Linear Models - Final Project

Kogan, Arye
303925671
aryekogan@gmail.com
aryeko/linear models @ GitHub

March 2020

1 Question 1

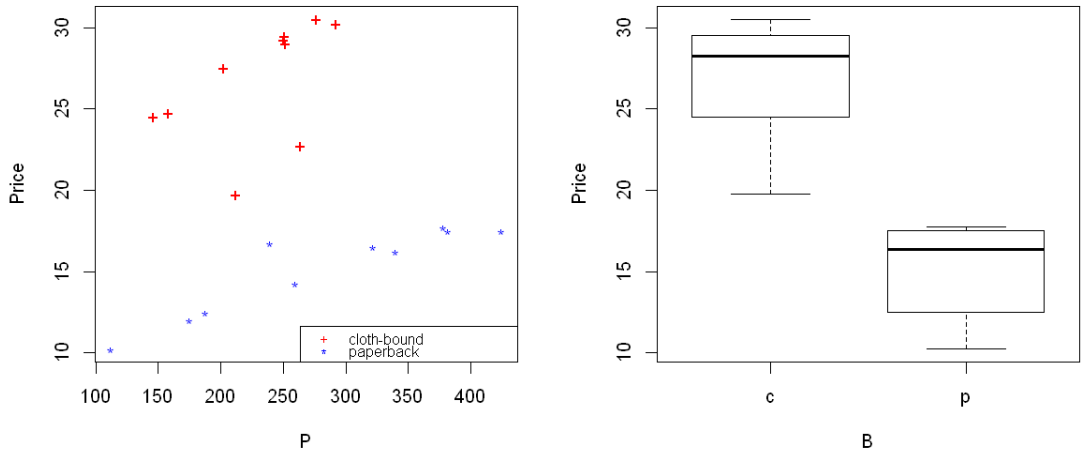
The data in the file Books.dat is compiled from the catalogue of American Goverment books at Spring, 1988. It lists prices Price, number of pages P and the binding B (p - paperback, c -) of books published by a certain publisher.

1. Find a reasonable linear model to this data using a price as the dependent variable and performing appropriate transformations of variables if necessary.
   - Examine the goodness-of-fit of your final model and comment the results.
2. Although most of the data are for books published in 1988, in fact, two of the cloth-bound books were published in 1970's, one of the paperbacks in 1989 and another in 1984.
   - Can you identify them?
   - Delete them from the data and find an adequate linear model for the reduced data set.
   - Did the omitted observations strongly affect the model?
3. Another possible way to reduce the influence of outliers is robust regression.
   - Fit robust regression(s) and comment the results.
4. What model(s) would you introduce to a client?
   - How would you interpret your results to him/her? (he is a complete "amateur" in statistics)
5. Estimate the price of a 200-page book for the two types of binding and give the corresponding 95% prediction intervals.

1.1 Find a reasonable linear model to this data using a price as the dependent variable and performing appropriate transformations of variables if necessary.

- Examine the goodness-of-fit of your final model and comment the results.

[1]:
```
books = read.table("Books.dat", header = T)

books$B <- as.factor(books$B)
head(books)
# Define color for each of the 2 categories
colors <- c("Red", "Blue")
colors <- colors[books$B]

# Define shapes
shapes = c('+', '*')
shapes <- shapes[books$B]

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(Price ~ P, data = books, col=colors, pch=shapes)
legend("bottomright", c("cloth-bound", "paperback"),
       col=c("Red", "Blue"), pch=c('+', '*'), cex=0.8)
plot(Price ~ B, data = books)
```



Looks like a parallel regression model will be appropriate for this data

[2]:
```
lm.fit=lm(Price ~ P + B + P*B, data = books)
# summary(lm.fit)
```

The interaction is not significant, lets test if it is rellevant

$$H_0 : \beta_{P*B} = 0$$

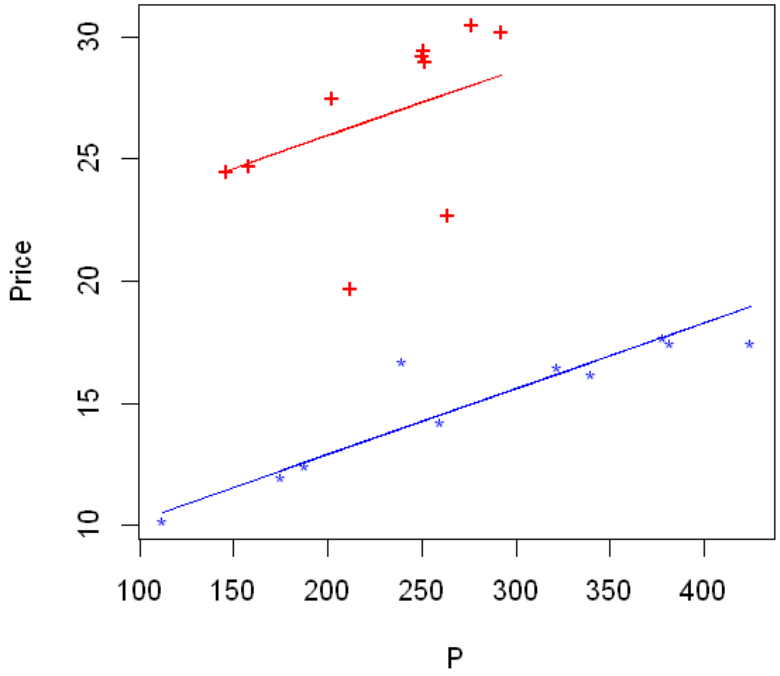
[3]:
```
lm.fit.no.interaction=lm(Price ~ P + B, data = books)
# summary(lm.fit.no.interaction)

anova(lm.fit.no.interaction, lm.fit)
```

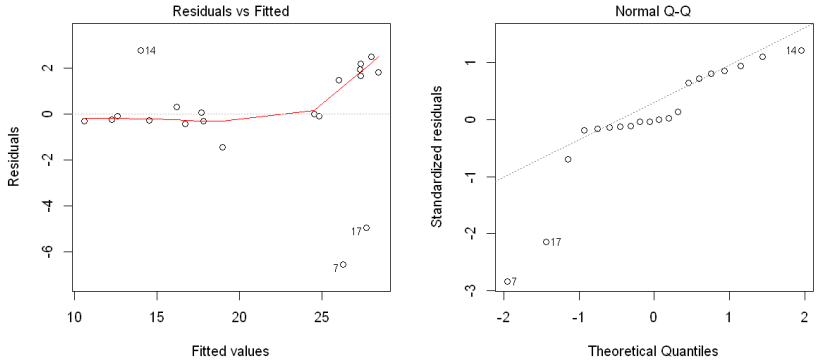
Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
17	100.28849	NA	NA	NA	NA
16	96.34474	1	3.943743	0.6549386	0.4302236

We will not reject H_0 and conclude that the interaction may be equal to zero

We will use the model without the interaction

[4]:
```
options(repr.plot.width=5, repr.plot.height=5)
par(mfrow = c(1,1))
plot(Price ~ P, data = books, col=colors, pch=shapes)
lines(books[books$B=='p',]$P, predict(lm.fit.no.interaction,␣
 ↪books[books$B=='p',]), col="Blue")
lines(books[books$B=='c',]$P, predict(lm.fit.no.interaction,␣
 ↪books[books$B=='c',]), col="Red")

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(lm.fit.no.interaction, which = 1:2)
```



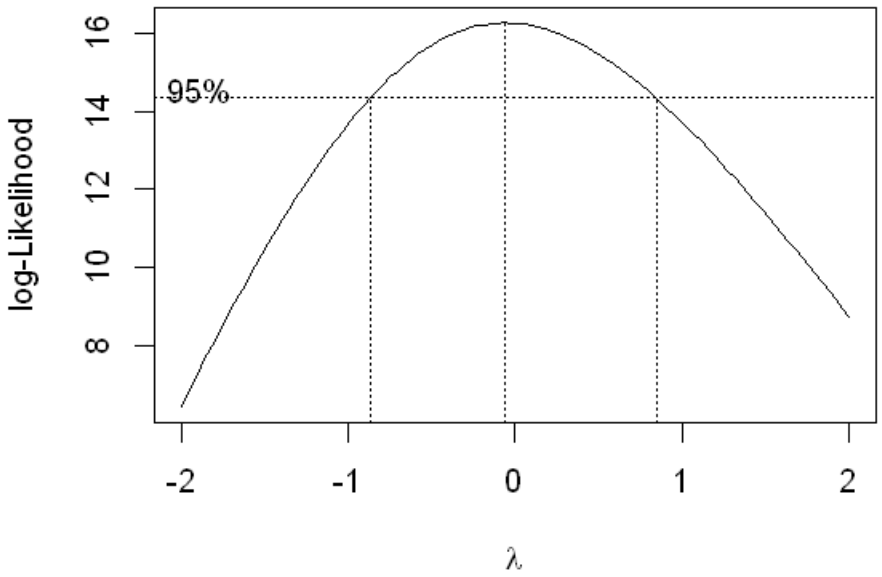


The summary of the model looks fine, there are clearly some outliers.

Looks like the residuals assumption aren't met but this data set is not big...

Let's try to improve it using transformation

[5]:
```
library(MASS)
options(repr.plot.width=5, repr.plot.height=4)
lamb = boxcox(lm.fit.no.interaction, lambda = seq(-2, 2, 1/10), plotit = TRUE,␣
 ↪eps = 1/50, xlab = expression(lambda), ylab = "log-Likelihood")
```



Log transformation to the dependant variable may be a good transformation here

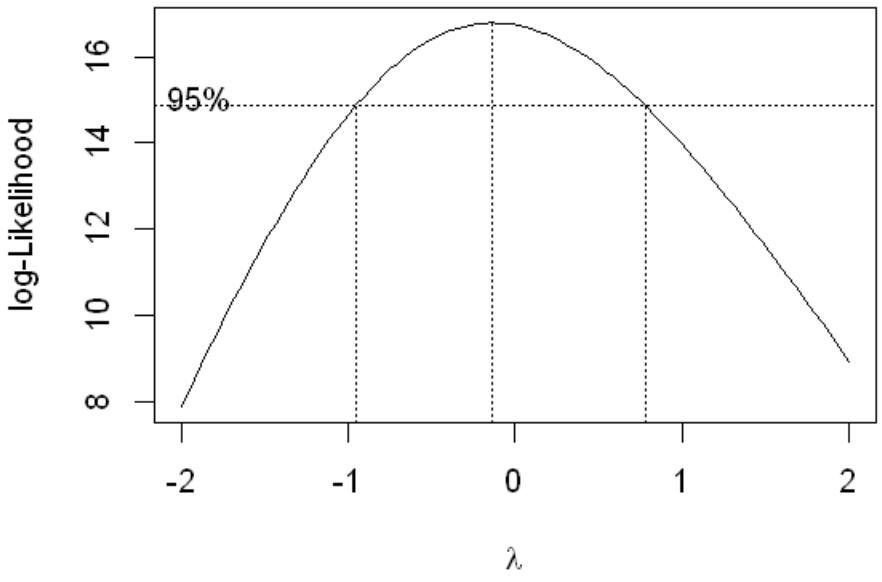
[6]:
```
lm.log.fit=lm(log(Price) ~ P + B, data = books)
# summary(lm.log.fit)

# options(repr.plot.width=10, repr.plot.height=5)
# par(mfrow = c(1,2))
# plot(lm.log.fit, which = 1:2)
```

Looks much better, also the RSS, R squared and coefficients are more significant!

Lets try a log-log model.

First, make sure log transformation is still relevant after log transformation on P

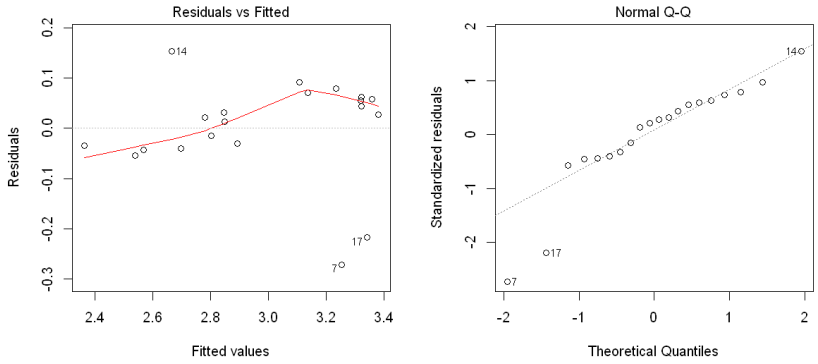
[7]:
```
lm.logP.fit=lm(Price ~ log(P) + B, data = books)
# summary(lm.logP.fit)

options(repr.plot.width=5, repr.plot.height=4)
par(mfrow = c(1,1))
lamb = boxcox(lm.logP.fit, lambda = seq(-2, 2, 1/10), plotit = TRUE, eps = 1/50,␣
 ↪xlab = expression(lambda), ylab = "log-Likelihood")
```



YES! Let's try a log-log model

[8]:
```
lm.loglog.fit=lm(log(Price) ~ log(P) + B, data = books)
summary(lm.loglog.fit)

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(lm.loglog.fit, which = 1:2)
```

```
Call:
lm(formula = log(Price) ~ log(P) + B, data = books)

Residuals:
     Min       1Q   Median       3Q      Max
-0.27165 -0.03651  0.02453  0.05926  0.15323

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.12797    0.39194   2.878   0.0104 *
log(P)       0.39705    0.07211   5.506 3.85e-05 ***
Bp          -0.63889    0.04827 -13.237 2.21e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.105 on 17 degrees of freedom
Multiple R-squared:  0.9144,Adjusted R-squared:  0.9044
F-statistic: 90.82 on 2 and 17 DF,  p-value: 8.418e-10
```



Better in RSS and R squared, but the residuals assumptions aren't met.

We have to retry it after removing the outliers

1.2 Although most of the data are for books published in 1988, in fact, two of the cloth-bound books were published in 1970's, one of the paperbacks in 1989 and another in 1984.

