
                                             Df  Sum Sq Mean Sq    F value    Pr(>F)
storage_issues_storage_issues                 1 2650.91 2650.91 79380.9178 < 2.2e-16 ***
w_est_year                                   26  206.18    7.93   237.4654 < 2.2e-16 ***
govt_cert                                     4   31.09    7.77   232.7116 < 2.2e-16 ***
storage_issues                                1    2.39    2.39    71.5010 < 2.2e-16 ***
transport_issue                               4    9.11    2.28    68.2085 < 2.2e-16 ***
temperature_regulation                        1    0.83    0.83    24.7061 6.793e-07 ***
storage_issues_storage_issues:w_est_year     26   33.65    1.29    38.7517 < 2.2e-16 ***
w_est_year:storage_issues                    26    4.00    0.15     4.6065 6.890e-14 ***
govt_cert:storage_issues                      4    0.48    0.12     3.5622  0.006567 **
storage_issues_storage_issues:storage_issues  1    0.18    0.18     5.4624  0.019450 *
Residuals                                  9606  320.79    0.03
```

- All predictors significantly contribute to explaining the variation
- Some predictors have very high Variance Inflation Factor (VIF > 2), indicating significant multicollinearity concerns

Multicollinearity

Iteration 0:

	GVIF	Df	GVIF^(1/(2*Df))
storage_issues_storage_issues	5.317517e+06	1	2305.974292
w_est_year	7.211596e+27	26	3.433455
govt_cert	1.725969e+00	4	1.070605
storage_issues	5.030673e+06	1	2242.916126
transport_issue	1.054377e+00	4	1.006641
temperature_regulation	1.480145e+00	1	1.216612
storage_issues_storage_issues:w_est_year	1.188174e+111	26	136.790326
w_est_year:storage_issues	1.216337e+111	26	136.851965
govt_cert:storage_issues	7.471416e+00	4	1.285806
storage_issues_storage_issues:storage_issues	4.319322e+02	1	20.782979

Iteration 1:

```
Residual standard error: 0.1827 on 9606 degrees of freedom
Multiple R-squared:  0.9016,    Adjusted R-squared:  0.9006
F-statistic: 936.2 on 94 and 9606 DF,  p-value: < 2.2e-16
```

consider setting type = 'predictor'; see ?vif

	GVIF	Df	GVIF^(1/(2*Df))
w_est_year	7.211596e+27	26	3.433455
govt_cert	1.725969e+00	4	1.070605
storage_issues	5.030673e+06	1	2242.916124
transport_issue	1.054377e+00	4	1.006641
temperature_regulation	1.480145e+00	1	1.216612
w_est_year:storage_issues_storage_issues	2.382228e+114	27	131.247876
w_est_year:storage_issues	1.216337e+111	26	136.851965
govt_cert:storage_issues	7.471416e+00	4	1.285806
storage_issues:storage_issues_storage_issues	4.319322e+02	1	20.782979

Best Subset MLR with Quadratic & Interaction terms after removing high VIF Values

Formula:

```
updated_both_model_mc2 <- lm(product_weight ~ w_est_year + govt_cert + storage_issues + transport_issue +
                             temperature_regulation, data =train_df)
```

Summary:

```
Residual standard error: 0.194 on 9664 degrees of freedom
Multiple R-squared:  0.8885,    Adjusted R-squared:  0.8881
F-statistic:  2139 on 36 and 9664 DF,  p-value: < 2.2e-16
```

Anova:

```
> anova(updated_both_model_mc2)
Analysis of Variance Table

Response: product_weight
                      Df  Sum Sq Mean Sq  F value    Pr(>F)
w_est_year            26 2546.34  97.936 2603.524 < 2.2e-16 ***
govt_cert              4   34.39   8.598  228.571 < 2.2e-16 ***
storage_issues         1  305.26 305.255 8114.867 < 2.2e-16 ***
transport_issue        4    9.24   2.311   61.431 < 2.2e-16 ***
temperature_regulation 1    0.84   0.837   22.250 2.428e-06 ***
Residuals           9664  363.53   0.038
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

VIF:

```
> vif(updated_both_model_mc2)
                             GVIF Df GVIF^(1/(2*Df))
w_est_year               3.270324 26        1.023048
govt_cert                1.555713  4        1.056796
storage_issues           3.102658  1        1.761436
transport_issue          1.026658  4        1.003294
temperature_regulation   1.456053  1        1.206670
```

