Лабораторна робота №2

Візуалізації даних в Python з Matplotlib і Seaborn

Мета роботи: набути навичок роботи з бібліотеками Python, опанувати основні методи бібліотек Seaborn і Matplotlib, навчитися проводити візуальний аналіз даних на представленому набір даних

Література

Галерея прикладів різної графіки в matplotlib - https://matplotlib.org/gallery.html
Повний список команд для pyplot - https://matplotlib.org/api/pyplot_summary.html
Документація: https://seaborn.pydata.org/generated/seaborn.heatmap.html
https://seaborn.pydata.org/generated/seaborn.JointGrid.html
https://seaborn.pydata.org/generated/seaborn.jointplot.html
https://seaborn.pydata.org/generated/seaborn.kdeplot.html
Styling plots with Seaborn -http://jose-coto.com/styling-with-seaborn

Зміст роботи

Завдання 1. Провести візуальний аналіз даних

Вхідні дані та структура даних представлена у лабораторній роботі №1.

Цільова ознака (яку необхідно буде прогнозувати):

Наявність серцево-судинних захворювань за результатами класичного лікарського огляду (cardio).

Вік заданий в днях. Значення показників холестерину і глюкози представлені одним з трьох класів: *норма, вище норми, значно вище норми*. Значення суб'єктивних ознак - бінарні.

З бібліотек знадобляться :

```
import numpy as np
import pandas as pd
import matplotlib.ticker
import matplotlib.pyplot as plt
import seaborn as sns

# ігноруємо warnings
import warnings
warnings.filterwarnings("ignore")
```

Провести налаштування зовнішнього вигляду графіків у seaborn:

```
sns.set_context(
 "notebook",
 font_scale = 1.5,
 rc = {
  "figure.figsize" : (12, 9),
  "axes.titlesize" : 18
   }
)
```

Зчитати дані з CSV-файлу в об'єкт pandas *DataFrame*.

```
df = pd.read_csv('mlbootcamp5_train.csv', sep=';',index_col='id')
```

або

```
df=pd.read_csv('http://nbviewer.jupyter.org/github/Yorko/mlcourse_open/blob/master/data/mlbootcamp5_train.csv', sep=';', index_col='id')
```

Подивитися на перші 5 записів та розмір Dataset.

Результат:

id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	17474	1	156	56.0	100	60	1	1	0	0	0	0

```
(70000, 12)
```

В рамках завдання для простоти необхідно буде працювати з вибіркою данних, що має кількісні і категоріальні ознаки. Чистити дані від викидів і помилок **НЕ ПОТРІБНО**, крім тих випадків, де про це явно зазначено.

Всі візуалізації рекомендовано проводити за допомогою бібліотеки *Seaborn.*

Проведемо невеликий EDA (розвідувальний аналіз даних)

Розвідувальний аналіз займається попереднім експрес-аналізом даних шляхом їх перетворення та/або представлення у зручному вигляді: графічному, табличному, за допомогою схем, діаграм тощо.

*Термін «розвідувальний аналіз» (*exploratory data analysis — EDA*) був вперше введений Дж. Тьюкі, він же сформулював основні його завдання:*

— максимальне проникнення в дані;

— вибір найважливіших ознак:

— аналіз основних структур;

— виявлення відхилень і аномалій;

— перевірка основних гіпотез щодо законів розподілу і взаємозв'язків;

— апробація моделей.

Отже, на етапі розвідувального аналізу формується уявлення про тип даних, оцінюється їхня однорідність, з'ясовується структура об'єкта моделювання, виявляються взаємозв'язки між ознаками. За допомогою дескриптивних статистик описуються й узагальнюються основні властивості об'єкта моделювання, частотний аналіз і графічна візуалізація

допомагають визначитися щодо методів подальшого аналізу і моделей, які треба застосувати, а також яких результатів можна очікувати. Без розвідувального аналізу даних моделювання буде наосліп.

Для початку завжди потрібно подивитися на значення, які приймають змінні.

