Columbia University
in the City of New York

Data Science for Business Analytics

Lecture 1

Thibault Vatter
Department of Statistics, Columbia University

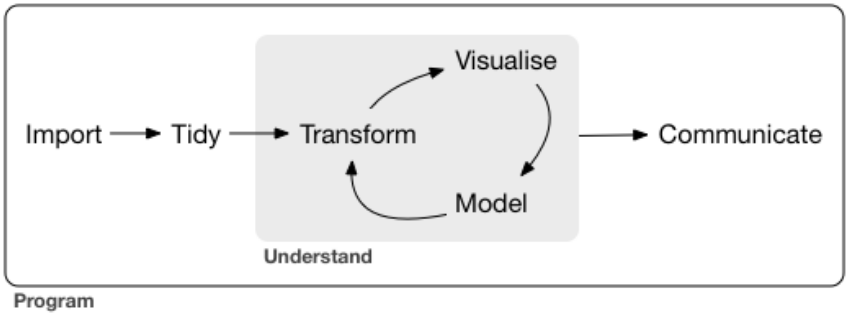
02/17/2020

Outline

A little about me

- Born and raised in Geneva
- Education:
  - B.Sc. Physics (EPFL, '10)
  - M.Sc. Physics with minor in Financial Engineering (EPFL, '12)
  - Ph.D. Statistics (HEC Lausanne, '16)
- Worked a bit as a quant in finance
- Currently:
  - Assistant Professor in Statistics at Columbia University
  - Live in New York city
- Hobbies:
  - Flying planes
  - Watching bay area teams (go 49ers and Warriors!)
  - Running
  - Beers (formerly at Satellite, now in Brooklyn micro-breweries)

What you will learn



- **Import** data from the web, a database, a stored file, etc.
- **Wrangle**:
  - **Tidy**: usually means that rows/columns are observations/variables.
  - **Transform**: narrowing in on observations of interests, creating new variables, calculating summary statistics.
- **Analyze**:
  - **Visualize**:
    - E.g., show unexpected things, or raise new questions.
    - Doesn't scale well as it requires human interpretation.
  - **Model**:
    - Sufficiently precise questions can be answered with a model.
    - Mathematical/computational tools generally scale well.
    - Even when it doesn't, computers are usually cheaper than brains!
- **Communicate** your results.
- Surrounding all these tools is **programming**.

Statistical computing & data science

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

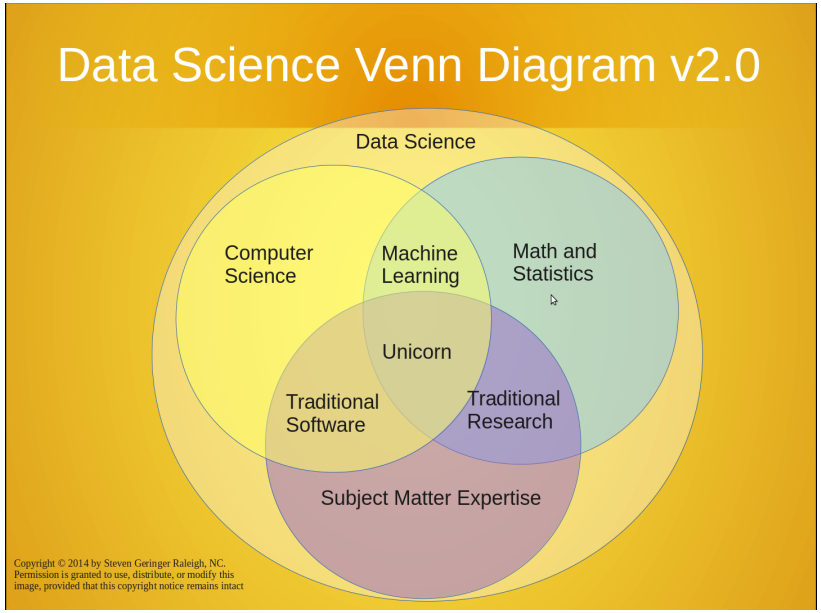
  "A data scientist is just a sexier word for statistician."
  — Nate Silver (outdated)

  "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?
  - There is not one correct answer.
  - Transform data into valuable information!
  - A data scientist spends a significant portion of time processing data and less time modeling data.

What is Data Science?

- **Wikipedia:** "the extraction of knowledge from data"
- Precise definition a bit unclear and controversed...
- Practitioners "agree" on the components of data science:
  - database management
  - gathering and cleaning
  - exploratory analysis
  - predictive modeling
  - data summary and visualization

