Data Handling: Import, Cleaning and Visualisation

Lecture 9:

Data Analysis and Basic Statistics with R

Prof. Dr. Ulrich Matter

02/12/2021

Updates

Reminder

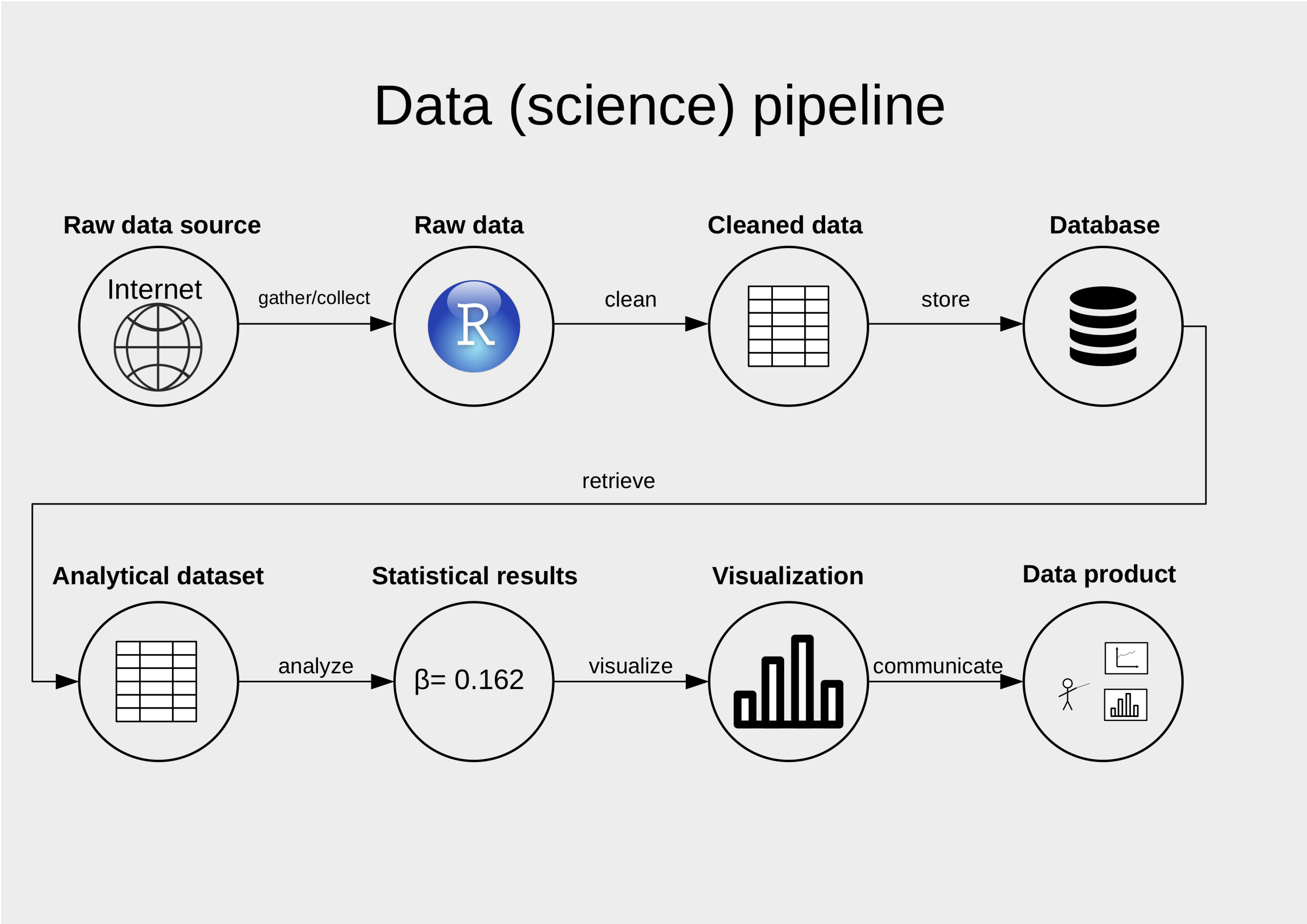
- Send questions for the Q&A session (last lecture)
- **ulrich.matter@unisg.ch**

Reminder: Guest Lecture by Corina Grünenfelder

9 December 2021, “Data Science in Insurance”

Corina Grünenfelder, M.Sc. Mathematics, Actuary SSA
Director, EY

Recap: Data Preparation


Data (science) pipeline
Raw data source
Internet
gather/collect
Raw data
R
clean
Cleaned data
store
Database
retrieve
Analytical dataset
analyze
Statistical results
β= 0.162
visualize
Visualization
communicate
Data product

The dataset is imported, now what?

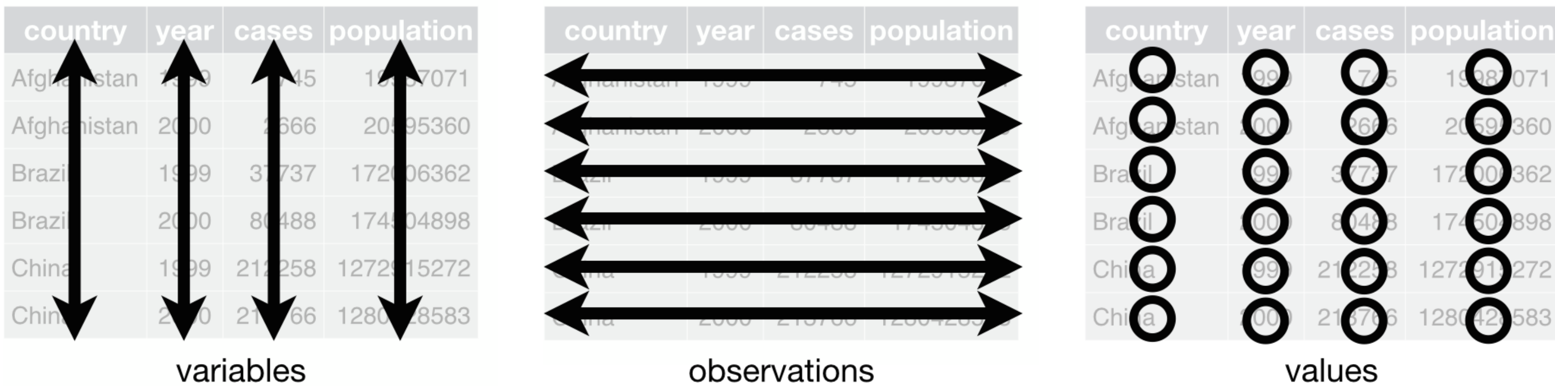
- In practice: still a long way to go.
- Parsable, but messy data: Inconsistencies, data types, missing observations, wide format.
- **Goal** of data preparation: Dataset is ready for analysis.
- **Key conditions**:
  1. Data values are consistent/clean within each variable.
  2. Variables are of proper data types.
  3. Dataset is in ‘tidy’ (in long format)!

