Исследование надежности заемщиков

Во второй части проекта вы выполните шаги 3 и 4. Их вручную проверит ревьюер. Чтобы вам не пришлось писать код заново для шагов 1 и 2, мы добавили авторские решения в ячейки с кодом.

Откройте таблицу и изучите общую информацию о данных

Задание 1. Импортируйте библиотеку pandas. Считайте данные из csv-файла в датафрейм и сохраните в переменную `data`. Путь к файлу:

`/datasets/data.csv`

In [3]:
```
import pandas as pd

try:
    data = pd.read_csv('/data.csv')
except:
    data = pd.read_csv('https://code.s3.yandex.net/datasets/data.csv')
```

Задание 2. Выведите первые 20 строчек датафрейма `data` на экран.

In [4]:
```
data.head(20)
```

Out[4]:

	children	days_employed	dob_years	education	education_id	family_status	family_status_id	gender	income_type	debt	total_
0	1	-8437.673028	42	высшее	0	женат / замужем	0	F	сотрудник	0	253875
1	1	-4024.803754	36	среднее	1	женат / замужем	0	F	сотрудник	0	112080
2	0	-5623.422610	33	Среднее	1	женат / замужем	0	M	сотрудник	0	145885
3	3	-4124.747207	32	среднее	1	женат / замужем	0	M	сотрудник	0	267628
4	0	340266.072047	53	среднее	1	гражданский брак	1	F	пенсионер	0	158616
5	0	-926.185831	27	высшее	0	гражданский брак	1	M	компаньон	0	255763
6	0	-2879.202052	43	высшее	0	женат / замужем	0	F	компаньон	0	240525
7	0	-152.779569	50	СРЕДНЕЕ	1	женат / замужем	0	M	сотрудник	0	135823
8	2	-6929.865299	35	ВЫСШЕЕ	0	гражданский брак	1	F	сотрудник	0	95856
9	0	-2188.756445	41	среднее	1	женат / замужем	0	M	сотрудник	0	144425
10	2	-4171.483647	36	высшее	0	женат / замужем	0	M	компаньон	0	113943
11	0	-792.701887	40	среднее	1	женат / замужем	0	F	сотрудник	0	77069
12	0	NaN	65	среднее	1	гражданский брак	1	M	пенсионер	0	
13	0	-1846.641941	54	неоконченное высшее	2	женат / замужем	0	F	сотрудник	0	130458
14	0	-1844.956182	56	высшее	0	гражданский брак	1	F	компаньон	1	165127
15	1	-972.364419	26	среднее	1	женат / замужем	0	F	сотрудник	0	116820
16	0	-1719.934226	35	среднее	1	женат / замужем	0	F	сотрудник	0	289202
17	0	-2369.999720	33	высшее	0	гражданский брак	1	M	сотрудник	0	90410
18	0	400281.136913	53	среднее	1	вдовец / вдова	2	F	пенсионер	0	56823
19	0	-10038.818549	48	СРЕДНЕЕ	1	в разводе	3	F	сотрудник	0	242831

Задание 3. Выведите основную информацию о датафрейме с помощью метода `info()`.

