Module 4 Lab Submission

Ben Tankus

First, we will explore the Brain size data in the data set `case0902` from the `Sleuth3` library. You can read more about this data set by viewing the help file:

```
help(case0902)
```

```
## starting httpd help server ... done
```

```
head(case0902)
```

```
##            Species  Brain    Body Gestation Litter
## 1         Aardvark    9.6    2.20        31    5.0
## 2         Acouchis    9.9    0.78        98    1.2
## 3 African elephant 4480.0 2800.00       655    1.0
## 4          Agoutis   20.3    2.80       104    1.3
## 5        Axis deer  219.0   89.00       218    1.0
## 6           Badger   53.0    6.00        60    2.2
```

1. **Fit a linear model with `Brain` as the response variable, and `Body`, `Gestation`, and `Litter` as the predictor variables.**

```
fit <- lm(Brain ~ Body + Gestation + Litter , data = case0902)
summary(fit)
```

```
##
## Call:
## lm(formula = Brain ~ Body + Gestation + Litter, data = case0902)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1026.68   -62.08    17.29    51.73   988.76
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -225.29213   83.05875  -2.712  0.00797 **
## Body           0.98588    0.09428  10.457  < 2e-16 ***
## Gestation      1.80874    0.35445   5.103 1.79e-06 ***
## Litter        27.64864   17.41429   1.588  0.11579
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 224.6 on 92 degrees of freedom
## Multiple R-squared:   0.81,  Adjusted R-squared:  0.8038
## F-statistic: 130.7 on 3 and 92 DF,  p-value: < 2.2e-16
```

2. **Calculate the case influence measures for this model using the `augment()` function from the package `broom`. Which species has the highest leverage for this model? Which species has the highest Cook's Distance?**

```
augFit <- augment(fit)
augFit$species <- case0902$Species
head(augFit)
```

```
## # A tibble: 6 x 11
##    Brain    Body Gestation Litter .fitted .resid .std.resid    .hat .sigma
##    <dbl>   <dbl>     <int>  <dbl>   <dbl>  <dbl>      <dbl>   <dbl>  <dbl>
## 1    9.6    2.2         31    5     -28.8    38.4      0.174  0.0361   226.
## 2    9.9    0.78        98    1.2   -14.1    24.0      0.108  0.0288   226.
## 3 4480   2800          655    1    3748.    732.       6.16   0.719    173.
## 4   20.3    2.8        104    1.3     1.52   18.8      0.0847 0.0250   226.
## 5  219     89          218    1     284.    -65.4     -0.294  0.0189   226.
## 6   53      6           60    2.2   -50.0   103.       0.465  0.0263   226.
## # ... with 2 more variables: .cooksd <dbl>, species <fct>
```

```
max_lev <- max(augFit$.hat)

augFit[augFit$.hat == max_lev, ]
```

```
## # A tibble: 1 x 11
##   Brain  Body Gestation Litter .fitted .resid .std.resid  .hat .sigma .cooksd
##   <dbl> <dbl>     <int>  <dbl>   <dbl>  <dbl>      <dbl> <dbl>  <dbl>   <dbl>
## 1  4480  2800       655      1   3748.   732.       6.16 0.719   173.    24.3
## # ... with 1 more variable: species <fct>
```

```
augFit[augFit$.cooksd == max(augFit$.cooksd), ]
```

```
## # A tibble: 1 x 11
##   Brain  Body Gestation Litter .fitted .resid .std.resid  .hat .sigma .cooksd
##   <dbl> <dbl>     <int>  <dbl>   <dbl>  <dbl>      <dbl> <dbl>  <dbl>   <dbl>
## 1  4480  2800       655      1   3748.   732.       6.16 0.719   173.    24.3
## # ... with 1 more variable: species <fct>
```

Elephants have the highest leverage at 0.72, and also the highest cook's distance at 24.29

Now we will continue investigating multicollinearity. Recall the simulated scenario considered in the `M4Lab-examples.Rmd` file, where we followed these steps:

1. Define $\beta_0 = 0.5$, $\beta_1 = 0.3$, and $\beta_2 = 0.7$
2. Define the mean of X_1 and X_2
3. Generate correlated/uncorrelated X_1 and X_2 data
4. Generate the response variable; use model equation and add N(0,1) noise
5. Fit a MLR model
6. Extract the coefficient estimate; $\hat{\beta}_0$, $\hat{\beta}_1$, or $\hat{\beta}_2$
7. Repeat steps (4) through (6) many times.