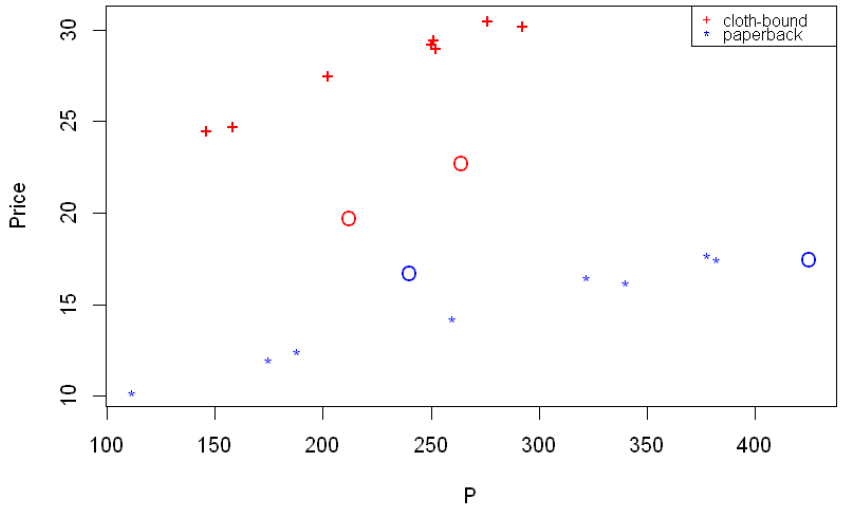
- Can you identify them?
- Delete them from the data and find an adequate linear model for the reduced data set.
- Did the omitted observations strongly affect the model?

First, from the given data, I can't be certain which books wasn't published in 1988, but:

1. From the plot it seems like the books has a very good linear relation (with diffrent slopes for each group) between the price and the number of pages in the book except some observations.
2. Given that we know that all the books was published by the same publisher and in the same year (1988), except exactly 2 books in each group.

I would guess that the observations that doesn't mach the linear relation (see marked with a circle) are the books that wasn't published in the same year (1988)

[9]:
```
options(repr.plot.width=7, repr.plot.height=5)
newshapes <- shapes
newshapes[7] <- 'O'
newshapes[17] <- 'O'
newshapes[14] <- 'O'
newshapes[15] <- 'O'

plot(Price ~ P, data = books, col=colors, pch=newshapes)

legend("topright", c("cloth-bound", "paperback"),
       col=c("Red", "Blue"), pch=c('+', '*'), cex=0.8)
```



Influential observations by cook's distance:

[10]:
```
cooksd <- cooks.distance(lm.loglog.fit)
sample_size <- nrow(books)

# influential row numbers
influential <- as.numeric(names(cooksd)[(cooksd > (4/sample_size))])

cat("influential: ", influential)
```

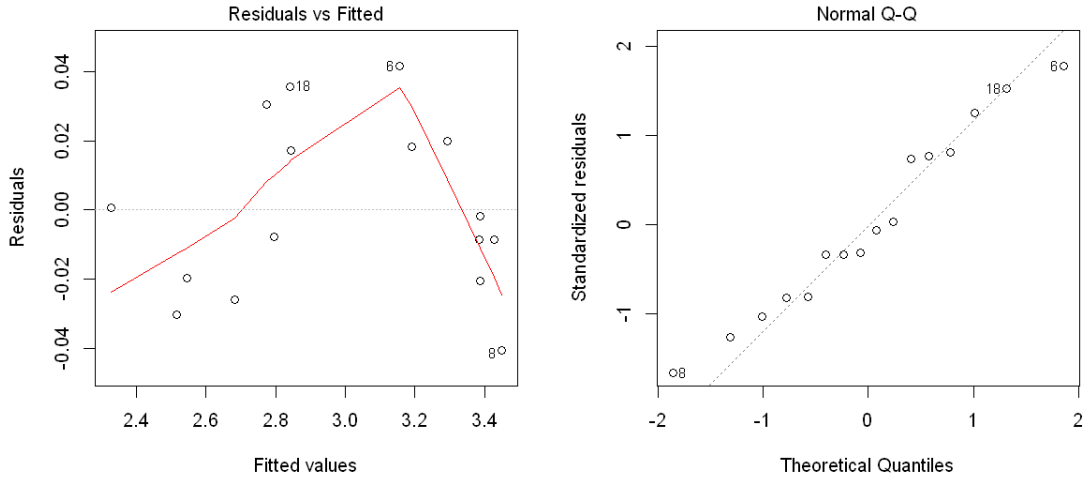
```
influential:  7 17
```

By cooks distance, observations are 7 and 17 are strongly affecting the model

[11]:
```
# Removing Outliers
data_screen <- books[-c(7, 17, 14, 15), ]


# Define color for each of the 2 categories
colors <- c("Red", "Blue")
colors <- colors[data_screen$B]

# Define shapes
shapes = c('+', '*')
shapes <- shapes[data_screen$B]
```

[12]:
```
lm.loglog.screen.fit=lm(log(Price) ~ log(P) + B, data = data_screen)
summary(lm.loglog.screen.fit)

# options(repr.plot.width=5, repr.plot.height=5)
# par(mfrow = c(1,1))
# plot(log(Price) ~ log(P), data = data_screen, col=colors, pch=shapes)
# lines(log(data_screen[data_screen$B=='p',]$P), predict(lm.loglog.screen.fit,␣
 ↪data_screen[data_screen$B=='p',]), col="Blue")
# lines(log(data_screen[data_screen$B=='c',]$P), predict(lm.loglog.screen.fit,␣
 ↪data_screen[data_screen$B=='c',]), col="Red")

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(lm.loglog.screen.fit, which = 1:2)
```

```
Call:
lm(formula = log(Price) ~ log(P) + B, data = data_screen)

Residuals:
      Min        1Q    Median        3Q       Max
-0.040530 -0.019983 -0.004761  0.018704  0.041541

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.05128    0.10662    9.86 2.12e-07 ***
log(P)       0.42256    0.01965   21.50 1.52e-11 ***
Bp          -0.71842    0.01350  -53.22  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.0266 on 13 degrees of freedom
Multiple R-squared:  0.9957,Adjusted R-squared:  0.995
F-statistic:  1496 on 2 and 13 DF,  p-value: 4.313e-16
```



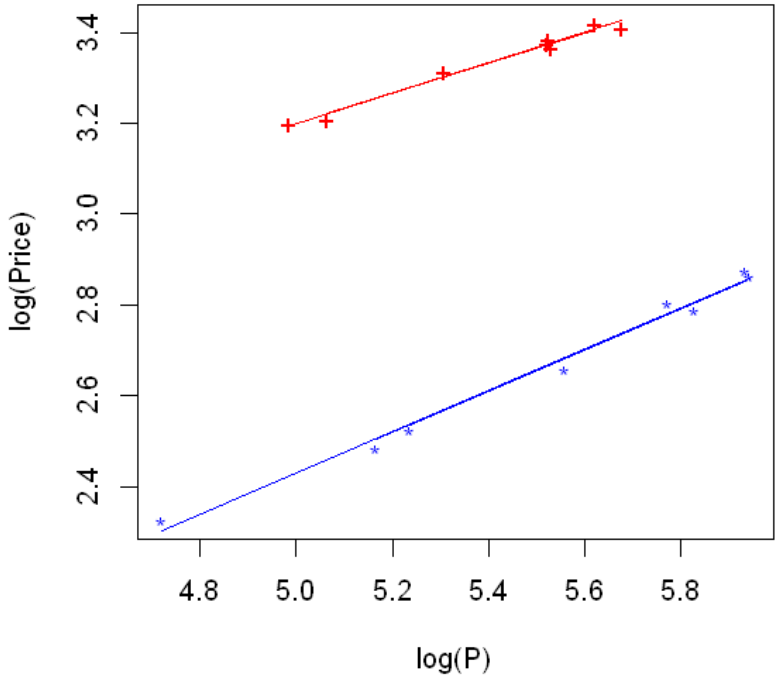
Better now in terms of RSS and R squared

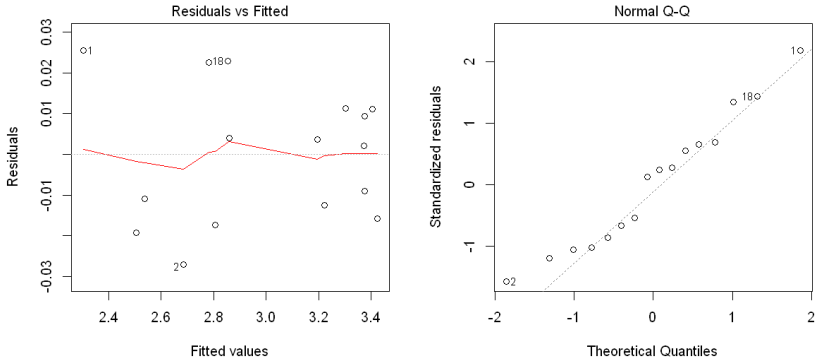
From the residuals plots looks like we forgot an explanatory variable.

Let's try to add the interaction back after removing the outliers

[13]:
```
lm.loglog.screen.full.fit=lm(log(Price) ~ log(P) + B + log(P) * B, data =␣
 ↪data_screen)
summary(lm.loglog.screen.full.fit)

options(repr.plot.width=5, repr.plot.height=5)
par(mfrow = c(1,1))
plot(log(Price) ~ log(P), data = data_screen, col=colors, pch=shapes)
lines(log(data_screen[data_screen$B=='p',]$P), predict(lm.loglog.screen.full.
 ↪fit, data_screen[data_screen$B=='p',]), col="Blue")
lines(log(data_screen[data_screen$B=='c',]$P), predict(lm.loglog.screen.full.
 ↪fit, data_screen[data_screen$B=='c',]), col="Red")

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(lm.loglog.screen.full.fit, which = 1:2)
```

```
Call:
lm(formula = log(Price) ~ log(P) + B + log(P) * B, data = data_screen)

Residuals:
      Min        1Q    Median        3Q       Max
-0.027044 -0.013326  0.002866  0.011150  0.025456

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.53956    0.14557  10.576 1.95e-07 ***
log(P)       0.33219    0.02691  12.343 3.53e-08 ***
Bp          -1.37786    0.16972  -8.118 3.23e-06 ***
log(P):Bp    0.12137    0.03119   3.891  0.00214 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01841 on 12 degrees of freedom
Multiple R-squared:  0.9981,Adjusted R-squared:  0.9976
F-statistic:  2087 on 3 and 12 DF,  p-value: < 2.2e-16
```





The residuals assumptions are much better now, and the interaction coefficient is significant!

Let's compare the final models

[14]:
```
crossVal <- function(lm.object)
{
    # Computes the cross-validation (CV), the generalized cross-validation (GCV)␣
 ↪and the cross-validation correlation coefficient (R^2_{CV}) for the specified␣
 ↪linear model
    res <- lm.object$residuals
    y <- lm.object$fitted.values + lm.object$residuals
    h <- lm.influence(lm.object)$hat
    n <- length(y)
    cv <- mean(res^2/(1 - h)^2)
    gcv <- (n * sum(res^2))/lm.object$df^2
    r2cv <- cor(y, y - res/(1 - h))^2
    return(cbind(cv, gcv, r2cv))
}
```

[15]:
```
lm.loglog.screen.full.fit.crossVal <- crossVal(lm.loglog.screen.full.fit)
lm.loglog.screen.fit.crossVal <- crossVal(lm.loglog.screen.fit)
models <- c("full log log model", "log log model")

results <- cbind(models,
                 rbind(lm.loglog.screen.full.fit.crossVal, lm.loglog.screen.fit.
 ↪crossVal))
results
```

models	cv	gcv	r2cv
full log log model	0.000591486576133862	0.000451903435589474	0.99574477928641
log log model	0.000864445412911331	0.000870942028517718	0.993499122250704

Both models are almost the same, but the full log log model meets the residuals assumption much better

SUMMARY

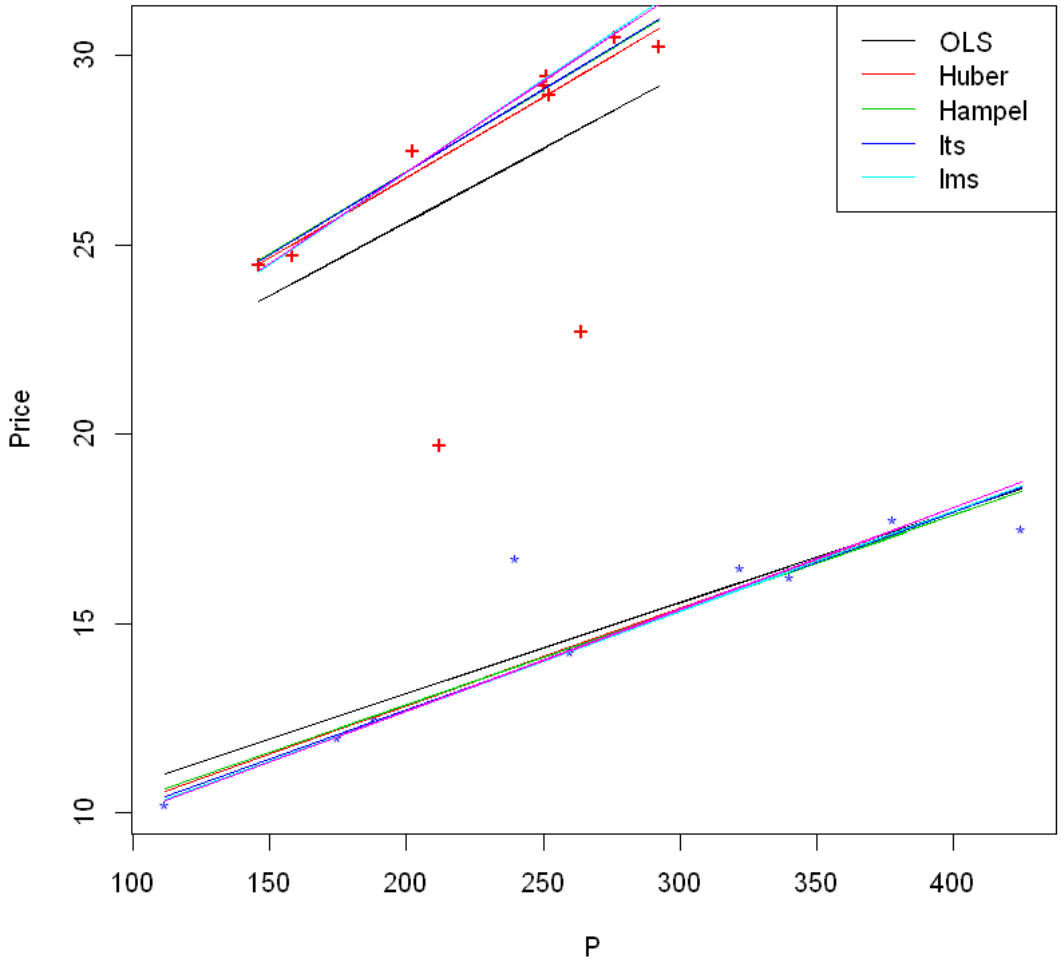
The full log log model is the choosen model for this data

1.3 Another possible way to reduce the influence of outliers is robust regression.

- Fit robust regression(s) and comment the results.

