Lecture 13: Multiple linear regression

Pratheepa Jeganathan

10/21/2019

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

Recap

- 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.

Outline

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

Inference for multiple regression

Regression function at one point

- One thing one might want to *learn* about the regression function in the prostate example is something about the regression function at some fixed values of $X_1, \ldots, X_7$, i.e. what can be said about the mean response

$$\beta_0 + 1.3 \cdot \beta_1 + 3.6 \cdot \beta_2 + 64 \cdot \beta_3 + \\ 0.1 \cdot \beta_4 + 0.2 \cdot \beta_5 - 0.2 \cdot \beta_6 + 25 \cdot \beta_7$$

  roughly the regression function at "typical" values of the predictors.

- The expression above is equivalent to

$$\sum_{j=0}^{7} a_j \beta_j = \boldsymbol{a}^T \boldsymbol{\beta}, \qquad \boldsymbol{a} = (1, 1.3, 3.6, 64, 0.1, 0.2, -0.2, 25).$$

Confidence interval for $\sum_{j=0}^{p} a_j \beta_j$

- Suppose we want a $(1-\alpha)\cdot 100\%$ CI for $\sum_{j=0}^{p} a_j \beta_j$.
- Just as in simple linear regression:

$$\sum_{j=0}^{p} a_j \widehat{\beta}_j \pm t_{1-\alpha/2,n-p-1} \cdot SE\left(\sum_{j=0}^{p} a_j \widehat{\beta}_j\right).$$

Standard error of $\sum_{j=0}^{p} a_j\widehat{\beta}_j$

- In order to form these confidence interval, we need the *SE* of our estimate $\sum_{j=0}^{p} a_j\widehat{\beta}_j$.
- Based on matrix approach to regression

$$\text{SE}\left(\sum_{j=0}^{p} a_j\widehat{\beta}_j\right) = \text{SE}\left(\boldsymbol{a}^T\widehat{\beta}\right) = \sqrt{\text{Cov}\left(\boldsymbol{a}^T\widehat{\beta}\right)} = \sqrt{\boldsymbol{a}^T\text{Cov}\left(\widehat{\beta}\right)\boldsymbol{a}}$$
$$= \sqrt{\widehat{\sigma}^2\boldsymbol{a}^T(X^TX)^{-1}\boldsymbol{a}}$$

. - Don't worry too much about specific implementation – for much of the effects we want `R` will do this for you in general.

Example

```r
library(xtable)
library(ElemStatLearn)
data(prostate)
prostate.lm = lm(lpsa ~ lcavol + lweight +
    age + lbph + svi + lcp + pgg45,
  data = prostate)
n = nrow(prostate)
Y = prostate$lpsa
X = model.matrix(prostate.lm)
beta_hat = as.numeric(solve(t(X) %*% X)
  %*% t(X) %*% Y)
names(beta_hat) = colnames(X)
```

```
Y.hat = X %*% beta_hat
sigma.hat = sqrt(sum((Y - Y.hat)^2)
  / (n - ncol(X)))
cov.beta_hat = sigma.hat^2 * solve(t(X) %*% X)
```

```r
print(xtable(data.frame(cov.beta_hat), digits = 4),
scalebox='0.6')
```

% latex table generated in R 3.6.0 by xtable 1.8-4 package % Mon Oct 21 01:17:35 2019

	X.Intercept.	lcavol	lweight	age	lbph	svi	lcp	pgg45
(Intercept)	0.7631	0.0100	-0.1127	-0.0058	0.0248	0.0103	0.0021	0.0000
lcavol	0.0100	0.0074	-0.0033	-0.0001	0.0004	-0.0026	-0.0035	0.0000
lweight	-0.1127	-0.0033	0.0394	-0.0004	-0.0045	-0.0041	0.0001	0.0001
age	-0.0058	-0.0001	-0.0004	0.0001	-0.0001	-0.0001	0.0001	-0.0000
lbph	0.0248	0.0004	-0.0045	-0.0001	0.0033	0.0021	-0.0001	-0.0000
svi	0.0103	-0.0026	-0.0041	-0.0001	0.0021	0.0567	-0.0089	-0.0001
lcp	0.0021	-0.0035	0.0001	0.0001	-0.0001	-0.0089	0.0080	-0.0001
pgg45	0.0000	0.0000	0.0001	-0.0000	-0.0000	-0.0001	-0.0001	0.0000

