

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
Национальный исследовательский технологический университет «МИСИС»
Институт информационных технологий и автоматизированных систем управления
Кафедра Бизнес-информатики и систем управления производством

Практическая работа №3

по дисциплине «Статистические методы анализа данных в принятии решений»
на тему «Подготовка и предварительный анализ данных»

Направление подготовки
38.03.05 Бизнес-информатика
Семестр 4

Выполнил:

Сычиков Владимир Андреевич

(ФИО студента)

ББИ-23-6

(№ группы)

09.03.2025

(дата сдачи)

Подпись:

Проверил:

(ФИО преподавателя)

(оценка)

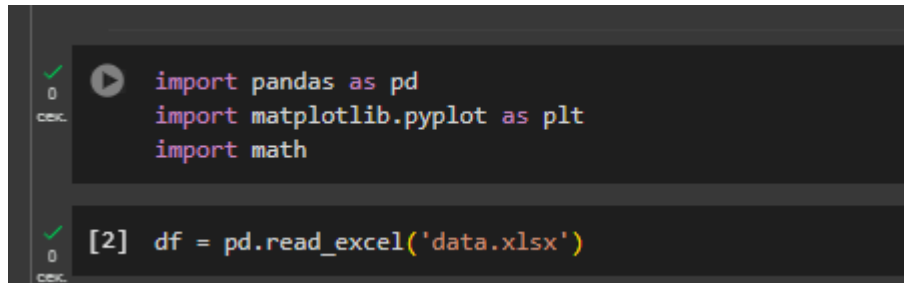
(дата проверки)

Подпись:

Москва – 2025

Ход работы:

Для начала работы я импортировал все необходимые библиотеки(pandas, matplotlib, math) с помощью команды import, а также задал элиасы для удобства обращения к библиотеке. Далее я подгрузил выборку и записал ее в датафрейм df.

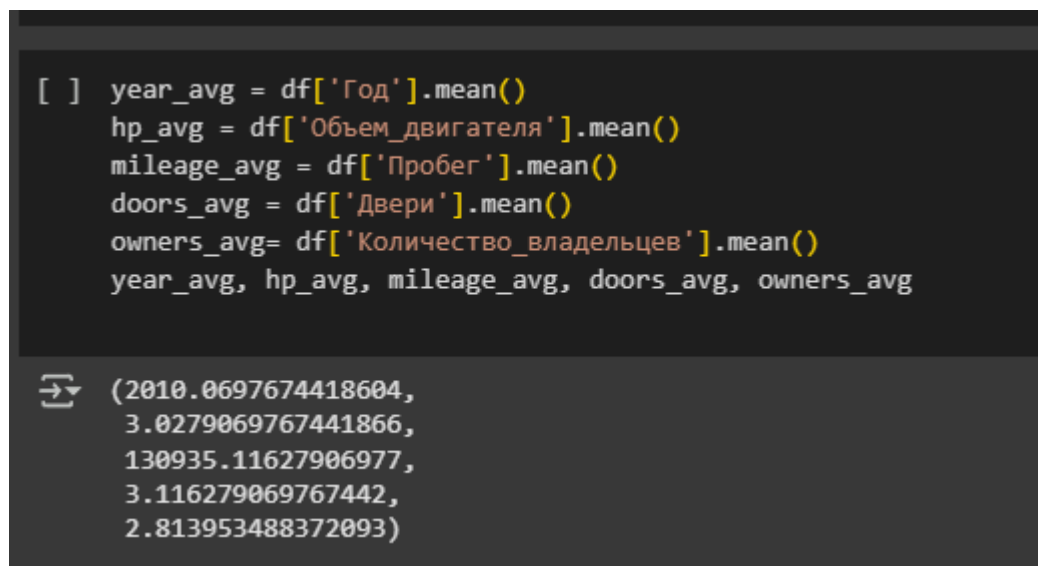


```
import pandas as pd
import matplotlib.pyplot as plt
import math

[2] df = pd.read_excel('data.xlsx')
```

Рисунок 1 - Импорт библиотек

На рисунке 2 реализуется процесс вычисления средних значений для нескольких столбцов из DataFrame. Код вычисляет средние значения для столбцов 'Год', 'Объем_двигателя', 'Пробег', 'Двери' и 'Количество_владельцев'. Для каждого столбца используется функция mean(), которая возвращает среднее арифметическое всех значений в столбце. Результаты вычислений сохраняются в переменные year_avg, hp_avg, mileage_avg, doors_avg и owners_avg. В конце кода эти переменные выводятся, показывая средние значения для каждого из столбцов.



```
[ ] year_avg = df['Год'].mean()
    hp_avg = df['Объем_двигателя'].mean()
    mileage_avg = df['Пробег'].mean()
    doors_avg = df['Двери'].mean()
    owners_avg = df['Количество_владельцев'].mean()
    year_avg, hp_avg, mileage_avg, doors_avg, owners_avg
```

