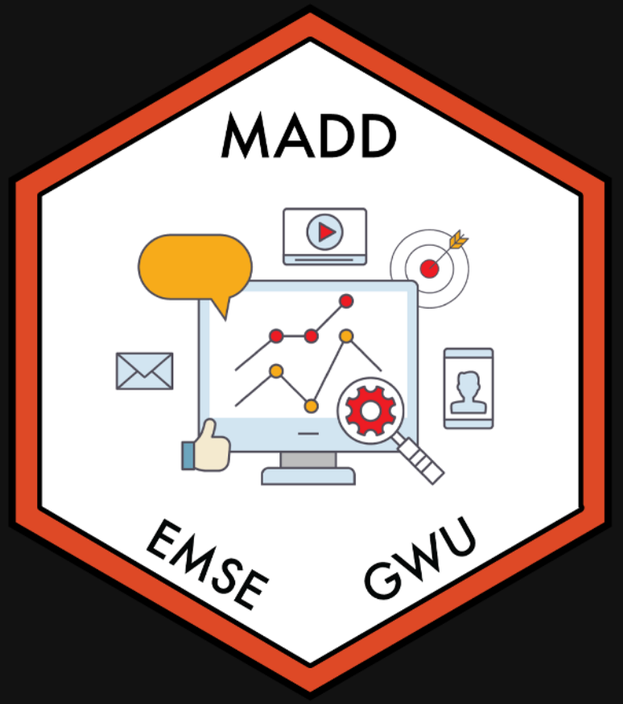


Getting Started

EMSE 6035: Marketing Analytics for Design Decisions

John Paul Helveston

August 26, 2026

Week 1: *Getting Started*

1. Course orientation
2. Intro to conjoint analysis
3. Introductions
4. Getting started with R & Positron

Week 1: *Getting Started*

1. Course orientation
2. Intro to conjoint analysis
3. Introductions
4. Getting started with R & Positron

Meet your instructor!

John Helveston, Ph.D.

- 2025 - Present: Associate Professor, EMSE
- 2018 - 2025: Assistant Professor, EMSE
- 2016-2018: Postdoc at Institute for Sustainable Energy, Boston University
- 2016: PhD in Engineering & Public Policy at Carnegie Mellon University
- 2015: MS in Engineering & Public Policy at Carnegie Mellon University
- 2010: BS in Engineering Science & Mechanics at Virginia Tech
- Website: www.jhelvy.com