- The standard error of regression function estimate at $\boldsymbol{a} = (1, 1.3, 3.6, 64, 0.1, 0.2, -0.2, 25)$ is $\sqrt{\boldsymbol{a}^T \mathrm{Cov}\left(\widehat{\beta}\right) \boldsymbol{a}}$

```r
a = c(1,1.3,3.6,64,0.1,0.2,-0.2,25)
sqrt(t(a)%*%cov.beta_hat%*%a)
```

```
##              [,1]
## [1,] 0.07101959
```

- The standard errors of each coefficient estimate are the square root of the diagonal entries.

```r
round(sqrt(diag(cov.beta_hat)), digits = 4)
```

```
## (Intercept)      lcavol     lweight         age           l
##      0.8736      0.0858      0.1984      0.0110         0.0
##         lcp       pgg45
##      0.0893      0.0034
```

- Generally, we can find our estimate of the covariance function $\text{Cov}\left(\hat{\beta}\right)$ as follows:

```
print(xtable(vcov(prostate.lm), digits = 4),
  scalebox='0.6')
```

% latex table generated in R 3.6.0 by xtable 1.8-4 package % Mon Oct 21 11:03:21 2019

	(Intercept)	lcavol	lweight	age	lbph	svi	lcp	pgg45
(Intercept)	0.7631	0.0100	-0.1127	-0.0058	0.0248	0.0103	0.0021	0.0000
lcavol	0.0100	0.0074	-0.0033	-0.0001	0.0004	-0.0026	-0.0035	0.0000
lweight	-0.1127	-0.0033	0.0394	-0.0004	-0.0045	-0.0041	0.0001	0.0001
age	-0.0058	-0.0001	-0.0004	0.0001	-0.0001	-0.0001	0.0001	-0.0000
lbph	0.0248	0.0004	-0.0045	-0.0001	0.0033	0.0021	-0.0001	-0.0000
svi	0.0103	-0.0026	-0.0041	-0.0001	0.0021	0.0567	-0.0089	-0.0001
lcp	0.0021	-0.0035	0.0001	0.0001	-0.0001	-0.0089	0.0080	-0.0001
pgg45	0.0000	0.0000	0.0001	-0.0000	-0.0000	-0.0001	-0.0001	0.0000

Example (confidence interval for regression at a given point)

```r
library(ElemStatLearn)
data(prostate)
prostate.lm = lm(lpsa ~ lcavol + lweight +
    age + lbph + svi + lcp + pgg45,
  data = prostate)
```

Example (confidence interval for regression at a given point)

- R will form these coefficients ($\boldsymbol{a}$) for each regression coefficient separately when using the `confint` function.
- If we have an observation, $X_1 = 1.3, X_2 = 3.6, X_3 = 64$, $X_4 = 0.1, X_5 = 0.2, X_6 = -0.2, X_7 = 25$.
- We can write $\boldsymbol{a} = (1.3, 3.6, 64, 0.1, 0.2, -0.2, 25)$.
-

```
predict(prostate.lm, list(lcavol = 1.3, lweight = 3.6,
  age = 64, lbph = 0.1,
  svi = 0.2, lcp = -.2, pgg45 = 25),
  interval='confidence',
  level=0.90)
```

```
##        fit      lwr      upr
## 1 2.422332 2.304287 2.540378
```

Confidence interval for individual regression coefficients

- If we want a confidence interval for β_1. We can write $\boldsymbol{a}$ as follows

$$\boldsymbol{a}_{\texttt{lcavol}} = (0, 1, 0, 0, 0, 0, 0, 0)^T$$

so that

$$\boldsymbol{a}_{\texttt{lcavol}}^T \boldsymbol{\beta} = \boldsymbol{\beta}_1$$

and

$$\boldsymbol{a}_{\texttt{lcavol}}^T \widehat{\beta} = \widehat{\beta}_1 = \texttt{coef(prostate.lm)}[2]$$

Confidence interval for regression coefficient

- Suppose we want a $(1-\alpha) \cdot 100\%$ CI for β_1.
- Just as in simple linear regression:

$$\begin{aligned} \boldsymbol{a}_{\texttt{lcavol}}^T \widehat{\boldsymbol{\beta}} \pm t_{1-\alpha/2,n-p-1} \cdot SE\left(\boldsymbol{a}_{\texttt{lcavol}}^T \widehat{\boldsymbol{\beta}}\right) \\ \widehat{\beta}_1 \pm t_{1-\alpha/2,n-p-1} \cdot SE\left(\widehat{\beta}_1\right). \end{aligned}$$