(2010.0697674418604,
3.0279069767441866,
130935.11627906977,
3.116279069767442,
2.813953488372093)

Рисунок 2 – Код для получения средних значений

Можем заметить, что средний автомобиль из выборки был в пользовании около 15 лет, имеет внушительный объем двигателя, характерный для спортивных автомобилей, а также пробег на уровне 9000км в год, что приемлемо для периодической городской езды. Можем заметить, что количество владельцев почти равно 3, что объясняется возрастом

автомобиля. Количество дверей в среднем 3 штуки, что говорит нам о высокой доле двухдверных автомобилей, что является особенностью спортивного типа автомобилей.

На рисунке 3 представлен процесс вычисления гармонических средних значений для нескольких столбцов из DataFrame. Код вычисляет гармонические средние для столбцов 'Год', 'Объем_двигателя', 'Пробег', 'Двери' и 'Количество_владельцев'. Для каждого столбца используется функция `hmean()`, которая возвращает гармоническое среднее всех значений в столбце. Гармоническое среднее часто используется для нахождения среднего значения в наборах данных, где важны отношения между числами, таких как скорости или коэффициенты. Результаты вычислений сохраняются в переменные `year_h`, `hp_h`, `mileage_h`, `doors_h` и `owners_h`. В конце кода эти переменные выводятся, показывая гармонические средние значения для каждого из столбцов.

```
[ ] year_h = hmean(df['Год'])
    hp_h = hmean(df['Объем_двигателя'])
    mileage_h = hmean(df['Пробег'])
    doors_h = hmean(df['Двери'])
    owners_h = hmean(df['Количество_владельцев'])
    year_h, hp_h, mileage_h, doors_h, owners_h

(2010.0471693781128,
 2.556661850528203,
 59079.15870166928,
 2.7623126338329764,
 2.023529411764706)
```

Рисунок 3 – Гармонические средние значения

Данные указывают на то, что в наборе преобладают относительно новые автомобили с небольшим объемом двигателя, умеренным пробегом и небольшим количеством владельцев. Это может быть полезно для потенциальных покупателей, которые ищут надежные и экономичные автомобили для повседневного использования.

На рисунке 3 реализуется процесс вычисления медианных значений для нескольких столбцов из DataFrame. Код вычисляет медианные значения для столбцов 'Год', 'Объем_двигателя', 'Пробег', 'Двери' и 'Количество_владельцев'. Для каждого столбца используется функция `median()`, которая возвращает медиану всех значений в столбце. Медиана представляет собой значение, которое разделяет набор данных на две равные половины, что делает её полезной для анализа данных с выбросами или асимметричным распределением. Результаты вычислений сохраняются в переменные `year_median`, `hp_median`, `mileage_median`, `doors_median` и `owners_median`. В конце кода эти переменные выводятся, показывая медианные значения для каждого из столбцов.

```
year_median = df['Год'].median()
hp_median = df['Объем_двигателя'].median()
mileage_median = df['Пробег'].median()
doors_median = df['Двери'].median()
owners_median = df['Количество_владельцев'].median()
year_median, hp_median, mileage_median, doors_median, owners_median
```

```
(2009.0, 3.1, 121816.0, 3.0, 3.0)
```

Рисунок 4 – Медианные значения

Медианное количество дверей составляет 3.0, что указывает на преобладание в данных автомобилей с 3 или 4 дверями, таких как седаны или хэтчбеки. Медианное количество владельцев составляет 3.0, что говорит о том, что половина автомобилей имела 3 владельца или меньше, а другая половина — больше. Это может свидетельствовать о частой смене владельцев, что важно учитывать при оценке истории и состояния автомобилей.

