BTRY 4030 - Fall 2018 - Homework 5 Q1

Put Your Name and NetID Here

Due Tuesday, December 4, 2018

You may either respond to the questions below by editing the hw5_2018_q1.Rmd to include your answers and compiling it into a PDF document, or by handwriting your answers and scanning these in.

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

Here we will add one more deletion diagnostic to our arsenal. When comparing two possible models, we often want to ask "Does one predict *future data* better than the other?" One way to do this is to divide your data into two collections of observations $(X_1, \mathbf{y}_1)$ and $(X_2, \mathbf{y}_2)$, say. We use $(X_1, \mathbf{y}_1)$ to obtain a linear regression model, with parameters $\hat{\beta}$ and look at the *prediction error* $(\mathbf{y}_2 - X_2\hat{\beta})^T(\mathbf{y}_2 - X_2\hat{\beta})$.

This is a bit wasteful – you could use $(X_2, \mathbf{y}_2)$ to improve your estimate of $\hat{\beta}$. However, we can assess how well this *type* of model does (for these data) as follows:

For each observation i

i. Remove $(\mathbf{x}_i, y_i)$ from the data and obtain $\hat{\beta}_{(i)}$ from the remaining $n-1$ data points.

ii. Use this to make a prediction $\hat{y}_{(i)i} = \mathbf{x}_i^T\hat{\beta}_{(i)}$.

Return the *cross validation* error $CV = \sum_{i=1}^{n}(y_i - \hat{y}_{(i)i})^2$

This can be used to compare a models that use different covariates, for example; particularly when the models are not nested. We will see an example of this in Question 2.

Here, we will find a way to calculate CV without having to manually go through removing observations one by one.

a. We will start by considering a separate test-set. As in the midterm, imagine that we have $X_2 = X_1$, but that the errors that produce $\mathbf{y}_2$ are independent of those that produce $\mathbf{y}_1$. We estimate $\hat{\beta}|$ using $(X_1, \mathbf{y}_1)$: $\hat{\beta} = (X_1^TX_1)^{-1}X_1^T\mathbf{y}$. Show that the in-sample average squared error, $(\mathbf{y}_1 - X_1\hat{\beta})^T(\mathbf{y}_1 - X_1\hat{\beta})/n$, is biased downwards as an estimate of σ, but the test-set average squared error, $(\mathbf{y}_2 - X_2\hat{\beta})^T(\mathbf{y}_2 - X_2\hat{\beta})/n$, is biassed upwards. (You may find the midterm solutions helpful.)

We know that

$$\frac{SSE}{n-p-1} = \frac{(y - X\hat{\beta})^T(y - X\hat{\beta})}{n-p-1}$$

is an unbiased estimate for σ^2, so when $\hat{\beta}$ is estimated with just the X_1's we have $(\mathbf{y}_1 - X_1\hat{\beta})^T(\mathbf{y}_1 X_1\hat{\beta}) = SSE$ so

$$\frac{(\mathbf{y}_1 - X_1\hat{\beta})^T(\mathbf{y}_1 - X_1\hat{\beta})}{n} = \frac{SSE}{n} < \frac{SSE}{n-p-1}$$

Giving that the expression is biased downwards.

Now looking at $X_2\hat{\beta}$ where $\hat{\beta}$ is predicted using X_1 we have

$$X_2\hat{\beta} \sim N(X_2\beta, \sigma^2X_2(X_1^TX_1)^{-1}X_2^T) = N(X_2\beta, \sigma^2H)$$

where H is the hat matrix formed with just the X_1 covariates (this follows since $X_1 = X_2$). So, looking at the i^{th} residual on the testing data we have,

$$E[\mathbf{y}_{2i} - \mathbf{x}_{2i}\hat{\beta}] = \mathbf{x}_{2i}\beta - \mathbf{x}_{2i}\beta = 0$$

and

$$Var(\mathbf{y}_{2i} - \mathbf{x}_{2i}\hat{\beta}) = Var(\mathbf{y}_{2i}) + Var(\mathbf{x}_{2i}\hat{\beta}) = \sigma^2(1 + H_{ii}).$$

Looking then at $E[(\mathbf{y}_2 - X_2\hat{\beta})^T(\mathbf{y}_2 - X_2\hat{\beta})]/n = E[\sum_{j=1}^n(\mathbf{y}_{2j} - \mathbf{x}_{2j}\hat{\beta})^2]/n$ we get

$$E[\sum_{j=1}^n(\mathbf{y}_{2j} - \mathbf{x}_{2j}\hat{\beta})^2]/n = \frac{1}{n}\sum_{j=1}^n \sigma^2(1 + H_{ii}) = \sigma^2\frac{n + Tr(H)}{n} > \sigma^2$$

so the estimate is biased upwards.

b. Suppose that $\beta_p = 0$, that is the final column of X_1 has no impact on prediction. Show that the expected *test set* error is smaller if we remove the final column from each of X_1 and X_2 than if we don't. (This makes using a test set a reasonable means of choosing what covariates to include.)

