Министерство цифрового развития

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Кафедра прикладной математики и кибернетики

Отчёт

по лабораторной работе № 1 «Первичный анализ и предобработка данных»

Выполнил:

студент группы ИП-212\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Глинский Вадим Юрьевич

ФИО студента

Работу проверил: Старший преподаватель

должность преподавателя Дементьева Кристина Игоревна

ФИО преподавателя

Новосибирск 2025 г.

# **Введение (задание )**

Цель работы: Освоить на практике основные методы первичного анализа данных (EDA), их визуализации и предобработки с использованием библиотек Python.

Для выполнения лабораторной работы был выбран набор данных Car Features and MSRP, содержащий данные о автомобилях, их характеристиках и их цене (https://www.kaggle.com/datasets/CooperUnion/cardataset)

# **Основная часть**

1. Загрузка и первичный осмотр:

import pandas as pd

path = "/content/drive/MyDrive/4mmo/data.csv"

df = pd.read\_csv(path)

df.info()

df.describe()

df.shape

Набор данных имеет 11914 записей, 16 признаков (Make, Model, Year, Engine Fuel Type, Engine HP, Engine Cylinders, Transmission Type, Driven\_Wheels, Number of Doors, Market Category, Vehicle Size, Vehicle Style, highway MPG, city mpg, Popularity, MSRP) (8 числовых и 8 типа object). В 5 признаках имеются пропуски.

2. Анализ пропусков:

missing = df.isna().sum()

print(missing)

missing\_percent = (df.isna().sum()/df.shape[0]).round(4)\*100

print(missing\_percent)

import seaborn as sns

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

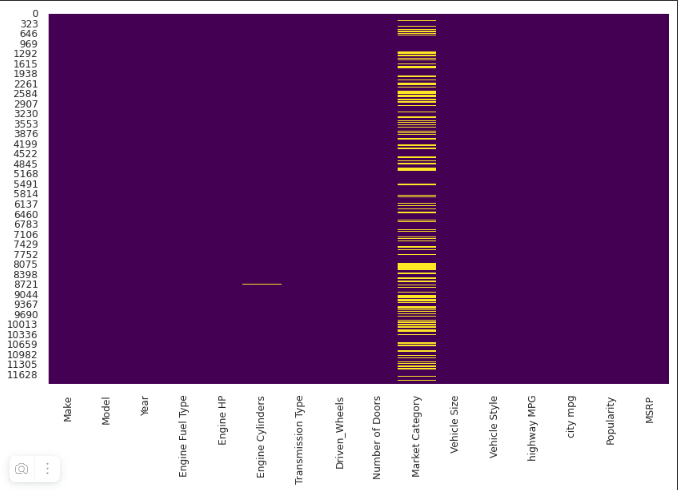
sns.set(style="dark", font\_scale=0.8)

sns.heatmap(df.isna(), cbar=False, cmap="viridis")

Пропуски в столбцах:

Engine Fuel Type – 3(0.03%), Engine HP - 69(0.58%), Engine Cylinders - 30(0.25%), Number of Doors - 6(0.05%), Market Category - 3742(31.41%)

Матрица пропусков при помощи heatmap



3. Анализ числовых признаков:

Постройка гистограмм и boxplot’ов для числовых столбцов

import numpy as np

numeric\_cols = df.select\_dtypes(include=[np.number]).columns.tolist()

fig, axes = plt.subplots(nrows=len(numeric\_cols), ncols=2, figsize=(20, 7\*len(numeric\_cols)))

for i, col in enumerate(numeric\_cols):

axes[i, 0].hist(df[col].dropna(), bins=30, alpha=0.7, edgecolor='black')

axes[i, 0].set\_title(f'Гистограмма: {col}')

axes[i, 0].set\_xlabel(col)

axes[i, 0].set\_ylabel('Частота')

axes[i, 0].grid(True, alpha=0.3)

