CPE232: Data Models

Portion 2: Midterm Exam Coding

Sections: A, B, RC

In [1]:
```
# # Run this cell if using Google Colab
# from google.colab import drive
# drive.mount('/content/drive')
```

ข้อมูลในชุดข้อมูล student_spending.csv **ที่ให้มาประกอบด้วยข้อมูลที่เกี่ยวข้องกับการใช้จ่ายของนักศึกษา ภารกิจของคุณคือการทำการวิเคราะห์ข้อมูลเชิงสำรวจ** (EDA) **ตามภารกิจย่อยห้าประการดังต่อไปนี้**

Subtask #1: รู้จักกับชุดข้อมูล

1.1) ศึกษาภาพรวมของชุดข้อมูล (Total points = 3)

[3 points] Display information of the data: size, shape, and number of dimensions. You can use any libraries of your choice (e.g. Numpy, Pandas).

แสดงรายละเอียดต่อไปนี้ของชุดข้อมูล: ขนาด, รูปร่าง, และจำนวนมิติ นักศึกษาสามารถใช้ไลบรารีใดก็ได้ตามต้องการ (เช่น Numpy, Pandas)

In [32]:
```
import pandas as pd
df = pd.read_csv('./student_spending.csv') #TODO: update the path and filename at this line
```

In [33]:
```
# Write your code here
print('size >>', df.size)
print('shape >>', df.shape)
print('dimensions >>', df.ndim)
```

```
size >> 18000
shape >> (1000, 18)
dimensions >> 2
```

1.2) ศึกษาสถิติของชุดข้อมูลนี้เพิ่มเติม (Total point = 15)

Use the command below:

ใช้คำสั่งต่อไปนี้:

`df.info()`

In [34]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 18 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Unnamed: 0               1000 non-null   int64
 1   age                      1000 non-null   int64
 2   gender                   1000 non-null   object
 3   year_in_school           1000 non-null   object
 4   major                    1000 non-null   object
 5   monthly_income           1000 non-null   int64
 6   financial_aid            1000 non-null   int64
 7   tuition                  1000 non-null   int64
 8   housing                  1000 non-null   int64
 9   food                     1000 non-null   int64
 10  transportation           1000 non-null   int64
 11  books_supplies           1000 non-null   int64
 12  entertainment            1000 non-null   int64
 13  personal_care            1000 non-null   int64
 14  technology               1000 non-null   int64
 15  health_wellness          1000 non-null   int64
 16  miscellaneous            1000 non-null   int64
 17  preferred_payment_method 1000 non-null   object
dtypes: int64(14), object(4)
memory usage: 140.8+ KB
```

[2 points] Obtain the following information and provide your answers:

- Number of columns of the type *Integer*
- Number of columns of the type *String*

หาค่าต่อไปนี้จากชุดข้อมูลและระบุคำตอบ:

- **จำนวนคอลัมน์ที่เป็น** *Integer*
- **จำนวนคอลัมน์ที่เป็น** *String*

ANS:

- **จำนวนคอลัมน์ที่เป็น** *Integer* = 14
- **จำนวนคอลัมน์ที่เป็น** *String* = 4

[1 point] Display the first 6 rows.

แสดง 6 **แถวแรกของข้อมูล**

In [35]:
```
# Write your code here
df.head(6)
```

Out[35]:

	Unnamed: 0	age	gender	year_in_school	major	monthly_income	financial_aid	tuition	housing
0	0	19	Non-binary	Freshman	Psychology	958	270	5939	70
1	1	24	Female	Junior	Economics	1006	875	4908	55
2	2	24	Non-binary	Junior	Economics	734	928	3051	66
3	3	23	Female	Senior	Computer Science	617	265	4935	65
4	4	20	Female	Senior	Computer Science	810	522	3887	82
5	5	25	Non-binary	Sophomore	Computer Science	523	790	3151	41

[1 point] Display the last 10 rows.

แสดง 10 แถวสุดท้ายของข้อมูล

In [36]:
```
# Write your code here
df.tail(10)
```

Out[36]:

	Unnamed: 0	age	gender	year_in_school	major	monthly_income	financial_aid	tuition	hous
990	990	20	Non-binary	Senior	Psychology	1412	155	5576	
991	991	24	Non-binary	Junior	Psychology	1391	259	3572	
992	992	20	Male	Freshman	Economics	1293	672	5635	
993	993	20	Male	Freshman	Psychology	1380	594	3658	
994	994	22	Male	Senior	Psychology	764	286	5430	
995	995	22	Female	Senior	Biology	1346	520	3688	
996	996	19	Female	Senior	Biology	1407	560	3380	
997	997	20	Male	Junior	Economics	957	393	3497	
998	998	22	Non-binary	Senior	Economics	1174	612	3649	
999	999	24	Non-binary	Sophomore	Computer Science	541	640	5965	

[1 point] Descriptive statistics of *ALL attributes*

สถิติเชิงพรรณนาของ *ทุกๆคุณลักษณะ*

In [37]:
```
# Write your code here
df.describe()
```

Out[37]:

	Unnamed: 0	age	monthly_income	financial_aid	tuition	housing	food
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.00000	1000.000000
mean	499.500000	21.675000	1020.650000	504.771000	4520.395000	696.00600	252.642000
std	288.819436	2.322664	293.841161	287.092575	860.657944	171.21862	86.949606
min	0.000000	18.000000	501.000000	0.000000	3003.000000	401.00000	100.000000
25%	249.750000	20.000000	770.750000	261.000000	3779.750000	538.75000	175.000000
50%	499.500000	22.000000	1021.000000	513.000000	4547.500000	704.50000	255.000000
75%	749.250000	24.000000	1288.250000	751.500000	5285.000000	837.25000	330.000000
max	999.000000	25.000000	1500.000000	1000.000000	6000.000000	1000.00000	400.000000

[1 point] Descriptive statistics of *one selected attribute*: `tuition`

สถิติเชิงพรรณนาของ *หนึ่งคุณลักษณะ*: `tuition`

In [38]:
```
# Write your code here
df.tuition.describe()
```

Out[38]:
```
count    1000.000000
mean     4520.395000
std       860.657944
min      3003.000000
25%      3779.750000
50%      4547.500000
75%      5285.000000
max      6000.000000
Name: tuition, dtype: float64
```

[4 points] Descriptive statistics of *four selected attribute*: `age` , `housing` , `food` , `transportation`

สถิติเชิงพรรณนาของ *สี่คุณลักษณะ*: `age` , `housing` , `food` , `transportation`

In [39]:
```
# Write your code here
filtered_df = df[['age', 'housing', 'food', 'transportation']]
filtered_df.describe()
```

Out[39]:

	age	housing	food	transportation
count	1000.000000	1000.00000	1000.000000	1000.00000
mean	21.675000	696.00600	252.642000	124.63700
std	2.322664	171.21862	86.949606	43.55799
min	18.000000	401.00000	100.000000	50.00000
25%	20.000000	538.75000	175.000000	88.00000
50%	22.000000	704.50000	255.000000	123.00000
75%	24.000000	837.25000	330.000000	162.25000
max	25.000000	1000.00000	400.000000	200.00000

[5 points] Display the number of occurrences of each unique value in ALL *non-integer* columns.

แสดงจำนวนครั้งที่แต่ละค่าที่ไม่ซ้ำกันปรากฏในทุกคอลัมน์ที่ *ไม่ใช่จำนวนเต็ม*

Hint: Example of the output may look like the following for the column named `genre` .

ตัวอย่างของผลลัพธ์อาจคล้ายผลต่อไปนี้สำหรับคอลัมน์ที่มีชื่อว่า `genre`

