A Session to understand Pandas and Seaborn

Saptarishi, Anton

2023-04-21

Briefing

The goal of this session was to understand a python codechunk and recreate it (if possible, better) with minor modifications.

The Problem: Code Chunk

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

sns.set_theme(style="darkgrid",palette="pastel")
df = pd.DataFrame(columns=["RGCs","SC", "LGN", "V1"])
df.loc[0, "RGCs"] = [4.3*10**5, 8.3*10**5,9.3*10**4]
df.loc[0, "SC"] = [6.2*10**5,5.8*10**5,7.6*10**5,5.5*10**5]
df.loc[0, "LGN"] = [1.4*10**5,3.1*10**5,2.5*10**5]
df.loc[0, "V1"] = [1.2*10**6,1.1*10**5,1.8*10**6,5.1*10**6,2.7*10**6,9.3*10**5,2.6*10**6]

plt.figure()

RGCs_mean = np.mean(df["RGCs"][0])
SC_mean = np.mean(df["SC"][0])
LGN_mean = np.mean(df["LGN"][0])
V1_mean = np.mean(df["V1"][0])
RGCs_std = np.std(df["RGCs"][0])
SC_std = np.std(df["SC"][0])
LGN_std = np.std(df["LGN"][0])
V1_std = np.std(df["V1"][0])

plt.bar([0,1,2,3],[RGCs_mean, SC_mean,LGN_mean,V1_mean], width=0.5)
```

```
## <BarContainer object of 4 artists>
```

```
plt.errorbar([0,1,2,3], [RGCs_mean, SC_mean,LGN_mean,V1_mean], yerr=[RGCs_std, SC_std,LGN_std,V1_std], f
```

```
## <ErrorbarContainer object of 3 artists>
```

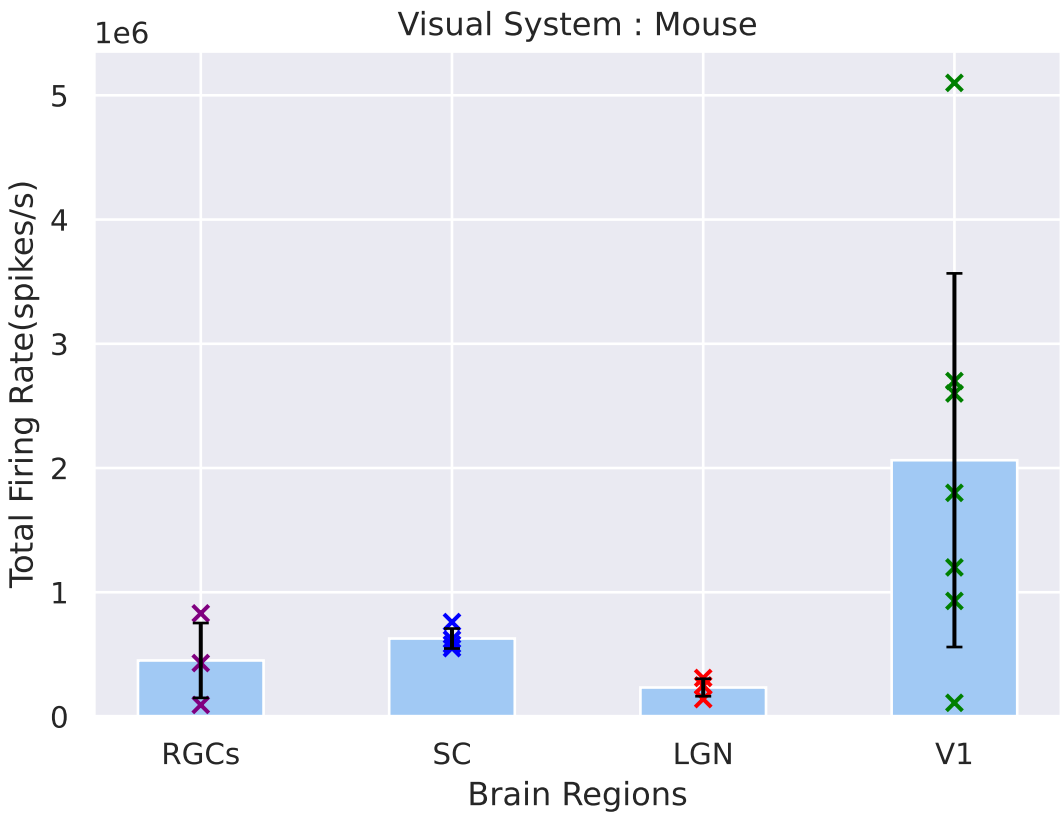
```
plt.scatter([0]*len(df["RGCs"][0]), df["RGCs"][0],color="purple",marker="x")
plt.scatter([1]*len(df["SC"][0]), df["SC"][0], color="blue",marker="x")
plt.scatter([2]*len(df["LGN"][0]), df["LGN"][0], color="red",marker="x")
plt.scatter([3]*len(df["V1"][0]), df["V1"][0], color="green",marker="x")
```