Example (confidence interval for regression coefficient)

```
confint(prostate.lm, level=0.90)
```

```
##                       5 %         95 %
## (Intercept) -0.9578488958  1.946158404
## lcavol       0.4268548240  0.712237239
## lweight      0.2845659251  0.944273708
## age         -0.0391601782 -0.002666755
## lbph         0.0016386253  0.193066445
## svi          0.3565053323  1.148289353
## lcp         -0.2534678904  0.043549074
## pgg45       -0.0003011464  0.010950077
```

- Confidence interval for β_1:

```
confint(prostate.lm, c("lcavol"), level=0.90)
```

```
##              5 %      95 %
## lcavol 0.4268548 0.7122372
```

Bonferroni correction (confidence interval for regression coefficient)

- Bonferroni correction is a multiple-comparison correction used when several dependent or independent statistical tests are being performed simultaneously

```r
confint(prostate.lm, c("lcavol",
  "lweight", "age", "lbph", "svi",
  "lcp", "pgg45"),
  level= 1-.1/7)
```

```
##                0.714 %    99.286 %
## lcavol    0.355005433 0.784086630
## lweight   0.118474394 1.110365239
## age      -0.048347956 0.006521023
## lbph     -0.046556258 0.241261328
## svi       0.157161587 1.347633098
## lcp      -0.328246457 0.118327641
## pgg45    -0.003133814 0.013782745
```

T-statistics revisited

- Of course, these confidence intervals are based on the standard ingredients of a T-statistic.
- Suppose we want to test

$$H_0 : \sum_{j=0}^{p} a_j \beta_j = h.$$

- As in simple linear regression, it is based on

$$T = \frac{\sum_{j=0}^{p} a_j \widehat{\beta}_j - h}{SE(\sum_{j=0}^{p} a_j \widehat{\beta}_j)}.$$

- If H_0 is true, then $T \sim t_{n-p-1}$, so we reject H_0 at level α if

$$|T| \geq t_{1-\alpha/2, n-p-1}, \qquad \text{OR}$$
$$\text{p-value} = 2 * (1 - \texttt{pt}(|\texttt{T}|, \texttt{n} - \texttt{p} - 1)) \leq \alpha.$$

Example (T-statistic)

- `R` produces these in the `coef` table `summary` of the linear regression model. Again, each of these linear combinations is a vector $\boldsymbol{a}$ with only one non-zero entry like $\boldsymbol{a}_{\texttt{lcavol}}$ above.

```
print(xtable(summary(prostate.lm)$coef,
  digits = 3), scalebox='0.6')
```

% latex table generated in R 3.6.0 by xtable 1.8-4 package % Wed Oct 23 11:19:57 2019

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.494	0.874	0.566	0.573
lcavol	0.570	0.086	6.634	0.000
lweight	0.614	0.198	3.096	0.003
age	-0.021	0.011	-1.905	0.060
lbph	0.097	0.058	1.691	0.094
svi	0.752	0.238	3.159	0.002
lcp	-0.105	0.089	-1.175	0.243
pgg45	0.005	0.003	1.573	0.119

Example (T-statistic)

- Let's do a quick calculation to remind ourselves the relationships of the columns in the table above.

```
T1 = 0.570 / 0.086
P1 = 2 * (1 - pt(abs(T1), 89))
print(round(c(T1, P1), digits = 3))
```

```
## [1] 6.628 0.000
```

- These were indeed the values for `lcavol` in the `summary` table.