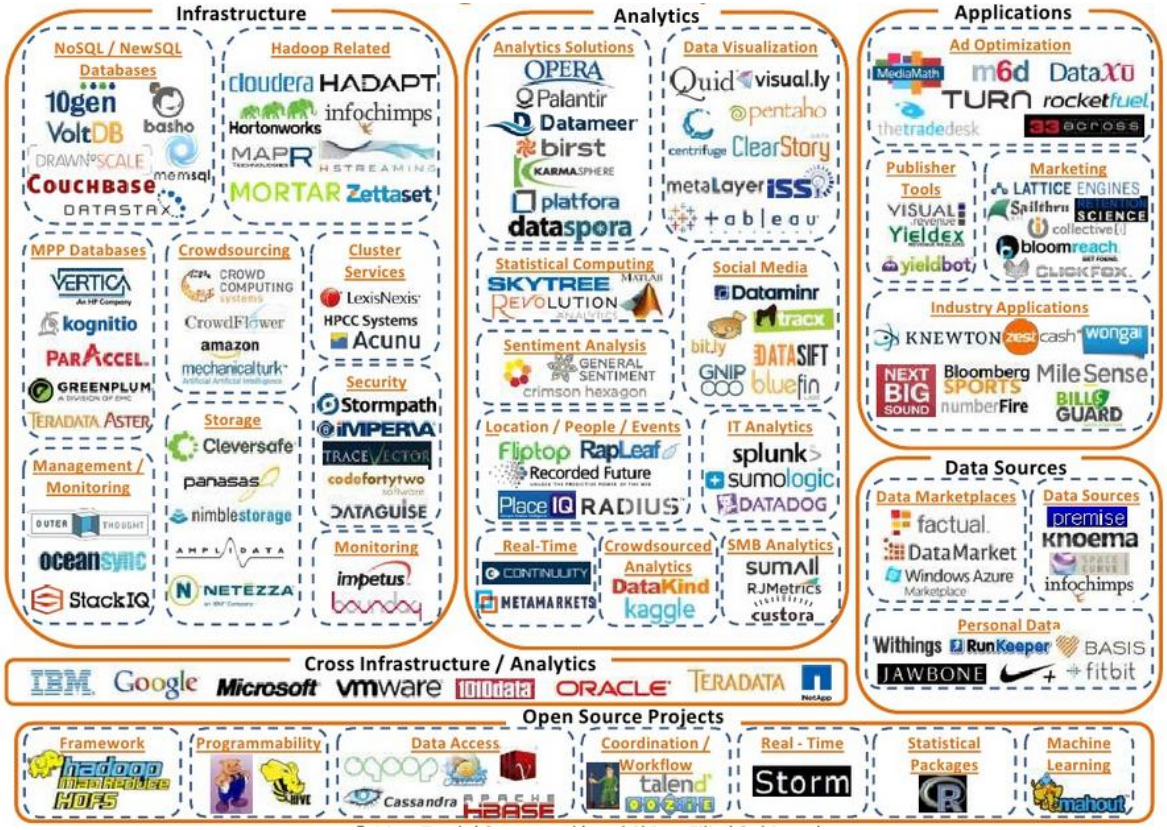


Applications

Some of the hiring partners of *The Data Incubator*

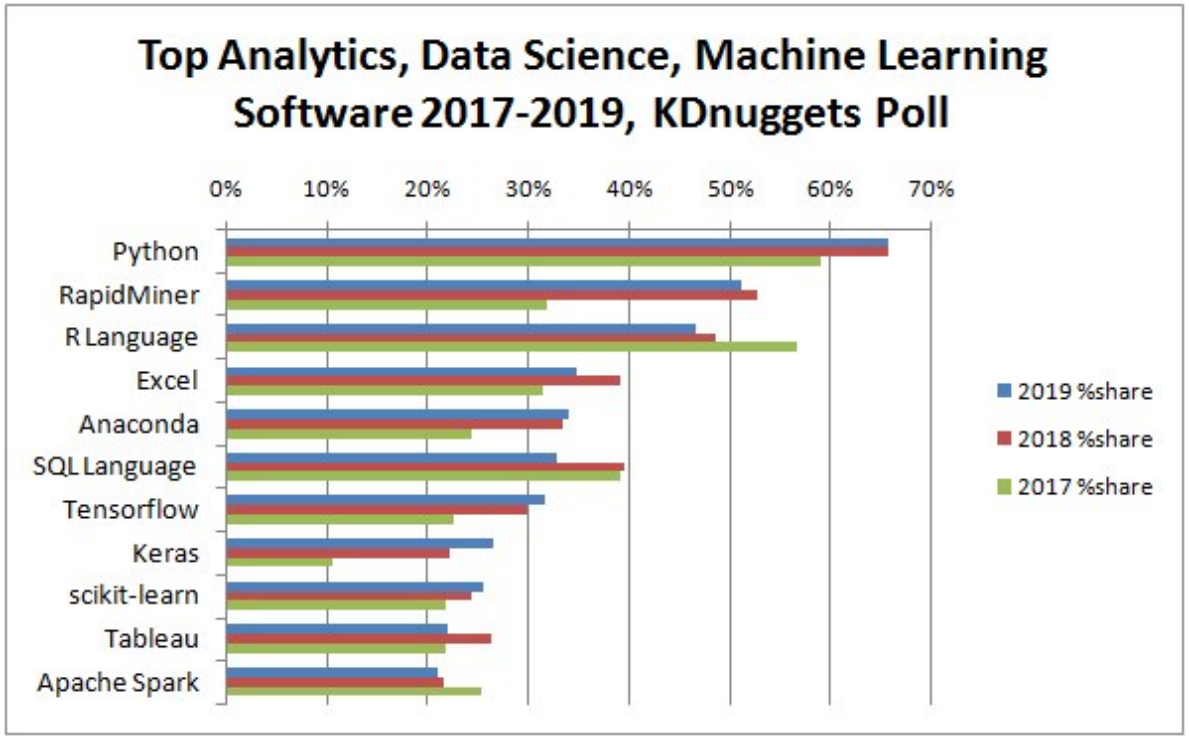
- E-marketing
- Recommender systems
- Sport analytics
- Biotechnology
- Image or speech recognition
- Fraud and risk detection
- Social media
- Credit scoring
- E-commerce
- Government analysis
- Gaming
- Price comparisons
- Airline routes planing
- Delivery logistics

