Data Handling: Import, Cleaning and Visualisation

Lecture 11: Visualization and Dynamic Documents

Prof. Dr. Ulrich Matter
(University of St. Gallen)

13/12/2018

1 Data display

- overview of last step in Data Science process
- low level: display data in R Murrell (2009) 9.10, only key aspects (use the practical aspects of this to start the workshop)
- visualization: plotting with gg (again, maybe part of the code examples in exercises)
- dynamic documents (partly last part of Murrell (2009) 9.10, rest from webmining: tables etc.), basics of markdown (focus particularly on this in exercises)

2 Data Visualization with R (`ggplot2`)

2.1 'Grammer of Graphics'

A few years back, Leland Wilkinson (statistician and computer scientist) wrote an influential book called 'The Grammar of Graphics'. In the book, Wilkinson develops a formal description ('grammar') of graphics used in statistics, illustrating how different types of plots (bar plot, histogram, etc.) are special cases of an underlying framework. Particularly, that we can think of graphics as consisting of different design-layers and that we can build and describe them layer by layer (see here for an illustration of this idea.

This framework got implemented in R with the very prominent `ggplot2`-package, building on the already very powerful R graphic engine. The result is a user-friendly environment to visualize data with R with enormous potential to plot almost any graphic illustrating data.

2.2 `ggplot2` basics

Using `ggplot2` to generate a basic plot in R is quite simple. Basically, it involves three key points:

1. The data must be stored in a `data.frame`
2. The starting point of a plot is always the function `ggplot()`
3. The first line of plot code declares the data and the 'aesthetics' (what variables are mapped to the x-/y-axes):

```r
ggplot(data = my_dataframe, aes(x= xvar, y= yvar))
```

2.3 Tutorial

In the following, we learn the basic functionality of `ggplot` by applying it to the `swiss` dataset introduced above.

2.3.1 Loading/preparing the data

First, we load and inspect the data. Among other variables it contains information about the share of inhabitants of a given Swiss province who indicate to be of Catholic faith (and not Protestant).

```
# load the R package
library(ggplot2)
# load the data
data(swiss)
# get details about the data set
# ?swiss
# inspect the data
head(swiss)
```

```
##              Fertility Agriculture Examination Education Catholic Infant.Mortality
## Courtelary        80.2        17.0          15        12     9.96             22.2
## Delemont          83.1        45.1           6         9    84.84             22.2
## Franches-Mnt      92.5        39.7           5         5    93.40             20.2
## Moutier           85.8        36.5          12         7    33.77             20.3
## Neuveville        76.9        43.5          17        15     5.16             20.6
## Porrentruy        76.1        35.3           9         7    90.57             26.6
```

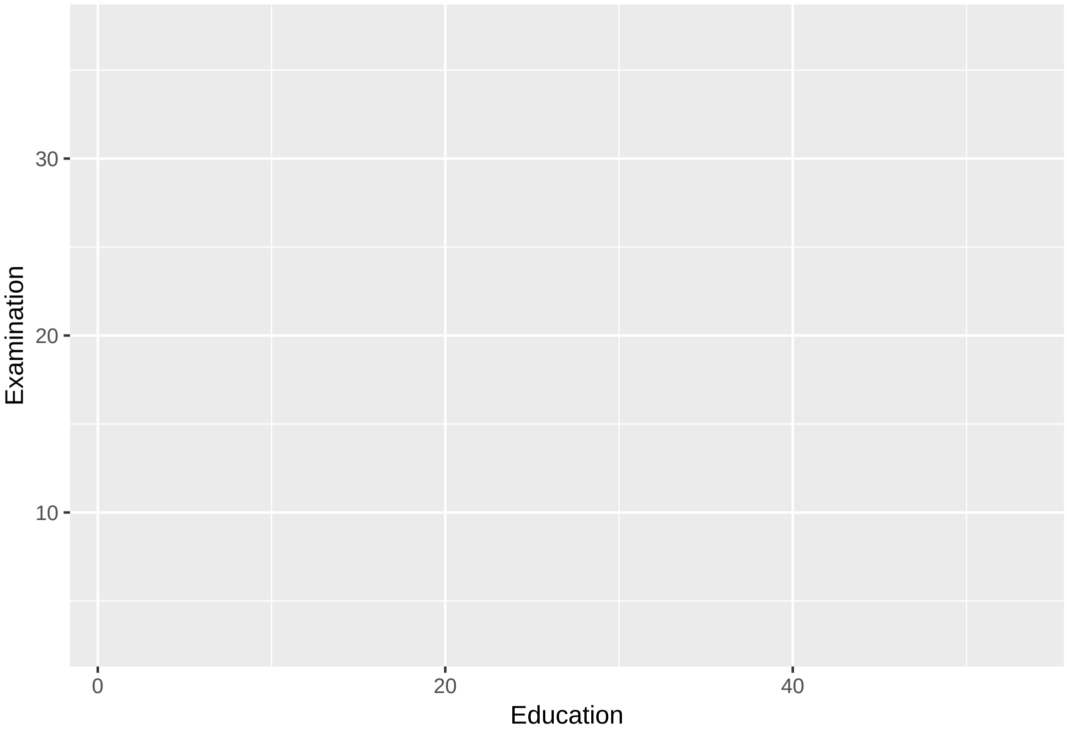
As we do not only want to use this continuous measure in the data visualization, we generate an additional factor variable called `Religion` which has either the value `'Protestant'` or `'Catholic'` depending on whether more then 50 percent of the inhabitants of the province are Catholics.

```
# code province as 'Catholic' if more than 50% are catholic
swiss$Religion <- 'Protestant'
swiss$Religion[50 < swiss$Catholic] <- 'Catholic'
swiss$Religion <- as.factor(swiss$Religion)
```