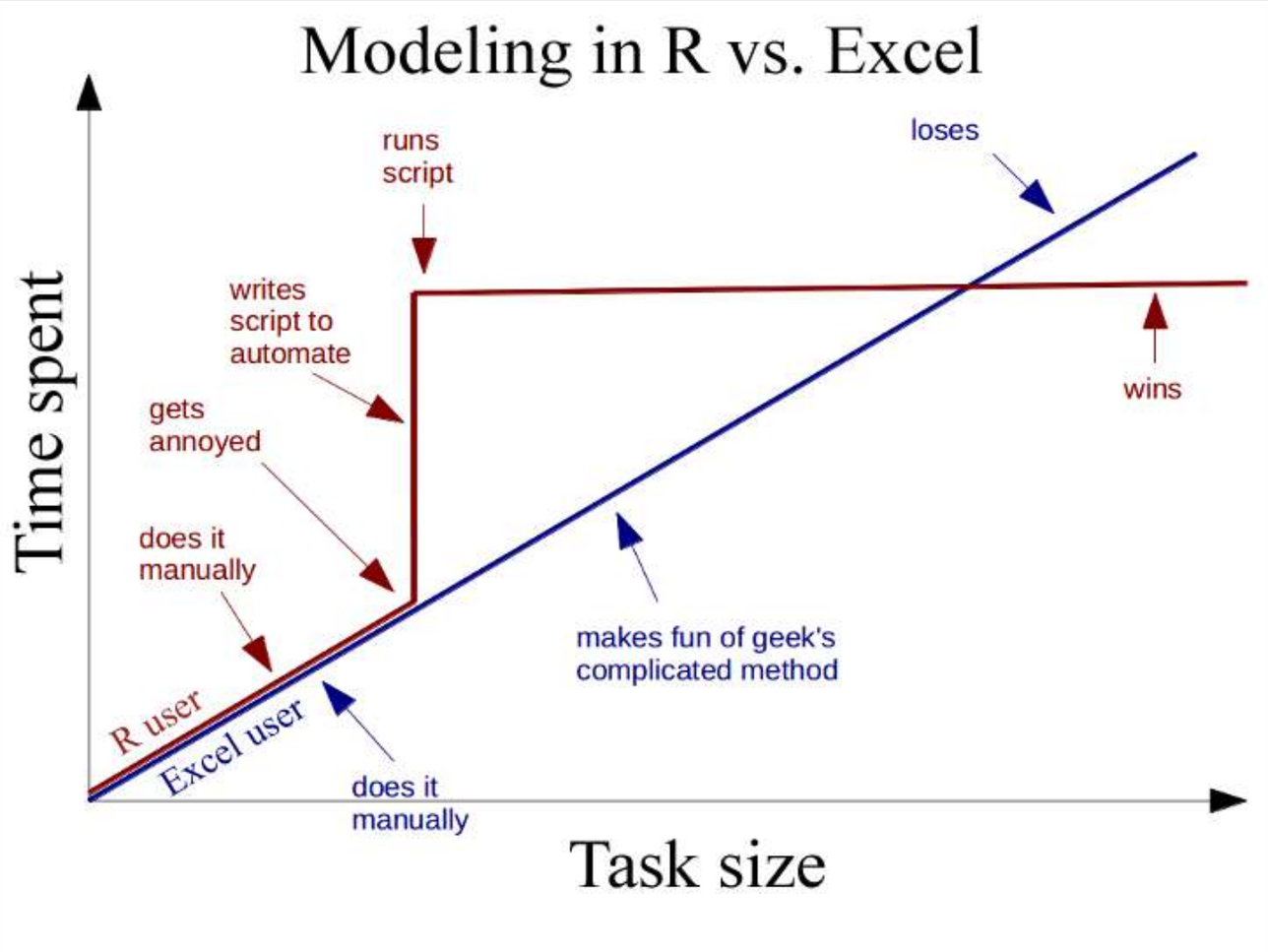
Tools

Course website: https://madd.seas.gwu.edu/2026-Fall/

Course slack: https://emse-madd-f26.slack.com

& Positron: Course Software Page

Why R?


Modeling in R vs. Excel
Time spent
Task size
runs script
loses
writes script to automate
wins
gets annoyed
does it manually
makes fun of geek's complicated method
R user
Excel user
does it manually

Wait, why aren’t we using Python?

The Marketing field out of Statistics, not Machine Learning

- Python is a general purpose language developed by **Guido van Rossum**, a computer scientist.
- Unlike R, Python was not originally developed for data analysis.
- Both languages are extremely useful, and **you should probably learn python too**.

Learning Objectives

After this class, you will know how to...

- ...work with data in R
- ...design effective surveys to get rich data
- ...analyze consumer choice data to model consumer preferences
- ...effectively communicate insights
- ...use Git and GitHub as everyday version control and collaboration tools
- ...direct agentic AI tools to write and audit code that supports analyses

Course prerequisites

This course requires prior exposure to:

- Probability theory
- Multivariable calculus
- Linear algebra
- Regression

Not sure?

Take this self assessment

Weekly HW (30% of grade)

Do some readings, recorded lectures, practice problems

Write a short reflection

- ~Every week (13 total)
- Due 11:59pm Monday before class
- ✓ Graded for completion (looking for engagement)

Quizzes (10% of grade)

📅 At the start of class every other week-ish. Make ups only for excused absences (i.e. don't be late).

📅 6 total, lowest dropped

🕒 10 minutes

Why quiz at all? The "retrieval effect" - basically, you have to *practice* remembering things, otherwise your brain won't remember them (see the book "Make It Stick: The Science of Successful Learning")

Exam (10% of grade)

In-person, first 45 minutes of class on 12/2

All hand-written (like a long quiz)

Semester Project (45% of grade)

