L12Ex_Voltage_Drop_Rick_Davila

Rick Davila

5/24/2020

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

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

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

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

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

rm(list = ls())

# Load data
Ex72 <- read.xlsx(
  "data-ex-7-2.xlsx",
  sheetIndex = 1, sheetName=NULL, rowIndex=NULL,
  startRow=NULL, endRow=NULL, colIndex= c(1,2,3),
  as.data.frame=TRUE, header=TRUE, colClasses=NA,
  keepFormulas=FALSE, encoding="unknown")

# Give labels to data columns
names(Ex72) <- c("obs",
                 "time",
                 "voltage_drop")
attach(Ex72)

# Output data to make sure it reads properly
out <- as.data.frame(c(Ex72))
colnames(out) <- c("obs",
                 "time",
                 "voltage_drop")
tab <- (xtable(out, digits=c(0,0,1,2)))
print(tab, type="html")
```

	obs	time	voltage_drop
1	1	0.0	8.33
2	2	0.5	8.23
3	3	1.0	7.17
4	4	1.5	7.14
5	5	2.0	7.31
6	6	2.5	7.60

7	7	3.0	7.94
8	8	3.5	8.30
9	9	4.0	8.76
10	10	4.5	8.71
11	11	5.0	9.71
12	12	5.5	10.26
13	13	6.0	10.91
14	14	6.5	11.67
15	15	7.0	11.76
16	16	7.5	12.81
17	17	8.0	13.30
18	18	8.5	13.88
19	19	9.0	14.59
20	20	9.5	14.05
21	21	10.0	14.48
22	22	10.5	14.92
23	23	11.0	14.37
24	24	11.5	14.63
25	25	12.0	15.18
26	26	12.5	14.51
27	27	13.0	14.34
28	28	13.5	13.81
29	29	14.0	13.79
30	30	14.5	13.05
31	31	15.0	13.04
32	32	15.5	12.60
33	33	16.0	12.05
34	34	16.5	11.15
35	35	17.0	11.15
36	36	17.5	10.14
37	37	18.0	10.08
38	38	18.5	9.78
39	39	19.0	9.80
40	40	19.5	9.95
41	41	20.0	9.51

```
# Output data structure and dimensions
str(Ex72)
```

'data.frame': 41 obs. of 3 variables: $ obs : num 1 2 3 4 5 6 7 8 9 10 … $ time : num 0 0.5 1 1.5 2 2.5 3 3.5 4 4.5 … $ voltage_drop: num 8.33 8.23 7.17 7.14 7.31 7.6 7.94 8.3 8.76 8.71 …

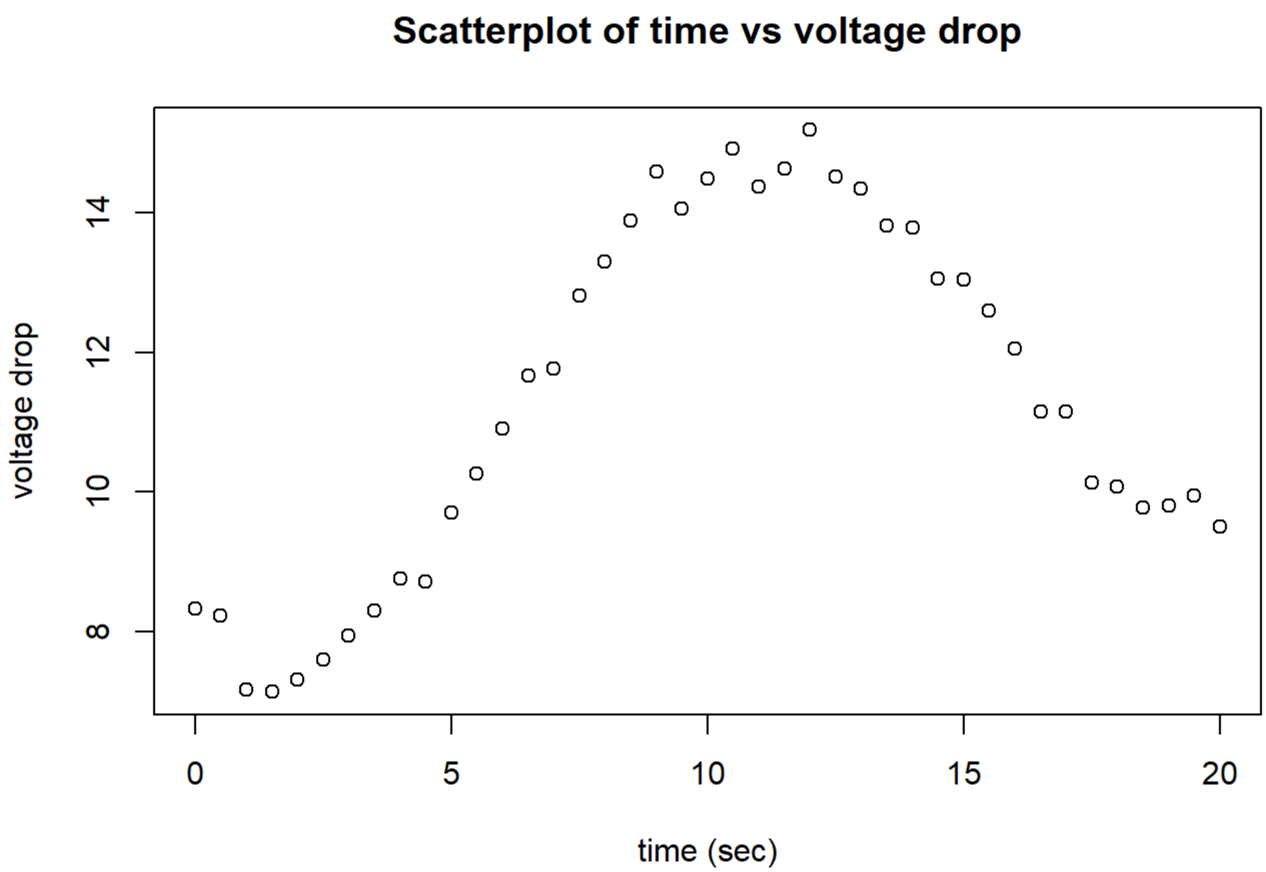
```
dim(Ex72)
```

[1] 41 3

Example 7.2 (p.231-234)

Create scatterplot