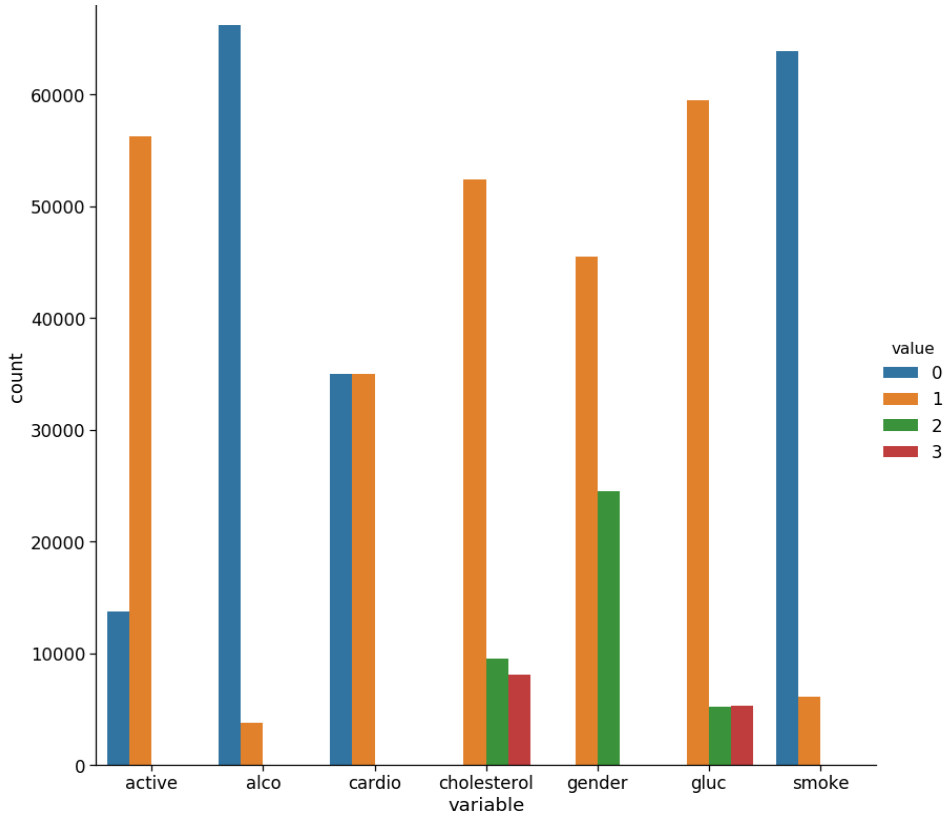
Переведемо дані в «Long Format» - представлення та візуалізуємо дані за допомогою *factorplot* **кількість значень, які приймають категоріальні змінні.**

```
df_uniques = pd.melt(frame=df, value_vars=['gender','cholesterol'
,'gluc', 'smoke', 'alco','active', 'cardio'])

df_uniques = pd.DataFrame(df_uniques.groupby(['variable','value']
)['value'].count())\
 .sort_index(level=[0, 1]) \
 .rename(columns={'value': 'count'}) \
 .reset_index()

sns.factorplot(x='variable', y='count', hue='value', data=df_uniq
ues, kind='bar', size=12)
```

Результат:



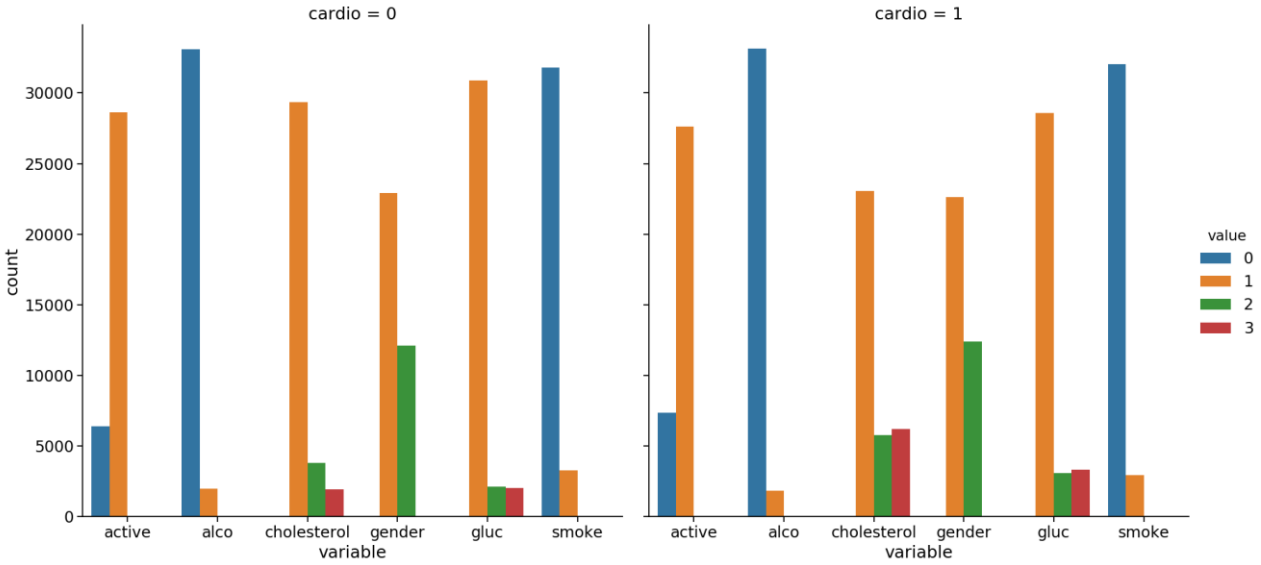
На графіку можна побачити, що класи цільової змінної *cardio* збалансовані, відмінно!