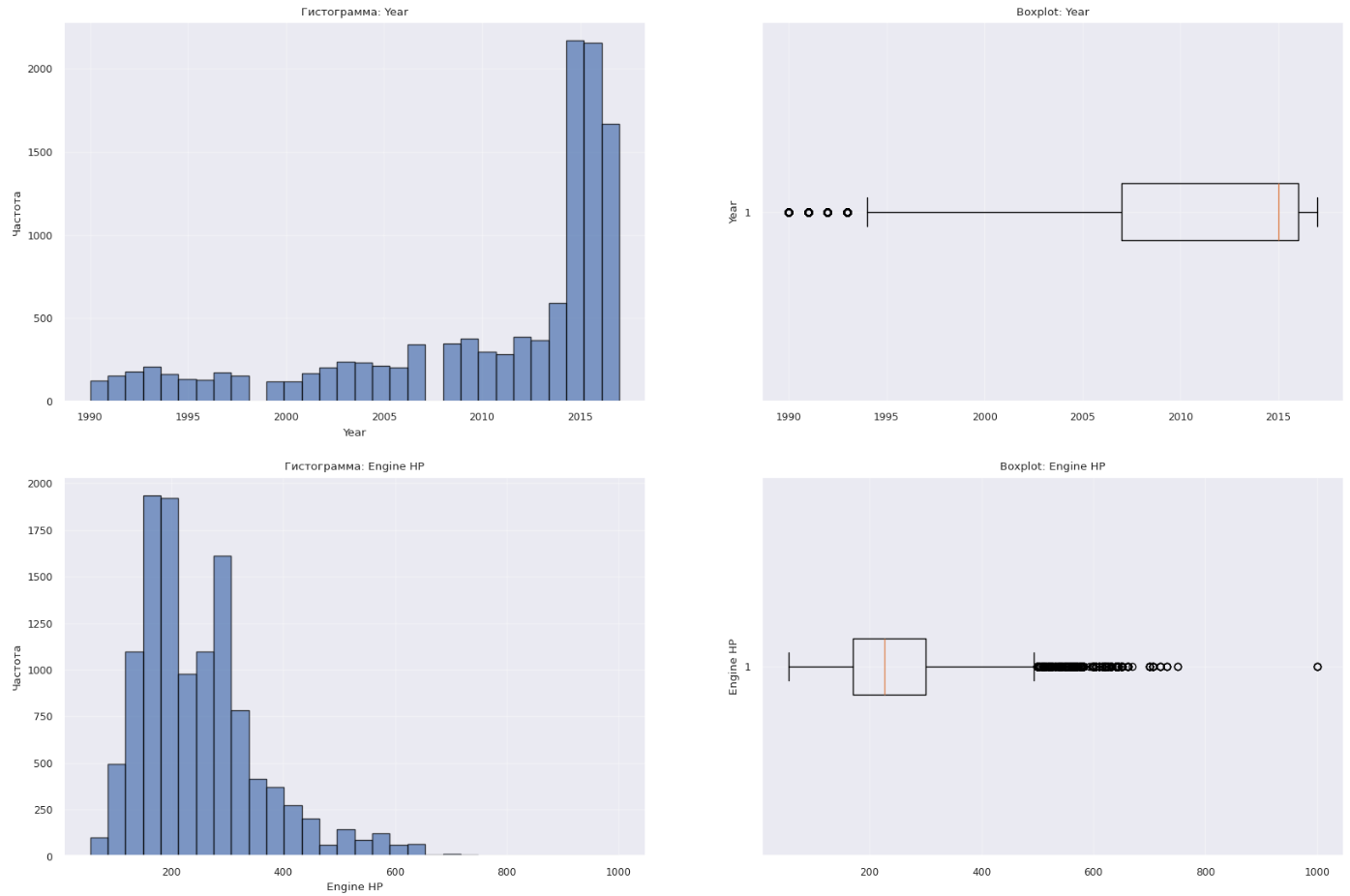
axes[i, 1].boxplot(df[col].dropna(), vert=False)

axes[i, 1].set\_title(f'Boxplot: {col}')

axes[i, 1].set\_ylabel(col)

axes[i, 1].grid(True, alpha=0.3)

plt.show()



