Notes for 751-752

Section 14

Martin Lindquist[*]

October 25, 2017

[*]Thanks to Brian Caffo and Ingo Ruczinski for supplying their notes, upon which much of this document is based.

14 Distributional results

In this chapter we assume that $\mathbf{Y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$. Under this assumption we will investigate some distributional results for the least-squares estimate $\hat{\boldsymbol{\beta}}$, and the residual variance estimate s^2.

14.1 Statistical properties

Note that $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ is a linear combination of normally distributed random variables, and hence itself normally distributed with moments:

$$E[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta} \quad \text{and} \quad \text{Var}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}'\mathbf{X})^{-1}\sigma^2.$$

The residual variance estimate is

$$s^2 = \frac{1}{n-p}\mathbf{e}'\mathbf{e},$$

which we previously showed was unbiased, i.e. $E[s^2] = \sigma^2$.

In addition, note that:

$$\frac{n-p}{\sigma^2}s^2 = \frac{1}{\sigma^2}\mathbf{y}'(\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{y}$$

is a quadratic form as discussed in the previous section. Furthermore,

$$\frac{1}{\sigma^2}\{\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\}\text{Var}(\mathbf{y}) = \{\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\},$$

which is idempotent. For symmetric idempotent matrices, the rank equals the trace; the latter of which is easily calculated as

$$\text{tr}\{\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\} = \text{tr}\{\mathbf{I}\} - \text{tr}\{\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\} = n - \text{tr}\{(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{X}\} = n - p.$$

Thus, $\frac{n-p}{\sigma^2}s^2$ is Chi-squared $n-p$. The special case of this where $\mathbf{X}$ has only an intercept yields the usual empirical variance estimate.

14.2 T statistics

We are now in a position to develop T statistics. Consider the linear contrast $\mathbf{c}'\hat{\boldsymbol{\beta}}$. First note that $\mathbf{c}'\hat{\boldsymbol{\beta}}$ is $N(\mathbf{c}'\boldsymbol{\beta}, \mathbf{c}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{c}\sigma^2)$. Furthermore,

$$\text{Cov}(\hat{\boldsymbol{\beta}}, \mathbf{e}) = \text{Cov}((\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}, \{\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\}\mathbf{Y}) = 0$$

since $(\mathbf{X}\mathbf{X})^{-1}\mathbf{X}'\{\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\} = \mathbf{0}$. Thus the residuals and estimated coefficients are independent, implying that $\mathbf{c}'\hat{\boldsymbol{\beta}}$ and s^2 are independent. Therefore,

$$\frac{\mathbf{c}'\hat{\boldsymbol{\beta}} - \mathbf{c}'\boldsymbol{\beta}}{\sqrt{\mathbf{c}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{c}\sigma^2}} / \sqrt{\frac{n-p}{\sigma^2} s^2/(n-p)} = \frac{\mathbf{c}'\hat{\boldsymbol{\beta}} - \mathbf{c}'\boldsymbol{\beta}}{\sqrt{\mathbf{c}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{c}s^2}}$$

is a standard normal divided by the square root of an independent Chi-squared over its degrees of freedom, thus is T distributed with $n - p$ degrees of freedom.

14.3 F statistic

Consider testing the hypothesis that $H_0 : \mathbf{K}\boldsymbol{\beta} = 0$ versus not equal for $\mathbf{K}$ of full row rank q. Notice that $\mathbf{K}\hat{\boldsymbol{\beta}} \sim N(\mathbf{K}\boldsymbol{\beta}, \mathbf{K}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{K}'\sigma^2)$ and thus

$$(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{K}\boldsymbol{\beta})'\{\mathbf{K}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{K}'\sigma^2\}^{-1}(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{K}\boldsymbol{\beta})$$

is Chi-squared with q degrees of freedom. Furthermore, it is independent of $\mathbf{e}$ being a function of $\hat{\boldsymbol{\beta}}$. Thus:

$$(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{K}\boldsymbol{\beta})'\{\mathbf{K}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{K}'\}^{-1}(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{K}\boldsymbol{\beta})/qs^2$$

forms the ratio of two independent Chi-squared random variables over their degrees of freedom, which follows an F distribution with q and $n - p$ degrees of freedom. .

