Lesson 8 R Activity

Rick Davila

5/11/2020

Lesson 8 - Install packages

Install necessary packages using library()

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
library(nlme) # Needed for lme()


rm(list = ls())
```

Example 5.5

Upload data-ex-PrintingInk.xlsx and label columns

```
exL8 <- read.xlsx("data-ex-PrintingInk.xlsx",
                  sheetIndex = 1,
                  colIndex = c(1,2,3,4,5),
                  as.data.frame = TRUE,
                  header = TRUE)
```

Assign labels to data columns using names() and attach() commands

```
names(exL8) <- c("x1","x2","x3","y_bar","s")
attach(exL8)
```

Output data to make sure it reads properly

```
out <- as.data.frame(c(exL8))
colnames(out) <- c("x1","x2","x3","y_bar","s")
tab <- (xtable(out, digits=c(0,0,0,0,1,1)))
print(tab, type="html")
```

	x1	x2	x3	y_bar	s
1	-1	-1	-1	24.0	12.5
2	0	-1	-1	120.3	8.4
3	1	-1	-1	213.7	42.8
4	-1	0	-1	86.0	3.7
5	0	0	-1	136.7	80.4
6	1	0	-1	340.7	16.2
7	-1	1	-1	112.3	27.6
8	0	1	-1	256.3	4.6
9	1	1	-1	271.7	23.6
10	-1	-1	0	81.0	0.0
11	0	-1	0	101.7	17.7
12	1	-1	0	357.0	32.9
13	-1	0	0	171.3	15.0
14	0	0	0	372.0	0.0
15	1	0	0	501.7	92.5
16	-1	1	0	264.0	63.5
17	0	1	0	427.0	88.6
18	1	1	0	730.7	21.1
19	-1	-1	1	220.7	133.8
20	0	-1	1	239.7	23.5
21	1	-1	1	422.0	18.5
22	-1	0	1	199.0	29.4
23	0	0	1	485.3	44.6
24	1	0	1	673.7	158.2
25	-1	1	1	176.7	55.5
26	0	1	1	501.0	138.9
27	1	1	1	1010.0	142.5

```
# Output data structure and dimensions
str(exL8)
```

'data.frame': 27 obs. of 5 variables: $ x1 : num -1 0 1 -1 0 1 -1 0 1 -1 … $ x2 : num -1 -1 -1 0 0 0 1 1 1 -1 … $ x3 : num -1 -1 -1 -1 -1 -1 -1 -1 -1 0 … $ y_bar: num 24 120 214 86 137 … $ s : num 12.5 8.4 42.8 3.7 80.4 16.2 27.6 4.6 23.6 0 …

```
dim(exL8)
```

[1] 27 5

Create a linear model relating the regressors to the standard deviation ('s' in th data file)

```
model.1 <- lm(s ~ x1 + x2 + x3)
summary(model.1)
```

Call: lm(formula = s ~ x1 + x2 + x3)

Residuals: Min 1Q Median 3Q Max -54.87 -27.51 -11.29 26.53 83.47

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 48.000 7.808 6.147 2.85e-06 * **x1 11.517 9.563 1.204 0.24072**
x2 15.322 9.563 1.602 0.12275
x3 29.172 9.563 3.051 0.00567 — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ’ ’ 1

Residual standard error: 40.57 on 23 degrees of freedom Multiple R-squared: 0.3668, Adjusted R-squared: 0.2842 F-statistic: 4.441 on 3 and 23 DF, p-value: 0.0133

```
xtable(summary(model.1))
```

	Estimate <dbl>	Std. Error <dbl>	t value <dbl>	Pr(>|t|) <dbl>
(Intercept)	48.00000	7.808187	6.147394	2.853040e-06
x1	11.51667	9.563037	1.204290	2.407249e-01
x2	15.32222	9.563037	1.602234	1.227502e-01
x3	29.17222	9.563037	3.050519	5.674023e-03

4 rows

```
xtable(anova(model.1))
```

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
x1	1	2387.405	2387.405	1.450314	0.240724942
x2	1	4225.869	4225.869	2.567154	0.122750231
x3	1	15318.334	15318.334	9.305664	0.005674023
Residuals	23	37860.992	1646.130	NA	NA

4 rows

Define model weights

```
# fitted standard deviations based on x1, x2 and x3 regressors
y_hat.1 <- model.1$fitted.values    # fitted values
wts.1 <- 1/(y_hat.1^2 + 1)          # weights using recommend approach
V <- diag(1/wts.1)                  # diag matrix V
W <- ginv(V,
          tol=.Machine$double.eps)  # diag matrix W
```

Solve weighted least squares using matrices

```
# Define X matrix of regressor observations
X <- cbind(matrix(1,length(x1),1),
           as.matrix(x1),
           as.matrix(x2),
           as.matrix(x3))
y <- as.matrix(y_bar)

# Calculate C Matrix -- defined tol to use higher precision
beta_matrix <- ginv(t(X) %*% W %*% X,
                    tol=.Machine$double.eps) %*% t(X) %*% W %*% y
```

Solve weighted least squares using lm() and entering weights

