Module 3 - Multiple regression model

Pawel Chilinski

November 5, 2013

Exercise 1.

Load data trees.

```
> # Color scatterplot matrix, colored and ordered by magnitude of r
> library(gclus)
> trees.r <- abs(cor(trees))
> cpairs(trees, order.single(trees.r), panel.colors = dmat.color(trees.r), gap = .5,
+        main = "Variables Ordered and Colored by Correlation")
```

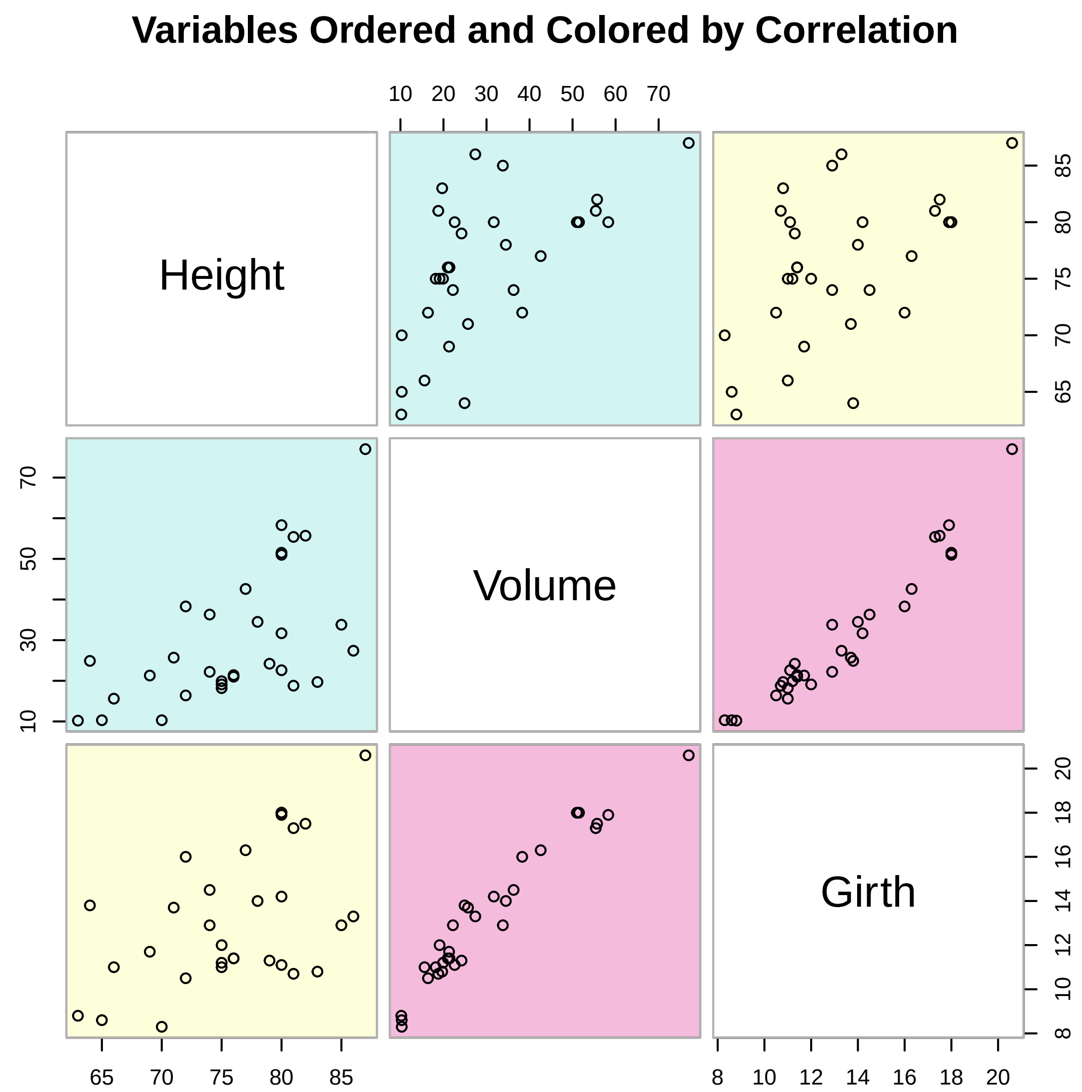

Figure 1: Data trees scatter plots orders by correlation.

- Fit least squares lines to both pairs of variables taking Volume as a response variable (use function lm()). Check how you can extract components of the object returned by function lm() such as coefficients, fitted values and residuals (use function names() on the returned object to see a full list of components).

```
> vol_girth_lm <- lm(Volume ~ Girth,trees)
> names(vol_girth_lm)
```

```
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"
```

```
> head(vol_girth_lm$coefficients)
```

```
(Intercept)       Girth
 -36.943459    5.065856
```

```
> #check if the formula gives the same results
> X=matrix(c(rep(1,nrow(trees)),trees$Girth),nrow=nrow(trees))
> Y=matrix(trees$Volume,nrow=nrow(trees))
> solve((t(X)%*%X))%*%t(X)%*%Y
```

```
           [,1]
[1,] -36.943459
[2,]   5.065856
```

```
> head(vol_girth_lm$fitted.values)
```

```
        1         2         3         4         5         6
 5.103149  6.622906  7.636077 16.248033 17.261205 17.767790
```

```
> head(vol_girth_lm$residuals)
```

```
        1         2         3         4         5         6
5.1968508 3.6770939 2.5639226 0.1519667 1.5387954 1.9322098
```

```
> vol_height_lm <- lm(Volume ~ Height,trees)
> vol_height_lm$coefficients
```

```
(Intercept)      Height
  -87.12361     1.54335
```

```
> head(vol_height_lm$fitted.values)
```

```
       1        2        3        4        5        6
20.91087 13.19412 10.10742 23.99757 37.88772 40.97442
```

```
> head(vol_height_lm$residuals)
```

```
           1            2            3            4            5            6
-10.61086922  -2.89412045   0.09257906  -7.59756873 -19.08771651 -21.27441602
```