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21525 entries, 0 to 21524
Data columns (total 12 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   children          21525 non-null  int64  
 1   days_employed     19351 non-null  float64
 2   dob_years         21525 non-null  int64  
 3   education         21525 non-null  object 
 4   education_id      21525 non-null  int64  
 5   family_status     21525 non-null  object 
 6   family_status_id  21525 non-null  int64  
 7   gender            21525 non-null  object 
 8   income_type       21525 non-null  object 
 9   debt              21525 non-null  int64  
 10  total_income      19351 non-null  float64
 11  purpose           21525 non-null  object 
dtypes: float64(2), int64(5), object(5)
memory usage: 2.0+ MB
```

Предобработка данных

Удаление пропусков

Задание 4. Выведите количество пропущенных значений для каждого столбца. Используйте комбинацию двух методов.

In [6]:
```
data.isna().sum()
```

Out[6]:
```
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
```

Задание 5. В двух столбцах есть пропущенные значения. Один из них — `days_employed`. Пропуски в этом столбце вы обработаете на следующем этапе. Другой столбец с пропущенными значениями — `total_income` — хранит данные о доходах. На сумму дохода сильнее всего влияет тип занятости, поэтому заполнить пропуски в этом столбце нужно медианным значением по каждому типу из столбца `income_type`. Например, у человека с типом занятости `сотрудник` пропуск в столбце `total_income` должен быть заполнен медианным доходом среди всех записей с тем же типом.

In [7]:
```
for t in data['income_type'].unique():
    data.loc[(data['income_type'] == t) & (data['total_income'].isna()), 'total_income'] = \
    data.loc[(data['income_type'] == t), 'total_income'].median()
```

Обработка аномальных значений

Задание 6. В данных могут встречаться артефакты (аномалии) — значения, которые не отражают действительность и появились по какой-то ошибке. таким артефактом будет отрицательное количество дней трудового стажа в столбце `days_employed`. Для реальных данных это нормально. Обработайте значения в этом столбце: замените все отрицательные значения положительными с помощью метода `abs()`.

In [8]:
```
data['days_employed'] = data['days_employed'].abs()
```

Задание 7. Для каждого типа занятости выведите медианное значение трудового стажа `days_employed` в днях.

In [9]:
```
data.groupby('income_type')['days_employed'].agg('median')
```

Out[9]:
```
income_type
безработный        366413.652744
в декрете            3296.759962
госслужащий          2689.368353
компаньон            1547.382223
пенсионер          365213.306266
предприниматель       520.848083
сотрудник            1574.202821
студент               578.751554
Name: days_employed, dtype: float64
```

У двух типов (безработные и пенсионеры) получатся аномально большие значения. Исправить такие значения сложно, поэтому оставьте их как есть. Тем более этот столбец не понадобится вам для исследования.

Задание 8. Выведите перечень уникальных значений столбца `children`.

In [10]:
```
data['children'].unique()
```

Out[10]:
```
array([ 1,  0,  3,  2, -1,  4, 20,  5], dtype=int64)
```

Задание 9. В столбце `children` есть два аномальных значения. Удалите строки, в которых встречаются такие аномальные значения из датафрейма `data`.

In [11]:
```
data = data[(data['children'] != -1) & (data['children'] != 20)]
```

Задание 10. Ещё раз выведите перечень уникальных значений столбца `children`, чтобы убедиться, что артефакты удалены.

In [12]:
```
data['children'].unique()
```

Out[12]:
```
array([1, 0, 3, 2, 4, 5], dtype=int64)
```

Удаление пропусков (продолжение)

Задание 11. Заполните пропуски в столбце `days_employed` медианными значениями по каждого типа занятости `income_type`.

In [13]:
```
for t in data['income_type'].unique():
    data.loc[(data['income_type'] == t) & (data['days_employed'].isna()), 'days_employed'] = \
    data.loc[(data['income_type'] == t), 'days_employed'].median()
```

Задание 12. Убедитесь, что все пропуски заполнены. Проверьте себя и ещё раз выведите количество пропущенных значений для каждого столбца с помощью двух методов.

In [14]:
```
data.isna().sum()
```

Out[14]:
```
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        0
purpose             0
dtype: int64
```

Изменение типов данных

Задание 13. Замените вещественный тип данных в столбце `total_income` на целочисленный с помощью метода `astype()`.

In [15]:
```
data['total_income'] = data['total_income'].astype(int)
```

Обработка дубликатов

Задание 14. Выведите на экран количество строк-дубликатов в данных. Если такие строки присутствуют, удалите их.

In [16]:
```
data.duplicated().sum()
```

Out[16]:
```
54
```

In [17]:
```
data = data.drop_duplicates()
```

Задание 15. Обработайте неявные дубликаты в столбце `education`. В этом столбце есть одни и те же значения, но записанные по-разному: с использованием заглавных и строчных букв. Приведите их к нижнему регистру. Проверьте остальные столбцы.

In [18]:
```
data['education'] = data['education'].str.lower()
```

Категоризация данных

Задание 16. На основании диапазонов, указанных ниже, создайте в датафрейме `data` столбец `total_income_category` с категориями:

- 0–30000 — `'E'`;
- 30001–50000 — `'D'`;
- 50001–200000 — `'C'`;
- 200001–1000000 — `'B'`;
- 1000001 и выше — `'A'`.

Например, кредитополучателю с доходом 25000 нужно назначить категорию `'E'`, а клиенту, получающему 235000, — `'B'`. Используйте собственную функцию с именем `categorize_income()` и метод `apply()`.

In [19]:
```
def categorize_income(income):
    try:
        if 0 <= income <= 30000:
            return 'E'
        elif 30001 <= income <= 50000:
            return 'D'
        elif 50001 <= income <= 200000:
            return 'C'
        elif 200001 <= income <= 1000000:
            return 'B'
        elif income >= 1000001:
            return 'A'
    except:
        pass
```

In [20]:
```
data['total_income_category'] = data['total_income'].apply(categorize_income)
```

Задание 17. Выведите на экран перечень уникальных целей взятия кредита из столбца `purpose`.

In [21]:
```
data['purpose'].unique()
```

Out[21]:
```
array(['покупка жилья', 'приобретение автомобиля',
       'дополнительное образование', 'сыграть свадьбу',
       'операции с жильем', 'образование', 'на проведение свадьбы',
       'покупка жилья для семьи', 'покупка недвижимости',
       'покупка коммерческой недвижимости', 'покупка жилой недвижимости',
       'строительство собственной недвижимости', 'недвижимость',
       'строительство недвижимости', 'на покупку подержанного автомобиля',
       'на покупку своего автомобиля',
       'операции с коммерческой недвижимостью',
       'строительство жилой недвижимости', 'жилье',
       'операции со своей недвижимостью', 'автомобили',
       'заняться образованием', 'сделка с подержанным автомобилем',
       'получение образования', 'автомобиль', 'свадьба',
       'получение дополнительного образования', 'покупка своего жилья',
       'операции с недвижимостью', 'получение высшего образования',
       'свой автомобиль', 'сделка с автомобилем',
       'профильное образование', 'высшее образование',
       'покупка жилья для сдачи', 'на покупку автомобиля', 'ремонт жилью',
       'заняться высшим образованием'], dtype=object)
```

Задание 18. Создайте функцию, которая на основании данных из столбца `purpose` сформирует новый столбец `purpose_category`, в который войдут следующие категории:

- `'операции с автомобилем'`,
- `'операции с недвижимостью'`,
- `'проведение свадьбы'`,
- `'получение образования'`.

Например, если в столбце `purpose` находится подстрока `'на покупку автомобиля'`, то в столбце `purpose_category` должна появиться строка `'операции с автомобилем'`.

Используйте собственную функцию с именем `categorize_purpose()` и метод `apply()`. Изучите данные в столбце `purpose` и определите, какие подстроки помогут вам правильно определить категорию.

In [22]:
```
def categorize_purpose(row):
    try:
        if 'автом' in row:
            return 'операции с автомобилем'
        elif 'жил' in row or 'недвиж' in row:
            return 'операции с недвижимостью'
        elif 'свад' in row:
            return 'проведение свадьбы'
        elif 'образов' in row:
            return 'получение образования'
    except:
        return 'нет категории'
```

In [23]:
```
data['purpose_category'] = data['purpose'].apply(categorize_purpose)
```

Исследуйте данные и ответьте на вопросы

Для исследования используем собственную функцию `category_factor()`, возвращающую отчет с данными о влиянии категории заемщика на вероятность возврата кредита в срок.

Для вывода графиков импортирум библиотеку `'seaborn'`

In [24]:
```
import seaborn as sb
import matplotlib.pyplot as plt
```

In [25]:
```
# Создание функции для формирования отчета с данными о влиянии категории заемщика на возврат кредита в срок, где
#   'cat' - категория заемщика, определяемая соответствующим столбцом в исходном датасете,
#   'column_sort' - номер столбца отчета, по которому сортируется отчет
#   'direct - направление сортировки, по которому сортируется отчет (1-по возрастанию, 0- по убыванию)

def category_factor(cat,column_sort,direct):

     # Количество заемщиков и надежных заемщиков, процент возврата
    loaner = len(data)
    well_loaner =  data.loc[data['debt']==0, 'debt'].count()
    well_loaner_ratio=  round(well_loaner/loaner*100,2)


    # Определение подкатегорий
    subcat = data[cat].unique()

    # формируем датасет
    table = []

    for s in range(len(subcat)):      # циклом проходим по каждой подкатегории и результат добавляем в общий датасет

        # кол-во людей в подкатегории
        subcat_peoples= data[cat].loc[data[cat]==subcat[s]].count()

        #  % людей в подкатегории от всех заемщиков
        subcat_peoples_ratio= round(subcat_peoples/loaner*100,2)

        # кол-во людей в подкатегории, выплачивающих в срок
        subcat_well_loaner=  data[cat].loc[(data[cat]==subcat[s]) & (data['debt']==0)].count()

