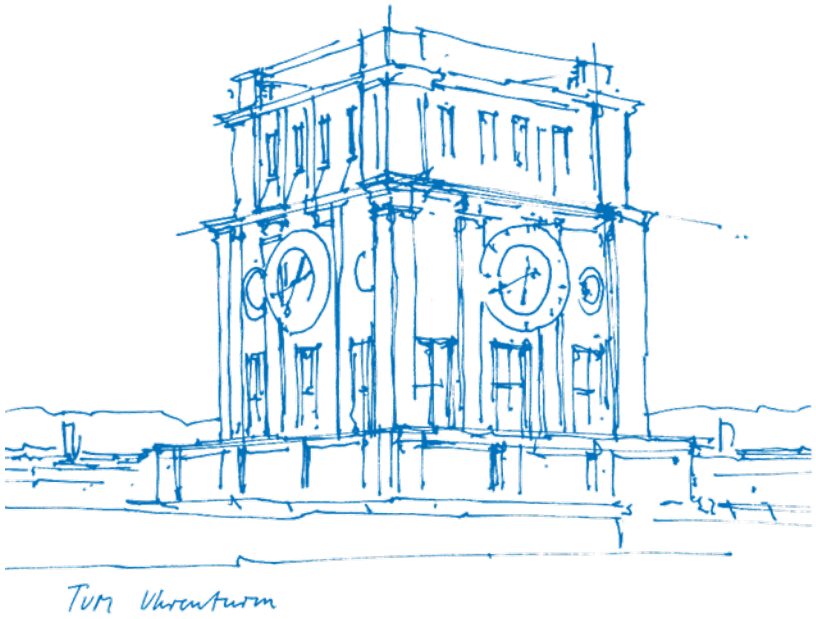
Professorship of International Political Economy
TUM School of Social Sciences and Technology
Technische Universität München

Quantitative Methods – Introduction

Timm Betz and Felix Hagemeister

October 21 and October 22, 2021



Your To Do List from Tuesday:

Your To Do List from Tuesday:

- download and install R Studio (it's free):
  download R Studio here

Your To Do List from Tuesday:

- download and install R Studio (it's free):
  download R Studio here

  need to install R first!
  R for Windows
  R for Mac

Your To Do List from Tuesday:

- download and install R Studio (it's free):
  download R Studio here

  need to install R first!
  R for Windows
  R for Mac

- join Slack, post a message once you installed R:
  click here for Slack

Your To Do List from Tuesday:

- download and install R Studio (it's free):
  download R Studio here

  need to install R first!
  R for Windows
  R for Mac

- join Slack, post a message once you installed R:
  click here for Slack

- read the syllabus

Your To Do List from Tuesday:

- download and install R Studio (it's free):
  download R Studio here

  need to install R first!
  R for Windows
  R for Mac

- join Slack, post a message once you installed R:
  click here for Slack

- read the syllabus

- pick a textbook

Any textbook will do. Good options:

- Imai, Kosuke. 2017. *Quantitative Social Science: An Introduction*. Princeton, NJ: Princeton University Press.
  Amazon link.

- Bailey, Michael A. 2016. *Real Stats: Econometrics for Political Science and Public Policy.* Oxford, UK: Oxford University Press.
  Amazon link.

- Wooldridge, Jeffrey M. 2013. *Introductory Econometrics: A Modern Approach*. South-Western Cengage Learning.
  Amazon link.

R has lots of built-in functions.

R has lots of built-in functions.

Additional `packages`, which in turn contain `libraries` with `functions`, can be downloaded and installed.

You usually install the package once; you need to load the corresponding library in every session in which you want to use it.

One useful package: `swirl`

– short exercises for learning R

One useful package: `swirl`

– short exercises for learning R

```
install.packages("swirl")
```

One useful package: `swirl`

– short exercises for learning R

```
install.packages("swirl")
library(swirl)
```

One useful package: `swirl`

– short exercises for learning R

```
install.packages("swirl")
library(swirl)
swirl()
```

Whenever you have a few minutes, you can pick up some R skills with swirl.

Whenever you have a few minutes, you can pick up some R skills with swirl.

By Tuesday, try out the swirl session
‘R Programming -> Basic Building Blocks.’

You can also install specific swirl courses. For example, the book by Kosuke Imai has swirl sessions for each chapter:

```
library(swirl)
```

You can also install specific swirl courses. For example, the book by Kosuke Imai has swirl sessions for each chapter:

```
library(swirl)
install_course_github(“kosukeimai”, “qss-swirl”)
```

You can also install specific swirl courses. For example, the book by Kosuke Imai has swirl sessions for each chapter:

```
library(swirl)
install_course_github(“kosukeimai”, “qss-swirl”)
swirl()
```

You can use R as a calculator. Try out some simple calculations.

You can use R as a calculator. Try out some simple calculations.

We will often want to refer back to previous information,
which we can store in `objects`.

We can assign a value (a number) or a string (a character) to `objects`.

We can assign a value (a number) or a string (a character) to `objects`.

```
m1 <- 2
```

We can assign a value (a number) or a string (a character) to `objects`.

```
m1 <- 2
m2 <- 4*5
```

We can assign a value (a number) or a string (a character) to `objects`.

```
m1 <- 2
m2 <- 4*5
m3 <- "School of Governance"
```

We can assign a value (a number) or a string (a character) to `objects`.

```
m1 <- 2
m2 <- 4*5
m3 <- "School of Governance"


class(m1)
class(m3)
class(swirl)
```

We can also assign an (ordered!) list of values or strings to `objects` using the `combine` or `concatenate` function `c()`.

We can also assign an (ordered!) list of values or strings to `objects` using the `combine` or `concatenate` function `c()`.

```
m4 <- c(2, 4*5, -2, 0)
```

We can also assign an (ordered!) list of values or strings to `objects` using the `combine` or `concatenate` function `c()`.

```
m4 <- c(2, 4*5, -2, 0)
m5 <- c(“School of Governance”, “HfP”, “TUM”)
```

We can also assign an (ordered!) list of values or strings to `objects` using the `combine` or `concatenate` function `c()`.

```
m4 <- c(2, 4*5, -2, 0)
m5 <- c(“School of Governance”, “HfP”, “TUM”)
m6 <- c(m1, m2)
```

We can also assign an (ordered!) list of values or strings to `objects` using the `combine` or `concatenate` function `c()`.

```
m4 <- c(2, 4*5, -2, 0)
m5 <- c(“School of Governance”, “HfP”, “TUM”)
m6 <- c(m1, m2)
m7 <- c(“m1”, “m2”)
```

We can look at the content of an object by typing its name.

We can look at the content of an object by typing its name.

```
m4
```

We can look at the content of an object by typing its name.

```
m4
m4[1]
```

`functions` use parentheses for their `arguments`;
brackets indicate positions within `objects`

To Do:

By Tuesday, try out the swirl session
‘R Programming -> Basic Building Blocks.’