lab3_writeup_1003998757

October 7, 2020

0.1 Lab 3 - Mental Rotation

In this experiment, 54 subjects performed a mental rotation task similar to Shepard & Metzler (1971): multiple pairs of 3D objects were presented at four different rotating angles. Data was obtained from Ganis & Kievit (2015).

The goal is to analyze the relationship between subjects' reaction times (dependent variable) and angles (independent variable) both within subjects [**Tasks 1-3**] and across subjects [**Tasks 4-5**].

Begin your solution as prompted below; everything else was written.

Refer to the lecture slides for the set of functions that you might use for this lab.

This lab must be done **individually**. The required packages have been imported for you below.

[64]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import pickle
from scipy.stats import linregress
```

Total number of subjects in the experiment that you will analyze.

[65]:
```
nsubject = 54
```

The four angles (in degrees) at which 3D objects were presented.

[66]:
```
unique_angles = [0,50,100,150]
```

Create place-holder variable for recording mean response times (see below).

[67]:
```
rt_matrix = np.zeros((nsubject,4))
```

Create place-holder variable for recording slopes (see below).

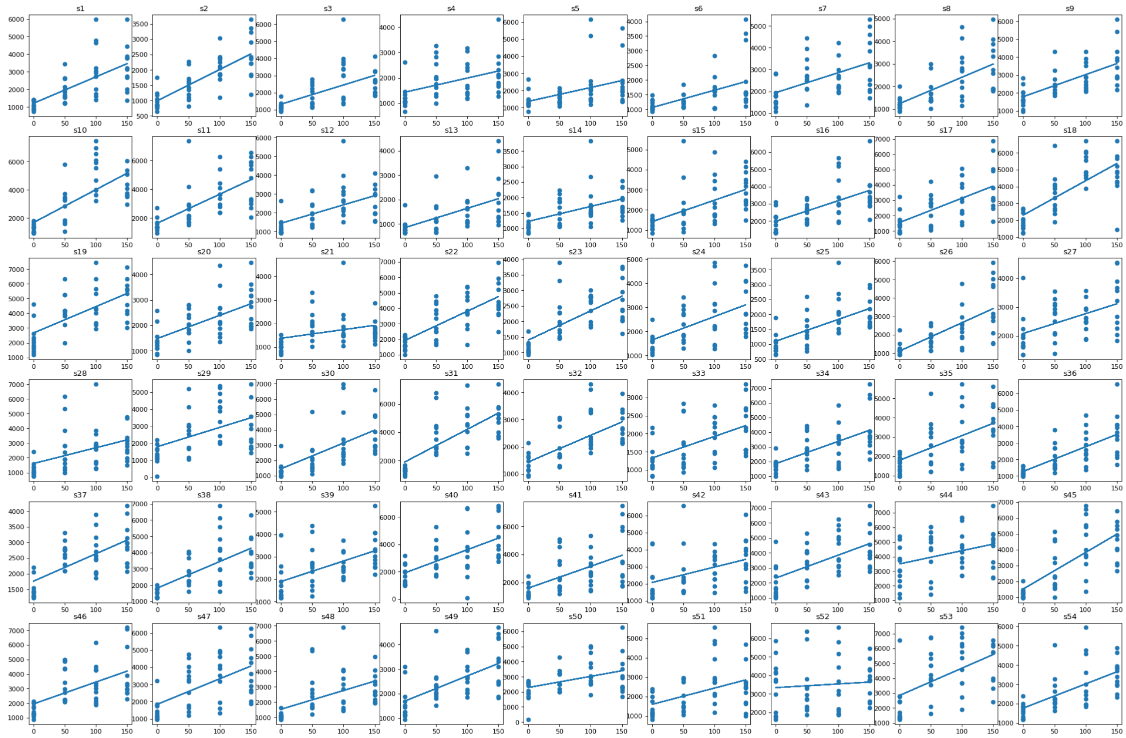
[68]:
```
slopes = np.zeros(nsubject)
```

Create a meta figure that visualizes results for individual subjects, then begin analysis.

[69]:
```
fig = plt.figure(figsize=(30, 20), dpi= 80, facecolor='w', edgecolor='k')
```

```
# Loop over the subjects
for s in range(0,nsubject):

    # Specify the file name of the subject in question
    fn_in = 'data-mental-rotation/sub'+str(s+1)

    # Read in experimental data
    with open(fn_in+'.pickle', 'rb') as f:  # Python 3: open(..., 'wb')
        stimuli, rt, angle, acc, ind_stimuli = pickle.load(f)