        # % людейв в подкатегории, выплачивающих в срок (в подкатегории)
        subcat_well_loaner_ratio_sc=  round(subcat_well_loaner/subcat_peoples*100,2)

        # % людей в подкатегории, выплачивающих в срок (от всех заемщиков)
        subcat_well_loaner_ratio_all=  round(subcat_well_loaner/loaner*100,2)

        #добавление данных в датасет
        table.append([cat, subcat[s], subcat_peoples, subcat_peoples_ratio, subcat_well_loaner, subcat_well_loaner_r


    # заголовки столбцов отчета
    heading=['категория  ','подкатегория  ','кол-во в подкатегории  ','% от всех  ','выплата в срок, чел  ','выпла

    #формирование  отчета
    rep = pd.DataFrame(data=table, columns=heading)

    #сортировка отчета по подкатегориям и направлению
    rep = rep.sort_values(by=heading[column_sort],ascending=direct).reset_index(drop=True)

    #предотчетная информация
    print (f'Количество всех заемщиков {loaner},    количество надежных заемщиков  {well_loaner}, процент возврата в
```

```
    if  direct == 0 :
        print(f'Столбец сортировки отчета:    {heading[column_sort]},   направление сортировки- по убыванию')
    elif  direct == 1 :
        print(f'Столбец сортировки отчета:   {heading[column_sort]},   направление сортировки- по возрастанию')

    #выводим график

    axis_x = list(rep[heading[column_sort]])
    axis_y = list(rep[heading[1]])

    sb.barplot(x=axis_x, y=axis_y)


    return rep
```

Задание 19. Есть ли зависимость между количеством детей и возвратом кредита в срок?

Для корректного отображения зависимостей на графиках меняем тип данных в столбце `'children'` с целочисленного на вещественный

In [26]:
```
data['children'] =  data['children'].astype(str)

#print(type(data.loc[1,'children']))  # проверка корректности  замены типа данных
```

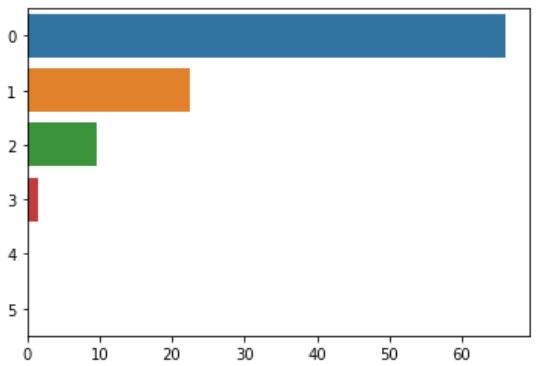
1. Общий обзор влияния количества детей на возврат кредита в срок

In [27]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)
report=category_factor('children',3,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    % от всех   ,   направление сортировки- по убыванию
```

Out[27]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	children	0	14107	66.08	13044	92.46	61.10
1	children	1	4809	22.53	4365	90.77	20.45
2	children	2	2052	9.61	1858	90.55	8.70
3	children	3	330	1.55	303	91.82	1.42
4	children	4	41	0.19	37	90.24	0.17
5	children	5	9	0.04	9	100.00	0.04



Предварительный вывод 1 : количество заемщиков обратно пропорционально количеству детей и для 4-5 детей составляет менее 1% от всех заемщиков. Охотнее всего берут кредиты заемщики без детей или с 1-2 детьми, в сумме их количество составляет 98,22 % от всех взявших кредит.

1. Сравнение по категории.

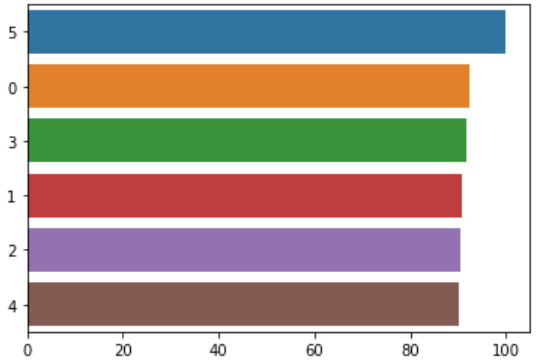
In [28]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)
```

```
report=category_factor('children',5,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % в подкатегории   ,   направление сортировки- по убыванию
```

Out[28]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	children	5	9	0.04	9	100.00	0.04
1	children	0	14107	66.08	13044	92.46	61.10
2	children	3	330	1.55	303	91.82	1.42
3	children	1	4809	22.53	4365	90.77	20.45
4	children	2	2052	9.61	1858	90.55	8.70
5	children	4	41	0.19	37	90.24	0.17



Предварительный вывод 2. Идеальные заемщики- имеющие 5 детей, они все до единого выполняют оплаты во время. В остальных подгруппах процент возврата больше 90%, с небольшим отрывом вперед выходят заемщики без детей и с 3 детьми.

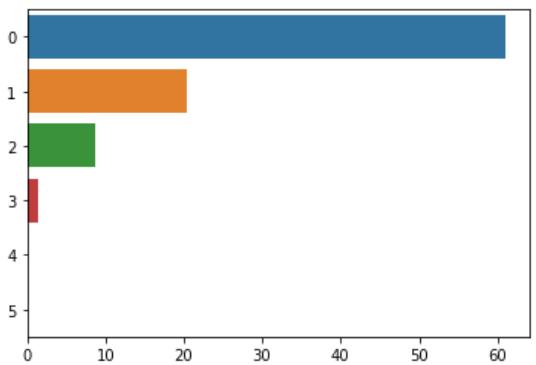
1. Сравнение с учетом количества.

In [29]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('children',6,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % от всех  ,   направление сортировки- по убыванию
```

Out[29]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	children	0	14107	66.08	13044	92.46	61.10
1	children	1	4809	22.53	4365	90.77	20.45
2	children	2	2052	9.61	1858	90.55	8.70
3	children	3	330	1.55	303	91.82	1.42
4	children	4	41	0.19	37	90.24	0.17
5	children	5	9	0.04	9	100.00	0.04



Предварительный вывод 3. Сказывается обратная зависимость количества детей и заемщиков. Заемщики с 5 детьми хотя и самые аккуратные, но в общей массе они почти ни на что не влияют. Также и худшие по категории заемщики с 4 детьми не оказывают существенного влияния на общие показатели по возвратам из-за небольшого количества таких заемщиков.

