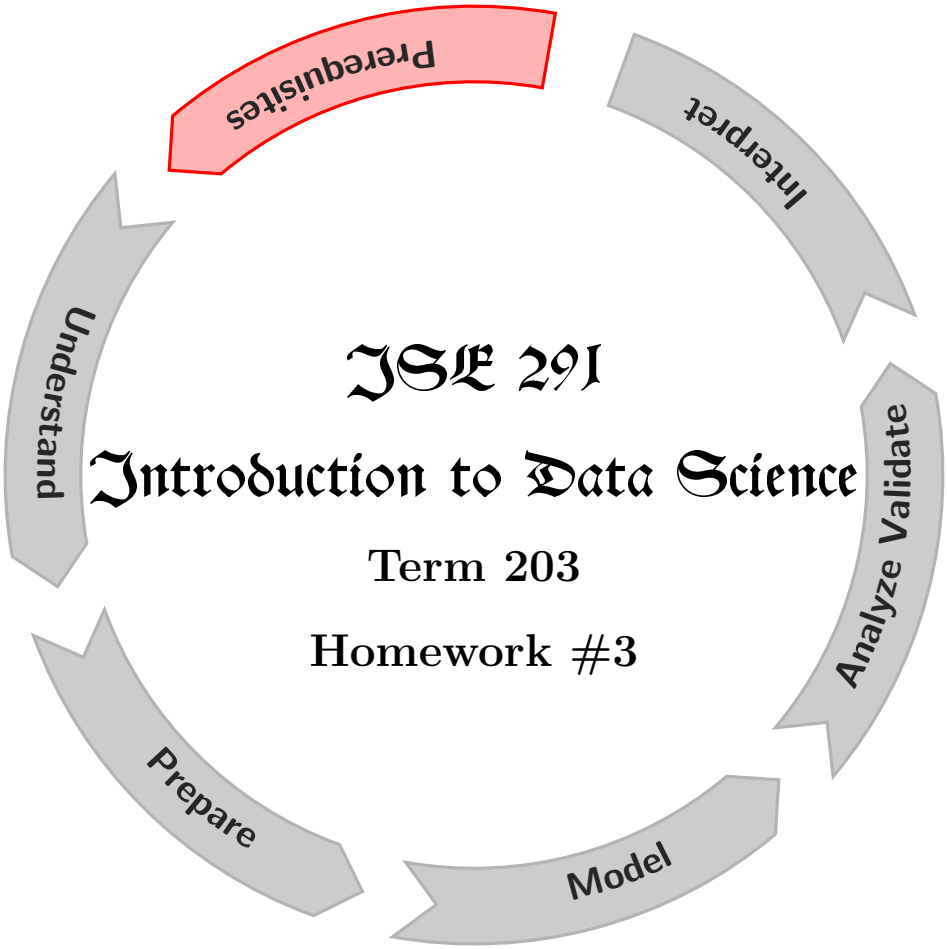
[The HW must be submitted as one .ipynb file. Write names & IDs of all the group members.]



JSE 291

Introduction to Data Science

Term 203

Homework #3

Homework Guidelines

To receive full credit, you should make sure you adhere to the following guidelines. For any questions/-comments contact your section instructor.

Homework Presentation & Submission:

- Every sub-problem (part) should be answered on a DIFFERENT CELL.
- EVERY CELL should have problem and part number clearly written in the first line.
- You should submit the solutions for the FIRST TWO problems only.
- All cells of your homework should be in CHRONOLOGICAL order. One cell per sub-problem.
- Submit entire HW as ONE single .ipynb document.
- ONE HW per group should be submitted.
- Your NAMEs, IDs, and the homework number should be clearly indicated in the FIRST CELL of the notebook.

Problem # A **50 marks**

Consider data given in EXCEL file **HW-3-Data-A** obtained from a public repository[1]. Excel sheet "studentsData" contains the data, and excel sheet "dataDescription" contains the data description.

Do the following tasks using the above data:
☞ *Note: Solve all the following questions using Python. Use* ***Pandas*** *library for all the following analysis.*

A-1: Assume the first row in "studentsData" contains the column headings. Read and display the first 10 rows of data.

A-2: Count the number of rows and columns in the data. Count the number of non-null rows for each column.

