

Федеральное государственное автономное образовательное учреждение  
высшего образования

«Пермский государственный национальный исследовательский  
университет»

Институт компьютерных наук и технологий

Отчет  
по лабораторной работе № 5

по дисциплине  
«Введение в анализ данных»

Студент Власова Елизавета Александровна

Группа ИТ-14-2023

Пермь 2025

## Оглавление

Основное задание (5 баллов).....	3
----------------------------------	---

## Основное задание (5 баллов)

Для визуализации можно использовать любые библиотеки на свой выбор, но обязательно продемонстрировать хотя бы 2 разных (например, matplotlib и seaborn).

1. Загрузите данные из файла «weather1.csv» о погоде в Перми.

Загрузите только следующие столбцы:

- a. Местное время в Перми;
- b. T (температура воздуха в градусах Цельсия);
- c. P (атмосферное давление в мм.рт.ст.);
- d. U (относительная влажность в %);
- e. Ff (скорость ветра в м/с);
- f. N (облачность);
- g. H (высота основания облаков, м);
- h. VV (горизонтальная дальность видимости в км).

### Решение:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 1
# загрузка данных
df = pd.read_csv(
    "weather1.csv",
    sep=";",
    quotechar='"',
    encoding="utf-8",
    usecols=["Местное время в Перми", "T", "P", "U", "Ff", "N", "H", "VV"]
)

# преобразуем столбец времени в datetime
df["Местное время в Перми"] = pd.to_datetime(df["Местное время в Перми"], errors="coerce")
```

2. (0.5 балла) Постройте точечную диаграмму (диаграмму рассеяния) по признакам температуры и относительной влажности.

### Решение:

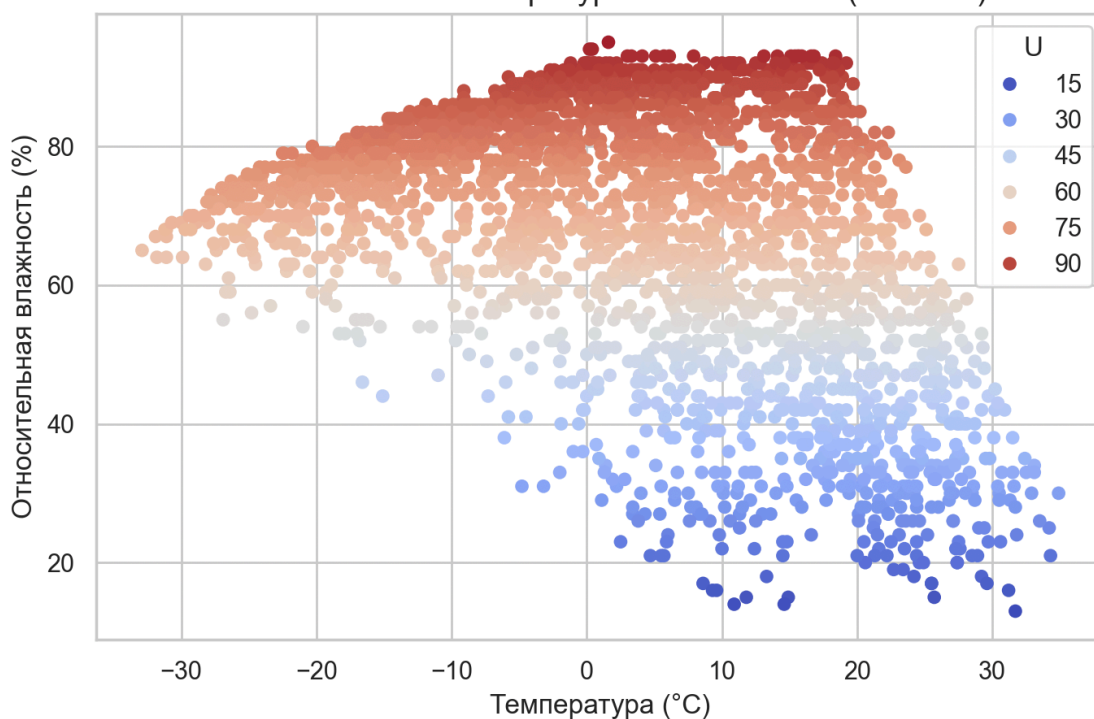
```
# 2
# диаграмма зависимости температуры от влажности с Matplotlib
plt.figure(figsize=(8, 5))
plt.scatter(df["T"], df["U"], color="dodgerblue", alpha=0.6, edgecolors="k")
plt.title("2.Зависимость температуры от влажности (Matplotlib)", fontsize=14)
plt.xlabel("Температура (°C)", fontsize=12)
plt.ylabel("Относительная влажность (%)", fontsize=12)
plt.grid(True)
plt.show()

