STAT2402: Analysis of Observations
Notes on Lecture 2

R.Nazim Khan*

2 August, 2022

Summary

This document contains some notes on the data analysis in Lecture 2. When reading the data into R, I assume that the relevant files are stored in a subdirectory, called `data`, of the working directory. You may have to modify the commands with which the data is read in, but all other commands should work as shown, and they should produce the output shown.

Exercise 1: Analysis of MultipleReg data

```
## Set directory to data source
setwd("C:/Users/rnazi/Documents/Teaching/STAT2402/STAT2402-2023/RMaterial")
```

Building a Linear model In this exercise we will build a linear model.

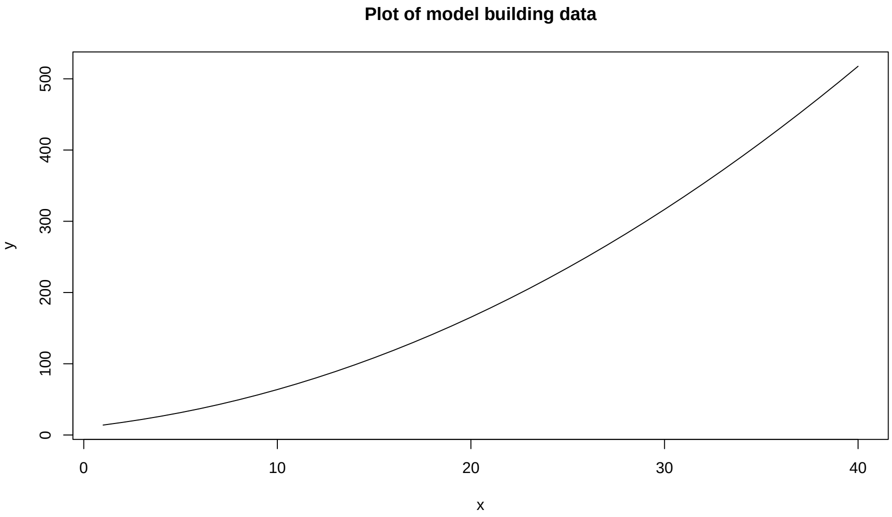
```
mult <- read.table("../Data/mult.txt", header = T, sep = "\t")
summary(mult)
```

```
##        x               y
##  Min.   : 1.00   Min.   : 13.95
##  1st Qu.:10.75   1st Qu.: 69.75
##  Median :20.50   Median :171.81
##  Mean   :20.50   Mean   :204.77
##  3rd Qu.:30.25   3rd Qu.:321.04
##  Max.   :40.00   Max.   :517.59
```

It is good to look at a summary of the data you read in, to make sure it has been read in correctly. Data contains two variables only. Let us look at a plot of the data.

```
with(mult, plot(y ~ x, type = "l", main = "Plot of model building data"))
```

*School of Mathematics and Statistics (M019), The University of Western Australia, 35 Stirling Highway, Crawley WA 6009, Australia. E-mail: `nazim.khan@uwa.edu.au`

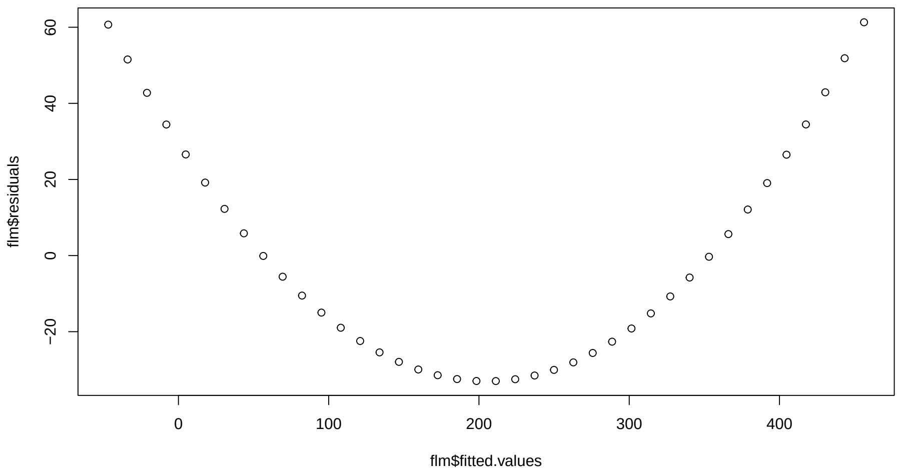


Plots indicates a slight curvature to the data. So how do we start the model building? Do we include an exponential term? Do we fit a model for $\log y$? Or shall we try a quadratic term? We start a simple linear model. Then by performing an analysis of residuals we select other terms and build up a model.

```
flm <- lm(y ~ x, data = mult)
summary(flm)

##
## Call:
## lm(formula = y ~ x, data = mult)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.973 -26.178  -8.142  21.016  61.307
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -59.6331     9.7454  -6.119 3.91e-07 ***
## x            12.8979     0.4142  31.137  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.24 on 38 degrees of freedom
## Multiple R-squared:  0.9623,Adjusted R-squared:  0.9613
## F-statistic: 969.5 on 1 and 38 DF,  p-value: < 2.2e-16

plot(flm$residuals ~ flm$fitted.values)
```