```
genre
pop          550
jazz         234
rock         294
country      146
Name: count, dtype: int64
```

In [40]:
```
# gender Unique
df.gender.value_counts()
```

Out[40]:
```
gender
Male          356
Female        323
Non-binary    321
Name: count, dtype: int64
```

In [41]:
```
# year_in_school Unique
df.year_in_school.value_counts()
```

Out[41]:
```
year_in_school
Senior       254
Freshman     253
Junior       247
Sophomore    246
Name: count, dtype: int64
```

In [42]:
```
# major Unique
df.major.value_counts()
```

Out[42]:
```
major
Biology              228
Economics            204
Computer Science     192
Engineering          192
Psychology           184
Name: count, dtype: int64
```

In [43]:
```
# preferred_payment_method Unique
df.preferred_payment_method.value_counts()
```

Out[43]:
```
preferred_payment_method
Mobile Payment App    350
Credit/Debit Card     340
Cash                  310
Name: count, dtype: int64
```

Subtask #2: ตรวจสอบข้อมูล

2.1) ตรวจสอบข้อมูลนักศึกษาโดยกำหนดเงื่อนไข (Total points = 14)

[4 points] Display *the first 10 rows* of records that meet the condition: *Students with a major in Computer Science with a spending on technology more than 100.*

แสดง 10 แถวแรก ของข้อมูลที่ตรงตามเงื่อนไขต่อไปนี้: *นักศึกษาที่เรียนสาขาวิทยาการคอมพิวเตอร์และมีค่าใช้จ่ายด้านเทคโนโลยีมากกว่า 100*

In [54]:
```
# Write your code here
# Students with a major in Computer Science with a spending on technology more than 100.
CS_major = df[df.major == 'Computer Science']
Over_Spending_CS_major = CS_major[CS_major.technology > 100]
Over_Spending_CS_major.head(10)
```

Out[54]:

	Unnamed: 0	age	gender	year_in_school	major	monthly_income	financial_aid	tuition	housin
3	3	23	Female	Senior	Computer Science	617	265	4935	65
5	5	25	Non-binary	Sophomore	Computer Science	523	790	3151	41
8	8	22	Non-binary	Senior	Computer Science	1402	248	5638	59
9	9	18	Female	Junior	Computer Science	1423	74	3977	62
32	32	24	Non-binary	Junior	Computer Science	522	555	5236	86
37	37	23	Non-binary	Senior	Computer Science	1309	265	5160	60
45	45	18	Male	Freshman	Computer Science	929	348	3854	59
52	52	19	Male	Senior	Computer Science	669	660	3823	83
56	56	24	Non-binary	Freshman	Computer Science	854	700	4824	52
71	71	21	Non-binary	Sophomore	Computer Science	1235	805	5442	74

[1 point] How many records are there that match the above condition?

มีทั้งหมดจำนวนกี่รายการที่ตรงกับเงื่อนไขข้างต้น?

Ans: 154 records

In [57]:
```
Over_Spending_CS_major.shape[0]
```

Out[57]:
```
154
```

[8 points] Display *the first 10 rows* of records that meet the condition: *Male Sophomore students with monthly income ranging from 600 to 1000.*

แสดง 10 แถวแรก ของข้อมูลที่ตรงตามเงื่อนไขต่อไปนี้: *นักศึกษาชั้นปีที่สอง (Sophomore) ที่เป็นเพศชายและมีรายได้ต่อเดือนอยู่ในช่วง 600 ถึง 1000*

In [67]:
```
# Write your code here
# Male Sophomore students with monthly income ranging from 600 to 1000.
Sophomore = df[df.year_in_school == 'Sophomore']
filter_Sophomore_by_income = Sophomore[(Sophomore.monthly_income >= 600) & (Sophomore.monthly
filter_Sophomore_by_income.head(10)
```

Out[67]:

	Unnamed: 0	age	gender	year_in_school	major	monthly_income	financial_aid	tuition	hous
12	12	21	Male	Sophomore	Economics	719	540	4863	
28	28	24	Non-binary	Sophomore	Psychology	905	671	4156	
41	41	25	Male	Sophomore	Economics	804	140	5332	
58	58	25	Non-binary	Sophomore	Biology	668	50	3650	
76	76	22	Male	Sophomore	Computer Science	983	862	5650	
77	77	24	Female	Sophomore	Computer Science	914	24	4881	
89	89	23	Male	Sophomore	Economics	800	933	5304	
97	97	18	Non-binary	Sophomore	Biology	767	457	5096	
102	102	20	Female	Sophomore	Psychology	920	149	3366	
108	108	20	Male	Sophomore	Computer Science	965	322	4992	

[1 point] How many records are there that match the above condition?

มีทั้งหมดจำนวนกี่รายการที่ตรงกับเงื่อนไขข้างต้น?

Ans: 111 records

In [65]:
```
filter_Sophomore_by_income.shape[0]
```

Out[65]:
```
111
```

2.2) ตรวจสอบว่ามีค่าที่ตกหล่นไปหรือไม่ (Total point = 1)

[1 point] How many attributes contain missing values?

มีคุณลักษณะ (attribute) กี่รายการที่มีค่าที่ตกหล่นไป

ANS: ทุก attribute ไม่มี missing values

In [68]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 18 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Unnamed: 0               1000 non-null   int64
 1   age                      1000 non-null   int64
 2   gender                   1000 non-null   object
 3   year_in_school           1000 non-null   object
 4   major                    1000 non-null   object
 5   monthly_income           1000 non-null   int64
 6   financial_aid            1000 non-null   int64
 7   tuition                  1000 non-null   int64
 8   housing                  1000 non-null   int64
 9   food                     1000 non-null   int64
 10  transportation           1000 non-null   int64
 11  books_supplies           1000 non-null   int64
 12  entertainment            1000 non-null   int64
 13  personal_care            1000 non-null   int64
 14  technology               1000 non-null   int64
 15  health_wellness          1000 non-null   int64
 16  miscellaneous            1000 non-null   int64
 17  preferred_payment_method 1000 non-null   object
dtypes: int64(14), object(4)
memory usage: 140.8+ KB
```

