Homework 8

Maya Mileva

due date: Nov 29, 2019

I did this homework by myself, with help from the book,the professor and www.displayr.com/variance-inflation-factors-vifs.

Exercises

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]).

```
## Create a new data frame
myCars <- data.frame(mtcars[,1:6])
head(myCars, n=2)
```

```
##               mpg cyl disp  hp drat    wt
## Mazda RX4      21   6  160 110  3.9 2.620
## Mazda RX4 Wag  21   6  160 110  3.9 2.875
```

```
dim(myCars) # 32 obs and 6 var
```

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

```
any(is.na(mtcars$mpg))
```

```
## [1] FALSE
```

```
var(myCars$mpg)
```

```
## [1] 36.3241
```

The data set comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models). The new data frame we created have the following columns:

[, 1] mpg Miles/(US) gallon

[, 2] cyl Number of cylinders

[, 3] disp Displacement (cu.in.)

[, 4] hp Gross horsepower

[, 5] drat Rear axle ratio

[, 6] wt Weight (1000 lbs)

```
str(myCars)
```

```
## 'data.frame':    32 obs. of  6 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
```

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)
```

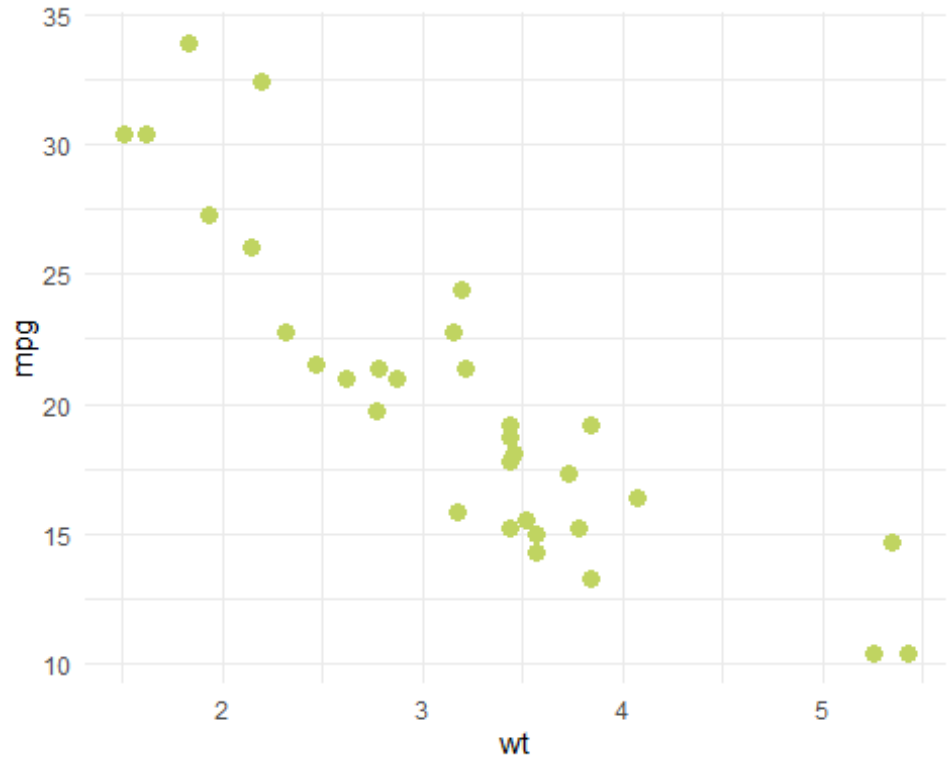
```
##             mpg        cyl       disp         hp       drat         wt
## mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.6811719 -0.8676594
## cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.6999381  0.7824958
## disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.7102139  0.8879799
##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

The Pearson Product-Moment Correlation, aka “r”, expresses the association between two metric variables on a scale of -1 to +1. Values of r near -1 or +1 are strong; values near 0 are weak.

