Lesson 9 Lecture Example

Rick Davila

5/16/2020

Lesson 9 - Install packages

Install necessary packages using library()

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

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

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

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

```
library(psych) # For geometric mean in Example 5.3
```

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

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

rm(list = ls())
```

```
exL <- read.xlsx("data-table-B09.xlsx",
                   sheetIndex = 1,
                   colIndex = c(1,2,3,4,5),
                   as.data.frame = TRUE,
                   header = TRUE)
```

Read data file (data-table-B09.xlsx)

```
names(exL) <- c("fluid_vel", "viscosity","mesh_open","fluid_gas_vel","pressure_drop")
attach(exL)
```

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

```
out <- as.data.frame(c(exL))
colnames(out) <- c("fluid_vel", "viscosity","mesh_open","fluid_gas_vel","pressure_drop")
tab <- (xtable(out, digits=c(0,2,1,2,3,1)))
print(tab, type="html")
```

Output data to make sure it reads properly fluid_vel

viscosity

mesh_open

fluid_gas_vel

pressure_drop

1

2.14

10.0

0.34

1.000

28.9

2

4.14

10.0

0.34

1.000

31.0

3

8.15

10.0

0.34

1.000

26.4

4

2.14

10.0

0.34

0.246

27.2

5

4.14

10.0

0.34

0.379

26.1

6

8.15

10.0

0.34

0.474

23.2

7

2.14

10.0

0.34

0.141

19.7

8

4.14

10.0

0.34

0.234

22.1

9

8.15

10.0

0.34

0.311

22.8

10

2.14

10.0

0.34

0.076

29.2

11

4.14

10.0

0.34

0.132

23.6

12

8.15

10.0

0.34

0.184

23.6

13

2.14

2.6

0.34

0.679

24.2

14

4.14

2.6

0.34

0.804

22.1

15

8.15

2.6

0.34

0.890

20.9

16

2.14

2.6

0.34

0.514

17.6

17

4.14

2.6

0.34

0.672

15.7

18

8.15

2.6

0.34

0.801

15.8

19

2.14

2.6

0.34

0.346

14.0

20

4.14

2.6

0.34

0.506

17.1

21

8.15

2.6

0.34

0.669

18.3

22

2.14

2.6

0.34

1.000

33.8

23

4.14

2.6

0.34

1.000

31.7

24

8.15

2.6

0.34

1.000

28.1

25

5.60

1.2

0.34

0.848

18.1

26

5.60

1.2

0.34

0.737

16.5

27

5.60

1.2

0.34

0.651

15.4

28

5.60

1.2

0.34

0.554

15.0

29

4.30

2.6

0.34

0.748

19.1

30

4.30

2.6

0.34

0.682

16.2

31

4.30

2.6

0.34

0.524

16.3

32

4.30

2.6

0.34

0.472

15.8

33

4.30

2.6

0.34

0.398

15.4

34

5.60

10.1

0.25

0.789

19.2

35

5.60

10.1

0.25

0.677

8.4

36

5.60

10.1

0.25

0.590

15.0

37

5.60

10.1

0.25

0.523

12.0

38

5.60

10.1

0.34

0.789

21.9

39

5.60

10.1

0.34

0.677

21.3

40

5.60

10.1

0.34

0.590

21.6

41

5.60

10.1

0.34

0.523

19.8

42

4.30

10.1

0.34

0.741

21.6

43

4.30

10.1

0.34

0.617

17.3

44

4.30

10.1

0.34

0.524

20.0

45

4.30

10.1

0.34

0.457

18.6

46

2.40

10.1

0.34

0.615

22.1

47

2.40

10.1

0.34

0.473

14.7

48

2.40

10.1

0.34

0.381

15.8

49

2.40

10.1

0.34

0.320

13.2

50

5.60

10.1

0.55

0.789

30.8

51

5.60

10.1

0.55

0.677

27.5

52

5.60

10.1

0.55

0.590

25.2

53

5.60

10.1

0.55

0.523

22.8

54

2.14

112.0

0.34

0.680

41.7

55

4.14

112.0

0.34

0.803

33.7

56

8.15

112.0

0.34

0.889

29.7

57

2.14

112.0

0.34

0.514

41.8

58

4.14

112.0

0.34

0.672

37.1

59

8.15

112.0

0.34

0.801

40.1

60

2.14

112.0

0.34

0.306

42.7

61

4.14

112.0

0.34

0.506

48.6

62

8.15

112.0

0.34

0.668

42.4

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

'data.frame': 62 obs. of 5 variables: $ fluid_vel : num 2.14 4.14 8.15 2.14 4.14 8.15 2.14 4.14 8.15 2.14 … $ viscosity : num 10 10 10 10 10 10 10 10 10 10 … $ mesh_open : num 0.34 0.34 0.34 0.34 0.34 0.34 0.34 0.34 0.34 0.34 … $ fluid_gas_vel: num 1 1 1 0.246 0.379 0.474 0.141 0.234 0.311 0.076 … $ pressure_drop: num 28.9 31 26.4 27.2 26.1 23.2 19.7 22.1 22.8 29.2 …

```
dim(exL)
```

[1] 62 5

```r
model <- lm(pressure_drop ~ fluid_vel + viscosity + mesh_open + fluid_gas_vel)

summary(model)
```

create multiple least squares model Call: lm(formula = pressure_drop ~ fluid_vel + viscosity + mesh_open + fluid_gas_vel)

Residuals: Min 1Q Median 3Q Max -9.9958 -3.3092 -0.2419 3.3924 10.5668

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.89453 4.32508 1.363 0.17828
fluid_vel -0.47790 0.34002 -1.406 0.16530
viscosity 0.18271 0.01718 10.633 3.78e-15 ***mesh_open 35.40284 11.09960 3.190 0.00232***
fluid_gas_vel 5.84391 2.90978 2.008 0.04935
— Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.014 on 57 degrees of freedom Multiple R-squared: 0.6914, Adjusted R-squared: 0.6697 F-statistic: 31.92 on 4 and 57 DF, p-value: 5.818e-14

```r
xtable(summary(model))
```

% latex table generated in R 4.0.2 by xtable 1.8-4 package % Mon Mar 15 21:54:34 2021

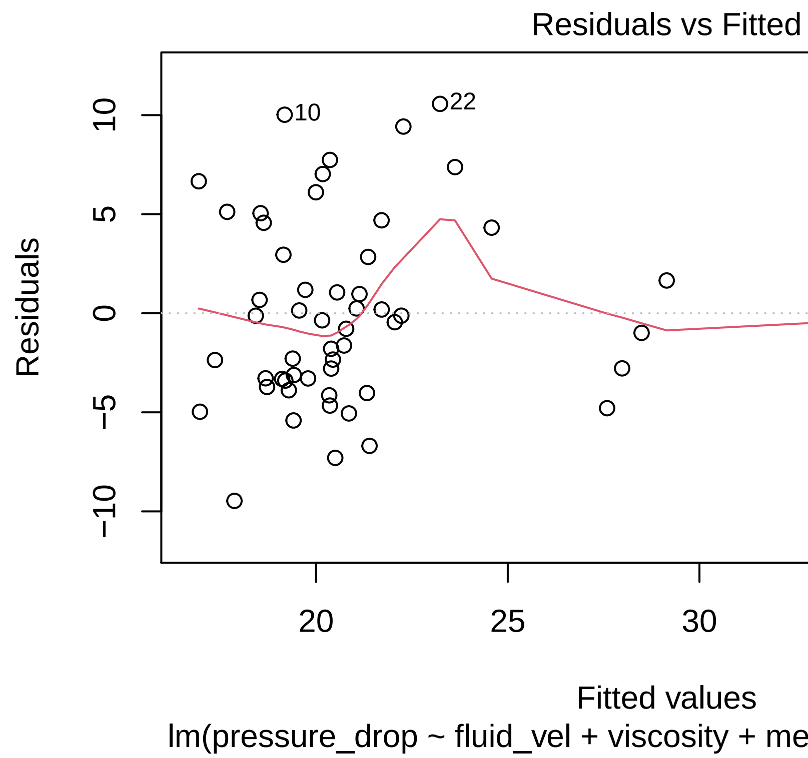
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	5.8945	4.3251	1.36	0.1783
fluid_vel	-0.4779	0.3400	-1.41	0.1653
viscosity	0.1827	0.0172	10.63	0.0000
mesh_open	35.4028	11.0996	3.19	0.0023
fluid_gas_vel	5.8439	2.9098	2.01	0.0494

