Task1

Hoda

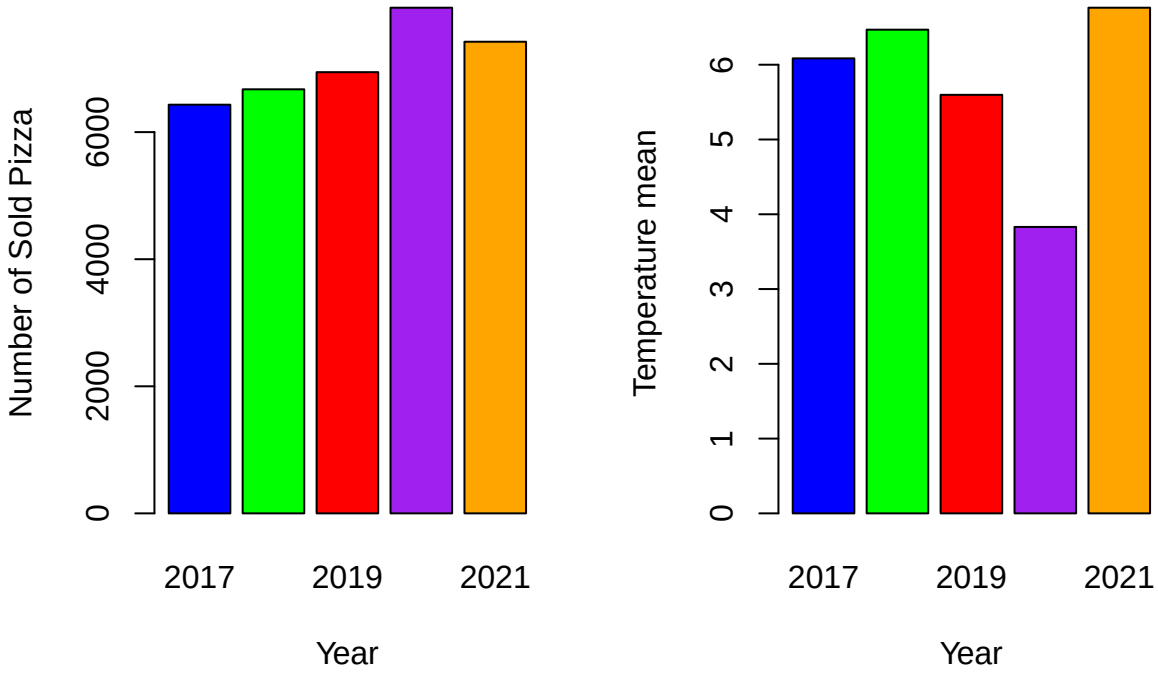
2023-01-22

TASK 1

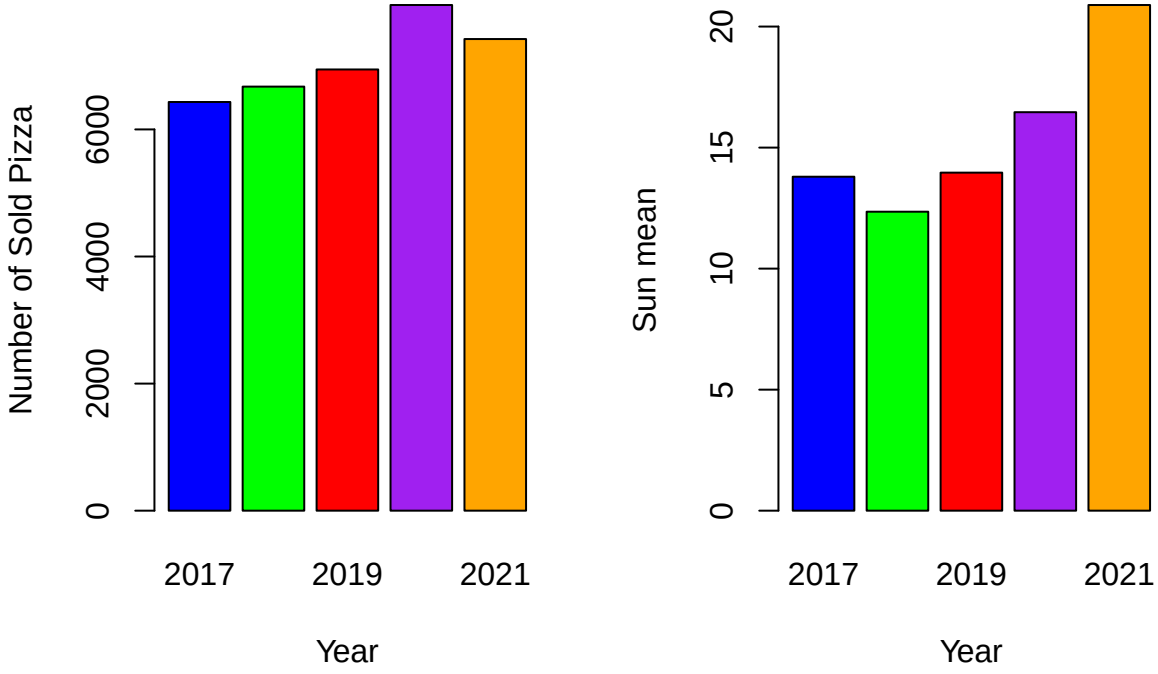
Barplot of number of sold PIZZA from 2017 - 2021

In the following the barplot of #PIZZA for Temperature,SUN,Price.offer and Precipitation can be see.

