# Data Science for Business Analytics

*Introduction*

HEC Lausanne
Aleksandr Shemendyuk [aleksandr.shemendyuk@unil.ch]

# Agenda

# A little about me

- Born and raised in Riga, Latvia
- **Education**:
  - ▶ B.Sc. in Applied Mathematics 2017 (HSE University, Moscow)
  - ▶ M.Sc. in Statistical Modelling 2019 (HSE University, Moscow)
  - ▶ Ph.D. in Actuarial Science 2024 (HEC Lausanne, UNIL)
- **Experience**:
  - ▶ Teaching assistant at HSE University and HEC Lausanne
  - ▶ Research using real data
  - ▶ Telegram Bot development in Python
  - ▶ AI in research and personal projects
  - ▶ AI automation
- **Hobbies**:
  - ▶ Chess
  - ▶ Guitar
  - ▶ Finance

# What you will learn

- Workflow of the 'modern' Data Scientist



Program
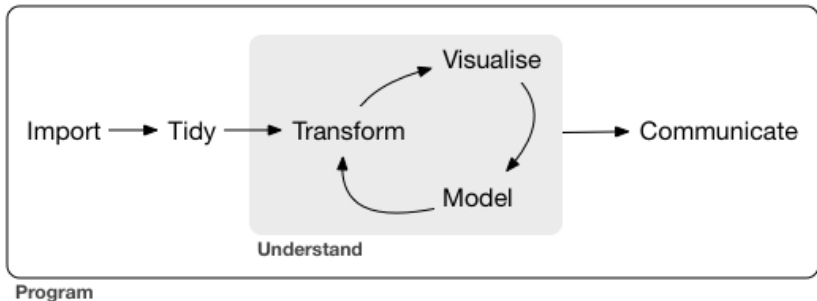
## Statistical computing & data science

- What's the difference between data science and statistics?

  *"A data scientist is just a sexier word for a statistician."*
  — *Nate Silver (outdated)*

  *"A data scientist is a better computer scientist than a statistician and is a better statistician than a computer scientist."*
  — *Unknown (still accurate)*

- What does a data scientist do?
  - ▶ *Jack of all trades.*
  - ▶ Transform data into valuable information!
  - ▶ A data scientist spends a significant portion of time processing data and less time modelling data.

# What is Data Science?

- **Wikipedia:** "the extraction of knowledge from data"
- The precise definition is a bit unclear and controversial...
- Practitioners "agree" on the components of data science:
  - ▶ database management
  - ▶ gathering and cleaning
  - ▶ exploratory analysis
  - ▶ predictive modelling
  - ▶ data summary and visualization
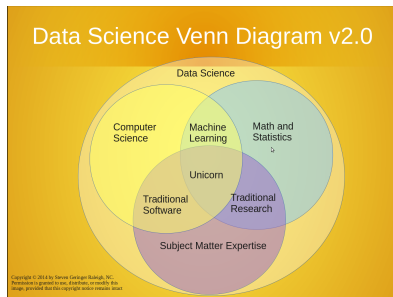
# Applications



Figure 1: Some of the hiring partners of *The Data Incubator*

- E-marketing
- Recommender systems
- Sport analytics
- Biotechnology
- Image or speech recognition
- Fraud and risk detection
- Social media

- redbfit scoring
- E-commerce
- Government analysis
- Gaming
- Price comparisons
- Airline routes planning
- Delivery logistics

# Agenda

# Course Description

- **Lectures** (10:15-12:00):

    ▶ Focus on introducing the concepts.
    ▶ Details are sometimes left for you to read up on
- **Exercise sessions** (12:30-14:00):

    ▶ Work on assignments and project.
- **Instructor**: Professor Alex (Aleksandr Shemendyuk).
- **Captains** (Teaching assistants):

    ▶ Ilia Azizi
    ▶ Léo Wenger
    ▶ Elwin Freudiger

# Grading

- The **Final grade** is composed based on two equally weighted components:
  - ▶ Online Exam
  - ▶ Group Project
- **Exam**:
  - ▶ Online closed/**open**-book exam (50%)
- **Group Project**:
  - ▶ Project proposal (5%)
  - ▶ Project update (5%)
  - ▶ Video presentation (15%)
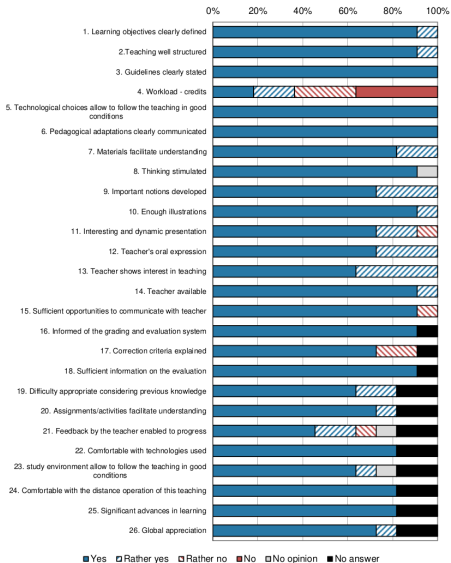  - ▶ Final report (25%)

For the group project:

- Groups of 3 (or 4) members
- More details to come...

