Programming for Analytics

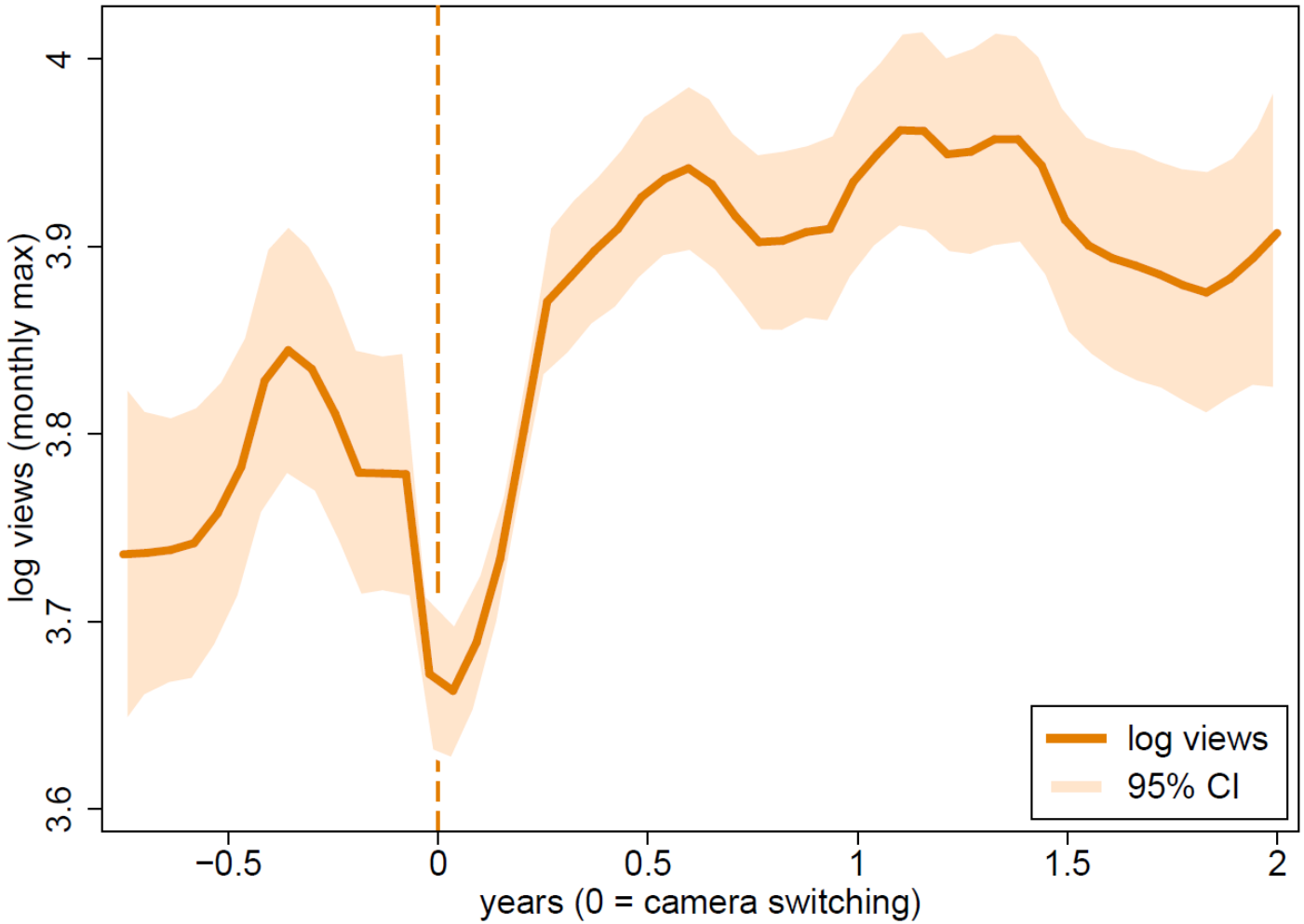
Yufeng Huang

Associate Professor of Marketing, Simon Business School

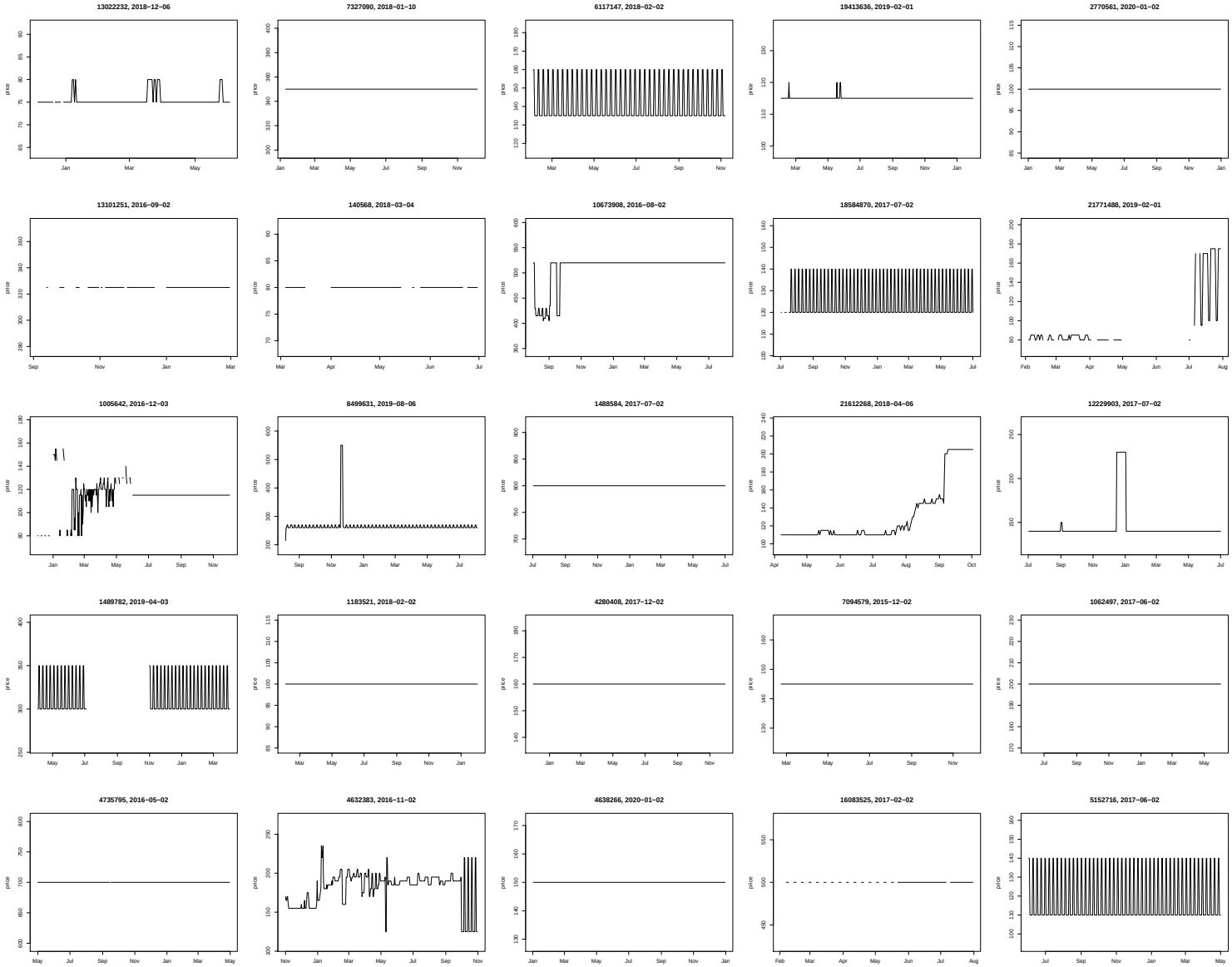
July 25, 2022

Hi, I'm Yufeng, and I write computer programs to learn something about consumers and businesses