Technology ecosystem



source: rosebt.com

Most popular?



source: kdnuggets.com

Outline

Course description

- 02.17/03.02/03.16/03.30/04.27/05.11 & 05.25 (presentations)
- Lectures:
  - Focus on introducing the concepts
  - 8:30-10:00am + 12:30-2:00pm
  - Classroom 237, Internef building
- Exercise sessions:
  - Focus on the assignments and project
  - 10:15-12:00pm + 2:15-4:00pm
  - Lab room 143, Internef building
- TA: https://irudnyts.github.io/, iegor.rudnytskyi@unil.ch

Date	Assignment
03.15	Project proposal
03.15	HW1
03.29	HW2
04.26	Project update
04.26	HW3
05.24	Project report

Lectures

Date	Topic
02.17 (am)	Introduction, R workflow and RMarkdown
02.17 (pm)	Wrangling I
03.02 (am)	Visualization I
03.02 (pm)	Wrangling II
03.16 (am)	Visualization II
03.16 (pm)	Modeling I
03.30 (am)	Modeling II
03.30 (pm)	Project coaching
04.27 (am)	Presentations, Dashboards, Interactivity
04.27 (pm)	Guest lectures
05.11 (am)	Project coaching
05.11 (pm)	Project coaching
05.25 (am)	Projects presentations
05.25 (pm)	Projects presentations

Lab sessions

Lab Date	Topic	Milestone
02.17 (am)	R, Rstudio and Github, R Refresher	HW0
02.17 (pm)	R workflow, RMarkdown, and data wrangling (I)	HW1
03.02 (am)	Project	Project
03.02 (pm)	Data wrangling (II) and visualization (I)	HW1
03.16 (am)	Project	Project
03.16 (pm)	Visualization (II) and modeling (I)	HW2
03.30 (am)	Project	Project
03.30 (pm)	Modeling (II)	HW3
04.27 (am)	Project	Project
04.27 (pm)	Project	Project
05.11 (am)	Project	Project
05.11 (pm)	Project	Project

Grading

- 3 assignments (30%) and one project (70%)
  - Detailed reports for each assignment and final project
  - Presentation during last lecture for the project
- Final grade
  - According to

$$GRADE = \frac{\sum_{i=1}^{3} \frac{HW_i}{3} \cdot 30 + PR \cdot 70}{100}$$

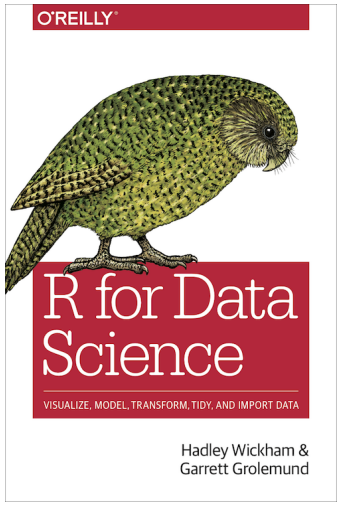
  - HW_i for $i = \{1, 2, 3, 3\}$ and PR are from 0 to 100
  - $GRADE$ will then be adjusted from 1 to 6
- Groups of 1 or 2 members
  - Email to legor with the group members
  - One email per group is enough
  - Deadline for group registration is **March 2**
- Grades based on academic performance only!

Course website

All lecture notes, the syllabus, assignments, and additional resources are available at:

https://tvatter.github.io/dsfba_2020/

Additional resources



- R for data science (Garrett Grolemund and Hadley Wickham)
- Rstudio cheat sheets
- The CRAN website

- Most of the material in the slides is taken from the book.
- It is available online for free, and the slides will be on the course's website.

Best place to look for answers?

Outline

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*

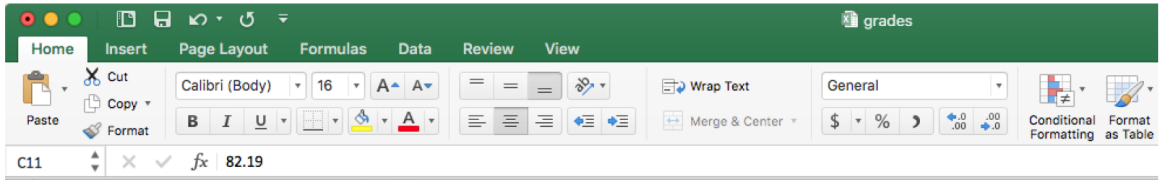
Some "technical" details about R

- **Part of the GNU free software project**
- Source code written primarily in C, Fortran, and R
- **Available for Windows, macOS, and Linux**
- 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. . .