2.3.2 Data and aesthetics

We initiate the most basic plot with `ggplot()` by defining which data we want to use and in the plot aesthetics which variable we want to use on the x and y axes. Here, we are interested in whether the level of education beyond primary school in a given district is related with how well draftees from the same district do in a standardized army examination (% of draftees that get the highest mark in the examination).

```
ggplot(data = swiss, aes(x = Education, y = Examination))
```

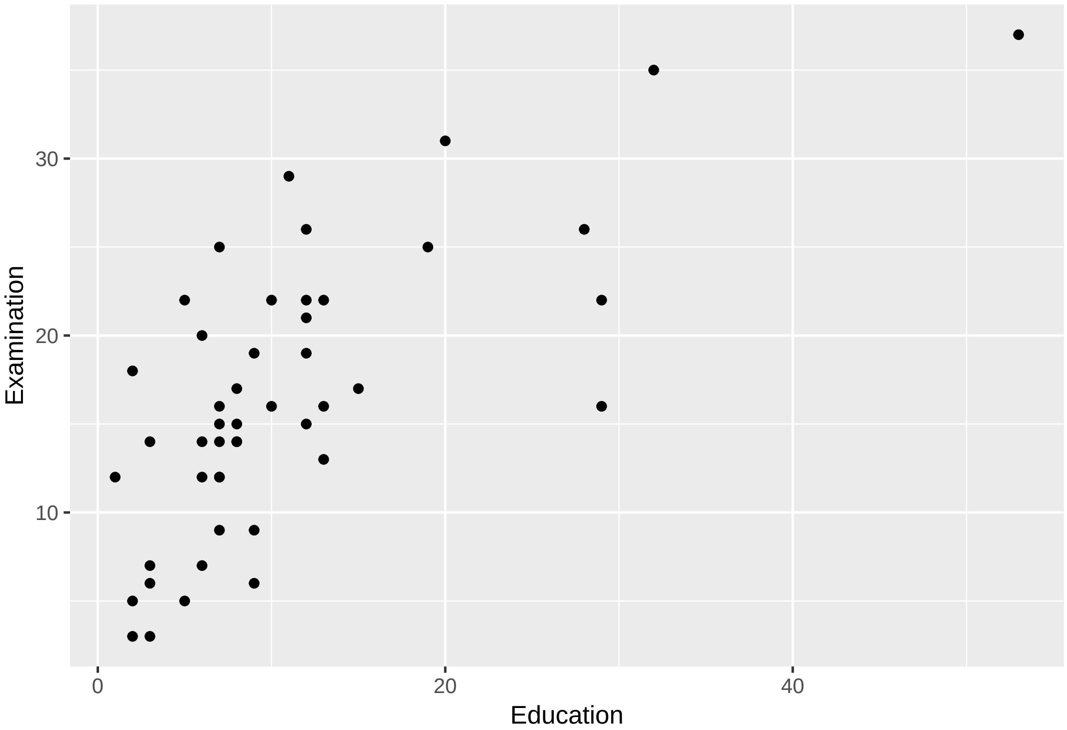


As we have not yet defined according to what rules the data shall be visualized, all we get is an empty ‘canvas’ and the axes (with the respective label and ticks indicating the range of the values).

