Data Science and R

David Gerard

2019-08-01

Learning Objectives

- Introduction to Course
- Placing R in the context of data science.
- Chapter 1 in RDS

Data Science

Components of Data Science

- Statistics
- Domain Knowledge
- Computation

Statistics

- Inferring general properties given data.
- Causal inference.
- Modeling (generative and predictive).
- Quantifying uncertainty.
- STAT 615 (Regression), STAT 627 (Machine Learning), most of the STAT curriculum.

Domain Knowledge

- Expertise in an area of application.
- E.g. biology, psychology, economics, chemistry, etc. . .
- Allows you to understand data in context.
- Let's you ask interesting questions.
- Let's you spot problems with existing analysis pipelines.
- Various "Tracks" in the data science program.

Computation – This class

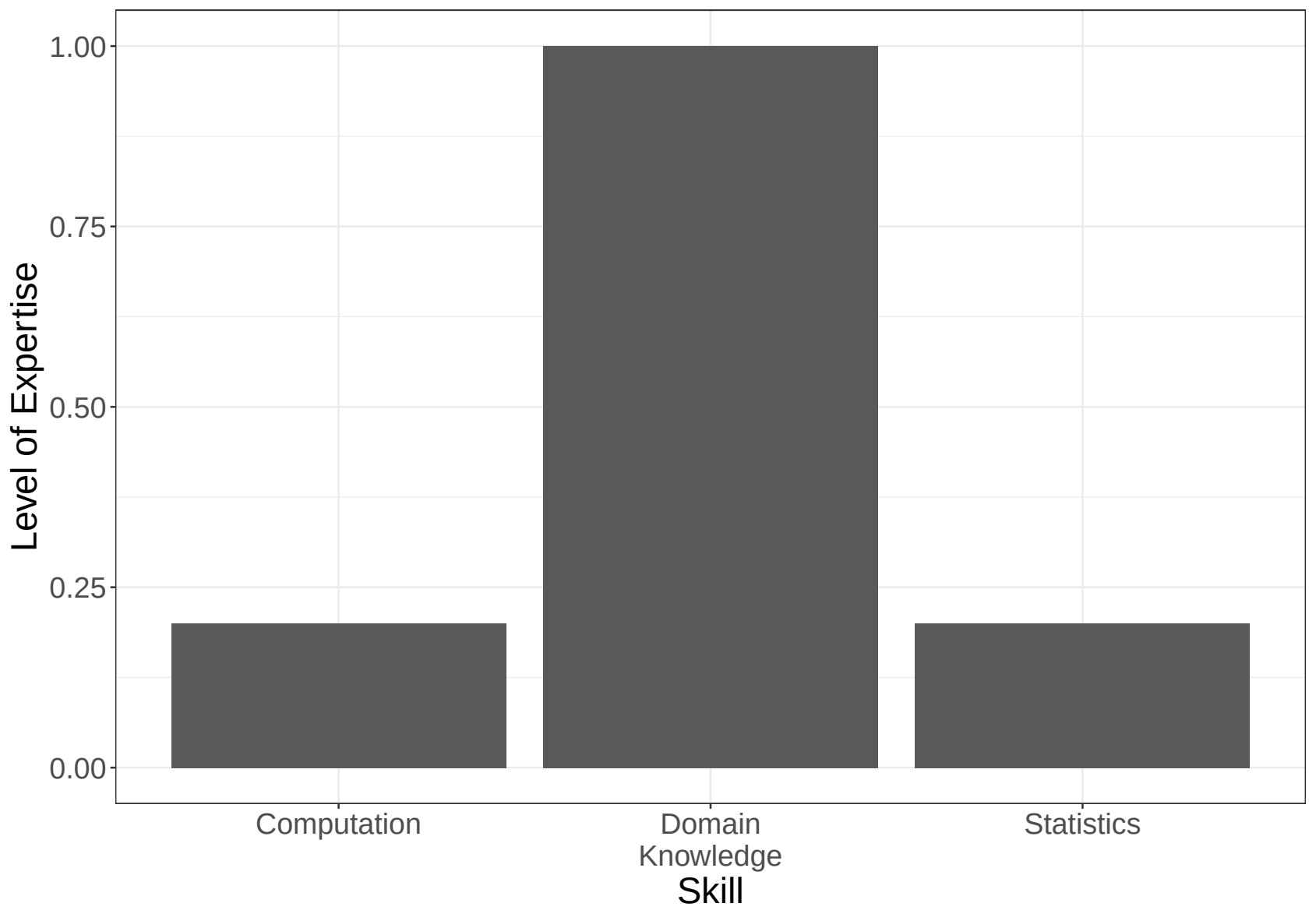
- Data gathering.
- Data preparation.
- Data exploration.
- Data transformation.
- Data visualization.
- STAT 612 (R programming), STAT 613 (Data Science), most of the CS curriculum.