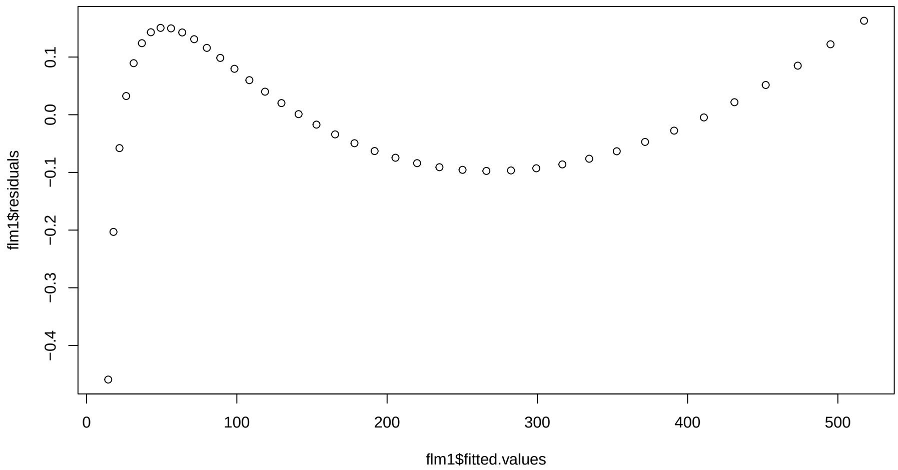



The residual plot indicates a quadratic term, so we include this.

```
flm1 <- update(flm, . ~ . + I(x^2))
summary(flm1)

##
## Call:
## lm(formula = y ~ x + I(x^2), data = mult)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45914 -0.07501 -0.00184  0.09164  0.16288
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.141e+01  6.114e-02   186.7   <2e-16 ***
## x           2.748e+00  6.878e-03   399.6   <2e-16 ***
## I(x^2)      2.475e-01  1.627e-04  1521.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1225 on 37 degrees of freedom
## Multiple R-squared:      1,Adjusted R-squared:      1
## F-statistic: 3.07e+07 on 2 and 37 DF,  p-value: < 2.2e-16

plot(flm1$residuals ~ flm1$fitted.values)
```

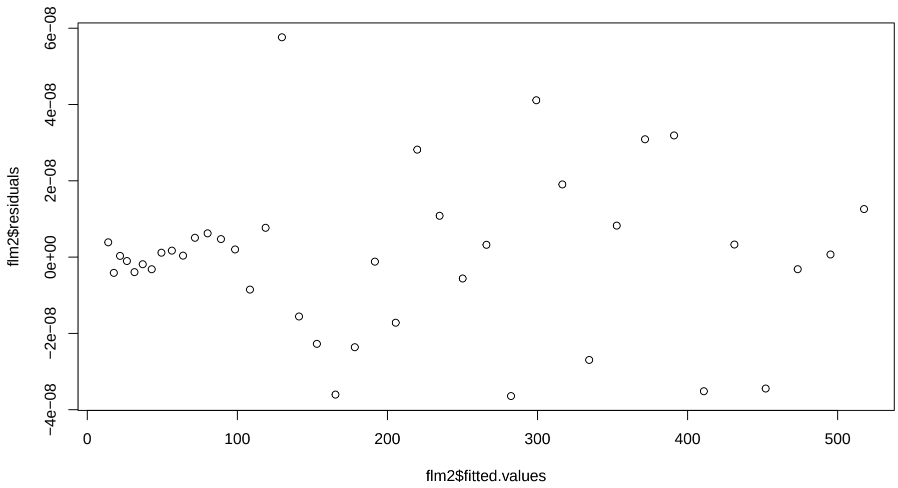



Already looking good, with very small residuals. There is still a pattern, which resembles a square root function, so we include this in the model.

```
flm2 <- update(flm1, . ~ . + I(sqrt(x)))
summary(flm2)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2) + I(sqrt(x)), data = mult)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.642e-08 -6.333e-09  5.320e-10  6.594e-09  5.762e-08
##
## Coefficients:
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept) 1.000e+01  4.163e-08 2.402e+08   <2e-16 ***
## x           2.500e+00  7.176e-09 3.484e+08   <2e-16 ***
## I(x^2)      2.500e-01  7.532e-11 3.319e+09   <2e-16 ***
## I(sqrt(x))  1.200e+00  3.419e-08 3.510e+07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.123e-08 on 36 degrees of freedom
## Multiple R-squared:      1,Adjusted R-squared:      1
## F-statistic: 6.815e+20 on 3 and 36 DF,  p-value: < 2.2e-16
```

```
plot(flm2$residuals ~ flm2$fitted.values)
```



Awesome! No patterns, *very* small residuals.