2.4 Geometries (~the type of plot)

To actually plot the data we have to define the ‘geometries’, defining according to which function the data should be mapped/visualized. In other words, geometries define which ‘type of plot’ we use to visualize the data (histogram, lines, points, etc.). In the example code below, we use `geom_point()` to get a simple point plot.

```r
ggplot(data = swiss, aes(x = Education, y = Examination)) +
     geom_point()
```

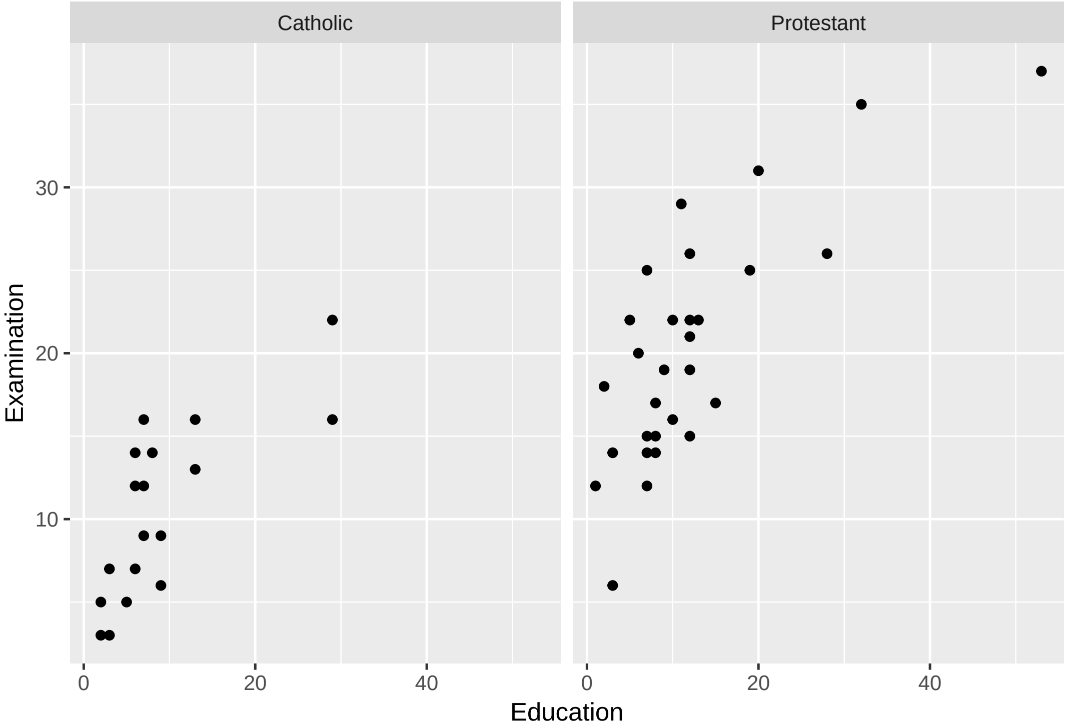


The result indicates that there is a positive correlation between the level of education and how well draftees do in the examination. We want to better understand this correlation. Particularly, what other factors could drive this picture.

2.4.1 Facets

According to a popular thesis, the protestant reformation and the spread of the protestant movement in Europe was driving the development of compulsory schooling. It would thus be reasonable to hypothesize that the picture we see is partly driven by differences in schooling between Catholic and Protestant districts. In order to make such differences visible in the data, we use 'facets' to show the same plot again, but this time separating observations from Catholic and Protestant districts:

```
ggplot(data = swiss, aes(x = Education, y = Examination)) +
     geom_point() +
     facet_wrap(~Religion)
```