- View the fitted model using function summary(). Check how you can extract components of the object returned by function summary()

```
> vol_girth_lm_sum <- summary(vol_girth_lm)
> vol_girth_lm_sum
```

```
Call:
lm(formula = Volume ~ Girth, data = trees)

Residuals:
   Min     1Q Median     3Q    Max
-8.065 -3.107  0.152  3.495  9.587

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -36.9435     3.3651  -10.98 7.62e-12 ***
Girth         5.0659     0.2474   20.48  < 2e-16 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 4.252 on 29 degrees of freedom
Multiple R-squared:  0.9353,	Adjusted R-squared:  0.9331
F-statistic: 419.4 on 1 and 29 DF,  p-value: < 2.2e-16
```

```
> names(vol_girth_lm_sum)
```

```
 [1] "call"          "terms"         "residuals"     "coefficients"
 [5] "aliased"       "sigma"         "df"            "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
> vol_girth_lm_sum$r.squared
```

```
[1] 0.9353199
```

```
> vol_height_lm_sum <- summary(vol_height_lm)
> vol_height_lm_sum
```

```
Call:
lm(formula = Volume ~ Height, data = trees)

Residuals:
    Min      1Q  Median      3Q     Max
-21.274  -9.894  -2.894  12.068  29.852

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -87.1236    29.2731  -2.976 0.005835 **
Height        1.5433     0.3839   4.021 0.000378 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 13.4 on 29 degrees of freedom
Multiple R-squared:  0.3579,	Adjusted R-squared:  0.3358
F-statistic: 16.16 on 1 and 29 DF,  p-value: 0.0003784
```

We see that b0 and b1 have statistically significant values (where model with girth variable has more significant coefficients).

- Draw fitted lines on respective scatter plots of data (use functions plot() and abline()).

```
> par(mfrow=c(1,2))
> plot(trees$Girth,trees$Volume, xlab="Girth",ylab="Volume",main="Volume ~ Girth")
> abline(vol_girth_lm,col="blue")
> legend(x="topleft",col=c("black","blue"),pch=c(1,NA),legend=c("data","fitted line"),lty=c(0,1))
> plot(trees$Height,trees$Volume, xlab="Height",ylab="Volume",main="Volume ~ Height")
> abline(vol_height_lm,col="blue")
> legend(x="topleft",col=c("black","blue"),pch=c(1,NA),legend=c("data","fitted line"),lty=c(0,1))
> par(mfrow=c(1,1))
```

