L16Ex_HeldCement

Rick Davila

6/09/2020

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

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

#Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jdk-14.0.1') # for 64-bit version
#library(rJava)

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

```
## Warning: package 'xlsx' was built under R version 4.0.4
```

```
library(MuMIn)
```

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

```
# Import data
Lex16_1 <- read.xlsx("data-ex-10-1.xlsx", sheetIndex = 1, sheetName=NULL, rowIndex=NULL, startRow=NULL, endRow=NULL, colInde
x= NULL, as.data.frame=TRUE, header=TRUE, colClasses=NA, keepFormulas=FALSE, encoding="unknown")

# Give labels to data columns
names(Lex16_1) <- c("y", "x1", "x2", "x3", "x4")
attach(Lex16_1)

# Output data to make sure it reads properly
Lex16_1
```

	y <dbl>	x1 <dbl>	x2 <dbl>	x3 <dbl>	x4 <dbl>
	78.5	7	26	6	60

y <dbl>	x1 <dbl>	x2 <dbl>	x3 <dbl>	x4 <dbl>
74.3	1	29	15	52
104.3	11	56	8	20
87.6	11	31	8	47
95.9	7	52	6	33
109.2	11	55	9	22
102.7	3	71	17	6
72.5	1	31	22	44
93.1	2	54	18	22
115.9	21	47	4	26

1-10 of 13 rows Previous **1** 2 Next

```
# Output data dimensions
dim(Lex16_1)
```

```
## [1] 13  5
```

```
### Example 11.1 (p. 375) ###
# Reproduce Table 11.1 on p. 375

# First, generate lm model using all possible regressions
Lex16_1_lm <- lm(y~x1+x2+x3+x4, data=Lex16_1, na.action = "na.fail") # Linear model of raw data

# Use dredge() function to automatically perform all regressors regression
combinations <- dredge(Lex16_1_lm, extra = c(R_Sq = function(x) summary(x)$r.squared,R_Sq_Adj = function(x) summary(x)$adj.
r.squared, MS_Res = function(x) summary(x)$sigma^2,Cp, MallowCp = function(x) summary(x)$sigma^2*df.residual(x)/summary(Lex1
6_1_lm)$sigma^2-dim(Lex16_1)[1]+2*length(x$coefficients)))
```

```
## Fixed term is "(Intercept)"
```

```
print(combinations)
```

```
## Global model call: lm(formula = y ~ x1 + x2 + x3 + x4, data = Lex16_1, na.action = "na.fail")
## ---
## Model selection table
##    (Intrc)    x1      x2      x3      x4   R_Sq R_Sq_Adj  MS_Res      Cp
## 4    52.58 1.468  0.6623                 0.9787   0.9744   5.790   92.65
## 12   71.65 1.452  0.4161         -0.2365 0.9823   0.9764   5.330   90.62
## 8    48.19 1.696  0.6569  0.2500         0.9823   0.9764   5.346   90.88
## 10  103.10 1.440                 -0.6140 0.9725   0.9670   7.476  119.60
## 14  111.70 1.052         -0.4100 -0.6428 0.9813   0.9750   5.648   96.02
## 15  203.60       -0.9234 -1.4480 -1.5570 0.9728   0.9638   8.202  139.40
## 16   62.41 1.551  0.5102  0.1019 -0.1441 0.9824   0.9736   5.983  107.70
## 13  131.30               -1.2000 -0.7246 0.9353   0.9223  17.570  281.20
## 7    72.07        0.7313 -1.0080         0.8470   0.8164  41.540  664.70
## 9   117.60                        -0.7382 0.6745   0.6450  80.350 1205.00
## 3    57.42        0.7891                 0.6663   0.6359  82.390 1236.00
## 11   94.16        0.3109         -0.4569 0.6801   0.6161  86.890 1390.00
## 2    81.48 1.869                         0.5339   0.4916 115.100 1726.00
## 6    72.35 2.312          0.4945         0.5482   0.4578 122.700 1963.00
## 5   110.20               -1.2560         0.2859   0.2210 176.300 2645.00
## 1    95.42                               0.0000   0.0000 226.300 3168.00
##    MallowCp df  logLik  AICc delta weight
## 4     2.678  4 -28.156  69.3  0.00  0.566
## 12    3.018  5 -26.933  72.4  3.13  0.119
## 8     3.041  5 -26.952  72.5  3.16  0.116
## 10    5.496  4 -29.817  72.6  3.32  0.107
## 14    3.497  5 -27.310  73.2  3.88  0.081
## 15    7.337  5 -29.734  78.0  8.73  0.007
## 16    5.000  6 -26.918  79.8 10.52  0.003
## 13   22.370  4 -35.372  83.7 14.43  0.000
## 7    62.440  4 -40.965  94.9 25.62  0.000
## 9   138.700  3 -45.872 100.4 31.10  0.000
## 3   142.500  3 -46.035 100.7 31.42  0.000
## 11  138.200  4 -45.761 104.5 35.21  0.000
## 2   202.500  3 -48.206 105.1 35.77  0.000
## 6   198.100  4 -48.005 109.0 39.70  0.000
## 5   315.200  3 -50.980 110.6 41.31  0.000
## 1   442.900  2 -53.168 111.5 42.22  0.000
## Models ranked by AICc(x)
```