We can look at the expected test set error, and if the final columns of the design matrices have no impact on prediction then

$$E[\sum_{j=1}^n(y_{2j} - x_{2j}\hat{\beta})^2] = \sigma^2(n + Tr(H))$$

by the above result, and if we remove one of the covariates the trace of the hat matrix shrinks by one, giving a smaller expected error in the test set.

c. Now we will turn to cross validation. Using the identity

$$\hat{\beta}_{(i)} = \hat{\beta} - \frac{1}{1 - h_{ii}}(X^TX)^{-1}\mathbf{x}_i\hat{e}_i$$

from class, to obtain an expression for the *out of sample* prediction $\mathbf{x}_i^T\hat{\beta}_{(i)}$ in terms of $\mathbf{x}_i$, y_i, $\hat{\beta}$ and h_{ii} only.

Using the identity above

$$\begin{aligned}\mathbf{x}_i^T\hat{\beta}_{(i)} &= \mathbf{x}_i^T\hat{\beta} - \frac{\mathbf{x}_i^T(X^TX)^{-1}\mathbf{x}_i}{1 - h_{ii}}\hat{e}_i \\ &= \mathbf{x}_i^T\hat{\beta} - \frac{h_{ii}}{1 - h_{ii}}\hat{e}_i\end{aligned}$$

Since $H = X(X^TX)^{-1}X^T$ and therefore $\mathbf{x}_i^T(X^TX)^{-1}\mathbf{x}_i = h_{ii}$ (noting that $\mathbf{x}_i$ is the ith row of X, but taken as a colulmn vector.)

d. Hence obtain an expression for the prediction error $y_i - \mathbf{x}_i^T\hat{\beta}_{(i)}$ using only y_i, $\hat{y}_i$ and h_{ii}. You may want to check this empirically using the first few entries of the data used in Question 2.

$$y_i - \mathbf{x}_i^T\hat{\beta}_{(i)} = y_i - \mathbf{x}_i^T\hat{\beta} + \frac{h_{ii}}{1 - h_{ii}}\hat{e}_i = \hat{e}_i + \frac{h_{ii}}{1 - h_{ii}}\hat{e}_i = \frac{\hat{e}_i}{1 - h_{ii}}$$

e. Show that the over-all CV score can be calculated from

$$\sum_{i=1}^n \frac{\hat{e}_i^2}{(1 - h_{ii})^2}$$

that is, *without* deleting observations, and only requiring the leverages h_{ii}.

We are looking for

$$\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T \hat{\beta}_{(i)})^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{(1 - h_{ii})^2}$$

Question 2

Here we will illustrate the results from Question 1 with a real world data set. We will use the study of mortality in 55 US cities as it is influenced by pollutants NOX (nitrous oxide) and SO2 (sulfur dioxide), while controlling weather (PRECIP) and sociological variables (EDUC and NONWHITE) that appeared on homework 4.

You can find the data in `airpollution.csv` on CMS.

a. Delete each of the first four observation in turn, fit a model with the remaining observations (ie, each model should be fit based on $n - 1$ observations) and use this to predict MORT in the left-out sample. Verfiy that your answer in Question 1d returns the same error.

```
dat = read.csv("./airpollution.csv", header=TRUE)
dat = dat[ , -1]
pred_mort = rep(NA, 4)
errs = rep(NA, 4)
computed_errors = rep(NA, 4)
full_mod = lm(MORT ~ ., data=dat)
des_mat = model.matrix(full_mod)
hat_mat = (des_mat %*% solve(t(des_mat) %*% des_mat) %*% t(des_mat))
for (ii in 1:4) {
    mod = lm(MORT ~ ., data=dat[-ii, ])
    pred_mort[ii] = predict(mod, dat[ii, ])
    errs[ii] = dat[ii, 1] - pred_mort[ii]

    ## compute the errors from 1d ##
    hii = hat_mat[ii, ii]
    computed_errors[ii] = (full_mod$residuals[ii])/(1 - hii)
    # computed_errors[ii] = dat$MORT[ii] - full_mod$fitted.values[ii] - 1/(1 - hii)
}

print(all(round(computed_errors, 5) == round(errs, 5))) # all in agreement
```

```
## [1] TRUE
```

b. Using your identity in Question 1e, compute the cross-validation score for a model using all covariates.

```
cv = 0
for (ii in 1:nrow(dat)) {
   cv = cv + full_mod$residuals[ii]^2/(1 - hat_mat[ii, ii])^2
}
print(paste("cross-validation =", cv))
```

```
## [1] "cross-validation = 77110.5491469181"
```

c. Calculate the cross-validation score in the sequence of models obtained by starting from the intercept and adding each column in the order given in the data (so every model should have one more covariate than the previous one). Which model has the lowest score?

```
cross_val <- function(mod){
    des_mat = model.matrix(mod)
```

```
    hat_mat = des_mat %*% solve(t(des_mat) %*% des_mat) %*% t(des_mat)
    cv = sum( (mod$residuals/(1-diag(hat_mat)))^2 )
    return(cv)
}

CV_scores = rep(0, 6)
names(CV_scores) = c("intercept", "precip", "educ", "nonwhite", "nox", "so2")

## just intercept ##
mod = lm(MORT ~ 1, data=dat)
CV_scores[1] = cross_val(mod)
## and precip ##
mod = lm(MORT ~ PRECIP, data=dat)
CV_scores[2] = cross_val(mod)
## and educ ##
mod = lm(MORT ~ PRECIP + EDUC, data=dat)
CV_scores[3] = cross_val(mod)
## and nonwhite ##
mod = lm(MORT ~ PRECIP + EDUC + NONWHITE, data=dat)
CV_scores[4] = cross_val(mod)
## and NOX ##
mod = lm(MORT ~ PRECIP + EDUC + NONWHITE + NOX, data=dat)
CV_scores[5] = cross_val(mod)
## and SO2 ##
mod = lm(MORT ~ PRECIP + EDUC + NONWHITE + NOX + SO2, data=dat)
CV_scores[6] = cross_val(mod)
CV_scores
```

```
## intercept    precip      educ  nonwhite       nox       so2
## 164241.17 160742.62 148834.42  99514.27  75063.52  77110.55
```

Looking at the `CV_scores` vector we see that the model with the lowest score removes SO2 but keeps all the other covariates in the model.

d. What happens if you add them in reverse order? Plot both sequences of scores versus the number of covariates in the model.