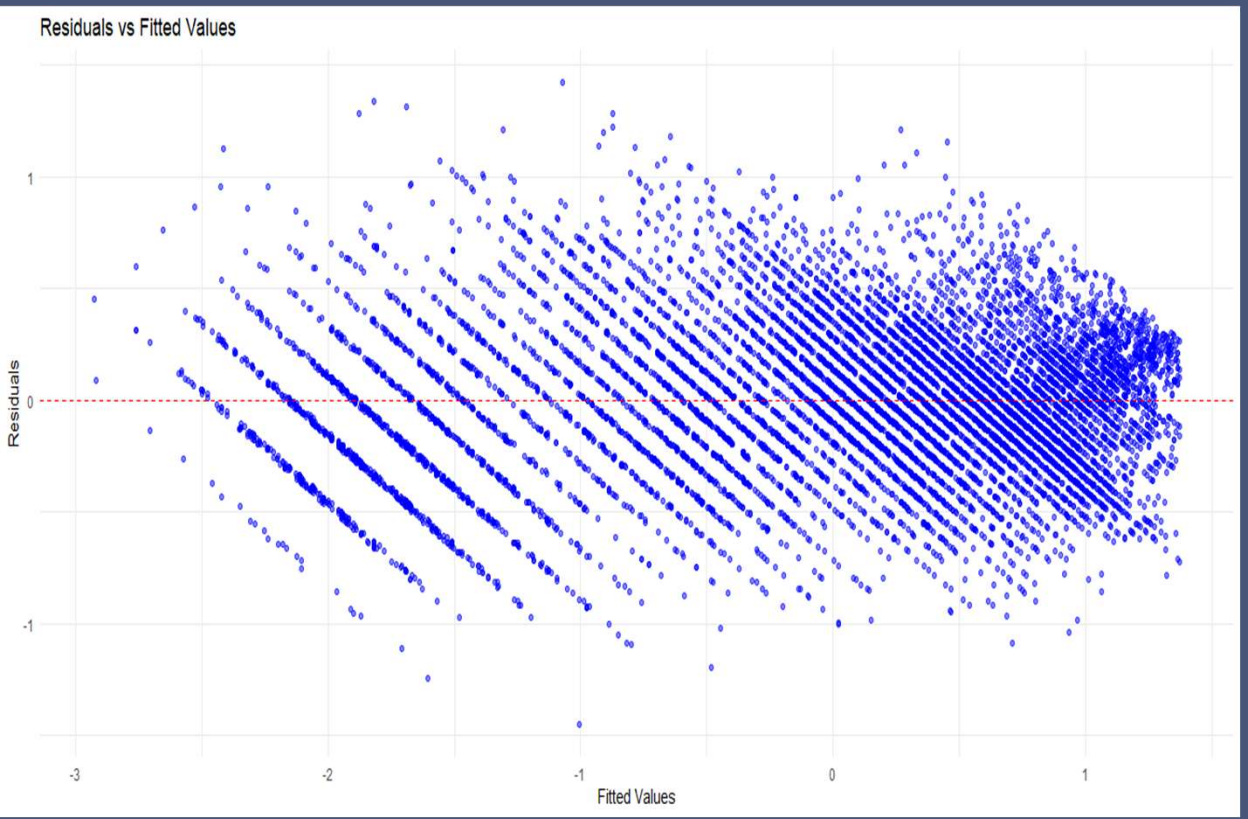
Model Comparison

Simple Linear Regression (SLR)	MLR w/o I&Q	MLR with I&Q
• BIC : 814.3759 • Adjusted R-square : 81.1%	• BIC : -3980.228 • Adjusted R-square : 88.81%	• BIC : -3980.228 • Adjusted R-square : 89.76%

