Lab 3 - On the Effect of Centering Variables

Lab Goals

The first two questions in this lab are designed to make you practice some theoretical calculations, while the third confirms these results using the `hills` data from Lab 2. When going through these, you may find it helps to have a very simple example to work through, either by hand or in R. Here we recommend using a $p = 2$ case with

$$x_1 = (1, 2, 2, 3),\ x_2 = (1, 1, 2, 4),\ \beta = (1, 1, 1),\ y = (2, 4, 6, 8)$$

We'll set this up in R:

```
x1 = c(1,2,2,3)
x2 = c(1,1,2,4)
beta = c(1,1,1)
y = c(2,4,6,8)
```

and we'll also define X and C matrices

```
X = cbind( rep(1,4), x1, x2)
C = diag(4) - matrix(1/4,4,4)
```

which we will use later. It will also help to define centered versions of covariates:

```
x1c = C%*%x1
x2c = C%*%x2
Xc = cbind( rep(1,4), x1c, x2c)
```

Centering, Linear Regression and ANOVAs

In class, we've relied on centered variables to clean up some calculations, and we've (rather blythely) said that this is OK (see the derivation of VIFs for example). Here we'll work through showing that this is the case:

1. By manipulating the equation
$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon$$
show that changing covariates from x_j to $x_j^* = Cx_j$ changes β_0 but not β_j.

Solution

$$\begin{aligned} y &= \beta_0 + \beta_1 \bar{x}_1 + \ldots + \bar{\beta}_p x_p + \beta_1(x_1 - \bar{x}_1) + \ldots + \beta_p(x_p - \bar{x}_p) + \epsilon \\ &= \beta_0^* + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon \end{aligned}$$

2. Show that the *estimated* $\hat{\beta}_j$ are also unaffected by centering the covariates. Do this by arguing that the *fitted values* are the same and these minimize mean squared error. Below, we will show that this is true by explicitly using matrix algebra.

Solution

The fitted values can be expressed as

$$\begin{aligned}\hat{y} &= \hat{\beta}_0 + \hat{\beta}_1\bar{x}_1 + \ldots + \hat{\bar{\beta}}_p x_p + \hat{\beta}_1(x_1 - \bar{x}_1) + \ldots + \hat{\beta}_p(x_p - \bar{x}_p) + \epsilon \\ &= \hat{\beta}_0^* + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_p x_p + \epsilon\end{aligned}$$

In R, we see that we can run either

```
mod = lm(y~x1+x2)
mod$coef
```

```
## (Intercept)          x1          x2
##  -0.3333333   2.0000000   0.6666667
```

or

```
modc = lm(y~x1c+x2c)
modc$coef
```

```
## (Intercept)         x1c         x2c
##   5.0000000   2.0000000   0.6666667
```

3. In simple linear regression, when x is centered, there is a simple formula for $\hat{\beta}_0$ and $\hat{\beta}_1$ and these are uncorrelated. When x is *not* centered, use the formula that for a 2×2 matrix,

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - cb}\begin{bmatrix} d & -c \\ -b & a \end{bmatrix}$$

   to find the covariance of $(\hat{\beta}_0, \hat{\beta}_1)$ and show that centering reduces the variance of $\hat{\beta}_0$ but leaves the variance of $\hat{\beta}_1$ unchanged.

Solution Here for X^TX we have $a = n$, $b = c = n\bar{x}$ and $d = x^Tx$ and the formula gives us that

$$(X^TX)^{-1} = \frac{1}{nx^Tx - n^2\bar{x}^2}\begin{bmatrix} x^TX & -n\bar{x} \\ -n\bar{x} & n \end{bmatrix} = \begin{bmatrix} (SXX + n\bar{x}^2)/nSXX & -\bar{x}/SXX \\ -\bar{x}/SXX & 1/SXX \end{bmatrix}$$

because $SXX = x^Tx - n\bar{x}^2$.

So $\text{cov}(\hat{beta}_0, \hat{\beta}_1) = -\bar{x}/SXX$ and we observe that if $\bar{x} = 0$ the 2,2 entry of this matrix does not change, but the 1,1 entry is reduced by $\bar{x}^2/SXX$.

