Шаг 1. Откройте таблицу и изучите общую информацию о данных

In [84]:

```
import math

import pandas as pd

data = pd.read_csv("./data.csv")

data.head(10)
```

Out[84]:

	children	days_employed	dob_years	education	education_id	family_status	family_status_id	gender	income_type	debt	
0	1	-8437.673028	42	высшее	0	женат / замужем	0	F	сотрудник	0	2
1	1	-4024.803754	36	среднее	1	женат / замужем	0	F	сотрудник	0	1
2	0	-5623.422610	33	Среднее	1	женат / замужем	0	M	сотрудник	0	1
3	3	-4124.747207	32	среднее	1	женат / замужем	0	M	сотрудник	0	2
4	0	340266.072047	53	среднее	1	гражданский брак	1	F	пенсионер	0	1
5	0	-926.185831	27	высшее	0	гражданский брак	1	M	компаньон	0	2
6	0	-2879.202052	43	высшее	0	женат / замужем	0	F	компаньон	0	2
7	0	-152.779569	50	СРЕДНЕЕ	1	женат / замужем	0	M	сотрудник	0	1
8	2	-6929.865299	35	ВЫСШЕЕ	0	гражданский брак	1	F	сотрудник	0	
9	0	-2188.756445	41	среднее	1	женат / замужем	0	M	сотрудник	0	1

Шаг 2. Предобработка данных

1. Проверим на пустые значения

In [85]:

```
data.isna().sum()
```

Out[85]:

```
children               0
days_employed       2174
dob_years              0
education              0
education_id           0
family_status          0
family_status_id       0
gender                 0
income_type            0
debt                   0
total_income        2174
purpose                0
```

```
dtype: int64
```

Вывод: Мы выяснили что единственный столбец с пропущенными значениями это days_employed - 2174 пропущенных значения. Отлично дальше мы сможем это исправить

Как мы видим у нас в столбце **children** есть значения **< 0.**

Вывод: Причиной подобного события вероятно есть потеря знака у числа. Её можно исправить: применим функцию модуль ко всем значениям столбца

In [86]:

```
data["children"] = data["children"].apply(lambda x: abs(x))

data.head(10)
```

Out[86]:

	children	days_employed	dob_years	education	education_id	family_status	family_status_id	gender	income_type	debt	
0	1	-8437.673028	42	высшее	0	женат / замужем	0	F	сотрудник	0	2
1	1	-4024.803754	36	среднее	1	женат / замужем	0	F	сотрудник	0	1
2	0	-5623.422610	33	Среднее	1	женат / замужем	0	M	сотрудник	0	1
3	3	-4124.747207	32	среднее	1	женат / замужем	0	M	сотрудник	0	2
4	0	340266.072047	53	среднее	1	гражданский брак	1	F	пенсионер	0	1
5	0	-926.185831	27	высшее	0	гражданский брак	1	M	компаньон	0	2
6	0	-2879.202052	43	высшее	0	женат / замужем	0	F	компаньон	0	2
7	0	-152.779569	50	СРЕДНЕЕ	1	женат / замужем	0	M	сотрудник	0	1
8	2	-6929.865299	35	ВЫСШЕЕ	0	гражданский брак	1	F	сотрудник	0	
9	0	-2188.756445	41	среднее	1	женат / замужем	0	M	сотрудник	0	1

Как мы видим у нас в столбце **days_employed** есть значения **< 0.**

Вывод: Причиной подобного события вероятно есть потеря знака у числа. Её можно исправить: применим функцию модуль ко всем значениям столбца

In [87]:

```
data["days_employed"] = data["days_employed"].apply(lambda x: abs(x))

data.head(10)
```

