linearregression.R

Batch1

2024-11-14

```
#1.load
prestige_data<-read.csv("C:/Users/batch1/Downloads/Prestige.csv")

#2.Apply data cleaning and data preprocessing to the prestige dataset
summary(prestige_data)
```

```
##    education          income          women           prestige
##  Min.   : 6.380   Min.   :  611   Min.   : 0.000   Min.   :14.80
##  1st Qu.: 8.445   1st Qu.: 4106   1st Qu.: 3.592   1st Qu.:35.23
##  Median :10.540   Median : 5930   Median :13.600   Median :43.60
##  Mean   :10.738   Mean   : 6798   Mean   :28.979   Mean   :46.83
##  3rd Qu.:12.648   3rd Qu.: 8187   3rd Qu.:52.203   3rd Qu.:59.27
##  Max.   :15.970   Max.   :25879   Max.   :97.510   Max.   :87.20
##      census         type
##  Min.   :1113   Length:102
##  1st Qu.:3120   Class :character
##  Median :5135   Mode  :character
##  Mean   :5402
##  3rd Qu.:8312
##  Max.   :9517
```

```
prestige_data<-na.omit(prestige_data)
head(prestige_data)
```

```
##   education income women prestige census type
## 1     13.11  12351 11.16     68.8   1113 prof
## 2     12.26  25879  4.02     69.1   1130 prof
## 3     12.77   9271 15.70     63.4   1171 prof
## 4     11.42   8865  9.11     56.8   1175 prof
## 5     14.62   8403 11.68     73.5   2111 prof
## 6     15.64  11030  5.13     77.6   2113 prof
```

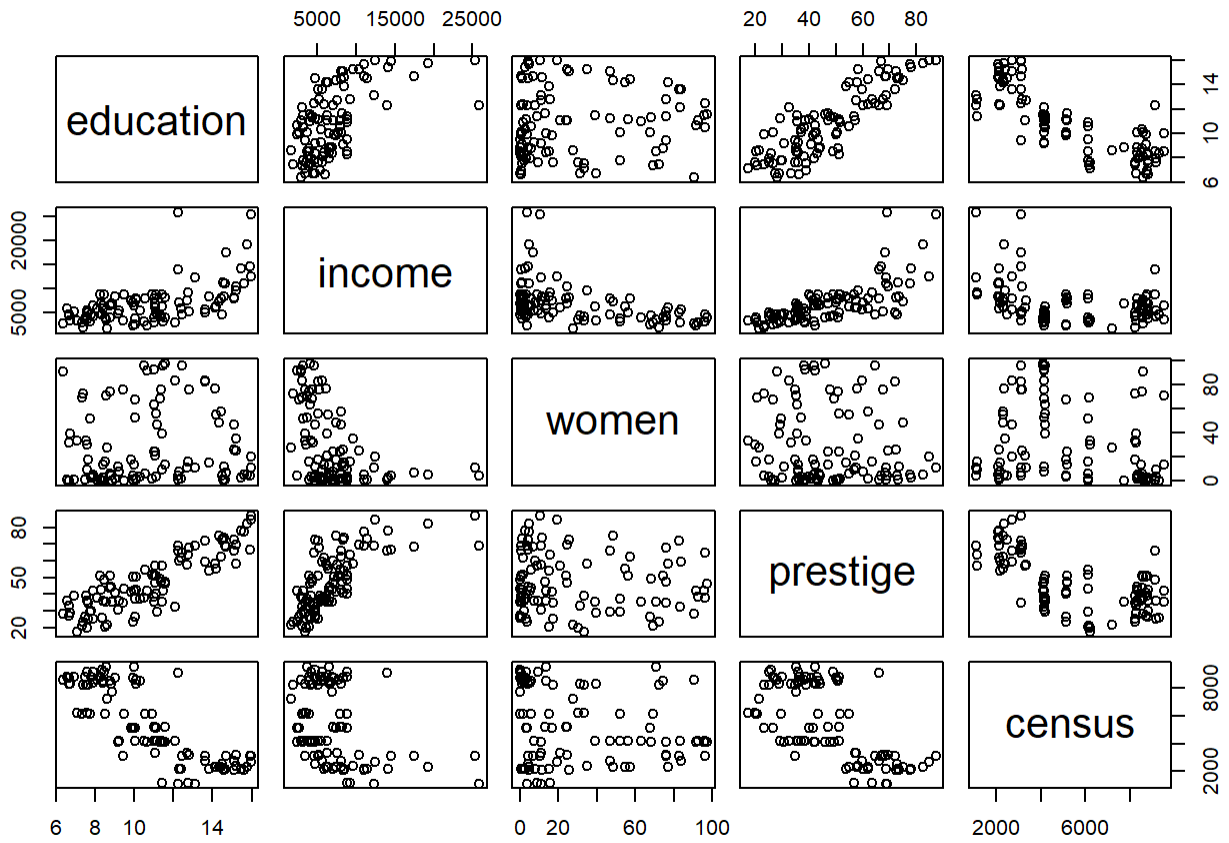
```
#3. Using plot or pair plot identify the relationship features and assign one independent fea
ture to X variable & the other dependent feature to Y variable.
