L13Ex_ToolLifeEx_Rick_Davila

Rick Davila

5/29/2020

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)
```

```
## java.home option:
```

```
## JAVA_HOME environment variable: C:\Program Files\Java\jdk-14.0.1
```

```
## Warning in fun(libname, pkgname): Java home setting is INVALID, it will be ignored.
## Please do NOT set it unless you want to override system settings.
```

```
library(e1071)
library("xlsx")
```

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

```
library(xtable)
# for laptop 2 -- result of reloading java JDK version


rm(list = ls())

# load data
Ex81 <- read.xlsx(
  "data-ex-8-1.xlsx",
  sheetIndex = 1, sheetName=NULL, rowIndex=NULL,
  startRow=NULL, endRow=NULL, colIndex= c(1,2,3,4),
  as.data.frame=TRUE, header=TRUE, colClasses=NA,
  keepFormulas=FALSE, encoding="unknown")

# Give labels to data columns
names(Ex81) <- c("Obs",
                 "y_i",
                 "x_1",
                 "Tool_Type")
attach(Ex81)

# Output data to make sure it reads properly
out <- as.data.frame(c(Ex81))
colnames(out) <- c("Obs",
                 "y_i",
                 "x_1",
                 "Tool_Type")
tab <- (xtable(out, digits=c(0,0,2,0,0)))
print(tab, type="html")
```

	Obs	y_i	x_1	Tool_Type
1	1	18.73	610	A
2	2	14.52	950	A
3	3	17.43	720	A
4	4	14.54	840	A
5	5	13.44	980	A
6	6	24.39	530	A
7	7	13.34	680	A
8	8	22.71	540	A
9	9	12.68	890	A
10	10	19.32	730	A
11	11	30.16	670	B
12	12	27.09	770	B
13	13	25.40	880	B
14	14	26.05	1000	B
15	15	33.49	760	B
16	16	35.62	590	B
17	17	26.07	910	B
18	18	36.78	650	B
19	19	34.95	810	B

20	20	43.67	500	B

