Regression

Darwin Khay

10/31/2020

```
asmr <- read.csv("C:\\Users\\khayd\\Documents\\FALL 2020 Files\\STAT 1601\\Datasets\\ASMR_data2.csv")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Multiple Linear Regression Model to predict Mood_After_watch

```
# Building a model to predict the mood after watching
mood.data <- asmr%>%
  select(BDI_TOTAL, Mood_Before_watch, Mood_During_Watch, Mood_After_watch, Mood_Daily)
bdimood.model <- lm(Mood_After_watch~., mood.data)
summary(bdimood.model)
```

```
##
## Call:
## lm(formula = Mood_After_watch ~ ., data = mood.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.479  -5.156  -0.332   4.918  40.283
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       10.55151    2.99850   3.519 0.000475 ***
## BDI_TOTAL          0.12024    0.06019   1.998 0.046309 *
## Mood_Before_watch  0.04903    0.05027   0.975 0.329879
## Mood_During_Watch  0.62592    0.03574  17.513  < 2e-16 ***
## Mood_Daily         0.22220    0.05198   4.275 2.32e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.536 on 470 degrees of freedom
## Multiple R-squared:  0.5761, Adjusted R-squared:  0.5725
## F-statistic: 159.7 on 4 and 470 DF,  p-value: < 2.2e-16
```

Using AIC to make a better multiple linear regression model

```
# Using correlation matrix to see which predictor variables are better
mood.data%>%
  cor()
```

```
##                    BDI_TOTAL Mood_Before_watch Mood_During_Watch
## BDI_TOTAL          1.0000000        -0.5321278        -0.2485346
## Mood_Before_watch -0.5321278         1.0000000         0.4657276
## Mood_During_Watch -0.2485346         0.4657276         1.0000000
## Mood_After_watch  -0.2633081         0.5085063         0.7223322
## Mood_Daily        -0.6303396         0.8366343         0.4506685
##                   Mood_After_watch Mood_Daily
## BDI_TOTAL               -0.2633081 -0.6303396
## Mood_Before_watch        0.5085063  0.8366343
## Mood_During_Watch        0.7223322  0.4506685
## Mood_After_watch         1.0000000  0.5251423
## Mood_Daily               0.5251423  1.0000000
```

```
# Using AIC pick out the best predictors
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
bdimood.model2 <- stepAIC(bdimood.model, direction = "both", trace = F)
detach("package:MASS")
summary(bdimood.model2)
```

```
##
## Call:
## lm(formula = Mood_After_watch ~ BDI_TOTAL + Mood_During_Watch +
##     Mood_Daily, data = mood.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.815  -5.158  -0.276   5.120  40.884
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       10.72823    2.99287   3.585 0.000373 ***
## BDI_TOTAL          0.11902    0.06017   1.978 0.048510 *
## Mood_During_Watch  0.63227    0.03514  17.992  < 2e-16 ***
## Mood_Daily         0.25858    0.03619   7.146 3.42e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.536 on 471 degrees of freedom
## Multiple R-squared:  0.5753, Adjusted R-squared:  0.5726
## F-statistic: 212.7 on 3 and 471 DF,  p-value: < 2.2e-16
```

Predicting using the AIC Multilinear regression model

```
# Creating values to predict in a data frame

newvalues <- data.frame(BDI_TOTAL = c(10,44,23), Mood_During_Watch = c(70,31,63), Mood_Daily = c(65,26,5

predict(bdimood.model2, newvalues, interval = "predict")
```

```
##        fit      lwr      upr
## 1 72.98531 54.19123 91.77938
## 2 42.28854 23.07345 61.50363
## 3 67.77937 48.91351 86.64523
```

Logistic Regression Model to predict EffectSleep

```
# First subsetting the data to only include the columns of the predictor variables and another one colu
effectsleep.data <- asmr%>%
  mutate(EffectSleep2 = ifelse(EffectSleep == "Yes", 1, 0))%>%
  select(BDI_TOTAL, BAI_TOTAL, Mood_Daily, Mood_Before_watch, Mood_After_watch, Num_ASMRVideos, EffectSl

# Building logistic regression model

logit <- glm(EffectSleep2~. , effectsleep.data, family = "binomial")


# Rebuilding logistic regression model to only include the most important variables by using AIC
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```r
new.logit <- stepAIC(logit, direction = "both", trace = F)
summary(new.logit)
```

```
##
## Call:
## glm(formula = EffectSleep2 ~ BDI_TOTAL + BAI_TOTAL + Mood_Before_watch +
##     Mood_After_watch, family = "binomial", data = effectsleep.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4238  -1.0555   0.5727   0.9974   1.5360
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -1.005931   0.661072  -1.522  0.12809
## BDI_TOTAL          0.054584   0.017554   3.109  0.00187 **
## BAI_TOTAL          0.036982   0.014616   2.530  0.01140 *
## Mood_Before_watch -0.022254   0.008724  -2.551  0.01074 *
## Mood_After_watch   0.022400   0.008581   2.611  0.00904 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 641.72  on 474  degrees of freedom
## Residual deviance: 569.01  on 470  degrees of freedom
## AIC: 579.01
##
## Number of Fisher Scoring iterations: 4
```

```r
detach("package:MASS")

exp(coef(new.logit))
```

```
##       (Intercept)         BDI_TOTAL         BAI_TOTAL Mood_Before_watch
##         0.3657038         1.0561014         1.0376740         0.9779921
##  Mood_After_watch
##         1.0226532
```

```r
# Using new values to predict the outcome of EffectSleep

newvalues2 <- data.frame(BDI_TOTAL = c(15,25,47), BAI_TOTAL = c(6,22,33), Mood_Before_watch = c(85,55,2
predict(new.logit, newvalues2, type = "response")
```

```
##         1         2         3
## 0.5397148 0.7993030 0.9508372
```