Subtask #3: จัดเตรียมข้อมูล

3.1 ลบคอลัมน์ที่ไม่จำเป็นออกจากชุดข้อมูล (Total points = 2)

[2 points] **ลบคอลัมน์** "Unnamed: 0" **ออกจากชุดข้อมูล**

*(Note: this column must no longer appear when displaying the dataframe again later; **หมายเหตุ: คอลัมน์นี้ต้องไม่ปรากฏอีกเมื่อแสดง** DataFrame **ในภายหลัง**)*

In [70]:
```
# Write your code here
df = df.drop(columns='Unnamed: 0')
df
```

Out[70]:

	age	gender	year_in_school	major	monthly_income	financial_aid	tuition	housing	food
0	19	Non-binary	Freshman	Psychology	958	270	5939	709	296
1	24	Female	Junior	Economics	1006	875	4908	557	365
2	24	Non-binary	Junior	Economics	734	928	3051	666	220
3	23	Female	Senior	Computer Science	617	265	4935	652	289
4	20	Female	Senior	Computer Science	810	522	3887	825	372
...	...	...	...	...	...	...	...	...	...
995	22	Female	Senior	Biology	1346	520	3688	969	152
996	19	Female	Senior	Biology	1407	560	3380	508	265
997	20	Male	Junior	Economics	957	393	3497	723	339
998	22	Non-binary	Senior	Economics	1174	612	3649	543	237
999	24	Non-binary	Sophomore	Computer Science	541	640	5965	609	270

1000 rows × 17 columns

3.2 สร้างคอลัมน์ใหม่ (Total points = 23)

[5 points] Create a new column and name it `major_expense`. This column contains values that are the sum of housing, food, and transportation.

