Лабораторная работа №2

Визуальный анализ данных

Подключение библиотек

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
%matplotlib inline
```

Загрузка данных

```
data_path = "/content/data2.csv"
data = pd.read_csv(data_path)
data.head(10)
# data.columns
```


```
{"summary":"{\n  \"name\": \"# data\",\n  \"rows\": 10,\n  \"fields\":
[\n    {\n      \"column\": \"State\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 9,\n
\"samples\": [\n          \"LA\",\n          \"OH\",\n
\"MA\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Account length\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 23,\n        \"min\": 75,\n
\"max\": 147,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          117,\n          107,\n          118\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Area code\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
40,\n        \"min\": 408,\n        \"max\": 510,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          415,\n
510,\n          419\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"International plan\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Yes\",\n          \"No\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Voice mail plan\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          \"No\",\n
\"Yes\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Number vmail messages\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 5,\n        \"min\": 10,\n
```

```
\"max\": 29,\n        \"num_unique_values\": 10,\n        \"samples\":
[\n          19,\n          14\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Total day minutes\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 49.34917425854257,\n
\"min\": 157.0,\n        \"max\": 299.4,\n
\"num_unique_values\": 10,\n        \"samples\": [\n          184.5,\n
161.6\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Total day calls\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 18,\n        \"min\": 71,\n
\"max\": 123,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          97,\n          123\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Total day charge\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
8.37146376421445,\n        \"min\": 26.69,\n        \"max\": 50.9,\n
\"num_unique_values\": 10,\n        \"samples\": [\n          31.37,\n
27.47\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Total eve minutes\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 96.16209059014194,\n        \"min\":
61.9,\n        \"max\": 351.6,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          351.6,\n          195.5\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Total eve calls\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
33,\n        \"min\": 80,\n        \"max\": 199,\n
\"num_unique_values\": 10,\n        \"samples\": [\n          80,\n
103\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Total eve charge\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 8.174559723108436,\n        \"min\":
5.26,\n        \"max\": 29.89,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          29.89,\n          16.62\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Total night minutes\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
45.229832829425504,\n        \"min\": 162.6,\n        \"max\": 326.4,\
n        \"num_unique_values\": 10,\n        \"samples\": [\n
215.8,\n          254.4\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Total night calls\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 12,\n        \"min\": 89,\n
\"max\": 121,\n        \"num_unique_values\": 9,\n        \"samples\":
[\n          90,\n          103\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Total night charge\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
```

```
2.035349000922337,\n        \"min\": 7.32,\n        \"max\": 14.69,\n
\"num_unique_values\": 10,\n        \"samples\": [\n          9.71,\n
11.45\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Total intl minutes\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2.521551550577981,\n        \"min\":
6.3,\n        \"max\": 13.7,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          8.7,\n          13.7\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Total intl calls\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 3,\n        \"max\": 7,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          5,\n
4\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Total intl charge\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.6799313690856356,\n        \"min\":
1.7,\n        \"max\": 3.7,\n        \"num_unique_values\": 10,\n
\"samples\": [\n          2.35,\n          3.7\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Customer service calls\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
1,\n        \"min\": 0,\n        \"max\": 3,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          0,\n
3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Churn\",\n      \"properties\": {\n        \"dtype\": \"boolean\",\n
\"num_unique_values\": 1,\n        \"samples\": [\n          true\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```


```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 64 entries, 0 to 63
Data columns (total 20 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   State                   64 non-null     object
 1   Account length          64 non-null     int64
 2   Area code               64 non-null     int64
 3   International plan      64 non-null     object
 4   Voice mail plan         64 non-null     object
 5   Number vmail messages   64 non-null     int64
 6   Total day minutes       64 non-null     float64
 7   Total day calls         64 non-null     int64
 8   Total day charge        64 non-null     float64
 9   Total eve minutes       64 non-null     float64
 10  Total eve calls         64 non-null     int64
 11  Total eve charge        64 non-null     float64
```

```
 12  Total night minutes     64 non-null     float64
 13  Total night calls       64 non-null     int64
 14  Total night charge      64 non-null     float64
 15  Total intl minutes      64 non-null     float64
 16  Total intl calls        64 non-null     int64
 17  Total intl charge       64 non-null     float64
 18  Customer service calls  64 non-null     int64
 19  Churn                   64 non-null     bool
dtypes: bool(1), float64(8), int64(8), object(3)
memory usage: 9.7+ KB
```