На рисунке 5 реализуется процесс вычисления модальных значений для нескольких столбцов из DataFrame. Код вычисляет моду для столбцов 'Год', 'Объем_двигателя', 'Пробег', 'Двери' и 'Количество_владельцев'. Для каждого столбца используется функция `mode()`, которая возвращает наиболее часто встречающееся значение в столбце. Мода полезна для определения наиболее типичных или распространенных значений в наборе данных. Результаты вычислений сохраняются в переменные `year_mode`, `hp_mode`, `mileage_mode`, `doors_mode` и `owners_mode`.

```
[ ] year_mode = mode(df['Год'])
    hp_mode = mode(df['Объем_двигателя'])
    mileage_mode = mode(df['Пробег'])
    doors_mode = mode(df['Двери'])
    owners_mode = mode(df['Количество_владельцев'])
    year_mode, hp_mode, mileage_mode, doors_mode, owners_mode
```

```
(ModeResult(mode=2007, count=5),
 ModeResult(mode=2.0, count=3),
 ModeResult(mode=5356, count=1),
 ModeResult(mode=2, count=17),
 ModeResult(mode=1, count=11))
```

Рисунок 5 – Мода значений

Модальное количество дверей — 2, что встречается 17 раз, указывает на то, что двухдверные автомобили являются наиболее распространенными в наборе данных. Это может говорить о популярности кузовов типа купе или компактных моделей. Модальное количество владельцев — 1, что встречается 11 раз, указывает на то, что большинство

автомобилей в наборе данных имели только одного владельца. Это может свидетельствовать о том, что автомобили не часто перепродавались, что может быть признаком их хорошего состояния или относительно недавней покупки. В целом, данные показывают, что наиболее типичные автомобили в наборе — это модели 2007 года выпуска с объемом двигателя 2.0, двухдверные и с одним владельцем.

На рисунке 6 реализуется процесс вычисления перцентилей для нескольких столбцов из DataFrame. Код вычисляет 25-й, 50-й и 90-й перцентили для столбцов 'Год', 'Объем_двигателя', 'Пробег', 'Двери' и 'Количество_владельцев'. Для каждого столбца используется функция `np.percentile()`, которая возвращает значение, ниже которого falls определенный процент данных. Перцентили полезны для понимания распределения данных и выявления выбросов. Результаты вычислений выводятся для каждого столбца.

```
percentile25 = np.percentile(df['Год'],25)
percentile50 = np.percentile(df['Год'],50)
percentile90 = np.percentile(df['Год'],90)
print(f"Перцентиль 25: {percentile25}, Перцентиль 50: {percentile50}, Перцентиль 90: {percentile90} для столбца Год")
percentile25 = np.percentile(df['Объем_двигателя'],25)
percentile50 = np.percentile(df['Объем_двигателя'],50)
percentile90 = np.percentile(df['Объем_двигателя'],90)
print(f"Перцентиль 25: {percentile25}, Перцентиль 50: {percentile50}, Перцентиль 90: {percentile90} для столбца Объем_двигателя")
percentile25 = np.percentile(df['Пробег'],25)
percentile50 = np.percentile(df['Пробег'],50)
percentile90 = np.percentile(df['Пробег'],90)
print(f"Перцентиль 25: {percentile25}, Перцентиль 50: {percentile50}, Перцентиль 90: {percentile90} для столбца Пробег")
percentile25 = np.percentile(df['Двери'],25)
percentile50 = np.percentile(df['Двери'],50)
percentile90 = np.percentile(df['Двери'],90)
print(f"Перцентиль 25: {percentile25}, Перцентиль 50: {percentile50}, Перцентиль 90: {percentile90} для столбца Двери")
percentile25 = np.percentile(df['Количество_владельцев'],25)
percentile50 = np.percentile(df['Количество_владельцев'],50)
percentile90 = np.percentile(df['Количество_владельцев'],90)
print(f"Перцентиль 25: {percentile25}, Перцентиль 50: {percentile50}, Перцентиль 90: {percentile90} для столбца Количество_владельцев")
```

```
Перцентиль 25: 2004.5, Перцентиль 50: 2009.0, Перцентиль 90: 2020.0 для столбца Год
Перцентиль 25: 2.1, Перцентиль 50: 3.1, Перцентиль 90: 4.58 для столбца Объем_двигателя
Перцентиль 25: 69726.5, Перцентиль 50: 121816.0, Перцентиль 90: 262214.2 для столбца Пробег
Перцентиль 25: 2.0, Перцентиль 50: 3.0, Перцентиль 90: 5.0 для столбца Двери
Перцентиль 25: 1.5, Перцентиль 50: 3.0, Перцентиль 90: 4.8000000000000004 для столбца Количество_владельцев
```