Добросовестные заемщики без детей или с 1-2 детьми составляют 90,25 % (из 91,89%) выполняющих оплаты вовремя от всех клиентов кредитной организации.

Вывод:

Среди 5-детных заемщиков наблюдается 100% возврат кредита вовремя. Среди остальных заемщиков с количеством детей от 0 до 4 возвращают кредит больше 90%, несколько лучше это делают заемщики без детей и с 3 детьми.

Учитывая количество заемщиков в общем количестве заемщиков, наибольшее влияние на % возвратов оказывают заемщики с 0-2 детьми. При этом 5-детные заемщики хотя и в 100% случаев выполняют оплаты вовремя, но не оказывают существенного влияния на процент выплат среди всех заемщиков.

Задание 20. Есть ли зависимость между семейным положением и возвратом кредита в срок?

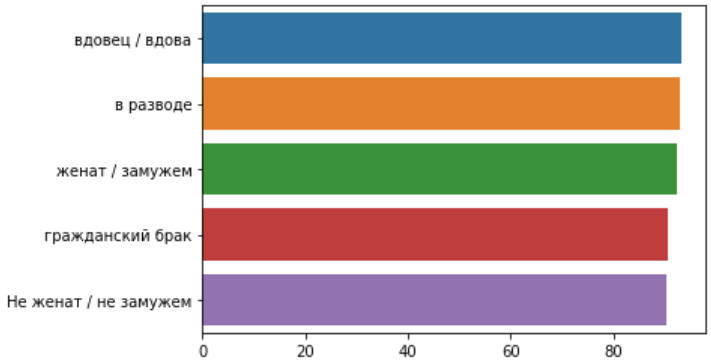
Сравнение по категории.

In [30]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('family_status',5,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % в подкатегории   ,   направление сортировки- по убыванию
```

Out[30]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	family_status	вдовец / вдова	951	4.45	888	93.38	4.16
1	family_status	в разводе	1189	5.57	1105	92.94	5.18
2	family_status	женат / замужем	12266	57.46	11339	92.44	53.12
3	family_status	гражданский брак	4146	19.42	3761	90.71	17.62
4	family_status	Не женат / не замужем	2796	13.10	2523	90.24	11.82



In [49]:
```
# все тоже самое, только через pivot_table
# Создание функции для формирования отчета с данными о влиянии категории заемщика на возврат кредита в срок, где
#   'cat' - категория заемщика, определяемая соответствующим столбцом в исходном датасете,
#   'column_sort' - номер столбца отчета, по которому сортируется отчет
#   'direct - направление сортировки, по которому сортируется отчет (1-по возрастанию, 0- по убыванию)

def pivot_category_factor(cat,column_sort,direct):



    # Количество заемщиков и надежных заемщиков, процент возврата кредитов
    loaner = len(data)
    well_loaner =  data.loc[data['debt']==0, 'debt'].count()
    well_loaner_ratio=  round(well_loaner/loaner*100,2)

    # заголовки столбцов отчета
    heading=['кол-во в подкатегории  ','неплательщики  ', 'надежные заемщики, чел', '% от всех   ','выплата в срок,
```

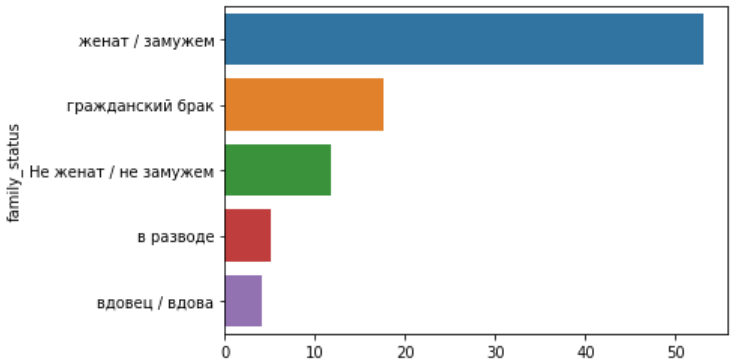
```
    # сводная таблица (pivot_table) по выбранной категории в cat  (число всех в подкатегории и ненадежные заемщики)
    piv_table = data.pivot_table(index=[cat], values='debt', aggfunc= ['count','sum'])



     # переименование столбцов отчета
    piv_table.columns=['кол-во в подкатегории  ','неплательщики  ']

    # надежные заемщики   в подкатегории
    piv_table['subcat_well_loaner']= piv_table ['кол-во в подкатегории  '] - piv_table ['неплательщики  ']

    #  % людей в подкатегории от всех заемщиков
    piv_table['subcat_peoples_ratio']=round(piv_table['кол-во в подкатегории  ']/loaner*100,2)

    # % людейв в подкатегории, выплачивающих в срок (в подкатегории)
    piv_table['subcat_well_loaner_ratio_sc']=round(piv_table['subcat_well_loaner']/piv_table['кол-во в подкатегории

    # % людей в подкатегории, выплачивающих в срок (от всех заемщиков)
    piv_table['']=round(piv_table['subcat_well_loaner']/loaner*100,2)


    # переименование столбцов отчета
    piv_table.columns=heading

    #сортировка отчета по подкатегориям и направлению
    piv_table =piv_table.sort_values(by=heading[column_sort],ascending=direct)

    #предотчетная информация
    print (f'Количество всех заемщиков {loaner},    количество надежных заемщиков  {well_loaner}, процент возврата в

    if  direct == 0 :
        print(f'Столбец сортировки отчета:    {heading[column_sort]},   направление сортировки- по убыванию')
    elif  direct == 1 :
        print(f'Столбец сортировки отчета:   {heading[column_sort]},   направление сортировки- по возрастанию')

