Lesson 9 R Activity

Rick Davila

5/13/2020

Lesson 9 - Install packages

Perform data housekeeping - upload, name columns, display to make sure it reads properly, etc.

```
knitr::opts_chunk$set(echo = TRUE)

library(e1071)
library(xtable)
library("xlsx") # Needed to read data
```

```
## Warning: package 'xlsx' was built under R version 4.0.3
```

```
library(MASS) # Needed for ginv() function

rm(list = ls())
```

Upload data-ex-3-1.xlsx data file and label columns

```
exL9 <- read.xlsx("data-ex-3-1.xlsx",
                  sheetIndex = 1,
                  colIndex = c(2,3,4),
                  as.data.frame = TRUE,
                  header = TRUE)

# Assign labels to data columns using names() and attach() commands
names(exL9) <- c("time","cases","distance")
attach(exL9)

# Output data to make sure it reads properly
out <- as.data.frame(c(exL9))
colnames(out) <- c("time","cases","distance")
tab <- (xtable(out, digits=c(0,2,0,0)))
print(tab, type="html")
```

	time	cases	distance
1	16.68	7	560
2	11.50	3	220
3	12.03	3	340
4	14.88	4	80
5	13.75	6	150
6	18.11	7	330

7	8.00	2	110
8	17.83	7	210
9	79.24	30	1460
10	21.50	5	605
11	40.33	16	688
12	21.00	10	215
13	13.50	4	255
14	19.75	6	462
15	24.00	9	448
16	29.00	10	776
17	15.35	6	200
18	19.00	7	132
19	9.50	3	36
20	35.10	17	770
21	17.90	10	140
22	52.32	26	810
23	18.75	9	450
24	19.83	8	635
25	10.75	4	150

```
# Output data structure and dimensions
str(exL9)
```

'data.frame': 25 obs. of 3 variables: $ time : num 16.7 11.5 12 14.9 13.8 … $ cases : num 7 3 3 4 6 7 2 7 30 5 … $ distance: num 560 220 340 80 150 330 110 210 1460 605 …

```
dim(exL9)
```

[1] 25 3

Example 6.1 (p.213-214)

Calculate hat matrix values (by hand)

```
X <- cbind(matrix(1,length(distance),1),as.matrix(cases),as.matrix(distance))
y <- as.matrix(time)

xTx <- t(X) %*% X
H_matrix <- X %*% ginv(xTx, tol=.Machine$double.eps) %*% t(X)

# get the diagonal
diag(H_matrix)
```

```
##  [1] 0.10180178 0.07070164 0.09873476 0.08537479 0.07501050 0.04286693
##  [7] 0.08179867 0.06372559 0.49829216 0.19629595 0.08613260 0.11365570
## [13] 0.06112463 0.07824332 0.04111077 0.16594043 0.05943202 0.09626046
## [19] 0.09644857 0.10168486 0.16527689 0.39157522 0.04126005 0.12060826
## [25] 0.06664345
```

Calculate hat matrix values automatically

```
# perform multiple least squares regression
model <-  lm(time ~ cases+distance)

# calculate hat matrix automatically
hat_diags <- lm.influence(model)$hat
hat_diags
```

```
##          1          2          3          4          5          6          7
## 0.10180178 0.07070164 0.09873476 0.08537479 0.07501050 0.04286693 0.08179867
##          8          9         10         11         12         13         14
## 0.06372559 0.49829216 0.19629595 0.08613260 0.11365570 0.06112463 0.07824332
##         15         16         17         18         19         20         21
## 0.04111077 0.16594043 0.05943202 0.09626046 0.09644857 0.10168486 0.16527689
##         22         23         24         25
## 0.39157522 0.04126005 0.12060826 0.06664345
```

Create data frame to reproduce Table 6.1 on p. 214 - start with column for Observation and h_ii

```
# sequence of observations
Obs <- seq(1, length(time))

influence_stats <- data.frame(cbind(Obs, hat_diags))

out <- influence_stats
colnames(out) <- c("Obs $i$", "$h_{ii}$")
tab <- (xtable(out, digits=c(0,0,5)))
print(tab, type="html")
```

	Obs i	h_{ii}
1	1	0.10180
2	2	0.07070
3	3	0.09873
4	4	0.08537
5	5	0.07501
6	6	0.04287
7	7	0.08180
8	8	0.06373
9	9	0.49829
10	10	0.19630
11	11	0.08613
12	12	0.11366
13	13	0.06112
14	14	0.07824
15	15	0.04111
16	16	0.16594
17	17	0.05943

18	18	0.09626
19	19	0.09645
20	20	0.10168
21	21	0.16528
22	22	0.39158
23	23	0.04126
24	24	0.12061
25	25	0.06664

