Технологическая практика

Работа с Pandas

In [2]:
```
import pandas as pd
%matplotlib inline
from scipy.stats import norm
import numpy as np
import matplotlib.pyplot as plt
```

Будем работать с датасетом Pima Indian Diabetes - это набор данных из Национального института диабета, болезней органов пищеварения и почек. Целью набора данных является диагностическое прогнозирование наличия диабета у пациента. Несколько ограничений были наложены на выбор этих экземпляров из большой базы данных. В частности, все пациенты здесь - женщины в возрасте от 21 года, индийского происхождения.

In [3]:
```
data = pd.read_csv('pima-indians-diabetes.csv')
data.tail(10)
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
758	1	106.0	76.0	NaN	NaN	37.5	0.19
759	6	190.0	92.0	NaN	NaN	35.5	0.27
760	2	88.0	58.0	26.0	16.0	28.4	0.76
761	9	170.0	74.0	31.0	NaN	44.0	0.40
762	9	89.0	62.0	NaN	NaN	22.5	0.14
763	10	101.0	76.0	48.0	180.0	32.9	0.17
764	2	122.0	70.0	27.0	NaN	36.8	0.34
765	5	121.0	72.0	23.0	112.0	26.2	0.24
766	1	126.0	60.0	NaN	NaN	30.1	0.34
767	1	93.0	70.0	31.0	NaN	30.4	0.31

Описание данных:

- **Pregnancies** - данная единица отображает количество беременностей, единицы измерения - целые числа от 0 до N. Тип переменной - количественная, дискретная.
- **Glucose** - данная единица отображает уровень глюкозы в крови, единицы измерения - целые числа. Тип переменной - количественная, дискретная.
- **BloodPressure** - данная единица отображает артериальное давление, единицы измерения - миллиметры р/с, целые числа. Тип переменной - количественная, дискретная.
- **SkinThickness** - данная единица отображает обхват трицепса в миллиметрах, целые числа. Тип переменной - количественная, дискретная.
- **Insulin** - данная единица отображает уровень инсулина в крови, целые числа. Тип переменной - количественная, дискретная.

- **BMI** - данная единица отображает индекс массы тела. Тип переменной - количественная, непрерывная.
- **DiabetesPedigreeFunction** - данная единица отображает риск наследственного диабета в зависимости наличия диабета у родственников. Выражается десятичной дробью от 0 до 1. Тип переменной - количественная, непрерывная.
- **Age** - данная единица отражает возраст в целых числах. Тип переменной - количественная, дискретная.
- **Class** - данная единица отражает наличие диабета у субъекта, выражена 0(здоров) или 1(болен). Тип переменной - категориальная, бинарная.

Задание 1.

Как вы видите, в данных много пропусков (NaN). Посчитайте количество пропусков в каждом из столбцов.

In [4]:
```
data[pd.isna(data)].size
```

Out[4]: 6912

Задание 2.

Замените все пропуски дискретных признаков соответствующими медианами, непрерывных признаков - средними значениями.

In [38]:
```
disck = ["Pregnancies", "Glucose", "BloodPressure", "SkinThickness", "Insulin
for col in disck:
    for index in data[col][pd.isna(data[col])].index:
        data[col][index] = data[col].mean()

data.head(10)
```

Out[38]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunct
0	6	148.0	72.000000	35.00000	155.548223	33.6	0.(
1	1	85.0	66.000000	29.00000	155.548223	26.6	0.:
2	8	183.0	64.000000	29.15342	155.548223	23.3	0.(
3	1	89.0	66.000000	23.00000	94.000000	28.1	0.:
4	0	137.0	40.000000	35.00000	168.000000	43.1	2.:
5	5	116.0	74.000000	29.15342	155.548223	25.6	0.:
6	3	78.0	50.000000	32.00000	88.000000	31.0	0.:
7	10	115.0	72.405184	29.15342	155.548223	35.3	0.:
8	2	197.0	70.000000	45.00000	543.000000	30.5	0.:
9	8	125.0	96.000000	29.15342	155.548223	NaN	0.:

Задание 3.

Вычислите основные статистики (минимум, максимум, среднее, дисперсию, квантили) для всех столбцов.

In [26]:
```
rows = ["min", "max", "Mean", "dispersion", "quantiles"]
statistics = pd.DataFrame(columns=data.columns)
```