[16]:
```
mod.huber <- rlm(Price ~ P + B + P * B, data = books, psi = psi.huber)
mod.hampel <- rlm(Price ~ P + B + P * B, data = books, psi = psi.hampel)
mod.tukey <- rlm(Price ~ P + B + P * B, data = books, psi = psi.bisquare)
mod.lms <- lqs(Price ~ P + B + P * B, data = books, method = "lms")
mod.lts <- lqs(Price ~ P + B + P * B, data = books, method = "lts")
```

[17]:
```
colors <- c("Red", "Blue")
colors <- colors[books$B]
shapes = c('+', '*')
shapes <- shapes[books$B]
options(repr.plot.width=7, repr.plot.height=7)
par(mfrow = c(1,1))
plot(Price ~ P, data = books, col=colors, pch=shapes)
lines(books[books$B=='p',]$P, predict(lm.fit, books[books$B=='p',]), col=1)
lines(books[books$B=='c',]$P, predict(lm.fit, books[books$B=='c',]), col=1)

lines(books[books$B=='p',]$P, predict(mod.huber, books[books$B=='p',]), col=2)
lines(books[books$B=='c',]$P, predict(mod.huber, books[books$B=='c',]), col=2)

lines(books[books$B=='p',]$P, predict(mod.hampel, books[books$B=='p',]), col=3)
lines(books[books$B=='c',]$P, predict(mod.hampel, books[books$B=='c',]), col=3)

lines(books[books$B=='p',]$P, predict(mod.tukey, books[books$B=='p',]), col=4)
lines(books[books$B=='c',]$P, predict(mod.tukey, books[books$B=='c',]), col=4)

lines(books[books$B=='p',]$P, predict(mod.lms, books[books$B=='p',]), col=5)
lines(books[books$B=='c',]$P, predict(mod.lms, books[books$B=='c',]), col=5)

lines(books[books$B=='p',]$P, predict(mod.lts, books[books$B=='p',]), col=6)
lines(books[books$B=='c',]$P, predict(mod.lts, books[books$B=='c',]), col=6)

legend(x="topright", legend = c("OLS", "Huber", "Hampel", "lts", "lms"), col=1:
↪6, lty=1)
```



Let's compare the models using leave one out CV

[18]:
```
none <- function(x){ return (x)}

lpo <- function(p = 1, un.trans = none, data = books, fit.func, ...){
    # A general "leave p out" CV calculation funcion
    n = nrow(data)
    pred = rep(0, n)
    step = p
    for(i in seq(1,n, by = step)){
        fit <- fit.func(..., data = data[-(i:(i+p-1)), ])
        pred[(i:(i+p-1))] <- un.trans(predict(fit, data[(i:(i+p-1)),-1]))
    }
    press <- sum((data[,1] - pred)^2)
    cv = round(press/n, digits = 5)

    return(cv)
}
```

[19]:
```
mod.huber.cv <- lpo(fit.func = rlm, formula = Price ~ P + B + P * B, psi = psi.
 ↪huber)
mod.hampel.cv <- lpo(fit.func = rlm, formula = Price ~ P + B + P * B, psi = psi.
 ↪hampel)
mod.tukey.cv <- lpo(fit.func = rlm, formula = Price ~ P + B + P * B, psi = psi.
 ↪bisquare)
mod.lms.cv <- lpo(fit.func = lqs, formula = Price ~ P + B + P * B, method =␣
 ↪"lms")
mod.lts.cv <- lpo(fit.func = lqs, formula = Price ~ P + B + P * B, method =␣
 ↪"lts")

# Note that I'm using un.trans to go back to the original scale
mod.loglog.cv <- lpo(un.trans = exp, fit.func = lm, formula = log(Price) ~␣
 ↪log(P) + B + log(P) * B)


models <- c("full log log model",
            "Huber",
            "Hampel",
            "Tukey",
            "lms",
            "lts"
           )
cv.original.scale <- rbind(mod.loglog.cv,
                       mod.huber.cv,
                       mod.hampel.cv,
                       mod.tukey.cv,
                       mod.lms.cv,
                       mod.lts.cv
                      )
```

```
results <- cbind(models, cv.original.scale)

colnames(results) <- c("Model", "CV on original scale")

results
```

	Model	CV on original scale
mod.loglog.cv	full log log model	6.72315
mod.huber.cv	Huber	6.02833
mod.hampel.cv	Hampel	6.09954
mod.tukey.cv	Tukey	6.09151
mod.lms.cv	lms	6.34038
mod.lts.cv	lts	6.33307

What about the model without the outliers?

[20]:
```
mod.loglog.screen.cv <- lpo(data = data_screen, un.trans = exp, fit.func = lm,
                            formula = log(Price) ~ log(P) + B + log(P) * B)

cat("cv: ", mod.loglog.screen.cv)
```

```
cv:  0.16211
```

Looks better than the robust models, but this cross validation didn't included the omited observations!

Let's add the mean prediction error to model's cv (for a fair comparision)

[21]:
```
outliers = c(7,17,14,15)
outliers.mean.error = round(
    mean((books[outliers,1] - exp(predict(lm.loglog.screen.full.fit,␣
 ↪books[outliers,-1])))^2)
                          , digits = 5)

mod.loglog.screen.cv = (mod.loglog.screen.cv + outliers.mean.error)/2
cat("cv: ", mod.loglog.screen.cv)
```

```
cv:  14.86557
```

Now we can see that the model without the outliers has a significant error

(compared to the robust regression models) when trying to predict the outliers

This model is higly overfitted to the screend data

1.4 What model(s) would you introduce to a client?

- How would you interpret your results to him/her? (he is a complete "amateur" in statistics)

Ill present the Huber model to the client

since it has the best percision and it is robust for observations with big measurment errors

Client interpertation:

We have a great model for predicting book price given its pages count and binding.

The model is very accurate and is not biased by mistakes in the data.

1.5 Estimate the price of a 200-page book for the two types of binding and give the corresponding 95% prediction intervals.

[23]:
```
data <- data.frame(books[1:2,-1])
data$P <- 200
data$B <- c('p', 'c')

prediction.intervals <- predict(mod.huber, data, interval = "prediction")
# prediction.intervals

cat("For paperback bindind, the price estimation is: ", prediction.
 ↪intervals[1,1],
    ". CI=(",prediction.intervals[1,2:3],")\n")

cat("For cloth bindind, the price estimation is: ", prediction.intervals[2,1],
    ". CI=(",prediction.intervals[2,2:3],")\n")
```

```
For paperback bindind, the price estimation is:  12.84051 . CI=( 11.74964
13.93138 )
For cloth bindind, the price estimation is:  26.77525 . CI=( 25.6976 27.8529 )
```

2 Question 2.

The file Girls.dat contains the data on the exercise histories of 138 teenaged girls hospitalized for eating disorders, and a group of 93 "control" subjects. The variables are:

- **subject** - an identification code;
  - there are several observations for each subject, but because the girls were hospitalized at different ages, the number of observations, and the age at the last observation, vary.
- **age** - the subject's age in years at the time of observation;
  - all but the last observation for each subject were collected retrospectively at intervals of two years, starting at 8.
- **exercise** - the amount of exercise in which the subject engaged, expessed as estimated hours per week
- **group** - a factor indicating whether the subject is "patient" or "control"

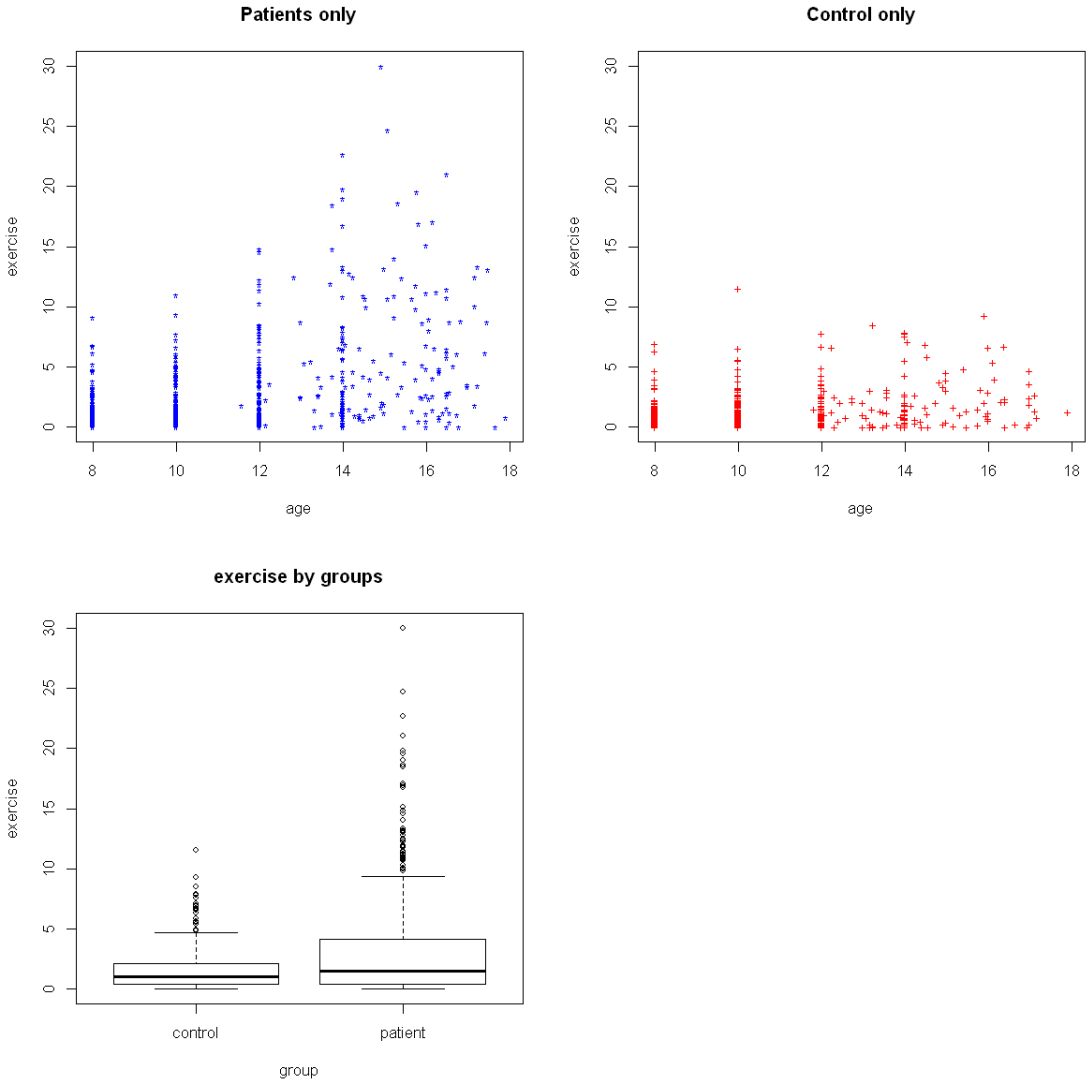
1. Perform initial examination of the data and make preliminary conclusions about the relationship of exercise to age for the two groups.
2. Fit an appropriate model performing transformations of original variables if necessary. Comment the results.
3. Is the relationship of exercise to age different in both groups?
4. Whether the amount of weekly hours of exercises does not change with the age for the control group?
5. Estimate the expected difference in the amount of weekly hours of exercises between the two groups of girls at age 15.

[1]:
```
girls = read.table("Girls.dat", header = T)

girls$group <- as.factor(girls$group)
head(girls)

# Define color for each of the 2 categories
colors <- c("Red", "Blue")
colors <- colors[girls$group]

# Define shapes
shapes = c('+', '*')
shapes <- shapes[girls$group]
```

subject	age	exercise	group
100	8.00	2.71	patient
100	10.00	1.94	patient
100	12.00	2.36	patient
100	14.00	1.54	patient
100	15.92	8.63	patient
101	8.00	0.14	patient

2.1 Perform initial examination of the data and make preliminary conclusions about the relationship of exercise to age for the two groups.