```
# Output data structure and dimensions
str(Ex81)
```

'data.frame': 20 obs. of 4 variables: $ Obs : num 1 2 3 4 5 6 7 8 9 10 … $ y_i : num 18.7 14.5 17.4 14.5 13.4 … $ x_1 : num 610 950 720 840 980 530 680 540 890 730 … $ Tool_Type: chr "A" "A" "A" "A" …

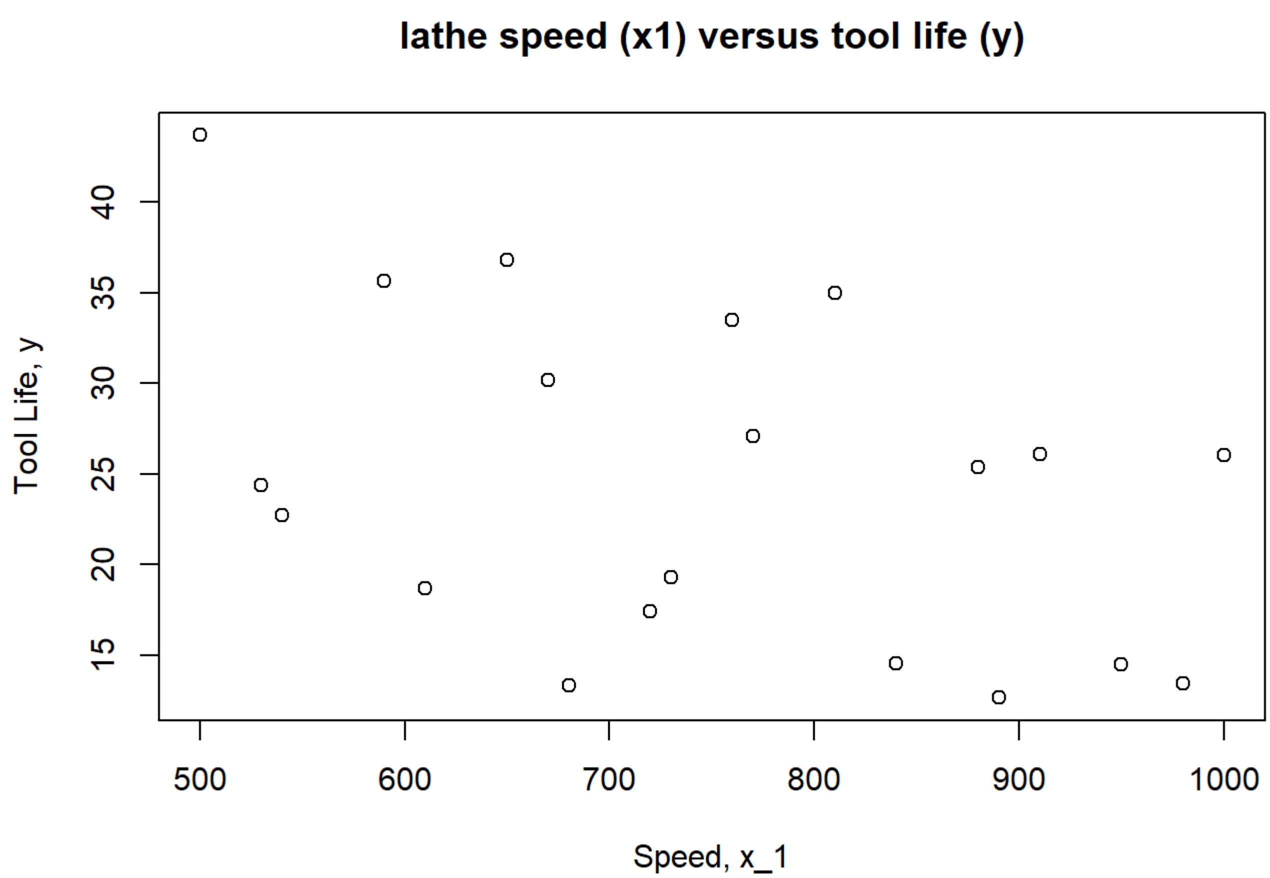
```
dim(Ex81)
```

[1] 20 4

Example 8.1 (p.262-264)

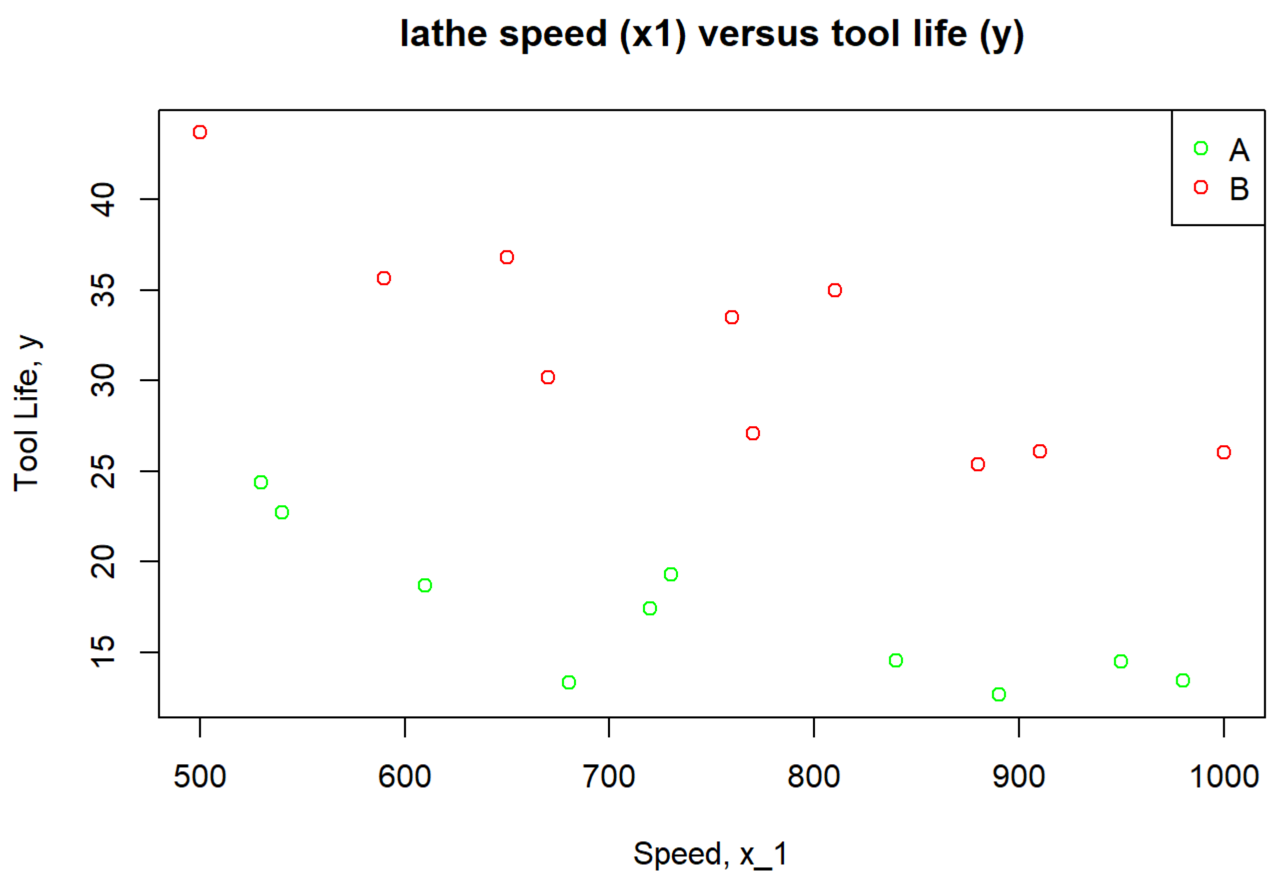
Create scatterplot to look at the data - all points look the same

```
plot(x_1,y_i, main = "lathe speed (x1) versus tool life (y)",
     xlab = "Speed, x_1",
     ylab = "Tool Life, y")
```



Create scatterplot - distinguish between Tool Type using points() command to make Tool Type A green and Tool Type B red.