Расчёт стандартной статистики (среднее, медиана, стандартное отклонение, асимметрия) для ключевых числовых признаков.

from scipy import stats

stats\_df = pd.DataFrame(index=numeric\_cols)

for col in numeric\_cols:

data = df[col].dropna()

stats\_df.loc[col, 'Среднее'] = np.mean(data)

stats\_df.loc[col, 'Медиана'] = np.median(data)

stats\_df.loc[col, 'Стандартное отклонение'] = np.std(data)

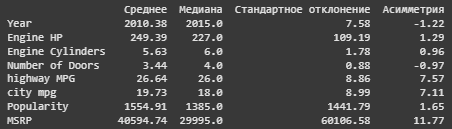
stats\_df.loc[col, 'Асимметрия'] = stats.skew(data)

formatted\_stats = stats\_df.copy()

for col in ['Среднее', 'Медиана', 'Стандартное отклонение', 'Асимметрия']:

formatted\_stats[col] = formatted\_stats[col].round(2)

print(formatted\_stats.to\_string())



При помощи boxplot’ов можно увидеть наличие выбросов, но для данного набора данных выбросы показывают разные категории авто, например, выбросы в цене появляются из-за наличия в наборе дорогих авто (Bugatti Veyron 16.4 – 2065902$) и их малого количества, по сравнению с автомобилями среднего ценового диапазона

4. Анализ категориальных признаков:

Столбчатые диаграммы для категориальных столбцов

categorical\_columns = df.select\_dtypes(include=['object']).columns.tolist()

cc = categorical\_columns

cc.remove('Model')

fig, axes = plt.subplots(nrows=len(cc), figsize=(10, 5\*len(cc)))

for i, col in enumerate(cc):

sns.countplot(x=col, data=df, ax=axes[i])

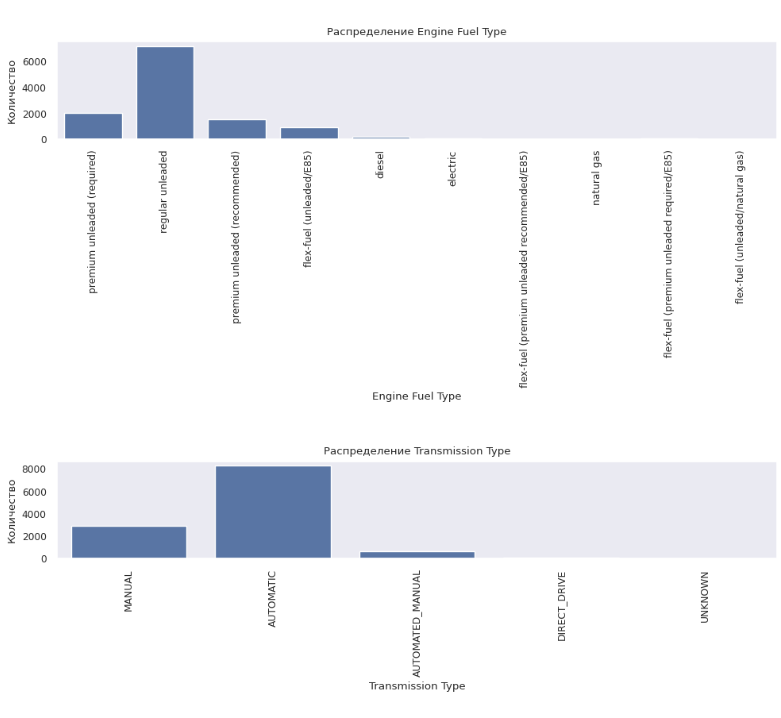
axes[i].set\_title(f'Распределение {col}')

axes[i].tick\_params(axis='x', rotation=90)

axes[i].set\_ylabel('Количество')

plt.tight\_layout()

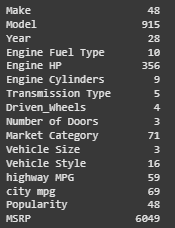
plt.show()