Some vocabulary

Following Wickham (2014):

- **Dataset**: Collection of **values** (numbers and strings).
- Every value belongs to a **variable** and an **observation**
- **Variable**: Contains all values that measure the same underlying attribute across units.
- **Observation**: Contains all values measured on the same unit (e.g., a person).

Tidy data



Tidy data. Source: Wickham and Grolemund (2017), licensed under the Creative Commons Attribution-Share Alike 3.0 United States license)

Data Analysis with R

Merging (Joining) datasets

- Combine data of two datasets in one dataset.
  - Why?
- Needed: Unique identifiers for observations (‘keys’).

Merging (joining) datasets: example

```
# load packages
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
## Warning: package 'tibble' was built under R version 3.6.2
```

```
## Warning: package 'tidyr' was built under R version 3.6.2
```

```
## Warning: package 'purrr' was built under R version 3.6.2
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

```
# initiate data frame on persons personal spending
df_c <- data.frame(id = c(1:3,1:3),
                   money_spent= c(1000, 2000, 6000, 1500, 3000, 5500),
                   currency = c("CHF", "CHF", "USD", "EUR", "CHF", "USD"),
                   year=c(2017,2017,2017,2018,2018,2018))
df_c
```

```
##   id money_spent currency year
## 1  1        1000      CHF 2017
```

Merging (joining) datasets: example

```
# initiate data frame on persons' characteristics
df_p <- data.frame(id = 1:4,
                   first_name = c("Anna", "Betty", "Claire", "Diane"),
                   profession = c("Economist", "Data Scientist", "Data Scientist", "E
df_p
```

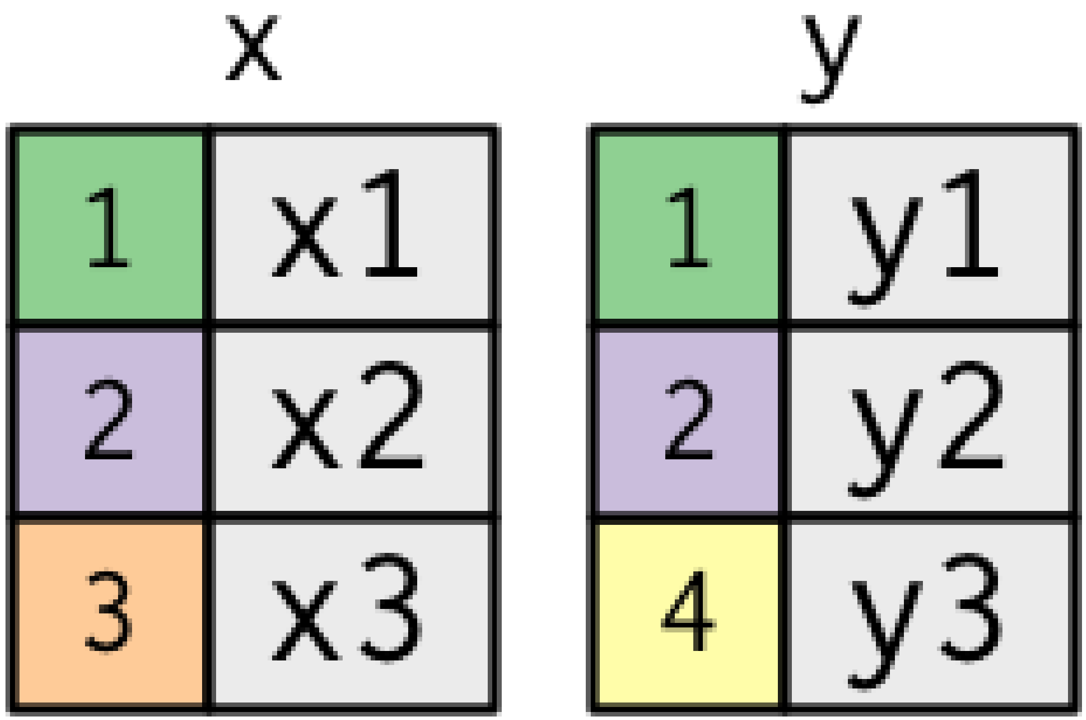
```
##   id first_name     profession
## 1  1       Anna      Economist
## 2  2      Betty Data Scientist
## 3  3     Claire Data Scientist
## 4  4      Diane      Economist
```

Merging (joining) Datasets: Example

