8004 Homework 4

Nooreen Dabbish
February 18, 2015

1 Problem 1 In the context of Problem 2 of Homework Assignment 3, use R matrix calculations to do the following in the (non-full-rank) Gauss-Markov normal linear model

(a) Find 90% two-sided confidence limits for σ.

The model described in HW3, Problem 2 in $\mathbf{Y} = \mathbf{X}\beta + \epsilon$ matrix form is:

$$\begin{pmatrix} y_{11} \\ y_{12} \\ y_{21} \\ y_{31} \\ y_{41} \\ y_{42} \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 4 \\ 6 \\ 3 \\ 5 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} + \begin{pmatrix} \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{21} \\ \epsilon_{31} \\ \epsilon_{41} \\ \epsilon_{42} \end{pmatrix}$$

Because the problem statement says this is a Gauss-Markov normal linear model, we know that $\mathbf{Y} \sim \mathrm{N}(\mathbf{X}\beta, \sigma^2\,\mathbf{I})$.

(a).1 Interval for σ using I

The Gauss-Markov normal linear model assumes that the var($\mathbf{Y}$) = σ^2 $\mathbf{I}$, and in this case we are able to solve for SSE directly from $\hat{\mathbf{Y}}$ and $\mathbf{X}$.

```
Yhat <- X %*% ginv(t(X) %*% X) %*% t(X) %*% Y

SSE1a3 <- t(Y-Yhat) %*% (Y-Yhat)

lowerchi <- qchisq(.05, df=(length(Y) - qr(X)$rank))
upperchi <- qchisq(.95, df=(length(Y) - qr(X)$rank))
```

For the Gauss-Markov linear model of HW3 Problem 2, we found an SSE of `2.5` and two-sided 90% confidence limits for σ of `0.5656` $< \sigma <$ `2.6656`.

(b) Find 90% two-sided confidence limits for $\mu + \tau_2$.

The following provides 90% confidence limits for $\mu + \tau_2$ in the Gauss-Markov model first, where $\mathbf{Y} \sim \mathrm{N}_6(\mathbf{X}\beta,\, \sigma^2\,\mathbf{I})$ and then in the GLS cases with var($\mathbf{Y}$) = σ^2 $\mathbf{V}_1$ and var($\mathbf{Y}$) = σ^2 $\mathbf{V}_2$.

(c) Find 90% two-sided confidence limits for τ_1 - τ_2.

Proceeding as in part b, here $\tau_1 - \tau_2 = \mathbf{a}'\beta = (0, 1, -1, 0, 0) \begin{pmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix}$. Note that the quantile for $t_{\alpha/2}$ and value for s are calculated above.

```
a_1c = matrix(c(0,1,-1,0,0))
```

```
quad_1c <- sqrt(t(a_1c) %*% ginv(t(W)%*%W) %*% a_1c)
upper1c <- t(a_1c) %*% Bhat_1b - t_1b * s_1b * quad_1c
lower1c <- t(a_1c) %*% Bhat_1b + t_1b * s_1b * quad_1c
```

We find that the 90% confidence limits for τ_1 - τ_2 are from `-12.8237` to `7.8237`.

(d) Find a *p*-value for testing the null hypothesis $H_0 : \tau_1 - \tau_2 = 0$ vs H_a : not H_0.

(d).1 General Linear Hypothesis Test

The general linear hypothesis test is the following F test for $H_0 : \mathbf{C}\beta = \mathbf{0}$ verus $H_1 : \mathbf{C}\beta \neq \mathbf{0}$, given $\mathbf{y} \sim N_n(\mathbf{X}\beta, \sigma^2 \mathbf{I})$, $\mathbf{C}$ q x (/k/+1), rank($\mathbf{C}$) = q, with SSH = the sum of squares due to the hypothesis or due to $\mathbf{C}\beta$. Note that
$\frac{SSH}{\sigma^2} = \frac{(\mathbf{C}\hat{\beta})'[\mathbf{C}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{C}']^{-1}\mathbf{C}\hat{\beta}}{\sigma^2} \sim \chi^2(q, \frac{(\mathbf{C}\beta)'[\mathbf{C}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{C}']^{-1}\mathbf{C}\beta}{2\sigma^2})$
and
$\frac{SSE}{\sigma^2} = \frac{\mathbf{y}'[\mathbf{I}-\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}']\mathbf{y}}{\sigma^2} \sim \chi^2(n-k-1)$.
Taking the ratio gives us our test statistic:

$$F = \frac{SSH/q}{SSE/(n-k-1)}$$

- If $H_0 : \mathbf{C}\beta = \mathbf{0}$ is false, F ~ F(q,n-k-1,λ), where $\lambda = \frac{(\mathbf{C}\beta)'[\mathbf{C}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{C}']^{-1}\mathbf{C}\beta}{2\sigma^2}$).
- Notice that if $\mathbf{C}\beta = \mathbf{0}$ is true, λ defined above = 0, giving F ~ F(q, n-k-1).