In the example above we are calculating a cross product between scores and taking the average of the sum of a cross product in order to find out a measure of association between two metric variables. It starts off as a quantity called the covariance. And then we standardize it into a scale that goes from minus 1 to plus 1. And that is r, the Pearson product-moment correlation.

If we take a look at the correlation matrix, we will see 1`s along the diagonal representing a correlation between a variable and itself. And then it has two halves, the lower triangle and the upper triangle, which are mirror images of one another. They contain the same information. We will be trying to predict the mpg variable. wt has the highest correlation with mpg(-0.87), which might be the single best predictor of mpg. Looks like car’s weight might affects fuel efficiency. The rest of the variables has high correlation too, but lets examine wt some more.

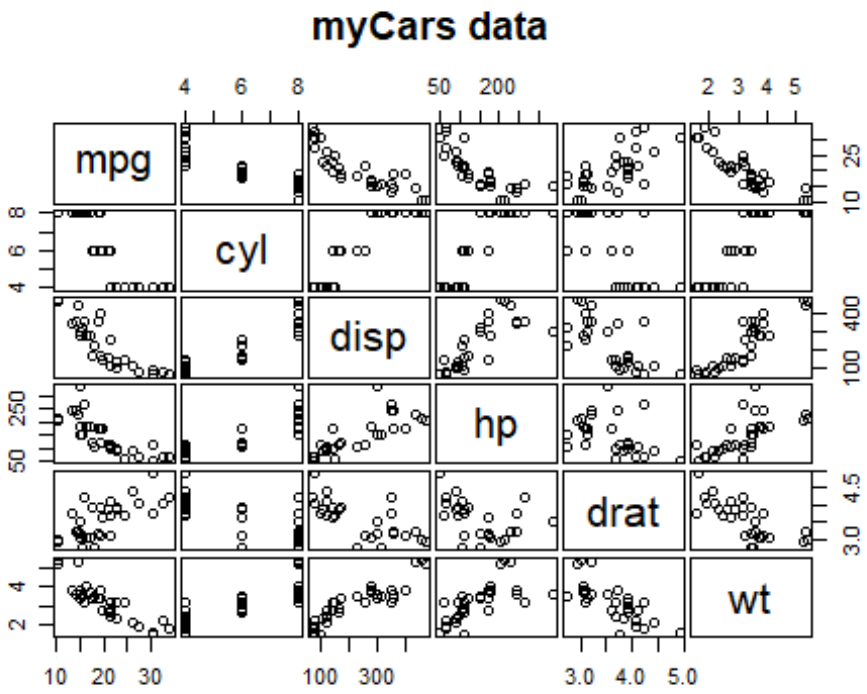
```
ggplot(myCars, aes(x=wt, y=mpg)) + geom_point(col = "#C0D461", size = 3) +
theme_minimal()
```



We can see a negative relationship: a higher weight means a higher miles per gallon and therefore a lower fuel efficiency in. We need to perform a statistical test to confirm that theory.

Lets look at all the variables.

```
pairs(myCars, main = "myCars data", gap = 1/4)
```



Pairs plot shows us the pattern of correlation using a scatterplot for each pair of variables. If we have a cigar-shaped scatterplot that points from the lower left the upper right, that's an indication of a strong positive variable. That same cigar shape pointing from the upper left down to the lower right indicates an inverted relationship, but again, a strong one. Any weak relationship between two variables and the scatterplot of points is more circular or more disordered. And that would indicate a relatively small value of r that's close to 0. We can see that disp,hp and drat have good correlation too with mpg.