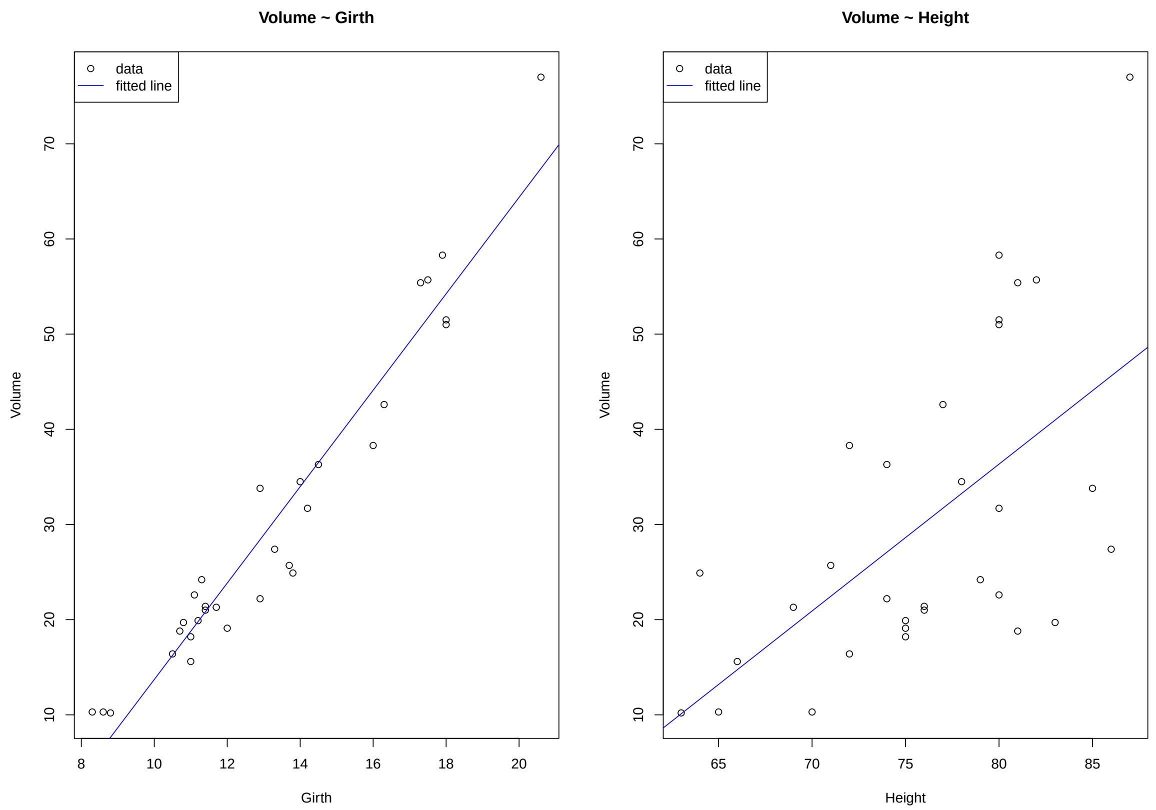

Figure 2: Data and fitted lines for Volume(Girth) and Volume(Height).

- Compare values of R2 between the two models.

  ```
  > vol_girth_lm_sum$r.squared

  [1] 0.9353199

  > vol_height_lm_sum$r.squared

  [1] 0.3579026
  ```

  So model containing Girth as explanatory variable explains more variance of Volume than model containing Height.

- Assuming that linear regression model is the true model for the data, can we say that there is a statistically significant relationship between variables Volume and Girth (use 5% significance level)? Answer the same question in the case of variables Volume and Height.

  ```
  > cor.test(trees$Volume,trees$Girth, conf.level = 0.95)

          Pearson's product-moment correlation

  data:  trees$Volume and trees$Girth
  t = 20.4783, df = 29, p-value < 2.2e-16
  alternative hypothesis: true correlation is not equal to 0
  95 percent confidence interval:
   0.9322519 0.9841887
  sample estimates:
        cor
  0.9671194

  > cor.test(trees$Volume,trees$Height, conf.level = 0.95)
  ```

```
        Pearson's product-moment correlation

data:  trees$Volume and trees$Height
t = 4.0205, df = 29, p-value = 0.0003784
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3095235 0.7859756
sample estimates:
      cor
0.5982497
```

So with 5% significance level we reject hypothesis about variables not being related (p<0.05 in both cases).

- Give the confidence interval for slope coefficient in the regression model Volume ∼ Girth?

  95% confidence interval:

```
> confint(vol_girth_lm)[2,]
```

```
   2.5 %   97.5 %
4.559914 5.571799
```

- What is the estimated variance of tree volume in this model?

```
> anova_vol_girth_lm <- anova(vol_girth_lm)
> anova_vol_girth_lm
```

```
Analysis of Variance Table

Response: Volume
          Df Sum Sq Mean Sq F value    Pr(>F)
Girth      1 7581.8  7581.8  419.36 < 2.2e-16 ***
Residuals 29  524.3    18.1
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> (anova_vol_girth_lm[1,2] + anova_vol_girth_lm[2,2])/(nrow(trees)-1)
```

```
[1] 270.2028
```

```
> #which is the same as
> var(trees$Volume)
```

```
[1] 270.2028
```

- What is the predicted value of volume in this model if the girth of a tree is equal to 15 inches?

```
> #using built-in function
> predict(vol_girth_lm,data.frame(Girth=c(15)))
```

```
       1
39.04439
```

```
> #or using formula directly
> vol_girth_lm$coefficients[1]+vol_girth_lm$coefficients[2]*15
```

```
(Intercept)
   39.04439
```

Exercise 2.

File anscombe quartet.txt contains four pairs of variables.

```
> anscombe_quartet <- read.table(file="anscombe_quartet.txt",header=T)
> names(anscombe_quartet)
```

```
[1] "Y1" "X1" "Y2" "X2" "Y3" "X3" "Y4" "X4"
```

- Fit least squares lines to all four pairs of variables.

```
> anscombe_quartet_models <- lapply(1:4,function(i){lm(as.formula(paste("Y",i,"~","X",i,sep="")),
+                                                        anscombe_quartet)})
```

- Compare fitted values of coefficients b0, b1, and values of R2 and correlations for the four models.

  Coefficients:

```
> lapply(1:4,function(i){data.frame(b0=anscombe_quartet_models[[i]]$coefficients[1],
+                                   b1=anscombe_quartet_models[[i]]$coefficients[2],row.names=NULL)})
```

```
[[1]]
        b0        b1
1 3.000091 0.5000909

[[2]]
        b0  b1
1 3.000909 0.5

[[3]]
        b0        b1
1 3.002455 0.4997273

[[4]]
        b0        b1
1 3.001727 0.4999091
```

  R2 of the models:

```
> anscombe_quartet_models_summaries <- lapply(anscombe_quartet_models,summary)
> sapply(anscombe_quartet_models_summaries,function(model_summary){model_summary$r.squared})
```

```
[1] 0.6665425 0.6662420 0.6663240 0.6667073
```

  Correlations (computed as b1*(sx/sy)):

```
> sapply(1:4,function(i){
+   anscombe_quartet_models_summaries[[i]]$coefficients[2,1]*
+     (sd(anscombe_quartet[[paste("X",i,sep="")]])/sd(anscombe_quartet[[paste("Y",i,sep="")]]))
+ })
```

```
[1] 0.8164205 0.8162365 0.8162867 0.8165214
```