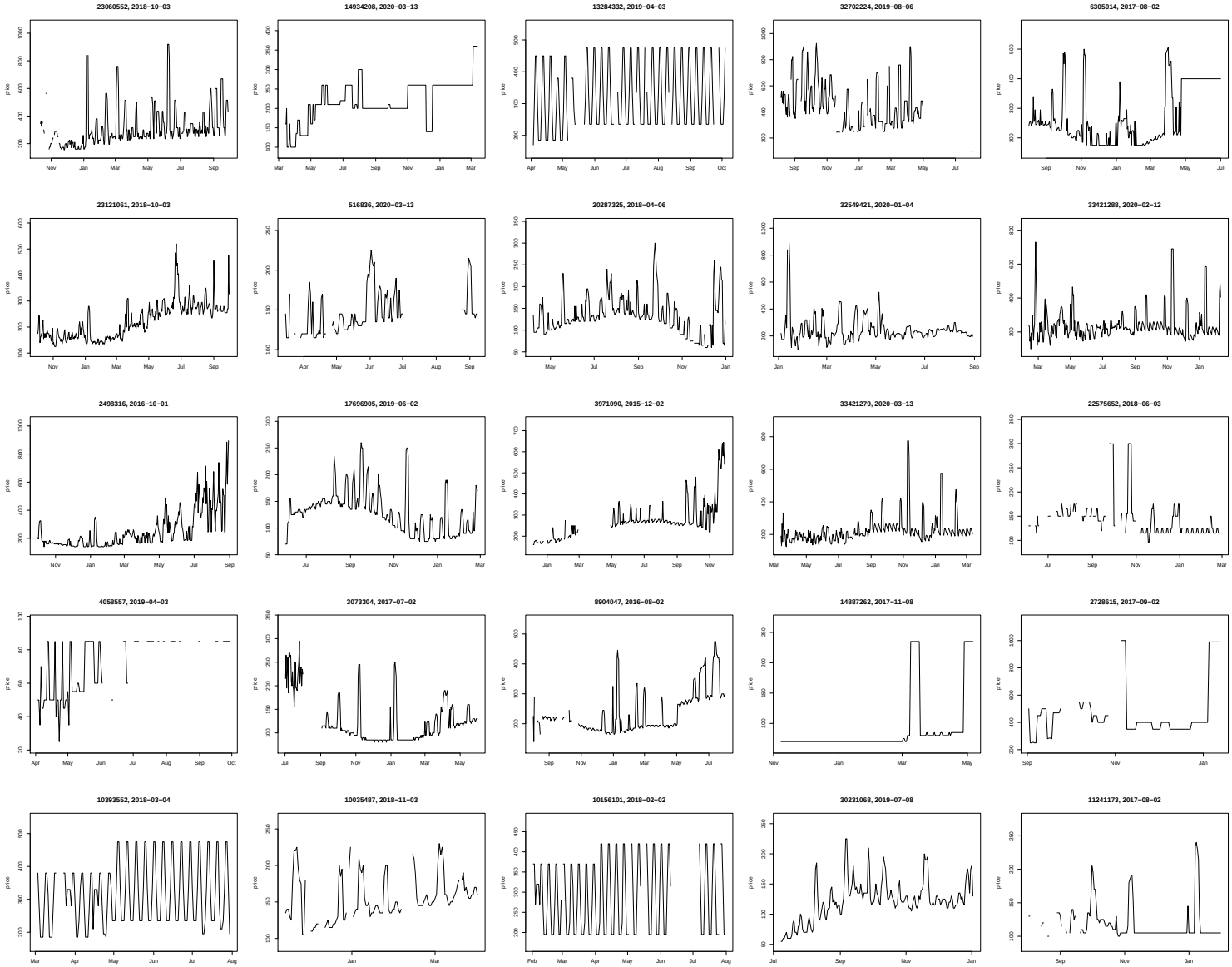
Temporary drop in picture quality when one switches to a new camera



Most Airbnb hosts use simple pricing



A small fraction of Airbnb hosts set algorithm-like prices



What I do

- I teach programming and PhD-level research topic classes
- Conduct research related to data analytics in marketing and economics (more on that later)
- Out of work: cooking, climbing/bouldering, video games, drone videography, building computers, 4WD motor vehicles

Your turn

- What is your undergraduate major (or previous degree)?
- How much experience do you have with data?
- How is your programming experience?
- See survey on BB
  - https://forms.gle/MtcbR7LtBixV9hT19 (MSMA)
  - https://forms.gle/aP2ZD1PrPhjjmjGZ8 (MSBA)

Syllabus

Syllabus

- Goal
- Structure and topics
- Attendance
- Assignments
- Exam
- Grading
- Do's and Don'ts

Learning goals

- Introductory-level course in programming, taught in R
- Goals
  - working with data in R
  - writing clean scripts in R
  - writing and executing structured (and efficient) programs in R
  - working with data and programming language to generate managerially-relevant reports

Structure

- Lectures
  - in total 13 meetings, 2 hours each, in-class
  - every Monday, Tuesday, and Thursday
- Labs
  - Kang Huang (PhD student in Marketing) teaches 1 one-hour lab every Friday, starting this week
  - reviews key concepts we covered in the week's class
  - optional, but *highly* recommended unless you find the class super trivial

Structure (con'd)

- Office hours
  - Kang, Minjie (TA), Siddhartha (TA), Siqi (TA), Xiaoyu (TA), and I will hold one office hours each week (Zoom or in-person? Not sure yet)
  - there will be an office hour every day except on Monday, and I REALLY encourage you to come to office hours to resolve your questions (and please try coming earlier than the assignment deadlines)
  - exact office hour schedules each week will be on Blackboard's "read me first!" section
  - urgent or quick questions: email me or any TA

Week 1: introduction and data structure basics

- Overview of the the R/RStudio environment
- Data types; vectors of numeric data (class starts here)
- Operators and functions on numeric variables

Week 2: data structure

- Arrays, factors, lists and data frame
- Data frame operations
- Text data, regular expressions and dates
- Data table
- Introduction to data visualization

Week 3-4: programming

- Expressions and flow control
- Functions and environments
- Algorithms
- Apply-class functions
- One class on advanced topics
- Review

General interactions

- Generally, interactions are important for learning (including, but not limited to, programming)
- I encourage informal interactions before/during/after class
  - exception is that during class, please direct questions to me instead of your peers
- Also important is interacting/discussing with your peers
  - including talking about the assignments
  - without *explicitly* asking/telling people what the approach is (which would violate academic integrity)