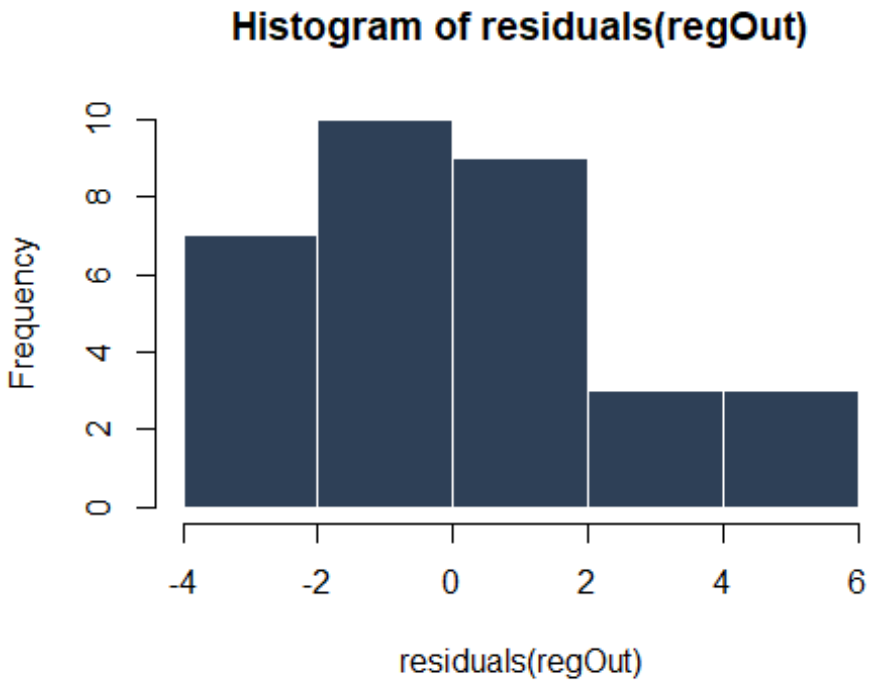
3. Run a multiple regression analysis on the myCars data with lm(), using mpg as the dependent variable and wt (weight) and hp (horsepower) as the predictors. Make sure to say whether or not the overall R-squared was significant. If it was significant, report the value and say in your own words whether 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.

```
## Predic mpg with wt (weight) and hp (horsepower)
regOut <- lm(mpg~ wt+hp, data = myCars)

## Show regression results
summary(regOut)

##
## 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
```

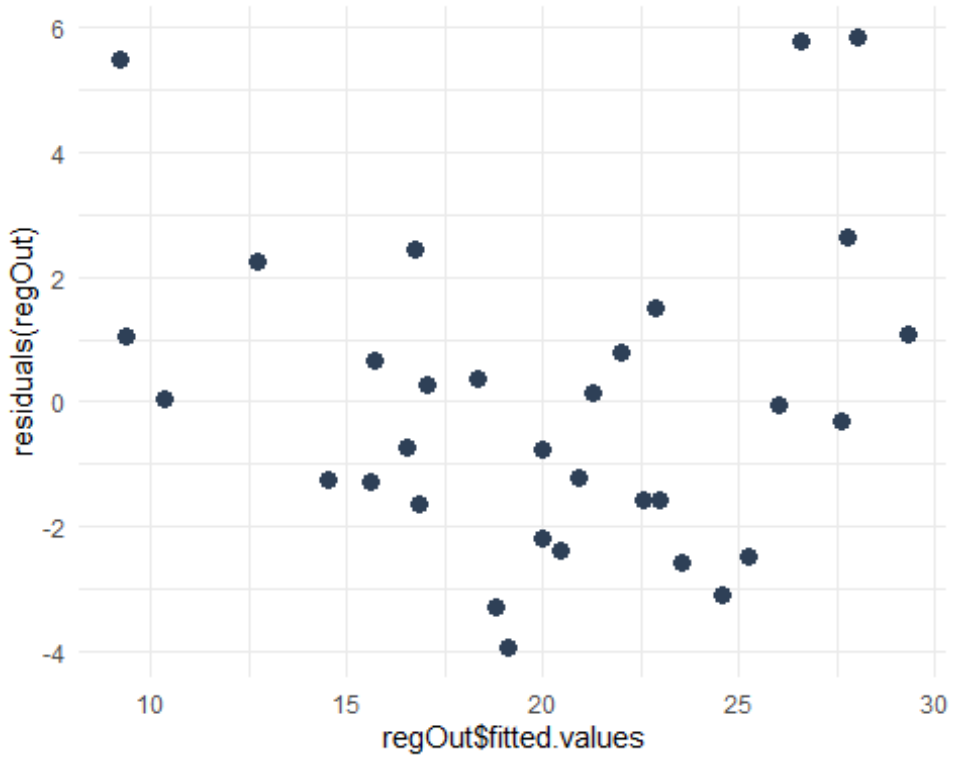
The first section is labeled residuals. Residuals are just the errors of prediction, and the lm() model generates a whole list of them in order to do its somewhat squared prediction errors calculation. In the output we see just a description of the distribution of them. It shows the minimum prediction error, the maximum prediction error, and then the first quartile, third quartile, and the median. They should be normally distributed with a median near zero. The fact that the median is -0.2 suggests that there is some skewness in the residuals, they are not that symmetrically distributed.

```
summary(residuals(regOut))

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -3.941  -1.600  -0.182   0.000   1.050   5.854

## Histogram of the residuals
hist(residuals(regOut)
     , col = "#2E4057"
     , border = "white"
     , main = "Histogram of residuals(regOut)")
```