Create shell of unnamed table on p. 213

```
Run <- c("9 and 22 in",
         "9 out",
         "22 out",
         "9 and 22 out")
beta_0 <- c(" "," "," "," ")
beta_1 <- c(" "," "," "," ")
beta_2 <- c(" "," "," "," ")
MS_Res <- c(" "," "," "," ")
R_sqrd <- c(" "," "," "," ")

unnamed_table <- data.frame(cbind(Run,
                                  beta_0,
                                  beta_1,
                                  beta_2,
                                  MS_Res,
                                  R_sqrd))
out <- unnamed_table
colnames(out) <- c("Run",
                   "beta_hat_0",
                   "beta_hat_1",
                   "beta_hat_2",
                   "$MS_{Res}$",
                   "$R_2$")
tab <- (xtable(out, digits=c(0,0,0,0,0,0,0)))
print(tab, type="html")
```

	Run	beta_hat_0	beta_hat_1	beta_hat_2	MS_{Res}	R_2
1	9 and 22 in					
2	9 out					
3	22 out					
4	9 and 22 out					

Create models for the four scenarios in the unnamed table on p. 213

Note: Deletions using subset= are done sequentially. So, subset=(1:N)[-1][-2] removes the first observation and then the second of the remaining observations.

```
# scenario 1, points 9 and 22 in
time.s1 <- time
cases.s1 <- cases
distance.s1 <- distance

model.s1 <-  lm(time.s1 ~ cases.s1 + distance.s1)

# scenario 2, point 9 out
time.s2 <- time[1:length(time)][-9]
cases.s2 <- cases[1:length(cases)][-9]
distance.s2 <- distance[1:length(distance)][-9]

model.s2 <-  lm(time.s2 ~ cases.s2 + distance.s2)

# scenario 3, point 22 out
time.s3 <- time[1:length(time)][-22]
cases.s3 <- cases[1:length(cases)][-22]
distance.s3 <- distance[1:length(distance)][-22]

model.s3 <-  lm(time.s3 ~ cases.s3 + distance.s3)

# scenario 4, points 9 and 22 out
time.s4 <- time[1:length(time)][-9][-21]
cases.s4 <- cases[1:length(cases)][-9][-21]
distance.s4 <- distance[1:length(distance)][-9][-21]

model.s4 <-  lm(time.s4 ~ cases.s4 + distance.s4)
```

Display completed (unnamed) table on bottom of p. 213

```
Run <- c("9 and 22 in",
         "9 out",
         "22 out",
         "9 and 22 out")

beta_0 <- as.data.frame(c(model.s1$coeff[1],
            model.s2$coeff[1],
            model.s3$coeff[1],
            model.s4$coeff[1]))

beta_1 <- as.data.frame(c(model.s1$coeff[2],
            model.s2$coeff[2],
            model.s3$coeff[2],
            model.s4$coeff[2]))

beta_2 <- as.data.frame(c(model.s1$coeff[3],
            model.s2$coeff[3],
            model.s3$coeff[3],
            model.s4$coeff[3]))

MS_Res <- as.data.frame(c(anova(model.s1)$'Mean Sq'[3],
            anova(model.s2)$'Mean Sq'[3],
            anova(model.s3)$'Mean Sq'[3],
            anova(model.s4)$'Mean Sq'[3]))

R_sqrd <- as.data.frame(c(summary(model.s1)$r.squared,
            summary(model.s2)$r.squared,
            summary(model.s3)$r.squared,
            summary(model.s4)$r.squared))

unnamed_table2 <- data.frame(cbind(Run,
                                  beta_0,
                                  beta_1,
                                  beta_2,
                                  MS_Res,
                                  R_sqrd))
out2 <- unnamed_table2

colnames(out2) <- c("Run",
                   "beta_hat_0",
                   "beta_hat_1",
                   "beta_hat_2",
                   "$MS_{Res}$",
                   "$R_2$")

rownames(out2) <- c("1","2","3","4")
tab2 <- (xtable(out2, digits=c(0,0,3,3,3,3,4)))
print(tab2, type="html")
```

	Run	beta_hat_0	beta_hat_1	beta_hat_2	MS_{Res}	R_2
1	9 and 22 in	2.341	1.616	0.014	10.624	0.9596
2	9 out	4.447	1.498	0.010	5.905	0.9487
3	22 out	1.916	1.786	0.012	10.066	0.9564

4	9 and 22 out	4.643	1.456	0.011	6.163	0.9072

Example 6.2 (p.216)