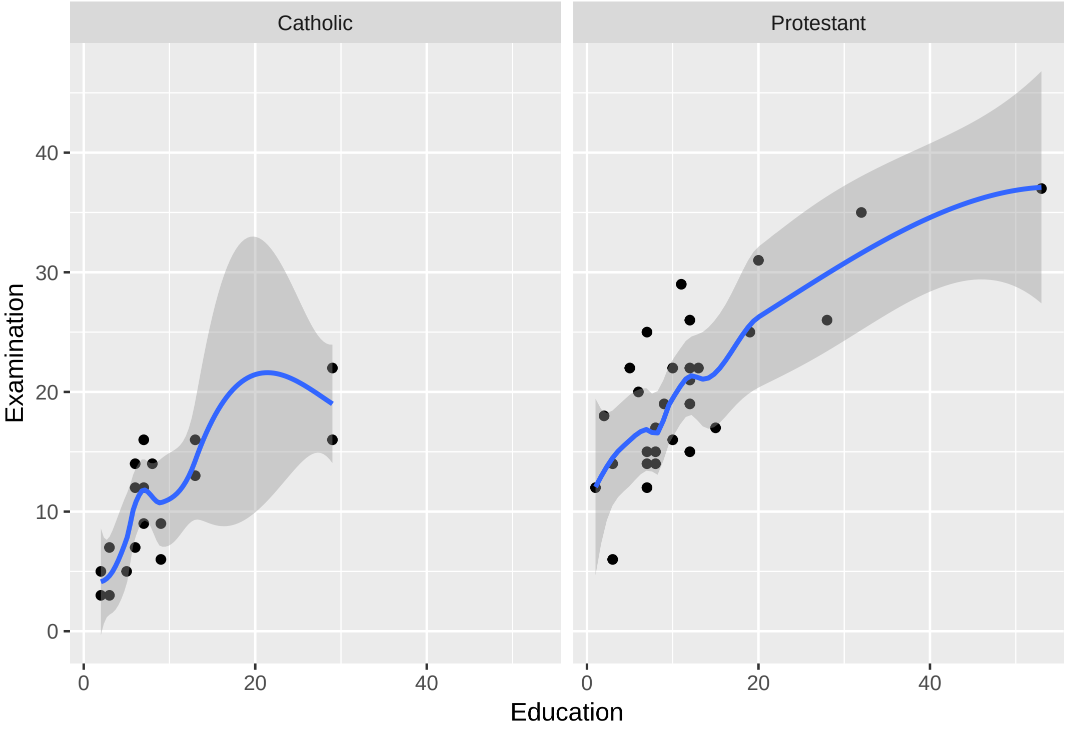


Draftees from protestant districts tend to do generally better (which might be an indication of better primary schools, or a generally stronger focus on scholastic achievements of Protestant children). However, the relationship between education (beyond primary schools) and examination success seems to hold for either type of districts.

