BTRY 4030 - Fall 2018 - Homework 2 Solutions

Greg Benton gwb67

Due Friday, September 28, 2018

Instructions:

Create your homework solution file by editing the "h21-2018_q1.Rmd" Rmarkdown file provided. Your solution to this homework assignment should include the relevant R code and output (fit summaries, ANOVA tables and computed statistics, as well as requested plots) in addition to written comments where requested. Do not include output that is not relevant to the question. You should turn in a .pdf version of your compiled code.

You may discuss the homework problems and computing issues with other students in the class. However, you must write up your homework solution on your own. In particular, do not share your homework RMarkdown file with other students.

Question 1

This question follows on from Q2 of HW 1. As in that case, the file "brainweight.txt" contains measurements of **brain weight** (B, in kilograms), **body weight** (W, in kilograms), and **gestation period** (G, in days) of a sample of 96 mammal species. The following commands can be used to read the data into R (although you will need point the **setwd** command to the directory on your computer that contains the data file.).

Here we will continue to look at multiple linear regression and how association between variables impacts inference. Question 2 will explore these results mathematically.

```
brain_data =read.table('BrainWeight.txt',head=TRUE)
colnames(brain_data)=c("B","W","G")
```

a. Fit a regression model to predict log(B) from log(W). Create the hat matrix for this regression.

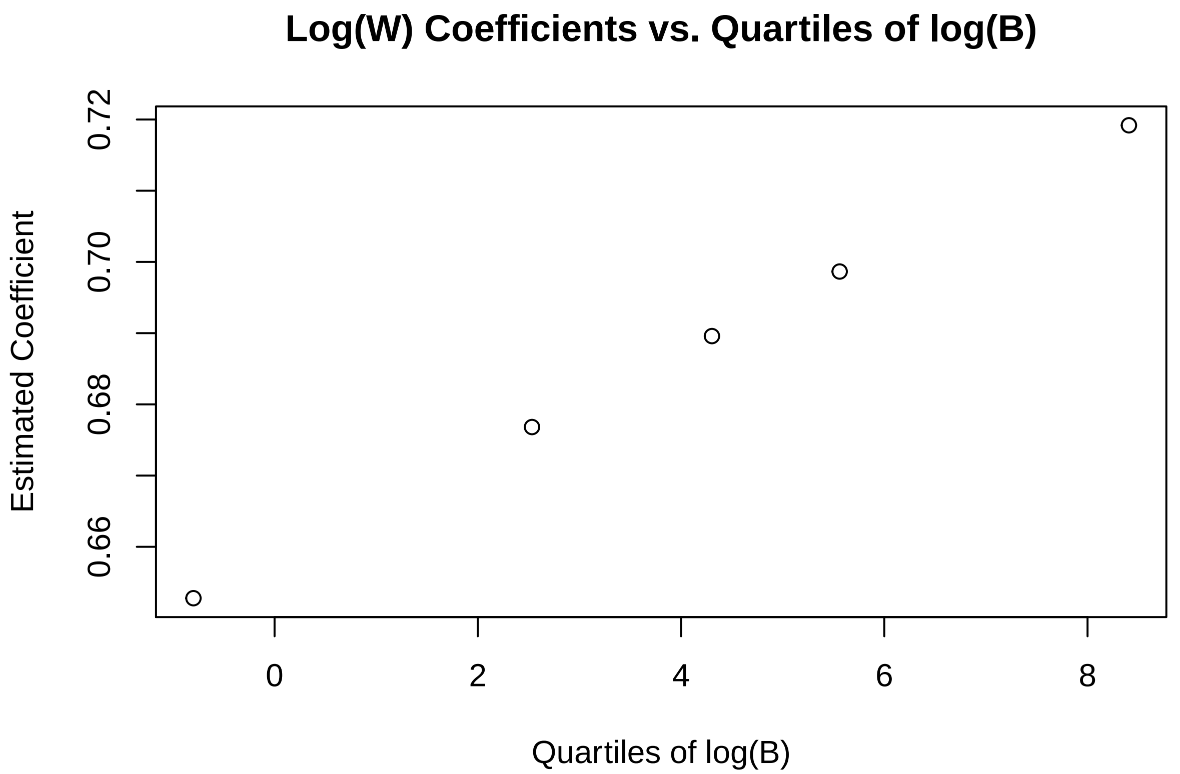
```
model = lm(B ~ W, data=log(brain_data)) # regression model
X = cbind(rep(1, nrow(brain_data)), log(brain_data$W)) # design matrix
hat_mat = X %*% solve(t(X) %*% X) %*% t(X) # hat matrix
```

b. Find the observation with the largest value of W, replace it's value of log(B) with each of the quartiles of log(B) in turn.
   For each quartile, re-fit the regression line. Plot the estimated coefficient of log(W) for each of these four models versus the corresponding quartile of log(W). Similarly, plot the fitted value for the observation you manipulated versus the quartile of log(W). Show that the slope of this line is equal to the leverage (the diagonal element of the Hat matrix) for that point.