Количество уникальных категорий в каждом признаке

unique\_counts = df.nunique()

print(unique\_counts)



Высокое количество уникальных значений в признаках модели авто, мощности двигателя, цене автомобиля

5. Анализ взаимосвязей:

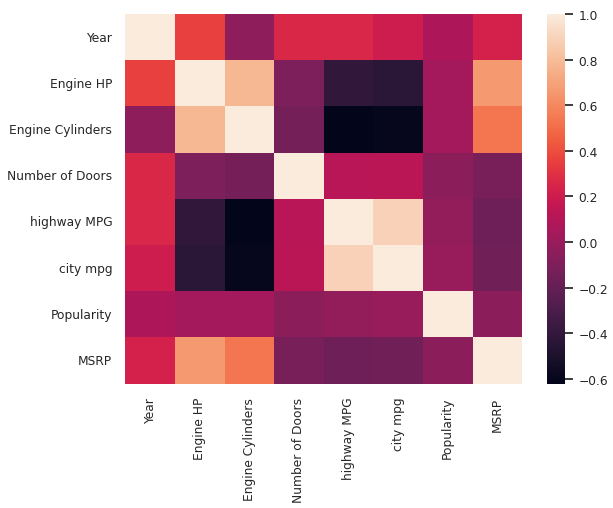
Матрица корреляций для числовых признаков с тепловой картой

numeric\_df = df.select\_dtypes(include=[np.number])

correlation\_matrix = numeric\_df.corr()

print(correlation\_matrix)

sns.heatmap(correlation\_matrix)



Взаимосвязь категориальных и числовых признаков при помощи boxplot’ов

fig, axes = plt.subplots(nrows = len(cc)\*4, ncols = 2, figsize=(20, 7\*len(cc)\*4))

axes = axes.reshape(len(cc)\*4, 2)

for i, category in enumerate(cc):

for j, num in enumerate(numeric\_cols):