```
df_merged <- merge(df_p, df_c, by="id")
df_merged
```

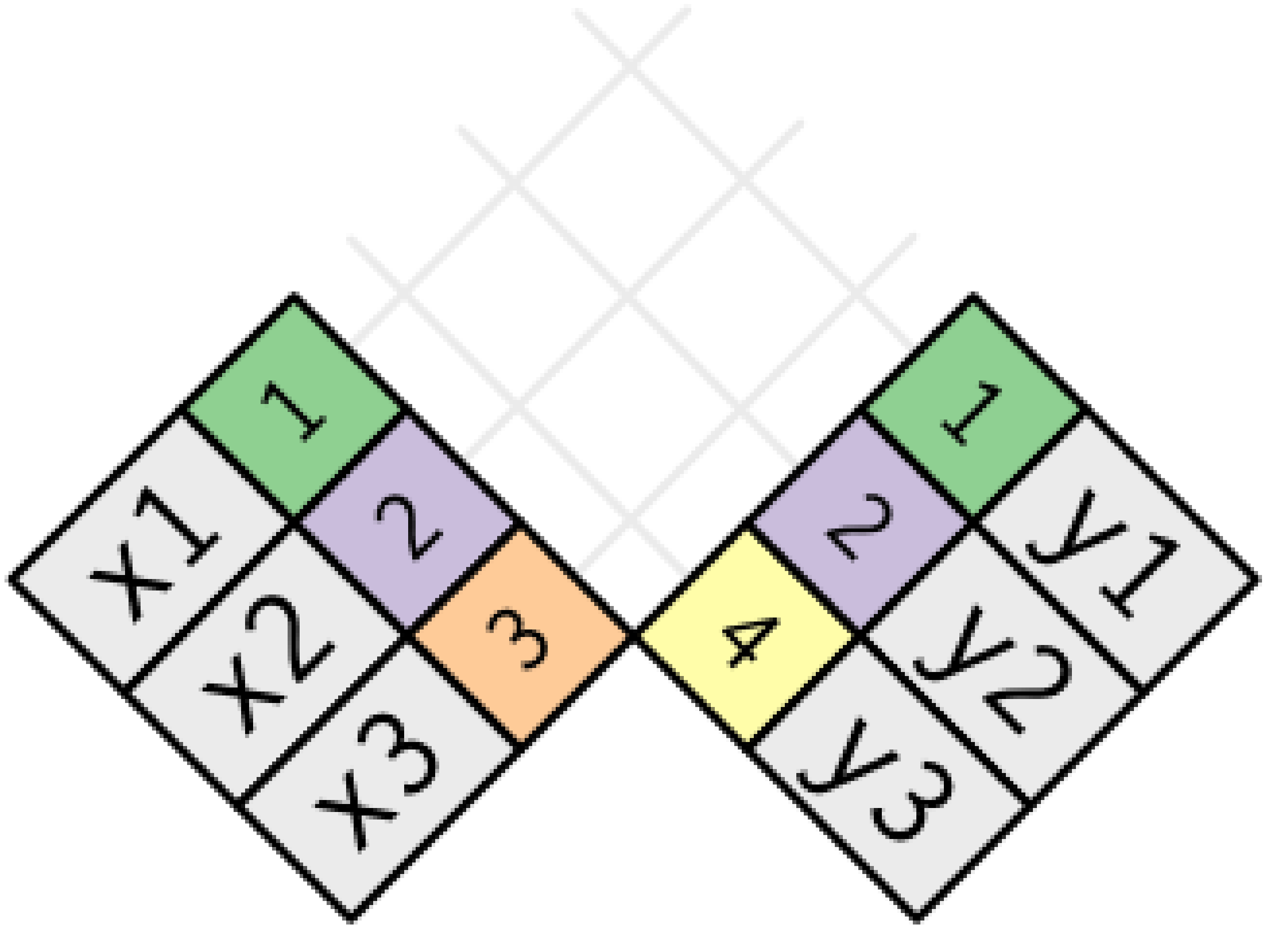
```
##   id first_name     profession money_spent currency year
## 1  1       Anna      Economist        1000      CHF 2017
## 2  1       Anna      Economist        1500      EUR 2018
## 3  2      Betty Data Scientist        2000      CHF 2017
## 4  2      Betty Data Scientist        3000      CHF 2018
## 5  3     Claire Data Scientist        6000      USD 2017
## 6  3     Claire Data Scientist        5500      USD 2018
```

Merging (joining) datasets: concept



Join setup. Source: Wickham and Grolemund (2017), licensed under the Creative Commons Attribution-Share Alike 3.0 United States license.

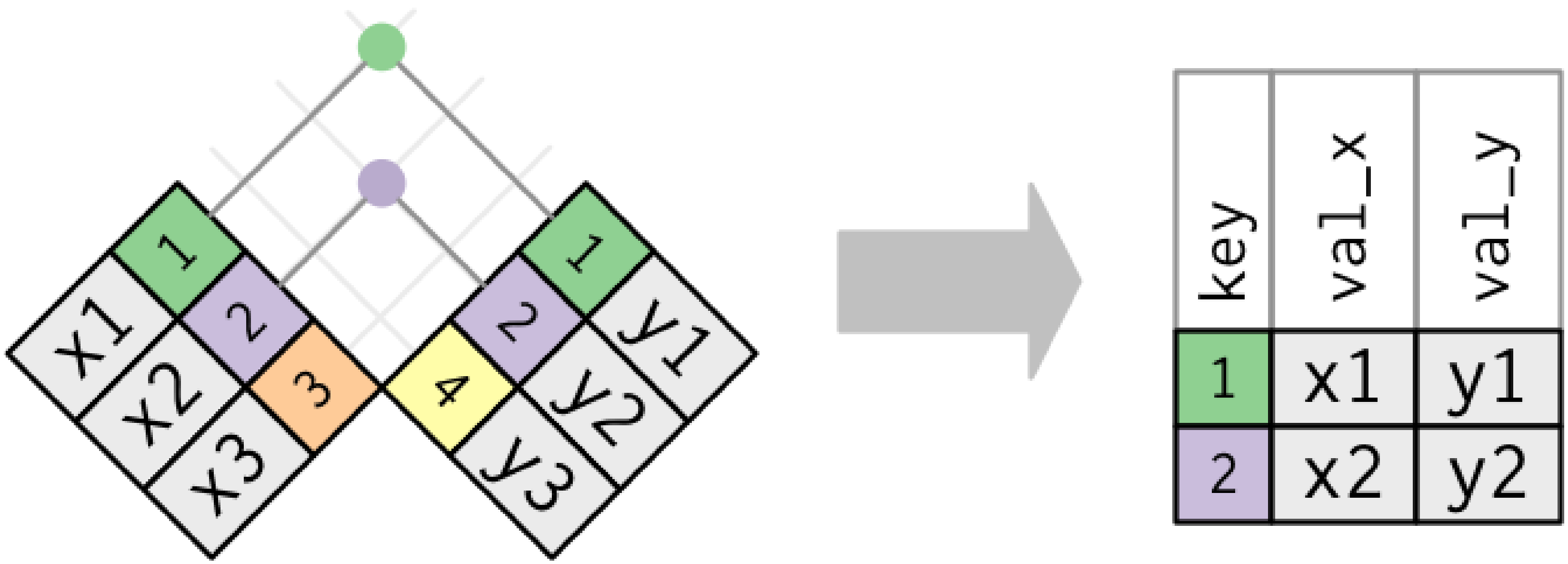
Merging (joining) datasets: concept



Join setup. Source: Wickham and Grolemund (2017), licensed under the [Creative Commons Attribution-Share Alike 3.0 United States] license.

Merging (joining) datasets: concept

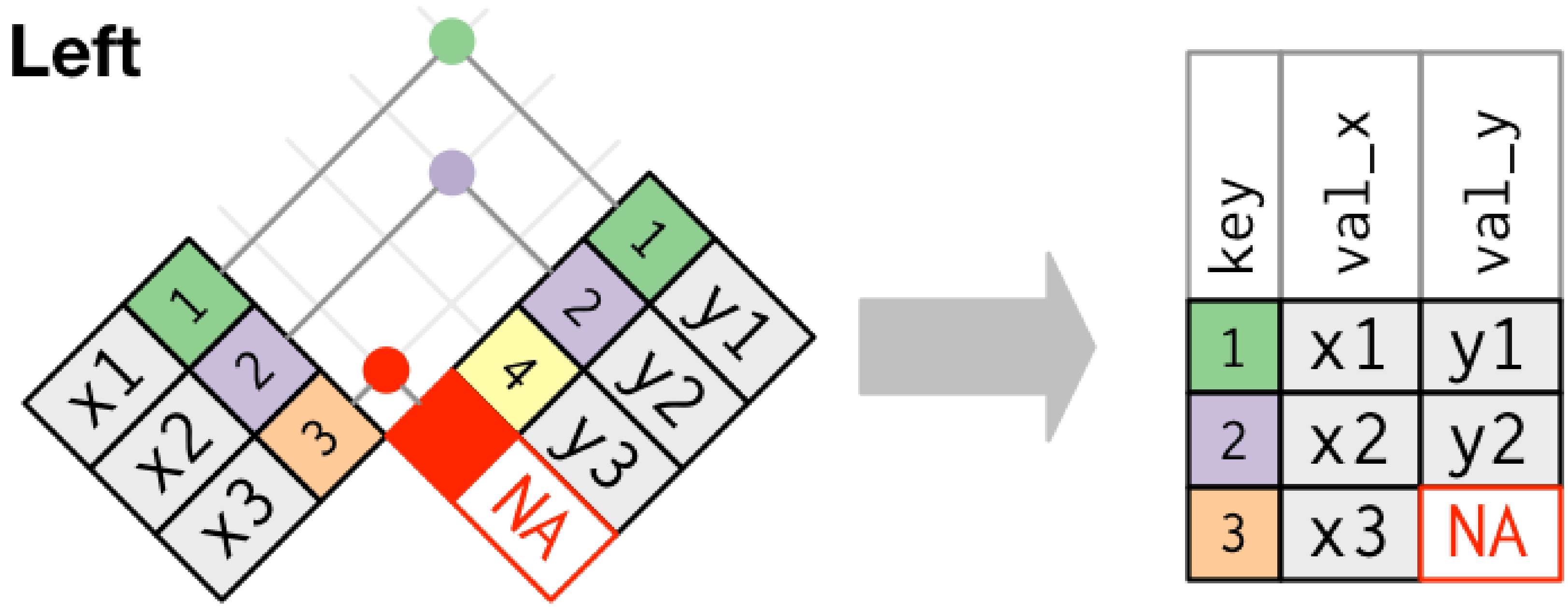
Merge: Inner join



Inner join. Source: Wickham and Grolemund (2017), licensed under the [Creative Commons Attribution-Share Alike 3.0 United States](#) license.

Merging (joining) datasets: concept

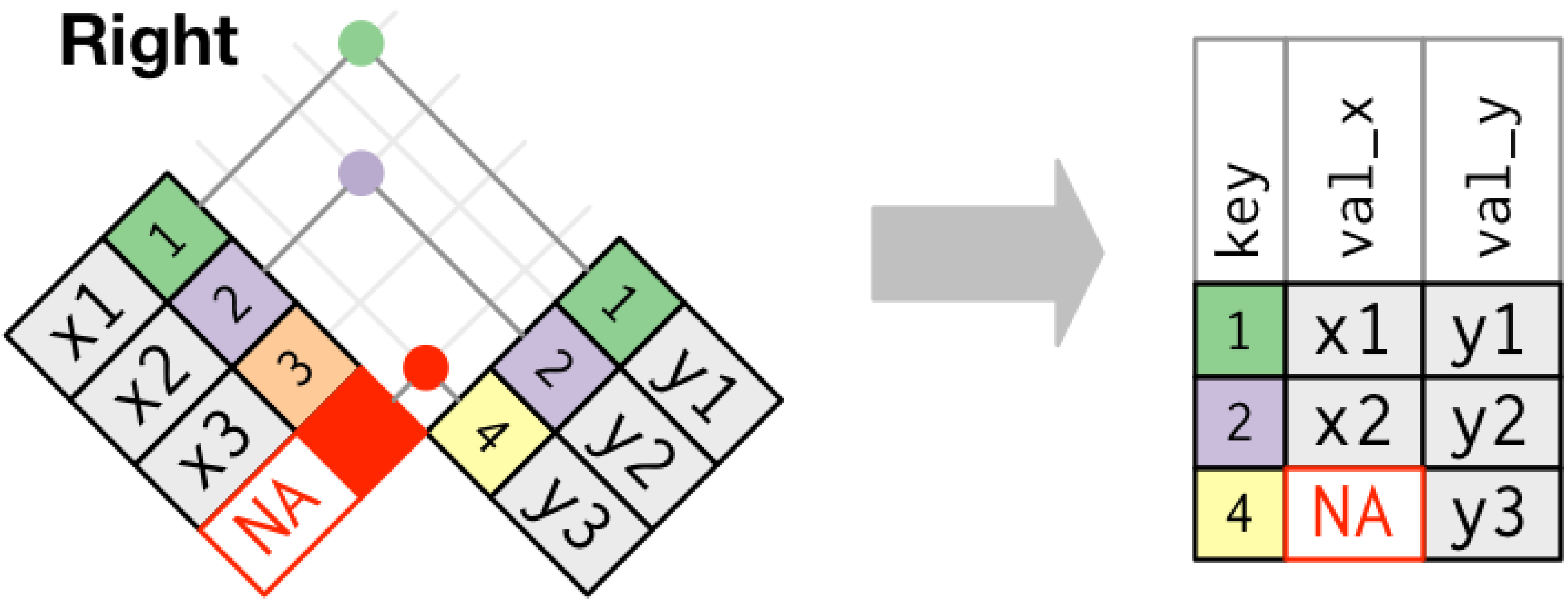
Merge all x: Left join



Outer join. Source: Wickham and Grolemund (2017), licensed under the [Creative Commons Attribution-Share Alike 3.0 United States](#) license.