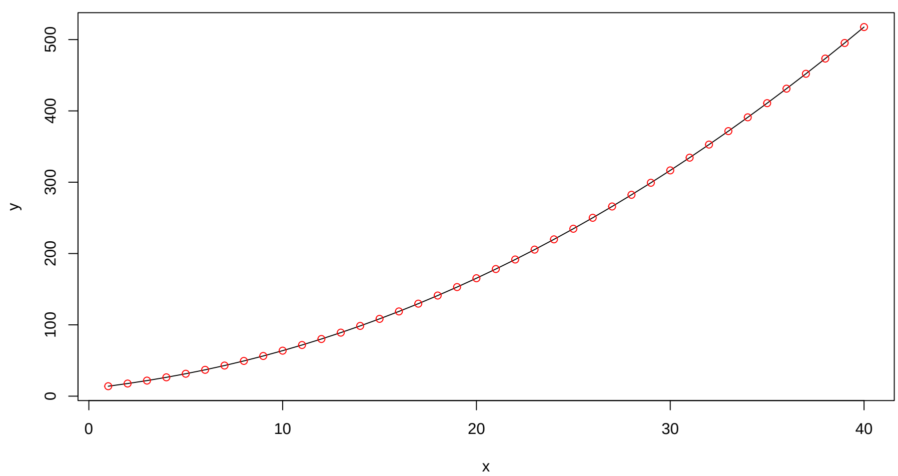
```
AIC(flm)
## [1] 390.201
AIC(flm1)
## [1] -49.56664
AIC(flm2)
## [1] -1294.127
```

The final model has lowest AIC and residual SE. Final model is

$$\hat{y} = 10 + 2.5x + 0.25x^{@} + 1.2\sqrt{x}.$$

This is the exact model used to generate the data. We plot the original data and the predicted values from the model.

```
with(mult, plot(y ~ x, type = "l"))
lines(predict.lm(flm2) ~ mult$x, type = "p", col = "red")
```



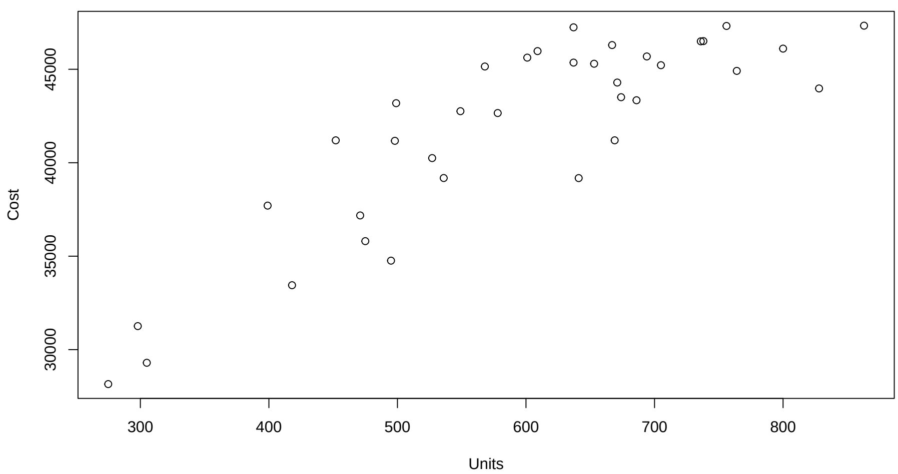
The fit is exact. So model building proceeds by fitting simple models first and then adding terms based on residual analysis.

Analysis of cost of power data We analyse the data `power.txt` to build a model for cost of power based on the usage. The data contains the cost of power and the usage in units. First read in the data and plot it.


```
power <- read.table("../Data/power.txt", header = T, sep = "\t")
summary(power)
```

```
##      Month            Cost           Units
##  Min.   : 1.00   Min.   :28157   Min.   :275.0
##  1st Qu.: 9.75   1st Qu.:39180   1st Qu.:497.2
##  Median :18.50   Median :43424   Median :623.0
##  Mean   :18.50   Mean   :41778   Mean   :593.7
##  3rd Qu.:27.25   3rd Qu.:45639   3rd Qu.:688.0
##  Max.   :36.00   Max.   :47332   Max.   :863.0
```

```
with(power, plot(Cost ~ Units))
```



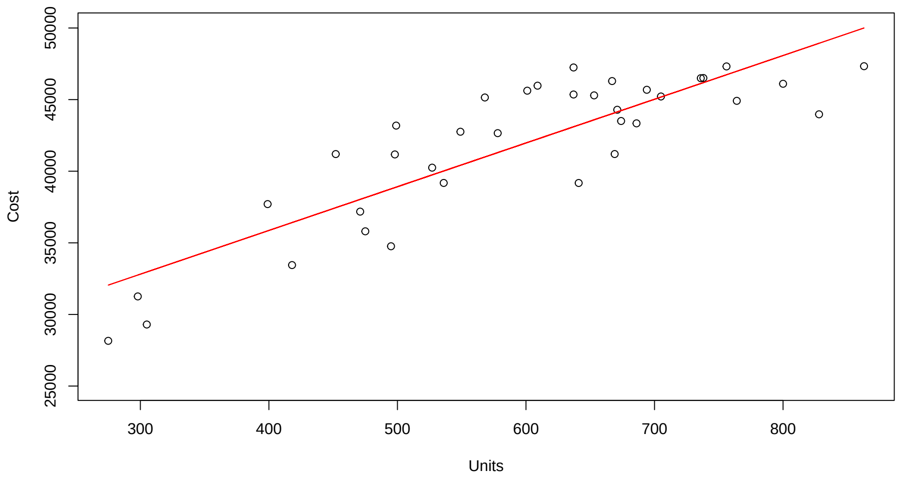
The cost appears to increase with usage, but seems to flatten off as cost increases. We start by fitting a simple linear model.

```
plm <- lm(Cost ~ Units, data = power)
summary(plm)
```

```
##
## Call:
## lm(formula = Cost ~ Units, data = power)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4958.9 -2136.0   236.4  2261.4  4297.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23651.489   1917.137  12.337 4.17e-14 ***
## Units          30.533      3.137   9.734 2.32e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2734 on 34 degrees of freedom
## Multiple R-squared:  0.7359,Adjusted R-squared:  0.7282
## F-statistic: 94.75 on 1 and 34 DF,  p-value: 2.317e-11
```

