МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа №__1__
по дисциплине «Методы машинного обучения»

Тема: «Создание "истории о данных" (Data Storytelling)»

ИСПОЛНИТЕЛЬ: _____Лу Сяои_____
ФИО

группа ИУ5И-22М _____
подпись

"28" Февраль 2023 г.

ПРЕПОДАВАТЕЛЬ: _____
ФИО

_____
подпись

"__"_____2023 г.

Москва - 2023

описание задания

- Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь.
- Для лабораторных работ не рекомендуется выбирать датасеты очень большого размера.
- Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:
- История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
- На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
- Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
- Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.
- История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.
- Сформировать отчет и разместить его в своем репозитории на github.

текст программы и экранные формы с примерами выполнения

Использование набора данных Sleep_Efficiency.csv

```
# подключение библиотек
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import random
import math as math
import seaborn as sns #
import matplotlib.pyplot as plt
import missingno as msno
import plotly.graph_objs as go
import plotly.express as px #
plt.style.use('seaborn-dark')
plt.style.context('grayscale')
%matplotlib inline
import re
from wordcloud import WordCloud, STOPWORDS
```

Importing Data

```
url = "/content/Sleep_Efficiency.csv"
df = pd.read_csv(url)
```

Df

	ID	Age	Gender	Bedtime	Wakeup time	Sleep duration	Sleep efficiency	REM sleep percentage	Deep sleep percentage	Light sleep percentage	Awakenings	con
0	1	65	Female	2021-03-06 01:00:00	2021-03-06 07:00:00	6.0	0.88	18	70	10	0.0	
1	2	69	Male	2021-12-05 02:00:00	2021-12-05 09:00:00	7.0	0.66	24	28	53	3.0	
2	3	40	Female	2021-05-25 21:30:00	2021-05-25 05:30:00	8.0	0.89	20	70	10	1.0	
3	4	40	Female	2021-11-03 02:30:00	2021-11-03 08:30:00	6.0	0.51	28	25	52	3.0	
4	5	57	Male	2021-03-13 01:00:00	2021-03-13 09:00:00	8.0	0.76	27	55	18	3.0	

The "Sleep efficiency" feature is a measure of the proportion of time spent in bed that is actually spent asleep. Additionally, the dataset includes information about each subject's caffeine and alcohol consumption in the 24 hours prior to bedtime, their smoking status, and their exercise frequency.

Caffeine consumption:the amount of caffeine consumed in the 24 hours prior to bedtime (in mg)

Alcohol consumption:the amount of alcohol consumed in the 24 hours prior to bedtime (in oz)

Exercise frequency:the number of times the test subject exercises each week

---

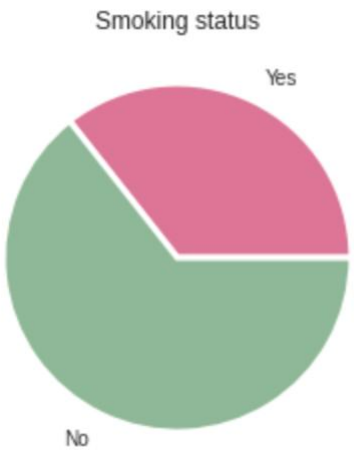
1.PIE

Single column data 'categoric' (Smoking status)

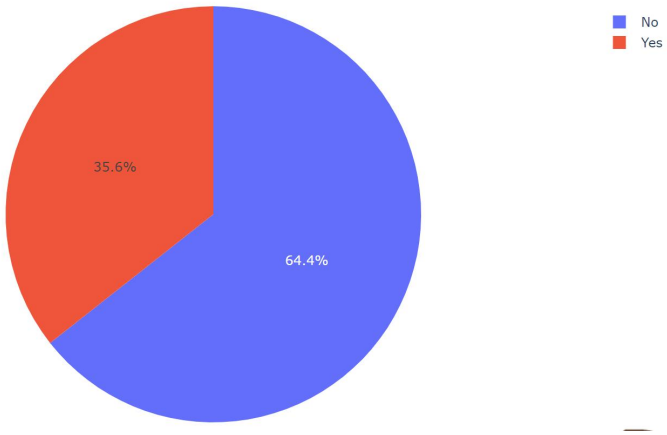
```
df['Smoking status'].value_counts()
```

```
No     291
Yes    161
Name: Smoking status, dtype: int64
```

```
# Method 1
df['Smoking status']
# create data: an array of values
size_of_groups=df['Smoking status']
count = [0, 0]
for i in size_of_groups:
  if i == 'Yes':
    count[0]+=1
  else:
    count[1]+=1
# Create a pieplot
plt.pie(count, labels=['Yes', 'No'],colors=['#DD7596', '#8EB897'],labeldistance=1.15,wedgeprops = { 'li
newidth' : 3, 'edgecolor' : 'white' })
plt.title("Smoking status")
plt.show()
```




```
# Method 2 import plotly.express as px
fig = px.pie(df,names='Smoking status')
fig.show()
```