```
plot(time, voltage_drop, main = "Scatterplot of time vs voltage drop",
     xlab = "time (sec)",
     ylab = "voltage drop")
```



Fit a cubic spline using two knots, one at t1=6.5 and one at t2=13

Create indicator functions for the two knots

```
t1 <- 6.5
t2 <- 13

x_t1 <- ifelse((time-t1)>0,(time-t1),0)
x_t2 <- ifelse((time-t2)>0,(time-t2),0)
```

Add columns to dataframe for the higher-order terms and spline terms

```
Ex72$x_b01 <- time
Ex72$x_b02 <- time^2
Ex72$x_b03 <- time^3
Ex72$x_b1 <- x_t1^3
Ex72$x_b2 <- x_t2^3

# Output data to make sure it reads properly
out <- as.data.frame(c(Ex72))
#colnames(out) <- c("obs",
#                 "time",
#                 "voltage_drop")
tab <- (xtable(out, digits=c(0,0,1,2,3,3,3,3,3)))
print(tab, type="html")
```

	obs	time	voltage_drop	x_b01	x_b02	x_b03	x_b1	x_b2
1	1	0.0	8.33	0.000	0.000	0.000	0.000	0.000
2	2	0.5	8.23	0.500	0.250	0.125	0.000	0.000
3	3	1.0	7.17	1.000	1.000	1.000	0.000	0.000
4	4	1.5	7.14	1.500	2.250	3.375	0.000	0.000
5	5	2.0	7.31	2.000	4.000	8.000	0.000	0.000
6	6	2.5	7.60	2.500	6.250	15.625	0.000	0.000
7	7	3.0	7.94	3.000	9.000	27.000	0.000	0.000
8	8	3.5	8.30	3.500	12.250	42.875	0.000	0.000
9	9	4.0	8.76	4.000	16.000	64.000	0.000	0.000
10	10	4.5	8.71	4.500	20.250	91.125	0.000	0.000
11	11	5.0	9.71	5.000	25.000	125.000	0.000	0.000
12	12	5.5	10.26	5.500	30.250	166.375	0.000	0.000
13	13	6.0	10.91	6.000	36.000	216.000	0.000	0.000
14	14	6.5	11.67	6.500	42.250	274.625	0.000	0.000
15	15	7.0	11.76	7.000	49.000	343.000	0.125	0.000
16	16	7.5	12.81	7.500	56.250	421.875	1.000	0.000
17	17	8.0	13.30	8.000	64.000	512.000	3.375	0.000
18	18	8.5	13.88	8.500	72.250	614.125	8.000	0.000
19	19	9.0	14.59	9.000	81.000	729.000	15.625	0.000
20	20	9.5	14.05	9.500	90.250	857.375	27.000	0.000
21	21	10.0	14.48	10.000	100.000	1000.000	42.875	0.000
22	22	10.5	14.92	10.500	110.250	1157.625	64.000	0.000
23	23	11.0	14.37	11.000	121.000	1331.000	91.125	0.000
24	24	11.5	14.63	11.500	132.250	1520.875	125.000	0.000
25	25	12.0	15.18	12.000	144.000	1728.000	166.375	0.000
26	26	12.5	14.51	12.500	156.250	1953.125	216.000	0.000
27	27	13.0	14.34	13.000	169.000	2197.000	274.625	0.000
28	28	13.5	13.81	13.500	182.250	2460.375	343.000	0.125
