lotto <- read.csv(file.choose(), header=TRUE)

attach(lotto)

model1 = lm(SALES ~ POP)

summary(model1)

Call:

lm(formula = SALES ~ POP)

Residuals:

    Min      1Q  Median      3Q     Max

-6046.7 -1460.9  -670.5   485.6 18229.5

Coefficients:

             Estimate Std. Error t value Pr(>|t|)

(Intercept) 469.70360   702.90619   0.668    0.507

POP           0.64709    0.04881  13.258   <2e-16 \*\*\*

---

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

Residual standard error: 3792 on 48 degrees of freedom

Multiple R-squared:  0.7855,           Adjusted R-squared:  0.781

F-statistic: 175.8 on 1 and 48 DF,  p-value: < 2.2e-16

newdata = data.frame(POP=10000)

# new piece of data not in original dataseet we want to predict for

predict(model1, newdata, interval="prediction", level=.95)

       fit       lwr      upr

1 6940.651 -759.3478 14640.65

predict(model1, newdata, interval="confidence", level=.95)

       fit     lwr      upr

1 6940.651 5860.36 8020.943

#fit is the y-hat value when we plug this x into our equation and

# the lwr and upr are the 95% Prediction (or confidence) Interval here

predict(model1, newdata)

       1

6940.651

predict(model1, newdata, interval="confidence", level=.95)

       fit     lwr      upr

1 6940.651 5860.36 8020.943

# confidence interval is a range for the average of a group, but prediction is the look at individual people's interval.

# confidence intervals will always be narrower than prediction.

predict(model1, newdata, interval="confidence", level=.95)

       fit     lwr      upr

1 6940.651 5860.36 8020.943

# CI = If the population has 10,000 people, the range of predicted sales for a group of towns would have an average lotto sales of between $5860.36 and $8020.94

#PI = If a town has 10,000, the range of predicted sales for an individual town would have lotto sales between $-759.35 and $14,640.65

outlier <- read.csv(file.choose(), header=TRUE)

attach(outlier)

# the data file is "OutlierExample"

outlier

     X   Y CODES

1  1.5 3.0     0

2  1.7 2.5     0

3  2.0 3.5     0

4  2.2 3.0     0

5  2.5 3.1     0

6  2.5 3.6     0

7  2.7 3.2     0

8  2.9 3.9     0

9  3.0 4.0     0

10 3.5 4.0     0

11 3.8 4.2     0

12 4.2 4.1     0

13 4.3 4.8     0

14 4.6 4.2     0

15 4.9 5.1     0

16 5.0 5.1     0

17 5.1 5.1     0

18 5.2 4.8     0

19 5.5 5.3     0

20 4.3 8.0     1

21 9.5 8.0     2

22 9.5 2.5     3

plot(x,y)

Error in plot(x, y) : object 'x' not found

plot(X, Y)

plot(X, Y, xlim=c(0,10), ylim=c(2,9), xlab="X axis", ylab="Y axis")

# x lim and y lim are the ranges of the axes and xlab & ylab alter the labels

model1 = lm(Y~X)

summary(model1)

Call:

lm(formula = Y ~ X)

Residuals:

    Min      1Q  Median      3Q     Max

-3.6674 -0.3891 -0.0534  0.4355  3.6163

Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept)   2.9087     0.6128   4.746 0.000123 \*\*\*

X             0.3430     0.1331   2.577 0.017987 \*

---

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

Residual standard error: 1.297 on 20 degrees of freedom

Multiple R-squared:  0.2493,           Adjusted R-squared:  0.2118

F-statistic: 6.643 on 1 and 20 DF,  p-value: 0.01799

# p-value = 0.01799, this is less than alpha = 0.05, which means our X is significant (doing a good job at predicting y value.)

# R-squared is 0.2493, this is low which is bad. 24.93% of the variation in Y is being explained by the X.

model2 = lm(Y~X, subset=-c(20))

# this removes all columns associated with observation #20

summary (model2)

Call:

lm(formula = Y ~ X, subset = -c(20))

Residuals:

    Min      1Q  Median      3Q     Max

-3.4540 -0.2708  0.0615  0.5901  2.0460

Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept)   2.7677     0.4857   5.699 1.71e-05 \*\*\*

X             0.3354     0.1052   3.190  0.00482 \*\*

---

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

Residual standard error: 1.025 on 19 degrees of freedom

Multiple R-squared:  0.3487,           Adjusted R-squared:  0.3145

F-statistic: 10.17 on 1 and 19 DF,  p-value: 0.004824

# p-value is smaller at 0.00482, which is better

# R-squared is higher at 0.3487, which is better

model3 = lm(Y~X, subset=-c(21))

summary (model3)

