EDA данных о развитии малого и среднего предпринимательства Пиморского края

СОДЕРЖАНИЕ

0 ИМПОРТИРОВАНИЕ БИБЛИОТЕК

Библиотеки:

1. Библиотека **pandas** потребуется для работы с данными в табличном представлении.
2. Библиотека **numpy** потребуется для округления данных в корреляционной матрице.
3. Библиотека **copy** потребуется для создания глубоких копий объектов.
4. Библиотека **missingno** потребуется для визуализации пропусков в данных.
5. Библиотеки **matplotlib, seaborn** потребуется для визуализации данных.
6. Библиотека **scipy** потребуется для тестирования гипотез.
7. Библиотека **scikit_posthocs** потребуется для тестирования гипотез (тест Данна).

In [1]:
```
import pandas as pd
import numpy as np
import copy
import missingno
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import normaltest
from scipy.stats import mstats
import scikit_posthocs as sp
```

Дополнительные настройки:

1. Настройка **pd.set_option('display.max_columns', None)** позволит вывести все колонки датафрейма не скрывая их.
2. Настройка **pd.set_option('display.float_format', lambda x: '%.2f' % x)** позволит вывести большие числа в числовом представлении а не в экспоненциальном.

In [2]:
```
pd.set_option('display.max_columns', None)
pd.set_option('display.float_format', lambda x: '%.2f' % x)
```

1 ПРОСМОТР И ОПИСАНИЕ ДАННЫХ

В данном блоке будет осуществлен исследовательский анализ данных, описаны типы данных, пропуски, дубликаты, меры центральной тенденци а также взаимосвязи между переменными.

1.1 Типы данных

Подгружаем датасет.

In [3]:
```
df = pd.read_csv('data_msp.csv') # Загрузка датасета
df.info()                        # Вывод информации по типам данных, колличеству и пропускам
df.head()                        # Вывод первых 5 строк датасета
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38366 entries, 0 to 38365
Data columns (total 53 columns):
 #   Column                                                                Non-Null Count
Dtype
---  ------                                                                --------------
-----
 0   Наименование МСП                                                      38366 non-null
object
 1   ИНН                                                                   38366 non-null
int64
 2   Муниципальное образование                                             38366 non-null
object
 3   Вид                                                                   38366 non-null
object
 4   Категория                                                             38366 non-null
object
 5   Код ОКВЭД                                                             38366 non-null
object
 6   Вид деятельности                                                      38366 non-null
object
 7   Доход                                                                 33319 non-null
float64
 8   Расход                                                                33319 non-null
float64
 9   ЕСХН                                                                  21974 non-null
float64
 10  УСН                                                                   21974 non-null
float64
 11  ЕНВД                                                                  21974 non-null
float64
 12  СРП                                                                   21974 non-null
float64
 13  Акцизы н                                                              7922 non-null
float64
 14  Водный налог н                                                        7922 non-null
float64
 15  ЕНВД н                                                                7922 non-null
float64
 16  ЕСХН н                                                                7922 non-null
float64
 17  Задолженность и перерасчеты по отмененным налогам н                   7922 non-null
float64
 18  Земельный налог н                                                     7922 non-null
float64
 19  Неналоговые доходы, администрируемые налоговыми органами н            7922 non-null
float64
 20  НДС н                                                                 7922 non-null
float64
 21  НДПИ н                                                                7922 non-null
float64
 22  НДФЛ н                                                                7922 non-null
float64
 23  Налог на имущество организаций н                                      7922 non-null
float64
 24  Налог на прибыль н                                                    7922 non-null
float64
 25  УСН н                                                                 7922 non-null
float64
 26  Сборы за пользование объектами животного мира и за пользование объектами ВБР н  7922 non-null
float64
 27  Страховые взносы мед н                                                7922 non-null
float64
 28  Страховые взносы соц н                                                7922 non-null
float64
 29  Страховые и другие взносы пенс н                                      7922 non-null
float64
 30  Торговый сбор н                                                       7922 non-null
float64
 31  Транспортный налог н                                                  7922 non-null
float64
 32  Акцизы у                                                              35103 non-null
float64
 33  Водный налог у                                                        35103 non-null
float64
 34  ЕНВД у                                                                35103 non-null
float64
 35  ЕСХН у                                                                35103 non-null
float64
 36  Задолженность и перерасчеты по отмененным налогам у                   35103 non-null
float64
 37  Земельный налог у                                                     35103 non-null
float64
 38  Неналоговые доходы, администрируемые налоговыми органами у            35103 non-null
float64
 39  НДС у                                                                 35103 non-null
float64
 40  НДПИ у                                                                35103 non-null
float64
 41  НДФЛ у                                                                35103 non-null
float64
 42  Налог на игорный бизнес у                                             35103 non-null
float64
 43  Налог на имущество организаций у                                      35103 non-null
float64
 44  Налог на прибыль у                                                    35103 non-null
float64
 45  УСН у                                                                 35103 non-null
float64
 46  Сборы за пользование объектами животного мира и за пользование объектами ВБР у  35103 non-null
float64
 47  Страховые взносы мед у                                                35103 non-null
float64
 48  Страховые взносы соц у                                                35103 non-null
float64
 49  Страховые и другие взносы пенс у                                      35103 non-null
float64
 50  Торговый сбор у                                                       35103 non-null
float64
 51  Транспортный налог у                                                  35103 non-null
float64
 52  Утилизационный сбор у                                                 35103 non-null
float64
dtypes: float64(46), int64(1), object(6)
memory usage: 15.5+ MB
```

Out[3]:

	Наименование МСП	ИНН	Муниципальное образование	Вид	Категория	Код ОКВЭД	Вид деятельности	Доход	Расход	ЕС
0	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "АГРО...	7017227901	ГОРОД ВЛАДИВОСТОК	Юридическое лицо	Микропредприятие	01.25.1	Выращивание прочих плодовых и ягодных культур	0.00	3000.00	0
1	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "КРИП...	2537135920	ГОРОД ВЛАДИВОСТОК	Юридическое лицо	Микропредприятие	63.99.1	Деятельность по оказанию консультационных и ин...	0.00	0.00	r
2	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "АЙДАР"	2537141850	ГОРОД ВЛАДИВОСТОК	Юридическое лицо	Микропредприятие	46.15.4	Деятельность агентов по оптовой торговле радио...	0.00	0.00	0
3	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "АВТО...	2537139280	ГОРОД ВЛАДИВОСТОК	Юридическое лицо	Микропредприятие	45.20	Техническое обслуживание и ремонт автотранспор...	nan	nan	r
4	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "САДО...	2537138751	ГОРОД ВЛАДИВОСТОК	Юридическое лицо	Микропредприятие	71.12	Деятельность в области инженерных изысканий, и...	0.00	0.00	0

Датасет содержит 38366 записей с 53 столбцами, есть нулевые значения, пропуски, а также несоответствующие типы данных для переменных.

Приведем типы данных в соответствие:

1. Переменную **ИНН** приведем к строковому значению так как это не число а идентификационый номер.
2. Для столбцов **Вид, Категория**, которые являются категориальными признаками, т.е. признаками, которые принимают ограниченное и обычно фиксированное количество возможных значений, установим тип данных как «категория».

In [4]:
```
# Приводим к типу данных строка
df['ИНН'] = df['ИНН'].astype('str')

# Приводим к типу данных категория
df['Категория'] = df['Категория'].astype('category')
df['Муниципальное образование'] = df['Муниципальное образование'].astype('category')
```

Проверим, прошли ли изменения.

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38366 entries, 0 to 38365
Data columns (total 53 columns):
 #   Column                                                                Non-Null Count
Dtype
---  ------                                                                --------------
-----
 0   Наименование МСП                                                      38366 non-null
object
 1   ИНН                                                                   38366 non-null
object
 2   Муниципальное образование                                             38366 non-null
category
 3   Вид                                                                   38366 non-null
object
 4   Категория                                                             38366 non-null
category
 5   Код ОКВЭД                                                             38366 non-null
object
 6   Вид деятельности                                                      38366 non-null
object
 7   Доход                                                                 33319 non-null
float64
 8   Расход                                                                33319 non-null
float64
 9   ЕСХН                                                                  21974 non-null
float64
 10  УСН                                                                   21974 non-null
float64
 11  ЕНВД                                                                  21974 non-null
float64
 12  СРП                                                                   21974 non-null
float64
 13  Акцизы н                                                              7922 non-null
float64
 14  Водный налог н                                                        7922 non-null
float64
 15  ЕНВД н                                                                7922 non-null
float64
 16  ЕСХН н                                                                7922 non-null
float64
 17  Задолженность и перерасчеты по отмененным налогам н                   7922 non-null
float64
 18  Земельный налог н                                                     7922 non-null
float64
 19  Неналоговые доходы, администрируемые налоговыми органами н            7922 non-null
float64
 20  НДС н                                                                 7922 non-null
float64
 21  НДПИ н                                                                7922 non-null
float64
 22  НДФЛ н                                                                7922 non-null
float64
 23  Налог на имущество организаций н                                      7922 non-null
float64
 24  Налог на прибыль н                                                    7922 non-null
float64
 25  УСН н                                                                 7922 non-null
float64
 26  Сборы за пользование объектами животного мира и за пользование объектами ВБР н  7922 non-null
float64
 27  Страховые взносы мед н                                                7922 non-null
float64
 28  Страховые взносы соц н                                                7922 non-null
float64
 29  Страховые и другие взносы пенс н                                      7922 non-null
float64
 30  Торговый сбор н                                                       7922 non-null
float64
 31  Транспортный налог н                                                  7922 non-null
float64
 32  Акцизы у                                                              35103 non-null
float64
 33  Водный налог у                                                        35103 non-null
float64
 34  ЕНВД у                                                                35103 non-null
float64
 35  ЕСХН у                                                                35103 non-null
float64
 36  Задолженность и перерасчеты по отмененным налогам у                   35103 non-null
float64
 37  Земельный налог у                                                     35103 non-null
float64
 38  Неналоговые доходы, администрируемые налоговыми органами у            35103 non-null
float64
 39  НДС у                                                                 35103 non-null
float64
 40  НДПИ у                                                                35103 non-null
float64
 41  НДФЛ у                                                                35103 non-null
float64
 42  Налог на игорный бизнес у                                             35103 non-null
float64
 43  Налог на имущество организаций у                                      35103 non-null
float64
 44  Налог на прибыль у                                                    35103 non-null
float64
 45  УСН у                                                                 35103 non-null
float64
 46  Сборы за пользование объектами животного мира и за пользование объектами ВБР у  35103 non-null
float64
 47  Страховые взносы мед у                                                35103 non-null
float64
 48  Страховые взносы соц у                                                35103 non-null
float64
 49  Страховые и другие взносы пенс у                                      35103 non-null
float64
 50  Торговый сбор у                                                       35103 non-null
float64
 51  Транспортный налог у                                                  35103 non-null
float64
 52  Утилизационный сбор у                                                 35103 non-null
float64
dtypes: category(2), float64(46), object(5)
memory usage: 15.0+ MB
```

Удалим столбец **вид** так как для последующего анализа он не представляет ценности.

In [6]:
```
del df['Вид']
```

1.2 Пропуски в данных

Посчитаем процентное соотношение пропусков и заполненых значений для всех колонок.

In [7]:
```
for col in df.columns:
    pct_missing = df[col].isnull().mean()
    print(f'{col} - {pct_missing :.1%}')
```