Various Professions

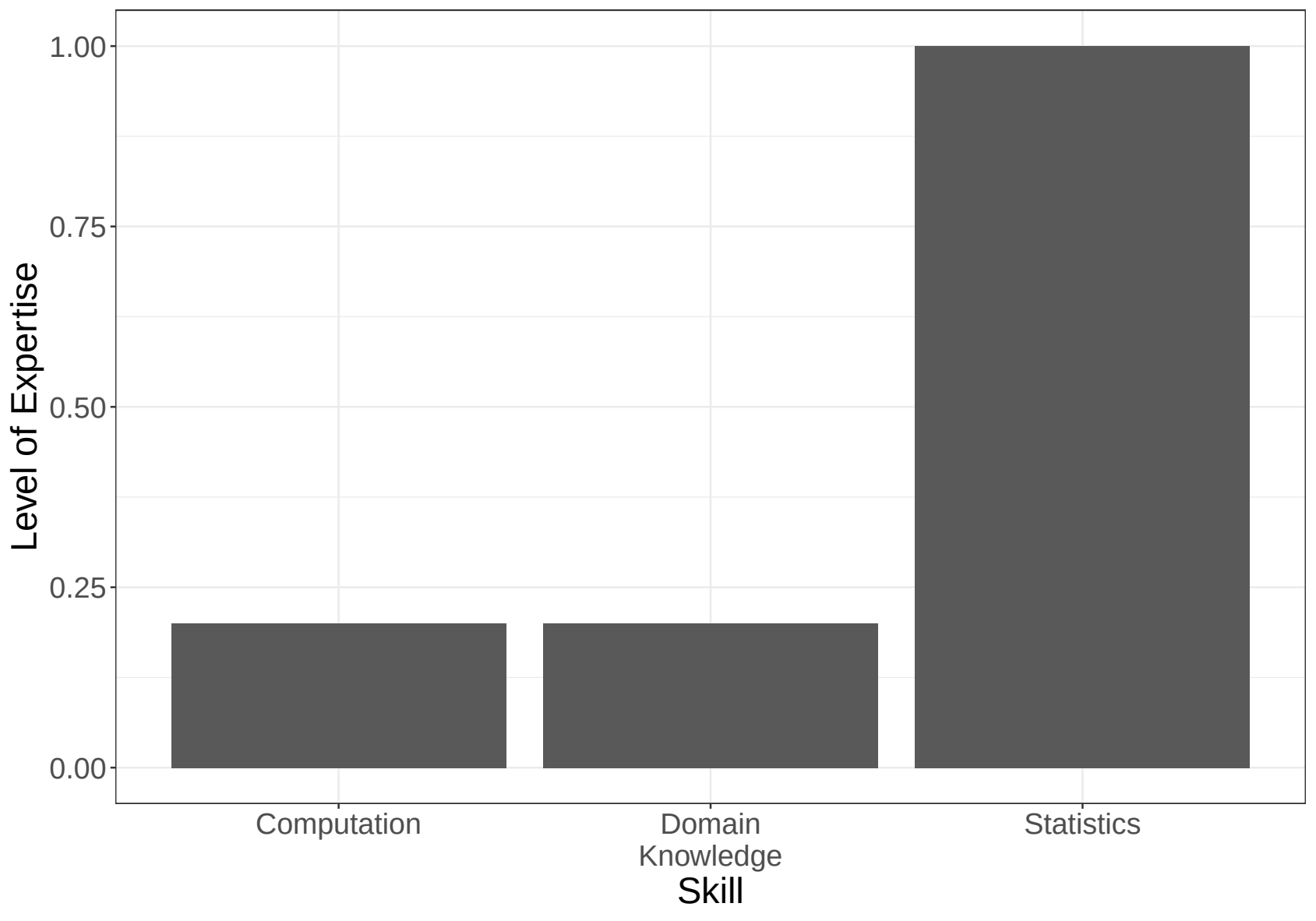
What makes a data scientist?