Рисунок 6 – Перцентили для столбцов

На основе перцентилей можно сделать следующие выводы: 25% автомобилей выпущены до 2004.5 года, 50% — до 2009 года, а 90% — до 2020 года. Объем двигателя у 25% автомобилей менее 2.1, у 50% — менее 3.1, а у 90% — менее 4.58. Пробег у 25% автомобилей менее 69726.5 км, у 50% — менее 121816 км, а у 90% — менее 262214 км. Количество дверей у 25% автомобилей — 2 или меньше, у 50% — 3 или меньше, а у 90% — 5 или меньше. Количество владельцев у 25% автомобилей — 1 или 2, у 50% — 3 или меньше, а у 90% — 4.8 или меньше. Эти данные показывают, что в наборе преобладают автомобили среднего возраста с умеренным объемом двигателя, значительным пробегом и несколькими владельцами.

На рисунке 7 реализуется процесс вычисления межквартильного размаха (IQR) для нескольких столбцов из DataFrame. Код вычисляет IQR для столбцов 'Год', 'Объем_двигателя', 'Пробег', 'Двери' и 'Количество_владельцев'. Для каждого столбца используется разница между 75-м и 25-м перцентилем, что позволяет определить разброс

данных в центральной части распределения. Межквартильный размах полезен для анализа вариативности данных и выявления выбросов. Результаты вычислений сохраняются в переменные `iqr_year`, `iqr_hp`, `iqr_mileage`, `iqr_doors` и `iqr_owners`.

```
[ ] iqr_year = df['Год'].quantile(0.75)-df['Год'].quantile(0.25)
    iqr_hp = df['Объем_двигателя'].quantile(0.75)-df['Объем_двигателя'].quantile(0.25)
    iqr_mileage = df['Пробег'].quantile(0.75)-df['Пробег'].quantile(0.25)
    iqr_doors = df['Двери'].quantile(0.75)-df['Двери'].quantile(0.25)
    iqr_owners = df['Количество_владельцев'].quantile(0.75)-df['Количество_владельцев'].quantile(0.25)
    iqr_year, iqr_hp, iqr_mileage, iqr_doors, iqr_owners
```

```
(12.0, 1.7999999999999998, 105608.0, 2.0, 2.5)
```

Рисунок 7 – Межквартильный размах значений

IQR для пробега демонстрирует значительный разброс. Это означает, что в данных есть как автомобили с небольшим пробегом, так и с очень высоким, что может влиять на их состояние и стоимость. IQR для количества дверей показывает, что большинство автомобилей имеют 3 или 4 двери. Это указывает на преобладание седанов или хэтчбеков в данных. IQR для количества владельцев указывает на то, что большинство автомобилей сменили от 1 до 3 владельцев. Это может свидетельствовать о том, что автомобили не часто перепродавались, что может быть признаком их хорошего состояния.

На рисунке 8 реализуется процесс вычисления коэффициентов асимметрии (скошенности) для нескольких столбцов из DataFrame. Код вычисляет асимметрию для столбцов 'Год', 'Объем_двигателя', 'Пробег', 'Двери' и 'Количество_владельцев'. Для каждого столбца используется функция `skew()`, которая возвращает коэффициент асимметрии, показывающий, насколько распределение данных отклоняется от симметричного. Положительное значение указывает на правостороннюю асимметрию, отрицательное — на левостороннюю, а значение близкое к нулю — на симметричное распределение. Результаты вычислений сохраняются в переменные `skewness_year`, `skewness_hp`, `skewness_mileage`, `skewness_doors` и `skewness_owners`.

```
skewness_year = skew(df['Цена'], nan_policy='omit')
skewness_hp = skew(df['Объем_двигателя'], nan_policy='omit')
skewness_mileage = skew(df['Пробег'], nan_policy='omit')
skewness_doors = skew(df['Двери'], nan_policy='omit')
skewness_owners = skew(df['Количество_владельцев'], nan_policy='omit')
skewness_year, skewness_hp, skewness_mileage, skewness_doors, skewness_owners
```

```
(-0.2782095091006514,
 0.041250407417632015,
 0.48117121501273835,
 0.4949743651674853,
 0.011016211295451054)
```

Рисунок 8 – Вычисление коэффициентов асимметрии

Год выпуска автомобилей имеет легкую левостороннюю асимметрию (-0.278), что указывает на преобладание более новых моделей с некоторым количеством старых. Объем двигателя демонстрирует почти симметричное распределение (0.041), что говорит о равномерном распределении объемов. Пробег имеет правостороннюю асимметрию (0.481), указывая на наличие автомобилей с очень высоким пробегом. Количество дверей также показывает правостороннюю асимметрию (0.495), что говорит о преобладании автомобилей с меньшим количеством дверей. Количество владельцев имеет почти симметричное распределение (0.011), что указывает на равномерное распределение