Considering above values of each subset models, the best subset model is

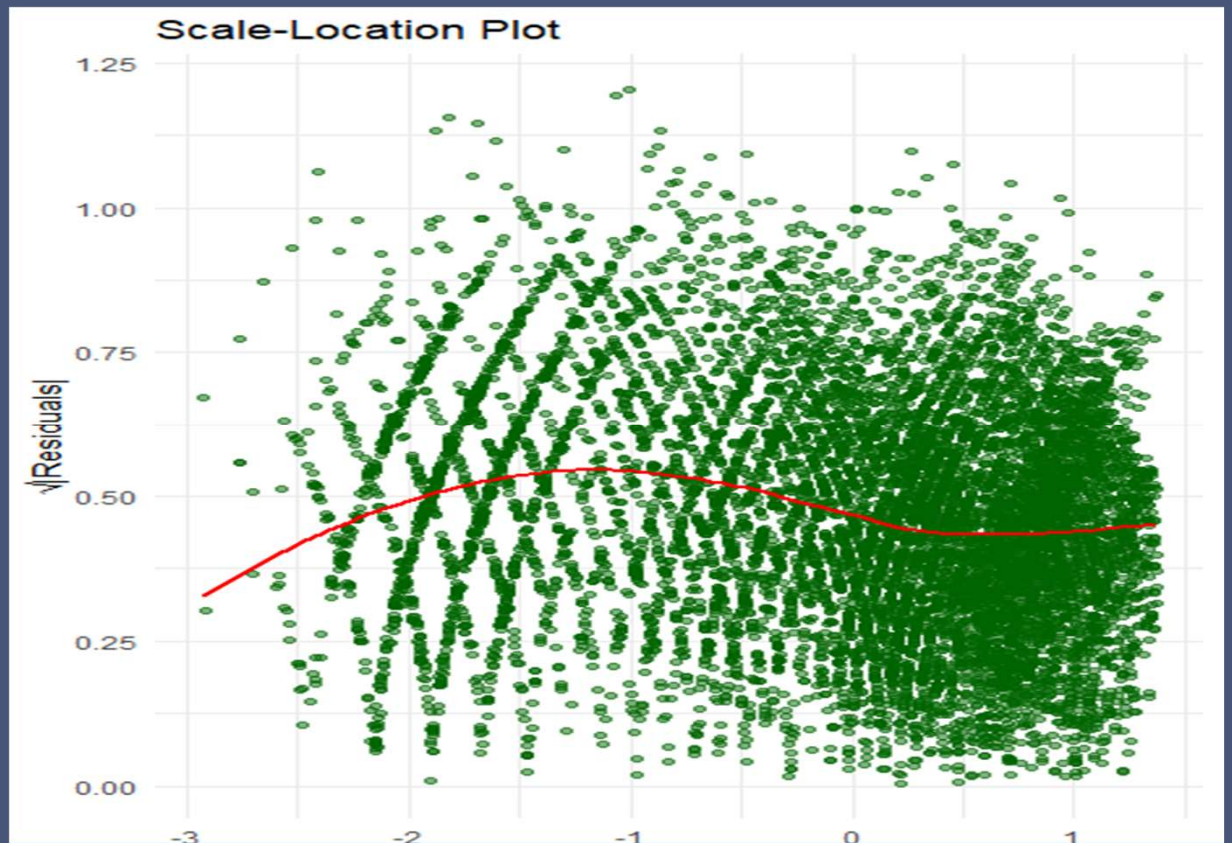
MLR with I&Q

Fitted vs residuals and test for lack of fitness



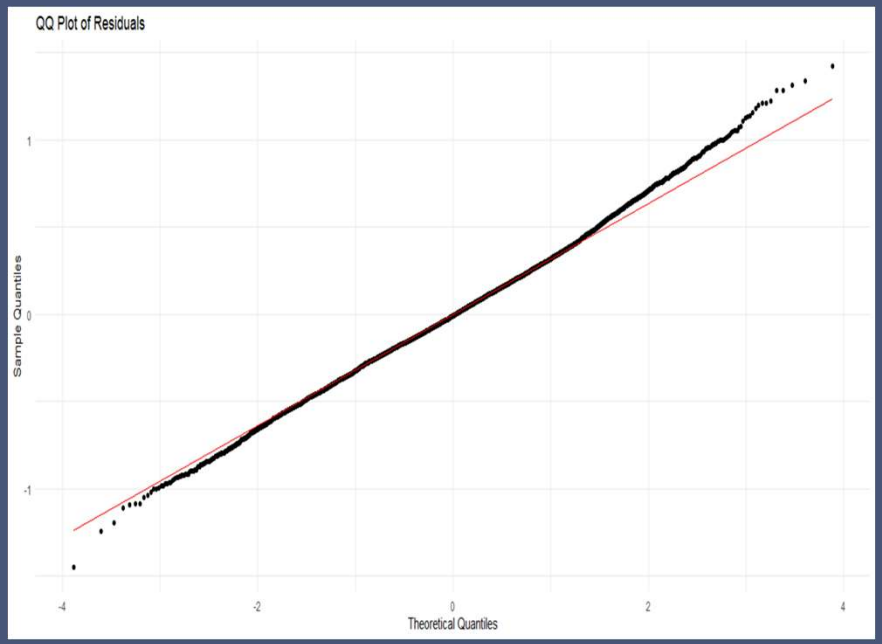

```
> cat("F-statistic for Lack-of-Fit:", F_stat, "\n")
F-statistic for Lack-of-Fit: 0.3460874
> cat("p-value for Lack-of-Fit:", p_value, "\n")
p-value for Lack-of-Fit: 1
> cat("Degrees of Freedom (LOF):", df_LOF, "\n")
Degrees of Freedom (LOF): 5574
> cat("Degrees of Freedom (Pure Error):", df_PE, "\n")
Degrees of Freedom (Pure Error): 4090
```

Scale-location and Brown test



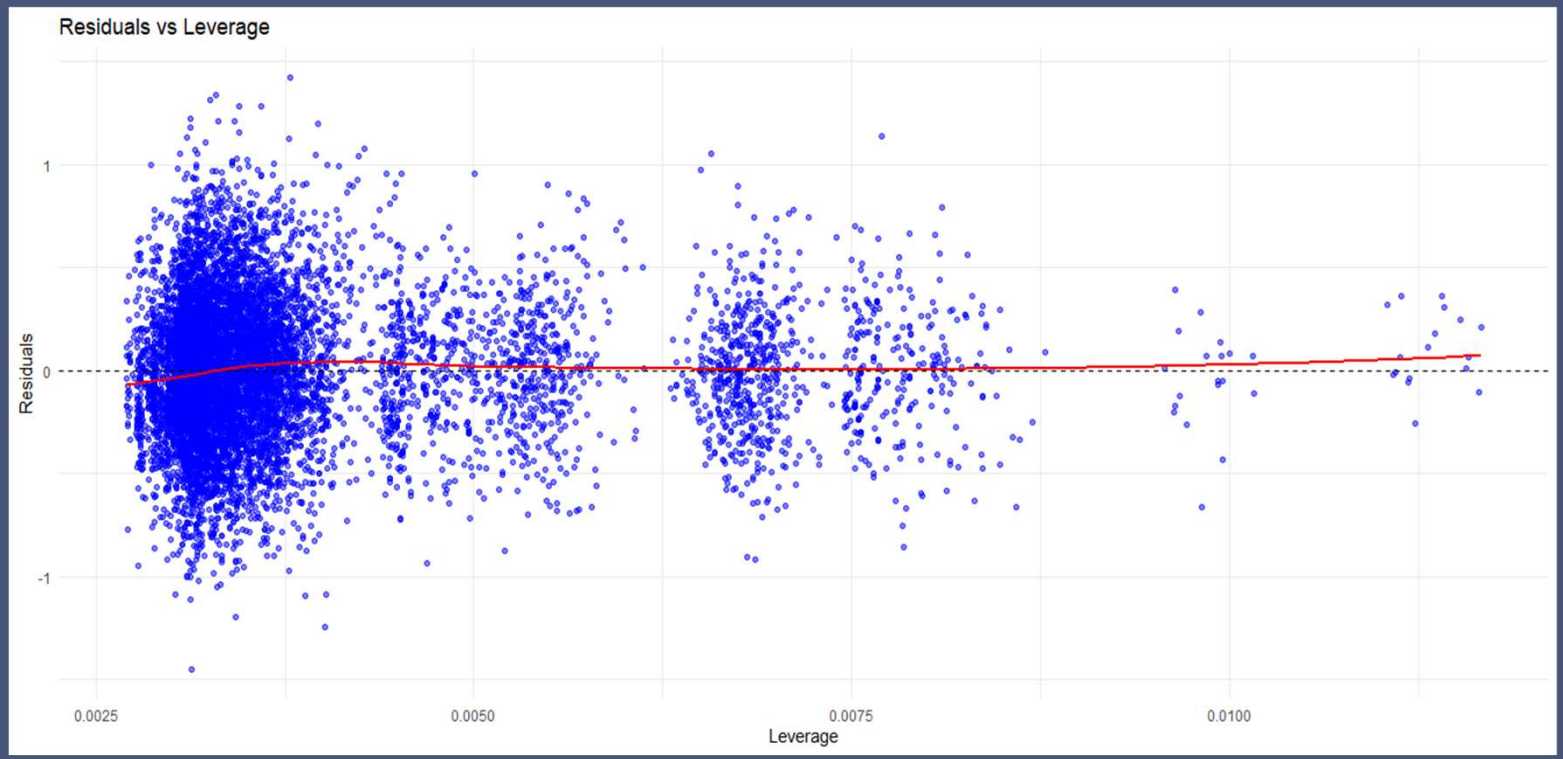

```
Brown-Forsythe Test Results:
> cat("F-statistic:", bf_test$statistic, "\n")
F-statistic: 44.79737
> cat("Degrees of Freedom:", bf_test$parameter, "
Degrees of Freedom: 3 5347.542
> cat("p-value:", bf_test$p.value, "\n")
p-value: 1.386091e-28
```

QQ plot




```
Shapiro-Wilk Test Results:
> cat("W-statistic:", shapiro_test$statistic, "\n")
W-statistic: 0.9972575
> cat("p-value:", shapiro_test$p.value, "\n")
p-value: 7.104407e-08
```