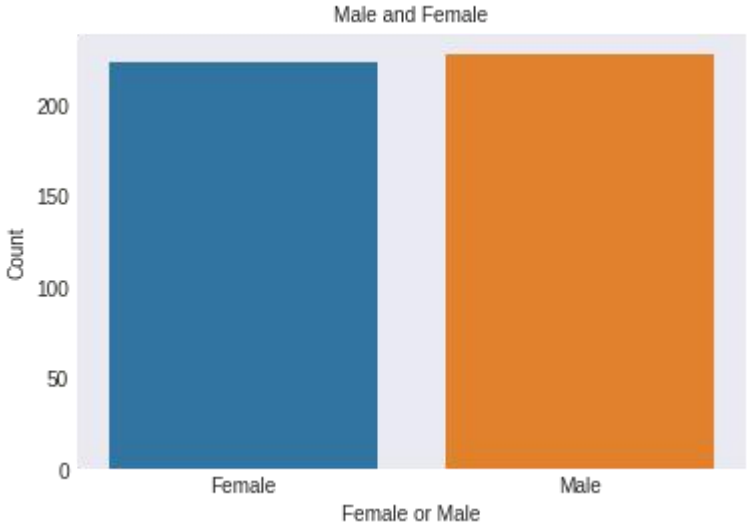


2.BARPLOT

```
df.Gender.value_counts()
```

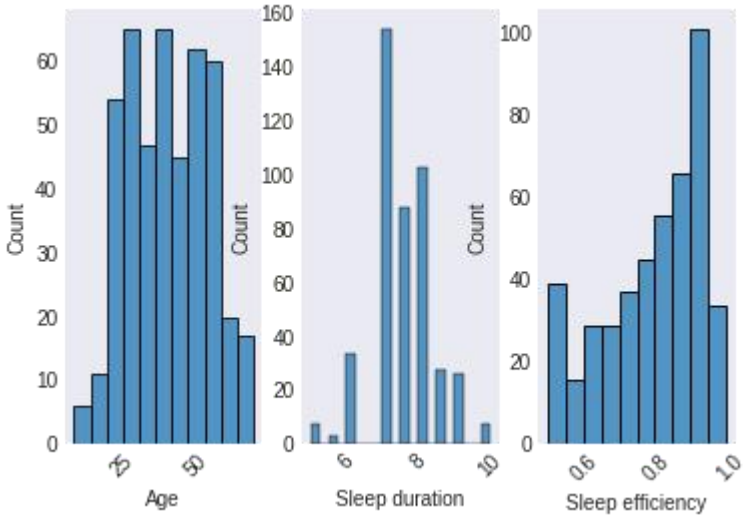
```
Male      228
Female    224
Name: Gender, dtype: int64
```

```
sns.countplot(data=df,x="Gender")
plt.xlabel("Female or Male",fontsize=10)
plt.ylabel("Count",fontsize=10)
plt.title("Male and Female",fontsize=10)
plt.show()
```



3.HISTOGRAM

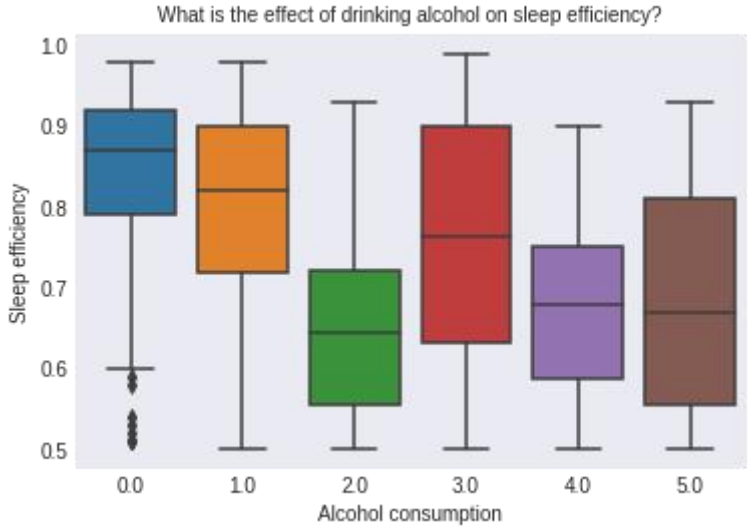
```
# Create a figure with 3 subplots per row
fig, axes = plt.subplots(nrows=1, ncols=3)
axes=axes.ravel()

# Use the first subplot to plot histograms
first_columns=["Age", "Sleep duration", "Sleep efficiency"]
for i, column in enumerate(first_columns):
   sns.histplot(data=df, x=column,ax=axes[i])
   # rotate the x-axis labels by 45 degrees
   axes[i].tick_params(axis='x', rotation=45)

plt.show()
```