Grading

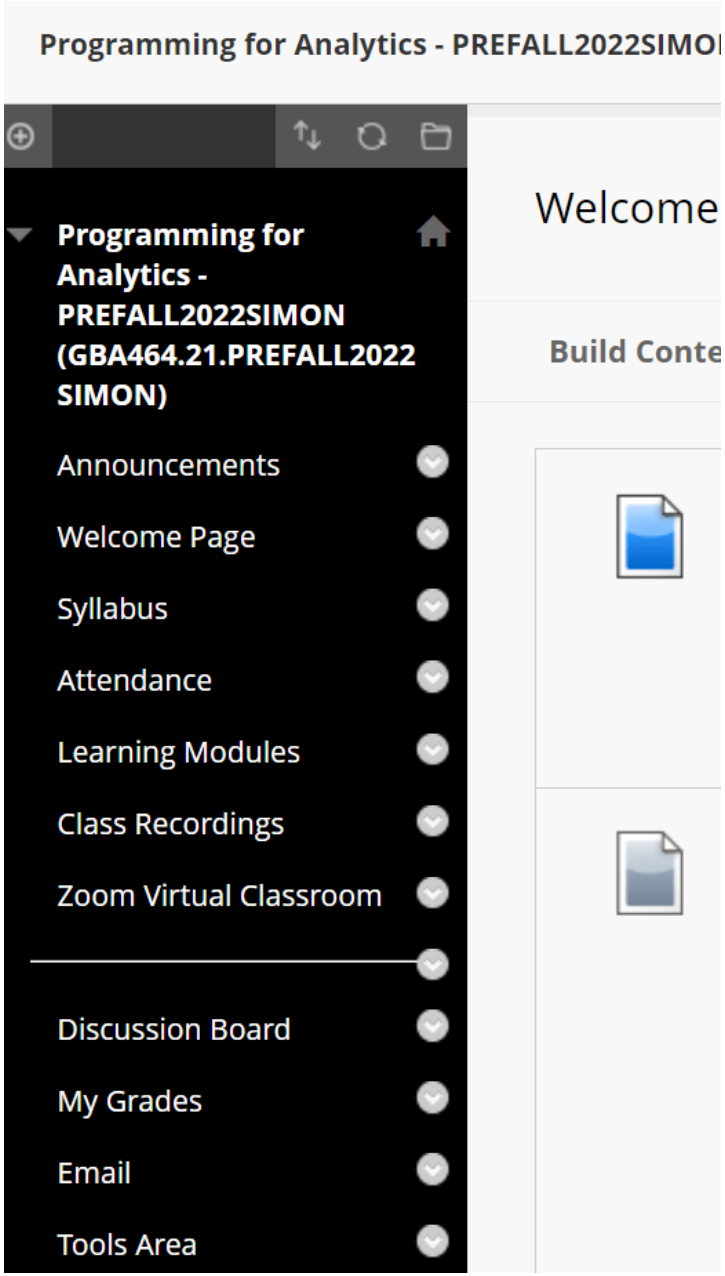
- Raw grades
  - 10% for attendance
  - 10% each of the four assignments
  - 50% final exam
- Raw grades are then curved (i.e. only ranking matters) to produce the letter grades
  - highest grade is A, lowest grade is usually B-

Assignments

- Assignments at the end of week 1, 2, 3 and 4
  - assigned mid-week and due Sunday, 11:59 pm sharp
- Three individual assignments, one group assignment
  - individual assignments are heavy
  - but working hard on assignment is the only way to learn programming
- Stick with your assigned team for group assignments (everyone gets the same grade)

Attendance

- Following school policy, I will track class attendance. Here's how it works
  - I will generate a sign-up code each class. Click on "Attendance" on Blackboard to enter the code and record your attendance
  - code valid for 15 minutes
  - late = missed the class
  - if sick or urgent business, email me before class and you will be excused (except for Covid, more on that later)
  - 12 classes (except today), each student can miss max. one class without any point deducted
    - e.g., miss one class = 10 participation points; miss two classes = 9.17 points; missed one class and 20 minutes late for another class = 9.17 points, and so on


Programming for Analytics - PREFALL2022SIMO
Programming for Analytics - PREFALL2022SIMON (GBA464.21.PREFALL2022 SIMON)
Announcements
Welcome Page
Syllabus
Attendance
Learning Modules
Class Recordings
Zoom Virtual Classroom
Discussion Board
My Grades
Email
Tools Area
Welcome
Build Conte

Covid-19

- If tested positive for Covid-19, follow the UHS's guidance and quarantine
  - email me with a picture of a positive test or the UHS's email instruction to obtain a Zoom link
  - Zoom attendance is tracked. Do not join Zoom unless with permission
- All other reasons: email me, no penalty on attendance, watch video after the class

Textbook

- Main focus on the lecture notes, no required textbook
- Optional reading/exercises
  - [entry-level] Roger D. Peng, "R Programming for Data Science", available on Leanpub
  - [more advanced than the class] Hadley Wickham, "Advanced R", at http://adv-r.had.co.nz/

Language: English

- In class, after class and during office hours
- Some of you might feel uncomfortable at the beginning
- But trust me, good for you in the long run

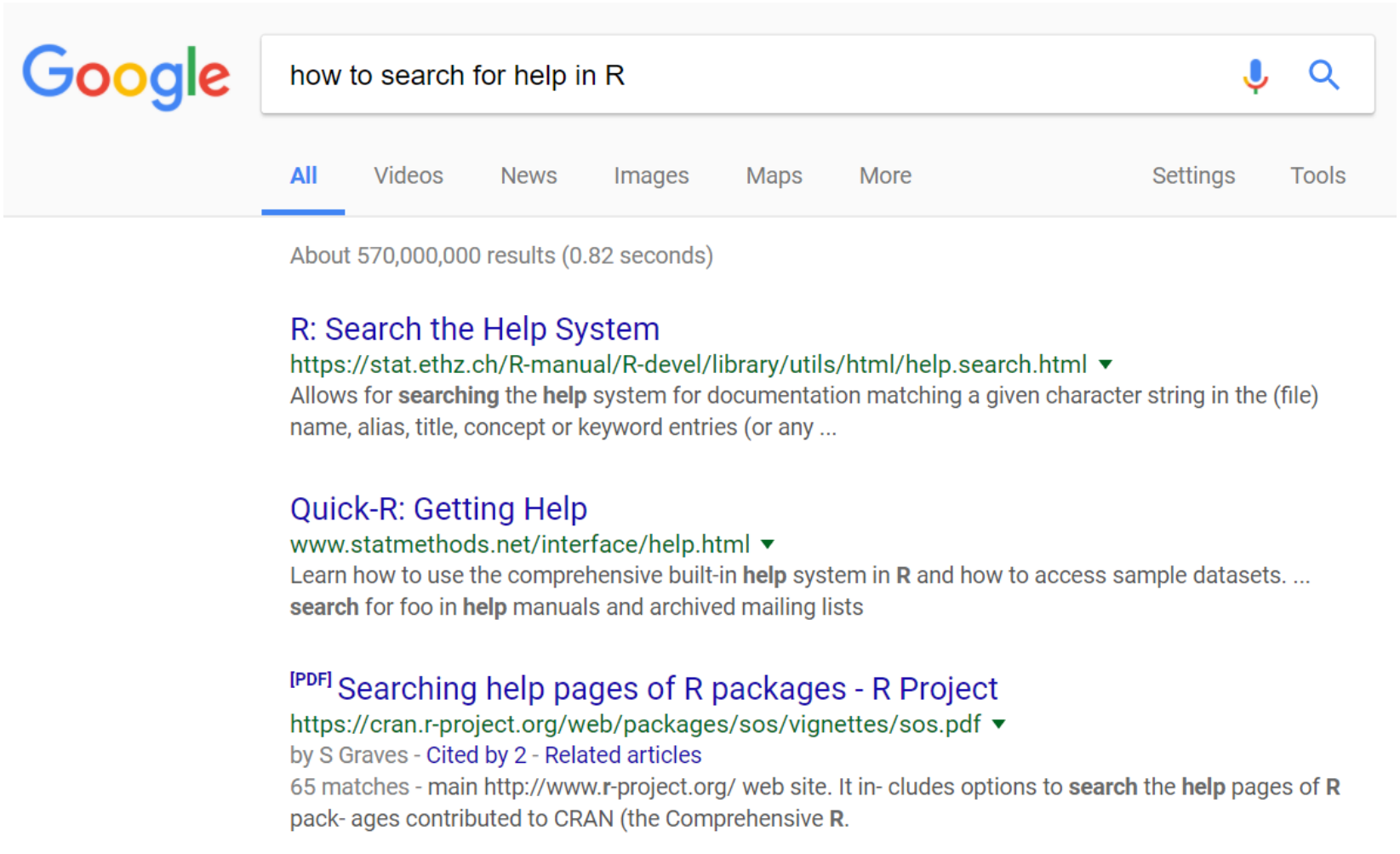
No plagiarism!