Teams of 3-4 students

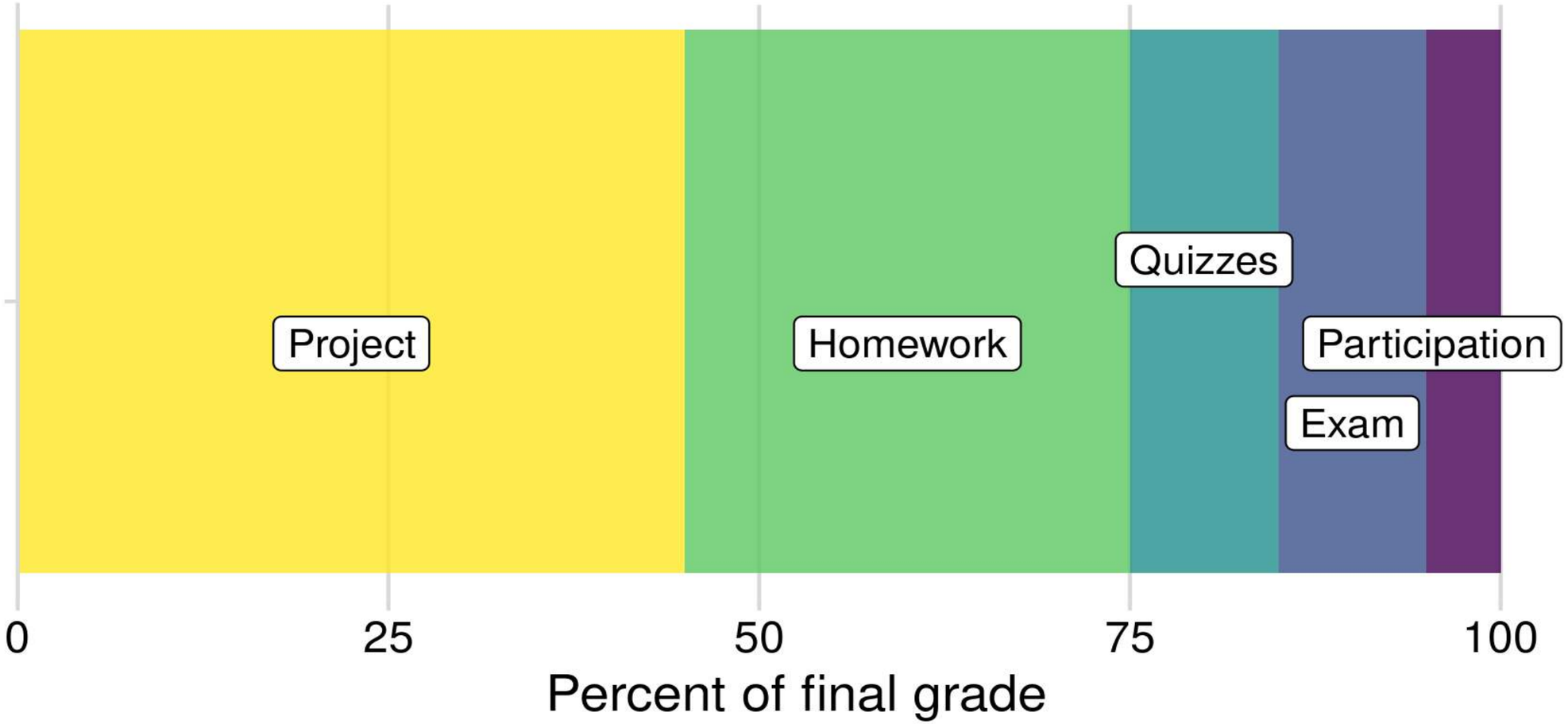
Goals:

- Assess market viability of a new technology or design
- Recommend best design choices for target market or application

Key deliverables:

Item	Weight	Due
Project Proposal	5 %	Sep. 21
Survey Plan	5 %	Sep. 30
Pilot Survey	5 %	Oct. 14
Pilot Analysis	5 %	Nov. 02
Final Survey	5 %	Nov. 16
Final Analysis Report	15 %	Dec. 07
Final Presentation	5 %	Dec. 09

Grades


Project
Homework
Quizzes
Participation
Exam
0
25
50
75
100
Percent of final grade

Grades

Item	Weight	Notes
Participation / Attendance	5%	(Yes, I take attendance)
Homework	30 %	13 assignments (12 × 2.5%, lowest dropped)
Quizzes	10 %	6 quizzes, lowest dropped
Final Exam	10 %	In class, closed book (12/2)
Project Proposal	5 %	Teams of 3-4 students
Survey Plan	5 %	
Pilot Survey	5 %	
Pilot Analysis	5 %	
Final Survey	5 %	
Final Analysis Report	15 %	
Final Presentation	5 %	

Course policies

- BE NICE
- BE HONEST
- DON’T CHEAT

Copying is good, stealing is bad

> “Plagiarism is trying to pass someone else’s work off as your own. Copying is about reverse-engineering.”
>
> – Austin Kleon, from Steal Like An Artist

Use of AI

We will heavily use AI tools this semester

- Large language models (LLMs) are pretty good
- Sometimes they suck.
- I will grade whatever you submit. **It should not suck.**

Ways to not have your work suck:

- Don’t submit code that doesn’t run (actually run it before submitting it).
- Actually read what the AI generates, and **don’t submit something you don’t understand**. (Ask the LLM to explain *why* something works or doesn’t work)

Use the approach I teach: agentic workflows (starting next week)

Late submissions

- **3** late days - use them anytime, no questions asked
- No more than **2** late days on any one assignment
- Contact me for special cases

How to succeed in this class

 Participate during class!