29	29	14.0	13.79	14.000	196.000	2744.000	421.875	1.000
30	30	14.5	13.05	14.500	210.250	3048.625	512.000	3.375
31	31	15.0	13.04	15.000	225.000	3375.000	614.125	8.000
32	32	15.5	12.60	15.500	240.250	3723.875	729.000	15.625
33	33	16.0	12.05	16.000	256.000	4096.000	857.375	27.000

34	34	16.5	11.15	16.500	272.250	4492.125	1000.000	42.875
35	35	17.0	11.15	17.000	289.000	4913.000	1157.625	64.000
36	36	17.5	10.14	17.500	306.250	5359.375	1331.000	91.125
37	37	18.0	10.08	18.000	324.000	5832.000	1520.875	125.000
38	38	18.5	9.78	18.500	342.250	6331.625	1728.000	166.375
39	39	19.0	9.80	19.000	361.000	6859.000	1953.125	216.000
40	40	19.5	9.95	19.500	380.250	7414.875	2197.000	274.625
41	41	20.0	9.51	20.000	400.000	8000.000	2460.375	343.000

Fit model; compare to values in Table 7.4, p. 233

```
model.72 <- lm(voltage_drop ~
                 Ex72$x_b01 +
                 Ex72$x_b02 +
                 Ex72$x_b03 +
                 Ex72$x_b1 +
                 Ex72$x_b2)

summary(model.72)
```

Call: lm(formula = voltage_drop ~ Ex72$x_b01 + Ex72$x_b02 + Ex72$x_b03 + Ex72$x_b1 + Ex72$x_b2)

Residuals: Min 1Q Median 3Q Max -0.45168 -0.18499 -0.03547 0.20577 0.61694

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 8.465678 0.200520 42.219 < 2e-16 ***Ex72***
$x_b01 - 1.453124 0.181586 - 8.002 2.04e - 09 *** Ex72$***x_b02 0.489889 0.043018 11.388 2.54e-13*** Ex72
$x_b03 - 0.029467 0.002848 - 10.347 3.44e - 12 *** Ex72$x_b1 0.024706 0.004039 6.116 5.43e-07
Ex72$x_b2 0.027112 0.003578 7.577 6.98e-09 — Signif. codes: 0 '***' 0.001 '**' *0.01* '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2678 on 35 degrees of freedom Multiple R-squared: 0.9904, Adjusted R-squared: 0.9891 F-statistic: 725.5 on 5 and 35 DF, p-value: < 2.2e-16

```
xtable(summary(model.72))
```

	Estimate <dbl>	Std. Error <dbl>	t value <dbl>	Pr(>|t|) <dbl>
(Intercept)	8.46567813	0.200519766	42.218672	1.292813e-31
Ex72$x_b01	-1.45312398	0.181586270	-8.002389	2.041364e-09
Ex72$x_b02	0.48988886	0.043017866	11.388033	2.542202e-13
Ex72$x_b03	-0.02946712	0.002847800	-10.347331	3.443568e-12
Ex72$x_b1	0.02470600	0.004039269	6.116454	5.425314e-07
Ex72$x_b2	0.02711180	0.003578004	7.577352	6.980038e-09

6 rows

Reproduce ANOVA table on p. 233

