STATMED Lecture 7: Linear regression II

Matti Pirinen

20.8.2024

In this lecture, we will study some more details about the linear regression model. In particular, how to make predictions from the model using R, and how to assess whether the model seems approriate for the data set at hand. Finally, we will study the interpretation of the coefficients in multiple regression setting where many predictors are simultaneously included in the model.

Let's start by reading in the same height-weight data set as previously.

```
measures = read.table("Davis_height_weight.txt", as.is = TRUE, header = TRUE)
head(measures, n = 3) #show first three rows
```

```
##   X sex weight height repwt repht
## 1 1   M     77    182    77   180
## 2 2   F     58    161    51   159
## 3 3   F     53    161    54   158
```

The columns `repwt`and `repht` are self-reported weight and self-reported height, respectively.

Let's fit separate linear regression models `lm.m` and `lm.f` in males and females, respectively.

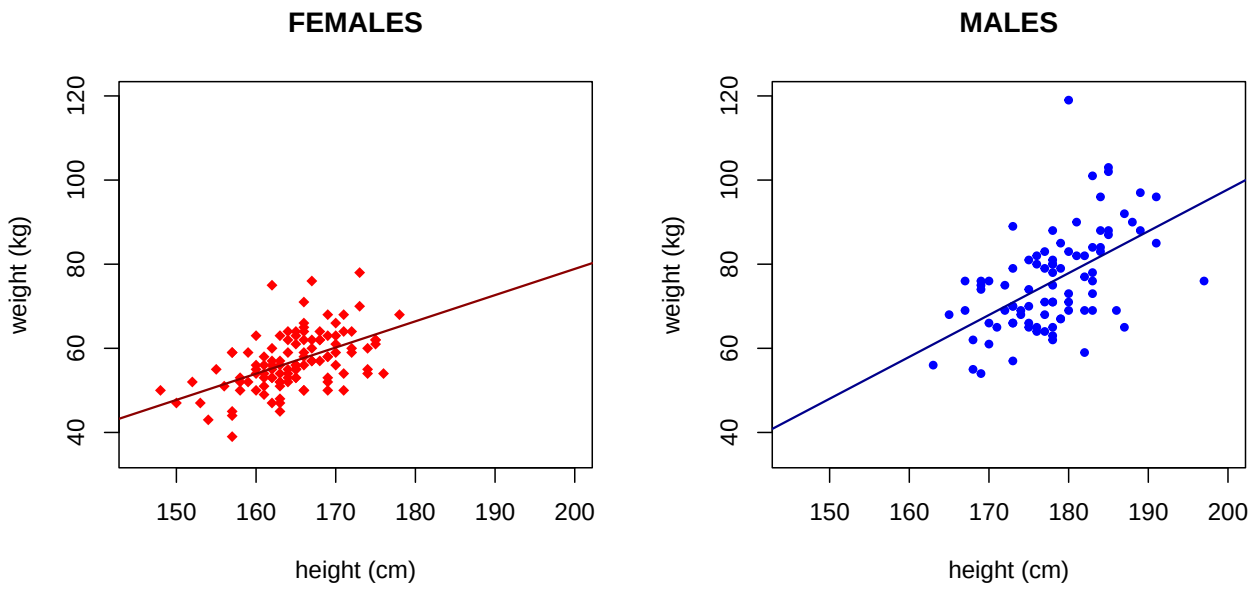
```
ii.m = (measures$sex == "M") #logical vector with 'TRUE' for Males and 'FALSE' for Females
lm.m = lm( weight ~ height, data = measures[ii.m,]) #fit model in males
lm.f = lm( weight ~ height, data = measures[!ii.m,]) #fit model in females

#use same range on the axes in both sexes to get a clear picture how they compare
x.range = c(145, 200) #height in cm
y.range = c(35, 120) #weight in kg
x.name = "height (cm)"
y.name = "weight (kg)"

par(mfrow = c(1,2)) #split plotting area to 1 x 2 grid

plot(measures$height[!ii.m], measures$weight[!ii.m], pch = 18, main = "FEMALES", col = "red",
     ylab = y.name, xlab = x.name, xlim = x.range, ylim = y.range)
abline(lm.f, col = "darkred", lwd = 1.5) #add the regression line

plot(measures$height[ii.m], measures$weight[ii.m], pch = 20, main = "MALES", col = "blue",
     ylab = y.name, xlab = x.name, xlim = x.range, ylim = y.range)
abline(lm.m, col = "darkblue", lwd = 1.5) #add the regression line
```



Predictions from the regression model

In the previous Lecture, we looked up the coefficients of the linear model and used them to compute the predicted value of `weight` for an individual with a given `height`. For example, to compute the predicted `weight` for a 180 cm tall male, we did

```
sum(lm.m$coeff*c(1,180)) #this computes a*1 + b*180 for a and b taken from male regression 'lm.m'
```

```
## [1] 77.87761
```

Very generally, in R, the fitted models can be used for prediction via the `predict(model, newdata)` function. Let's predict `weight` for two new individuals who are 160 cm and 180 cm tall. We need to give the input data as parameter `newdata =` formatted as a data.frame that contains (at least) the same column names that were used in the model fitted by the `lm()` function. In the current case, we need to define only the column `height`. We will call the data that goes in to the prediction as `data.in`.

```
data.in = data.frame(height = c(160, 180))
data.in # Check that this looks correct.
```

```
##   height
## 1    160
## 2    180
```

```
predict(lm.m, newdata = data.in) # predictions for males of 160cm and 180cm
```

```
##        1        2
## 57.96565 77.87761
```

We can present these results in a `data.frame` together with the input data:

```r
data.frame(data.in, predicted.weight = predict(lm.m, newdata = data.in))
```

```
##   height predicted.weight
## 1    160         57.96565
## 2    180         77.87761
```

The uncertainty of prediction can be asked at the level of individual value `interval = "prediction"` or as a confidence interval for the population mean at the given parameter values `interval = "confidence"`. Note that the option names in R commands can be abbreviated as long as the abbreviation remains unique among all possible options; hence here we can use `pred` and `conf`, for example.

```r
predict(lm.m, newdata = data.in, interval = "pred") #for males of 160cm and 180cm
```

```
##        fit      lwr      upr
## 1 57.96565 36.95574 78.97555
## 2 77.87761 57.73251 98.02271
```

The interpretation of the 2nd prediction interval here is that, in 95% of the data sets, when we do this prediction procedure for a new male who is 180 cm tall, the new male will fall within the prediction interval, which here is from 58 kg to 98 kg.

```r
predict(lm.m, newdata = data.in, interval = "conf") #for males of 160cm and 180cm
```

```
##        fit      lwr      upr
## 1 57.96565 51.59499 64.33631
## 2 77.87761 75.64286 80.11236
```

The interpretation of the confidence interval for the mean here is that, in 95% of the data sets, when we do this prediction procedure for male population of height of 180 cm, the true population mean will be within the confidence interval, which here is from 75.6 kg to 81.1 kg.

WARNING: A common error happens when `newdata` parameter is not correctly specified because then `predict()` will not complain but simply outputs the fitted values of the original `lm()` model fit. For example, if we misspell `newdata` as `new.data`, R will not complain and the first two predicted values are actually for the first 2 males in the original data, NOT for 160cm and 180cm as the user might think!

```r
predict(lm.m, new.data = data.in)[1:2] #GOES WRONG because newdata has not been specified!
```

```
##        1        4
## 79.86881 74.89082
```

```r
predict(lm.m, newdata = data.in)[1:2] #This is correct
```

```
##        1        2
## 57.96565 77.87761
```

Example 7.1

1. What is the predicted weight of a male who is 170 cm tall? What is the predicted population average weight of all males who are 170 cm tall? Give appropriate 95% intervals in both cases.

```r
#For completeness, let's re-fit the model first in males only
lm.m = lm( weight ~ height, data = measures[measures$sex == "M",])
#predicted weight for a male 170 cm
predict(lm.m, newdata = data.frame(height = c(170)), interval = "pred")
```

```
##        fit      lwr      upr
## 1 67.92163 47.61119 88.23207
```

```r
#predicted population average at 170 cm
predict(lm.m, newdata = data.frame(height = c(170)), interval = "conf")
```

```
##        fit      lwr     upr
## 1 67.92163 64.50355 71.3397
```

We estimate that the population average weight for a male who is 170 cm tall is 67.9 kg (95%CI: 64.5,. . . ,71.3). We predict that 95% CI for weight values for 170 cm males is (47.6,. . . ,88.2) (and the mean weight is the same 67.9 kg as the estimate for the population mean).

2. Fit a linear model regressing `height` on `weight` in females. What is the predicted height of a woman who weighs 60kg?

```r
lm.h.on.w = lm( height ~ weight, data = measures[measures$sex == "F",])
#predicted weight for a female of 60kg
predict(lm.h.on.w, newdata = data.frame(weight = c(60)), interval = "pred")
```

```
##        fit      lwr      upr
## 1 166.0199 156.2594 175.7805
```

Predicted height is 166 cm with 95%CI of (156,. . . ,176).