# диаграмма зависимости температуры от влажности с Seaborn
sns.set(style="whitegrid")
plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x="T", y="U", hue="U", palette="coolwarm", edgecolor="none")
plt.title("2.Зависимость температуры от влажности (Seaborn)", fontsize=14)
plt.xlabel("Температура (°C)", fontsize=12)
plt.ylabel("Относительная влажность (%)", fontsize=12)
plt.show()
```

**Результаты работы программы:**



## 2. Зависимость температуры от влажности (Seaborn)



3. (0.5 балла) На построенной в предыдущем пункте диаграмме выделите точки разными цветами в зависимости от облачности: синим — для которых облачность составляет 100%; красным — все остальные.

### Решение:

```
# 3
# диаграмма с выделением точек по облачности с Matplotlib

# облачность (N) может содержать строки вида "100%." — извлекаем числовые значения
df["N_clean"] = df["N"].astype(str).str.extract(r"(\d+)") # достаём цифры
df["N_clean"] = pd.to_numeric(df["N_clean"], errors="coerce") # преобразуем в числа

cloud_100 = df[df["N_clean"] == 100]
cloud_other = df[df["N_clean"] != 100]

plt.figure(figsize=(8, 5))
plt.scatter(cloud_other["T"], cloud_other["U"], color="red", alpha=0.6, label="Облачность ≠ 100%")
plt.scatter(cloud_100["T"], cloud_100["U"], color="blue", alpha=0.6, label="Облачность = 100%")

plt.title("3. Температура и влажность (окраска по облачности)", fontsize=14)
plt.xlabel("Температура (°C)", fontsize=12)
plt.ylabel("Относительная влажность (%)", fontsize=12)
plt.legend()
plt.grid(True)
plt.show()
```

**Результаты работы программы:**



4. (0.5 балла) Постройте линейную диаграмму (график) изменения температуры в зависимости от местного времени.

**Решение:**