 Start assignments early and **read carefully**!

 Get sleep and take breaks often!

 Ask for help!

Getting Help

Use Slack to ask questions.

Schedule a meeting w/Prof. Helveston:

- Mondays from 8:00-4:30pm
- Tuesdays from 8:00-4:30pm
- Fridays from 8:00-4:00pm

GW Coders

Week 1: *Getting Started*

1. Course orientation
2. Intro to conjoint analysis
3. Introductions
4. Getting started with R & Positron

Engineers often design things nobody wants!

We want to answers to questions like...

- Higher prices decrease demand, but by how much?
- How much more is a consumer willing to pay for increased performance in X?
- How will my product compete against competitors in the market?

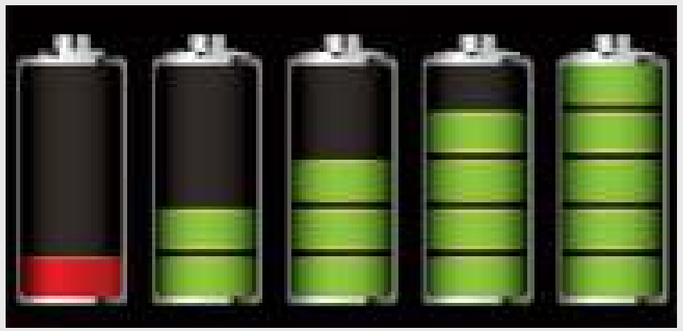
Answers depend on knowing what people want

Directly asking people what they want isn’t always helpful

(People want everything)

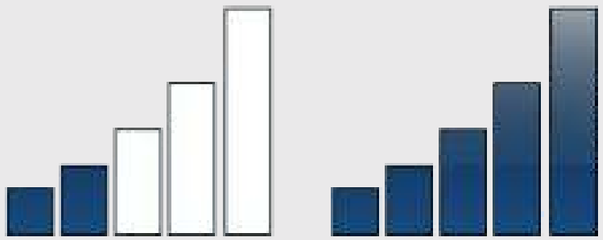
Which feature do you care more about?

Battery Life?

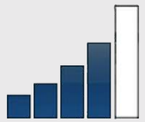
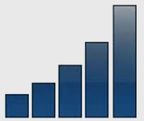
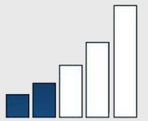


Brand?

Signal quality?



Conjoint approach:
Use consumer choice data to model preferences

Attribute	Phone 1	Phone 2	Phone 3
Price	$400	$450	$350
Brand			
Battery Life			
Signal Quality			
N chosen:	350	250	400

Use random utility framework to predict probability of choosing phone *j*

1. $u_j = \beta_1 \text{price}_j + \beta_2 \text{brand}_j + \beta_3 \text{battery}_j + \beta_4 \text{signal}_j + \varepsilon_j$
2. Assume $\varepsilon_j \sim$ iid extreme value
3. Probability of choosing phone *j*: $P_j = \frac{e^{\beta' x_j}}{\sum_k^J e^{\beta' x_k}}$
4. Estimate $\beta_1, \beta_2, \beta_3, \beta_4$ by minimizing $-L = -\sum_n^N \sum_j^J y_{nj} \ln P_{nj}$