We want to check and make sure that they are normally distributed and centered on 0 (that undicates that the regression model is doing good). Most of the cases are around -1, and the range that we saw earlier is between -3.94 and 5.85. Some improvement of the model can be done.

```
ggplot(regOut, aes(x=regOut$fitted.values, y=residuals(regOut))) +
  geom_point(colour = "#2E4057", size = 3) + theme_minimal()
```



Most of the values are on the negative side, between 0 and -2 like we saw in the histogram. There is an evidence of a curvilinear relationship (plot has an uneven distribution and the hint of a parabola). A curvilinear relationship is a type of relationship between two variables that has a pattern of correspondence or association between the two variables that change as the values of the variables change (increase or decrease).

```
## Check the correlation between the predictive variables
cor(myCars$wt, myCars$hp)

## [1] 0.6587479
```

We should begin interpreting a regression equation by making sure that the R squared is significant. If we look at the bottom, we see Multiple R squared of 0.8268 and an Adjusted R squared of 0.8148. Adjusted R-squared is slightly penalized relative to the multiple R-squared because we have two predictors. In social science, R squared could be anywhere from about 0.1 to about 0.5. In our case we have really high value, which means that hp and wt accounted for about 82% variability in mpg. That's backed up by the F-test. F-value is 69.1. That's a large F value, the expected F value is 1, so anything that's wildly in excess of 1 is likely to be significant. The degrees of freedom is 2 and 29. So the 2 refers to the number of the predictors, and 29 = 32-2-1, starting with 32 observation, 2 df are lost for the predictors and one for calculating the Y-intercept (F of 2 and 26 is equal to 69.21). The associated p-value is vanishingly small. It certainly is below that conventional threshold of 0.05 so we are rejecting the null hypothesis that R squared in the population is equal to 0. We've passed the omnibus test and we can go on and interpret the coefficients.

The coefficient section begins to show us the key results we need to know. The fist column is “Estimate” and show us the intercept in the first line and the slopes/B-weights on the predictors in the second and third line. So the upper number, 37.22, is the intercept, and the lower numbers, -3.88 and -0.3, are the B-weights. For each, the estimate is the statistical value of interest. The std.error estimates variability of the underlying sampling distribution. Together, these two values to calculate “t” and an associated null hypothesis test that each estimate coefficient is equal to 0. In all the cases, the value of “t” and the associated probability clearly indicates that we should reject the null hypothesis (p <.001 for intercept and wt, p<.01 for 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.

y(mpg) = 37.22727 -3.87783(wt)-0.03177(hp)

```
37.22727  -3.87783*3 -0.03177*110 # mpg = 22.09908
```

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

We have to consider the prediction errors when we are reporting those results.

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?

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

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

The Bayes factor is an odd ratio showing the likelihood of the stated alternative hypothesis (model with nonzero weights for the predictors) divided by the likelihood of the null model (intercept only model).