- Lots of professions use those three skills to analyze data.
- My (very subjective) opinion is that these professions differ by their level of expertise.

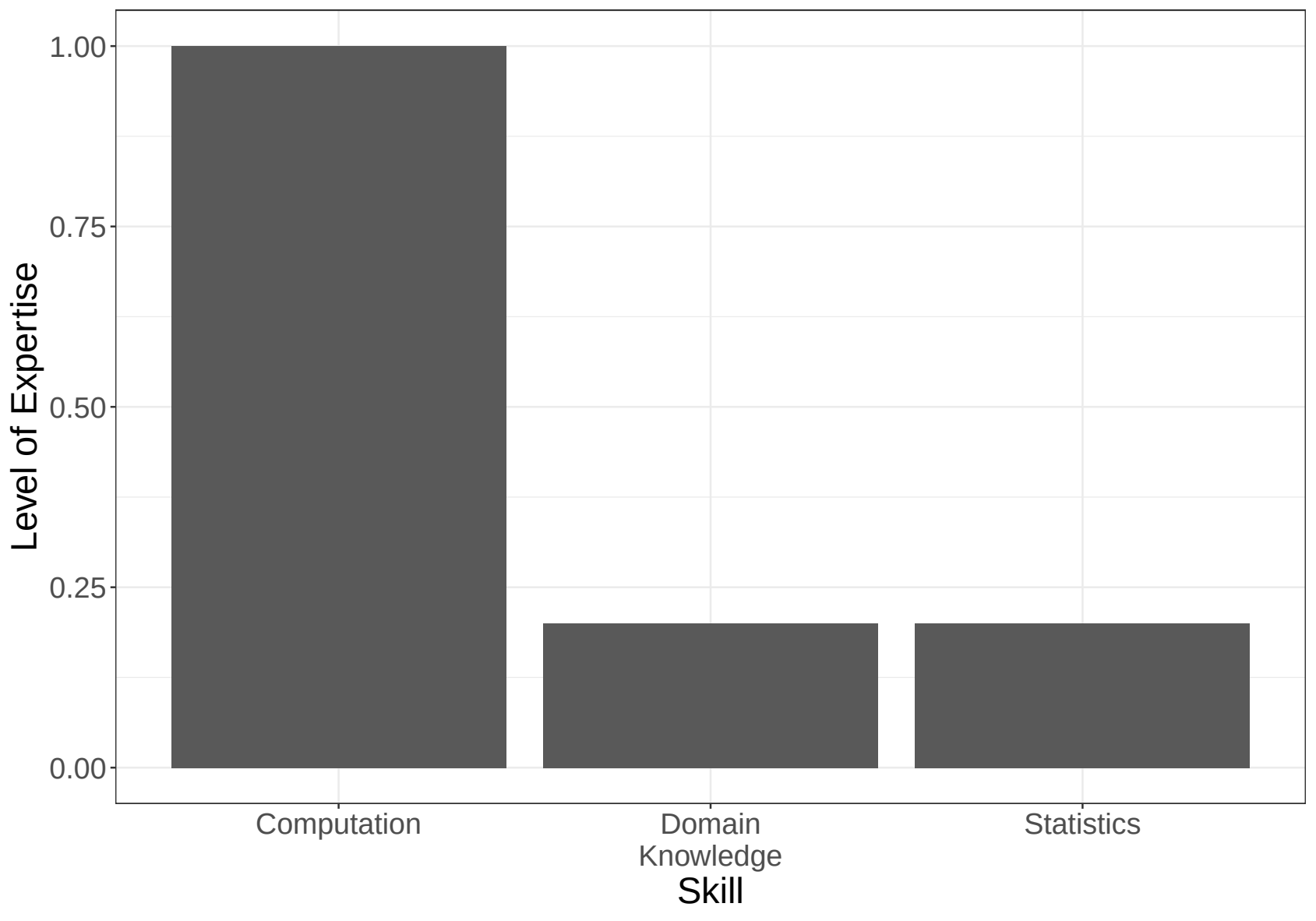
Scientist


1.00
0.75
0.50
0.25
0.00
Level of Expertise
Computation
Domain Knowledge
Statistics
Skill