Out[87]:

	children	days_employed	dob_years	education	education_id	family_status	family_status_id	gender	income_type	debt	
0	1	8437.673028	42	высшее	0	женат / замужем	0	F	сотрудник	0	2
						женат /					

	children	days_employed	dob_years	education	education_id	family_status	family_status_id	gender	income_type	debt	
1	1	4024.803754	36	среднее	1	[illegible]	0	F	сотрудник	0	1
2	0	5623.422610	33	Среднее	1	женат / замужем	0	M	сотрудник	0	1
3	3	4124.747207	32	среднее	1	женат / замужем	0	M	сотрудник	0	2
4	0	340266.072047	53	среднее	1	гражданский брак	1	F	пенсионер	0	1
5	0	926.185831	27	высшее	0	гражданский брак	1	M	компаньон	0	2
6	0	2879.202052	43	высшее	0	женат / замужем	0	F	компаньон	0	2
7	0	152.779569	50	СРЕДНЕЕ	1	женат / замужем	0	M	сотрудник	0	1
8	2	6929.865299	35	ВЫСШЕЕ	0	гражданский брак	1	F	сотрудник	0	
9	0	2188.756445	41	среднее	1	женат / замужем	0	M	сотрудник	0	1

Теперь все **2174** пропущенных значения в столбце **days_employed** мы заменим на медианное значение

In [88]:

```
data["days_employed"] = data["days_employed"].fillna(data["days_employed"].median())
data.isnull().sum()
```

Out[88]:

```
children               0
days_employed          0
dob_years              0
education              0
education_id           0
family_status          0
family_status_id       0
gender                 0
income_type            0
debt                   0
total_income        2174
purpose                0
dtype: int64
```

Вывод: Столбце days_employed пришел в норму

Заменим вещественный тип данных на целочисленный в столбце **total_income.** И предварительно изменив все пустые значения на медианное значение столбца

In [89]:

```
data["total_income"] = data["total_income"].fillna(data["total_income"].median())
data["total_income"] = data["total_income"].astype(int)

data["total_income"].head(10)
```

Out[89]:

```
0    253875
1    112080
2    145885
3    267628
4    158616
5    255763
```

```
6    240525
7    135823
8     95856
9    144425
Name: total_income, dtype: int32
```

Вывод: в столбце total_income есть только целочисленные значения

2. Обработка дубликатов

1) Удалим явные дубликаты в таблице **2)** Найдем неявные дубликаты и почистим их

1. Удалим явные дубликаты в таблице

In [90]:

```
data = data.drop_duplicates().reset_index(drop=True)
data
```

Out[90]:

	children	days_employed	dob_years	education	education_id	family_status	family_status_id	gender	income_type	del
0	1	8437.673028	42	высшее	0	женат / замужем	0	F	сотрудник	
1	1	4024.803754	36	среднее	1	женат / замужем	0	F	сотрудник	
2	0	5623.422610	33	Среднее	1	женат / замужем	0	M	сотрудник	
3	3	4124.747207	32	среднее	1	женат / замужем	0	M	сотрудник	
4	0	340266.072047	53	среднее	1	гражданский брак	1	F	пенсионер	
...	...	...	...	...	...	...	...	...	...	...
21466	1	4529.316663	43	среднее	1	гражданский брак	1	F	компаньон	
21467	0	343937.404131	67	среднее	1	женат / замужем	0	F	пенсионер	
21468	1	2113.346888	38	среднее	1	гражданский брак	1	M	сотрудник	
21469	3	3112.481705	38	среднее	1	женат / замужем	0	M	сотрудник	
21470	2	1984.507589	40	среднее	1	женат / замужем	0	F	сотрудник	

21471 rows × 12 columns

Вывод: все явные дубликаты были удалены метод drop_duplicates и последующее очистка старых индексов