The results shows that the odds are in favor of the alternative hypotheses(very strong posite evidence that “wt” and “hp” are nonzero), in the sense that the model containing “wt” and “hp” as predictors is hugely favored over a model that only contains the Y-intercept.

Taken together, the result of the conventional analysis and the Bayesian analysis indicate that we we do have a solid predictive model witch weakens the conclusion we made earlier.

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.

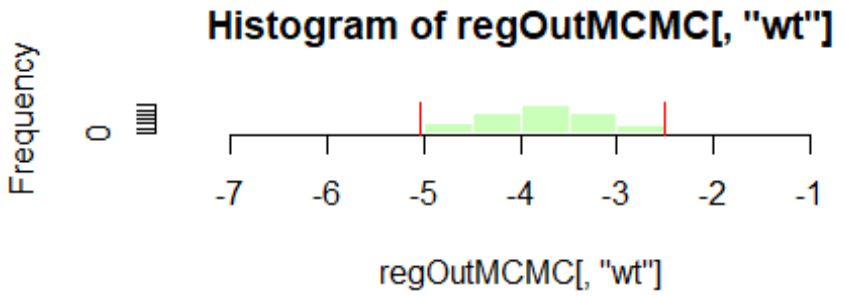
```
## Run a multiple regression analysis on the myCars data
regOutMCMC <- lmBF(mpg ~ wt + hp, data = myCars, posterior = TRUE,
                   iterations = 10000)
summary(regOutMCMC)

##
## 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   20.09277  0.491122 4.911e-03      4.829e-03
## wt   -3.78249  0.655739 6.557e-03      6.557e-03
## hp   -0.03094  0.009374 9.374e-05      9.374e-05
## sig2  7.51403  2.247293 2.247e-02      2.781e-02
## g     3.85664 15.344465 1.534e-01      1.534e-01
##
## 2. Quantiles for each variable:
##
##          2.5%     25%      50%      75%    97.5%
## mu   19.12664 19.7724 20.09611 20.41327 21.05938
## wt   -5.04891 -4.2085 -3.77985 -3.36049 -2.49853
## hp   -0.04915 -0.0373 -0.03103 -0.02472 -0.01213
## sig2  4.34911  5.9620  7.12863  8.63078 12.93048
## g     0.34193  0.9209  1.71810  3.46772 17.96518
```

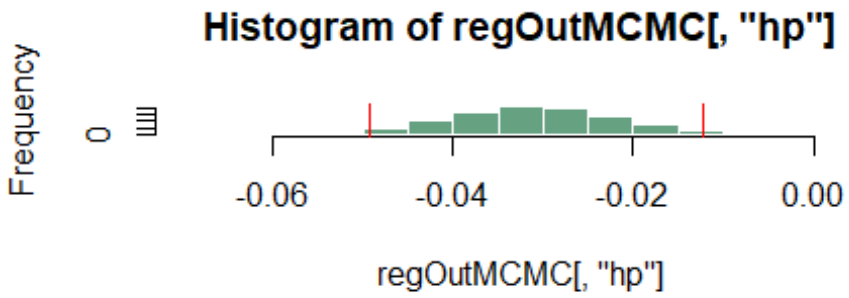
In the command above, we run lmBF() with posterior = TRUE and iterations= 10000 to sample from the posterior distribution using MCMC technique. Looking at the MCMC output first, we see both the means of the respective distribution and the 95% HDI.

If we compare the Bayesian output and the LM procedure, we need to look at the mean column. The values are really close (-3.78 compared to -3.88; -0.319 compared to -0.317). The top line of the tables is labeled mu - the grand mean for the data set. There are two specific columns: Naive SE and Time-series SE. SE stands for standard error (indication of dispersion in a sampling distribution). Naive SE calculates the standard error in the normal way. The Time-series SE accounts for the fact that there might be some, autocorrelation among the different estimates of the standard error.