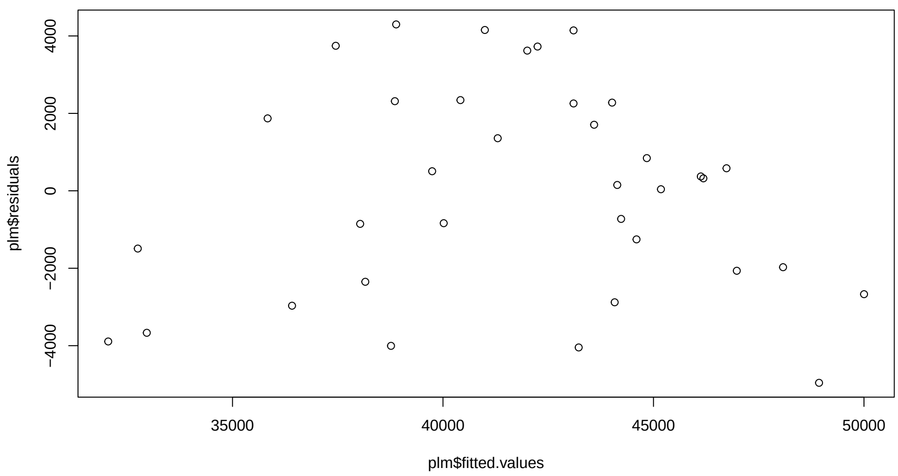
Some diagnostics now. First the plot of the fit.

```
with(power, plot(Cost ~ Units, ylim = c(25000, 50050)))
lines(predict.lm(plm) ~ power$Units, col = "red")
```

Not a very good fit, especially at the ends. Examine the plot of residuals to see how to improve this model.

```
plot(plm$residuals ~ plm$fitted.values)
```



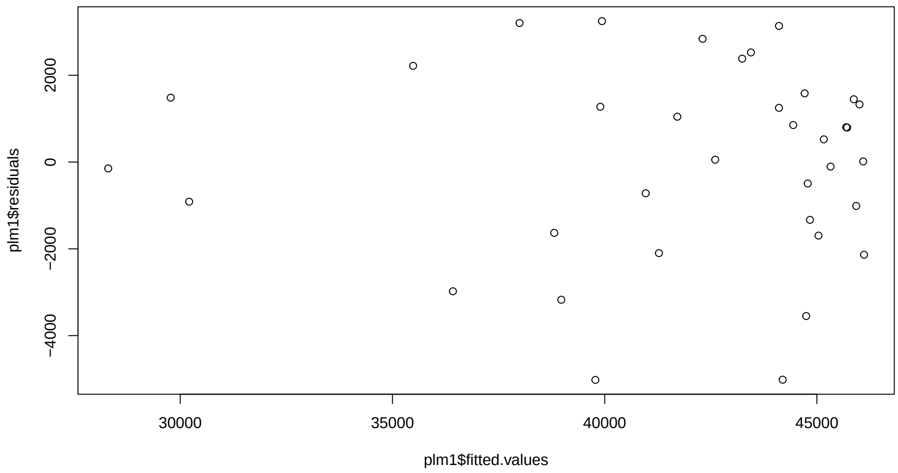
Nothing too clear here! But a closer look does indicate some curvature and a quadratic trend. Let us include a quadratic term in the model.

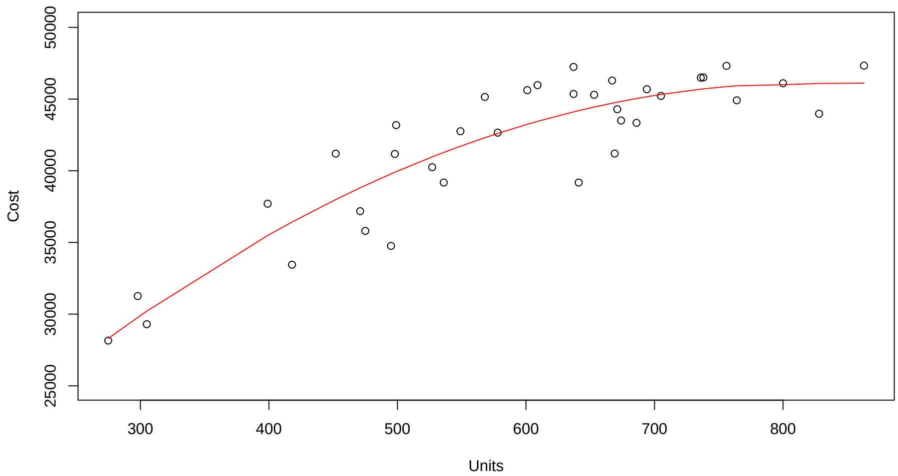
```
plm1 <- lm(Cost ~ Units + I(Units^2), data = power)
summary(plm1)
##
## Call:
## lm(formula = Cost ~ Units + I(Units^2), data = power)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5020.4 -1406.9   288.9  1456.9  3248.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5792.79829 4763.05850   1.216 0.232539
## Units         98.35039   17.23690   5.706  2.3e-06 ***
## I(Units^2)    -0.05997    0.01507  -3.981 0.000356 ***
## ---
```



```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2281 on 33 degrees of freedom
## Multiple R-squared:  0.8216,Adjusted R-squared:  0.8108
## F-statistic: 75.98 on 2 and 33 DF,  p-value: 4.453e-13
```

```
plot(plm1$residuals ~ plm1$fitted.values)
with(power, plot(Cost ~ Units, ylim = c(25000, 50050)))
lines(sort(predict.lm(plm1)) ~ sort(power$Units), col = "red")
```