Assessing model fit

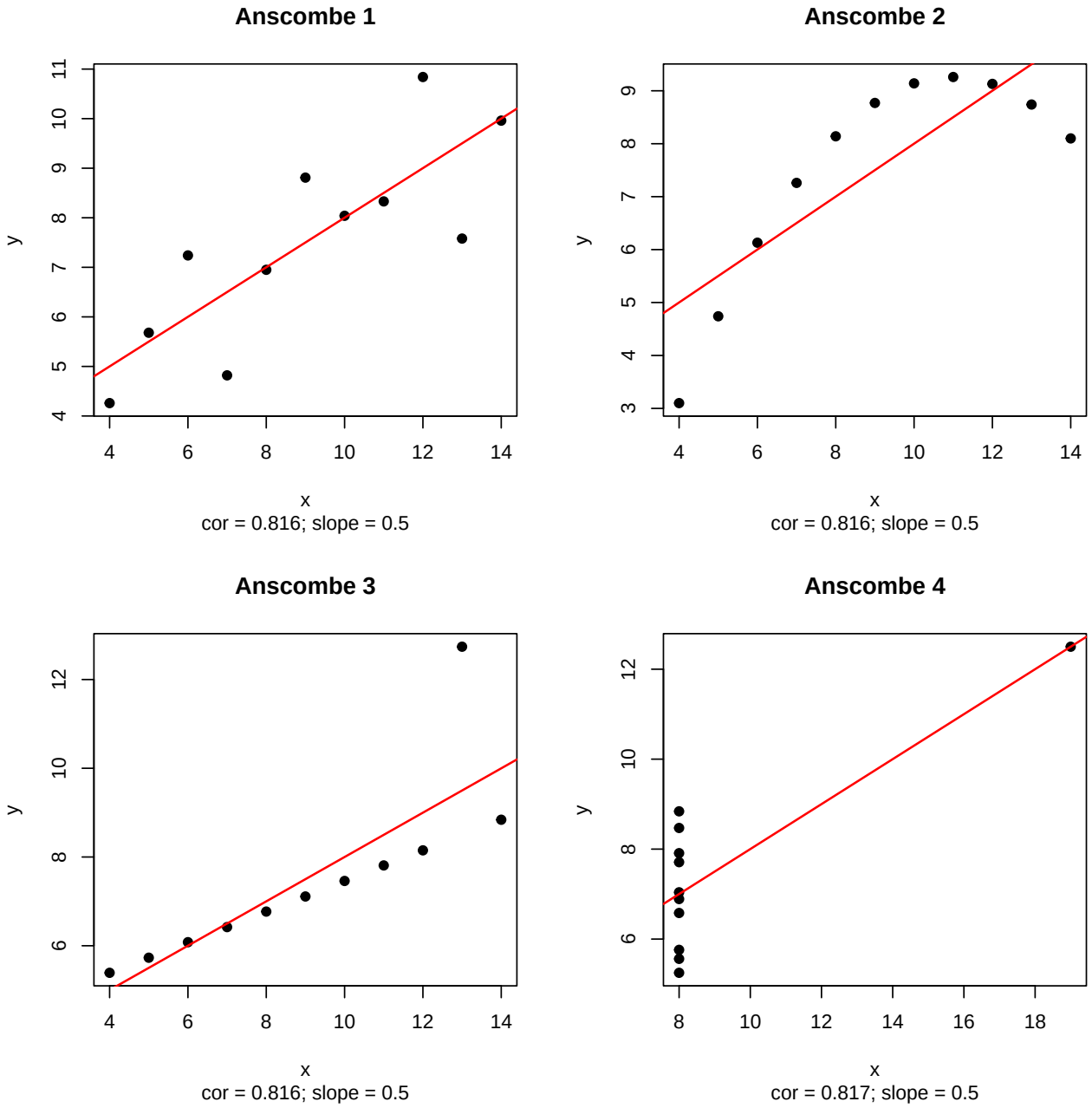
Francis Anscombe generated four data sets with 11 observations in each that all had similar summary statistics (means, variances, correlations) but look very different when plotted. His point was to emphasize the importance to plot the data. In R, these data sets are in the columns of the data.frame called `anscombe`. Let's plot the data sets, fit linear model and compute correlations and slopes that we put in a subtitle for each plot.

```r
anscombe[1:2,] # show first two rows
```

```
##   x1 x2 x3 x4   y1   y2   y3   y4
## 1 10 10 10  8 8.04 9.14 7.46 6.58
## 2  8  8  8  8 6.95 8.14 6.77 5.76
```

```r
par(mfrow = c(2,2)) #2 x 2 plotting area for 4 plots
for(ii in 1:4){
  x = anscombe[,ii] #x-coordinates for Anscombe's data set ii
  y = anscombe[,4 + ii] #y-coordinates for Anscombe's data set ii
  lm.fit = lm(y ~ x)
```

```
  plot(x, y, main = paste("Anscombe",ii), pch = 19, xlab = "x", ylab = "y",
       sub = paste0("cor = ",signif(cor(x,y), 3),
                    "; slope = ",signif(coefficients(lm.fit)[2], 3)))
  abline(lm.fit, col = "red", lwd = 1.5)
}
```



We see that the data sets are identical with respect to the x-y correlation and the linear model slope, but we also see that the data sets are very different in other respects, and, in particular, the linear model does not seem an appropriate description of the X-Y relationship for the data sets 2, 3 and 4.

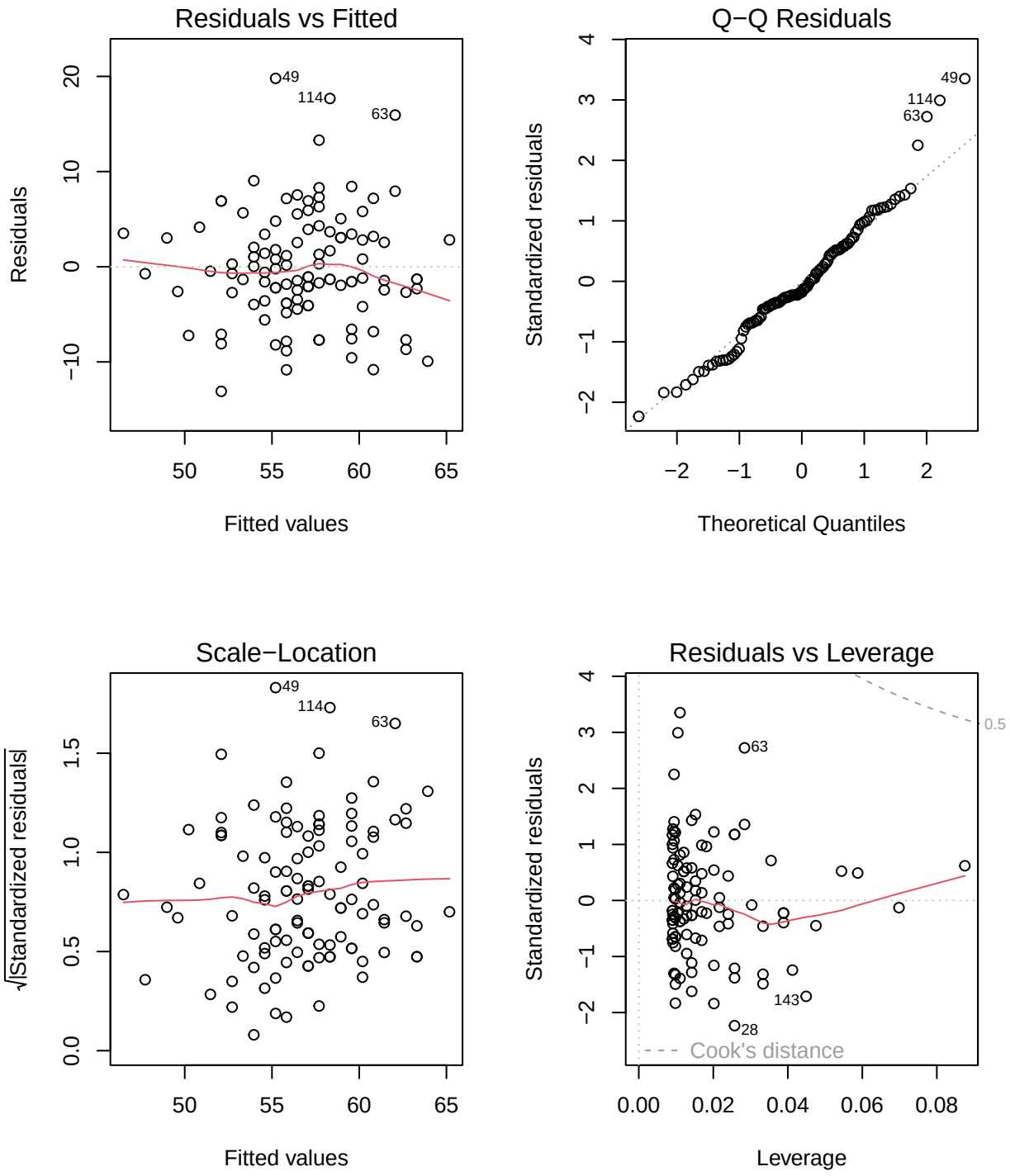
How can we know whether a linear model is an approriate model for our data set? First, we should consider this conceptually: Is there a reason to think that a linear relationship between the variables is appropriate for the research question we have in mind? If yes, then we should proceed to assess how the linear model

is fitting the data and whether the modeling assumptions seem valid? The main assumptions for the linear model are:

1. The relationship between Y and X is linear, that is, there are values a and b for which $Y = a + b \cdot X + \varepsilon$ and the distribution of the **error term** ε does not depend on the value of X.
2. In particular, errors ε have constant variance across all values of X.
3. Errors ε for different observations are independent of each other.
4. Errors ε have a Normal distribution in order that SE and P-value from `lm()` are valid.

