Lecture 15: Multiple linear regression

Pratheepa Jeganathan

10/25/2019

Recap

- What is a regression model?
- Descriptive statistics – graphical
- Descriptive statistics – numerical
- Inference about a population mean
- Difference between two population means
- Some tips on R
- Simple linear regression (covariance, correlation, estimation, geometry of least squares)
  - Inference on simple linear regression model
  - Goodness of fit of regression: analysis of variance.
  - F-statistics.
  - Residuals.
  - Diagnostic plots for simple linear regression (graphical methods).

Recap

- Multiple linear regression
  - Specifying the model.
  - Fitting the model: least squares.
  - Interpretation of the coefficients.
  - Inference for multiple regression
    - T-statistics revisited.

Outline

- Inference for multiple regression
  - More F statistics.
  - Tests involving more than one β.

Inference for multiple regression

Questions about many (combinations) of β_j's

- In multiple regression we can ask more complicated questions than in simple regression.
- For instance, we could ask whether `lcp` and `pgg45` explains little of the variability in the data, and might be dropped from the regression model.
- These questions can be answered by F-statistics.
- **Note: This hypothesis should really be formed *before* looking at the output of `summary`.**
- Later we'll see some examples of the messiness when forming a hypothesis after seeing the `summary`.

Dropping one or more variables

- Suppose we wanted to test the above hypothesis.
- Formally, the null hypothesis is:

$$H_0 : \beta_{\texttt{lcp}}(= \beta_6) = \beta_{\texttt{pgg45}}(= \beta_7) = 0$$

and the alternative is

$$H_a = \text{one of } \beta_{\texttt{lcp}}, \beta_{\texttt{pgg45}} \text{ is not 0.}$$

- This test is an F-test based on two models

$$\text{Reduced}(\text{H}_0) : Y_i = \beta_0 + \sum_{j=1}^{5} \beta_j X_{ij} + \varepsilon_i$$

$$\text{Full}(\text{H}_a) : Y_i = \beta_0 + \sum_{j=1}^{7} X_{ij} \beta_j + \varepsilon_i$$

- **Note: The reduced model R must be a special case of the full model F to use the F-test.**

SSE of a model

- A "model", $\mathcal{M}$ is a subspace of $\mathbb{R}^n$ (e.g. column space of X).
- Least squares fit $=$ projection onto the subspace of $\mathcal{M}$, yielding predicted values $\widehat{Y}_{\mathcal{M}}$
- Error sum of squares:

$$SSE(\mathcal{M}) = \|Y - \widehat{Y}_{\mathcal{M}}\|^2.$$

F-statistic for $H_0 : \beta_{\texttt{lcp}} = \beta_{\texttt{pgg45}} = 0$

- We compute the F statistic the same to compare any models

$$F = \frac{\frac{SSE(R) - SSE(F)}{2}}{\frac{SSE(F)}{n-1-p}} \sim F_{2,n-p-1} \qquad \text{(if } H_0 \text{ is true)}$$

- Reject H_0 at level α if $F \geq F_{1-\alpha,2,n-1-p}$.
- When comparing two models, one a special case of the other (i.e. one nested in the other), we can test if the smaller model (the special case) is roughly as good as the larger model in describing our outcome. This is typically tested using an F test based on comparing the two models. The following function does this.

```
f.test.lm = function(R.lm, F.lm) {
    SSE.R = sum(resid(R.lm)^2)
    SSE.F = sum(resid(F.lm)^2)
    df.num = R.lm$df - F.lm$df
    df.den = F.lm$df
    F = ((SSE.R - SSE.F) / df.num) / (SSE.F / df.den)
    p.value = 1 - pf(F,
      df.num, df.den)
    return(data.frame(F,
      df.num, df.den, p.value))
}
```

- R has a function that does essentially the same thing as `f.test.lm`: the function is `anova`. It can be used several ways, but it can be used to compare two models.

```
prostate.lm.reduced = lm(lpsa ~ lcavol +
    lbph + lweight + age + svi,
  data=prostate)
print(f.test.lm(prostate.lm.reduced, prostate.lm))
```

```
##          F df.num df.den   p.value
## 1 1.372057      2     89 0.2588958
```

```
anova(prostate.lm.reduced, prostate.lm)
```

```
## Analysis of Variance Table
##
## Model 1: lpsa ~ lcavol + lbph + lweight + age + svi
## Model 2: lpsa ~ lcavol + lweight + age + lbph + svi + lc
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     91 44.437
## 2     89 43.108  2    1.3291 1.3721 0.2589
```

Dropping an arbitrary subset

- For an arbitrary model, suppose we want to test

$$H_0 : \beta_{i_1} = \cdots = \beta_{i_j} = 0$$
$$H_a : \text{one or more of } \beta_{i_1}, \ldots \beta_{i_j} \neq 0$$

for some subset $\{i_1, \ldots, i_j\} \subset \{0, \ldots, p\}$.