Residuals vs leverage



Outlier Analysis

- Outlying on X-direction
- Outlying on Y-direction

Outlying on X-direction

Leverage

```
> print(leverage_values[1:100])
      2463       2511      10419       8718       2986       1842       9334       3371      11638       4761       6746       9819       2757       5107
0.004860591 0.005216644 0.003292527 0.003207057 0.003798827 0.003868538 0.002924252 0.003571887 0.003126874 0.003387086 0.003947998 0.003103695 0.002929928 0.003344535
      9145       9209      10205       2888       6170       2567       9642       9982       2980       1614        555       4469       9359      10784
0.003309296 0.003657003 0.003113874 0.003161913 0.003064277 0.003726881 0.003870662 0.003308809 0.003051036 0.003147204 0.003072748 0.005160794 0.007113160 0.003291134
     10730       7789       9991       9097       1047       7067       3004       3207       7989       3995       8358        217       9506       8157
0.003050264 0.002979499 0.003322366 0.003402535 0.003808636 0.003424650 0.004005103 0.005308909 0.003066913 0.003180115 0.003143815 0.003015472 0.002873995 0.003376031
     10821       6216       8780       1599       4237       3937       4089       2907        294       8469         41       8508       7391       6672
0.003017502 0.003289932 0.003406746 0.003124626 0.003677307 0.003217329 0.004724301 0.004489622 0.003227021 0.003160906 0.003133393 0.004598362 0.003082566 0.003592935
      7284      11014      10987       2504       6742       9375       8944      11473       8566      10034       6129      10274       4612       2117
0.003720002 0.002727357 0.007149746 0.006963385 0.003286767 0.003428946 0.006496545 0.003533157 0.003309420 0.005541308 0.003202771 0.002883122 0.003316257 0.003218690
      6134        755       6553       5428       9198      10777       7127      10531       9640       3358       3980       9326       3230       5603
0.004525913 0.005442854 0.004090316 0.003166183 0.003539017 0.003496599 0.004022285 0.003434569 0.003175163 0.003329090 0.003045571 0.003515962 0.003285182 0.005469338
     10126       9693       4576       3783       7831      10106       5967       9301       7816       9267      11338       1386      10476       4706
0.003508819 0.003196796 0.004028936 0.003197762 0.003179434 0.003128139 0.003751731 0.002966690 0.003726166 0.003446045 0.003227876 0.003285304 0.003187171 0.003232885
      2378       4044
0.002962967 0.003307305
```

Threshold:

```
> print(threshold)
[1] 0.00762808
```

Outlying Points:

```
> print(high_leverage[1:100])
  712  5509  2432  3082  9587 10054   193 10307  7274  1352  9174  4494 10719   703  1722  3101  5868  1406  2558 12117  1762  5988  8412  9713   216  8284  1427 10247  4307
  165   240   353   406   436   448   473   569   638   687   711   759   802   919   948  1021  1025  1179  1200  1328  1471  1527  1529  1554  1582  1591  1611  1617  1642
 6425  4023  5466 10482  9386  7956  8903  3909 10558  4455  9947  9233  4621  1158  5455 11332  7965  8137  2416  5019  4931  6076 12052  8096  3058 11408 11201  5012  8495
 1680  1741  1753  1759  1767  1849  1978  2024  2040  2077  2201  2297  2327  2352  2403  2417  2449  2479  2493  2555  2664  2709  2740  2792  2795  2863  2874  2881  3087
 3486 10072  9036   326  5131 11815  1000  9284  1334   559  8668  6227   345   229  3819 11279 11669  9738  6073  1009  7553 11776 11931  4796  9538  9992  4208  6101   708
 3112  3162  3222  3278  3330  3331  3352  3381  3423  3433  3558  3627  3660  3685  3724  3744  3756  3776  3837  3839  3860  3884  3918  3919  3948  4101  4186  4207  4225
10779  9290  2282  3131  1669   918 11275  4123  7793  3266  1637  2301 11690
 4226  4267  4274  4376  4531  4568  4578  4587  4598  4738  4750  4754  4803
>
```

Outlying on Y-direction

Studentize Residuals

```
> print(studentized_residuals[1:100])
        2463         2511        10419         8718         2986         1842         9334         3371        11638         4761         6746         9819
-1.3213784151 -0.8839438763  0.3864978053 -0.5032369112 -0.1229676465  0.6453171514 -0.4485389882 -1.3679537281 -1.0174509299  0.4084843838 -0.3821674860  0.0468643263
        2757         5107         9145         9209        10205         2888         6170         2567         9642         9982         2980         1614
-0.5459484573  0.0497590060  0.6605877238 -0.1700320428 -1.3069970709  1.1053065008  1.8605833235  1.3284268063 -0.8934353609  0.3575324748  0.8505842214 -0.7900156494
         555         4469         9359        10784        10730         7789         9991         9097         1047         7067         3004         3207
 0.2155256095 -0.3561155309  1.0505414767  0.8291727564  1.0166072157  0.3386150569  0.1624547773 -1.1024177200  0.4636905247  0.2183443201  0.2157156720  1.1685221030
        7989         3995         8358          217         9506         8157        10821         6216         8780         1599         4237         3937
-0.5891205071  1.3582598571  0.7935224888  0.3165441788 -0.1007427391  0.1500145533  0.2468846062  0.0166427516 -1.5350934691  0.4329459814  0.3613733109 -0.3469576487
        4089         2907          294         8469           41         8508         7391         6672         7284        11014        10987         2504
-0.7374177632  0.0109627850 -1.4315935145 -0.5758344334  0.1454599847  0.4389783889  0.7409180324  0.6924888798 -0.3154100733  0.2557759876  1.7235488801 -0.4226569528
        6742         9375         8944        11473         8566        10034         6129        10274         4612         2117         6134          755
 1.1973899955 -0.4482049681 -0.7546016280  1.4102138793  0.6307288217  0.9691016979 -0.4800378600 -0.7409239246 -0.8758024598 -0.0953140428 -1.1769105687  0.2601489402
        6553         5428         9198        10777         7127        10531         9640         3358         3980         9326         3230         5603
 0.0969714835  0.4522659344  2.4213260595  0.3650171977 -0.1090873877 -0.8096437963  1.0701481116 -0.6343828241  0.7972852816  0.2625880644  0.9357530232  0.7932115990
       10126         9693         4576         3783         7831        10106         5967         9301         7816         9267        11338         1386
-0.4311229977 -0.1427828576  0.2750784063 -0.2270429758  1.7332889929  0.3679150858 -0.4141947779 -0.0007272014  1.3093970252  0.9346213078 -0.8022297108  0.3607837786
       10476         4706         2378         4044
 0.3844645693  0.4301999427 -0.6697274751 -0.5933508772
```

