R code for polynomial regression Ex 8.6 p 336

```
# R code for steroid levels, ex8.6, p336
steroid=read.table("C:/Users/Mihinda/Desktop/steroid.txt", header=1)
#the data file
steroid
plot(steroid$x, steroid$y)
# x= ahe of the subject
# y = steroid level
sorted <- steroid[order(steroid$x),]
sorted
# The plots require the data to be sorted in this case.
fit1 <- lm(y ~ poly(x, 1, raw=TRUE), data=sorted)
summary(fit1)
anova(fit1)
fit2 <- lm(y ~ poly(x, 2, raw=TRUE), data=sorted)
# if we do not specify raw=TRUE , the function poly give us the values
#of the beta parameters of orthogonal polynomials, which is different
from
#what we want here.
summary(fit2)
anova(fit2)
par(mfrow=c(2,2))
plot(sorted$x, sorted$y)
points(sorted$x, predict(fit2), type="l", col="red")

plot(fitted(fit1),residuals(fit1))

plot(fitted(fit2),residuals(fit2))


fit3 <- lm(y ~ poly(x, 3, raw=TRUE), data=sorted)
summary(fit3)
anova(fit3)
x0 <- data.frame(x=15)
predict(fit2, x0)
#CI for the mean of Y at x=x0
predict(fit2, x0, interval="confidence", level=0.95)
```

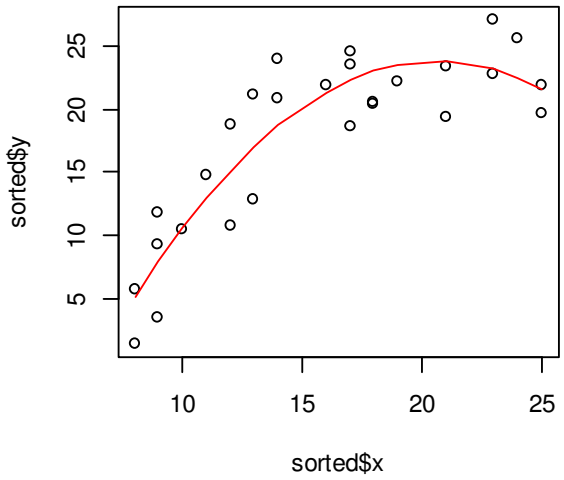
R output

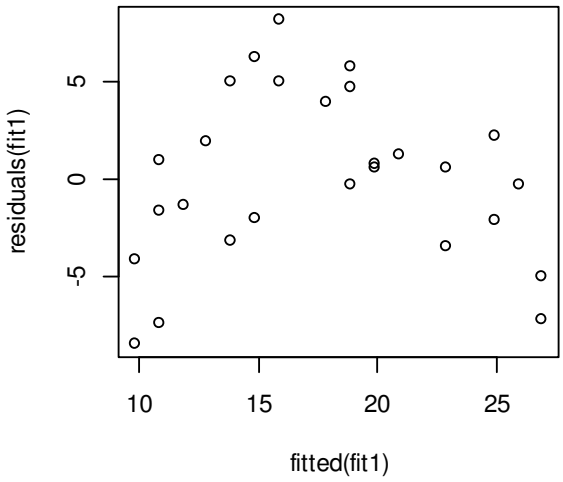
```
> # R code for steroid levels, ex8.6, p336
> steroid=read.table("C:/Users/Mihinda/Desktop/steroid.txt", header=1)
#the data file
> steroid
      y  x
1  27.1 23
2  22.1 19
3  21.9 25
4  10.7 12
5   1.4  8
6  18.8 12
7  14.7 11
8   5.7  8
9  18.6 17
10 20.4 18
11  9.2  9
12 23.4 21
13 10.5 10
14 19.7 25
15 11.8  9
16 24.6 17
17  3.4  9
18 22.8 23
19 21.1 13
20 24.0 14
21 21.8 16
22 23.5 17
23 19.4 21
24 25.6 24
25 12.8 13
26 20.8 14
27 20.6 18
```