- You can talk in general about how you solve something (encouraged in assignments)
- But, you MUST NOT
  - show others your code
  - or ask others to do so
- Plagiarism is strictly not accepted in the university
  - no help when you go to an interview or generally after you graduate
- Instead, you are encouraged to:
  - seek help from me
  - from the TAs
  - from help() or the internet (which is very important as we go through this class)

Ask questions any time during the class!! (Interrupt me at any time)

Bring your laptop in class

Seek help(), Stackoverflow, and Google!!



Motivation: Why learn programming?

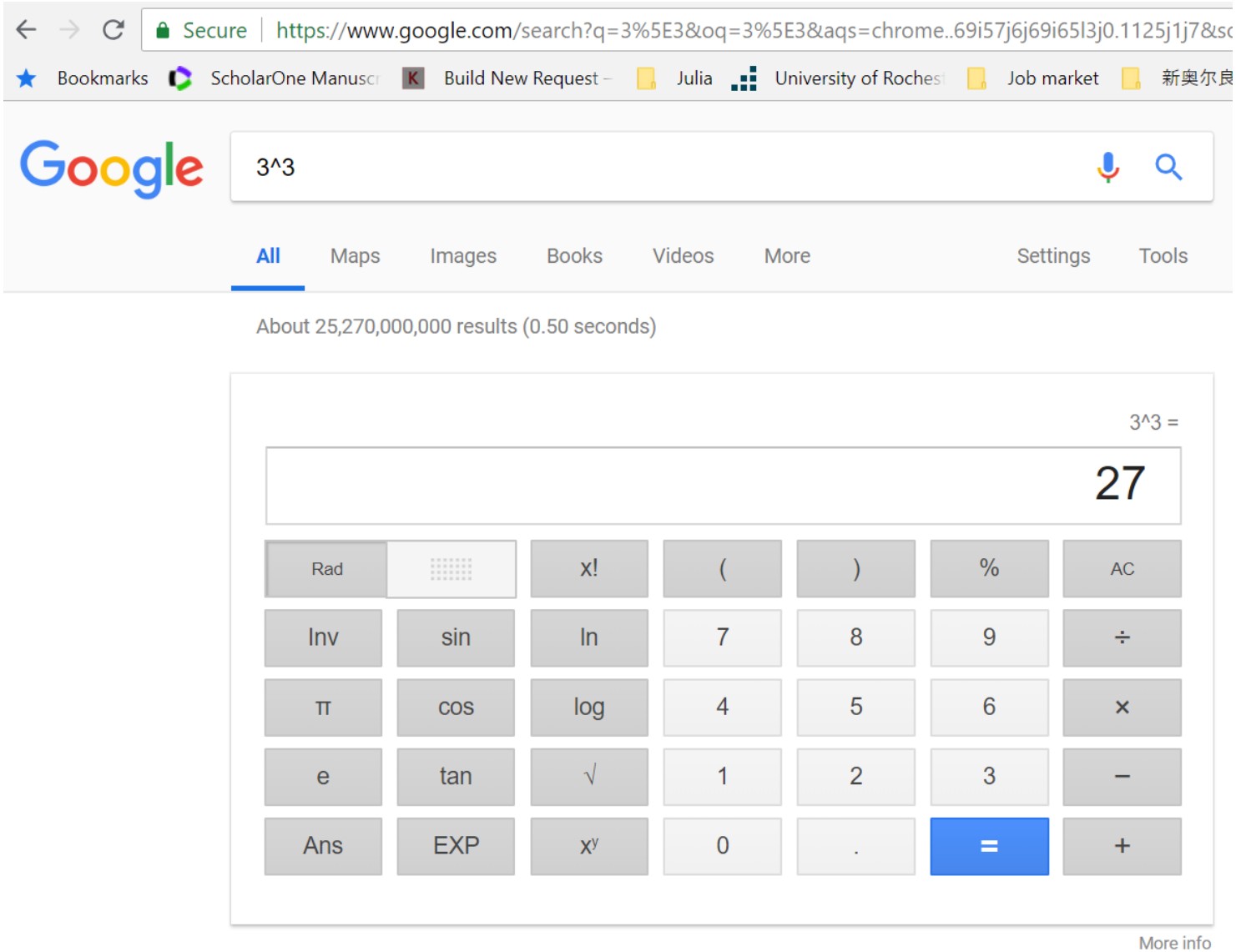
"The Excel Depression"[1]

- Reinhart and Rogoff (2010), "Growth in the time of debt"
- When debt/GDP ratio goes above 90%, growth rate falls by >=1%
- By 2013, this is accepted as gold standard limit for debt policy
  - just happened that Greece was a big topic those days
  - e.g. Wash. Post, we are "dangerously near the 90 percent mark that economists regard as a threat to sustainable economic growth."
- ...until a U. Mass. PhD student requested their Excel sheet
  - and discovered that they did not drag their average function long enough that they excluded 5 rows

[1] Term from Krugman, "The Excel Depression", NY Times, April 2013

- What I meant to say is that Excel is un-debug-gable
  - it's easy to make a mistake by hand
  - in a formal programming language you need to explicitly take average across the wrong data range to get the same mistake
  - even then the mistake is easy to spot
- Excel's a good software but it's not really for good data analytics practices
  - un-scalable: not easy to apply one procedure to other variables/datasets
  - un-portable: files are good when stored in .xlsx and not easy to work with (non-Microsoft) programs
  - later in this class, I'll explain what scalability/portability/debuggability are and why they matter

Example 1a: Looking for a calculator



Example 1a: Using R to do basic math[2][3]

```
2 + 3
```

```
## [1] 5
```

```
3 * 4
```

```
## [1] 12
```

```
3^3
```

```
## [1] 27
```