สร้างคอลัมน์ใหม่ชื่อ `major_expense` โดยมีค่าที่ได้จากผลรวมของค่าใช้จ่ายด้านที่อยู่อาศัย อาหาร และการเดินทาง

In [75]:
```
# Write your code here
major_expense = df.housing + df.food + df.transportation
df['major_expense'] = major_expense
df
```

Out[75]:

	age	gender	year_in_school	major	monthly_income	financial_aid	tuition	housing	food
0	19	Non-binary	Freshman	Psychology	958	270	5939	709	296
1	24	Female	Junior	Economics	1006	875	4908	557	365
2	24	Non-binary	Junior	Economics	734	928	3051	666	220
3	23	Female	Senior	Computer Science	617	265	4935	652	289
4	20	Female	Senior	Computer Science	810	522	3887	825	372
...	...	...	...	...	...	...	...	...	...
995	22	Female	Senior	Biology	1346	520	3688	969	152
996	19	Female	Senior	Biology	1407	560	3380	508	265
997	20	Male	Junior	Economics	957	393	3497	723	339
998	22	Non-binary	Senior	Economics	1174	612	3649	543	237
999	24	Non-binary	Sophomore	Computer Science	541	640	5965	609	270

1000 rows × 18 columns

[8 points] Create another new column and name it `major_expense_ratio` which is based on the following formula:

สร้างคอลัมน์ใหม่ชื่อ `major_expense_ratio` **โดยอิงจากสูตรการคำนวณต่อไปนี้**

```
major_expense_ratio = (major_expense * 100) / monthly_income
```

In [76]:

```
# Write your code here
major_expense_ratio = (df.major_expense * 100) / df.monthly_income
df['major_expense_ratio'] = major_expense_ratio
df
```

Out[76]:

	age	gender	year_in_school	major	monthly_income	financial_aid	tuition	housing	food
0	19	Non-binary	Freshman	Psychology	958	270	5939	709	296
1	24	Female	Junior	Economics	1006	875	4908	557	365
2	24	Non-binary	Junior	Economics	734	928	3051	666	220
3	23	Female	Senior	Computer Science	617	265	4935	652	289
4	20	Female	Senior	Computer Science	810	522	3887	825	372
...	...	...	...	...	...	...	...	...	...
995	22	Female	Senior	Biology	1346	520	3688	969	152
996	19	Female	Senior	Biology	1407	560	3380	508	265
997	20	Male	Junior	Economics	957	393	3497	723	339
998	22	Non-binary	Senior	Economics	1174	612	3649	543	237
999	24	Non-binary	Sophomore	Computer Science	541	640	5965	609	270

1000 rows × 19 columns

[10 points] According to the results in previous cell(s), do most students experience financial difficulties as a result of exceeding their monthly income? What is the percentage of those who experience financial difficulties and those who do not? Show your work and and analysis below.

จากผลลัพธ์ในเซลล์ก่อนหน้า นักศึกษาส่วนใหญ่ประสบปัญหาทางการเงินเนื่องจากใช้จ่ายเกินรายได้ต่อเดือนหรือไม่? คำนวณเปอร์เซ็นต์ของนักศึกษาที่ประสบปัญหาทางการเงินและนักศึกษาที่ไม่ประสบปัญหา พร้อมบรรยายผลการวิเคราะห์

ANS: ใช่ นักศึกษาส่วนใหญ่ประสบปัญหาทางการเงินเนื่องจากใช้จ่ายเกินรายได้ต่อเดือน คิดเป็นเปอร์เซ็นต์ ดังนี้

- Percentage of Financial Difficulties Student >> 53.5 %
- Percentage of Non Financial Difficulties Student >> 46.5 %

สื่อให้เห็นว่าจำนวนนักศึกษาที่ประสบปัญหาทางการเงินนั้นมีมากกว่า นักศึกษาที่ไม่ประสบปัญหาทางการเงิน ถึง 7% และ หากเทียบเป็นจำนวนคน พบว่ามีนักศึกษาที่ประสบปัญหาทางการเงินเนื่องจากใช้จ่ายเกินรายได้ต่อเดือน จำนวน 535 คน

In [89]:
```
def Financial_Diff(x):
    if( x > 100 ):
        return 'Yes'
    else:
        return 'No'
```

In [103…
```
total_student = df.shape[0]

df['Financial_Diff'] = df['major_expense_ratio'].apply(Financial_Diff)

Ratio = df.Financial_Diff.value_counts() / total_student * 100
```

```
print(f'Percentage of Financial Difficulties Student >> {Ratio.iloc[0]} %')
print(f'Percentage of Non Financial Difficulties Student >> {Ratio.iloc[1]} %')
```

```
Percentage of Financial Difficulties Student >> 53.5 %
Percentage of Non Financial Difficulties Student >> 46.5 %
```

Subtask #4: สร้างแผนภาพ (Visualizations)

In [105…
```
import matplotlib.pyplot as plt
```

4.1 วิเคราะห์สาขาวิชาต่างๆ (Total points = 10)

[5 points] Create a *pie chart* to demonstrate unique values of the attribute `major`. In your visualization, also display chart title, percentage of distribution, and a legend.