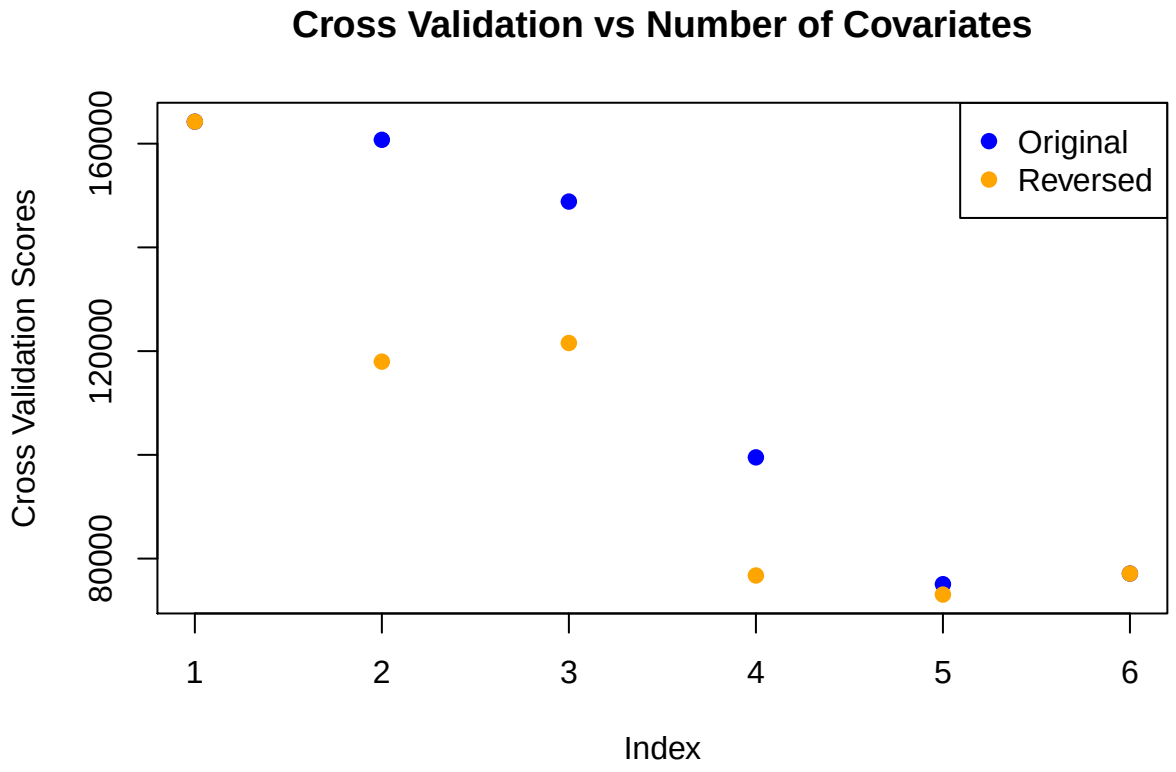
```
rev_scores = rep(0, 6)
names(rev_scores) = rev(c("intercept", "precip", "educ", "nonwhite", "nox", "so2"))

## just intercept ##
mod = lm(MORT ~ 1, data=dat)
rev_scores[1] = cross_val(mod)
## Just SO2 ##
mod = lm(MORT ~ SO2, data=dat)
rev_scores[2] = cross_val(mod)
## and NOX ##
mod = lm(MORT ~ SO2 + NOX , data=dat)
rev_scores[3] = cross_val(mod)
## and NONWHITE ##
mod = lm(MORT ~ SO2 + NOX + NONWHITE , data=dat)
rev_scores[4] = cross_val(mod)
## and EDUC ##
mod = lm(MORT ~ SO2 + NOX + NONWHITE + EDUC , data=dat)
rev_scores[5] = cross_val(mod)
```

```
## and PRECIP ##
mod = lm(MORT ~ SO2 + NOX + NONWHITE + EDUC + PRECIP  , data=dat)
rev_scores[6] = cross_val(mod)
rev_scores
```

```
##       so2       nox  nonwhite      educ    precip intercept
## 164241.17 117976.24 121570.21  76742.00  73065.13  77110.55
```

Plotting:

```
plot(CV_scores, pch = 19, col="blue",
     ylim=range(c(CV_scores, rev_scores)),
     ylab = "Cross Validation Scores",
     main = "Cross Validation vs Number of Covariates")
points(rev_scores, pch=19, col="orange")
legend("topright", legend=c("Original", "Reversed"), pch=19, col=c("blue", "orange"))
```



Here we drop the last covariate as well, but this time it's PRECIP.

e. An alternative (fairly classical) means of selecting models in linear regression in Mallows C_p score. This can be expressed as