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 4.2.3
```

```
numeric_data<-prestige_data[sapply(prestige_data,is.numeric)]
pairs(numeric_data)
```




```
head(prestige_data)
```

```
##   education income women prestige census type
## 1     13.11  12351 11.16     68.8   1113 prof
## 2     12.26  25879  4.02     69.1   1130 prof
## 3     12.77   9271 15.70     63.4   1171 prof
## 4     11.42   8865  9.11     56.8   1175 prof
## 5     14.62   8403 11.68     73.5   2111 prof
## 6     15.64  11030  5.13     77.6   2113 prof
```

```
tail(prestige_data)
```

```
##     education income women prestige census type
## 97       8.49   8845  0.00     48.9   9131   bc
## 98       7.58   5562  9.47     35.9   9171   bc
## 99       7.93   4224  3.59     25.1   9173   bc
## 100      8.37   4753  0.00     26.1   9313   bc
## 101     10.00   6462 13.58     42.2   9511   bc
## 102      8.55   3617 70.87     35.2   9517   bc
```

```
x<-prestige_data$education
y<-prestige_data$prestige

#Splitting the dataset into the Training set and Test set using 'caTools' with 2/3 and 1/3.
library(caTools)
```

```
## Warning: package 'caTools' was built under R version 4.2.3
```

```
set.seed(123)
split<-sample.split(prestige_data$prestige,SplitRatio=2/3)
training_set<-subset(prestige_data,split==TRUE)
test_set<-subset(prestige_data,split==FALSE)

#5.Fitting Simple Linear Regression to the Training set using the function lm (dependent vs i
ndependent variable inside the function to be used) and store in the object 'reg1' as model n
ame. Get help from R for the function lm.
reg1<-lm(prestige~education,data = training_set)
summary(reg1)
```

```
##
## Call:
## lm(formula = prestige ~ education, data = training_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.506  -6.657   0.380   7.274  17.890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -9.5860     4.6706  -2.052   0.0443 *
## education     5.1937     0.4249  12.225   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.058 on 63 degrees of freedom
## Multiple R-squared:  0.7035, Adjusted R-squared:  0.6987
## F-statistic: 149.4 on 1 and 63 DF,  p-value: < 2.2e-16
```

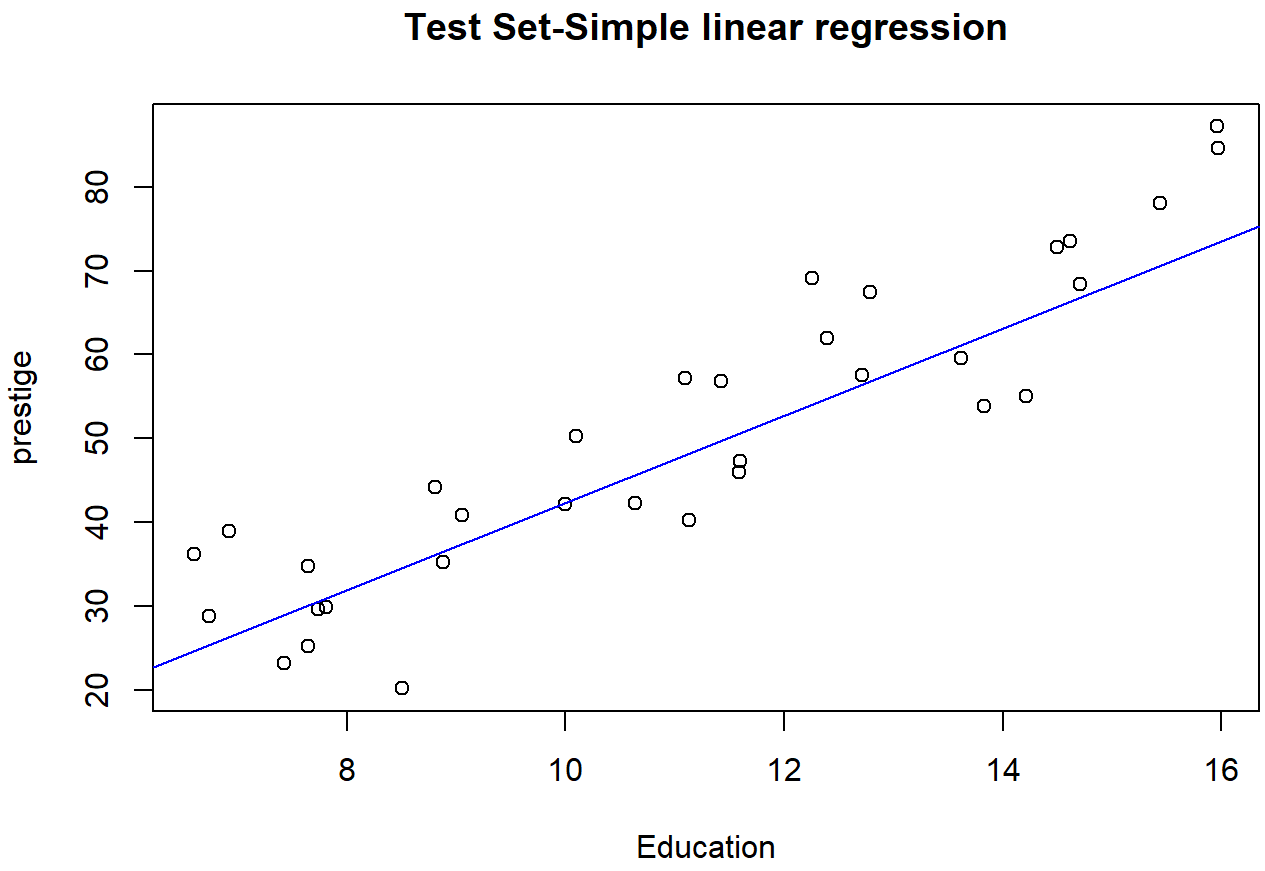
```
#6. Check for residuals error for the model created (summary)
summary(reg1)
```

```
##
## Call:
## lm(formula = prestige ~ education, data = training_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.506  -6.657   0.380   7.274  17.890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -9.5860     4.6706  -2.052   0.0443 *
## education     5.1937     0.4249  12.225   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.058 on 63 degrees of freedom
## Multiple R-squared:  0.7035, Adjusted R-squared:  0.6987
## F-statistic: 149.4 on 1 and 63 DF,  p-value: < 2.2e-16
```

```
#7.Predicting the Test set results using predict function and assign to y_pred
y_pred<-predict(reg1,newdata=test_set)
y_pred
```

```
