HW8_Videtti

##1. The data sets package in R contains a small data set called mtcars that contains n = 32 observations of the characteristics of different automobiles. Create a new data frame from part of this data set using this command: myCars <- data.frame(mtcars[,1:6]).

```
myCars <- data.frame(mtcars[,1:6])
myCars
```

```
##                      mpg cyl  disp  hp drat    wt
## Mazda RX4           21.0   6 160.0 110 3.90 2.620
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875
## Datsun 710          22.8   4 108.0  93 3.85 2.320
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440
## Valiant             18.1   6 225.0 105 2.76 3.460
## Duster 360          14.3   8 360.0 245 3.21 3.570
## Merc 240D           24.4   4 146.7  62 3.69 3.190
## Merc 230            22.8   4 140.8  95 3.92 3.150
## Merc 280            19.2   6 167.6 123 3.92 3.440
## Merc 280C           17.8   6 167.6 123 3.92 3.440
## Merc 450SE          16.4   8 275.8 180 3.07 4.070
## Merc 450SL          17.3   8 275.8 180 3.07 3.730
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345
## Fiat 128            32.4   4  78.7  66 4.08 2.200
## Honda Civic         30.4   4  75.7  52 4.93 1.615
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835
## Toyota Corona       21.5   4 120.1  97 3.70 2.465
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520
## AMC Javelin         15.2   8 304.0 150 3.15 3.435
## Camaro Z28          13.3   8 350.0 245 3.73 3.840
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140
## Lotus Europa        30.4   4  95.1 113 3.77 1.513
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770
## Maserati Bora       15.0   8 301.0 335 3.54 3.570
## Volvo 142E          21.4   4 121.0 109 4.11 2.780
```

##2. Create and interpret a bivariate correlation matrix using cor(myCars) keeping in mind the idea that you will be trying to predict the mpg variable. Which other variable might be the single best predictor of mpg?

```
cor(myCars)
```

```
##             mpg        cyl       disp         hp       drat         wt
## mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.6811719 -0.8676594
## cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.6999381  0.7824958
## disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.7102139  0.8879799
## hp   -0.7761684  0.8324475  0.7909486  1.0000000 -0.4487591  0.6587479
## drat  0.6811719 -0.6999381 -0.7102139 -0.4487591  1.0000000 -0.7124406
## wt   -0.8676594  0.7824958  0.8879799  0.6587479 -0.7124406  1.0000000
```

#We see that mpg is most strongly correlated with wt, although cyl and disp are very close second and third options, respectively. Because of this, we can assume that the wt variable might be the best predictor for mpg in these data.

##3. Run a multiple regression analysis on the myCars data with lm(), using mpg as the dependent variable and wt (weight) and hp (horsepower) as the predictors. Make sure to say whether or not the overall R-squared was significant. If it was significant, report the value and say in your own words whether it seems like a strong result or not. Review the significance tests on the coefficients (B-weights). For each one that was significant, report its value and say in your own words whether it seems like a strong result or not.

```
summary(lm(mpg~wt + hp, data = myCars))
```

```
## 
## Call:
## lm(formula = mpg ~ wt + hp, data = myCars)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -3.941 -1.600 -0.182  1.050  5.854 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 37.22727    1.59879  23.285  < 2e-16 ***
## wt          -3.87783    0.63273  -6.129 1.12e-06 ***
## hp          -0.03177    0.00903  -3.519  0.00145 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.593 on 29 degrees of freedom
## Multiple R-squared:  0.8268, Adjusted R-squared:  0.8148 
## F-statistic: 69.21 on 2 and 29 DF,  p-value: 9.109e-12
```

#We see that the p-value for the F-test is extremely low at 9.109e-12, so we will reject the null hypothesis that R-squared is equal to zero, thus, the overall R-squared was significant. The result was 0.8268, which is usually considered a very strong result, but could be considered weak in certain

contexts. The intercept, wt, and hp, all are significant, as all have Pr(>|t|) < 0.05. The intercept B-weight is approximately 37.2, the B-weight of wt is approximately -3.9, and the B-weight of hp is approximately -0.03. I would say that the wt variable is a much stronger result since its value is so much larger than that of hp.

##4. Using the results of the analysis from Exercise 2, construct a prediction equation for mpg using all three of the coefficients from the analysis (the intercept along with the two B-weights). Pretend that an automobile designer has asked you to predict the mpg for a car with 110 horsepower and a weight of 3 tons. Show your calculation and the resulting value of mpg.

```
Exercise4 <- 37.22727 + (-3.87783*3) + (-0.03177*110)
Exercise4
```

```
## [1] 22.09908
```

##5. Run a multiple regression analysis on the myCars data with lmBF(), using mpg as the dependent variable and wt (weight) and hp (horsepower) as the predictors. Interpret the resulting Bayes factor in terms of the odds in favor of the alternative hypothesis. If you did Exercise 2, do these results strengthen or weaken your conclusions?

```
library(BayesFactor)
```

```
## Loading required package: coda

## Loading required package: Matrix

## ************
## Welcome to BayesFactor 0.9.12-4.3. If you have questions, please contact Richard Morey (richarddmorey@gmail.com).
##
## Type BFManual() to open the manual.
## ************
```

```
lmBF(mpg~wt+hp, data = myCars)
```

```
## Bayes factor analysis
## --------------
## [1] wt + hp : 788547604 ±0%
##
## Against denominator:
##   Intercept only
```

```
## ---
## Bayes factor type: BFlinearModel, JZS
```

#We see that the Bayes factor is 788547604, meaning that there are 788547604:1 odds in favor of the alternative hypothesis (our model with wt and hp) over the null hypothesis (intercept only model). In Exercise 2, we said that wt may be the best predictor of mpg in the mtcars data, and while we haven't necessarily proven that, we have shown here that this specific model that contains wt is a very strong one.