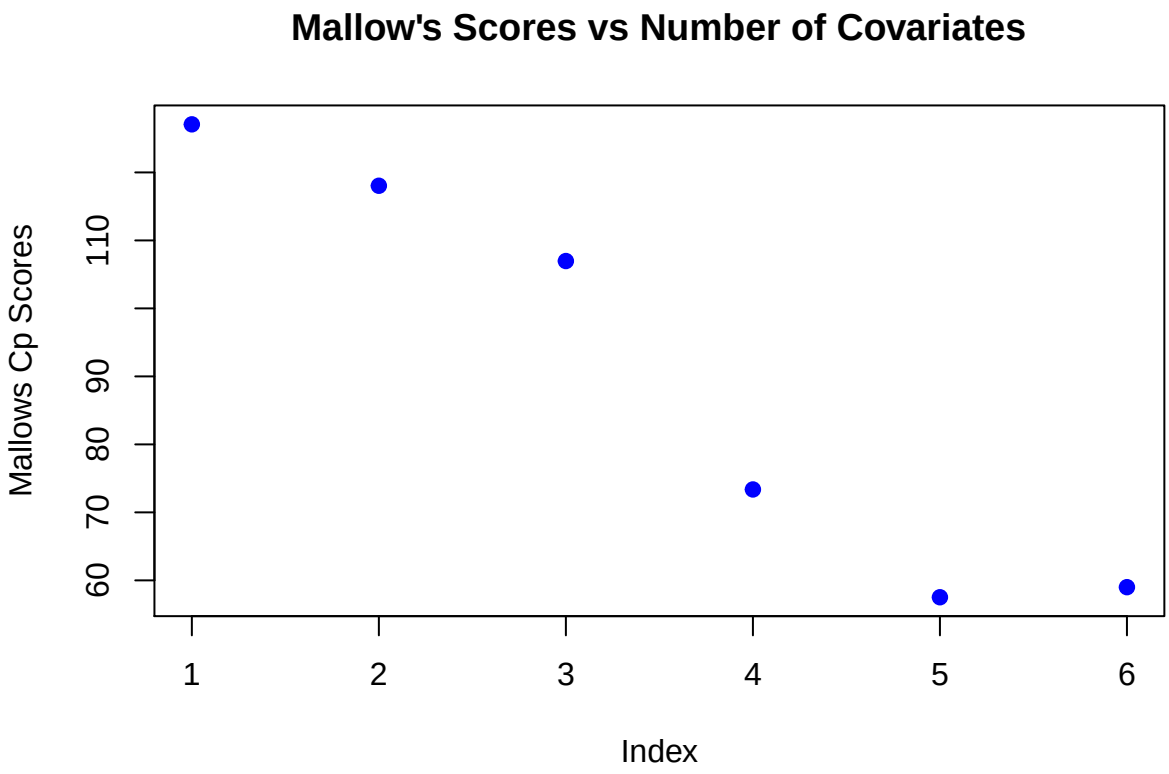
$$C_j = \frac{\mathbf{y}^T(I - H_j)\mathbf{y}}{\mathbf{y}^T(I - H)\mathbf{y}/(n - p - 1)} - 2j$$

often also written as $SSE_j/\hat{\sigma^2} - 2j$, where SSE_j is the SSE for a model with j covariates, and $\hat{\sigma^2}$ is calculated from a model with all covariates.
Obtain C_j for each of your models in part c, how does this compare with cross validation?

```r
# just a helper function #
mallows <- function(model, sig_hat, j){
    sse = sum(model$residuals^2)
    return(sse/sig_hat^2 + 2*j)
}

# build full model to get sig hat #
sig_hat = summary(lm(MORT ~ ., data=dat))$sigma
cp_scores = rep(0, 6)
names(cp_scores) = c("intercept", "precip", "educ", "nonwhite", "nox", "so2")

## just intercept ##
mod = lm(MORT ~ 1, data=dat)
cp_scores[1] = mallows(mod, sig_hat, 0)
## and precip ##
mod = lm(MORT ~ PRECIP, data=dat)
cp_scores[2] = mallows(mod, sig_hat, 1)
## and educ ##
mod = lm(MORT ~ PRECIP + EDUC, data=dat)
cp_scores[3] = mallows(mod, sig_hat, 2)
## and nonwhite ##
mod = lm(MORT ~ PRECIP + EDUC + NONWHITE, data=dat)
cp_scores[4] = mallows(mod, sig_hat, 3)
## and NOX ##
mod = lm(MORT ~ PRECIP + EDUC + NONWHITE + NOX, data=dat)
cp_scores[5] = mallows(mod, sig_hat, 4)
## and SO2 ##
mod = lm(MORT ~ PRECIP + EDUC + NONWHITE + NOX + SO2, data=dat)
cp_scores[6] = mallows(mod, sig_hat, 5)
cp_scores
```

```
## intercept    precip      educ  nonwhite       nox       so2
##  127.0696  118.0385  106.9658   73.3712   57.5191   59.0000
```

```r
# plotting #
plot(cp_scores, pch=19, col="blue",
     ylab="Mallows Cp Scores",
     main="Mallow's Scores vs Number of Covariates")
```



From this plot we see that the scales are very different for Mallow's C_p scores and cross-validation scores, but the trends are similar. Mallow's C_p score indicates we should use all covariates except $SO2$.