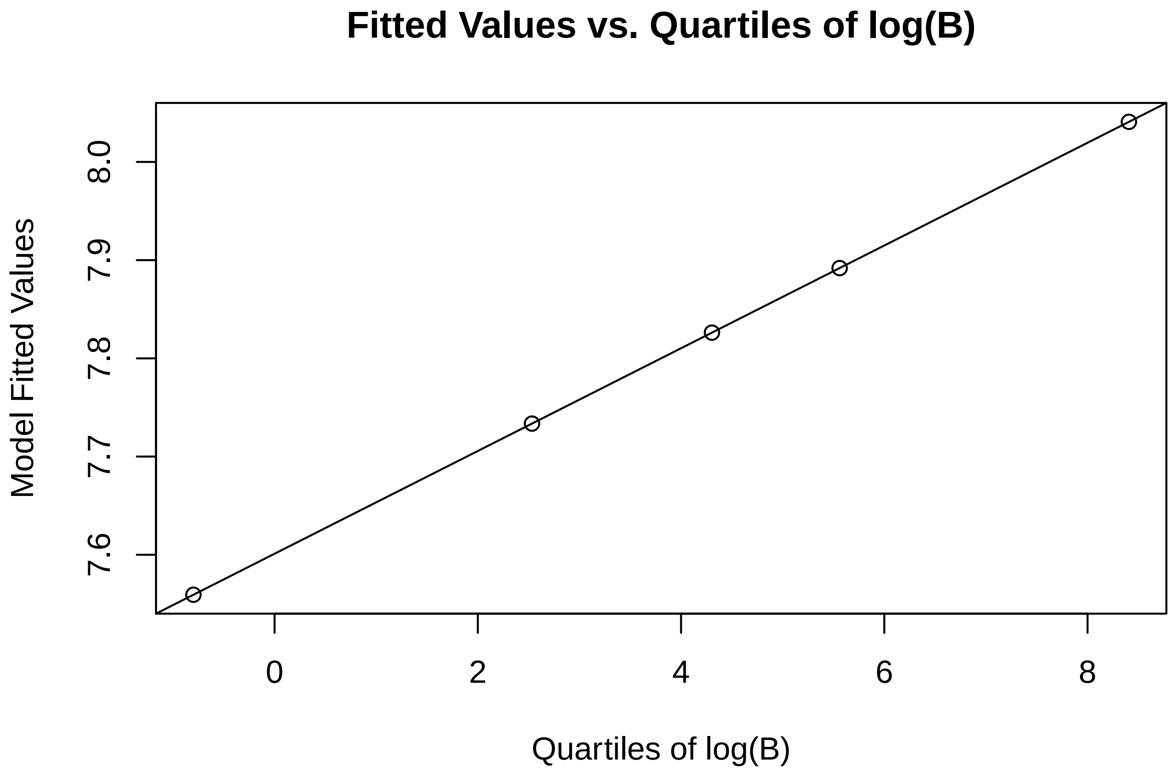
```
max_ind = which.max(brain_data$W)
quant_probs = c(0, 0.25, 0.5, 0.75, 1)
n_quants = length(quant_probs)
log_b_quartiles = quantile(log(brain_data$B), probs=quant_probs)
log_w_coeffs = rep(NA, n_quants)
max_ind_fitted_vals = rep(NA, n_quants)

for (ii in 1:n_quants) {
    temp_data = log(brain_data)
    temp_data$B[max_ind] = log_b_quartiles[ii]
    model = lm(B ~ W, data=temp_data)
    log_w_coeffs[ii] = model$coefficients[2]
    max_ind_fitted_vals[ii] = model$fitted.values[max_ind]
}
log_w_quartiles = quantile(log(brain_data$W), probs=quant_probs)

## plot the log(w) coeffs vs. log(b) quartiles ##
plot(log_w_coeffs ~ log_b_quartiles,  main="Log(W) Coefficients vs. Quartiles of log(B)",
    ylab = "Estimated Coefficient", xlab="Quartiles of log(B)")
```



```
## plot the adjusted fitted values against log(b) quartiles ##
leverage_model = lm(max_ind_fitted_vals ~ log_b_quartiles)
plot(max_ind_fitted_vals ~ log_b_quartiles,  main="Fitted Values vs. Quartiles of log(B)",
    ylab = "Model Fitted Values", xlab="Quartiles of log(B)")
abline(leverage_model)
```



```
print(paste("Difference between empirical leverage and diagonal entry of hat matrix:",
            (leverage_model$coefficients[2]-hat_mat[max_ind, max_ind])))
```

```
## [1] "Difference between empirical leverage and diagonal entry of hat matrix: 2.70616862252382e-16"
```