14.4 Coding example

Consider the `swiss` dataset. Let's first make sure that we can replicate the coefficient table obtained by R.

```
> ## First let's see the coeficient table
> fit = lm(Fertility ~ ., data = swiss)
> round(summary(fit)$coef, 3)
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)        66.915     10.706   6.250    0.000
Agriculture        -0.172      0.070  -2.448    0.019
Examination        -0.258      0.254  -1.016    0.315
Education          -0.871      0.183  -4.758    0.000
Catholic            0.104      0.035   2.953    0.005
Infant.Mortality    1.077      0.382   2.822    0.007
> # Now let's do it more manually
> x = cbind(1, as.matrix(swiss[,-1]))
> y = swiss$Fertility
> beta = solve(t(x) %*% x, t(x) %*% y)
> e = y - x %*% beta
> n = nrow(x); p = ncol(x)
> s = sqrt(sum(e^2) / (n - p))
> #Compare with lm
> c(s, summary(fit)$sigma)
[1] 7.165369 7.165369
> #calculate the t statistics
> betaVar = solve(t(x) %*% x) * s ^ 2
> ## Show that standard errors agree with lm
> cbind(summary(fit)$coef[,2], sqrt(diag(betaVar)))
                        [,1]        [,2]
(Intercept)      10.70603759 10.70603759
Agriculture       0.07030392  0.07030392
Examination       0.25387820  0.25387820
Education         0.18302860  0.18302860
Catholic          0.03525785  0.03525785
Infant.Mortality  0.38171965  0.38171965
> # Show that the tstats agree
> tstat = beta / sqrt(diag(betaVar))
> cbind(summary(fit)$coef[,3],  tstat)
                      [,1]      [,2]
(Intercept)       6.250229  6.250229
Agriculture      -2.448142 -2.448142
Examination      -1.016268 -1.016268
Education        -4.758492 -4.758492
Catholic          2.952969  2.952969
Infant.Mortality  2.821568  2.821568
> # Show that the P-values agree
> cbind(summary(fit)$coef[,4],  2 * pt(- abs(tstat), n - p)
                         [,1]         [,2]
(Intercept)      1.906051e-07 1.906051e-07
Agriculture      1.872715e-02 1.872715e-02
Examination      3.154617e-01 3.154617e-01
Education        2.430605e-05 2.430605e-05
Catholic         5.190079e-03 5.190079e-03
Infant.Mortality 7.335715e-03 7.335715e-03
> # Get the F statistic
> # Set K to grab everything except the intercept
> k = cbind(0, diag(rep(1, p - 1)))
> kvar = k %*% solve(t(x) %*% x) %*% t(k)
> fstat = t(k %*% beta) %*% solve(kvar) %*% (k %*% beta) / (p - 1) / s ^ 2
> #Show that it's equal to what lm is giving
> cbind(summary(fit)$fstat, fstat)
> #Calculate the p-value
> pf(fstat, p - 1, n - p, lower.tail = FALSE)
             [,1]
[1,] 5.593799e-10
> summary(fit)
## ... only showing the one relevant line ...
F-statistic: 19.76 on 5 and 41 DF,  p-value: 5.594e-10
```

14.5 Confidence ellipsoids

An hyper-ellipsoid with center $\mathbf{v}$ is defined as the solutions in $\mathbf{x}$ of $(\mathbf{x}-\mathbf{v})'\mathbf{A}(\mathbf{x}-\mathbf{v}) = 1$. The eigenvalues of $\mathbf{A}$ determine the length of the axes of the ellipsoid. The set of points $\{\mathbf{x} \mid (\mathbf{x}-\mathbf{v})'\mathbf{A}(\mathbf{x}-\mathbf{v}) \leq 1\}$ lie in the interior of the hyper-ellipsoid.

Now consider our F statistic from earlier on in the chapter:

$$(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})'\{\mathbf{K}(\mathbf{X}\mathbf{X})^{-1}\mathbf{K}'\}^{-1}(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})/vS^2$$

We would fail to reject $H_0 : \mathbf{K}\boldsymbol{\beta} = \mathbf{m}$ is less than the appropriate cut off from an F distribution, say $F_{1-\alpha}$. So, the set of points

$$\{\mathbf{m} \mid (\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})'\{\mathbf{K}(\mathbf{X}\mathbf{X})^{-1}\mathbf{K}'\}^{-1}(\mathbf{K}\hat{\boldsymbol{\beta}} - \mathbf{m})/vS^2F_{1-\alpha} \leq 1\}$$

forms a confidence set. From the discussion above, we see that this is a hyper ellipsoid. This multivariate form of confidence interval is called a confidence ellipse. These are of course most useful when the dimension is 2 or 3 so that we can visualize it as an actual ellipse.

14.5.1 Coding example

```
fit = lm(mpg ~ disp + hp , mtcars)
open3d()
plot3d(ellipse3d(fit), col = "red", alpha = .5, aspect = TRUE)

## Doing it directly
beta = coef(fit)
Sigma = vcov(fit)
n = nrow(mtcars); p = length(beta)
```

```
A = Sigma * (3 * qf(.95, 3, n - p))
nms = names(beta)

open3d()
## Using the definition of an elipse
##(x - b)' A (x - b) = 1
plot3d(ellipse3d(A, centre = beta, t = 1),
                 color = "blue",  alpha = .5, aspect = TRUE,
       xlab = nms[1], ylab = nms[2], zlab = nms[3])

## Using the more statistical version
## Provide ellipse3d with the variance covariance matrix
plot3d(ellipse3d(Sigma, centre = beta, t = sqrt(3 * qf(.95, 3, n - p))),
                 color = "green",  alpha = .5, aspect = TRUE,
       xlab = nms[1], ylab = nms[2], zlab = nms[3], add = TRUE)
```