- You guessed it: it is based on the two models:

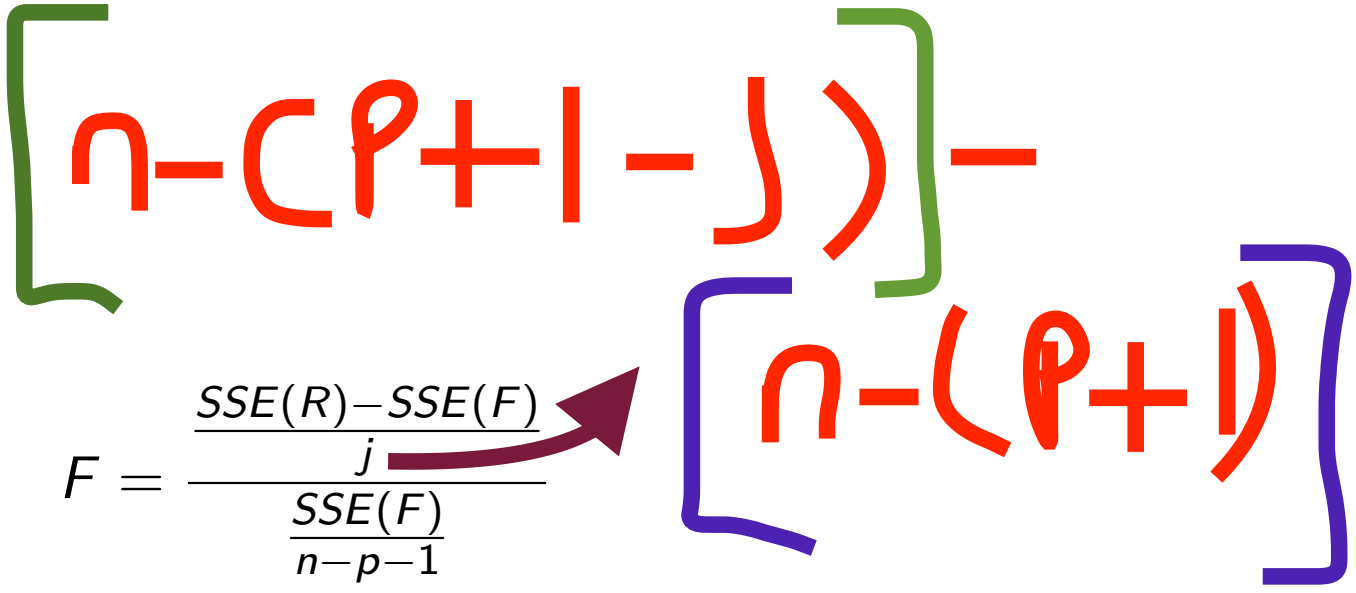
$$R : Y_i = \sum_{l=0, l \notin \{i_1, \ldots, i_j\}}^{p} \beta_l X_{il} + \varepsilon_i$$

$$F : Y_i = \sum_{l=0}^{p} \beta_l X_{il} + \varepsilon_i$$

where $X_{i0} = 1$ for all i.

- **Note: This hypothesis should really be formed *before* looking at the output of `summary`. Looking at `summary` before deciding which to drop is problematic!**

Dropping an arbitrary subset



- Statistic:

$$F = \frac{\frac{SSE(R)-SSE(F)}{j}}{\frac{SSE(F)}{n-p-1}}$$

$$\sim F_{j,n-p-1} \qquad \text{(if } H_0 \text{ is true)}$$

- Reject H_0 at level α if $F \geq F_{1-\alpha,j,n-1-p}$.

General F-tests

- Given two models $R \subset F$ (i.e. R is a subspace of F), we can consider testing

$$H_0 : R \text{ is adequate (i.e. } \mathbb{E}(Y) \in R)$$

vs.

$$H_a : F \text{ is adequate (i.e. } \mathbb{E}(Y) \in F)$$

- The test statistic is

$$F = \frac{(SSE(R) - SSE(F))/(df_R - df_F)}{SSE(F)/df_F}.$$

- If H_0 is true, $F \sim F_{df_R - df_F, df_F}$ so we reject H_0 at level α if $F \geq F_{1-\alpha, df_R - df_F, df_F}$.

Constraining $\beta_{\texttt{lcavol}} = \beta_{\texttt{svi}}$

- In this example, we might suppose that the coefficients for `lcavol` and `svi` are the same and want to test this. We do this, again, by comparing a "full model" and a "reduced model".
- Full model:

$$Y_i = \beta_0 + \beta_1 X_{i,\texttt{lcavol}} + \beta_2 X_{i,\texttt{lweight}} + \beta_3 X_{i,\texttt{age}} + \beta_4 X_{i,\texttt{lbph}} + \beta_5 X_{i,\texttt{svi}} + \beta_6 X_{i,\texttt{lcp}} + \beta_7 X_{i,\texttt{pgg45}} + \varepsilon_i$$

- Reduced model:

$$Y_i = \beta_0 + \tilde{\beta}_1 X_{i,\texttt{lcavol}} + \beta_2 X_{i,\texttt{lweight}} + \beta_3 X_{i,\texttt{age}} + \beta_4 X_{i,\texttt{lbph}} + \tilde{\beta}_1 X_{i,\texttt{svi}} + \beta_6 X_{i,\texttt{lcp}} + \beta_7 X_{i,\texttt{pgg45}} + \varepsilon_i$$