(d).2 *p*-value from the F statistic

We need to find the F statistic described above. Here $\mathbf{C}$ is $\mathbf{a}$' from above, $\mathbf{a}$'=(0,1,-1,0,0), and $\mathbf{C}$ is 1 x 5 of rank 1, so q = 1. Note also that n=6, k=4, n-k-1=1.

```
SSH <- t(t(a_1c) %*% Bhat_1b) %*% ginv(t(a_1c)%*%ginv(t(W)%*%W)%*%a_1c)%*%t(a_1c)%*%Bhat_1b

p_1d <- pf(SSH/SSE, 1, 1, lower.tail=FALSE)
```

The *p*-value obtained was `0.7048`. This is the probability that the central F distribution exceeds the observed F. This suggests that we should accept the null hyposthesis.

(e) Find 90% two-sided predition limits for the sample mean of /n/=10 future observations from the first set of conditions.

(e).1 A t statistic for prediction

Consider future observation y_0, $y_0 = \mathbf{x}_0' \beta + \epsilon_0$ with $\hat{y}_0 = \mathbf{x_0'}\hat{\beta}$, where $\hat{y}_0$ is computed from n observations and y_0 is obtained independently. We find that $E(y_0 - \hat{y}_0) = 0$ and
$var(y_0 - \hat{y}_0) = var(\epsilon_0) + var(\mathbf{x_0'}\hat{\beta}) = \sigma^2[1 + \mathbf{x_0'}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x_0}]$, where $\widehat{var(y- \hat{y})} = s^2 2[1 + \mathbf{x_0'}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x_0}]$. Because of the independence of s^2 and y_0 and $\hat{y}_0$, we have the following t statistic:

$$t = \frac{y_0 - \hat{y}_0 - 0}{s\sqrt{1 + \mathbf{x_0'}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x_0}} \sim t(n-k-1)}$$

Therefore,

$$P = \left[-t_{\alpha/2,n-k-1} \le \frac{y_0 - \hat{y}_0 - 0}{s\sqrt{1 + \mathbf{x_0'}(\mathbf{X'X})^{-1}\mathbf{x_0}}} \le t_{alpha/2,n-k-a} \right] = 1 - \alpha$$

Re-arranging in terms of $\mathbf{x_0'}\hat{\beta} = \hat{y}_0$ gives:

$$\mathbf{x_0'}\hat{\beta} \pm t_{\alpha/2,n-k-1} s\sqrt{1 + \mathbf{x_0'}(\mathbf{X'X})^{-1}\mathbf{x_0}}.$$

(f) Find 90% two-sided prediction limits for the difference between a pair of future values, one from the first set of conditions (i.e. with mean $\mu + \tau_1$) and one from the second set of conditions (i.e. with mean $\mu + \tau_2$).

(g) Find a *p*-value for testing the following: What is the practical interpretation of this test?

$$H_0 : \begin{pmatrix} 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

(h) Find a *p*-value for testing:

$$H_0 : \begin{pmatrix} 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} \mu \\ \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{pmatrix} = \begin{pmatrix} 10 \\ 0 \end{pmatrix}.$$

2 Problem 2 In the following make use of the data in Problem 4 of Homework Assignment 3. Consider a regression of *y* on $x_1, x_2, \ldots, x_5$. Use R matrix calculations to do the following in a full rank Gauss-Markov normal linear model.

(a) Find 90% two-sided condifence limits for σ.

(b) Find 90% two-sided condifence limits for the mean response under the conditions of data point #1.

(c) Find 90% two-sided condifence limits for the difference in mean responses under the conditions of data points #1 and #2. .

(d) Find a *p*-value for testing the hypothesis that the conditions of data points #1 and #2 produce the same mean response.

(e) Find 90% two-sided prediction limits for an additional response for the set of conditions $x1 = 0.005, x2 = 0.45, x3 = 7, x4 = 45,$ and $x_5 = 6$.