14.6 Confidence intervals

Consider creating a confidence interval for $\mathbf{c}'\boldsymbol{\beta}$. Begin by recalling that

$$\frac{\mathbf{c}'\hat{\boldsymbol{\beta}} - \mathbf{c}'\boldsymbol{\beta}}{s\sqrt{\mathbf{c}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{c}}} \sim t_{n-p}.$$

Thus, we can show that a $100(1-\alpha)\%$ confidence interval for $\mathbf{c}'\boldsymbol{\beta}$ is given by:

$$\mathbf{c}'\hat{\boldsymbol{\beta}} \pm t_{\alpha/2,n-p} s\sqrt{\mathbf{c}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{c}}$$

14.7 Prediction intervals

It's worth discussing prediction intervals. The obvious prediction at a new set of covariates, $\mathbf{x}_0$, is $\mathbf{x}_0'\hat{\boldsymbol{\beta}}$. This is then just a linear contrast of the $\boldsymbol{\beta}$ and so the interval would be

$$\mathbf{x}_0'\hat{\boldsymbol{\beta}} \pm t_{n-p,1-\alpha/2} s\sqrt{\mathbf{x}_0'(\mathbf{X}\mathbf{X})^{-1}\mathbf{x}_0}.$$

If you've taken an introductory regression class, you it will have noted the difference between a prediction interval and a confidence interval for a mean at a new value of $\mathbf{x}$. For a prediction interval, we want to estimate a range of possible values for y at that value of $\mathbf{x}$, a different statement than trying to estimate the average value of y at that value of $\mathbf{x}$. As we collect infinite data, we should get the average value exactly. However, predicting a new value involves intrinsic variability that can't go away no matter how much data we use to build our model.

As an example, consider the following two tasks: (i) guessing the sales price of a diamond given its weight; and (ii) guessing the average sales price of diamonds given a particular weight. With enough data, we should be able to estimate the average sale price very precisely. However, we still won't know the exact sales price of an individual diamond of that weight.

To account for this, we develop prediction intervals. These are not confidence intervals, because they are trying to estimate something random, not a fixed parameter. Consider estimating Y_0 at $\mathbf{x}$ value $\mathbf{x}_0$. Note that

$$Var(y_0 - \mathbf{x}_0'\hat{\boldsymbol{\beta}}) = (1 + \mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_0)\sigma^2$$

For homework, show that

$$\frac{y_0 - \mathbf{x}_0'\hat{\boldsymbol{\beta}}}{s\sqrt{1 + \mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_0\sigma^2}}$$

follows a T distribution with $n - p$ degrees of freedom. Finish, by showing that

$$P\{y_0 \in [\mathbf{x}_0'\hat{\boldsymbol{\beta}} \pm t_{n-p,1-\alpha/2}s\sqrt{1 + \mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_0}]\} = 1 - \alpha.$$

This is called a prediction interval. Notice the variability under consideration contains $\mathbf{x}_0'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_0$, which goes to 0 as we get more $\mathbf{X}$ variability and 1, which represents the intrinsic part of the variability that doesn't go away.

14.8 Coding example

Let's try to predict a car's MPG from other characteristics.

```
> fit = lm(mpg ~ hp + wt, data = mtcars)
> newcar = data.frame(hp = 90, wt = 2.2)
> predict(fit, newdata = newcar)
       1
25.83648
> predict(fit, newdata = newcar, interval = "confidence")
       fit      lwr      upr
1 25.83648 24.46083 27.21212
> predict(fit, newdata = newcar, interval = "prediction")
       fit      lwr      upr
1 25.83648 20.35687 31.31609
```

```
> #Doing it manually
> library(dplyr)
> y = mtcars$mpg
> x = as.matrix(cbind(1, select(mtcars, hp, wt)))
> n = length(y)
> p = ncol(x)
> xtxinv = solve(t(x) %*% x)
> beta = xtxinv %*% t(x) %*% y
> x0 = c(1, 90, 2.2)
> yhat = x %*% beta
> e = y - yhat
> s = sqrt(sum(e^2 / (n - p)))
> yhat0 = sum(x0 * beta)
> # confidence interval
> yhat0 + qt(c(0.025, .975), n - p) * s * sqrt(t(x0) %*% xtxinv %*% x0)
[1] 24.46083 27.21212
> # prediction interval
> yhat0 + qt(c(0.025, .975), n - p) * s * sqrt(1 + t(x0) %*% xtxinv %*% x0)
[1] 20.35687 31.31609
```

14.9 Confidence interval for the variance

We can use this result to develop a confidence interval for the variance. Let $\chi^2_{n-p,\alpha}$ be the α quantile from the chi squared distribution with $n-p$ degrees of freedom. Then inverting the probability statement

$$1-\alpha = P\left(\chi^2_{n-p,\alpha/2} \leq \frac{\mathbf{e}'\mathbf{e}}{n-p} \leq \chi^2_{n-p,1-\alpha/2}\right)$$

yields the $100(1-\alpha)\%$ confidence interval

$$\frac{(n-p)s^2}{\chi^2_{n-p,1-\alpha/2}} \leq \sigma^2 \leq \frac{(n-p)s^2}{\chi^2_{n-p,\alpha/2}}$$