bonus: When there is no natural ordering of the covariates, one way to create one is to first choose the covariate that produces the smallest SSE among all models in 1 covariate. Then, keeping that in the model look for the best covariate to add to it. Continue this process until all covariates are in the model. If you do this, what ordering do you get? What is your optimal model?

```
## evaluate all single predictors ##
for (cov_ind in c(2:6)) {
    mod = lm(MORT ~ dat[ , cov_ind], data=dat)
    print(paste("SSE for ind ",cov_ind, " is ", sum(mod$resid^2)))
}
```

```
## [1] "SSE for ind  2  is  144578.838830966"
## [1] "SSE for ind  3  is  133281.752919119"
## [1] "SSE for ind  4  is  100726.854026237"
## [1] "SSE for ind  5  is  108701.0921071"
## [1] "SSE for ind  6  is  109955.389945021"
```

get that NONWHITE is the best single covariate, now pick second covariate

```
for (cov_ind in c(2, 3, 5, 6)) {
    mod = lm(MORT ~ NONWHITE +  dat[ , cov_ind], data=dat)
    print(paste("SSE for ind ",cov_ind, " is ", sum(mod$resid^2)))
}
```

```
## [1] "SSE for ind  2  is  100071.620880787"
## [1] "SSE for ind  3  is  84105.8537126839"
```

```
## [1] "SSE for ind  5  is  74113.8874391216"
## [1] "SSE for ind  6  is  69527.9921565895"
```

So the next best predictor is SO2, now pick the next covariate

```
for (cov_ind in c(2, 3, 5)) {
    mod = lm(MORT ~ NONWHITE + SO2 + dat[ , cov_ind], data=dat)
    print(paste("SSE for ind ",cov_ind, " is ", sum(mod$resid^2)))
}
```

```
## [1] "SSE for ind  2  is  65724.0740926635"
## [1] "SSE for ind  3  is  62703.8057435706"
## [1] "SSE for ind  5  is  69443.4999424066"
```

Then we add EDUC and continue

```
for (cov_ind in c(2, 5)) {
    mod = lm(MORT ~ NONWHITE + SO2 + EDUC + dat[ , cov_ind], data=dat)
    print(paste("SSE for ind ",cov_ind, " is ", sum(mod$resid^2)))
}
```

```
## [1] "SSE for ind  2  is  61587.5967247313"
## [1] "SSE for ind  5  is  61941.8489504311"
```

Then PRECIP, leaving just NOX, and then computing the cross validation error for the full model gives

```
mod = lm(MORT ~ NONWHITE + SO2 + EDUC + PRECIP + NOX, data=dat)
    print(paste("SSE for ind ",cov_ind, " is ", sum(mod$resid^2)))
```

```
## [1] "SSE for ind  5  is  61051.82470919"
```

The ordering we get is then NONWHITE, SO2, EDUC, PRECIP, then NOX.

bonus: Simulate data from the model that you get before SO2 and NOX are entered (that is, fit a model with just PRECIP, EDUC and NONWHITE and simulate data with the estimated coefficients and residual variance). Carry out the model-selection step in part c for each of 100 simulations. How frequently does cross validation choose the right model?

```
## what do we use as predictor variables? ##
mod = lm(MORT ~ PRECIP + EDUC + NONWHITE, data=dat)
beta = mod$coefficients
des_mat = model.matrix(mod)
sig = summary(mod)$sigma
n_obs = nrow(dat)

correct_pred_counter = 0
for (simulation in c(1:100)) {
    CV_scores = rep(NA, 6)
    sim_dat = des_mat %*% beta + rnorm(n_obs, 0, sd=sig)

    ## just intercept ##
    mod = lm(sim_dat ~ 1, data=dat)
    CV_scores[1] = cross_val(mod)
    ## and precip ##
    mod = lm(sim_dat ~ PRECIP, data=dat)
    CV_scores[2] = cross_val(mod)
    ## and educ ##
    mod = lm(sim_dat ~ PRECIP + EDUC, data=dat)
    CV_scores[3] = cross_val(mod)
```

```
    ## and nonwhite ##
    mod = lm(sim_dat ~ PRECIP + EDUC + NONWHITE, data=dat)
    CV_scores[4] = cross_val(mod)
    ## and NOX ##
    mod = lm(sim_dat ~ PRECIP + EDUC + NONWHITE + NOX, data=dat)
    CV_scores[5] = cross_val(mod)
    ## and SO2 ##
    mod = lm(sim_dat ~ PRECIP + EDUC + NONWHITE + NOX + SO2, data=dat)
    CV_scores[6] = cross_val(mod)

    if(which.min(CV_scores) == 4){
        correct_pred_counter = correct_pred_counter + 1
    }
}
print(paste("Correctly selected model ", correct_pred_counter, "% of the time over 100 trials"))
```

```
## [1] "Correctly selected model  79 % of the time over 100 trials"
```

Only get correct predictions ~9% of the time (sometimes as low as 4%, sometimes as high as 11%).

Question 3

This question will consider a random effect model for data longitudinal data. In Question 4, we will examine the effect of different diets on the growth rate of rats. Here we will look at a simplfied version of this where we simply measure the size of rats (given by the width between their temples) as the grow over 5 weeks. There will be n rats in total.