    #выводим график

    axis_x = list(piv_table[heading[column_sort]])
    axis_y = piv_table.index

    sb.barplot(x=axis_x, y=axis_y)


    return piv_table
```

In [50]:
```
report_pv=pivot_category_factor('family_status',5,0)
report_pv
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % от всех  ,   направление сортировки- по убыванию
```

Out[50]:

family_status	кол-во в подкатегории	неплательщики	надежные заемщики, чел	% от всех	выплата в срок, % в подкатегории	выплата в срок, % от всех
женат / замужем	12266	927	11339	57.46	92.44	53.12
гражданский брак	4146	385	3761	19.42	90.71	17.62
Не женат / не замужем	2796	273	2523	13.10	90.24	11.82
в разводе	1189	84	1105	5.57	92.94	5.18
вдовец / вдова	951	63	888	4.45	93.38	4.16



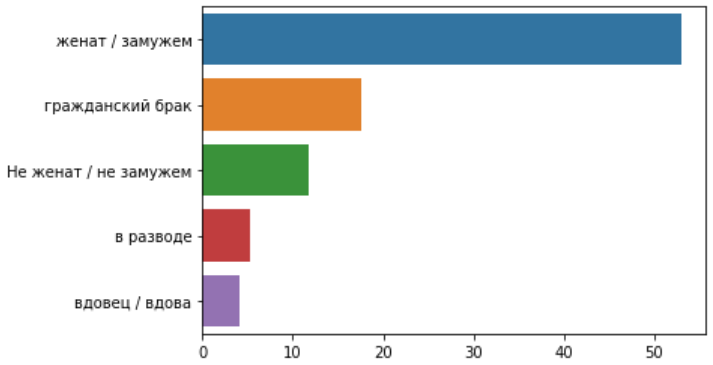
Сравнение с учетом количества.

In [33]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('family_status',6,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % от всех  ,   направление сортировки- по убыванию
```

Out[33]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	family_status	женат / замужем	12266	57.46	11339	92.44	53.12
1	family_status	гражданский брак	4146	19.42	3761	90.71	17.62
2	family_status	Не женат / не замужем	2796	13.10	2523	90.24	11.82
3	family_status	в разводе	1189	5.57	1105	92.94	5.18
4	family_status	вдовец / вдова	951	4.45	888	93.38	4.16



Вывод:

Разброс по категории небольшой. Сравнение показало, что самые аккуратные - вдовцы, потом те, кто в разводе и женатые/замужние. Живущие в гражданском браке или не женатые/незамужние почти не отличаются и показывают наименьший процент надежности по категории.

Если смотреть на рейтинг надежности заемщиков с учетом их количества, то наиболее ответственными будут женатые/замужние заемщики - они составляют 53,12 % от всех заемщиков в целом.

Задание 21. Есть ли зависимость между уровнем дохода и возвратом кредита в срок?

Категории клиентов по доходу

- 0–30000 — `'E'`;
- 30001–50000 — `'D'`;
- 50001–200000 — `'C'`;
- 200001–1000000 — `'B'`;
- 1000001 и выше — `'A'`.

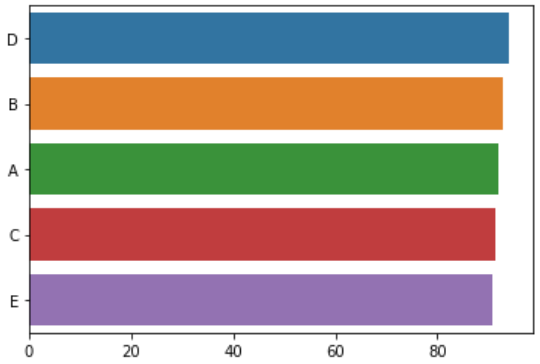
Сравнение по категории.

In [34]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('total_income_category',5,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % в подкатегории   ,   направление сортировки- по убыванию
```

Out[34]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	total_income_category	D	349	1.63	328	93.98	1.54
1	total_income_category	B	5014	23.49	4660	92.94	21.83
2	total_income_category	A	25	0.12	23	92.00	0.11
3	total_income_category	C	15938	74.66	14585	91.51	68.32
4	total_income_category	E	22	0.10	20	90.91	0.09



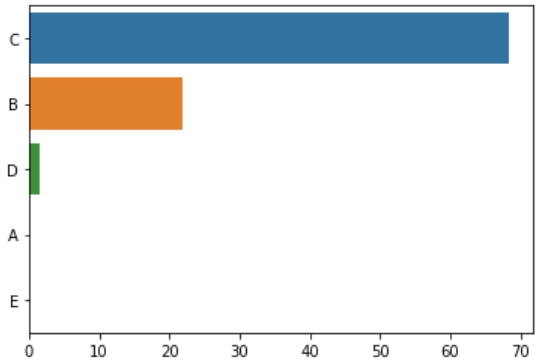
Сравнение с учетом количества.

In [35]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('total_income_category',6,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % от всех  ,   направление сортировки- по убыванию
```

Out[35]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	total_income_category	C	15938	74.66	14585	91.51	68.32
1	total_income_category	B	5014	23.49	4660	92.94	21.83
2	total_income_category	D	349	1.63	328	93.98	1.54
3	total_income_category	A	25	0.12	23	92.00	0.11
4	total_income_category	E	22	0.10	20	90.91	0.09



In [ ]:

Вывод:

Разброс по категории небольшой, наиболее ответственные - заемщики категории D с доходом 30 001- 50 000.

С учетом количества заемщиков, лидируют подгруппы C (50 001-200 000) и B (200 001-1 000 000), в сумме у них 90,15% вовзращающих кредиты от общего числа заемщиков**

Задание 22. Как разные цели кредита влияют на его возврат в срок?

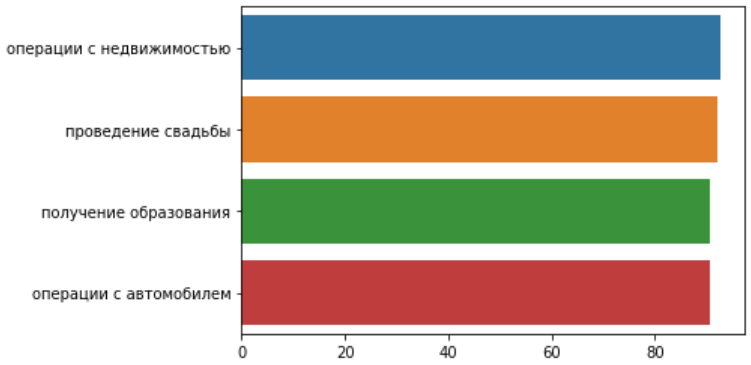
Сравнение по категории.