4.BOX PLOT

What is the effect of drinking alcohol on sleep efficiency?

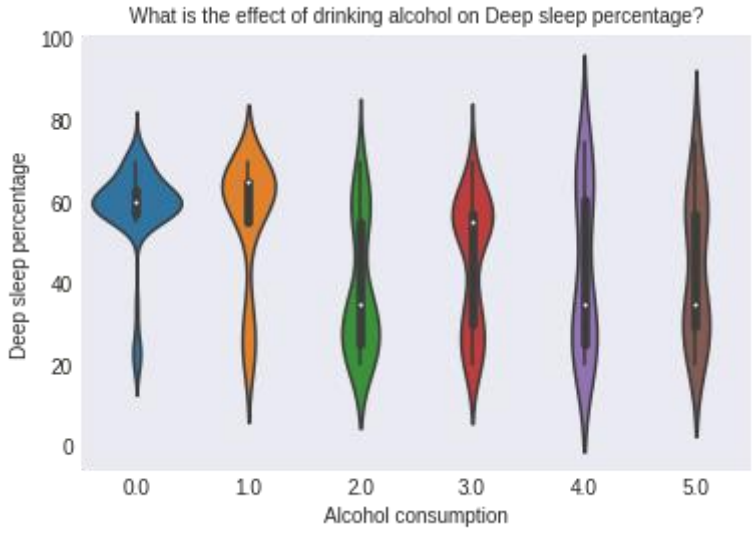
```
sns.boxplot(data=df,x="Alcohol consumption",y="Sleep efficiency")
plt.xlabel("Alcohol consumption",fontsize=10)
plt.ylabel("Sleep efficiency",fontsize=10)
plt.title("What is the effect of drinking alcohol on sleep efficiency?",fontsize=10)
plt.show()
```



Basically, the less alcohol you drink, the more efficient you are at sleeping.

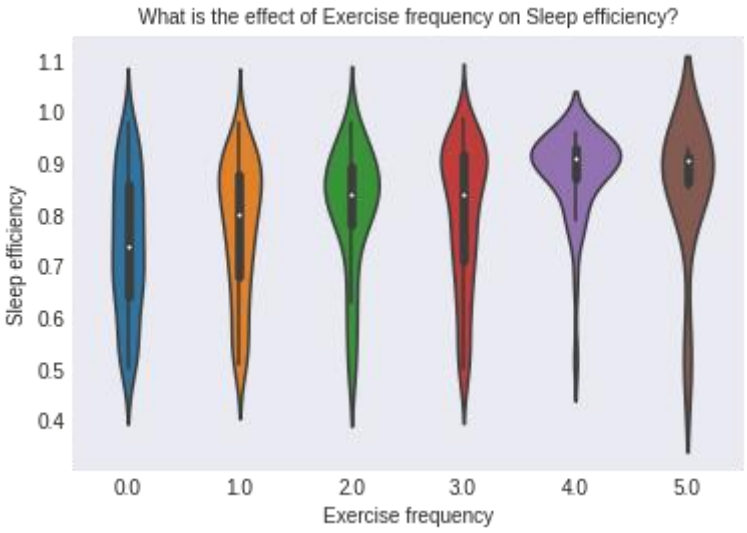
5.VIOLIN PLOT

```
sns.violinplot(data=df,x="Alcohol consumption",y="Deep sleep percentage")
plt.xlabel("Alcohol consumption",fontsize=10)
plt.ylabel("Deep sleep percentage",fontsize=10)
plt.title("What is the effect of drinking alcohol on Deep sleep percentage?",fontsize=10)
plt.show()
```



Basically, the lower the alcohol consumption, the higher the percentage of deep sleep.

```
sns.violinplot(data=df,x="Exercise frequency",y="Sleep efficiency")
plt.xlabel("Exercise frequency",fontsize=10)
plt.ylabel("Sleep efficiency",fontsize=10)
plt.title("What is the effect of Exercise frequency on Sleep efficiency?",fontsize=10)
plt.show()
```