```
plot(x_1,y_i, main = "lathe speed (x1) versus tool life (y)",
     xlab = "Speed, x_1",
     ylab = "Tool Life, y")

# in the for loop, use points() function to distinguish between tool types
for (pltpts in seq(1:length(y_i))){

  if (Tool_Type[pltpts] == "A") {
    points(x_1[pltpts], y_i[pltpts], col = "green")}
  else if (Tool_Type[pltpts] == "B") {
    points(x_1[pltpts], y_i[pltpts], col = "red")}
}

legend("topright", legend=c("A", "B"),
       col=c("green", "red"),
       pch=c(1,1))
```



Define indicator variables for Tool Type and add to dataframe

```
x_2 <- vector()

for (i in seq(1:length(y_i))){

  if (Tool_Type[i] == "A") {
    x_2[i] = 1}
  else if (Tool_Type[i] == "B") {
    x_2[i] = 0}
}

Ex81$x_2 <- c(x_2)

# Output data to make sure it reads properly
out <- as.data.frame(c(Ex81))
colnames(out) <- c("Obs",
                   "y_i",
                   "x_1",
                   "Tool_Type",
                   "x_2")
tab <- (xtable(out, digits=c(0,0,2,0,0,0)))
print(tab, type="html")
```

	Obs	y_i	x_1	Tool_Type	x_2
1	1	18.73	610	A	1
2	2	14.52	950	A	1
3	3	17.43	720	A	1
4	4	14.54	840	A	1
5	5	13.44	980	A	1
6	6	24.39	530	A	1
7	7	13.34	680	A	1
8	8	22.71	540	A	1
9	9	12.68	890	A	1
10	10	19.32	730	A	1
11	11	30.16	670	B	0
12	12	27.09	770	B	0
13	13	25.40	880	B	0
14	14	26.05	1000	B	0
15	15	33.49	760	B	0
16	16	35.62	590	B	0
17	17	26.07	910	B	0
18	18	36.78	650	B	0
19	19	34.95	810	B	0
20	20	43.67	500	B	0

Create linear model; compare to Table 8.2 on p. 264

```
model.81 <- lm(y_i ~ x_1 + x_2)