```
Наименование МСП - 0.0%
ИНН - 0.0%
Муниципальное образование - 0.0%
Категория - 0.0%
Код ОКВЭД - 0.0%
Вид деятельности - 0.0%
Доход - 13.2%
Расход - 13.2%
ЕСХН - 42.7%
УСН - 42.7%
ЕНВД - 42.7%
СРП - 42.7%
Акцизы н - 79.4%
Водный налог н - 79.4%
ЕНВД н - 79.4%
ЕСХН н - 79.4%
Задолженность и перерасчеты по отмененным налогам н - 79.4%
Земельный налог н - 79.4%
Неналоговые доходы, администрируемые налоговыми органами н - 79.4%
НДС н - 79.4%
НДПИ н - 79.4%
НДФЛ н - 79.4%
Налог на имущество организаций н - 79.4%
Налог на прибыль н - 79.4%
УСН н - 79.4%
Сборы за пользование объектами животного мира и за пользование объектами ВБР н - 79.4%
Страховые взносы мед н - 79.4%
Страховые взносы соц н - 79.4%
Страховые и другие взносы пенс н - 79.4%
Торговый сбор н - 79.4%
Транспортный налог н - 79.4%
Акцизы у - 8.5%
Водный налог у - 8.5%
ЕНВД у - 8.5%
ЕСХН у - 8.5%
Задолженность и перерасчеты по отмененным налогам у - 8.5%
Земельный налог у - 8.5%
Неналоговые доходы, администрируемые налоговыми органами у - 8.5%
НДС у - 8.5%
НДПИ у - 8.5%
НДФЛ у - 8.5%
Налог на игорный бизнес у - 8.5%
Налог на имущество организаций у - 8.5%
Налог на прибыль у - 8.5%
УСН у - 8.5%
Сборы за пользование объектами животного мира и за пользование объектами ВБР у - 8.5%
Страховые взносы мед у - 8.5%
Страховые взносы соц у - 8.5%
Страховые и другие взносы пенс у - 8.5%
Торговый сбор у - 8.5%
Транспортный налог у - 8.5%
Утилизационный сбор у - 8.5%
```

Наибольшее количество пропусков составляют колонки:

79.4%:

- Акцизы н
- Водный налог н
- ЕНВД н
- ЕСХН н
- Задолженность и перерасчеты по отмененным налогам н
- Земельный налог н
- Неналоговые доходы, администрируемые налоговыми органами н
- НДС н
- НДПИ н
- НДФЛ н
- Налог на имущество организаций н
- Налог на прибыль н
- УСН н
- Сборы за пользование объектами животного мира и за пользование объектами ВБР н
- Страховые взносы мед н
- Страховые взносы соц н
- Страховые и другие взносы пенс н
- Торговый сбор н
- Транспортный налог н

42.7%:

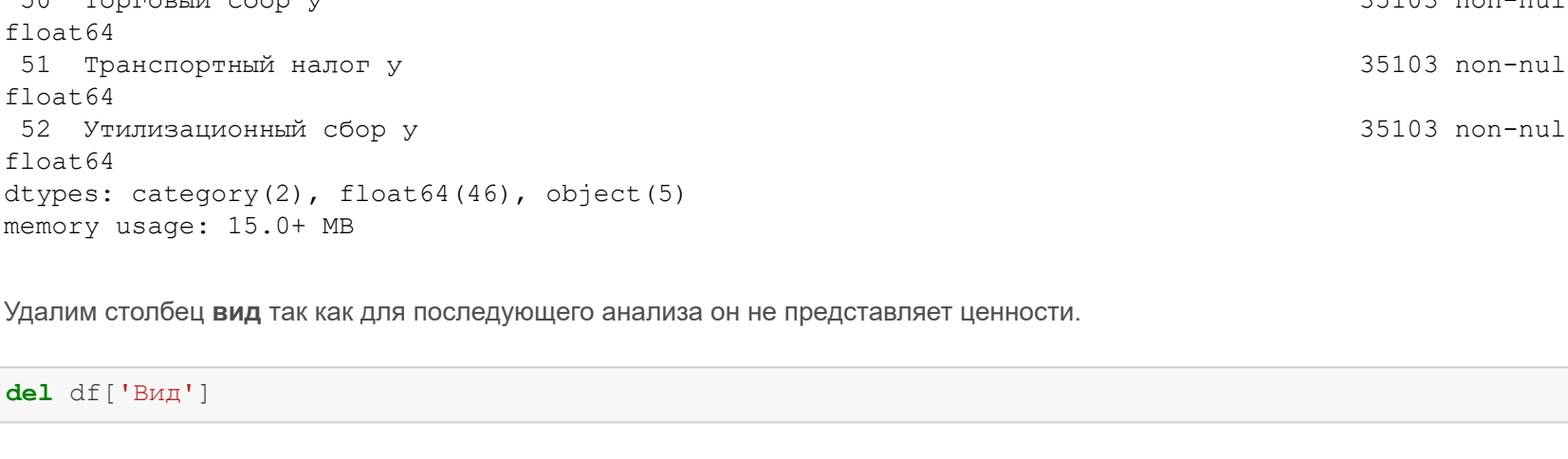
- ЕСХН
- УСН
- ЕНВД
- СРП

По остальным колонкам пропуски составляют небольшую долю в данных.

Визуализируем матрицу пропусков.

In [8]:
```
print('Визуализация пропущенных значений со всеми пропусками.')
missingno.matrix(df, fontsize = 14) # Передаем в качестве аргументов датафрейм, а также указываем разме
р шрифта
plt.show()
```

Визуализация пропущенных значений со всеми пропусками.



In [9]:
```
print('Колонки с наибольшим количеством пропусков.')
df.iloc[:, 12:31].columns # Берем срез от датафрейма и выводим названия колонок
```

Колонки с наибольшим количеством пропусков.

Out[9]:
```
Index(['Акцизы н', 'Водный налог н', 'ЕНВД н', 'ЕСХН н',
       'Задолженность и перерасчеты по отмененным налогам н',
       'Земельный налог н',
       'Неналоговые доходы, администрируемые налоговыми органами н', 'НДС н',
       'НДПИ н', 'НДФЛ н', 'Налог на имущество организаций н',
       'Налог на прибыль н', 'УСН н',
       'Сборы за пользование объектами животного мира и за пользование объектами ВБР н',
       'Страховые взносы мед н', 'Страховые взносы соц н',
       'Страховые и другие взносы пенс н', 'Торговый сбор н',
       'Транспортный налог н'],
      dtype='object')
```

In [10]:
```
print('Визуализация пропущенных значений, в случае если убрать наибольшую часть пропусков в датасете.')
missingno.matrix(df[df['Акцизы н'].notnull()], fontsize = 14) # Передаем в качестве аргументов датафрей
м,
                                                               # а также указываем размер шрифта
plt.show()
```

Визуализация пропущенных значений, в случае если убрать наибольшую часть пропусков в датасете.

In [11]:
```
print('Остались следующие колонки с наибольшим количеством пропусков.')
df[df['Доход'].notnull()].iloc[:, 6:12].columns # Берем срез от датафрейма и выводим названия колонок
```

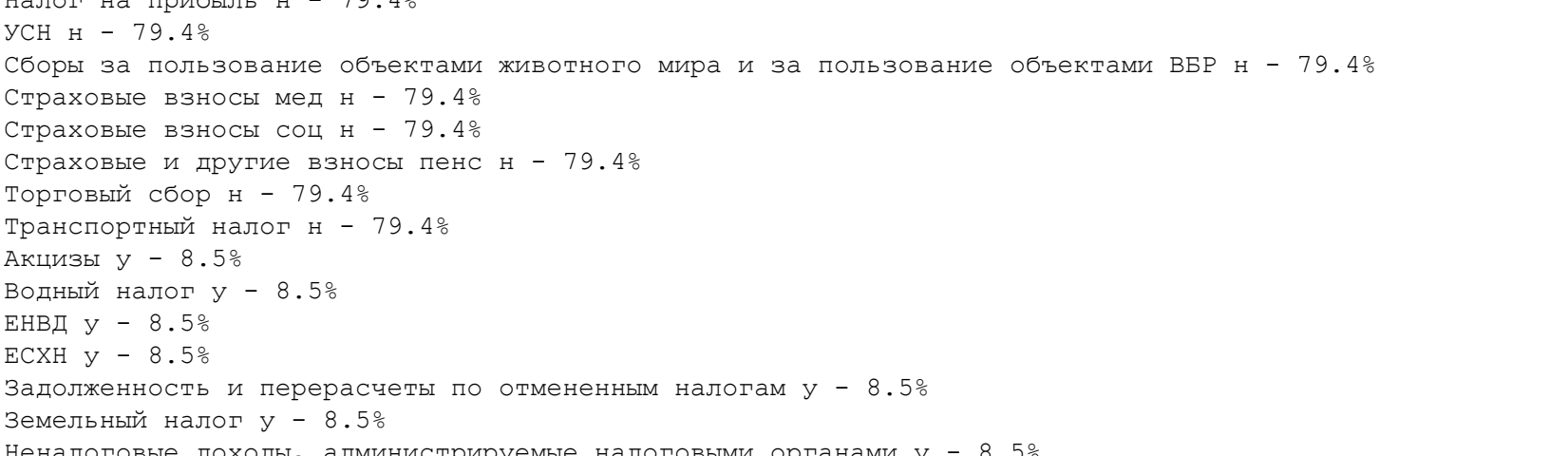
Остались следующие колонки с наибольшим количеством пропусков.

Out[11]:
```
Index(['Доход', 'Расход', 'ЕСХН', 'УСН', 'ЕНВД', 'СРП'], dtype='object')
```

В дальнейшем, колонки с системами налогообложения ценности особой не представляют, а вот информация о доходе и расходе у субъектов малого и среднего предпринимательства может потребоваться. Почистим в них пропуски и сформируем итоговый датафрейм.

In [12]:
```
print('Визуализация пропущенных значений, в случае если оставить только необходимую для дальнейшего ана
лиза информацию.')
# Передаем в качестве аргументов датафрейм, а также указываем размер шрифта
missingno.matrix(df[df['Акцизы н'].notnull() & df['Доход'].notnull()], fontsize = 14)
plt.show()
```

Визуализация пропущенных значений, в случае если оставить только необходимую для дальнейшего анализа информацию.



In [13]:
```
print('Колонки с наибольшим количеством пропусков.')
df[df['Акцизы н'].notnull() & df['Доход'].notnull()].iloc[:, 8:12].columns
```

Колонки с наибольшим количеством пропусков.

Out[13]:
```
Index(['ЕСХН', 'УСН', 'ЕНВД', 'СРП'], dtype='object')
```

Так как проводить анализ по данным, содержащим подавляющее число пропусков нецелесообразно, создадим датафрейм с максимально возможной информацией без пропусков для последующего анализа.

In [14]:
```
data_for_analysis = copy.deepcopy(df[df['Акцизы н'].notnull() & df['Доход'].notnull()])
```

Проверим количество муниципальных образований в данных. Важно что бы их было 34, так как в дальнейшем планируется проверить гипотезу где потребуются данные по каждому муниципалитету.

In [15]:
```
if len(set(data_for_analysis['Муниципальное образование'])) == 34:
    print('Данные содержат информацию по каждому муниципальному образованию')
else:
    print('Данные содержат информацию не по каждому муниципальному образованию')
```