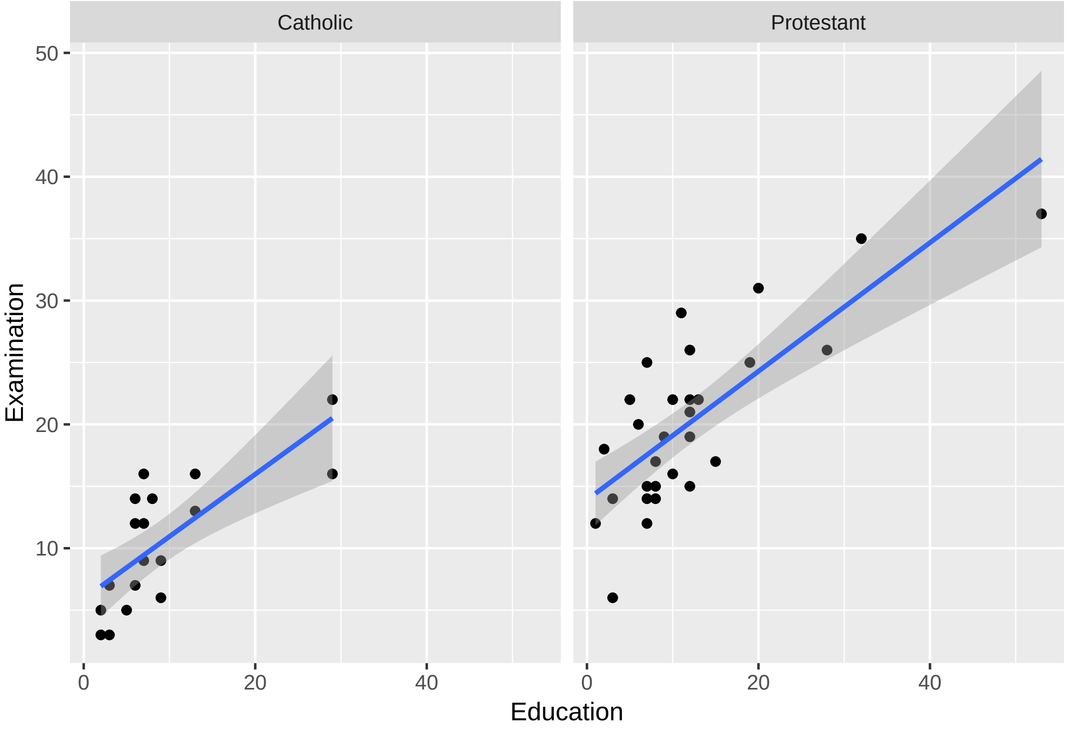
2.4.2 Additional layers and statistics

Let's visualize this relationship more clearly by drawing trend-lines through the scatter diagrams. Once with the non-parametric 'loess'-approach and once forcing a linear model on the relationship between the two variables.

```
ggplot(data = swiss, aes(x = Education, y = Examination)) +
     geom_point() +
     geom_smooth(method = 'loess') +
     facet_wrap(~Religion)
```



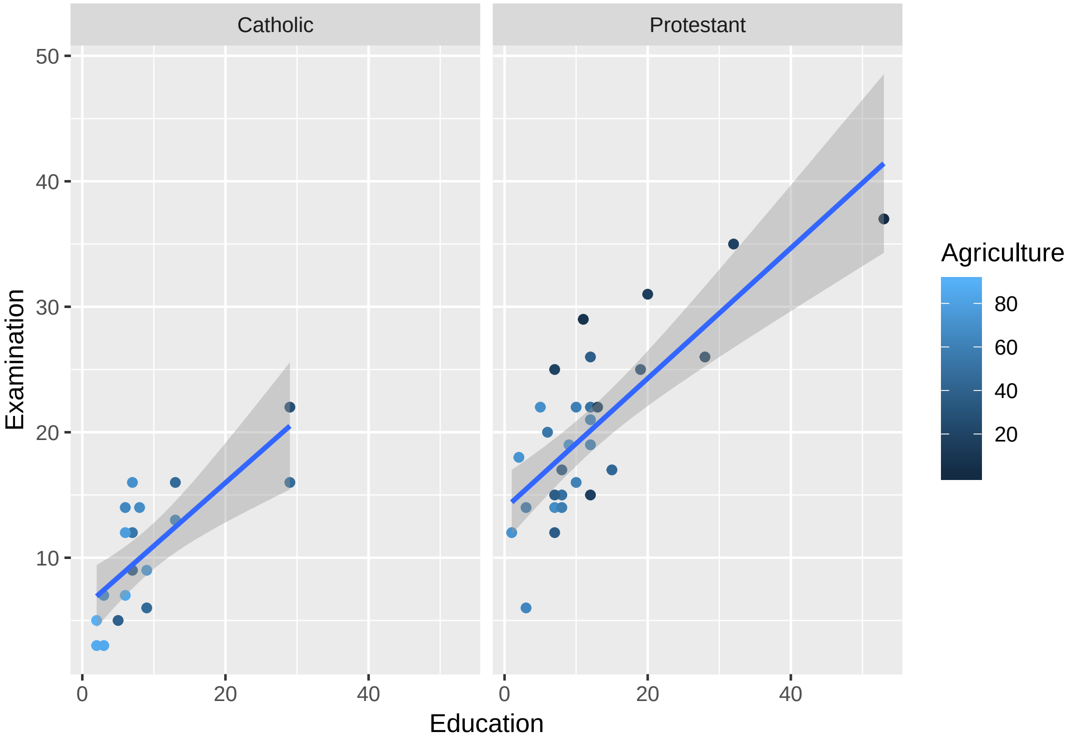
```r
ggplot(data = swiss, aes(x = Education, y = Examination)) +
     geom_point() +
     geom_smooth(method = 'lm') +
     facet_wrap(~Religion)
```