```
summary(model.72)
```

Call: lm(formula = voltage_drop ~ Ex72$x_b01 + Ex72$x_b02 + Ex72$x_b03 + Ex72$x_b1 + Ex72$x_b2)

Residuals: Min 1Q Median 3Q Max -0.45168 -0.18499 -0.03547 0.20577 0.61694

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 8.465678 0.200520 42.219 < 2e-16 ***Ex72*** $x_b01 - 1.453124 0.181586 - 8.002 2.04e - 09 *** Ex72$***x_b02 0.489889 0.043018 11.388 2.54e-13*** Ex72 $x_b03 - 0.029467 0.002848 - 10.347 3.44e - 12 *** Ex72$x_b1 0.024706 0.004039 6.116 5.43e-07 ***Ex72$x_b2 0.027112 0.003578 7.577 6.98e-09*** — Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2678 on 35 degrees of freedom Multiple R-squared: 0.9904, Adjusted R-squared: 0.9891 F-statistic: 725.5 on 5 and 35 DF, p-value: < 2.2e-16

```
xtable(anova(model.72))
```

	Df <int>	Sum Sq <dbl>	Mean Sq <dbl>	F value <dbl>	Pr(>F) <dbl>
Ex72$x_b01	1	48.162914	48.16291361	671.52626	1.989173e-24
Ex72$x_b02	1	170.493247	170.49324703	2377.15463	9.097170e-34
Ex72$x_b03	1	11.788218	11.78821797	164.36086	8.785188e-15
Ex72$x_b1	1	25.615992	25.61599208	357.15886	6.047372e-20
Ex72$x_b2	1	4.117984	4.11798435	57.41627	6.980038e-09
Residuals	35	2.510255	0.07172156	NA	NA

6 rows

Test significance of spline terms using Partial F-test. Use alpha = 0.01 as the significance level

```
# full model including spline terms
model.full <- lm(voltage_drop ~
                 Ex72$x_b01 +
                 Ex72$x_b02 +
                 Ex72$x_b03 +
                 Ex72$x_b1 +
                 Ex72$x_b2)

# reduced model excluding spline terms
model.reduced <- lm(voltage_drop ~
                 Ex72$x_b01 +
                 Ex72$x_b02 +
                 Ex72$x_b03)

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

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	37	32.244231	NA	NA	NA	NA
2	35	2.510255	2	29.73398	207.2876	3.955041e-20

2 rows

```
# F crit
alpha <- 0.01
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)
```

We're investigating the contribution of the spline terms to the model. That is, we wish to test