Willingness to Pay

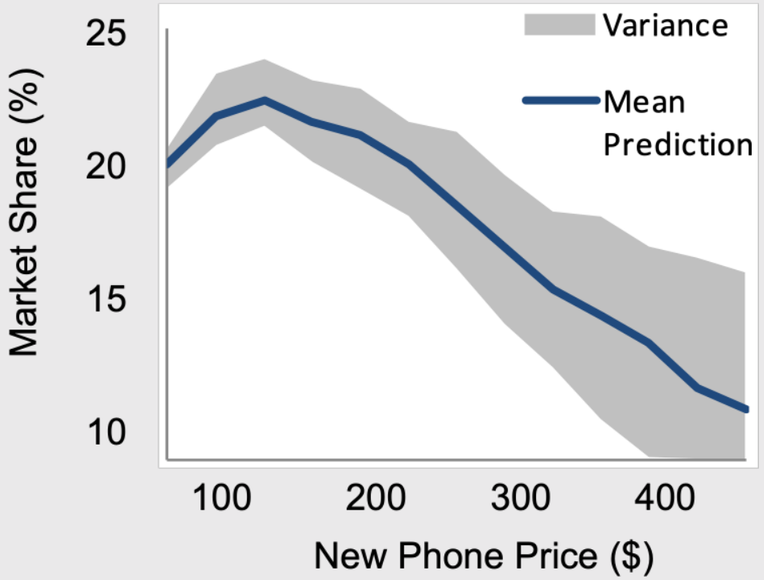
$$u_j = \beta' x_j + \alpha p_j + \varepsilon_j$$

$$\omega = \frac{\beta}{-\alpha}$$

Respondents on average are willing to pay $XX to improve battery life by XX%

Make predictions

$$P_j = \frac{e^{\beta' x_j}}{\sum_k^J e^{\beta' x_k}}$$



Example: *Pocket Charge*

A Flexible, Portable Solar Charger

Product Diagram

Durability

Degradation Protections

Attribute Units

Price– USD
Weight– Kg
Power Output – Watts
Durability – Months
Portability – LxWxH

Power Output

Power Density

Packing Design

Decision Variable Units

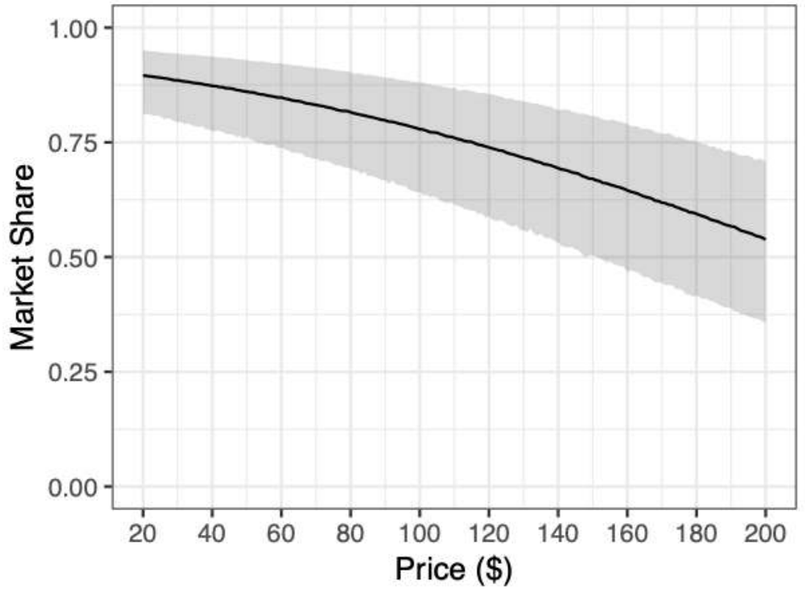
Power Density – W/Kg
Degradation Rate – Hours
Packing Design – Cm^3