Можна також розбити елементи навчаючої вибірки за значеннями цільової змінної: іноді на таких графіках можна відразу побачити найбільш значиму ознаку.

```
df_uniques = pd.melt(frame=df, value_vars=['gender','cholesterol'
,'gluc','smoke', 'alco',
             'active'],id_vars=['cardio'])
df_uniques = pd.DataFrame(df_uniques.groupby(['variable', 'value'
,
  'cardio'])['value'].count()) \
 .sort_index(level=[0, 1]) \
 .rename(columns={'value': 'count'}) \
 .reset_index()

sns.factorplot(x='variable', y='count', hue='value', col='cardio'
, data=df_uniques, kind='bar', size=9)
```

Результат:



По отриманим графікам можна побачити, що в залежності від цільової змінної (*cardio*) сильно змінюється розподіл холестерину і глюкози.

Статистику за унікальними значеннями ознак можна отримати за допомогою наступного коду:

```
for c in df.columns:
  n = df[c].nunique()
  print(c)
  if n <= 3:
    print(n, sorted(df[c].value_counts().to_dict().items()))
  else:
    print(n)
print(10 * '-')
```

Результат:

```
age
8076
gender
2 [(1, 45530), (2, 24470)]
height
109
weight
287
ap_hi
153
ap_lo
157
cholesterol
3 [(1, 52385), (2, 9549), (3, 8066)]
gluc
3 [(1, 59479), (2, 5190), (3, 5331)]
smoke
2 [(0, 63831), (1, 6169)]
alco
2 [(0, 66236), (1, 3764)]
active
2 [(0, 13739), (1, 56261)]
cardio
2 [(0, 35021), (1, 34979)]
----------
```

Разом:

- П’ять кількісних ознак (без id)
- Сім категоріальних

Завдання 2. Самостійно проведіть візуальний аналіз даних

1. Кореляційна матриця

Для того щоб краще зрозуміти ознаки в Dataset, можна порахувати матрицю коефіцієнтів кореляції між ознаками.

Побудуйте кореляційну матрицю (heatmap). Матриця формується засобами Pandas, зі стандартним значенням параметрів.

Визначте які дві ознаки найбільше корелюють (за Пірсоном) з ознакою height?

2. Розподіл росту людини за гендерною ознакою

В процесі дослідження унікальних значень стать кодується значеннями 1 або 2, визначити хто є хто потрібно було у лабораторній роботі №1, для цього потрібно було використовувати середні значення зросту (або ваги) при різних значеннях ознаки *gender*. Тепер представте те ж саме, але графічно.

Проведіть візуалізацію даних: зріст і стать (violinplot). Використовуйте параметри:

- *hue* ***- для розбивки за статтю;***
- *scale* ***- для оцінки кількості кожної статі.***

Для коректного відтворення, перетворіть *DataFrame* в «Long Format»-представлення за допомогою функції *melt* в *pandas*.

Побудуйте violinplot для статі, росту і ваги.

Побудуйте на одному графіку два окремих *kdeplot* росту і ваги, окремо для чоловіків і жінок. На ньому різниця буде більш наочною, але не можна буде оцінити кількість чоловіків / жінок.

3. Рангова кореляція

У більшості випадків достатньо скористатися лінійним коефіцієнтом кореляції Пірсона для виявлення закономірностей у даних, але для подальших розрахунків використаємо рангову кореляцію Спірмена, яка допоможе виявити пари, в яких менший ранг з варіаційного ряду однієї ознаки завжди передує більшому іншого (або навпаки, в разі негативної кореляції).