```
Данные содержат информацию по каждому муниципальному образованию
```

1.3 Дубликаты

1.3.1 Наименование МСП

In [17]:
```
if df['Наименование МСП'].duplicated().any() == True:
    print('Данные содержат дубликаты.')
else:
    print('Данные не содержат дубликаты.')
```

```
Данные содержат дубликаты.
```

Данные о наименовании субъекта МСП содержат дубликаты по причине того, что названия могут быть одинаковые. В рамках данного анализа информация не представляет особого интереса.

1.3.2 ИНН

In [18]:
```
if df['ИНН'].duplicated().any() == True:
    print('Данные содержат дубликаты.')
else:
    print('Данные не содержат дубликаты.')
```

```
Данные не содержат дубликаты.
```

Данные об ИНН субъекта МСП не содержат дубликаты по причине того, что ИНН это уникальный номер, он не может быть один у двух и более юридических лиц или индивидуальных предпримиателей. По данной информации можно строить агрегаты, а также использовать в качестве уникального ключа, идентифицирующего конкретную запись о субъекте МСП.

1.3.3 Муниципальное образование

In [19]:
```
if df['Муниципальное образование'].duplicated().any() == True:
    print('Данные содержат дубликаты.')
else:
    print('Данные не содержат дубликаты.')
```

```
Данные содержат дубликаты.
```

Данные о муниципальном образовании, в котором зарегистрирован субъект МСП содержат дубликаты по причине того, что муниципальное образование это территория, на которой может осуществлять свою деятельность более одного субъекта МСП. По данной категориальной переменной можно строить агрегированную информацию в разрезе муниципальных образований.

1.3.4 Категория

In [20]:
```
if df['Категория'].duplicated().any() == True:
    print('Данные содержат дубликаты.')
else:
    print('Данные не содержат дубликаты.')
```

```
Данные содержат дубликаты.
```

Данные о категории, к которой отнесен субъект МСП содержат дубликаты по причине того, что к одной и той же категории может относится более одного субъекта МСП. По данной категориальной переменной можно строить агрегированную информацию в разрезе категорий МСП.

1.3.5 Код ОКВЭД

In [21]:
```
if df['Код ОКВЭД'].duplicated().any() == True:
    print('Данные содержат дубликаты.')
else:
    print('Данные не содержат дубликаты.')
```