- In one screen make four scatter plots of the respective data (use command par(mfrow=c(2,2))). In which case fitting a linear model is reasonable? Are numerical summaries sufficient for assessing a regression model?

```
> par(mfrow=c(2,2))
> for(i in 1:4){
+   x_var <- paste("X",i,sep="")
+   y_var <- paste("Y",i,sep="")
+   plot(anscombe_quartet[[x_var]],anscombe_quartet[[y_var]], xlab=x_var,ylab=y_var,
+        main=paste(y_var,"~",x_var))
+   abline(anscombe_quartet_models[[i]],col="blue")
+   legend(x="topleft",col=c("black","blue"),pch=c(1,NA),legend=c("data","fitted line"),lty=c(0,1))
+ }
> par(mfrow=c(1,1))
```

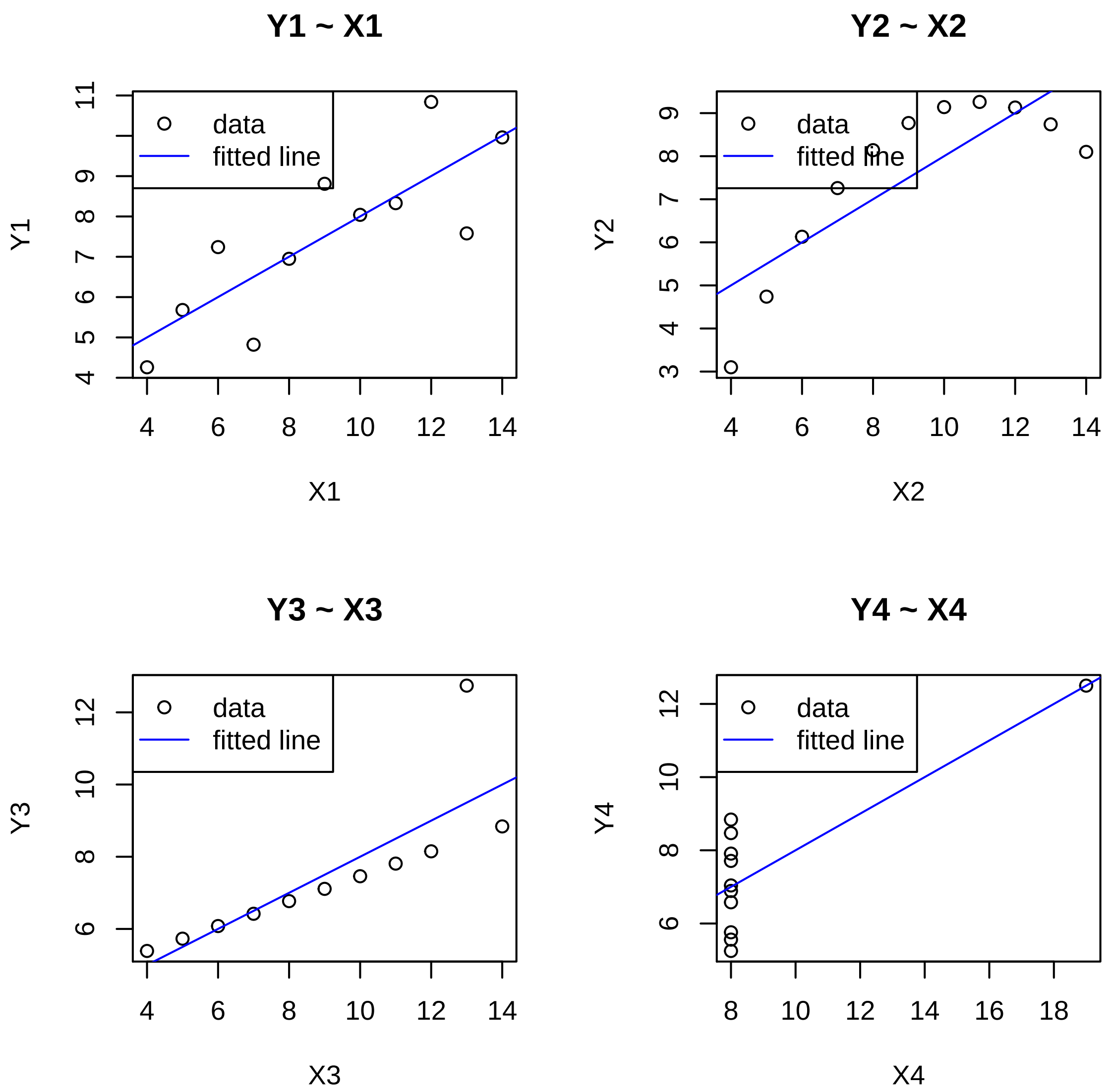

Figure 3: Data and fitted models for anscombe quartet

To assess if the model is appropriate for the data we can plot the residuals and check if they spread evenly around 0:

```
> par(mfrow=c(2,2))
> for(i in 1:4){
+   model <- anscombe_quartet_models[[i]]
+   plot(model$residuals, main=paste("residuals for model",i))
+   abline(a=0,b=0,col="blue")
+ }
> par(mfrow=c(1,1))
```