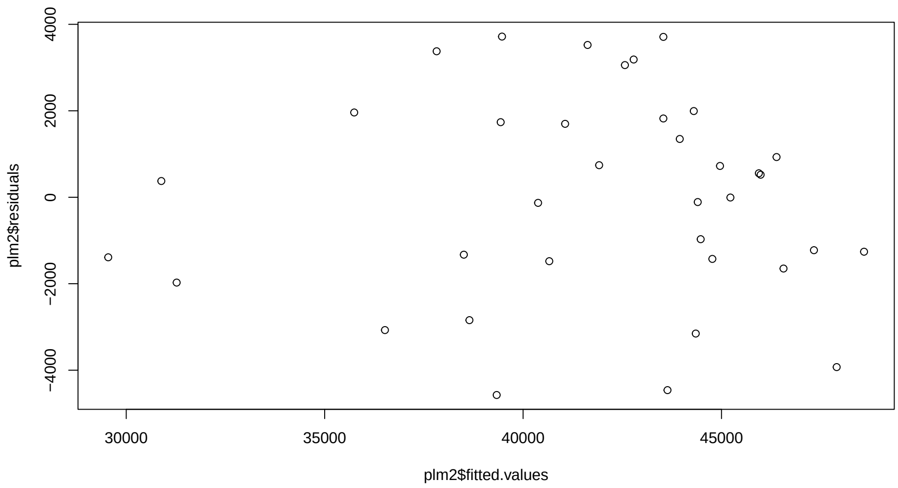
Fit looks much better. Problem with the model is that the coefficient of the square term is negative, so the Cost will decrease as Units increases. This is not reasonable. Let us try a log model.

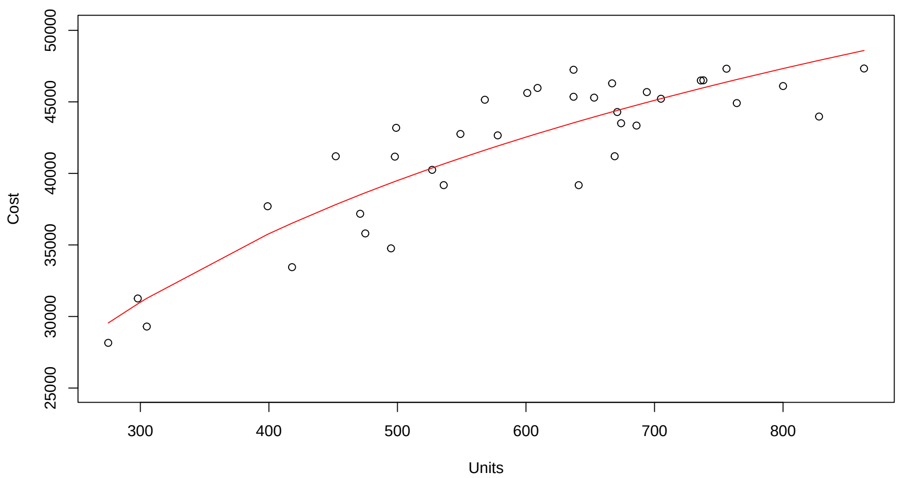
```
plm2 <- lm(Cost ~ I(log(Units)), data = power)
summary(plm2)
```

```
##
## Call:
## lm(formula = Cost ~ I(log(Units)), data = power)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4573.6 -1439.2   184.7  1758.1  3716.3
##
## Coefficients:
```



```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -63993       9144  -6.998 4.49e-08 ***
## I(log(Units))    16654       1438  11.578 2.41e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2393 on 34 degrees of freedom
## Multiple R-squared:  0.7977,Adjusted R-squared:  0.7917
## F-statistic:   134 on 1 and 34 DF,  p-value: 2.409e-13
plot(plm2$residuals ~ plm2$fitted.values)
with(power, plot(Cost ~ Units, ylim = c(25000, 50050)))
lines(sort(predict.lm(plm2)) ~ sort(power$Units), col = "red")
AIC(plm1)
## [1] 663.7554
AIC(plm2)
## [1] 666.2827
```

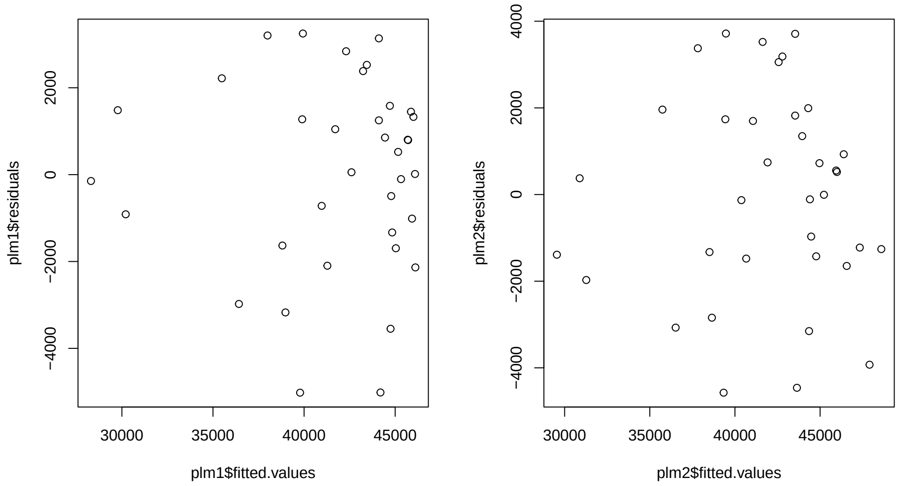




Not a bad fit but not as good as quadratic. AIC also confirms quadratic model as better.