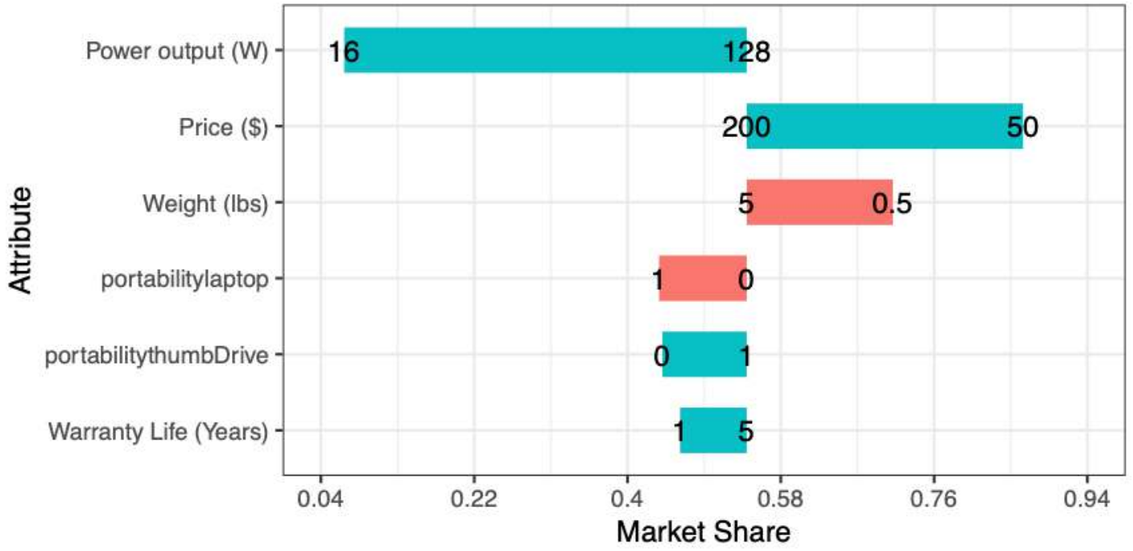
Weight

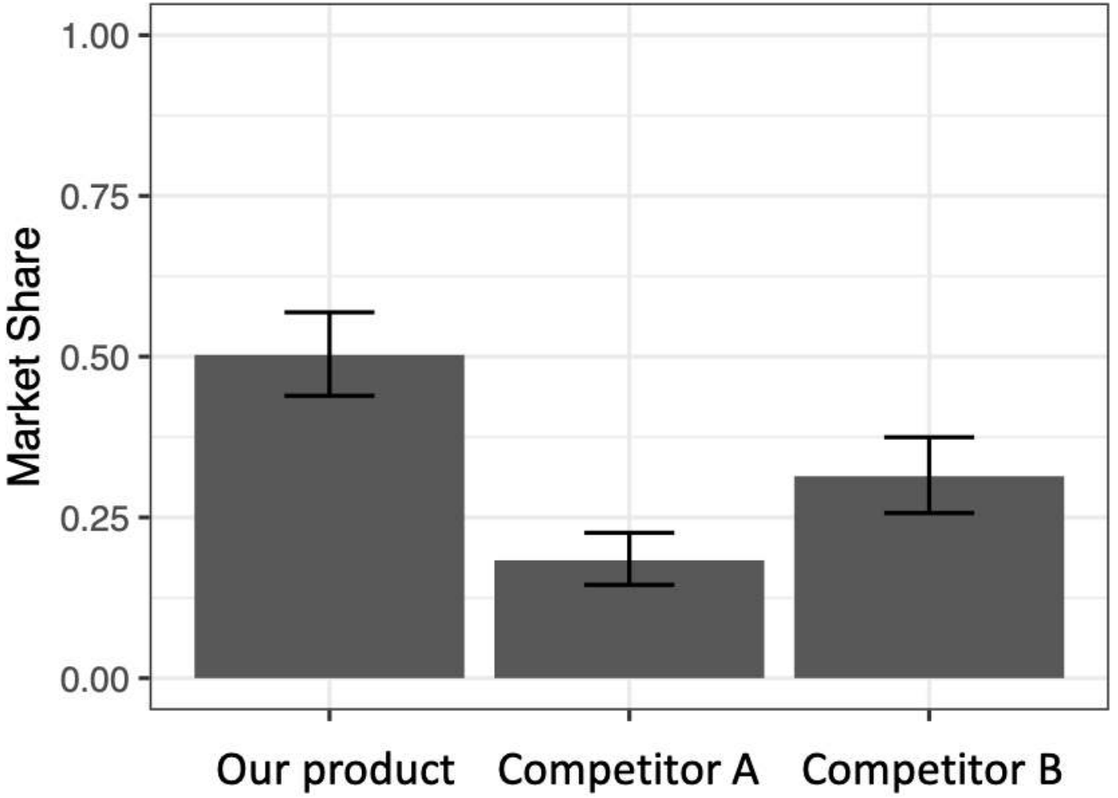
Image Sources:
1.https://www.deviantart.com/kefa321/Art/Rainy-Sun-344719413
2.https://www.pennscale.com/products/ntep-certified-scales/label-printing-scale/
3. https://www.thesolarmonkey.com/portable-foldable-solar-panel-charger/

Example survey choice question

	Choice 1	Choice 2	Choice 3
Price (USD)	**200**	**50**	**100**
Weight (lbs)	**3**	**0.5**	**0.5**
Power Output (Watts)	**16 (One cellphone in 2 hours)**	**16 (One cellphone in 2 hours)**	**16 (One cellphone in 2 hours)**
Years of Manufacturers Warranty	**1**	**1**	**3**
Portability	**Compacted size of a thumb drive**	**Compacted size of a deck of cards**	**Compacted size of a box fan**
	○	○	○


1.00
0.75
0.50
0.25
0.00
Market Share
20
40
60
80
100
120
140
160
180
200
Price ($)


Power output (W)
16
128
Price ($)
200
50
Weight (lbs)
5
0.5
portabilitylaptop
1
0
portabilitythumbDrive
0
1
Warranty Life (Years)
1
5
Attribute
0.04
0.22
0.4
0.58
0.76
0.94
Market Share


1.00
0.75
0.50
0.25
0.00
Market Share
Our product
Competitor A
Competitor B

Your project starts now!

