Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ_____

КАФЕДРА ____СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)____

Курс «Технологии машинного обучения»

Отчет по лабораторной работе №1

«Разведочный анализ данных. Исследование и визуализация данных»

Выполнила:
студентка группы ИУ5-61Б
Покшубина Софья
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2023 г.

Цель лабораторной работы:

Изучение различных методов визуализация данных.

Описание задания:

- Выбрать набор данных (датасет)
- Для первой лабораторной работы рекомендуется использовать датасет без пропусков в данных

  Для лабораторных работ не рекомендуется выбирать датасеты большого размера.
- Создать ноутбук, который содержит следующие разделы:

  1. Текстовое описание выбранного Вами набора данных.
  2. Основные характеристики датасета.
  3. Визуальное исследование датасета.

Описание датасета

Этот датасет изучает психическое здоровье, семейный статус, наличие консультаций со специалистом студентов университета и их влияние на средний балл диплома. Он содержит важную демографическую информацию и ответы студентов для исчерпывающей картины влияния психического состояния студентов на их средний балл. С помощью этого исследования можно добиться лучшего понимания того, как психическое здоровье влияет на учебу. После изучения отдельных факторов, которые могут способствовать разным результатам, университеты смогут работать над улучшением образовательных систем на благо учащихся.

Описание столбцов

- 'Choose your gender' - пол студента
- 'Age' - возраст студента
- 'What is your course?' - специальность обучения
- 'Your current year of Study' - текущий курс
- 'What is your CGPA?' - средний балл диплома
- 'Marital status' - семейное положение
- 'Do you have Depression?' - наличие депрессии
- 'Do you have Anxiety?' - наличие тревожности
- 'Do you have Panic attack?' - наличие панических атак
- 'Did you seek any specialist for a treatment?' - искал ли студент помощи у специалиста

Выгрузка библиотек и датасета

B [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

B [2]:
```
df = pd.read_csv("Student Mental health.csv")
df.head(10)
```

Out[2]:

	Timestamp	Choose your gender	Age	What is your course?	Your current year of Study	What is your CGPA?	Marital status	Do you have Depression?	Do you have Anxiety?	Do you have Panic attack?	Did you seek any specialist for a treatment?
0	8/7/2020 12:02	Female	18.0	Engineering	year 1	3.00 - 3.49	No	Yes	No	Yes	No
1	8/7/2020 12:04	Male	21.0	Islamic education	year 2	3.00 - 3.49	No	No	Yes	No	No
2	8/7/2020 12:05	Male	19.0	BIT	Year 1	3.00 - 3.49	No	Yes	Yes	Yes	No
3	8/7/2020 12:06	Female	22.0	Laws	year 3	3.00 - 3.49	Yes	Yes	No	No	No
4	8/7/2020 12:13	Male	23.0	Mathemathics	year 4	3.00 - 3.49	No	No	No	No	No
5	8/7/2020 12:31	Male	19.0	Engineering	Year 2	3.50 - 4.00	No	No	No	Yes	No
6	8/7/2020 12:32	Female	23.0	Pendidikan islam	year 2	3.50 - 4.00	Yes	Yes	No	Yes	No
7	8/7/2020 12:33	Female	18.0	BCS	year 1	3.50 - 4.00	No	No	Yes	No	No
8	8/7/2020 12:35	Female	19.0	Human Resources	Year 2	2.50 - 2.99	No	No	No	No	No
9	8/7/2020 12:39	Male	18.0	Irkhs	year 1	3.50 - 4.00	No	No	Yes	Yes	No

Характеристики датасета

Информация о датасете:

B [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101 entries, 0 to 100
Data columns (total 11 columns):
 #   Column                                        Non-Null Count  Dtype
---  ------                                        --------------  -----
 0   Timestamp                                     101 non-null    object
 1   Choose your gender                            101 non-null    object
 2   Age                                           100 non-null    float64
 3   What is your course?                          101 non-null    object
 4   Your current year of Study                    101 non-null    object
 5   What is your CGPA?                            101 non-null    object
 6   Marital status                                101 non-null    object
 7   Do you have Depression?                       101 non-null    object
 8   Do you have Anxiety?                          101 non-null    object
 9   Do you have Panic attack?                     101 non-null    object
 10  Did you seek any specialist for a treatment?  101 non-null    object
dtypes: float64(1), object(10)
memory usage: 8.8+ KB
```

Удалим столбец 'Timestamp' за ненадобностью:

B [4]:
```
df.drop(['Timestamp'], axis = 1, inplace = True)
```

Переименуем колонки:

В [5]:
```
df_column = {'Choose your gender':'Gender','What is your course?':'Course',
             'Your current year of Study':'Year of Study','What is your CGPA?':'CGPA',
             'Do you have Depression?':'Depression', 'Marital status':'Married',
             'Do you have Anxiety?':'Anxiety', 'Do you have Panic attack?':'Panic Attack',
             'Did you seek any specialist for a treatment?':'Treatment'}
df.rename(columns = df_column, inplace=True)
df.head()
```

Out[5]:

	Gender	Age	Course	Year of Study	CGPA	Married	Depression	Anxiety	Panic Attack	Treatment
0	Female	18.0	Engineering	year 1	3.00 - 3.49	No	Yes	No	Yes	No
1	Male	21.0	Islamic education	year 2	3.00 - 3.49	No	No	Yes	No	No
2	Male	19.0	BIT	Year 1	3.00 - 3.49	No	Yes	Yes	Yes	No
3	Female	22.0	Laws	year 3	3.00 - 3.49	Yes	Yes	No	No	No
4	Male	23.0	Mathematics	year 4	3.00 - 3.49	No	No	No	No	No

Заменим значения:

В [6]:
```
df.replace({'Yes': 1,'No': 0}, inplace = True)
df.head()
```

Out[6]:

	Gender	Age	Course	Year of Study	CGPA	Married	Depression	Anxiety	Panic Attack	Treatment
0	Female	18.0	Engineering	year 1	3.00 - 3.49	0	1	0	1	0
1	Male	21.0	Islamic education	year 2	3.00 - 3.49	0	0	1	0	0
2	Male	19.0	BIT	Year 1	3.00 - 3.49	0	1	1	1	0
3	Female	22.0	Laws	year 3	3.00 - 3.49	1	1	0	0	0
4	Male	23.0	Mathematics	year 4	3.00 - 3.49	0	0	0	0	0

Форма датасета:

В [7]:
```
df.shape
```

Out[7]:
```
(101, 10)
```

Проверим пропуски:

В [8]:
```
df.isnull().sum()
```

Out[8]:
```
Gender          0
Age             1
Course          0
Year of Study   0
CGPA            0
Married         0
Depression      0
Anxiety         0
Panic Attack    0
Treatment       0
dtype: int64
```

Найдем пропущенное значение в столбце 'Age':

В [9]:
```
df[df.isna().any(axis=1)]
```

Out[9]:

	Gender	Age	Course	Year of Study	CGPA	Married	Depression	Anxiety	Panic Attack	Treatment
43	Male	NaN	BIT	year 1	0 - 1.99	0	0	0	0	0

Заполним пропуск медианным значением:

В [10]:
```
df['Age'].fillna(df['Age'].mean(),inplace=True)
df['Age'] = df["Age"].astype(int)
df.iloc[[43]]
```

Out[10]:

	Gender	Age	Course	Year of Study	CGPA	Married	Depression	Anxiety	Panic Attack	Treatment
43	Male	20	BIT	year 1	0 - 1.99	0	0	0	0	0

Приведем текстовые столбцы к нижнему регистру:

В [11]:
```
df['Course'] = df['Course'].str.lower()
df['Year of Study'] = df['Year of Study'].str.lower()
```

Подсчитаем количество уникальных значений:

В [12]:
```
df.nunique()
```

Out[12]:
```
Gender            2
Age               7
Course           42
Year of Study     4
CGPA              6
Married           2
Depression        2
Anxiety           2
Panic Attack      2
Treatment         2
dtype: int64
```

Проверим уникальные значения в столбце 'CGPA':

В [13]:
```
df["CGPA"].unique()
```

Out[13]:
```
array(['3.00 - 3.49', '3.50 - 4.00', '3.50 - 4.00 ', '2.50 - 2.99',
       '2.00 - 2.49', '0 - 1.99'], dtype=object)
```

Заменим значение:

В [14]:
```
df.replace({'3.50 - 4.00 ':'3.50 - 4.00'},regex = True,inplace = True)
df["CGPA"].unique()
```

Out[14]:
```
array(['3.00 - 3.49', '3.50 - 4.00', '2.50 - 2.99', '2.00 - 2.49',
       '0 - 1.99'], dtype=object)
```

Визуализация

Посчитаем количество студентов каждого пола:

В [15]:
```
gen_count = pd.DataFrame(df['Gender'].value_counts().reset_index())
gen_count.rename(columns = {'index':'Gender','Gender':'Number of Students'}, inplace=True)
gen_count
```

Out[15]:

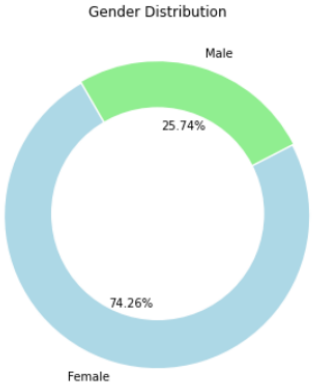
	Gender	Number of Students
0	Female	75
1	Male	26

Распределение студентов по полу:

В [16]:
```
plt.figure(figsize=(8,6))
plt.pie(gen_count['Number of Students'], explode=(0.015,0),
        labels=gen_count['Gender'],
        colors=['lightblue','lightgreen'], autopct='%1.2f%%',
        startangle=120)

centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.title("Gender Distribution")
plt.show()
```



Отберем студентов, у которых есть депрессия, тревожность или панические атаки:

В [17]:
```
dep = df[df["Depression"]== 1]
anx = df[df["Anxiety"]== 1]
pa = df[df["Panic Attack"]== 1]
```

Распределение студентов по полу и их психическому здоровью:

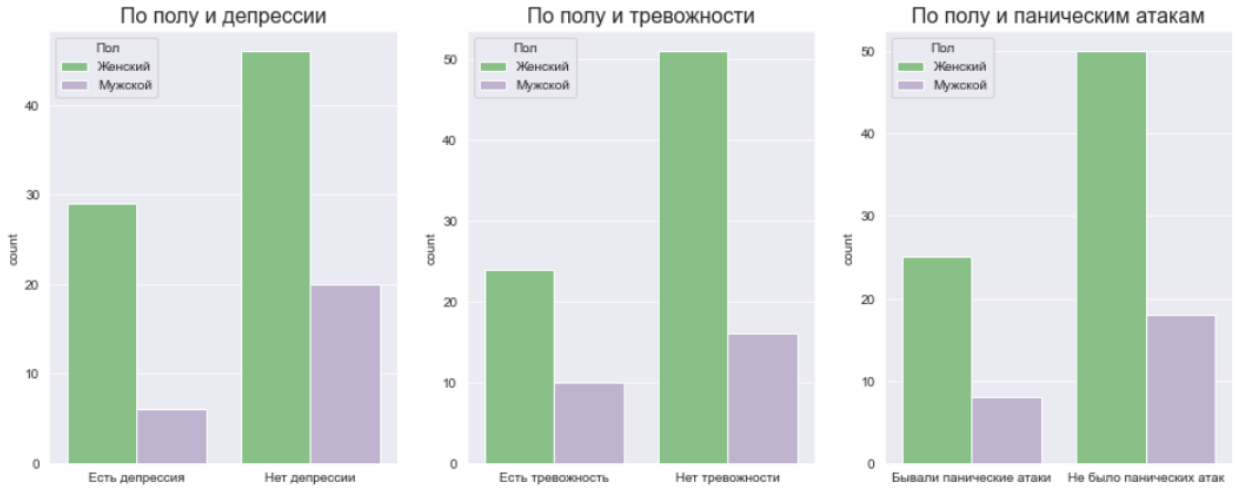
В [18]:
```
sns.set_style('darkgrid')
fig, ax = plt.subplots(1, 3, figsize=(16,6))

sns.countplot(x='Depression', hue='Gender',
              data=df, order=[1,0],
              palette='Accent', ax=ax[0])
ax[0].set_xticklabels(["Есть депрессия","Нет депрессии"])
ax[0].set_xlabel(None)
ax[0].set_title("По полу и депрессии", fontsize = 16)
ax[0].legend(["Женский","Мужской"], title="Пол")

sns.countplot(x='Anxiety', hue='Gender',
              data=df, order=[1,0],
              palette='Accent', ax=ax[1])
ax[1].set_xticklabels(["Есть тревожность","Нет тревожности"])
ax[1].set_xlabel(None)
ax[1].set_title("По полу и тревожности", fontsize = 16)
ax[1].legend(["Женский","Мужской"], title="Пол")

sns.countplot(x='Panic Attack', hue='Gender',
              data=df, order=[1,0],
              palette='Accent', ax=ax[2])
ax[2].set_xticklabels(["Бывали панические атаки","Не было панических атак"])
ax[2].set_xlabel(None)
ax[2].set_title("По полу и паническим атакам", fontsize = 16)
ax[2].legend(["Женский","Мужской"], title="Пол")

plt.show()
```



Посчитаем количество студентов с каждой специальности:

```
В [28]: pd.DataFrame(df['Course'].value_counts()).rename(columns={'Course':'Number of Students'})
```

Out[28]:

	Number of Students
bcs	18
engineering	17
bit	10
koe	6
biomedical science	4
benl	3
psychology	3
laws	2
engine	2
islamic education	2
kirkhs	2
pendidikan islam	2

Распределение студентов по специальности и психическому здоровью:

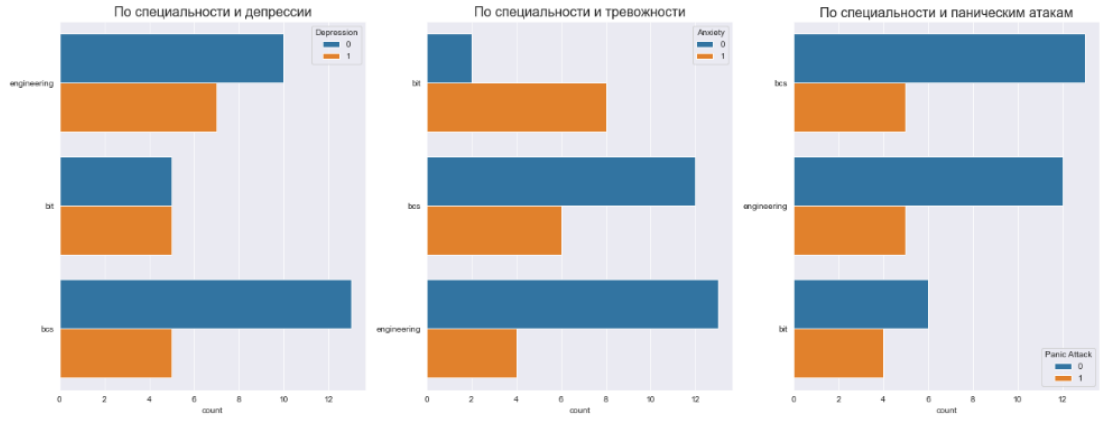
```
В [29]: sns.set_style('darkgrid')
fig, ax = plt.subplots(1, 3, figsize=(22,8))

sns.countplot(y='Course', hue='Depression',
              data=df, order=dep['Course'].value_counts(ascending=False).head(3).index,
              ax=ax[0])
ax[0].set_ylabel(None)
ax[0].set_title("По специальности и депрессии", fontsize = 16)

sns.countplot(y='Course', hue='Anxiety',
              data=df, order=anx['Course'].value_counts(ascending=False).head(3).index,
              ax=ax[1])
ax[1].set_ylabel(None)
ax[1].set_title("По специальности и тревожности", fontsize = 16)

sns.countplot(y='Course', hue='Panic Attack',
              data=df, order=pa['Course'].value_counts(ascending=False).head(3).index,
              ax=ax[2])
ax[2].set_ylabel(None)
ax[2].set_title("По специальности и паническим атакам", fontsize = 16)

plt.show()
```



Посчитаем количество людей с каждым средним баллом:

В [30]:
```
CGPA_count = pd.DataFrame(df['CGPA'].value_counts().reset_index())
CGPA_count.rename(columns = {'index':'CGPA','CGPA':'Number of Students'}, inplace=True)
CGPA_count
```

Out[30]:

	CGPA	Number of Students
0	3.50 - 4.00	48
1	3.00 - 3.49	43
2	2.50 - 2.99	4
3	0 - 1.99	4
4	2.00 - 2.49	2

Распределение людей по психическому здоровью и среднему баллу:

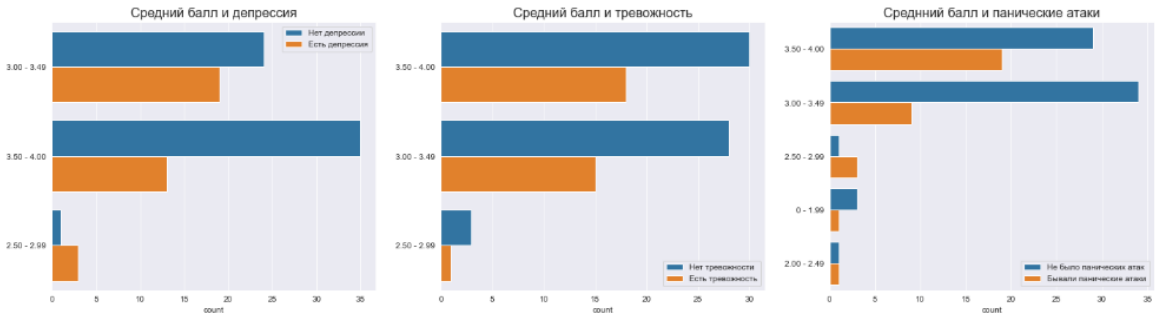
В [31]:
```
sns.set_style('darkgrid')
fig, ax = plt.subplots(1, 3, figsize=(24,6))