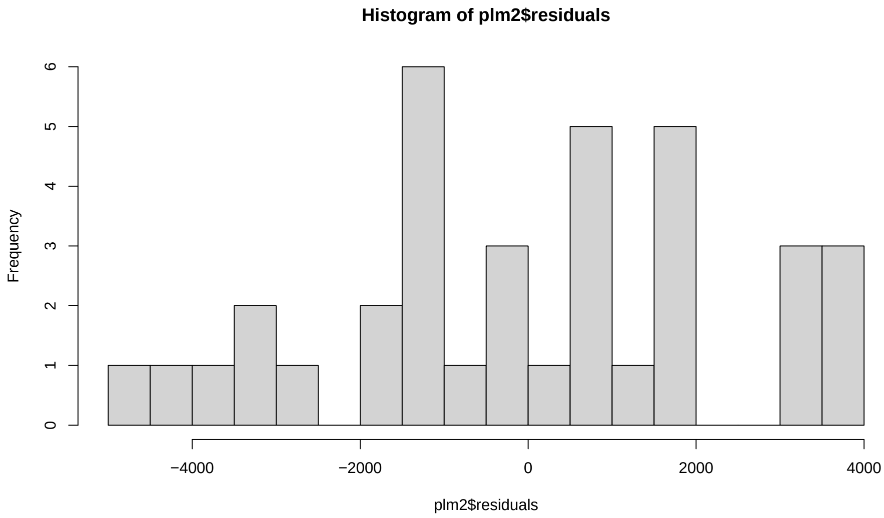
```
oldpar <- par(mfrow = c(1, 2))
plot(plm1$residuals ~ plm1$fitted.values)
plot(plm2$residuals ~ plm2$fitted.values)
```

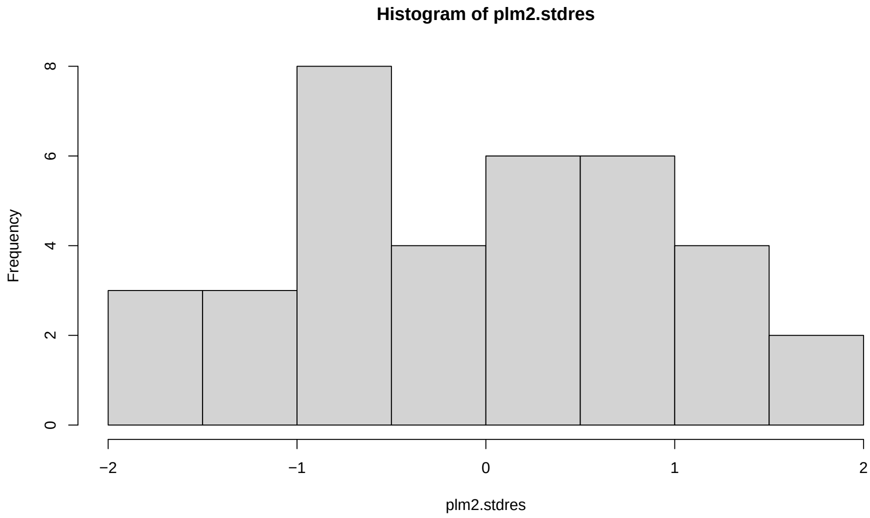
```
par(oldpar)
```

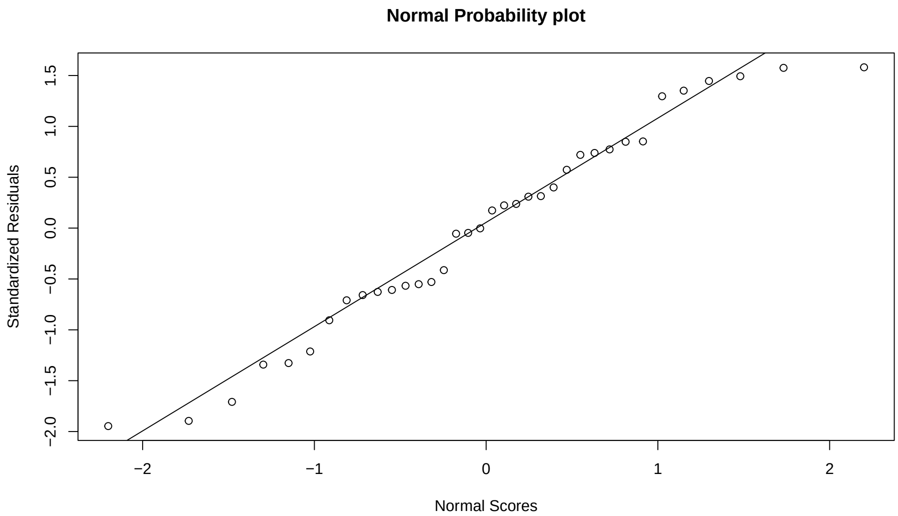


Residual plots look similar, and residuals are fairly large. But note the response value range is in tens of thousands. Let us look at normality assumption.

```
hist(plm2$residuals, nclass = 20)
plm2.stdres = rstandard(plm2)
hist(plm2.stdres)
qqnorm(plm2.stdres, ylab = "Standardized Residuals", xlab = "Normal Scores", main = "Normal Probability p
qqline(plm2.stdres)
```