```r
xtable(anova(model))
```

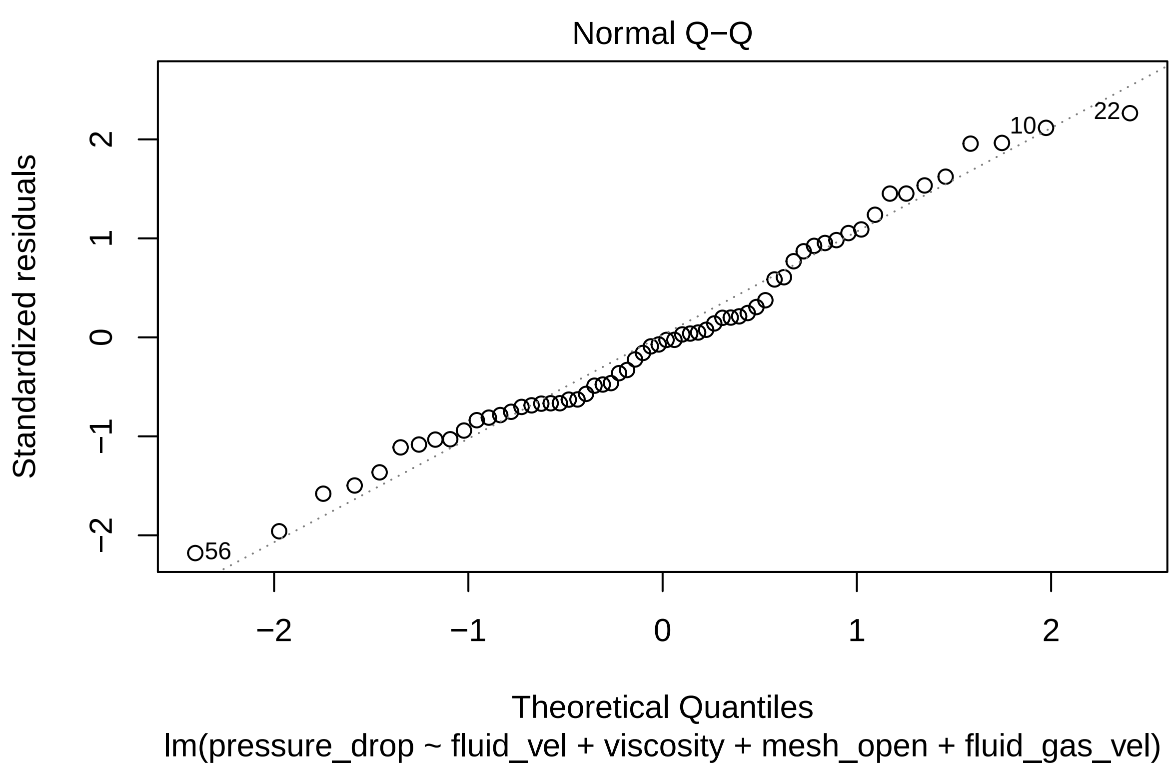
% latex table generated in R 4.0.2 by xtable 1.8-4 package % Mon Mar 15 21:54:34 2021

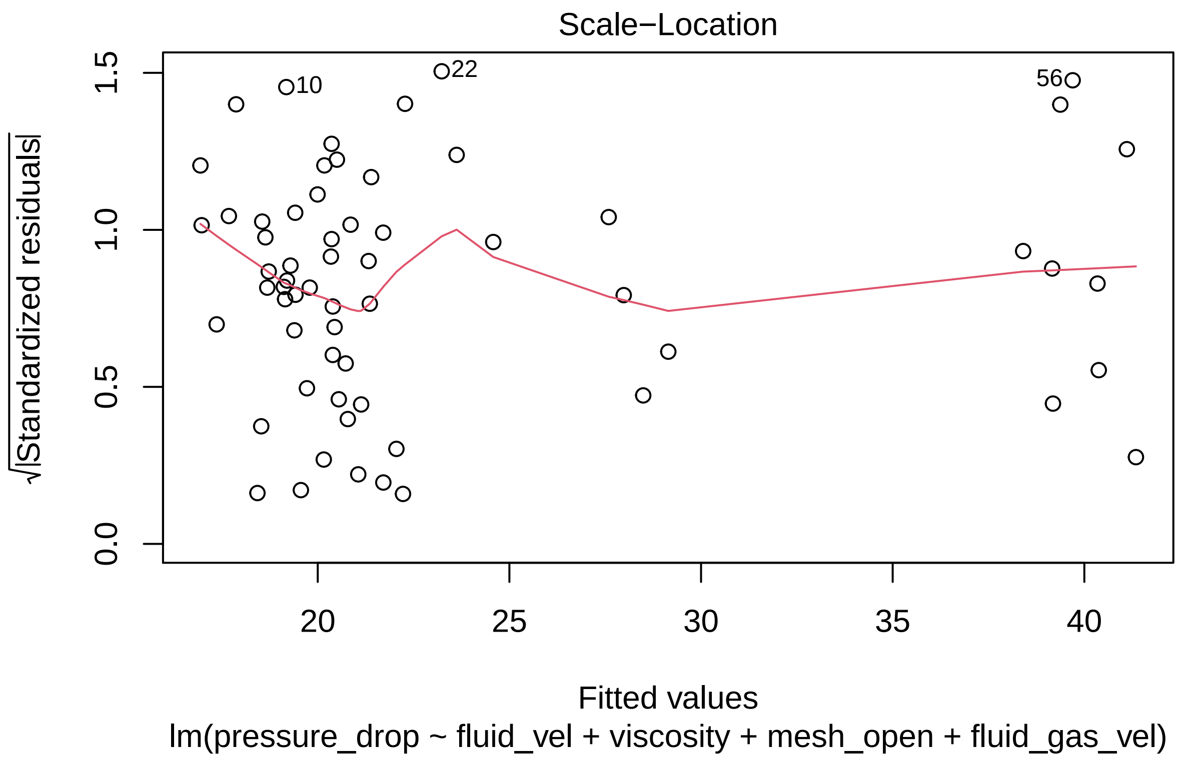
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
fluid_vel	1	9.60	9.60	0.38	0.5391
viscosity	1	2839.78	2839.78	112.97	0.0000
mesh_open	1	258.95	258.95	10.30	0.0022
fluid_gas_vel	1	101.39	101.39	4.03	0.0494
Residuals	57	1432.79	25.14		

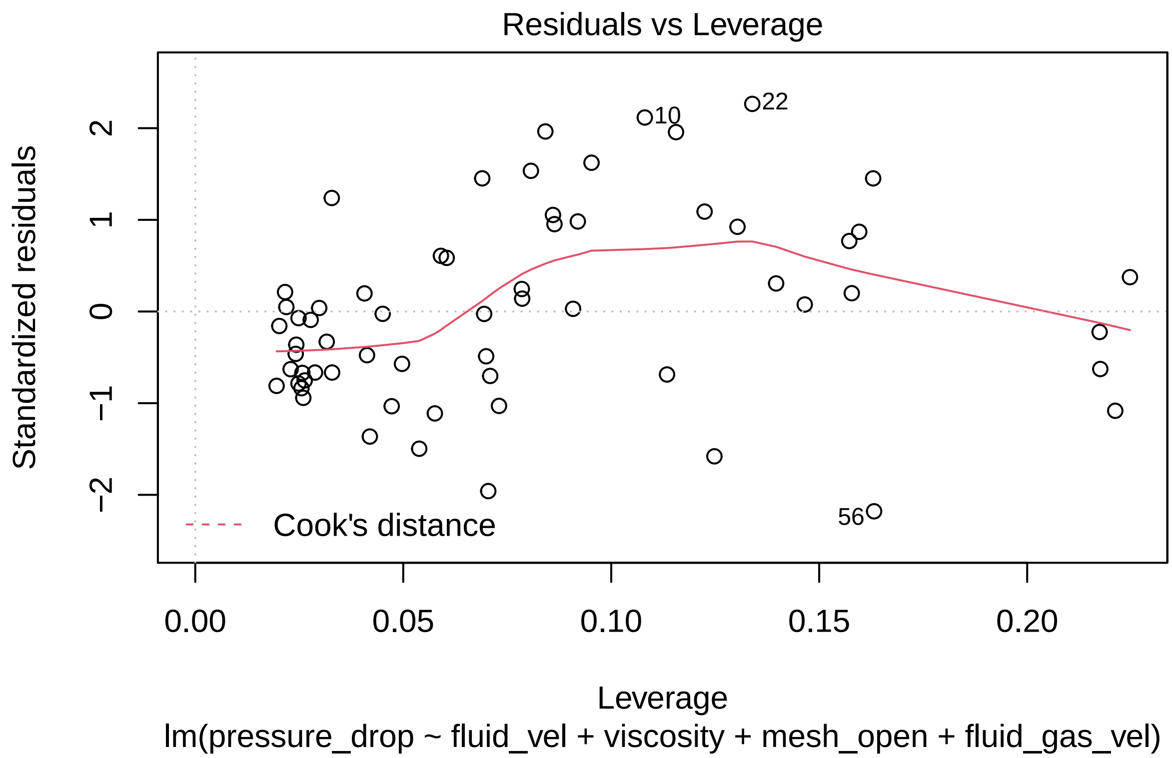
```r
plot(model)
```



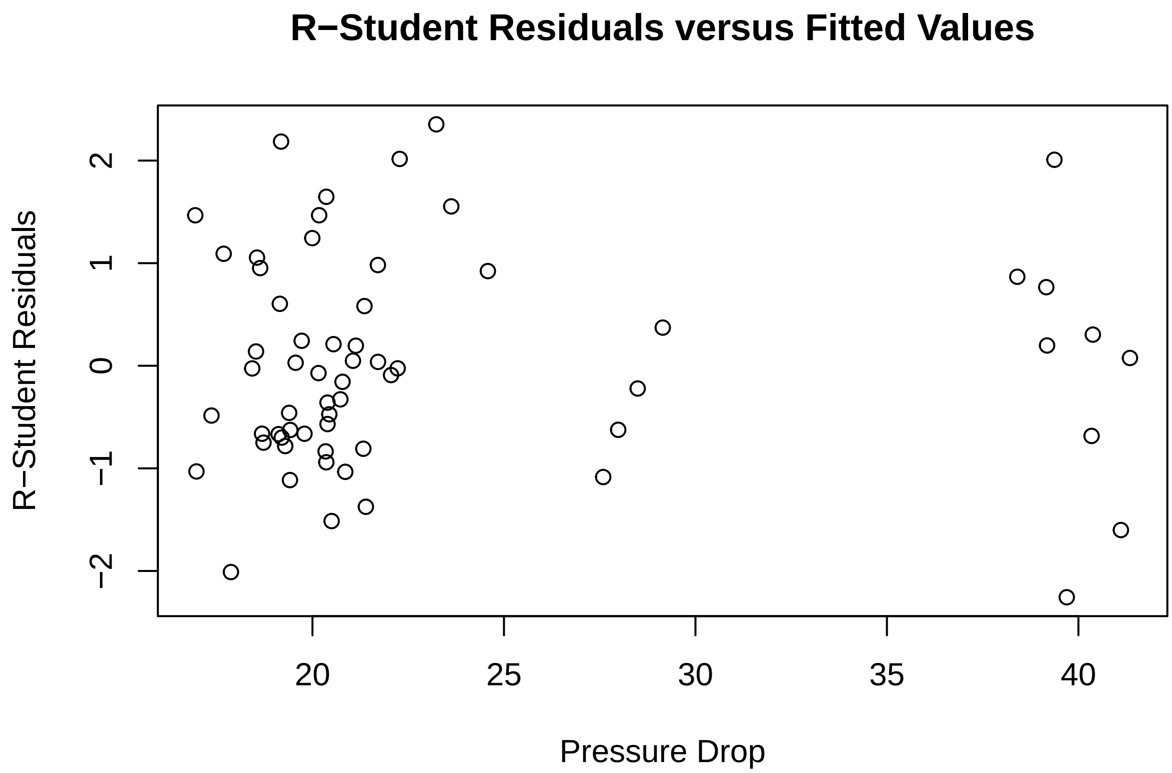
make a plot of the multiple least squares model



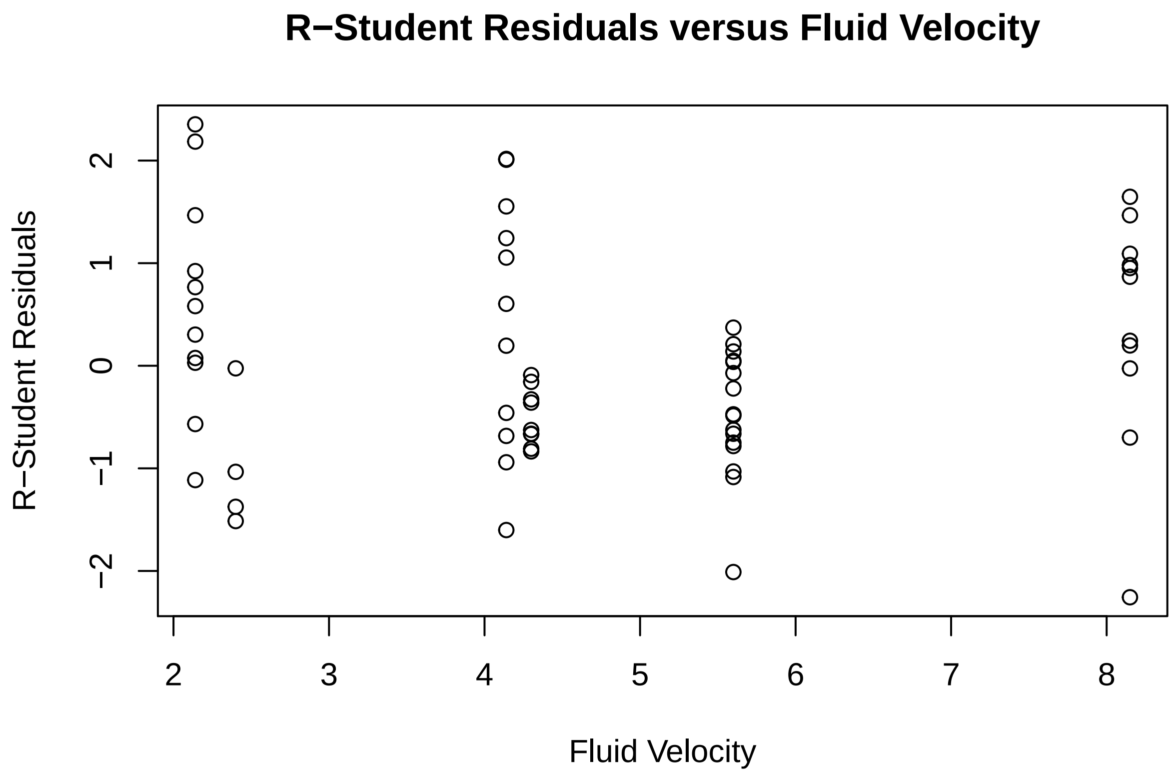

Scale−Location
0.0
0.5
1.0
1.5
√|Standardized residuals|
10
22
56
20
25
30
35
40
Fitted values
lm(pressure_drop ~ fluid_vel + viscosity + mesh_open + fluid_gas_vel)



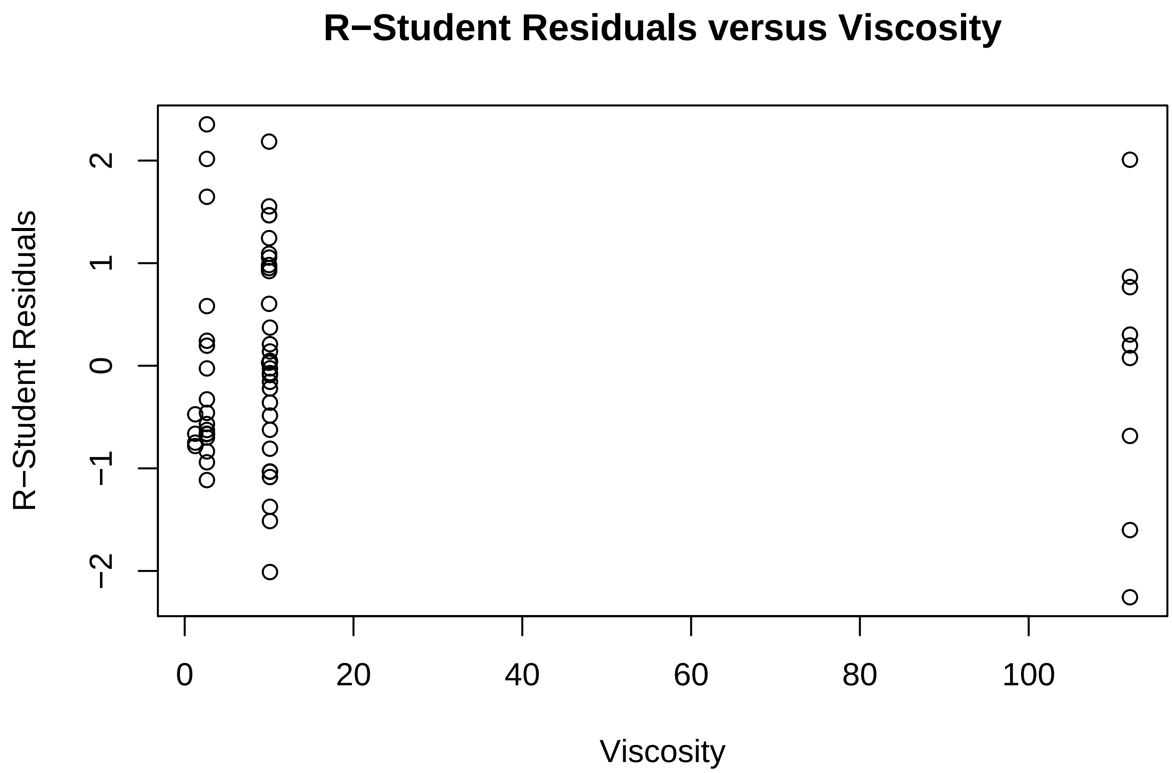