[2]:
```
options(repr.plot.width=10, repr.plot.height=10)
par(mfrow = c(2,2))
plot(exercise ~ age, data = girls, main = "Patients only", ylim=c(0,30), subset␣
 ↪= girls$group == 'patient', col=colors, pch=shapes)
plot(exercise ~ age, data = girls, main = "Control only", ylim=c(0,30), subset =␣
 ↪girls$group == 'control', col=colors, pch=shapes)
plot(exercise ~ group, data = girls, main = "exercise by groups", ylim=c(0,30))
```



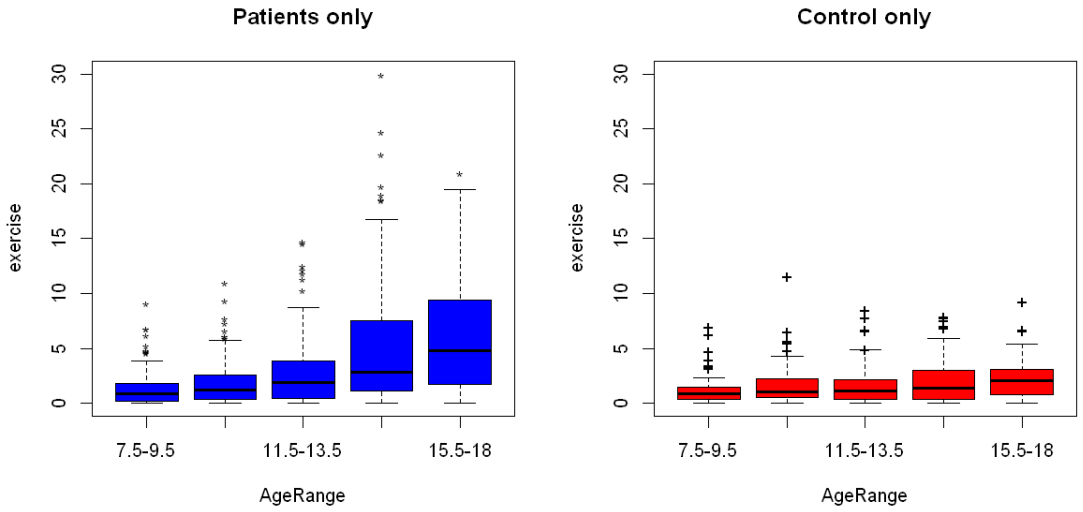
Seems like part of the girls in the 'patient' group tend to excercise more when their age increases, while the patients in the 'control' group seems to NOT excercise more.

2.1.1 Group oservations by age range

Because the observations was sampled every 2 years, we'll use the following age ranges:

- 7.5-9.5
- 9.5-11.5
- 11.5-13.5
- 13.5-15.5
- 15.5-18

[4]:
```
library(dplyr)
girls.age.range <- girls %>% group_by(AgeRange = cut(age, breaks = c(7.5,9.5,11.
↪5,13.5,15.5,18),
                        labels = c("7.5-9.5", "9.5-11.5", "11.5-13.5", "13.5-15.
↪5", "15.5-18")))

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(exercise ~ AgeRange, data = girls.age.range, main = "Patients only",␣
↪ylim=c(0,30), subset = girls$group == 'patient', col=colors, pch=shapes)
plot(exercise ~ AgeRange, data = girls.age.range, main = "Control only",␣
↪ylim=c(0,30), subset = girls$group == 'control', col=colors, pch=shapes)
```



When grouping the girls by age ranges, it is easyer to see the trend.

The exercise groth trend in the patients group looks exponential.

2.2 Fit an appropriate model performing transformations of original variables if necessary

- Comment the results

Given that in each group, there is a difference in the excersice mean between ages and different slopes, I would use a full model using subject's effect as a random effect

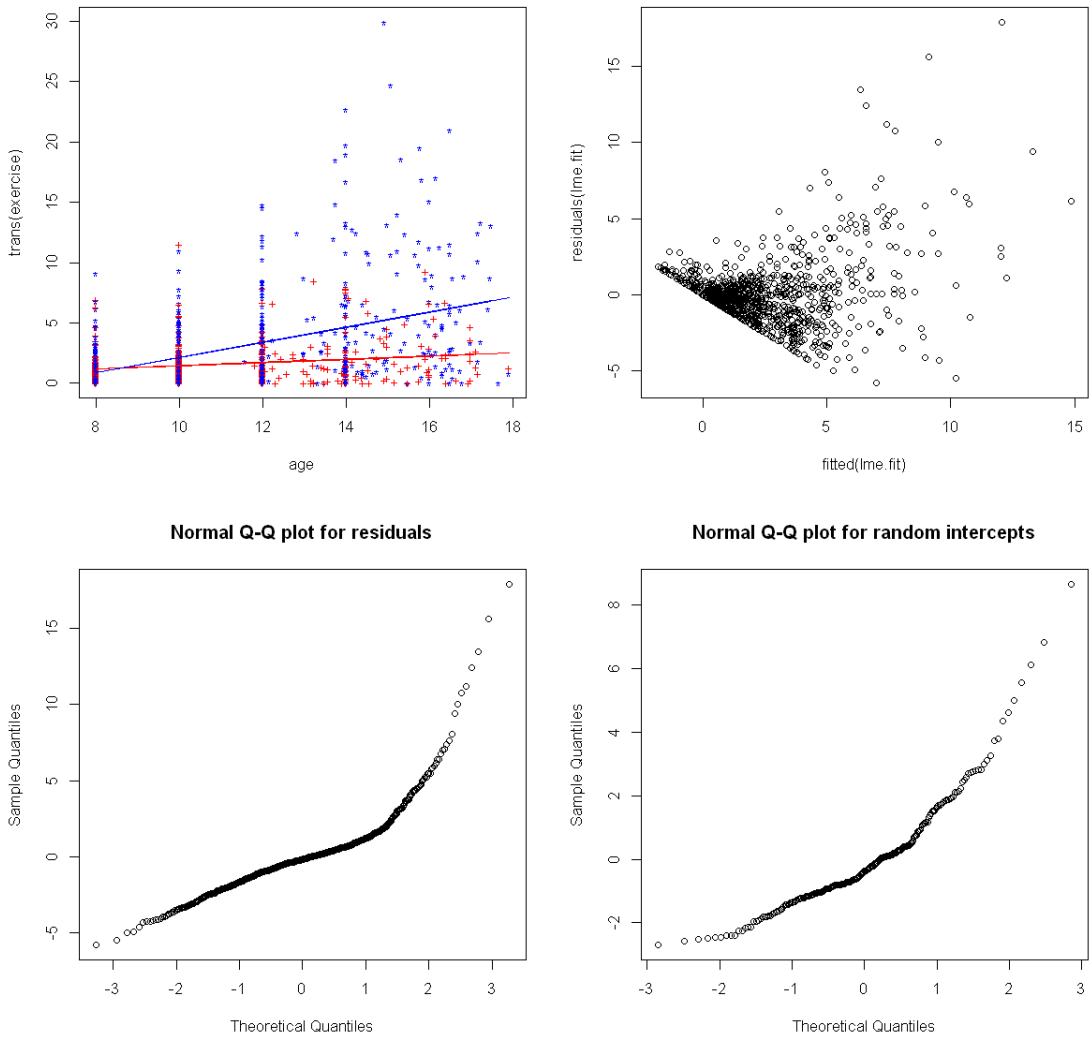
[18]:
```
none <- function(x){ return (x)}

girls.lme.plot <- function(lme.fit, trans = none){
    # A general function to plot girls data for a given model and its error
    options(repr.plot.width=10, repr.plot.height=10)
    par(mfrow = c(2,2))

    sub.control = lme.fit$data$group == 'control'
    sub.patient = lme.fit$data$group == 'patient'

    plot(trans(exercise) ~ age, data = lme.fit$data, col=colors, pch=shapes)
    lines(lme.fit$data$age[sub.control], lme.fit$fitted[sub.control, 1]␣
 ↪,col="red")
    lines(lme.fit$data$age[sub.patient], lme.fit$fitted[sub.patient, 1]␣
 ↪,col="blue")

    plot(fitted(lme.fit),residuals(lme.fit))
    qqnorm(residuals(lme.fit),main = "Normal Q-Q plot for residuals")
    qqnorm(ranef(lme.fit)[,1],main = "Normal Q-Q plot for random intercepts")
}
```

2.2.1 Random intercept, ML

$$exercise_{ij}|\alpha_{0i} \sim N(\beta_0 + \beta_1 age_{ij} + \beta_2 group_{ij} + \beta_3 group_{ij} : age_{ij} + \alpha_{0i}, \sigma^2)$$

$$\alpha_{0i} \sim N(0, \sigma^2_{\alpha_0})$$

[19]:
```
library("nlme")
fit1 <- lme(exercise ~ age*group, random = ~ 1|subject, method = "ML", data =␣
 ↪girls)
# summary(fit1,cor=F)

girls.lme.plot(fit1)
```



All the coefficients are significat except the intercept.

The normallity with zero mean residuals assumption is clearly not met

The exercise growth trend seems to be exponential - we'll try a log transformation.

Since we have 0 (zero) exercise values, we'll use [$log(exercise + 1)$] transformation

This transformation will keep variance and the zero observations as zero:

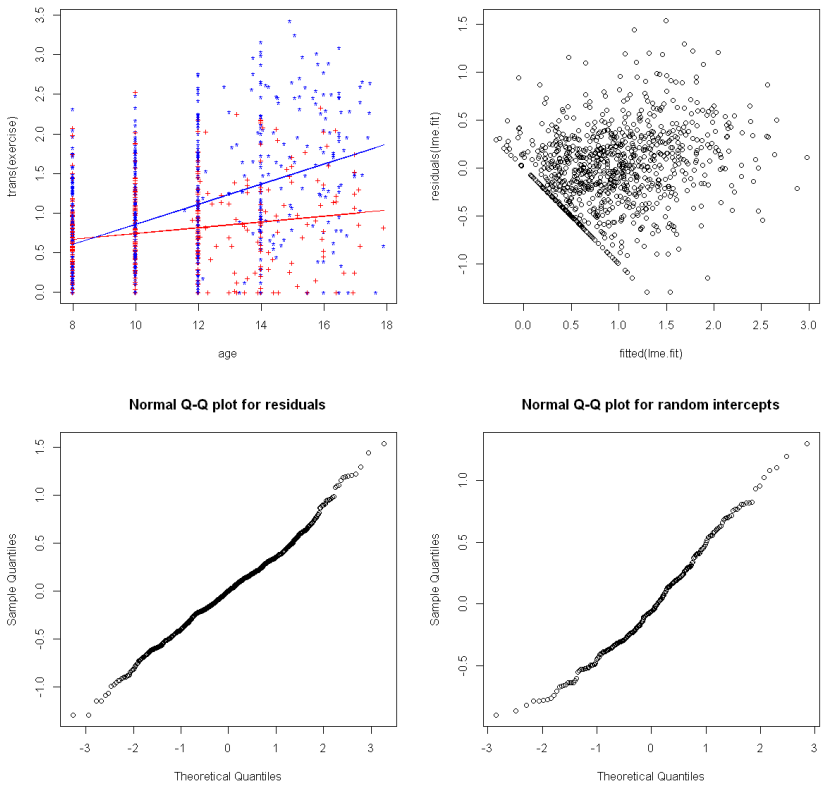
$$log(exercise + 1)|_{exrecise=0} = 0$$

2.2.2 Random intercept with log transformation, ML

$$log(exercise_{ij})|\alpha_{0i} \sim N(\beta_0 + \beta_1 age_{ij} + \beta_2 group_{ij} + \beta_3 group_{ij} : age_{ij} + \alpha_{0i} - 1, \sigma^2)$$

$$\alpha_{0i} \sim N(0, \sigma^2_{\alpha_0})$$

[20]:
```
fit1.log <- lme(log(exercise + I(1)) ~ age*group, random = ~ 1|subject, method =␣
 ↪"ML", data = girls)
# summary(fit1.log,cor=F)

my.trans <- function(x){ return (log(x+1))}

girls.lme.plot(fit1.log, trans = my.trans)
```



Now all the coefficients are significat including the intercept and the residuals looks better

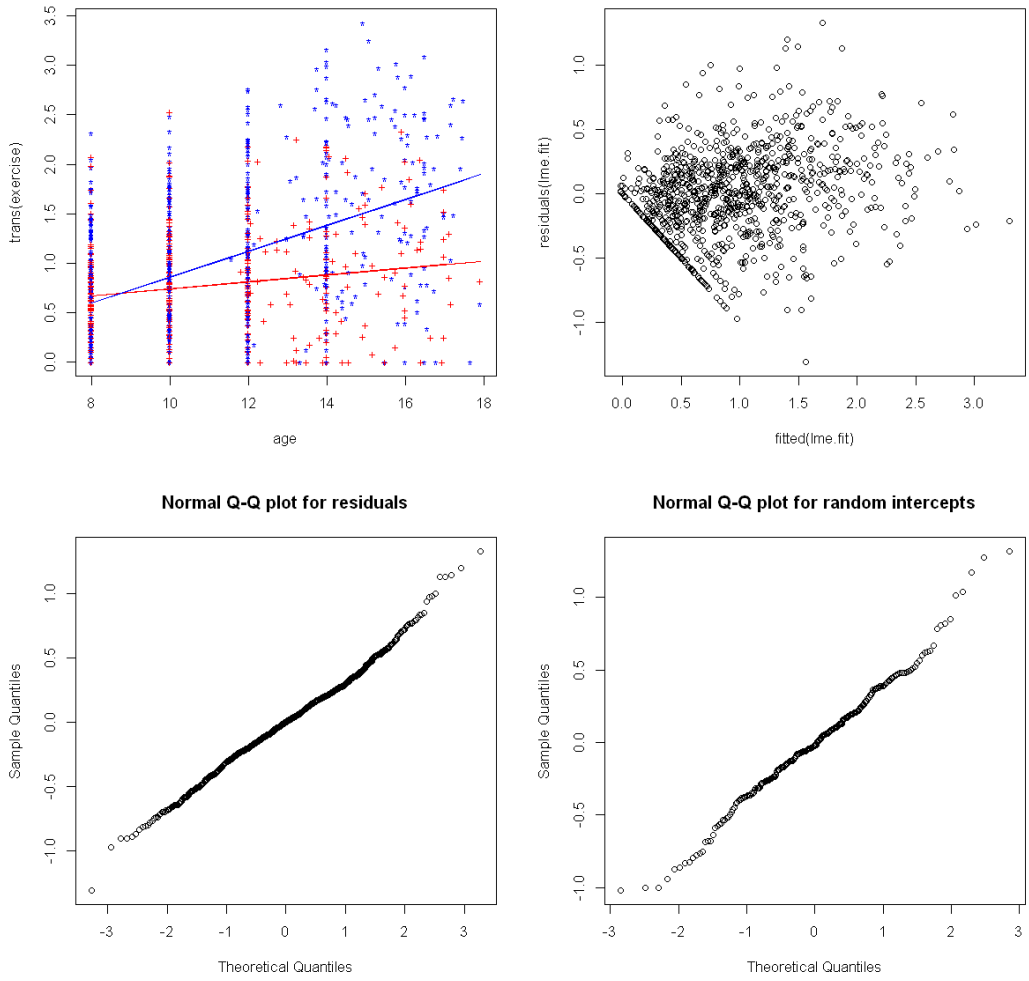
Let's try to add a random slope

2.2.3 Random intercept and random slope with log transformation, ML

$$log(exercise_{ij})|\alpha_{0i}, \alpha_{1i} \sim N(\beta_0 + \beta_1 age_{ij} + \beta_2 group_{ij} + \beta_3 group_{ij} : age_{ij} + \alpha_{0i} + \alpha_{1i} age_{ij} - 1, \sigma^2)$$
$$(\alpha_{0i}, \alpha_{1i})^t \sim N(0, V)$$