print(xtable(summary(model.81),
             digits=c(0,3,3,2,3),
             display=c("s","f","f","f","g"))
      ,type="html")
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	51.990	3.541	14.68	4.34e-11
x_1	-0.027	0.005	-5.89	1.79e-05
x_2	-15.004	1.360	-11.04	3.59e-09

```
print(xtable(anova(model.81),
             digits=c(0,0,3,3,3,3),
             display=c("s","d","f","f","f","g"))
      ,type="html")
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
x_1	1	293.005	293.005	31.716	2.99e-05
x_2	1	1125.028	1125.028	121.776	3.59e-09
Residuals	17	157.055	9.239		

The least squares fit equation is:

$$\hat{y} = 51.9898518 + (-0.0266072)x_1 + (-15.0042506)x_2$$

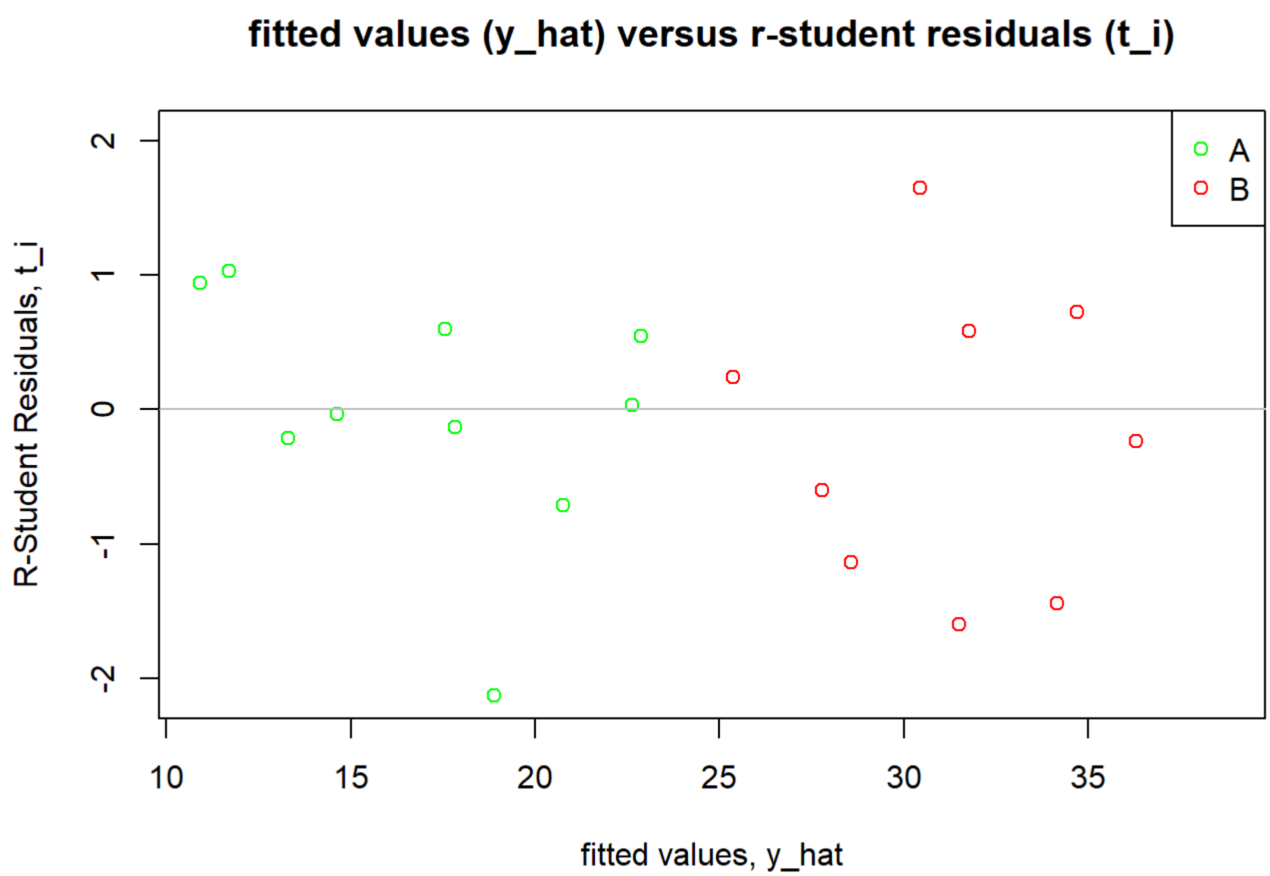
$$R^2 = 0.9002884$$

$$R^2_{adj} = 0.8885576$$

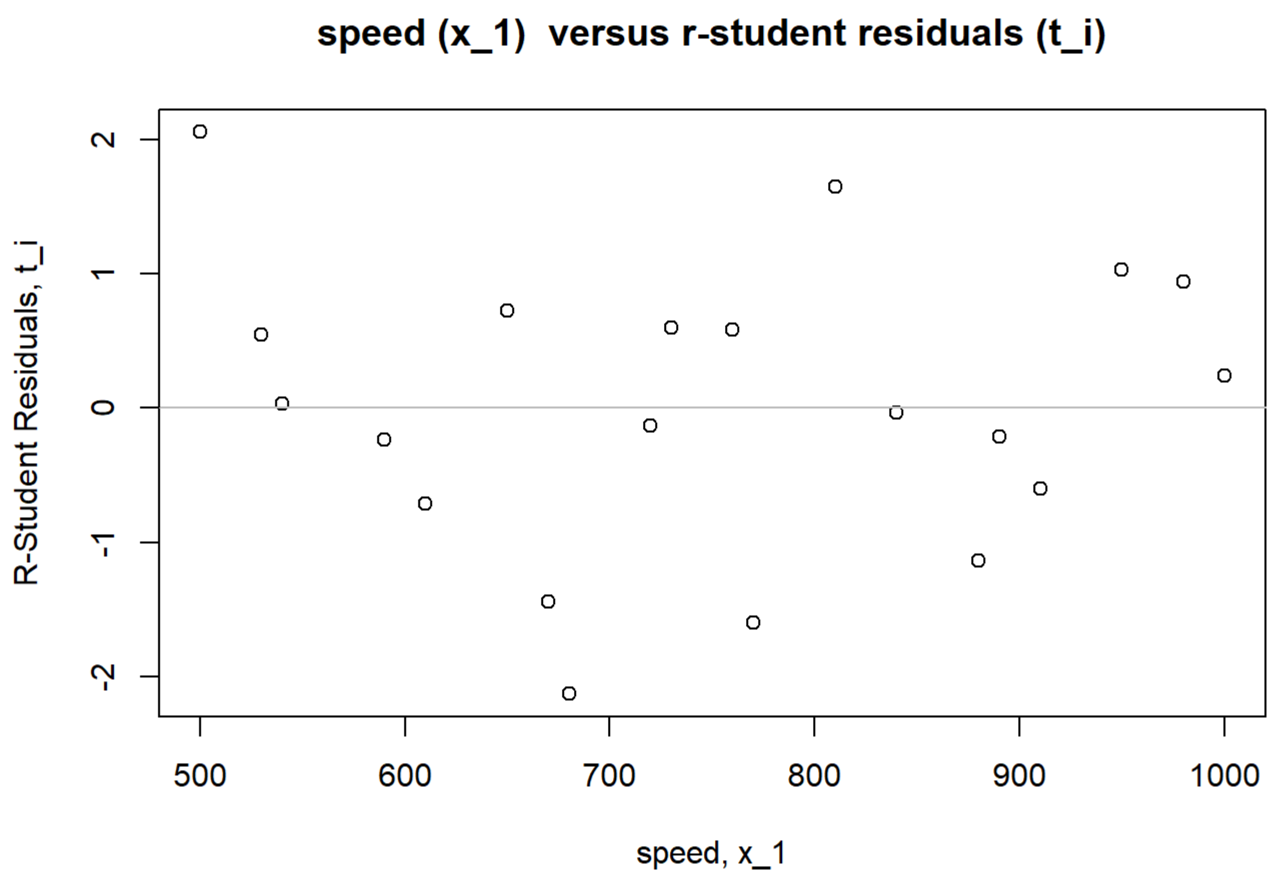
$$F_0 = 76.7458491$$

Check model adequacy with residual plots

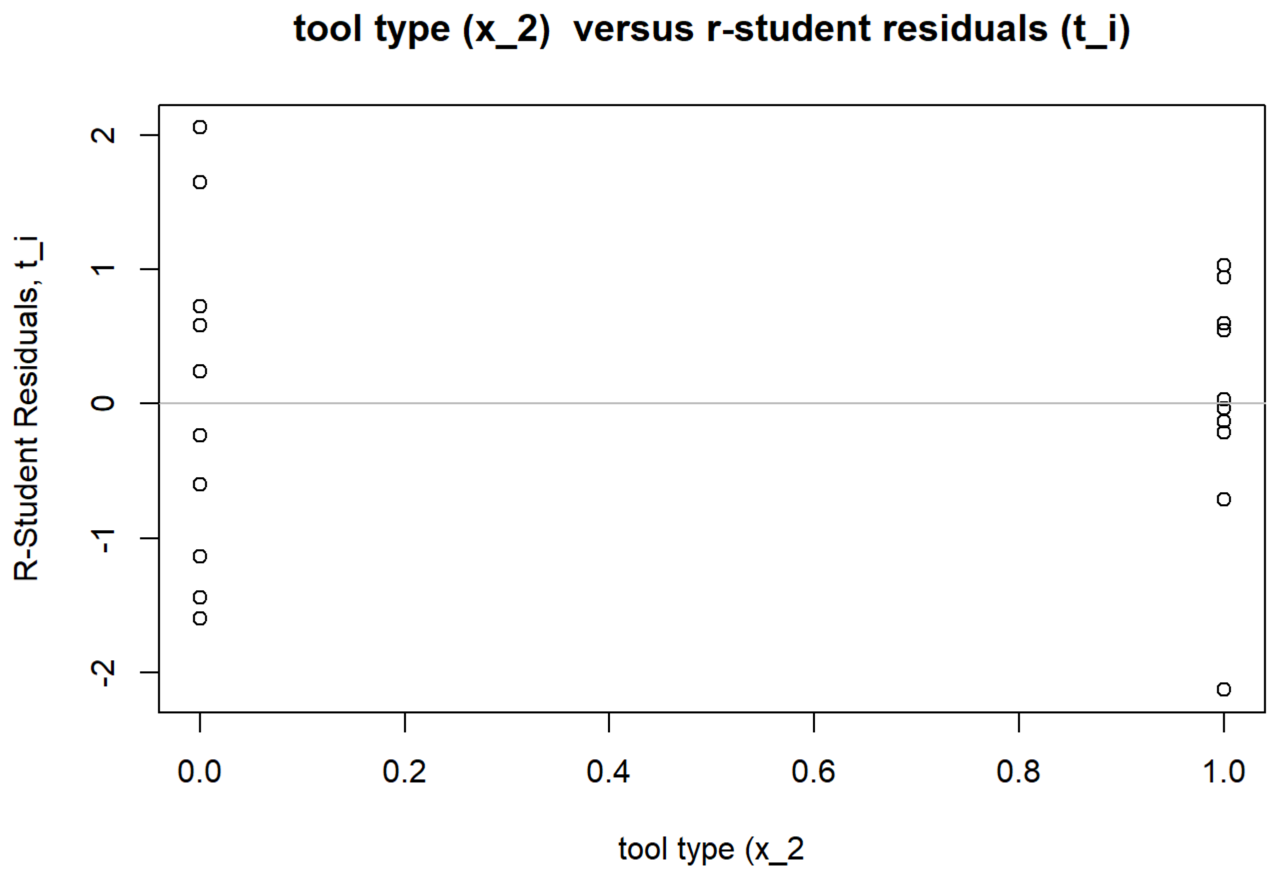
```
# fitted values (y_hat) versus R-student residuals (t_i)
plot(model.81$fitted.values, rstudent(model.81),
     main = "fitted values (y_hat) versus r-student residuals (t_i)",
     xlab = "fitted values, y_hat",
     ylab = "R-Student Residuals, t_i")