```
# Let's look at a plot of the R-student residuals versus the fitted values

R_Student_Residuals <- rstudent(model)
y_hat <- model$fitted.values
plot(y_hat, R_Student_Residuals, ylab = "R-Student Residuals", xlab = "Pressure Drop", main = "R-Studen
```



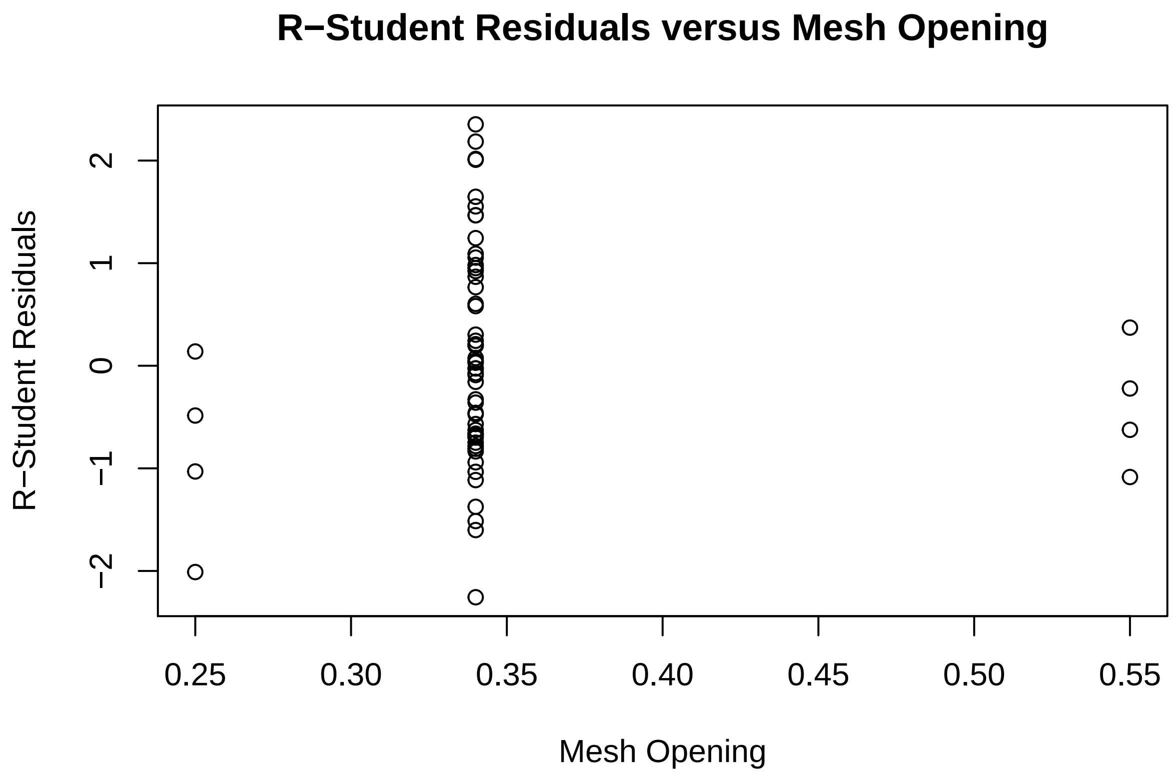

```
plot(fluid_vel, R_Student_Residuals, ylab = "R-Student Residuals", xlab = "Fluid Velocity", main = "R-St
```



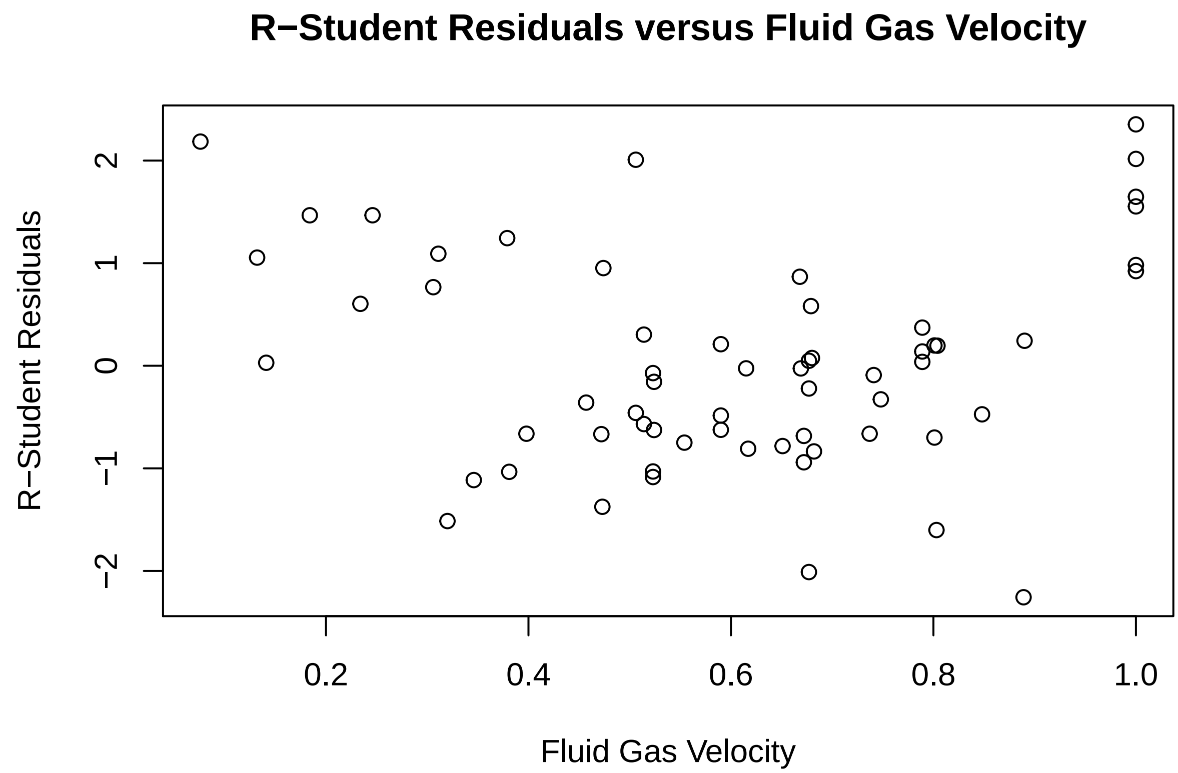

```
plot(viscosity, R_Student_Residuals, ylab = "R-Student Residuals", xlab = "Viscosity", main = "R-Studen
```




```
plot(mesh_open, R_Student_Residuals, ylab = "R-Student Residuals", xlab = "Mesh Opening", main = "R-Stud
```




```
plot(fluid_gas_vel, R_Student_Residuals, ylab = "R-Student Residuals", xlab = "Fluid Gas Velocity", mai
```



Since there's a huge gap in the residuals versus viscosity data, we'll try a natural log on the regressor. Redo the exercise above, but, with the lm() using log of vicosity.

```
model <- lm(pressure_drop ~ fluid_vel + log(viscosity) + mesh_open + fluid_gas_vel)

summary(model)
```

Call: lm(formula = pressure_drop ~ fluid_vel + log(viscosity) + mesh_open + fluid_gas_vel)

Residuals: Min 1Q Median 3Q Max -13.4818 -3.5114 -0.2417 2.9716 12.8961

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.8207 4.9007 -0.167 0.86760
fluid_vel -0.5492 0.3739 -1.469 0.14739
log(viscosity) 5.0960 0.5574 9.143 9.03e-13 ***mesh_open 28.0374 12.1975 2.299 0.02521***
fluid_gas_vel 10.3162 3.2096 3.214 0.00215 — Signif. codes: 0 '***' 0.001 '**' *0.01* '' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.514 on 57 degrees of freedom Multiple R-squared: 0.6267, Adjusted R-squared: 0.6005 F-statistic: 23.92 on 4 and 57 DF, p-value: 1.201e-11