Example

```
prostate$Z = prostate$lcavol + prostate$svi
equal.lm = lm(Y ~ Z + lweight + age +
    lbph + lcp + pgg45, data = prostate)
f.test.lm(equal.lm, prostate.lm)
```

```
##           F df.num df.den   p.value
## 1 0.4818657      1     89 0.4893864
```

Constraining $\beta_{\texttt{lcavol}} + \beta_{\texttt{svi}} = 1$

- Full model:

$$Y_i = \beta_0 + \beta_1 X_{i,\texttt{lcavol}} + \beta_2 X_{i,\texttt{lweight}} + \beta_3 X_{i,\texttt{age}} + \beta_4 X_{i,\texttt{lbph}} + \beta_5 X_{i,\texttt{svi}} + \beta_6 X_{i,\texttt{lcp}} + \beta_7 X_{i,\texttt{pgg45}} + \varepsilon_i$$

- Reduced model:

$$Y_i = \beta_0 + \tilde{\beta}_1 X_{i,\texttt{lcavol}} + \beta_2 X_{i,\texttt{lweight}} + \beta_3 X_{i,\texttt{age}} + \beta_4 X_{i,\texttt{lbph}} + (1 - \tilde{\beta}_1) X_{i,\texttt{svi}} + \beta_6 X_{i,\texttt{lcp}} + \beta_7 X_{i,\texttt{pgg45}} + \varepsilon_i$$

Example

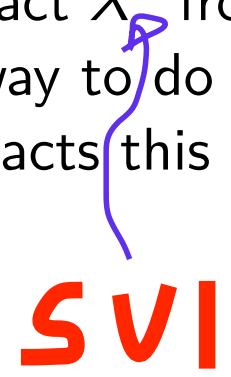
```
prostate$Z2 = prostate$lcavol - prostate$svi
constrained.lm = lm(lpsa ~ Z2 + lweight + age +
    lbph + lcp + pgg45,
  data=prostate, offset=svi)
anova(constrained.lm, prostate.lm)
```

```
## Analysis of Variance Table
##
## Model 1: lpsa ~ Z2 + lweight + age + lbph + lcp + pgg45
## Model 2: lpsa ~ lcavol + lweight + age + lbph + svi + lc
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     90 43.961
## 2     89 43.108  1   0.85359 1.7623 0.1877
```

```
f.test.lm(constrained.lm, prostate.lm)
```

```
##          F df.num df.den   p.value
## 1 1.762322      1     89 0.1877303
```

- What we had to do above was subtract X_{-} from Y on the right hand side of the formula. R has a way to do this called using an *offset*. What this does is it subtracts this vector from Y before fitting.



General linear hypothesis

- An alternative version of the F test can be derived that does not require refitting a model.
- Suppose we want to test

$$H_0 : C_{q\times(p+1)}\beta_{(p+1)\times 1} = h$$

versus

$$H_a : C_{q\times(p+1)}\beta_{(p+1)\times 1} \neq h.$$

- This can be tested via an F test:

$$F = \frac{(C\hat{\beta} - h)^T \left(C(X^TX)^{-1}C^T\right)^{-1} (C\hat{\beta} - h)/q}{SSE(F)/df_F} \overset{H_0}{\sim} F_{q,n-p-1}.$$

Note: we are assuming that rank$(C(X^TX)^{-1}C^T) = q$**.**

- Here's a function that implements this computation.

```
general.linear = function(model.lm,
  linear_part, null_value=0) {
    # shorthand
    C = linear_part
    h = null_value

    beta.hat = coef(model.lm)
    V = as.numeric(C %*% beta.hat - null_value)
    # the MSE is included in vcov
    invcovV = solve(C %*% vcov(model.lm) %*% t(C))

    df.num = nrow(C)
    df.den = model.lm$df.resid
    F = t(V) %*% invcovV %*% V / df.num
    p.value = 1 - pf(F, df.num, df.den)
    return(data.frame(F, df.num,
      df.den, p.value))
}
```

Example (General linear hypothesis)

- Let's test verify this work with our test for testing $\beta_{\texttt{lcp}} = \beta_{\texttt{pgg45}} = 0$.

```
A = matrix(0, nrow = 2, ncol = 8)
A[1,7] = 1
A[2,8] = 1
print(A)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
## [1,]    0    0    0    0    0    0    1    0
## [2,]    0    0    0    0    0    0    0    1
```

- Rank of **A**

```
qr(A)$rank
```

```
## [1] 2
```

Example (General linear hypothesis)

```
general.linear(prostate.lm, A)
```

```
##           F df.num df.den   p.value
## 1 1.372057      2     89 0.2588958
```

```
f.test.lm(prostate.lm.reduced, prostate.lm)
```

```
##           F df.num df.den   p.value
## 1 1.372057      2     89 0.2588958
```

References

- **CH** Chapter 3.10, 3.12
- Lecture notes of Jonathan Taylor .