Рангова кореляція — метод кореляційного аналізу, який використовується для сукупностей невеликого обсягу і для кількісних ознак, якщо їхня сукупність не має нормального розподілу.

Для змінних, що належать до порядкової шкали, або для змінних, що не підкоряються нормальному розподілу, а також для змінних, приналежних до інтервальної шкали, замість коефіцієнта Пірсона розраховується рангова кореляція за Спірменом. Для цього окремим значенням змінних привласнюються рангові місця, що з годом обробляються за допомогою відповідних формул.

Ще одним варіантом рангових коефіцієнтів кореляції є коефіцієнти Кендалла. У цьому методі одна змінна представляється у вигляді монотонної послідовності в порядку зростання величин; іншій змінній привласнюються відповідні рангові місця. Кількість інверсій (порушень монотонності в порівнянні з першим рядом) використовується у формулі для кореляційних коефіцієнтів. Застосування коефіцієнта Кендалла є кращим, якщо у вихідних даних зустрічаються викиди.

***Рангова кореляція Спірмена** (кореляція рангів) — найпростіший спосіб визначення міри зв'язку між факторами.*

```
sns.heatmap(df.corr(method='spearman'))
```

Назва методу свідчить про те, що зв'язок визначають між рангами, тобто рядами одержаних кількісних значень, ранжованих у порядку зниження або зростання. Треба мати на увазі, що, по-перше, рангову кореляцію не рекомендовано проводити, якщо зв'язок пар менший чотирьох і більший двадцяти; по-друге, рангова кореляція дає змогу визначати зв'язок і в іншому випадку, якщо значення мають напівкількісний характер, тобто не мають числового виразу, відображають чіткий порядок прямування цих величин; по-третє, рангову кореляцію доцільно застосовувати в тих випадках, коли достатньо одержати приблизні дані.

Побудуйте кореляційну матрицю, використовуючи коефіцієнт Спірмена.

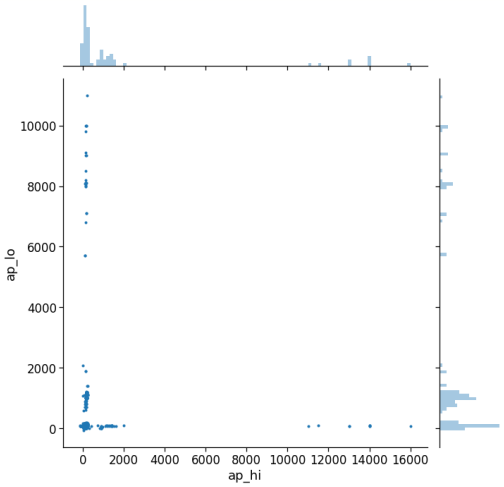
3.1 Які ознаки найбільше корелюють одна з одною за Спірменом?

3.2 Чому значення рангової кореляції в цих ознаках таке велике (відносно)?

4. Спільний розподіл ознак

Побудуйте спільний графік розподілу (*jointplot)* двох ознак, що найбільш корелюють між собою за Спирменом.

```
sns.jointplot(df['ap_hi'],
              df['ap_lo'],
              size=10,
              marker='.',
              marginal_kws=dict(bins=100, rug=False, hist_kws={'log'
: True}))
```



Здається, графік вийшов неінформативним через викиди в значеннях. Необхідно побудувати цей графік, але за логарифмічною шкалою (щоб не отримувати *OverflowError* необхідно відфільтрувати значення менше або рівні нулю).

```
data_filtered = df[(df['ap_hi'] > 0) & (df['ap_lo'] > 0)][['ap_lo', 'ap_hi
']].apply(np.log1p)
```

```
data_filtered.describe()
```

	ap_lo	ap_hi
count	69971.000000	69971.000000
mean	4.438763	4.839140
std	0.325333	0.184955
min	0.693147	0.693147
25%	4.394449	4.795791
50%	4.394449	4.795791
75%	4.510860	4.948760
max	9.305741	9.681656

```
g = sns.jointplot(
```

```
    data_filtered['ap_hi'],
    data_filtered['ap_lo'],
    size=10,
    stat_func=None,
    marginal_kws=dict(bins=100, rug=False,
   hist_kws={'log': True}),marker='.'
)
```

