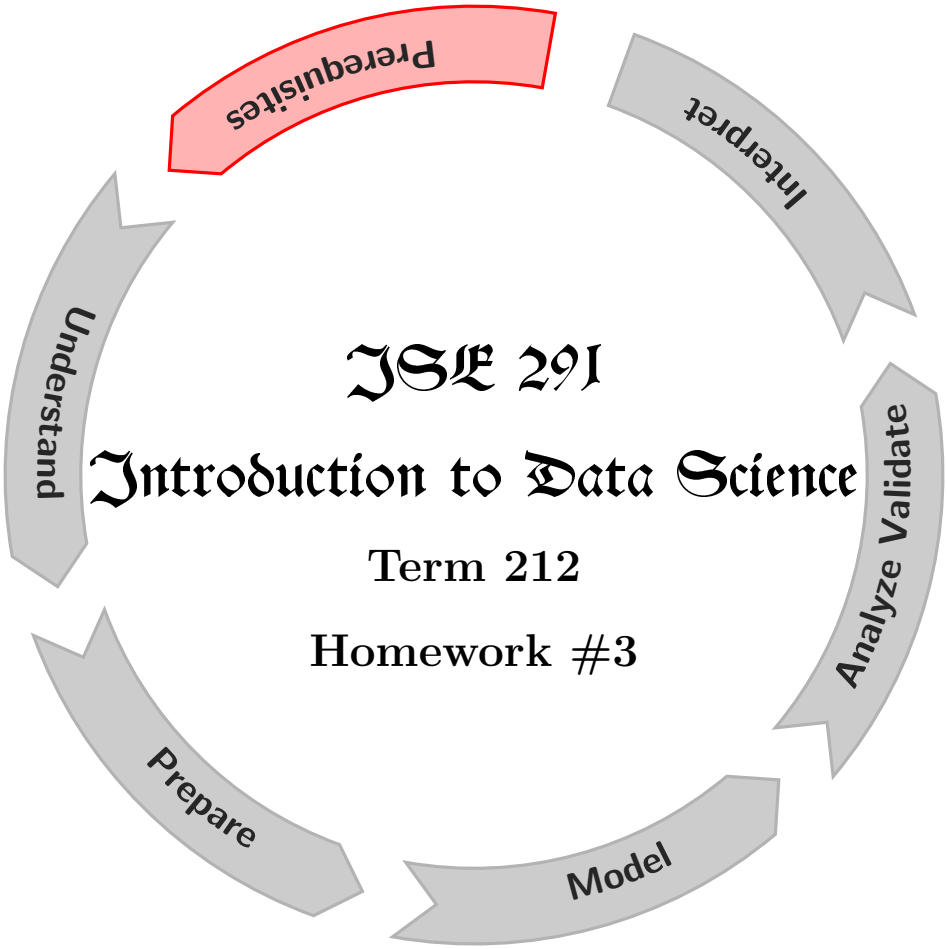
[The HW must be submitted as one .ipynb file. Write name & ID in the provided template.]



JSE 291

Introduction to Data Science

Term 212

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:

- You should submit the solutions for the **FIRST TWO** problems only.
- Every sub-problem (part) should be answered on a DIFFERENT CELL as given in the template.
- EVERY CELL should have problem and part number clearly written in the first line.
- All cells of your homework should be in CHRONOLOGICAL order. One cell per sub-problem.
- Any text should be written as comment in the code cell. Do NOT modify code cell into markdown cell.
- Submit entire HW as ONE single .ipynb document.
- **Do NOT add/delete** any cell in the given template.

Problem # A **50 marks**

Consider data given in EXCEL file **HW3_Data_A** obtained from a public repository[1]. Each row/record corresponds to one car/observation.

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: Read the data (Assume the first row in **HW3_Data_A** file contains the column headings). Then display 15 randomly selected rows.

A-2: Display the names of the column headers and row labels. Next, Count the number of rows and columns in the data. Last, Count the number of null rows for each column.

A-3: Display statistical summaries for numeric columns. Display statistical summaries for non-numeric columns.

A-4: Count the number of rows in the data that corresponds to "toyota" car (the information is available in **make** column) . Display the statistical summaries of all columns, where the column **make** value is "toyota".

A-5: What is the range of **price** column for the records that have **fuel-type** as "gas" and **horsepower** between 100 and 130?

A-6: What is the proportion of cars having "two" doors and **length** greater than or equal to 170.

A-7: Display the 15 cars in the data that has the highest price. Display the statistical summaries of all the non-numerical columns for the selected 15 cars.

A-8: Modify **aspiration** column such that all "std" values should be replaced by "standard". Round to the nearest integer all values in columns: **length, width, height**.

A-9: Add two new columns (**city-kpl** and **highway-kpl**), fill the two columns by converting the corresponding values from **city-mpg** and **highway-mpg** columns from Mile per Gallon (mpg) to Kilometer per Liter (kpl) (Hint: 1mpg=0.425kpl).

[1] modified data for ISE 291 HW.

Problem #B **50 marks**

Consider the data given in CSV file **HW3_Data_B**. Do the following tasks using the above data:.

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: Using the histograms generated in *B*-1, provide descriptive comments on the distribution of the following columns:

- loan_amount
- rate_of_interest
- loan_type
- property_value

Hint: the descriptive comments may include about the similarity of the distribution with normal distribution or uniform distribution, about skewness, about mode, etc.

B-3: Draw a plot between **loan_amount** and **Gender**. What can you conclude from the plot?

B-4: Draw a plot between **property_value** and **loan_amount**, differentiated by **Status**. For the defaulted loans, what is the relationship between **property_value** and **loan_amount**. (*Note: A **status** of 1 indicates that the loan has defaulted.*)

B-5: Draw a plot between **property_value** and **loan_amount**, differentiated by **Status** and **Region**. Make the figsize=(9,4), and dpi=200. *Hint: dpi is a keyword argument in plt.figure() method.*

B-6: Display a count plot of **credit_type**, differentiated by **loan_purpose**.

B-7: Draw a plot on the **loan_amount** differentiated by **business_or_commercial** and **occupancy_type**.

B-8: Add two new columns to the dataframe with headers: **property value multiple** and **loan multiple**. The values in the new columns can be calculated using the following formulas:

$$\text{property value multiple} = \frac{\text{property\_value}}{\textit{income} \times 12}$$

$$\text{loan multiple} = \frac{\text{loan\_amount}}{\textit{income} \times 12}$$

B-9: Plot the **Credit_Score** in ascending order on the x-axis, and the above two new columns on y-axis (*Hint: use line plots*)

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()
```