PIZZA , Temperature

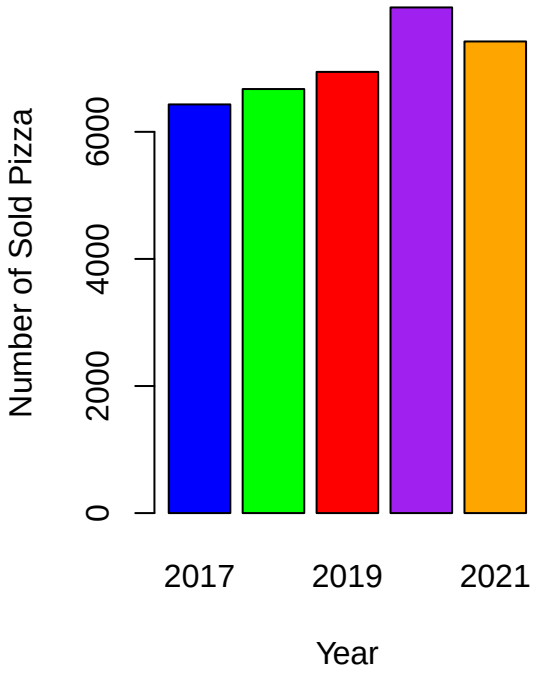


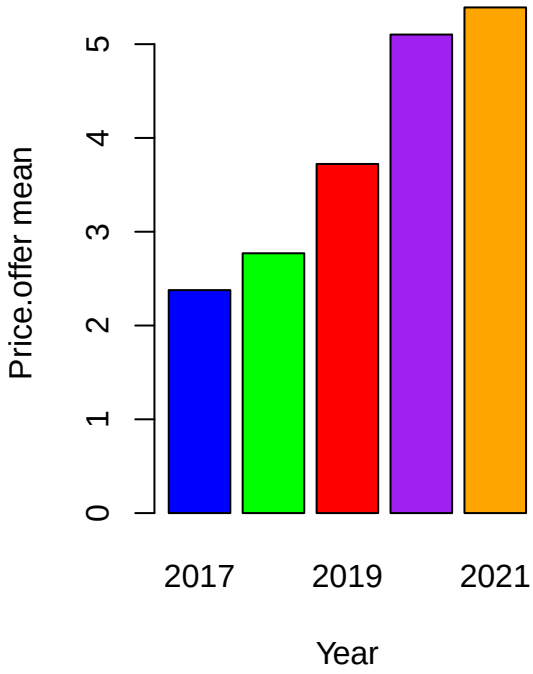
PIZZA , Sun



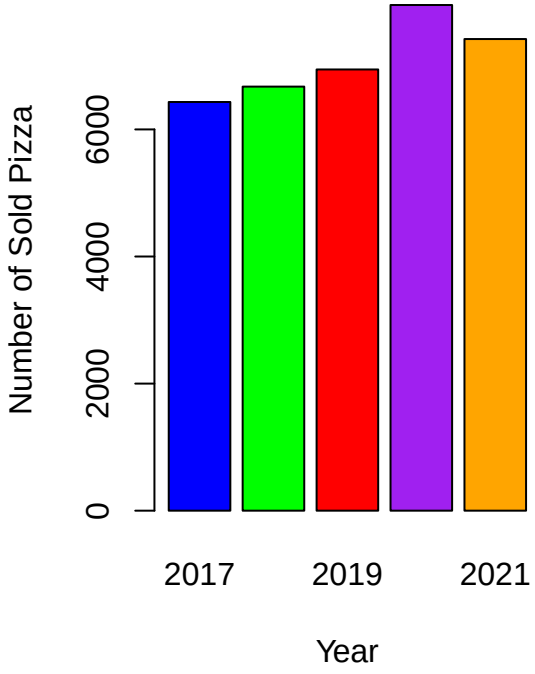
PIZZA , Price.offer

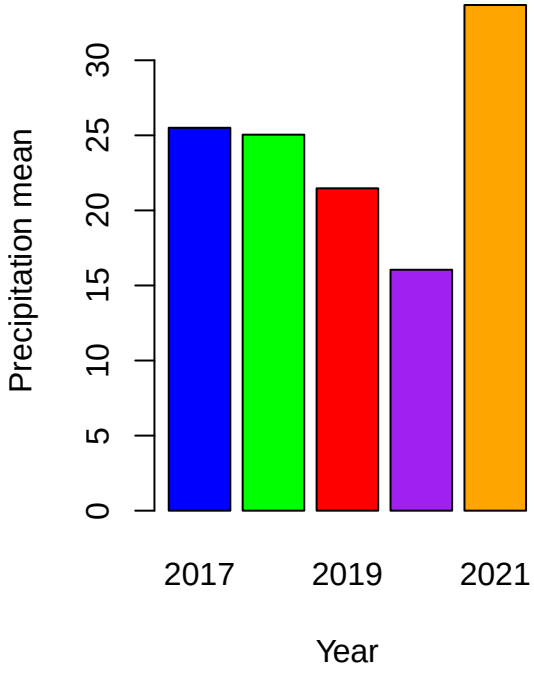
```
par(mfrow = c(1, 2))
barplot(c(sum(Year2017$Pizza), sum(Year2018$Pizza), sum(Year2019$Pizza), sum(Year2020$Pizza), sum(Year2
        names.arg = years,
        xlab = "Year", ylab = "Number of Sold Pizza",col=colors)

# Create the second bar plot
barplot(c(mean(Year2017$Price.offer), mean(Year2018$Price.offer), mean(Year2019$Price.offer), mean(Year
        names.arg = years,
        xlab = "Year", ylab = "Price.offer mean",col=colors)
```