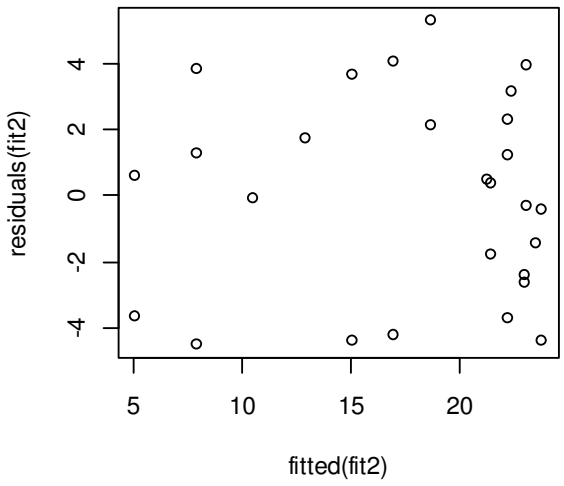
```
> # x= ahe of the subject
> # y = steroid level
> sorted <- steroid[order(steroid$x),]
> sorted
      y  x
5   1.4  8
8   5.7  8
11  9.2  9
15 11.8  9
17  3.4  9
13 10.5 10
7  14.7 11
4  10.7 12
6  18.8 12
19 21.1 13
25 12.8 13
20 24.0 14
26 20.8 14
21 21.8 16
9  18.6 17
16 24.6 17
22 23.5 17
10 20.4 18
27 20.6 18
2  22.1 19
12 23.4 21
23 19.4 21
1  27.1 23
18 22.8 23
24 25.6 24
3  21.9 25
14 19.7 25
> # The plots require the data to be sorted in this case.

# First let's look at the linear model
> fit1 <- lm(y ~ poly(x, 1, raw=TRUE), data=sorted)
> summary(fit1)

Call:
lm(formula = y ~ poly(x, 1, raw = TRUE), data = sorted)

Residuals:
    Min      1Q  Median      3Q     Max
-8.4340 -2.6239  0.5114  3.0677  8.1408

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)            1.8005     2.6363   0.683    0.501
poly(x, 1, raw = TRUE)   1.0042     0.1581   6.352  1.2e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 4.434 on 25 degrees of freedom
Multiple R-squared: 0.6174,     Adjusted R-squared: 0.6021
F-statistic: 40.35 on 1 and 25 DF,  p-value: 1.196e-06

> anova(fit1)
Analysis of Variance Table

Response: y
                        Df Sum Sq Mean Sq F value    Pr(>F)
poly(x, 1, raw = TRUE)   1 793.28  793.28  40.348 1.196e-06 ***
Residuals               25 491.53   19.66
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

# Quadratic model
> fit2 <- lm(y ~ poly(x, 2, raw=TRUE), data=sorted)
> # if we do not specify raw=TRUE , the function poly give us the
values
> #of the beta parameters of orthogonal polynomials, which is
different from
> #what we want here.
> summary(fit2)

Call:
lm(formula = y ~ poly(x, 2, raw = TRUE), data = sorted)

Residuals:
    Min      1Q  Median      3Q     Max
-4.5463 -2.5369  0.3868  2.1973  5.3020

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             -26.32541    5.88154  -4.476 0.000157 ***
poly(x, 2, raw = TRUE)1   4.87357    0.77515   6.287 1.69e-06 ***
poly(x, 2, raw = TRUE)2  -0.11840    0.02347  -5.045 3.71e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.153 on 24 degrees of freedom
Multiple R-squared: 0.8143,     Adjusted R-squared: 0.7989
F-statistic: 52.63 on 2 and 24 DF,  p-value: 1.678e-09

> anova(fit2)
Analysis of Variance Table

Response: y
                        Df  Sum Sq Mean Sq F value    Pr(>F)
poly(x, 2, raw = TRUE)   2 1046.27  523.13  52.633 1.678e-09 ***
Residuals               24  238.54    9.94
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> par(mfrow=c(2,2))
> plot(sorted$x, sorted$y)
> points(sorted$x, predict(fit2), type="l", col="red")
>
> plot(fitted(fit1),residuals(fit1))
>
> plot(fitted(fit2),residuals(fit2))
>
```







```
>
# cubic model
> fit3 <- lm(y ~ poly(x, 3, raw=TRUE), data=sorted)
> summary(fit3)

Call:
lm(formula = y ~ poly(x, 3, raw = TRUE), data = sorted)

Residuals:
    Min      1Q  Median      3Q     Max
-5.3854 -2.2414  0.3715  2.1975  4.5595

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             -52.173251  20.482576  -2.547   0.0180 *
poly(x, 3, raw = TRUE)1  10.339563   4.224095   2.448   0.0224 *
poly(x, 3, raw = TRUE)2  -0.474299   0.271491  -1.747   0.0940 .
poly(x, 3, raw = TRUE)3   0.007224   0.005491   1.316   0.2012
---
Residual standard error: 3.106 on 23 degrees of freedom
Multiple R-squared: 0.8273,     Adjusted R-squared: 0.8048
F-statistic: 36.73 on 3 and 23 DF,  p-value: 6.122e-09

> anova(fit3)
Analysis of Variance Table

Response: y
                      Df  Sum Sq Mean Sq F value    Pr(>F)
poly(x, 3, raw = TRUE)  3 1062.96  354.32  36.735 6.122e-09 ***
Residuals              23  221.84    9.65
---

# Confidence interval etc
> x0 <- data.frame(x=15)

> predict(fit2, x0)
       1
20.13792
> #CI for the mean of Y at x=x0
> predict(fit2, x0, interval="confidence", level=0.95)
       fit      lwr      upr
1 20.13792 18.29535 21.98049
```