2.4.3 Additional aesthetics

Knowing a little bit about Swiss history and geography, we realize that particularly rural cantons in mountain regions remained Catholic during the reformation. In addition, cantonal school systems historically took into account that children have to help their parents on the farms during the summers. Thus in some rural cantons schools were closed from spring until autumn. Hence, we might want to indicate in the plot which point refers to a predominantly agricultural district. We use the aesthetics of the point geometry to color the points according to the '`Agriculture`'-variable (the % of males involved in agriculture as occupation).

```
ggplot(data = swiss, aes(x = Education, y = Examination)) +
     geom_point(aes(color = Agriculture)) +
     geom_smooth(method = 'lm') +
     facet_wrap(~Religion)
```

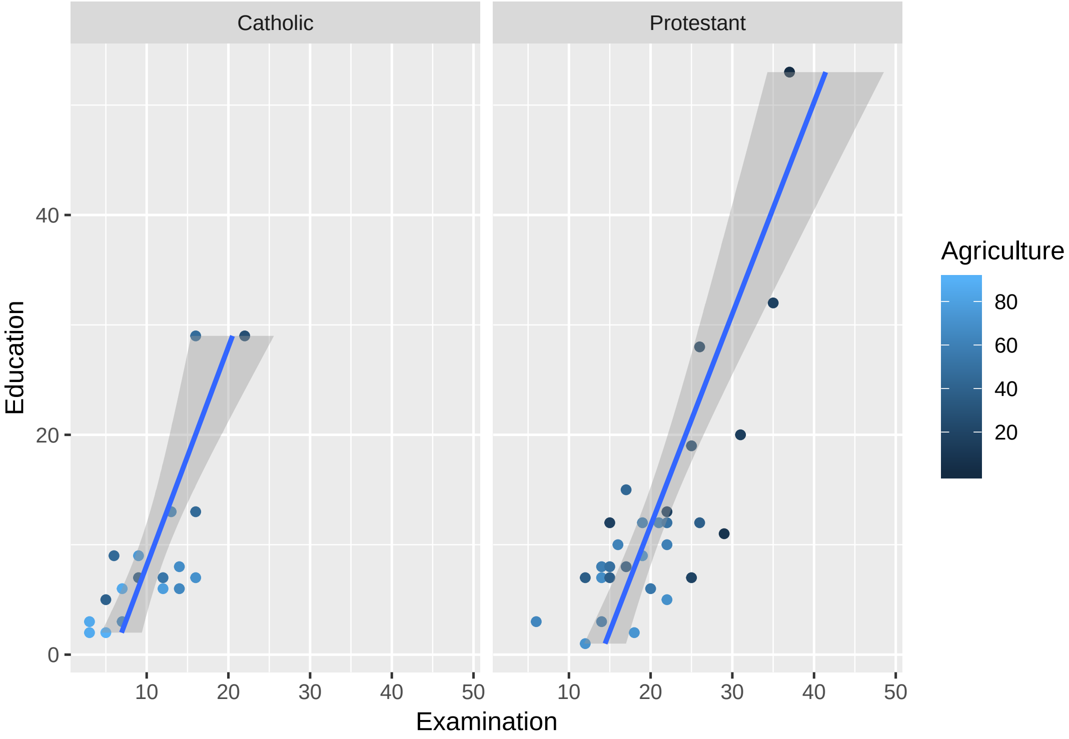


The resulting picture is in line with what we have expected. Overall, the districts with a lower share of occupation in agriculture tend to have rather higher levels of education as well as higher achievements in the examination.