We used a function, included here, to perform steps 4. through 6., and then repeated that function many times (step 7.)

```
fitmodel <- function(X1, X2, beta0, beta1, beta2){
  n <- length(X1)
  Y <- beta0 + beta1*X1 + beta2*X2 + rnorm(n, 0, 1) # Generate/calculate response
  fit <- lm(Y ~ X1 + X2) # Fit the model
  fit$coefficients # Return estimated coefficient values
}
```

To run this function, we have to define the coefficient values (Step 1.), and set the mean and covariance matrix to generate predictor variables (Steps 2. and 3.).

```
# Step 1
beta0 <- 0.5 # define beta_0
beta1 <- 0.3 # define beta_1,
beta2 <- 0.7 # define beta_2

# Step 2
mu <- matrix(c(0,0)) # Set means for X_1, X_2
sigma1 <- matrix(c(1, 0, 0, 1), ncol = 2) # Cov Matrix: Cov(X_1, X_2) = 0
sigma2 <- matrix(c(1, 0.9, 0.9, 1), ncol = 2) # Cov Matrix: Cov(X_1, X_2) = 0


# Step 3
set.seed(1822) # Francis Galton born, invented regression concept

#UNCORRELATED
n <- 250
X <- mvrnorm(n, mu=c(0,0), Sigma=sigma1)
X1 <- X[,1]
X2 <- X[,2]

#CORRELATED
X2_2 <- mvrnorm(n, mu=c(0,0), Sigma=sigma2)
X1_2 <- X2_2[,1]
X2_2 <- X2_2[,2]


# Step 7
beta_estimates <- replicate(10000, fitmodel(X1, X2, beta0, beta1, beta2))
beta_estimatesCOV <- replicate(10000, fitmodel(X1_2, X2_2, beta0, beta1, beta2))

#fitmodel(X1, X2, beta0, beta1, beta2)
```

Finally, we calculated the standard deviation of the estimates of β_0 that resulted from these simulated datasets:

```
print(’uncorrelated’)
```

```
## [1] "uncorrelated"
```

```
print(paste("Intercept:", sd(beta_estimates[1,])))
```

```
## [1] "Intercept: 0.0634905573129528"
```

```
print(paste("X1:", sd(beta_estimates[2,])))
```

```
## [1] "X1: 0.0613978756737243"
```

```
print(paste("X2:", sd(beta_estimates[3,])))
```

```
## [1] "X2: 0.061527906687893"
```

```
print(’’)
```

```
## [1] ""
```

```
print(’correlated’)
```

```
## [1] "correlated"
```

```
print(paste("Intercept:", sd(beta_estimatesCOV[1,])))
```

```
## [1] "Intercept: 0.0625873800650354"
```

```
print(paste("X1:", sd(beta_estimatesCOV[2,])))
```

```
## [1] "X1: 0.141096204862987"
```

```
print(paste("X2:", sd(beta_estimatesCOV[3,])))
```

```
## [1] "X2: 0.141873309268063"
```

3. **Now it is your turn to calculate the standard deviation of the estimates of β_1 and β_2 in the uncorrelated case; and β_0, β_1, and β_2 in the correlated case. As you run the simulations, fill in the standard errors in the table below. Note: In the correlated case, use `sigma2 <- matrix(c(1, 0.9, 0.9, 1), ncol = 2)` to define the covariance matrix.**

Parameter	$SE(\hat{\beta}_i)$
Uncorrelated	
β_0	0.0635
β_2	0.614
β_3	0.062
Correlated	
β_0	0.0625
β_2	0.141
β_3	0.142

4. **The variances (and therefore standard deviations) of $\hat{\beta}_1$ and $\hat{\beta}_2$ are much larger when X_1 and X_2 are correlated than when they are uncorrelated. Does it make sense that $\hat{\beta}_0$ is unaffected? Explain your reasoning.**

$\hat{\beta}_0$ being unaffected by correlation is not a suprise. This occurs based on the mathematical equation for standard error. The two terms on the right simplify to 1, leaving only SE = $\hat{\sigma}$.

$$SE = \hat{\sigma} * \sqrt{\frac{1}{1 - R^2_{x_j}}}\sqrt{\frac{1}{\sum(X_ij - \bar{X}_j)^2}}$$

5. **Recall the sample VIFs calculated (in `M4Lab-examples.Rmd`) for some simulated data in the correlated case:**

```
      X1       X2
5.304359 5.304359
```

Compare the variances (*squared standard deviations*) in the table above for the correlated predictor setting to the variances for the uncorrelated predictor setting: what is the ratio of the variance of $\hat{\beta}_1$ in the correlated predictor setting to the variance of $\hat{\beta}_1$ in the uncorrelated predictor setting? Similarly, what is the variance of $\hat{\beta}_2$ in the correlated predictor setting to the variance of $\hat{\beta}_2$ in the uncorrelated predictor setting? Do these ratios seem close to the VIFs that we calculated?