Interpretation: y_outliers <- which(abs(studentized_residuals) > 3)

```
> cat("Outliers on Y-direction", y_outliers, "\n")
Outliers on Y-direction 122 219 259 532 1101 1301 1470 1791 2610 2617 2797 3197 3547 3677 3678 3934 3997 4023 4389 4481 5293 5331 6034 6075 6377 6682 6997 7380 7415 7788 7837 8048
8064 8376 8386 8596 8665 9589
```

Influential Cases Detection

- DEFITTS
- Cook’s Distance
- DFBETAS

DFFITS

It measures the influence of the observations in the predicted variable

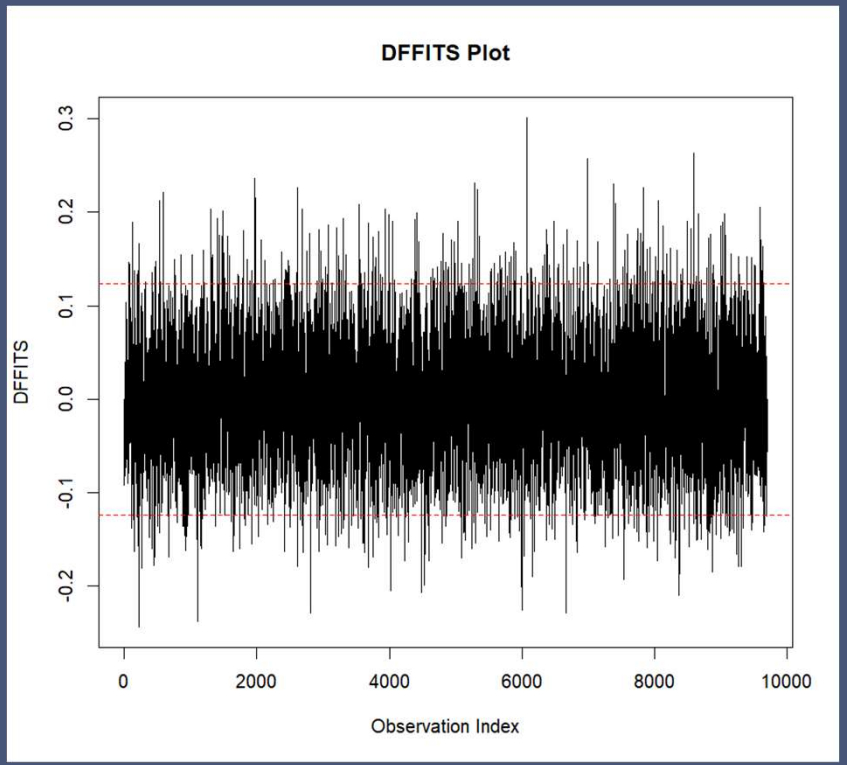
```
> print(dfbetas_values[1:100])
  [1] -1.030405e-02  1.147424e-03 -8.627184e-04  1.219008e-03  5.703592e-04  8.747898e-04 -7.955768e-05  4.584525e-04 -6.052667e-04  8.058309e-04  1.778491e-03 -6.860254e-05
 [13]  7.684681e-04 -1.260949e-04  4.674512e-03 -5.209136e-04  1.539730e-03 -1.302435e-03  1.066934e-02 -5.089646e-03  3.947268e-03  8.353963e-04 -3.751487e-04 -1.651833e-03
 [25]  1.157419e-03  1.645887e-03  5.573523e-03  2.769231e-03 -3.676081e-04  4.212894e-05 -7.019775e-05 -2.518104e-03  1.549864e-04  5.074593e-04  3.925665e-04  4.982940e-03
 [37] -3.055144e-03  1.918902e-03 -6.740058e-04 -2.968705e-04  1.166025e-04 -3.579917e-04 -4.643323e-04 -9.989668e-06  5.491319e-03 -4.153989e-04 -7.088631e-04  7.023354e-04
 [49]  2.149346e-03 -1.365677e-05 -1.833996e-03 -1.974604e-04 -7.239931e-05 -1.150435e-03  4.049627e-03 -6.643864e-04 -1.172176e-03 -3.341852e-04  2.005666e-03  8.976523e-04
 [61] -2.961203e-03  1.692448e-03  6.230190e-04  3.968820e-03  9.868678e-04 -3.289777e-04  8.436896e-04  7.381254e-05 -3.513929e-03  1.170829e-04  3.717003e-04 -1.081901e-03
 [73] -5.911531e-04 -5.927768e-04 -3.496305e-04 -9.992058e-04  4.113214e-04 -3.525497e-03  2.436848e-03  1.242111e-03 -1.187887e-03  6.850201e-04 -2.236647e-03 -3.411274e-03
 [85]  4.875427e-04  3.062334e-04  6.988477e-04 -8.825309e-04  1.495734e-03 -7.120717e-04  7.193679e-04 -4.447155e-06  6.196517e-04 -2.466676e-03 -5.890988e-03  6.329622e-04
 [97]  4.463558e-04 -7.512411e-04  3.045642e-04 -5.563699e-04
```

Cutoff value:

```
> print(cutoff)
[1] 0.1235158
```

Influential points:

```
> print(influential[1:100])
10987  9198  6387  8011  4685  8536 11067   185  7933  3799  3581  1905  3625  9617 11360
   59    75   105   122   135   144   155   192   193   203   206   219   224   259   319
 4357  1948  4374  7727 10054  1557  9071  7042 11187  7392  1498  3556  2788  2004 11032
  376   394   414   418   448   454   466   476   499   530   532   539   580   584   663
 4184  2801 11525  9185  1963 10719  7994  8601  4244 10915  1807  2082  4338  4311  5656
  673   755   771   793   796   802   853   881   886   890   905   927   933   947  1028
 5317  5799  3851 10118  2244  2387  6689  4312  5883 10645 12117  2705  8870  9496  9471
 1096  1101  1154  1162  1183  1191  1209  1301  1317  1320  1328  1336  1379  1401  1416
11543  6195  1611  6880 11364  2586  8806 10247  2908  5222  6927   810  9595 11851 12102
 1429  1470  1487  1496  1510  1563  1589  1617  1641  1660  1663  1708  1730  1738  1778
10021  5386   743  4879  1984  7753  3749  8371  6819  3645  1781  8903  4405  5301  3215
 1791  1806  1811  1823  1828  1850  1897  1917  1918  1965  1967  1978  2022  2065  2127
10211  5875  9063  7754 10998 10311  3023  9349 12025  4886
 2167  2245  2252  2268  2374  2413  2416  2421  2428  2435
```



Cook’s Distance

Measures the influence of the observations in overall model.