row = i \* 4 + j // 2

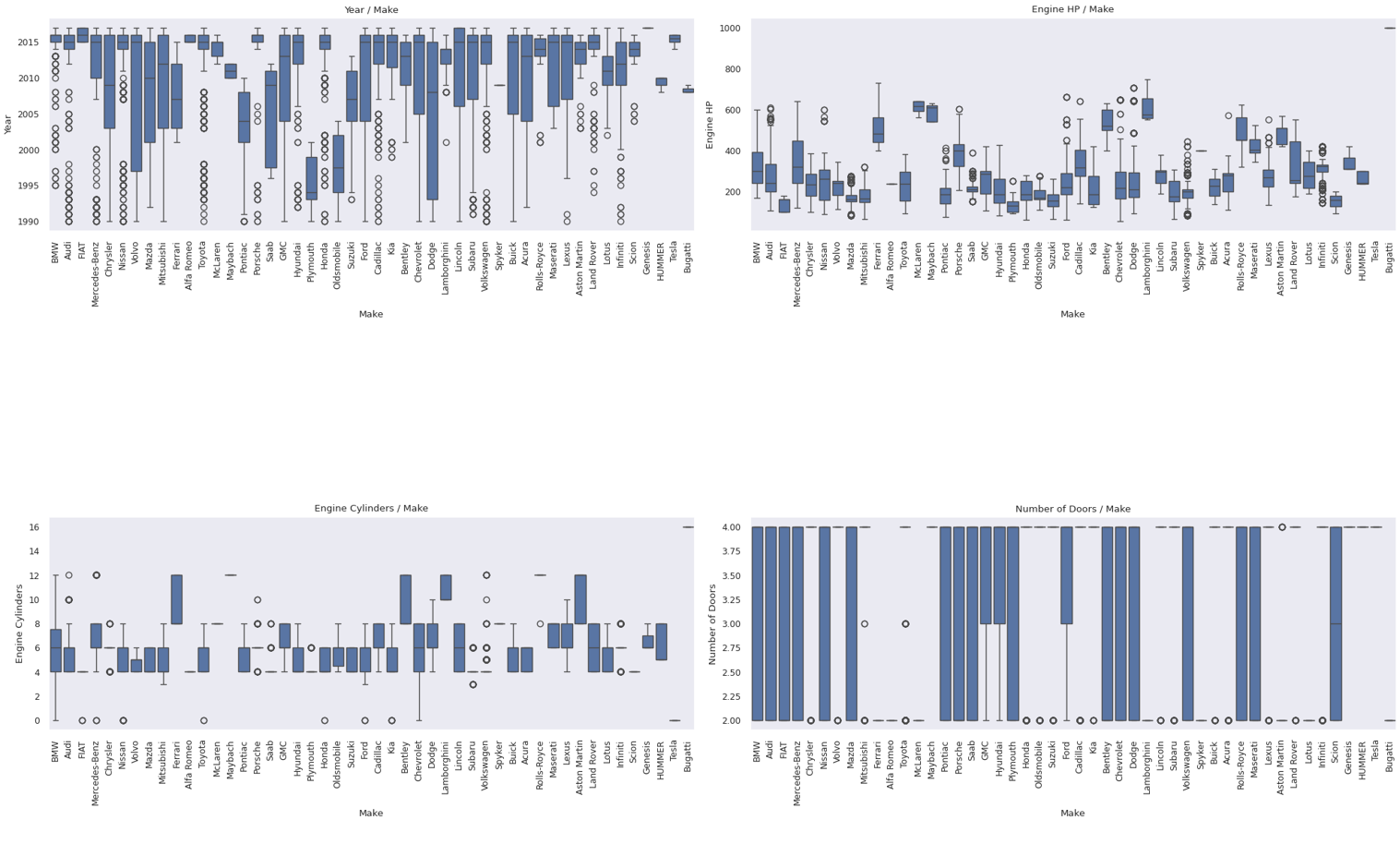
col = j % 2

sns.boxplot(x=category, y=num, data=df, ax=axes[row][col])

axes[row][col].set\_title(f'{num} / {category}')

axes[row][col].tick\_params(axis='x', rotation=90)

plt.tight\_layout()



6. Базовая предобработка:

Изменение формата названия столбцов (нижний регистр, замена пробелов на \_)

df.columns = df.columns.str.lower().str.replace(' ', '\_')

print(df.columns.to\_list())

Обработка пропусков

df['engine\_fuel\_type'] = df['engine\_fuel\_type'].fillna(df['engine\_fuel\_type'].mode()[0])

df['engine\_hp'] = df.groupby(['make', 'model', 'year'])['engine\_hp'].transform(

lambda x: x.fillna(x.median() if not pd.isna(x.median()) else df['engine\_hp'].median()))

df['engine\_cylinders'] = df.groupby(['make', 'engine\_hp', 'msrp'])['engine\_cylinders'].transform(

lambda x: x.fillna(x.median() if not pd.isna(x.median()) else df['engine\_cylinders'].median()))

df['number\_of\_doors'] = df.groupby(['vehicle\_style', 'vehicle\_size', 'year'])['number\_of\_doors'].transform(

lambda x: x.fillna(x.median() if not pd.isna(x.median()) else df['number\_of\_doors'].median()))

df['market\_category'] = df['market\_category'].fillna('unknown')

print("Пропуски после обработки:")

print(df.isnull().sum())

Пропуски заполнены:

Тип топлива - при помощи моды

Мощность двигателя - медианой по производителю, модели и году выпуска

Количество цилиндров – медианой по производителю, мощности и цене

Количество дверей – медианой по типу авто, размеру авто, году выпуска

Рыночная категория – создана отдельная категория «unknown»

7. Обработка выбросов

Для обработки выбросов была выбрана цена авто, так как этот признак имел наибольшие выбросы

Q1 = df['msrp'].quantile(0.25)