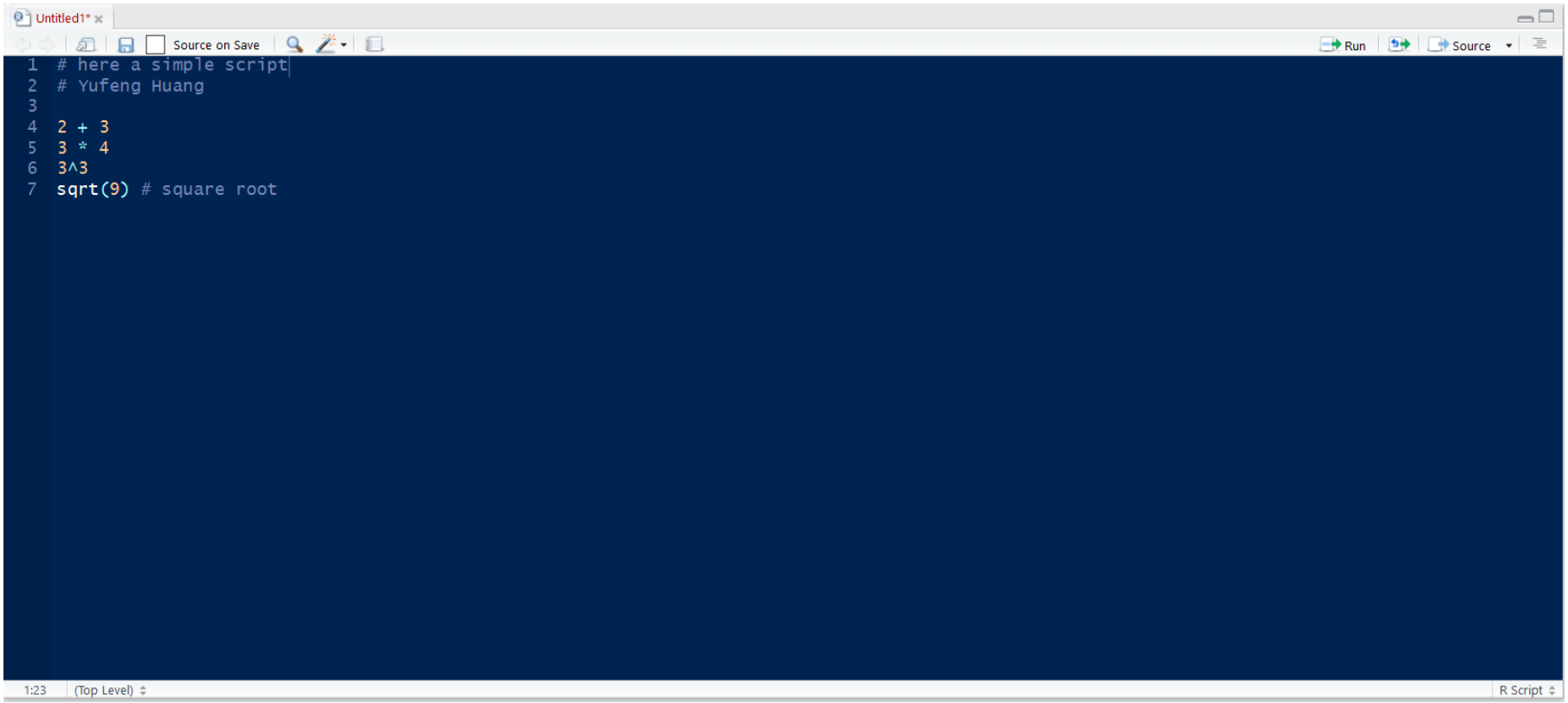
```
sqrt(9) # square root
```

```
## [1] 3
```

[2] By the way, I'm running R when compiling the slides

[3] Btw, script convention in the lecture notes: green = number, dark red = function, light red = comment (not part of the code), "##": console output

A good habit to develop: use the editor (instead of the console) in RStudio

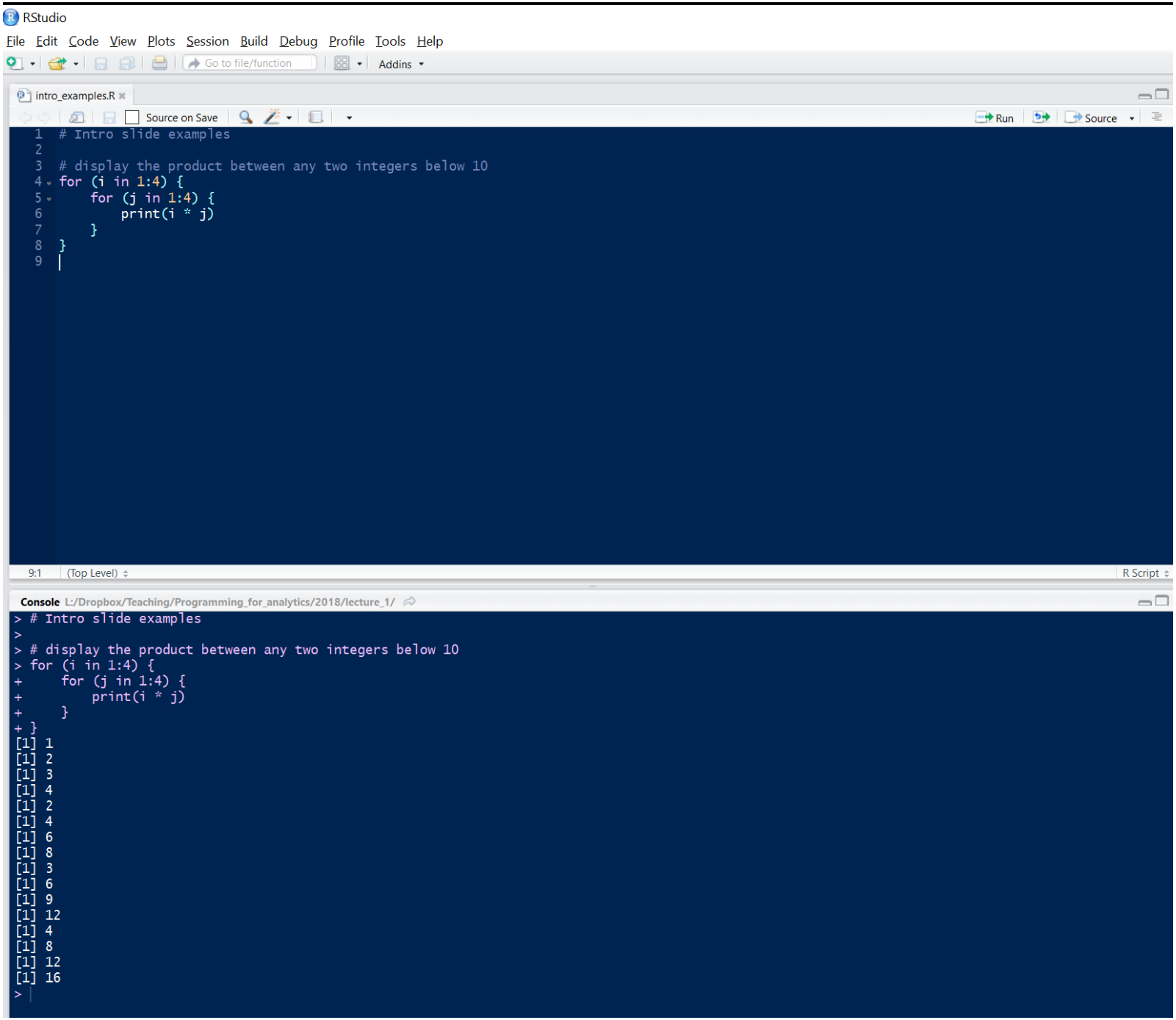


Example 1b ("advanced calculator")

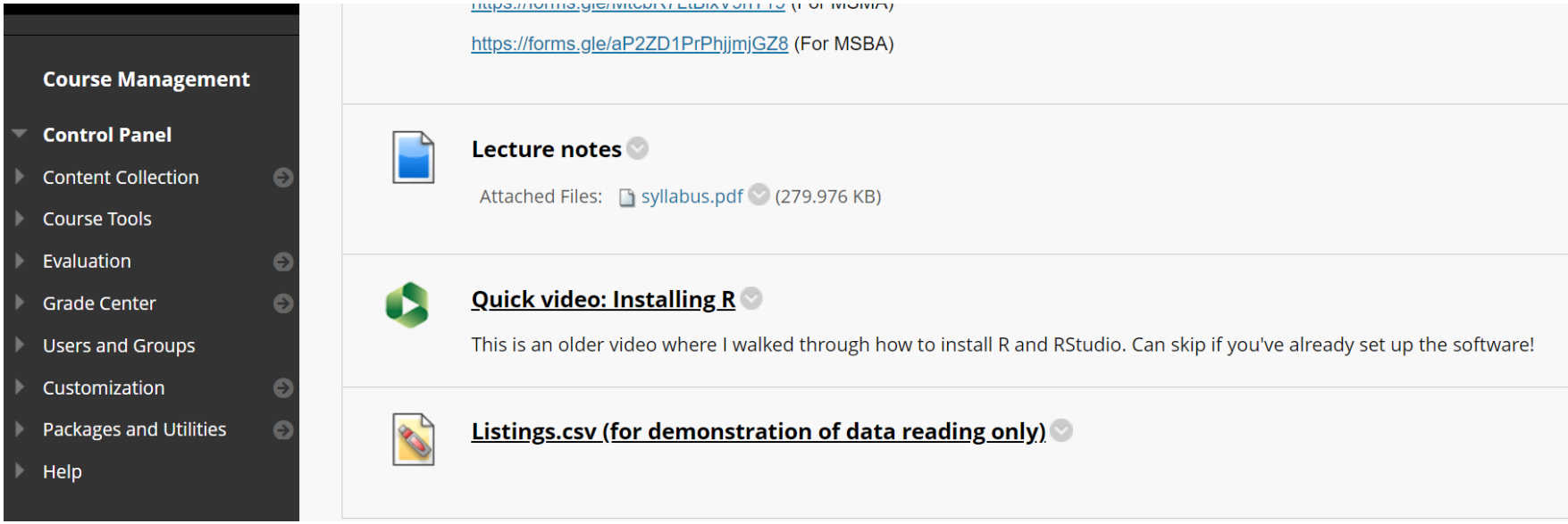
```
# display the product between any two integers below 4
for (i in 1:4) {
    for (j in 1:4) {
        print(i * j)
    }
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 2
## [1] 4
## [1] 6
## [1] 8
## [1] 3
## [1] 6
## [1] 9
## [1] 12
## [1] 4
## [1] 8
## [1] 12
## [1] 16
```