```
# 4
# линейная диаграмма изменения температуры во времени с Matplotlib
plt.figure(figsize=(10, 5))
plt.plot(df["Местное время в Перми"], df["T"], color="green", linewidth=1.5)
plt.title("4.Изменение температуры во времени (Пермь)", fontsize=14)
plt.xlabel("Местное время", fontsize=12)
plt.ylabel("Температура (°C)", fontsize=12)
plt.grid(True)
plt.tight_layout()
plt.show()
```

**Результаты работы программы:**



5. (1 балл) Посчитайте по имеющимся данным среднемесячную температуру и постройте столбчатую диаграмму (вертикальную) зависимости средней температуры от месяца. Подсказка: создайте отдельный столбец с номером месяца (вычислив его из столбца «Местное время»), а затем сгруппируйте данные по этому столбцу.

**Решение:**

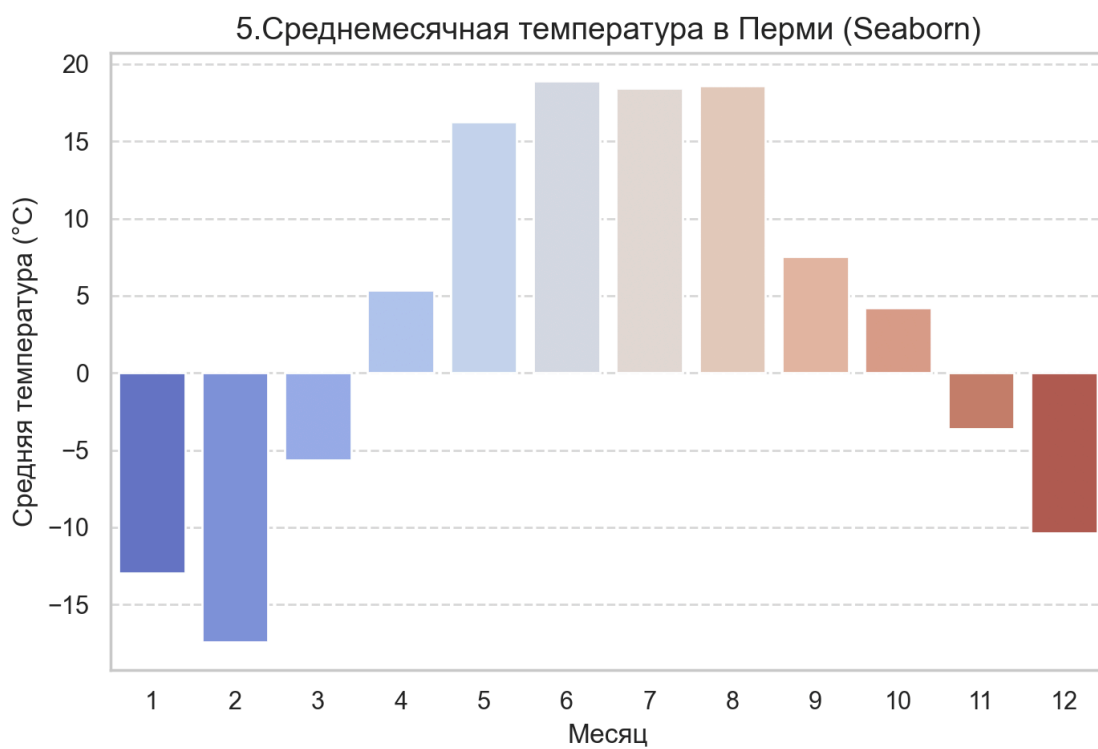
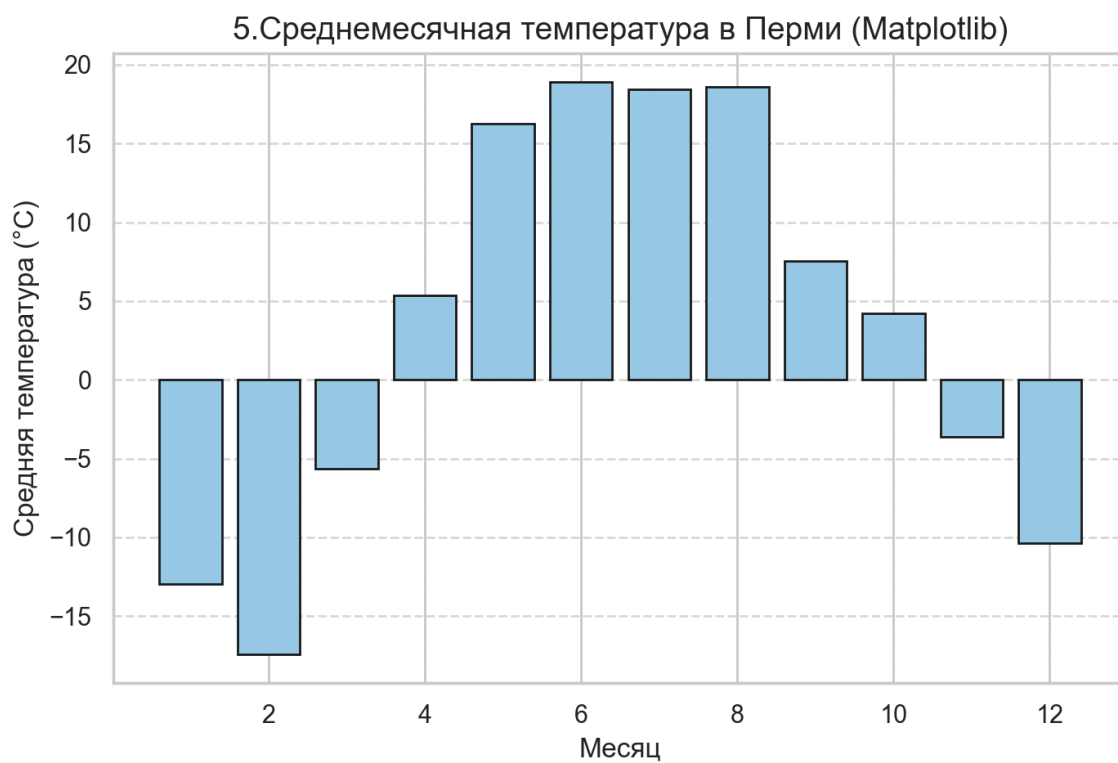
```
# 5
# среднемесячная температура и столбчатая диаграмма
# добавляем столбец "Месяц"
df["Месяц"] = df["Местное время в Перми"].dt.month