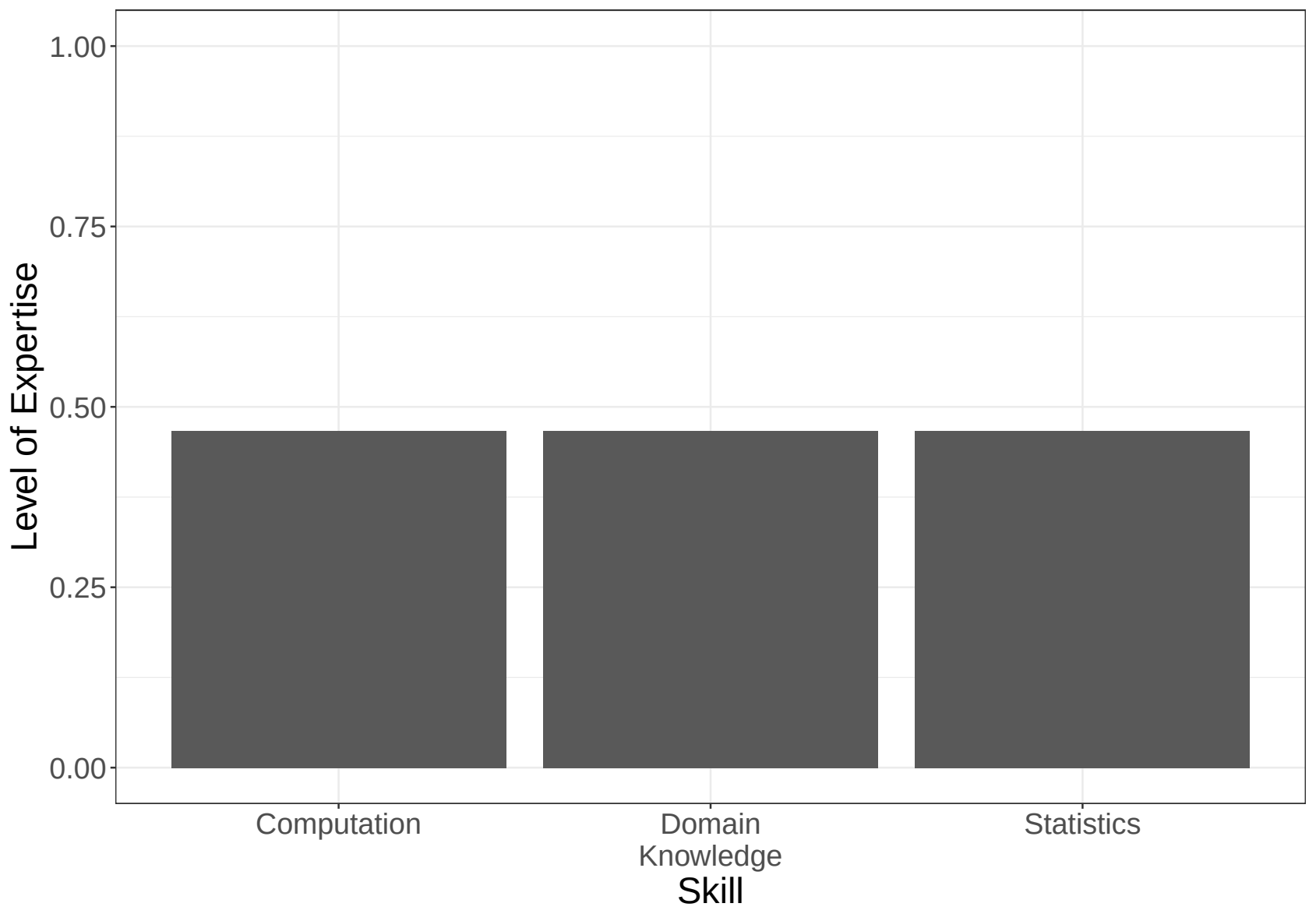
Statistician



Computer Scientist

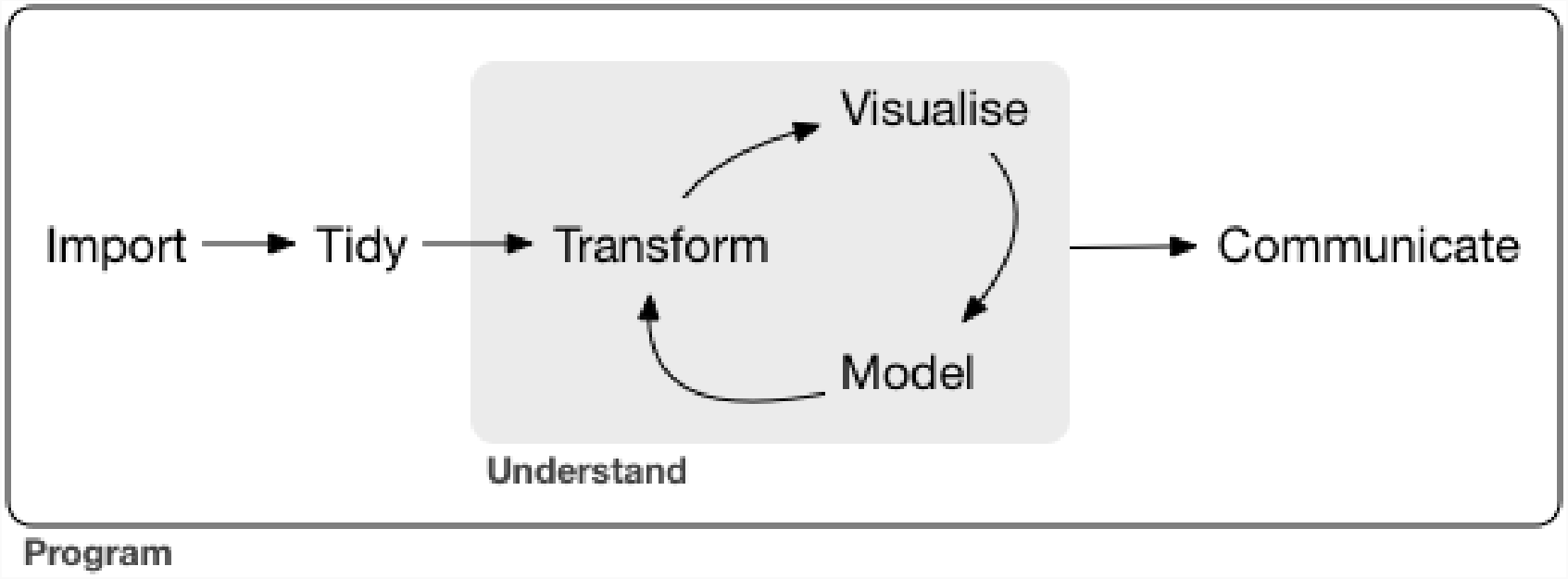


Data Scientist



The steps of an analysis and R

Steps of a data analysis



Tools

- Many tools exist for these steps:
- General data tools: R, Python, Julia, Matlab, STATA, SAS
- Other tools: SQL (data import), git (version control), map/reduce software (for big data).
- Advantages and disadvantages to each.