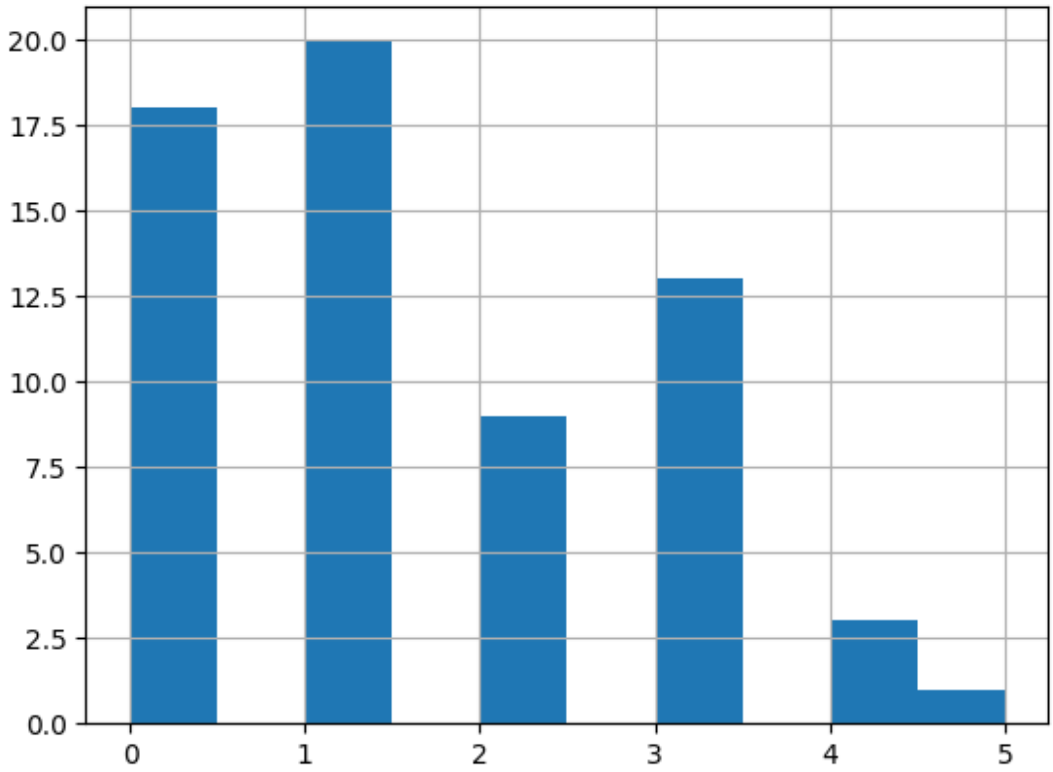
Одиночные признаки

Количественные признаки

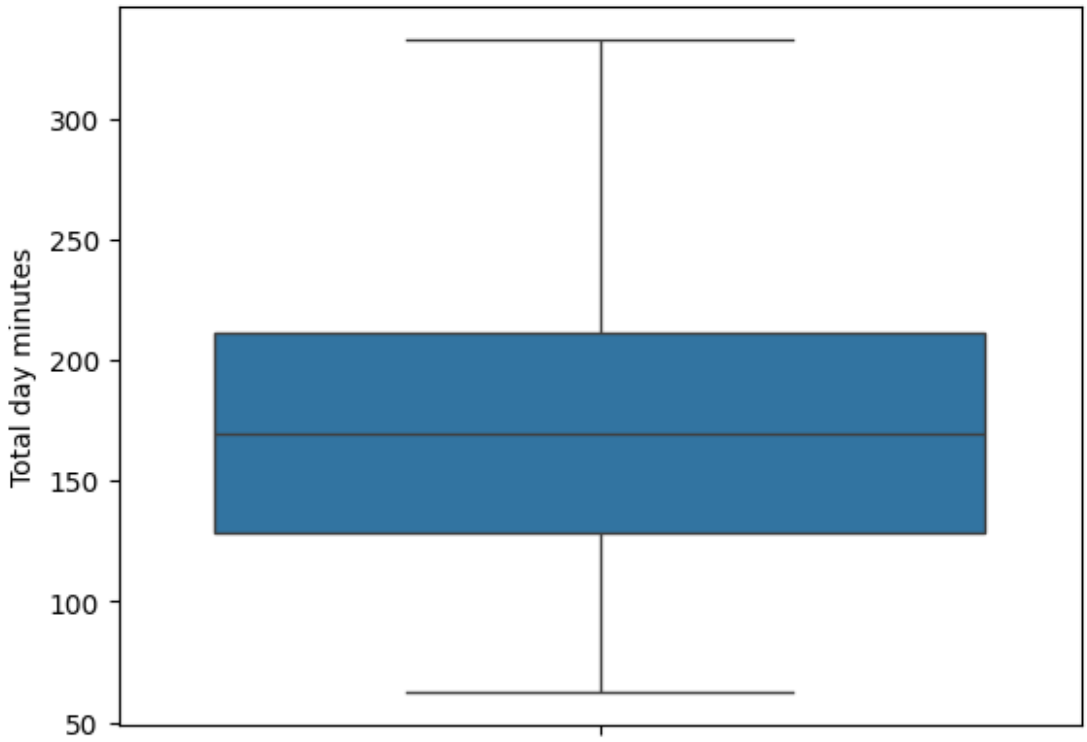
```
data.columns
```

```
Index(['State', 'Account length', 'Area code', 'International plan',
       'Voice mail plan', 'Number vmail messages', 'Total day
minutes',
       'Total day calls', 'Total day charge', 'Total eve minutes',
       'Total eve calls', 'Total eve charge', 'Total night minutes',
       'Total night calls', 'Total night charge', 'Total intl
minutes',
       'Total intl calls', 'Total intl charge', 'Customer service
calls',
       'Churn'],
      dtype='object')
```

```
# Применение pandas для визуализации данных
# Pandas работает как настройка над matplotlib
data['Customer service calls'].hist();
```

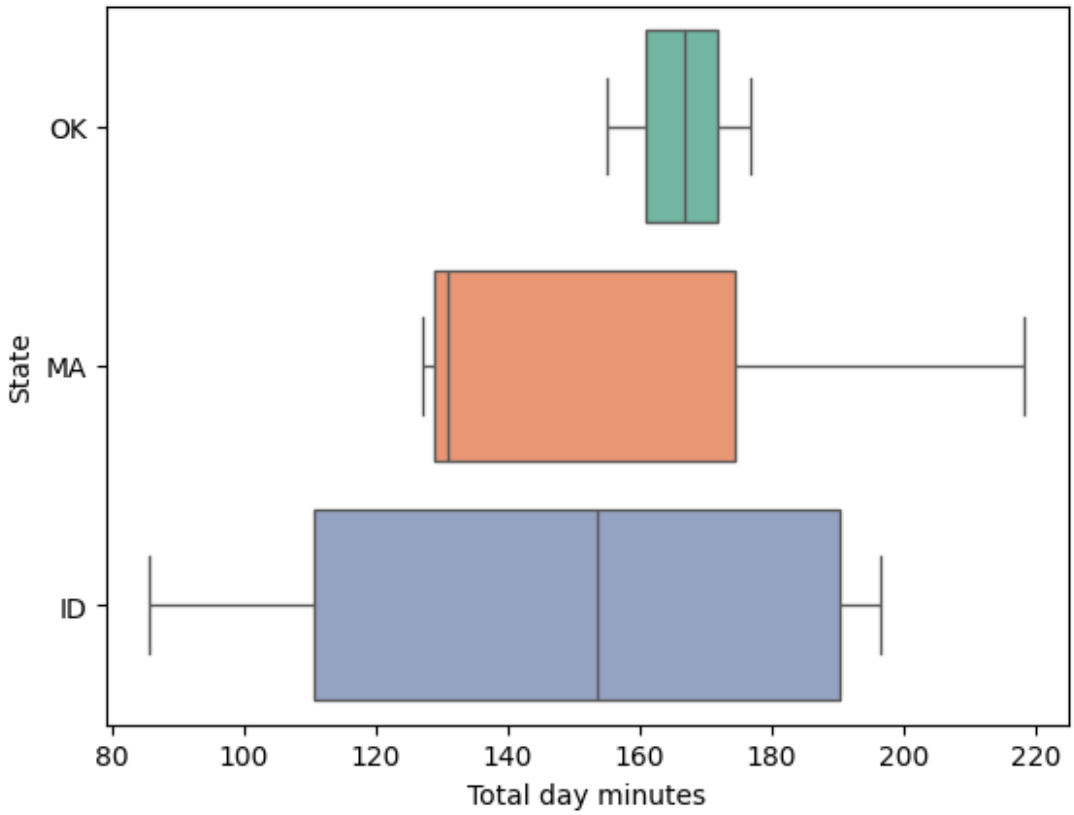



```
# использование Seaborn
# Построение диаграммы типа "ящик с усами"
# по диаграмме можно определить медиану, квартили,
# интерквартильный размах, выбросы
sns.boxplot(data['Total day minutes']);
```