Merging (joining) datasets: concept

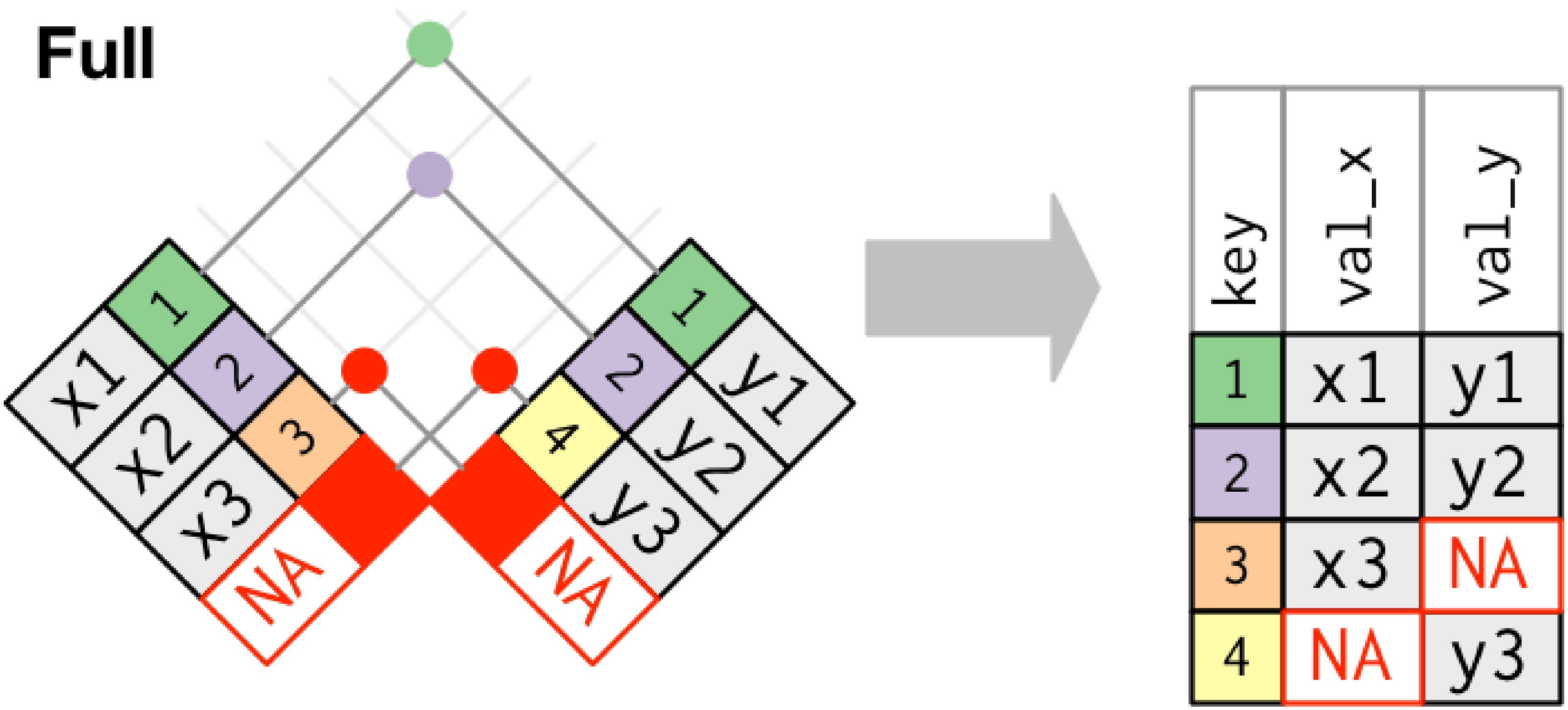
Merge all y: Right join



Outer join. Source: Wickham and Grolemund (2017), licensed under the [Creative Commons Attribution-Share Alike 3.0 United States](#) license.

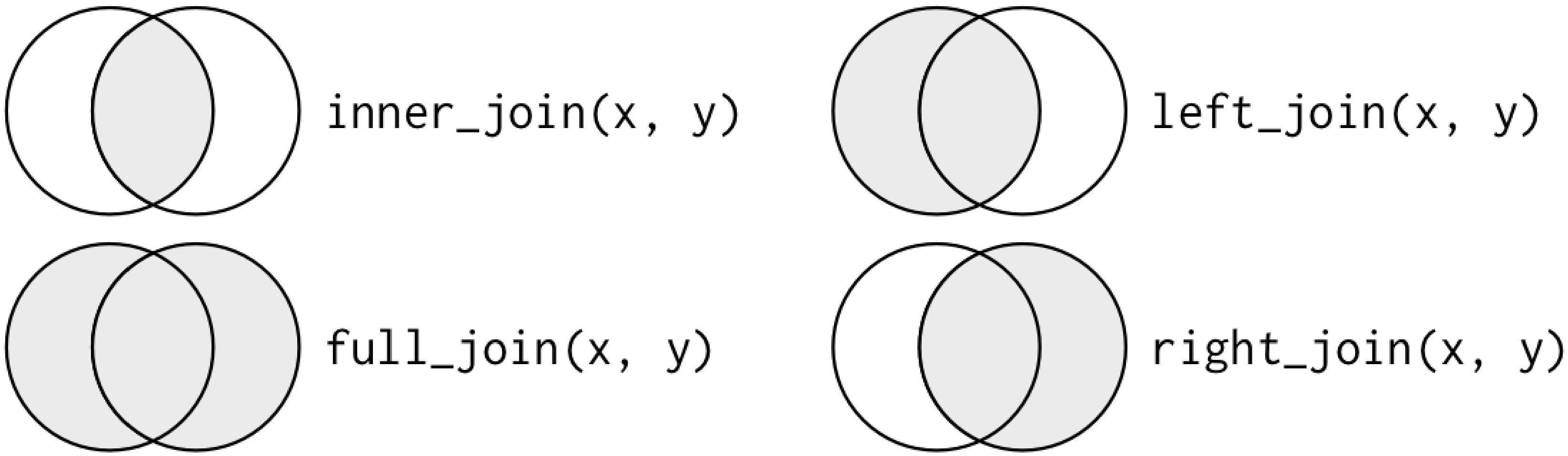
Merging (joining) datasets: concept

Merge all x and all y: Full join



Outer join. Source: Wickham and Grolemund (2017), licensed under the [Creative Commons Attribution-Share Alike 3.0 United States]() license.

Merging (joining) datasets: concept



Join Venn Diagramm. Source: Wickham and Grolemund (2017), licensed under the [Creative Commons Attribution-Share Alike 3.0 United States](#) license.

Merging (joining) datasets: R

Overview by Wickham and Grolemund (2017):

dplyr (tidyverse)	base::merge
`inner_join(x, y)`	`merge(x, y)`
`left_join(x, y)`	`merge(x, y, all.x = TRUE)`
`right_join(x, y)`	`merge(x, y, all.y = TRUE)`,
`full_join(x, y)`	`merge(x, y, all.x = TRUE, all.y = TRUE)`

Data Summaries: Selecting, Filtering, and Mutating

Data summaries

- First step of analysis.
- Get overview over dataset.
- Show key aspects of data.
  - Inform your own statistical analysis.
  - Inform audience (helps understand advanced analytics parts)

Data summaries: first steps

- **Select** subset of variables (e.g., for comparisons).