    # Focus on trials where stimulus pair is matched
    inds= np.where(ind_stimuli==0)

    # Record the reaction time (rt) for each trial in a vector
    rt = np.float32(rt[inds])
    # Record the angle of the object pair for each trial in a vector
    angle = np.float32(angle[inds])


    #========== Your solution begins here ==========

    # Task 1: Line fitting [3pts]

    #-------Task 1.1-------
    # Fit a line (linear regression) between reaction time (rt) and angle
    # i.e. rt = angle x slope + intercept
    mask = ~np.isnan(rt) & ~np.isnan(angle)
    slope, intercept, r_value, p_value, std_err = linregress(angle[mask],␣
↪rt[mask])
    line = angle * slope + intercept

    #-------Task 1.2-------
    # Record the slope for this subject in place-holder variable "slopes"
    slopes[s] = slope

    # Create a subplot for this subject
    plt.subplot(6,9,s+1)

    # Task 2: Within-subject visualization [2pts]

    #-------Task 2.1-------
    # Scatter plot reaction times (y-axis) against angles (x-axis)
    plt.scatter(angle, rt)

    #-------Task 2.2-------
    # Juxtapose the fitted line onto this scatter plot
    plt.plot(angle, line)
```

```
    # Specify title of the plot by subject index
    plt.title('s'+str(s+1))

    # Task 3: Within-subject statistics [1pt]

    #-------Task 3.1-------
    # Calculate within-subject mean response time across trials, for each angle
    angle_zero = np.nanmean(rt[np.where(angle == unique_angles[0])])
    angle_fifty = np.nanmean(rt[np.where(angle == unique_angles[1])])
    angle_hundred = np.nanmean(rt[np.where(angle == unique_angles[2])])
    angle_onefifty = np.nanmean(rt[np.where(angle == unique_angles[3])])




    #-------Task 3.2-------
    # Record this value in place-holder variable "rt_matrix"
    rt_matrix[s][0] = angle_zero
    rt_matrix[s][1] = angle_fifty
    rt_matrix[s][2] = angle_hundred
    rt_matrix[s][3] = angle_onefifty

plt.show()
```



0.1.1 Task 4: Across-subject visualization [2pts]

Task 4.1: Take the average of mean response times across subjects, for each angle.

[70]:
```
# Write your code here.
avg = rt_matrix.mean(0)
print("Average response time is: {}".format(avg))
```

```
Average response time is: [1537.90277552 2508.85030337 3120.91149676
3353.57024694]
```

Task 4.2: Take the standard deviation of mean response times across subjects, for each angle.

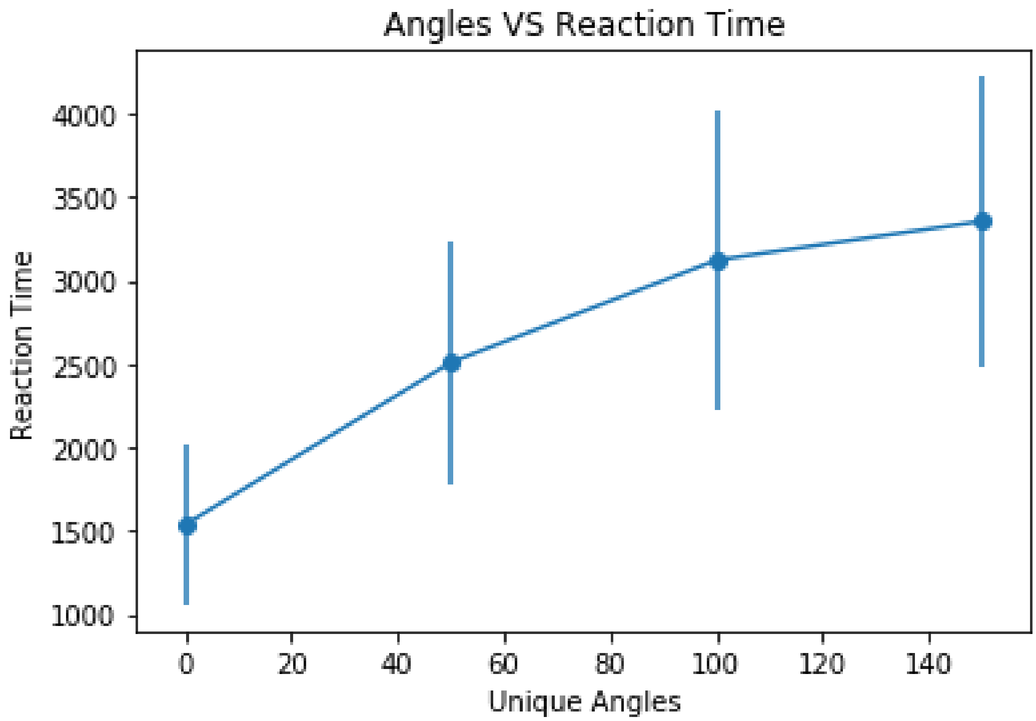
[71]:
```
# Write your code here.
std = rt_matrix.std(0)
print("Standard deviation of response time is: {}".format(std))
```