```
Данные содержат дубликаты.
```

Данные о коде ОКВЭД, указанном как основной вид деятельности у субъекта МСП, содержат дубликаты по причине того, что субъекты МСП могут осуществлять одни и теже виды деятельности. По данной категориальной переменной можно строить агрегированную информацию в разрезе ОКВЭД МСП (в данном анализе это не представляет особого интереса).

1.3.6 Вид деятельности

In [22]:
```
if df['Вид деятельности'].duplicated().any() == True:
    print('Данные содержат дубликаты.')
else:
    print('Данные не содержат дубликаты.')
```

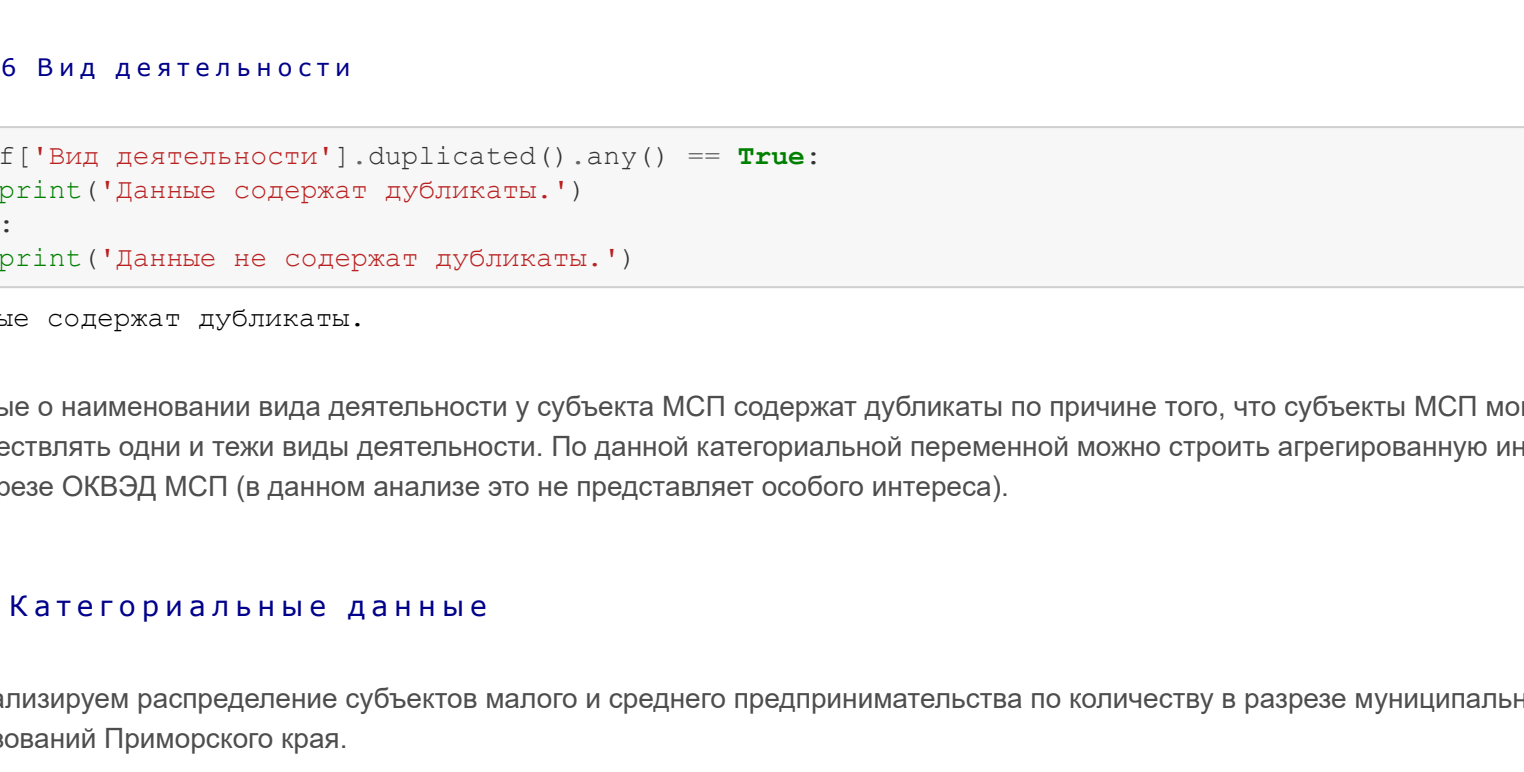
```
Данные содержат дубликаты.
```

Данные о наименовании вида деятельности у субъекта МСП содержат дубликаты по причине того, что субъекты МСП могут осуществлять одни и теже виды деятельности. По данной категориальной переменной можно строить агрегированную информацию в разрезе ОКВЭД МСП (в данном анализе это не представляет особого интереса).

1.4 Категориальные данные

Визуализируем распределение субъектов малого и среднего предпринимательства по количеству в разрезе муниципальных образований Приморского края.

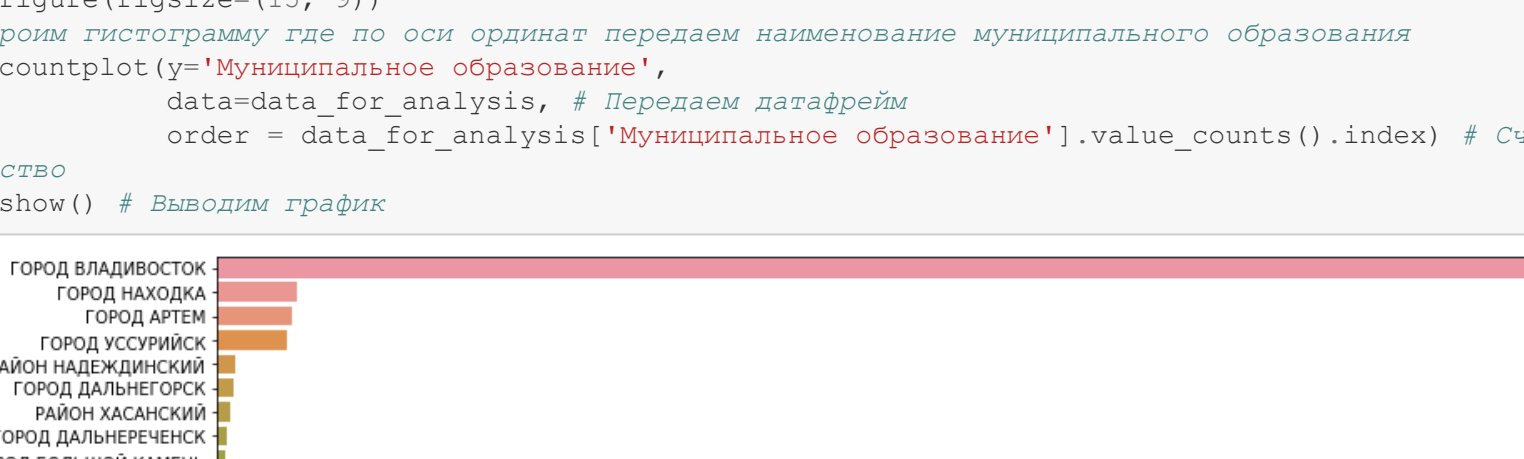
In [23]:
```
# Задаем размеры визуализации
plt.figure(figsize=(15, 9))
# Строим гистограмму где по оси ординат передаем наименование муниципального образования
sns.countplot(y='Муниципальное образование',
              data=data_for_analysis, # Передаем датафрейм
              order = data_for_analysis['Муниципальное образование'].value_counts().index) # Считаем ко
личество
plt.show() # Выводим график
```



Как видно из данной визуализации подавляющее число субъектов малого и среднего предпринимательства сосредоточено в городе Владивостоке, далее идет город Находка, Артем и Усурийск. В остальных муниципалитетах сосредоточенно примерно одинаковое количество субъектов малого и среднего предпринимательтсва региона что вполне соответствует реальности.

Визуализируем распределение субъектов малого и среднего предпринимательства по количеству в разрезе категорий.

In [24]:
```
# Задаем размеры визуализации
plt.figure(figsize=(15, 2))
# Строим гистограмму где по оси ординат передаем категорию бизнеса
sns.countplot(y='Категория',
              data=data_for_analysis, # Передаем датафрейм
              order = data_for_analysis['Категория'].value_counts().index) # Считаем количество
plt.show() # Выводим график
```



Как видно из данной визуализации подавляющее число субъектов малого и среднего предпринимательства относится к категории микропредприятий, затем идут малые предприятия и самую незначительную долю составляет средний бизнес.

1.5 Числовые данные

Выведем меры центральной тенденции.

In [26]:
```
data_for_analysis.describe().T # Выводим количесвто, ско, среднее, процентили, минимальное и максимальное значение, размах
```

Out[26]:

	count	mean	std	min	25%	50%	75%	max
Доход	7397.00	43374076.03	1843882471.72	-173000.00	0.00	711000.00	8653000.00	158305175000.00
Расход	7397.00	42458360.09	1843900718.30	0.00	0.00	767000.00	7906000.00	158305175000.00
ЕСХН	4503.00	0.00	0.06	0.00	0.00	0.00	0.00	1.00
УСН	4503.00	0.95	0.23	0.00	1.00	1.00	1.00	1.00
ЕНВД	4503.00	0.13	0.34	0.00	0.00	0.00	0.00	1.00
СРП	4503.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Акцизы н	7397.00	2.72	164.85	0.00	0.00	0.00	0.00	13629.00
Водный налог н	7397.00	182.83	14609.82	0.00	0.00	0.00	0.00	1255666.22
ЕНВД н	7397.00	615.90	10508.81	0.00	0.00	0.00	0.00	431879.57
ЕСХН н	7397.00	29.74	2526.31	0.00	0.00	0.00	0.00	217271.33
Задолженность и перерасчеты по отмененным налогам н	7397.00	61.19	2088.25	0.00	0.00	0.00	0.00	157474.34
Земельный налог н	7397.00	4721.40	301574.71	0.00	0.00	0.00	0.00	25824655.82
Неналоговые доходы, администрируемые налоговыми органами н	7397.00	801.03	19891.68	0.00	0.00	0.00	0.00	1543050.00
НДС н	7397.00	102702.42	2372022.00	0.00	0.00	0.00	0.00	117242849.65
НДПИ н	7397.00	39.49	2957.19	0.00	0.00	0.00	0.00	252380.83
НДФЛ н	7397.00	17603.05	518681.96	0.00	0.00	0.00	0.00	39585371.85
Налог на имущество организаций н	7397.00	26588.82	1247140.17	0.00	0.00	0.00	0.00	80281759.62
Налог на прибыль н	7397.00	15042.21	521480.18	0.00	0.00	0.00	0.00	38338171.78
УСН н	7397.00	3662.49	44706.53	0.00	0.00	0.00	0.00	2191959.24
Сборы за пользование объектами животного мира и за пользование объектами ВБР н	7397.00	2.13	142.03	0.00	0.00	0.00	0.00	11999.53
Страховые взносы мед н	7397.00	7022.70	203966.81	0.00	0.00	0.00	0.60	15372111.02
Страховые взносы соц н	7397.00	2457.02	52521.74	0.00	0.00	0.00	0.41	2551248.71
Страховые и другие взносы пенс н	7397.00	29957.21	844876.04	0.00	0.00	0.00	6.51	61508654.23
Торговый сбор н	7397.00	12.37	1031.27	0.00	0.00	0.00	0.00	88656.40
Транспортный налог н	7397.00	2106.30	118731.67	0.00	0.00	0.00	0.00	9955895.29
Акцизы у	7397.00	5702.88	289392.53	0.00	0.00	0.00	0.00	22223936.00
Водный налог у	7397.00	61.19	2531.79	0.00	0.00	0.00	0.00	196115.00
ЕНВД у	7397.00	9040.63	73817.52	0.00	0.00	0.00	0.00	3225074.00
ЕСХН у	7397.00	81.65	5267.52	0.00	0.00	0.00	0.00	448552.00
Задолженность и перерасчеты по отмененным налогам у	7397.00	0.38	16.55	0.00	0.00	0.00	0.00	872.56
Земельный налог у	7397.00	16863.84	143667.65	0.00	0.00	0.00	0.00	4051650.00
Неналоговые доходы, администрируемые налоговыми органами у	7397.00	118.00	5421.83	0.00	0.00	0.00	0.00	394728.00
НДС у	7397.00	273657.71	2300684.66	0.00	0.00	0.00	0.00	100595330.00
НДПИ у	7397.00	2784.71	116179.49	0.00	0.00	0.00	0.00	9010000.00
НДФЛ у	7397.00	267.42	5203.04	0.00	0.00	0.00	0.00	326056.00
Налог на игорный бизнес у	7397.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Налог на имущество организаций у	7397.00	16092.43	209374.10	0.00	0.00	0.00	0.00	11005118.00
Налог на прибыль у	7397.00	105148.86	1151649.12	0.00	0.00	0.00	0.00	55659310.00
УСН у	7397.00	92981.14	381700.79	0.00	0.00	0.00	28535.00	9823589.88
Сборы за пользование объектами животного мира и за пользование объектами ВБР у	7397.00	1314.68	77312.87	0.00	0.00	0.00	0.00	6516869.00
Страховые взносы мед у	7397.00	52468.28	249257.35	0.00	0.00	4541.70	26968.81	7465047.00
Страховые взносы соц у	7397.00	28470.63	125833.68	0.00	0.00	2610.00	15307.65	3081898.21
Страховые и другие взносы пенс у	7397.00	230091.76	1041065.91	0.00	0.00	20268.86	118428.00	26984659.00
Торговый сбор у	7397.00	47.65	2929.90	0.00	0.00	0.00	0.00	202500.00
Транспортный налог у	7397.00	3209.13	28869.77	0.00	0.00	0.00	0.00	983910.00
Утилизационный сбор у	7397.00	2285.39	196556.48	0.00	0.00	0.00	0.00	16905000.00

In [29]:
```
data_for_analysis.mode()[:1] # Смотрим на моду в данных
```

Out[29]:

	Наименование МСП	ИНН	Муниципальное образование	Категория	Код ОКВЭД	Вид деятельности	Доход	Расход	ЕСХН	УСН	ЕНВД
0	ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ "ВОСТОК"	1831187536	ГОРОД ВЛАДИВОСТОК	Микропредприятие	41.20	Строительство жилых и нежилых зданий	0.00	0.00	0.00	1.00	0.00

Выводы:.

1. Данные имеют большой размах;
2. В данных присутствуют нулевые значения (естественные, не отсутствие значения);
3. Данные имеют выбросы (естественные);
4. Присутствуют отрицательные значения;
5. В большенстве колонок (уплата / не уплата) до 75% процентиля лежат нули;
6. Среднее значение сильно отличается от медианы;
7. Колонка "налог на игорный бизнес у" содержит только нулевые значения;
8. В числовых данных наиболее часто встречаемое значение равное нулю (чаще всего бизнес платит налоги в полном объеме);
9. Ниболее часто встречаемый вид деятельности 41.20 "Строительство жилых и нежилых зданий";
10. Нииболее часто встречаемая категория МСП микропредприятие;
11. Наиболее часто встречаемый муниципалитет город Владивосток.

Удалим колонки, которые в дальнейшем не потребуются.

In [30]:
```
data_for_analysis.drop(['Наименование МСП',
                        'Код ОКВЭД',
                        'Вид деятельности',
                        'ЕСХН',
                        'УСН',
                        'ЕНВД',
                        'СРП',
                        'Налог на игорный бизнес у'], axis=1, inplace=True)
```

Визуализируем распределения по всем числовым признакам датафрейма. Построим боксплот и частотный график для каждой колонки.

In [31]:
```
def vizualization_number_data(df, start_column, width, height):
    """
        Функция возвращающая для каждой числовой колонки датафрейма
        боксплот и частотный график

    Аргументы функции:
        df - датафрейм
        start_column - начало среза (с какой колонки начать)
        width - ширина графика
        height - высота графика

    Возвращаемое значение:
        Функция визуализирует для каждой числовой переменной частотный график и боксплот