A simple model for a given rat is

$$\mathbf{y}_i = \beta_0 + b_{i0} + \beta_1 \mathbf{t} + \epsilon_i$$

where $\mathbf{t}$ is coded as $(-2, -1, 0, 1, 2)^T$ so it is orthogonal to the intercept and $\mathbf{y}_i$ is the set of five measurements for the ith rat.

We will treat $\epsilon_i \sim N(0, \sigma^2 I_5)$ and we will also let each rat start with it's own size offset b_{i0} and model differences between rats by saying that $b_{i0} \sim N(0, \tau^2)$ independent of all other quantities.

Note that this means that β_0 is the size of the average rate at 3 weeks (corresponding to $t_3 = 0$).

We will also use $X = [\mathbf{1}_5, \mathbf{t}]$ as the "design matrix" for a single rat.

a. We'll first start by considering an estimate for each rat separately

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

What is the mean vector and variance matrix of $\hat{\beta}_i$ (accounting for the variance in b_{i0})? You should use the values in $\mathbf{t}$ to express this explicitly.

The mean vector can be calculated in the standard way

$$\begin{aligned} E[\hat{\beta}_i] &= (X^T X)^{-1} X^T E[\mathbf{y}] \\ &= (X^T X)^{-1} X^T [(\beta_0 + E[b_{i0}])\mathbf{1} + \beta_1 \mathbf{t} + E[\epsilon_i]] \\ &= (X^T X)^{-1} X^T [\beta_0 \mathbf{1} + \beta_1 \mathbf{t}] \\ &= (X^T X)^{-1} X^T X \beta = \beta \end{aligned}$$

But we can use the values of $\mathbf{t}$ to compute $(X^T X)^{-1} X^T$ explicitly:

```r
t_vec = c(-2, -1, 0, 1, 2)
X = cbind(rep(1, 5), t_vec)
mat = solve(t(X) %*% X) %*% t(X)
print(mat)
```

```
##       [,1] [,2] [,3] [,4] [,5]
##        0.2  0.2  0.2  0.2  0.2
## t_vec -0.2 -0.1  0.0  0.1  0.2
```

So we have

$$E[\hat{\beta}_i] = \begin{pmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ -0.2 & -0.1 & 0 & 0.1 & 0.2 \end{pmatrix} [\beta_0 + \beta_1 \mathbf{t}]$$

Then for the variance we have

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

$$= \begin{pmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ -0.2 & -0.1 & 0 & 0.1 & 0.2 \end{pmatrix} ((\tau^2 J_5 + \sigma^2 I_5)) \begin{pmatrix} 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ -0.2 & -0.1 & 0 & 0.1 & 0.2 \end{pmatrix}^T.$$

$$= \begin{pmatrix} \sigma^2/5 + \tau^2 & 0 \\ 0 & \sigma^2/10 \end{pmatrix}$$

b. We can estimate a global $\hat{\beta}_1$ (which applies to all rats) from averating the n $\hat{\beta}_{i1}$ values. What is the variance of this estimate?

$$Var(\sum_{i=1}^{n} \hat{\beta}_{i1}/n) = \frac{1}{n^2} Var(\sum_{i=1}^{n} \hat{\beta}_{i1}) = \frac{1}{n^2} \sum_{i=1}^{n} Var(\hat{\beta}_{i1}) = \frac{n\sigma^2/10}{n^2} = \frac{\sigma^2/10}{n}.$$

c. To produce confidence intervals, we need to estimate $\hat{\sigma}^2$. Show that the average MSE from each model in Part a. provides an unbiassed estimate. Also show how this is related to a χ^2 and give it's degrees of freedom. Why is it independent of $\hat{\beta}_1$? (Hint: think about what you know about what results from running each model in turn.)

We can express the average MSE over all subjects as

$$\frac{1}{n} \sum_{i=1}^{n} \frac{1}{5-2} \sum_{j=1}^{5} (y_{ij} - \hat{y}_{ij})^2$$

but note that the estimates for the y's control for the β terms as well as the b_{i0} terms, meaning that the quantity $y_{ij} - \hat{y}_{ij}$ is mean zero with variance σ^2. Specifically with H being the fitted values for the matrix $X = [\mathbf{1}, \mathbf{t}]$ we have

$$(I - H)\mathbf{y} = (I - H)[\mathbf{1}(\beta_0 + b_{i0}) + \mathbf{t}\beta_1 + \epsilon] = (I - H)\epsilon.$$

So in the standard fashion (since we have 5-2 degrees of freedom for each subject) we get

$$E\left[\frac{1}{5-2} SSE\right] = E\left[\frac{1}{5-2} \sum_{j=1}^{5} (y_{ij} - \hat{y}_{ij})^2\right] = \sigma^2$$

and therefore the average MSE will also be an unbiased estimate for σ^2.

In terms of a Chi-squared distribution we can re-write the expression for the average MSE as

$$\frac{1}{n(5-2)} \sum_{i=1}^{n} \sum_{j=1}^{5} (y_{ij} - \hat{y}_{ij})^2$$

which is the sum of $5n$ mean zero normals with variance σ^2 and we have estimated two parameters for each subject, so we get that

$$\frac{1}{\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{5}(y_{ij}-\hat{y}_{ij})^2 \sim \chi^2_{n(5-2)}$$

We have shown previously and in class that the sum of squared errors is independent from the estimated coefficients, $\hat{\beta}$, giving that an estimate formed from the sum of squares will be independent from the estimated coefficients, including $\hat{\beta}_1$.

d. If we estimate $\hat{\beta}_0$ by the average of the $\hat{\beta}_{i0}$, what is its variance? Show that the estimate

$$\frac{1}{n-1}\sum(\hat{\beta}_{i0}-\hat{\beta}_0)^2$$

can be used to provide an estimate of this. Hence obtain a t-statistic for estimate β_0.