$$H_0 : \beta_1 = \beta_2 = 0$$

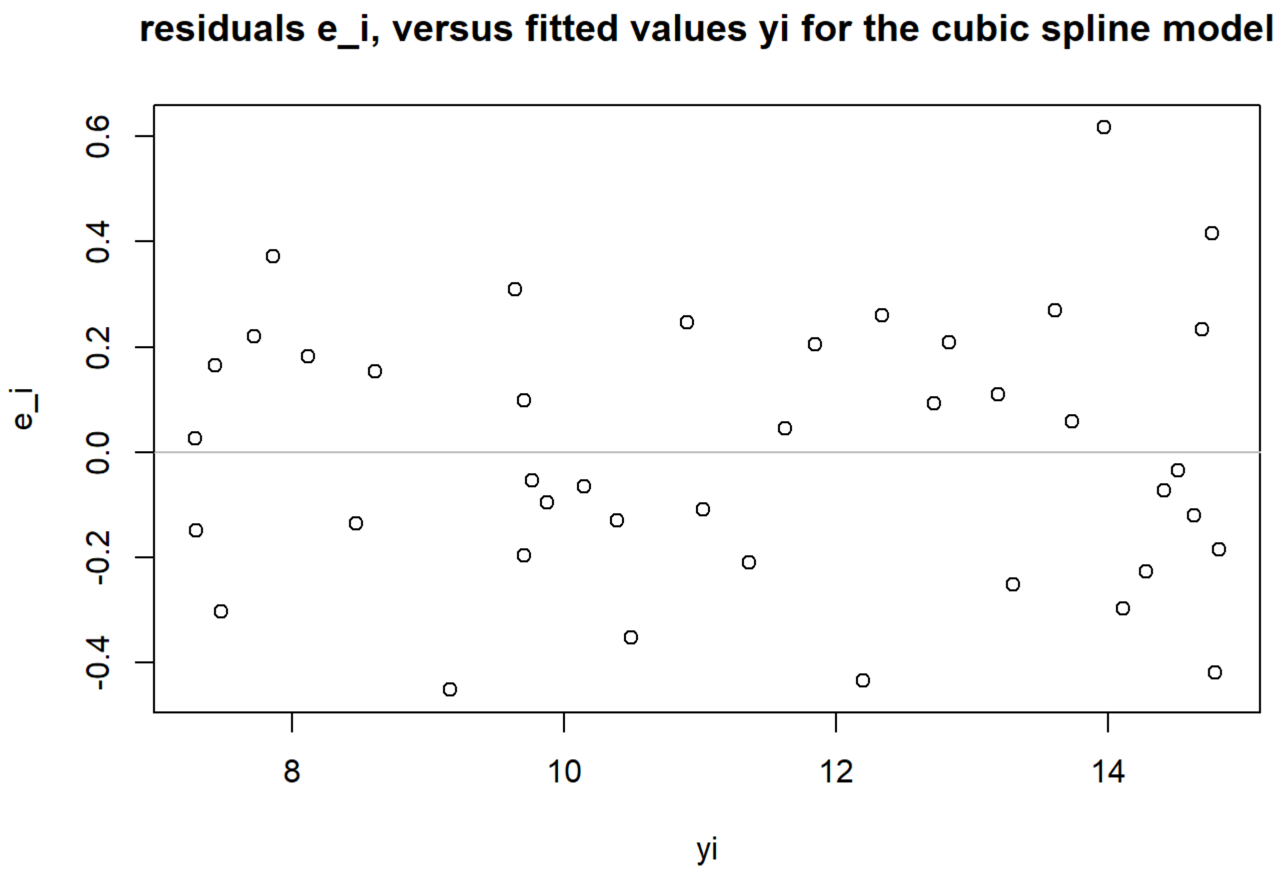
we have

$$F_0 = \frac{SS_R(\beta_1, \beta_2 | \beta_{00}, \beta_{01}, \beta_{02}, \beta_{03}/2}{MS_{Res}} = \frac{29.7339764/2}{0.0717216} = 207.2875639$$

and since $F_{0.01,2,35} = 5.2679413$, we have $F_0 > F_{0.01,2,35}$ and as a result, we reject the null hypothesis and conclude that the spline terms contributes significantly to the model.

Create residuals versus fits plot for each model

```
plot(model.72$fitted.values, model.72$residuals, main = "residuals e_i, versus fitted values ŷi
 for the cubic spline model", xlab = "ŷi", ylab = "e_i")
abline(0, 0, col = "gray")
```




```
# assumption, for the second model, like in the book, is to fit a cubic polynomial model and sho
w a residuals versus fits plot

x_b01b <- time
x_b02b <- time^2
x_b03b <- time^3

model.72b <- lm(voltage_drop ~
                 x_b01b +
                 x_b02b +
                 x_b03b)

summary(model.72b)
```