    """
    # Задаем количество строк вывода
    quantity_string = len(df.iloc[:, start_column:].columns)
    # Создаем счетчик графика
    counter_graph = 0

    # В цикле итерируемся по каждой колонке
    for i in df.iloc[:, start_column:]:
        # Задаем размеры графиков
        plt.figure(figsize=(width, height))
        # Наращиваем порядковый номер графика
        counter_graph += 1
        # Задаем параметры отображения графика
        # quantity_string - количество строк
        # 2 - количество графиков в строке
        # counter_graph - порядковый номер графика
        plt.subplot(quantity_string, 2, counter_graph)
        # Создаем бокс плот
        # В x передаем наименование колонки
        # В data передаем датафрейм
        sns.boxplot(x=i, data=df)
        # Наращиваем порядковый номер графика
        counter_graph += 1
        # Задаем параметры отображения графика
        # quantity_string - количество строк
        # 2 - количество графиков в строке
        # counter_graph - порядковый номер графика
        plt.subplot(quantity_string, 2, counter_graph)
        # Создаем частотный график
        # В i передаем название колонки
        # kde=False убираем линию с частотного графика
        sns.distplot(df[i], kde=False)
        plt.show()
```

In [32]:
```
vizualization_number_data(data_for_analysis, 3, 20, 155)
```

Из построенных визуализаций видно, что данные имеют выбросы, и гамма распределение (сильная ассиметрия данных).

Для установления связей между числовыми переменными построим и визуализируем корреляционную матрицу.

In [33]:
```
# Создадим датафрейм для корреляционной матрицы
data_matrix = data_for_analysis[['Доход', 'Расход','Акцизы н', 'Водный налог н', 'ЕНВД н', 'ЕСХН н',
                                 'Задолженность и перерасчеты по отмененным налогам н','Земельный налог н',
                                 'Неналоговые доходы, администрируемые налоговыми органами н', 'НДС н',
                                 'НДПИ н', 'НДФЛ н', 'Налог на имущество организаций н', 'Налог на прибыль н', 'УСН н',
                                 'Сборы за пользование объектами животного мира и за пользование объектами ВБР н',
                                 'Страховые взносы мед н', 'Страховые взносы соц н', 'Страховые и другие взносы пенс н',
                                 'Торговый сбор н', 'Транспортный налог н', 'Акцизы у', 'Водный налог у', 'ЕНВД у',
                                 'ЕСХН у', 'Задолженность и перерасчеты по отмененным налогам у', 'Земельный налог у',
                                 'Неналоговые доходы, администрируемые налоговыми органами у', 'НДС у', 'НДПИ у',
                                 'НДФЛ у', 'Налог на имущество организаций у', 'Налог на прибыль у', 'УСН у',
                                 'Сборы за пользование объектами животного мира и за пользование объектами ВБР у',
                                 'Страховые взносы мед у', 'Страховые взносы соц у', 'Страховые и другие взносы пенс у',
                                 'Торговый сбор у', 'Транспортный налог у', 'Утилизационный сбор у']]
```

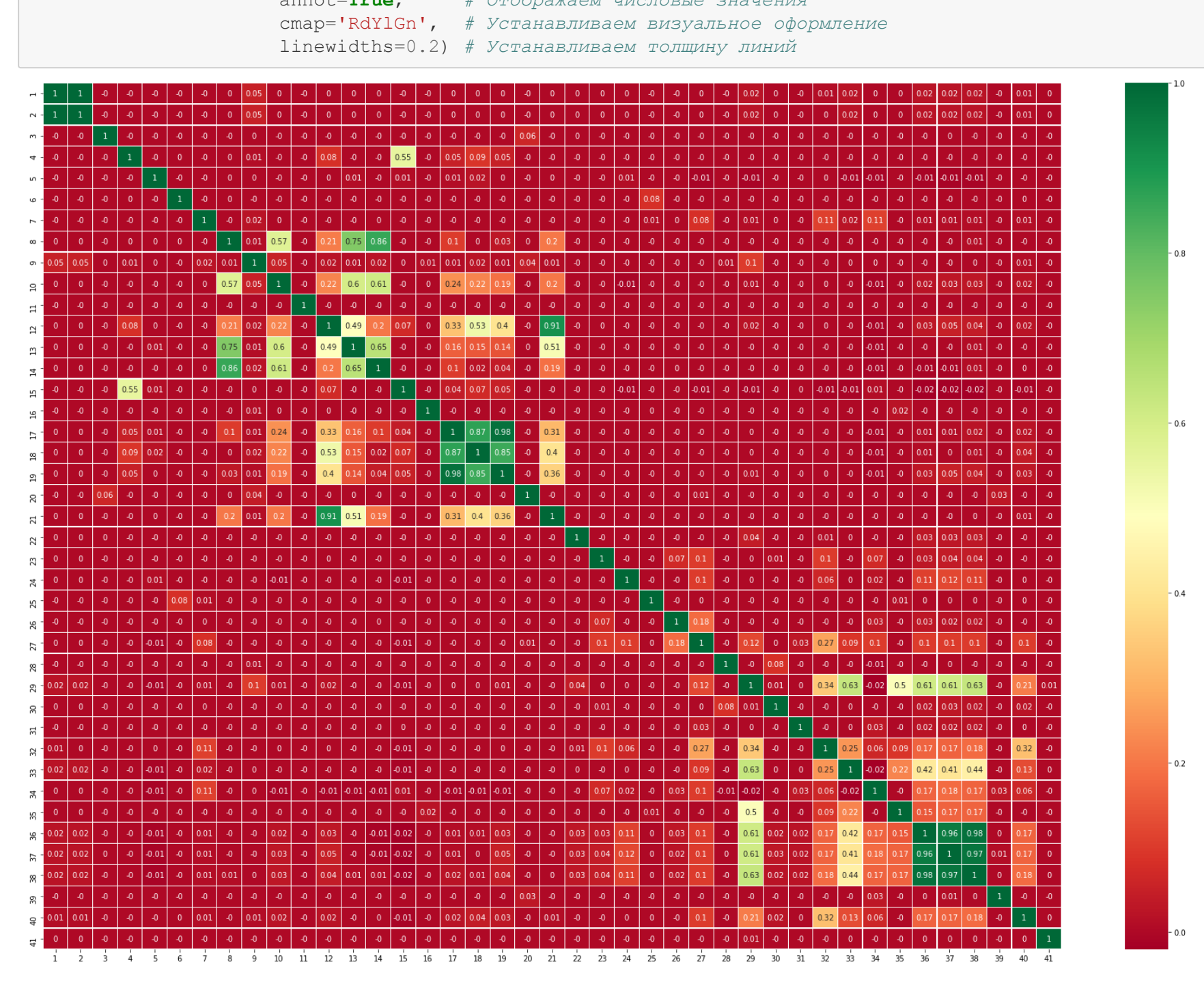
Переименуем названия признаков для удобства визуализации корреляционной матрицы:

1. Доход.
2. Расход.
3. Акцизы н.
4. Водный налог н.
5. ЕНВД н.
6. ЕСХН н.
7. Задолженность и перерасчеты по отмененным налогам н.
8. Земельный налог н.
9. Неналоговые доходы, администрируемые налоговыми органами н.
10. НДС н.
11. НДПИ н.
12. НДФЛ н.
13. Налог на имущество организаций н.
14. Налог на прибыль н.
15. УСН н.
16. Сборы за пользование объектами животного мира и за пользование объектами ВБР н.
17. Страховые взносы мед н.
18. Страховые взносы соц н.
19. Страховые и другие взносы пенс н.
20. Торговый сбор н.
21. Транспортный налог н.
22. Акцизы у.
23. Водный налог у.
24. ЕНВД у.
25. ЕСХН у.
26. Задолженность и перерасчеты по отмененным налогам у.
27. Земельный налог у.
28. Неналоговые доходы, администрируемые налоговыми органами у.
29. НДС у.
30. НДПИ у.
31. НДФЛ у.
32. Налог на имущество организаций .
33. Налог на прибыль у.
34. УСН у.
35. Сборы за пользование объектами животного мира и за пользование объектами ВБР у.
36. Страховые взносы мед у.
37. Страховые взносы соц у.
38. Страховые и другие взносы пенс у.
39. Торговый сбор у.
40. Транспортный налог у.
41. Утилизационный сбор у.

In [34]:
```
# Зададим новые названия колонок
data_matrix.columns = ['1', '2', '3', '4', '5', '6',
                       '7', '8', '9', '10', '11', '12',
                       '13', '14', '15', '16', '17', '18',
                       '19', '20', '21', '22', '23', '24',
                       '25', '26', '27', '28', '29', '30',
                       '31', '32', '33', '34', '35', '36',
                       '37', '38', '39', '40', '41']
```

In [35]:
```
# Построим корреляционную матрицу
matrix_corr = np.round(data_matrix.corr(), decimals=2)
```

In [36]:
```
# Создадим размеры визуализации
plt.figure(figsize=(30, 21))
# Визуализируем корреляционную матрицу
heatmap = sns.heatmap(matrix_corr,     # Передаем датафрейм
                      annot=True,      # Отображаем числовые значения
                      cmap='RdYlGn',   # Устанавливаем визуальное оформление
                      linewidths=0.2) # Устанавливаем толщину линий
```



Корреляционная матрица показывает:

1. Отсутствие обратной линейной зависимости между числовыми признаками датасета;
2. Наличие сильной прямой линейной зависимости между доходом и расходом у субъектов малого и среднего предпринимательства что вполне более чем логично. Как правило чем выше доходы компании тем крупнее компания и тем больше нужно нести расходов, обеспечивающих компании прибыль и эффективное функционирование;
3. Наличие средней прямой линейной зависимости между суммами неуплаченных налогов что более чем логично. Обычно, недобросовестные компании неупалчивают все типы налогов (за исключением компаний, где мог быть расчитан не корректно какой то конкретный налог и соответственно произошло недопоступление в бюджет), таким образом чем выше неуплата по одному налогу тем выше она может быть и по другим налогам (например налог на прибыль и НДС);
4. Наличие средней прямой линейной зависимости между суммами уплаченных налогов что более чем логично. Обычно, с ростом одного налога к уплате и растет сумма иных налогов (например налог на прибыль и ндс);
5. Наличие сильной прямой линейной зависимости между суммами неуплаченных страховых взносов что вполне логично, так как база для расчета единая для всех тарифов страховых взносов;
6. Наличие сильной прямой линейной зависимости между суммами уплаченных страховых взносов что вполне логично, так как база для расчета единая для всех тарифов страховых взносов.

Так как цель анализа установить различия налоговой платежеспособности у субъектов малого и среднего предпринимательства в муниципальных образованиях, в такой декомпозиции данных нет необходимости. Объединим группы данных в новые переменные.

In [38]:
```
# Создадим новую колонку с суммой неуплаченных налогов
data_for_analysis['Налоги долг'] = data_for_analysis[['Акцизы н', 'Водный налог н', 'ЕНВД н', 'ЕСХН н',
                                    'Земельный налог н', 'НДС н', 'НДПИ н', 'НДФЛ н',
                                    'Налог на имущество организаций н', 'Налог на прибыль н',
                                    'УСН н', 'Торговый сбор н', 'Транспортный налог н']].sum(axis=1)