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Student	Midterm	Final	Asn #1	Asn #2	Asn #3	Asn #4	Asn #5	Asn #6	V1	V2	Final Points	Letter Grade
2	Student 1	95.5	91.78	100	100	100	100	100	100	94.54	93.42	94.54	A
3	Student 2	93.2	89.04	100	100	100	100	100	100	92.48	91.23	92.48	A
4	Student 3	95.5	86.3	100	100	100	100	100	100	91.80	89.04	91.80	A
5	Student 4	94.3	86.3	100	100	100	100	100	100	91.44	89.04	91.44	A
6	Student 5	95.5	82.88	100	100	75	100	100	100	89.26	85.47	89.26	A
7	Student 6	79.5	86.3	100	100	100	100	100	100	87.00	89.04	89.04	A
8	Student 7	84.1	85.6	100	100	100	100	100	100	88.03	88.48	88.48	A
9	Student 8	94.3	80.14	100	100	100	100	100	100	88.36	84.11	88.36	A
10	Student 9	94.3	80	100	100	100	100	100	100	88.29	84.00	88.29	A
11	Student 10	89.8	82.19	100	100	100	100	100	100	88.04	85.75	88.04	A
12	Student 11	90.9	81.51	100	100	100	100	100	100	88.03	85.21	88.03	A
13	Student 12	93.2	78.77	100	100	100	100	100	100	87.35	83.02	87.35	A
14	Student 13	89.8	81.5	100	75	100	100	100	100	86.86	84.37	86.86	A
15	Student 14	86.4	81.5	100	100	100	100	100	100	86.67	85.20	86.67	A
16	Student 15	93.2	76.71	100	100	100	100	100	100	86.32	81.37	86.32	A
17	dent 16	97.7	71.23	100	100	100	100	100	100	84.93	76.98	84.93	A-
18	Student 17	86.4	76.71	100	100	100	100	100	100	84.28	81.37	84.28	A-
19	Student 18	87.5	77.4	100	100	100	100	75	100	84.12	81.09	84.12	A-
20	Student 19	90.9	75.34	100	100	100	75	100	100	84.11	79.44	84.11	A-
21	Student 20	81.8	78.77	100	100	100	100	100	100	83.93	83.02	83.93	A-
22	Student 21	76.1	78.77	100	100	100	100	100	100	82.22	83.02	83.02	B+
23	Student 22	84.1	71.92	100	100	100	100	100	100	81.19	77.54	81.19	B+
24	Student 23	85.2	71.23	100	100	100	100	100	100	81.18	76.98	81.18	B+
25	Student 24	67	76.03	100	100	100	100	100	100	78.12	80.82	80.82	B+
26	Student 25	85.2	69.18	100	100	100	100	100	100	80.15	75.34	80.15	B+
27	Student 26	81.8	70.55	100	100	100	100	100	100	79.82	76.44	79.82	B+
28	Student 27	81.8	70.55	100	100	100	100	100	100	79.82	76.44	79.82	B+

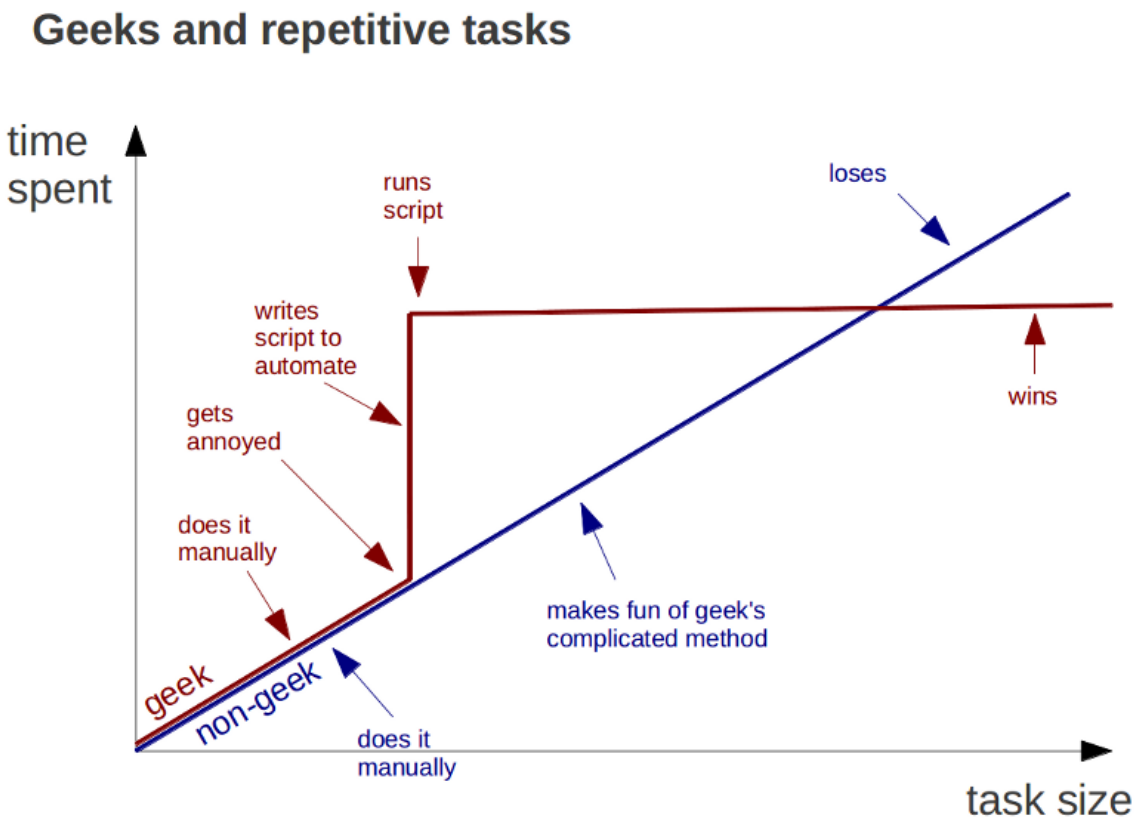
source: github.com/jdwilson4

. . . 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 organization
- 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



source: trendct.org

How about Python?

source: python.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.