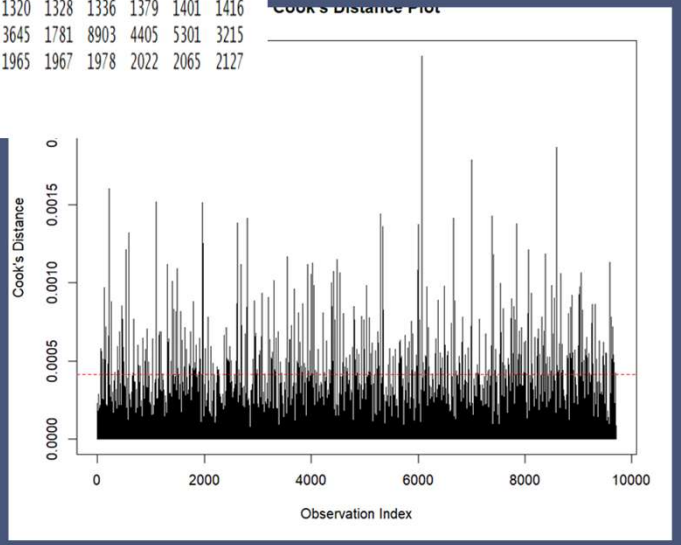
```
> print(cooks_d[1:100])
          2463           2511          10419           8718           2986           1842           9334           3371          11638           4761           6746           9819           2757
2.304753e-04 1.107440e-04 1.333804e-05 2.202311e-05 1.558572e-06 4.371204e-05 1.594855e-05 1.812816e-04 8.775950e-05 1.532803e-05 1.564729e-05 1.848234e-07 2.367361e-05
          5107           9145           9209          10205           2888           6170           2567           9642           9982           2980           1614            555           4469
2.245831e-07 3.916155e-05 2.868269e-06 1.442020e-04 1.047319e-04 2.875059e-04 1.784047e-04 8.383072e-05 1.147043e-05 5.984388e-05 5.325745e-05 3.869924e-06 1.778210e-05
          9359          10784          10730           7789           9991           9097           1047           7067           3004           3207           7989           3995           8358
2.136893e-04 6.135918e-05 8.546087e-05 9.261679e-06 2.377933e-06 1.121408e-04 2.221860e-05 4.428238e-06 5.057794e-06 1.969576e-04 2.885834e-05 1.590568e-04 5.367325e-05
           217           9506           8157          10821           6216           8780           1599           4237           3937           4089           2907            294           8469
8.191704e-06 7.906900e-07 2.060554e-06 4.986416e-06 2.471221e-08 2.176851e-04 1.588032e-05 1.302805e-05 1.050233e-05 6.976541e-05 1.465038e-08 1.793065e-04 2.841909e-05
            41           8508           7391           6672           7284          11014          10987           2504           6742           9375           8944          11473           8566
1.797658e-06 2.406166e-05 4.587880e-05 4.673689e-05 1.004042e-05 4.836018e-06 5.780479e-04 3.385839e-05 1.277759e-04 1.868271e-05 1.006390e-04 1.905566e-04 3.570280e-05
          10034           6129          10274           4612           2117           6134            755           6553           5428           9198          10777           7127          10531
1.414377e-04 2.001257e-05 4.290252e-05 6.897821e-05 7.929318e-07 1.701940e-04 1.001109e-05 1.043921e-06 1.756042e-05 5.624821e-04 1.263662e-05 1.299017e-06 6.106160e-05
          9640           3358           3980           9326           3230           5603          10126           9693           4576           3783           7831          10106           5967
9.858861e-05 3.633304e-05 5.248505e-05 6.576033e-06 7.800364e-05 9.352111e-05 1.768988e-05 1.767258e-06 8.273638e-06 4.469862e-06 2.589304e-04 1.148098e-05 1.746260e-05
          9301           7816           9267          11338           1386          10476           4706           2378           4044
4.253194e-11 1.732975e-04 8.163858e-05 5.632908e-05 1.159674e-05 1.277441e-05 1.622453e-05 3.602756e-05 3.157647e-05
```

Threshold:

```
> threshold <- 4 / (n - k)
> print(threshold)
[1] 0.1111111
```

Influential Points:

```
> print(influential[1:100])
10987  9198  6387  8011  4685  8536 11067   185  7933  3799  3581  1905  3625  9617 11360  4357  1948  4374  7727 10054  1557  9071  7042 11187  7392  1498  3556  2788  2004 11032
   59    75   105   122   135   144   155   192   193   203   206   219   224   259   319   376   394   414   418   448   454   466   476   499   530   532   539   580   584   663
 4184  2801 11525  9185  1963 10719  7994  8601  4244 10915  1807  2082  4338  4311  5656  5317  5799  3851 10118  2244  2387  6689  4312  5883 10645 12117  2705  8870  9496  9471
  673   755   771   793   796   802   853   881   886   890   905   927   933   947  1028  1096  1101  1154  1162  1183  1191  1209  1301  1317  1320  1328  1336  1379  1401  1416
11543  6195  1611  6880 11364  2586  8806 10247  2908  5222  6927   810  9595 11851 12102 10021  5386   743  4879  1984  7753  3749  8371  6819  3645  1781  8903  4405  5301  3215
 1429  1470  1487  1496  1510  1563  1589  1617  1641  1660  1663  1708  1730  1738  1778  1791  1806  1811  1823  1828  1850  1897  1917  1918  1965  1967  1978  2022  2065  2127
10211  5875  9063  7754 10998 10311  3023  9349 12025  4886
 2167  2245  2252  2268  2374  2413  2416  2421  2428  2435
```