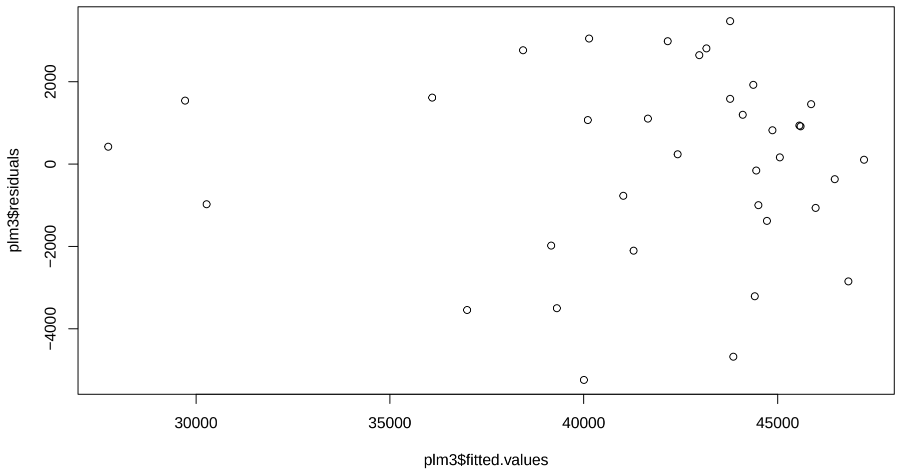
The histogram of residuals does not look to be from a normal distribution. The normal probability plot is expected to be close to a straight line. In the given plot the departures from straight line are not severe, so there is not reason to doubt the normality assumption. The departures are at either end. At both ends the plot flattens off, indicating that the normal scores continue but the residuals stop. This indicates a "cliff", that is, a short tail for the data.

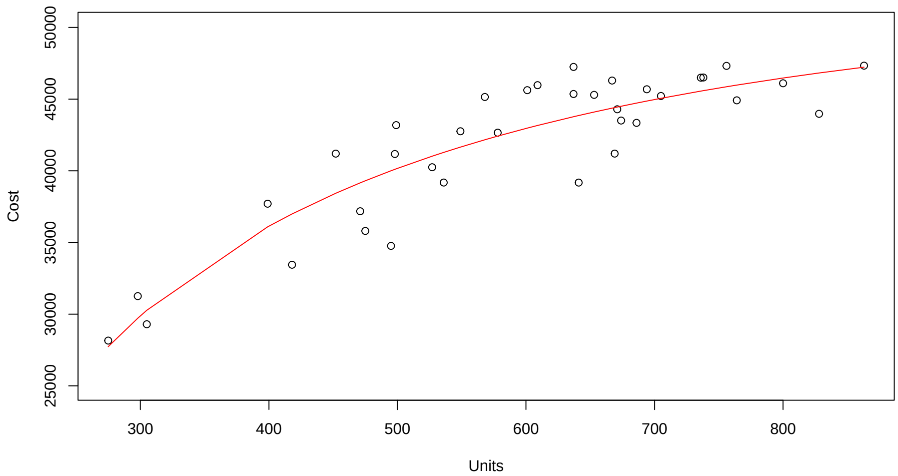
Can this model be improved? Let us try an extra term in the model.

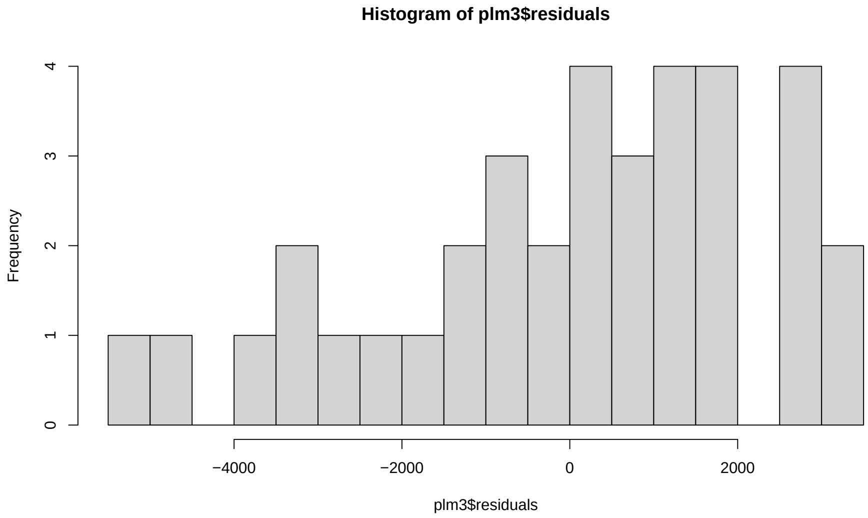
```
plm3 <- lm(Cost ~ I(log(Units)) + I(sqrt(log(Units))), data = power)
summary(plm3)
##
## Call:
## lm(formula = Cost ~ I(log(Units)) + I(sqrt(log(Units))), data = power)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5241.7 -1143.2   329.8  1551.9  3471.1
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -1117903     624539  -1.790   0.0826 .
## I(log(Units))         -153338     100736  -1.522   0.1375
## I(sqrt(log(Units)))    846805     501759   1.688   0.1009
## ---
```