R

- R is a statistical scripting language.
- You write code (a series of commands) to perform some task.
- R can be used to perform **all** of the tasks of a data analysis.

Motivation for R

1. It's free.
    - You will always have access to R.
    - Not true for other statistical softwares (Matlab, STATA, SAS).

Motivation for R

1. It's free.
    - You will always have access to R.
    - Not true for other statistical softwares (Matlab, STATA, SAS).
2. It's widely used.
    - If you need to do some special analysis, someone has probably already made an R package for it.
    - Just use Google to check.

Motivation for R

1. It's free.
    - You will always have access to R.
    - Not true for other statistical softwares (Matlab, STATA, SAS).
2. It's widely used.
    - If you need to do some special analysis, someone has probably already made an R package for it.
    - Just use Google to check.
3. It's easy (especially graphics and data munging).

Motivation for R

1. It's free.
   - You will always have access to R.
   - Not true for other statistical softwares (Matlab, STATA, SAS).
2. It's widely used.
   - If you need to do some special analysis, someone has probably already made an R package for it.
   - Just use Google to check.
3. It's easy (especially graphics and data munging).
4. It makes reproducible research easy.
   - When part of the pipeline is copying and pasting excel spreadsheets, people make mistakes.
   - E.g. an excel mistake led countries to adopt austerity measures to increase economic growth.
   - In R, you can automate your analysis, reducing the chance for mistakes and making your analysis transparent to the wider research community.

Could Also Learn Python

- Python is also a very good tool for data science.
- Computer scientists tend to prefer it because its syntax is more like a standard computer language. But I think this makes it harder to learn for a non-programmer.
- Main reason to use either tool is based on what your collaborators use.

Two main flavors of R

- There are two flavors of R programmers: Base R users and tidyverse users.
- Base R is more general (not fighting against the system when you want to accomplish a unique task that isn't designed to fit within the tidyverse).
- tidyverse is much more convenient for the vast majority of tasks as long as you drink the kool aid.

This Class

Tentative Schedule

- Week 1: Intro (chapters 1–2), R basics (chapter 4), data frames (chapter 10), R Markdown (chapter 27)
- Week 2: R Programming (pipes, functions, R scripts) (Chapters 6, 17–19)
- Week 3: Data Visualization (chapter 3)
- Week 4: Manipulating data frames with dplyr (chapter 5)
- Week 5: dplyr lab (chapter 5)
- Week 6: Exam 1
- Week 7: Data import and exploratory data analysis: (chapters 7, 11)
- Week 8: tidyr (chapter 12)
- Week 9: tidyr lab (chapter 12)
- Week 10: Exam 2
- Week 11: Relational Data (chapter 13)
- Weeks 12/13: Strings (Chapter 14)
- Week 14: Factors

Books and Resources:

- All material used in this course is free online.
- R for Data Science: https://r4ds.had.co.nz/
- Tidyverse Style Guide: https://style.tidyverse.org/
- Rstudio Cheat Sheets: https://www.rstudio.com/resources/cheatsheets/
- Hands-on Programming with R: https://rstudio-education.github.io/hopr/

Grading

- Assignments: 40% for grad students, 50% for undergrads
- 2 midterms: 12% each
- Final exam: 16%
- Grad students: Final Project: 20%
- Undergrads: Attendance/participation: 10%

Exams

- The exams will be in-class coding assignments. You will have 2.5 hours to complete some coding questions.

Curve

- No curve for individual assignments/exams/projects.
- Curve your overall grade at the end of the semester.
- Curve the median up to 85%.
- Usual cutoffs for A/A-/B+/B/B-/C+/C/C-/D/F.
- My classes usually finish with the pre-curve median being around 75%.

One last note

- The course notes will **not** contain everything you need for the assignments.
- You will need to use Google. Just make sure you understand any support you get and cite your sources.
- This is how most people learn how to code now.