from the combinations, select the first to models

```
# model 1
model.1 <- lm(y ~ x1 + x2)
summary(model.1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -2.893 -1.574 -1.302  1.363  4.048
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 52.57735    2.28617   23.00 5.46e-10 ***
## x1           1.46831    0.12130   12.11 2.69e-07 ***
## x2           0.66225    0.04585   14.44 5.03e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.406 on 10 degrees of freedom
## Multiple R-squared:  0.9787, Adjusted R-squared:  0.9744
## F-statistic: 229.5 on 2 and 10 DF,  p-value: 4.407e-09
```

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

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
x1	1	1450.07633	1450.076328	250.4256	2.088092e-08
x2	1	1207.78227	1207.782266	208.5818	5.028960e-08
Residuals	10	57.90448	5.790448	*NA*	*NA*

3 rows

```
# model 2
model.2 <- lm(y ~ x1 + x2 + x4)
summary(model.2)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0919 -1.8016  0.2562  1.2818  3.8982
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  71.6483    14.1424   5.066 0.000675 ***
## x1            1.4519     0.1170  12.410 5.78e-07 ***
## x2            0.4161     0.1856   2.242 0.051687 .
## x4           -0.2365     0.1733  -1.365 0.205395
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.309 on 9 degrees of freedom
## Multiple R-squared:  0.9823, Adjusted R-squared:  0.9764
## F-statistic: 166.8 on 3 and 9 DF,  p-value: 3.323e-08
```

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

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
x1	1	1450.076328	1450.076328	272.043870	4.933517e-08
x2	1	1207.782266	1207.782266	226.587908	1.094167e-07
x4	1	9.931754	9.931754	1.863262	2.053954e-01
Residuals	9	47.972729	5.330303	*NA*	*NA*

4 rows

model 1 fitted equation

$$\hat{y} = 52.5773489 + (1.4683057)x_1 + (0.6622505)x_2$$

model 2 fitted equation

$$\hat{y} = 71.648307 + (1.451938)x_1 + (0.4161098)x_2 + (-0.2365402)x_4$$

Example 11.1 (p. 375)

Reproduce Table 11.1 on p. 375

```
library(e1071)
library(xtable)

# using x1,x2 and x4 data values, calculate respective model y_hats
model.1.out <- model.1$coefficients[1] +
  model.1$coefficients[2] * x1  +
  model.1$coefficients[3] * x2

model.2.out <- model.2$coefficients[1] +
  model.2$coefficients[2] * x1 +
  model.2$coefficients[3] * x2 +
  model.2$coefficients[4] * x4

table_models <- data.frame(cbind(y,x1,x2,x3,x4,model.1.out,model.2.out))

out <- table_models
colnames(out) <- c("$y$",
                   "$x_1$",
                   "$x_2$",
                   "$x_3$",
                   "$x_4$",
                   "model 1",
                   "model 2")

tab <- (xtable(out, digits=c(0,1,0,0,0,0,3,3)))
print(tab, type="html")
```

	y	x_1	x_2	x_3	x_4	model 1	model 2

1	78.5	7	26	6	60	80.074	78.438
2	74.3	1	29	15	52	73.251	72.867
3	104.3	11	56	8	20	105.815	106.191
4	87.6	11	31	8	47	89.258	89.402
5	95.9	7	52	6	33	97.293	95.644
6	109.2	11	55	9	22	105.152	105.302
7	102.7	3	71	17	6	104.002	104.129
8	72.5	1	31	22	44	74.575	75.592
9	93.1	2	54	18	22	91.275	91.818
10	115.9	21	47	4	26	114.538	115.546
11	83.8	1	40	23	34	80.536	81.702
12	113.3	11	66	9	12	112.437	112.244
13	109.4	10	68	8	12	112.293	111.625