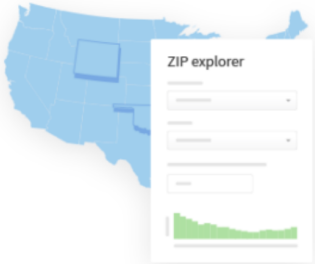
source: cran.r-project.org

RStudio

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

RStudio

RStudio makes R easier to use. It includes a code editor, debugging & visualization tools.



Shiny

Shiny helps you make interactive web applications for visualizing data. Bring R data analysis to life.

R Packages

Our developers create popular packages to expand the features of R. Includes ggplot2, dplyr, R Markdown & more.

source: rstudio.com

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 perform the same tasks as `Base R`, but:
  - Provides 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.
- `readdr`: reading in rectangular data.
- `purrr`: enhancing R's functional programming (FP).
- `tibble`: a `tibble`, or `tbl_df`, is a modern reimagining of the `data.frame`.
- `stringr`: functions designed to make working with strings as easy as possible.
- `forcats`: useful tools that solve common problems with factors.

More on the tidyverse website!

Base R versus tidyverse

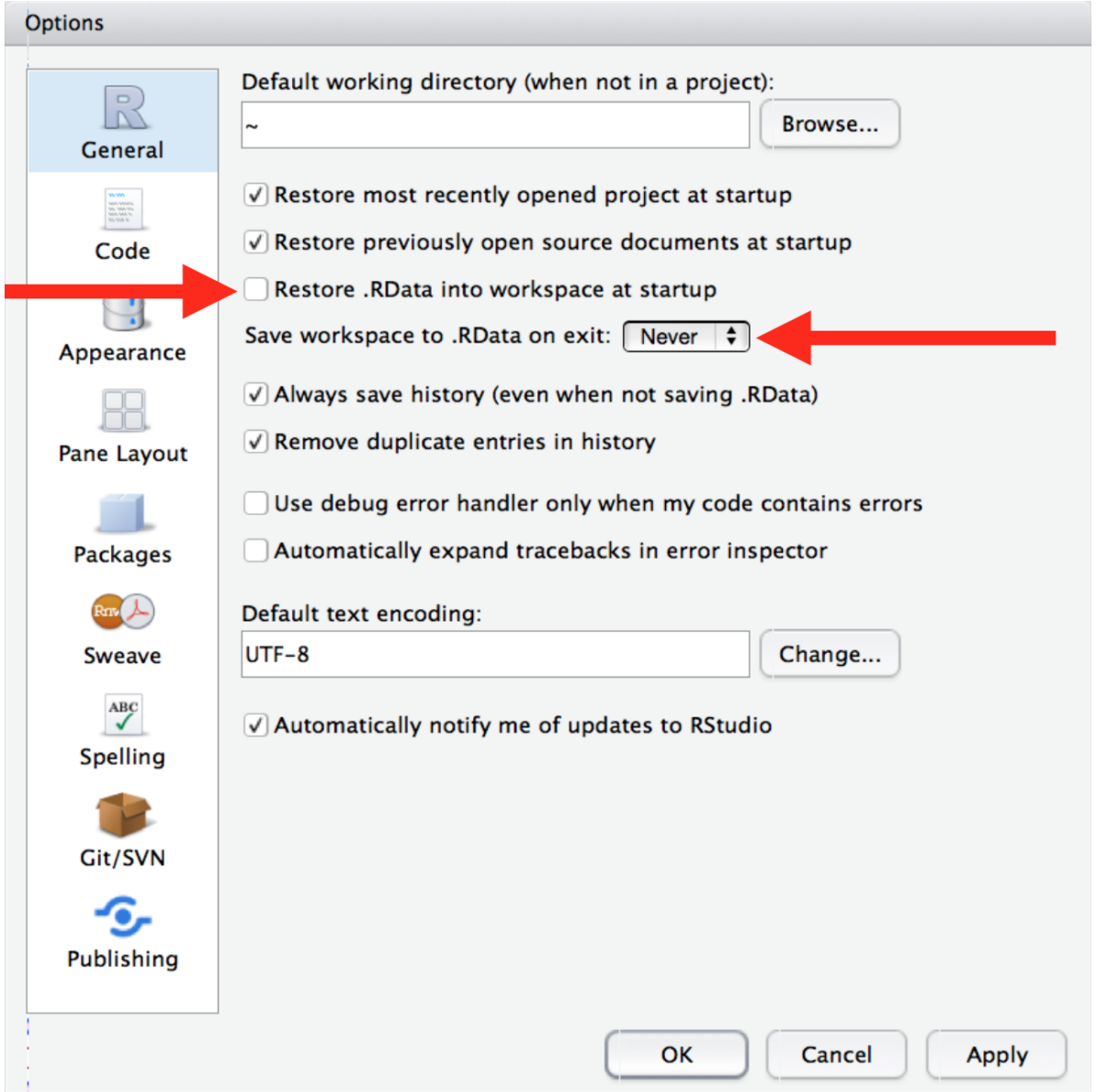
- Why ever use `Base R`?
  - Gets the job done!
  - To become an expert, you have to know `Base R`.
  - Some `Base R` functions are very common/useful, e.g., `mean()`.
- What should you learn first? `Base R` or `tidyverse`?
  - Some believe you should learn `Base R` first, others the `tidyverse` first.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

Outline

Two questions

- What about your analysis is "real"?
- Where does your analysis "live"?