In [36]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('purpose_category',5,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % в подкатегории   ,   направление сортировки- по убыванию
```

Out[36]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	purpose_category	операции с недвижимостью	10754	50.37	9974	92.75	46.72
1	purpose_category	проведение свадьбы	2324	10.89	2141	92.13	10.03
2	purpose_category	получение образования	3989	18.69	3620	90.75	16.96
3	purpose_category	операции с автомобилем	4281	20.05	3881	90.66	18.18



In [ ]:

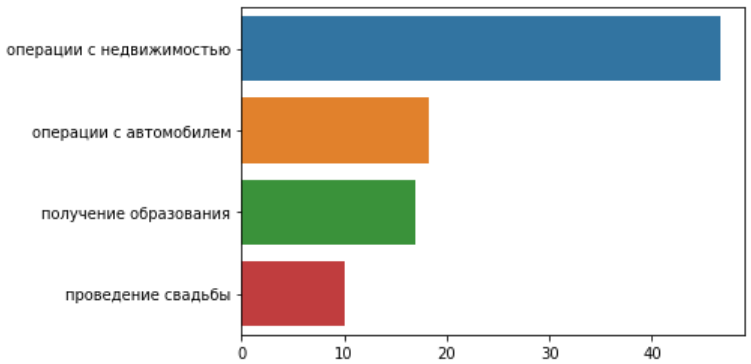
Сравнение с учетом количества.

In [37]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('purpose_category',6,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % от всех   ,   направление сортировки- по убыванию
```

Out[37]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	purpose_category	операции с недвижимостью	10754	50.37	9974	92.75	46.72
1	purpose_category	операции с автомобилем	4281	20.05	3881	90.66	18.18
2	purpose_category	получение образования	3989	18.69	3620	90.75	16.96
3	purpose_category	проведение свадьбы	2324	10.89	2141	92.13	10.03



In [ ]:

Вывод:

Разброс по категории в % незначителен, самые ответственные- заемщики с кредитами на недвижимость. Близко к ним находится кредитование на свадьбу. Заемщики с кредитами на получение образования и покупку автомобиля показывают близкий процент возврата кредита.

Если учитывать общее количество заемщиков, то наибольший процент возвращаемых среди кредитов на недвижимость- 46,72% от всех выданных кредитов. Покупка автомобиля и кредит на образование дают сопоставимый процент возврата и вместе показывают 35,14 % возвращаемых кредитов.

Задание 23. Как пол заемщика влияет на возврат кредита в срок?

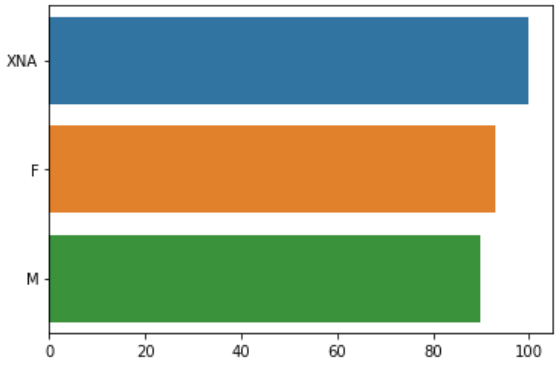
Сравнение по категории.

In [38]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('gender',5,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % в подкатегории   ,   направление сортировки- по убыванию
```

Out[38]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	gender	XNA	1	0.00	1	100.00	0.00
1	gender	F	14107	66.08	13118	92.99	61.45
2	gender	M	7240	33.91	6497	89.74	30.43



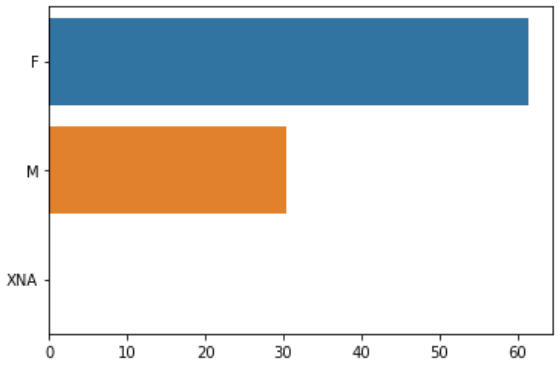
Сравнение с учетом количества.

In [39]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('gender',6,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % от всех  ,   направление сортировки- по убыванию
```

Out[39]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	gender	F	14107	66.08	13118	92.99	61.45
1	gender	M	7240	33.91	6497	89.74	30.43
2	gender	XNA	1	0.00	1	100.00	0.00



Вывод:

Однозначные лидеры в надежности заемщиков- мужчины. Они лидируют как в сравнении по категории, так и в общем количественном сравнении.

Отдельной категорией является неопределившийся с полом заемщик. Он, хотя и 100% надежный, но в количестве 1 человека является исключением для данной категории и на общие результаты не влияет.

Задание 24. Как образование заемщика влияет на возврат кредита в срок?

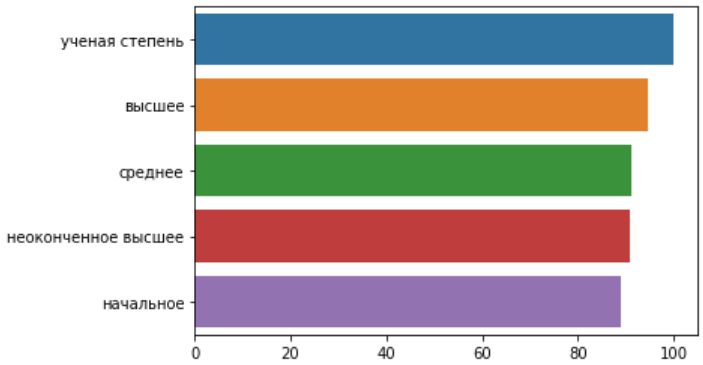
Сравнение по категории.

In [40]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('education',5,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % в подкатегории   ,   направление сортировки- по убыванию
```

Out[40]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	education	ученая степень	6	0.03	6	100.00	0.03
1	education	высшее	5228	24.49	4950	94.68	23.19
2	education	среднее	15091	70.69	13736	91.02	64.34
3	education	неоконченное высшее	741	3.47	673	90.82	3.15
4	education	начальное	282	1.32	251	89.01	1.18



In [ ]:
```
```

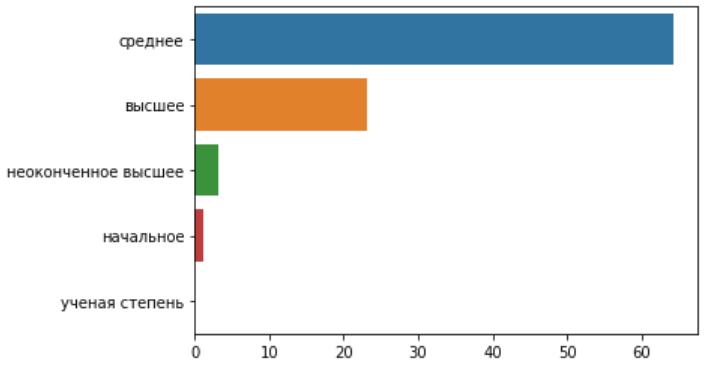
Сравнение с учетом количества.

In [41]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('education',6,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % от всех  ,   направление сортировки- по убыванию
```

Out[41]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	education	среднее	15091	70.69	13736	91.02	64.34
1	education	высшее	5228	24.49	4950	94.68	23.19
2	education	неоконченное высшее	741	3.47	673	90.82	3.15
3	education	начальное	282	1.32	251	89.01	1.18
4	education	ученая степень	6	0.03	6	100.00	0.03



Вывод:

Лучшим в сравнении по категории будет заемщик с ученой степенью. Но количественный учет показывает, что надежные заемщики с ученой степенью практически не влияют на общие результаты (0,03% от всех заемщиков).