2. Поиск и удаление неявных дубликатов

In [91]:

```
data
```

Out[91]:

	children	days_employed	dob_years	education	education_id	family_status	family_status_id	gender	income_type	del
0	1	8437.673028	42	высшее	0	женат / замужем	0	F	сотрудник	
1	1	4024.803754	36	среднее	1	женат / замужем	0	F	сотрудник	
2	0	5623.422610	33	Среднее	1	женат / замужем	0	M	сотрудник	
3	3	4124.747207	32	среднее	1	женат / замужем	0	M	сотрудник	
4	0	340266.072047	53	среднее	1	гражданский брак	1	F	пенсионер	
...	...	...	...	...	...	...	...	...	...	
21466	1	4529.316663	43	среднее	1	гражданский брак	1	F	компаньон	
21467	0	343937.404131	67	среднее	1	женат / замужем	0	F	пенсионер	
21468	1	2113.346888	38	среднее	1	гражданский брак	1	M	сотрудник	
21469	3	3112.481705	38	среднее	1	женат / замужем	0	M	сотрудник	
21470	2	1984.507589	40	среднее	1	женат / замужем	0	F	сотрудник	

21471 rows × 12 columns

Как мы видим есть 2 потенциальных столбца с неявными дубликатами это: education, family_status

In [92]:

```
data["education"] = data["education"].apply(lambda x: x.lower())
data["family_status"] = data["family_status"].apply(lambda x: x.lower())

data = data.drop_duplicates().reset_index(drop=True)
data
```

Out[92]:

	children	days_employed	dob_years	education	education_id	family_status	family_status_id	gender	income_type	del
0	1	8437.673028	42	высшее	0	женат / замужем	0	F	сотрудник	
1	1	4024.803754	36	среднее	1	женат / замужем	0	F	сотрудник	
2	0	5623.422610	33	среднее	1	женат / замужем	0	M	сотрудник	
3	3	4124.747207	32	среднее	1	женат / замужем	0	M	сотрудник	
4	0	340266.072047	53	среднее	1	гражданский брак	1	F	пенсионер	
...	...	...	...	...	...	...	...	...	...	
21449	1	4529.316663	43	среднее	1	гражданский брак	1	F	компаньон	
21450	0	343937.404131	67	среднее	1	женат / замужем	0	F	пенсионер	
21451	1	2113.346888	38	среднее	1	гражданский брак	1	M	сотрудник	
						женат /				

	children	days_employed	dob_years	education	education_id	family_status	family_status_id	gender	income_type	del
21452	3	3112.481705	38	среднее	1	женат / замужем	0	M	сотрудник	
21453	**2**	**1984.507589**	**40**	среднее	**1**	женат / замужем	**0**	**F**	сотрудник	

21454 rows × 12 columns

Вывод: все неявные дубликаты были превращены в явные путем преобразования строк и дальшего удаление как в пункте 1

Создадим два новых датафрейма, в которых:

1) каждому уникальному значению из **education** соответствует уникальное значение **education_id; 2)** каждому уникальному значению из **family_status** соответствует уникальное значение **family_status_id .**

In [93]:

```
education_unique = data["education"].unique()
education_id_unique = data["education_id"].unique()

education_data = pd.DataFrame({
    'education': education_unique,
    'education_id': education_id_unique
})

education_data
```

Out[93]:

	education	education_id
0	высшее	**0**
1	среднее	**1**
2	неоконченное высшее	**2**
3	начальное	**3**
4	ученая степень	**4**

In [94]:

```
family_status_unique = data["family_status"].unique()
family_status_id_unique = data["family_status_id"].unique()

family_data = pd.DataFrame({
    'family_status': family_status_unique,
    'family_status_id': family_status_id_unique
})

family_data
```

Out[94]:

	family_status	family_status_id
0	женат / замужем	**0**
1	гражданский брак	**1**
2	вдовец / вдова	**2**
3	в разводе	**3**
4	не женат / не замужем	**4**

Удалим из исходного датафрейма столбцы **education** и **family_status**

In [95]:

```
data = data.drop(columns=["education", "family_status"], axis=1)
```