##6. Run lmBF() with the same model as for Exercise 4, but with the options posterior=TRUE and iterations=10000. Interpret the resulting information about the coefficients.

```
summary(lmBF(mpg~wt+hp, data = myCars,posterior = TRUE, iterations = 10000))
```

```
## 
## Iterations = 1:10000
## Thinning interval = 1 
## Number of chains = 1 
## Sample size per chain = 10000 
## 
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
## 
##          Mean        SD  Naive SE Time-series SE
## mu    20.0905  0.485830 4.858e-03      4.858e-03
## wt    -3.7741  0.669361 6.694e-03      6.694e-03
## hp    -0.0311  0.009488 9.488e-05      9.032e-05
## sig2   7.4697  2.150995 2.151e-02      2.662e-02
## g      3.9648 15.623983 1.562e-01      1.562e-01
## 
## 2. Quantiles for each variable:
## 
##           2.5%     25%      50%      75%    97.5%
## mu   19.13941 19.7740 20.09206 20.40863 21.04940
## wt   -5.05921 -4.2245 -3.78609 -3.33034 -2.43188
## hp   -0.04992 -0.0373 -0.03115 -0.02489 -0.01231
## sig2  4.40965  5.9507  7.08731  8.54627 12.73510
## g     0.36470  0.9518  1.71354  3.43545 18.80267
```

#The means for each of the variables are essentially Bayesian estimates for the population value of the B-weights for each corresponding coefficient. The first section also contains the standard deviation of each variable in the

```
list of 10000 values from our 10000 iterations, as well as the Naive and Time-series standard errors.
#The second section has the quantiles for each of the variables. These can be used to construct HDI's, with the most intriguing quantiles being the 2.5% and 97.5% quantiles, which allow us to find the 95% HDI for each coefficient's B-weight.
```

```
##7. Run install.packages() and library() for the "car" package. The car package is "companion to applied regression" rather than more data about automobiles. Read the help file for the vif() procedure and then look up more information online about how to interpret the results. Then write down in your own words a "rule of thumb" for interpreting vif.
#install.packages("car")
library(car)

## Loading required package: carData

help(vif)

## starting httpd help server ...

##  done

#Per Wikipedia, the variance inflation factor is the ratio of the variance of estimating some parameter in a model that includes multiple other terms by the variance of a model constructed using only one term. It quantifies the severity of multicollinearity in an ordinary least squares regression analysis. The square root of the variance inflation factor indicates how much larger the standard error increases compared to if that variable had 0 correlation to other predictor variables in the model. For example, if the variance inflation factor of a predictor variable were 5.27 (√5.27 = 2.3), this means that the standard error for the coefficient of that predictor variable is 2.3 times larger than if that predictor variable had 0 correlation with the other predictor variables.

#Multiple other online sources state that a variance inflation factor between 4 and 10 indicates a chance of mutlicollinearity, and that a vif of 10 or above indicates high multicollinearity.

#Rule of Thumb for Interpreting VIF:
#VIF = 1: no multicollinearity
#VIF >= 4 and < 10: chance of multicollinearity
#VIF >= 10: high multicollinearity
```

```
##8. Run vif() on the results of the model from Exercise 2. Interpret the
results. Then run a model that predicts mpg from all five of the predictors
in myCars. Run vif() on those results and interpret what you find.
vif(lm(mpg~wt + hp, data = myCars))

##       wt       hp
## 1.766625 1.766625

#There does not appear to be any multicollinearity in this model.
sqrt(vif(lm(mpg~wt + hp, data = myCars)))

##       wt       hp
## 1.329144 1.329144

#The standard error for both wt and hp is 1.329144 times larger than if they
had 0 correlation with each other.

vif(lm(mpg~., data = myCars))

##       cyl      disp        hp      drat        wt
##  7.869010 10.463957  3.990380  2.662298  5.168795

#Per our rule of thumb in Exercise 7, there is cause for concern for
multicollinearity with 4 out of 5 of these variables. We see that the vif is
above 10 for disp, which indicates high multicollinearity. The cyl and wt
variables have a vif above 4, which indicates a chance of multicollinearity,
although hp is almost there as well at 3.99.
sqrt(vif(lm(mpg~., data = myCars)))

##      cyl     disp       hp     drat       wt
## 2.805176 3.234804 1.997594 1.631655 2.273498

#The standard error for cyl is 2.805176 times larger than if it had 0
correlation with the other predictor variables in this model.
#The standard error for disp is 3.234804 times larger than if it had 0
correlation with the other predictor variables in this model.
#The standard error for hp is 1.997594 times larger than if it had 0
correlation with the other predictor variables in this model.
#The standard error for drat is 1.631655 times larger than if it had 0
correlation with the other predictor variables in this model.
#The standard error for wt is 2.273498 times larger than if it had 0
correlation with the other predictor variables in this model.
```