```
# us weights determined in previous section

model.2 <- lm(y_bar ~ x1 + x2 + x3, weights = wts.1)
summary(model.2)
```

```
##
## Call:
## lm(formula = y_bar ~ x1 + x2 + x3, weights = wts.1)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6254 -1.2823  0.1667  1.3308  4.2558
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   289.53      18.90  15.319 1.47e-13 ***
## x1            108.20      10.61  10.203 5.23e-10 ***
## x2             69.31      11.57   5.989 4.16e-06 ***
## x3            100.28      17.93   5.594 1.08e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.954 on 23 degrees of freedom
## Multiple R-squared:  0.8671, Adjusted R-squared:  0.8498
## F-statistic: 50.02 on 3 and 23 DF,  p-value: 3.081e-10
```

```
xtable(summary(model.2))
```

	Estimate <dbl>	Std. Error <dbl>	t value <dbl>	Pr(>|t|) <dbl>
(Intercept)	289.53470	18.90086	15.318596	1.471686e-13
x1	108.20059	10.60503	10.202764	5.227332e-10
x2	69.30814	11.57181	5.989394	4.161321e-06
x3	100.28229	17.92827	5.593527	1.083511e-05

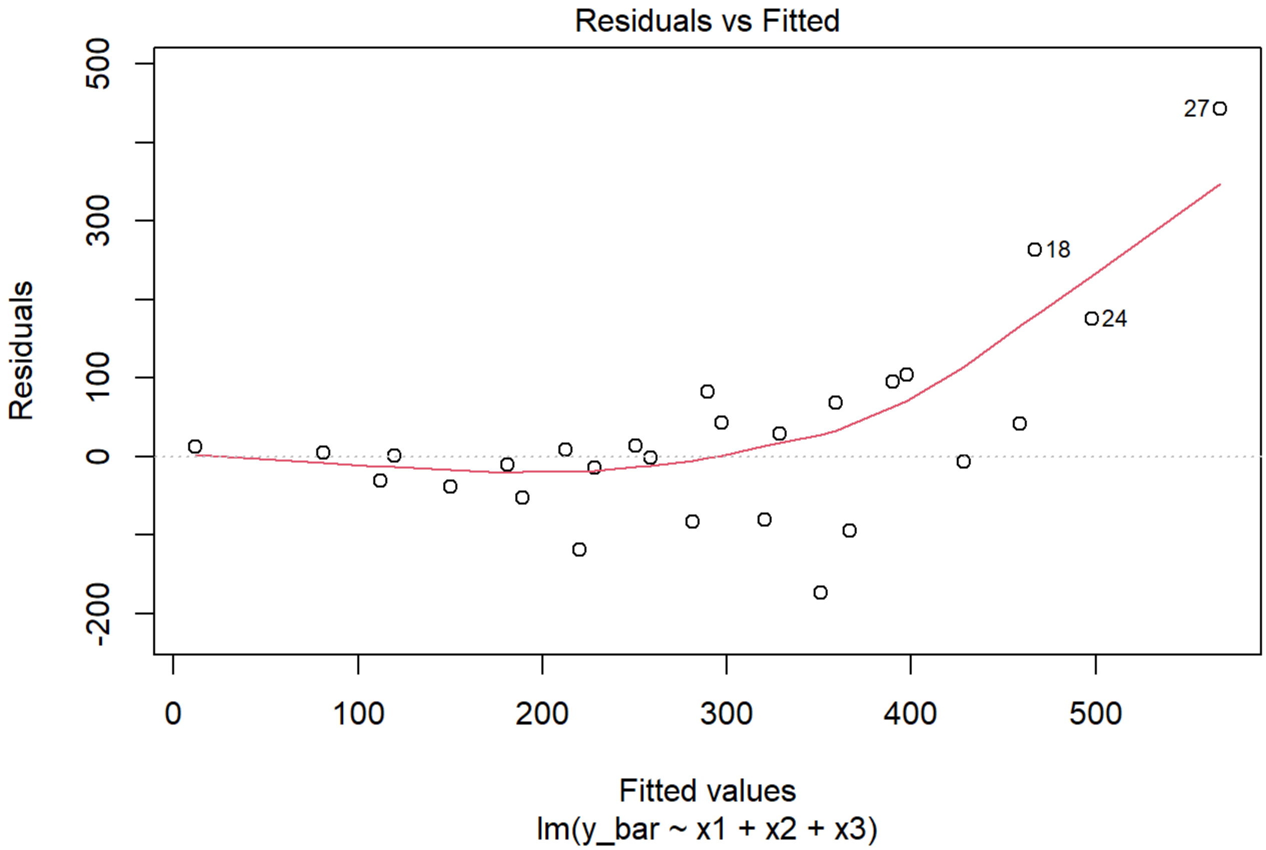
4 rows

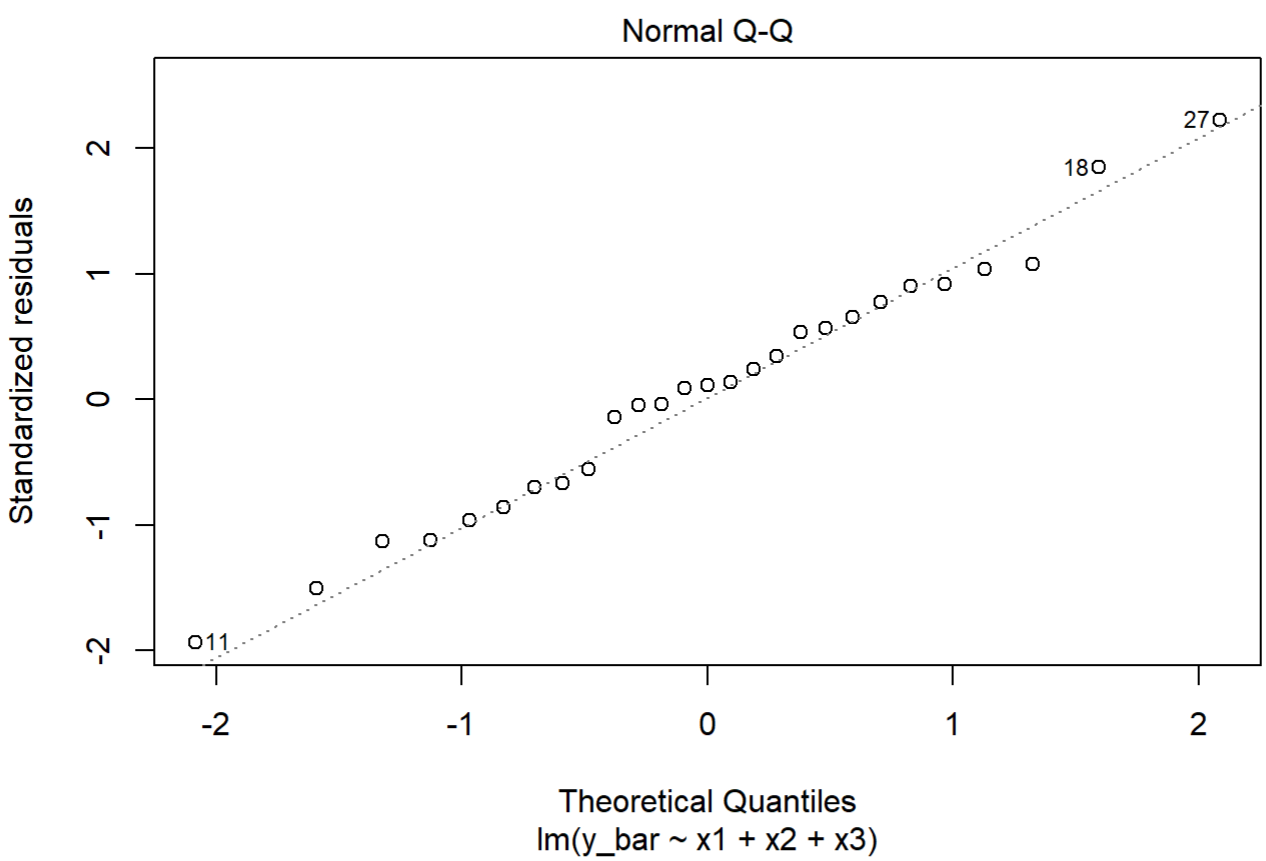
```
xtable(anova(model.2))
```

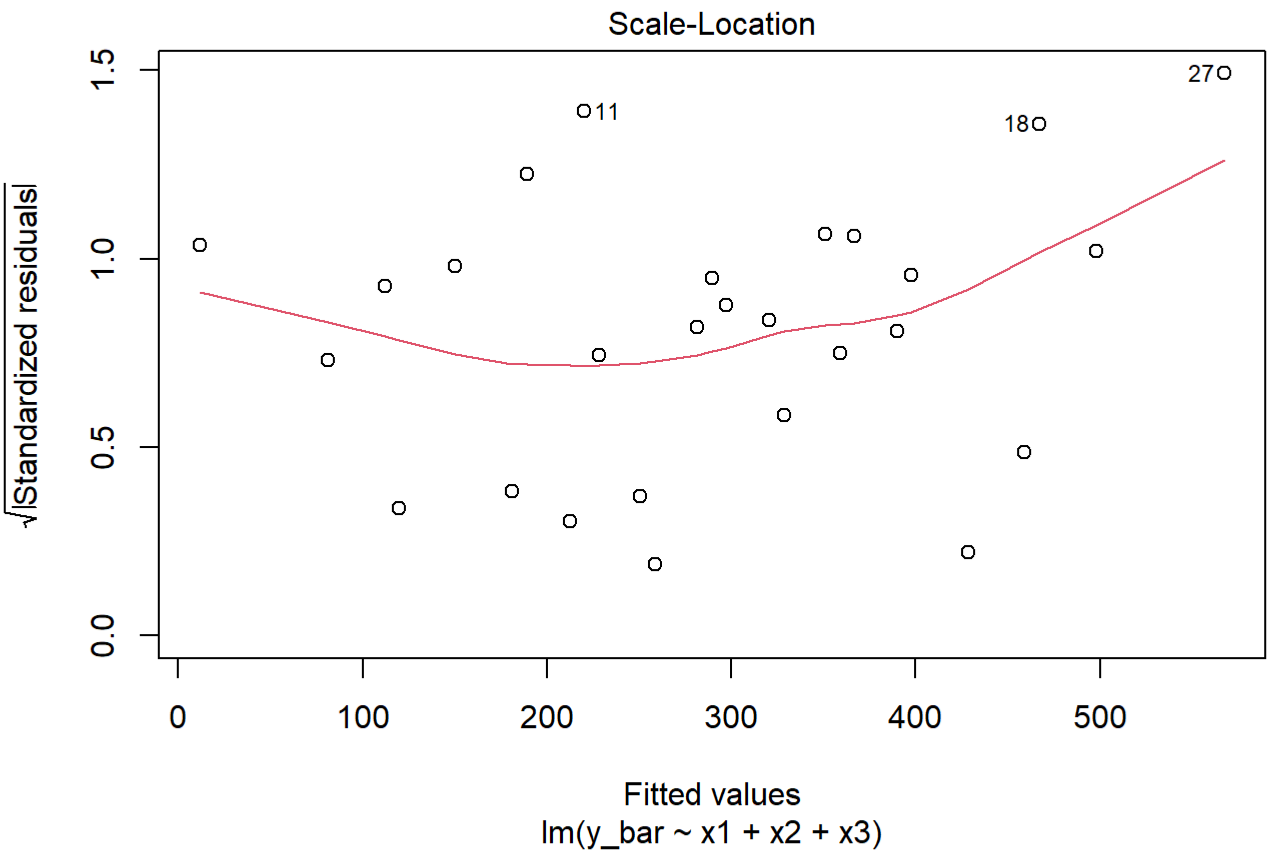
	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
x1	1	270.08404	270.084037	70.76544	1.792084e-08
x2	1	183.25088	183.250880	48.01405	4.600139e-07
x3	1	119.41235	119.412349	31.28755	1.083511e-05
Residuals	23	87.78201	3.816609	NA	NA

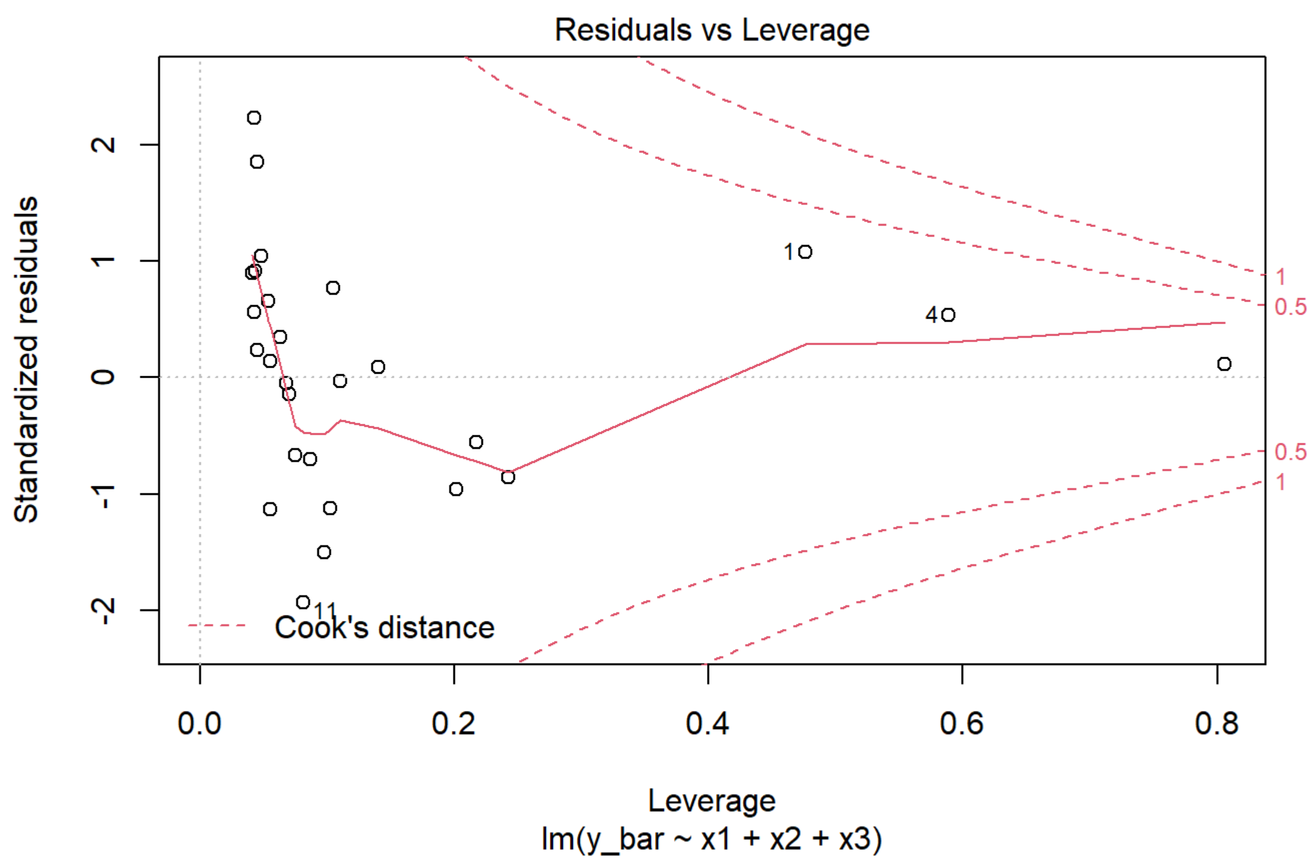
4 rows