There are 4 standard plots that we can use to assess these assumptions and they are generated simply by calling `plot()` on the regression model object. Let's first check how our model of `weight ~ height` in females behaves in these diagnostic plots:

```
par(mfrow = c(2,2)) #plot diagnostics to 2 x 2  area
plot(lm.f) #calling plot on an lm-object generates 4 diagnostic plots
```



To interpret these plot, read the detailed explanation from here: https://data.library.virginia.edu/diagnostic-plots/.

A short summary of the four plots:

1. Residuals ($y_i - \widehat{y}_i$) vs. fitted ($\widehat{y}_i$) should not show a pattern where the distribution of residuals varies depending on the fitted values. Thus, in a good plot, the red line showing the mean of the data points is close to a horizontal line.

2. QQ-plot should ideally be on the diagonal line, in which case the residuals ($y_i - \widehat{y}_i$) are Normally

distributed and SEs and P-values of the model coefficients are reliable. However, small deviations from the line are not a problem.

3. Standardized residuals vs. fitted values should show a similar magnitude along y-axis for each region on the x-axis. Otherwise, the variance of the residuals is changing and a more appropriate model (than the standard `lm`) should take this variation into account. Again, a good graph has the red line being close to a horizontal line.

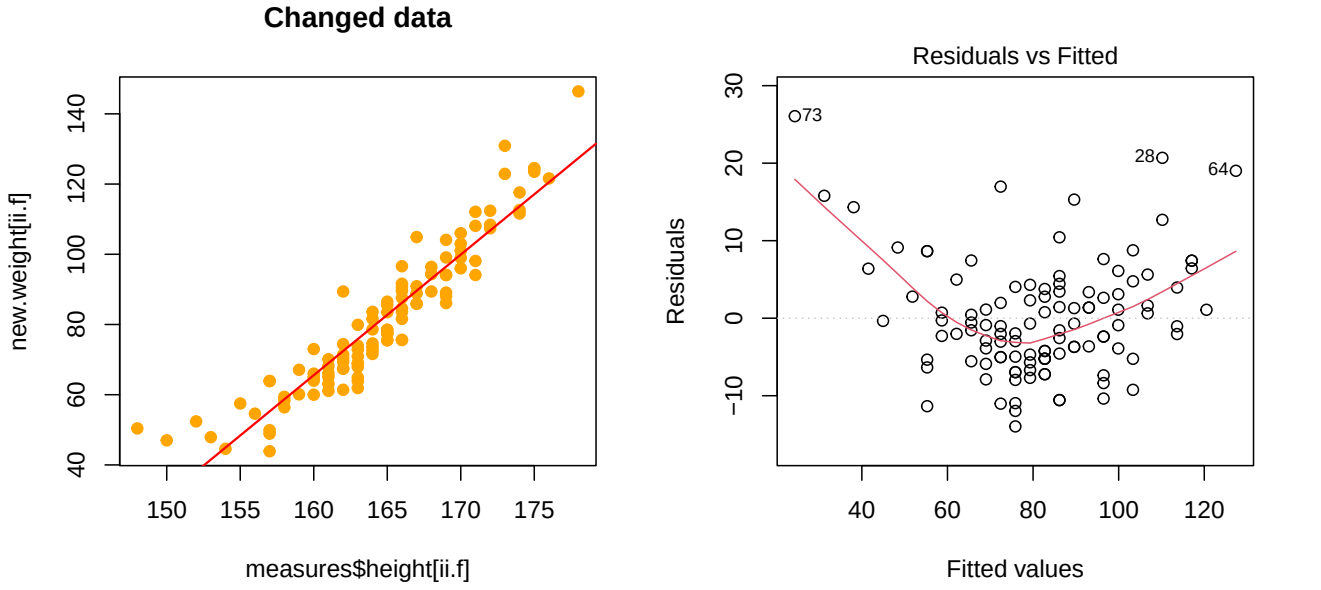
4. Standardized residuals vs. leverage should not show points that reach outside the curved areas of Cook's distance 1. Otherwise, such points would have a high influence on the slope of the linear model, and hence the `lm()` estimates from such data would not be reliable.

For our test data, the diagnostics look ok. There are 4 slightly outlying observations, that have to such a degree higher weight than predicted by their height, that they separate slightly from the line on the QQ-plot. However, these are still minor deviations and, importantly, they are not having a large influence on the fitted line, which can be seen on the lower right plot, where no points reach even a Cook's distance of 0.5. (Only a small region beyond Cook's distance of 0.5 can be seen at the top right corner of the plot, but no region with Cook's distance > 1 can be seen in this plot.)

Examples 7.2. Let's demonstrate how deviations from the linear model assumptions show up in the diagnostic plots. For that, let's change the existing female height-weight data in a few different ways.

1. Let's make weight to have an extra contribution of `+0.1*(height-150)^2` so that weight increases also quadratically, and not just linearly, as a function of height. Let's then fit the linear model and plot the model fit and the diagnostic plot 1.

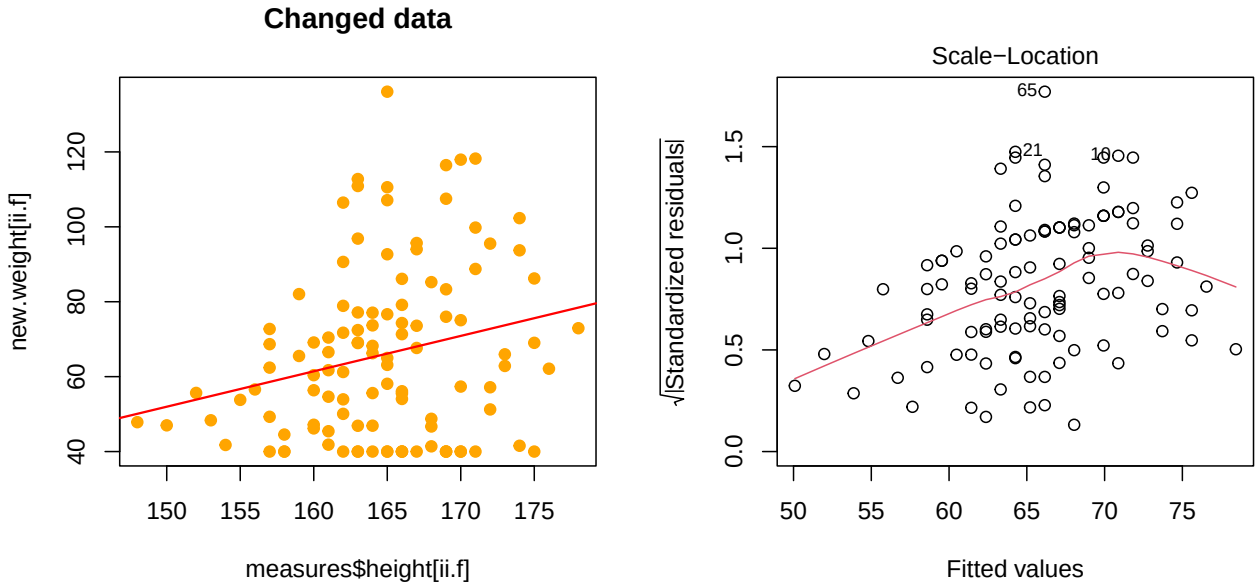
```
new.weight = measures$weight + 0.1*(measures$height - 150)^2
ii.f = (measures$sex == "F")
lm.new = lm(new.weight[ii.f] ~ measures$height[ii.f])
par(mfrow = c(1,2))
plot(measures$height[ii.f], new.weight[ii.f], pch = 19, col = "orange", main = "Changed data")
abline(lm.new, col = "red", lwd = 1.5)
plot(lm.new, 1) #plot only diagnostic plot no. 1
```



In the left-hand plot, we see the new data and the linear model fit that tends to underestimate weight at both ends of the height distribution. In the right-hand plot, we see a parabolic pattern of residuals vs. fitted values, where the residuals are systematically positive at both ends of the fitted values. This indicates that a linear model is not adequate and suggests that a quadratic term of the predictor should be included in the model. (We will come back to how to do this later in this lecture.)