2.4.4 Themes: Fine-tuning the plot

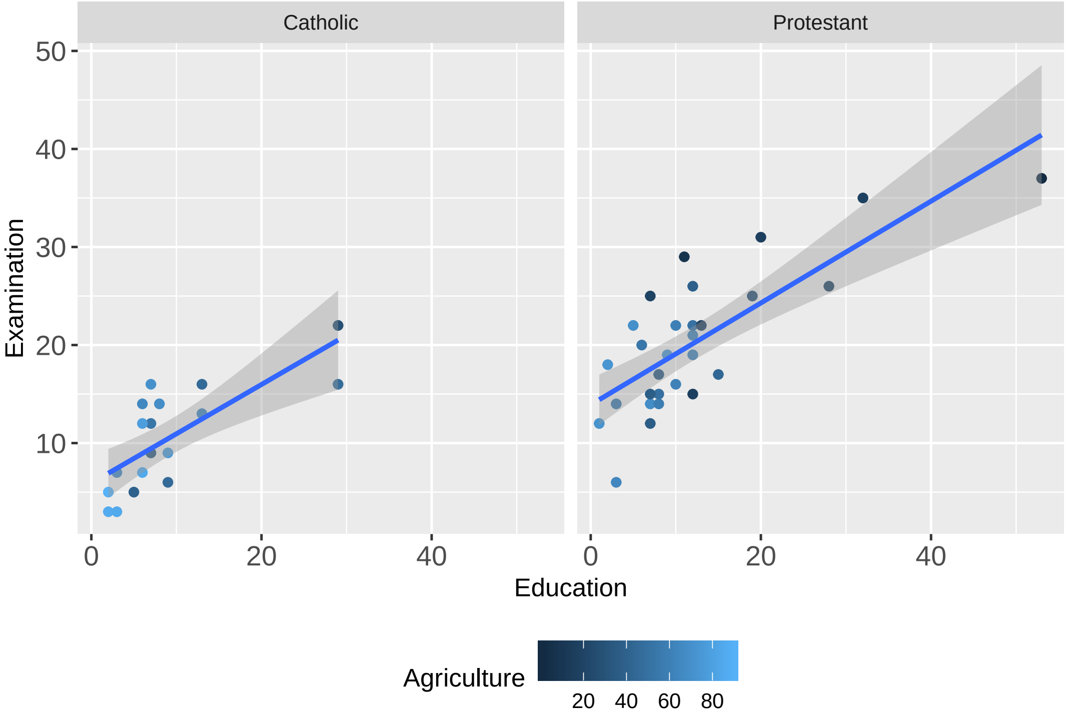
Finally, there are countless options to further refine the plot. For example, we can easily change the orientation/coordinates of the plot:

```
ggplot(data = swiss, aes(x = Education, y = Examination)) +
     geom_point(aes(color = Agriculture)) +
     geom_smooth(method = 'lm') +
     facet_wrap(~Religion) +
     coord_flip()
```



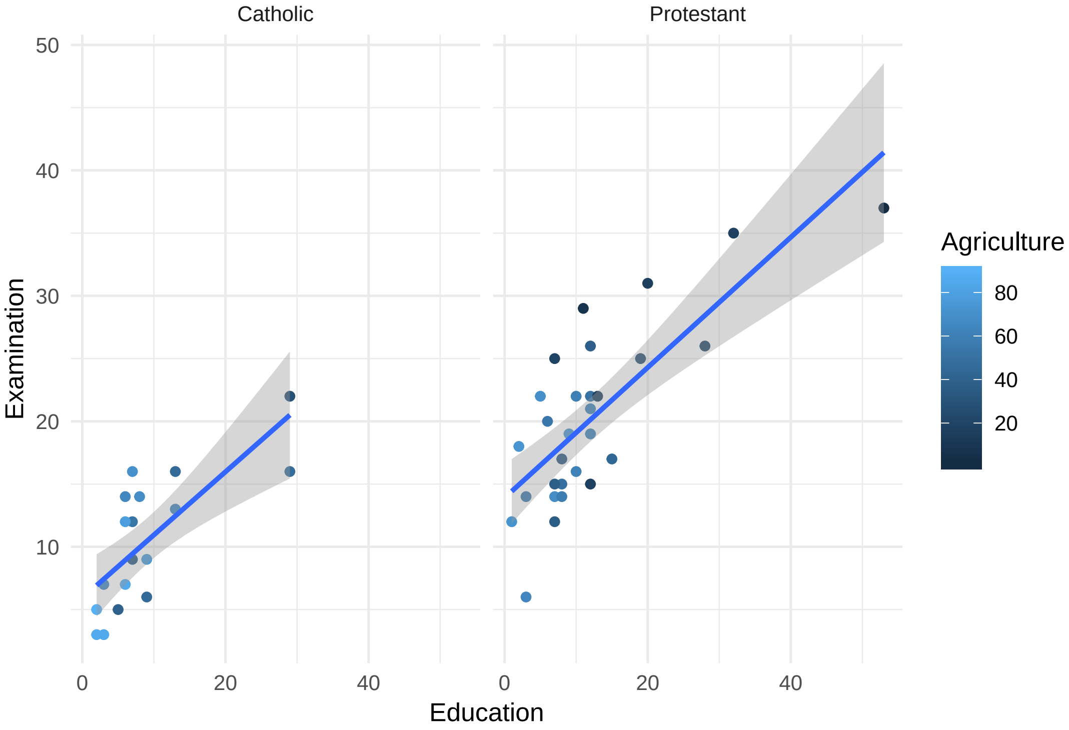
In addition, the theme()-function allows to change almost every aspect of the plot (margins, font face, font size, etc.). For example, we might prefer to have the plot legend at the bottom and have larger axis labels.

```
ggplot(data = swiss, aes(x = Education, y = Examination)) +
     geom_point(aes(color = Agriculture)) +
     geom_smooth(method = 'lm') +
     facet_wrap(~Religion) +
     theme(legend.position = "bottom", axis.text=element_text(size=12) )
```