The above value indicates that the difference between the leverage calculated from the slope in the above plot is numerically equivalent to the diagonal entry of the hat matrix.

c. Now we will consider fitting log(B) to both log(G) and log(W). Do this using the **lm** function two ways: 1. with log(W) listed before log(G) and 2. with log(G) before log(W). Use the **anova** command to obtain sequential anova tables for both models; do thse give different sum of squares?

```
model1 = lm(B ~ G + W, data = log(brain_data))
model2 = lm(B ~ W + G, data = log(brain_data))

anova(model1)
```

```
## Analysis of Variance Table
##
## Response: B
##           Df Sum Sq Mean Sq F value    Pr(>F)
## G          1 355.74  355.74 1480.37 < 2.2e-16 ***
## W          1  69.72   69.72  290.13 < 2.2e-16 ***
## Residuals 93  22.35    0.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(model2)
```

```
## Analysis of Variance Table
##
## Response: B
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## W          1 416.40  416.40 1732.785 < 2.2e-16 ***
## G          1   9.06    9.06   37.713 2.002e-08 ***
## Residuals 93  22.35    0.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

These give different sums of squares for the individual predictors, but not for the residuals.

d. Show that the **Anova** command in the **car** package gives the same non-sequential sums of squares for both models.

```r
library(car)
```

```
## Warning: package 'car' was built under R version 3.5.1
```

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

```r
Anova(model1)
```

```
## Anova Table (Type II tests)
##
## Response: B
##           Sum Sq Df F value    Pr(>F)
## G          9.063  1  37.713 2.002e-08 ***
## W         69.719  1 290.126 < 2.2e-16 ***
## Residuals 22.349 93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(model2)
```

```
## Anova Table (Type II tests)
##
## Response: B
##           Sum Sq Df F value    Pr(>F)
## W         69.719  1 290.126 < 2.2e-16 ***
## G          9.063  1  37.713 2.002e-08 ***
## Residuals 22.349 93
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The above printouts show that the `Anova' function from`car’ gives the same non-sequential sums of squares for both models (i.e. Sum Sq for W and G are the same in both versions).

e. Fit a model to predict log(G) from log(W) and extract the residuals from this model into an object called GW. Now examine regressing log(B) on log(W) and GW. Show that this model has the same fitted values as those in c (You can look at the maximum of the absolute differences between the fitted value vectors) and show that in this case the order in which you specify log(W) and GW makes no difference to the sequential ANOVA table.

```r
GW = lm(G ~ W, data=log(brain_data))$residuals
part_e_mod = lm(log(brain_data$B) ~ log(brain_data$W) + GW)
```

```
print(max(abs(part_e_mod$fitted.values - model1$fitted.values))) # within tolerance for 0
```

```
## [1] 3.774758e-14
```

The above printout shows that the maximum difference between fitted values of this model and the models from part c is (within some numerical tolerance) zero, so all of the fitted values are the same.

```
print(anova(lm(log(brain_data$B) ~ log(brain_data$W) + GW)))
```

```
## Analysis of Variance Table
##
## Response: log(brain_data$B)
##                   Df Sum Sq Mean Sq  F value    Pr(>F)
## log(brain_data$W)  1 416.40  416.40 1732.785 < 2.2e-16 ***
## GW                 1   9.06    9.06   37.713 2.002e-08 ***
## Residuals         93  22.35    0.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(anova(lm(log(brain_data$B) ~ GW + log(brain_data$W))))
```

```
## Analysis of Variance Table
##
## Response: log(brain_data$B)
##                   Df Sum Sq Mean Sq  F value    Pr(>F)
## GW                 1   9.06    9.06   37.713 2.002e-08 ***
## log(brain_data$W)  1 416.40  416.40 1732.785 < 2.2e-16 ***
## Residuals         93  22.35    0.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

These printouts show that even when using sequential Anova tables the sums of squares are the same for each covariate.

f. We can examine this by the observation that if

$$\log(G) = \gamma_0 + \gamma_1 \log(W) + WG$$

then we can re-write the model from part c as

$$\begin{aligned}\beta_0 + \beta_1 \log(W) + \beta_2 \log(G) &= \beta_0 + \beta_1 \log(W) + \beta_2 \left(\gamma_0 + \gamma_1 \log(W) + WG\right) \\ &= \beta_0 + \beta_2\gamma_0 + (\beta_1 + \gamma_1) \log(W) + \beta_2 WG.\end{aligned}$$

Show that the coefficients in your models satisfy this relationship.