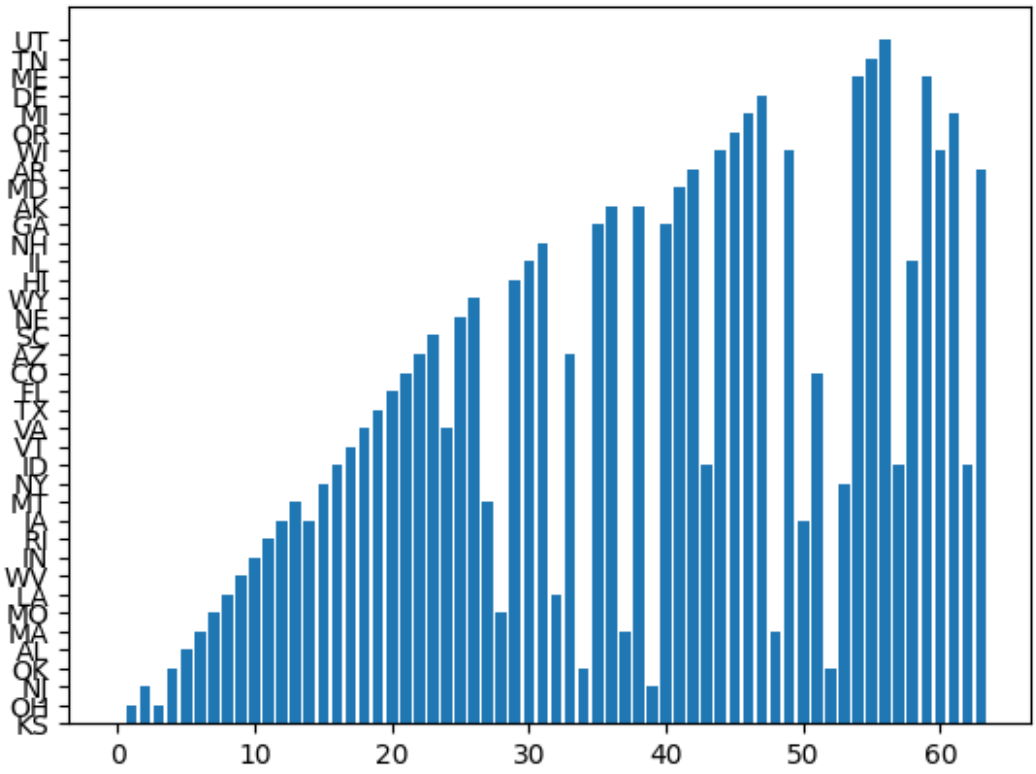



```
top_data = data[['State','Total day minutes']]
top_data = top_data.groupby('State').sum()
top_data = top_data.sort_values('Total day minutes',ascending=False)
top_data = top_data[:3].index.values
sns.boxplot(y='State',
            x='Total day minutes',
            data=data[data.State.isin(top_data)], palette='Set2');
```

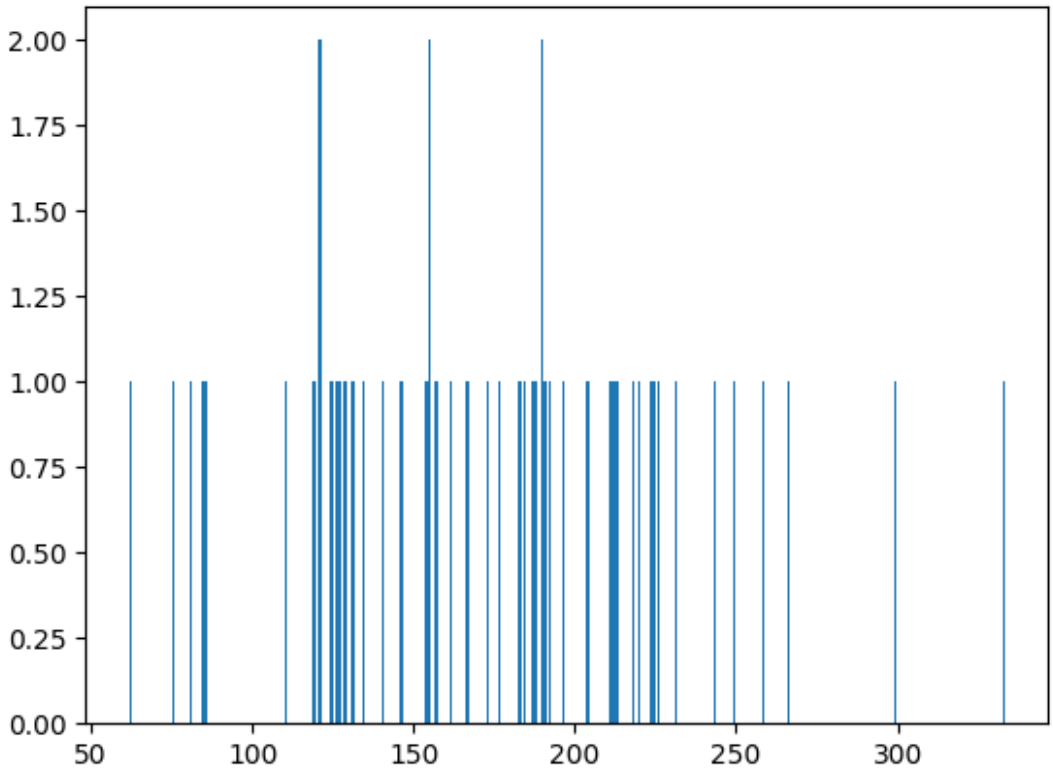
```
<ipython-input-71-add119f26021>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `y` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(y='State',
```




```
plt.bar(data.index, data['State'])
plt.show()
```