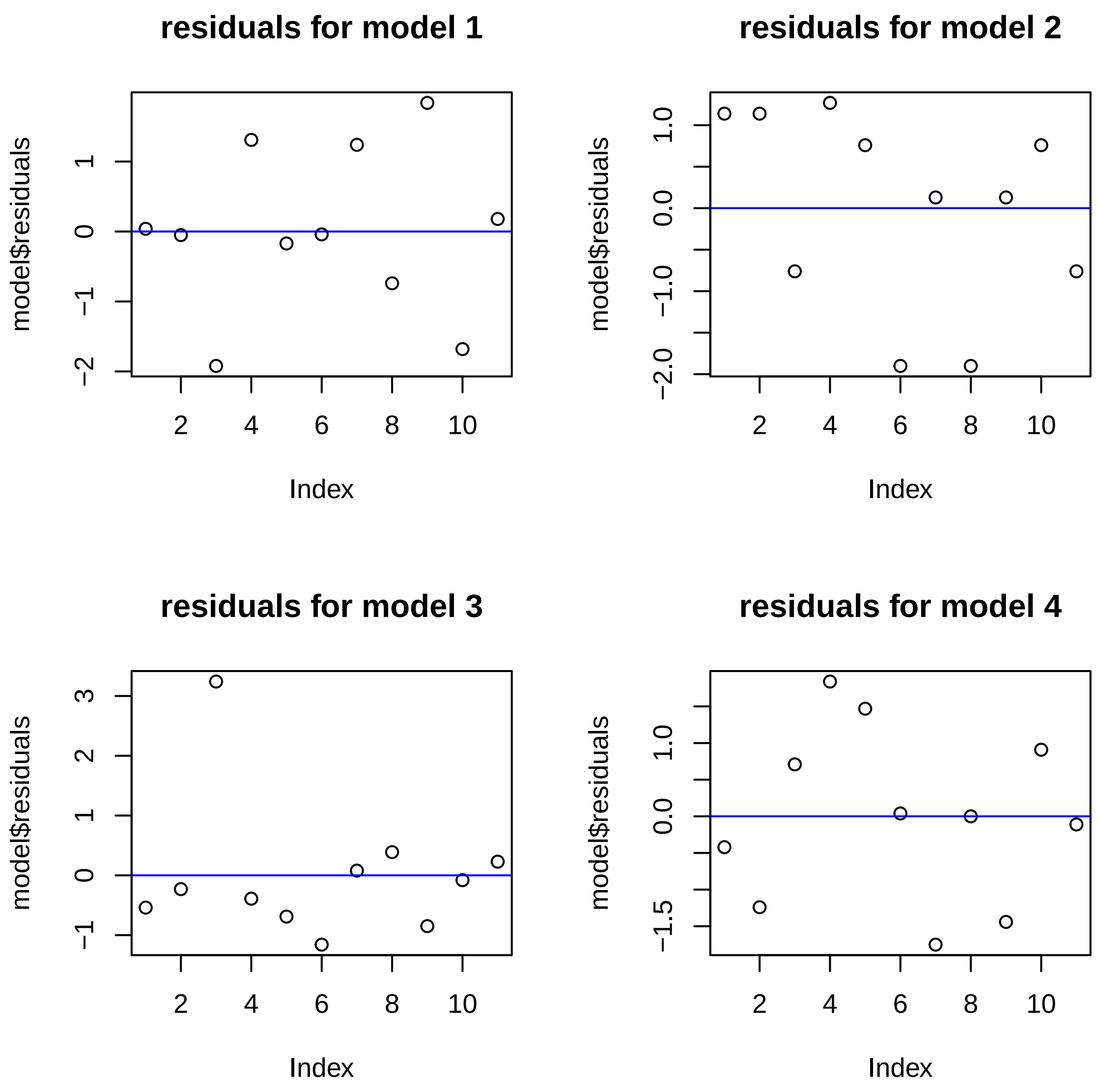

Figure 4: Residuals

QQ plots for residuals to assess their normality:

```
> par(mfrow=c(2,2))
> for(i in 1:4){
+   model <- anscombe_quartet_models[[i]]
+   qqnorm(model$residuals,main=paste("residuals for model",i))
+ }
> par(mfrow=c(1,1))
```