A-3: Identify the columns that are numeric. Identify the columns that are non-numeric. Display statistical summaries for all the columns.

A-4: Display the statistical summaries of all the numerical columns, whose rows are related to male students.

A-5: Display the statistical summaries of all the non-numerical columns, where students age is either 15, 17 or 19.

A-6: Display the summary statistics of column "famsize" for those students who score above average in columns G1, G2 and G3, respectively.

A-7: Count the number of students whose one(or both) of the parents job is categorized as teachers, health, or services.

A-8: Select the students whose one(or both) of the parents is teacher, and the parents live together. Display top 10 (sorted by G3 score) such students.

A-9: Modify columns G1, G2 and G3 in-place as follows: Any decimal value ≤ 0.5 should be set to 0.5, and any decimal value > 0.5 should be rounded-up. Display the summary statistics of columns G1, G2 and G3.

HINT: To use **if-condition** with `lambda` function (or in single line), see **Problem # D-14** and **Problem # E** *Code-6*.

A-10: Create a new column called "advising", which contains the following values:

- Any student who has ≤ 10 of "absences" should have a value "normal" in the "advising" column.
- Any student who has > 10 and ≤ 25 of "absences" should have a value "follow-up" in the "advising" column.
- Any student who has > 25 of "absences" and < 3 in health should have a value "medical" in the "advising" column.
- Any student who has > 25 of "absences" and ≥ 3 in health should have a value "concern" in the "advising" column.

Display the value counts for each of the above category.

HINT: `df.apply(funcName,axis=1)` will send entire row as input to the funcName function.

[1] modified data for ISE 291 HW. Details of the data will be provided in the solutions.

Problem #B **50 marks**

Consider the data given in CSV file **HW-3-Data-B**. The data fields/variables has description similar to the data given in **Problem #A**.

Answer the following questions:

☞ *Note: Solve all the following questions using Python (not by hand). Use **Seaborn** libraries for all the following plots. "differentiated by" in the following problem means that either have multiple plots, one plot with different colors, or one plot with sub-plots.*

B-1: Draw the histograms of all numeric and non-numeric columns.

B-2: Draw box plots for **G1** differentiated by **gender**.

B-3: Draw scatter plot between columns **G1** and **G2**, where the size of the marker is based on column **G3**.

B-4: Plot **G3** in ascending order on the x axis, and the corresponding average of **G1** and **G2** on the y axis. Use line plot.

B-5: Display count plots of **Mjob**, differentiated by **famsize**.

B-6: Display violin-plots of **G3**, differentiated by **famsize**.

B-7: Depict **G1** vs **G2** differentiated by **activities** and **internet**. Use relationship-plot.

B-8: Draw relationship-plot of columns **G1** and **G2**, where the size of the marker is based on column **absences**, color/hue is based on **reason**, and plots are separated by **famsize**.

B-9: Draw box plot for **G3** differentiated by **gender** for students that have score in **G3** above 10.

B-10: Display count plots of **Mjob**, differentiated by **famsize** for students that have score in **G3** above 10.

Problem #C (Practice only. No submission required.)

Consider the following python methods, available in naive python, or pandas/seaborn libraries:

C-1: `pandas.DataFrame()`

C-2: `pandas.read_csv()`

C-3: `pandas.DataFrame.head()`

C-4: `pandas.DataFrame.index`

C-5: `pandas.DataFrame.columns`

C-6: `pandas.DataFrame.describe()`

C-7: `pandas.DataFrame.info()`

C-8: `pandas.DataFrame.loc()`

C-9: `pandas.DataFrame.iloc()`

C-10: `pandas.DataFrame.sort_values()`

C-11: `pandas.DataFrame.isin()`

C-12: `pandas.DataFrame.value_counts()`

C-13: `pandas.DataFrame.apply()`

C-14: `pandas.DataFrame.applymap()`

C-15: `seaborn.relplot()`

C-16: `seaborn.pairplot()`

C-17: `seaborn.catplot()`

Answer the following questions for each of the above methods:

- State the purpose/usage of the method/attribute.
- List all the argument of the method.
- Classify the arguments as positional or keyword arguments.
- Write the default values for each of the keyword arguments.

☞ *Note: You must use* ***help()*** *function from python to answer all the above questions.*

Problem #D (Practice only. No submission required.)