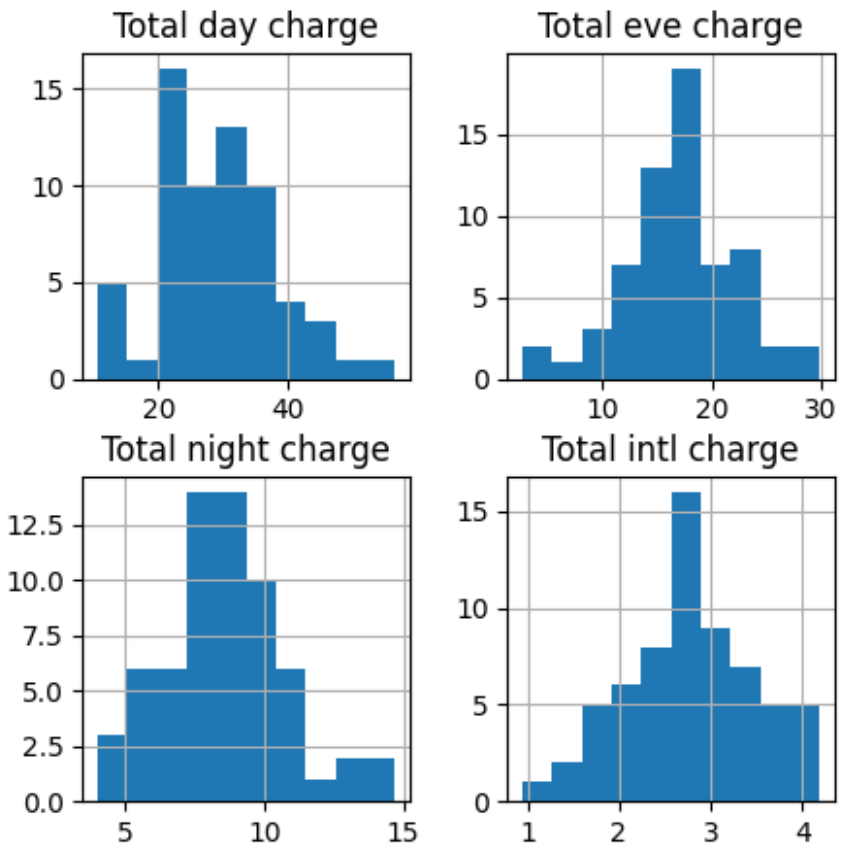

```
hist = data['Total day minutes'].value_counts()
plt.bar(hist.index, hist);
```




```
# jn,jh ghbpyfrjd
feats=[f for f in data.columns if 'charge' in f]
feats
```

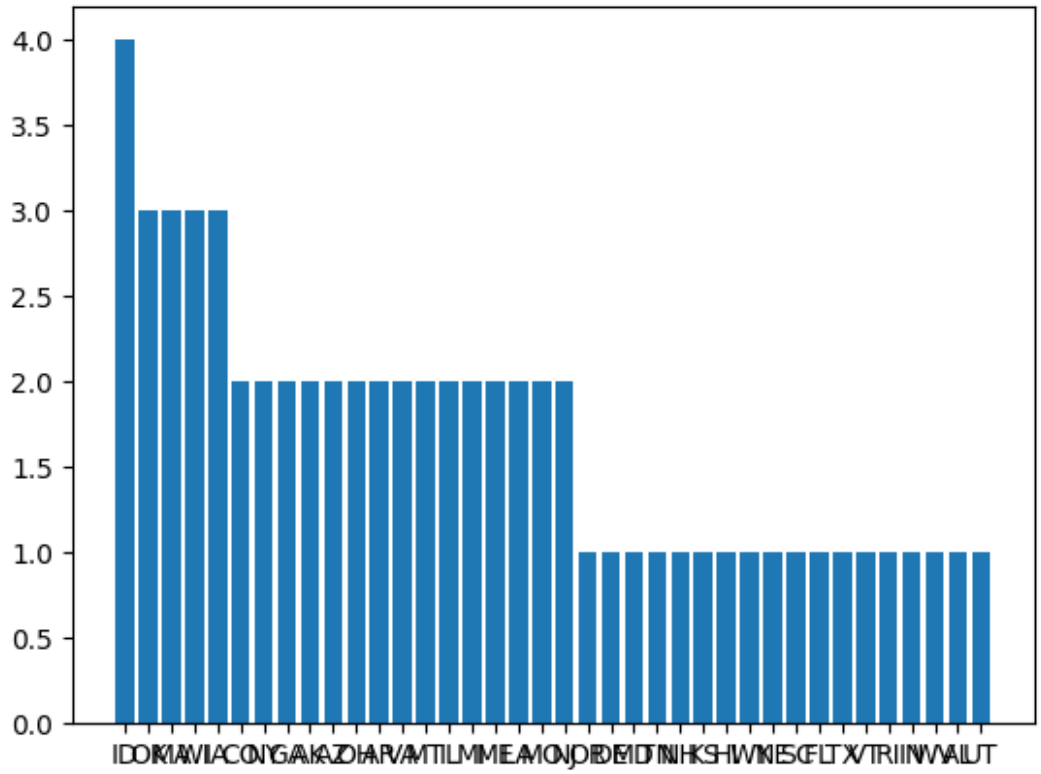
```
['Total day charge',
 'Total eve charge',
 'Total night charge',
 'Total intl charge']
```

```
# построение гистограммы для нескольких признаков
data[feats].hist(figsize=(5,5));
```



Категориальные признаки

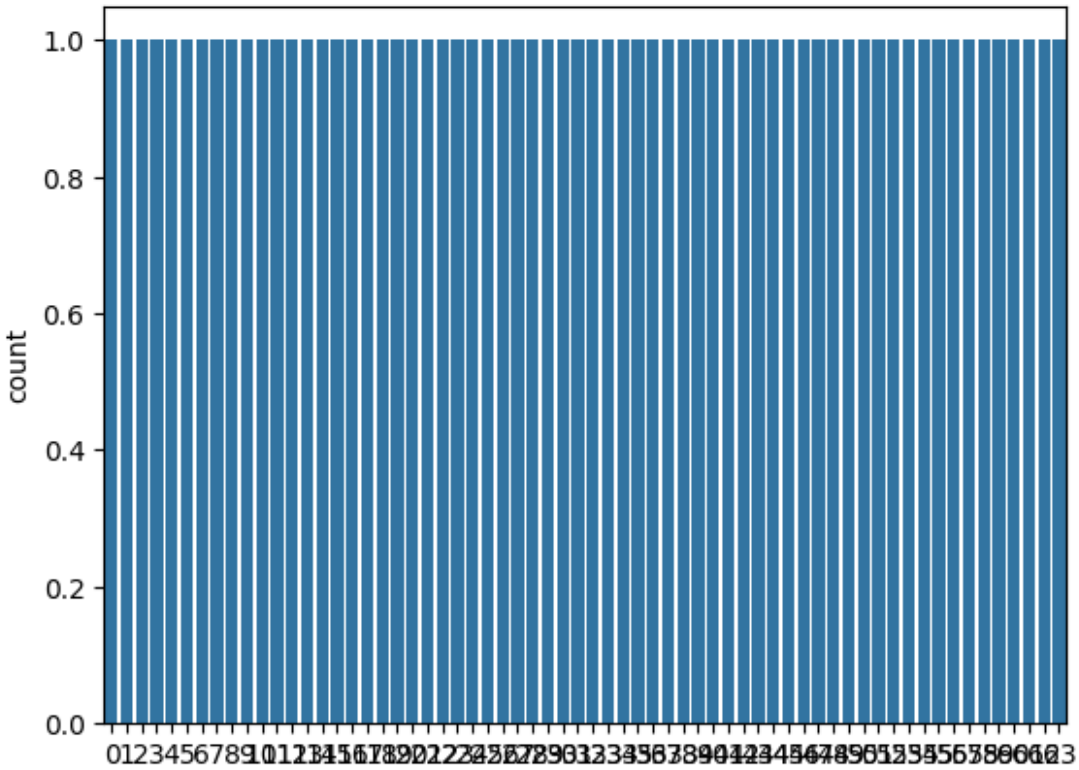
```
# определение первых n "популярных" штатов
# data['State'].value_counts().head(10)
hist = data['State'].value_counts()
plt.bar(hist.index, hist);
```



