Practical examples to the course of Statistical Analysis

Anna Melnykova, Daria Bystrova

Abstract

This file serves as a complement and an illustration to the theoretical course on Statistical Analysis. It is by no mean complete and may contain errors, misprints and minor inaccuracies. You are encouraged to ask your questions and send any comments by e-mail, so that you will help to improve this document for the future generations and leave your trace in history.

Session 1. Simple Linear regression

Let us first recall the setting: we observe two random variables X and Y, and assume that they are linked by a linear relation, defined as follows:

$$Y = \beta_1 X + \beta_0 + \varepsilon, \tag{1}$$

where ε is an error of the measurements (of unknown distribution), Y is a dependent variable, X is an explanatory variable. β_1 and β_0 are unknown parameters of the model, which we will try to estimate.

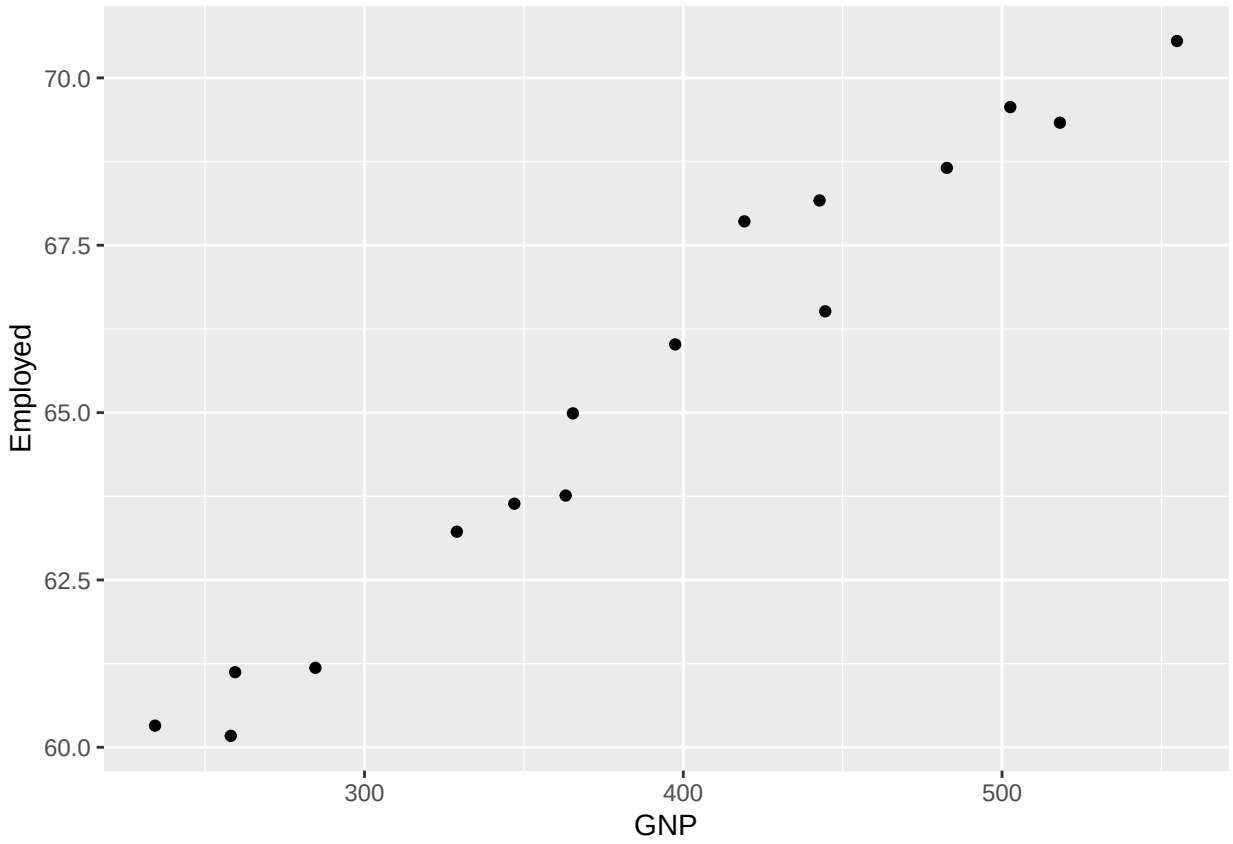
We start our experiment with setting up the environment and load the data. First, let us load some libraries which we will use. The first one is the set of standard datasets, the second one allows to create nice plots (but not only!). If the last library are not installed on your machine, you can use the command `install.packages("ggplot2")` in your R console prior to executing the next chunk of code.