Use a method `.unique()` method to obtain unique values in the attribute.

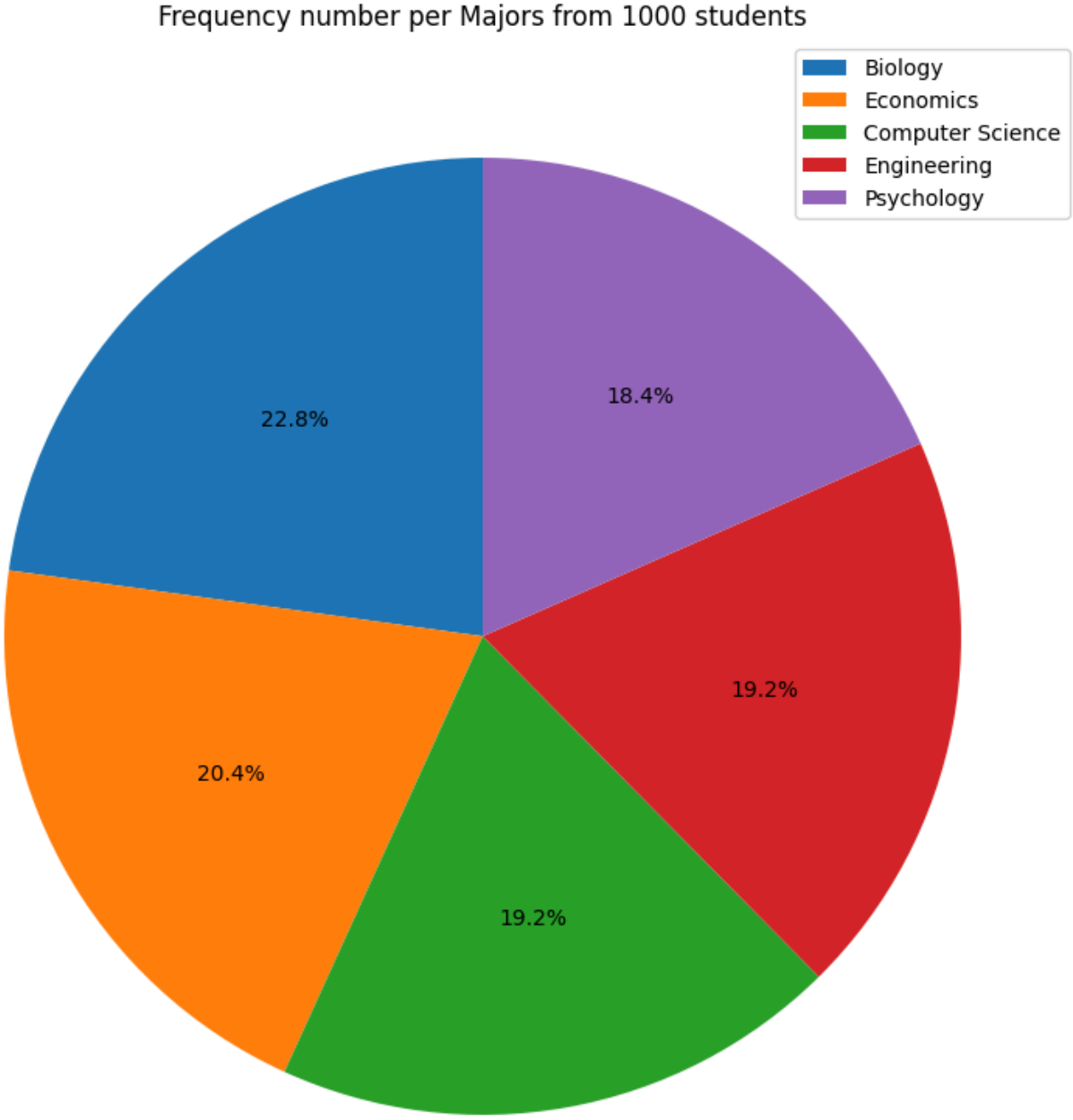
สร้าง แผนภูมิวงกลม (pie chart) **เพื่อแสดงค่าที่ไม่ซ้ำกันของคุณลักษณะ `major` ในการแสดงผล ให้แสดงชื่อแผนภูมิ, เปอร์เซ็นต์การกระจาย, และคำอธิบายสัญลักษณ์** (legend)

ใช้เมธอด** .unique() **เพื่อดึงค่าที่ไม่ซ้ำกันของคุณลักษณะนั้น

In [110…
```
# Write your code here
Majors = df.major.value_counts()
Majors
```

Out[110…
```
major
Biology             228
Economics           204
Computer Science    192
Engineering         192
Psychology          184
Name: count, dtype: int64
```

In [155…
```
plt.figure(figsize=(10,10))
plt.pie(x=Majors, startangle=90, autopct='%1.1f%%')
plt.title('Frequency number per Majors from 1000 students')
plt.legend(labels= Majors.index)
plt.show()
```



[5 points] Describe this visualization in your own words. What information does it convey?

อธิบายแผนภูมิที่สร้างขึ้น แผนภูมินี้สื่อถึงข้อมูลอะไร?

