Report (for colleagues)

David Meyer

8. Mai 2016

Starting point

In a software development company, four metric variables were collected for 21 bugs (no missing values):

- time needed to fix the bug (*duration*)
- number of codelines (*codelines*)
- number of use cases (*usecases*)
- age of the programmer (*age*)

The objective of the analysis is to find a model that can be used to predict the time needed to fix a bug, and to make predictions for 200.000 and 600.000 lines of code, respectively.

Data management

The data were read using `read.table()`:

```
bugfixes = read.table("Y:/SS 2016/FH Technikum/BWI-2 DL DAS/Module_Books/Data/bugfixes.csv",
                      header = TRUE)
head(bugfixes)
```

```
##   duration codelines usecases age
## 1      120    183000       49  35
## 2      174    386000       86  44
## 3      188    467000       95  37
## 4      161    309000       70  35
## 5      157    305000       71  21
## 6      178    243000       85  36
```

Summary statistics

The distribution of *duration* is graphed using a boxplot (see figure 1). The five-number summary is given by:

```
summary(bugfixes$duration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   120.0   159.0   174.0   169.8   178.0   207.0
```

The distribution is left-skewed, the minimum (120 minutes) and the maximum (207 minutes) are considered as outliers. The median is 174 minutes, the medmed is 19.2738.

The boxplot for *codelines* is given in figure 2. The distribution is symmetric and doesn't show any conspicuous feature. The five-number summary is given by:

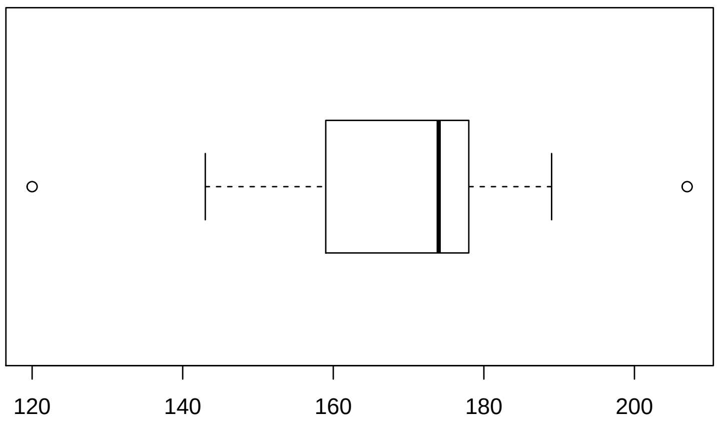

Figure 1: Boxplot of *duration*

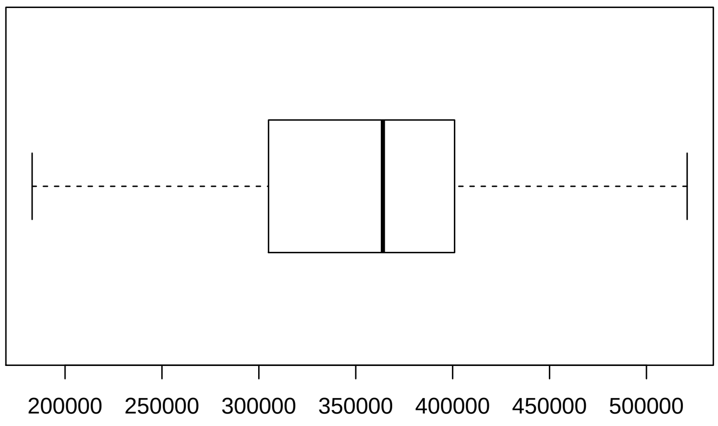

Figure 2: Boxplot of *codelines*

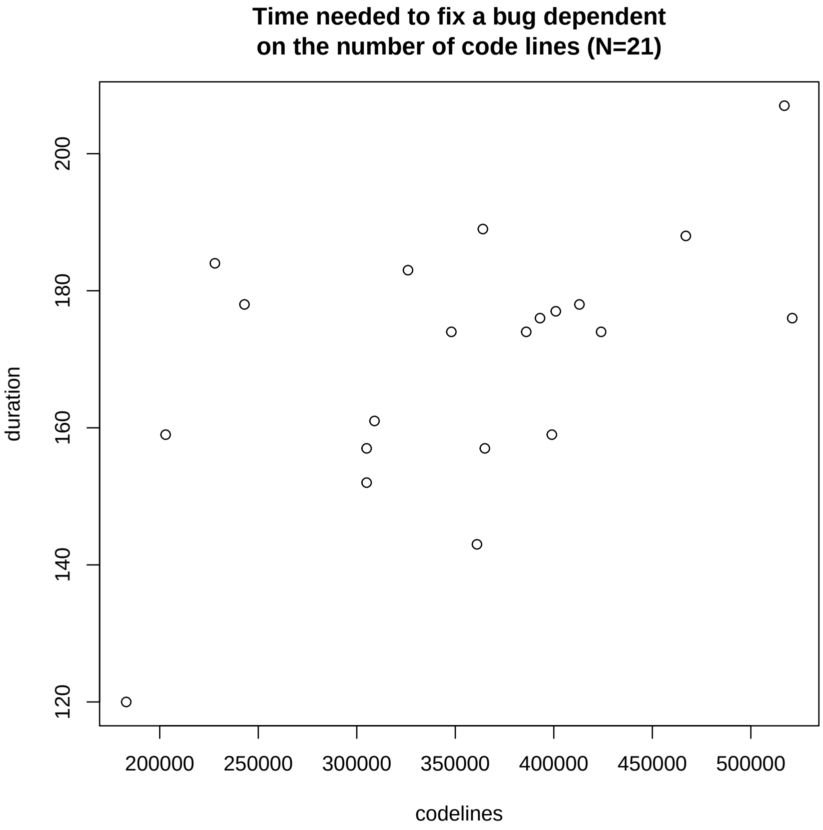

Figure 3: Scatterplot of *duration* and *codelines*

```
summary(bugfixes$codelines)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  183000  305000  364000  355300  401000  521000
```

According to this, the median is 364000, and the medmed is 81543.

Descriptive analysis

The joint distribution of *duration* and *codelines* is visualized using a scatterplot (see figure 3):

```
plot(duration ~ codelines, data = bugfixes,
     main = "Time needed to fix a bug dependent\non the number of code lines (N=21)")
```

There seems to be a positive linear relationship between *duration* and *codelines*.

Model estimation

Using `lm()`, we fit a linear model of the form: $y = a + bx$:

```r
model = lm(duration ~ codelines, data = bugfixes)
```

The coefficients are:

```r
round(coef(model), 5)
```

```
## (Intercept)   codelines
##   130.27545     0.00011
```

We get a model summary writing:

```r
summary(model)
```

```
##
## Call:
## lm(formula = duration ~ codelines, data = bugfixes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.639 -12.214   1.768   6.136  28.354
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.303e+02  1.414e+01   9.216 1.93e-08 ***
## codelines   1.113e-04  3.856e-05   2.885  0.00948 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.92 on 19 degrees of freedom
## Multiple R-squared:  0.3047, Adjusted R-squared:  0.2681
## F-statistic: 8.326 on 1 and 19 DF,  p-value: 0.009477
```

Both intercept and the coefficient of *codelines* are significant at the 0.05-level. The coefficient of determination amounts to 0.30 (p-value: 0.0094, hence also significant at the 0.05-level), i.e., the model explains approximately 30% of the variance of *duration*. In addition, we compute 95% confidence intervals for the parameters:

```r
round(confint(model), 5)
```

```
##                 2.5 %    97.5 %
## (Intercept) 100.69032 159.86059
## codelines     0.00003   0.00019
```

Model diagnostics

Before making any predictions, we have to assess the validity of the model assumptions. Since the sample is quite small, we have to check the normality assumption for the residuals in any case. This is done using a Q-Q-Plot, where the quantiles of the residuals are plotted against the theoretical values of the normal distribution (see figure 4).