One-sided tests

- Suppose, instead, we wanted to test the one-sided hypothesis

$$H_0 : \sum_{j=0}^{p} a_j\beta_j \leq h, \text{ vs. } H_a : \sum_{j=0}^{p} a_j\beta_j > h$$

- We reject H_0 at level α if

$$\begin{gathered} T \geq t_{1-\alpha,n-p-1}, \qquad \text{OR} \\ p-\text{value} = (1 - \mathtt{pt}(\mathtt{T}, \mathtt{n}-\mathtt{p}-1)) \leq \alpha. \end{gathered}$$

- **Note: the decision to do a one-sided T test should be made *before* looking at the T statistic. Otherwise, the probability of a type I error is doubled!**

Prediction interval

- Basically identical to simple linear regression.
- Prediction interval at $X_{1,new}, \dots, X_{p,new}$:

$$\widehat{\beta}_0 + \sum_{j=1}^{p} X_{j,new}\widehat{\beta}_j \pm t_{1-\alpha/2,n-p-1}\sqrt{\widehat{\sigma}^2 + SE\left(\widehat{\beta}_0 + \sum_{j=1}^{p} X_{j,new}\widehat{\beta}_j\right)^2}.$$

- If we take $\boldsymbol{a} = \left(1, X_{1,new}, \cdots, X_{p,new}\right)^T$,
- $(1-\alpha)\,100\%$ prediction interval for the response is

$$\boldsymbol{a}^T\hat{\beta} \pm t_{1-\alpha/2,n-p-1}\sqrt{\widehat{\sigma}^2 + \boldsymbol{a}^T\mathrm{Cov}\left(\hat{\beta}\right)\boldsymbol{a}}.$$

Forming intervals by hand

- While `R` computes most of the intervals we need, we could write a function that explicitly computes a confidence interval (and can be used for prediction intervals with the "extra" argument).
- This exercise shows the calculations that R is doing under the hood: the function *predict* is generally going to be fine for our purposes.

```
interaval.lm = function(cur.lm, a, level=0.95, extra=0) {
     # the center of the confidence interval
     center = sum(a*cur.lm$coef)
     # the estimate of sigma^2
     sigma.hat.sq = sum(resid(cur.lm)^2) /
       cur.lm$df.resid
     # the standard error of sum(a*cur.lm$coef)
     se = sqrt(extra * sigma.hat.sq +
         sum((a %*% vcov(cur.lm)) * a))
     # the degrees of freedom for the t-statistic
     df = cur.lm$df
     # the quantile used in the confidence interval
     q = qt((1 - level)/2, df,
       lower.tail=FALSE)
     # upper, lower limits
     upper = center + se * q
     lower = center - se * q
     return(data.frame(center,
       lower, upper))
}
```

Example (prediction intervals)

- By using the extra $= 1$ argument, we can make prediction intervals.

```
print(interaval.lm(prostate.lm, c(1, 1.3, 3.6,
  64, 0.1, 0.2, -0.2, 25),
  extra=1))
```

```
##     center    lower    upper
## 1 2.422332 1.032301 3.812363
```

```
predict(prostate.lm,list(lcavol=1.3,
  lweight = 3.6,age = 64,lbph = 0.1,
  svi = 0.2,lcp = -0.2, pgg45 = 25),
  interval='prediction')
```

```
##        fit      lwr      upr
## 1 2.422332 1.032301 3.812363
```

Example (confidence interval for mean response)

```r
print(interaval.lm(prostate.lm,
  c(1, 1.3, 3.6, 64, 0.1,
    0.2, -0.2, 25), extra = 0))
```

```
##     center    lower    upper
## 1 2.422332 2.281218 2.563447
```

```r
predict(prostate.lm, list(lcavol = 1.3,
  lweight = 3.6,
  age = 64, lbph = 0.1,svi = 0.2,
  lcp = -0.2, pgg45 = 25),
  interval='confidence')
```

```
##        fit      lwr      upr
## 1 2.422332 2.281218 2.563447
```

Arbitrary contrasts

- If we want, we can set the intercept term to 0. This allows us to construct confidence interval for, say, how much the `lpsa` score will change will increase if we change `age` by 2 years and `svi` by 0.5 units, leaving everything else unchanged.
- Therefore, what we want is a confidence interval for 2 times the coefficient of `age` + 0.5 times the coefficient of `lbph`:

$$2 \cdot \beta_{\texttt{age}} + 0.5 \cdot \beta_{\texttt{svi}}$$

- Most of the time, *predict* will do what you want so this won't be used too often.

Example (Arbitrary contrasts)

```r
print(interaval.lm(prostate.lm,
  c(0,0,0,2,0,0.5,0,0),  extra = 0))
```

```
##        center      lower     upper
## 1 0.3343717 0.09496226 0.5737812
```

References

- **CH** Chapter 3.9, 3.11.
- Lecture notes of Jonathan Taylor .