##        2        4        5        8       11       13       16       20
## 54.08862 49.72592 66.34572 70.60455 54.76380 62.24271 64.21631 65.72248
##       21       22       24       26       31       32       33       35
## 73.35720 61.15203 73.30526 66.81315 56.84127 48.01200 56.42578 50.60884
##       38       51       55       60       61       69       71       72
## 45.67484 48.21975 50.66078 30.61314 34.56034 36.53395 30.09377 30.09377
##       73       75       77       86       88       91       92       93
## 28.95116 25.41945 36.17039 42.87025 37.41687 26.35432 24.69234 30.97670
##      101
## 42.35088
```

```
#8. Visualize the test set results with data points and abline (regressor line)
plot(test_set$education,test_set$prestige,main="Test Set-Simple linear regression",xlab="Educ
ation",ylab ="prestige")
abline(reg1,col="blue") #regression line
```




```
#9. Multiple linear regression - Log Transformation
#create a linear model 'reg2', (LHS - Independent variable Vs RHS all the dependent features
to be added, for income use log(income))
#summarize reg2 and check for the residual error
#Compare predicted and residual values
#First fit the prediction (use fitted function) and typecast to dataframe
#Second add the residual values (use residuals function for the same ) and typecase to datafr
ame
#Combine (1) first and (2) second questions to the single dataframe to check for predicted an
d residual errors
#Use a ggplot or qqPlot for the reg2 to visualize the linear regression

reg2<-lm(prestige~education+log(income)+women,data=training_set)

summary(reg2)
```

```
##
## Call:
## lm(formula = prestige ~ education + log(income) + women, data = training_set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5009  -4.1508  -0.1051   3.4637  14.6564
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -180.11323   23.59669  -7.633 1.88e-10 ***
## education      2.68604    0.46398   5.789 2.64e-07 ***
## log(income)   22.33595    3.03034   7.371 5.33e-10 ***
## women          0.12492    0.03786   3.300  0.00162 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.378 on 61 degrees of freedom
## Multiple R-squared:  0.8576, Adjusted R-squared:  0.8506
## F-statistic: 122.5 on 3 and 61 DF,  p-value: < 2.2e-16
```

```
predictions<-as.data.frame(fitted(reg2))
predictions
```

```
##     fitted(reg2)
## 1       66.93290
## 3       60.17982
## 6       70.44869
## 7       65.06994
## 9       67.62000
## 10      67.22504
## 12      51.84573
## 14      66.70649
## 15      63.50091
## 17      83.25167
## 18      62.25081
## 19      69.92048
## 23      67.07096
## 25      77.35203
## 27      53.80748
## 28      36.95848
## 29      57.44361
## 30      70.49465
## 36      42.64363
## 37      45.93626
## 39      46.91045
## 40      35.08373
## 41      41.70196
## 42      39.22129
## 43      38.01706
## 44      37.21325
## 45      40.12460
## 46      44.90085
## 47      47.26127
## 48      50.73731
## 49      43.00249
## 50      47.68849
## 52      30.93567
## 54      20.58487
## 56      50.45757
## 57      53.11136
## 58      48.42985
## 59      52.54754
## 62      49.39782
## 64      27.06368
## 65      25.27171
## 66      26.79484
## 68      12.01098
## 70      30.63939
## 74      29.37552
## 76      48.03363
## 78      39.05886
## 79      37.82666
## 80      37.82858
## 81      40.54149
```