First,

$$Var(\hat{\beta}_0) = \frac{1}{n^2}\sum_{i=1}^{n} Var(\hat{\beta}_{i0}) = \frac{(\tau^2+\sigma^2/5)}{n}.$$

Now for estimation, note $\hat{\beta}_{i0}-\hat{\beta}_0$ is mean zero (since they have the same expectation). Therefore if we divide by the variance of $\hat{\beta}_{i0}$ we get

$$\frac{\sum(\hat{\beta}_{i0}-\hat{\beta}_0)^2}{(\tau^2+\sigma^2/5)} \sim \chi^2_{n-1}$$

meaning

$$E\left[\frac{\sum(\hat{\beta}_{i0}-\hat{\beta}_0)^2}{n-1}\right] = (\tau^2+\sigma^2/5)$$

so

$$E\left[\frac{\sum(\hat{\beta}_{i0}-\hat{\beta}_0)^2}{n(n-1)}\right] = (\tau^2+\sigma^2/5)/n = Var(\hat{\beta}_0)$$

Therefore

$$\frac{\hat{\beta}_0}{Q^{1/2}} \sim t_{n-1},$$

where $Q = \frac{\sum(\hat{\beta}_{i0}-\hat{\beta}_0)^2}{n(n-1)}$

e. Given your answers above, how would you estimate τ^2?

We have that the MSE is an unbiased estimate of σ^2, so

$$E\left[\frac{\sum(\hat{\beta}_{i0}-\hat{\beta}_0)^2}{(n-1)} - 0.2MSE\right] = \tau^2$$

f. Give an estimate of the conditional expectation of b_{i0} given the data and our estimates above.

We have that

$$\begin{pmatrix} b_{i0} \\ \hat{\beta}_{i0} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ \beta_0 \end{pmatrix}, \begin{pmatrix} \tau^2 & \tau^2 \\ \tau^2 & \tau^2+\sigma^2/5 \end{pmatrix}\right)$$

Thus

$$b_{i0}|\hat{\beta}_{i0} \sim N\left(\frac{\tau^2}{\tau^2+\sigma^2/5}(\hat{\beta}_{i0}-\beta_0), \tau^2 - \frac{\tau^4}{\tau^2+\sigma^2/5}\right) = N\left(\frac{\tau^2}{\tau^2+\sigma^2/5}(\hat{\beta}_{i0}-\beta_0), \tau^2\frac{\sigma^2}{\tau^2+\sigma^2/5}\right)$$

bonus What is the joint covariance of the full vector $\mathbf{y}$ combining all the rats (you may express this in terms of blocks).

First, in terms of indices:

$$cov(y_{ij}, y_{kl}) = \begin{cases} 0 & i \neq k \\ \tau^2 & i = k, j \neq l \\ \tau^2 + \sigma^2 & i = k, j = l \end{cases}$$

which can be expressed as

$$Var(\mathbf{y}) = \tau^2 J_R \otimes I_K + \sigma^2 I_{RK}$$

where R is the number of observations for each subject and K is the number of subjects.

bonus Derive the maximum likelihood estimates of the parameters $(\beta_0, \beta_1, \sigma^2, \tau^2)$.

Question 4

The data in `rats.csv` contain the results of an longitudinal study of rat growth. Rats were given three different diets and their size (calipers around their heads) were recorded at a number of different times. Each rat grows approximately linearly, but they all start from different sizes and grow at different rates, even in the same group.

Within a group, we can express this as

$$y_{ij} = \beta_0 + \beta_1 t_{ij} + b_{i0} + b_{i1} t_{ij} + \epsilon_{ij}$$

for the measurement of the ith rat at time t_{ij} with random intercept b_{i0} and random slope b_{i1} where we specify $\epsilon_{ij} \sim N(0, \sigma_e^2)$, $b_{i0} \sim N(0, \sigma_{b_0}^2)$ and $b_{i1} \sim N(0, \sigma_{b_1}^2)$.

a) Write down a model for y_{ijk} to cover rat i in group k at observation time j in which the *average* growth rate in each group is different between rats.

We can write the model as:

$$y_{ijk} = \beta_0 + \beta_{1k} t_{ijk} + b_{i0} + b_{i1} t_{ijk} + \epsilon_{ijk}.$$

b) Fit this model in `lme4`, can you conclude that the growth rate is different between different groups?

```
library(lme4)
```

```
## Warning: package 'lme4' was built under R version 3.5.1
```

```
## Loading required package: Matrix
```

```
rats = read.csv("./rats.csv", header=TRUE)
rats$GROUP = as.factor(rats$GROUP)
mod = lmer(RESPONSE ~ 1 + (TIME|SUBJECT) + TIME:GROUP, data=rats)
summary(mod)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: RESPONSE ~ 1 + (TIME | SUBJECT) + TIME:GROUP
##    Data: rats
##
## REML criterion at convergence: 1080.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.13289 -0.58315  0.03061  0.58849  2.32303
##
```

```
## Random effects:
##  Groups   Name        Variance  Std.Dev. Corr
##  SUBJECT  (Intercept) 2.967e+00 1.722444
##           TIME        4.181e-06 0.002045 1.00
##  Residual             2.729e+00 1.652021
## Number of obs: 251, groups:  SUBJECT, 50
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 63.113984   0.503689  125.30
## TIME:GROUP1  0.201142   0.007311   27.51
## TIME:GROUP2  0.182102   0.007298   24.95
## TIME:GROUP3  0.200141   0.008695   23.02
##
## Correlation of Fixed Effects:
##             (Intr) TIME:GROUP1 TIME:GROUP2
## TIME:GROUP1 -0.647
## TIME:GROUP2 -0.633  0.410
## TIME:GROUP3 -0.624  0.404       0.395
```