Q3 = df['msrp'].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

print(f"Q1 (25-й перцентиль): {Q1:.2f}")

print(f"Q3 (75-й перцентиль): {Q3:.2f}")

print(f"IQR: {IQR:.2f}")

print(f"Нижняя граница: {lower\_bound:.2f}")

print(f"Верхняя граница: {upper\_bound:.2f}")

df\_processed = df.copy()

df\_processed['msrp\_processed'] = df['msrp'].clip(lower=lower\_bound, upper=upper\_bound)

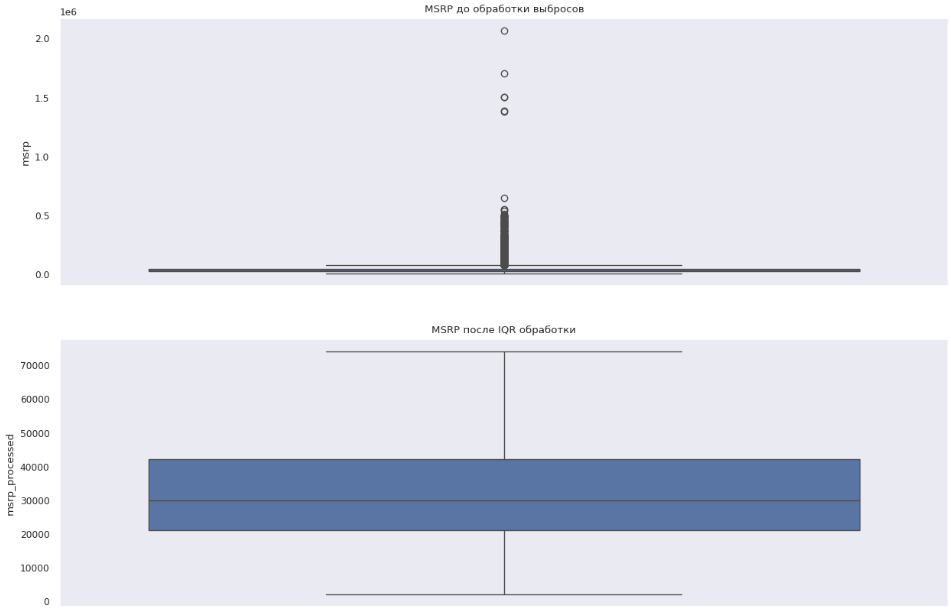
fig, axes = plt.subplots(2, figsize=(15, 10))

sns.boxplot(y=df['msrp'], ax=axes[0])

axes[0].set\_title('MSRP до обработки выбросов')

sns.boxplot(y=df\_processed['msrp\_processed'], ax=axes[1])

axes[1].set\_title('MSRP после IQR обработки')



Для данного признака оптимальным вариантом было отсутствие обработки выбросов, так как отсекаются различные «эксклюзивные» авто с высокой ценой

# **Заключение**

Общие выводы по проведенному анализу:

* 1. Какие основные проблемы данных были выявлены?
  2. Какие методы предобработки были применены и почему?
  3. Как проведенная работа повлияет на дальнейшее построение моделей?

1. К основным проблемам набора данных относится большое количество пропусков в некоторых столбцах
2. Методы предобработки, которые были применены – это заполнение пропусков (для каждого из признаков был выбран оптимальный вариант заполнения), обработка выбросов
3. Проведенная работа может сильно повлиять на дальнейшее построение моделей, так как большое количество автомобилей было отброшено при обработке выбросов, из-за чего у модели не будет информации о дорогих авто. Так же пропуски могли быть заполнены неправильными данными, из-за чего у модели может быть неправильное представление о некоторых автомобилях