```
## 82      37.65747
## 83      40.05128
## 84      26.01108
## 85      35.28034
## 87      26.64122
## 89      44.90207
## 90      45.16952
## 94      39.86181
## 95      24.96916
## 96      66.20508
## 97      45.67162
## 98      34.04882
## 99      28.10809
## 100     31.47697
## 102     34.71308
```

```
residuals<-as.data.frame(residuals(reg2))
results<-data.frame(Predicted=predictions,Residuals=residuals)
colnames(results)<-c("Predicted","Residuals")
results
```

```
##     Predicted    Residuals
## 1    66.93290   1.86710483
## 3    60.17982   3.22018336
## 6    70.44869   7.15131142
## 7    65.06994   7.53005830
## 9    67.62000   5.48000364
## 10   67.22504   1.57495639
## 12   51.84573   8.15427261
## 14   66.70649  -4.50649471
## 15   63.50091  11.39908900
## 17   83.25167  -0.95167133
## 18   62.25081  -4.15081294
## 19   69.92048 -11.62048236
## 23   67.07096  -0.97095909
## 25   77.35203 -10.65202782
## 27   53.80748  10.89251672
## 28   36.95848  -2.05847591
## 29   57.44361  14.65639164
## 30   70.49465  -1.19465261
## 36   42.64363  -0.74363016
## 37   45.93626   3.46373600
## 39   46.91045   0.78954754
## 40   35.08373  -4.18373176
## 41   41.70196  -9.00196212
## 42   39.22129  -0.52129146
## 43   38.01706  -1.91706384
## 44   37.21325  -0.01325437
## 45   40.12460  -2.02459712
## 46   44.90085 -15.50085477
## 47   47.26127   3.83872741
## 48   50.73731 -15.03731418
## 49   43.00249  -7.40248904
## 50   47.68849  -6.18848760
## 52   30.93567  -4.43567417
## 54   20.58487   2.71512537
## 56   50.45757  -3.35757228
## 57   53.11136  -2.01135875
## 58   48.42985  -4.92984937
## 59   52.54754  -0.94754278
## 62   49.39782   5.50217927
## 64   27.06368  -6.26367947
## 65   25.27171  -7.97171139
## 66   26.79484  -6.69483957
## 68   12.01098   9.48902074
## 70   30.63939   8.26060753
## 74   29.37552   3.92447992
## 76   48.03363  -5.53363381
## 78   39.05886  -3.15885804
## 79   37.82666   3.97333808
## 80   37.82858  -1.92858105
## 81   40.54149   3.15851450
```

```
## 82   37.65747  13.14252772
## 83   40.05128  -2.85128084
## 84   26.01108   2.18891869
## 85   35.28034   2.81965574
## 87   26.64122   0.65878185
## 89   44.90207   5.29793041
## 90   45.16952   5.93047723
## 94   39.86181   3.03819285
## 95   24.96916   1.53084201
## 96   66.20508  -0.10508405
## 97   45.67162   3.22837690
## 98   34.04882   1.85118270
## 99   28.10809  -3.00808533
## 100  31.47697  -5.37696656
## 102  34.71308   0.48692026
```

```
head(results)
```

```
##    Predicted Residuals
## 1   66.93290  1.867105
## 3   60.17982  3.220183
## 6   70.44869  7.151311
## 7   65.06994  7.530058
## 9   67.62000  5.480004
## 10  67.22504  1.574956
```

```
#10.
library(ggplot2)

ggplot(results,aes(sample=Residuals))+stat_qq()+stat_qq_line()+ggtitle("QQ plot of residualso
f reg2 model")
```


QQ plot of residualsof reg2 model
10
0
-10
y
-2
-1
0
1
2
x