# Создадим новую колонку с суммой уплаченных налогов
data_for_analysis['Налоги уплата'] = data_for_analysis[['Акцизы у', 'Водный налог у', 'ЕНВД у', 'ЕСХН у',
                                    'Земельный налог у', 'НДС у', 'НДПИ у', 'НДФЛ у',
                                    'Налог на имущество организаций у', 'Налог на прибыль у',
                                    'УСН у', 'Торговый сбор у', 'Транспортный налог у']].sum(axis=1)

# Создадим новую колонку с суммой неуплаченных страховых взносов
data_for_analysis['Страховые неуплата'] = data_for_analysis.loc[:, 'Страховые взносы мед н':\
                                    'Страховые и другие взносы пенс н'].sum(axis=1)

# Создадим новую колонку с суммой уплаченных страховых взносов
data_for_analysis['Страховые уплата'] = data_for_analysis.loc[:, 'Страховые взносы мед у':\
                                    'Страховые и другие взносы пенс у'].sum(axis=1)

# Создадим новую колонку с суммой неуплаченых иных платежей
data_for_analysis['Иные платежи долг'] = data_for_analysis[['Задолженность и перерасчеты по отмененным налогам н',
                                    'Неналоговые доходы, администрируемые налоговыми органами н',
                                    'Сборы за пользование объектами животного мира и за пользование объектами ВБР н']].sum(axis=1)

# Создадим новую колонку с суммой уплаченых иных платежей
data_for_analysis['Иные платежи уплата'] = data_for_analysis[['Задолженность и перерасчеты по отмененным налогам у',
                                    'Неналоговые доходы, администрируемые налоговыми органами у',
                                    'Сборы за пользование объектами животного мира и за пользование объектами ВБР у',
                                    'Утилизационный сбор у']].sum(axis=1)

# Создадим датафрейм с новыми признаками
data_new_features = copy.deepcopy(data_for_analysis[['ИНН', 'Муниципальное образование', 'Категория',
                                    'Доход', 'Расход', 'Налоги долг', 'Иные платежи долг',
                                    'Налоги уплата', 'Иные платежи уплата', 'Страховые неуплата', 'Страховые уплата']])
data_new_features.head()
```

Out[38]:

	ИНН	Муниципальное образование	Категория	Доход	Расход	Налоги долг	Иные платежи долг	Налоги уплата	Иные платежи уплата	Страховые неуплата
5	2537138078	ГОРОД ВЛАДИВОСТОК	Микропредприятие	10448000.00	10324000.00	91725.74	0.00	36810.26	0.00	0.00
16	2537136507	ГОРОД ВЛАДИВОСТОК	Микропредприятие	106000.00	791000.00	0.00	0.00	0.00	0.00	67.92
21	2537136352	ГОРОД ВЛАДИВОСТОК	Микропредприятие	17592000.00	17410000.00	0.13	0.00	58538.00	0.00	0.00
31	2537135648	ГОРОД ВЛАДИВОСТОК	Микропредприятие	60251000.00	58506000.00	11.71	0.00	487455.00	0.00	273.08
33	2537138053	ГОРОД ВЛАДИВОСТОК	Микропредприятие	0.00	0.00	640333.00	0.00	0.00	0.00	0.00

2 ТЕСТИРОВАНИЕ ГИПОТЕЗ

В данном блоке ответим на следующие вопросы:

1. Отличаются ли муниципальные образования по уровню налоговой платежеспособности у субъектов МСП?
2. В каких муниципальных образованиях наиболее низкий уровень налоговой платежеспособности?

1. Отличаются ли муниципальные образования по уровню налоговой платежеспособности у субъектов МСП?

H0 - Муниципальные образования Приморского края не отличаются по уровню налоговой платежеспособности субъектов МСП.
H1 - Муниципальные образования Приморского края отличаются по уровню налоговой платежеспособности субъектов МСП.

Переведем все числовые данные к целочисленному формату для корректного проведения тестов.

In [39]:
```
# На каждой итерации берем колонку и приводим
# формат числа к целочисленному
for col_name in data_new_features.iloc[:, 3:]:
    data_new_features[col_name] = data_new_features[col_name].astype('int64')
```

In [40]:
```
# Выведем типы данных и отобразим первые 5 строк датафрейма
data_new_features.info()
data_new_features.head()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7397 entries, 5 to 38364
Data columns (total 11 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   ИНН                        7397 non-null   object
 1   Муниципальное образование  7397 non-null   category
 2   Категория                  7397 non-null   category
 3   Доход                      7397 non-null   int64
 4   Расход                     7397 non-null   int64
 5   Налоги долг                7397 non-null   int64
 6   Иные платежи долг          7397 non-null   int64
 7   Налоги уплата              7397 non-null   int64
 8   Иные платежи уплата        7397 non-null   int64
 9   Страховые неуплата         7397 non-null   int64
 10  Страховые уплата           7397 non-null   int64
dtypes: category(2), int64(8), object(1)
memory usage: 594.0+ KB
```

Out[40]:

	ИНН	Муниципальное образование	Категория	Доход	Расход	Налоги долг	Иные платежи долг	Налоги уплата	Иные платежи уплата	Страховые неуплата	Страховые уплата
5	2537138078	ГОРОД ВЛАДИВОСТОК	Микропредприятие	10448000	10324000	91725	0	36810	0	0	12870
16	2537136507	ГОРОД ВЛАДИВОСТОК	Микропредприятие	106000	791000	0	0	0	0	67	147720
21	2537136352	ГОРОД ВЛАДИВОСТОК	Микропредприятие	17592000	17410000	0	0	58538	0	0	435171
31	2537135648	ГОРОД ВЛАДИВОСТОК	Микропредприятие	60251000	58506000	11	0	487455	0	273	363008
33	2537138053	ГОРОД ВЛАДИВОСТОК	Микропредприятие	0	0	640333	0	0	0	0	0

In [41]:
```
# Создадим датафрейм для проведения теста
df_test_mo = copy.deepcopy(data_new_features[['Муниципальное образование', 'Налоги долг']])
# Выведем первые 5 строк датафрейма
df_test_mo.head()
```

Out[41]:

	Муниципальное образование	Налоги долг
5	ГОРОД ВЛАДИВОСТОК	91725
16	ГОРОД ВЛАДИВОСТОК	0
21	ГОРОД ВЛАДИВОСТОК	0
31	ГОРОД ВЛАДИВОСТОК	11
33	ГОРОД ВЛАДИВОСТОК	640333

Проверим распределение на нормальность. Используем тест нормальности Андерсона-Дарлинга, так как он работает с большими выборками.

In [42]:
```
def normal_test(df, column_name, alpha):
    """
        Функция, проверяющая распределение на нормальность

    Аргументы функции:
        df - датафрейм
        column_name - нименование колонки для проверки нормальности распределения
        alpha - уровень значимости

    Возвращаемое значение:
        Функция возвращает статистику критерия и p-value,
        а также принятие или отклонение H0.

    """
    print('Тест нормальности Андерсона-Дарлинга:')

    # Проводим тест
    stat, pval = normaltest(df[column_name])

    # Выводим статистику критерия и p-value
    print('Statistic:', f'{stat:.3f}')
    print('P-Value:', f'{pval:.10f}')

    # Проверяе условие для принятия или отклонения H0
    if pval > alpha:
        print('Принимаем H0 - Данные распределены нормально.')
    if pval < alpha:
        print('Отклоняем H0 - Данные распределены не нормально.')
```

In [43]:
```
normal_test(df_test_mo, 'Налоги долг', 0.05)
```

```
Тест нормальности Андерсона-Дарлинга:
Statistic: 22925.015
P-Value: 0.0000000000
Отклоняем H0 - Данные распределены не нормально.
```

Признак 'Налоги долг' распределен не нормально. Учитывая данный факт для сравнение выборок будем использовать **непараметрический критерий Краскела-Уоллиса.**

Разобъем данные на 34 группы (так как в Приморском крае 34 муниципальных образования) для тестирования и проведем тест Краскела-Уоллиса.

In [44]:
```
def kruskalwallis_test(df, group_name, values, alpha):
    """
        Функция, реализующая проведение
        теста Краскела-Уоллиса

    Аргументы функции:
        df - датафрейм
        group_name - нименование колонки с значениями для группировки
        values - наименование колонки с непрерывной величиной
        alpha - уровень значимости

    Возвращаемое значение:
        Функция возвращает статистику критерия и p-value,
        а также принятие или отклонение H0

    """
    # Создаем пустой словарь, куда поместим группы для тестирования
    groups = {}
    # На каждой итерации цикла берем уникальное название для последующей фильтрации
    for grp in df[group_name].unique():
        # Создаем ключ словаря присвоив ему название группы для тестирования
        # передав значения, полученные в результате фильтрации датафрейма
        # (фильтруем исходный датафрейм по названию ключа, оставив только соответствующме значения)
        groups[grp] = df[values][df[group_name]==grp].values

    # Создаем тестовые группы, оставив только значения словаря
    test_group = groups.values()

    print('Тест Краскела-Уоллиса:')
    # Проводим тест
    H, pval = mstats.kruskalwallis(*test_group)

    # Выводим статистику критерия и p-value
    print('H-statistic:', f'{H:.3f}')
    print('P-Value:', f'{pval:.10f}')

    # Проверяе условие для принятия или отклонения H0
    if pval > alpha:
        print('Принимаем H0 - Между группами нет значительных различий.')
    if pval < alpha:
        print('Отклоняем H0 - Между группами существуют значительные различия.')
```

In [45]:
```
kruskalwallis_test(df_test_mo, 'Муниципальное образование', 'Налоги долг', 0.05 )
```