```
## RHS COEFFICIENTS ###
betas = as.numeric(model2$coefficients)

### LHS COEFFICIENTS ###
gammas = as.numeric(lm(G ~ W, data=log(brain_data))$coeff)
lhs_coeffs = c(betas[1] + betas[3]*gammas[1],
               betas[2] + betas[3]*gammas[2],
               betas[3])

## Generate Predictions ##
rhs_preds = (cbind(rep(1, nrow(brain_data)), log(brain_data$W), log(brain_data$G)) %*%
                      betas)
lhs_preds = cbind(rep(1, nrow(brain_data)), log(brain_data$W), GW) %*% lhs_coeffs
```

```
# Compare #
print(max(abs(rhs_preds - lhs_preds))) ## essentially 0
```

```
## [1] 3.552714e-15
```

From the above we see that the maximum absolute difference between the left and right sides is nearly 0, showing that the forms are the same.

Question 2

Here we will explore a mathematical explanation of the statistical behavior we saw in Question 1. For this, we will suppose that we have a model of the form

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$$

where the averages are zero: $\bar{x}_1 = \bar{x}_2 = 0$. In particular, we will write $X = [1_n, \mathbf{x}_1, \mathbf{x}_2]$.

As a hint: most of the questions below can be solved by writing fitted values and residuals in terms of the relevant hat matrices.

a. Suppose in addition that $\mathbf{x}_1^T \mathbf{x}_2 = 0$. Show that $X^T X$ is diagonal.

Using the form of X from above we can see

$$X^T X = [1_n, \mathbf{x}_1, \mathbf{x}_2]^T [1_n, \mathbf{x}_1, \mathbf{x}_2] = \begin{pmatrix} 1_n^T 1_n & 1_n^T \mathbf{x}_1 & 1_n^T \mathbf{x}_2 \\ \mathbf{x}_1^T 1_n & \mathbf{x}_1^T \mathbf{x}_1 & \mathbf{x}_1^T \mathbf{x}_2 \\ \mathbf{x}_2^T 1_n & \mathbf{x}_2^T \mathbf{x}_1 & \mathbf{x}_2^T \mathbf{x}_2 \end{pmatrix}.$$

But note that if $\mathbf{x}_1$ and $\mathbf{x}_2$ are mean 0 we have $\frac{1}{n}\sum_{i=1}^n x_{i,k} = 0$ for $k = 1, 2$, which (by multiplying both sides by n) shows that $\sum_{i=1}^n x_{i,k} = 0$. But this is the same as $1_n^T \mathbf{x}_k = \mathbf{x}_k^T 1_n = 0$ for $k = 1, 2$. So all the entries of this form are 0, and we are given that $\mathbf{x}_1^T \mathbf{x}_2 = \mathbf{x}_2^T \mathbf{x}_1 = 0$ so all entries of this form are 0, leaving just

$$X^T X = \begin{pmatrix} 1_n^T 1_n & 0 & 0 \\ 0 & \mathbf{x}_1^T \mathbf{x}_1 & 0 \\ 0 & 0 & \mathbf{x}_2^T \mathbf{x}_2 \end{pmatrix},$$

which is diagonal. b. Here we could think of fitting β_1 and β_2 three different ways:

1. Performing simple linear regression to predict y_i from x_{i1} and y_i from x_{i2} separately.

Since both $\mathbf{x}_1$ and $\mathbf{x}_2$ are mean 0 I will show the form of the slope of simple linear regression for the general case of a mean 0 covariate vector then plug in $\mathbf{x}_1$ and $\mathbf{x}_2$.

We have as a result from part a that when $X = [1_n, \mathbf{x}]$ we get

$$X^T X = \begin{pmatrix} 1_n^T 1_n & 0 \\ 0 & \mathbf{x}_1^T \mathbf{x}_1 \end{pmatrix} = \begin{pmatrix} n & 0 \\ 0 & \mathbf{x}_1^T \mathbf{x}_1 \end{pmatrix}.$$

Then (by properties of diagonal matrices)

$$[X^T X]^{-1} = \begin{pmatrix} \frac{1}{n} & 0 \\ 0 & \frac{1}{\mathbf{x}_1^T \mathbf{x}_1} \end{pmatrix}.$$

Then, plugging the above into the form $\hat{\beta} = [X^T X]^{-1} X^T \mathbf{y}$ gives

$$\hat{\beta} = \begin{pmatrix} \frac{1}{n} & 0 \\ 0 & \frac{1}{\mathbf{x}^T \mathbf{x}} \end{pmatrix} [1_n, \mathbf{x}]^T [y_1 \cdots y_n]^T = \begin{bmatrix} 1_n^T / n \\ \mathbf{x}^T / \mathbf{x}^T \mathbf{x} \end{bmatrix} [y_1 \cdots y_n]^T = \begin{bmatrix} 1_n^T \mathbf{y} / n \\ \mathbf{x}^T \mathbf{y} / \mathbf{x}^T \mathbf{x} \end{bmatrix}.$$