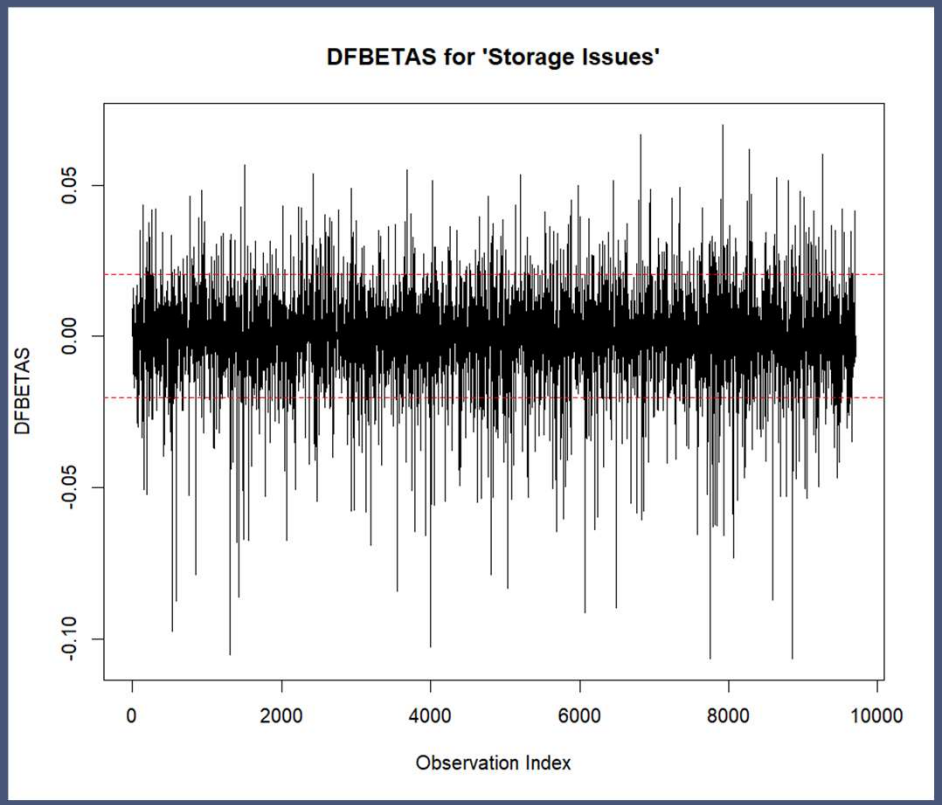


DFBETAS

Analyzes how individual observations affect specific regression coefficients.

```
$storage_issues
10987 11473  9198  6387  8011  4685  9985  8536 11067  2037  2313  6379   185   413
   59    64    75   105   122   135   139   144   155   179   180   190   192   194
 3581  3201  1905  6815  9637  5793  4213  4807  9426  9617  2470 10748  5214 11360
  206   216   219   228   232   236   241   256   258   259   275   276   307   319
 4357 10078  7727  8124  5423  7880  7042  3318  1517  8812 11054  7392  1498 11287
  376   403   418   419   423   426   476   497   512   521   525   530   532   537
 1831  4546 11528  2004   606   602  7032 11032  7639  8778  1352 11132  2801 11525
  546   557   564   584   586   618   643   663   671   676   687   726   755   771
 6483  7575  5003  7994  8601  7601  2082  4338  6736  4311  4812 11205  7053  3149
  790   805   821   853   881   923   927   933   939   947   959   962   964   967
 5278 11657  7319  9020  2970  5317  5799  8558 11310  8390   956  9027  2306  2437
 1030  1036  1040  1059  1085  1096  1101  1128  1130  1146  1155  1163  1171  1185
 7324  2287  3822  4312  1899  1003  9336  5216  5883 10645  1571  8870  9496 11543
 1192  1199  1222  1301  1304  1307  1308  1314  1317  1320  1343  1379  1401  1429
 5664  4815  9315  6195  5736 10623  1611  6880 11364  2745   572  6123  8684  2586
 1451  1453  1465  1470  1478  1484  1487  1496  1510  1519  1521  1541  1546  1563
10030  7242 11485  1761 10496  2908   793  5222 10147  2229 10718  7531 12102 11228
 1577  1585  1593  1613  1618  1641  1653  1660  1687  1692  1697  1760  1778  1782
 5386  4879  4376  2871  1887  2523  7419  3919  1701  8371  6819  7139  2873  9170
 1806  1823  1839  1845  1848  1869  1877  1881  1910  1917  1918  1948  1950  1956
 1781   774  6370  4405  9940  2021  6001  2173  5301  4001  1007  3295  7285  9124
 1967  1993  2021  2022  2023  2036  2046  2048  2065  2067  2080  2082  2105  2117
 3215   474  6143  6606  5961  1964  4725 10322  7754  2195 11995  4010  7868  9356
```

```
> #threshold
> n <- nrow(train_df)  # Number of observations
> threshold <- 2 / sqrt(n)
> print(threshold)
[1] 0.02030588
>
```


DFBETAS for 'Storage Issues'
DFBETAS
0.05
0.00
-0.05
-0.10
0
2000
4000
6000
8000
10000
Observation Index

Model Evaluation

Train-Test Split:

```
> R_squared <- 1 - (SS_Residual / SS_Total)
> R_squared
[1] 0.8868954
```

The MSPR value and R-square value of test data is 0.03829644 and 0.8868954

The ratio of MSPR and MSE is: 1.021966

K-cross Validation:

```
Linear Regression

12127 samples
    5 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 9701, 9700, 9702, 9702, 9703
Resampling results:

  RMSE       Rsquared   MAE
  0.1947053  0.8874002  0.1530196
```

Inference In Regression Analysis

$$b_k - s(b_k)t\left(1-\frac{\alpha}{2};n-p\right) \leq \beta_k \leq b_k + s(b_k)t\left(1-\frac{\alpha}{2};n-p\right)$$