What about your analysis is "real"?



Where does your analysis live?

- The console
- R scripts
- RStudio projects: make it straightforward to divide your work into multiple contexts, each with their own working directory, workspace, history, and source documents.

DEMO!

The workflow with RStudio projects

- Create an RStudio project for each data analysis project.
- Keep data files there.
- Keep scripts there.
- Save your outputs (plots and cleaned data) there.
- Only ever use relative paths (e.g., with `here::here`), not absolute paths.

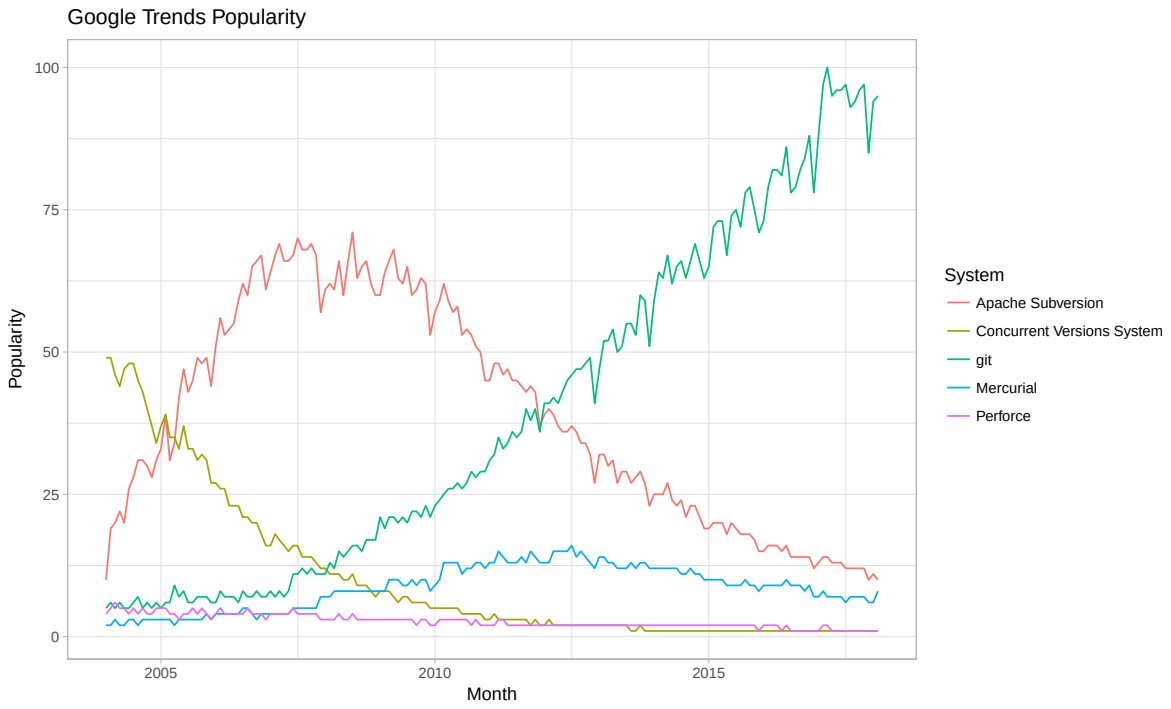
Everything you need is in one place, and cleanly separated from all the other projects that you are working on.

Outline

Version control

"Management of changes to documents, computer programs, large web sites, and other collections of information."

