Lab 4 - Variance Inflation Factors

The purpose of this lab is to explore variance inflation factors. As with understanding sums of squares, we'll try to set up an experiment in which we can construct an alternative estimator which attains the variance implied by VIFs.

Obtaining Minimal Variances

Variance Inflation Factors are somewhat slippery concepts. They refer to the variance in $\hat{\beta}_j$ if you were doing a simple linear regression of y on x_j, *but you had the same error variance as the multiple regression model.* Here we will explore just what we mean.

Continuing our example from Lab 3:

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

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

```
## [1] 1.154701
```

1. By expressing $X = [1, x_1, X_{-1}]$ and considering a sequential ANOVA table, show that if we ignore all the covariates except x_1, we then the Mean Square Error is larger than for the full regression. The VIF does not account for an increase in our *estimate* of $\hat{\sigma}^2$.
2. In fact, what would the mean and variance (*not* the estimated mean and variance) of $(\hat{\beta}_0, \hat{\beta}_1)$ be if we just regressed y on x_1?
3. As an alternative way to think about VIF's, consider transforming X_{-1} to $X^*_{-1} = (I - H_1)X_{-1}$ (note adding a star to a variable just means its had some transform, not necessarily centered) and using $[1, x_1, X^*_{-1}]$ as a design matrix.

- 3a. Show that $x_1^T X^*_{-1} = 0$. In our example, this amounts to regressing x_2 on x_1 and taking the residuals

```
x2mod = lm(x2~x1)
r2 = x2mod$residual

t(X[,1:2])%*%r2
```

```
##    [,1]
##       0
## x1    0
```

- 3b. Hence show that using these covariates, $\hat{\beta}_1$ has its minimum possible variance. In R, we'll just look at a new version of $(X^T X)^{-1}$

```
Xr2 = cbind( rep(1,4), x1, r2)
solve( t(Xr2)%*%Xr2 )
```

```
##            x1        r2
##     2.25 -1.0 0.0000000
## x1 -1.00  0.5 0.0000000
## r2  0.00  0.0 0.6666667
```

```
# Compare to
solve( t(X[,1:2])%*%X[,1:2] )
```

```
##            x1
##     2.25 -1.0
## x1 -1.00  0.5
```

- 3c. Interpret this procedure in terms of how we are partitioning the regression sums of squares between x_1 and all the other covariates.

4. Do we achieve this variance if instead of changing X_{-1}, we change $x_1^* = (I - H_{-1})x_1$?

```
r1 = lm(x1~x2)$residuals
Xr1 = cbind( rep(1,4), r1, x2)
solve( t(Xr1)%*%Xr1 )
```

```
##                           r1            x2
##     9.166667e-01  3.423188e-16 -3.333333e-01
## r1  3.423188e-16  2.000000e+00 -1.850372e-16
## x2 -3.333333e-01 -1.850372e-16  1.666667e-01
```

In Real Data

We'll examine this from the data used in Lab 2.

1. Fit a model to predict Time from Distance and Climb and obtain VIF's for their coefficients using the function `vif()` in the `car` package.

```
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
```

```r
library(car)
```

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

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

```r
vif(fit)
```

```
## Distance    Climb
## 1.740812 1.740812
```

```r
sigma = summary(fit)$sigma
```

2. Using the the result above, what is the smallest possible error standard deviation for the coefficient of Distance? (Remember that VIF's are in terms of variances rather than standard deviations).

```r
0.6/sqrt(1.74)
```

```
## [1] 0.4548588
```

3. Fitting a linear regression of Time onto Distance, only, does its coefficient have this estimated standard error?

```r
fit1=lm(Time~Distance,data=hills)
summary(fit1)
```

```
##
## Call:
## lm(formula = Time ~ Distance, data = hills)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -35.745  -9.037  -4.201   2.849  76.170
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.8407     5.7562  -0.841    0.406
## Distance      8.3305     0.6196  13.446 6.08e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.96 on 33 degrees of freedom
## Multiple R-squared:  0.8456, Adjusted R-squared:  0.841
## F-statistic: 180.8 on 1 and 33 DF,  p-value: 6.084e-15
```

4. Fit a model to predict Climb from Distance and obtain residuals from this model; call them rClimb. Now predict Time from Distance and rClimb and observe that the covariate of Distance has estimated standard error that you found in Part (2).

```r
rClimb = lm(Climb~Distance,data=hills)$residuals
fit2 = lm(Time~Distance+rClimb,data=hills)
summary(fit2)
```

```
##
## Call:
## lm(formula = Time ~ Distance + rClimb, 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) -4.840720   4.233160  -1.144    0.261
## Distance     8.330456   0.455623  18.284  < 2e-16 ***
## rClimb       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
```