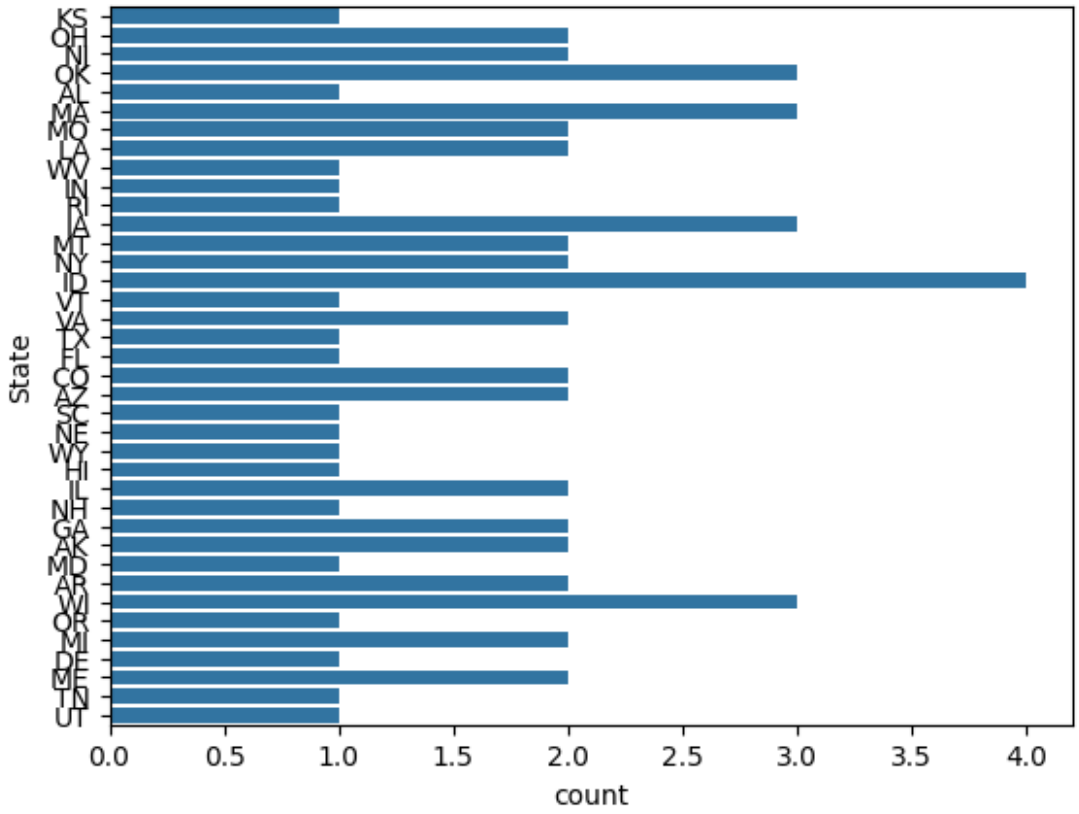

```
# фактически бинарный признак
data['Churn'].value_counts()
```

```
Churn
True     60
False     4
Name: count, dtype: int64
```

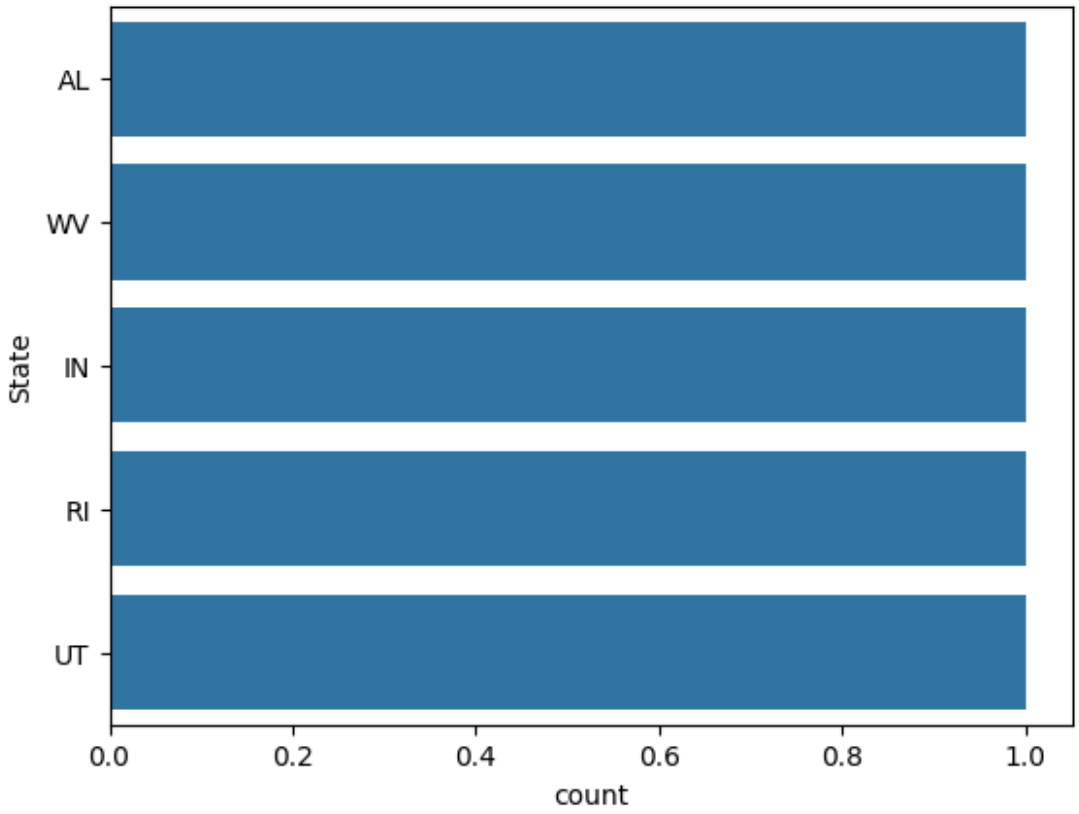
```
sns.countplot(data['Churn']);
```




```
# гистограмма для всех штатов
sns.countplot(data['State']);
```




```
# гистограмма "популярных" штатов
sns.countplot(data[data['State'].isin(data['State'].value_counts().tai
l(5).index)]['State']);
```