(f) Find a *p*-value for testing the hypothesis that a model including only x_1, x_3, and x_5 is adequare for "explaining" home price.

(Hint: write it in the form of $H_0 : \mathbf{C}\beta = \mathbf{0}$). The full model in this problem is $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 + \epsilon$. The reduced model to test is $H_0 : \beta_2 = \beta_4 = 0$ or $y = \beta_0 + \beta_1 x_1 + \beta_3 x_3 + \beta_5 x_5 + \epsilon$. This can be written $\mathbf{C}\beta = \mathbf{0}$, with C = (0 0 1 0 1 0).

We can create a p-value to test these models using an F statistic, constructed out of the ratio of the difference in regression sum of squares between the full (SSR_{full}) and reduced($\text{SSR}_{\text{reduced}}$) models and the sum of squared error (SSE). These quantities are independent and follow a non-central $\chi^2(h,\lambda)$ and central $\chi^2(n\text{-}k\text{-}1)$ respectively where n is the number of observations, k is the number of parameters in the full model, and h is the difference in the number of parameters between the full and reduced models. The non-centrality parameter λ can be written $\beta_2\mathbf{'[X_2'X_2} - \mathbf{X_2'X_1(X_1'X_1)^{-1}X_1'X_2]}\beta_2\mathbf{/2}\sigma^2$ where $\mathbf{X}_1$ and $\mathbf{X}_2$ form a partition of $\mathbf{X}$ such that we can write:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon = (\mathbf{X}_1, \mathbf{X}_2)\begin{pmatrix}\beta_1\\ \beta_2\end{pmatrix} + \epsilon = \mathbf{X}_1\beta_1 + \mathbf{X}_2\beta_2 + \epsilon$$

And the reduced model would be $\mathbf{y} = \mathbf{X}_1\beta_1{}^\star + \epsilon^\star$.

```
#Find SSR in the full model.
SSR_Bf <- t(bhat_B) %*% t(X_B) %*% Y_B - (length(Y_B)*(mean(Y_B))^2)

#create reduced model design matric and X1_B and estimator bhat1_B
X1_B <- X_B[,-c(3,5)]
bhat1_B <- ginv(t(X1_B)%*%X1_B) %*% t(X1_B) %*% Y_B
SSR_Br <- t(bhat1_B) %*% t(X1_B) %*% Y_B - (length(Y_B)*(mean(Y_B))^2)

SSE_B <- t(Y_B)%*%Y_B - t(bhat_B)%*%t(X_B)%*%Y_B

F_2f <- ((SSR_Bf - SSR_Br)/2)/(SSE_B/(length(Y_B) - qr(X_B)$rank))

pf_2f <- pf(F_2f, 2, (length(Y_B)-(qr(X_B)$rank)), lower.tail=F)
pf_2f
```

This gives us a p-value of `3.19090353910822e-13`.

3 Problem 3

(a) In the context of Problem 1, part g), suppose that in fact $\tau_1 = \tau_2$, $\tau_3 = \tau_4 = \tau_1$ - dσ. What is the distribution of the F statistic?

(b) Use R to plot the power of the α = 0.05 level test as a function of *d* for *d* $\in$ [-5,5], that is plotting *P* (F > the cut-off value) against *d*. The R function pf(q,df1,df2,ncp) will compute cumulative (non-central) F probabilities for you corresponding to the value q, for degrees of freedom df1 and df2 when the noncentrality parameter is ncp.

