Multiple Regression/Linear Prediction Homework

Tim Hulak

```
options(scipen = 999)
library(BayesFactor)
```

```
## Loading required package: coda

## Loading required package: Matrix

## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard Morey (richarddmorey
##
## Type BFManual() to open the manual.
## ************
```

```
library(car)
```

```
## Loading required package: carData
```

I produced the material below with no assistance

Exercises 1-8 on pages 181-182 of *Reasoning with Data: An Introduction to Traditional and Bayesian Statistics Using R*

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]).

```
data("mtcars")
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```r
dim(mtcars)
```

```
## [1] 32 11
```

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

```
##                    mpg cyl disp  hp drat    wt
## Mazda RX4         21.0   6  160 110 3.90 2.620
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875
## Datsun 710        22.8   4  108  93 3.85 2.320
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215
## Hornet Sportabout 18.7   8  360 175 3.15 3.440
## Valiant           18.1   6  225 105 2.76 3.460
```

```r
dim(myCars)
```

```
## [1] 32  6
```

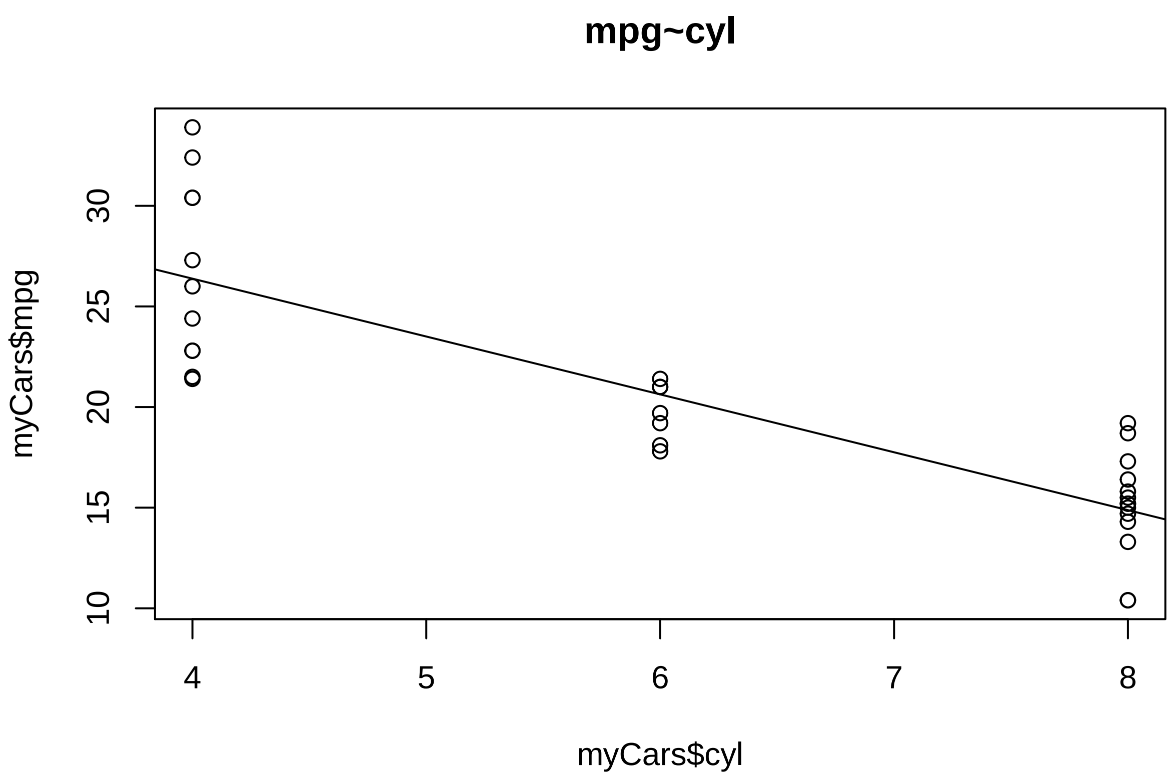
---

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?