```
xtable(summary(model))
```

% latex table generated in R 4.0.2 by xtable 1.8-4 package % Mon Mar 15 21:54:34 2021

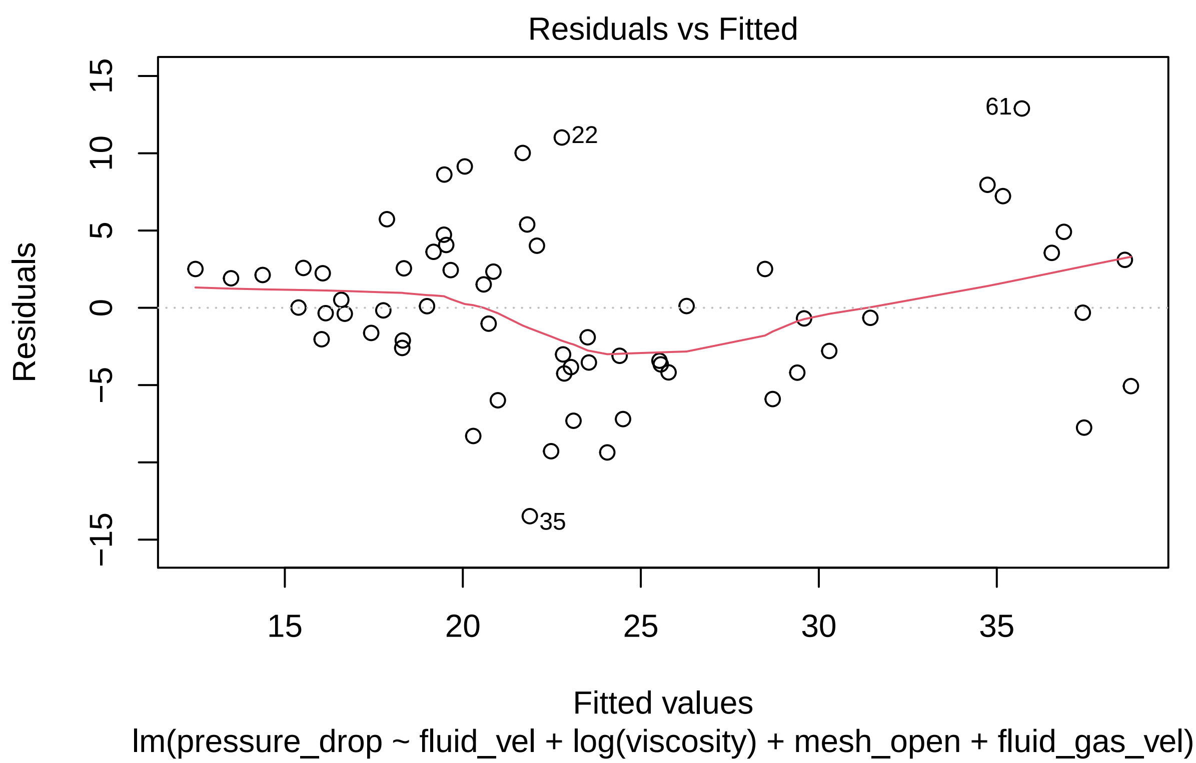
	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	-0.8207	4.9007	-0.17	0.8676
fluid_vel	-0.5492	0.3739	-1.47	0.1474
log(viscosity)	5.0960	0.5574	9.14	0.0000
mesh_open	28.0374	12.1975	2.30	0.0252
fluid_gas_vel	10.3162	3.2096	3.21	0.0022

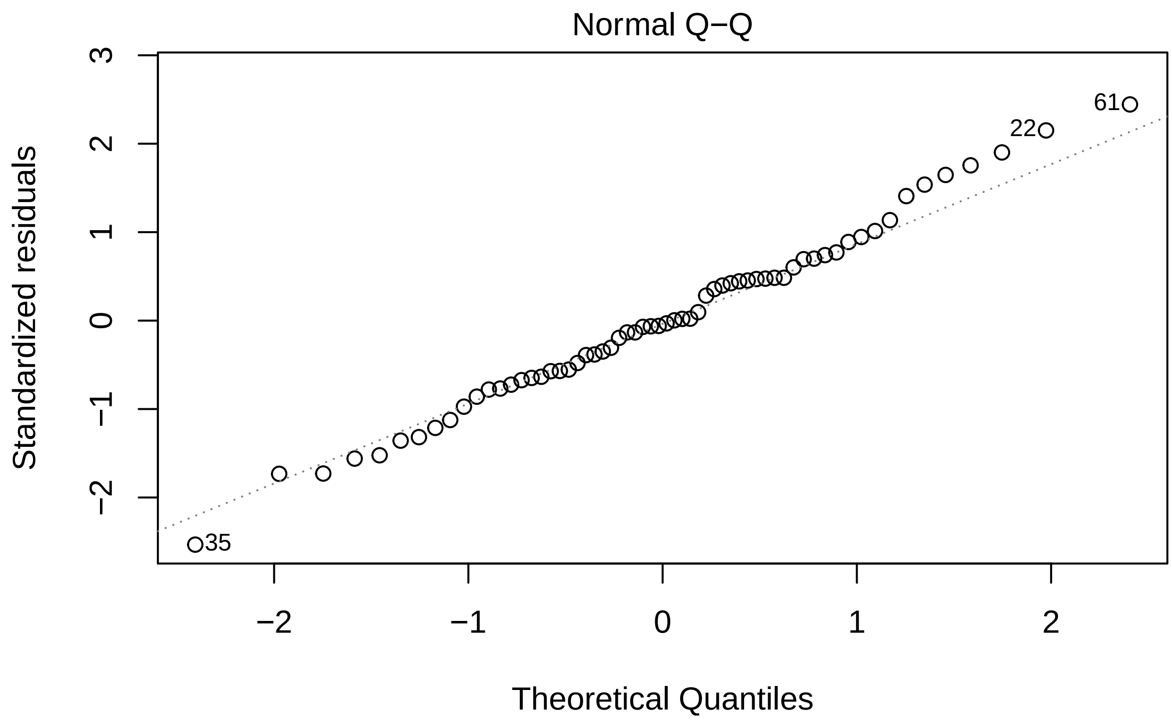
```
xtable(anova(model))
```

% latex table generated in R 4.0.2 by xtable 1.8-4 package % Mon Mar 15 21:54:34 2021

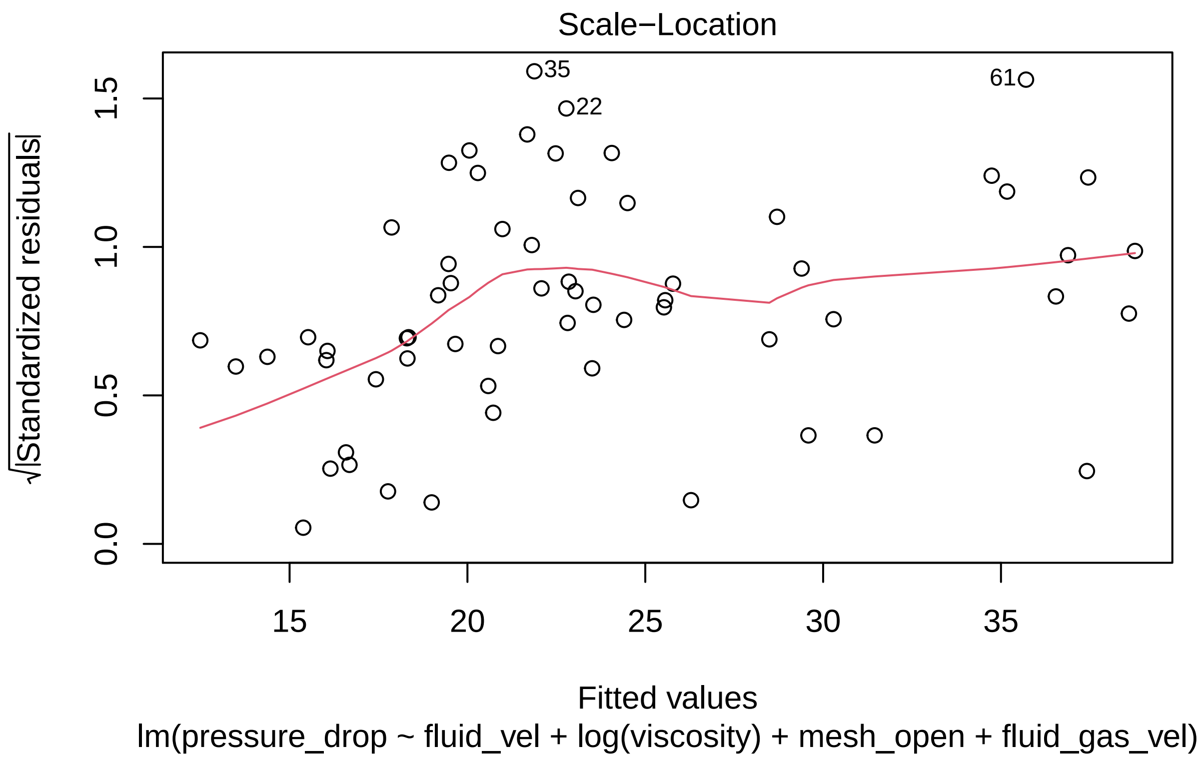
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
fluid_vel	1	9.60	9.60	0.32	0.5765
log(viscosity)	1	2420.86	2420.86	79.62	0.0000
mesh_open	1	164.82	164.82	5.42	0.0235
fluid_gas_vel	1	314.12	314.12	10.33	0.0022
Residuals	57	1733.11	30.41		

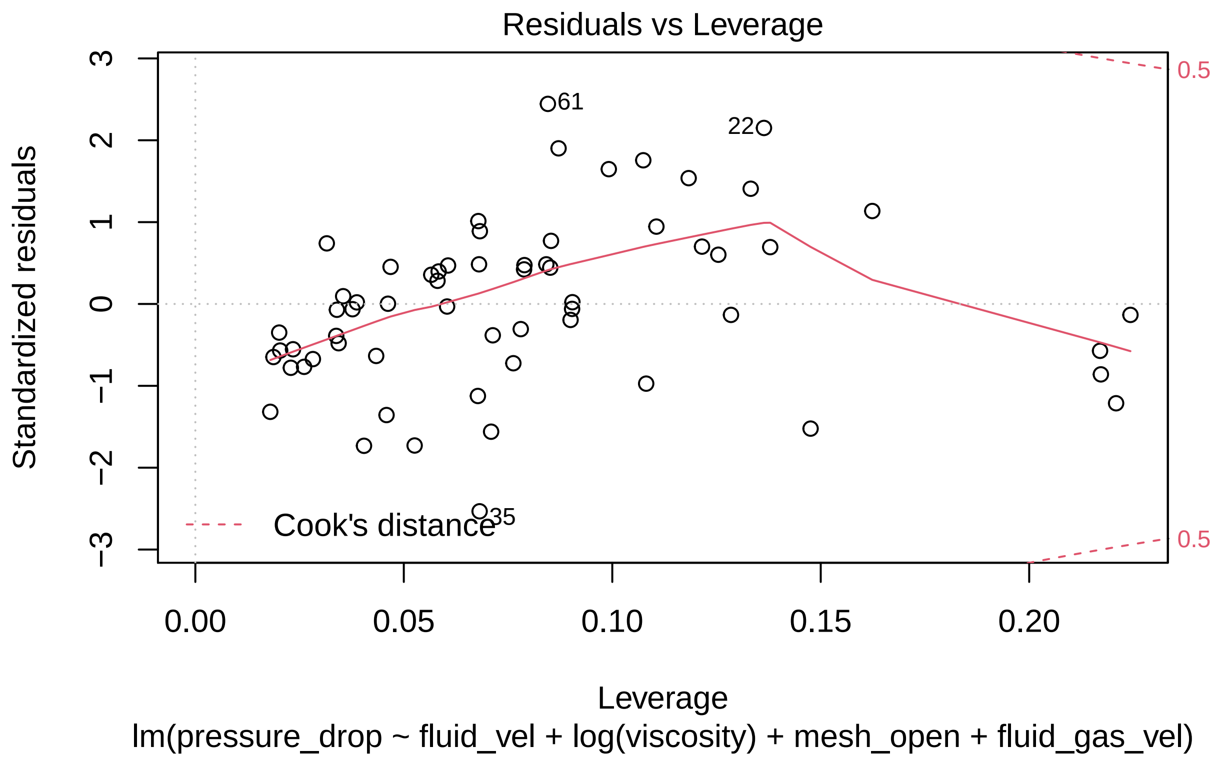
```
plot(model)
```



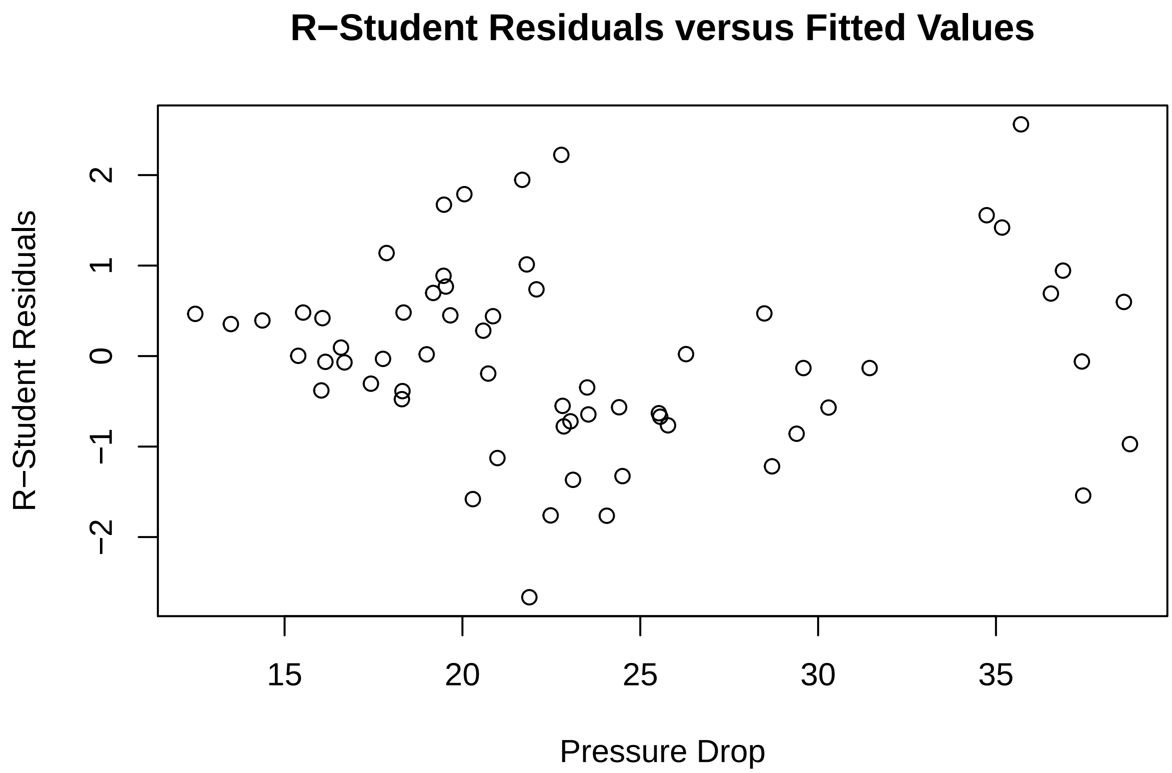


lm(pressure_drop ~ fluid_vel + log(viscosity) + mesh_open + fluid_gas_vel)

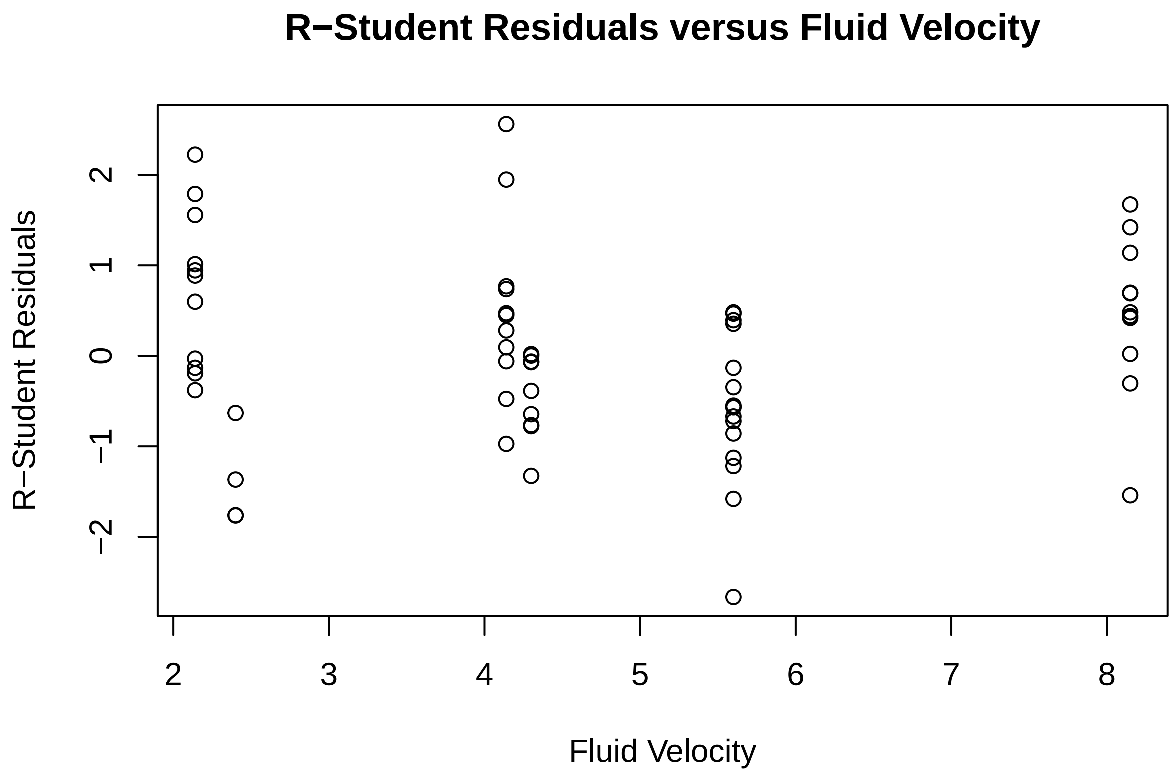

Scale−Location
35
22
61
1.5
1.0
0.5
0.0
√|Standardized residuals|
15
20
25
30
35
Fitted values
lm(pressure_drop ~ fluid_vel + log(viscosity) + mesh_open + fluid_gas_vel)



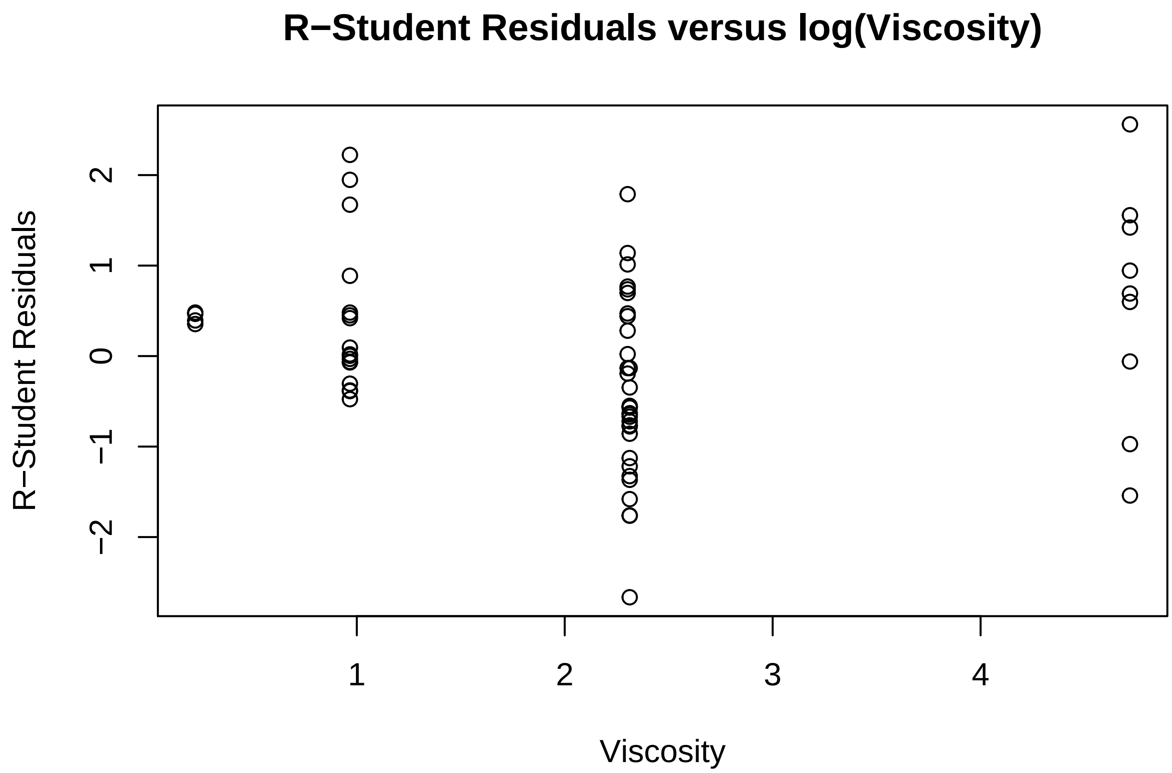

