wk7_1

November 11, 2025

0.1 Attendance code for today: 40872913

1 Where are we? Where are we going?

Python

- Arrays
- Loops
- Dataframes

Next: If statements (making decisions)

Data concepts

- simulation - sampling from a simulated world
  - Simulate one boy/girl child (50% chance of each)
  - Simulate one Swain juror (26% black, 74% white)
  - Simulate union and non-union workers at Fiat
- Take a sample by repeating
  - Y children
  - 12 jurors
  - The number of workers re-hired
- Simulate the 'no difference'/'fair' world
  - Boys and girls are equally likely
  - Jurors are picked according to their percentage in the community
  - Union and non-union workers are picked according to their percentage in the workforce
- Compare the **range** of results (a distribution) to the **actual** outcome
  - Did Henry VIII have an unusual number of female children?
  - Was the Swain jury selection fair?
  - Did Fiat discriminate against union workers?
  - **Visualize** the distribution from the 'fair' world and compare to the single real-world value
  - Was the real value unusual in the fair world?
- Calculate the probability of the real outcome in the fair world
  - Count the number of **simulated** juries with 0 black jurors
  - Divide by the total number of juries
  - Count the number of times 2 **or fewer** boys occur in **simulated** families with 15 children
  - Divide by the total number of families simulated

1.0.1 The concepts here are very important. We will repeat them in the rest of the course

1.0.2 Mid-module review results

- What you liked
  - Exercises on vlab
  - Interactive sessions (live coding)
  - Resources and textbook
  - The pace
- What you didn't like
  - Time to set up at the beginning of lectures
  - Finding a group for the project
  - The rooms are hard to do interactive work in
  - The pace (too slow and too fast)
  - Talking in class
  - Not fully understanding or feeling lost
- Suggestions
  - Interactive activities
  - Pre-allocate project groups
  - More challenging content for more advanced students
  - More practicing in lectures
  - Less lecture, more workshop
  - Drop-in sessions

1.0.3 What can we change?

- More interactive sessions - less lecture/more workshop
  - Slower, so more material will need to be done independently outside of class
  - Need good ways to interact (vevox?)
  - Some working time, some lecture
- Project groups
  - For people who want to be assigned I can assign some groups
  - Send me a request to be assigned
- Setup time
  - Computer problems - hopefully solved this week?
  - 6-8 weeks to replace a computer
- Challenging material
  - Material gets more challenging now
  - Projects can take on challenge
  - Increasing work will be necessary on projects
- Drop in sessions
  - There will be more
  - NOTE HOWEVER - 1st drop-in had 4 teaching staff (me + 3 TA) and 2 students
    - This did not indicate a high level of **actual** interest
- Talking in class
  - Balance between interactivity and lecture
  - BUT when we are doing lecture, need to be aware of other students
- I will cover fewer historical data scientists, but not eliminate all together

- More information about the project format and marking criteria will come
  - We have presented basic information
  - More details will come shortly
  - I didn't ask about this, but the other survey will have done.

1.0.4 What can't we change

- The room
  - Computer rooms are not large enough, except for the CTL
  - The CTL was worse - 2 separate rooms
  - There are benefits to working on your own laptops rather than sitting behind machines
  - Constant challenge as class size has grown
- More sessions
  - Sessions numbers are organized at the beginning and constrained by staffing
- Many individual help sessions
  - I do encourage students to ask me questions
  - I can meet with students
  - I can't provide many individual sessions - the class is too large
  - Note, however, that piazza has been somewhat underutilized
  - Interactivity may help
  - Does vevox help encourage questions?

1.0.5 Pandas

Putting pandas to use

- Selecting data
- Sorting
- Summarizing
- Plotting

1.0.6 Who has more fat? McDonalds or Starbucks

Read in the menu data - the file is: `McD_vs_StarB_menus.csv`

Extract all of the McDonalds values for one category (which column name has 'Calories')?

```
Try Calories. Then you could try Fat, Carbs, Fiber, Protein
Plot the distribution for the McDonalds values
```

Extract all of the Starbucks values for the same category

```
Plot the distribution for the Starbucks values
```

Which fast food provider has higher values?

```
Does the shape of the distribution have things that are important to notice?
```

1.0.7 Read in the data

[ ]:
```
# read in the data here
```

1.0.8 Select the McDonalds fat values

[ ]:
```
# select the McDonalds fat values here
```

1.0.9 Select the Starbucks fat values

[ ]:
```
# select the Starbucks fat values here
```

1.0.10 Plot the two sets of fat values – How do you want the plot to look?

[ ]:
```
# experiment with some plots here
```

1.0.11 What do you notice?

[ ]:
```
# do you want to do something else?
```

1.0.12 How would you test whether one restaurant has a fattier menu?

2 Permutation test: Another simulation test

3 -Mosquitoes and beer-

4 Are you more attractive to malaria mosquitoes after a beer?

4.0.1 (Remember this? - see video from the beginning of class - week 1, session 2)

Mosquitoes and beer experiment

- Put one person in a tent to capture odor from breath and body
- Open the other tent to the outside air
- Draw air from these two sources into boxes at the end of a Y-junction
- Release mosquitos
- Do they go left to the box connected to the person?
- Do they go right to the box connected to outside air?