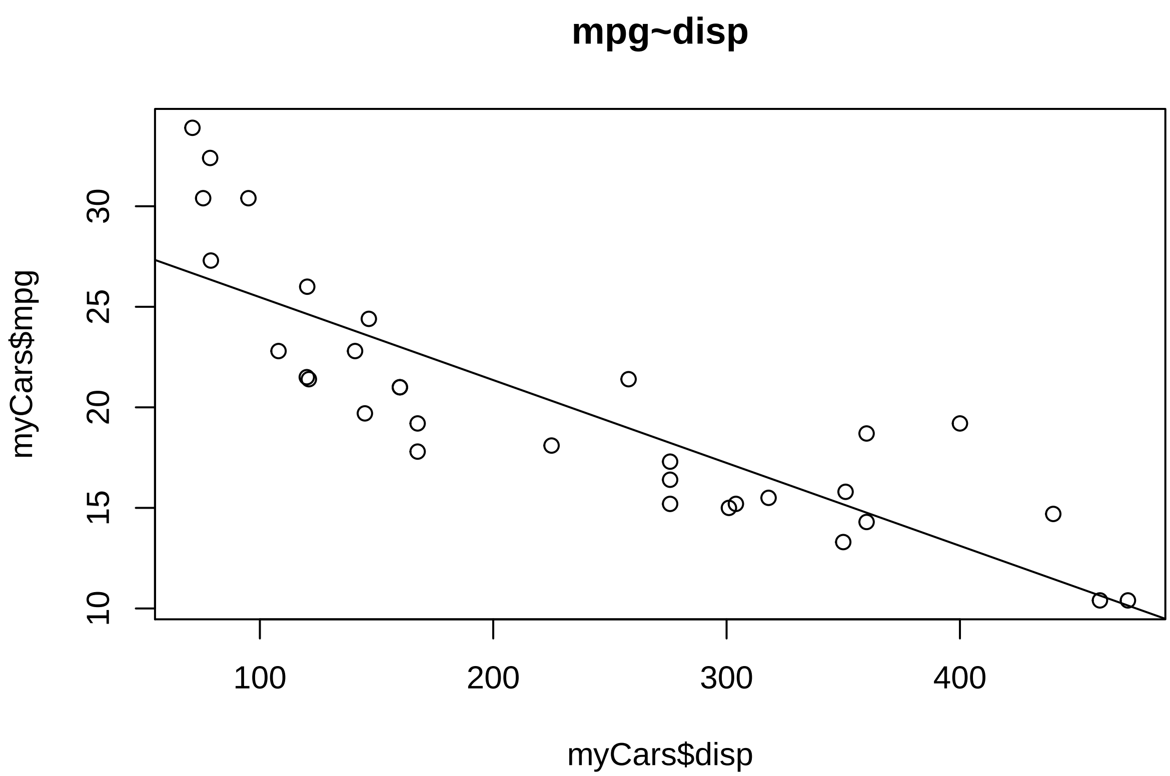
```r
cor(myCars)
```

```
##             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
```

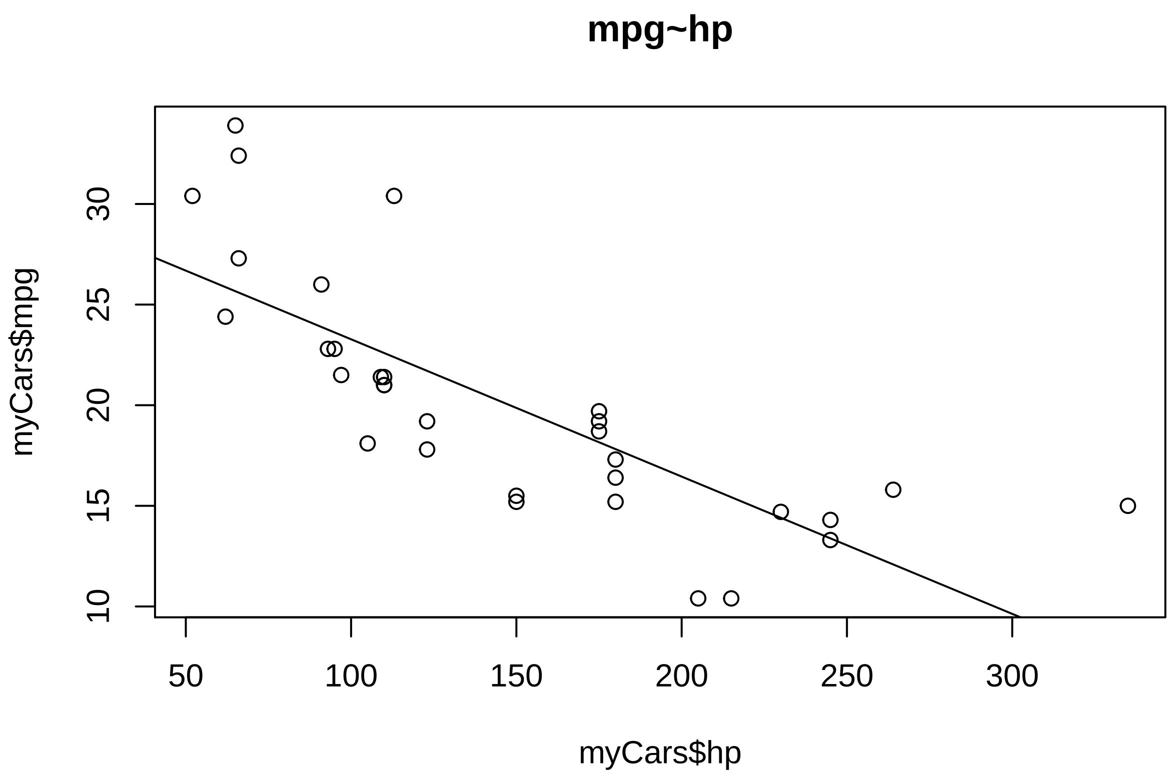
```r
plot(myCars$cyl, myCars$mpg, main="mpg~cyl")
abline(lm(mpg ~ cyl, data = myCars))
```