```
Тест Краскела-Уоллиса:
H-statistic: 101.258
P-Value: 0.0000000074
Отклоняем H0 - Между группами существуют значительные различия.
```

По итогам проведенного теста получены статистически значимые различия между медианными значениями (данный стат критерий проверяет равенство медианных значений в выборках, что так же является плюсом так как средние значения сильно завышены за счет естественных выбросов) по уплате налогов бизнесом в муниципальных образованиях Приморского края. Это потдтверждается критически низким значением p-value. Вероятность совершения ошибки первого рода равна 0.0000000074.

Учитывая, что в данных присутствуют естественные выбросы они могли повлиять на результаты теста. Проверим данный факт. Уберем выбросы (оставим диапазон +- 3 сигмы (не смотря на то, что данное правило применимо к данным, распределенным нормально, используем его для создания диапазона) и повторим тестирование.

In [46]:
```
# Считаем среднее
m = df_test_mo.mean()
# Считаем стандартное отклонение
std = df_test_mo.std()
# Считаем нижнюю границу
lower_bound = int(round(m - (3 * std)))
# Считаем верхнюю границу
upper_bound = int(round(m + (3 * std)))
# Создаем датафрейм без выбросов
mo_remove_outliers = df_test_mo[df_test_mo['Налоги долг'].between(lower_bound,  # Указываем нижнюю гран
ицу
                                                                    upper_bound)] # Указываем верхнюю гра
ницу
print(f'Число записей всего {len(df_test_mo)} запись(ей).')
print(f'Число записей без выбросов составляет {len(mo_remove_outliers)} запись(ей).')
```

```
Число записей всего 7397 запись(ей).
Число записей без выбросов составляет 7380 запись(ей).
```

Сравним статистики после удаления выбросов.

In [47]:
```
print(f'Среднее до: {df_test_mo["Налоги долг"].mean()} / Среднее после: {mo_remove_outliers["Налоги дол
г"].mean()}')
print(f'Медиана до: {df_test_mo["Налоги долг"].median()} / Медиана после: {mo_remove_outliers["Налоги д
олг"].median()}')
print(f'Мода до: {df_test_mo["Налоги долг"].mode()[0]} / Мода после: {mo_remove_outliers["Налоги долг"]
.mode()[0]}')
print(f'Стандартное отклонение до: {df_test_mo["Налоги долг"].std()} / Стандартное отклонение после: {m
o_remove_outliers["Налоги долг"].std()}')
print(f'Дисперсия до: {df_test_mo["Налоги долг"].var()} / Дисперсия после: {mo_remove_outliers["Налоги
 долг"].var()}')
```

```
Среднее до: 173309.4472083277 / Среднее после: 39963.469783197834
Медиана до: 18.0 / Медиана после: 17.5
Мода до: 0 / Мода после: 0
Стандартное отклонение до: 4141243.9524956928 / Стандартное отклонение после: 364253.812159629
Дисперсия до: 17149901474082.148 / Дисперсия после: 132680839672.82228
```

После удаления выбросов значительно изменилось среднее. Это обусловленно удалением из даных экстемально высоких значений касательно неуплаты налогов, соответственно снизился разброс данных (стандартное отклонение и дисперсия). Медиана почти не изменилась, мода осталась неизменной. Так как критерий Краскела-Уоллиса проверяет равенство медианных значений в выборках удаление выбросов сильно не сказалось на изменении мер центральной тенденции, на основании которых проводится расчет, поэтому можно провести повторный тест по датафрейму без выбросов.

Проверим полученное распределение на нормальность.</p

In [48]:
```
normal_test(mo_remove_outliers, 'Налоги долг', 0.05)
```

```
Тест нормальности Андерсона-Дарлинга:
Statistic: 14693.809
P-Value: 0.0000000000
Отклоняем H0 - Данные распределены не нормально.
```

Снова разобъем данные на 34 группы и проведем тестирование.

H0 - Группы не отличаются.
H1 - Группы имеют существенные отличия.

In [49]:
```
kruskalwallis_test(mo_remove_outliers, 'Муниципальное образование', 'Налоги долг', 0.05 )
```

```
Тест Краскела-Уоллиса:
H-statistic: 103.747
P-Value: 0.0000000031
Отклоняем H0 - Между группами существуют значительные различия.
```

Проведя два теста по данным с выбросами и по данным, взятым в диапазоне +- 3 сигма установлено, что при частичном удалении выбросов вероятность получить статистически значимые различия в группах возрастает, но учитывая низкую вероятности совершения ошибки первого рода проводя тестирование на данных с выбросами, отчистка данных от выбросов в данном анализе не повлияла не результат.

С выбросами:

H-statistic: 101.258

P-Value: 0.0000000074

+ - 3 сигмы:

H-statistic: 103.747

P-Value: 0.0000000031

Вывод:

На основании проведенного теста можно сделать вывод, что все муниципальные образования Приморского края отличаются по уровню налоговой платежеспособности субъектов малого и среднего предпринимательства. Это подтверждается низким уровне p-value, полученным в ходе тестирования. Вероятность совершения ошибки первого рода равна 0.0000000074.

2. В каких муниципальных образованиях наиболее низкий уровень налоговой платежеспособности?

Для того, чтобы ответить на данный вопрос, а также используя полученную информацию по результатам предидущего теста (установлены различия между муниципалитетами по уровню неуплаты налогов бизнесом) объединим муниципалитеты в следующие группы:

1. **low** - медианная неуплата налога на одного предпринимателя меньше или равна 500 рублей.
2. **medium** - медианная неуплата налога на одного предпринимателя больше 500 рублей и меньше или равна 1500 рублей.
3. **high** - медианная неуплата налога на одного предпринимателя больше 1500 рублей.

Создадим датафрейм с наименованием муниципального образования и медианным значением неуплаты налога в данном муниципалитете.

In [50]:
```
# Группируем данные по мун образованиям, агрегируем данные по недоимке
mo_median_pay = df_test_mo.groupby('Муниципальное образование').agg({'Налоги долг': ['median', 'count'
]}).reset_index()
# Формируем новый датафрейм объединив информацию по мун образованиям и недоимке
mo_merge = mo_median_pay['Муниципальное образование'].reset_index().merge(mo_median_pay['Налоги долг'].
reset_index())
# Удаляем столбец с индексом
del mo_merge['index']
# приводим данные по недоимке к целочисленному формату
mo_merge['median'] = mo_merge['median'].astype('int64')
# Сортируем данные по недоимке по убыванию
mo_merge.sort_values('median', ascending=False)
```

Out[50]:

	Муниципальное образование	median	count
17	РАЙОН ЛАЗОВСКИЙ	8787	8
12	РАЙОН АНУЧИНСКИЙ	3686	3
32	РАЙОН ШКОТОВСКИЙ	1510	16
9	ГОРОД СПАССК-ДАЛЬНИЙ	1462	30
24	РАЙОН ПОЖАРСКИЙ	1000	31
14	РАЙОН КАВАЛЕРОВСКИЙ	943	20
22	РАЙОН ПАРТИЗАНСКИЙ	907	22
13	РАЙОН ДАЛЬНЕРЕЧЕНСКИЙ	710	8
25	РАЙОН СПАССКИЙ	588	9
23	РАЙОН ПОГРАНИЧНЫЙ	501	10
21	РАЙОН ОЛЬГИНСКИЙ	469	10
11	ГОРОД ФОКИНО	383	21
6	ГОРОД ЛЕСОЗАВОДСК	249	30
29	РАЙОН ХОРОЛЬСКИЙ	244	11
2	ГОРОД БОЛЬШОЙ КАМЕНЬ	139	34
18	РАЙОН МИХАЙЛОВСКИЙ	135	12
8	ГОРОД ПАРТИЗАНСК	131	27
15	РАЙОН КИРОВСКИЙ	106	7
7	ГОРОД НАХОДКА	84	346
5	ГОРОД ДАЛЬНЕРЕЧЕНСК	68	38
28	РАЙОН ХАСАНСКИЙ	36	53
1	ГОРОД АРТЕМ	26	326
10	ГОРОД УССУРИЙСК	20	300
31	РАЙОН ЧУГУЕВСКИЙ	16	12
3	ГОРОД ВЛАДИВОСТОК	15	5760
0	ГОРОД АРСЕНЬЕВ	7	28
19	РАЙОН НАДЕЖДИНСКИЙ	7	79
16	РАЙОН КРАСНОАРМЕЙСКИЙ	6	17
4	ГОРОД ДАЛЬНЕГОРСК	3	73
26	РАЙОН ТЕРНЕЙСКИЙ	1	13
27	РАЙОН ХАНКАЙСКИЙ	0	8
20	РАЙОН ОКТЯБРЬСКИЙ	0	20
30	РАЙОН ЧЕРНИГОВСКИЙ	0	10
33	РАЙОН ЯКОВЛЕВСКИЙ	0	5

Присвоим каждому муниципальному образованию принадлежность к одной из трех групп.

In [51]:
```
mo_merge.loc[mo_merge['median'] <= 500, 'Категория'] = 'low'
mo_merge.loc[(mo_merge['median'] > 500) & (mo_merge['median'] <= 1500) , 'Категория'] = 'medium'
mo_merge.loc[mo_merge['median'] > 1500, 'Категория'] = 'high'
```

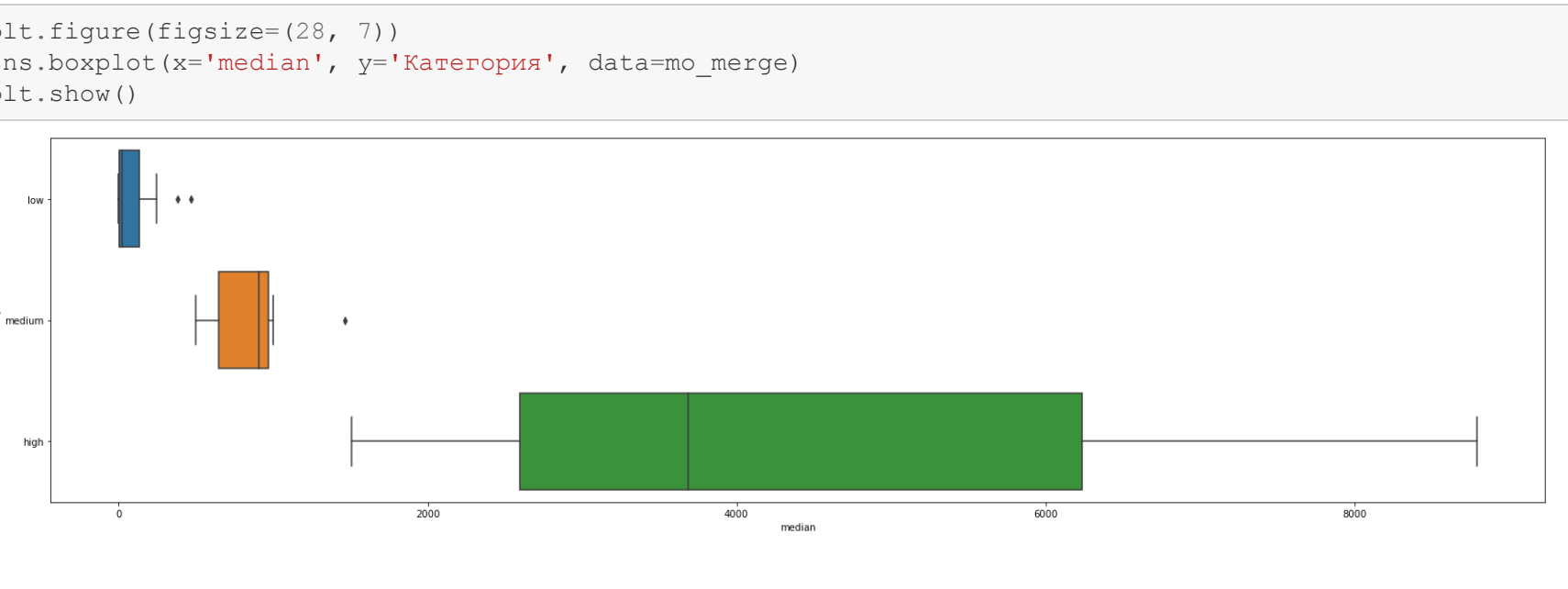
In [52]:
```
mo_merge.sort_values('median', ascending=False)
```

Out[52]:

	Муниципальное образование	median	count	Категория
17	РАЙОН ЛАЗОВСКИЙ	8787	8	high
12	РАЙОН АНУЧИНСКИЙ	3686	3	high
32	РАЙОН ШКОТОВСКИЙ	1510	16	high
9	ГОРОД СПАССК-ДАЛЬНИЙ	1462	30	medium
24	РАЙОН ПОЖАРСКИЙ	1000	31	medium
14	РАЙОН КАВАЛЕРОВСКИЙ	943	20	medium
22	РАЙОН ПАРТИЗАНСКИЙ	907	22	medium
13	РАЙОН ДАЛЬНЕРЕЧЕНСКИЙ	710	8	medium
25	РАЙОН СПАССКИЙ	588	9	medium
23	РАЙОН ПОГРАНИЧНЫЙ	501	10	medium
21	РАЙОН ОЛЬГИНСКИЙ	469	10	low
11	ГОРОД ФОКИНО	383	21	low
6	ГОРОД ЛЕСОЗАВОДСК	249	30	low
29	РАЙОН ХОРОЛЬСКИЙ	244	11	low
2	ГОРОД БОЛЬШОЙ КАМЕНЬ	139	34	low
18	РАЙОН МИХАЙЛОВСКИЙ	135	12	low
8	ГОРОД ПАРТИЗАНСК	131	27	low
15	РАЙОН КИРОВСКИЙ	106	7	low
7	ГОРОД НАХОДКА	84	346	low
5	ГОРОД ДАЛЬНЕРЕЧЕНСК	68	38	low
28	РАЙОН ХАСАНСКИЙ	36	53	low
1	ГОРОД АРТЕМ	26	326	low
10	ГОРОД УССУРИЙСК	20	300	low
31	РАЙОН ЧУГУЕВСКИЙ	16	12	low
3	ГОРОД ВЛАДИВОСТОК	15	5760	low
0	ГОРОД АРСЕНЬЕВ	7	28	low
19	РАЙОН НАДЕЖДИНСКИЙ	7	79	low
16	РАЙОН КРАСНОАРМЕЙСКИЙ	6	17	low
4	ГОРОД ДАЛЬНЕГОРСК	3	73	low
26	РАЙОН ТЕРНЕЙСКИЙ	1	13	low
27	РАЙОН ХАНКАЙСКИЙ	0	8	low
20	РАЙОН ОКТЯБРЬСКИЙ	0	20	low
30	РАЙОН ЧЕРНИГОВСКИЙ	0	10	low
33	РАЙОН ЯКОВЛЕВСКИЙ	0	5	low

Посмотрим на распределения в группах.

In [53]:
```
plt.figure(figsize=(28, 7))
sns.boxplot(x='median', y='Категория', data=mo_merge)
plt.show()
```



Визуально из построенных боксплотов видно отличие групп. Подтвердим это проведя тест. Сформулируем нулевую и альтернативную гипотезы.

Разобъем данные на 3 группы и проведем тестирование.

H0 - Группы не отличаются.
H1 - Группы имеют существенные отличия.

In [54]:
```
kruskalwallis_test(mo_merge, 'Категория', 'median', 0.05)
```