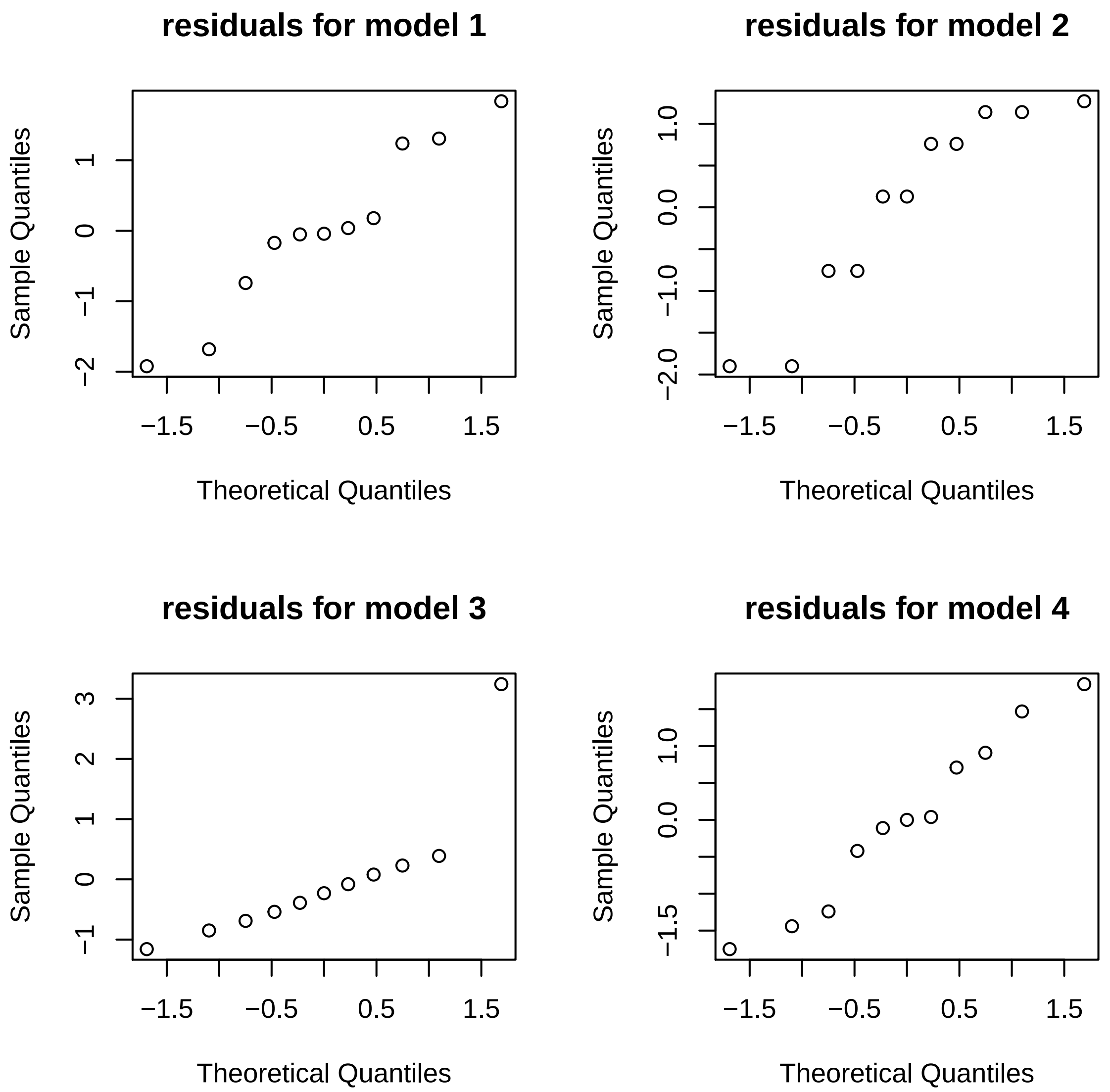

Figure 5: QQ plots of residuals

So looking at scatter plots and distribution of residuals we can conclude that only first data set should be explained by linear model because its residuals seem with no relationship whatsoever with explanatory variable (this cannot be said about residuals from other models). The second data set is not linear. The third data set contains one outlier which added to linear data completely changes the fitted model. The forth data set with one outlier which all but one data points show no relationship between variables. Looking at numerical summaries we can see that all are almost the same, are not enough to assess the models and even are deceptive when making conclusions about models (so we have to look at scatter plots and other visualization tools to help us make correct decision):

```
> anscombe_quartet_models_summaries

[[1]]

Call:
lm(formula = as.formula(paste("Y", i, "~", "X", i, sep = "")),
    data = anscombe_quartet)

Residuals:
     Min       1Q   Median       3Q      Max
-1.92127 -0.45577 -0.04136  0.70941  1.83882
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0001     1.1247   2.667  0.02573 *
X1            0.5001     0.1179   4.241  0.00217 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 1.237 on 9 degrees of freedom
Multiple R-squared:  0.6665,	Adjusted R-squared:  0.6295
F-statistic: 17.99 on 1 and 9 DF,  p-value: 0.00217


[[2]]

Call:
lm(formula = as.formula(paste("Y", i, "~", "X", i, sep = "")),
    data = anscombe_quartet)

Residuals:
    Min      1Q  Median      3Q     Max
-1.9009 -0.7609  0.1291  0.9491  1.2691

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.001      1.125   2.667  0.02576 *
X2             0.500      0.118   4.239  0.00218 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 1.237 on 9 degrees of freedom
Multiple R-squared:  0.6662,	Adjusted R-squared:  0.6292
F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002179


[[3]]

Call:
lm(formula = as.formula(paste("Y", i, "~", "X", i, sep = "")),
    data = anscombe_quartet)

Residuals:
    Min      1Q  Median      3Q     Max
-1.1586 -0.6146 -0.2303  0.1540  3.2411

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0025     1.1245   2.670  0.02562 *
X3            0.4997     0.1179   4.239  0.00218 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 1.236 on 9 degrees of freedom
Multiple R-squared:  0.6663,	Adjusted R-squared:  0.6292
F-statistic: 17.97 on 1 and 9 DF,  p-value: 0.002176


[[4]]

Call:
lm(formula = as.formula(paste("Y", i, "~", "X", i, sep = "")),
    data = anscombe_quartet)

Residuals:
   Min     1Q Median     3Q    Max
-1.751 -0.831  0.000  0.809  1.839
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.0017     1.1239   2.671  0.02559 *
X4            0.4999     0.1178   4.243  0.00216 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 1.236 on 9 degrees of freedom
Multiple R-squared:  0.6667,	Adjusted R-squared:  0.6297
F-statistic:    18 on 1 and 9 DF,  p-value: 0.002165
```

Exercise 3.