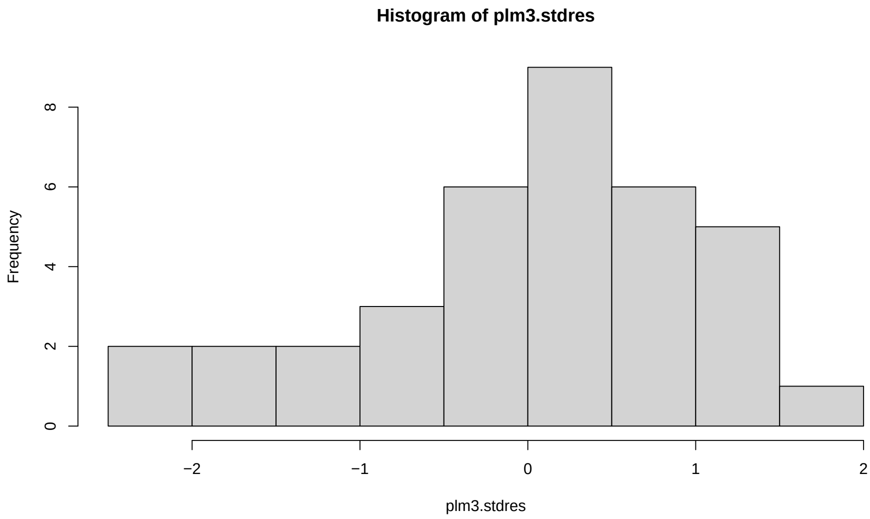


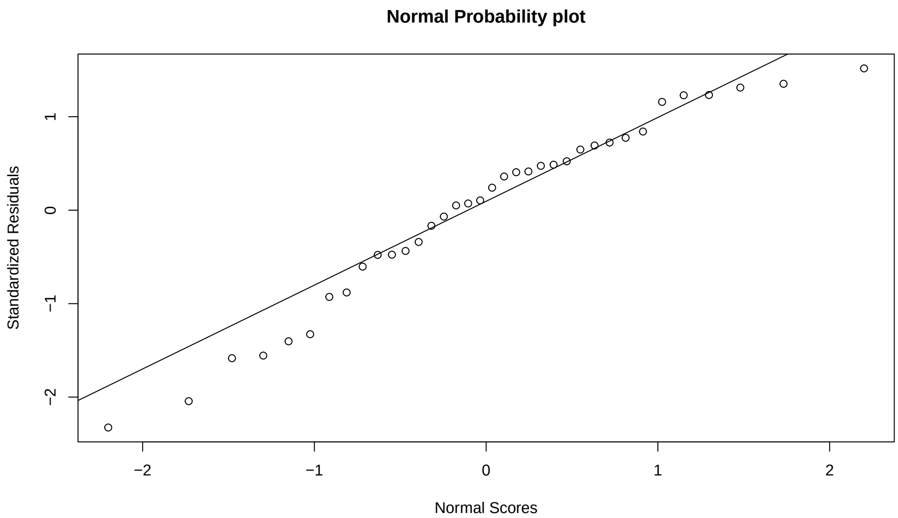
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2330 on 33 degrees of freedom
## Multiple R-squared:  0.8138,Adjusted R-squared:  0.8025
## F-statistic: 72.09 on 2 and 33 DF,  p-value: 9.048e-13
plot(plm3$residuals ~ plm3$fitted.values)
with(power, plot(Cost ~ Units, ylim = c(25000, 50050)))
lines(sort(predict.lm(plm3)) ~ sort(power$Units), col = "red")
AIC(plm1)
## [1] 663.7554
AIC(plm2)
## [1] 666.2827
AIC(plm3)
## [1] 665.3024
hist(plm3$residuals, nclass = 20)
plm3.stdres = rstandard(plm3)
hist(plm3.stdres)
qqnorm(plm3.stdres, ylab = "Standardized Residuals", xlab = "Normal Scores", main = "Normal Probability p
qqline(plm3.stdres)
```











Looks better than the previous model, but harder to interpret! I will be happy with just the log model.

Any other ways of improving the model? Well, if you examine the plot of residuals against fitted values for model 2, you will see some evidence of heterogenous variance. That topic is this week's lecture material. We will cover this in the lab class.

SessionInfo

This document was prepared using the following settings:

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19044)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8  LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.39
##
## loaded via a namespace (and not attached):
##  [1] digest_0.6.29   formatR_1.12    magrittr_2.0.3  evaluate_0.15   highr_0.9
##  [6] rlang_1.0.4     stringi_1.7.6   cli_3.3.0       rstudioapi_0.13 rmarkdown_2.14
## [11] tools_4.2.1     stringr_1.4.0   xfun_0.31       yaml_2.3.5      fastmap_1.1.0
## [16] compiler_4.2.1  htmltools_0.5.3
```