for (pltpts in seq(1:length(y_i))){

  if (Tool_Type[pltpts] == "A") {
    points(model.81$fitted.values[pltpts],
           rstudent(model.81)[pltpts],
           col = "green")}
  else if (Tool_Type[pltpts] == "B") {
    points(model.81$fitted.values[pltpts],
           rstudent(model.81)[pltpts],
           col = "red")}
}

legend("topright", legend=c("A", "B"),
       col=c("green", "red"),
       pch=c(1,1))
abline(0,0,col="gray")
```



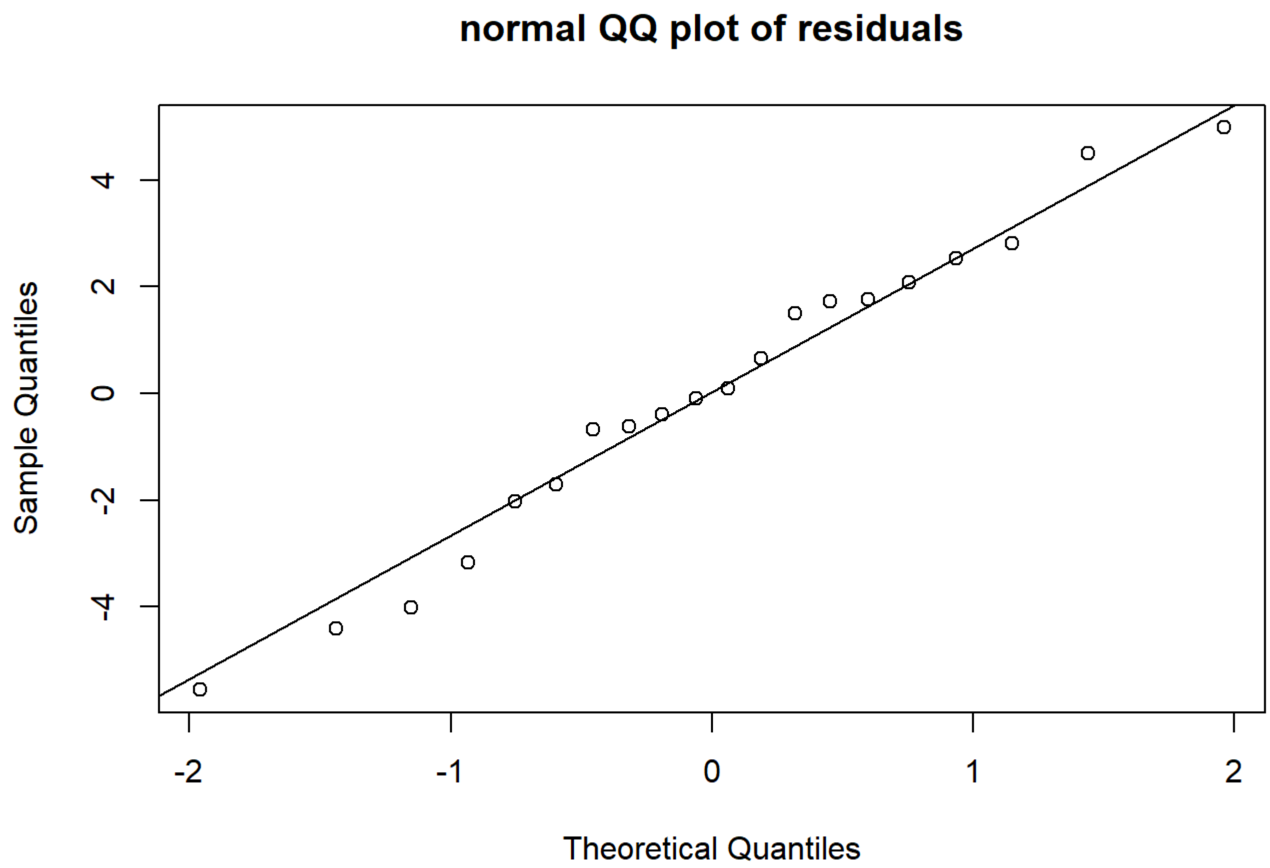
```
# speed versus rstudent residuals
plot(x_1, rstudent(model.81),
     main = "speed (x_1)  versus r-student residuals (t_i)",
     xlab = "speed, x_1",
     ylab = "R-Student Residuals, t_i")
abline(0,0,col="gray")
```



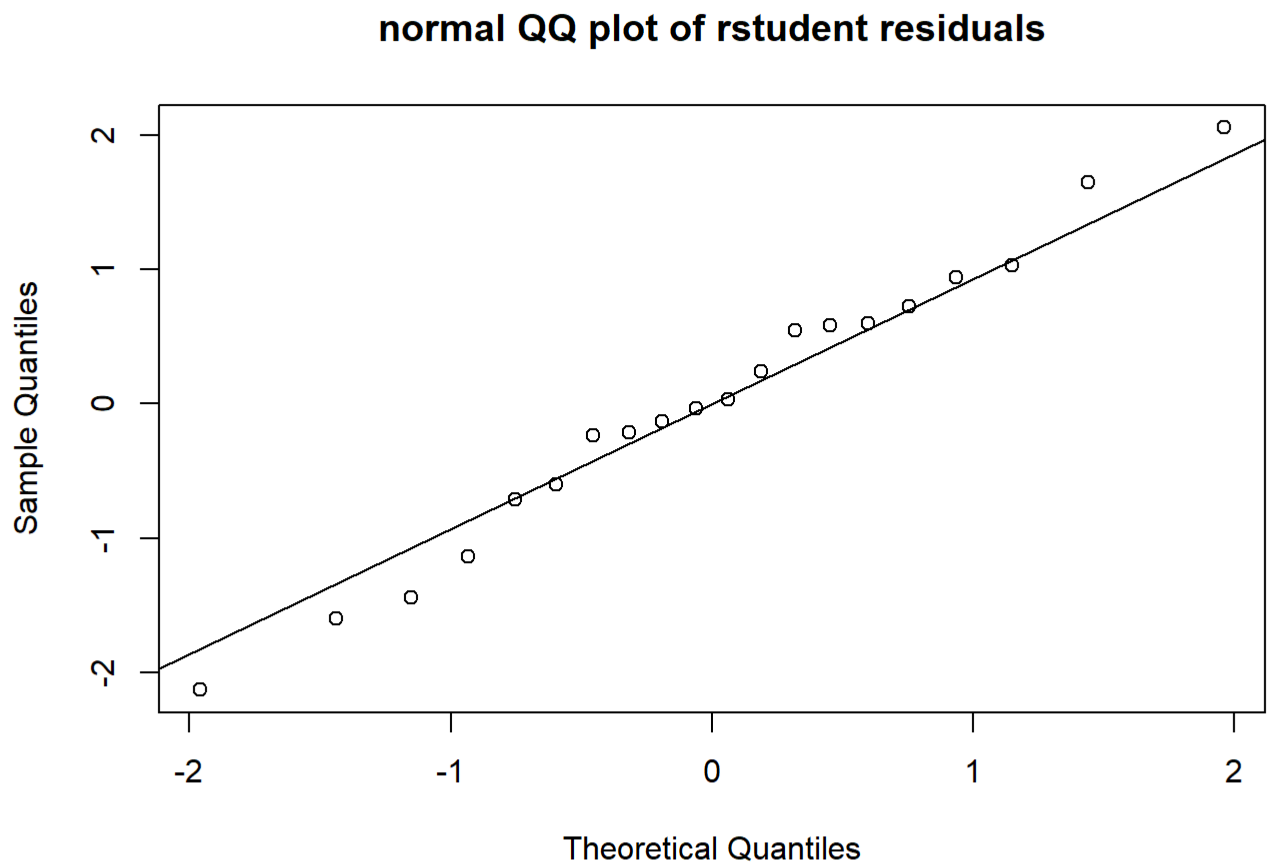