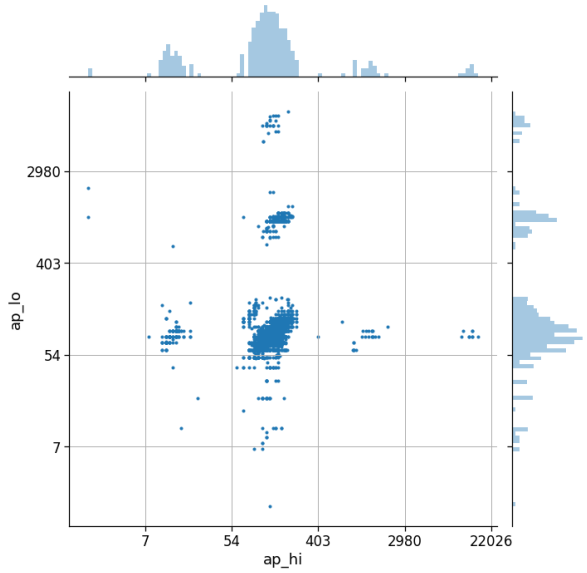
Побудуйте спільний графік розподілу (jointplot) двох ознак, що найбільш корелюють між собою за Спірменом.

Для побудови сітки на графіку скористайтеся кодом:

```
g.ax_joint.grid(True)
```

Для перетворення логоріфмфчних значень на реальні:

```
g.ax_joint.yaxis.set_major_formatter(matplotlib.ticker.FuncFormatter(lambd
a x, pos: str(round(int(np.exp(x))))))
g.ax_joint.xaxis.set_major_formatter(matplotlib.ticker.FuncFormatter(lambd
a x, pos: str(round(int(np.exp(x))))))
```



– ***Скільки чітко виражених кластерів вийшло на спільному графіку обраних ознак, за логарифмічною шкалою? Під кластером іноді розуміють щільне скупчення точок, в околиці яких досить мало одиночних і які візуально відділені від інших кластерів.***

5. Вік

Порахуємо, скільки повних років було респондентам на момент їх занесення в базу.

```
df['age_years'] = (df['age'] // 365.25).astype(int)
```

Побудуйте Countplot, де на осі абсцис буде відзначений вік, на осі ординат - кількість. Кожне значення віку повинне мати два стовпці, що

відповідають кількості осіб кожного класу cardio (здоровий / хворий) даного віку.

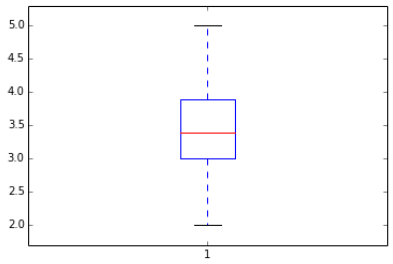
– ***В якому віці кількість пацієнтів з ССЗ вперше стає більше, ніж здорових?***

Контрольні запитання

1. Для чого використовується «Long Format»- представлення?

2. За допомогою якого графіка можна дізнатися середнє значення (мат. очікування) і розкид значень (дисперсію) для різних категорій даних.

3. Опишіть основні елементи наступної діаграми і яким чином можна її отримати:



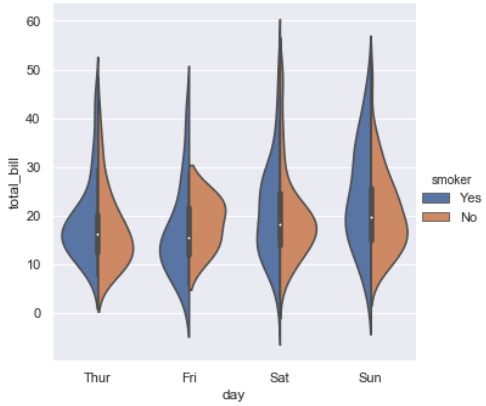
4. Якщо використовувати метод *DataFrame.plot()* з параметром *kind = 'bar'*, який вид діаграми можна отримати?

5. Тип графіків *Pie Chart* (Пиріжковий графік) відмінно підходить для відображення часток, які належать частини даних. Опішить метод побудови.

6. Опишіть призначення графіка *Heat Map* (Теплова карта).

7. Для чого призначені функції *plt.show()* і *plt.draw()?*

8. Як можна отримати наступний графік:



9. Який тип графіка можна отримати:

```
sns.pairplot(data=iris, hue="species")
```