На рисунке 9 реализуется процесс вычисления коэффициентов эксцесса для нескольких столбцов из DataFrame. Код вычисляет эксцесс для столбцов 'Год', 'Объем_двигателя', 'Пробег', 'Двери' и 'Количество_владельцев'. Для каждого столбца используется функция `kurtosis()`, которая возвращает коэффициент эксцесса, показывающий, насколько остроконечным или плоским является распределение данных по сравнению с нормальным распределением. Отрицательный эксцесс указывает на более плоское распределение, а положительный — на более остроконечное. Результаты вычислений сохраняются в переменные `kurtosis_year`, `kurtosis_hp`, `kurtosis_mileage`, `kurtosis_doors` и `kurtosis_owners`.

```
[ ] kurtosis_year = kurtosis(df['Цена'], nan_policy='omit', bias=False)
    kurtosis_hp = kurtosis(df['Объем_двигателя'], nan_policy='omit', bias=False)
    kurtosis_mileage = kurtosis(df['Пробег'], nan_policy='omit', bias=False)
    kurtosis_doors = kurtosis(df['Двери'], nan_policy='omit', bias=False)
    kurtosis_owners = kurtosis(df['Количество_владельцев'], nan_policy='omit', bias=False)
    kurtosis_year, kurtosis_hp, kurtosis_mileage, kurtosis_doors, kurtosis_owners
```

```
(-0.29761186943254003,
-1.1064851981571966,
-0.6224675361517815,
-1.1210316470831305,
-1.2973081567453517)
```

Рисунок 9 – Код для вычисления коэффициента эксцесса

Год выпуска автомобилей имеет плоское распределение (-0.298), что указывает на равномерное распределение данных. Объем двигателя также демонстрирует плоское распределение (-1.106), говорящее о равномерности значений. Пробег показывает умеренно плоское распределение (-0.622), что указывает на некоторую равномерность с легкой концентрацией вокруг среднего. Количество дверей имеет плоское распределение (-1.121), что говорит о равномерном распределении данных. Количество владельцев демонстрирует очень плоское распределение (-1.297), указывая на равномерность значений.

На основе всех предоставленных данных можно сделать следующие выводы. Средний год выпуска автомобилей составляет примерно **2010.07**, с медианой **2009.0** и

модой **2007**, что указывает на преобладание относительно новых моделей, но с некоторым количеством старых. Объем двигателя имеет среднее гармоническое **2.56**, медиану **3.1** и моду **2.0**, что говорит о преобладании автомобилей с небольшим объемом двигателя, но с равномерным распределением (эксцесс **-1.106**).

Пробег имеет среднее гармоническое **59079.16**, медиану **121816.0** и моду **5356**, что указывает на значительный разброс данных с правосторонней асимметрией (0.481) и умеренно плоским распределением (эксцесс **-0.622**). Это говорит о наличии как автомобилей с небольшим пробегом, так и с очень высоким.

Количество дверей имеет среднее гармоническое **2.76**, медиану **3.0** и моду **2**, что указывает на преобладание автомобилей с меньшим количеством дверей, но с равномерным распределением (эксцесс **-1.121**).

Количество владельцев имеет среднее гармоническое **2.02**, медиану **3.0** и моду **1**, что говорит о том, что большинство автомобилей имели одного или двух владельцев, с почти симметричным распределением (0.011) и очень плоским распределением (эксцесс **-1.297**).

В целом, данные показывают, что в наборе преобладают относительно новые автомобили с небольшим объемом двигателя, умеренным пробегом и небольшим количеством владельцев. Распределения данных в основном равномерные, с некоторыми отклонениями, такими как правосторонняя асимметрия пробега и левосторонняя асимметрия года выпуска.

Вывод:

Проведенная работа была направлена на анализ и исследование данных с использованием статистических методов и визуализации. В процессе работы были применены различные подходы, включая расчет средних значений, медиан, мод, перцентилей, межквартильных размахов, коэффициентов асимметрии и эксцесса, что позволило оценить характер распределения данных, выявить особенности и тенденции. С помощью библиотек Python, таких как pandas, numpy и scipy, удалось провести детальный анализ данных, включая расчет гармонических средних, медиан, мод и других статистических показателей. В процессе работы я научился применять различные статистические методы для анализа данных, интерпретировать результаты расчетов и делать выводы на основе полученных данных.