```
# speed versus rstudent residuals
plot(x_2, rstudent(model.81),
     main = "tool type (x_2)  versus r-student residuals (t_i)",
     xlab = "tool type (x_2",
     ylab = "R-Student Residuals, t_i")
abline(0,0,col="gray")
```




```
#Assess normality of the observations as a whole using the residuals.
qqnorm(model.81$residuals,main="normal QQ plot of residuals")
qqline(model.81$residuals)
```




```
#Assess normality of the observations as a whole using the r-student residuals.
qqnorm(rstudent(model.81),main="normal QQ plot of rstudent residuals")
qqline(rstudent(model.81))
```



No issues with the normal probability plot. Constant variance with the fitted values and regression variables versus residuals; the seems to be no issues with the model fits.

Example 8.2 (p. 267-268)

Fit a model were ToolType is expected to influence both slope and intercept; compare the two models using the Partial F-test

Add column to dataframe for Interaction

```
x_12 <- x_1 * x_2
Ex81$x_12 <- x_12

Ex81$x_12 <- c(x_12)

# Output data to make sure it reads properly
out <- as.data.frame(c(Ex81))
colnames(out) <- c("Obs",
                  "y_i",
                  "x_1",
                  "Tool_Type",
                  "x_2",
                  "x_12")
tab <- (xtable(out, digits=c(0,0,2,0,0,0,0)))
print(tab, type="html")
```

	Obs	y_i	x_1	Tool_Type	x_2	x_12
1	1	18.73	610	A	1	610
2	2	14.52	950	A	1	950
3	3	17.43	720	A	1	720
4	4	14.54	840	A	1	840
5	5	13.44	980	A	1	980
6	6	24.39	530	A	1	530
7	7	13.34	680	A	1	680
8	8	22.71	540	A	1	540
9	9	12.68	890	A	1	890
10	10	19.32	730	A	1	730
11	11	30.16	670	B	0	0
12	12	27.09	770	B	0	0
13	13	25.40	880	B	0	0
14	14	26.05	1000	B	0	0
15	15	33.49	760	B	0	0
16	16	35.62	590	B	0	0
17	17	26.07	910	B	0	0
18	18	36.78	650	B	0	0
19	19	34.95	810	B	0	0
20	20	43.67	500	B	0	0

Create full model; compare to p. 267