# группируем и считаем среднюю температуру по каждому месяцу
monthly_avg = df.groupby("Месяц")["T"].mean().reset_index()

# диаграмма Matplotlib
plt.figure(figsize=(8, 5))
plt.bar(monthly_avg["Месяц"], monthly_avg["T"], color="skyblue", edgecolor="k")
plt.title("5.Среднемесячная температура в Перми (Matplotlib)", fontsize=14)
plt.xlabel("Месяц")
plt.ylabel("Средняя температура (°C)")
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()

# альтернатива Seaborn
plt.figure(figsize=(8, 5))
sns.barplot(data=monthly_avg, x="Месяц", y="T", palette="coolwarm")
plt.title("5.Среднемесячная температура в Перми (Seaborn)", fontsize=14)
plt.xlabel("Месяц")
plt.ylabel("Средняя температура (°C)")
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()
```

## Результаты работы программы:



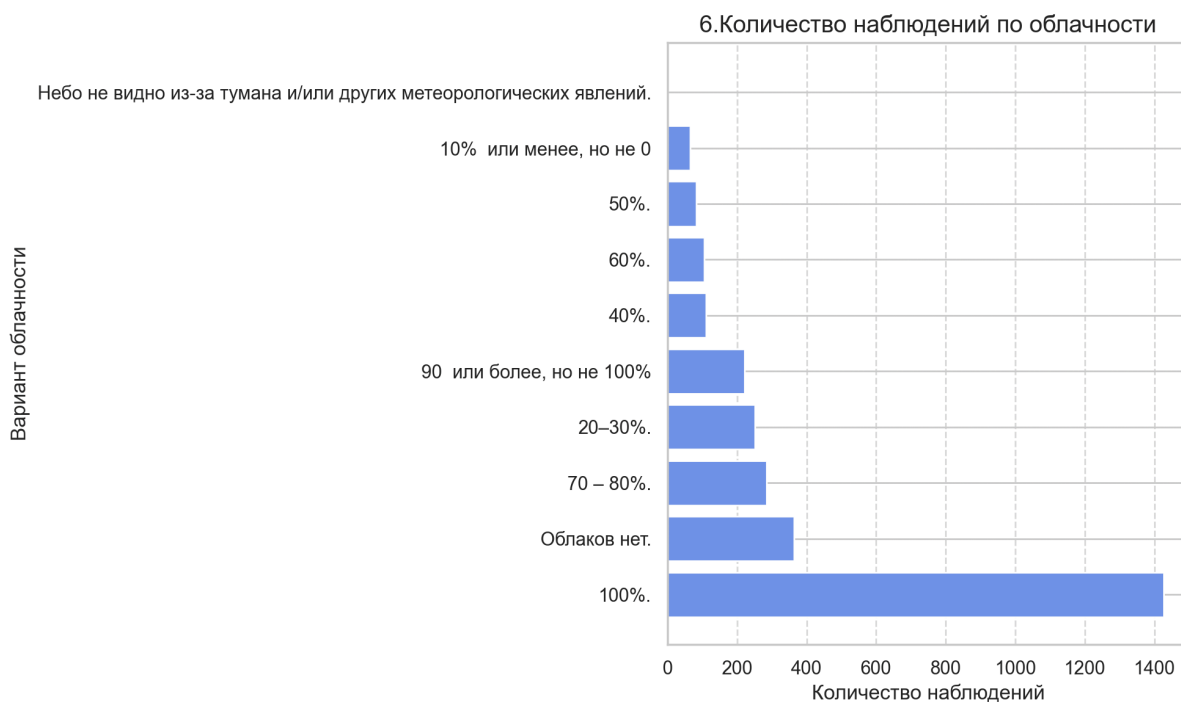
6. (0.5 балла) Постройте ленточную диаграмму (горизонтальную), отразив на ней количество имеющихся наблюдений для каждого варианта облачности.

**Решение:**

```
# 6
# ленточная (горизонтальная) диаграмма по облачности
# считаем количество наблюдений для каждого варианта облачности
cloud_counts = df["N"].value_counts().reset_index()
cloud_counts.columns = ["Облачность", "Количество"]

