Lesson 3 R Activity

Rick Davila

4/25/2020

Install packages

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

library(e1071)
library("xlsx")
library(xtable)

rm(list = ls())
```

Read in rocket propellant data

Copy and paste the following information to your completed Lesson 1 R Activity .R file. Read data from Excel spreadsheet using the read.xlsx() command

```
# uncomment for laptop 1
ex2_1 <- read.xlsx("data-ex-2-1.xlsx",
                   sheetIndex = 1,
                   colIndex = c(2,3),
                   as.data.frame = TRUE,
                   header = TRUE)
```

Rocket propellant data - table printout

```
xtable(ex2_1)
```

% latex table generated in R 4.0.2 by xtable 1.8-4 package % Mon Dec 28 21:22:17 2020

Data structure and dimensions of data

Output data structure and dimensions using the str() and dim() commands

```
# output dataframe structure
str(ex2_1)
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ Shear.Strength..psi..y_i     : num  2159 1678 2316 2061 2208 ...
##  $ Age.of.Propellant..weeks..x_i: num  15.5 23.8 8 17 5.5 ...
```

	Shear.Strength..psi..y_i	Age.of.Propellant..weeks..x_i
1	2158.70	15.50
2	1678.15	23.75
3	2316.00	8.00
4	2061.30	17.00
5	2207.50	5.50
6	1708.30	19.00
7	1784.70	24.00
8	2575.00	2.50
9	2357.90	7.50
10	2256.70	11.00
11	2165.20	13.00
12	2399.55	3.75
13	1779.80	25.00
14	2336.75	9.75
15	1765.30	22.00
16	2053.50	18.00
17	2414.40	6.00
18	2200.50	12.50
19	2654.20	2.00
20	1753.70	21.50

```r
# dim of data 'matrix' (i.e., should be 20 rows by 2 columns)
dim(ex2_1)
```

```
## [1] 20  2
```

Print out tabel using revised column names

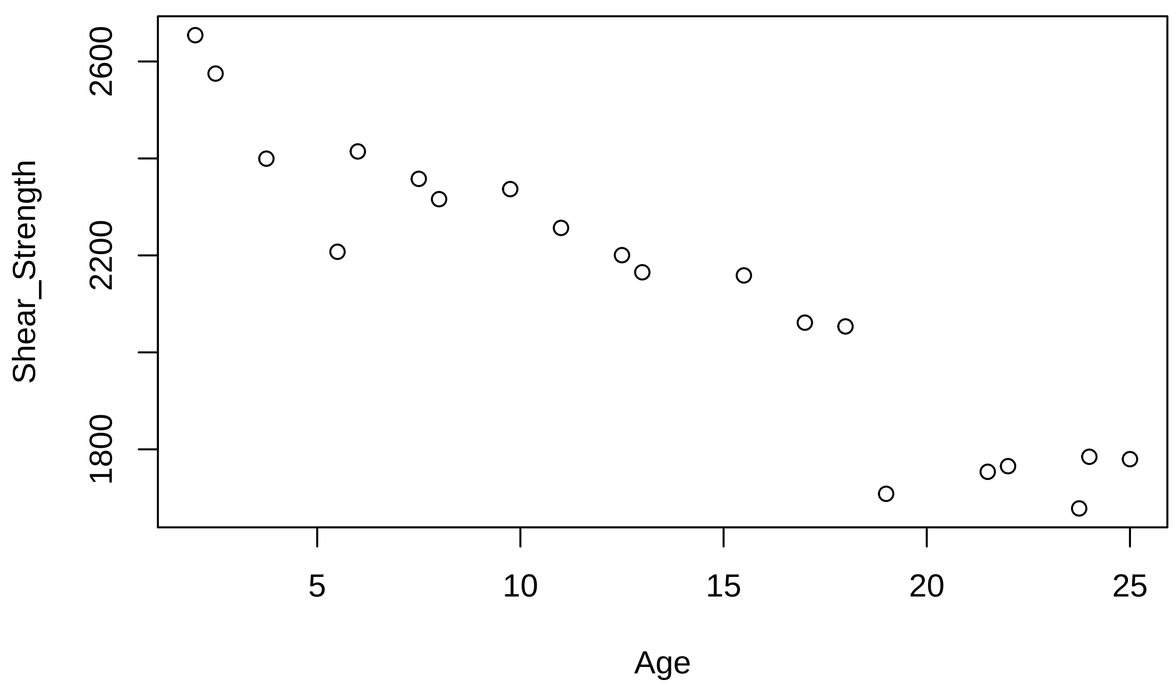
```r
names(ex2_1) <- c("Shear_Strength", "Age")
attach(ex2_1)