- **Filter** the dataset (some observations not needed in **this** analysis).
- **Mutate** the dataset: additional values needed

Select, filter, mutate in R (tidyverse)

- `select()`
- `filter()`
- `mutate()`

Data Summaries: Aggregate Statistics

Descriptive/aggregate statistics

- Overview of key characteristics of main variables used in analysis.
- Key characteristics:
    - mean
    - standard deviation
    - No. of observations
    - etc.

Aggregate statistics in R

1. Function to compute statistic (`mean()`).
2. Function to apply the statistics function to one or several columns in a tidy dataset.

- All values.
- By group (observation categories, e.g. by gender)

Aggregate statistics in R

- `summarise()` (in `tidyverse`)
- `group_by()` (in `tidyverse`)
- `sapply()`, `apply()`, `lapply()`, etc. (in `base`)

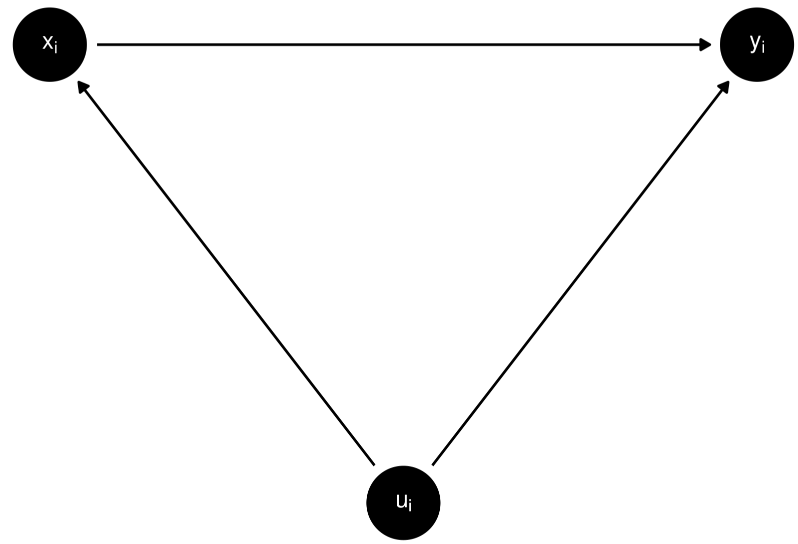
Data Analytics: a primer in linear regression/OLS

Statistical modelling perspective

- **Dependent variable**: y_i.
- **Explanatory variable**: x_i.
- "All the rest": u_i (the **'residuals'** or the 'error term').

$$y_i = \alpha + \beta x_i + u_i\,.$$

Statistical modelling perspective: causality?



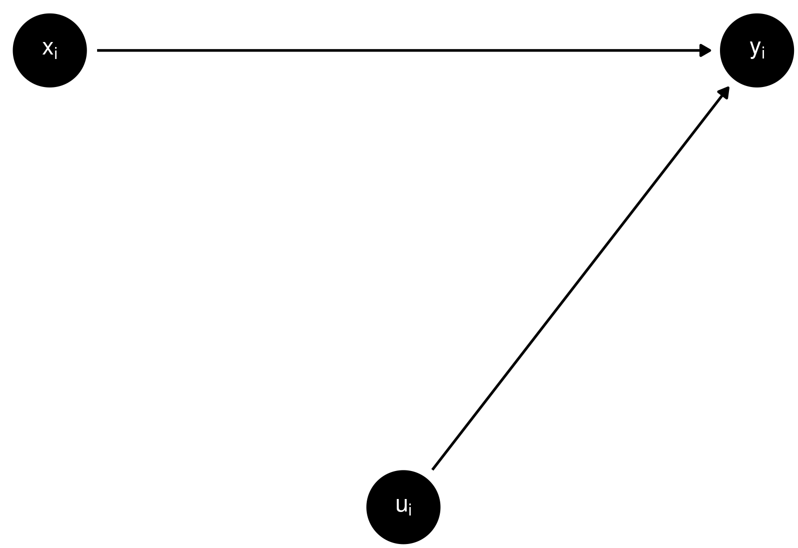


Illustration with pseudo-data

First, we define the key parameters for the simulation. We choose the actual values of α and β, and set the number of observations N.