2. Let's make weight to have more variance for the taller individuals by adding an extra contribution of `+ rnorm(, mean = 0, sd = 2*|height-150|/100)` to the wieght. Let's then fit the linear model and plot the model fit and the diagnostic plot 3.

```
new.weight = measures$weight + rnorm(nrow(measures), 0, 2*abs(measures$height - 150))
new.weight[new.weight < 40] = 40 #Let's set minimum new weight to 40kg
lm.new = lm(new.weight[ii.f] ~ measures$height[ii.f])
par(mfrow = c(1,2))
plot(measures$height[ii.f], new.weight[ii.f], pch = 19, col = "orange", main = "Changed data")
abline(lm.new, col = "red", lwd = 1.5)
plot(lm.new, 3) #plot only diagnostic plot no. 3
```

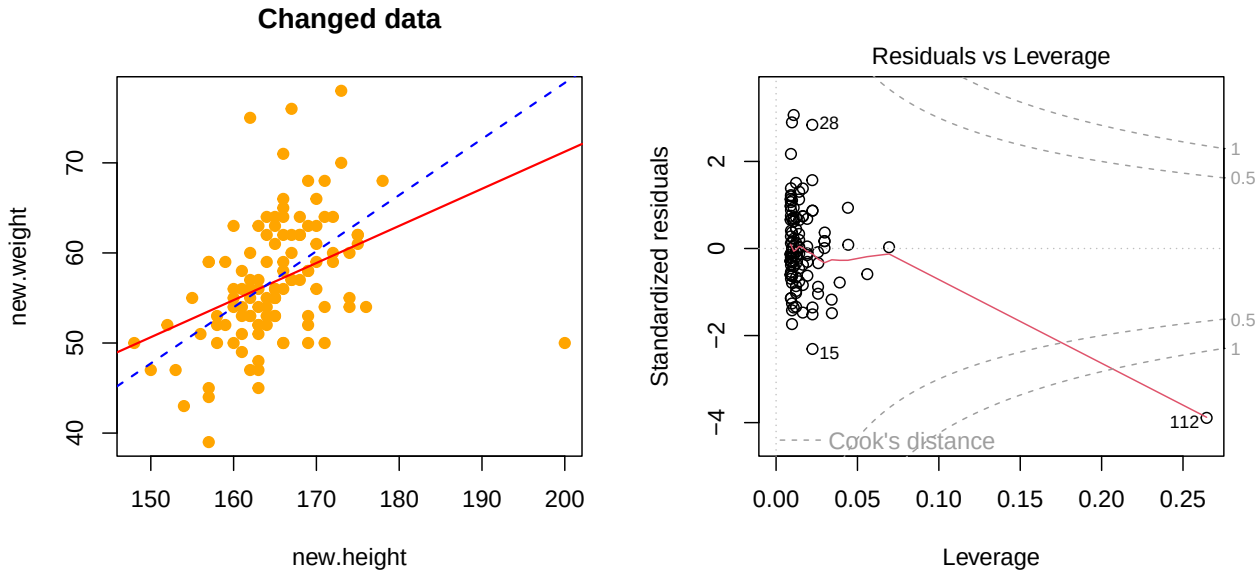


In the left-hand plot, we see that in the new data, the variance increases with height. In the right-hand plot, we see an increasing pattern in the standardized residuals, which indicates that the variance of residuals is changing as a function of the fitted values. This violates the assumptions of the linear regression model.

3. Let's add one new individual to data with `height = 200` and `weight = 50`. This individual will be an outlier (meaning that residual value is large) because weight is so small compared to height. The individual will also be influential for the slope of the line, which is shown by the observation reaching Cook's distance of 1 in Stndardized residuals vs. Leverage plot. Let's show the original fitted line without the outlier from `lm.f` object as a blue dashed line, and the new fitted line with the outlier individual included as a red line.

```
new.weight = c(measures$weight[ii.f], 50) #existing females + new_value 50
new.height = c(measures$height[ii.f], 200) #existing females + new_value 200
lm.new = lm(new.weight ~ new.height)
par(mfrow = c(1,2))
plot(new.height, new.weight, pch = 19, col = "orange", main = "Changed data")
```

```
abline(lm.new, col = "red", lwd = 1.5)
abline(lm.f, col = "blue", lwd = 1.5, lty = 2)
plot(lm.new, 5) #plot only diagnostic plot no. 5
```



In the left-hand plot, we see the outlier point at $x = 200, y = 50$. We also see that this one point changes the fitted line considerably from the blue line to the red line. In the right-hand plot, we see how the diagnostic plot no. 5 shows that the outlier point (indexed 112) reaches Cook's distance 1, which indicates that it is very influential for the slope of the line. In such cases, it is important to understand what is the reason for the outlier points and whether they could/should be excluded from the analysis. A typical reason for coarse outliers is a measurement error of some kind at some step of the data collection, but outlier points can also be completely real and valid observations.

Example analysis: Social factors in 1998

Let's study some social factors from around the world at the turn of the millenium using a UN data sets from 1998. This file is in csv format (comma separated values) and we can read it in using `read.csv()` function that is simply a version of `read.table()` function with the default parameter values set to correspond to a typical csv file. (And to make sure that the read.csv command indeed treats comma as the separator between values, we can explicitly require `sep = ","`.)

```
y = read.csv("UN98.csv", as.is = TRUE, header = TRUE, sep = ",")
head(y) #show some first rows
```

```
##                X region  tfr contraception educationMale educationFemale
## 1    Afghanistan   Asia 6.90            NA            NA              NA
## 2        Albania Europe 2.60            NA            NA              NA
## 3        Algeria Africa 3.81            52          11.1             9.9
## 4 American.Samoa   Asia   NA            NA            NA              NA
## 5        Andorra Europe   NA            NA            NA              NA
## 6         Angola Africa 6.69            NA            NA              NA
##   lifeMale lifeFemale infantMortality GDPperCapita economicActivityMale
## 1     45.0       46.0             154         2848                 87.5
## 2     68.0       74.0              32          863                   NA
## 3     67.5       70.3              44         1531                 76.4
```

```
## 4     68.0       73.0              11           NA                  58.8
## 5       NA         NA              NA           NA                    NA
## 6     44.9       48.1             124          355                    NA
##   economicActivityFemale illiteracyMale illiteracyFemale
## 1                    7.2         52.800            85.00
## 2                     NA             NA               NA
## 3                    7.8         26.100            51.00
## 4                   42.4          0.264             0.36
## 5                     NA             NA               NA
## 6                     NA             NA               NA
```

Read the description of the variables from http://vincentarelbundock.github.io/Rdatasets/doc/carData/UN98.html.

Let's give these variables shorter names to avoid writing long names.

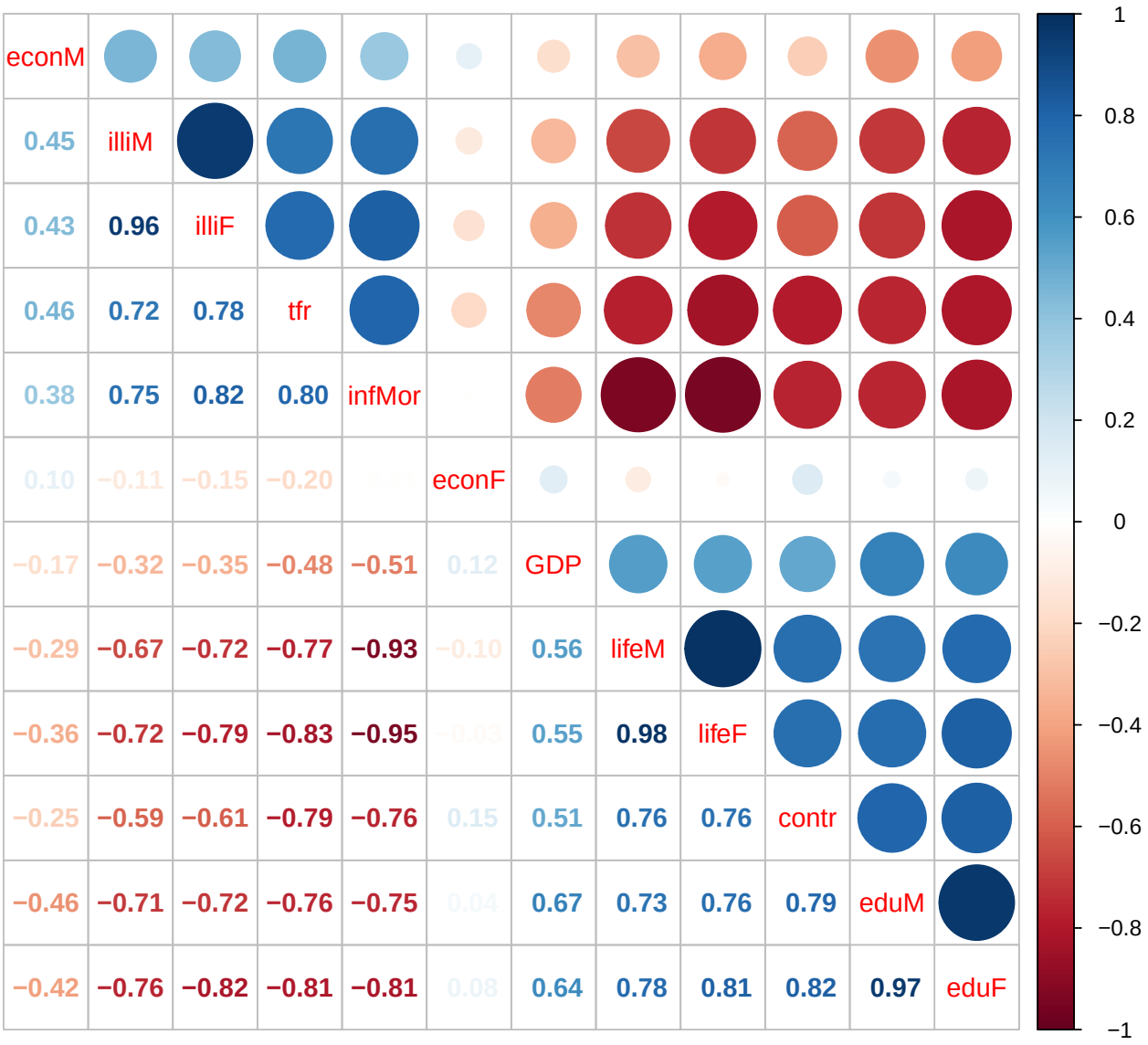
```
colnames(y) = c("country","region","tfr","contr","eduM","eduF","lifeM",
                "lifeF","infMor","GDP","econM","econF","illiM","illiF")
```