So $\hat{\beta}_1 = \mathbf{x}_1^T \mathbf{y} / \mathbf{x}_1^T \mathbf{x}_1$ when $\mathbf{x}_1$ is the covariate, and $\hat{\beta}_1 = \mathbf{x}_2^T \mathbf{y} / \mathbf{x}_2^T \mathbf{x}_2$ when the covariate is $\mathbf{x}_2$.

2. Predict y_i from x_{i1} and obtain the residuals r_i from this fit. Then predict r_i from x_{i2}.

For notation I will call β_i the i^{th} coefficient from the first model ($\mathbf{y} \sim \mathbf{x}_1$), and γ_i the i^{th} coefficient from the second model ($\mathbf{r}_1 \sim \mathbf{x}_2$). Additionally, I will denote X_1 as the design matrix with a column of ones and the other column as $\mathbf{x}_1$, likewise for X_2.

First note that the coefficient β_1 (the slope of model 1) will be the same as in part 1, since the models are identical.

Now we define the residuals of this model as

$$r_1 = \mathbf{y} - X_1(X_1^T X_1)^{-1} X_1^T \mathbf{y}.$$

Plugging this in as the response variable to a model with $\mathbf{x}_2$ as the covariate gives:

$$\hat{\gamma} = (X_2^T X_2)^{-1} X_2^T r_1 = (X_2^T X_2)^{-1} X_2^T (\mathbf{y} - X_1(X_1^T X_1)^{-1} X_1^T \mathbf{y})$$
$$\hat{\gamma} = (X_2^T X_2)^{-1} X_2^T \mathbf{y} - (X_2^T X_2)^{-1} X_2^T X_1(X_1^T X_1)^{-1} X_1^T \mathbf{y}).$$

We can recognize the first term in this expression as the form of the predictors when doing simple linear regression with $\mathbf{y}$ as the response and $\mathbf{x}_2$ as the predictor. So if we can show that the second entry in the second term (a 2 × 1 vector) is 0, we will have that $\hat{\gamma}_1$ is what we would get from the regression model in part 1.

Looking at the second term, we can see $\hat{\beta}$ and rewrite as

$$(X_2^T X_2)^{-1} X_2^T X_1(X_1^T X_1)^{-1} X_1^T \mathbf{y}) = (X_2^T X_2)^{-1} X_2^T X_1 \hat{\beta}$$

Now, writing out as matrices (taking results from question 1) we have

$$(X_2^T X_2)^{-1} X_2^T X_1 \hat{\beta} = \begin{pmatrix} \frac{1}{n} & 0 \\ 0 & \frac{1}{\mathbf{x}_2^T \mathbf{x}_2} \end{pmatrix} \begin{pmatrix} n & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix} = \begin{pmatrix} \frac{1}{n} & 0 \\ 0 & \frac{1}{\mathbf{x}_2^T \mathbf{x}_2} \end{pmatrix} \begin{pmatrix} n\hat{\beta}_0 \\ 0 \end{pmatrix} = \begin{pmatrix} \hat{\beta}_0 \\ 0 \end{pmatrix}.$$

It doesn't really matter what the first term here is, just that the second term is 0. Plugging this back into the expression for $\hat{\gamma}$ we can see that $\hat{\gamma}_1$ will be the same slope we would have gotten if we had just done simple linear regression on $\mathbf{x}_2$ as in part 1.

3. Fit a multiple linear regression.

Here we have

$$\hat{\beta} = (X^T X)^{-1} X^T \mathbf{y}$$

where

$$X = [1_n, \mathbf{x}_1, \mathbf{x}_2] \text{ and } X^T X = \begin{pmatrix} n & 0 & 0 \\ 0 & \mathbf{x}_1^T \mathbf{x}_1 & 0 \\ 0 & 0 & \mathbf{x}_2^T \mathbf{x}_2 \end{pmatrix}.$$

Multplying $(X^T X)^{-1}$ (which is the same as above just with reciprocals along the diagonal) by X^T gives

$$(X^T X)^{-1} X^T = \begin{pmatrix} 1/n & 0 & 0 \\ 0 & \frac{1}{\mathbf{x}_1^T \mathbf{x}_1} & 0 \\ 0 & 0 & \frac{1}{\mathbf{x}_2^T \mathbf{x}_2} \end{pmatrix} \begin{pmatrix} 1_n^T \\ \mathbf{x}_1^T \\ \mathbf{x}_2^T \end{pmatrix} = \begin{pmatrix} 1_n^T/n \\ \frac{\mathbf{x}_1^T}{\mathbf{x}_1^T \mathbf{x}_1} \\ \frac{\mathbf{x}_2^T}{\mathbf{x}_2^T \mathbf{x}_2} \end{pmatrix}$$

And plugging this into the form for $\hat{\beta}$ gives

$$\hat{\beta} = \begin{pmatrix} 1_n^T/n \\ \frac{\mathbf{x}_1^T}{\mathbf{x}_1^T \mathbf{x}_1} \\ \frac{\mathbf{x}_2^T}{\mathbf{x}_2^T \mathbf{x}_2} \end{pmatrix} \mathbf{y} = \begin{pmatrix} 1_n^T \mathbf{y}/n \\ \frac{\mathbf{x}_1^T \mathbf{y}}{\mathbf{x}_1^T \mathbf{x}_1} \\ \frac{\mathbf{x}_2^T \mathbf{y}}{\mathbf{x}_2^T \mathbf{x}_2} \end{pmatrix}$$