```
plot(model.2)
```


Residuals vs Fitted
Residuals
500
300
100
0
-200
0
100
200
300
400
500
27
18
24
Fitted values
lm(y_bar ~ x1 + x2 + x3)


Normal Q-Q
Standardized residuals
2
1
0
-1
-2
-2
-1
0
1
2
27
18
11
Theoretical Quantiles
lm(y_bar ~ x1 + x2 + x3)

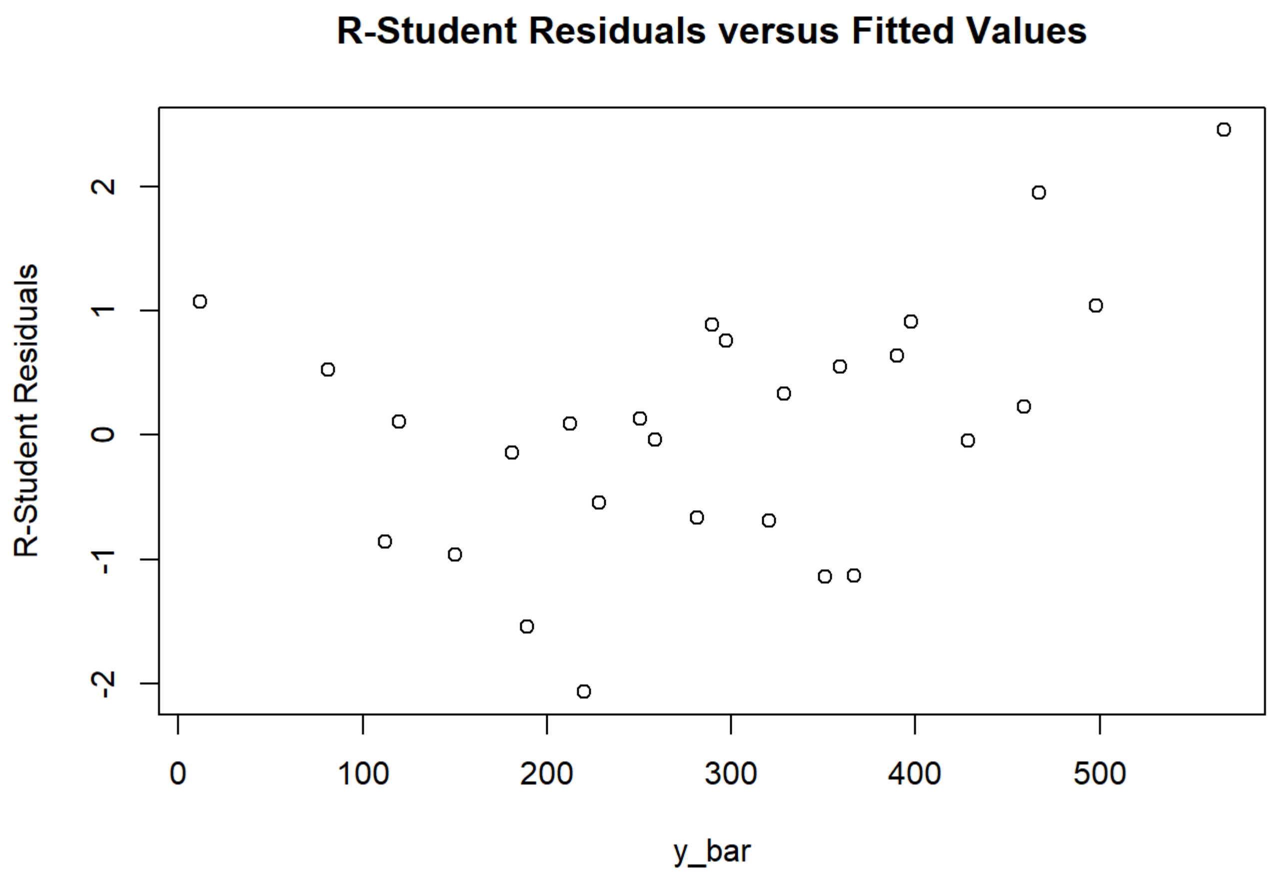

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
100
200
300
400
500
Fitted values
lm(y_bar ~ x1 + x2 + x3)
11
18
27


Residuals vs Leverage
Standardized residuals
-2
-1
0
1
2
0.0
0.2
0.4
0.6
0.8
Leverage
lm(y_bar ~ x1 + x2 + x3)
Cook's distance
1
0.5
0.5
1
1
4
11