Number of Sold Pizza
0
2000
4000
6000
2017
2019
2021
Year

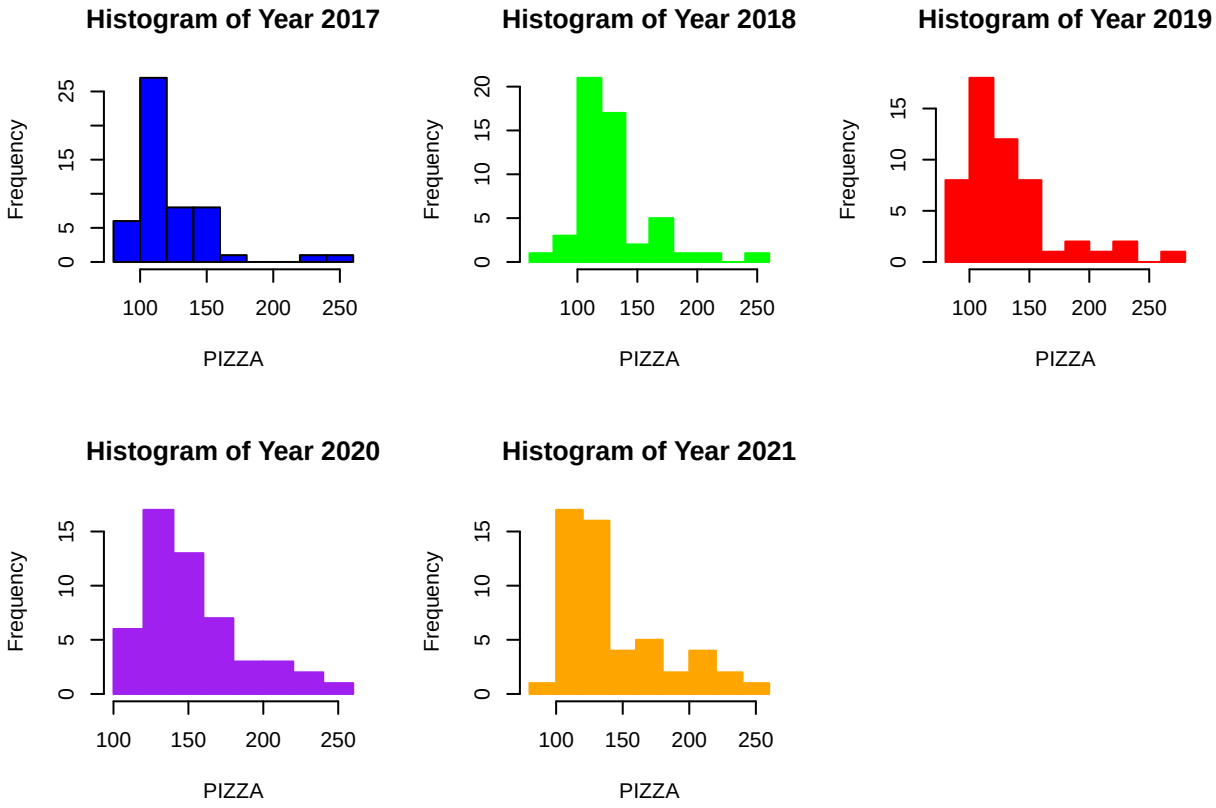

Price.offer mean
0
1
2
3
4
5
2017
2019
2021
Year

PIZZA , Precipitation

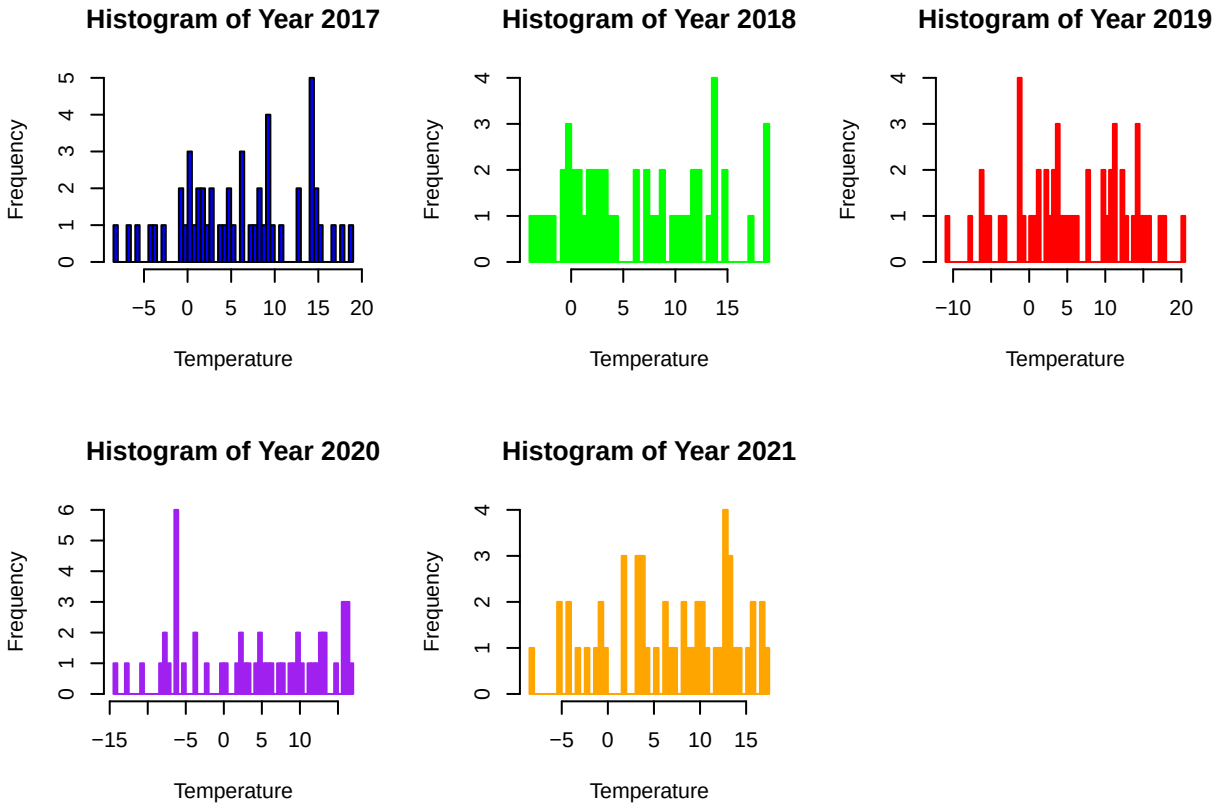




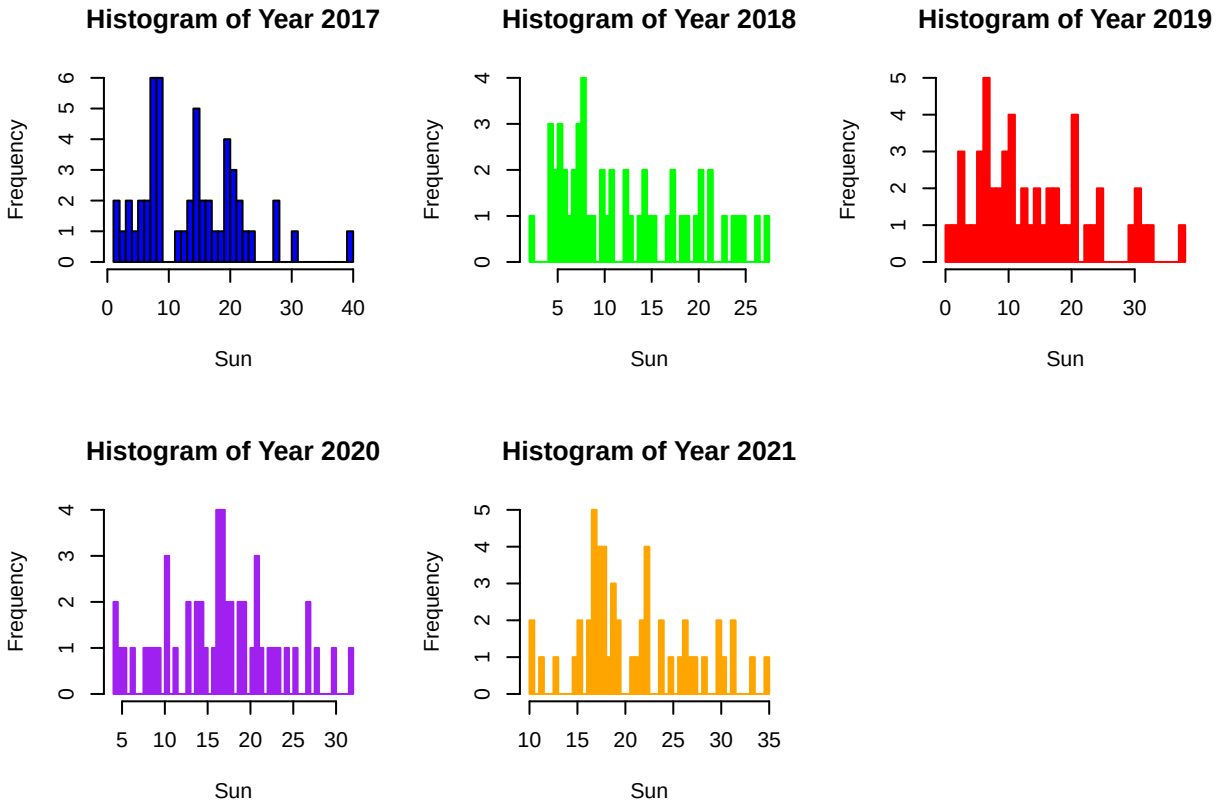
Histogram of sold PIZZA from 2017 - 2021



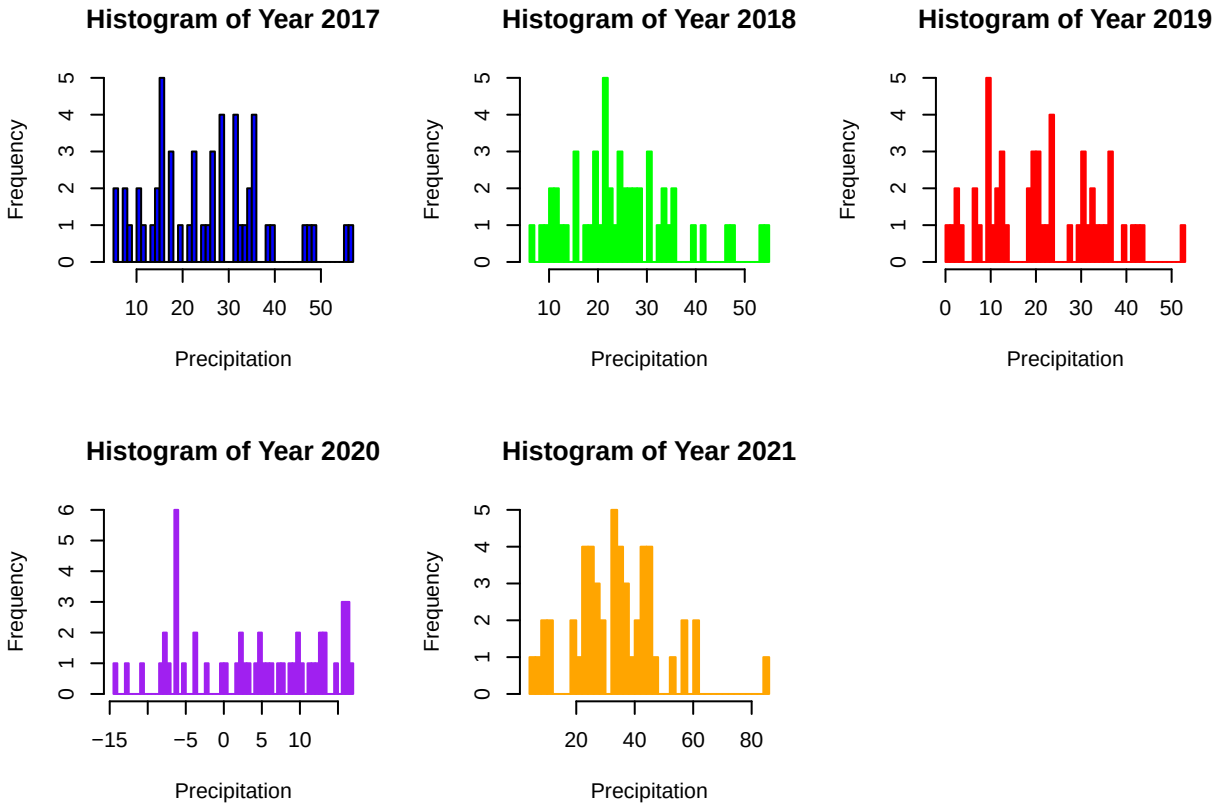
Histogram of Temperature from 2017 - 2021