In this example we will use the data from `longley` dataset, which contains economic data about the employment, unemployment, population, GNP and so on of the US population in years 1947-1962.

```
library(datasets)
library(ggplot2)
data("longley")
gnp <- longley$GNP # That will be our X
emp <- longley$Employed # That will be our Y
```

Let us visualize the variables of interest. Without going into details of the `ggplot` function, note that as a first parameter we give the dataset loaded in our workspace by a command `data("longley")`, and then we use the names of variables (not the exctracted vectors!) to build the scatter plot (that's what the command `geom_point()` is doing).

```
scat_plot <- ggplot(longley, aes(GNP, Employed))+geom_point()
scat_plot
```



We see a clear linear dependency between the GNP and the employed population. That is, we assume that we can express each observation of Y as a linear function of X, written as follows:

$$y_i = \beta_1 x_i + \beta_0 + \varepsilon_i,$$

where $i = 1, \ldots, n$, ε_i are independent indentically distributed error of measurements. Our aim is now to fit the regression line which explains the link between the variables. Recall that the natural estimators of the unknown parameters are such that they minimize the mean square error (i.e., distance between y_i and its "predicted" value $\beta_1 x_i + \beta_0$). Thus, the estimators are defined as follows:

$$\hat{\beta}_0, \hat{\beta}_1 = \underset{\beta_0, \beta_1}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \beta_1 x_i - \beta_0\right)^2.$$

Then, following the derivation which we have seen in the course, the point estimate of parameters β_1 (slope) and β_0 (intercept), are given by the following formula:

$$\hat{\beta}_1 = \frac{c_{xy}}{s_x^2}, \qquad \hat{\beta}_0 = \bar{y} + \frac{c_{xy}}{s_x^2} \bar{x}.$$

where c_{xy} denotes the empirical covariance between the vectors x and y, and s_x^2 is the empirical variance, $\bar{x}$ and $\bar{y}$ are empirical means of x and y respectively.

Important note: don't forget that the estimators $\hat{\beta}_1$ and $\hat{\beta}_0$ are, in fact, random variables, and for another realization of X and Y we will have slightly different values! What is important to know that those estimators are consistent (proof will be given in the course):

$$\mathbb{E}[\hat{\beta}_1] = \beta_1 \qquad \mathbb{E}[\hat{\beta}_0] = \beta_0.$$

For computing the value of the estimators in our specific case, you can use the following pre-computed values:

```
mean(gnp); var(gnp)
```

```
## [1] 387.6984
```

```
## [1] 9879.354
```

```
mean(emp); var(emp)
```

```
## [1] 65.317
```

```
## [1] 12.33392
```

```
cov(gnp, emp)
```

```
## [1] 343.3302
```

```
cor(gnp, emp)
```

```
## [1] 0.9835516
```

Note that the correlation coefficient is close to 1, which clearly indicates the existing linear dependency between the variables. Please note, however, that the linear correlation in data does not imply that there is an evidence of two events being dependent on the other, or that one thing causes another! In our example, if we compute the correlation coefficients between each pair of variables in a dataset, we could come to conclusion that the increase of GNP can be also explained by a total population grows, or that a population growth can be traced back to the increase of employed people, or that all those factors depend linearly on the year we are in (which is not true, of course!). More funny plots and spurious correlations can be found on this website: https://www.tylervigen.com/spurious-correlations

But let's go back to the point and check if your computations were correct!

```
b1 <- cov(gnp, emp)/var(gnp); b1
```

```
## [1] 0.03475229
```

```
b0 <- mean(emp) - b1*mean(gnp); b0
```