The second table is overview of the highest density interval. We have the 2.5% and 97.5% boundaries of the HDI for each of the B-weights. These boundaries mark the edges of the central region of the posterior distribution for each B-weight. sig2 is the error variance in the regression equation. And we can use that error variance to calculate a distribution of r-squared values.

```
par(mfrow = c(2,1))
hist(regOutMCMC[,"wt"], col = "#CAFFB9", border = "white")
abline(v=quantile(regOutMCMC[,"wt"], c(0.025)), col = "red")
abline(v=quantile(regOutMCMC[,"wt"], c(0.975)), col = "red")

hist(regOutMCMC[,"hp"], col = "#66A182", border = "white")
abline(v=quantile(regOutMCMC[,"hp"], c(0.025)), col = "red")
abline(v=quantile(regOutMCMC[,"hp"], c(0.975)), col = "red")
```

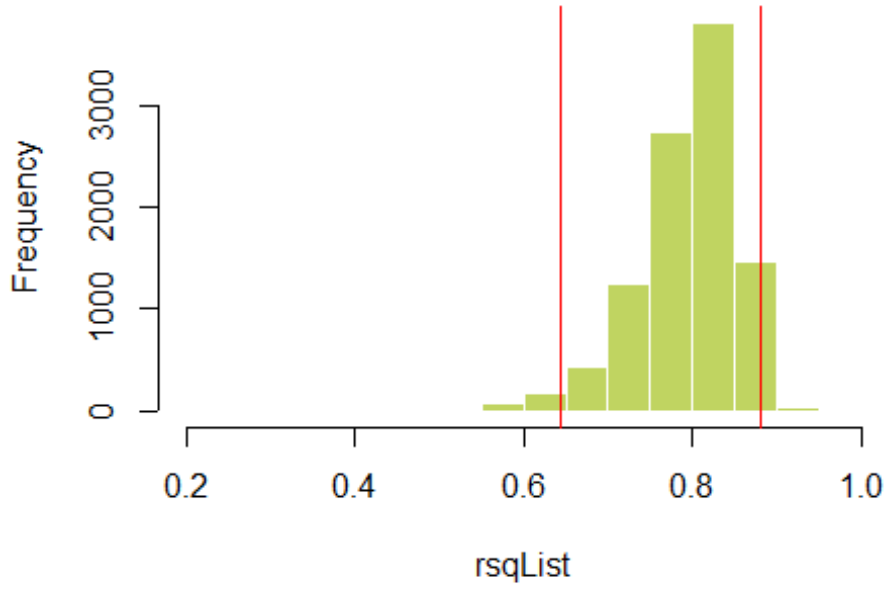




We can see pretty symmetric distribution and the lower and upper bounds of the HDI. The intervals does not overlap with 0, providing evidence that the population value of those B-weights credibly differ from 0.

```
## Calculate a distribution of r-squared values
rsqList <- 1 - (regOutMCMC[,"sig2"] / 36.3241)
hist(rsqList
     , xlim = c(0.2,1)
     , main = ""
     , col = "#C0D461", border = "white")
mean(rsqList) # 0.79
```

```
## [1] 0.7931394

## Draw boundaries of the 95% HDI
abline(v=quantile(rsqList,c(0.025)), col="red")
abline(v=quantile(rsqList,c(0.975)), col="red")
```



The mean value of this distribution came out to 0.79, which is slightly lower that the R-square that we obtained from the lm() model, but closer to the Adjuster R-square(0.81) we obtained from that model. The mean B-weight on "wt" is also smaller in magnitude for the Bayesian analysis, than for the conventional analysis (-3.78 compared to -3.88) which might be the reason for that difference in the R-square. Bayesian analysis finds that "wt" is not as strong predictor as the conventional analysis indicated. Bayesian model can give us a clear-eyed view of the likely range of the possibilities for the predictive strength of our model. In the underlying population, it is credible for us to expect R-square as low as about 0.64 or as high as about 0.88, with the most likely value pf R-square in that central region surrounding 0.79.

These results strengthen the conclusions we made. Using the "frequentist" (traditional) null hypothesis test, we rejected the null hypothesis for all two predictors as well as the overall R-squared. The Highest Density Intervals (HDIs) from the MCMC output showed estimates for the coefficients and R squared that concur with the frequentist model. The Bayes Factor overwhelmingly favored a model that includes the two predictors.