In our toy example in R, we can consider only the first covariate and look at

```
mod1 = lm(y~x1)
vcov(mod1)
```

```
##             (Intercept)   x1
## (Intercept)        2.25 -1.0
## x1                -1.00  0.5
```

versus

```
mod1c = lm(y~x1c)
vcov(mod1c)
```

```
##             (Intercept) x1c
## (Intercept)        0.25 0.0
## x1c                0.00 0.5
```

4. If X is the design matrix, including the intercept, for the model above, find A so that $XA = X^*$ where the columns of X^* are all centered except for the first column of 1's.

Solution

$$A = \begin{bmatrix} 1 & -\bar{x}^T \\ 0 & I \end{bmatrix}$$

where $\bar{x}^T = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_p)$ and 0 is a vector of p 0's.

As an example in R, if we consider

```
A = matrix( c(1,-2,-2,0,1,0,0,0,0,1),3,3,byrow=TRUE)
X%*%A
```

```
##      [,1] [,2] [,3]
## [1,]    1   -1   -1
## [2,]    1    0   -1
## [3,]    1    0    0
## [4,]    1    1    2
```

compare to

```
Xc
```

```
##      [,1] [,2] [,3]
## [1,]    1   -1   -1
## [2,]    1    0   -1
## [3,]    1    0    0
## [4,]    1    1    2
```

5. Find A^{-1} from the previous question (hint: A will be given in terms of the $\bar{x}_j$, replace each of these with $-\bar{x}_j$ and show that this gives you the inverse).

Solution

$$\begin{bmatrix} 1 & -\bar{x}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} 1 & \bar{x}^T \\ 0 & I \end{bmatrix} = \begin{bmatrix} 1 & \bar{x}^T - \bar{x}^T \\ 0 & I \end{bmatrix} = I_{p+1}$$

```
solve(A)
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    2
## [2,]    0    1    0
## [3,]    0    0    1
```

5. By expanding $(X^{*T}X^*)^{-1}$ in terms of $(X^TX)^{-1}$ and A^{-1}, show that the result from (3) is also true of multiple regression. Hence our Variance Inflation Factor identity also holds for un-centered covariates.

Solution I will write $S = (X^TX)_1$ for convenientce, then

$$A^{-1}(X^TX)^{-1}(A^{-1})^T = \begin{bmatrix} S_{11} + S_{1,-1}\bar{x} + \bar{x}^T S_{-1,1} + \bar{x}^T S_{-1,-1}\bar{x} & S_{1,-1} + \bar{x}^T S_{-1,-1} \\ S_{-1,1} + S_{-1,-1}\bar{x} & S_{-1,-1} \end{bmatrix}$$

where we seee that only the first row and column are changed, and that the (1,1)th entry is smallest when $\bar{x} = 0$.

```
vcov(mod)
```

```
##             (Intercept)        x1         x2
## (Intercept)   3.8888889 -2.666667  0.8888889
## x1           -2.6666667  2.666667 -1.3333333
## x2            0.8888889 -1.333333  0.8888889
```

```
vcov(modc)
```

```
##             (Intercept)       x1c        x2c
## (Intercept)   0.3333333  0.000000  0.0000000
## x1c           0.0000000  2.666667 -1.3333333
## x2c           0.0000000 -1.333333  0.8888889
```

In Data

Let's look at what we actually get in data. For this question, we use the data used in Lab 2.

```
hills=read.csv("hills.csv")
fit=lm(Time~Distance+Climb,data=hills)
summary(fit)
```

```
##
## Call:
## lm(formula = Time ~ Distance + Climb, data = hills)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.215  -7.129  -1.186   2.371  65.121
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.992039   4.302734  -2.090   0.0447 *
## Distance     6.217956   0.601148  10.343 9.86e-12 ***
## Climb        0.011048   0.002051   5.387 6.45e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.68 on 32 degrees of freedom
## Multiple R-squared:  0.9191, Adjusted R-squared:  0.914
## F-statistic: 181.7 on 2 and 32 DF,  p-value: < 2.2e-16
```