Which is exactly what we got in the first case when we just did simple linear regression on each predictor (and was shown to be the same as in the second case).

Show that all three cases give you the same coefficients β_1 and β_2.

Done above.

c. In a general regression $y_i = \mathbf{x}_i^T \beta + \epsilon_i$, show that the residuals of a linear regression are orthogonal to the fitted values.

First recall that the hat matrix H is both symmetric and idempotent. Then we can write the inner product of the fitted values and residuals out and just perform substitutions and simplifications to get

$$\hat{\mathbf{y}}^T \mathbf{r} = \hat{\mathbf{y}}^T (\mathbf{y} - \hat{\mathbf{y}}) = (H\mathbf{y})^T (\mathbf{y} - H\mathbf{y}) = \mathbf{y}^T H (\mathbf{y} - H\mathbf{y}) = \mathbf{y}^T H \mathbf{y} - \mathbf{y}^T H H \mathbf{y} = \mathbf{y}^T \hat{\mathbf{y}} - \mathbf{y}^T \hat{\mathbf{y}} = 0.$$

So the two vectors are orthogonal.

d. Here we will try to resolve the ANOVA decomposition. Let z_i be the residual after regressing x_{i2} on x_{i1}. That is $x_{i2} = \alpha_0 + \alpha_1 x_{i1} + z_i$.
Show that the sequential ANOVA decomposition for the model $y_i = \gamma_0 + \gamma_1 x_{i1} + \gamma_2 z_i + \epsilon_i$ does not change with the order in which x_{i1} and z_i are entered.

If we let H_1 be the hat matrix for regressing on $\mathbf{x}_1$ then $\mathbf{z} = (I - H_1)\mathbf{x}_2$ and hence $\mathbf{z}^T \mathbf{1}_n = \mathbf{z}^T \mathbf{x}_1 = 0$. Thus, taking $X = [\mathbf{1}_n, \mathbf{x}_1, \mathbf{z}]$ we have

$$X^T X = \begin{bmatrix} \mathbf{1}_n^T \mathbf{1}_n & 0 & 0 \\ 0 & \mathbf{x}_1^T \mathbf{x}_1 & 0 \\ 0 & 0 & \mathbf{z}^T \mathbf{z} \end{bmatrix}$$

Which lets us write the full hat matrix as

$$\begin{aligned} H_2 &= X(X^T X)^{-1} X^T \\ &= [1, \mathbf{x}_1, \mathbf{z}] \begin{bmatrix} \mathbf{1}/n & 0 & 0 \\ 0 & 1/\mathbf{x}_1^T \mathbf{x}_1 & 0 \\ 0 & 0 & 1/\mathbf{z}^T \mathbf{z} \end{bmatrix} \begin{bmatrix} \mathbf{1}^T \\ \mathbf{x}_1^T \\ \mathbf{z}^T \end{bmatrix} \\ &= [1, \mathbf{x}_1, \mathbf{z}] \begin{bmatrix} \mathbf{1}^T/n \\ \mathbf{x}_1^T / \mathbf{x}_1^T \mathbf{x}_1 \\ \mathbf{z}^T / \mathbf{z}^T \mathbf{z} \end{bmatrix} \\ &= \frac{1}{n} \mathbf{1}\mathbf{1}^T + \frac{1}{\mathbf{x}_1^T \mathbf{x}_1} \mathbf{x}_1 \mathbf{x}_1^T + \frac{1}{\mathbf{z}^T \mathbf{z}} \mathbf{z}\mathbf{z}^T \end{aligned}$$

and we observe that similar calculations show that if $X_1 = [1, \mathbf{x}_1]$ then

$$H_1 = \frac{1}{n} \mathbf{1}\mathbf{1}^T + \frac{1}{\mathbf{x}_1^T \mathbf{x}_1} \mathbf{x}_1 \mathbf{x}_1^T$$

and if $\tilde{X}_1 = [1, \mathbf{z}]$ then the corresponding hat matrix is

$$\tilde{H}_1 = \frac{1}{n} \mathbf{1}\mathbf{1}^T + \frac{1}{\mathbf{z}^T \mathbf{z}} \mathbf{z}\mathbf{z}^T$$

so that if we take $\mathbf{x}_1$ before $\mathbf{z}$ we have the sums of squares determined by

$$H_1 - H_0 = \frac{1}{\mathbf{x}_1^T \mathbf{x}_1} \mathbf{x}_1 \mathbf{x}_1^T \text{ and } H_2 - H_1 = \frac{1}{\mathbf{z}^T \mathbf{z}} \mathbf{z}\mathbf{z}^T$$

and the other way, we have

$$\tilde{H}_1 - H_0 = \frac{1}{\mathbf{z}^T \mathbf{z}} \mathbf{z}\mathbf{z}^T \text{ and } H_2 - \tilde{H}_1 = \frac{1}{\mathbf{x}_1^T \mathbf{x}_1} \mathbf{x}_1 \mathbf{x}_1^T$$

resulting in the same answers.

e. What is the relationship between the γ_j above and the β_j in the model at the start of this question?

We want to compare the models