Если не учитывать заемщика с ученой степенью в сравнении по категории и в количествено сравнении, то картина меняется: лучшим/ худшим в категори будут заемщики с высшим и начальным образованием соответственно. При количественном сравнении лучшим/худшим становятся заемщики со средним и начальным образованием соответственно.

Отдельно стоит отметить количественный состав заемщиков: среднее образование у 70,69 %, высшее- у 24,49 % , остальные составляют менее 5%.

Задание 25. Как возраст заемщика влияет на возврат кредита в срок?

Сперва необходимо категоризировать возраст заемщика по следующим диапазонам:

1. `'children-18 '` - дети до 18 лет
2. `'adults-18-29'` - взрослые от 18 до 29 лет включительно
3. `'adults-30-49'` - взрослые от 30 до 49 лет включительно
4. `'adults-50-64'` - взрослые от 50 до 64 лет включительно
5. `'retires_65  '` - взрослые пенсионеры от 65 лет включительно

In [42]:
```python
def categorize_age(income):
    try:
        if 0 <= income < 18:
            return 'children-18'
        elif 18 <= income <30:
            return 'adults-18-29'
        elif 30 <= income < 50:
            return 'adults-30-49'
        elif 50 <= income < 65:
            return 'adults-50-64'
        elif income >= 65:
            return 'retires_65'
    except:
        pass
```

In [43]:
```python
data['age_category'] = data['dob_years'].apply(categorize_age)
```

Количественный обзор.

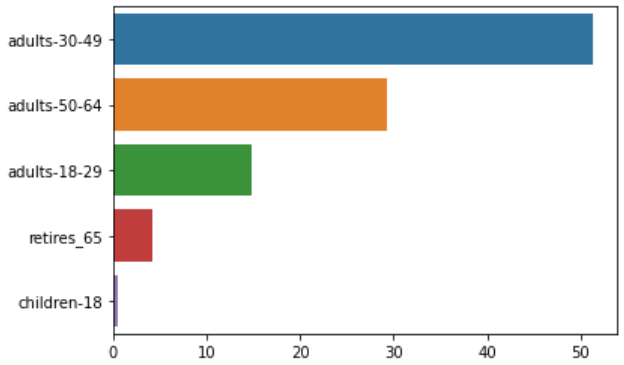
In [44]:
```python
# вызов функции  'category_factor' с параметрами:
# столбец -
```

```
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('age_category',3,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    % от всех   ,   направление сортировки- по убыванию
```

Out[44]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	age_category	adults-30-49	10948	51.28	9999	91.33	46.84
1	age_category	adults-50-64	6237	29.22	5860	93.96	27.45
2	age_category	adults-18-29	3167	14.84	2818	88.98	13.20
3	age_category	retires_65	896	4.20	847	94.53	3.97
4	age_category	children-18	100	0.47	92	92.00	0.43



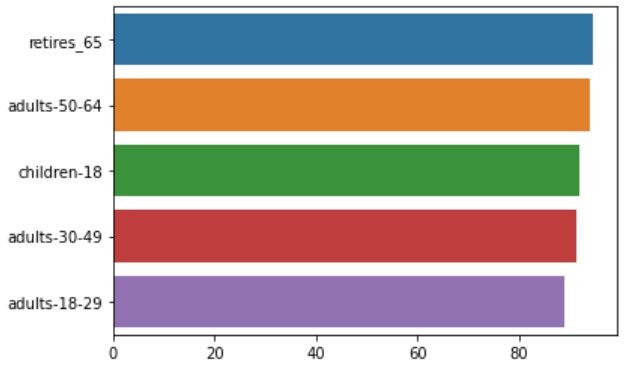
Сравнение по категории.

In [45]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('age_category',5,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % в подкатегории   ,   направление сортировки- по убыванию
```

Out[45]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	age_category	retires_65	896	4.20	847	94.53	3.97
1	age_category	adults-50-64	6237	29.22	5860	93.96	27.45
2	age_category	children-18	100	0.47	92	92.00	0.43
3	age_category	adults-30-49	10948	51.28	9999	91.33	46.84
4	age_category	adults-18-29	3167	14.84	2818	88.98	13.20



Сравнение с учетом количества.

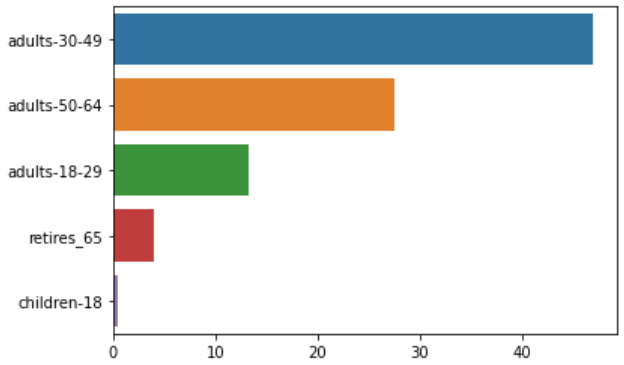
In [46]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)
```

```
report=category_factor('age_category',6,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % от всех  ,   направление сортировки- по убыванию
```

Out[46]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	age_category	adults-30-49	10948	51.28	9999	91.33	46.84
1	age_category	adults-50-64	6237	29.22	5860	93.96	27.45
2	age_category	adults-18-29	3167	14.84	2818	88.98	13.20
3	age_category	retires_65	896	4.20	847	94.53	3.97
4	age_category	children-18	100	0.47	92	92.00	0.43



Вывод:

В сравнении по категории лучшими/худшими являются пенсионеры (от 65) и молодежь (до 30 лет). Дети занимают середину рейтинга.

Но учет количества меняет картину. Дети и пенсионеры составляют вместе меньше 5% от всех заемщиков. Среди остальных лидируют взрослые в возрасте 30-49 лет. На них приходится 46,84% выплат всех кредитов вовремя. Менее ответственная молодежь показывает 13,2 % возвратов от всех кредитов.