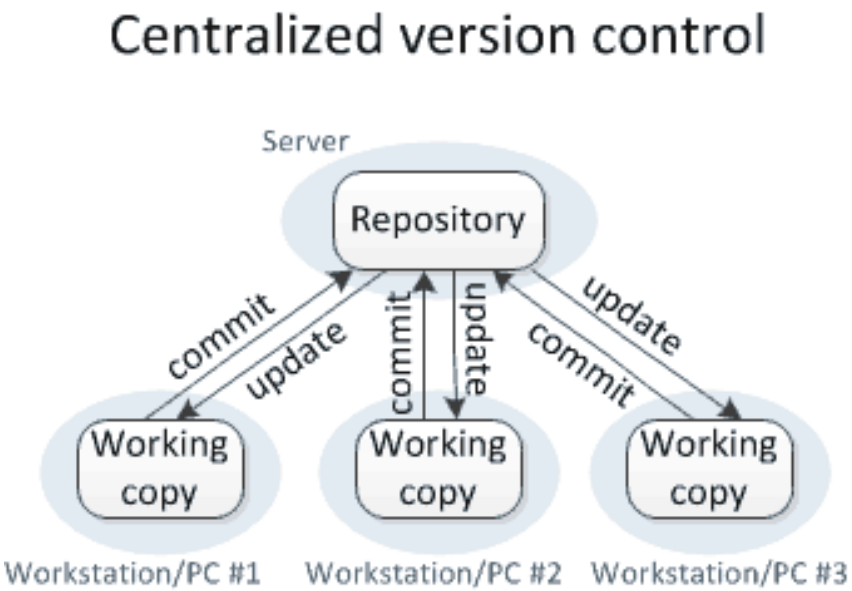
— Wikipedia

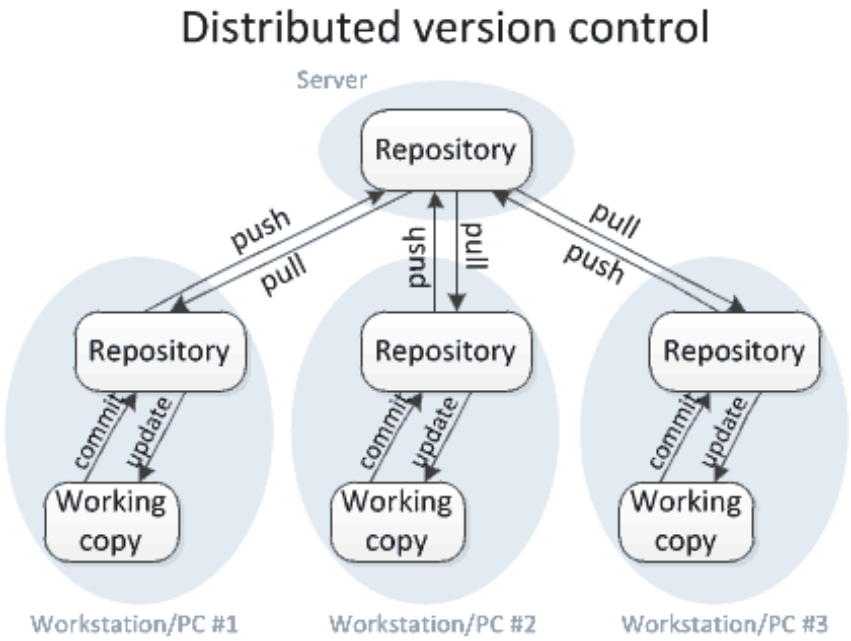


Git: "the stupid content tracker"

- Created by Linus Torvalds in 2005, his criteria:
  - Patching should take < 3 seconds
  - CVS as an ex. of what not to do (in doubt, do the opposite)
  - Distributed workflow
  - Strong safeguards against corruption (accidental or malicious)
- Maintained by Junio Hamano since 2005
- Part of the GNU free software project
- Source code written primarily in C, Shell, Perl, Tcl, Python
- Available for Windows, macOS, and Linux

Centralized vs distributed workflows





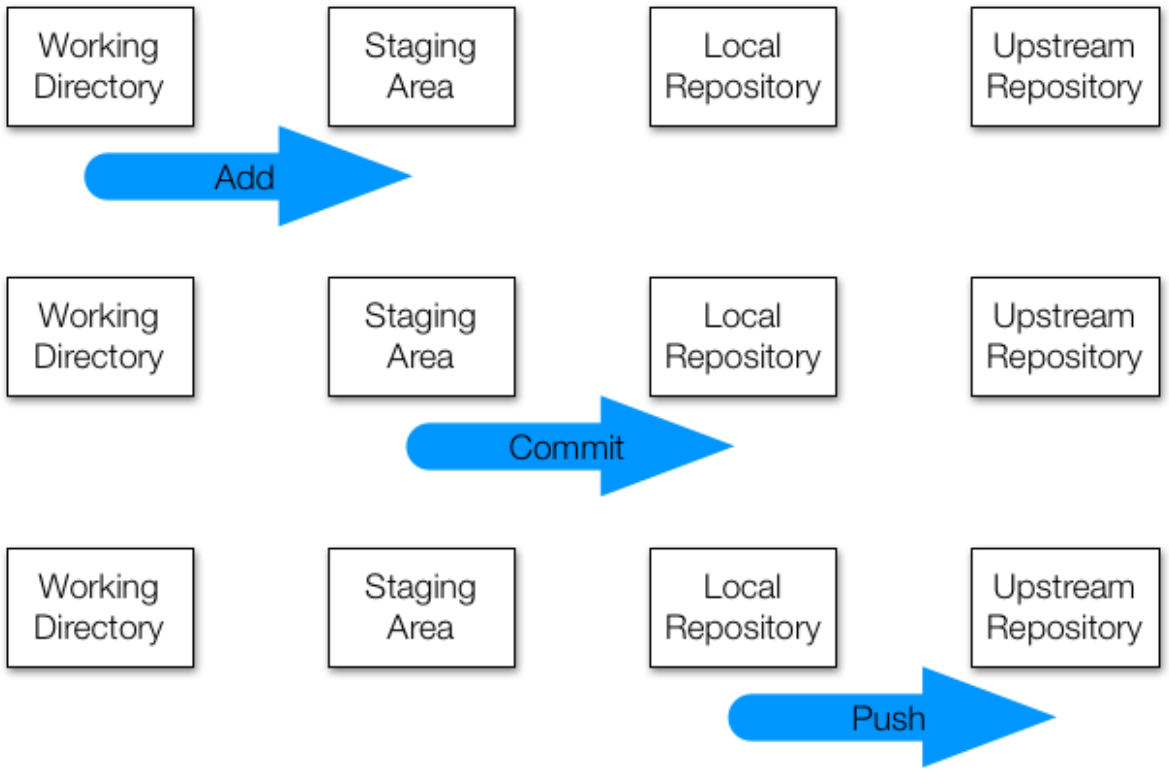
"Version control concepts and best practices" by Michael Ernst

GitHub

- Web-based version control service using git
- Bug tracking, feature requests, task management, and wikis for every project
- 40+ million users and 100+ million repositories (January 2020)
- Private and public repos
- GitHub Student Developer Pack
- Create account & email user to Iegor by **March 2**.