```
for col in data:
    statistics[col] = [data[col].min(), data[col].max(), data[col].mean(), np
statistics.index = rows
statistics.head(5)
```

Out[26]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedig
min	0	44	24	7	14	18.2	
max	17	199	122	99	846	67.1	
Mean	3.84505	121.687	72.4052	29.1534	155.548	32.4575	
dispersion	11.3393	925.141	146.131	77.18	7219.18	47.8921	
quantiles	[0, 17]	[44.0, 199.0]	[24.0, 122.0]	[7.0, 99.0]	[14.0, 846.0]	[nan, nan]	

Задание 4.

У скольких женщин старше 50 лет обнаружен диабет?

In [27]:
```
len(data["Class"][data["Age"] > 50])
```

Out[27]: 81

Задание 5.

Найдите трех женщин с наибольшим числом беременностей.

In [28]:
```
max_1 = data["Pregnancies"].max()
max_2 = data["Pregnancies"][data["Pregnancies"] != max_1].max()
max_3 = data["Pregnancies"][(data["Pregnancies"] != max_1) & (data["Pregnanci
print(max_1, max_2, max_3)
```

```
17 15 14
```

Задание 6.

Сколько женщин возраста между 30 и 40 успело родить 3 или более детей?

In [29]:
```
data["Pregnancies"][(data["Age"] >= 30) & (data["Age"] <= 30)].size
```

Out[29]: 21

Задание 7.

Нормальным кровяным давлением будем считать давление в диапазоне [80-89]. У какого процента женщин давление нормальное?

In [30]:
```
data["BloodPressure"][(data["BloodPressure"] >= 80) & (data["BloodPressure"]
```

Out[30]: 18.880208333333336

Задание 8.

Считается, что BMI >= 30 - это признак ожирения. У скольких женщин с признаками ожирения кровяное давление выше среднего?

In [31]:
```
mean = data["BloodPressure"].mean()
obesity = data[data["BMI"] >= 30]
len(obesity[obesity > mean])
```

Out[31]: 472

Задание 9.

Сравните средние значения для признаков **Glucose, BloodPressure, Insulin** среди тех, у кого обнаружен диабет, и тех, у кого его нет.

In [32]:
```
# Glucose ( ͡° ͜ʖ ͡°)づ ━ ☆・*。
Glucose = [data["Glucose"][data["Class"] == 1].mean(), data["Glucose"][data['
print(Glucose)
```

```
[142.16557285655603, 110.71012057667103]
```

In [33]:
```
# BloodPressure ( ͡° ͜ʖ ͡°)づ ━ ☆・*。
BloodPressure = [data["BloodPressure"][data["Class"] == 1].mean(), data["Bloc
print(BloodPressure)
```

```
[75.1473244283358, 70.93539699863574]
```

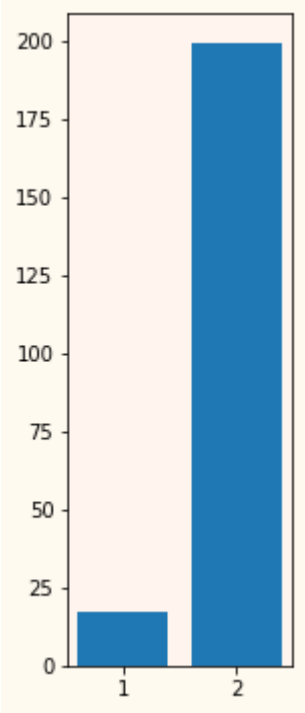
In [34]:
```
# Insulin ( ͡° ͜ʖ ͡°)づ ━ ☆・*。
Insulin = [data["Insulin"][data["Class"] == 1].mean(), data["Insulin"][data['
print(Insulin)
```