[22]:
```
fit2.log <- lme(log(exercise + I(1)) ~ age*group, random = ~ age|subject, method␣
 ↪= "ML", data = girls)
summary(fit2.log,cor=F)

girls.lme.plot(fit2.log, trans = my.trans)
```

```
Linear mixed-effects model fit by maximum likelihood
 Data: girls
       AIC      BIC    logLik
  1539.141 1577.951 -761.5705

Random effects:
 Formula: ~age | subject
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev     Corr
(Intercept) 0.66997684 (Intr)
age         0.06484378 -0.747
Residual    0.40120992

Fixed effects: log(exercise + I(1)) ~ age * group
                     Value  Std.Error  DF   t-value p-value
(Intercept)       0.3951994 0.11964946 712  3.302977  0.0010
age               0.0348769 0.01096666 712  3.180268  0.0015
grouppatient     -0.8416968 0.15161763 229 -5.551444  0.0000
age:grouppatient  0.0958513 0.01381010 712  6.940668  0.0000
 Correlation:
                  (Intr) age    grpptn
age               -0.892
grouppatient      -0.789  0.704
age:grouppatient  0.708 -0.794 -0.887

Standardized Within-Group Residuals:
         Min           Q1          Med           Q3          Max
-3.258207791 -0.518976515  0.004806002  0.509390532  3.308646955

Number of Observations: 945
Number of Groups: 231
```



Also here all the coefficients are significat and the residuals very similar

Let's check if the random slope is relevant by testing the hypothesis:

$$H_0 : \sigma^2_{\alpha_1 patient} = \sigma^2_{\alpha_1 control}$$

[23]:
```
print(anova(fit1.log, fit2.log))
```

```
          Model df      AIC      BIC    logLik   Test  L.Ratio p-value
fit1.log      1  6 1584.486 1613.593 -786.2427
fit2.log      2  8 1539.141 1577.951 -761.5705 1 vs 2 49.34438  <.0001
```

We will reject H_0 and conclude with very high confidence that the random slope variance is different between the groups

2.3 Is the relationship of exercise to age different in both groups?

The relationship of the age and the exersice is clearly different between the groups.

From the conclusion in the previous section, we can see that the random slope (for age) variance is different between both groups

$$\sigma^2_{age:control} \neq \sigma^2_{age:patient}$$

2.4 Whether the amount of weekly hours of exercises does not change with the age for the control group?

Let's test the following hypothesis:

$$H_0 : \sigma^2_{age:control} = 0$$

[24]:
```
fit.cont        <- lme(log(exercise + I(1)) ~ age,
                       random = ~ 1|subject, method = "ML", data = girls,␣
 ↪subset=group=='control')
fit.cont.rslope <- lme(log(exercise + I(1)) ~ age,
                       random = ~ age|subject, method = "ML", data = girls,␣
 ↪subset=group=='control')

print(anova(fit.cont, fit.cont.rslope))
```

```
               Model df      AIC      BIC    logLik   Test  L.Ratio p-value
fit.cont           1  4 474.5766 490.1099 -233.2883
fit.cont.rslope    2  6 455.2744 478.5743 -221.6372 1 vs 2 23.30219  <.0001
```

We will reject H_0 and conclude with very high confidence

that the weekly hours of exercise in the control group do change with age

2.5 Estimate the expected difference in the amount of weekly hours of exercises between the two groups of girls at age 15.

$$\hat{E}(Y_{patient}|_{age=15} - Y_{control}|_{age=15}) = \widehat{Y_{patient}|_{age=15} - Y_{control}|_{age=15}} = \hat{Y}_{patient}|_{age=15} - \hat{Y}_{control}|_{age=15}$$

Until now we used the models for hypothesis testing. For estimation we'll use REML

[25]:
```
fit.reml <- lme(log(exercise + I(1)) ~ age*group, random = ~ age|subject, method␣
 ↪= "REML", data = girls)
test.data <- girls[1:2, -3]
test.data$subject <- c(1,2) #new subject
test.data$age <- c(15, 15)
test.data$group <- c('patient', 'control')

# predict with level zero - corresponding to the population predictions
pred <- predict(fit.reml, test.data, level = 0)

# un-transform log(exercise + 1) by exp(prediction) - 1
pred <- exp(pred) - 1
cat("expected difference in exercise weekly hours at age 15: ", pred[1] -␣
 ↪pred[2])
```

```
expected difference in exercise weekly hours at age 15:  2.042764
```

The expected difference in the amount of weekly hours of exercises between the two groups of girls at age 15 is:

$$\hat{E}(Y_{patient}|_{age=15} - Y_{control}|_{age=15}) = 2.042$$

3 Question 3.

The dataset Boston from the library MASS consists of 506 median prices of owner-occupoed homes in $1000s (medv) in various places in Boston. Alongside with price, the dataset also provide various geographic and socio-economic information such as:

- crim – per capita crime rate by town
- zn – proportion of residential land zoned for lots over 25,000 sq.ft
- indus – proportion of non-retail business acres per town
- chas – Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- nox – nitrogen oxides concentration (parts per 10 million)
- rm – average number of rooms per dwelling
- age – proportion of owner-occupied units built prior to 1940
- dis – weighted mean of distances to five Boston employment centres
- rad – index of accessibility to radial highways
- tax – full-value property-tax rate per $10,000
- ptratio – pupil-teacher ratio by town
- black – 1000(Bk0.63)^2 where Bk is the proportion of blacks by town
- lstat – lower status of the population (percent)

The goal is to find the relations between these factors and the house prices.

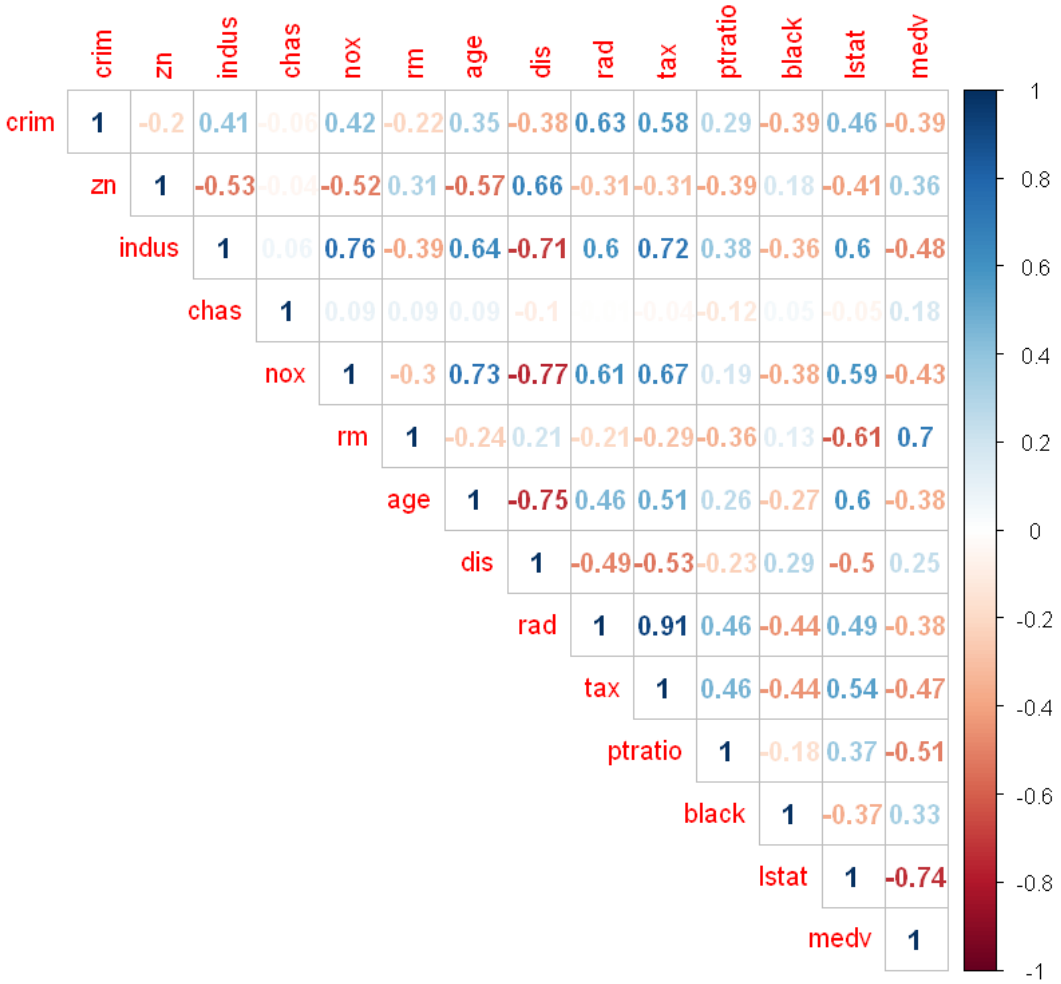
1. Analyze the data to get some first impressions and make some preliminary comments.
2. Split randomly (why?) the data into a training and test sets of 80% and 20% of the data respectively. Put a test set meanwhile aside and consider a training set:
   - Start from the main effects model, verify its adequacy
   - If you're not satisfied, try to add paired interactions, perform transformations if necessary.
   - Perform model selection w.r.t. various model selection criteria. Compare the resulting models and comment the results.
3. Test and compare the goodness-of-fit of those models on the test set. Comment the results, choose the 'final' model and explain it.

[1]:
```
library(MASS)
attach(Boston)
boston = as.data.frame(Boston)
boston$chas = as.factor(boston$chas)
```

3.1 Analyze the data to get some first impressions and make some preliminary comments.

3.1.1 Correlation matrix

[2]:
```
library(corrplot)

corr_matrix<-cor(Boston)
options(repr.plot.width=7, repr.plot.height=7)
corrplot(corr_matrix, type="upper", method = "number")
```

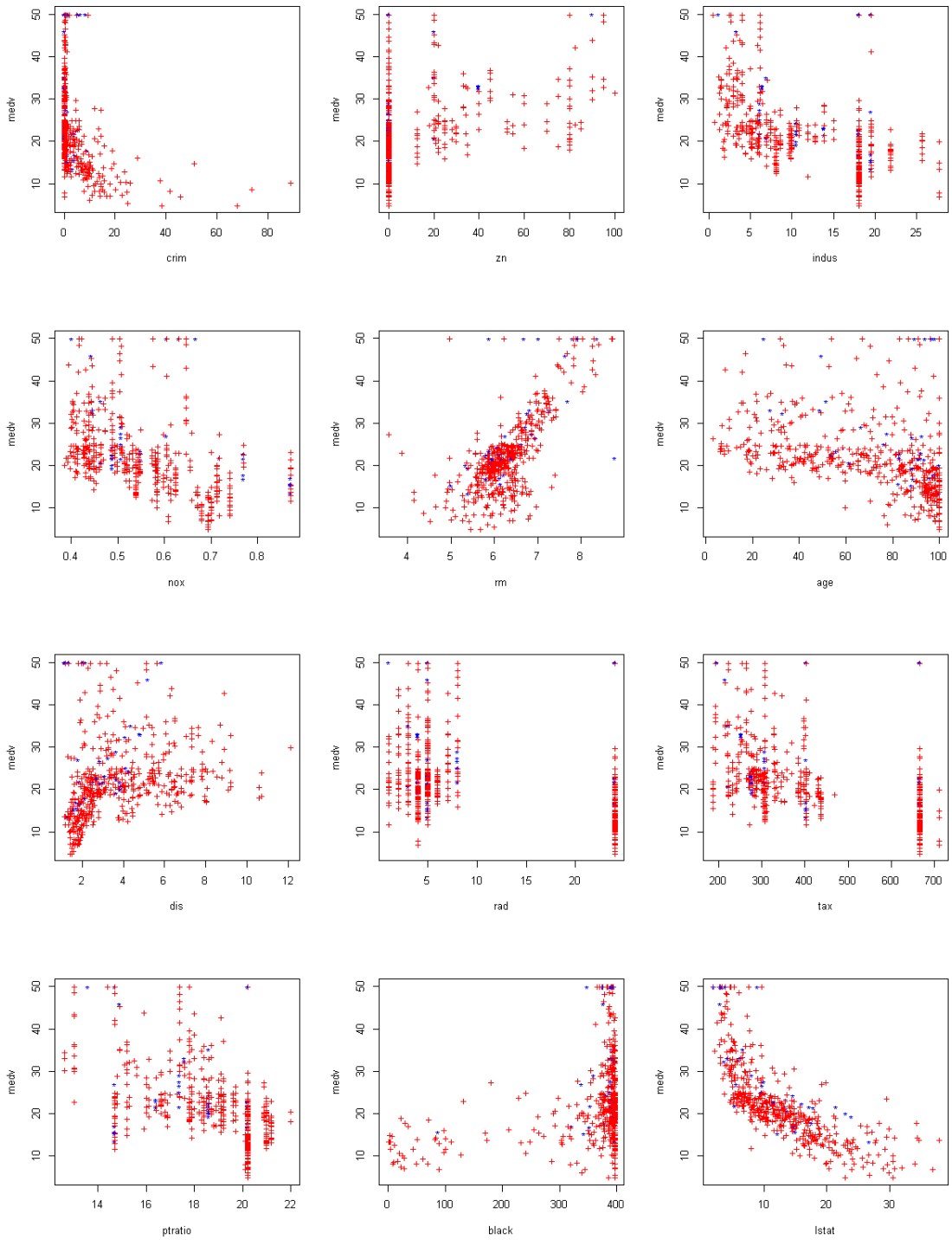


From the correlation matrix:

1. Median value of owner-occupied homes (**mdev**)
    - Increases as average number of rooms per dwelling (**rm**) increases
      - A house with more rooms is usually more expensive
    - Decreases as percent of lower status population (**lstat**) in the area increases
      - Houses are cheaper when there are more people that cannot afford an expensive house (no demand for expensive huses)
2. Full-value property-tax rate (**tax**)
    - Increases as index of accessibility to radial highways (**rad**) increases
      - Access to more highways raises the taxes which make sense because the municipality is investing more money (in higeways) in this area
    - Increases as proportion of non-retail business acres per town (**indus**) increases
      - Maybe more businesses needs more maintenence (cleaning, infrastructure, etc..) by the municipality, so it needs more money
3. Weighted mean of distances to five Boston employment centres (**dis**)
    - Decreases as proportion of non-retail business acres per town (**indus**) increases
      - There are more non-retail business as you get closer to commercial centers
    - Decreases as nitrogen oxides concentration (**nox**) (Air pollution) increases
      - There is more nitrogen oxides (air pollution) as you get closer to commercial centers
    - Decreases as proportion of owner-occupied units built prior to 1940 (**age**) increases
      - Probably new business owners preferred to open their businesses in areas that was already associated as commercial centers (because the old businesses)
4. Nitrogen oxides concentration (**nox**) (Air pollution)
    - Increases as proportion of non-retail business acres per town (**indus**)
      - There is more nitrogen oxides (air pollution) as you get closer to commercial centers
    - Increases as proportion of owner-occupied units built prior to 1940 (**age**)
      - There are more commercial centers around old businesses (which we saw in the previous point) so there are more air pollution

3.1.2 Median value of owner-occupied homes to the explenatory variables plots

[3]:
```
# Define color for each of the 2 categories
colors <- c("Red", "Blue")
colors <- colors[boston$chas]

# Define shapes
shapes = c('+', '*')
shapes <- shapes[boston$chas]

options(repr.plot.width=10, repr.plot.height=3.33*4)
par(mfrow = c(4,3))
plot(medv ~ crim , data = boston, col=colors, pch=shapes)
plot(medv ~ zn, data = boston, col=colors, pch=shapes)
plot(medv ~ indus, data = boston, col=colors, pch=shapes)
# plot(medv ~ chas, data = boston, col=colors, pch=shapes)
plot(medv ~ nox, data = boston, col=colors, pch=shapes)
plot(medv ~ rm, data = boston, col=colors, pch=shapes)
plot(medv ~ age, data = boston, col=colors, pch=shapes)
plot(medv ~ dis, data = boston, col=colors, pch=shapes)
plot(medv ~ rad, data = boston, col=colors, pch=shapes)
plot(medv ~ tax, data = boston, col=colors, pch=shapes)
plot(medv ~ ptratio, data = boston, col=colors, pch=shapes)
plot(medv ~ black, data = boston, col=colors, pch=shapes)
plot(medv ~ lstat, data = boston, col=colors, pch=shapes)
```


medv
crim
zn
indus
nox
rm
age
dis
rad
tax
ptratio
black
lstat

From the plots

1. Seems like there are more crime in areas with lower median value of owner-occupied homes (**medv**)
2. We can see the good relation between **medv** to the explenatory variables **lstat** and **rm** as presented in the correlation matrix section
3. Seems like there are more cheaper houses in areas with a lot of owner-occupied units built prior to 1940 (**age**), probably because there are more old areas among those areas
4. Seems like most of the samples are from areas with low black ratio (black is between 300 and 400 when bk is between 0 and 0.08)
5. The following variables seems to have a polynomial relation to **medv**:
   - crim - 2nd degree
   - indus - 2nd degree
   - lstat - 2nd degree
   - dis - 3rd degree

   It may be worth to try and add their polynomial effect

3.2 Split randomly (why?) the data into a training and test sets of 80% and 20% of the data respectively.

- Put a test set meanwhile aside and consider a training set

We split the data randomly to get an equal data distribution in both, the train and test sets.

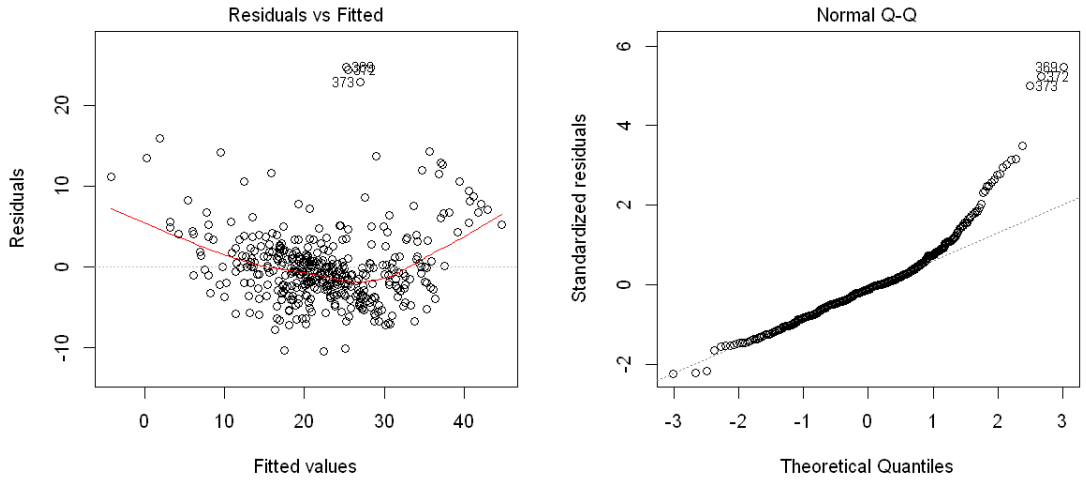
We don't want to train only on certain types of samples and not see at all some other types which will end in overfitting the model to the sample types in the trian set

[4]:
```
set.seed(123)
train.indexes <- sample(nrow(boston), nrow(boston) * 0.80)
boston.train <- boston[train.indexes, ]
boston.test <- boston[-train.indexes, ]
```

3.2.1 Start from the main effects model, verify its adecuacy.

Main effect model

[5]:
```
lm.maineffects.fit <- lm(medv ~ ., data = boston.train)
# summary(lm.maineffects.fit)

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(lm.maineffects.fit, which = 1:2)
```

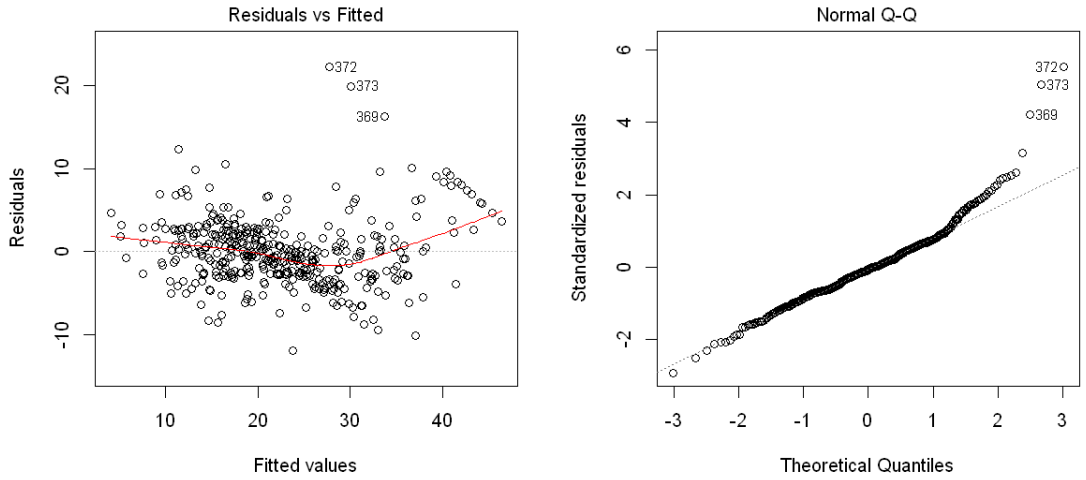


Main effect model - summary

1. All the explenatory variables are significant except **age** and **indus**, but we know that these variables are higly correlated with each other and other variables such as **nox** and **dis** so we'll not remove them yet.
2. R squared is not very good but fair (0.7441)
3. From the residuals vs fitted plot:
   - The residuals assumtions are clearly not met
   - The square trend maybe hints that we forgot an explanatory variable (interaction? squared variable?)
   - We may need a transformation
4. From the Q-Q plot:
   - Seems like the residuals distribution has a "shorter" right tail which means also that our residuals normality assumption doesn't met

Main effect model with polynomial effects

[6]:
```
lm.poly.fit <- lm(
    medv ~ . + I(crim^2) + I(indus^2) + I(lstat^2) + I(dis^2) + I(dis^3),
    data = boston.train)
# summary(lm.poly.fit)

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(lm.poly.fit, which = 1:2)
```



Main effect model with polynomial effects - summary

1. Now **age** is significant, but **indus** and it's squared effect are not significant. All the rest of the variables are significant.
2. R squared is better (0.807 vs 0.7441), but we added more explanatory variables so this is not a good evidence of improvment
3. From the residuals vs fitted plot:
   - The residuals assumtions are yet not met but better
   - The square trend releaxed a bit but we still may missing interactions
   - We still may need a transformation
4. From the Q-Q plot:
   - Got better but still not perfect.

Test if the polynomial effects are dependant on medv Let's test the hypothesis:

$$H_0 : \beta_{lstat^2} = \beta_{crim^2} = \beta_{indus^2} = \beta_{dis^2} = \beta_{dis^3} = 0$$

[7]:
```
anova(lm.maineffects.fit, lm.poly.fit)
```

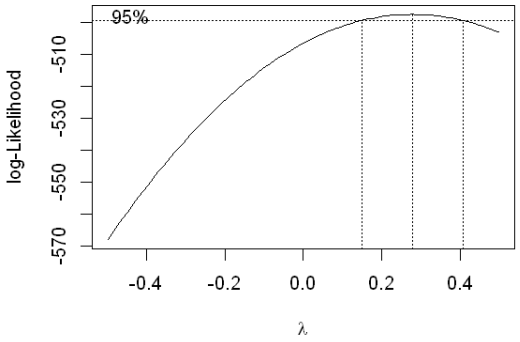
Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
390	8729.934	NA	NA	NA	NA
385	6584.015	5	2145.919	25.0965	6.581013e-22

We will reject H_0 and conclude that the polynomial effects coefficients are not equal to zero

3.3 If you're not satisfied, try to add paired interactions, perform transformations if necessary.

3.3.1 Boxcox transformation for the dependant variable

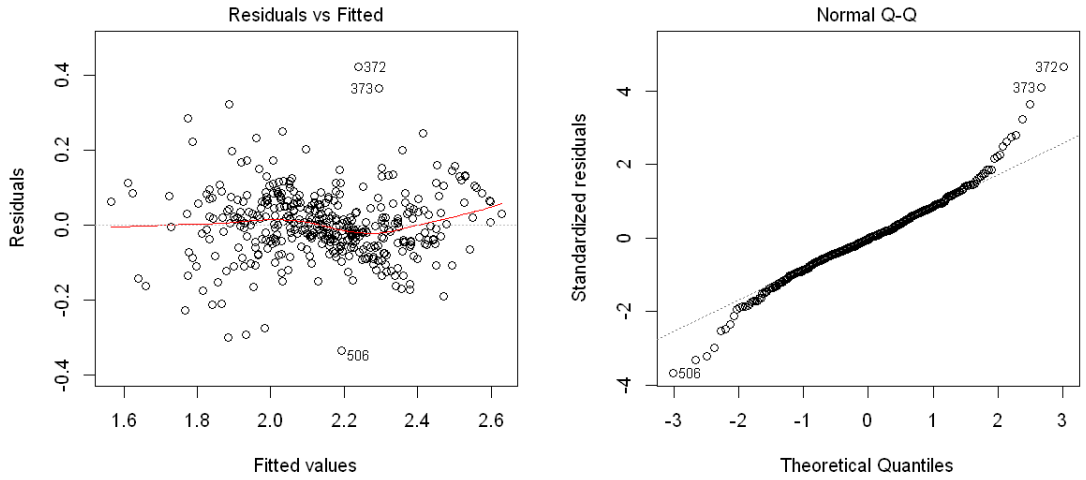
[8]:
```
options(repr.plot.width=10, repr.plot.height=4)
par(mfrow = c(1,2))

boxcox(lm.poly.fit, lambda = seq(-0.5, 0.5, 1/10), plotit = TRUE,
       eps = 1/50, xlab = expression(lambda), ylab = "log-Likelihood")
```



For both models, 4th root transformation may be a good transformation: $\sqrt[4]{medv}$

polynomial model with 4th root transformation

[9]:
```
lm.poly.trans.fit <- lm(
    medv^0.25 ~ . + I(crim^2) + I(indus^2) + I(lstat^2) + I(dis^2) + I(dis^3),
    data = boston.train)
# summary(lm.poly.trans.fit)

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(lm.poly.trans.fit, which = 1:2)
```



polynomial model with 4th root transformation - summary

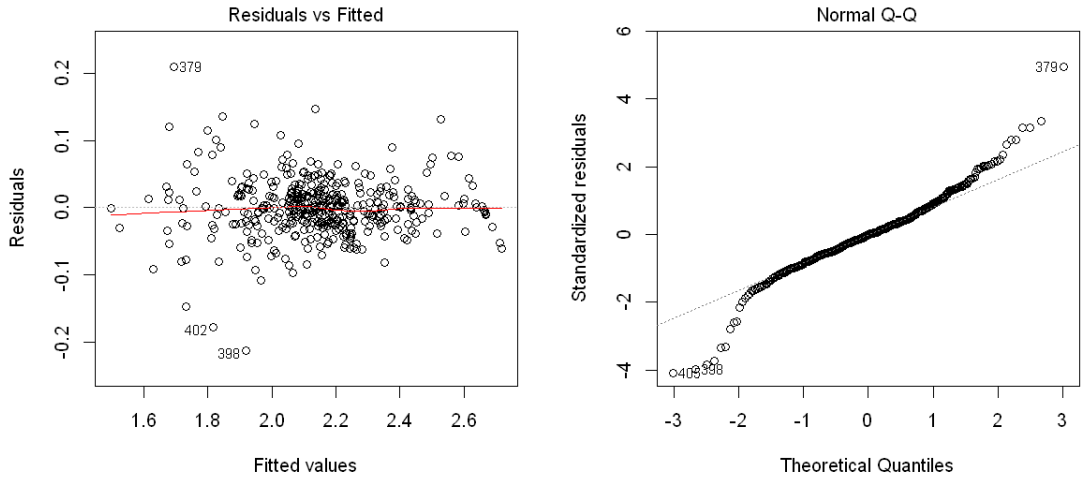
1. **indus** and **dis** polynomial effects are not significat now
2. R squared is a bit better compared to the untransformed polynomial model (0.8255 vs 0.807)
3. Residuals:
    - The residuals zero mean assumption is much better now.
    - We may need to add interactions