Calculate Cook's D using Equation 6.5

```
# rstudent residual calculation
model.1 <- lm(time ~ cases + distance)

# Calculate studentized residuals, r_i (eqn 4.8)
e_i <- model.1$residuals
MS_Res <- anova(model.1)$'Mean Sq'[3]
r_i <- e_i/sqrt(MS_Res * (1-hat_diags))

p <- sum(hat_diags)

D_i <- ((r_i)^2/p) * (hat_diags/(1-hat_diags))

D_i
```

```
##            1            2            3            4            5            6 
## 1.000921e-01 3.375704e-03 9.455785e-06 7.764718e-02 5.432217e-04 1.231067e-04 
##            7            8            9           10           11           12 
## 2.171604e-03 3.051135e-03 3.419318e+00 5.384516e-02 1.619975e-02 1.596392e-03 
##           13           14           15           16           17           18 
## 2.294737e-03 3.292786e-03 6.319880e-04 3.289086e-03 4.013419e-04 4.397807e-02 
##           19           20           21           22           23           24 
## 1.191868e-02 1.324449e-01 5.086063e-02 4.510455e-01 2.989892e-02 1.023224e-01 
##           25 
## 1.084694e-04
```

Calculate Cook's D using cooks.distance(). Does this give the same answer as the "by hand" approach?

```
D_i_auto <- cooks.distance(model.1)
D_i_auto
```

```
##            1            2            3            4            5            6 
## 1.000921e-01 3.375704e-03 9.455785e-06 7.764718e-02 5.432217e-04 1.231067e-04 
##            7            8            9           10           11           12 
## 2.171604e-03 3.051135e-03 3.419318e+00 5.384516e-02 1.619975e-02 1.596392e-03 
##           13           14           15           16           17           18 
## 2.294737e-03 3.292786e-03 6.319880e-04 3.289086e-03 4.013419e-04 4.397807e-02 
##           19           20           21           22           23           24 
## 1.191868e-02 1.324449e-01 5.086063e-02 4.510455e-01 2.989892e-02 1.023224e-01 
##           25 
## 1.084694e-04
```

cooks.distance() matches the output from the by-hands approach.

Add Cook's D to the Table 6.1 dataframe

```
# obtain and add Cook's D to table 6.1 dataframe
influence_stats$Cooks_D <- c(D_i_auto)
```

Example 6.3 (p.218-219)

Calculate DFFITS and DFBETAS using R

```
influence_stats$DFFITS <- c(dffits(model.1))
dfbetas.col <- dfbetas(model.1)
influence_stats$DFBETAS_0 <- c(dfbetas.col[,1])
influence_stats$DFBETAS_1 <- c(dfbetas.col[,2])
influence_stats$DFBETAS_2 <- c(dfbetas.col[,3])
```

Update Table 6.1

```
out <- influence_stats
colnames(out) <- c("Obs $i$",
                   "$h_{ii}$",
                   "$D_i$",
                   "$DFFITS_i$",
                   "$DFBETAS_{0i}$",
                   "$DFBETAS_{1i}$",
                   "$DFBETAS_{2i}$")

tab <- (xtable(out, digits=c(0,0,5,5,4,4,4,4)))
print(tab, type="html")
```

	Obs i	h_{ii}	D_i	$DFFITS_i$	$DFBETAS_{0i}$	$DFBETAS_{1i}$	$DFBETAS_{2i}$
1	1	0.10180	0.10009	-0.5709	-0.1873	0.4113	-0.4349
2	2	0.07070	0.00338	0.0986	0.0898	-0.0478	0.0144
3	3	0.09873	0.00001	-0.0052	-0.0035	0.0039	-0.0028
4	4	0.08537	0.07765	0.5008	0.4520	0.0883	-0.2734
5	5	0.07501	0.00054	-0.0395	-0.0317	-0.0133	0.0242
6	6	0.04287	0.00012	-0.0188	-0.0147	0.0018	0.0011
7	7	0.08180	0.00217	0.0790	0.0781	-0.0223	-0.0110
8	8	0.06373	0.00305	0.0938	0.0712	0.0334	-0.0538
9	9	0.49829	3.41932	4.2961	-2.5757	0.9287	1.5076
10	10	0.19630	0.05385	0.3987	0.1079	-0.3382	0.3413
11	11	0.08613	0.01620	0.2180	-0.0343	0.0925	-0.0027
12	12	0.11366	0.00160	-0.0677	-0.0303	-0.0487	0.0540
13	13	0.06112	0.00229	0.0813	0.0724	-0.0356	0.0113
14	14	0.07824	0.00329	0.0974	0.0495	-0.0671	0.0618
15	15	0.04111	0.00063	0.0426	0.0223	-0.0048	0.0068
16	16	0.16594	0.00329	-0.0972	-0.0027	0.0644	-0.0842
17	17	0.05943	0.00040	0.0339	0.0289	0.0065	-0.0157
18	18	0.09626	0.04398	0.3653	0.2486	0.1897	-0.2724
19	19	0.09645	0.01192	0.1862	0.1726	0.0236	-0.0990
20	20	0.10168	0.13244	-0.6718	0.1680	-0.2150	-0.0929