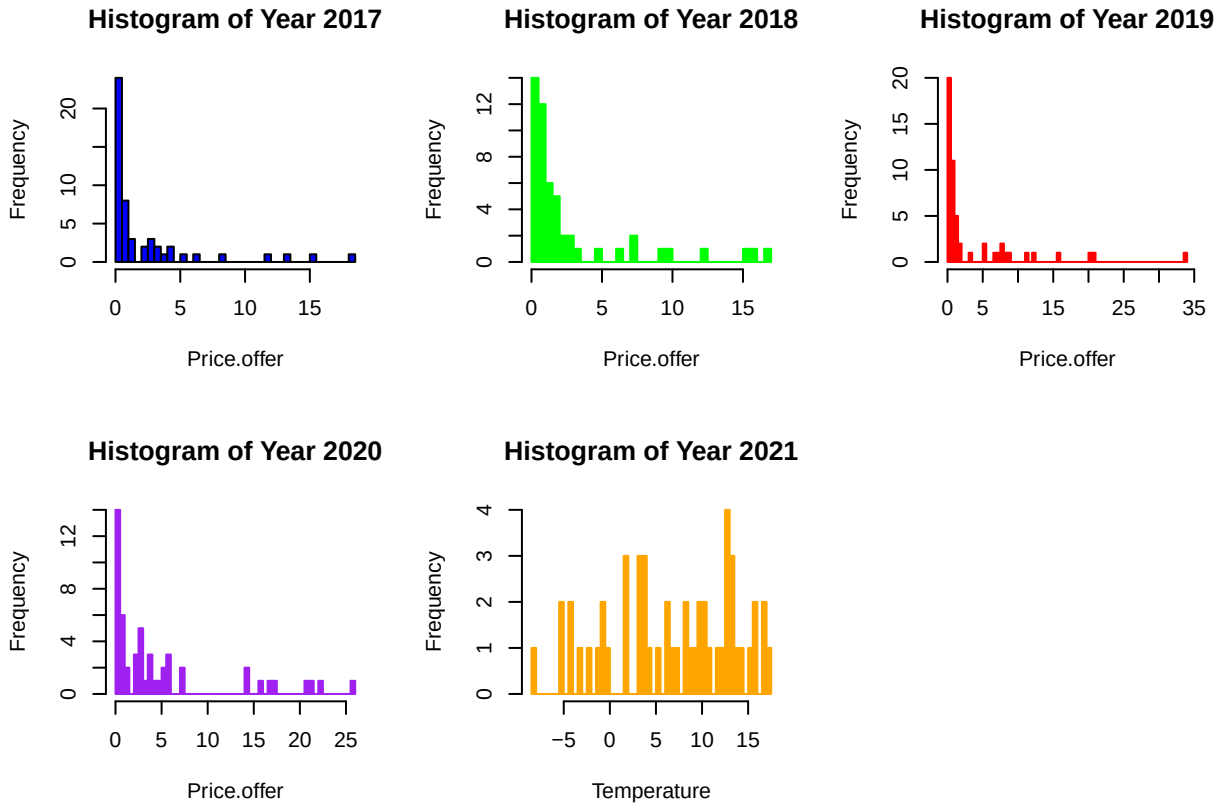
Histogram of Sun from 2017 - 2021



Histogram of Precipitation from 2017 - 2021



Histogram of Price.offer from 2017 - 2021



creating the model

Preprocessing and split to train and test

```
data = read.csv('data.csv')
summary(data)
```

```
##   Year.week            Pizza         Price.offer        Christmas
##  Length:261         Min.   : 66.87   Min.   : 0.0000   Min.   :0.00000
##  Class :character   1st Qu.:111.99   1st Qu.: 0.3317   1st Qu.:0.00000
##  Mode  :character   Median :127.96   Median : 1.0008   Median :0.00000
##                     Mean   :135.74   Mean   : 3.8722   Mean   :0.01916
##                     3rd Qu.:147.06   3rd Qu.: 4.2067   3rd Qu.:0.00000
##                     Max.   :274.78   Max.   :33.9935   Max.   :1.00000
##
##     New.year       X17th.of.May...National.day     Easter
##  Min.   :0.00000   Min.   :0.00000             Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.00000             1st Qu.:0.00000
##  Median :0.00000   Median :0.00000             Median :0.00000
##  Mean   :0.01533   Mean   :0.01916             Mean   :0.08812
##  3rd Qu.:0.00000   3rd Qu.:0.00000             3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.00000             Max.   :3.00000
##
##  Kr..Himmelfart..Ascension.day. Pinse..pentecost. Summer.vacation
```

```
##  Min.   :0.00000                Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000                1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000                Median :0.00000   Median :0.0000
##  Mean   :0.01916                Mean   :0.03831   Mean   :0.1533
##  3rd Qu.:0.00000                3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.00000                Max.   :1.00000   Max.   :1.0000
##
##   Temperature       Precipitation         Sun           Campaign.1
##  Min.   :-14.3405   Min.   :-29.01   Min.   : 0.000   Min.   :  0.000
##  1st Qu.:  0.4127   1st Qu.: 13.39   1st Qu.: 8.565   1st Qu.:  0.000
##  Median :  6.0156   Median : 23.35   Median :16.109   Median :  0.000
##  Mean   :  5.7483   Mean   : 24.34   Mean   :15.487   Mean   :  4.285
##  3rd Qu.: 12.3484   3rd Qu.: 33.66   3rd Qu.:20.717   3rd Qu.:  0.000
##  Max.   : 20.0514   Max.   : 85.90   Max.   :39.886   Max.   :228.000
##
##    Campaign.2        Campaign.3        Campaign.4       Campaign.5
##  Min.   :  0.000   Min.   :  0.000   Min.   : 0.000   Min.   :  0.000
##  1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.: 0.000   1st Qu.:  0.000
##  Median :  0.000   Median :  0.000   Median : 0.000   Median :  0.000
##  Mean   :  4.277   Mean   :  3.162   Mean   : 1.301   Mean   :  1.001
##  3rd Qu.:  0.000   3rd Qu.:  0.000   3rd Qu.: 0.000   3rd Qu.:  0.000
##  Max.   :190.500   Max.   :124.200   Max.   :86.300   Max.   :105.800
##
##    Campaign.6        Campaign.7       Campaign.8        Campaign.9
##  Min.   :  0.000   Min.   : 0.000   Min.   :  0.000   Min.   :  0.000
##  1st Qu.:  0.000   1st Qu.: 0.000   1st Qu.:  0.000   1st Qu.:  0.000
##  Median :  0.000   Median : 0.000   Median :  0.000   Median :  0.000
##  Mean   :  2.482   Mean   : 1.393   Mean   :  4.607   Mean   :  1.993
##  3rd Qu.:  0.000   3rd Qu.: 0.000   3rd Qu.:  0.000   3rd Qu.:  0.000
##  Max.   :196.300   Max.   :85.800   Max.   :173.100   Max.   :199.700
##
##   Campaign.10       Campaign.11       Campaign.12       Campaign.13
##  Min.   :  0.000   Min.   :  0.000   Min.   :  0.000   Min.   :  0.000
##  1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:  0.000   1st Qu.:  0.000
##  Median :  0.000   Median :  0.000   Median :  0.000   Median :  0.000
##  Mean   :  1.612   Mean   :  3.066   Mean   :  3.157   Mean   :  3.503
##  3rd Qu.:  0.000   3rd Qu.:  0.000   3rd Qu.:  0.000   3rd Qu.:  0.000
##  Max.   :177.000   Max.   :173.400   Max.   :170.500   Max.   :174.700
##
##   Campaign.14  Campaign.15  Campaign.16   Campaign17  Campaign.18
##  Min.   :0    Min.   :0    Min.   :0    Min.   :0    Min.   :0
##  1st Qu.:0    1st Qu.:0    1st Qu.:0    1st Qu.:0    1st Qu.:0
##  Median :0    Median :0    Median :0    Median :0    Median :0
##  Mean   :0    Mean   :0    Mean   :0    Mean   :0    Mean   :0
##  3rd Qu.:0    3rd Qu.:0    3rd Qu.:0    3rd Qu.:0    3rd Qu.:0
##  Max.   :0    Max.   :0    Max.   :0    Max.   :0    Max.   :0
##
##   Competitor.1      Competitor.2    Competitor.3     Competitor.4
##  Min.   :  0.000   Min.   :  0.0   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  0.000   1st Qu.:  0.0   1st Qu.:  0.00   1st Qu.:  0.00
##  Median :  0.000   Median :  0.0   Median :  0.00   Median :  0.00
##  Mean   :  7.577   Mean   : 38.2   Mean   : 47.33   Mean   : 13.58
##  3rd Qu.:  0.000   3rd Qu.: 89.6   3rd Qu.:109.50   3rd Qu.:  0.00
##  Max.   :272.301   Max.   :202.7   Max.   :258.10   Max.   :234.70
```

```
##                                                 NA's   :1
##   Competitor.5     Competitor.6     Competitor.7
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  0.00   1st Qu.: 41.20   1st Qu.:  0.00
##  Median :  0.00   Median : 61.50   Median :  0.00
##  Mean   : 12.15   Mean   : 62.58   Mean   : 20.81
##  3rd Qu.:  0.00   3rd Qu.: 77.90   3rd Qu.:  0.00
##  Max.   :180.20   Max.   :212.90   Max.   :243.30
##
```

```
df2=data[, -c(1)]
#remove 0 column
temp = c("Campaign.14", "Campaign.15", "Campaign.16","Campaign17","Campaign.18" )
df2 = df2[ ,! names(df2) %in% temp]
df3=as.data.frame(scale(df2))

data=df3
for(i in 1:ncol(data)) {
  data[is.na(data[,i]), i] <- min(data[,i], na.rm = TRUE)
}

# Replace missing values with the median of the column
missing_rows <- sapply(data, function(x) any(is.na(x)))
missing_rows_index <- which(missing_rows)


n=dim(data)[1]
set.seed(12345)
id=sample(1:n, floor(n*0.7))

# split to train and test

train=data[id,]
test=data[-id,]
trainX=as.matrix(train[,-1])
testX=as.matrix(test[,-1])
```