File realest.txt contains data related to houses in Chicago such as: Price (price of house), Bedroom (number of bedrooms), Space (area in squared feet), Room (number of rooms), Lot (width of front lot), Tax (property tax per year), Bathroom (number of bathrooms), Garage (number of parking lots in garage), Condition (0 indicates good condition, 1 - bad condition). Fit a linear regression model taking price of house as a response variable and the rest of variables in the data set as explanatory variables.

```
> realest <- read.table(file="realest.txt",header=T)
> realest_model <- lm(Price ~ Bedroom + Space + Room + Lot + Tax + Bathroom + Garage + Condition, realest)
```

- How will the price of house change if number of bedrooms is increased by 1 and values of the rest of variables stay unchanged? Explain apparent incorrect result. Compare this result with the analogous result in a single regression model Price âĹij Bedroom.

```
> realest_model

Call:
lm(formula = Price ~ Bedroom + Space + Room + Lot + Tax + Bathroom +
    Garage + Condition, data = realest)

Coefficients:
(Intercept)      Bedroom        Space         Room          Lot          Tax
  13.712572    -7.756208     0.011626     5.097706     0.228063     0.003374
   Bathroom       Garage    Condition
   5.718372     3.613603    -2.162027
```

So we could conclude that increase in number of bedroom by 1 decreases the price by 7.75 which seems to be false. But we cannot interpret this coefficient in this way because now we have interaction with other variables. Additionally when we have linear dependence between explanatory variables then there are infinitely many fitting models (when $X'X$ is close to not invertible matrix). So the coefficients cannot be used to reason about how changes in explanatory variables affects explained variable. Example of the linear dependence between explanatory variables:

```
> plot(realest$Bedroom,realest$Room,xlab="Bedroom",ylab="Room",main="Room vs Bedroom")
> text(5,11,labels=paste("correlation = ",round(cor(realest$Bedroom,realest$Room),digits=2)),col="blue")
```