#Printout revised table using new column names
xtable(ex2_1)
```

% latex table generated in R 4.0.2 by xtable 1.8-4 package % Mon Dec 28 21:22:17 2020

Create scatter plot

```r
# Create scatterplot using plot()
plot(Age, Shear_Strength)
```

	Shear_Strength	Age
1	2158.70	15.50
2	1678.15	23.75
3	2316.00	8.00
4	2061.30	17.00
5	2207.50	5.50
6	1708.30	19.00
7	1784.70	24.00
8	2575.00	2.50
9	2357.90	7.50
10	2256.70	11.00
11	2165.20	13.00
12	2399.55	3.75
13	1779.80	25.00
14	2336.75	9.75
15	1765.30	22.00
16	2053.50	18.00
17	2414.40	6.00
18	2200.50	12.50
19	2654.20	2.00
20	1753.70	21.50



Create linear model "by hand"

First, perform intermediary calculations - Syy, Sxx, Sxy, x_bar, y_bar, num

```
# num = length(Age)
num = length(Age)

Syy = sum(Shear_Strength^2) - (sum(Shear_Strength))^2/num
print(sprintf("Syy = %f",Syy))
```

```
## [1] "Syy = 1693737.601375"
```

```
# From Eq. (2.9) in the e-book
Sxx = sum(Age^2) - (sum(Age))^2/num
print(sprintf("Sxx = %f",Sxx))
```

```
## [1] "Sxx = 1106.559375"
```

```
# From Eq. (2.10) in the e-book
Sxy=sum(Age*Shear_Strength)-sum(Age)*sum(Shear_Strength)/num
print(sprintf("Sxy = %f",Sxy))
```

```
## [1] "Sxy = -41112.654375"
```

```
#x_bar and y_bar ...
x_bar = sum(Age)/num
y_bar = sum(Shear_Strength)/num
print(sprintf("x_bar = %f and y_bar = %f",x_bar, y_bar))
```

```
## [1] "x_bar = 13.362500 and y_bar = 2131.357500"
```

Determine estimates for slope and intercept

```
# slope, from Eq. (2.11) in the e-book
beta_1 = Sxy/Sxx
print(sprintf("beta_1 = %f",beta_1))
```

```
## [1] "beta_1 = -37.153591"
```

```
# intercept, from Eq. (2.6) in the e-book
beta_0 = y_bar - beta_1*x_bar
print(sprintf("beta_0 = %f",beta_0))
```

```
## [1] "beta_0 = 2627.822359"
```

```
print('The Least Squares regression line by hand:')
```

```
## [1] "The Least Squares regression line by hand:"
```

```r
print(sprintf(" y_hat = %f + (%f)x",beta_0, beta_1))
```

```
## [1] " y_hat = 2627.822359 + (-37.153591)x"
```

The equation is

$$\hat{y} = (-37.15)x + (2627.82)$$

Output the model 'by hand'

```r
# Output 'by hand' model by hand
y_hat = beta_0 + beta_1*Age
Results = data.frame(Shear_Strength, y_hat)
names(Results) <- c("Observed Values","Fitted Values")
xtable(Results)
```

% latex table generated in R 4.0.2 by xtable 1.8-4 package % Mon Dec 28 21:22:17 2020

	Observed Values	Fitted Values
1	2158.70	2051.94
2	1678.15	1745.42
3	2316.00	2330.59
4	2061.30	1996.21
5	2207.50	2423.48
6	1708.30	1921.90
7	1784.70	1736.14
8	2575.00	2534.94
9	2357.90	2349.17
10	2256.70	2219.13
11	2165.20	2144.83
12	2399.55	2488.50
13	1779.80	1698.98
14	2336.75	2265.57
15	1765.30	1810.44
16	2053.50	1959.06
17	2414.40	2404.90
18	2200.50	2163.40
19	2654.20	2553.52
20	1753.70	1829.02

Create the linear model using lm() command and display using the summary() command

```r
# Create linear model using lm() command and display output using the summary() command
model=lm(Shear_Strength~Age)
summary(model)
```

Call: lm(formula = Shear_Strength ~ Age)

Residuals: Min 1Q Median 3Q Max -215.98 -50.68 28.74 66.61 106.76

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 2627.822 44.184 59.48 < 2e-16 ***Age -37.154 2.889 -12.86 1.64e-10*** — Signif. codes: 0 '' ***0.001*** '' *0.01* '' 0.05 '.' 0.1 ' ' 1

Residual standard error: 96.11 on 18 degrees of freedom Multiple R-squared: 0.9018, Adjusted R-squared: 0.8964 F-statistic: 165.4 on 1 and 18 DF, p-value: 1.643e-10

Example 2.2 (p.21-22)

Obtain ANOVA table elements (by hand) - SST, SS_Regression, SS_Residual