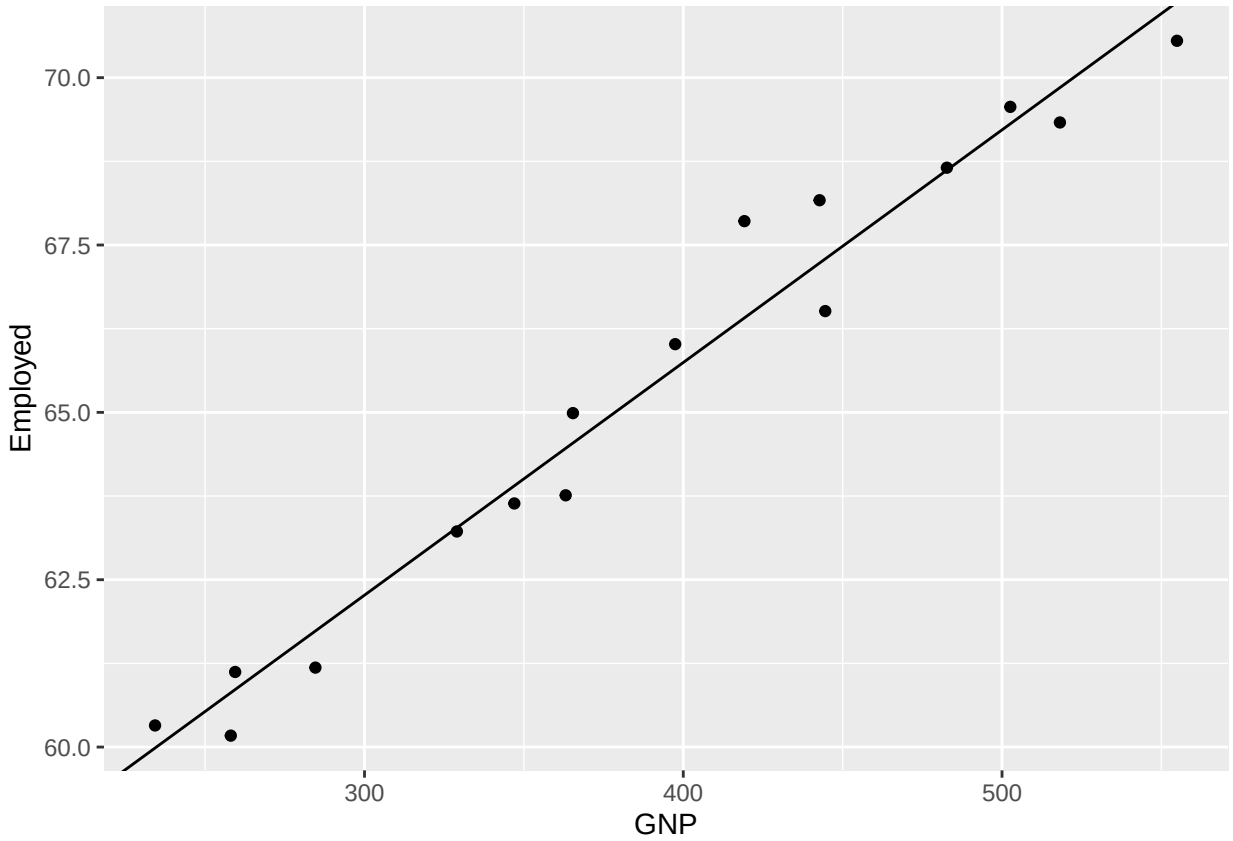
```
## [1] 51.84359
```

Now, we will check if it corresponds to the result of the built-in function in R (function `lm`, which stands for *linear model*), and plot the obtained line!

```
model_fit <- lm(Employed~GNP, data = longley)
model_fit
```

```
##
## Call:
## lm(formula = Employed ~ GNP, data = longley)
##
## Coefficients:
## (Intercept)          GNP
##    51.84359      0.03475
```

```
scat_plot + geom_abline(intercept = b0, slope = b1)
```