Взаимосвязанные признаки

Количественный - количественный

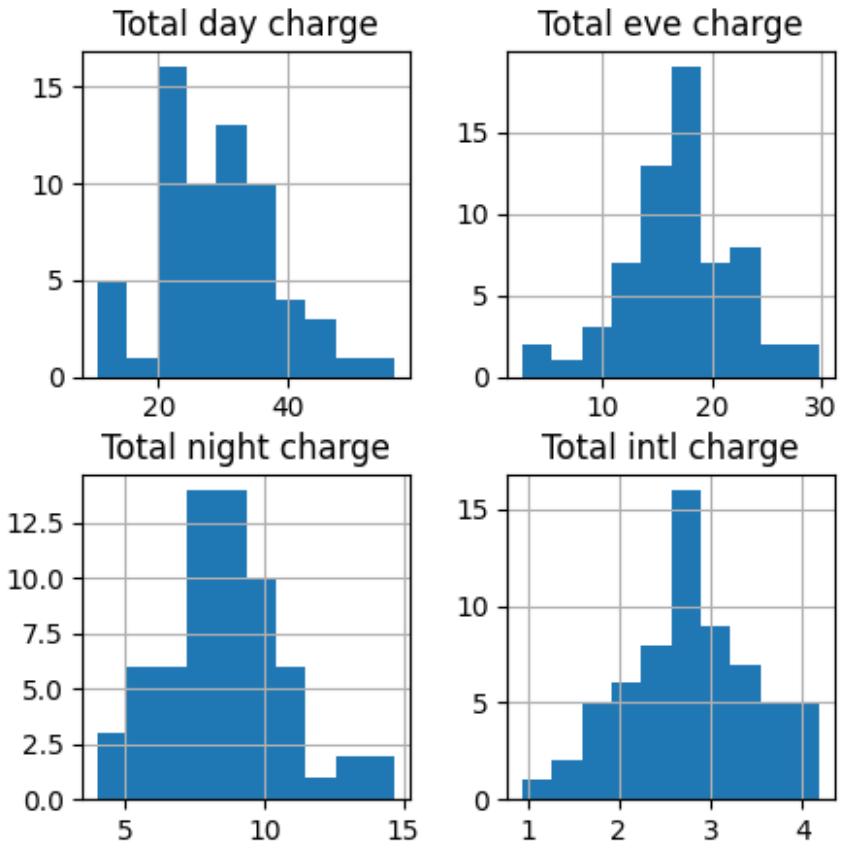
```
# список колонок
data.columns
```

```
Index(['State', 'Account length', 'Area code', 'International plan',
       'Voice mail plan', 'Number vmail messages', 'Total day
minutes',
       'Total day calls', 'Total day charge', 'Total eve minutes',
       'Total eve calls', 'Total eve charge', 'Total night minutes',
       'Total night calls', 'Total night charge', 'Total intl
minutes',
       'Total intl calls', 'Total intl charge', 'Customer service
calls',
       'Churn'],
      dtype='object')
```

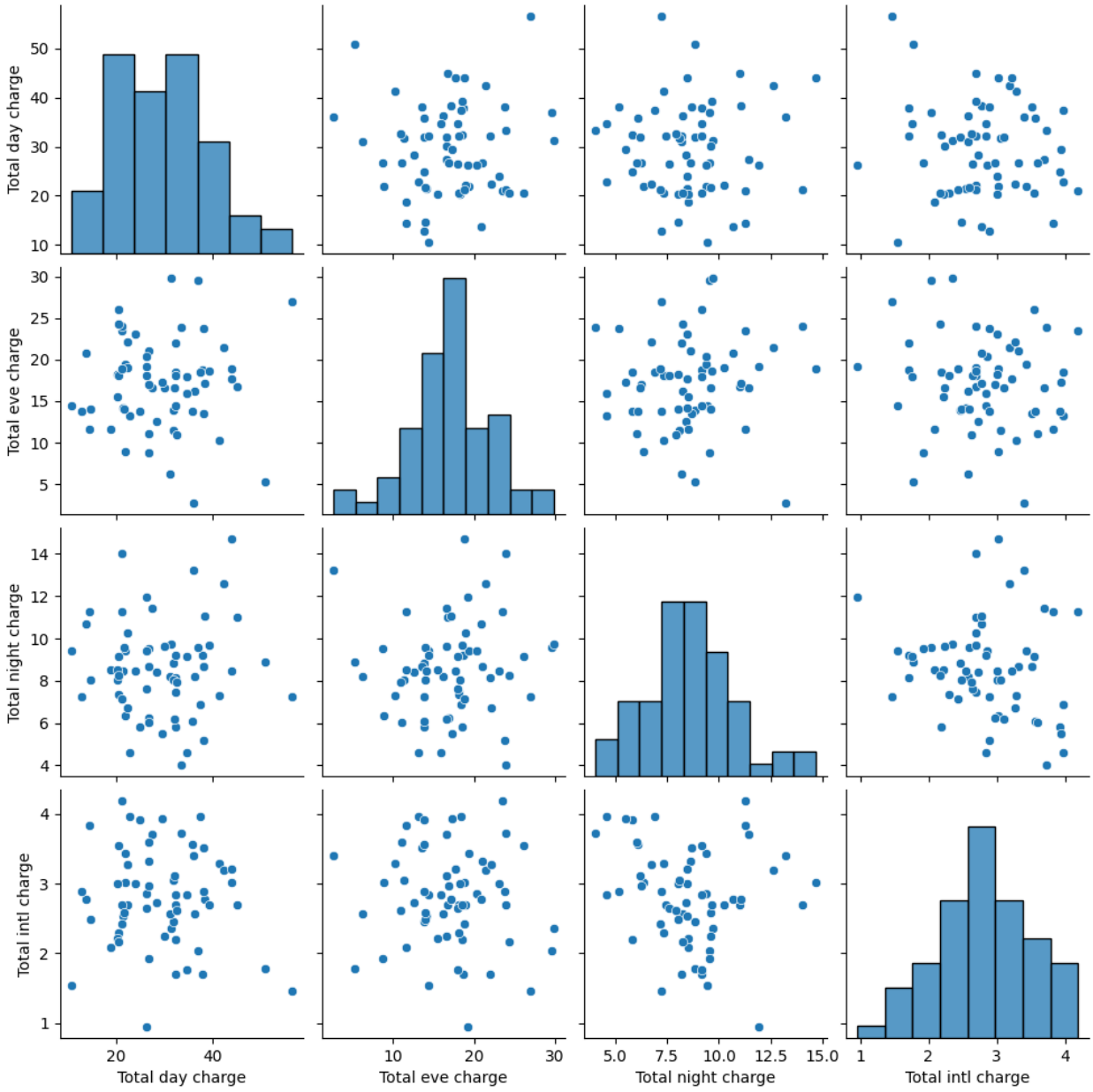
```
# Отбор числовых признаков, содержащих слово 'charge'
feats = [f for f in data.columns if 'charge' in f]
len(feats)
# feats=['Total day calls', 'Total day charge']
```

```
4
```

```
# строим отдельные гистограммы
# для нескольких признаков
data[feats].hist(figsize=(5,5));
```