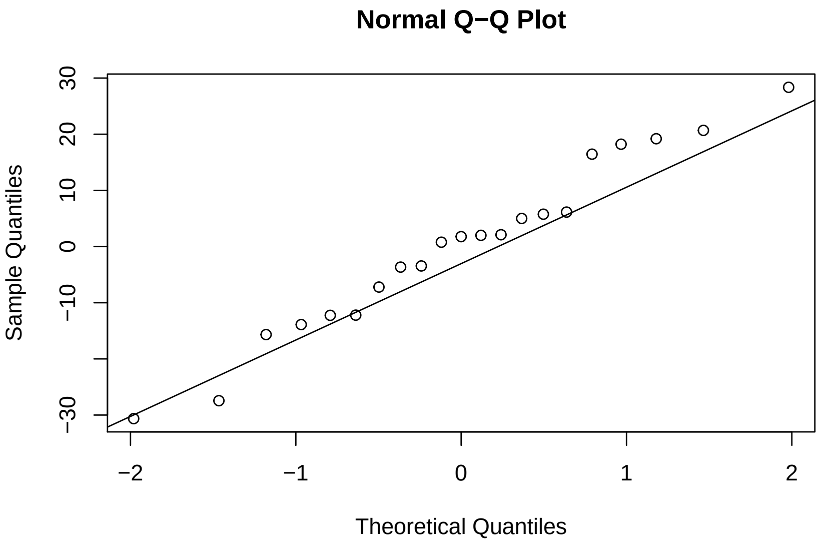

Figure 4: Q-Q-Plot of the residuals

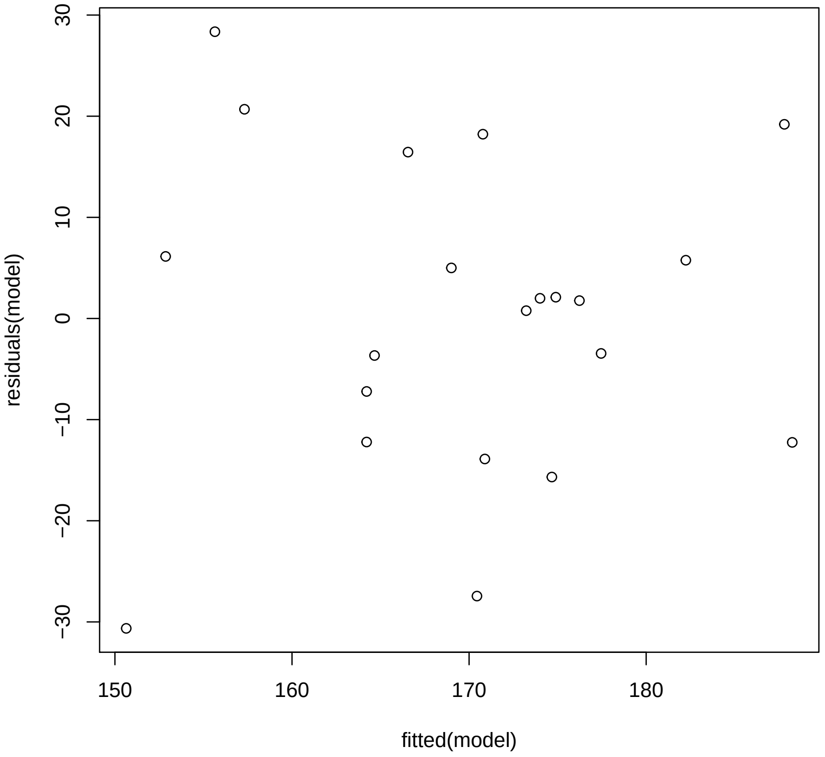

Figure 5: Scatterplot of the residuals against the fitted values.