```
Standard deviation of response time is: [478.46774874 729.36486799 889.53126205
865.22197137]
```

Task 4.3: Plot the average and standard deviation values calculated above, against the four angles.

Task 4.4: Label both axes.

[72]:
```
plt.figure()

# # Write your code here.
plt.errorbar(unique_angles, avg, yerr=std)
plt.title("Angles VS Reaction Time")
plt.xlabel("Unique Angles")
plt.ylabel("Reaction Time")
plt.scatter(unique_angles, avg)

plt.show()
```

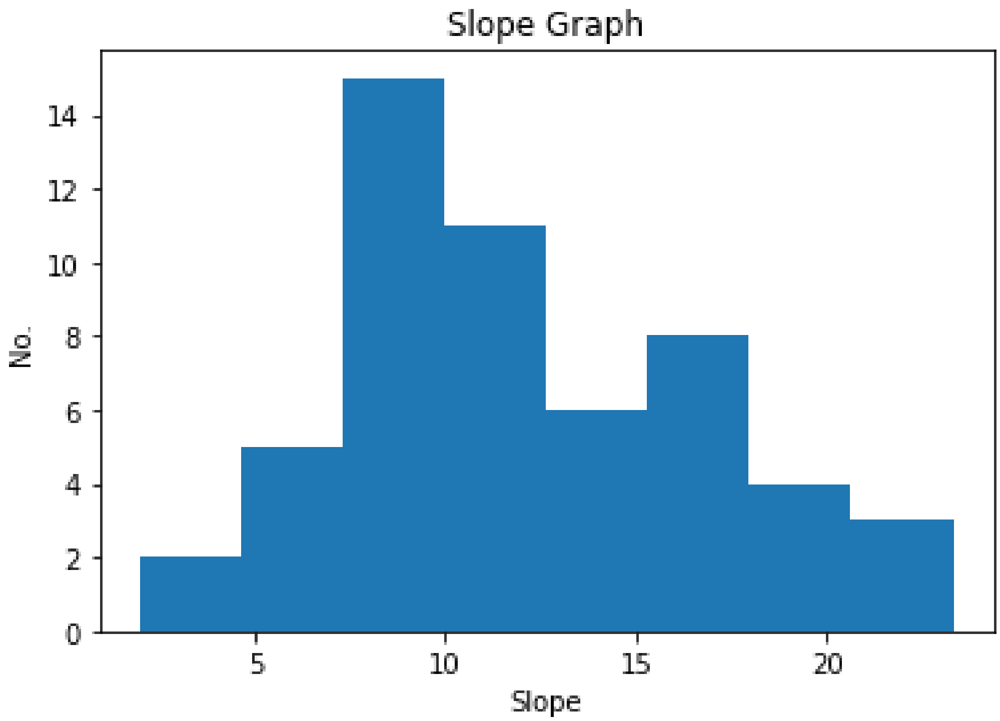


0.1.2 Task 5: Across-subject statistics [2pts]

Task 5.1: Plot a histogram of slopes (across subjects) from the line fits in **Tasks 1.1** and **1.2**.

Task 5.2: Label both axes.

[73]:
```
# Write your code here.
plt.hist(slopes,8)
# bins -> number of bars
plt.title("Slope Graph")
plt.ylabel("No.")
plt.xlabel("Slope")
plt.show()
```



Export and submit a **fully executable** Python Jupyter Notebook and a PDF copy of your notebook showing all results. Please follow the usual naming convention for **both** your Jupyter Notebook and PDF files: `lab3_writeup_YourStudentNumber.pdf`

```
[ ]:
```