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i \text{ and } y_i = \gamma_0 + \gamma_1 x_{i1} + \gamma_2 z_i + \epsilon_i.$$

First, we know that the ϵ_i's will be the same for both models. Geometrically we interpret the fitted values of the model as the orthogonal projection onto the span of the covariates. Since we can write $\mathbf{z}$ as a linear combination of $\mathbf{x}_1$ and $\mathbf{x}_2$ (with non-zero coefficients) it must be the case that the span of $\mathbf{z}$ and $\mathbf{x}_1$ is the same as the span of $\mathbf{x}_1$ and $\mathbf{x}_2$. So the orthogonal projections of $\mathbf{y}$ will be the same for either space. Thus the errors ϵ_i will be the same.

Thus we can set the models equal to one another and cancel the ϵ_i's, giving

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} = \gamma_0 + \gamma_1 x_{i1} + \gamma_2 z_i$$

Now we just need to substitute our form for z_i back in. And note that since $\mathbf{x}_1$ and $\mathbf{x}_2$ are both mean 0, we will have that $\alpha_0 = 0$ (this is just a special case of what was shown in part a1).

So we have $z_i = x_{i2} - \alpha_1 x_{i1}$. Plugging into the above yields

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} = \gamma_0 + \gamma_1 x_{i1} + \gamma_2 (x_{i2} - \alpha_1 x_{i1})$$

Now we can just match coefficients to get

$$\gamma_0 = \beta_0 \text{ and } \gamma_1 - \gamma_2 \alpha_1 = \beta_1 \text{ and } \gamma_2 = \beta_2.$$

f. Show further that the ANOVA sum of squares for the model in part d is equivalent to the sequential ANOVA for x_{i1} and x_{i2} where the sums of squares for x_{i1} is calculated first.

For this we recall that X and XA give the same hat matrix so long as A is invertible. Here we have that

$$[\mathbf{1}, \mathbf{x}_1, \mathbf{x}_2] \begin{bmatrix} 1 & 0 & -\alpha_0 \\ 0 & 1 & -\alpha_1 \\ 0 & 0 & 1 \end{bmatrix} = [\mathbf{1}, \mathbf{x}_1, \mathbf{z}]$$

if α_0 and α_1 are chosen by least-squares for predicting $\mathbf{x}_2$ from $\mathbf{x}_1$. Hence $H_2 - H_1$ is the same for both cases. Note that

$$A^{-1} = \begin{bmatrix} 1 & 0 & \alpha_0 \\ 0 & 1 & \alpha_1 \\ 0 & 0 & 1 \end{bmatrix}$$

Question 3

a. Let A and B be $p \times n$ and $q \times n$ matrices with columns $(\mathbf{a}_1, \ldots, \mathbf{a}_n)$ and $(\mathbf{b}_1, \ldots, \mathbf{b}_n)$ respectively.

Show that

$$AB^T = \sum_{k=1}^{n} \mathbf{a}_k \mathbf{b}_k^T$$

by expressing the elements $(AB^T)_{ij}$ in terms of the individual entries a_{ij} and b_{ij}.

First look at the i, j entry as we would calculate it from the left hand side. We know that, from matrix multipliction, this corresponds to the vector product of the i^{th} row of A with the j^{th} column of B^T (which is just the j^{th} row of B).

So

$$AB^T_{ij} = A_{i:} B^T_{:j} = A_{i:} B_{j:} = \sum_{k=1}^{n} A_{ik} B_{jk}$$

or, written in terms of the column vectors,

$$AB^T_{ij} = \sum_{k=1}^{n} (\mathbf{a}_k)_i (\mathbf{b}_k)_j$$

where $(\mathbf{a}_k)_i$ is the i^{th} entry of the vector $\mathbf{a}_k$.

Now calculate the i, j entry from the right hand side. The i, j entry of the outer product $\mathbf{a}_k \mathbf{b}_k^T$ is the i^{th} element of $\mathbf{a}_k$ multiplied by the j^{th} element of $\mathbf{b}_k$ (i.e. $(\mathbf{a}_k)_i(\mathbf{b}_k)_j$). Taking the sum over k of the i, j entry of these outer products will give us the i, j entry of the resulting matrix, therefore

$$\left[\sum_{k=1}^{n} \mathbf{a}_k \mathbf{b}_k^T\right]_{ij} = \sum_{k=1}^{n} (\mathbf{a}_k)_i (\mathbf{b}_k)_j$$