```
[180.4315478445337, 142.2107614213198]
```

Задание 10.

Постройте гистограммы для любых двух количественных признаков.

In [48]:
```
# ( ͡° ͜ʖ ͡°)づ ━ ☆・*。

x = [1, 2]
y = [data["Pregnancies"].max(), data["Glucose"].max()]

fig, ax = plt.subplots()

ax.bar(x, y)

ax.set_facecolor('seashell')
fig.set_facecolor('floralwhite')
fig.set_figwidth(2)    #  ширина Figure
fig.set_figheight(6)    #  высота Figure

plt.show()
```

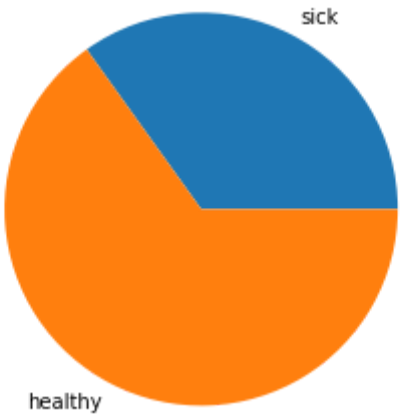


Задание 11.

Постройте круговую диаграмму для признака **Class**.

In [51]:
```
# ( ͡° ͜ʖ ͡°)づ ━ ☆・*。
vals = [data["Class"][data["Class"] == 1].size / data["Class"].size,
        data["Class"][data["Class"] == 0].size / data["Class"].size]
labels = ["sick", "healthy"]

fig, ax = plt.subplots()
ax.pie(vals, labels=labels)
ax.axis("equal")
```

Out[51]: (-1.1103917189999, 1.1004948773786571, -1.106452141145052, 1.111637571647217)



Задание 12.

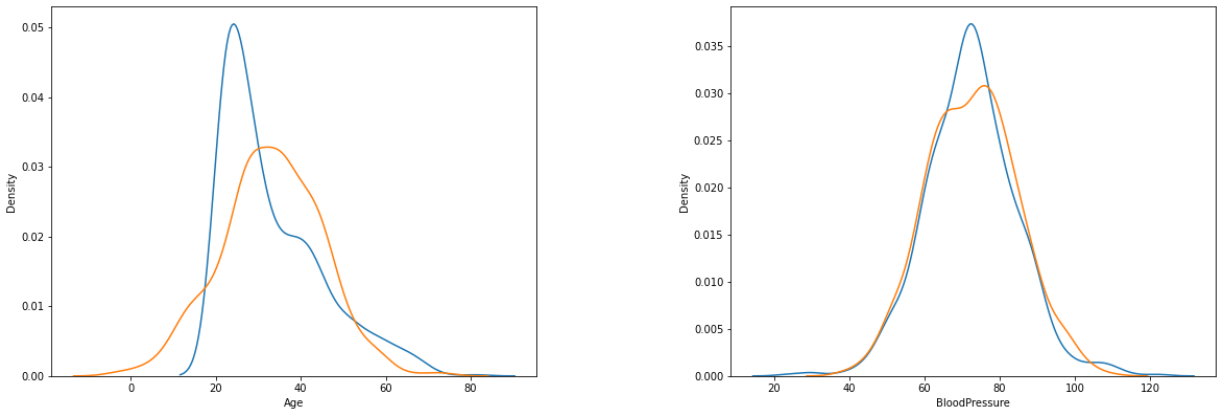
Постройте распределения для признаков **Age** и **BloodPressure** и сравните оба распределения с нормальным.

In [58]:
```
# ( ͡° ͜ʖ ͡°)づ ━ ☆・*。
# plt.scatter(data["Age"], np.arange(data["Age"].size))
import seaborn as sns
x = np.random.normal(loc=data['Age'].mean(), scale=data['Age'].std(), size=le
y = np.random.normal(loc=data['BloodPressure'].mean(), scale=data['BloodPress
```

```
fig, axs = plt.subplots(1, 2,figsize=(20, 5))
plt.subplots_adjust(top = 0.99, bottom=0.01, hspace=1.5, wspace=0.4)
sns.distplot(data['Age'], hist=False, ax=axs[0],label='распределение для приз
sns.distplot(x, hist=False, ax=axs[0],label='Нормальное распределение')

sns.distplot(data['BloodPressure'], hist=False, ax=axs[1],label='распределени
sns.distplot(y, hist=False, ax=axs[1],label='Нормальное распределение')
```