Click here to view project ideas

Week 1: *Getting Started*

1. Course orientation
2. Intro to conjoint analysis
3. Introductions
4. Getting started with R & Positron

Introduce yourself

- Preferred name
- Degree program
- Prior experience
- What do you hope to gain from this class?
- Project interests?

Break

1. If you haven’t already, install everything on the software page
2. Add your **GitHub username** to this sheet (you need this to get your course repo!)
3. Stand up, meet each other, (maybe form teams?...use this sheet)

05:00

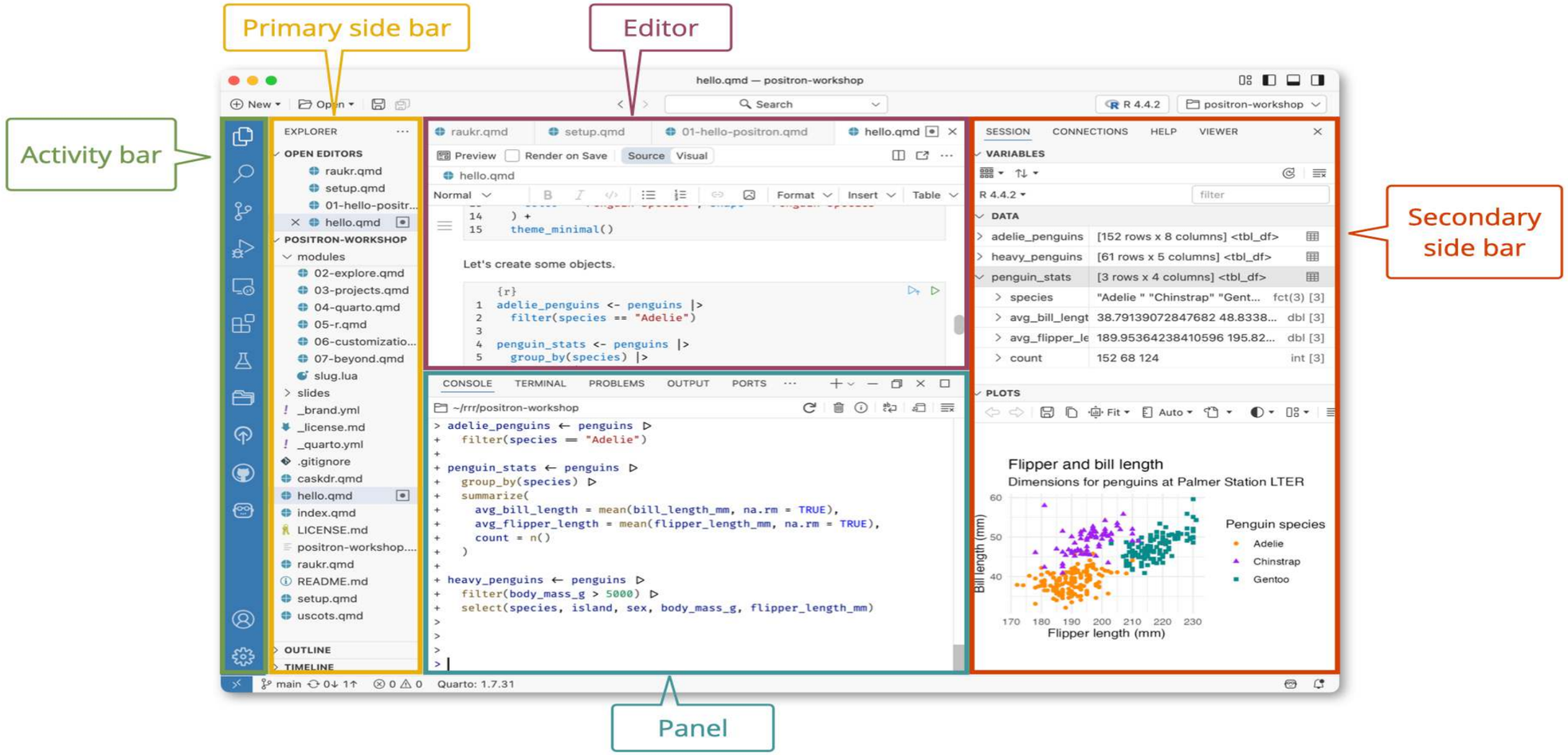
Week 1: *Getting Started*

1. Course orientation
2. Intro to conjoint analysis
3. Introductions
4. Getting started with R & Positron

This semester we use Positron

(not RStudio)