ANS: แผนภูมินี้สื่อถึงสัดส่วนของนักศึกษาในแต่ละ majors จากนักศึกษา 1000 คน ซึ่งจากแผนภูมิแสดงให้เห็นว่าจากนักศึกษา 1000 คน Majors of Biology มีจำนวนนักศึกษามากที่สุด ตามมาด้วย Majors of Economics จากนั้น Major of Computer Science และ Major of Engineering ซึ่งมีจำนวนนักศึกษาเท่ากัน และสุดท้าย Major of Psycology ซึ่งมีจำนวนนักศึกษาน้อยที่สุด

4.2 ศึกษาการกระจายตัวในข้อมูล (Total points = 24)

[7 points] Create a bar chart to demonstrate the distribution of gender. In your visualization, also display the chart title and data labels.

สร้างแผนภูมิแท่ง (bar chart) เพื่อแสดงการกระจายของเพศ (gender) ในการแสดงผล ให้แสดงชื่อแผนภูมิ และคำอธิบายสัญลักษณ์ (legend)

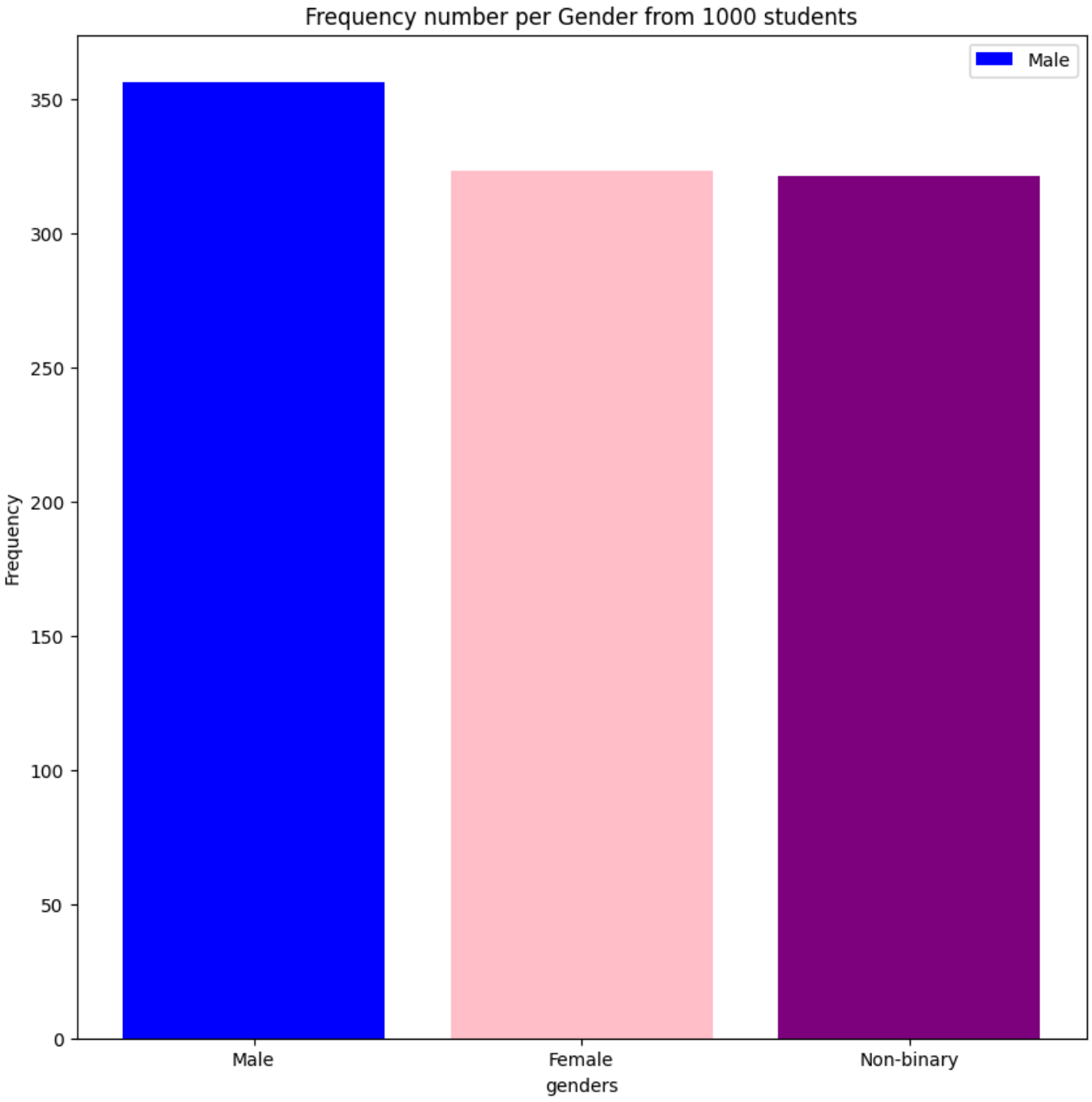
In [142…

```
# Write your code here
Genders = df.gender.value_counts()
Genders
```

Out[142…
```
gender
Male          356
Female        323
Non-binary    321
Name: count, dtype: int64
```

In [164…
```
plt.figure(figsize=(10,10))
plt.bar(x = Genders.index, height= Genders, color=['blue','pink','purple'])
plt.title('Frequency number per Gender from 1000 students')
plt.xlabel('genders')
plt.ylabel('Frequency')

plt.legend(labels = Genders.index)
plt.show()
```



[5 points] Describe this visualization in your own words. What information does it convey?