First linear regression model

```
m1=lm(Pizza~., data=train)
summary(m1)
```

```
##
## Call:
## lm(formula = Pizza ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.91418 -0.20079 -0.03899  0.17470  1.47286
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.011473   0.032310   0.355 0.723021
## Price.offer                   0.804713   0.034404  23.390  < 2e-16 ***
```

```
## Christmas                      -0.049193   0.061855  -0.795 0.427697
## New.year                       -0.039959   0.032068  -1.246 0.214674
## X17th.of.May...National.day    -0.054058   0.032352  -1.671 0.096820 .
## Easter                         -0.140834   0.035385  -3.980 0.000107 ***
## Kr..Himmelfart..Ascension.day. -0.030181   0.061019  -0.495 0.621596
## Pinse..pentecost.              -0.001868   0.034476  -0.054 0.956850
## Summer.vacation                -0.057143   0.042001  -1.361 0.175706
## Temperature                    -0.047488   0.058428  -0.813 0.417640
## Precipitation                   0.020872   0.041278   0.506 0.613851
## Sun                            -0.049563   0.048680  -1.018 0.310253
## Campaign.1                      0.060041   0.036972   1.624 0.106486
## Campaign.2                      0.083098   0.030319   2.741 0.006873 **
## Campaign.3                     -0.032094   0.035063  -0.915 0.361492
## Campaign.4                     -0.000291   0.032676  -0.009 0.992906
## Campaign.5                      0.196570   0.029091   6.757 2.92e-10 ***
## Campaign.6                      0.006248   0.061912   0.101 0.919749
## Campaign.7                     -0.016799   0.069128  -0.243 0.808332
## Campaign.8                     -0.005534   0.040767  -0.136 0.892198
## Campaign.9                     -0.002947   0.027794  -0.106 0.915697
## Campaign.10                     0.039436   0.028281   1.394 0.165252
## Campaign.11                     0.025638   0.041142   0.623 0.534129
## Campaign.12                     0.057737   0.028132   2.052 0.041872 *
## Campaign.13                    -0.040139   0.032054  -1.252 0.212441
## Competitor.1                   -0.015781   0.037006  -0.426 0.670387
## Competitor.2                   -0.053541   0.034980  -1.531 0.127974
## Competitor.3                    0.007253   0.035089   0.207 0.836511
## Competitor.4                    0.076498   0.034944   2.189 0.030129 *
## Competitor.5                    0.009757   0.033628   0.290 0.772111
## Competitor.6                   -0.006125   0.036182  -0.169 0.865795
## Competitor.7                    0.040652   0.039021   1.042 0.299189
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4126 on 150 degrees of freedom
## Multiple R-squared:  0.849,  Adjusted R-squared:  0.8178
## F-statistic: 27.21 on 31 and 150 DF,  p-value: < 2.2e-16
```

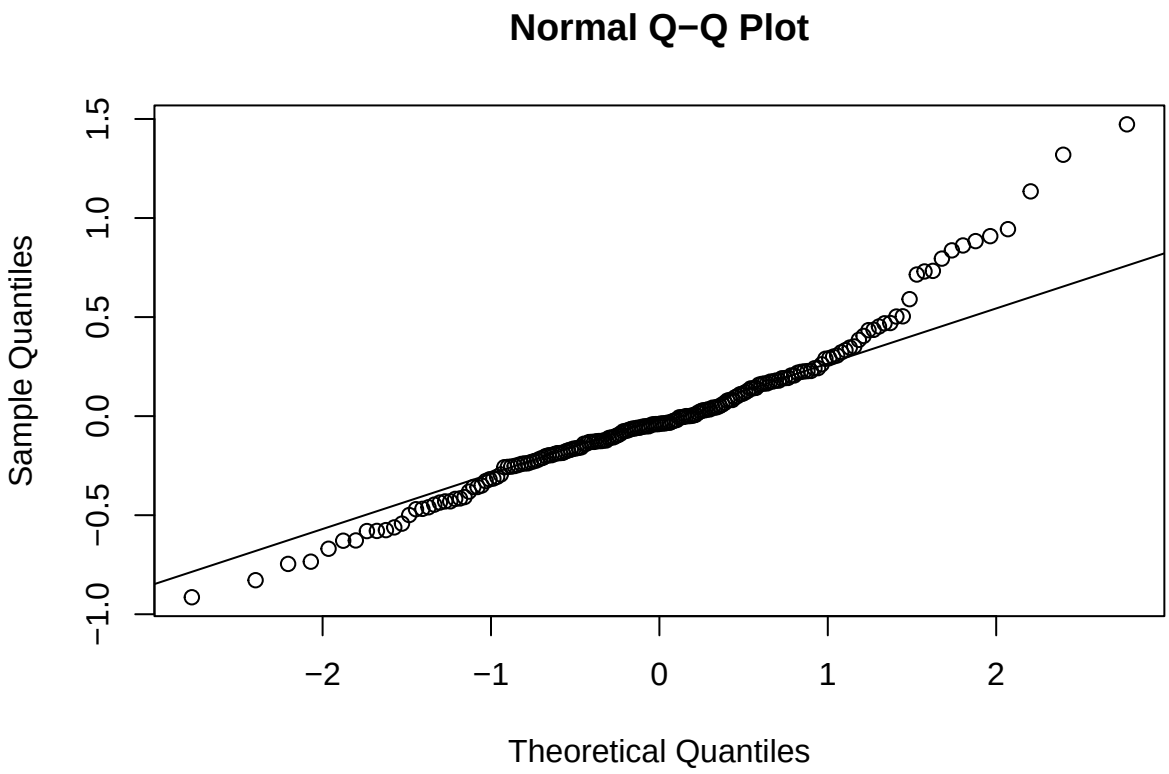
```
cat('mean square error on train set = ', mean((train$Pizza-predict(m1,train))^2,na.rm=TRUE),'\n')
```

```
## mean square error on train set =  0.1402895
```

```
cat('mean square error on test set = ', mean((test$Pizza-predict(m1,test))^2,na.rm=TRUE),'\n')
```

```
## mean square error on test set =  0.2150179
```

```
qqnorm(residuals(m1))
qqline(residuals(m1))
```