sns.countplot(y='CGPA', hue='Depression',
              data=df, order=dep['CGPA'].value_counts(ascending=False).index,
              ax=ax[0])
ax[0].set_ylabel(None)
ax[0].set_title("Средний балл и депрессия", fontsize = 16)
ax[0].legend(["Нет депрессии","Есть депрессия"])

sns.countplot(y='CGPA', hue='Anxiety',
              data=df, order=anx['CGPA'].value_counts(ascending=False).index,
              ax=ax[1])
ax[1].set_ylabel(None)
ax[1].set_title("Средний балл и тревожность", fontsize = 16)
ax[1].legend(["Нет тревожности","Есть тревожность"])

sns.countplot(y='CGPA', hue='Panic Attack',
              data=df, order=pa['CGPA'].value_counts(ascending=False).index,
              ax=ax[2])
ax[2].set_ylabel(None)
ax[2].set_title("Средний балл и панические атаки", fontsize = 16)
ax[2].legend(["Не было панических атак","Бывали панические атаки"])

plt.show()
```



Анализ психического здоровья:

В [32]:
```
print("Количество студентов, имеющих депрессию: {} студентов\n".format(len(dep['Depression'])))
print("Количество студентов, имеющих тревожность: {} студента\n".format(len(anx['Anxiety'])))
print("Количество студентов, имеющих панические атаки: {} студента\n".format(len(pa['Panic Attack'])))
print("Количество студентов, искавших квалифицированную психологическую помощь: {} студентов\n".format(len(df[df['Treatment']==
```

```
Количество студентов, имеющих депрессию: 35 студентов

Количество студентов, имеющих тревожность: 34 студента

Количество студентов, имеющих панические атаки: 33 студента

Количество студентов, искавших квалифицированную психологическую помощь: 6 студентов
```

Распределение студентов по психическому здоровью:

В [33]:
```
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(15,4))

ax1.pie(df['Depression'].value_counts(), explode=(0.025,0.025),
            labels=("Нет депрессии","Есть депрессия"),
            colors=['lightblue','lightgreen'], autopct='%1.2f%%', startangle=120)

ax2.pie(df['Anxiety'].value_counts(), explode=(0.025,0.025),
            labels=("Нет тревожности","Есть тревожность"),
            colors=['lightblue','lightgreen'], autopct='%1.2f%%', startangle=120)

ax3.pie(df['Panic Attack'].value_counts(), explode=(0.025,0.025),
            labels=("Не было панических атак","Бывали панические атаки"),
            colors=['lightblue','lightgreen'], autopct='%1.2f%%', startangle=120)

plt.suptitle("Распределение студентов по психическому здоровью")
plt.show()
```

Посчитаем количество студентов, столкнувшихся с депрессией и искавших квалифицированную помощь:

В [34]:
```
dep_treat = dep.groupby('Treatment')[['Depression']].count()
dep_treat.rename(columns = {'Depression':'Number of Students (Depression)'},
                 index={0: 'No', 1:'Yes'}, inplace=True)
dep_treat
```

Out[34]:

Treatment	Number of Students (Depression)
No	29
Yes	6

Отобразим на круговой диаграмме данное распределение:

В [35]:
```
plt.figure(figsize=(12,6))
plt.pie(dep_treat['Number of Students (Depression)'], explode=(0.02,0),
        labels=dep_treat.index,
        colors=['palegreen','yellowgreen'], autopct='%1.2f%%', startangle=180)

centre_circle = plt.Circle((0,0),0.8,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.title("(Из студентов, имеющих депрессию)")
plt.suptitle('Искали ли квалифицированную психологическую помощь?')
plt.show()
```

Корреляция

B [36]:
```
plt.figure(figsize = (18, 6))
sns.heatmap(df.corr(), annot = True, cmap = 'viridis', linewidth = 1, fmt = '.3f')
plt.show()
```