## Milestones

| Date | Assignment |
| --- | --- |
| October 14 | Project proposal |
| November 11 | Project update |
| December 16 | Video presentation |
| December 23 | Final Report |

# Course Evaluations in Fall 2022

- **Workload** - **credits**: ~33% of students found it too high!

# Schedule outline

| Date | Topic | Reading |
|------|-------|---------|
| September 23 | Introduction | |
| September 30 | Data Structures and Subsetting | ADVR 3+4 |
| October 7 | Control Flows and Functions | ADVR 5, 6, 9, 11 |
| October 14 | Data Wrangling | R4DS 4, 6, 8, 13-20 |
| October 21 | Visualisation | R4DS 2, 10-12 |
| October 28 | R Projects and Quarto | R4DS 3, 5, 7, 9, 29, 30 |
| November 4 | Project Coaching | |
| November 11 | Data Wrangling | R4DS 4, 6, 8, 13-20 |
| November 18 | Visualisation | R4DS 2, 10-12 |
| November 25 | Project Coaching | |
| December 2 | Presentations / Dashboards / Interactivity | R4DS 29+30, htmlwidgets |
| December 9 | Projects Coaching | |
| December 16 | Projects Presentations | |

(numbers in the third column are book chapters)

# Course website

**Data Science in Business Analytics 2024-2025**

- Course website:

  ▶ **https://moodle.unil.ch/course/view.php?id=32207**

  ▶ Registration key: **dsiba2024**

  ▶ **A discussion forum**

  ▶ PDFs of the lectures

  ▶ Additional resources

# Additional resources
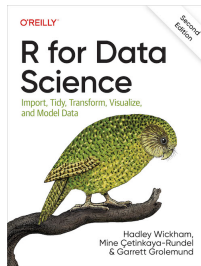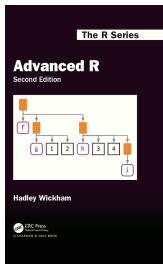




- Books:
  - ▶ Advanced R
  - ▶ R for data science
- Additionally:
  - ▶ Rstudio cheat sheets
  - ▶ The CRAN website

... **and our forum on Moodle.**

# Agenda

# S and R

- S

  - ▶ A statistical programming language
  - ▶ First appeared in 1976
  - ▶ Developed by John Chambers and (in earlier versions) Rick Becker and Allan Wilks of Bell Labs
  - ▶ John Chambers, *[the aim is] to turn ideas into software, quickly and faithfully*

- R

  - ▶ Modern implementation of S
  - ▶ First appeared in 1993
  - ▶ Created by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand
  - ▶ Currently developed by the *R Development Core Team*

## Some "technical" details about R
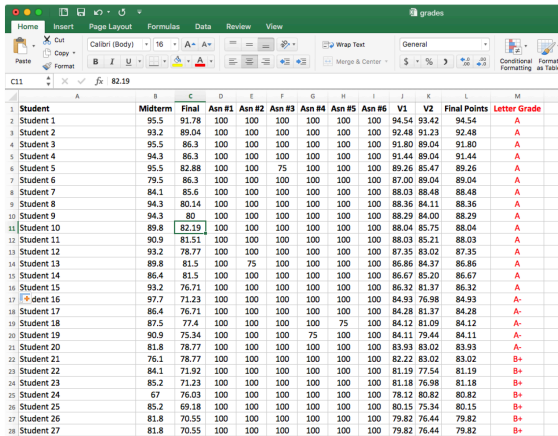
- Technical features:
  - ▶ **Available for Windows, macOS, and Linux**
  - ▶ Source code written primarily in C, Fortran, and R
  - ▶ Multi-paradigm: object-oriented, functional, procedural
  - ▶ Dynamically typed
  - ▶ Scripting language (interpreted)
  - ▶ **Wide variety of statistical and graphical techniques**
  - ▶ **Easily extensible through functions and packages**
  - ▶ **Read/write from/to various data sources**

# What about Excel?



source: fantasyfootballanalytics.net

# Excel is great for certain things…



source: github.com/jdwilson4

# ...but not everything

### R's advantages:

- **Easier automation**
- **Better reproducibility**
- Faster computation
- Supports larger data sets
- Reads any type of data
- More powerful data manipulation capabilities
- Easier project organisation