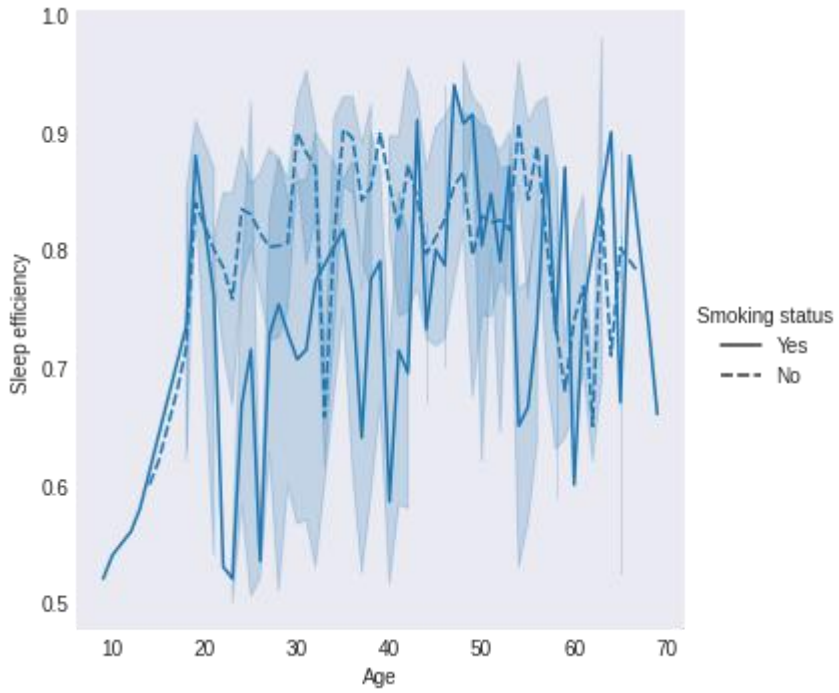


The more frequently you exercise during the week, the more efficient your sleep will be.

6.LINEPLOT

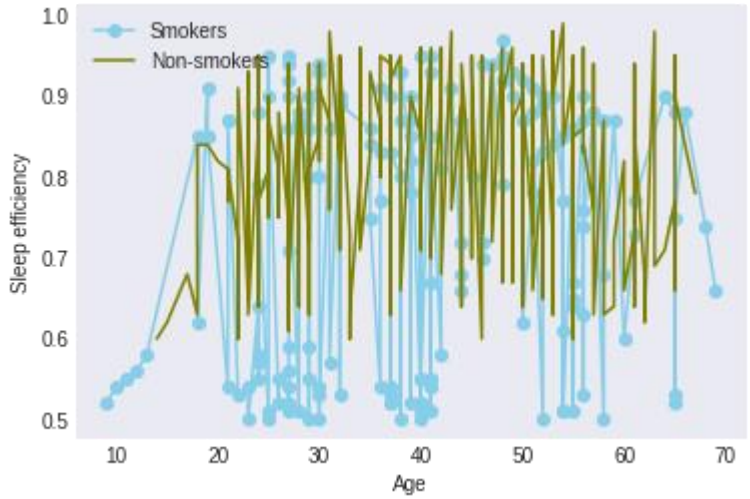
ONE CAT SEVERAL NUM, A NUM. IS ORDERED (Age)

```
# Method 1
sns.relplot(
   data=df, kind="line",
   x="Age", y="Sleep efficiency", style="Smoking status",
)
plt.show()
```



```
# Method 2
# Data
df1=df[['Smoking status','Age','Sleep efficiency']].copy()
df2=df1.sort_values(by=['Age'],ascending=True)

se_yes=list(df2[df2['Smoking status']=='Yes']['Sleep efficiency'])
```

```
se_no=list(df2[df2['Smoking status']=='No']['Sleep efficiency'])

age_yes=list(df2[df2['Smoking status']=='Yes']['Age'])
age_no=list(df2[df2['Smoking status']=='No']['Age'])

# multiple line plots
plt.plot(age_yes,se_yes,label="Smokers",marker='o',color='skyblue')
plt.plot(age_no,se_no,label="Non-smokers",marker='',color='olive')

# show legend
plt.legend()
plt.xlabel("Age")
plt.ylabel("Sleep efficiency")
# show graph
plt.show()
```