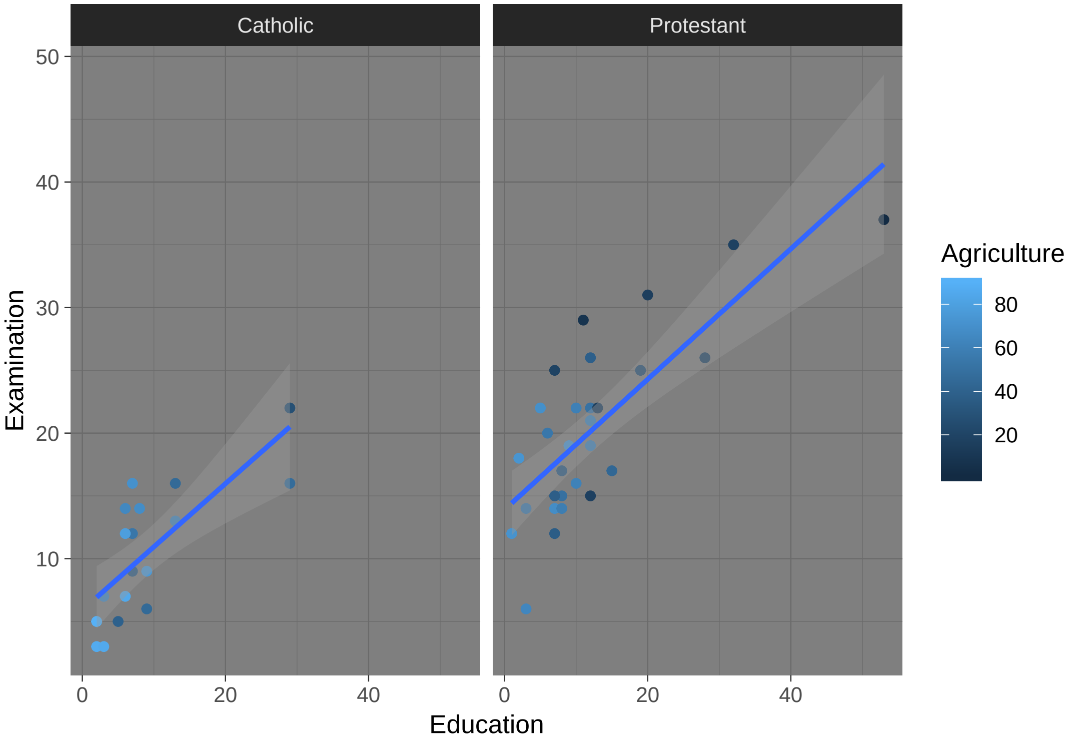


Moreover, several theme-templates offer ready-made designs for plots:

```
ggplot(data = swiss, aes(x = Education, y = Examination)) +
     geom_point(aes(color = Agriculture)) +
     geom_smooth(method = 'lm') +
     facet_wrap(~Religion) +
     theme_minimal()
```



```
ggplot(data = swiss, aes(x = Education, y = Examination)) +
     geom_point(aes(color = Agriculture)) +
     geom_smooth(method = 'lm') +
     facet_wrap(~Religion) +
     theme_dark()
```



3 Dynamic Documents: basic idea (focus on HTML because they already know it)

References

Murrell, Paul. 2009. *Introduction to Data Technologies*. London, UK: CRC Press.