```
Тест Краскела-Уоллиса:
H-statistic: 21.136
P-Value: 0.0000257215
Отклоняем H0 - Между группами существуют значительные различия.
```

Вывод:

На основании проведенного теста можно сделать вывод, что между группами low, medium и hight существуют различия по уровню налоговой платежеспособности субъектов малого и среднего предпринимательства. Это подтверждается низким уровнем p-value, полученным в ходе тестирования. Вероятность совершения ошибки первого рода равна 0.0000257215.

Проведем еще одно тестирование, чтобы понять между какими конкретно группами существуют различия (возможно не все группы отличаются между собой). Воспользуемся непараметрическим апостериорным критерием Данна и сравним группы попарно. Кроме того в целях минимизации получить ошибку первого рода применим поправку Бонферрони и Холмса.

Сравнение групп **high**, **medium** и **low**.

In [58]:
```
pd.set_option('display.float_format', lambda x: '%.6f' % x)
```

In [59]:
```
print('Тест данна с поправкой Бонферрони:')
sp.posthoc_dunn(mo_merge, val_col='median', group_col='Категория', p_adjust = 'bonferroni')
```

```
Тест данна с поправкой Бонферрони:
```

Out[59]:

	high	low	medium
high	1.000000	0.002300	1.000000
low	0.002300	1.000000	0.000862
medium	1.000000	0.000862	1.000000

In [60]:
```
print('Тест данна с поправкой Холмса:')
sp.posthoc_dunn(mo_merge, val_col='median', group_col='Категория', p_adjust = 'holm')
```

```
Тест данна с поправкой Холмса:
```

Out[60]:

	high	low	medium
high	1.000000	0.001534	0.466479
low	0.001534	1.000000	0.000862
medium	0.466479	0.000862	1.000000

Вывод:

По результатам теста Данна полученные следующие результаты:

1. Скорректированное значение p-value для разницы между группами **high** и **low** составляет:
Бонферрони - 0.002300
Холмс - 0.001534

2. Скорректированное значение p-value для разницы между группами **high** и **medium** составляет:
Бонферрони - 1.000000
Холмс - 0.466479

3. Скорректированное значение p-value для разницы между группами **low** и **medium** составляет:
Бонферрони - 0.000862
Холмс - 0.000862

Таким образом, группы, которые статистически значимо отличаются при α = 0,05, - это группы **high и low** и группы **medium и low**

Результаты проведенных тестов показали, что между муниципальными образованиями существуют группы, отличающиеся по уровню налоговой платежеспособности субъектов малого и среднего предпринимательства.

3 ИТОГОВЫЙ ВЫВОД ПО РЕЗУЛЬТАТАМ ПРОВЕДЕННОГО АНАЛИЗА

По результатам проведенного анализа удалось ответить на следующие вопросы:

1. Отличаются ли муниципальные образования по уровню налоговой платежеспособности у субъектов МСП?
2. В каких муниципальных образованиях наиболее низкий уровень налоговой платежеспособности?

В ходе проведенного анализа установлено, что между муниципальными образованиями Приморского края существуют различия по уровню неуплаты налогов субъектами малого и среднего предпринимательства. В целях установления групп муниципалитетов по уровню неуплаты налогов бизнесом муниципальные образования были объединены в группы по медианному уровню неуплаты налогов на одного предпринимателя. В результате установлено:

1. В Лазовском, Анучинском, Шкотовском, Пожарском, Кавалеровском, Партизанском, Дальнереченском, Спасском, Пограничном районах, а также в городе Спасс-Дальний наблюдается самая высокая недобросовестность бизнеса по уплате налоговых платежей. Данные муниципальтеты входят в группу high и medium. Медианная неуплата налога на одного предпринимателя в группе high составляет 3 686 рублей. Средняя неуплата налога на одного предпринимателя в группе medium составляет 907 рублей;

2. В Ольгинском, Хорольском, Михайловском, Кировском, Хасанском, Чугуевском, Надеждинском, Красноармейском, Тернейском, Ханкайском, Октябрьском, Черниговском, Яковлевском, а также в городах Фокино, Лесозаводске, Большом Камне, Партизанске, Находке, Дальнереченске, Артеме, Уссурийске, Владивостоке, Арсеньеве, Дальнегорске, работает самый добросовестный бизнес. Средняя неуплата налога на одного предпринимателя в данной группе муниципалитетов составляет 23 рубля.

Таким образом, в муниципальных образованиях Приморского края входящих в группу high и medium осуществляет деятельность бизнес, либо плохо знающий налоговое законадательство, либо бизнес, целенаправлено не платящий налоги. При формировании годового плана обучающих мероприятий бизнеса налоговой грамотности стоит обратить внимание на данные категории муниципалитетов. Кроме того, в целях повышения уровня поступления налогов в консалидированный бюджет Приморского края, необходимо усилить налоговый контроль на данных территориях.

Сформируем датасет для последующей визуализации в powe bi.

In [61]:
```
mo_merge.to_csv('data_for_powebi.csv', encoding = 'utf-8')
```