95 % Confidence Interval for βk of MLR Model

```
> print(conf_intervals)
                                Coefficient     Estimate  Lower_95_CI   Upper_95_CI
(Intercept)                     (Intercept) 10.257407831 10.224011222 10.2908044400
w_est_year1997                w_est_year1997 -0.004355300 -0.044139953  0.0354293534
w_est_year1998                w_est_year1998  0.008010816 -0.029095884  0.0451175159
w_est_year1999                w_est_year1999  0.012617894 -0.024596558  0.0498323460
w_est_year2000                w_est_year2000  0.016378655 -0.020238453  0.0529957629
w_est_year2001                w_est_year2001  0.001709980 -0.035414332  0.0388342922
w_est_year2002                w_est_year2002 -0.006838073 -0.043855797  0.0301796507
w_est_year2003                w_est_year2003 -0.003757591 -0.041205335  0.0336901537
w_est_year2004                w_est_year2004  0.020890102 -0.016255381  0.0580355851
w_est_year2005                w_est_year2005  0.003440587 -0.034220861  0.0411020346
w_est_year2006                w_est_year2006 -0.035805483 -0.072602676  0.0009917103
w_est_year2007                w_est_year2007 -0.111135743 -0.148102476 -0.0741690097
w_est_year2008                w_est_year2008 -0.162147227 -0.199687966 -0.1246064892
w_est_year2009                w_est_year2009 -0.182804207 -0.220530478 -0.1450779363
w_est_year2010                w_est_year2010 -0.177422926 -0.214791284 -0.1400545687
w_est_year2011                w_est_year2011 -0.175330558 -0.212822288 -0.1378388274
w_est_year2012                w_est_year2012 -0.241695438 -0.279178942 -0.2042119343
w_est_year2013                w_est_year2013 -0.297396082 -0.335353959 -0.2594382051
w_est_year2014                w_est_year2014 -0.342639799 -0.380660858 -0.3046187399
w_est_year2015                w_est_year2015 -0.340294635 -0.378514590 -0.3020746807
w_est_year2016                w_est_year2016 -0.414797171 -0.453286804 -0.3763075377
w_est_year2017                w_est_year2017 -0.502793043 -0.541835121 -0.4637509648
w_est_year2018                w_est_year2018 -0.608203669 -0.647995597 -0.5684117402
w_est_year2019                w_est_year2019 -0.701686722 -0.741718000 -0.6616554440
w_est_year2020                w_est_year2020 -0.752543095 -0.793526154 -0.7115600359
w_est_year2021                w_est_year2021 -0.817342081 -0.861098411 -0.7735857518
w_est_year2022                w_est_year2022 -0.849939096 -0.898736501 -0.8011416917
govt_certA+                      govt_certA+  0.011089449 -0.003573829  0.0257527279
govt_certB                       govt_certB -0.117898904 -0.130128683 -0.1056691257
govt_certB+                      govt_certB+ -0.105981699 -0.118264515 -0.0936988836
govt_certC                       govt_certC -0.036786122 -0.048966769 -0.0246054744
storage_issues                storage_issues  0.305306369  0.298504182  0.3121085572
transport_issue1            transport_issue1 -0.044119312 -0.054515531 -0.0337230921
transport_issue2            transport_issue2 -0.074750500 -0.093833292 -0.0556677078
transport_issue3            transport_issue3 -0.124880136 -0.148633758 -0.1011265132
transport_issue4            transport_issue4 -0.097039647 -0.120795679 -0.0732836148
temperature_regulation1 temperature_regulation1  0.022787837  0.013317928  0.0322577454
```

95 % Confidence Interval for E(Xh)

$$\hat{Y}_h - s(\hat{Y}_h)t\left(1-\frac{\alpha}{2};n-p\right) \leq E(X_h) \leq \hat{Y}_h + s(\hat{Y}_h)t\left(1-\frac{\alpha}{2};n-p\right)$$

```
The 95% confidence interval for E(X_h) is:
 8.7814 <= E(X_h) <= 8.7917
```

Prediction Interval for New Observation $Yh(new)$

$$\hat{Y}_h - \text{s(pred)}\, t\left(1-\frac{\alpha}{2}; n-p\right) \leq \boldsymbol{Y_{h(new)}} \leq \hat{Y}_h + \text{s(pred)}\, t\left(1-\frac{\alpha}{2}; n-p\right)$$

```
The 95% prediction interval for New Observation is:
 8.4054 <= Y_h_new <= 9.1677
```

$$\hat{Y}_h - \text{s(meanpred)}\, t\left(1-\frac{\alpha}{2}; n-p\right) \leq \textbf{mean of m obsr.} \leq \hat{Y}_h + \text{s(meanpred)}\, t\left(1-\frac{\alpha}{2}; n-p\right)$$

```
The 95% prediction interval for New Observation is:
 8.7598 <= mean of m obsr <= 8.8133
```

Confidence of Region (CR) for Regression Surface

$$\hat{Y}_h - Ws(\hat{Y}_h) \leq \text{CR at } X_h \leq \hat{Y}_h + Ws(\hat{Y}_h)$$

```
The 95% confidence interval for E(X_h) is:
 8.7814 <= E(X_h) <= 8.7917
```

Conclusion

- The regression model shows strong performance with an R-squared value of 0.8885, explaining 88.85% of the variation in product weight
- The model performs consistently, as indicated by the MSPR/MSE of 1.021966
- All the confidence intervals are narrow that shows key coefficients reflects reliable and stable population estimates
- The prediction interval for new observations is precise, indicating robustness in model predictions

Recommendations

- Dimensionality reduction: Use Principal Component Analysis to transform highly correlated predictors into orthogonal components to reduce multicollinearity
- Using optimization algorithms like weighted linear regression or Elastic Net Regression, combining Lasso L1 and Ridge L2 penalties to balance feature selection and coefficient shrinkage
- Future Validation: Regularly validate the model with updated data to ensure its robustness and adaptability to changing patterns in production and operations

References

Montgomery, D. C., Peck, E. A., & Vining, G. G. (2021). Introduction to linear regression analysis (6th ed.). Wiley.

Kutner, M. H., Nachtsheim, C. J., & Neter, J. (2004). Applied linear statistical models (5th ed.). McGraw-Hill Education.