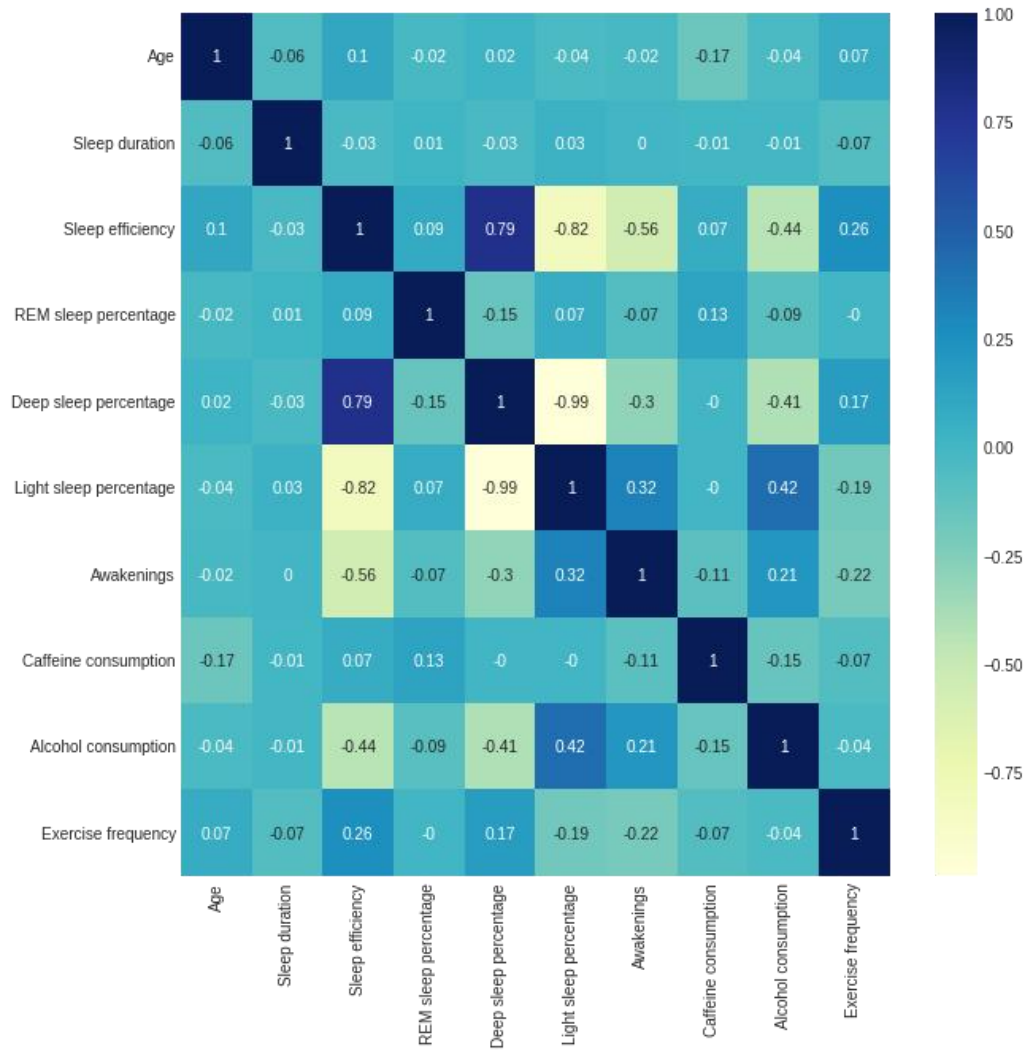


7.HEATMAP

```
# Drop the "ID" 'Bedtime','Wakeup time' column
dfnew = df.drop(labels=['ID','Bedtime','Wakeup time'], axis=1)
dfnew.head()
```

	Age	Gender	Sleep duration	Sleep efficiency	REM sleep percentage	Deep sleep percentage	Light sleep percentage	Awakenings	Caffeine consumption	Alcohol consumption	Smoki stat
0	65	Female	6.0	0.88	18	70	10	0.0	0.0	0.0	Y
1	69	Male	7.0	0.66	24	28	53	3.0	0.0	3.0	Y
2	40	Female	8.0	0.89	20	70	10	1.0	0.0	0.0	N
3	40	Female	6.0	0.51	28	25	52	3.0	50.0	5.0	Y
4	57	Male	8.0	0.76	27	55	18	3.0	0.0	3.0	N

```
# plot the correlation matrix after cleaning
corr = dfnew.corr()
plt.figure(figsize=(10,10))
sns.heatmap(corr.round(2), annot=True, cmap='YlGnBu')
plt.show()
```



'Sleep efficiency' and 'Deep sleep percentage' are significantly correlated