Second linear model, exclude competitor

```
#model2
model <- lm(Pizza ~ Price.offer +Temperature +Precipitation + Sun + Christmas + New.year + X17th.of.May
              Kr..Himmelfart..Ascension.day. + Pinse..pentecost. + Summer.vacation + Campaign.1 +
              Campaign.2 + Campaign.3 + Campaign.4 + Campaign.5 + Campaign.6 + Campaign.7 +
              Campaign.8 + Campaign.9 + Campaign.10 + Campaign.11 + Campaign.12 + Campaign.13 , data =
summary(model)
```

```
##
## Call:
## lm(formula = Pizza ~ Price.offer + Temperature + Precipitation +
##     Sun + Christmas + New.year + X17th.of.May...National.day +
##     Easter + Kr..Himmelfart..Ascension.day. + Pinse..pentecost. +
##     Summer.vacation + Campaign.1 + Campaign.2 + Campaign.3 +
##     Campaign.4 + Campaign.5 + Campaign.6 + Campaign.7 + Campaign.8 +
##     Campaign.9 + Campaign.10 + Campaign.11 + Campaign.12 + Campaign.13,
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90305 -0.20041 -0.04081  0.21062  1.44546
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.020245   0.032730   0.619 0.537113
```

```
## Price.offer                     0.806998   0.034340  23.500  < 2e-16 ***
## Temperature                    -0.038280   0.057166  -0.670 0.504079
## Precipitation                   0.031663   0.040402   0.784 0.434391
## Sun                            -0.041366   0.047286  -0.875 0.383022
## Christmas                      -0.059912   0.061040  -0.982 0.327842
## New.year                       -0.039445   0.032212  -1.225 0.222590
## X17th.of.May...National.day    -0.046880   0.031955  -1.467 0.144362
## Easter                         -0.141254   0.035466  -3.983 0.000104 ***
## Kr..Himmelfart..Ascension.day. -0.033865   0.060307  -0.562 0.575227
## Pinse..pentecost.              -0.012560   0.034500  -0.364 0.716303
## Summer.vacation                -0.066322   0.041526  -1.597 0.112247
## Campaign.1                      0.060789   0.037148   1.636 0.103762
## Campaign.2                      0.118722   0.028165   4.215 4.20e-05 ***
## Campaign.3                     -0.040582   0.035357  -1.148 0.252818
## Campaign.4                      0.004692   0.032338   0.145 0.884825
## Campaign.5                      0.197072   0.029304   6.725 3.09e-10 ***
## Campaign.6                      0.005778   0.063053   0.092 0.927103
## Campaign.7                     -0.022238   0.070275  -0.316 0.752085
## Campaign.8                     -0.007679   0.039291  -0.195 0.845311
## Campaign.9                      0.009738   0.027248   0.357 0.721291
## Campaign.10                     0.039322   0.028783   1.366 0.173849
## Campaign.11                     0.018351   0.041756   0.439 0.660917
## Campaign.12                     0.052405   0.027733   1.890 0.060648 .
## Campaign.13                    -0.046937   0.031074  -1.510 0.132927
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4215 on 157 degrees of freedom
## Multiple R-squared:  0.8351, Adjusted R-squared:  0.8099
## F-statistic: 33.12 on 24 and 157 DF,  p-value: < 2.2e-16
```

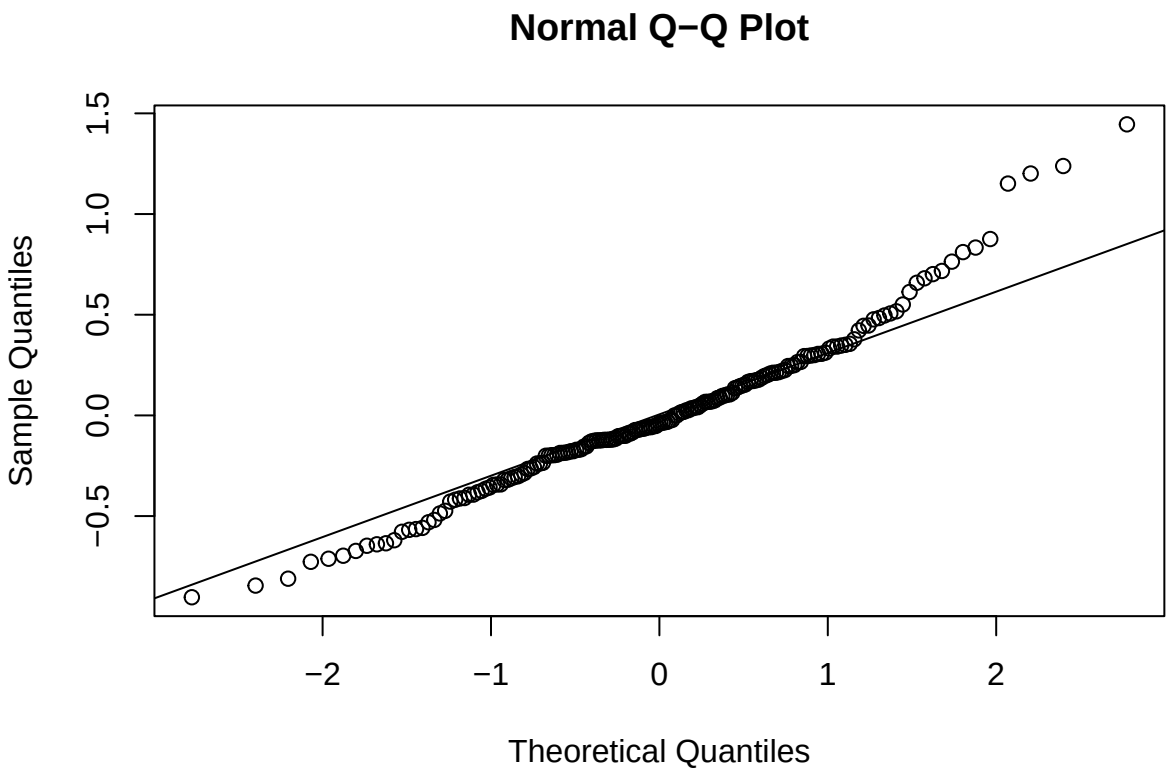
```
cat('mean square error on train set = ', mean((train$Pizza-predict(model,train))^2,na.rm=TRUE),'\n')
```

```
## mean square error on train set =  0.1532723
```

```
cat('mean square error on test set = ', mean((test$Pizza-predict(model,test))^2,na.rm=TRUE),'\n')
```

```
## mean square error on test set =  0.2015195
```

```
qqnorm(residuals(model))
qqline(residuals(model))
```




```
#####################################################################

lm2 = lm(Pizza ~ Price.offer+Temperature+Precipitation + Sun,data = train)
summary(lm2)
```

Finiding the most important feature among: Price.offer, Temperature, Precipitation, Sun

```
##
## Call:
## lm(formula = Pizza ~ Price.offer + Temperature + Precipitation +
##     Sun, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.20746 -0.33146 -0.09257  0.28221  2.18729
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.04740    0.03808   1.245   0.2149
## Price.offer    0.84183    0.03966  21.226   <2e-16 ***
## Temperature   -0.01900    0.05142  -0.369   0.7122
## Precipitation  0.04649    0.04115   1.130   0.2601
## Sun           -0.09316    0.05000  -1.863   0.0641 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.5135 on 177 degrees of freedom
## Multiple R-squared:  0.7241, Adjusted R-squared:  0.7178
## F-statistic: 116.1 on 4 and 177 DF,  p-value: < 2.2e-16
```

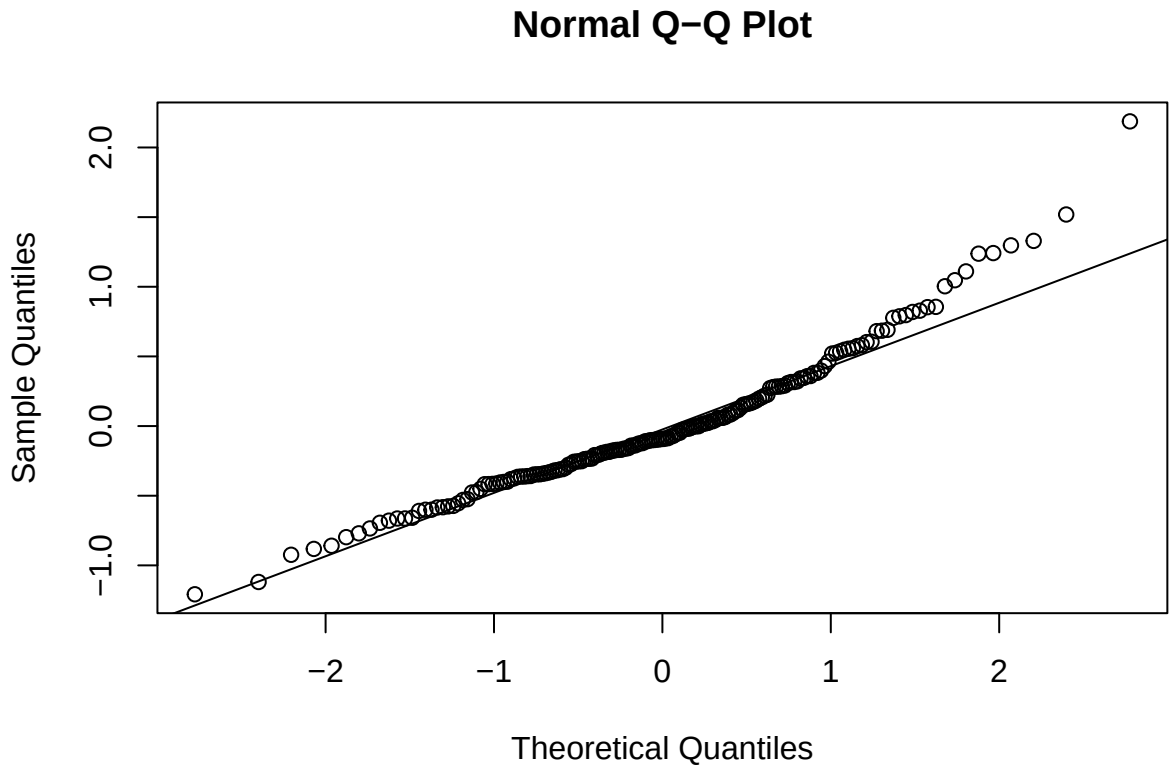
```
cat('mean square error on train set = ', mean((train$Pizza-predict(lm2,train))^2,na.rm=TRUE),'\n')
```

```
## mean square error on train set =  0.2564177
```

```
cat('mean square error on test set = ', mean((test$Pizza-predict(lm2,test))^2,na.rm=TRUE),'\n')
```

```
## mean square error on test set =  0.3176253
```

```
qqnorm(residuals(lm2))
qqline(residuals(lm2))
```