Positron Orientation



Open `intro-to-R.R` file and follow along

Staying organized

1) Save your code in .R files

2) Keep all related work in folders

3) Always open your folder from Positron

Your turn

A. Practice getting organized

1. Make a new folder called `week1`, then open it in Positron (**File > Open Folder...**).
2. Create a new R script and save it as `practice.R`.
3. Open the `practice.R` file and write your answers to these questions in it.

10:00

B. Creating & working with objects

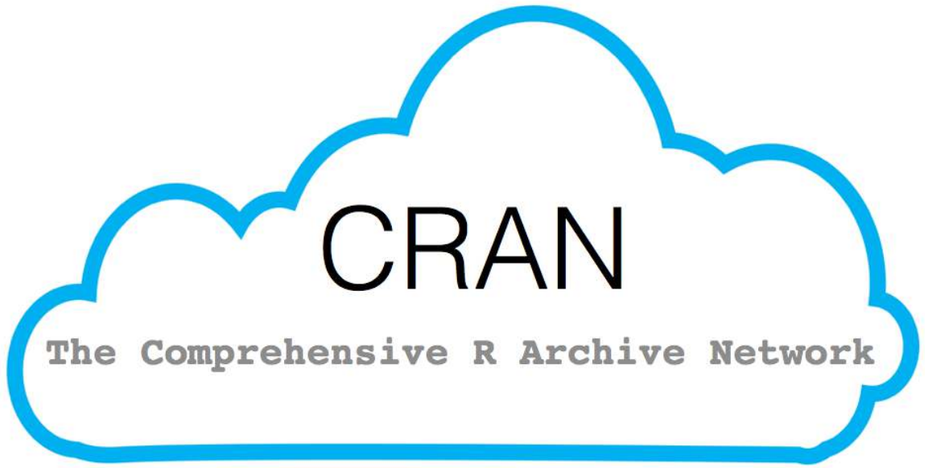
1). Create objects to store the values in this table:

City	Area (sq. mi.)	Population (thousands)
San Francisco, CA	47	884
Chicago, IL	228	2,716
Washington, DC	61	694

2. Using the objects you created, answer the following questions:

- Which city has the highest density?
- How many *more* people would need to live in DC for it to have the same population density as San Francisco?

>20,000 packages on the CRAN



Installing packages

`install.packages("packagename")`

(The package name **must** be in quotes)

```
1 install.packages("packagename") # This works
2 install.packages(packagename) # This doesn't work
```

You only need to install a package once!

Loading packages

`library(packagename)`: Loads all the functions in a package
(The package name *doesn't* need to be in quotes)

```
1 library("packagename") # This works
2 library(packagename) # This also works
```

You need to *load* the package every time you use it!

Installing vs. Loading

INSTALL ONCE:

```
install.packages("light")
```

USE MANY TIMES:

```
library("light")
```

Example: **wikifacts**

Install the Wikifacts package, by Keith McNulty:

```
1 install.packages("wikifacts")
```

Load the package:

```
1 library(wikifacts) # Load the library
```

Use one of the package functions

```
1 wiki_randomfact()
```

```
#> [1] "Did you know that the Romans built a temple possibly dedicated to a boar
```

Example: **wikifacts**

Now, restart your R session in Positron: > Session → Restart R

Try using the package function again:

```
1 wiki_randomfact()
```

```
#> Error in `wiki_randomfact()`:
#> ! could not find function "wiki_randomfact"
```

Using only *some* package functions

You don’t always have to load the whole library.

Functions can be accessed with this pattern:

```
packagename::functionname()
```

```
1 wikifacts::wiki_randomfact()
```

```
#> [1] "Here's some news from 05 May 2026. The Antigua and Barbuda Labour Party,
```

If you haven’t yet, install these packages

Back `intro-to-R.R` for the rest of class!