5. Is the value of the estimated coefficient of Distance the same in all three models? Does it mean the same thing?

6. Conduct a simulation (copy from Lab 2) in which you generate new y's from the fitted values of your first model, but then use these to re-fit regressions on Distance only as well as on Distance and rClimb. Record the coefficient of Distance in both fitted models. What is the standard error from your simulated estimates?

```
N=1000            # This many simulations
B1=matrix(0,N,2)   # Distance Only
B2=matrix(0,N,3)  # Distance and rClimb

for (i in 1:N)
  {
    y=fit$fitted+rnorm(nrow(hills),sd=sigma) # new simulated data
    B1[i,]=lm(y~hills$Distance)$coef
    B2[i,] = lm(y~hills$Distance+rClimb)$coef
  }

sd(B1[,2])
```

```
## [1] 0.4547821
```

```
sd(B2[,2])
```

```
## [1] 0.4547821
```

Some Theory Practice

Giles is interested in whether the size of the bags that students bring to class varies with their study program. He proposes measuring the volume of bags (by pouring water into them, of course!) as they come to class and also recording their height and whether they are Statistical Science majors (SS), Biometry and Statistics majors (BS) or MPS (MPS).

He then proposes to form a covariate matrix $X = [1, x_1, x_2, x_3, x_4]$ where - x_1 indicates the students height, - x_2 is 1 if SS and 0 otherwise, - x_3 is 1 if BS and 0 otherwise, - x_4 is 1 if MPS and 0 otherwise

He will then use this to provide an equation to predict bag volume from height and program via the usual linear regression procedures.

1. Do you see any problems with the design of the matrix X? In particular, does it have full rank? If not, how do you suggest proceeding?

2. Giles is particularly interested in whether there is a difference between SS and BS bags. To that end he wants to examine the difference $\beta_2 - \beta_3$.

i. What is the mean and variance of $\hat{\beta}_2 - \hat{\beta}_3$? Give an expression in terms of X.

ii. Produce a t-test procedure to test the null hypothesis

$$H_0 : \beta_2 - \beta_3 = \delta$$

State your test statistic and test distribution including degrees of freedom.

iii. Hence derive a confidence interval for δ.

3. There is good reason to believe that the error variance should be known at 1 gallon^2 (bags tend to come in whole-number volumes). If this were the case, how would your test and confidence interval in part 2 change?

4. How would you test the hypothesis $H_0 : \sigma^2 = 1$? We would especially like to make sure that σ^2 is not larger than we think it should be, so the sensible alternative would be $H_A : \sigma^2 > 1$. *Hint:* since σ^2 is known under H_0, we should be able to normalize SSE to get a known distribution.

Additional Practice Problems (not for lab)

- From Christiensen: Ex 2.10.2, 2.10.4, 3.9.3,
- Moser Chapter 2, Questions 3, 5; Chapter 5, Questions 5, 7,
- A final practice: in the hills data in lab, we know that a test of

$$H_0 : \beta_1 = \beta_2 = 0$$

is given by MSR/MSE, which we compare to a critical value. Here we would like to use this test to obtain a joint confidence *region* for (β_1, β_2).

i. For a candiate $H_0 : (\beta_1, \beta_2) = (b_1, b_2)$, show that replacing y with $z = y - b_1 x_1 - b_2 x_2$ yields a model in which H_0 is true.

ii. If the estimates using z are $(\tilde{\beta}_1, \tilde{\beta}_2)$, and those with y are $(\hat{\beta}_1, \hat{\beta}_2)$ show that

$$(\tilde{\beta}_1, \tilde{\beta}_2) = (\hat{\beta}_1 - b_1, \hat{\beta}_2 - b_2)$$

iii. Hence show that we reject this null hypothesis for

$$\frac{(\hat{\beta} - b)^T A (\hat{\beta} - b)}{\text{MSE}} > F^p_{n-p-1,\alpha}$$

Specify the matrix A. What shape does this correspond to?

iv. We can visualize this by simply using a set of candidate (b_1, b_2) that form a grid in space. Here we will do this simply as

```r
hills = read.csv('hills.csv',head=TRUE)
```

Fit a linear regression

```r
mod1 = lm(Time~Distance+Climb,data=hills)

summary(mod1)
```

```
##
## 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
```

We'll set up a sequence of points coverin the confidence intervals for the coefficients

```
b1 = seq(4,8,len=101); b2 = seq(0.005,0.018,len=101)
```

Now for each combination of b1 and b2, we'll remove the signal from Time, fit a new linear model and see if it is significant.

First we need a matrix to store whether or not the test is above the critical values

```
is.sig = matrix(0,101,101)
```

and the critical value for F

```
fcrit = qf(0.95,2,32)
```

Now we run through the values of b_1 and b_2 where we subtract the signal

```
for(i in 1:101){
 for(j in 1:101){
  # Subtract signal
  z = hills$Time - b1[i]*hills$Distance - b2[j]*hills$Climb
  # Fit linear model
  zmod = lm(z~hills$Distance+hills$Climb)
  # See if it is significant
  is.sig[i,j] = (summary(zmod)$fstatistic[1] > fcrit)
 }
}

# And plot

image(b1,b2,is.sig)
```

Here the elipse in red represents the combined values of (β_1, β_2) that are consistent with the data.