Let's plot the pairwise correlations using `corrplot` to an increased plotting area of size `fig.width = 8` and `fig.height = 8` (defined in the Rmd code block definition).

```
#install.packages("corrplot")
library(corrplot)
```

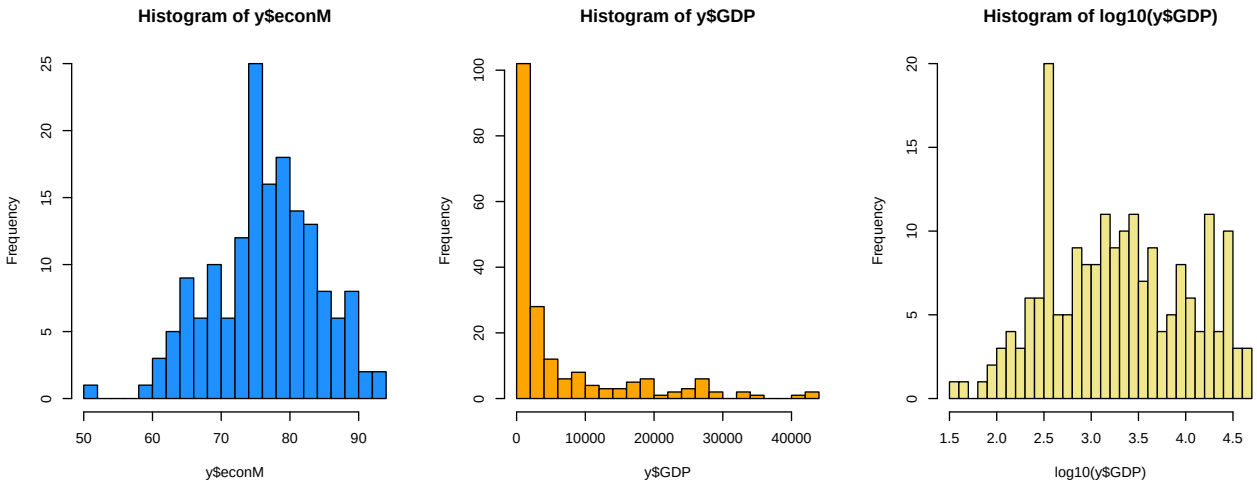
```
## corrplot 0.94 loaded
```

```
x = as.matrix(y[, 3:14]) #make a matrix of numeric columns of data.frame y for cor()
corr = cor(x, use = "pairwise")
corrplot.mixed(corr, order = "hclust")
```



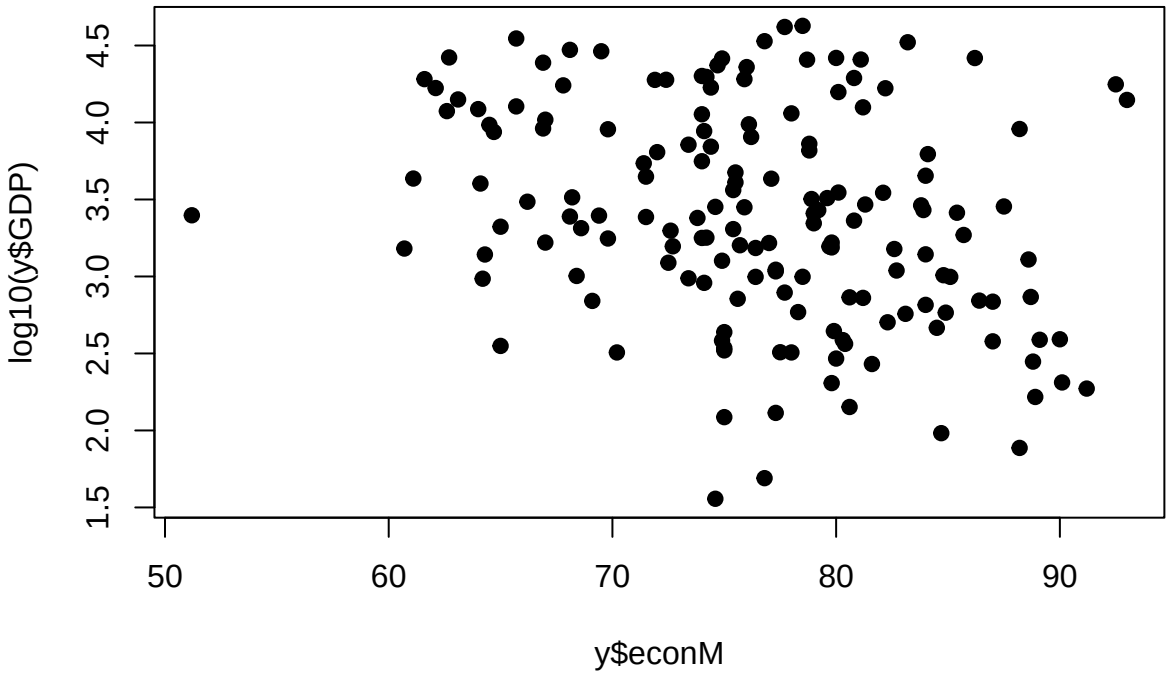
We see two clear blocks of highly correlated variables. First block is about positive things like life expectancy, education and availability of contraception and the second contains negative things like illiteracy and infant mortality. We observe that "percentage of economically active women" does not correlate much with the other variables. Perhaps surprisingly, GDP (gross domestic product, bruttokansantuote) doesn't correlate positively with economic activity in males either. Let's have a more careful look at these two variables.

```
par(mfrow = c(1,3))
hist(y$econM, col = "dodgerblue", breaks = 25)
hist(y$GDP, col = "orange", breaks = 25)
hist(log10(y$GDP), col = "khaki", breaks = 25)
```



We see that activity is roughly symmetric and very roughly Normally distributed but GDP is an example of a highly skewed distribution with a strong tail to right. This is typical of variables describing wealth or, more generally, variables that take only non-negative values. Often logarithm of such a variable is a much better input variable to linear regression as it is often more symmetric and Normally distributed, as is the case here as well. Thus, we will use log10 of GDP. (Note that variables need not be Normally distributed in order to be used in linear regression, but an approximate Normality often makes the model fit better.)

```
plot(y$econM, log10(y$GDP), pch = 19)
```



There does not seem to be clear patterns between these two variables, which agrees with the small magnitude of correlation.

Let's print out statistics for countries with over 90% Males "economically active".