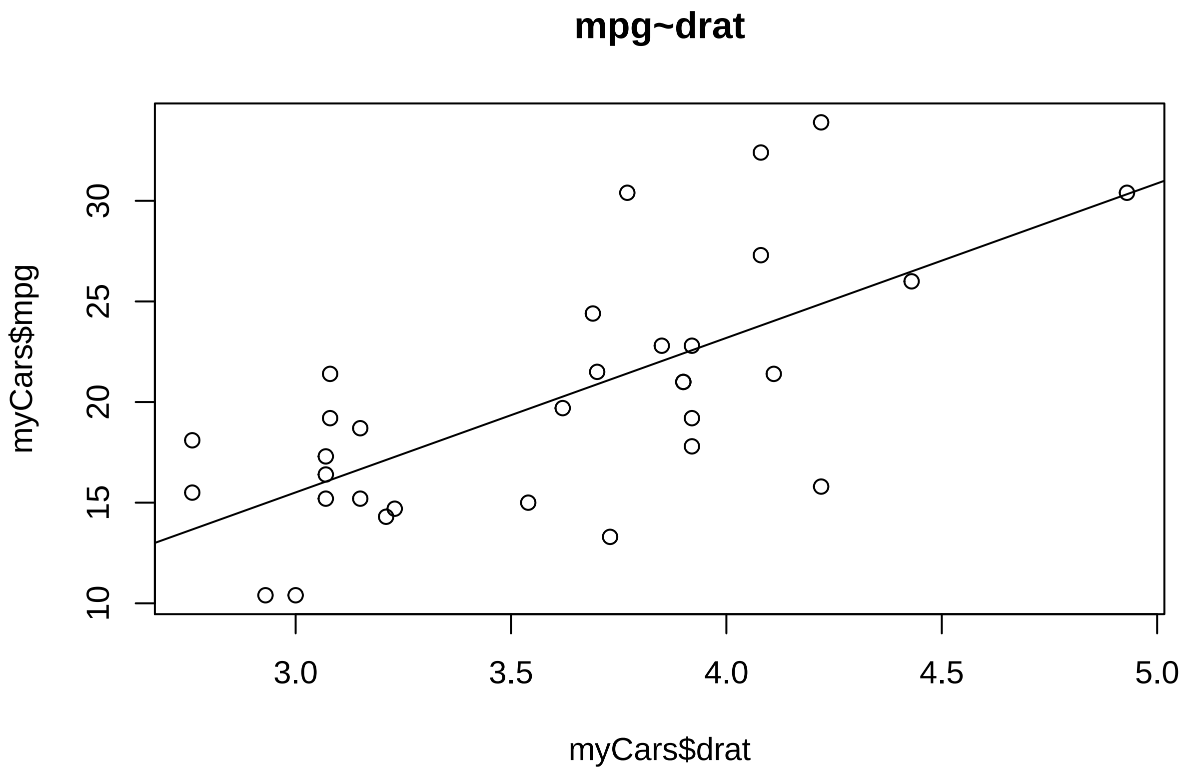

```
plot(myCars$disp, myCars$mpg, main="mpg~disp")
abline(lm(mpg ~ disp, data = myCars))
```