21	21	0.16528	0.05086	-0.3885	-0.1619	-0.2972	0.3364
22	22	0.39158	0.45105	-1.1950	0.3986	-1.0254	0.5731
23	23	0.04126	0.02990	-0.3075	-0.1599	0.0373	-0.0527
24	24	0.12061	0.10232	-0.5711	-0.1197	0.4046	-0.4654
25	25	0.06664	0.00011	-0.0176	-0.0168	0.0008	0.0056

Example 6.4 (p. 219)

Calculate Covariance Ratio using R

```
influence_stats$covratio <- c(covratio(model.1))
```

Update Table 6.1

```
out <- influence_stats
colnames(out) <- c("Obs $i$",
                   "$h_{ii}$",
                   "$D_i$",
                   "$DFFITS_i$",
                   "$DFBETAS_{0i}$",
                   "$DFBETAS_{1i}$",
                   "$DFBETAS_{2i}$",
                   "$COVRATIO_i$")

tab <- (xtable(out, digits=c(0,0,5,5,4,4,4,4,4)))
print(tab, type="html")
```

	Obs i	h_{ii}	D_i	$DFFITS_i$	$DFBETAS_{0i}$	$DFBETAS_{1i}$	$DFBETAS_{2i}$	$COVRATIO_i$
1	1	0.10180	0.10009	-0.5709	-0.1873	0.4113	-0.4349	0.8711
2	2	0.07070	0.00338	0.0986	0.0898	-0.0478	0.0144	1.2149
3	3	0.09873	0.00001	-0.0052	-0.0035	0.0039	-0.0028	1.2757
4	4	0.08537	0.07765	0.5008	0.4520	0.0883	-0.2734	0.8760
5	5	0.07501	0.00054	-0.0395	-0.0317	-0.0133	0.0242	1.2396
6	6	0.04287	0.00012	-0.0188	-0.0147	0.0018	0.0011	1.1999
7	7	0.08180	0.00217	0.0790	0.0781	-0.0223	-0.0110	1.2398
8	8	0.06373	0.00305	0.0938	0.0712	0.0334	-0.0538	1.2056
9	9	0.49829	3.41932	4.2961	-2.5757	0.9287	1.5076	0.3422
10	10	0.19630	0.05385	0.3987	0.1079	-0.3382	0.3413	1.3054
11	11	0.08613	0.01620	0.2180	-0.0343	0.0925	-0.0027	1.1717
12	12	0.11366	0.00160	-0.0677	-0.0303	-0.0487	0.0540	1.2906
13	13	0.06112	0.00229	0.0813	0.0724	-0.0356	0.0113	1.2070
14	14	0.07824	0.00329	0.0974	0.0495	-0.0671	0.0618	1.2277
15	15	0.04111	0.00063	0.0426	0.0223	-0.0048	0.0068	1.1918
16	16	0.16594	0.00329	-0.0972	-0.0027	0.0644	-0.0842	1.3692
17	17	0.05943	0.00040	0.0339	0.0289	0.0065	-0.0157	1.2192
18	18	0.09626	0.04398	0.3653	0.2486	0.1897	-0.2724	1.0692
19	19	0.09645	0.01192	0.1862	0.1726	0.0236	-0.0990	1.2153
20	20	0.10168	0.13244	-0.6718	0.1680	-0.2150	-0.0929	0.7598

21	21	0.16528	0.05086	-0.3885	-0.1619	-0.2972	0.3364	1.2377
22	22	0.39158	0.45105	-1.1950	0.3986	-1.0254	0.5731	1.3981
23	23	0.04126	0.02990	-0.3075	-0.1599	0.0373	-0.0527	0.8897
24	24	0.12061	0.10232	-0.5711	-0.1197	0.4046	-0.4654	0.9476
25	25	0.06664	0.00011	-0.0176	-0.0168	0.0008	0.0056	1.2311

Identify observations that exceed limits of 1 +/- 3p/n for COVRATIO using which() and the “or” logical operator (|). Are these the same points identified in the textbook?

```
n <- length(time)
limit_plus <- (1 + 3*p/n)
limit_minus <- (1 - 3*p/n)
points <- which(influence_stats$covratio > limit_plus | influence_stats$covratio < limit_minus)
```

Points $9, 16, 22$ exceed the cutoff $COVRATIO_i$ limits of 0.64 and 1.36. The textbook identified points 9 and 22, but not point 16. For my calculations, point 16 barely exceeds the 1.36 limit.