```
R_Student_Residuals <- rstudent(model)
y_hat <- model$fitted.values
plot(y_hat, R_Student_Residuals, ylab = "R-Student Residuals", xlab = "Pressure Drop", main = "R-Studen
```



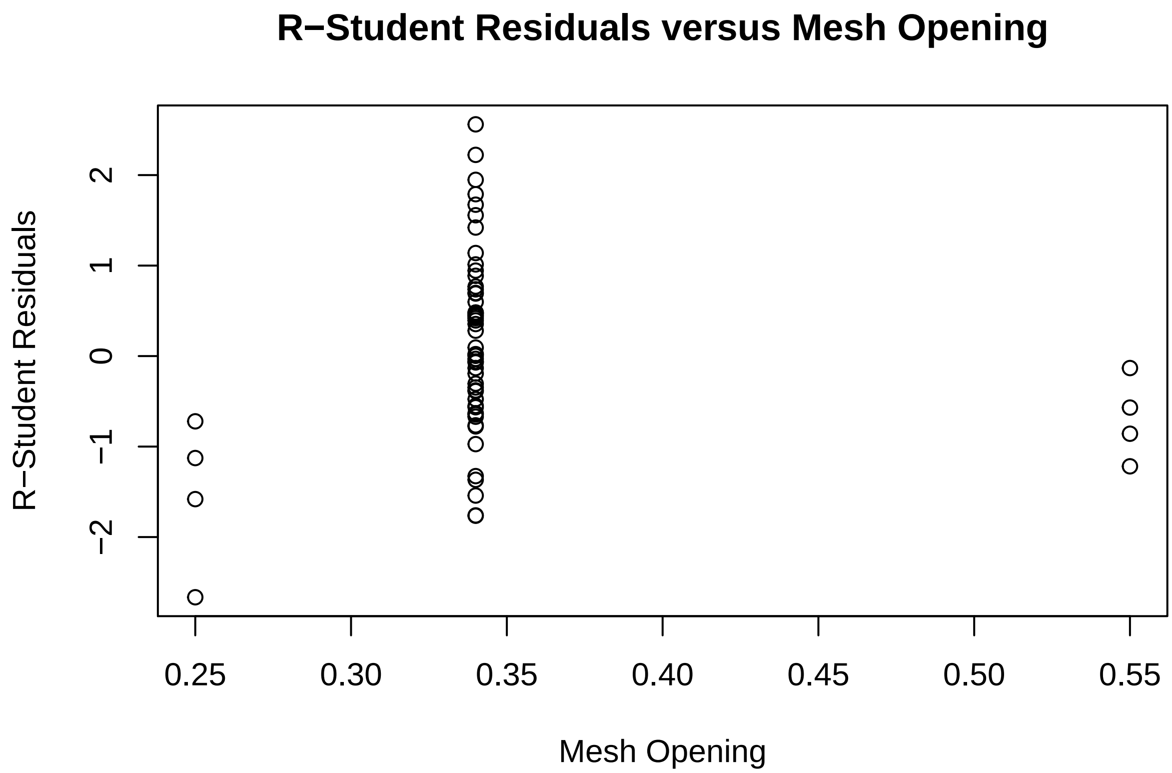