Which is the same expression found above, giving that the equivalence relation above holds.

b. Show that $tr(AB) = tr(BA)$ by writting down each of these in terms of the elements of A and B.

Using the form found above we can show that both sides are equivalent. First note that the trace of AB, $tr(AB)$, is equal to the sum from 1 to n of all the diagonal entries,

$$tr(AB) = \sum_{i=1}^{n} AB_{ii}.$$

Then substituting from part a and moving things around we have

$$tr(AB) = \sum_{i=1}^{n}\sum_{k=1}^{n} A_{ik}B_{ki} = \sum_{k=1}^{n}\sum_{i=1}^{n} A_{ik}B_{ki} = \sum_{k=1}^{n}\sum_{i=1}^{n} B_{ki}A_{ik}.$$

But if we swap the i's and the k's (just labeling so it doesn't change anything) we have

$$\sum_{k=1}^{n}\sum_{i=1}^{n} B_{ki}A_{ik} = \sum_{i=1}^{n}\sum_{k=1}^{n} B_{ik}A_{ki}$$

which is the sum over i down the diagonal of the matrix product BA, which is $tr(BA)$.

Giving $tr(AB) = tr(BA)$.

c. We claimed in class that for a square symmetric matrix A with eigenvector/eigenvalue pairs $(\mathbf{e}_i, d_i)$ we can write $A = EDE^T$, where D is a diagonal matrix with diagonal entries d_i and E is a matrix with columns given by the the $\mathbf{e}_i$. Show that this is true by

i) Showing that $AE = ED$

ii) Observing that $EE^T = I$, produce $A = EDE^T$.

iii) We can think about the product AE column by column. The first column of the matrix product AE is the first column of E ($E_{:1}$) multiplied by the matrix A. Since this is an eigenvector, this will just be $\lambda_1 E_{:1}$, where λ_1 is the corresponding eigenvalue. Following this logic, AE will just be E where each of the columns is multiplied by its corresponding eigenvalue.

Recalling the result from the first question in homework 1, we can write this as E multiplied on the right by a diagonal matrix D where the entries along the diagonal are eigenvalues.

Giving,

$$AE = ED.$$

ii) Note that for a real symmetric matrix we have orthogonal eigenvectors (and assumed they are scaled correctly they are orthonormal). An easy sketch of the proof is as follows:

assume $\mathbf{u}$ and $\mathbf{v}$ are eigenvectors of a real symmetric matrix A corresponding to distinct eigenvalues, μ and λ respectively.

Then consider the inner product $\mathbf{u}^T\mathbf{v}$. We can multiply by μ to get

$$\mu\mathbf{u}^T\mathbf{v} = (A\mathbf{u})^T\mathbf{v} = \mathbf{u}^T A^T\mathbf{v} = \mathbf{u}^T A\mathbf{v} = \lambda\mathbf{u}^T\mathbf{v}$$

Then subtracting the leftmost and rightmost sides gives

$$\mu\mathbf{u}^T\mathbf{v} - \lambda\mathbf{u}^T\mathbf{v} = (\mu\lambda)\mathbf{u}^T\mathbf{v} = 0.$$

Since $\mu \neq \lambda$ by assumption it must be the case that $\mathbf{u}^T\mathbf{v} = 0$. Thus if we take E to be a matrix of normalized eigenvectors of A, we get $EE^T = I$.

So taking the expression from i) and right multiplying by E^T we get

$$AEE^T = EDE^T \text{ giving } A = EDE^T.$$

d. Using the above results, show that we can write

$$\sum_{i=1}^{n} d_i\mathbf{e}_i\mathbf{e}_i^T = EDE^T$$

We know that ED is the same as E with each column multiplied by the corresponding eigenvalue. So let $B = ED$ where $B = (d_1\mathbf{e}_1, \ldots, d_n\mathbf{e}_n)$. Then $EDE^T = BE^T$. And using the result from a) we know we can write

$$EDE^T = BE^T = \sum_{i=1}^{n} d_i\mathbf{e}_i\mathbf{e}_i^T.$$

e. Hence show that we can express $\text{tr}(A) = \sum d_i$.

By substitution we have $tr(A) = tr(EDE^T)$, now again let $B = ED$ so $tr(A) = tr(BE^T)$, but from b) we know this is the same as $tr(E^TB) = tr(E^TED) = tr(ID) = tr(D)$ which is just the sum of the d_i's. So

$$tr(A) = \sum_{i=1}^{n} d_i.$$