4 Appendix: Tangled R code

```
library(MASS); library(xtable)
  lvector <- function(x, dig = 2, dsply=rep("f",ncol(x)+1)) {
   x <- xtable(x, align=rep("",ncol(x)+1),display=dsply,digits=dig) # We repeat empty string 6 time
   print(x, floating=FALSE, tabular.environment="pmatrix",
     hline.after=NULL, include.rownames=FALSE, include.colnames=FALSE)
   }

#Variables from Problem 2 of HW3:
  V1 <- diag(c(1,9,9,1,1,9))
  Y <- matrix(c(2, 1, 4, 6, 3, 5), nrow=6, ncol=1)
  X <- matrix(c(rep(1,6),
                1,1,0,0,0,0,
                0,0,1,0,0,0,
                0,0,0,1,0,0,
                0,0,0,0,1,1),nrow = 6,byrow=FALSE)

  V2 <- diag(c(1,9,9,1,1,9))
  V2[1,2] <- 1
  V2[2,1] <- 1
  V2[4,3] <- -1
  V2[3,4] <- -1
  V2[6,5] <- -1
  V2[5,6] <- -1


#Variables from Problem 4 of HW3:
data(Boston)
Y_B = as.matrix(Boston$medv)
X_B = as.matrix(Boston[,c('crim','nox','rm','age','dis')])
X_B = cbind(rep(1,dim(Boston)[1]),X_B)
bhat_B <- ginv(t(X_B)%*%X_B) %*% t(X_B) %*% Y_B
Yhat_B <- X_B %*% bhat_B
err_B <- Y_B - Yhat_B
sigsqhat_B <- t(err_B) %*% err_B / (dim(X_B)[1] - qr(X_B)$rank)

#Find V^(-1/2)
Vh1 <-solve(V1^(1/2))

#Transform model to OLS
U <- Vh1 %*% Y
W <- Vh1 %*% X

Uhat <- W %*% ginv(t(W) %*% W) %*% t(W) %*% U

SSE <- t(U-Uhat) %*% (U-Uhat)

qr(W)$rank
```

```
lowerchi <- qchisq(.05, df=(length(U) - qr(W)$rank))
upperchi <- qchisq(.95, df=(length(U) - qr(W)$rank))

SSE/lowerchi
SSE/upperchi

#Find V^(-1/2) using spectral decompostion
Vh2 <-solve(eigen(V2)$vectors %*% diag(sqrt(eigen(V2)$values)) %*% t(eigen(V2)$vectors))

#Transform model to OLS
U <- Vh2 %*% Y
W <- Vh2 %*% X

Uhat <- W %*% ginv(t(W) %*% W) %*% t(W) %*% U

SSE <- t(U-Uhat) %*% (U-Uhat)

qr(W)$rank

lowerchi <- qchisq(.05, df=(length(U) - qr(W)$rank))
upperchi <- qchisq(.95, df=(length(U) - qr(W)$rank))

Yhat <- X %*% ginv(t(X) %*% X) %*% t(X) %*% Y

SSE <- t(Y-Yhat) %*% (Y-Yhat)

lowerchi <- qchisq(.05, df=(length(Y) - qr(X)$rank))
upperchi <- qchisq(.95, df=(length(Y) -qr(X)$rank))

#Find the t distribution quantile
t_1b <- qt(.05, (length(Y) - qr(W)$rank - 1) )

a_1b = matrix(c(1,0,1,0,0))
s_1b <- sqrt(SSE/(length(Y) - qr(W)$rank - 1))
Bhat_1b <- ginv(t(W) %*% W) %*% t(W) %*% U
quad_1b <- sqrt(t(a_1b) %*% ginv(t(W)%*%W) %*% a_1b)
upper1b <- t(a_1b) %*% Bhat_1b - t_1b * s_1b * quad_1b
lower1b <- t(a_1b) %*% Bhat_1b + t_1b * s_1b * quad_1b

a_1c = matrix(c(0,1,-1,0,0))

quad_1c <- sqrt(t(a_1c) %*% ginv(t(W)%*%W) %*% a_1c)
upper1c <- t(a_1c) %*% Bhat_1b - t_1b * s_1b * quad_1c
lower1c <- t(a_1c) %*% Bhat_1b + t_1b * s_1b * quad_1c

SSH <- t(t(a_1c) %*% Bhat_1b) %*% ginv(t(a_1c)%*%ginv(t(W)%*%W)%*%a_1c)%*%t(a_1c)%*%Bhat_1b

p_1d <- pf(SSH/SSE, 1, 1, lower.tail=FALSE)
```

```
#Find SSR in the full model.
SSR_Bf <- t(bhat_B) %*% t(X_B) %*% Y_B - (length(Y_B)*(mean(Y_B))^2)

#create reduced model design matric and X1_B and estimator bhat1_B
X1_B <- X_B[,-c(3,5)]
bhat1_B <- ginv(t(X1_B)%*%X1_B) %*% t(X1_B) %*% Y_B
SSR_Br <- t(bhat1_B) %*% t(X1_B) %*% Y_B - (length(Y_B)*(mean(Y_B))^2)

SSE_B <- t(Y_B)%*%Y_B - t(bhat_B)%*%t(X_B)%*%Y_B

F_2f <- ((SSR_Bf - SSR_Br)/2)/(SSE_B/(length(Y_B) - qr(X_B)$rank))

pf_2f <- pf(F_2f, 2, (length(Y_B)-(qr(X_B)$rank)), lower.tail=F)
pf_2f
```