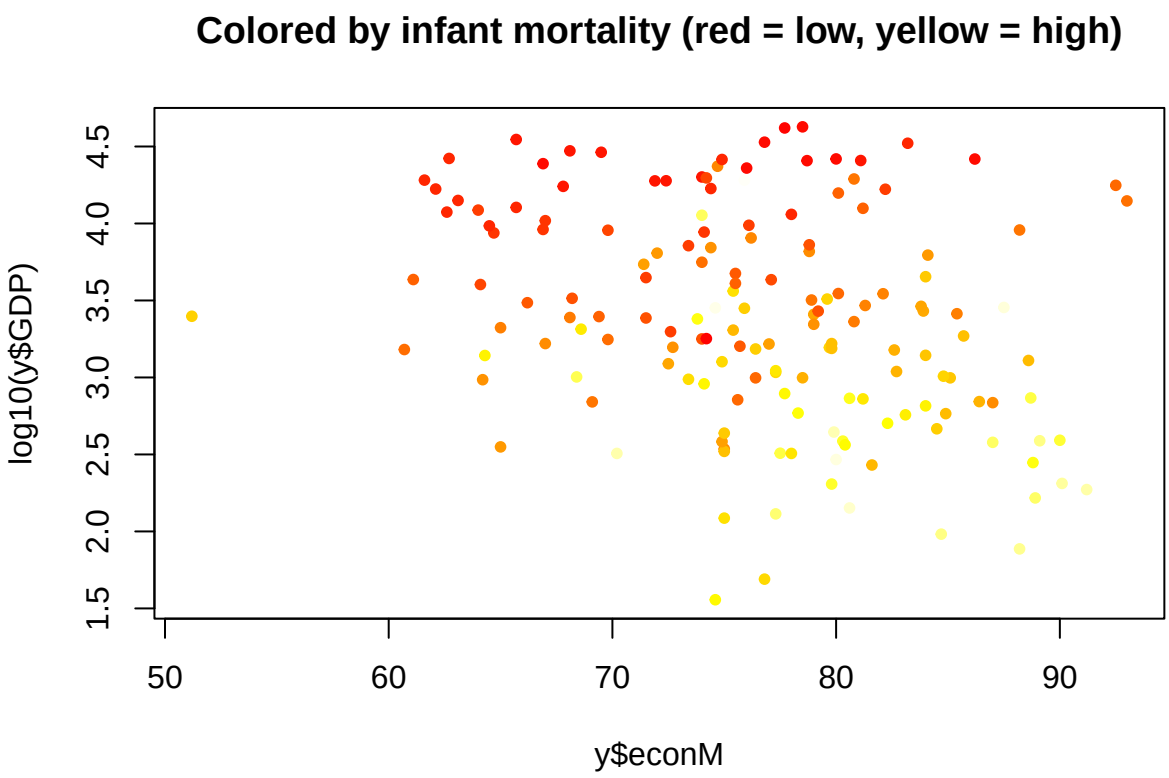
```
y[which(y$econM > 90),] #we use which() here because that automatically filters out NAs
```

```
##                 country region  tfr contr eduM eduF lifeM lifeF infMor   GDP
```

```
## 29                Burundi Africa 6.28     9  5.1  4.0  45.5  48.8    114   205
## 35                   Chad Africa 5.51    NA   NA   NA  46.3  49.3    115   187
## 153                 Qatar   Asia 3.77    32 10.6 11.6  70.0  75.4     17 14013
## 194 United.Arab.Emirates   Asia 3.46    NA  9.8 10.3  73.9  76.5     15 17690
##     econM econF illiM illiF
## 29   90.1  90.6  50.7  77.5
## 35   91.2  25.3  37.9  65.3
## 153  93.0  27.5  20.8  20.1
## 194  92.5  24.2  21.1  20.2
```

We have poor countries like Burundi and Chad and rich countries like Qatar and United Arab Emirates that both have a high proportion of male economic activity. One clear factor separating these groups is infant mortality. Let's look at the same plot but now coloring the points by infant mortality. We make as many colors as there are countries by using the `heat.colors(n)` palette that returns n colors from red to yellow. We index these colors by `rank` of each country in `infMor` variable (rank = 1 for the smallest `infMor` and rank = n for the largest `infMor`).

```
n = nrow(y)
plot(y$econM, log10(y$GDP), pch = 20, col = heat.colors(n)[rank(y$infMor)],
     main = "Colored by infant mortality (red = low, yellow = high)")
```



We see that `infMor` seems predictive of GDP (red associates with high GDP and yellow with low GDP). What does the linear model say when we predict GDP by both Economic activity and Infant mortality?

```
lm.0 = lm(log10(y$GDP) ~ econM + infMor, data = y, na.action = "na.exclude")
summary(lm.0)
```

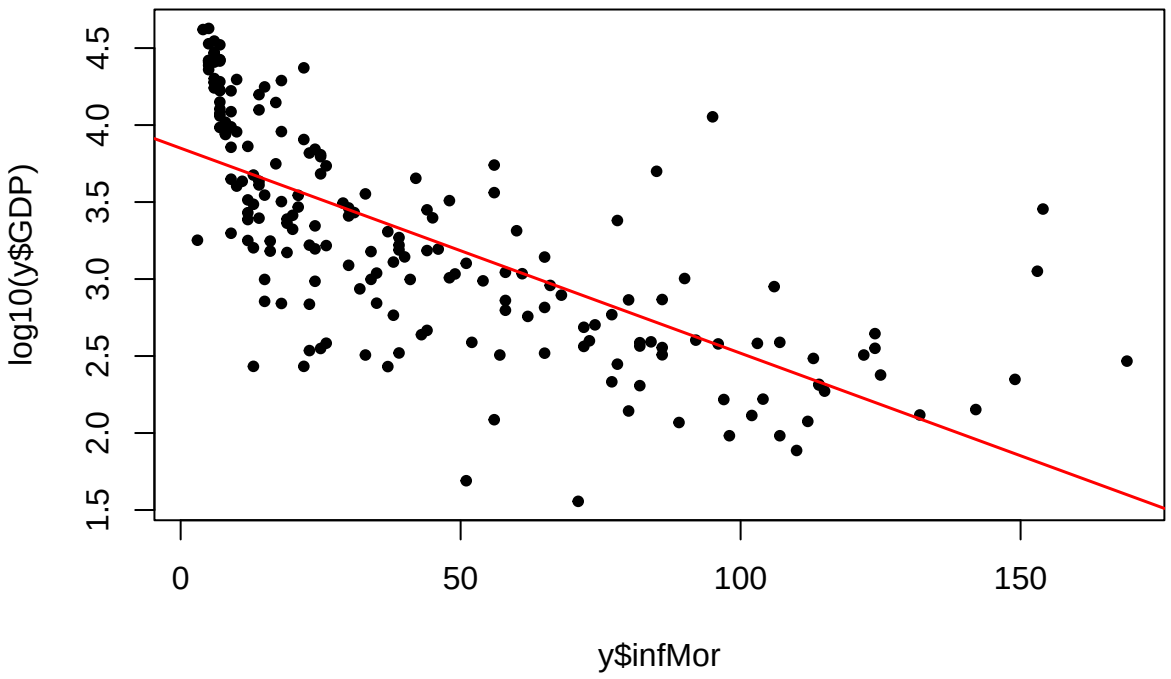
```
##
```

```
## Call:
## lm(formula = log10(y$GDP) ~ econM + infMor, data = y, na.action = "na.exclude")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49103 -0.29882 -0.04389  0.31470  1.75052
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.112476   0.417481   9.851   <2e-16 ***
## econM       -0.002795   0.005644  -0.495    0.621
## infMor      -0.014051   0.001251 -11.236   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.496 on 155 degrees of freedom
##   (49 observations deleted due to missingness)
## Multiple R-squared:    0.5,  Adjusted R-squared:  0.4936
## F-statistic: 77.51 on 2 and 155 DF,  p-value: < 2.2e-16
```

Economic activity is not an important predictor whereas Infant mortality is, and the model explains about 50% of the variation in GDP.

Let's do a linear model using Infant mortality as the only predictor.

```
lm.1 = lm(log10(GDP) ~ infMor, data = y, na.action = "na.exclude")
plot(y$infMor, log10(y$GDP), pch = 20)
abline(lm.1, col = "red", lwd = 1.5)
```




```
summary(lm.1)
```