- Explanation: for each variable "i" and variable "j", each taking values from 1 to 4, display the value i * j
- This is called a "for-loop", which we systematically study in the second half of the course


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
intro_examples.R
Source on Save
Run
Source
1 # Intro slide examples
2
3 # display the product between any two integers below 10
4 for (i in 1:4) {
5 for (j in 1:4) {
6 print(i * j)
7 }
8 }
9
9:1
(Top Level)
R Script
Console L:/Dropbox/Teaching/Programming_for_analytics/2018/lecture_1/
> # Intro slide examples
>
> # display the product between any two integers below 10
> for (i in 1:4) {
+ for (j in 1:4) {
+ print(i * j)
+ }
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 2
[1] 4
[1] 6
[1] 8
[1] 3
[1] 6
[1] 9
[1] 12
[1] 4
[1] 8
[1] 12
[1] 16
>

Example 2: Reading a dataset



- A tempting thing to do when you download a .csv (comma separated variable file format) file is to double click and open it with Excel
- How long does that take?

Reading a moderate-sized dataset

```
# standard R
data_1 <- read.csv('listings.csv')
```

```
## Time difference of 7.324047 secs
```

- Already way faster than Excel, do you agree?

Reading a moderate-sized dataset: faster functions

```
# data table way
library('data.table')   # need to install the package first
data_2 <- fread('listings.csv')
```

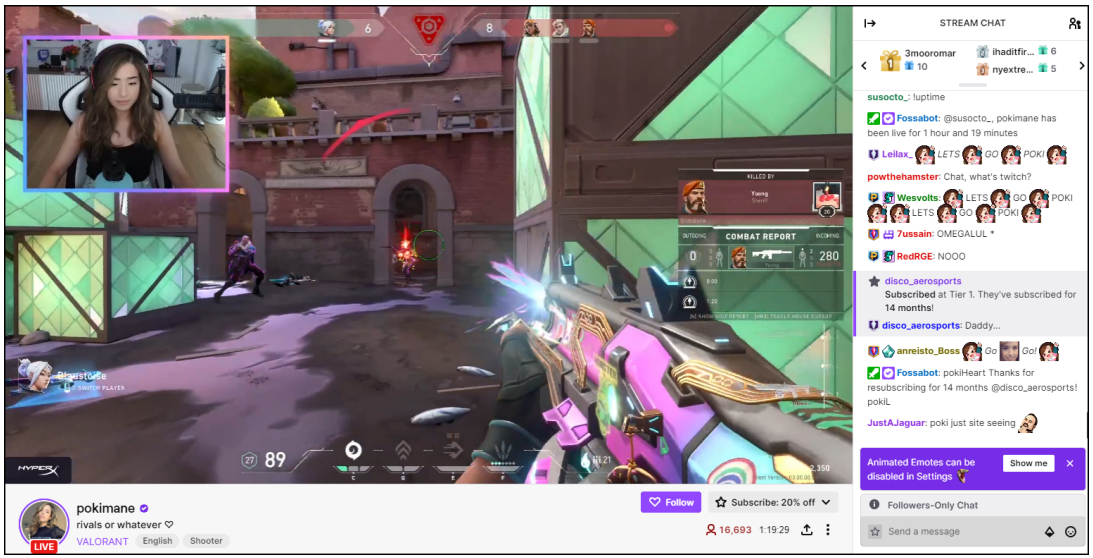
```
## Time difference of 1.874187 secs
```

- R is constant developing; e.g. the *data.table* package allows way faster reading and data processing speed
- R is free both in the sense of "free speech" and in the sense of "free beer"