We can see a strong correlation between the residuals and the theoretical values of the normal distribution, all points lie quite close to the line.

Furthermore, the correct specification of the model has to be verified by a scatterplot of the residuals against the fitted (= estimated) values (see figure 5). There is no visible pattern, thus the linear model should cover the information present in the data well.

Model summary

The regression equation for the expected time needed to fix a bug, given the lines of code, is:

$$E(\text{duration}|\text{codelines}) = 130.27 + 0.00011 \times \text{codelines}$$

including the margins of error:

- Intercept: [100.69, 159.86]
- Coefficient of codelines: [0.00003, 0.00019]

Prediction

We want to make predictions for 200.000 and 600.000 lines of code, respectively:

```
newdata = data.frame(codelines = c(200000, 600000))
format(newdata, scientific = FALSE)
```

```
##   codelines
## 1    200000
## 2    600000
```

The point and interval predictions for the *expected* values of *duration* are given by:

```
predict(model, newdata, interval = "confidence")
```

```
##        fit      lwr      upr
## 1 152.5303 138.0389 167.0216
## 2 197.0399 175.9910 218.0887
```

Die Interval predictions for the single observations are:

```
predict(model, newdata, interval = "prediction")[, -1]
```

```
##        lwr      upr
## 1 116.1865 188.8740
## 2 157.6200 236.4597
```

Figure 6 once again shows the scatterplot for *duration* and *codelines*, now including regression line (red), confidence bands (blue) and prediction bands (magenta).

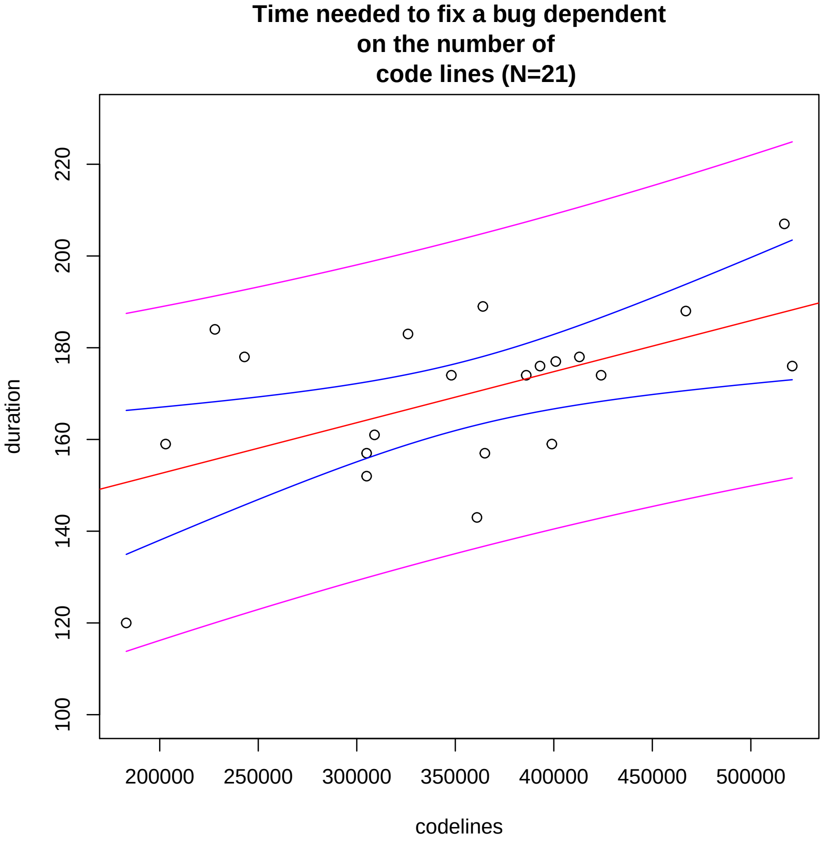

Figure 6: Regression line, confidence bands, and prediction bands for the time needed to fix a bug dependent on the number of code lines (N=21)