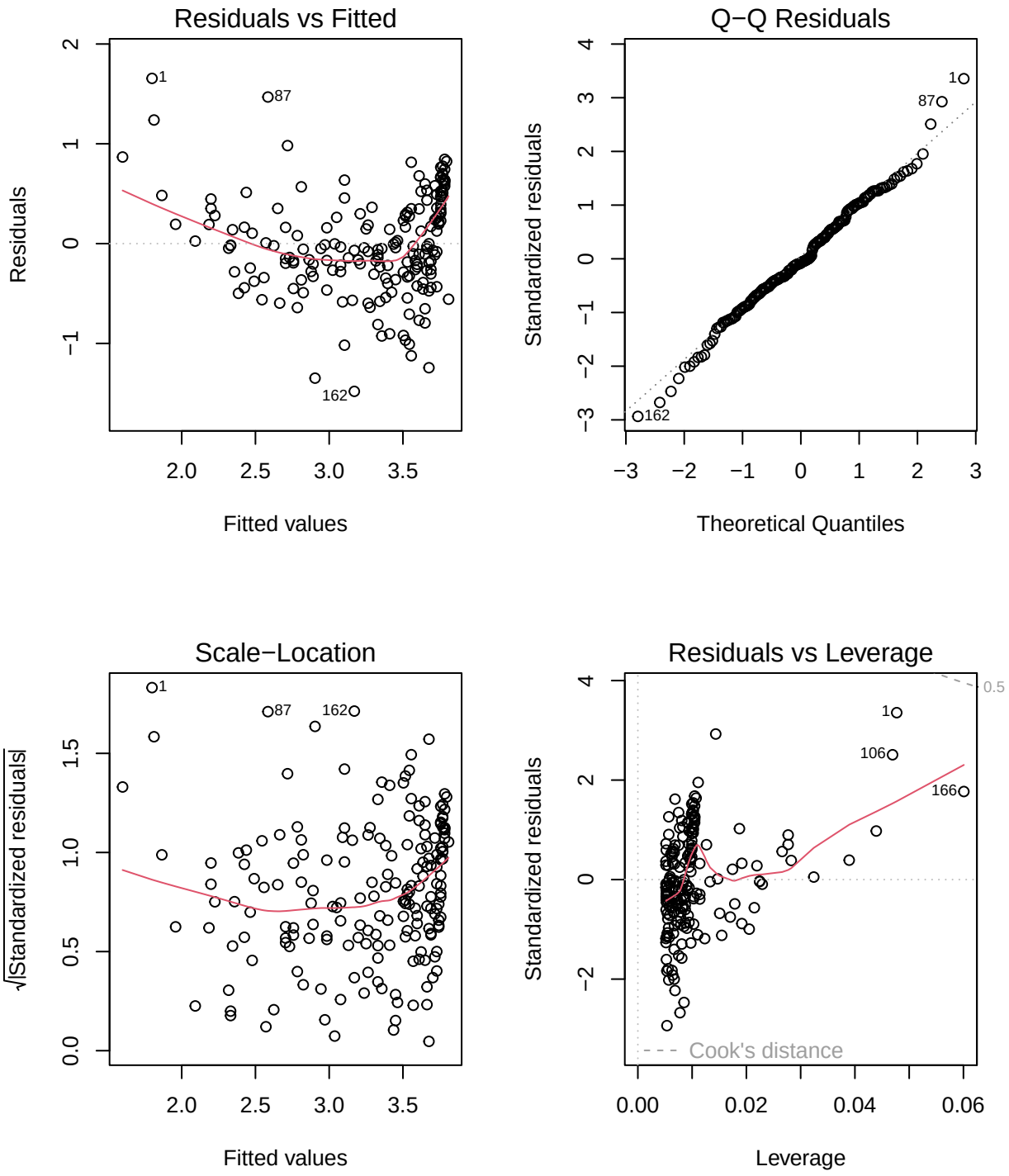
```
##
```

```
## Call:
## lm(formula = log10(GDP) ~ infMor, data = y, na.action = "na.exclude")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48033 -0.30369 -0.04039  0.34797  1.65554
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.8496233  0.0551925   69.75   <2e-16 ***
## infMor      -0.0133157  0.0009465  -14.07   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5057 on 191 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.5089, Adjusted R-squared:  0.5063
## F-statistic: 197.9 on 1 and 191 DF,  p-value: < 2.2e-16
```

We see that log10 of GDP is decreasing by 0.0133 for each unit of infant deaths (per 1000 infants). In other words, when infant mortality increases one per mille, GDP drops multiplicatively by a factor of 10^(-0.0133157)=0.9698047, i.e., it drops about 3%. This model alone explains about 50% of the variation in GDP.

Does this model fit the data well? It looks like there is a tendency of residuals being positive at the ends, which would suggest adding the second order term in the model. Let's check the diagnostic plots.

```
par(mfrow = c(2,2))
plot(lm.1)
```



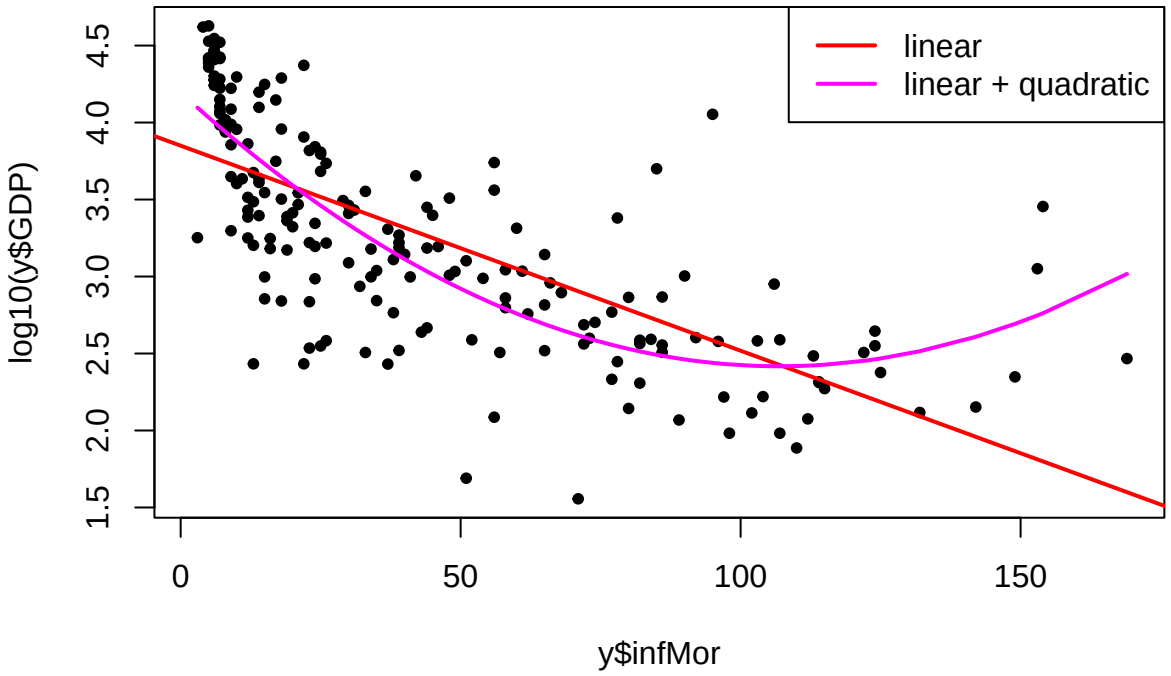
No other clear problems except that the residuals having a shape of a prabola in the top-left plot. Let's add the quadratic term in the model and our model becomes `log10(GDP) ~ InfMor + InfMor^2`. The quadratic term needs to be input through `I()` notation in the model formula as is done below.

```
lm.2 = lm(log10(GDP) ~ infMor + I(infMor^2), data = y, na.action = "na.exclude")
summary(lm.2)
```

```
##
## Call:
```

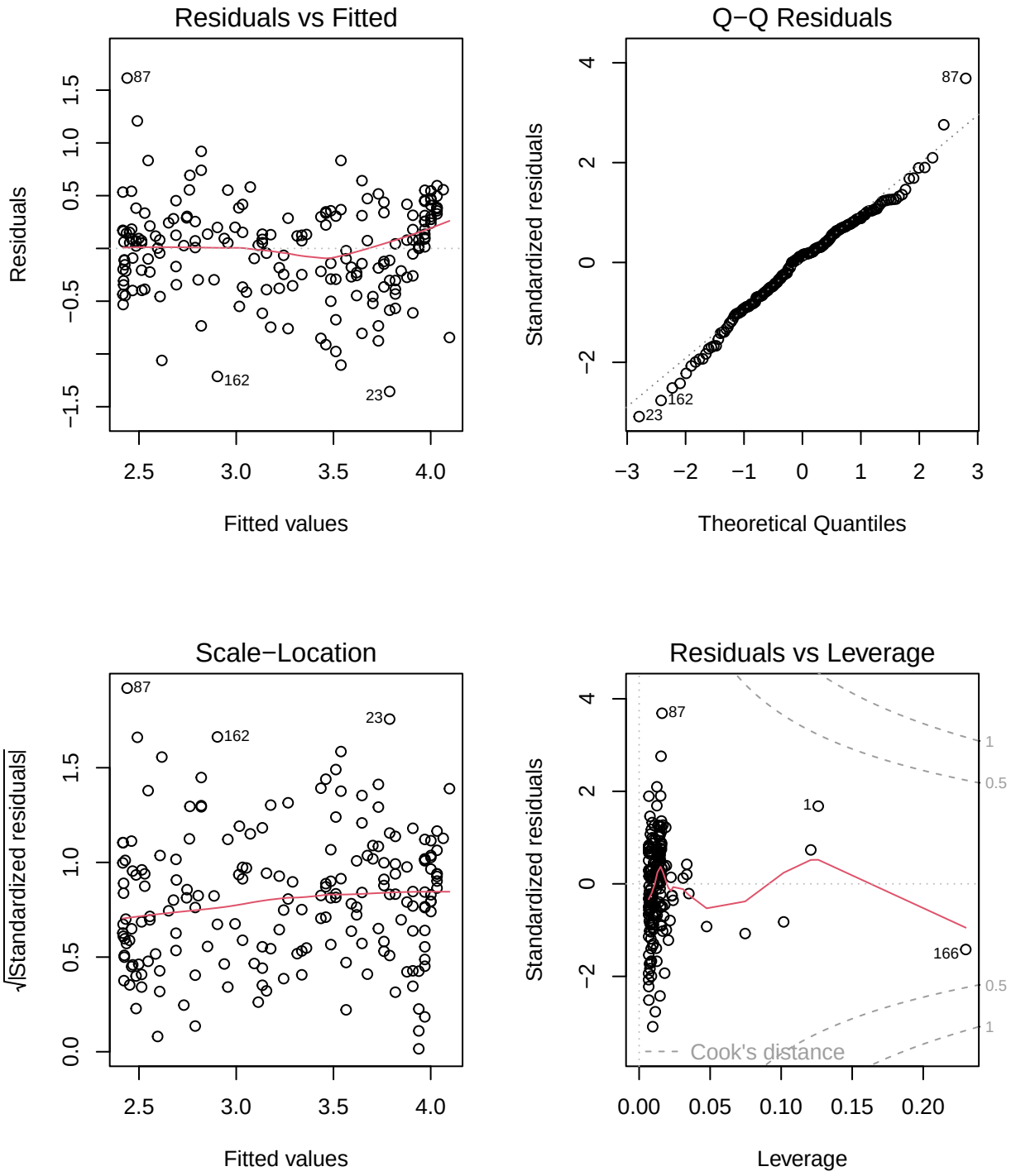
```
## lm(formula = log10(GDP) ~ infMor + I(infMor^2), data = y, na.action = "na.exclude")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.35590 -0.27433  0.06152  0.30104  1.61427
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.195e+00  6.546e-02   64.08  < 2e-16 ***
## infMor      -3.326e-02  2.690e-03  -12.36  < 2e-16 ***
## I(infMor^2)  1.555e-04  1.996e-05    7.79 4.23e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4414 on 190 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.6278, Adjusted R-squared:  0.6239
## F-statistic: 160.2 on 2 and 190 DF,  p-value: < 2.2e-16
```

```
plot(y$infMor, log10(y$GDP), pch = 20)
abline(lm.1, col = "red", lwd = 2) #linear model fit without squared term
#In order to show the lm.2 model fit in the same Figure, we connect the fitted values via lines,
# but we need to do that by ordering the points from the smallest infMor to the largest
x = y$infMor
x[is.na(y$GDP)] = NA #add NAs where GDP is missing
ii = order(x) #indexes from smallest infMor to largest infMor
lines(x[ii], fitted(lm.2)[ii], col = "magenta", lwd = 2) #show model fit with the squared term
legend("topright", lwd = 2, legend = c("linear","linear + quadratic"), col = c("red","magenta"))
```