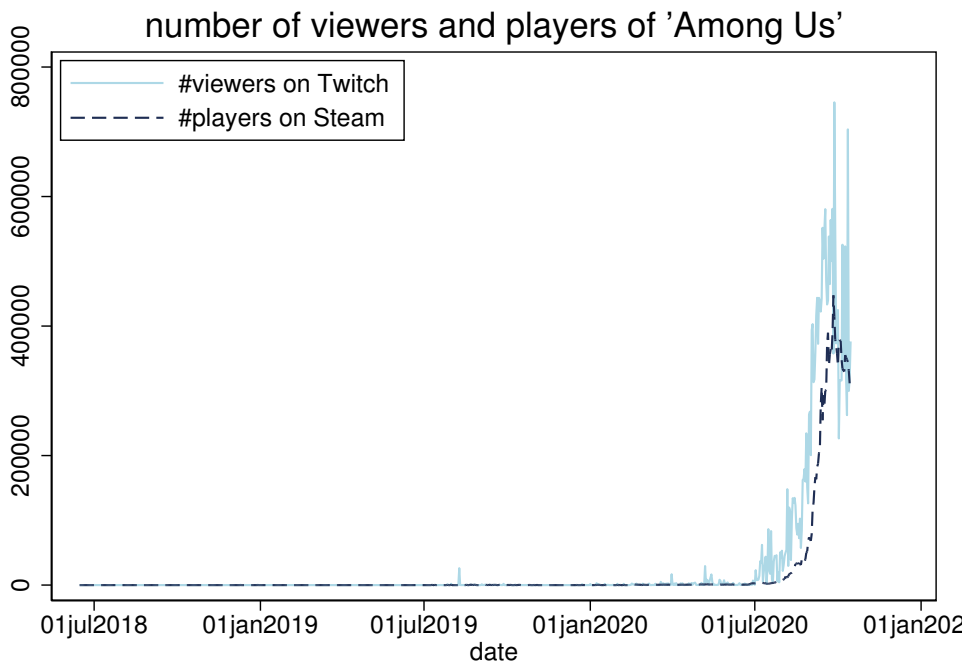
“Less-trivial” examples

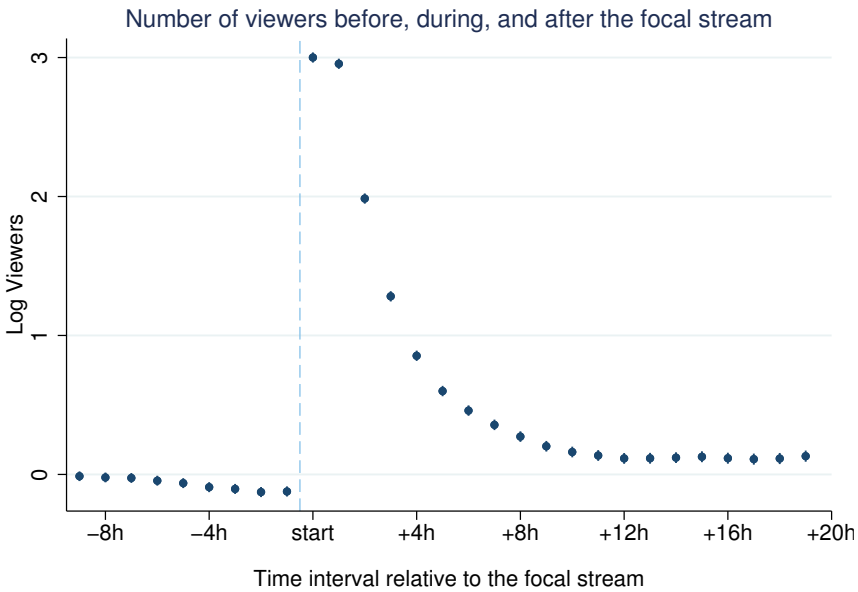
Example 3: one of my recent projects

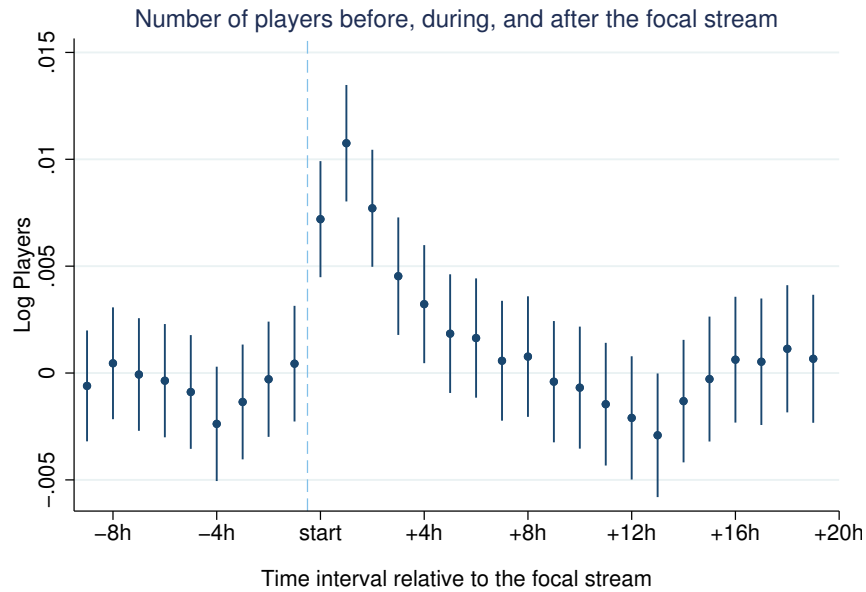
- Are Twitch streams useful to promote the product?

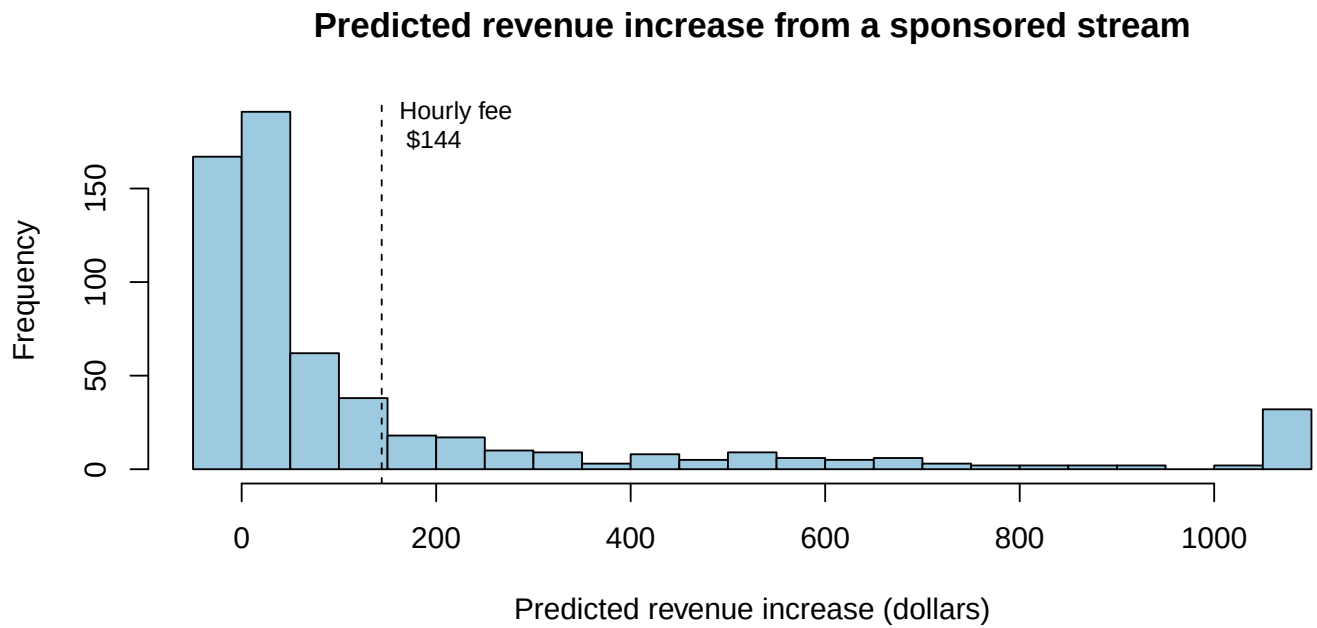


Difficulty: game trending up *because of* or *lead to* Twitch streams?

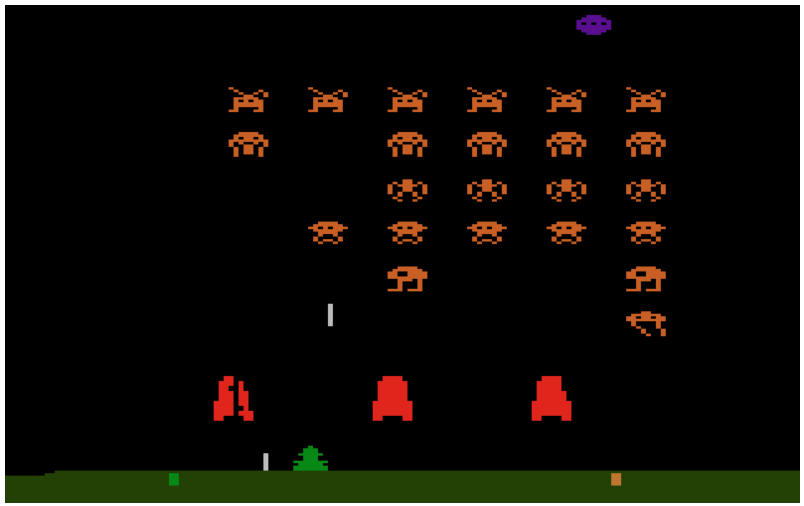



Number of viewers before, during, and after the focal stream
Log Viewers
0
1
2
3
−8h
−4h
start
+4h
+8h
+12h
+16h
+20h
Time interval relative to the focal stream