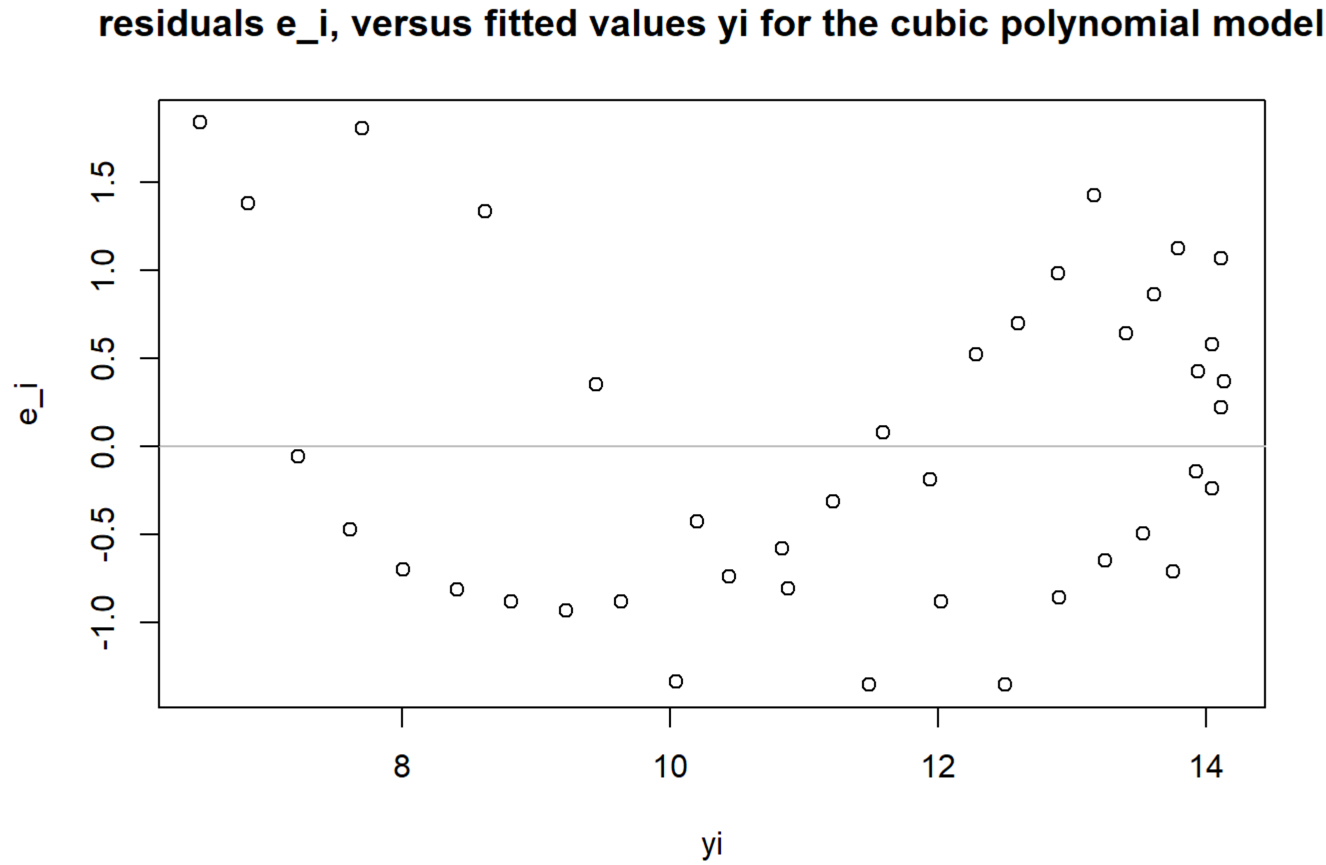
```
##
## Call:
## lm(formula = voltage_drop ~ x_b01b + x_b02b + x_b03b)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3503 -0.7340 -0.1859  0.6440  1.8390
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.4910163  0.5336473  12.163 1.71e-14 ***
## x_b01b       0.7031952  0.2339552   3.006 0.004738 **
## x_b02b       0.0340179  0.0273762   1.243 0.221829
## x_b03b      -0.0033072  0.0008992  -3.678 0.000743 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9335 on 37 degrees of freedom
## Multiple R-squared:  0.8773, Adjusted R-squared:  0.8673
## F-statistic: 88.14 on 3 and 37 DF,  p-value: < 2.2e-16
```

```
xtable(summary(model.72b))
```

	Estimate <dbl>	Std. Error <dbl>	t value <dbl>	Pr(>|t|) <dbl>
(Intercept)	6.491016346	0.5336472749	12.163496	1.705415e-14
x_b01b	0.703195219	0.2339551713	3.005684	4.738282e-03
x_b02b	0.034017947	0.0273761684	1.242612	2.218288e-01
x_b03b	-0.003307211	0.0008992137	-3.677892	7.431685e-04

4 rows

```
plot(model.72b$fitted.values, model.72b$residuals, main = "residuals e_i, versus fitted values ŷ
i for the cubic polynomial model", xlab = "ŷi", ylab = "e_i")
abline(0, 0, col = "gray")
```


residuals e_i, versus fitted values yi for the cubic polynomial model
e_i
-1.0
-0.5
0.0
0.5
1.0
1.5
8
10
12
14
yi