In [96]:

```
data
```

Out[96]:

	children	days_employed	dob_years	education_id	family_status_id	gender	income_type	debt	total_income	
0	1	8437.673028	42	0	0	F	сотрудник	0	253875	покупк
1	1	4024.803754	36	1	0	F	сотрудник	0	112080	приоб авто
2	0	5623.422610	33	1	0	M	сотрудник	0	145885	покупк
3	3	4124.747207	32	1	0	M	сотрудник	0	267628	дополни обра
4	0	340266.072047	53	1	1	F	пенсионер	0	158616	
...	...	...	...	...	...	...	...	...	...	
21449	1	4529.316663	43	1	1	F	компаньон	0	224791	опе
21450	0	343937.404131	67	1	0	F	пенсионер	0	155999	с автом
21451	1	2113.346888	38	1	1	M	сотрудник	1	89672	недви
21452	3	3112.481705	38	1	0	M	сотрудник	1	244093	на авто
21453	2	1984.507589	40	1	0	F	сотрудник	0	82047	на авто

21454 rows × 10 columns

Вывод: столбцы **education** и **family_status** были удалены

На основании диапазонов, указанных ниже, создадим столбец **total_income_category** с категориями:

1) 0–30000 — 'E'; 2) 30001–50000 — 'D'; 3) 50001–200000 — 'C'; 4) 200001–1000000 — 'B'; 5) 1000001 и выше — **'A'.**

In [97]:

```
def get_category(value: int) -> str:
    if value <= 3*10**4:
        return "E"
    elif 3*10**4 < value <= 5*10**4:
        return "D"
    elif 5*10**4 < value <= 2*10**5:
        return "C"
    elif 2*10**5 < value <= 10**6:
        return "B"
    elif 10**6 < value:
        return "A"


data["total_income_category"] = data["total_income"].apply(get_category)
data
```

Out[97]:

	children	days_employed	dob_years	education_id	family_status_id	gender	income_type	debt	total_income	
0	1	8437.673028	42	0	0	F	сотрудник	0	253875	покупк
1	1	4024.803754	36	1	0	F	сотрудник	0	112080	приоб авто
2	0	5623.422610	33	1	0	M	сотрудник	0	145885	покупк
3	3	4124.747207	32	1	0	M	сотрудник	0	267628	дополни обра
4	0	340266.072047	53	1	1	F	пенсионер	0	158616	
...	...	...	...	...	...	...	...	...	...	
21449	1	4529.316663	43	1	1	F	компаньон	0	224791	опе
21450	0	343937.404131	67	1	0	F	пенсионер	0	155999	с автом
21451	1	2113.346888	38	1	1	M	сотрудник	1	89672	недви
21452	3	3112.481705	38	1	0	M	сотрудник	1	244093	на авто
21453	2	1984.507589	40	1	0	F	сотрудник	0	82047	на авто

21454 rows × 11 columns

Вывод: Мы разделили клиентов на классы от E до A

Создадим функцию, которая на основании данных из столбца **purpose** сформирует новый столбец **purpose_category,** в который войдут следующие категории:

1. операции с автомобилем
2. операции с недвижимостью
3. проведение свадьбы
4. получение образования

In [98]:

```
def get_category_purpose(string: str) -> str:
    string = string.lower()

    if "недвижимость" in string or "жиль" in string:
        return "операции с недвижимостью"
    elif "автомобил" in string:
        return "операции с автомобилем"
    elif "свадьб" in string:
        return "проведение свадьбы"
    elif "образовани" in string:
        return "получение образования"
    return ""



data["purpose_category"] = data["purpose"].apply(get_category_purpose)
data
```