```
# Calculate r_student residuals
R_Student_Residuals.2 <- rstudent(model.2)

y_hat.2 <- model.2$fitted.values

plot(y_hat.2, R_Student_Residuals.2, ylab = "R-Student Residuals", xlab = "y_bar", main = "R-Stu
dent Residuals versus Fitted Values")
```



Display both sets of coefficients - should be the same (minor differences possible due to rounding)

```
model.1coeff <- as.data.frame(c(beta_matrix[1],
                                beta_matrix[2],
                                beta_matrix[3],
                                beta_matrix[4]))
model.2coeff <- as.data.frame(c(model.2$coefficients[1],
                                model.2$coefficients[2],
                                model.2$coefficients[3],
                                model.2$coefficients[4]))

out <- as.data.frame(c(model.1coeff, model.2coeff))
colnames(out) <- c("matrix",
                   "using lm()")
rownames(out) <- c("beta_0","beta_1","beta_2","beta_3")
tab <- (xtable(out, digits=c(0,6,6)))
print(tab, type="html")
```

	matrix	using lm()
beta_0	289.534697	289.534697
beta_1	108.200587	108.200587
beta_2	69.308140	69.308140
beta_3	100.282292	100.282292

Example 5.6 (p. 196-199)

Upload data-ex-5-6.xlsx and label columns

```
exL8_5pt6 <- read.xlsx("data-ex-5-6.xlsx",
                   sheetIndex = 1,
                   colIndex = c(1,2,3,4,5,6),
                   as.data.frame = TRUE,
                   header = TRUE)

names(exL8_5pt6) <- c("Helicopter",
                      "Aspect",
                      "Paper",
                      "Interaction",
                      "Rep",
                      "Time")
attach(exL8_5pt6)

out <- as.data.frame(c(exL8_5pt6))
colnames(out) <- c("Helicopter",
                      "Aspect",
                      "Paper",
                      "Interaction",
                      "Rep",
                      "Time")
tab <- (xtable(out, digits=c(0,0,0,0,0,0,2)))
print(tab, type="html")
```

	Helicopter	Aspect	Paper	Interaction	Rep	Time

1	1	1	-1	-1	1	3.60
2	1	1	-1	-1	1	3.85
3	1	1	-1	-1	1	3.98
4	2	-1	-1	1	1	6.44
5	2	-1	-1	1	1	6.37
6	2	-1	-1	1	1	6.78
7	3	-1	1	-1	1	6.84
8	3	-1	1	-1	1	6.90
9	3	-1	1	-1	1	7.18
10	4	-1	1	-1	2	6.37
11	4	-1	1	-1	2	6.38
12	4	-1	1	-1	2	6.58
13	5	1	1	1	1	3.44
14	5	1	1	1	1	3.43
15	5	1	1	1	1	3.75
16	6	1	-1	-1	2	3.75
17	6	1	-1	-1	2	3.73
18	6	1	-1	-1	2	4.10
19	7	1	1	1	2	4.59
20	7	1	1	1	2	4.64
21	7	1	1	1	2	5.02
22	8	-1	-1	1	2	6.50
23	8	-1	-1	1	2	6.33
24	8	-1	-1	1	2	6.92

```
# Output data structure and dimensions
str(exL8_5pt6)
```

'data.frame': 24 obs. of 6 variables: $ Helicopter : num 1 1 1 2 2 2 3 3 3 4 … $ Aspect : num 1 1 1 -1 -1 -1 -1 -1 -1 -1 … $ Paper : num -1 -1 -1 -1 -1 -1 1 1 1 1 … $ Interaction: num -1 -1 -1 1 1 1 -1 -1 -1 -1 … $ Rep : num 1 1 1 1 1 1 1 1 1 2 … $ Time : num 3.6 3.85 3.98 6.44 6.37 6.78 6.84 6.9 7.18 6.37 …

```
dim(exL8_5pt6)
```

[1] 24 6

Create a linear model predicting time as a function of aspect, paper,and the interaction. Results should match Table 5.11.

```
model.3 <- lm(Time ~ Aspect + Paper + Interaction)
summary(model.3)
```

```
##
## Call:
## lm(formula = Time ~ Aspect + Paper + Interaction)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.71500 -0.22875 -0.07083  0.23375  0.87500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.31125    0.08339  63.689  < 2e-16 ***
## Aspect      -1.32125    0.08339 -15.844 8.74e-13 ***
## Paper        0.11542    0.08339   1.384    0.182
## Interaction  0.03958    0.08339   0.475    0.640
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4085 on 20 degrees of freedom
## Multiple R-squared:  0.9268, Adjusted R-squared:  0.9158
## F-statistic: 84.39 on 3 and 20 DF,  p-value: 1.583e-11
```