```
SST = sum((Shear_Strength - y_bar)^2)
SS_Regression = sum((y_hat - y_bar)^2)
SS_Residual = sum((Shear_Strength - y_hat)^2)

print(sprintf("note, Syy = SST = %f and SS_Reg+SS_Res = %f",Syy, SS_Regression+SS_Residual))
```

```
## [1] "note, Syy = SST = 1693737.601375 and SS_Reg+SS_Res = 1693737.601375"
```

```
# Define degrees of freedom (by hand)
# num = length(Age) - number of observations
df_SST = num - 1          # 20 - 1 = 19
df_SS_Regression = 1              # 1
df_SS_Residual = num - 2  # 20 - 2 = 18

# Obtain residual degrees of freedom automatically using df.residual()
dfresidual = df.residual(model)
print(sprintf("residual degrees of freedom using df.residals = %i",dfresidual))
```

```
## [1] "residual degrees of freedom using df.residals = 18"
```

```
# Obtain estimated error variance (MS_Residual)
MS_Residual = SS_Residual/dfresidual
print(sprintf("sigma^2, estimated error variance = %f",MS_Residual))
```

```
## [1] "sigma^2, estimated error variance = 9236.381004"
```

```
#Obtain ANOVA table using aov() command or anova() command and display output
aov(model)
```

```
## Call:
##    aov(formula = model)
##
## Terms:
##                        Age Residuals
## Sum of Squares  1527482.7  166254.9
## Deg. of Freedom         1        18
##
## Residual standard error: 96.10609
## Estimated effects may be unbalanced
```

```r
anova(model)
```

```
## Analysis of Variance Table
##
## Response: Shear_Strength
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## Age        1 1527483 1527483  165.38 1.643e-10 ***
## Residuals 18  166255    9236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example 2.3 (p.25)

Test significance of slope parameter (by hand) using the qt() and pt() commands for critical value and p-value for the t-distribution

```r
se_beta1 = sqrt(MS_Residual/Sxx)
test_statistic = beta_1/se_beta1

# for alpha = 0.05
alpha = 0.05
tcritical_value = abs(qt(alpha/2.0, df = df_SS_Residual))
p_value = 2*pt(-abs(test_statistic), df = df_SS_Residual)
print(sprintf("test statistic = %f, critical value = %f and p-value = %e",
              test_statistic, tcritical_value, p_value))
```

```
## [1] "test statistic = -12.859889, critical value = 2.100922 and p-value = 1.643344e-10"
```

The absolute value of the test statistic is greater than the critical value

$$12.8 > 2.1$$

and p value is

$$1.64 \times 10^{-10}$$

… there's a linear relationship between the shear strength and the age of the propellant

Example 2.4 (p. 28)

Test for significance of Regression (F-test) at .01 significance level

(1) By hand

```r
# By hand:
# F_o test statistic
siglevel = 0.01

F_o = (SS_Regression/df_SS_Regression)/(SS_Residual/df_SS_Residual)
print(sprintf("F_o test statistic = %f", F_o))
```

```
## [1] "F_o test statistic = 165.376758"
```

```r
Fcritical = qf(1-siglevel, df_SS_Regression, df_SS_Residual)
print(sprintf("F statistic = %f",Fcritical))
```

```
## [1] "F statistic = 8.285420"
```

```r
# as a check, from table A.4 in the e-book, F_(0.01,1,18) = 8.29
p_val = 1 - pf(F_o, df_SS_Regression, df_SS_Residual)
print(sprintf("F p value = %e",p_val))
```

```
## [1] "F p value = 1.643343e-10"
```

Since

$$F_0 = 165.36 > F_{(0.01,1,18)}$$

We can reject the null hypotheses, there's a relationship between the shear strength and the age of the rocket propellant