```
plt.xticks([0, 1,2,3], ["RGCs", "SC","LGN","V1"])
```

```
## ([<matplotlib.axis.XTick object at 0x7f655a070250>, <matplotlib.axis.XTick object at 0x7f655a070220>
```

```
plt.xlabel('Brain Regions')
plt.ylabel('Total Firing Rate(spikes/s)')
plt.title("Visual System : Mouse")

plt.show()
```



Our Solution

Note

The problem with the previous dataframe was that the rows for a column were clubbed together in a single cell of the dataframe, which made it very difficult to access the values.

Due to the nature of the data, we considered to construct the dataframe in long format. This was just done to get started, we will try to improve on a more "real" way of gathering or writing the preprocessed data.

Code (Action)

Creating the dataframe df2

First we noted down all the Cell Types and Characters/markers we wanted them to be represented with

```
Types = np.array(["RGC","SC", "LGN","VI"])
Character = np.array(["X","O"])
```

Next, we put this data into a data frame (df2)

```
df2 = pd.DataFrame({
    'Type':np.repeat(Types,[3,4,5,7]),
    'Values':np.array([1,4,77,7,21,14,15,3,24,55,67,14,8,9,5,2,11,14,11])

})
```

Then, since the assignment of markers was very random, we couldn't find an easier way to add it to column of the df2. We thus wrote it down 'by-hand' in a new column of the 'df2'.

```
df2["Character"] = np.array(["X","X","O",
                                        "X","X","X","O",
                                        "X","X","X","X","O",
                                        "X","X","O","O","O","O","O"])
```

Now, looking at the first 5 values of the dataframe - df2 (Just to ensure its alright)

```
df2.head()
```

```
##   Type  Values Character
## 0  RGC       1         X
## 1  RGC       4         X
## 2  RGC      77         O
## 3   SC       7         X
## 4   SC      21         X
```

Finding the mean using groupby() method of pandas dataframe

Now, having constructed the dataframe, we noticed that hte Values column was not numeric. Its got sadly defined as a character. We thus had to convert the column (Values) into numeric using pd.to_numeric()

```
df2['Values_New'] = pd.to_numeric(df2['Values'])
```

Then, we grouped the dataframe values using Type and CHaracter to then find mean of Values_New

```
df2['Mean'] = df2.groupby(['Type', 'Character'])['Values_New'].transform('mean')
```

Similarly we found the standard deviation - std

```
df2['Std'] = df2.groupby(['Type','Character'])['Values_New'].transform('std').fillna(0)
```

Checking the modified df2 with added columns of mean and sd

```
df2.head()
```

```
##   Type  Values Character  Values_New  Mean      Std
## 0  RGC       1         X           1   2.5  2.12132
## 1  RGC       4         X           4   2.5  2.12132
## 2  RGC      77         O          77  77.0  0.00000
## 3   SC       7         X           7  14.0  7.00000
## 4   SC      21         X          21  14.0  7.00000
```

Plotting

Final Plot just using seaborn library

```
plt.clf()
sns.barplot(data = df2, x ="Type", hue= "Character", y ="Values_New")
sns.stripplot(data = df2, x ="Type", hue= "Character", y ="Values_New", dodge = True)

plt.show()
```