- Easier to find and fix errors
- Free & open source
- Advanced statistics capabilities
- State-of-the-art graphics
- Runs on many platforms
- Anyone can contribute packages to improve its functionality

# Automation and reproducibility

**Geeks and repetitive tasks**
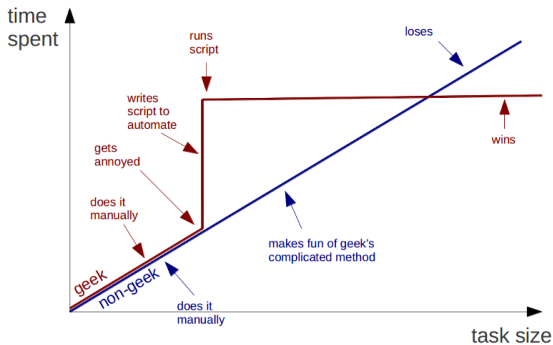


source: trendct.org

# CRAN

The Comprehensive R Archive Network

**Download and Install R**

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

**Source Code for all Platforms**

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!
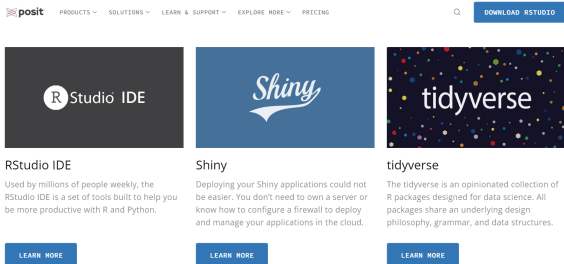
- The latest release (2017-11-30, Kite-Eating Tree) R-3.4.3.tar.gz, read what's new in the latest version.

- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).

- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.

- Source code of older versions of R is available here.

- Contributed extension packages

**Questions About R**

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

**source: cran.r-project.org**

# RStudio

- An open-source integrated development environment (IDE)
- RStudio Desktop available for Windows, macOS, and Linux



**source: posit.co**

## Base R

- What is `Base R`?

  *"The package named 'base' is in a way the core of R and contains the basic functions of the language, particularly, for reading and manipulating data."*

  — *R for Beginners, Emmanuel Paradis*

- `Base R` includes all default code for performing common data manipulation and statistical tasks.
- You might recognize some `Base R` functions:
  - ▶ `mean()`, `median()`, `lm()`, `summary()`, `sort()`
  - ▶ `data.frame()`, `read.csv()`, `cbind()`, `grep()`, `regexpr()`
  - ▶ Many many more…
- If you don't recognize any `Base R` functions, don't worry!

# The tidyverse

- Common criticisms of `Base R`:

  ▶ The code doesn't flow as well as other languages.
  ▶ Function names/arguments are often inconsistent/confusing.
  ▶ `Base R` functions sometimes don't return type-stable objects.
  ▶ `Base R` functions are not refined to run as fast as possible.
  ▶ Other complaints exist…

- So what is the **tidyverse**? A collection of R packages

  ▶ designed for data science,
  ▶ sharing an underlying design philosophy, grammar, and data structures.

- Often, it performs the same tasks as `Base R`, but:

  ▶ Relies on a **pipe** operator to help with the flow of the code.
  ▶ More descriptive function names and consistent inputs.
  ▶ Type-stable.
  ▶ Often faster than common `Base R` functions.

# Core `tidyverse` packages

- `ggplot2`: declarative graphics, based on The Grammar of Graphics.
- `dplyr`: grammar of data manipulation.
- `tidyr`: functions that help you get to tidy data.
- `readr`: reading in rectangular data.
- `purrr`: enhancing R's functional programming (FP).
- `tibble`: a modern `data.frame`.
- `stringr`: makes working with strings as easy as possible.
- `forcats`: useful tools for working with factors.

See more on the tidyverse website.

# Base R vs tidyverse

- Why ever use Base R?
  - ▶ Gets the job done!
  - ▶ To become an expert, you have to know Base R.
  - ▶ Some Base R functions are very common/useful, e.g., mean().
- What should you learn first? Base R or tidyverse?
  - ▶ Some believe you should learn Base R first, others the tidyverse first.
  - ▶ Lately, more are shifting to tidyverse…

# Install links

- R:
  - ▶ MacOS
    - ▶ Be mindful of the difference between Apple silicon (M1/M2) Macs and older Intel Macs
  - ▶ Windows
- RStudio Desktop:
  - ▶ MacOS 12+
  - ▶ Windows 10/11