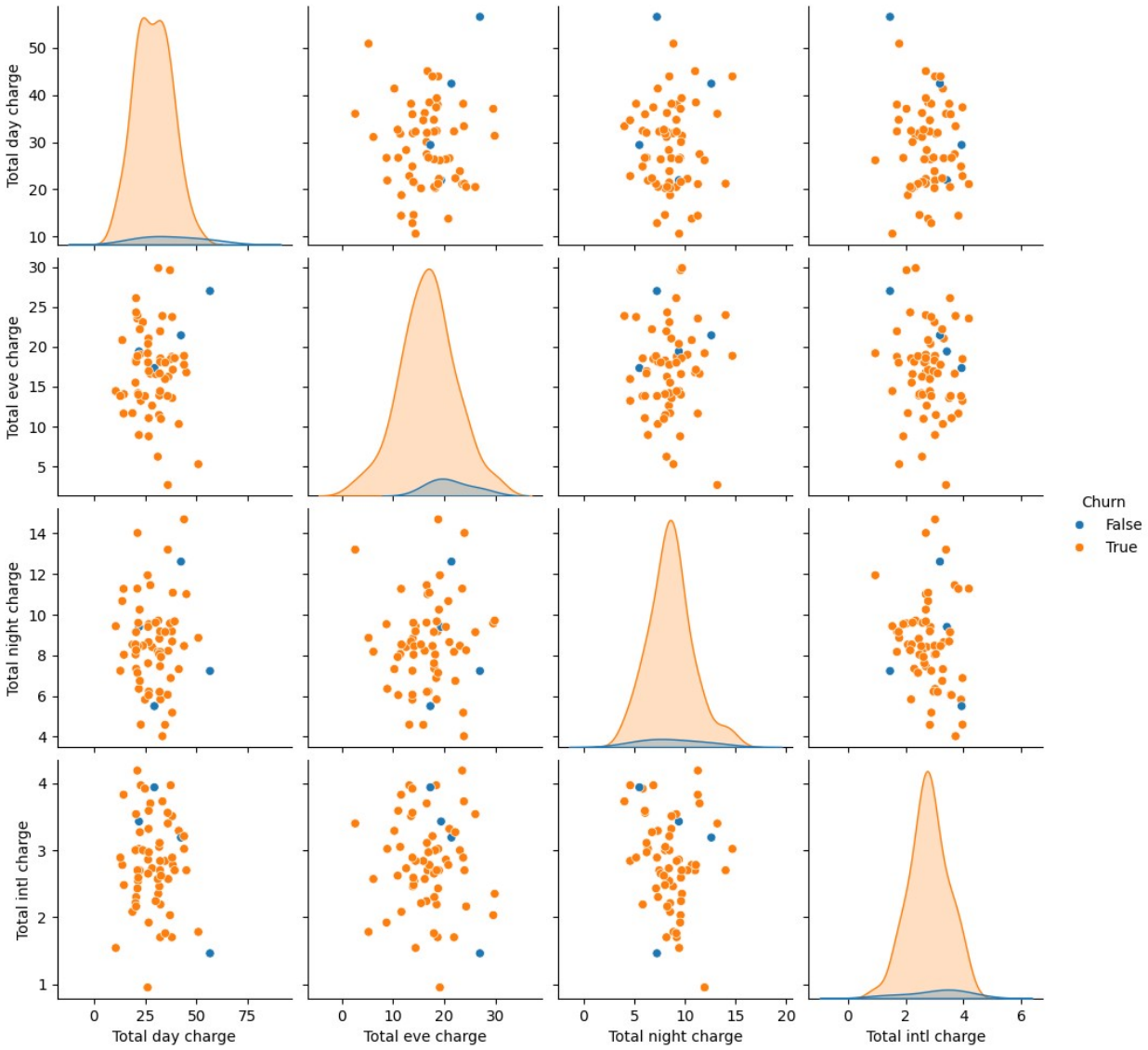



```
# Попарное распределение признаков
# Применение Seaborn
sns.pairplot(data[feats]);
```



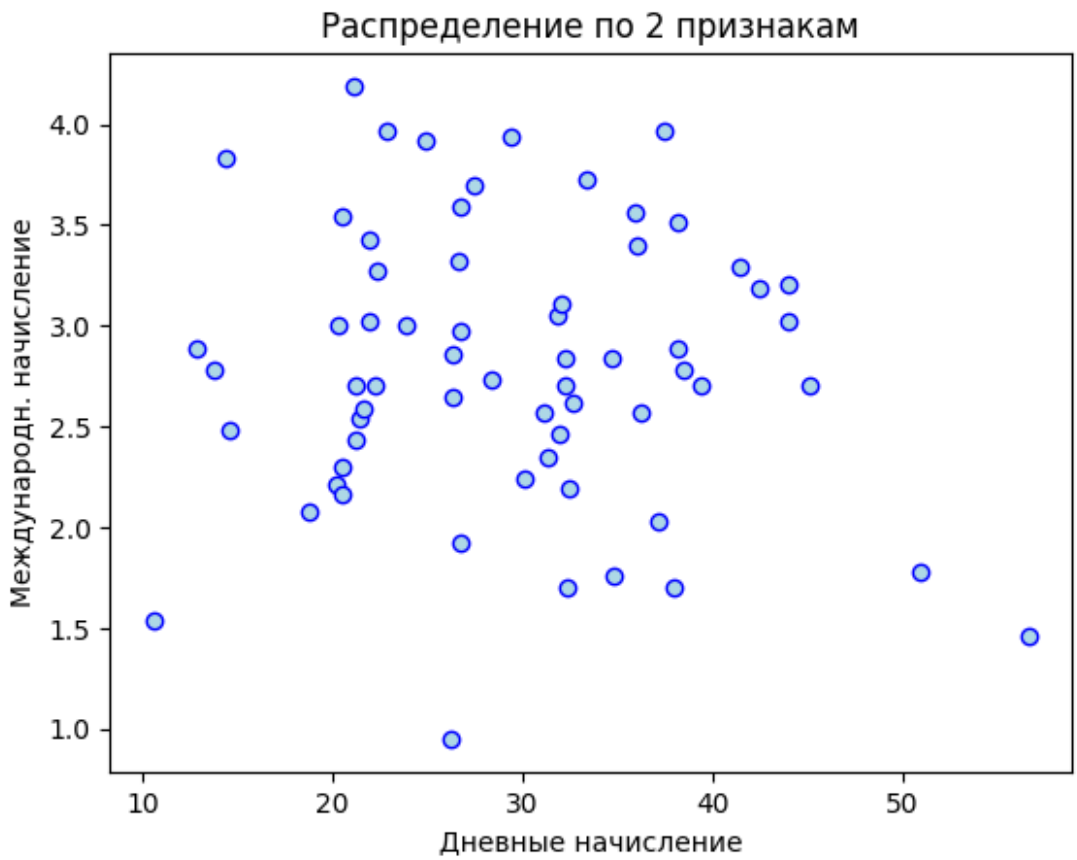
Можно строить более сложные попарные распределения признаков

```
sns.pairplot(data[feats + ['Churn']], hue='Churn');
```