```
alpha <- 30
beta <- 0.9
N <- 1000
```

Illustration with pseudo-data

Now, we initiate a vector x of length N drawn from the uniform distribution (between 0 and 0.5). This will be our explanatory variable.

```
x <- runif(N, 0, 50)
```

Illustration with pseudo-data

Next, we draw a vector of random errors (residuals) u (from a normal distribution with mean=0 and SD=0.05) and compute the corresponding values of the dependent variable y.

```
# draw the random errors (all the other factors also affecting y)
epsilon <- rnorm(N, sd=10)
# compute the dependent variable values
y <- alpha + beta*x + epsilon
```

Illustration with pseudo-data

```
plot(x,y)
abline(a = alpha, b=beta, col="red")
```

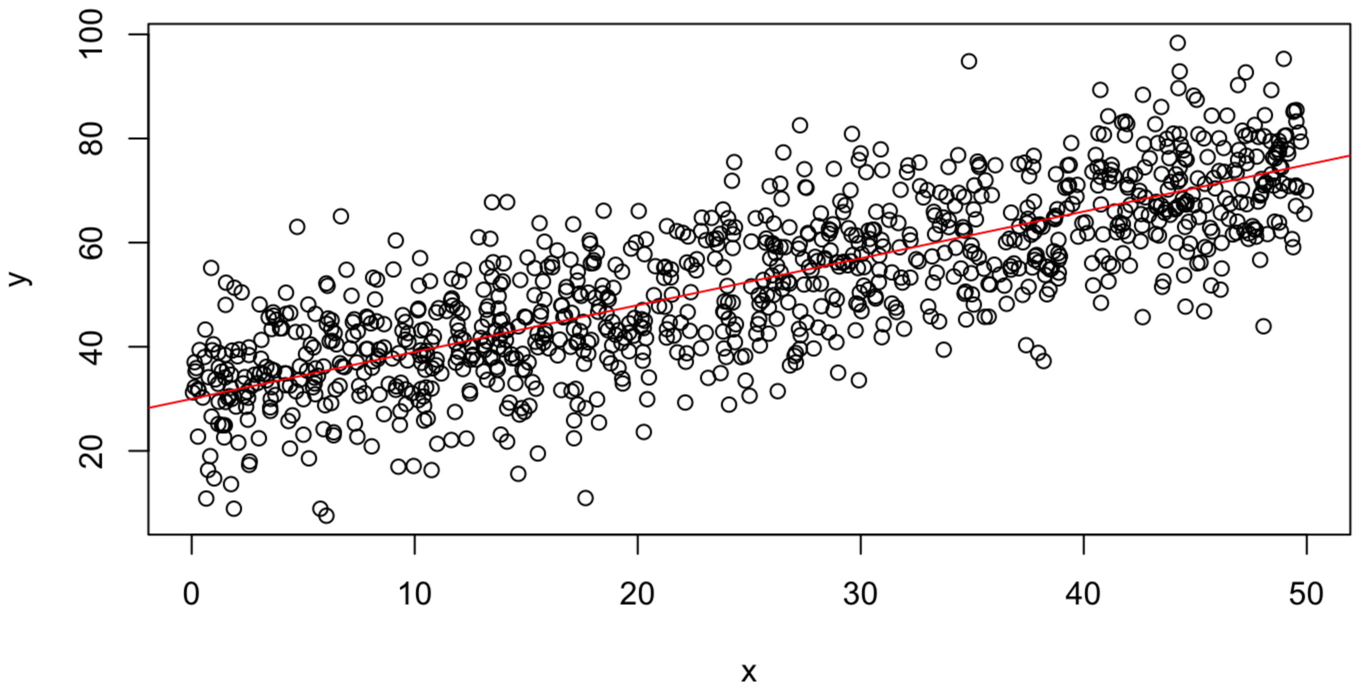


Illustration with pseudo data: “averaging”