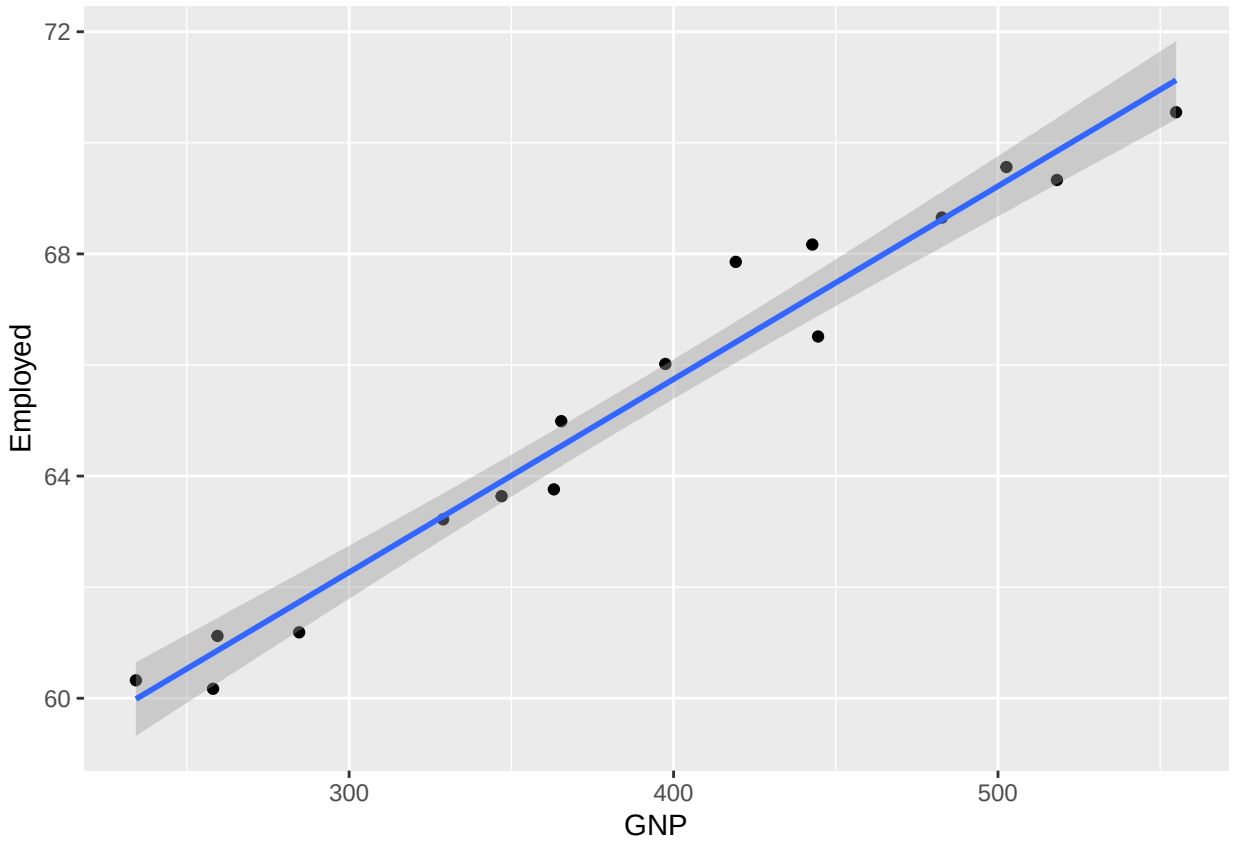


Feel free to execute the commands on your machine and have a look on all the information encoded in the variable `model_fit` now. In particular, you will find there a vector of residuals (`model_fit$residuals`), i.e. error of "approximation" of each observed value by the line, the respective fitted values (`model_fit$fitted_values`) and other interesting things. More generally, if you type `?lm` in your console, you will find all the info about the parameters which this function take, the output and so on.

To finish with, let me show how the same result can be obtained directly with the built-in methods of `ggplot`. Note that on the plot below, the built-in command is "smarter" and not only gives the point estimates of β_0 and β_1, but a confidence interval as well (95% by default), which you see as a grey region on the plot. Finally, in order to obtain the confidence intervals (that's something we are not going to do right now), one should have more information about the distribution of the estimators $\hat{\beta}_0, \hat{\beta}_1$. These properties are linked to the properties of ε_i (for example, if we assume the error to be gaussian, the estimators would be normal). That will be a subject of further lectures.

```
# scat_plot + stat_smooth() # The first function is commented out as we don't really want to talk about
scat_plot + stat_smooth(method = lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```


72
68
64
60
Employed
300
400
500
GNP