```
xtable(summary(model.3))
```

	Estimate <dbl>	Std. Error <dbl>	t value <dbl>	Pr(>|t|) <dbl>
(Intercept)	5.31125000	0.0833931	63.6893193	1.427502e-24
Aspect	-1.32125000	0.0833931	-15.8436363	8.736922e-13
Paper	0.11541667	0.0833931	1.3840073	1.816076e-01
Interaction	0.03958333	0.0833931	0.4746596	6.401715e-01

4 rows

```
xtable(anova(model.3))
```

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
Aspect	1	41.89683750	41.89683750	251.0208101	8.736922e-13
Paper	1	0.31970417	0.31970417	1.9154763	1.816076e-01
Interaction	1	0.03760417	0.03760417	0.2253017	6.401715e-01
Residuals	20	3.33811667	0.16690583	NA	NA

4 rows

Create a linear model of groups, with the response being the group average (i.e. use Table 5.14 as input data). Results should match Table 5.15. Okay to manually input the

Table 5.14 information.

```
Helicopter <- c(1,2,3,4,5,6,7,8)
Aspect <- c(1,-1,-1,-1,1,1,1,-1)
Paper <- c(-1,-1,1,1,1,-1,1,-1)
Interaction <- c(-1,1,-1,-1,1,-1,1,1)
Time <- c(3.81, 6.53, 6.973, 6.443, 3.54, 3.86, 4.75, 6.583)

model.4 <- lm(Time ~ Aspect + Paper + Aspect*Paper)
summary(model.4)
```

```
##
## Call:
## lm(formula = Time ~ Aspect + Paper + Aspect * Paper)
##
## Residuals:
##       1       2       3       4       5       6       7       8
## -0.0250 -0.0265  0.2650 -0.2650 -0.6050  0.0250  0.6050  0.0265
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.31112    0.16537  32.116  5.6e-06 ***
## Aspect       -1.32113    0.16537  -7.989  0.00133 **
## Paper         0.11537    0.16537   0.698  0.52382
## Aspect:Paper  0.03963    0.16537   0.240  0.82241
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4677 on 4 degrees of freedom
## Multiple R-squared:  0.9415, Adjusted R-squared:  0.8976
## F-statistic: 21.45 on 3 and 4 DF,  p-value: 0.006292
```

```
xtable(summary(model.4))
```

	Estimate <dbl>	Std. Error <dbl>	t value <dbl>	Pr(>|t|) <dbl>
(Intercept)	5.311125	0.1653741	32.1158297	5.603673e-06
Aspect	-1.321125	0.1653741	-7.9887078	1.331028e-03
Paper	0.115375	0.1653741	0.6976608	5.238156e-01
Aspect:Paper	0.039625	0.1653741	0.2396083	8.224113e-01

4 rows

```
xtable(anova(model.4))
```

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
Aspect	1	13.96297013	13.96297013	63.81945188	0.001331028
Paper	1	0.10649112	0.10649112	0.48673063	0.523815552
Aspect:Paper	1	0.01256113	0.01256113	0.05741215	0.822411297
Residuals	4	0.87515450	0.21878863	NA	NA

4 rows

Example 5.7 (p.201-202)

Upload data-ex-5-7.xlsx and label columns

```
exL8_5pt7 <- read.xlsx("data-ex-5-7.xlsx",
                   sheetIndex = 1,
                   colIndex = c(2,3,4,5),
                   as.data.frame = TRUE,
                   header = TRUE)

names(exL8_5pt7) <- c("time",
                      "cases",
                      "distance",
                      "city")
attach(exL8_5pt7)

out <- as.data.frame(c(exL8_5pt7))
colnames(out) <- c("time",
                   "cases",
                   "distance",
                   "city")
tab <- (xtable(out, digits=c(0,2,0,0,0)))
print(tab, type="html")
```

	time	cases	distance	city
1	16.68	7	560	San Diego
2	11.50	3	220	San Diego
3	12.03	3	340	San Diego
4	14.88	4	80	San Diego
5	13.75	6	150	San Diego
6	18.11	7	330	San Diego
7	8.00	2	110	San Diego
8	17.83	7	210	Boston
9	79.24	30	1460	Boston
10	21.50	5	605	Boston
11	40.33	16	688	Boston
12	21.00	10	215	Boston
13	13.50	4	255	Boston

14	19.75	6	462	Boston
15	24.00	9	448	Boston
16	29.00	10	776	Boston
17	15.35	6	200	Boston
18	19.00	7	132	Austin
19	9.50	3	36	Austin
20	35.10	17	770	Austin
21	17.90	10	140	Austin
22	52.32	26	810	Austin
23	18.75	9	450	Austin
24	19.83	8	635	Minneapolis
25	10.75	4	150	Minneapolis

```
# Output data structure and dimensions
str(exL8_5pt7)
```

'data.frame': 25 obs. of 4 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 … $ city : chr "San Diego" "San Diego" "San Diego" "San Diego" …

```
dim(exL8_5pt7)
```

[1] 25 4

Run regression model using City as a random factor using lme(). Compare StdDev for (Intercept) and Residual to variances in Figure 5.12.