```
plot(fluid_vel, R_Student_Residuals, ylab = "R-Student Residuals", xlab = "Fluid Velocity", main = "R-St
```



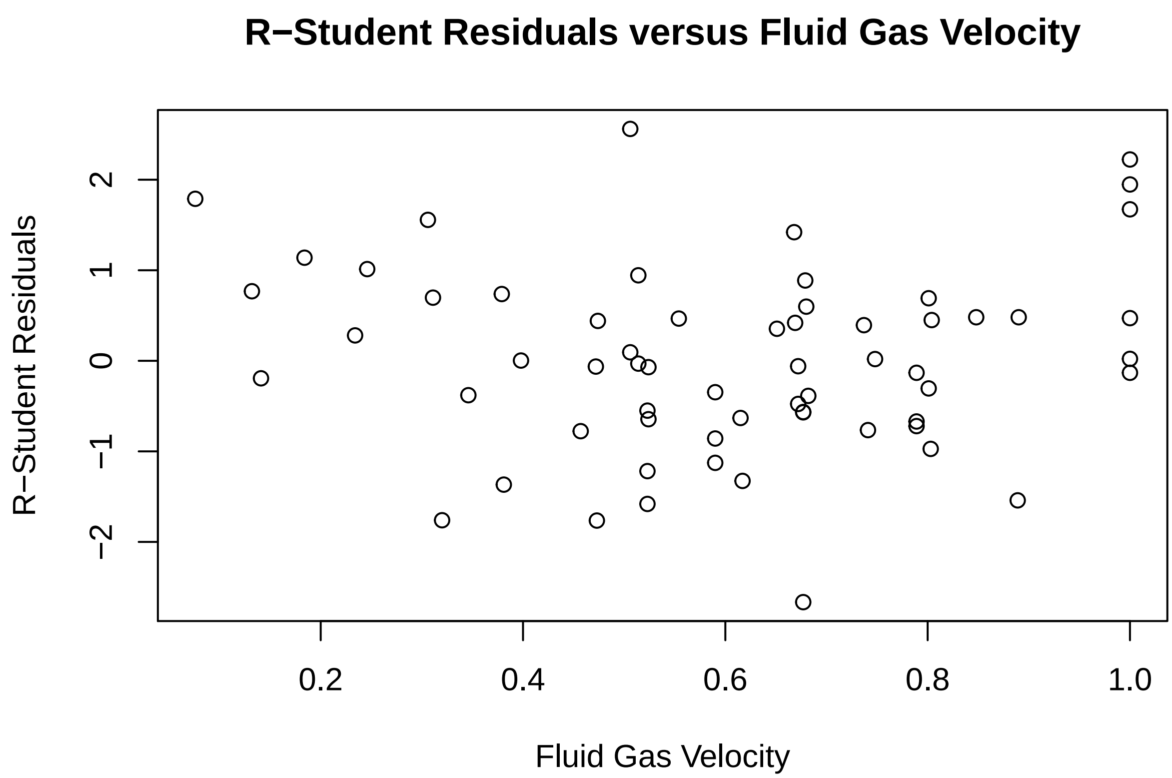

```
plot(log(viscosity), R_Student_Residuals, ylab = "R-Student Residuals", xlab = "Viscosity", main = "R-St
```




```
plot(mesh_open, R_Student_Residuals, ylab = "R-Student Residuals", xlab = "Mesh Opening", main = "R-Stu
```




```
plot(fluid_gas_vel, R_Student_Residuals, ylab = "R-Student Residuals", xlab = "Fluid Gas Velocity", mai
```



Lesson 9 additions

```
exL2 <- read.xlsx("data-table-B13.xlsx",
                   sheetIndex = 1,
                   colIndex = c(2,3,4,5,6,7,8),
                   as.data.frame = TRUE,
                   header = TRUE)

names(exL2) <- c("thrust",
                 "primary",
                 "secondary",
                 "fuel",
                 "press",
                 "exhaust",
                 "ambient")
attach(exL2)

# Output data structure and dimensions
str(exL2)
```

```
## 'data.frame':    40 obs. of  7 variables:
##  $ thrust   : num  4540 4315 4095 3650 3200 ...
##  $ primary  : num  2140 2016 1905 1675 1474 ...
```

```
##  $ secondary: num  20640 20280 19860 18980 18100 ...
##  $ fuel     : num  30250 30010 29780 29330 28960 ...
##  $ press    : num  205 195 184 164 144 216 206 196 171 149 ...
##  $ exhaust  : num  1732 1697 1662 1598 1541 ...
##  $ ambient  : num  99 100 97 97 97 87 87 87 85 85 ...
```

```
dim(exL2)
```

```
## [1] 40  7
```

```
out <- as.data.frame(c(exL2))
colnames(out) <- c("thrust", "primary","secondary","fuel","press","exhaust","ambient")
tab <- xtable(out, digits=c(0,0,0,0,0,0,0,0))
print(tab, type="html")
```

thrust

primary

secondary

fuel

press

exhaust

ambient

1

4540

2140

20640

30250

205

1732

99

2

4315

2016

20280

30010

195

1697

100

3

4095

1905

19860

29780

184

1662

97

4

3650

1675

18980

29330

164

1598

97

5

3200

1474

18100

28960

144

1541

97

6

4833

2239

20740

30083

216

1709

87

7

4617

2120

20305

29831

206

1669

87

8

4340

1990

19961

29604

196

1640

87

9

3820

1702

18916

29088

171

1572

85

10

3368

1487

18012

28675

149

1522

85

11

4445

2107

20520

30120

195

1740

101

12

4188

1973

20130

29920

190

1711

100

13

3981

1864

19780

29720

180

1682

100

14

3622

1674

19020

29370

161

1630

100

15

3125

1440

18030

28940

139

1572

101

16

4560

2165

20680

30160

208

1704

98

17

4340

2048

20340

29960

199

1679

96

18

4115

1916

19860

29710

187

1642

94

19

3630

1658

18950

29250

164

1576

94

20

3210

1489

18700

28890

145

1528

94

21

4330

2062

20500

30190

193

1748

101

22

4119

1929

20050

29960

183

1713

100

23

3891

1815

19680

29770

173

1684

100

24

3467

1595

18890

29360

153

1624

99

25

3045

1400

17870

28960

134

1569

100

26

4411

2047

20540

30160

193

1746

99

27

4203

1935

20160

29940

184

1714

99

28

3968

1807

19750

29760

173

1679

99

29

3531

1591

18890

29350

153

1621

99

30

3074

1388

17870

28910

133

1561

99

31

4350

2071

20460

30180

198

1729

102

32

4128

1944

20010

29940

186

1692

101

33

3940

1831

19640

29750

178

1667

101

34

3480

1612

18710

29360

156

1609

101

35

3064

1410

17780

28900

136

1552

101

36

4402

2066

20520

30170

197

1758

100

37

4180

1954

20150

29950

188

1729

99

38

3973

1835

19750

29740

178

1690

99

39

3530

1616

18850

29320

156

1616

99

40

3080

1407

17910

28910

137

1569

100

```r
# Perform preliminary calculations that we might use at some point
N <- length(thrust)

X <- matrix(c(rep(1, N),
              exL2[, "primary"],
              exL2[, "secondary"],
              exL2[, "fuel"],
              exL2[, "press"],
              exL2[, "exhaust"],
              exL2[, "ambient"]), ncol = 7, byrow = FALSE) # Define X matrix of regressor observations
xTx <- t(X) %*% X # Calulate the matrix product of X_Transpose and X
C <- ginv(xTx, tol=.Machine$double.eps) # Define C matrix
```

Jet Turbine Ex – lm() fit

```r
model.jet <- lm(thrust ~ primary + secondary + fuel + press + exhaust + ambient)

model.jet
```

```
##
## Call:
## lm(formula = thrust ~ primary + secondary + fuel + press + exhaust +
##     ambient)
##
## Coefficients:
## (Intercept)      primary    secondary         fuel        press      exhaust
##  -4.738e+03    1.119e+00   -3.018e-02    2.306e-01    3.850e+00    8.219e-01
##     ambient
##  -1.695e+01
```

```r
summary(model.jet)
```

```
##
## Call:
## lm(formula = thrust ~ primary + secondary + fuel + press + exhaust +
##     ambient)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.949 -19.028  -1.572  17.139  49.606
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.738e+03  2.445e+03  -1.938 0.061213 .
## primary      1.119e+00  2.865e-01   3.904 0.000441 ***
## secondary   -3.018e-02  3.823e-02  -0.789 0.435478
## fuel         2.306e-01  1.180e-01   1.954 0.059231 .
## press        3.850e+00  2.686e+00   1.433 0.161246
## exhaust      8.219e-01  3.508e-01   2.343 0.025298 *
## ambient     -1.695e+01  2.620e+00  -6.468 2.45e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.51 on 33 degrees of freedom
## Multiple R-squared:  0.9977, Adjusted R-squared:  0.9972
## F-statistic:  2350 on 6 and 33 DF,  p-value: < 2.2e-16
```

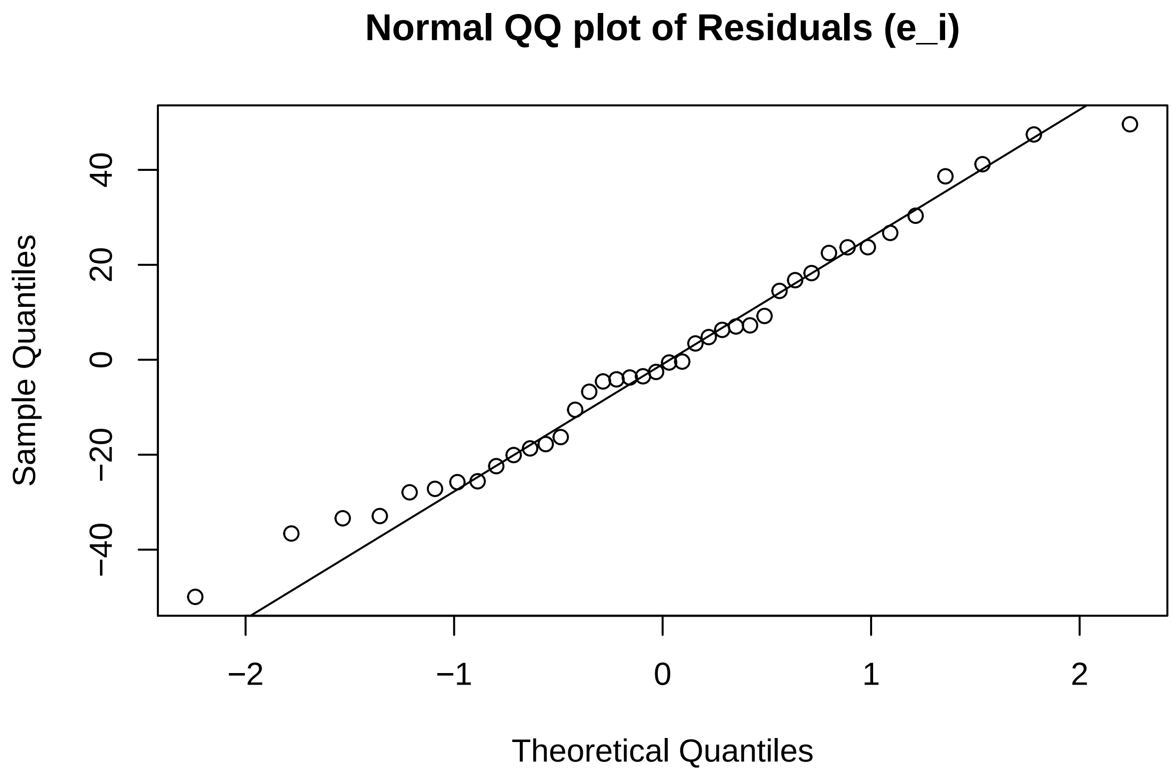
```r
anova(model.jet)
```