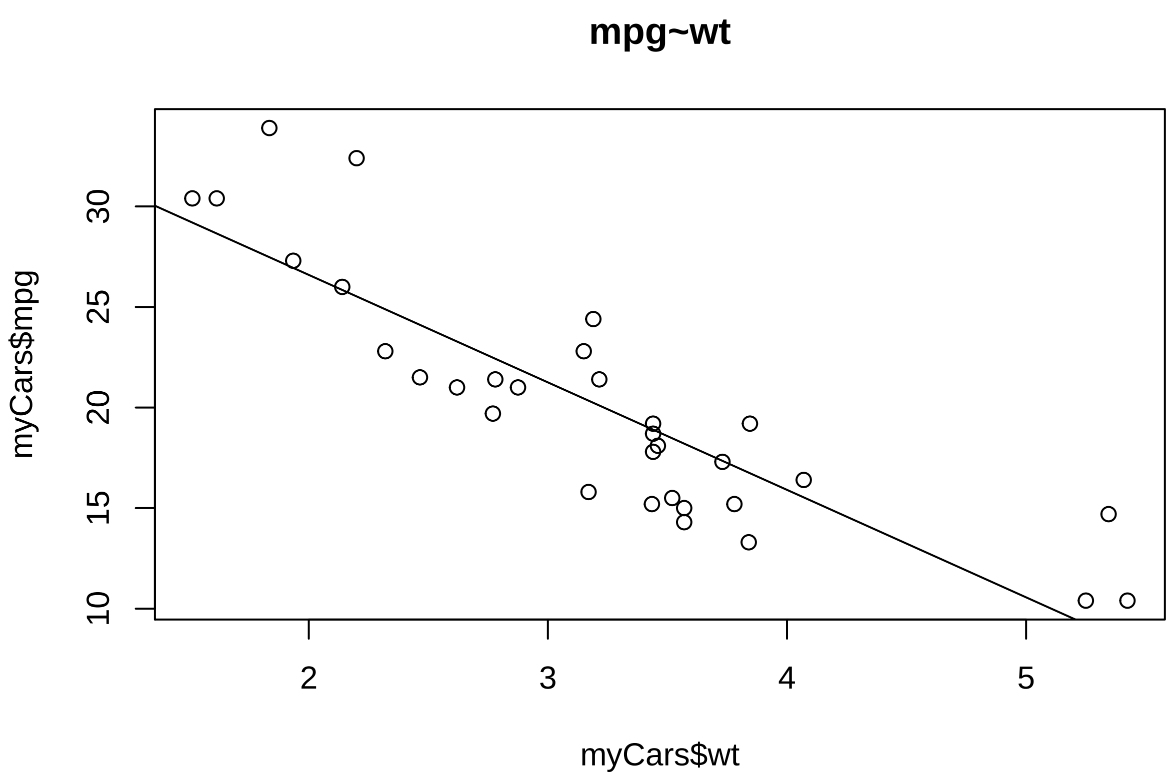
```
plot(myCars$hp, myCars$mpg, main="mpg~hp")
abline(lm(mpg ~ hp, data = myCars))
```




```
plot(myCars$drat, myCars$mpg, main="mpg~drat")
abline(lm(mpg ~ drat, data = myCars))
```




```
plot(myCars$wt, myCars$mpg, main="mpg~wt")
abline(lm(mpg ~ wt, data = myCars))
```



Answer: There appears to be a strong negative correlation between *mpg* and *disp* (**cor = -0.847**), *mpg* and *wt* (**cor = -0.867**), and *mpg* and *cyl* (**cor = -0.852**) This could be interpreted as the higher the *disp* or *wt*, the lower the *mpg*. There appears to be a strong positive correlation between *mpg* and *drat* (**cor = 0.681**). This could be interpreted as the higher the *drat*, the better the *mpg*. The strongest correlation is between *mpg* and *wt* (**cor = -0.867**) which means that *wt* might be the single best predictor of *mpg*.