```r
# compute average y per x intervals
lower <- c(0,10,20,30,40)
upper <- c(lower[-1], 50)
n_intervals <- length(lower)
y_bars <- list()
length(y_bars) <- n_intervals
x_bars <- list()
length(x_bars) <- n_intervals
for (i in 1:n_intervals){
  y_bars[i] <- mean(y[lower[i] <= x & x < upper[i]])
  x_bars[i] <- mean(x[lower[i] <= x & x < upper[i]])

}
y_bars <- unlist(y_bars)
x_bars <- unlist(x_bars)

# add to plot
plot(x,y)
abline(a = alpha, b=beta, col="red")
points(x_bars, y_bars, col="blue", lwd=10)
```

Parameter estimation: “average out the u”

Clearly, the average values are much closer to the real values. That is, we can ‘average out’ the u in order to get a good estimate for the effect of x on y (to get an estimate of β). With this understanding, we can now formalize how to compute β (or, to be more precise, an estimate of it: $\hat{\beta}$). For simplicity, we take $\alpha = 30$ as given.

In a first step we take the averages of both sides of our initial regression equation:

$$\frac{1}{N}\sum y_i = \frac{1}{N}\sum (30 + \beta x_i + u_i)\ ,$$

rearranging and using the common ‘bar’-notation for means, we get

$$\bar{y} = 30 + \beta\bar{x} + \bar{u}\,,$$

and solving for β and some rearranging then yields

$$\beta = \frac{\bar{y} - 30 - \bar{u}}{\bar{x}}\,.$$

Parameter estimation: “average out the u”

While the elements in $\bar{u}$ are unobservable, we can use the rest to compute an estimate of β:

$\hat{\beta} = \frac{\bar{y} - 30}{\bar{x}}$.

```
(mean(y) -30)/mean(x)
```

```
## [1] 0.8916402
```

Data analytics perspective and estimation: data

```r
# load the data
data(swiss)
# look at the description
?swiss
```

Data analytics perspective and estimation: research question

- Do more years of schooling improve educational outcomes?
- Approximate educational attainment with the variable `Education` and educational outcomes with the variable `Examination`.
- Make use of the simple linear model to investigate whether more schooling improves educational outcomes (on average)?

Model specification

$$Examination_i = \alpha + \beta Education_i \,,$$

- Intuitive hypothesis: β is positive, indicating that a higher share of draftees with more years of schooling results in a higher share of draftees who reach the highest examination mark.
- **Problems**?

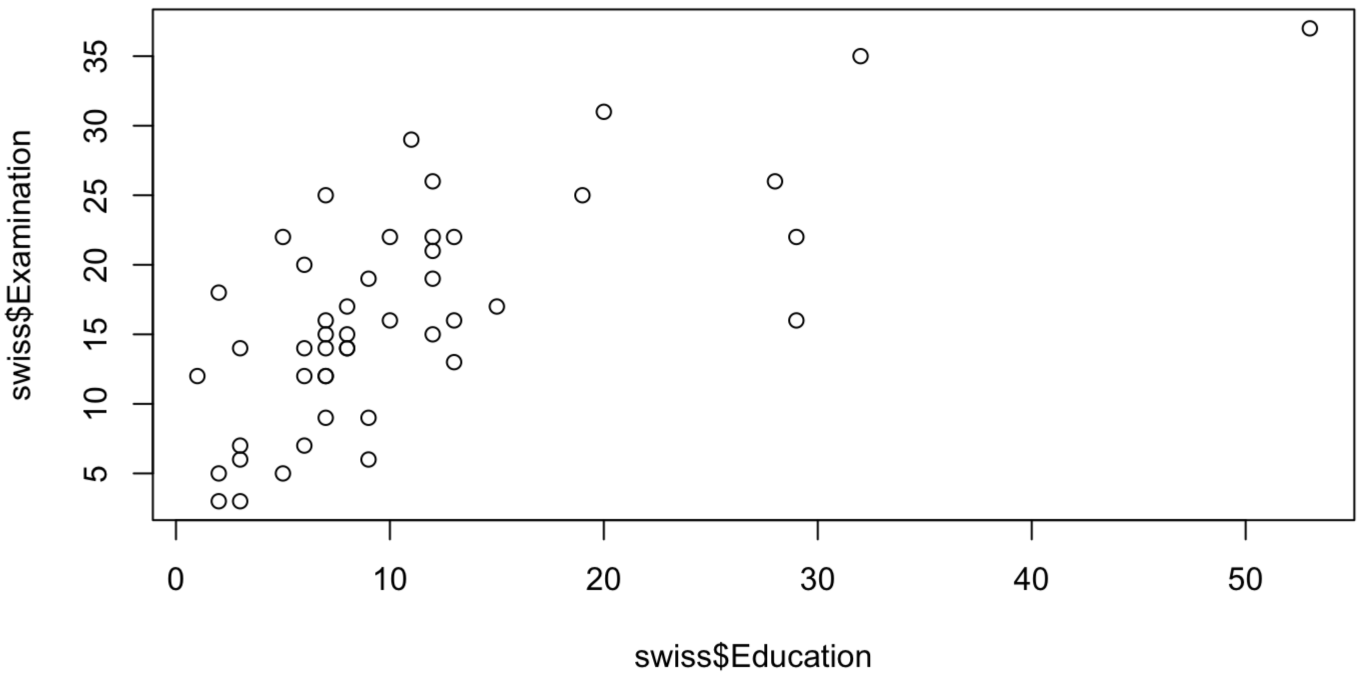
Model specification

To formally acknowledge that other factors might also play a role, we extend our model with the term u_i. For the moment, we thus subsume all other potentially relevant factors in that term:

$$Examination_i = \alpha + \beta Education_i + u_i \,.$$

Raw data