```
## Analysis of Variance Table
##
## Response: thrust
##           Df  Sum Sq Mean Sq    F value    Pr(>F)
## primary    1 9833160 9833160 13993.0720 < 2.2e-16 ***
## secondary  1    5992    5992     8.5274  0.006263 **
## fuel       1   25819   25819    36.7413 8.052e-07 ***
## press      1    4707    4707     6.6982  0.014240 *
## exhaust    1    9772    9772    13.9056  0.000721 ***
## ambient    1   29397   29397    41.8337 2.450e-07 ***
## Residuals 33   23190     703
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

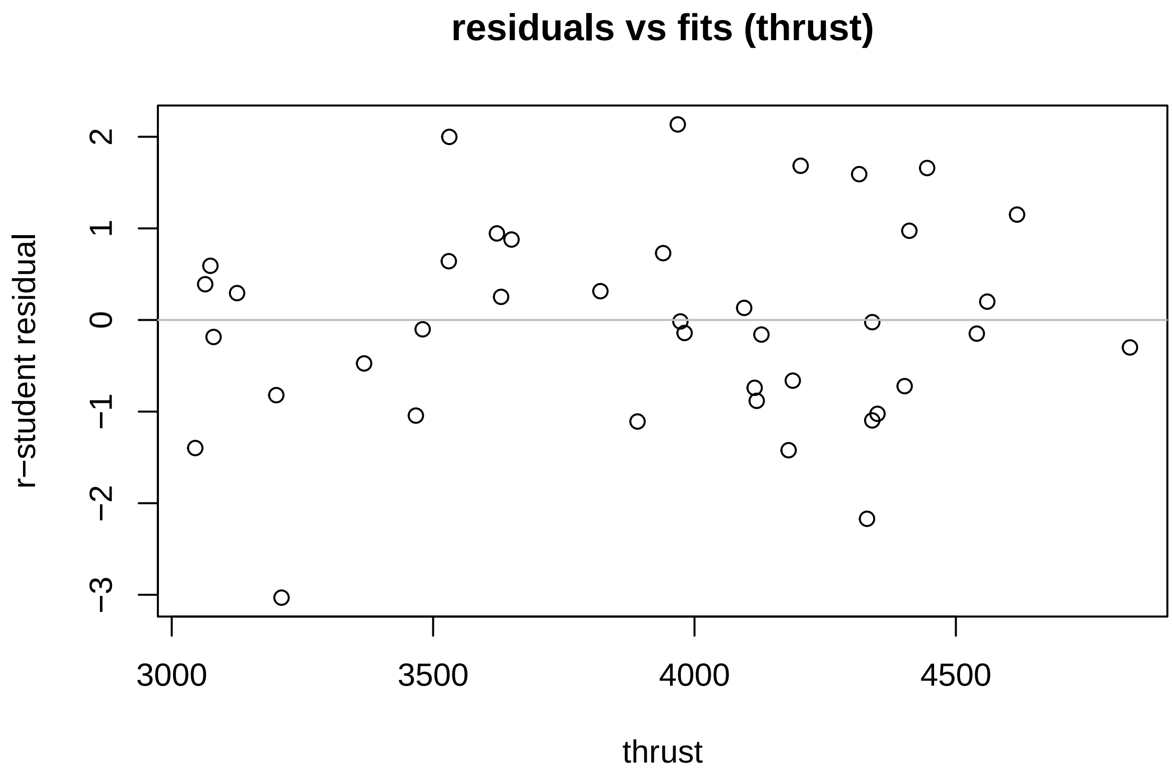
```r
df_T <- N - 1
df_Regression <- length(model.jet$coefficients) - 1 # Number of predictor variables, two in this case
k <- df_Regression # Sometimes helpful to use the variable
p <- k + 1 # Same reasoning
df_Residual <- df_T - df_Regression

y_observed <- exL2[, "thrust"] # Create vector of dependent variable observations
y_predicted <- (X %*% ginv(t(X)%*%X,tol=.Machine$double.eps) %*% t(X)) %*% y_observed # Hat matrix appr
y_bar <- mean(y_observed)
```

```
xTy <- t(X) %*% y_observed # Define rhs of least squares normal equations in matrix form
BetaCoefficients <- C %*% xTy # Calculate using ginv()
ResidualVector <- y_observed - X%*%BetaCoefficients # Define vector of residuals for each observation
SS_Residuals <- t(ResidualVector) %*% ResidualVector # Dot product of residual vector and its transpose

SST_dt <- sum((y_observed - y_bar)^2)

SS_Regression_dt <- sum((y_predicted - y_bar)^2)
SS_Residual_dt <- SST_dt - SS_Regression_dt

MS_Regression <- SS_Regression_dt / df_Regression
MS_Residual <- SS_Residual_dt / df_Residual

F_test <- MS_Regression / MS_Residual
t_test <- BetaCoefficients / sqrt(MS_Residual * diag(C)) # Formula on p.88

significanceLevel <- 0.05
F_critical <- qf(1-significanceLevel/2, df_Regression, df_Residual)
t_critical <- qt(1-significanceLevel/2, df_Residual)
#p_val <- 2*(1 - pt(t_test, df_Residual)) # Multiply by two since it is a two-tailed hypothesis test

# compare with my calculations
alpha = 0.05

tcritical_value = abs(qt(alpha/2, df = df_Residual))
p_values = 2*pt(-abs(t_test), df = df_Residual)
```

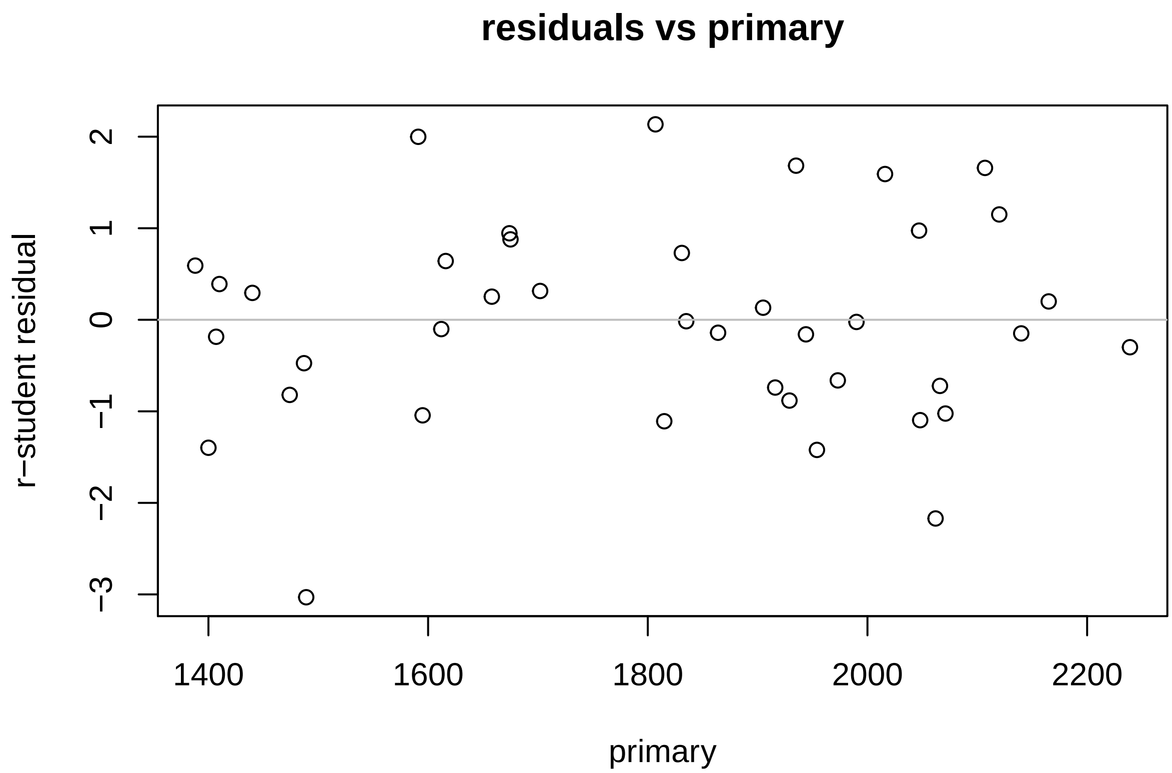
```
qqnorm(model.jet$residuals,main="Normal QQ plot of Residuals (e_i)")
qqline(model.jet$residuals)
```



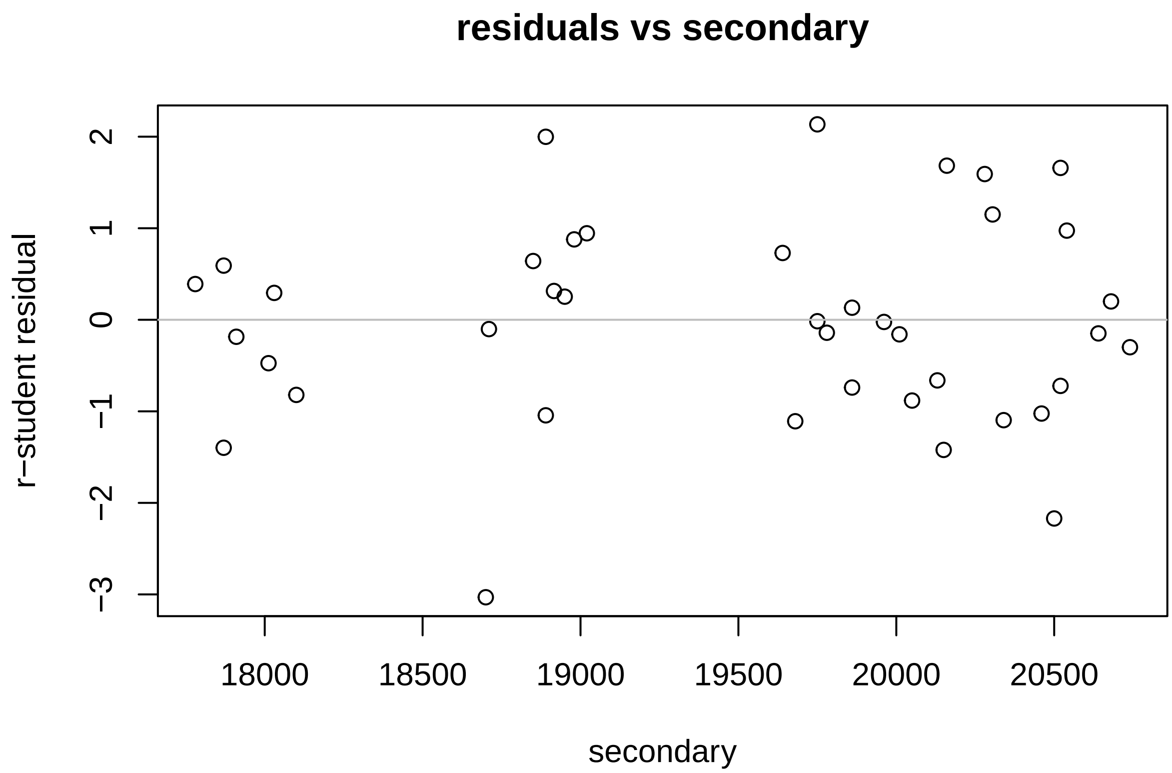

```
plot(thrust,rstudent(model.jet), main="residuals vs fits (thrust)",
     ylab="r-student residual")
abline(0, 0, col="gray")
```