```
model.82 <- lm(y_i ~ x_1 + x_2 + x_12)

summary(model.82)
```

```
##
## Call:
## lm(formula = y_i ~ x_1 + x_2 + x_12)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.1750 -1.4999  0.4849  1.7830  4.8652
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  56.745353   4.934565  11.500  3.8e-09 ***
## x_1          -0.032914   0.006425  -5.123 0.000102 ***
## x_2         -23.970593   6.768973  -3.541 0.002716 **
## x_12          0.011944   0.008842   1.351 0.195533
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.968 on 16 degrees of freedom
## Multiple R-squared:  0.9105, Adjusted R-squared:  0.8937
## F-statistic: 54.25 on 3 and 16 DF,  p-value: 1.319e-08
```

The least squares fit equation is:

$$\hat{y} = 56.7453533 + (-0.0329143)x_1 + (-23.9705934)x_2 + (0.011944)x_1x_2$$

$$R^2 = 0.9104965$$

$$R^2_{adj} = 0.8937147$$

$$F_0 = 54.2546849$$

Calculate SS and df for Regression and Residual for full model; compare to Table 8.3, p. 268

```
x <- anova(model.82)

SS_Reg <- sum(x$`Sum Sq`[1:3])
df_Reg <- sum(x$Df[1:3])

SS_Res <- x$`Sum Sq`[4]
df_Res <- x$Df[4]
```

Sum of Squares (SS) Regression: 1434.1123507 and Degrees of Freedom (df) Regression: 3

Sum of Squares (SS) Residual: 140.9758293 and Degrees of Freedom (df) Residual: 16

Test significance of interaction term using Partial F-test. Use alpha = 0.05 as the significance level; compare to values on p. 268

```
# full model including interaction term
model.full <- lm(y_i ~ x_1 + x_2 + x_12)

# reduced model excluding interaction term
model.reduced <- lm(y_i ~ x_1 + x_2)

# anova -- comparision of reduced to full model
anova(model.reduced, model.full)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	17	157.0546	NA	NA	NA	NA
2	16	140.9758	1	16.07873	1.82485	0.195533

2 rows

```
# F crit
alpha <- 0.05
df_SS_R <- anova(model.reduced, model.full)$'Df'[2]
df_SS_Res <- anova(model.reduced, model.full)$'Res.Df'[2]

F_crit <- qf(1-alpha,df_SS_R,df_SS_Res)

F_0 <- anova(model.reduced, model.full)$'F'[2]
p_value <-  anova(model.reduced, model.full)$'Pr(>F)'[2]
```

The appropriate hypotheses are

$$H_0 : \beta_3 = 0, H_1 : \beta_3 \neq 0$$

To test

$$H_0 : \beta_3 = 0$$

form the test statistic

$$F_0 = \frac{SS_{reg}(\beta_3|\beta_2, \beta_1, \beta_0)/1}{MS_{Res}} = \frac{16.0787331}{8.8109893} = 1.8248499$$

and since $F_{0.05,1,16} = 4.4939985$, we have $F_0 < F_{0.05,1,16}$ and a p-value of $P = 0.195533$ which is greater than the 0.05 threshold. We don't reject the null hypothesis and conclude that the interaction term is not significant.

Test hypothesis that the regression lines for each tool type are identical; compare to p. 268.

```
# full model including interaction term
model.full <- lm(y_i ~ x_1 + x_2 + x_12)

# reduced model excluding interaction term
model.reduced <- lm(y_i ~ x_1)

# anova -- comparision of reduced to full model
anova(model.reduced, model.full)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	18	1282.0828	NA	NA	NA	NA
2	16	140.9758	2	1141.107	64.75476	2.137119e-08

2 rows

```
# F crit
alpha <- 0.05
df_SS_R <- anova(model.reduced, model.full)$'Df'[2]
df_SS_Res <- anova(model.reduced, model.full)$'Res.Df'[2]

F_crit <- qf(1-alpha,df_SS_R,df_SS_Res)
```

To test the hypothesis that the two regression lines are identical,

$$H_0 : \beta_2 = \beta_3 = 0$$

use the statistic

$$F_0 = \frac{SS_R(\beta_2, \beta_3|\beta_1, \beta_0/2}{MS_{Res}} = \frac{1141.1069474/2}{8.8109893} = 64.7547571$$

and since $F_{0.05,2,16} = 3.6337235$, we have $F_0 > F_{0.05,2,16}$ and also the P statistic is $P = 2.1371192 \times 10^{-8}$. We reject the null hypothesis and conclude that the two regression lines are not identical.