# **Код программы**

import pandas as pd

path = "/content/drive/MyDrive/4mmo/data.csv"

df = pd.read\_csv(path)

df.info()

df.describe()

df.shape

missing = df.isna().sum()

print(missing)

missing\_percent = (df.isna().sum()/df.shape[0]).round(4)\*100

print(missing\_percent)

import seaborn as sns

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

sns.set(style="dark", font\_scale=0.8)

sns.heatmap(df.isna(), cbar=False, cmap="viridis")

import numpy as np

numeric\_cols = df.select\_dtypes(include=[np.number]).columns.tolist()

fig, axes = plt.subplots(nrows=len(numeric\_cols), ncols=2, figsize=(20, 7\*len(numeric\_cols)))

for i, col in enumerate(numeric\_cols):

axes[i, 0].hist(df[col].dropna(), bins=30, alpha=0.7, edgecolor='black')

axes[i, 0].set\_title(f'Гистограмма: {col}')

axes[i, 0].set\_xlabel(col)

axes[i, 0].set\_ylabel('Частота')

axes[i, 0].grid(True, alpha=0.3)

axes[i, 1].boxplot(df[col].dropna(), vert=False)

axes[i, 1].set\_title(f'Boxplot: {col}')

axes[i, 1].set\_ylabel(col)

axes[i, 1].grid(True, alpha=0.3)

plt.show()

from scipy import stats

stats\_df = pd.DataFrame(index=numeric\_cols)

for col in numeric\_cols:

data = df[col].dropna()

stats\_df.loc[col, 'Среднее'] = np.mean(data)

stats\_df.loc[col, 'Медиана'] = np.median(data)

stats\_df.loc[col, 'Стандартное отклонение'] = np.std(data)

stats\_df.loc[col, 'Асимметрия'] = stats.skew(data)

formatted\_stats = stats\_df.copy()

for col in ['Среднее', 'Медиана', 'Стандартное отклонение', 'Асимметрия']:

formatted\_stats[col] = formatted\_stats[col].round(2)

print(formatted\_stats.to\_string())

categorical\_columns = df.select\_dtypes(include=['object']).columns.tolist()

cc = categorical\_columns

cc.remove('Model')

fig, axes = plt.subplots(nrows=len(cc), figsize=(10, 5\*len(cc)))

for i, col in enumerate(cc):

sns.countplot(x=col, data=df, ax=axes[i])

axes[i].set\_title(f'Распределение {col}')

axes[i].tick\_params(axis='x', rotation=90)

axes[i].set\_ylabel('Количество')

plt.tight\_layout()

plt.show()

unique\_counts = df.nunique()

print(unique\_counts)

numeric\_df = df.select\_dtypes(include=[np.number])

correlation\_matrix = numeric\_df.corr()

print(correlation\_matrix)

sns.heatmap(correlation\_matrix)

fig, axes = plt.subplots(nrows = len(cc)\*4, ncols = 2, figsize=(20, 7\*len(cc)\*4))

axes = axes.reshape(len(cc)\*4, 2)

for i, category in enumerate(cc):

for j, num in enumerate(numeric\_cols):

row = i \* 4 + j // 2

col = j % 2

sns.boxplot(x=category, y=num, data=df, ax=axes[row][col])

axes[row][col].set\_title(f'{num} / {category}')

axes[row][col].tick\_params(axis='x', rotation=90)

plt.tight\_layout()

df.columns = df.columns.str.lower().str.replace(' ', '\_')

print(df.columns.to\_list())

Q1 = df['msrp'].quantile(0.25)

Q3 = df['msrp'].quantile(0.75)

IQR = Q3 - Q1

lower\_bound = Q1 - 1.5 \* IQR

upper\_bound = Q3 + 1.5 \* IQR

print(f"Q1 (25-й перцентиль): {Q1:.2f}")

print(f"Q3 (75-й перцентиль): {Q3:.2f}")

print(f"IQR: {IQR:.2f}")

print(f"Нижняя граница: {lower\_bound:.2f}")

print(f"Верхняя граница: {upper\_bound:.2f}")

df\_processed = df.copy()

df\_processed['msrp\_processed'] = df['msrp'].clip(lower=lower\_bound, upper=upper\_bound)

fig, axes = plt.subplots(2, figsize=(15, 10))

sns.boxplot(y=df['msrp'], ax=axes[0])

axes[0].set\_title('MSRP до обработки выбросов')

sns.boxplot(y=df\_processed['msrp\_processed'], ax=axes[1])

axes[1].set\_title('MSRP после IQR обработки')