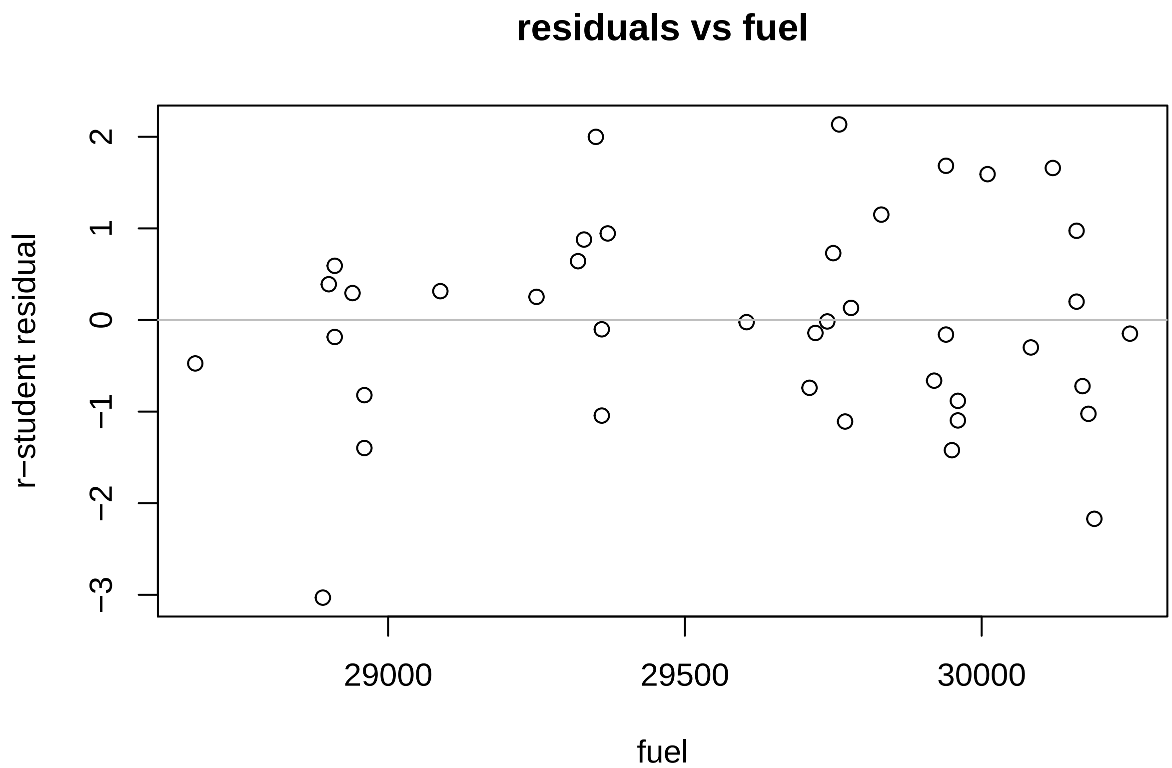

```
plot(primary,rstudent(model.jet), main="residuals vs primary",
     ylab="r-student residual")
abline(0, 0, col="gray")
```



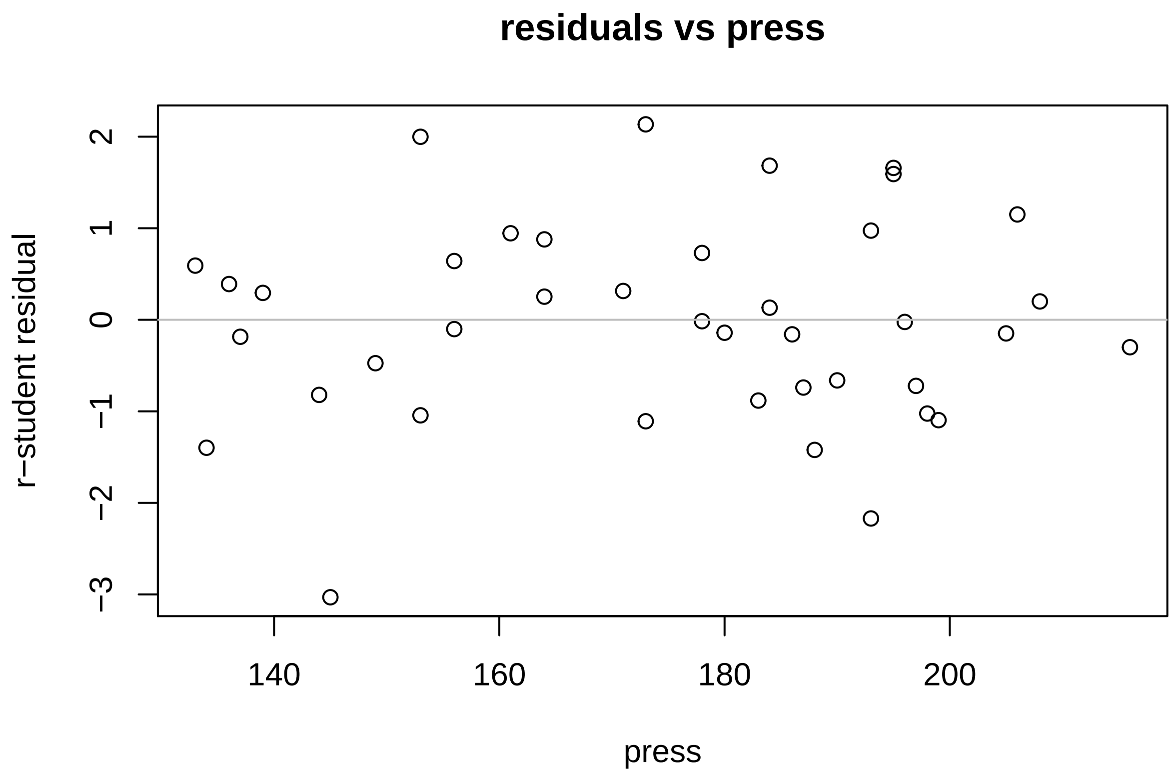

```
plot(secondary,rstudent(model.jet), main="residuals vs secondary",
     ylab="r-student residual")
abline(0, 0, col="gray")
```



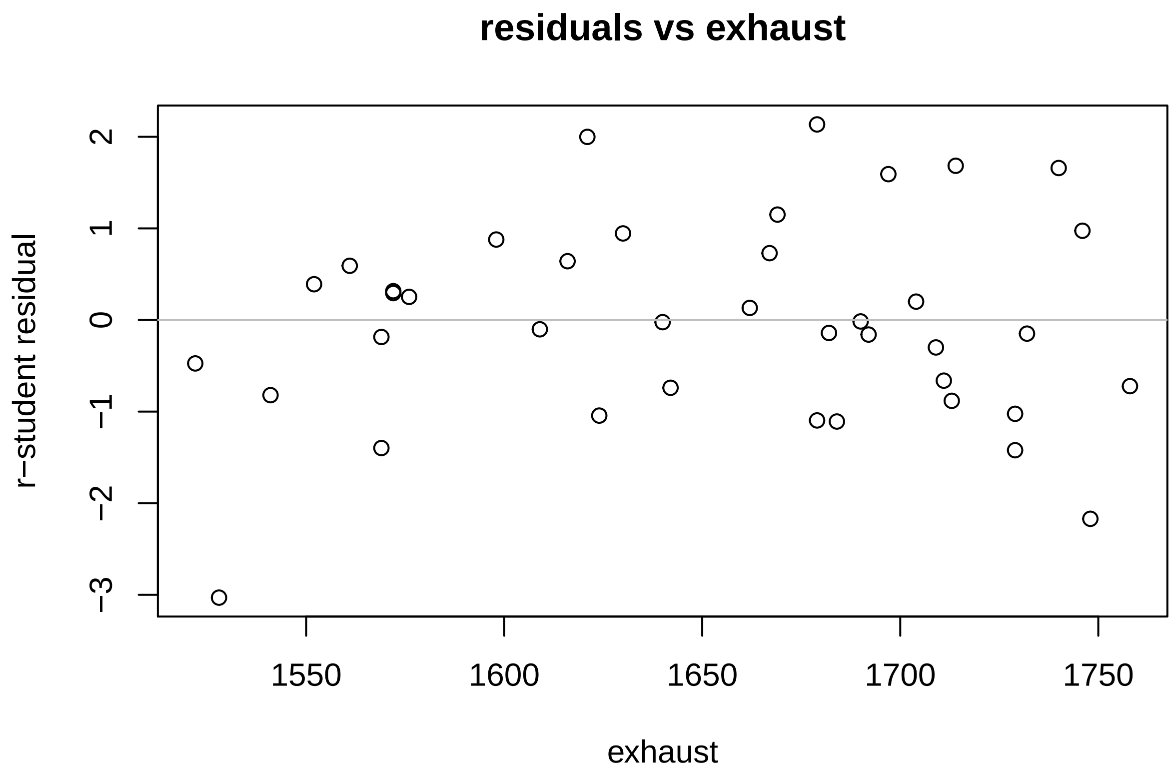

```
plot(fuel,rstudent(model.jet), main="residuals vs fuel",
     ylab="r-student residual")
abline(0, 0, col="gray")
```



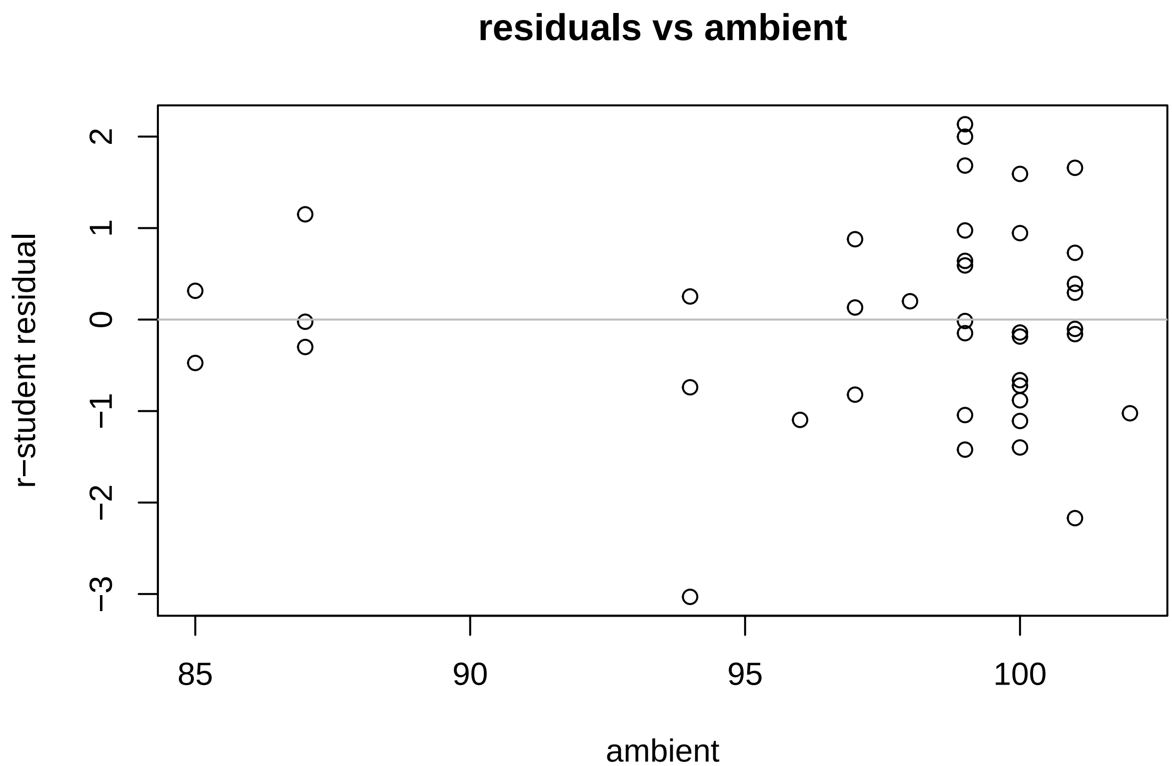

```
plot(press,rstudent(model.jet), main="residuals vs press",
     ylab="r-student residual")
abline(0, 0, col="gray")
```




```
plot(exhaust,rstudent(model.jet), main="residuals vs exhaust",
     ylab="r-student residual")
abline(0, 0, col="gray")
```




```
plot(ambient,rstudent(model.jet), main="residuals vs ambient",
     ylab="r-student residual")
abline(0, 0, col="gray")
```


residuals vs ambient
r−student residual
2
1
0
−1
−2
−3
85
90
95
100
ambient