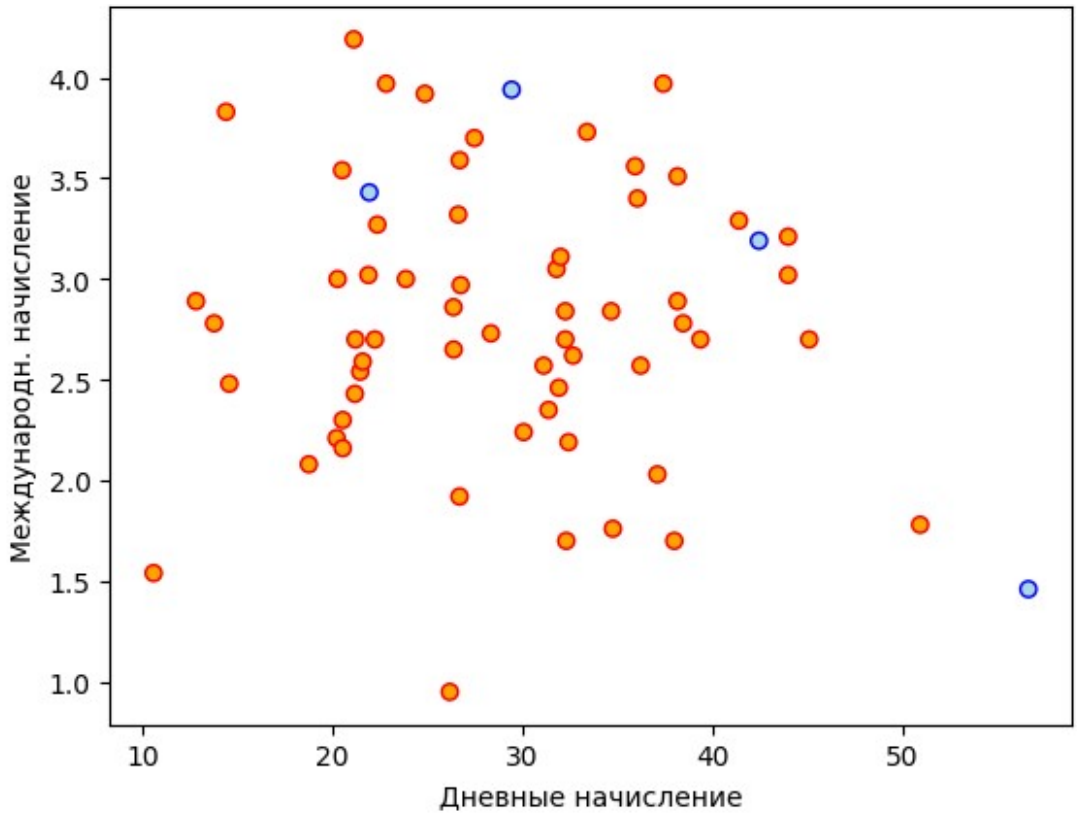


Использование matplotlib, подписей данных, заголовков Использование простейших пользовательских цветов

```
plt.scatter(data['Total day charge'],
            data['Total intl charge'],
            color='lightblue', edgecolors='blue')
plt.xlabel('Дневные начисление')
plt.ylabel('Международн. начисление')
plt.title('Распределение по 2 признакам');
```




```
# Раскрашивание данных
# Цвет в зависимости от ухода клиента
c = data['Churn'].map({False: 'lightblue', True: 'orange'})
edge_c = data['Churn'].map({False: 'blue', True: 'red'})
# Настройка графика
plt.scatter(data['Total day charge'], data['Total intl charge'],
            color=c, edgecolors=edge_c
           )
plt.xlabel('Дневные начисление')
plt.ylabel('Международн. начисление');
```




```
# Раскраска лояльных и ушедших клиентов,
# добавление легенды

# Ушедшие клиенты
data_churn = data[data['Churn']]
# Оставшиеся клиенты
data_loyal = data[~data['Churn']]

plt.scatter(data_churn['Total day charge'],
            data_churn['Total intl charge'],
            color='orange',
            edgecolors='red',
            label='Ушли'
           )
plt.scatter(data_loyal['Total day charge'],
            data_loyal['Total intl charge'],
            color='lightblue',
            edgecolors='blue',
            label='Остались'
           )
plt.xlabel('Дневные начисление')
plt.ylabel('Международн. начисление')
plt.title('Распределение клиентов')
plt.legend();
```

Корреляция признаков

Из карты heatmap видно, что некоторые признаки коррелируют: например сильная корреляция в парах (total day charge, total day minutes), (total night charge, total night minutes). Из таких пар можно удалить один признак

```
# Удаление коррелирующих признаков
data_uncorr = data.drop(feats, axis=1)
data_uncorr.columns
```

```
Index(['State', 'Account length', 'Area code', 'International plan',
       'Voice mail plan', 'Number vmail messages', 'Total day
minutes',
       'Total day calls', 'Total eve minutes', 'Total eve calls',
       'Total night minutes', 'Total night calls', 'Total intl
minutes',
       'Total intl calls', 'Customer service calls', 'Churn'],
      dtype='object')
```

Перестраиваем heatmap без коррелирующих признаков