3.3.2 Adding paired interactions

- We'll now add all the paired interactions and try to reduce the model later

polynomial model with all the paired interactions

[10]:
```
lm.poly.trans.int.fit <- lm(
    medv^0.25 ~ (. + I(crim^2) + I(indus^2) + I(lstat^2) + I(dis^2) +␣
 ↪I(dis^3))^2,
    data = boston.train)
# summary(lm.poly.trans.int.fit)

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(lm.poly.trans.int.fit, which = 1:2)
```



polynomial model with all the paired interactions - summary

1. We have 172 coefficients - most of them are not significant which is not supprosing since they are higly correlated
2. R squared is fantastic! (0.9612) which is also not supprosing since we added **A LOT** of explanatory variables
3. From the residuals vs fitted plot:
   - Great! Now the zero mean assumption on the residuals seems to met
4. From the Q-Q plot:
   - We are not much normal - more like t distribution

3.4 Perform model selection w.r.t. various model selection criteria.

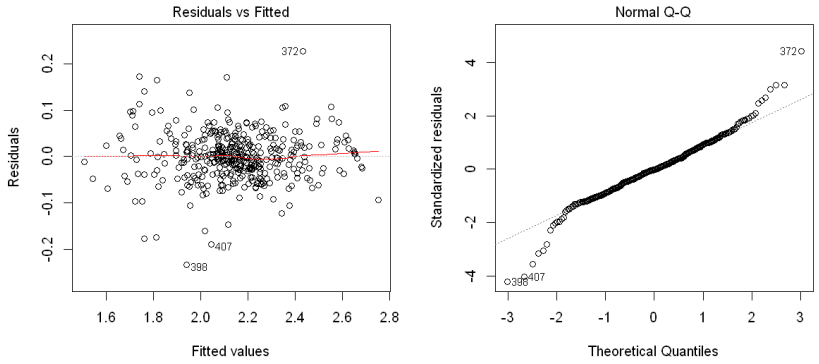
- Compare the resulting models
- Comment the results

3.4.1 AIC

Running on the transformed polynomial model.

Starting from a model with all the paired interactions

[11]:
```
mod.aic <- stepAIC(lm.poly.trans.fit,
                   scope=list(upper= medv ~ .^2, lower= ~1),
                   direction = "both",
                   trace=F,
                   k=2)
# summary(mod.aic)
# length(mod.aic$coefficients)

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(mod.aic, which = 1:2)
```



AIC - summary

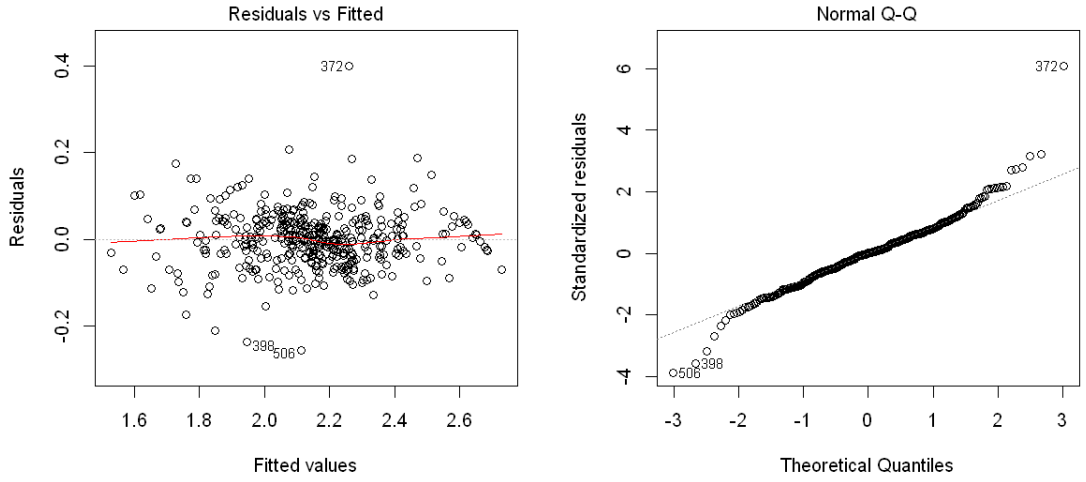
1. AIC has selected **62** variables
2. R squared is fantastic! (0.942)
3. From the residuals vs fitted plot:
   - The zero mean assumption on the residuals still met (except few outliers)
4. From the Q-Q plot:
   - Better, smoother than before but still has tails like t distribution

3.4.2 BIC

Running on the transformed polynomial model.

Starting from a model with all the paired interactions

[12]:
```
n=nrow(boston.train)

mod.bic <- stepAIC(lm.poly.trans.fit,
                   scope=list(upper= medv ~ .^2, lower= ~1),
                   direction = "both",
                   trace=F,
                   k=log(n))
# summary(mod.bic)
# length(mod.bic$coefficients)

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(mod.bic, which = 1:2)
```



BIC - summary

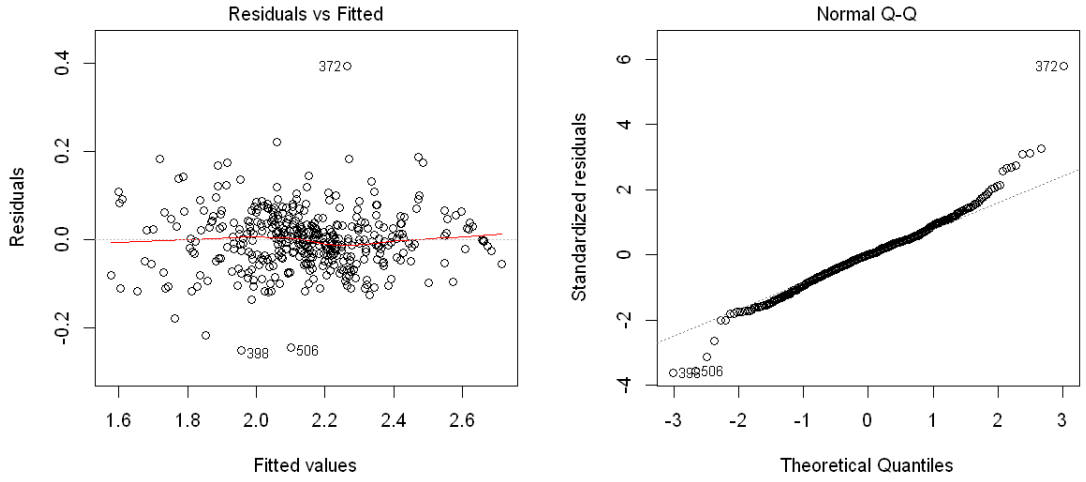
1. BIC has selected **27** variables
2. R squared is fantastic also here (0.9102) - less than AIC, but using less then half of the variables
3. From the residuals vs fitted plot:
   - The zero mean assumption on the residuals still met (except few outliers)
4. From the Q-Q plot:
   - Similar to AIC - still has tails like t distribution

3.4.3 RIC

Running on the transformed polynomial model.

Starting from a model with all the paired interactions

[13]:
```
p=length(lm.poly.trans.int.fit$coefficients)-1

mod.ric <- stepAIC(lm.poly.trans.fit,
                   scope=list(upper= medv ~ .^2, lower= ~1),
                   direction = "both",
                   trace=F,
                   k=2*log(p))

# summary(mod.ric)
# length(mod.ric$coefficients)

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(mod.ric, which = 1:2)
```



RIC - summary

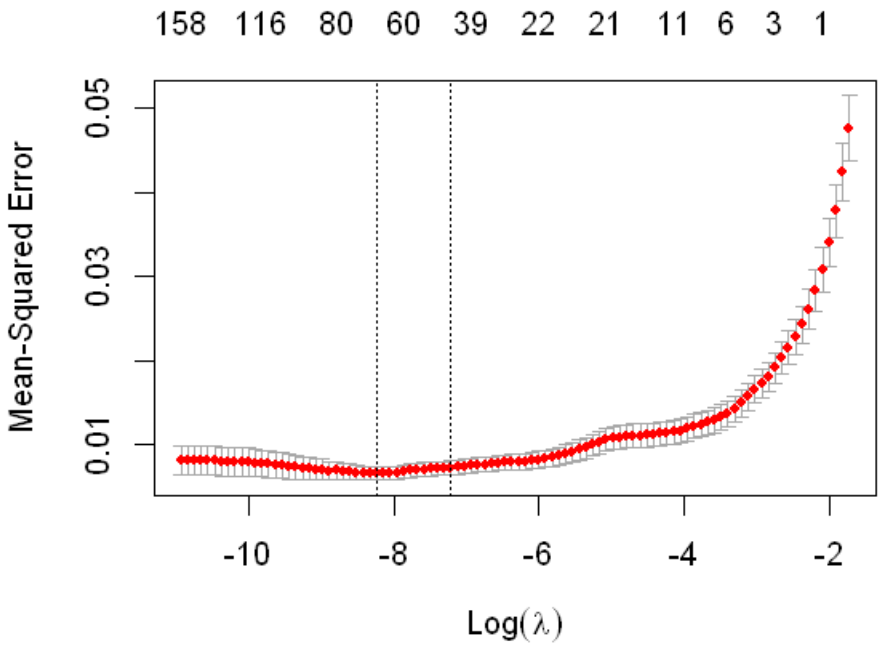
1. RIC has selected **23** variables
2. R squared is fantastic also here (0.9032) - less that AIC & BIC, but using less variables
3. From the residuals vs fitted plot:
   - The zero mean assumption on the residuals still met (except few outliers)
4. From the Q-Q plot:
   - Also similar to AIC & BIC - still has tails like t distribution

3.4.4 Lasso

Running on the transformed polynomial model.

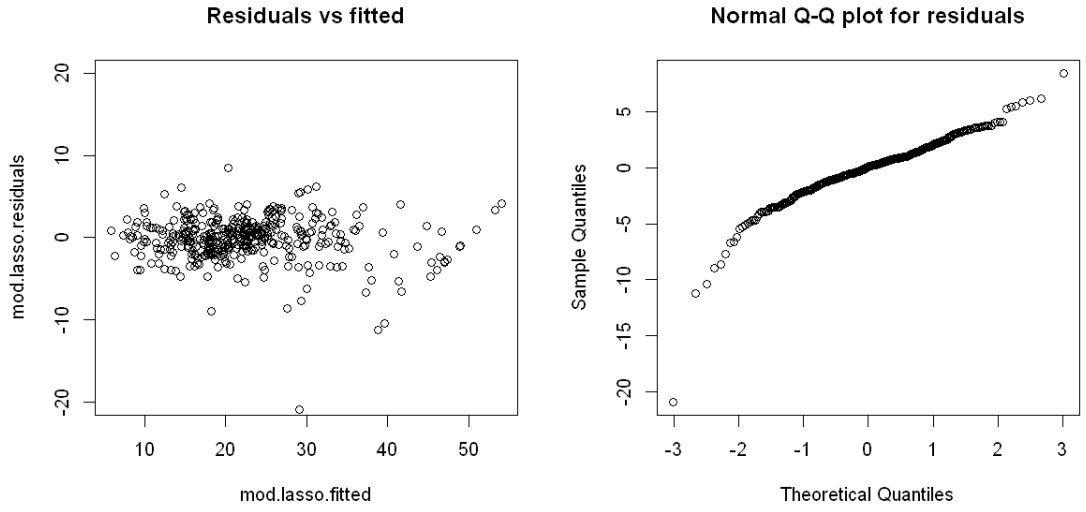
Starting from a model with all the paired interactions

[14]:
```
library(glmnet)

X.lasso <- model.matrix(lm.poly.trans.int.fit)

mod.cv.lasso <- cv.glmnet(X.lasso,
                          boston.train$medv^0.25, #Use a transformed medv
                          alpha=1,
                          nfolds=nrow(X.lasso), #Choose lambda using LOO CV
                          grouped=FALSE)

options(repr.plot.width=5, repr.plot.height=4)
plot(mod.cv.lasso)
```



[15]:
```
mod.lasso <- glmnet(X.lasso, boston.train$medv^0.25,alpha=1,lambda=mod.cv.
 ↪lasso$lambda.min)

mod.lasso.fitted = predict(mod.lasso, X.lasso)^4
mod.lasso.residuals = mod.lasso.fitted - boston.train$medv

# number of coefficients selected by lasso
non.zero.indexes = mod.lasso$beta[, 1] != 0
mod.lasso.length <- length(mod.lasso$beta[non.zero.indexes])

mod.lasso.R2 <- cor(boston.train$medv, as.numeric(mod.lasso.fitted))^2
mod.lasso.R2

options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
plot(mod.lasso.fitted, mod.lasso.residuals, ylim = c(-20,20), main = "Residuals␣
 ↪vs fitted")
qqnorm(mod.lasso.residuals, main = "Normal Q-Q plot for residuals")
```