This quadratic term clearly improves model fit (R2 increases from 50% to 62%) and the fitted values take into account the curved pattern in the data. Let's see the diagnostic plots.

```
par(mfrow = c(2,2))
plot(lm.2)
```



They look good.

We have found very strong (inverse) association between Infant mortality and GDP. Can we now say that Infant mortality is **the cause** of differences in GDP? The answer is **NO**. Causal statements can never be made from regression analysis alone! **Correlation is not causation** is an important fact to remember in all analyses of observational data! This is because there can be some third factor Z, that could be

causally influencing both X and Y, and hence the effect of Z could cause the observed correlation between X and Y even though neither X nor Y was causally influencing the other. Such a third variable Z is called a **confounder** when it is *confounding* the association between X and Y. A common statistics textbook example is the positive correlation between ice-cream sales and deaths by drowning. Is ice-cream causing drowning? No, it is sun and hot temperature that increase ice-cream sales and also increase activity on water, which then also leads to more accidents on water, and consequently deaths by drowning. So weather is confounding the association between ice-cream sales and drowning and the positive correlation between these two variables is not due to a causal relationship.

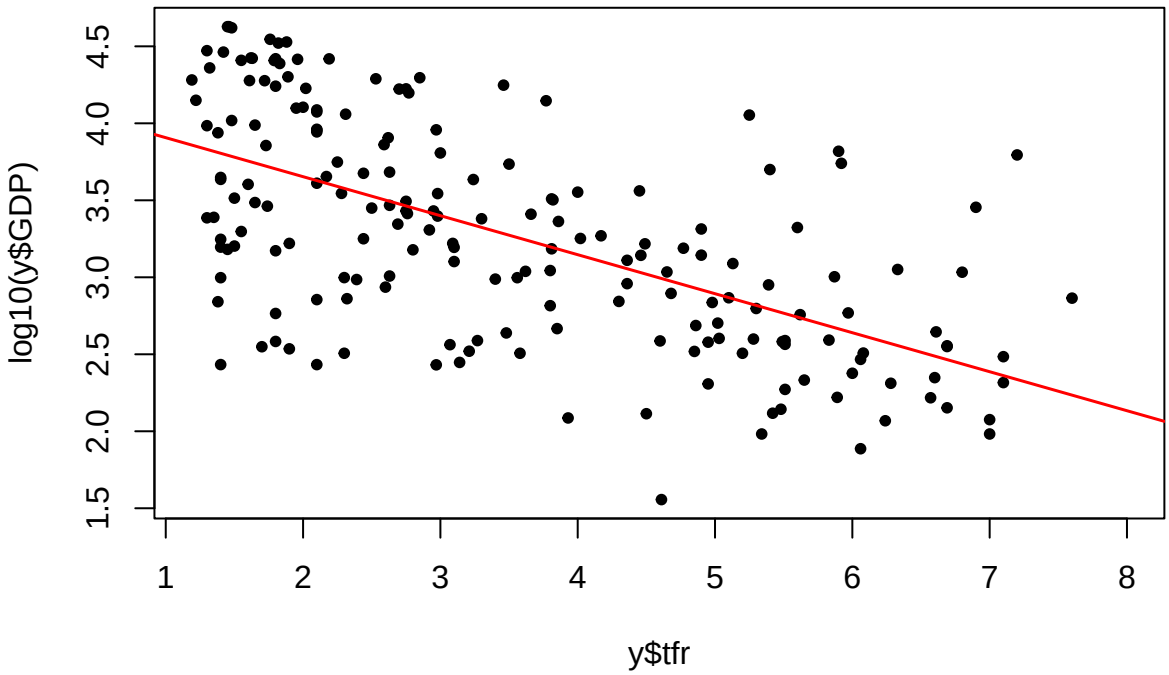
Since we can never measure all variables in observational studies, it remains impossible to do causal conclusions based only on observational data as some additional, unmeasured confounders could influence the regression estimates. Luckily, in clinical medicine and experimental science, there is an important method to overcome the problem of unmeasured variables: **randomized controlled trial (RCT)**. The situation is different for observational population-level epidemiology or social sciences, for example, where it is difficult to validate any causal statements.

Multiple regression with social factors Let's think more about what happens when we include multiple predictors in the same linear regression model. Above we established that Infant Mortality was a strong predictor of GDP. What about Total Fertility Rate (`tfr`)?

```
lm.3 = lm(log10(GDP) ~ tfr, data = y, na.action = "na.exclude")
summary(lm.3)
```

```
##
## Call:
## lm(formula = log10(GDP) ~ tfr, data = y, na.action = "na.exclude")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4358 -0.3703 -0.0134  0.4197  1.4586
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.15995    0.09179   45.32   <2e-16 ***
## tfr         -0.25333    0.02343  -10.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5647 on 188 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.3834, Adjusted R-squared:  0.3801
## F-statistic: 116.9 on 1 and 188 DF,  p-value: < 2.2e-16
```

```
plot(y$tfr, log10(y$GDP), pch = 20)
abline(lm.3, col = "red", lwd = 1.5)
```



A very clear association where higher fertility associates with decreasing GDP. The model explains about 38% of the variation. Let's then include both Infant Mortality `infMor` and `tfr` in the same model. This is an example of a multiple regression model.

```
lm.4 = lm(log10(GDP) ~ tfr + infMor, data = y, na.action = "na.exclude")
summary(lm.4)
```

```
##
## Call:
## lm(formula = log10(GDP) ~ tfr + infMor, data = y, na.action = "na.exclude")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.35185 -0.30586 -0.04307  0.34557  1.62657
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.887919   0.088177  44.092  < 2e-16 ***
## tfr         -0.021036   0.037179  -0.566    0.572
## infMor      -0.012434   0.001677  -7.416 4.14e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4958 on 186 degrees of freedom
##   (18 observations deleted due to missingness)
## Multiple R-squared:  0.5206, Adjusted R-squared:  0.5154
## F-statistic:   101 on 2 and 186 DF,  p-value: < 2.2e-16
```

In this model, only `infMor` seems important (effect is many SEs away from zero as denoted by small P-value) whereas the association with `tfr` has disappeared (effect dropped from -0.25 to -0.02 and P-value went from <2e-16 to 0.57). Such dramatic changes are possible because `infMor` and `tfr` are correlated and therefore part of what each of them explains alone in a regression model, can be explained by the other

when they are jointly included in a multiple regression model. In these data, it seems that the effect of `tfr` could be completely explained by the correlated variable `InfMor`. Multiple regression has here produced new insights over pairwise correlations. Namely, out of the two highly correlated variables `infMor` and `tfr` (with correlation $r = 0.8$), the former seems much better to predict changes in GDP than the latter.

We say that the effect size of `tfr` from multiple regression model "has been adjusted for" `infMor` whereas the effect of `tfr` from the simple regression model was not so adjusted. The interpretation for the effect estimates from multiple regression is that they tell what happens to the outcome variable (here log10(GDP)) per a unit change in the focal predictor (here `tfr`), when the other predictors (here `infMor`) are kept constant. Thus, here we have learned that if we look among the countries with a constant the level of infant mortality, then the fertility rate does not anymore noticeably affect GDP.

In Finnish we say: "Hedelmällisyysluvulla ei ole tilastollista yhteyttä BKT:hen kun analyysi on *vakioitu* lapsikuolleisuudella."