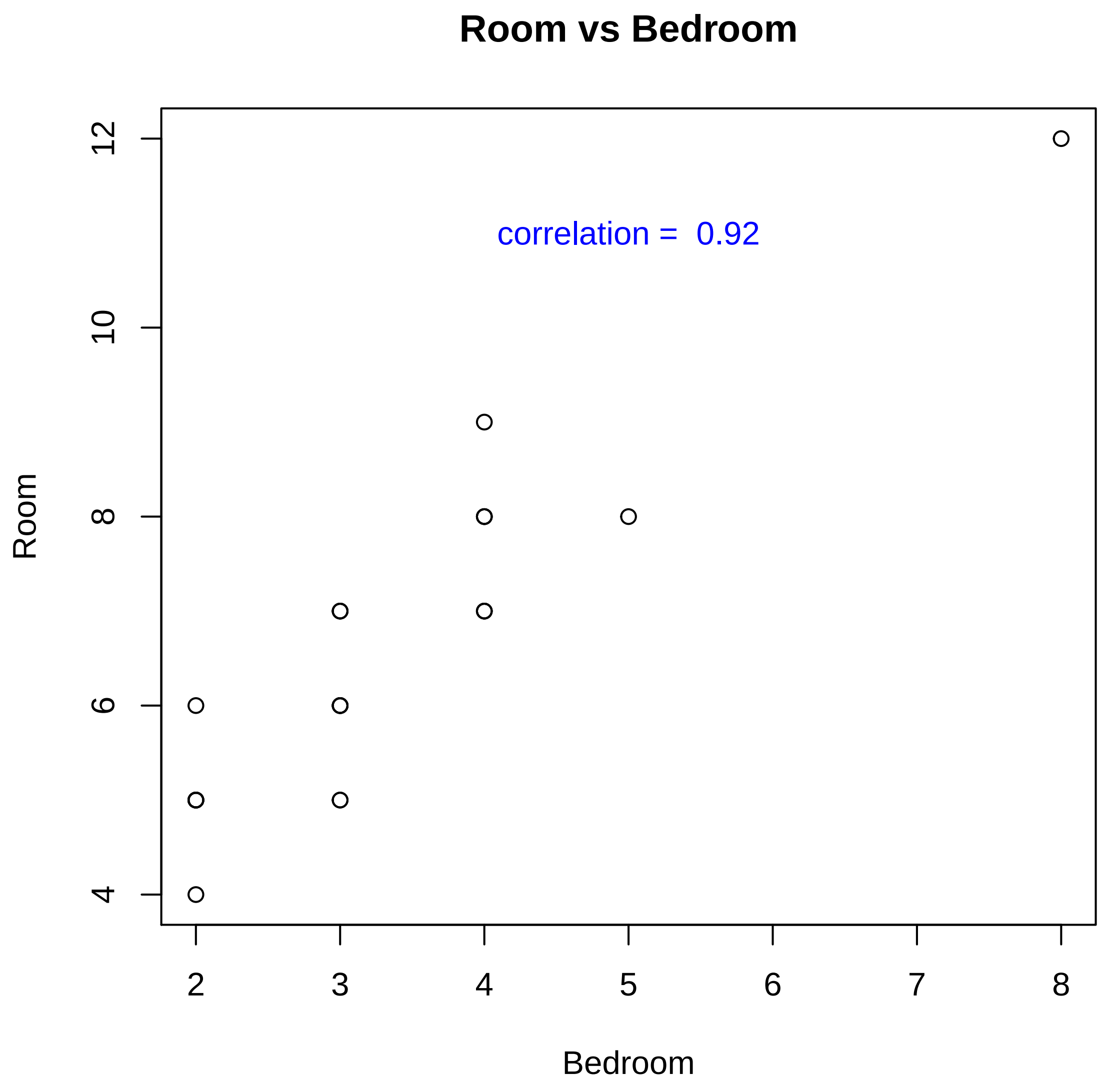

Figure 6: Room vs Bedroom linear dependence

But looking at the simpler model Price ∼ Bedroom we see more plausible prediction i.e. increasing number of bedrooms by 1 increases price by 3.921.

```
> realest_price_bedroom_model<-lm(Price ~ Bedroom, realest)
> realest_price_bedroom_model

Call:
lm(formula = Price ~ Bedroom, data = realest)

Coefficients:
(Intercept)      Bedroom
     43.487        3.921
```

- What price would you predict for a house in a good condition, with 3 bedrooms, 8 rooms, 2 bathrooms, 1 parking lot, 1500 square feet of area, 40 feet of lot width and 1000 dollars of tax amount (use function predict())? Find confidence interval for the predicted:
  - mean value of response (parameter interval="confidence" in function predict()),
  - value of response (parameter interval="prediction" in function predict()).

  Predicted price:

```
> realest_model<-lm(Price ~ Bedroom + Room + Bathroom + Garage + Space + Lot + Tax, realest)
> predict(realest_model,data.frame(Bedroom=3,Room=8,Bathroom=2,Garage=1,Space=1500,Lot=40,Tax=1000))
```

```
       1
74.88175
```

Mean value of response and its 0.95 confidence interval:

```
> predict(realest_model,data.frame(Bedroom=3,Room=8,Bathroom=2,Garage=1,Space=1500,Lot=40,Tax=1000),
+         interval="confidence")

       fit      lwr      upr
1 74.88175 65.98707 83.77642
```

Value of predicted response and its 0.95 confidence interval:

```
> predict(realest_model,data.frame(Bedroom=3,Room=8,Bathroom=2,Garage=1,Space=1500,Lot=40,Tax=1000),
+         interval="prediction")

       fit      lwr      upr
1 74.88175 57.35735 92.40614
```

Exercise 4.

File cheese.txt contains data describing taste of cheese (variable cheese) and other parameters such as:

Acetic - logarithm of acetic acid content,
Lactic - lactic acid content,
H2S - logarithm of hydrogen sulphide content.

Consider two linear models:

taste versus Acetic
taste versus Acetic, Lactic, H2S.

Perform an F test for testing hypothesis that smaller model is better fitted to the data than the larger one (take significance level as 0.05).

Read data and create models:

```
> cheese <- read.table("cheese.txt", header=T)
> simple_model <- lm(taste ~ Acetic, cheese)
> complex_model <- lm(taste ~ Acetic + Lactic + H2S, cheese)
```

We see that in more complex model the Acetic and Intercept coefficients are not statistically significant:

```
> simple_model_sum <- summary(simple_model)
> simple_model_sum

Call:
lm(formula = taste ~ Acetic, data = cheese)

Residuals:
    Min      1Q  Median      3Q     Max
-29.642  -7.443   2.082   6.597  26.581

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -61.499     24.846  -2.475  0.01964 *
Acetic        15.648      4.496   3.481  0.00166 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 13.82 on 28 degrees of freedom
Multiple R-squared:  0.302,       Adjusted R-squared:  0.2771
F-statistic: 12.11 on 1 and 28 DF,  p-value: 0.001658

> complex_model_sum <- summary(complex_model)
> complex_model_sum

Call:
lm(formula = taste ~ Acetic + Lactic + H2S, data = cheese)

Residuals:
```

```
    Min      1Q  Median      3Q     Max
-17.390  -6.612  -1.009   4.908  25.449

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -28.8768    19.7354  -1.463  0.15540
Acetic        0.3277     4.4598   0.073  0.94198
Lactic       19.6705     8.6291   2.280  0.03108 *
H2S           3.9118     1.2484   3.133  0.00425 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

Residual standard error: 10.13 on 26 degrees of freedom
Multiple R-squared:  0.6518,	Adjusted R-squared:  0.6116
F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

Compare the models:

```
> #R^2 is signifacntly different
> anova(simple_model,complex_model)
```

```
Analysis of Variance Table

Model 1: taste ~ Acetic
Model 2: taste ~ Acetic + Lactic + H2S
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     28 5348.7
2     26 2668.4  2    2680.3 13.058 0.0001186 ***
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

```
> #R^2 of complex model is bigger
> simple_model_sum$r.squared
```

```
[1] 0.3019934
```

```
> complex_model_sum$r.squared
```

```
[1] 0.6517747
```

So the p-value of the test with H_0 that R^2 of the complex model is the same as R^2 of the simple model is very small i.e. much less than 0.05. So we can reject this hypothesis and assume that complex model explains the data better.