Experimental design: - Do this once before drinking anything (control condition) - Do this again after drinking either beer or water (experimental condition)

5 Experimental setup

5.1 Data Wrangling

These are the real data. The video presented simplified data that were already pre-processed

Here we give an example of some data understanding and data wrangling problems

Then we do our first permutation test

5.1.1 Data

Data are in ‘mosquito_beer.csv’

Let’s read in the data...

```
[2]: mozzie_data = pd.read_csv("mosquito_beer.csv")
     mozzie_data.info() # shows information about columns and column names
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86 entries, 0 to 85
Data columns (total 12 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   volunteer      86 non-null     object
 1   group          86 non-null     object
 2   test           86 non-null     object
 3   nb_released    86 non-null     int64
 4   no_odour       86 non-null     int64
 5   volunt_odour   86 non-null     int64
 6   activated      86 non-null     int64
 7   co2no          81 non-null     float64
 8   co2od          80 non-null     float64
 9   temp           82 non-null     float64
 10  trapside       86 non-null     object
 11  datetime       86 non-null     object
dtypes: float64(3), int64(4), object(5)
memory usage: 8.2+ KB
```

```
[3]: # we should also check that data seem to have been read in ok
     mozzie_data.head()
```

```
[3]:   volunteer group    test  nb_released  no_odour  volunt_odour  activated  \
     0     subj1  beer  before           50         7             9         16
     1     subj2  beer  before           50        26             7         33
     2     subj3  beer  before           50         5            10         15
     3     subj4  beer  before           50         3             7         10
     4     subj5  beer  before           50         2             8         10

        co2no  co2od  temp trapside             datetime
     0  305.0  321.0  36.1        A  2007-08-28 19:00:00
     1  338.0  720.0  35.3        B  2007-08-28 21:00:00
     2  348.0  355.0  36.1        B  2007-09-15 19:00:00
     3  349.0  437.0  35.6        A  2007-09-25 17:00:00
     4  396.0  475.0  37.0        B  2007-09-25 18:00:00
```

5.1.2 You need to know about what is in each column

consult the source of the data...

- group: beer/water
- test: before/after
- no_odour: air trap
- volunt_odour: person trap
- activated: number of mosquitoes that went towards the traps

5.1.3 What measure are we interested in?

5.1.4 Possible measure

- % of activated mozzies that went toward the volunteer rather than open air
- Take difference between beer/water

```
[4]: # Calculate the % of activated mozzies that went to the volunteer

     # calculate percent
     mozzie_data['percent_act_person'] = mozzie_data['volunt_odour']/
      ↪mozzie_data['activated']

     # Make a data frame with just the values we are interested in.
     # select columns
     new_moz_data = mozzie_data[['volunteer','group','test',
                                 'no_odour','volunt_odour',
                                 'activated','percent_act_person']].copy()
     # check the result
     new_moz_data.head()
```

```
[4]:   volunteer group    test  no_odour  volunt_odour  activated  \
     0     subj1  beer  before         7             9         16
     1     subj2  beer  before        26             7         33
     2     subj3  beer  before         5            10         15
     3     subj4  beer  before         3             7         10
     4     subj5  beer  before         2             8         10

        percent_act_person
     0            0.562500
     1            0.212121
     2            0.666667
     3            0.700000
     4            0.800000
```

5.2 What about the before/after column? What is that? Why is it there?

- People may differ in how much they attract mosquitos even without drink
- What can you do about that?
- Remember we would like a single value that tells us about how attractive a person is after they have had either beer or water

5.2.1 If 'drink' changes preference, then % of activated mozzies should increase in the 'after' condition

5.2.2 If 'beer' changes the preference more, then the *increase* should be bigger for beer compared to water

What do we need? - before % for people who will drink water - after % for people who will drink water - calculate the difference (after % - before %) - same for beer people

- Is the before/after difference bigger for beer?

5.2.3 How would you do that?

5.2.4 You may want to investigate the pandas function 'merge'

- Merge is a fundamental database operation
- It is frequently used to put datasets together when there is a column that links them
  - E.g. one dataset has lung cancer by region
  - Another dataset has air pollution by region
  - Region is shared
  - Combine the data so lung cancer and air pollution are present for each region