Create Conda Environment

Download miniconda, install and create an environment for running the demos and mini-projects.

You will need the following libraries in the environment:

Core Scientific Stack

- **python ≥3.9**
- **numpy** – vectorized math, simulations
- **pandas** – tabular data (reaction times, accuracies)
- **matplotlib** – plotting (histograms, error bars, line plots)
- **seaborn** (optional) – prettier statistical visualizations
- **scikit-learn** – logistic regression, confusion matrices

Language Models

Optional, but used in Week 13 (Transformers & Surprisal):

- **transformers** (Hugging Face) – for tokenization + toy GPT-2;

Utilities

- **jupyter** – for interactive coding and project notebooks
- **scipy** (optional) – if you want extra distributions/optimizers

Deep Learning / ML

- **pytorch** – main DL library for feedforward nets, RNNs, LSTMs, CNNs, DQN
- **torchvision** – image transforms, loaders (Week 8/12 augmentation/adversarial)
- **Pillow (PIL)** – synthetic image generation (Week 8/12 shapes/adversarial)

APSY 780: Computational Methods for AI models

Welcome!

Stroop Task

Name the font colour:

congruent	incongruent
Green	Green
Yellow	Yellow
Red	Red
Blue	Blue
Orange	Orange
Purple	Purple

Mini-project 1

Write Python code to:

* Simulate a Stroop-like task with randomized congruent vs. incongruent trials and generate RTs;
* Save the data to a CSV file;
* Plot distributions of RTs;
* Push code to GitHub.

Demo 1: Variables

Write Python code to:

* Create four variables for a participant in an experiment:
  * Participant's Name
  * Participant's Age
  * Participant's Height
  * Whether participant is a student

Dynamic typing

Demo 2: lists

Write Python code to:

* Create a list with reaction times for 10 trials
* Find the average reaction time and print it
* Print the reaction time for the first & last trials
* Print the reaction times for first half of trials
* (Try it) Calculate the median reaction time instead of the mean
* (Try it) Calculate the std dev of reaction times

Demo 3: dictionaries

Write Python code to:

* Create a data structure for each participant.
* For example:

```
Participant 1 {
    ID → xfkgot
    Age → 23
    Condition → ‘Congruent’
}
```

Demo 4: functions

Let us make the code reusable

* Re-write the code for finding the mean as a function

Demo 5: loops and conditions

Write a Python function to go through all RTs and classify them as fast or slow.

* Re-write the code for finding the median as a function

* (Try it) Write a function for finding variance of a given list of reaction times and test it

Demo 6: numpy

Python has a wide range of libraries. Probably the one that we will use the most is numpy

* Create an array of RTs, this time as an numpy array
* Re-write the code for finding mean, median, std using numpy
* Generate RTs randomly from a normal distribution

Demo 7: pandas

A second Python library that is really useful is Pandas, used for data storage, manipulation, and analysis

* Store data from a sample experiment in a CSV file
* Load data from CSV file into a pandas dataframe
* Print the mean for congruent and incongruent conditions

Demo 8: matplotlib

Let us plot the RTs

* Create a time series plot of RTs
* Change the label along x-axis to “Trials”
* Change the label along the y-axis to “RT (ms)”
* Change the title to “RTs across trials”
* Save the plot to a png file

Demo 9: Seaborn

Let us plot the RTs using a different package: seaborn

* Create a time series plot of RTs
* Create a box plot for RTs under the congruent and incongruent conditions

Stroop Task Simulator

Now your turn (in-class **mini-project**):

1. Simulate a Stroop task:

    1. Write a generate_rts() function that generates RTs in ‘congruent’ and ‘incongruent’ conditions from distributions with given parameters

    2. Store these RTs in CSV file

    3. Write a analyse_results() function that reads RTs from CSV file and performs various statistical analyses on it. Visualise the data.

Mini-project extension (Take-home)

Extension

Extend the simulation to multiple participants, each with RTs with different mean and variance. Remove the first N trials as practice trials. Then plot both individual-level Stroop effects and the group average. Bonus: Check if Stroop effect increases or decreases during the experiment.

Expected Output

A set of plots showing within-participant variability and an aggregated group plot, highlighting that Stroop effects are robust across individuals but noisy at the trial level.

Create a GitHub page and upload your data, code and output graphs to the page.