2) Using aov and anova commands

Using aov and anova functions:

aov model output

```r
aov(model)
```

```
## Call:
##    aov(formula = model)
##
## Terms:
##                       Age Residuals
## Sum of Squares  1527482.7  166254.9
## Deg. of Freedom         1        18
##
## Residual standard error: 96.10609
## Estimated effects may be unbalanced
```

anova model output

```r
anova(model)
```

```
## Analysis of Variance Table
##
## Response: Shear_Strength
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## Age        1 1527483 1527483  165.38 1.643e-10 ***
## Residuals 18  166255    9236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example 3 2.5 (p 30)

Construct a 95% confidence interval (CI) for the slope parameter (by hand) The 100 $(1-\alpha)$ percent confidence interveral (CI) of the slope β_1 is given by

$$\hat{\beta}_1 - t_{\alpha/2,n-2} se(\hat{\beta}_1) \leq \hat{\beta}_1 \leq \hat{\beta}_1 + t_{\alpha/2,n-2} se(\hat{\beta}_1)$$

We construct a 95 CI on β_1 by using the standard error of $\hat{\beta}_1$ is $se(\hat{\beta}_1) = \sqrt{MS_{Res}/S_{xx}}$ and $t_{0.025,18} = 2.100922$

```
# Example 2.5 (p. 30)
# Construct a 95% confidence interval for the slope parameter (by hand)

se_beta_1 = sqrt(MS_Residual/Sxx)
CIleft_wing = beta_1 - tcritical_value * se_beta_1
CIright_wing = beta_1 + tcritical_value * se_beta_1
```

The 95% CI on the slope is

$$-37.15 - (2.1)(2.89) \leq \hat{\beta}_1 \leq -37.15 + (2.1)(2.89)$$

or

$$-43.2233786 \leq \hat{\beta}_1 \leq -31.0838033$$

check using the Confint command

```
# Check using Confint command
confint(model, level = 0.95)
```

```
##                   2.5 %    97.5 %
## (Intercept) 2534.99540 2720.6493
## Age          -43.22338  -31.0838
```

Construct 95% C.I. for the point x0 = 13.3625

Consider finding a 95% CI on $E(y|x_0)$. The CI per Eq. (2.43) in the e-book is

$$\hat{\mu}_{y|x_0} - t_{\alpha/2,n-2}\sqrt{MS_{Res}\left(\frac{1}{n} + \frac{(x_0-\bar{x})^2}{Sxx}\right)} \leq E(y|x_0) \leq \hat{\mu}_{y|x_0} + t_{\alpha/2,n-2}\sqrt{MS_{Res}\left(\frac{1}{n} + \frac{(x_0-\bar{x})^2}{Sxx}\right)}$$

for the Rocket Propellant data, we have

$$\hat{\mu}_{y|x_0} - (2.1)\sqrt{9236.38\left(\frac{1}{20} + \frac{(x_0-13.36)^2}{1106.56}\right)} \leq E(y|x_0) \leq \hat{\mu}_{y|x_0} + (2.1)\sqrt{9236.38\left(\frac{1}{20} + \frac{(x_0-13.36)^2}{1106.56}\right)}$$

For this case, $\hat{\mu}_{y|x_0=13.3625}$ is

```
x0 = 13.3625
mu_hat_x0 = beta_1 * x0 + beta_0
mu_hat_x0
```

```
## [1] 2131.358
```

and the CI at $\hat{\mu}_{y|x_0=13.3625} = 2131.3575$ is

```
Sqrt_quantity = sqrt(MS_Residual*(1/num + (x0 - x_bar)^2/Sxx))
CIx0_left_wing = mu_hat_x0 - tcritical_value * Sqrt_quantity
CIx0_right_wing = mu_hat_x0 + tcritical_value * Sqrt_quantity
CIx0_left_wing
```

```
## [1] 2086.209
```

```
CIx0_right_wing
```

```
## [1] 2176.506
```

or

$$2086.2087367 \leq E(y|x_0 = 13.3625) \leq 2176.5062633$$

Example 2.7 (p. 34-35)

Construct a 95% prediction interval for $x_0 = 10$ weeks

The $100(1-\alpha)$ percent prediction interval on a future observation at x_0 is

$$\hat{y_0} - t_{\alpha/2,n-2}\sqrt{MS_{Res}\left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{Sxx}\right)} \leq y_o \leq \hat{y_0} + t_{\alpha/2,n-2}\sqrt{MS_{Res}\left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{Sxx}\right)}$$

for $x_0 = 10$ weeks, $\hat{y_0}$ is

```
x0 = 10
y_hat = beta_1 * x0 + beta_0
y_hat
```

```
## [1] 2256.286
```

$\hat{y_0} = 2256.2864496$

so, the interval becomes

$$2256.29 - (2.1)\sqrt{9236.38\left(1 + \frac{1}{20} + \frac{(10 - 13.36)^2}{1106.56}\right)} \leq y_0 \leq 2256.29 + (2.1)\sqrt{9236.38\left(1 + \frac{1}{20} + \frac{(10 - 13.36)^2}{1106.56}\right)}$$

which simplies to