```
/home/dima/botva/1semestr/technological_prackt/MLS/mls/lib/python3.8/site-pac
kages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibility) or `k
deplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/home/dima/botva/1semestr/technological_prackt/MLS/mls/lib/python3.8/site-pac
kages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibility) or `k
deplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/home/dima/botva/1semestr/technological_prackt/MLS/mls/lib/python3.8/site-pac
kages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibility) or `k
deplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
/home/dima/botva/1semestr/technological_prackt/MLS/mls/lib/python3.8/site-pac
kages/seaborn/distributions.py:2551: FutureWarning: `distplot` is a deprecate
d function and will be removed in a future version. Please adapt your code to
use either `displot` (a figure-level function with similar flexibility) or `k
deplot` (an axes-level function for kernel density plots).
  warnings.warn(msg, FutureWarning)
```

```
Out[58]: <AxesSubplot:xlabel='BloodPressure', ylabel='Density'>
```

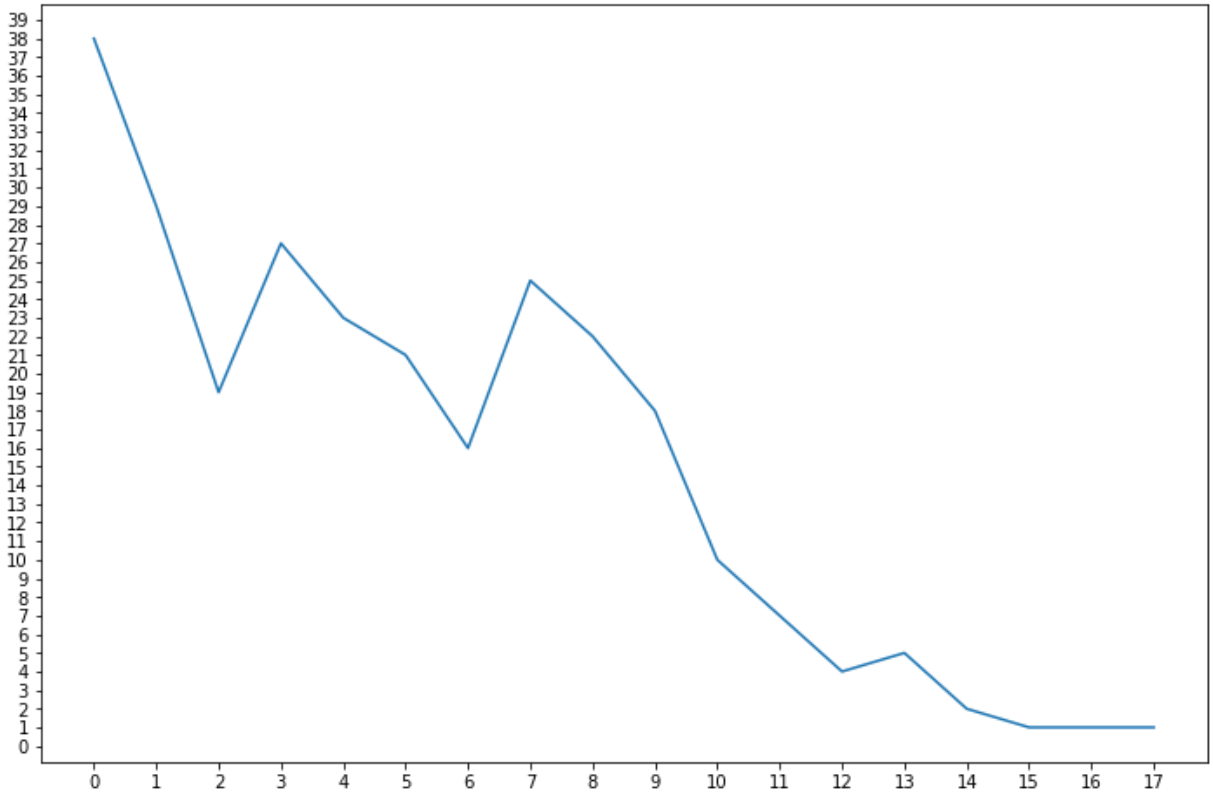


Задание 13.

Постройте следующий график: среднее число больных диабетом в зависимости от числа беременностей.