```
plot(swiss$Education, swiss$Examination)
```



Derivation and implementation of OLS estimator

From the model equation we easily see that these 'differences' between the predicted and the actual values of y are the remaining unexplained component u:

$y_i - \hat{\alpha} - \hat{\beta}x_i = u_i$.

Hence, we want to minimize the **sum of squared residuals (SSR)**: $\sum u_i^2 = \sum (y_i - \hat{\alpha} - \hat{\beta}x_i)^2$. Using calculus, we define the two first order conditions:

$$\frac{\partial SSR}{\partial \hat{\alpha}} = \sum -2(y_i - \hat{\alpha} - \hat{\beta}x_i) = 0$$

$$\frac{\partial SSR}{\partial \hat{\beta}} = \sum -2x_i(y_i - \hat{\alpha} - \hat{\beta}x_i) = 0$$

Derivation and implementation of OLS estimator

The first condition is relatively easily solved by getting rid of the -2 and considering that $\sum y_i = N\bar{y}$: $\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$.

Derivation and implementation of OLS estimator

By plugging the solution for $\hat{\alpha}$ into the first order condition regarding $\hat{\beta}$ and again considering that $\sum y_i = N\bar{y}$, we get the solution for the slope coefficient estimator:

$$\frac{\sum x_i y_i - N\bar{y}\bar{x}}{\sum x_i^2 - N\bar{x}^2}.$$

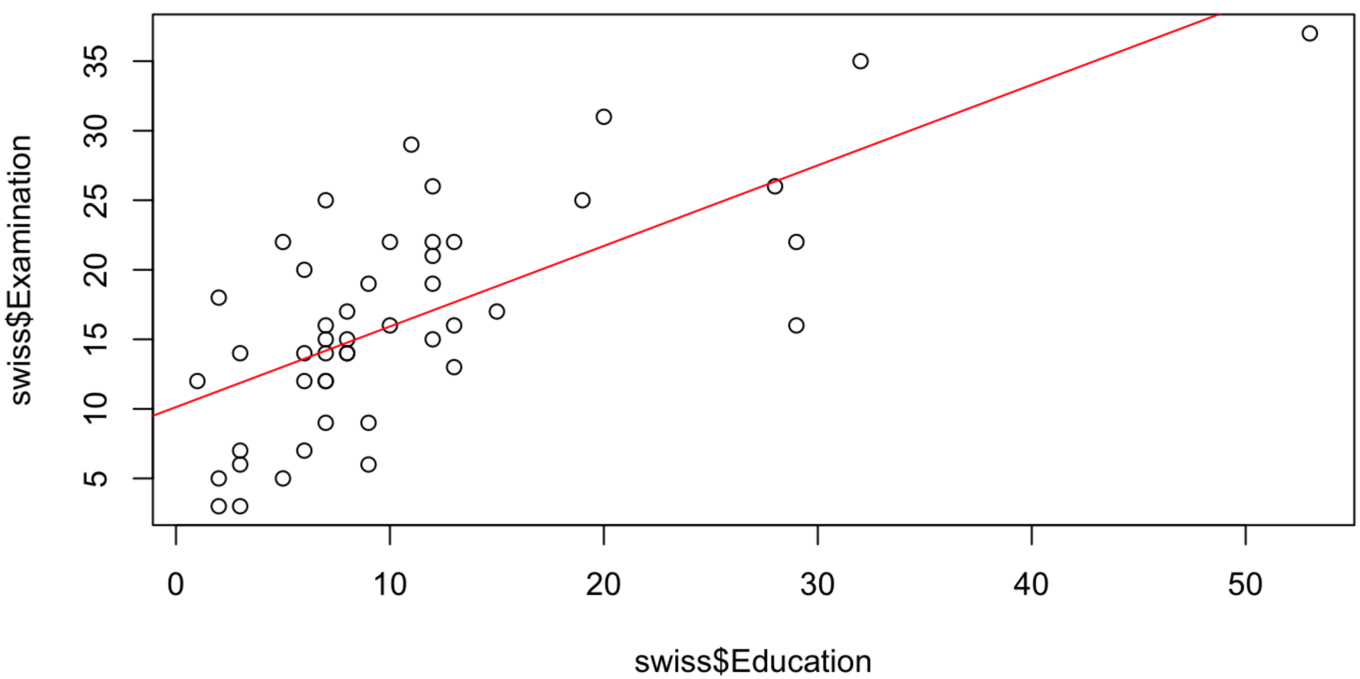
OLS estimator in R!

```
# implement the simple OLS estimator
# verify implementation with simulated data from above
# my_ols(y,x)
# should be very close to alpha=30 and beta=0.9
my_ols <-
  function(y,x) {
    N <- length(y)
    betahat <- (sum(y*x) - N*mean(x)*mean(y)) / (sum(x^2)-N*mean(x)^2)
    alphahat <- mean(y)-betahat*mean(x)

    return(list(alpha=alphahat,beta=betahat))
  }

# estimate effect of Education on Examination
estimates <- my_ols(swiss$Examination, swiss$Education)
estimates
```

```
## $alpha
## [1] 10.12748
##
## $beta
## [1] 0.5794737
```

Simple visualisation

```
plot(swiss$Education, swiss$Examination)
abline(estimates$alpha, estimates$beta, col="red")
```



Regression toolbox in R

```
estimates2 <- lm(Examination~Education, data=swiss)
estimates2
```

```
##
## Call:
## lm(formula = Examination ~ Education, data = swiss)
##
## Coefficients:
## (Intercept)    Education
##     10.1275       0.5795
```

With one additional line of code we can compute all the common statistics about the regression estimation:

```
summary(estimates2)
```

```
##
## Call:
## lm(formula = Examination ~ Education, data = swiss)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9322  -4.7633  -0.1838   3.8907  12.4983
##
## Coefficients:
```

Q&A

References

Wickham, Hadley. 2014. “Tidy Data.” **Journal of Statistical Software** 59 (10): 1–23. https://doi.org/10.18637/jss.v059.i10.

Wickham, Hadley, and Garrett Grolemund. 2017. Sebastopol, CA: O’Reilly. http://r4ds.had.co.nz/.