Summary of the results

We tested a model of fuel consumption that used two variables to predict mpg(miles per gallon): hp (horse power) and wt(weight). A Bayesian analysis of this model showed a mean posterior estimate for R-squared of 0.79, with the highest density interval ranging from roughly 0.64 to 0.88. The traditional analysis confirmed this result with a slightly more optimistic R-squared of 0.81. The F-test on this value was $F(2, 29)=69.21$, $p<.05$, so we reject the null hypothesis that R-squared was equal to zero. All three predictors were also significant with B-weights of -3.88 (weight)and -0.03 (horse power). The Bayes factor of 788547604 was strongly in favor of the two predictor model (in comparison with an intercept-only model).

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.

Multicollinearity problems consist of including, in the model, different variables that have a similar predictive relationship with the outcome. This can be assessed for each predictor by computing the VIF value. The variance inflation factor (VIF) quantifies the extent of correlation between one predictor and the other predictors in a model. It is used for diagnosing collinearity/multicollinearity. Higher values signify that it is difficult to impossible to assess accurately the contribution of predictors to a model.

How to interpret the VIF

A VIF can be computed for each predictor in a predictive model. A value of 1 means that the predictor is not correlated with other variables. The higher the value, the greater the correlation of the variable with other variables. Values of more than 4 or 5 are sometimes regarded as being moderate to high, with values of 10 or more being regarded as very high. These numbers are just rules of thumb; in some contexts a VIF of 2 could be a great problem (e.g., if estimating price elasticity), whereas in straightforward predictive applications very high VIFs may be unproblematic.

If one variable has a high VIF it means that other variables must also have high VIFs. In the simplest case, two variables will be highly correlated, and each will have the same high VIF.

Where a VIF is high, it makes it difficult to disentangle the relative importance of predictors in a model, particularly if the standard errors are regarded as being large. This is particularly problematic in two scenarios, where:

The focus of the model is on making inferences regarding the relative importance of the predictors. The model is to be used to make predictions in a different data set, in which the correlations may be different. The higher the VIF, the more the standard error is inflated, and the larger the confidence interval and the smaller the chance that a coefficient is determined to be statistically significant.

Ref: https://www.displayr.com/variance-inflation-factors-vifs/

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(regOut)

##       wt       hp
## 1.766625 1.766625
```

The Variance Inflating Factor (VIF) tells you how much higher the variance are when "wt" and "hp" are correlated compared to when they are uncorrelated. In our case, the variance is higher by a factor 1.7. High VIFs are a sign of multicollinearity.

We established that a value of 1 means that the predictor is not correlated with other variables, there are only 2 variables so their VIF score is the same(1.76), slightly higher than 1. VIF of 1,76 tells us that the variance of a particular coefficient is around 90% bigger than what you would expect if there was no multicollinearity — if there was no correlation with other predictors.

The variance inflation factor of a predictor variables is 1.76 ($\sqrt{1.76} = 1.33$), this means that the standard error for the coefficient of that predictor variable is 1.33 times larger than if that predictor variable had 0 correlation with the other predictor variables.

```
## Model that predicts mpg from all five of the predictors in myCars
regOut1 <- lm(mpg ~ wt + hp + cyl + disp + drat, data = myCars)
vif(regOut1)

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

In this example, the VIF score for the predictor variable "dist" is **very high**(VIF = 10.46). This might be problematic, makes it difficult to disentangle the relative importance of the predictor in the model. The score for "cyl" is **high**. Variables "wt" score can be considered **moderate**, variable "hp" is almost at the 4 point border. The higher the value, the greater the correlation of the variable with other variables. Predictor variable "drat" seems to be less correlated with the rest. We have to mention that these numbers are just rules of thumb; in some contexts a VIF of 2 could be a great problem.

Any variable with a high VIF value (above 5 or 10) can be removed from the model. This can leads to a simpler model without compromising the model accuracy, which is good.