Consider the data given in CSV file **HW-3-Data-D**.

Answer the following questions:

D-1: Read the data, and assume that the first row of the file contains the name of the columns.

D-2: Change the names of the header with serialno, gre, toefl, rating, sop, lor, cgpa, research, chance. You can use the function `df.rename`. The changes made should be permanent.

D-3: Display statistical summary for all the columns.

D-4: The CGPA is given using the scale of 0:10, this is not standard practice, convert the cgpa from the scale of 0:4. The change should be permanent.

D-5: Convert the serialno into KFUPM style student ID, assuming every student is from year 2009. So, for example a serial number, 23 will become "200900023". Add this column, with the header 's_id' into the DataFrame.

D-6: After last part, you can drop the serialno column. Use `df.drop` function of DataFrame.

D-7: Based on evaluator's experience, any student who has less than 100 in toefl will not get admission. Display summary of all the students who has score less than 100.

D-8: Similarly, any student, who have weak "Statement of Purpose" (sop) or "Letters of Recommendations" (lor) score will not get the admission. Display summary for those students, whose either score is less than 2.5.

D-9: Display summary of all the students, whose toefl score is greater than 110.

D-10: Compute the average chance (last column) of all students whose toefl score is greater than 110, university rating is 4 or 5 and cgpa is 3.75 or higher.

D-11: Following $D-10$, what proportion of these students have gre score of more than 330.

D-12: Draw separate histograms for each numeric columns.

D-13: Draw separate boxplots for columns, toefl, gre, rating, sop, lor, and cgpa. Are they any outliers?

D-14: Create a column 'accept' which has value of '1' if chance is > 0.8 and '-1' if chance is < 0.5 and 0 otherwise. You may use the lambda function. To have if-condition in lambda function, see the following:

```
In [1]: 1 (lambda x: '+' if x >0 else '-')(4)
        2 (lambda x: '+' if x >0 else '-' if x <0 else 0)(0)
```

D-15: Draw a box plot of the numerical columns differentiated by column 'accept'

D-16: Draw scatter plot of toefl and gre, where the size of the marker is based on chance.

☞ *Note: Solve all the above questions using Python (not by hand). Use* ***Pandas*** *&* ***Seaborn*** *libraries for all the above analysis. "differentiated by" in the above problem means that either have multiple plots, one plot with different colors, or one plot with sub-plots. For the methods highlighted in blue fonts, you may use* `help()` *to know more about the methods and their usage.*

Problem #E (Practice only. No submission required.)

Explain the following *Python* codes. Assume `df` represents an existing pandas' dataframe, where the columns are `C1, c2,...`. The columns with odd numbers are categorical, and columns with even numbers are numerical:

Code-1:

In [1]:
```
display(df.describe())
display(df.describe(include='all'))
display(df.describe(include='object'))
display(df.describe(exclude='number'))
```

Code-2:

In [2]:
```
print(df.head())
print(df.tail())
print(df.sample(5))
print(df.info())
print(df.count())
print(df.shape())
```

Code-3:

In [3]:
```
S1 = df["C1"].isin(["Value 1","Value 2"]) & ((df["C2"]>=10) & (df["C2"]<=20))
S2 = ((df["C1"] == "Value 1") | (df["C1"] == "Value 1")) & df["C2"].between(10,20)
print(S1==S2)
```

Code-4:

In [4]:
```
ndf = df.loc[:,["C1","C2","C3"]]
ndf.set_index("C2",inplace=True)
ndf.sort_index(inplace=True)
ndf.to_csv('data.csv', index = False, header=True)
```

Code-5:

In [5]:
```
df.loc[:,["C1","C3"]]=df.loc[:,["C1","C3"]].applymap(lambda x: x.replace(" ","") )
```

Code-6:

In [6]:
```
def cost(x):
   return x*5 if x> 50 else x if x< 10 else 2*x
df.loc[:,"C2"]=df.loc[:,"C2"].apply(cost)*df.loc[:,"C4"].apply(lambda x: abs(x))
```

Code-7:

In [7]:
```
num_columns = df.select_dtypes(exclude='object').columns
for c in num_columns:
   plt.figure()
   sns.histplot(x=c,bins=10,data=df);
   plt.show()
```