R^2 : 0.918695505016414



Lasso - summary

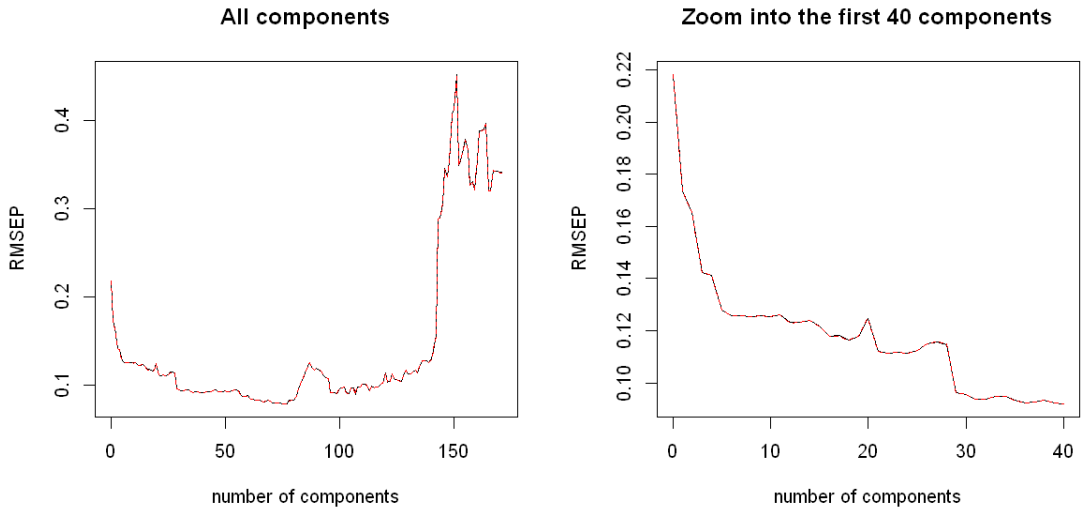
1. Lasso has selected **66** variables
2. R squared is fantastic also here (0.9186)
3. The residuals are not perfect

3.4.5 PCR

Using a polynomial model with all the paired interactions

[16]:
```
library(pls)

mod.pcr = pcr(medv^0.25 ~ (. + I(crim^2) + I(indus^2) + I(lstat^2) + I(dis^2) +
 ↪I(dis^3))^2,
              data=boston.train, scale=T, validation="LOO")

# summary(mod.pcr)
```

[17]:
```
options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
validationplot(mod.pcr, main="All components")
validationplot(mod.pcr, ncomp = 1:40, main="Zoom into the first 40 components")
```



PCR - Summary

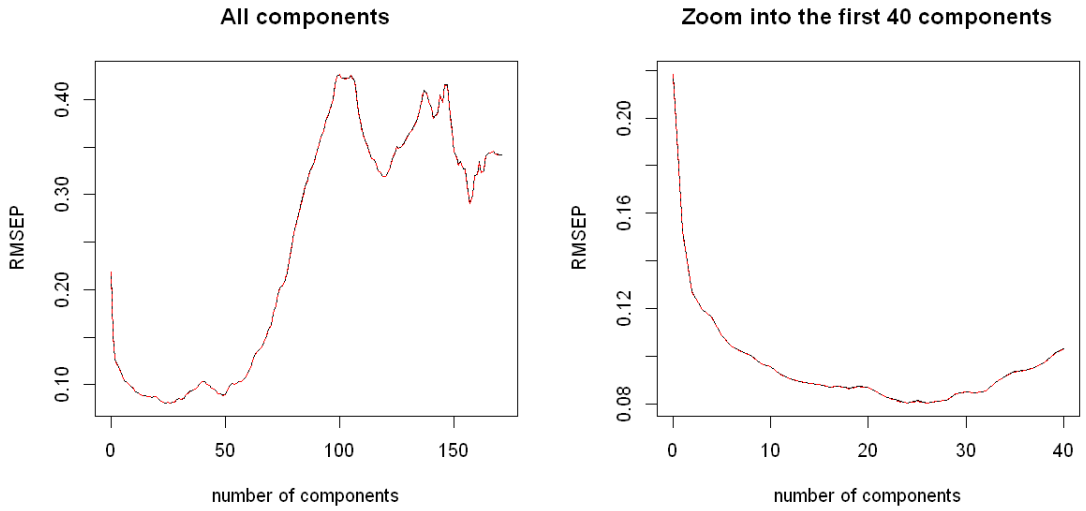
- The best CV value achieved using **76** components (0.07918), but from **29** components (CV=0.09620) the improvment is negligible.
- In addition, 29 components explain 99.05% of Xvariance
- Projecting the data on the first 29 eigenvectors will significantly reduce the data dimensions and will explain most of it's variance
- We know that there is a problem with modeling the data using PCR - it doesn't take the response variable (**medv**) in consideration

3.4.6 PLS

Using a polynomial model with all the paired interactions

[18]:
```
mod.pls = plsr(medv^0.25 ~ (. + I(crim^2) + I(indus^2) + I(lstat^2) + I(dis^2) +␣
 ↪I(dis^3))^2,
               data=boston.train, scale=T, validation="LOO")

# summary(mod.pls)
```

[19]:
```
options(repr.plot.width=10, repr.plot.height=5)
par(mfrow = c(1,2))
validationplot(mod.pls, main="All components")
validationplot(mod.pls, ncomp = 1:40, main="Zoom into the first 40 components")
```



PLS - Summary

- The best CV value achieved using **24** components (0.08052).
- In addition, 24 components explain 97.15% of Xvariance
- Projecting the data on the first 24 eigenvectors will significantly reduce the data dimensions and will explain most of it's variance

3.5 Test and compare the goodness-of-fit of those models on the test set.

- Comment the results
- choose the 'final' model and explain it.

3.5.1 Models comparison

We'll compare the models using:

1. **CV** - Leave one out cross validation
2. **GCV** - Generalized cross validation
3. **R2CV** - Squared correlation between the response variable to the corresponding LOO predicted response vector
4. **MSPE** - Mean square prediction error **using the test data**
5. **Size** - Number of the variables used by the model

NOTE → The comparision will be made on the original scale

Functions for calculating the above

[20]:
```
none <- function(x){ return (x)}
power.4 = function(x){return (x^4)}

mspe <- function(lm.object, un.trans = none, testdata = boston.test, ...){
    # Computes the mean square prediction error
    pred.test <- un.trans(predict(lm.object, testdata, ...))
    mspe <- as.numeric(mean((pred.test - boston.test$medv) ^ 2))

    return (round(cbind(mspe), 4))
}
```

[21]:
```
lm.crossVal <- function(lm.object, un.trans = none)
{
    # Computes the cross-validation (CV), the generalized cross-validation (GCV)␣
 ↪and the cross-validation correlation coefficient (R^2_{CV}) for the specified␣
 ↪linear model
    res.1 <- lm.object$residuals
    y <- un.trans(lm.object$fitted.values + lm.object$residuals)
    res <- y - un.trans(lm.object$fitted.values)
    h <- lm.influence(lm.object)$hat
    n <- length(y)
    cv <- mean(res^2/(1 - h)^2)
    gcv <- (n * sum(res^2))/lm.object$df^2
    r2cv <- cor(y, y - res/(1 - h))^2
    return(round(cbind(cv, gcv, r2cv), 4))
}
```

[22]:
```
pls.crossVal <- function(pls.object, p, un.trans = none)
{
    # Computes the cross-validation (CV), the generalized cross-validation (GCV)␣
 ↪and the cross-validation correlation coefficient (R^2_{CV}) for the specified␣
 ↪PLS model
    y <- un.trans(pls.object$fitted.values[,,][,p] + pls.
 ↪object$residuals[,,][,p])
    res <- y - un.trans(pls.object$fitted.values[,,][,p])

    n <- length(y)

    # CV
    press = sum((un.trans(pls.object$validation$pred[,,][,p]) - y)^2)
    cv = press/n

    # GCV
    rss = sum(res^2)
    gcv = (n/(n-p)^2)*rss

    # R2CV
    r2cv = cor(y, un.trans(pls.object$validation$pred[,,][,p]))^2

    return(round(cbind(cv, gcv, r2cv), 4))
}
```

[23]:
```
mod.size <- function(mod){
    return (length(mod$coefficients))
}
```

[24]:
```
lm.maineffect.crossVal <- lm.crossVal(lm.maineffects.fit)
lm.poly.crossVal <- lm.crossVal(lm.poly.fit)
lm.poly.trans.crossVal <- lm.crossVal(lm.poly.trans.fit, un.trans = power.4)
lm.poly.trans.int.crossVal <- lm.crossVal(lm.poly.trans.int.fit, un.trans =␣
 ↪power.4)
mod.aic.crossVal <- lm.crossVal(mod.aic, un.trans = power.4)
mod.bic.crossVal <- lm.crossVal(mod.bic, un.trans = power.4)
mod.ric.crossVal <- lm.crossVal(mod.ric, un.trans = power.4)
mod.pcr.crossVal <- pls.crossVal(mod.pcr, 29, un.trans = power.4)
mod.pls.crossVal <- pls.crossVal(mod.pls, 24, un.trans = power.4)
```

[25]:
```
lm.maineffect.mspe <- mspe(lm.maineffects.fit)
lm.poly.mspe <- mspe(lm.poly.fit)
lm.poly.trans.mspe <- mspe(lm.poly.trans.fit, un.trans = power.4)
lm.poly.trans.int.mspe <- mspe(lm.poly.trans.int.fit, un.trans = power.4)
mod.aic.mspe <- mspe(mod.aic, un.trans = power.4)
mod.bic.mspe <- mspe(mod.bic, un.trans = power.4)
mod.ric.mspe <- mspe(mod.ric, un.trans = power.4)
mod.pcr.mspe <- mspe(mod.pcr, un.trans = power.4, ncomp=29)
mod.pls.mspe <- mspe(mod.pls, un.trans = power.4, ncomp=24)
```

[26]:
```
#calculate lasso CV
# lambda.cv.index = which(mod.cv.lasso$lambda == mod.cv.lasso$lambda.min)
# mod.lasso.cv = mod.cv.lasso$cvm[lambda.cv.index][[1]]

#calculate lasso GCV
mod.lasso.rss = sum((predict(mod.lasso, X.lasso)^4 - boston.train$medv)^2)
n = nrow(X.lasso)
p = mod.lasso.length
mod.lasso.gcv = round((n/(n-p)^2)*mod.lasso.rss, 4)

#calculate lasso R2CV
mod.lasso.cv.pred <- lapply(1:nrow(X.lasso), function(i){
    fit <- glmnet(as.matrix(X.lasso[-i,]),
                  boston.train$medv[-i]^0.25,
                  alpha = 1,
                  lambda = mod.cv.lasso$lambda.min)
    pred <- predict(fit, t(X.lasso[i,]))[1]^4

    return(pred)
})
mod.lasso.r2cv = round(cor(boston.train$medv, as.numeric(mod.lasso.cv.pred))^2,␣
 ↪4)

#calculate lasso CV
mod.lasso.cv = round(sum(as.numeric(mod.lasso.cv.pred) - boston.train$medv)^2/n,␣
 ↪4)

#calculate lasso MSPE
newx <- model.matrix(as.formula(medv^0.25 ~ (. + I(crim^2) + I(indus^2) +␣
 ↪I(lstat^2) + I(dis^2) + I(dis^3))^2), boston.test)
mod.lasso.mspe <- mspe(mod.lasso, testdata = newx, un.trans = power.4)
```

[27]:
```
options(scipen=999)
results <- rbind(
    cbind(lm.maineffect.crossVal, lm.maineffect.mspe, mod.size(lm.maineffects.
 ↪fit)),
    cbind(lm.poly.crossVal, lm.poly.mspe, mod.size(lm.poly.fit)),
    cbind(lm.poly.trans.crossVal, lm.poly.trans.mspe, mod.size(lm.poly.trans.
 ↪fit)),
    cbind(lm.poly.trans.int.crossVal, lm.poly.trans.int.mspe, mod.size(lm.poly.
 ↪trans.int.fit)),
    cbind(mod.aic.crossVal, mod.aic.mspe, mod.size(mod.aic)),
    cbind(mod.bic.crossVal, mod.bic.mspe, mod.size(mod.bic)),
    cbind(mod.ric.crossVal, mod.ric.mspe, mod.size(mod.ric)),
    cbind(mod.lasso.cv, mod.lasso.gcv, mod.lasso.r2cv, mod.lasso.mspe, mod.lasso.
 ↪length),
    cbind(mod.pcr.crossVal, mod.pcr.mspe, 29),
    cbind(mod.pls.crossVal, mod.pls.mspe, 24)

)
```

[28]:
```
models = c(
    "Main effect",
    "polynomial effects",
    "polynomial 4th root transformed",
    "polynomial and interactions transformed",
    "AIC",
    "BIC",
    "RIC",
    "Lasso",
    "PCR",
    "PLS"
)

comparision <- as.data.frame(cbind(models, results))
names(comparision) <- c("model", "CV", "GCV", "R2cv", "MSPE", "Size")

comparision[order(comparision$MSPE),]
```

Results sorted by MSPE:

	model	CV	GCV	R2cv	MSPE	Size
5	AIC	7.254	6.2406	0.9143	11.3871	62
8	Lasso	6.553	9.9088	0.8715	11.8316	66
7	RIC	9.7129	8.969	0.8852	11.8507	23
6	BIC	9.4054	8.719	0.8888	12.7963	27
10	PLS	11.342	7.7941	0.8663	12.9068	24
3	polynomial 4th root transformed	17.1092	16.687	0.7985	18.8912	19
9	PCR	16.4079	15.2476	0.8065	19.4128	29
2	polynomial effects	18.5488	17.9453	0.7805	19.5318	19
4	polynomial and interactions transformed	81.6144	7.9461	0.5026	21.811	172
1	Main effect	23.8818	23.188	0.7174	23.6786	14

Selected model The selected model is the RIC model:

- The RIC model has a very good MSPE value (negligable difference from the minimal MSPE) **while the number of variables used in this model is fairly small (only 23)**
- In "goodness of fit" terms:
  - It is not the best but very satisfying
  - Also the residuals assumptions in this model seems to by met (except few outliers)