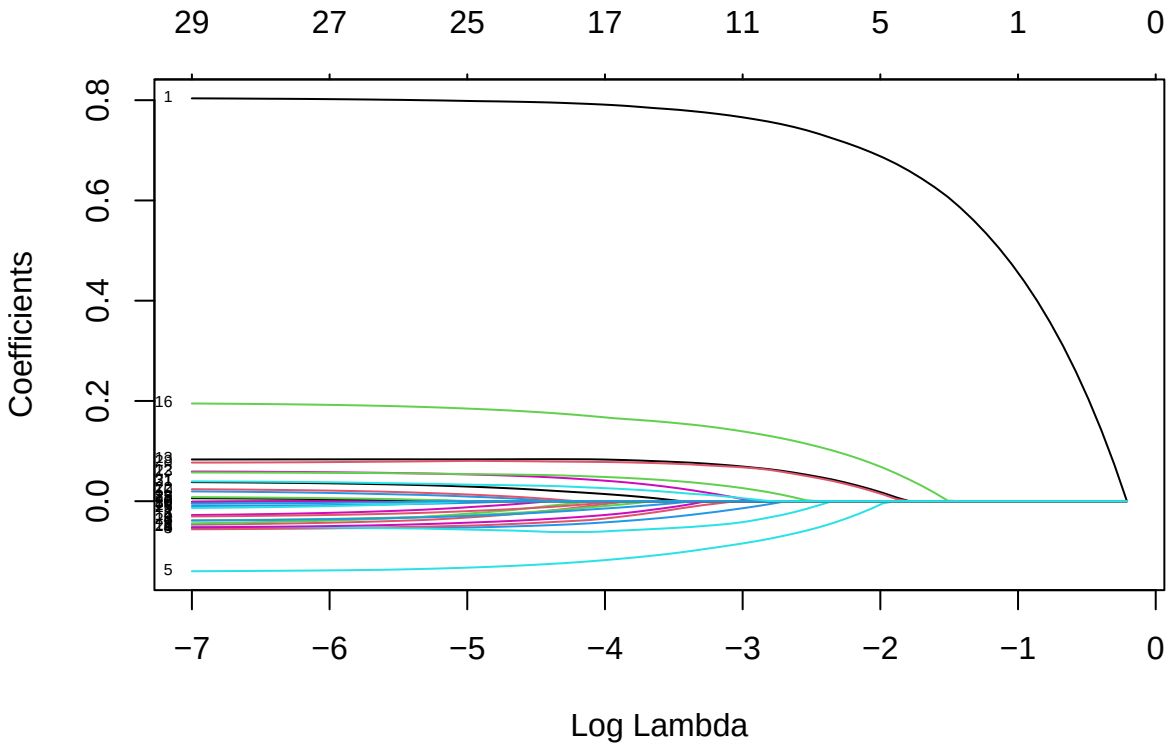



```
#################################################################
```

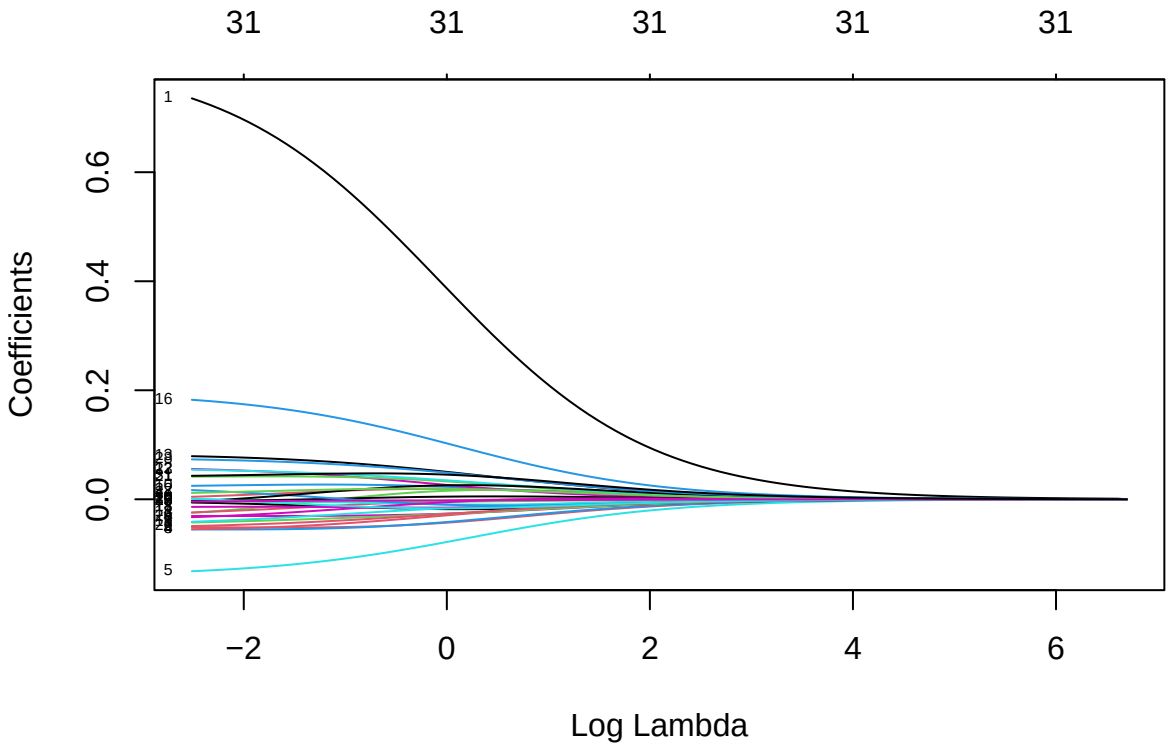
Based on the coef Price.Offer is the most important one.

LASSO and Ridge Regression model

```
library(glmnet)
model0=glmnet(trainX, train$Pizza, alpha=1,family="gaussian")
plot(model0, xvar="lambda", label=TRUE)
```