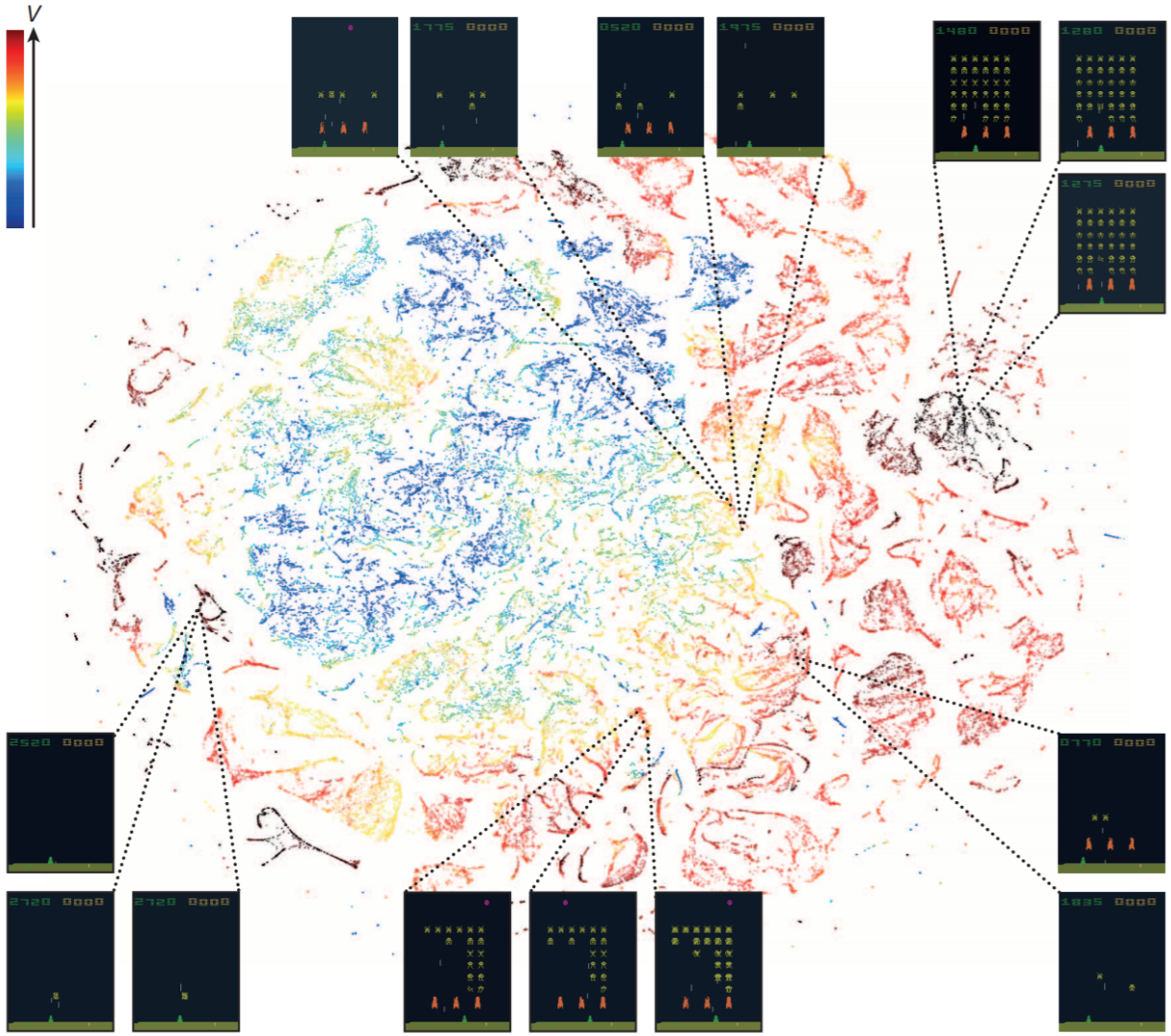

Number of players before, during, and after the focal stream
Log Players
−.005
0
.005
.01
.015
−8h
−4h
start
+4h
+8h
+12h
+16h
+20h
Time interval relative to the focal stream


Predicted revenue increase from a sponsored stream
Hourly fee
$144
Frequency
0
50
100
150
0
200
400
600
800
1000
Predicted revenue increase (dollars)

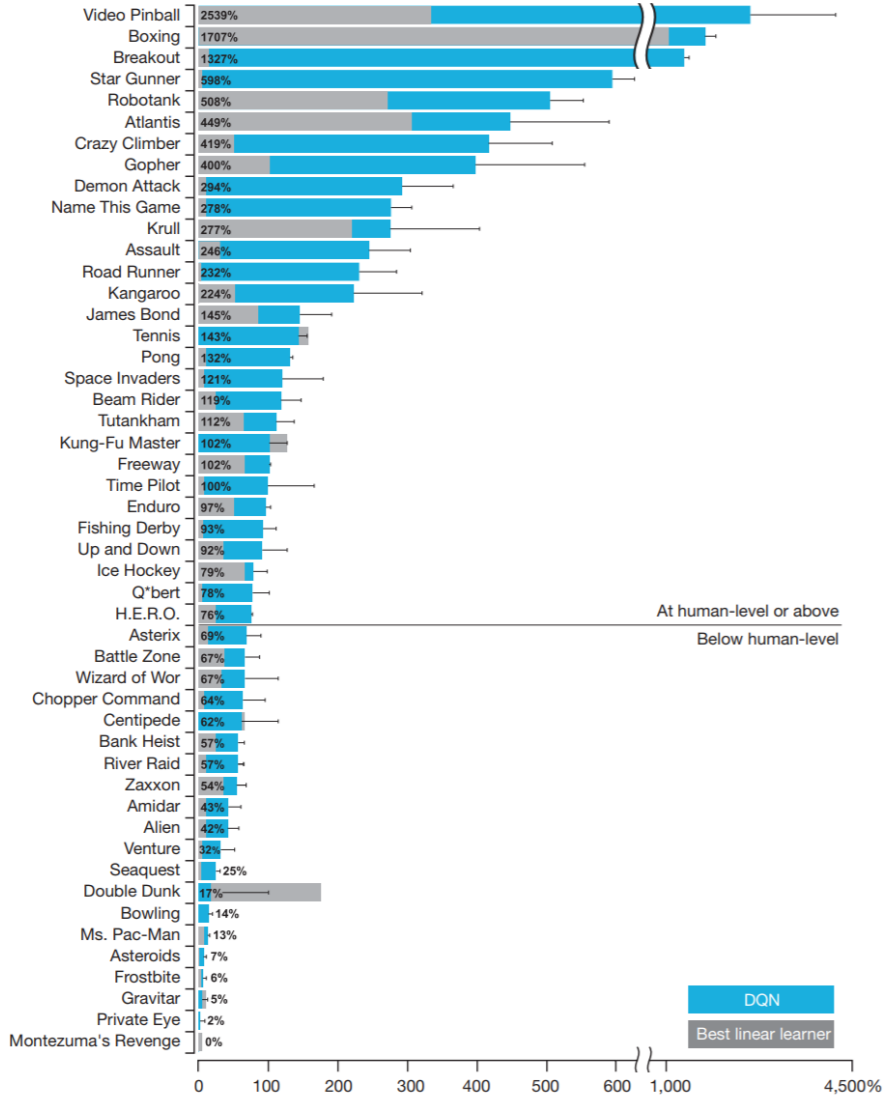
Example 4: Artificial intelligence playing Atari games



- (Google) Deep Mind trained an AI to play Atari 2600 games, including "Space Invader"
  - if you're curious, see Mnih and coauthors (2015), "Human-level control through deep reinforcement learning", Nature
- Common mis-understanding: you teach the AI rules of the game
  - no, you don't; the AI "figures out" the best way to play the game
  - the same algorithm across all Atari games and later behind Alpha Go



- "Value function", or V, represents the AI's prediction of the total score, from looking at the current sceen
  - V is high when e.g. there are few enemies left (bottom-left)



- Performance improvement over a human player: AI does not do everything well yet, but does well in many games

- In 2019, Google DeepMind's *Alpha Star* beat a professional Starcraft 2 player 10 games in a row

Summary

- Intro of me, you, and the class
- Some "inspirational" examples, all of which you'll get your hands on at some point (yes, I'm sure about that)
- Homework for today:
  - install R and RStudio, see video on Blackboard if necessary