อธิบายแผนภูมิที่สร้างขึ้น แผนภูมินี้สื่อถึงข้อมูลอะไร?

ANS: แผนภูมินี้สื่อถึงจำนวนนักศึกษาแต่ละเพศ จากนักศึกษา 1000 คน ซึ่งจากแผนภูมิแสดงให้เห็นว่า จากนักศึกษา 1000 คน มีคนเป็นเพศชายมากที่สุด ตามด้วย เพศหญิง และไม่ระบุ ตามลำดับ อาจสื่ออย่างมีนัยสำคัญได้ว่าในสถานศึกษาแห่งนี้มีประชากรชายมากกว่าประชากรหญิง

[7 points] Create a *horizontal* bar chart to demonstrate the distribution of year in school. In your visualization, also display the chart title and data labels.

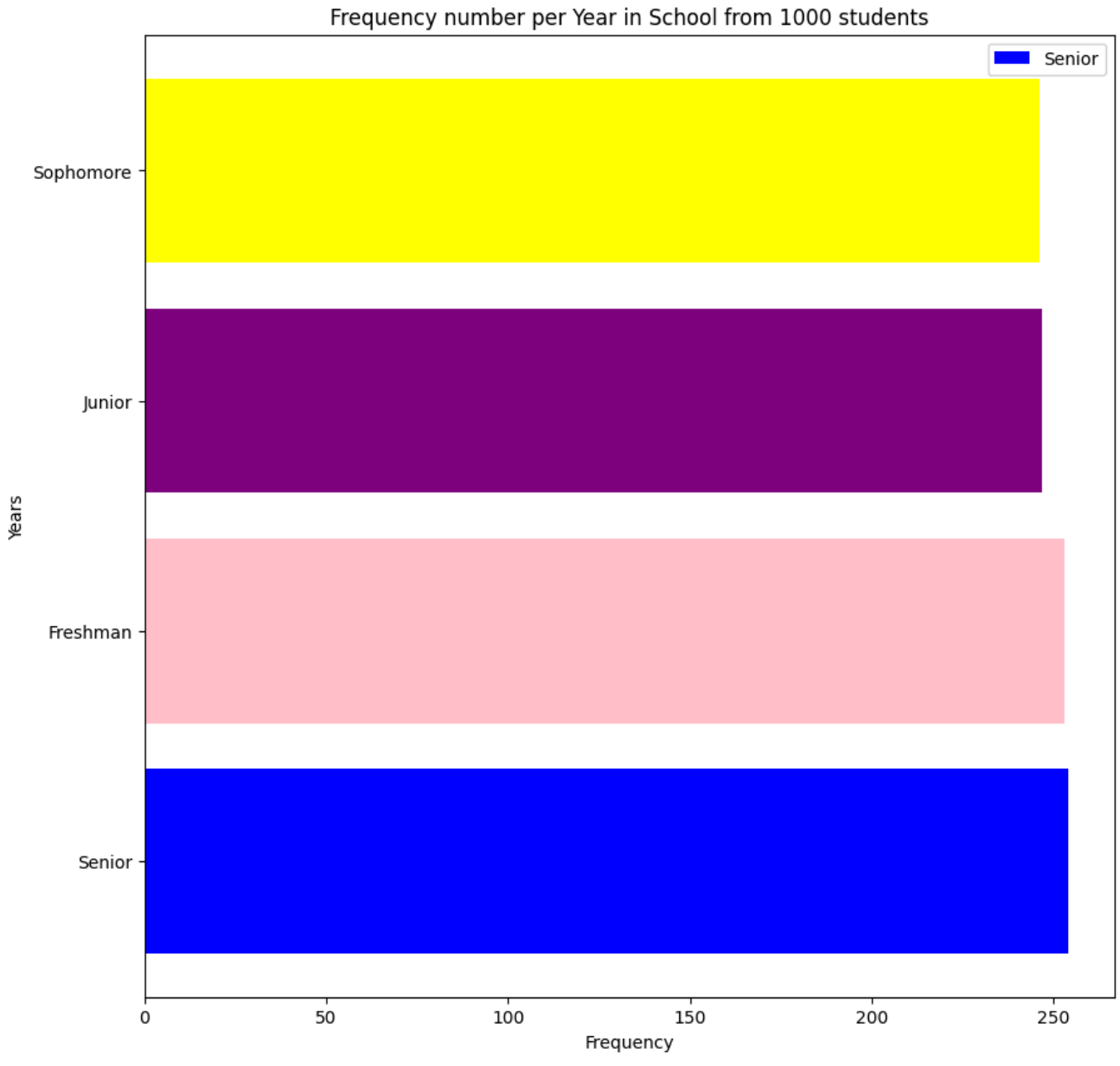
สร้าง แผนภูมิแท่งแนวนอน (horizontal bar chart) **เพื่อแสดงการกระจายของปีที่ศึกษา** (year in school) **ในการแสดงผล ให้แสดงชื่อแผนภูมิ และคำอธิบายสัญลักษณ์** (legend)