DEMO!

- Work on your assignment
- Commit changes to your local repository
- Push the changes to the GitHub repo



Best practices

- Use a descriptive commit message
- Make each commit a logical unit
- Avoid indiscriminate commits
- Incorporate others' changes frequently
- Share your changes frequently
- Coordinate with your co-workers
- Remember that the tools are line-based
- Don't commit generated files

See "Version control concepts and best practices" by Michael Ernst

Final comments

- Last commit before midnight of due date as final submission
  - If there are commits after midnight, we will take the last commit up to the due date at 11:59 pm as the final version
- Check that the final commit is showing in your GitHub repo
  - "I forgot to push" is not an acceptable excuse
- Detailed tutorials (with lots of pictures):
  - The best
  - Setting-up GitHub
  - Git and RStudio
  - GitHub and RStudio (alternative)
  - GitHub and RStudio (alternative 2)

Outline

R Markdown

- The two components:
  - Literate programming
  - Markdown

Literate programming

- Motivation: helps peers understand and replicate your results, find errors and suggest enhancements
- Introduced by Donald Knuth

> *"a program is given as an explanation of the program logic in a natural language, such as English, interspersed with snippets of macros and traditional source code, from which a compilable source code can be generated [... It] represents a move away from writing programs in the manner and order imposed by the computer, and instead enables programmers to develop programs in the order demanded by the logic and flow of their thoughts."*
>
> — *Wikipedia*

What does this R code do?


```r
data(women)
plot(women)
fit <- lm(weight ~ height, data = women)
abline(fit)
```

And this one?


```r
# Analysis of the 'women' dataset in R
data(women) # Load the data
attach(women) # Attach data to path
plot(weight ~ height) # Make a scatter plot
fit <- lm(weight ~ height) # Fit linear model
abline(fit) # Add a line of best fit to the plot
```

Two competing "views"

"Real programmers don't comment their code. If it was hard to write, it should be hard to understand."

— unknown

"If you can't write clearly, you probably don't think nearly as well as you think you do."

— Kurt Vonnegut

Can't we do better?

The **World Almanac and Book of Facts** (1975) includes a dataset of heights (in) and weights (lbs) of 15 American women aged 30–39. It is built into R:

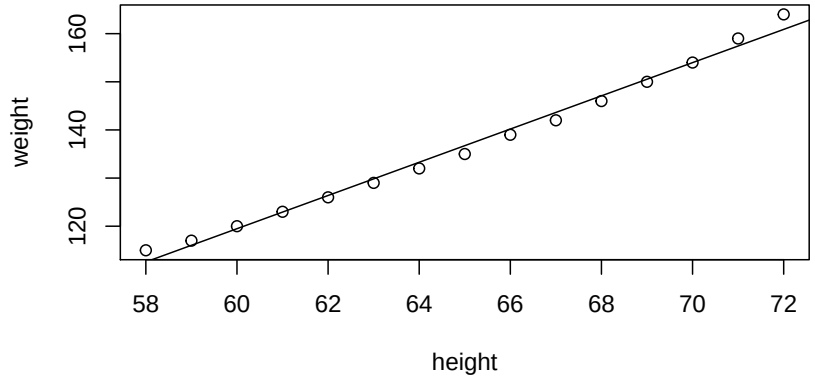
```
data(women)
```

Weight appears to increase (almost) linearly with height: every inch in height adds approximately 3.45 lbs. This was determined by fitting a simple linear regression model of weight against height:

```
fit <- lm(weight ~ height, data = women)
```

The resulting least-squares regression line can be drawn on a scatter plot of height against weight, where the models seems appropriate:

```
plot(weight ~ height, data = women)
abline(fit)
```



How did I do that?


```
The __World Almanac and Book of Facts__ (1975) includes a dataset of
heights (in) and weights (lbs) of 15 American women aged 30-39.
It is built into R:

```{r}
data(women)
```

Weight appears to increase (almost) linearly with height: every inch in
height adds approximately 3.45 lbs. This was determined by fitting a
simple linear regression model of weight against height:

```{r}
fit <- lm(weight ~ height, data = women)
```

The resulting least-squares regression line can be drawn on a scatter plot
of height against weight, where the models seems appropriate:

```{r}
plot(weight ~ height, data = women)
abline(fit)
```
```

A lightweight markup language

- **Markup:**
  - A system for annotating a document in a way that is syntactically distinguishable from the text
  - E.g., LaTeX and HyperText Markup Language (HTML)
- **Lightweight:**
  - A markup language with simple, unobtrusive syntax
  - E.g., Markdown and R markdown

Markup vs lightweight markup

Here is some text:

- in *italics*,
- in **boldface**.

In Latex:

```
Here is some text:
\begin{itemize}
\item in \textit{italics},
\item in \textbf{boldface}.
\end{itemize}
```

In Markdown:

```
Here is some text:
* in *italics*,
* in **boldface**.
```

What is R markdown?

A markdown-based literate programming system

DEMO!

Useful resources

- Essential: R Markdown cheat sheet
- RStudio's R markdown website
  - Tutorial (to get you started)
  - Output formats (e.g., HTML, Word documents, PDFs, presentations, etc.)
- stuff written by Yihui
  - knitr and especially [its options page](https://yihui.name/knitr/options/
  - bookdown to write technical reports
  - blogdown to even build your own website