```
model.5 <- lme(time ~ cases + distance, random =~1|city)

summary(model.5)
```

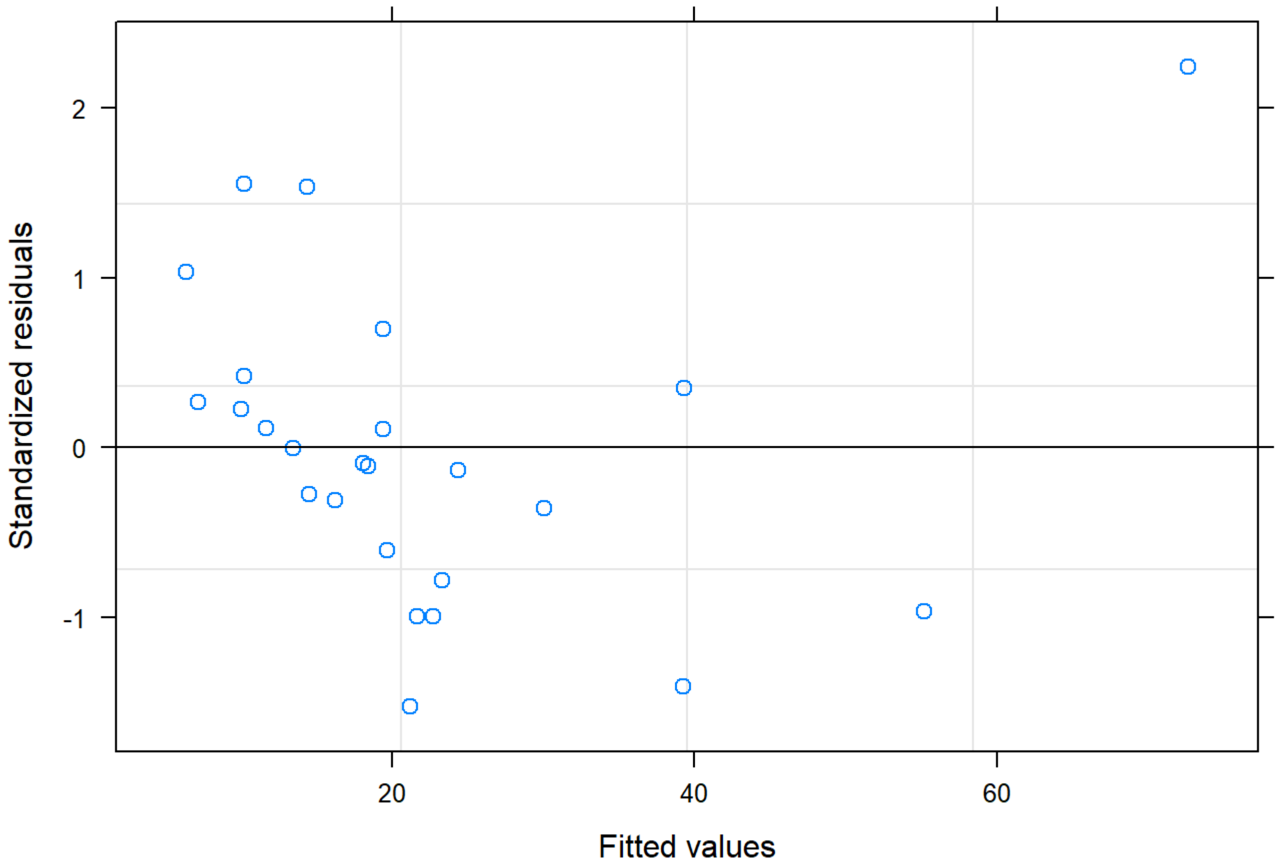
```
## Linear mixed-effects model fit by REML
##   Data: NULL
##        AIC      BIC    logLik
##   146.684 152.1392 -68.34199
##
## Random effects:
##  Formula: ~1 | city
##         (Intercept) Residual
## StdDev:    1.609268 2.964796
##
## Fixed effects:  time ~ cases + distance
##                 Value Std.Error DF  t-value p-value
## (Intercept) 2.0754319 1.3204411 19 1.571772  0.1325
## cases       1.7148234 0.1719497 19 9.972817  0.0000
## distance    0.0120317 0.0035580 19 3.381543  0.0031
##  Correlation:
##          (Intr) cases
## cases    -0.213
## distance -0.143 -0.836
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -1.52550355 -0.60242771 -0.09382676  0.34986657  2.24343282
##
## Number of Observations: 25
## Number of Groups: 4
```

```
xtable(anova(model.5))
```

	numDF <int>	**denDF** <dbl>	**F-value** <dbl>	**p-value** <dbl>
(Intercept)	1	19	430.78289	1.620926e-14
cases	1	19	545.54731	1.887379e-15
distance	1	19	11.43483	3.132507e-03

3 rows

```
plot(model.5)
```




```
sd_res <- model.5$sigma
var_res <- sd_res^2
```

From lme(), the standard deviation for the residual is 2.9647961, which, when squared is equal to the variance in Figure 5.12 in the book, 8.7900161.

Similarly, for the Intercept, the standard deviation is 1.609268, and the variance is 2.589743, matching the value in Figure 5.12.