If we look at a summary of this model we see that the t values associated with the interaction terms between the TIME and GROUP variables are all >20, which is extreme enough that we should believe that there are differences in the growth rates.

c) Why does it not make sense to compare σ_e^2 and $\sigma_{b_1}^2$ when looking at how much variation is due to what source? If you did want to compare sources of variation, suggest a way to do it.

It doesn't make sense to compare σ_e^2 and $\sigma_{b_1}^2$ since one is related to raw noise and one is related to noise in the slope and hence has different units.

If we wanted to compare these sources of variation we could consider the empirical estimate for σ_e^2 (provided by `lmer`) and then getting an estimate for $\sigma_{b_1}^2$ from the estimates of the b_{i1}'s (likewise with $\sigma_{b_0}^2$). With the estimates for $\sigma_{b_1}^2$ and σ_e^2 we can look at all of the estimated covariances of the indivual y_{ijk}'s where

$$cov(y_{ijk}, y_{ijk})_{est.} = \hat{\sigma}_{b_0}^2 + t_{ijk}^2 \hat{\sigma}_{b_1}^2 + \hat{\sigma}_e^2.$$

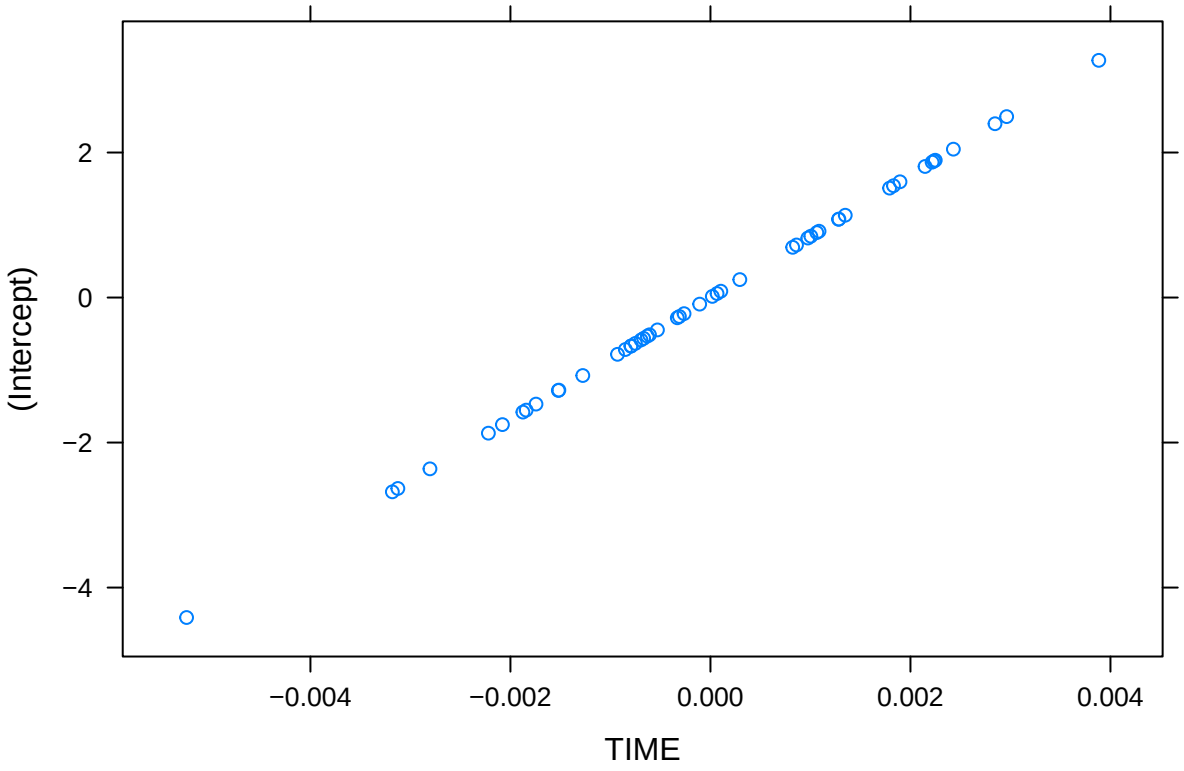
then we can (at least point by point) look at the variability due to each source.

d) Produce a scatter plot of the random intercepts versus random slopes for each rat (You can get this from `ranef`). Do these appear related?

If we plot the random effects (i.e. b_{i0} and b_{i1}) we see a nearly perfectly linear trend between the two:

```
plot(ranef(mod))
```

```
## $SUBJECT
```


(Intercept)
2
0
−2
−4
−0.004
−0.002
0.000
0.002
0.004
TIME