Out[98]:

	children	days_employed	dob_years	education_id	family_status_id	gender	income_type	debt	total_income	
0	1	8437.673028	42	0	0	F	сотрудник	0	253875	покупк

	children	days_employed	dob_years	education_id	family_status_id	gender	income_type	debt	total_income	
1	1	4024.803754	36	1	0	F	сотрудник	0	112080	приоб авто
2	0	5623.422610	33	1	0	M	сотрудник	0	145885	покупк
3	3	4124.747207	32	1	0	M	сотрудник	0	267628	дополни обра
4	0	340266.072047	53	1	1	F	пенсионер	0	158616	
...	...	...	...	...	...	...	...	...	...	
21449	1	4529.316663	43	1	1	F	компаньон	0	224791	опе
21450	0	343937.404131	67	1	0	F	пенсионер	0	155999	с автом
21451	1	2113.346888	38	1	1	M	сотрудник	1	89672	недви
21452	3	3112.481705	38	1	0	M	сотрудник	1	244093	на авто
21453	2	1984.507589	40	1	0	F	сотрудник	0	82047	на авто

21454 rows × 12 columns

Вывод: Мы разбили на 4 категории запросы

Шаг 3. Ответьте на вопросы

1) Есть ли зависимость между количеством детей и возвратом кредита в срок**? 2)** Есть ли зависимость между семейным положением и возвратом кредита в срок**? 3)** Есть ли зависимость между уровнем дохода и возвратом кредита в срок**? 4)** Как разные цели кредита влияют на его возврат в срок**?**

1.Есть ли зависимость между количеством детей и возвратом кредита в срок?

Разделим людей на 2 группы:

1) С задолжностью **2)** Без задолжностей

In [98]:

```
```

In [99]:

```
debt_people = data.groupby("children")["debt"].value_counts(normalize=True)
debt_people
```

Out[99]:

```
children  debt
0         0       0.924562
          1       0.075438
1         0       0.908342
          1       0.091658
2         0       0.905458
          1       0.094542
3         0       0.918182
          1       0.081818
4         0       0.902439
          1       0.097561
5         0       1.000000
20        0       0.894737
```

```
          1       0.105263
Name: proportion, dtype: float64
```

Вывод: Как мы видим наличие детей и их количество не сильно влияет статистически на погашение кредита в срок

2. Есть ли зависимость между семейным положением и возвратом кредита в срок?

In [100]:

```
debt_of_family = data.groupby("family_status_id")["debt"].value_counts(normalize=True)
debt_of_family
```

Out[100]:

```
family_status_id  debt
0                 0       0.924548
                  1       0.075452
1                 0       0.906529
                  1       0.093471
2                 0       0.934307
                  1       0.065693
3                 0       0.928870
                  1       0.071130
4                 0       0.902491
                  1       0.097509
Name: proportion, dtype: float64
```

Вывод: Как мы видим зависимость между семейным статусом и погашением кредита во-время нет.

3. Есть ли зависимость между уровнем дохода и возвратом кредита в срок?

In [101]:

```
debt_total_income = data.groupby("total_income_category")["debt"].value_counts(normalize
=True)
debt_total_income
```

Out[101]:

```
total_income_category  debt
A                      0       0.920000
                       1       0.080000
B                      0       0.929379
                       1       0.070621
C                      0       0.915085
                       1       0.084915
D                      0       0.940000
                       1       0.060000
E                      0       0.909091
                       1       0.090909
Name: proportion, dtype: float64
```

Вывод: Как мы видим зависимость между уровнем дохода и погашением кредита во-время нет.

4. Как разные цели кредита влияют на его возврат в срок?

In [102]:

```
debt_purpose_category = data.groupby("purpose_category")["debt"].value_counts(normalize=
True)
debt_purpose_category
```

Out[102]:

```
purpose_category          debt
```

```
                            0       0.926978
                            1       0.073022
операции с автомобилем      0       0.906410
                            1       0.093590
операции с недвижимостью    0       0.928034
                            1       0.071966
получение образования       0       0.907800
                            1       0.092200
проведение свадьбы          0       0.919966
                            1       0.080034
Name: proportion, dtype: float64
```

Вывод: Как мы видим зависимость между целью заема и погашением кредита во-время нет