```
In [44]: # ( ͡° ͜ʖ ͡°)づ ━━ ☆•*｡
import matplotlib.ticker as ticker

diabetes = data[data["Class"] == 1]
unique_pregnancies = dict.fromkeys(np.sort(diabetes["Pregnancies"].unique()),

for i in unique_pregnancies:
    unique_pregnancies[i] = diabetes["Pregnancies"][diabetes["Pregnancies"] =

fig, ax = plt.subplots()
ax.plot(unique_pregnancies.keys(),unique_pregnancies.values())
```

```
#  Устанавливаем интервал основных делений:
ax.xaxis.set_major_locator(ticker.MultipleLocator(1))
ax.yaxis.set_major_locator(ticker.MultipleLocator(1))


fig.set_figwidth(12)
fig.set_figheight(8)

plt.show()
```



Задание 14.

Добавьте новый бинарный признак:

wasPregnant ∈ {0,1} - была женщина беременна (1) или нет (0)

In [74]:
```
# ( ͡° ͜ʖ ͡°)づ ━ ☆ﾟ.*･｡ﾟ
data["wasPregnant"] = [1 if i != 0 else 0 for i in data["Pregnancies"]]
data.head(10)
```

Out[74]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunct
0	6	148.0	72.000000	35.00000	155.548223	33.6	0.(
1	1	85.0	66.000000	29.00000	155.548223	26.6	0.:
2	8	183.0	64.000000	29.15342	155.548223	23.3	0.(
3	1	89.0	66.000000	23.00000	94.000000	28.1	0.:
4	0	137.0	40.000000	35.00000	168.000000	43.1	2.:
5	5	116.0	74.000000	29.15342	155.548223	25.6	0.:
6	3	78.0	50.000000	32.00000	88.000000	31.0	0.:
7	10	115.0	72.405184	29.15342	155.548223	35.3	0.:
8	2	197.0	70.000000	45.00000	543.000000	30.5	0.:
9	8	125.0	96.000000	29.15342	155.548223	NaN	0.:

Задание 15.

Сравните процент больных диабетом среди женщин, которые были беременны и не были.

In [76]:
```
# ( ͡° ͜ʖ ͡°)づ ━ ☆ﾟ.*･｡ﾟ
print(data["wasPregnant"][data["wasPregnant"] == 1].size / data["wasPregnant"
      data["wasPregnant"][data["wasPregnant"] == 0].size / data["wasPregnant"]
```

```
85.546875 14.453125
```

Задание 16.

Добавьте новый категориальный признак **bodyType** на основе столбца BMI:

BMI Categories:

Underweight = <18.5

Normal weight = 18.5–24.9

Overweight = 25–29.9

Obesity = BMI of 30 or greater

Признак должен принимать значения Underweight, Normal weight, Overweight и Obesity.

In [79]:
```
# ( ͡° ͜ʖ ͡°)づ ━ ☆ﾟ.*･｡ﾟ
data["bodyType"] = ["Underweight" if i <= 18.5 else "Normal weight" if i > 18
                    else "Normal weight" if i > 25 and i < 29.9 else "Obesity
data.head(15)
```

Out[79]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunc
0	6	148.0	72.000000	35.00000	155.548223	33.6	0
1	1	85.0	66.000000	29.00000	155.548223	26.6	0
2	8	183.0	64.000000	29.15342	155.548223	23.3	0
3	1	89.0	66.000000	23.00000	94.000000	28.1	0
4	0	137.0	40.000000	35.00000	168.000000	43.1	2
5	5	116.0	74.000000	29.15342	155.548223	25.6	0
6	3	78.0	50.000000	32.00000	88.000000	31.0	0
7	10	115.0	72.405184	29.15342	155.548223	35.3	0
8	2	197.0	70.000000	45.00000	543.000000	30.5	0
9	8	125.0	96.000000	29.15342	155.548223	NaN	0
10	4	110.0	92.000000	29.15342	155.548223	37.6	0
11	10	168.0	74.000000	29.15342	155.548223	38.0	0
12	10	139.0	80.000000	29.15342	155.548223	27.1	1
13	1	189.0	60.000000	23.00000	846.000000	30.1	0
14	5	166.0	72.000000	19.00000	175.000000	25.8	0

Задание 17.

Будем считать "здоровыми" тех, у кого нормальный вес и кровяное давление. Какой процент "здоровых" женщин больны диабетом?

In [88]:
```
# ( ͡° ͜ʖ ͡°)づ ━ ☆ﾟ.*･｡ﾟ
data[(data["BloodPressure"] > 80) & (data["BloodPressure"] < 89)
                    & (data["bodyType"] == "Normal weight") & (data["Class"] =
```

Out[88]:
```
0.6510416666666667
```