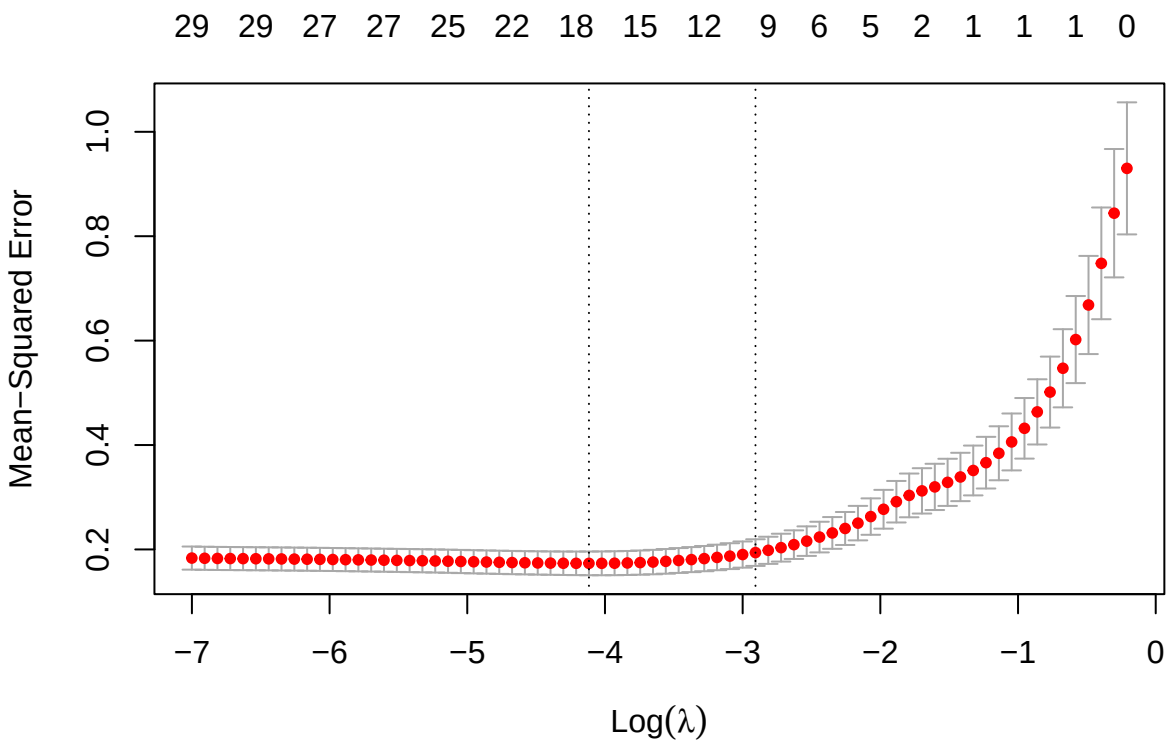

```
model1=glmnet(trainX, train$Pizza, alpha=0,family="gaussian")
plot(model1, xvar="lambda", label=TRUE)
```




```
set.seed(12345)
model=cv.glmnet(trainX, train$Pizza, alpha=1,family="gaussian")
model$lambda.min
```

```
## [1] 0.01630411
```

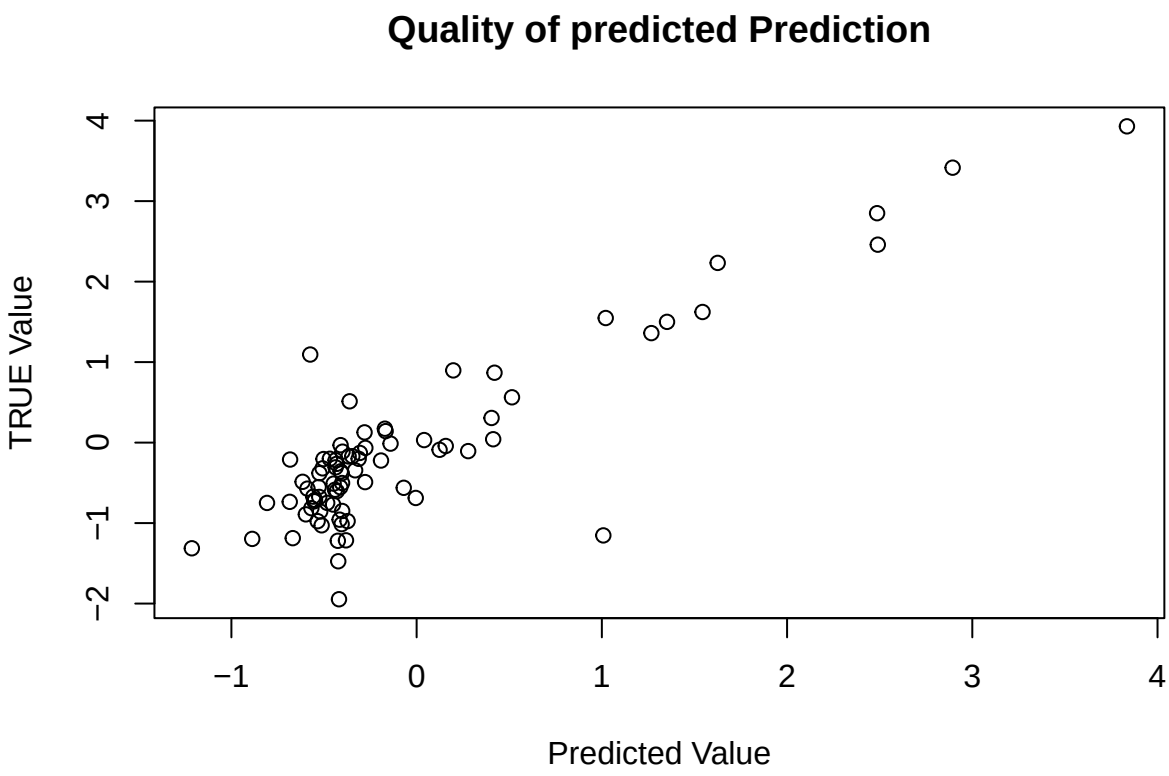
```
plot(model)
```




```
mB=glmnet(trainX, train$Pizza, alpha=1,family="gaussian", lambda = model$lambda.min)
cat('mean square error on test set = ', mean((test$Pizza-predict(mB,testX))^2),'\n')
```

```
## mean square error on test set =  0.2521334
```

```
plot(predict(mB, testX),test$Pizza,ylab = 'TRUE Value', xlab = 'Predicted Value',main = 'Quality of pred
```



The most important featurs

```
coefs <- coef(mB, s = "lambda.min")

important_feature_indices <- order(abs(coefs), decreasing = TRUE)[1:6]
#Feature_name_most = names(train)
Feature_name_most = coefs@Dimnames[[1]]
important_features <- Feature_name_most[important_feature_indices]
cat('The most important features: ')
```

```
## The most important features:
```

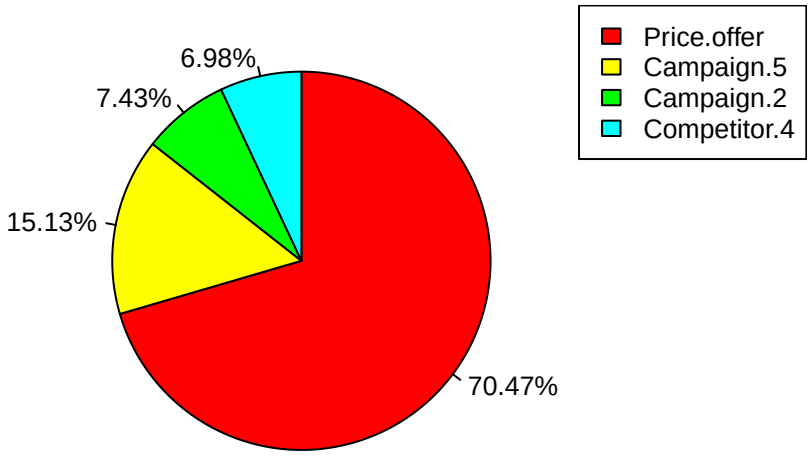
```
important_features_df = data.frame(data = t(coefs[important_feature_indices]) )
names(important_features_df) = important_features
print(important_features_df)
```

```
##   Price.offer Campaign.5     Easter Campaign.2 Competitor.4         Sun
## 1   0.7925594  0.1701207 -0.1200364 0.08354925   0.07847286 -0.06130093
```

Easter and Sun has negative effect in selling the highest number of PIZZA.

Pie chart of

```
values <- c(0.79255940,0.17012066,0.08354925,0.07847286)
labels <- c('Price.offer','Campaign.5','Campaign.2','Competitor.4')
pie(values, labels = paste0(round(values/sum(values)*100,2),"%"), main = "2D Percentage Pie Chart", col
legend("topright",legend=labels, cex=0.8,fill=rainbow(6))
```



The least important Features

```
zero_coef_indices_least <- which(coefs == 0)
Feature_name = coefs@Dimnames[[1]]
zero_coef_features <- Feature_name[zero_coef_indices_least]
cat('The least important features: ')
```

```
## The least important features:
```

```
print(zero_coef_features)
```

```
##  [1] "Kr..Himmelfart..Ascension.day." "Pinse..pentecost."
##  [3] "Precipitation"                  "Campaign.4"
##  [5] "Campaign.6"                     "Campaign.7"
##  [7] "Campaign.8"                     "Campaign.9"
##  [9] "Campaign.11"                    "Competitor.1"
## [11] "Competitor.3"                   "Competitor.5"
## [13] "Competitor.6"
```