---

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.

```r
initial_model <- lm(mpg ~ wt + hp, myCars)
summary(initial_model)
```

```
##
## 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 < 0.0000000000000002 ***
## wt          -3.87783    0.63273  -6.129           0.00000112 ***
## 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: 0.000000000009109
```

```r
model_1 <- lm(mpg ~ cyl + disp + hp + drat + wt, myCars)
summary(model_1)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + drat + wt, data = myCars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7014 -1.6850 -0.4226  1.1681  5.7263
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 36.00836    7.57144   4.756 0.000064 ***
## cyl         -1.10749    0.71588  -1.547  0.13394
## disp         0.01236    0.01190   1.039  0.30845
```

```
## hp          -0.02402    0.01328  -1.809  0.08208 .
## drat         0.95221    1.39085   0.685  0.49964
## wt          -3.67329    1.05900  -3.469  0.00184 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.538 on 26 degrees of freedom
## Multiple R-squared:  0.8513, Adjusted R-squared:  0.8227
## F-statistic: 29.77 on 5 and 26 DF,  p-value: 0.0000000005618
```

```r
model_2 <- lm(mpg ~ cyl + disp + hp  + wt, myCars)
summary(model_2)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + disp + hp + wt, data = myCars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0562 -1.4636 -0.4281  1.2854  5.8269
##
## Coefficients:
##             Estimate Std. Error t value           Pr(>|t|)
## (Intercept) 40.82854    2.75747  14.807 0.0000000000000176 ***
## cyl         -1.29332    0.65588  -1.972           0.058947 .
## disp         0.01160    0.01173   0.989           0.331386
## hp          -0.02054    0.01215  -1.691           0.102379
## wt          -3.85390    1.01547  -3.795           0.000759 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.513 on 27 degrees of freedom
## Multiple R-squared:  0.8486, Adjusted R-squared:  0.8262
## F-statistic: 37.84 on 4 and 27 DF,  p-value: 0.0000000001061
```

```r
model_3 <- lm(mpg ~ cyl + hp  + wt, myCars)
summary(model_3)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + hp + wt, data = myCars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9290 -1.5598 -0.5311  1.1850  5.8986
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept) 38.75179    1.78686  21.687 < 0.0000000000000002 ***
## cyl         -0.94162    0.55092  -1.709             0.098480 .
## hp          -0.01804    0.01188  -1.519             0.140015
## wt          -3.16697    0.74058  -4.276             0.000199 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.512 on 28 degrees of freedom
## Multiple R-squared:  0.8431, Adjusted R-squared:  0.8263
## F-statistic: 50.17 on 3 and 28 DF,  p-value: 0.00000000002184
```

```
model_4 <- lm(mpg ~ cyl + wt, myCars)
summary(model_4)
```

```
##
## Call:
## lm(formula = mpg ~ cyl + wt, data = myCars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2893 -1.5512 -0.4684  1.5743  6.1004
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept)  39.6863     1.7150  23.141 < 0.0000000000000002 ***
## cyl          -1.5078     0.4147  -3.636             0.001064 **
## wt           -3.1910     0.7569  -4.216             0.000222 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.568 on 29 degrees of freedom
## Multiple R-squared:  0.8302, Adjusted R-squared:  0.8185
## F-statistic: 70.91 on 2 and 29 DF,  p-value: 0.000000000006809
```

Answer: First, a linear model using mpg as the dependent variable and wt (weight) and hp (horsepower) as the predictors was built. The p-value of *wt* was **0.00000112** and the p-value or of *hp* was **0.00145**. This is evidence that both variables are statistically significant when predicting *mpg*. The R-squared value was **0.8268** and the Adjusted R-squared value was **0.8148**. This means that the model accounted for over 80% of the variability in the data.