```
x0 = 10
Sqrt_quantity = sqrt(MS_Residual*(1 + 1/num + (x0 - x_bar)^2/Sxx))
PI_left_wing = y_hat - tcritical_value * Sqrt_quantity
PI_right_wing = y_hat + tcritical_value * Sqrt_quantity
PI_left_wing
```

```
## [1] 2048.385
```

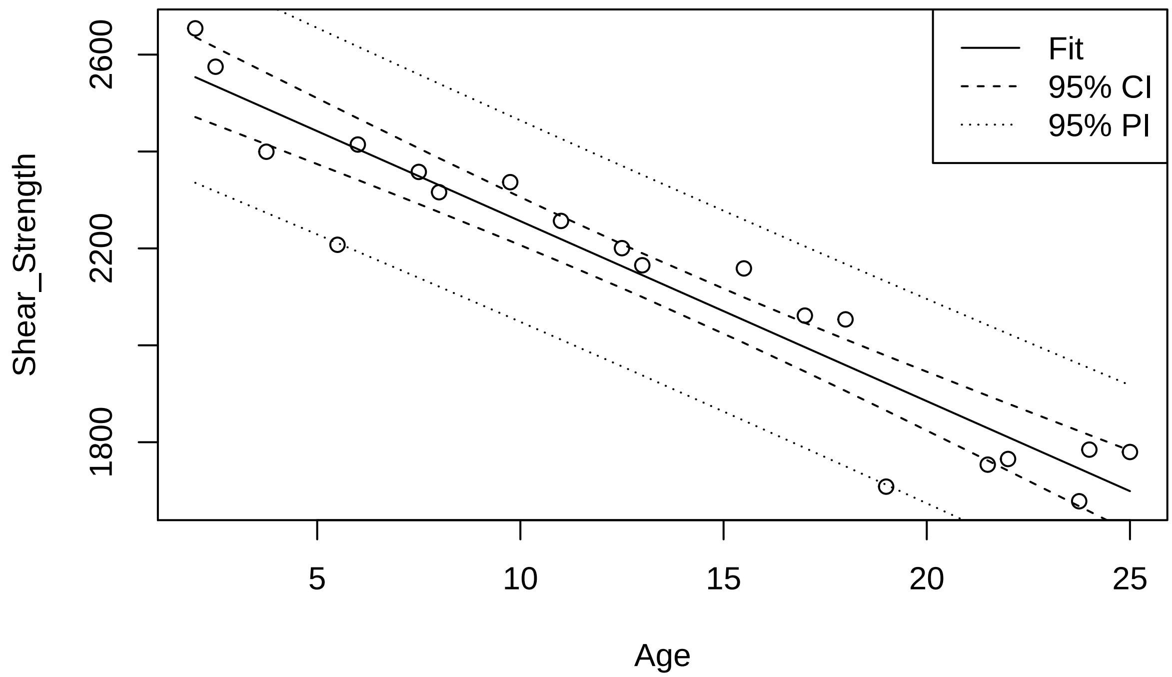
```
PI_right_wing
```

```
## [1] 2464.188
```

$$2048.3845936 \le y_0 \le 2464.1883055$$

Superimpose a 95% prediction and confidence interval plots onto the scatterplot of data using predict(), plot(), lines() and order() commands.

```
# Obtain prediction interval lines automatically using predict()
# sorting data frame by Age ... recall from Lesson 1, attach(ex2_1) dataset

sorted_data <- ex2_1[order(Age),]
ci_band = predict(model, sorted_data, interval = "confidence", level = 0.95)

# Obtain confidence interval lines automatically using predict()
pi_band = predict(model, sorted_data, interval = "prediction", level = 0.95)

# Create scatter plot using plot()
plot(Age, Shear_Strength)

# Add line of predicted y_hat values using lines().
# Be sure to use order() to ensure that x-values are in ascending (versus random) order

# sorted Age column, or x-values, in ascending order are now , contained in sorted_data[,2]
y_hat = beta_1*sorted_data[,2] + beta_0
lines(sorted_data[,2], y_hat, lty = 1)

# Add prediction interval lines using lines() and order() commands
lines(sorted_data[,2], ci_band[,2], lty = 2)
lines(sorted_data[,2], ci_band[,3], lty = 2)


# Add confidence interval lines using lines() and order() commands
lines(sorted_data[,2], pi_band[,2], lty = 3)
lines(sorted_data[,2], pi_band[,3], lty = 3)

legend("topright", legend = c("Fit","95% CI","95% PI"), lty = c(1,2,3))
```


Fit
95% CI
95% PI
Shear_Strength
2600
2200
1800
5
10
15
20
25
Age