Задание 26. Как тип занятости заемщика влияет на возврат кредита в срок?

Сравнение по категории.

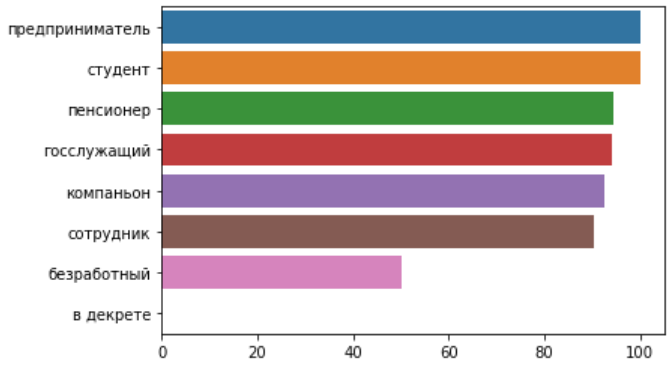
In [47]:

```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('income_type',5,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % в подкатегории  ,   направление сортировки- по убыванию
```

Out[47]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	income_type	предприниматель	2	0.01	2	100.00	0.01
1	income_type	студент	1	0.00	1	100.00	0.00
2	income_type	пенсионер	3820	17.89	3604	94.35	16.88
3	income_type	госслужащий	1451	6.80	1365	94.07	6.39
4	income_type	компаньон	5049	23.65	4675	92.59	21.90
5	income_type	сотрудник	11022	51.63	9968	90.44	46.69
6	income_type	безработный	2	0.01	1	50.00	0.00
7	income_type	в декрете	1	0.00	0	0.00	0.00



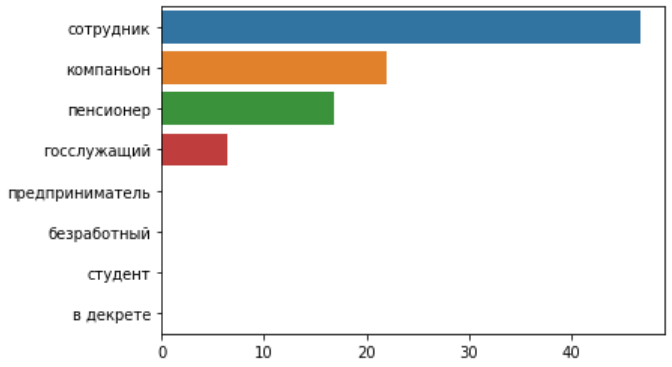
Сравнение с учетом количества.

In [48]:
```
# вызов функции  'category_factor' с параметрами:
# столбец -
# сортировка по столбцу отчета (1..6)
# направление сортировки (1-по возрастанию, 0- по убыванию)

report=category_factor('income_type',6,0)
report
```

```
Количество всех заемщиков 21348,    количество надежных заемщиков  19616, процент возврата в целом  91.89 %
Столбец сортировки отчета:    выплата в срок,  % от всех  ,   направление сортировки- по убыванию
```

Out[48]:

	категория	подкатегория	кол-во в подкатегории	% от всех	выплата в срок, чел	выплата в срок, % в подкатегории	выплата в срок, % от всех
0	income_type	сотрудник	11022	51.63	9968	90.44	46.69
1	income_type	компаньон	5049	23.65	4675	92.59	21.90
2	income_type	пенсионер	3820	17.89	3604	94.35	16.88
3	income_type	госслужащий	1451	6.80	1365	94.07	6.39
4	income_type	предприниматель	2	0.01	2	100.00	0.01
5	income_type	безработный	2	0.01	1	50.00	0.00
6	income_type	студент	1	0.00	1	100.00	0.00
7	income_type	в декрете	1	0.00	0	0.00	0.00



Вывод: в сравнении по категориям лучшими являются предпринимтаели и студенты, худшими - безработные и в декрете. Но тех и других пренебрежимо мало по сравнению с остальными. Если их исключить из выборки, то лучшим/худшим заемщиком будут пенсионер и сотрудник соответственно.

Если смотреть влияние на общую массу выплат, то сотрудники создают 46,69% выплат кредитов от общего их количества.

Задание 23. Приведите возможные причины появления пропусков в исходных данных.

Применительно к датасету настоящего исследования можно выделить следующие причины пропусков данных.

1. Неверно указанные данные при анкетировании заемщика.
2. Заведомо пропущенные данные при анкетировании заемщика.
3. Ошибки загрузки данных анкетирования в базу данных.
4. Ошибки выгрузки данных из БД.

Задание 24. Объясните, почему заполнить пропуски медианным значением — лучшее решение для количественных переменных.

Не вполне согласен, что заполнение пропусков медианным значением- лучшее решение. Необходимо анализировать характер пропусков и причину их появления. Возможно, что пропущенные значения в действительности были нулевыми, минимальными или максимальными, а не медианными. Пропущенные данные в величине дохода заемщика могут говорить как об отсутствии дохода (для безработных), так и об уровне дохода, величину которого заемщик предпочитает не афишировать. В целом заполнение пропуска данных медианным значением для группы позволит сохранить данные группы от выбросов значений, перекоса в значениях и,в целом, относительно корректно обработать данные всей группы не теряя отдельные строки из-за пропусков.

Общий вывод.

Анализ по категориям выявляет *лучшего* заемщика со следующими признаками: Вдовец с 5 детьми (или без детей), мужчина 65+, предпринимтель/пенсионер, с доходом 30 001-50 000, с ученой степенью или высшим образованием и кредитом на приобретение недвижимости.

Анализ по категориям выявляет *худшего* заемщика со следующими признаками: Незамужня женщина с 4 детьми, 18-29 лет, сотрудница с доходом 0-30 000, с начальным образованием и кредитом на автомобиль.

С точки зрения общего количества, наиболее влияющим на выплату кредитов вовремя будет заемщик: Мужчина в браке, без детей, 30-49 дет, сотрудник с доходом 50 000- 200 000, со средним или высшим образованием и кредитом на приобретение недвижимости.

Для более полной картины следует проверить величину кредита на вероятность его выплаты, а также влияение заемщиков с разными суммами кредитов на общую картину выплат. Например, один безответственный заемщик с кредитом в 10 000 000 принесет больше убытков, чем доход от 100 добросовестных заемщиков по 10 000. Но этих данных в датасете нет и такое исследование выходит за рамки настоящего проекта.