For the sake of being thorough, all of the variables were passed into a linear model. This seemed to yield an R-squared of **0.8513** and an Adjusted R-squared of **0.8227** (meaning that the model is accounting for about 83% of the variability). The highest p-value was for *drat* at **0.49964**. So, *drat* was removed and the model was run again without it. In *model 2*, the highest p-value was *disp* at **0.331386** (which was also the second highest p-value in the first model). Therefore, *disp* was removed and the model was run again. In *model 3*, *hp* had a p-value of **0.140015**. *hp* was removed and the model was run for a fourth time with only *cyl* and *wt*. *Model 4* maintained an R-squared of **0.8302** and an Adjusted R-squared of **0.8185** (meaning that it accounted for around 82% of the variability in the data. which was better than the initial model of *hp* and *wt* predicting *mpg*). The p-value of *cyl* was **0.001064** and the p-value of *wt* was **0.000222** (incidentally, the p-value of *cyl* was above the traditional 0.05 threshold in the other 3 models). In *model 4*, both of those variables were statistically significant and the *wt* variable might be the single best predictor of mpg.

---

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.

```r
initial_model <- lm(mpg ~ wt + hp, myCars)
summary(initial_model)
```

```
##
## 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 < 0.0000000000000002 ***
## wt          -3.87783    0.63273  -6.129           0.00000112 ***
## 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: 0.000000000009109
```

```r
Intercept <- 37.22727
wt_B <- -3.87783
weight <- 3
hp_B <- -0.03177
horsepower <- 110

predicted_mpg <- Intercept + (wt_B*weight) + (hp_B*horsepower)
predicted_mpg
```

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

Answer: 22.09908 Miles Per Gallon is predicted for a car with 110 horsepower and a weight of 3 tons

---

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?

```r
model_bf <- lmBF(mpg ~ wt + hp, data=myCars,posterior=F)
summary(model_bf)
```

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

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

Answer: The ratio for the relationship between the variables is **12690355.3299492:1**. This means that there is strong evidence to reject the null hypothesis that there is no relationship between mileage and the variables. This strengthens the conclusions from Exercise 2 because we are able to reject the null hypothesis that is no relationship (meaning there *is* a relatonship)

---

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.

```
model_bf_2 <- lmBF(mpg ~ wt + hp, data=myCars,posterior=T, iterations=10000)
summary(model_bf_2)
```

```
##
## 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   20.09204  0.478936 0.00478936     0.00466042
## wt   -3.78024  0.659708 0.00659708     0.00711054
## hp   -0.03098  0.009452 0.00009452     0.00009452
## sig2  7.49308  2.148104 0.02148104     0.02536110
## g     4.10847 16.676811 0.16676811     0.17433685
##
## 2. Quantiles for each variable:
##
##          2.5%      25%      50%      75%    97.5%
## mu   19.14463 19.77924 20.09038 20.40776 21.03472
## wt   -5.08298 -4.20337 -3.77856 -3.36305 -2.47122
## hp   -0.04921 -0.03718 -0.03112 -0.02488 -0.01192
## sig2  4.34823  5.98107  7.16291  8.57827 12.63858
## g     0.35468  0.93711  1.73161  3.46094 19.72867
```

Answer: The HDI for *wt* had a lower bound of **-5.11112** and an upper bound of **-2.46356**. The HDI for *hp* had a lower bound of **-0.04944** and an upper bound of **-0.01246**. Neither of these straddle **0**, so we have a sense of certainty that the correlation is negative. There is strong evidence that we can reject the null hypothesis.

---

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.

```
?vif()
```

```
vif(initial_model)
```

```
##       wt       hp
## 1.766625 1.766625
```

Answer: According to the documentation, VIF (Variance Inflation Factors) "Calculates variance-inflation and generalized variance-inflation factors for linear, generalized linear, and other models." A "rule of thumb" for interpreting vif can be: 1 = not correlated, Between 1 and 5 = moderately correlated, and Greater than 5 = highly correlated. (Source: https://www.statisticshowto.com/variance-inflation-factor/) Since the values if the *wt* and *hp* is **1.766625**, this would be considered *moderately correlated*, if we go by that rule of thumb.

---

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(model_1)
```

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

Answer: If we go off of the rule of thumb (*1 = not correlated, Between 1 and 5 = moderately correlated, and Greater than 5 = highly correlated*), then the following can be assumed:

- mpg & cyl (7.869010): highly correlated because it is Greater than 5
- mpg & disp (10.463957): highly correlated because it is Greater than 5
- mpg & hp (3.990380): moderately correlated because it is Between 1 and 5
- mpg & drat (2.662298): moderately correlated because it is Between 1 and 5
- mpg & wt (5.168795): highly correlated because it is Greater than 5