1. Create a centering matrix and apply this to center Distance and Climb. Show that applying lm to these data gives you the same slopes, but changes the intercept.

```
# Centering matrix
C = diag(35) - matrix(1/35,35,35)

# Here we'll center
Chills = hills
Chills[,2] = C%*%hills[,2]
Chills[,3] = C%*%hills[,3]

# And call lm
Cfit = lm(Time~Distance+Climb,data=Chills)
summary(Cfit)
```

```
##
## Call:
## lm(formula = Time ~ Distance + Climb, data = Chills)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -16.215  -7.129  -1.186   2.371  65.121
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 57.875714   2.480615  23.331  < 2e-16 ***
## Distance     6.217956   0.601148  10.343 9.86e-12 ***
## Climb        0.011048   0.002051   5.387 6.45e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.68 on 32 degrees of freedom
## Multiple R-squared:  0.9191, Adjusted R-squared:  0.914
## F-statistic: 181.7 on 2 and 32 DF,  p-value: < 2.2e-16
```

2. Check that the intercept is, indeed, changed by the sum of the slopes times the average of each of distance and time: $\beta_1\bar{x}_1 + \beta_2\bar{x}_2$.

```
fit$coef[1] - Cfit$coef[1]
```

```
## (Intercept)
##   -66.86775
```

```
Cfit$coef[2]*mean(hills$Distance) + Cfit$coef[3]*mean(hills$Climb)
```

```
## Distance
## 66.86775
```

3. Confirm that X^TX is block diagonal when the centered covariates are used.

```
X = as.matrix(cbind(rep(1,35),Chills[,2:3]))
t(X)%*%X
```

```
##              rep(1, 35)     Distance        Climb
## rep(1, 35) 3.500000e+01 2.842171e-14 9.094947e-13
## Distance   2.842171e-14 1.037471e+03 1.983777e+05
## Climb      9.094947e-13 1.983777e+05 8.913605e+07
```

```
# Here we note that the off-diagonals on the first row and column are very close to zero.
```

4. Now we will re-create the functions in

```
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: Time
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Distance   1  71997   71997 334.293 < 2.2e-16 ***
## Climb      1   6250    6250  29.018 6.445e-06 ***
## Residuals 32   6892     215
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

a. First, create the matrices $X_1 = [\mathbf{1}_n, \mathbf{x}_1]$, and the matrix $X_2 = [X_1, \mathbf{x}_2]$ along with the corresponding hat matrices H_1 and H_2.

```
X1 = as.matrix(cbind(rep(1,35),hills[,2]))
X2 = as.matrix(cbind(X1,hills[,3]))
```

```
H1 = X1%*%solve(t(X1)%*%X1)%*%t(X1)
H2 = X2%*%solve(t(X2)%*%X2)%*%t(X2)
```

b. Form the matrices for the sums of squares $A_1 = H_1 C H_1$, $A_2 = H - H_1$, $A_3 = I - H$, what are their traces?

```
A1 = H1%*%C%*%H1
sum(diag(A1))
```

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

```
A2 = H2 - H1
sum(diag(A2))
```

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

```
A3 = diag(35) - H2
sum(diag(A3))
```

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

c. Show that $\mathbf{y}^T A \mathbf{y}$ gives the sum of squares for the corresponding term in *anova(fit)* above.

```
y = hills$Time

y%*%A1%*%y
```

```
##          [,1]
## [1,] 71996.89
```

```
y%*%A2%*%y
```

```
##          [,1]
## [1,] 6249.738
```

```
y%*%A3%*%y
```

```
##          [,1]
## [1,] 6891.867
```

d. Confirm that these sum to $\mathbf{y}^T c \mathbf{y}$.

```
y%*%C%*%y
```

```
##          [,1]
## [1,] 85138.49
```

```
y%*%A1%*%y + y%*%A2%*%y + y%*%A3%*%y
```

```
##          [,1]
## [1,] 85138.49
```

e. Verify that these are also respectively the sum of squared differences between
   i) a regression on Distance and the mean
   ii) a regression on both Distance and Climb and a regression on distance
   iii) the observations and a regression on both variables Use the hat matrices that you created to do this.

```
# First A1
y%*%A1%*%y
```

```
##          [,1]
```

```
## [1,] 71996.89
```

```
yhat1 = H1%*%y
sum( (yhat1 - mean(y))^2 )
```

```
## [1] 71996.89
```

```
# Now A2
y%*%A2%*%y
```

```
##           [,1]
## [1,] 6249.738
```

```
yhat2 = H2%*%y
sum( (yhat2 - yhat1)^2 )
```

```
## [1] 6249.738
```

```
# And A3
y%*%A3%*%y
```

```
##           [,1]
## [1,] 6891.867
```

```
sum( (y - yhat2)^2 )
```

```
## [1] 6891.867
```