Call:

lm(formula = Y ~ X, subset = -c(21))

Residuals:

    Min      1Q  Median      3Q     Max

-2.6746 -0.7131 -0.0785  0.5754  3.7754

Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept)   3.4391     0.6449   5.332  3.8e-05 \*\*\*

X             0.1827     0.1524   1.199    0.245

---

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

Residual standard error: 1.224 on 19 degrees of freedom

Multiple R-squared:  0.07033,        Adjusted R-squared:  0.0214

F-statistic: 1.437 on 1 and 19 DF,  p-value: 0.2453

# p-value is larger at 0.245, which is worse, now the X is not significant

# R-squared is lower at 0.07033, which is worse

model4 = lm(Y~X, subset=-c(22))

summary(model4)

Call:

lm(formula = Y ~ X, subset = -c(22))

Residuals:

    Min      1Q  Median      3Q     Max

-0.7011 -0.4069 -0.1541  0.0981  3.2981

Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept)   1.8473     0.4344   4.252 0.000431 \*\*\*

X             0.6639     0.1026   6.468 3.37e-06 \*\*\*

---

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

Residual standard error: 0.8246 on 19 degrees of freedom

Multiple R-squared:  0.6877,           Adjusted R-squared:  0.6713

F-statistic: 41.84 on 1 and 19 DF,  p-value: 3.369e-06

**# p-value is smaller at 0.00000337, which is the best**

**# R-squared is highest at 0.6877, which is the best**

model5 = lm(Y~X, subset=-c(20,22))

summary(model5)

Call:

lm(formula = Y ~ X, subset = -c(20, 22))

Residuals:

     Min       1Q   Median       3Q      Max

-0.51763 -0.28094  0.03452  0.23586  0.44581

Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept)  1.77463    0.15020   11.81 6.48e-10 \*\*\*

X            0.63978    0.03551   18.02 5.81e-13 \*\*\*

---

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

Residual standard error: 0.2849 on 18 degrees of freedom

Multiple R-squared:  0.9474,           Adjusted R-squared:  0.9445

F-statistic: 324.5 on 1 and 18 DF,  p-value: 5.808e-13

model6 = lm(Y~X, subset=-c(20,21,22))

summary(model6)

Call:

lm(formula = Y ~ X, subset = -c(20, 21, 22))

Residuals:

    Min      1Q  Median      3Q     Max

-0.4790 -0.2708  0.0711  0.2263  0.4094

Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept)  1.86874    0.19583   9.543 3.06e-08 \*\*\*

X            0.61094    0.05219  11.705 1.47e-09 \*\*\*

---

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

Residual standard error: 0.2883 on 17 degrees of freedom

Multiple R-squared:  0.8896,           Adjusted R-squared:  0.8831

F-statistic:   137 on 1 and 17 DF,  p-value: 1.471e-09

#not as good to remove all three. THe best is to remove point 20 and 22. But this is too tedious to evaluate individually, so we

# will learn how to evaluate all points at the same time

# the model that removes both 20 and 22 seems to be the best combination here

#HOMEWORK 1 NOTES: make a boxplot with a quantitative (Y variable) by the categorical (X)

# boxplot (Y ~ X)

term <-read.csv(file.choose(), header=TRUE)

# Data file is 'TermLife'

attach(term)

dim(term)

[1] 500  18

# means it has 500 observations and 18 variables

term1 = subset(term, subset=FACE>0)

# takes a subset of the term datast and only uses people who have a positive FACE value of their policy, they purchased

# a policiy. Call it term1.

dim(term1)

[1] 275  18

#If you want a subset that only equals a certain value, it would look like this:

# term1 = subset(term, subset=FACE==0)

# if you want a subset that only doesn't equal a certain value, it would look like this:

# term1 = subset(term, subset=FACE!=0)       use the ! to mean doesn't equal

attach(term1)

The following objects are masked from term:

    AGE, BORROWCVLIFEPOL, CASHCVLIFEPOLICIES, CHARITY, EDUCATION, ETHNICITY, FACE, FACECVLIFEPOLICIES, GENDER, INCOME,

    MARSTAT, NETVALUE, NUMHH, SAGE, SEDUCATION, SGENDER, SMARSTAT, TOTINCOME

hist(FACE)

# This is strongly right skewed

hist(INCOME)

# Also strongly right skewed

logFACE = log(FACE)

hist(logFACE)

# much improved and more symmetric

> logINCOME = log(INCOME)

> hist(logINCOME)

> # much improved and more symmetric