In [165…
```
# Write your code here
Year_in_school = df.year_in_school.value_counts()
Year_in_school
```

Out[165…
```
year_in_school
Senior       254
Freshman     253
Junior       247
Sophomore    246
Name: count, dtype: int64
```

In [170…
```
plt.figure(figsize=(10,10))
plt.barh(y = Year_in_school.index, width= Year_in_school, color=['blue','pink','purple','yell
plt.title('Frequency number per Year in School from 1000 students')
plt.xlabel('Frequency')
plt.ylabel('Years')

plt.legend(labels = Year_in_school.index)
plt.show()
```



[5 points] Describe this visualization in your own words. What information does it convey?

อธิบายแผนภูมิที่สร้างขึ้น แผนภูมินี้สื่อถึงข้อมูลอะไร?

ANS: แผนภูมินี้สื่อถึงจำนวนนักศึกษาแต่ละชั้นปี จากนักศึกษา 1000 คน ซึ่งจากแผนภูมิแสดงให้เห็นว่า จากนักศึกษา 1000 คน ทุกชั้นปีมีจำนวนนักศึกษาใกล้เคียงกันมากๆ อาจสื่อได้ถึงอัตราการผ่านที่สูงของทุกๆชั้นปี แต่หากดูในเชิงลึกจะเห็นได้ว่า นักศึกษาชั้นปีที่ Senior จะเยอะที่สุด ตามด้วย Freshman, junior และ Sophomore ตามลำดับ

Subtask #5: จัดกลุ่มข้อมูล

[6 points] Group the data by `year_in_school` and `major`, then display the sum of these attributes: `entertainment`, `personal_care`, `technology`, `health_wellness`, and `miscellaneous`.

จัดกลุ่มข้อมูลตาม `year_in_school` และ `major` แล้วแสดงผลรวมของคุณลักษณะต่อไปนี้: `entertainment`, `personal_care`, `technology`, `health_wellness`, และ `miscellaneous`

In [171…
```
# Write your code here
df.groupby(['year_in_school', 'major'])[['entertainment', 'personal_care', 'technology', 'hea
```

Out[171…

year_in_school	major	entertainment	personal_care	technology	health_wellness
Freshman	Biology	5173	3392	10912	7033
	Computer Science	4405	3300	9931	6069
	Economics	3832	2638	7530	4605
	Engineering	3844	3097	7838	5705
	Psychology	4536	3289	9551	5676
Junior	Biology	5306	3562	9975	6997
	Computer Science	3886	2733	8375	5589
	Economics	3596	2689	9160	5231
	Engineering	4429	2802	9212	5122
	Psychology	3768	2602	7286	5564
Senior	Biology	4557	3426	10858	6924
	Computer Science	3705	3164	7285	5406
	Economics	5166	3520	9589	6350
	Engineering	4188	3516	10223	5404
	Psychology	3498	2671	7783	5039
Sophomore	Biology	4784	3053	9851	6778
	Computer Science	3770	2626	8200	5262
	Economics	4708	3158	10048	5612
	Engineering	3820	2404	6879	4373
	Psychology	3843	3057	7818	5571

[2 points] Describe your understanding from this output.

อธิบายความเข้าใจจากผลลัพธ์ที่ได้

ANS: ผลลัพธ์สื่อให้เห็นถึงผลรวมของค่าใช้จ่ายที่นักศึกษาของแต่ละคณะของแต่ละชั้นปีได้เสียไปให้กับ entertainment, personal_care, technology **และ** health_wellness **เนื่องจากผลลัพธ์นี้มีตัวเลขจำนวนมาก และ ยังไม่ใช่การ** visualize **ที่ดีเท่าที่ควร ทำให้การ** observe trend **ต่างๆ หรือการ** comparison **กันระหว่างคณะ และระหว่างชั้นปี มีความยากมาก หากจะให้ดีควรจะมี** Total **ของแต่ละชั้นปี เพื่อการเปรียบเทียบที่ชัดเจนมากขึ้น**

แต่หากสังเกตโดยที่ไม่มีการ process **ใดๆ ทั้งสิ้นจะสรุปได้คร่าวๆว่า**

- **คณะ และ ชั้นปี ที่เสียเงินให้กับ** entertainment **มากที่สุด ได้แก่** Biology Junior
- **คณะ และ ชั้นปี ที่เสียเงินให้กับ** personal_care **มากที่สุด ได้แก่** Biology Junior
- **คณะ และ ชั้นปี ที่เสียเงินให้กับ** technology **มากที่สุด ได้แก่** Biology Freshman
- **คณะ และ ชั้นปี ที่เสียเงินให้กับ** health_wellness **มากที่สุด ได้แก่** Biology Freshman

สังเกตได้ว่าในทุกๆด้าน คณะ Biology **มักจะเป็นคณะที่เสียค่าใช้จ่ายมากที่สุด อาจสื่อได้ถึง ฐานะของนักศึกษาที่เรียนในคณะนี้**