# диаграмма Matplotlib
plt.figure(figsize=(10, 6))
plt.barh(cloud_counts["Облачность"], cloud_counts["Количество"], color="cornflowerblue")
plt.title("6.Количество наблюдений по облачности", fontsize=14)
plt.xlabel("Количество наблюдений")
plt.ylabel("Вариант облачности")
plt.grid(axis="x", linestyle="--", alpha=0.7)
plt.tight_layout()
plt.show()
```

**Результаты работы программы:**

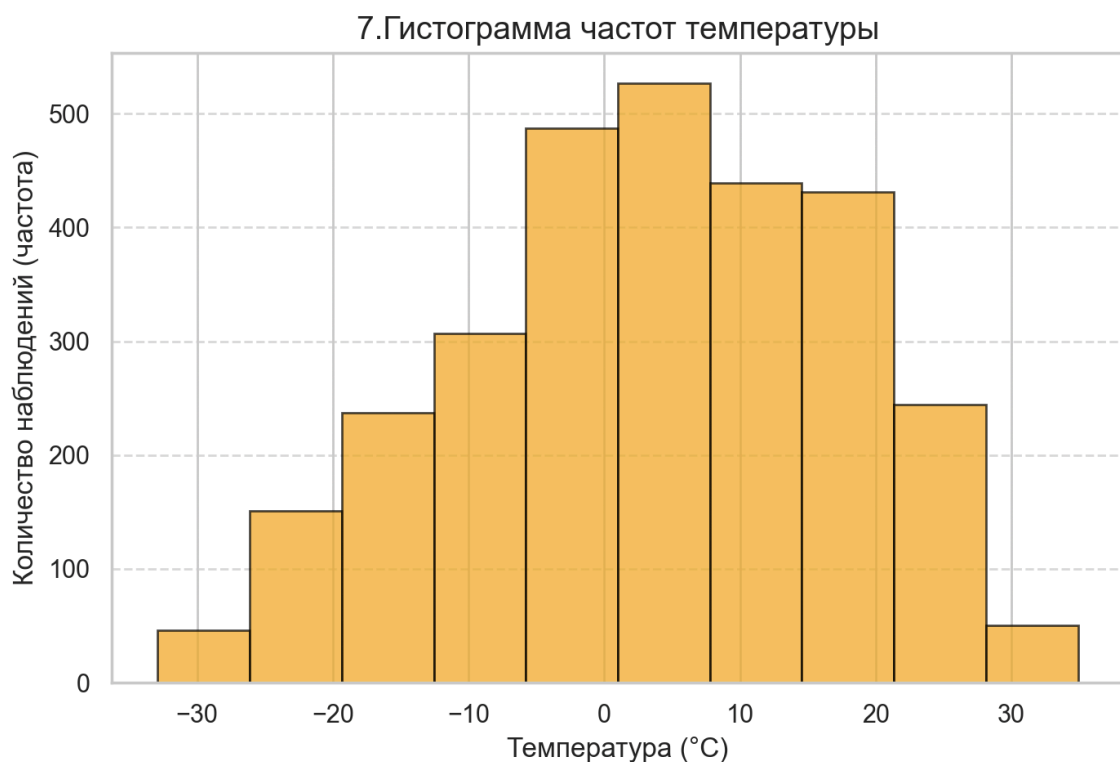


7. (0.5 балла) Постройте гистограмму частот для температуры. На гистограмме должно быть 10 диапазонов температуры.

**Решение:**

```
# 7
# гистограмма частот температуры Matplotlib
plt.figure(figsize=(8, 5))
plt.hist(df["T"], bins=10, color="orange", edgecolor="black", alpha=0.7)
plt.title("7.Гистограмма частот температуры", fontsize=14)
plt.xlabel("Температура (°C)")
plt.ylabel("Количество наблюдений (частота)")
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()
```

**Результаты работы программы:**



8. (1 балл) Разбейте данные на 3 группы по значению горизонтальной дальности видимости (одна группа – дальность видимости менее 5 км, вторая – от 5 до 15 км(включительно), третья – более 15 км). В одной области для каждой группы постройте boxplot (диаграмму «ящик с усами») для признака «атмосферное давление».

**Решение:**

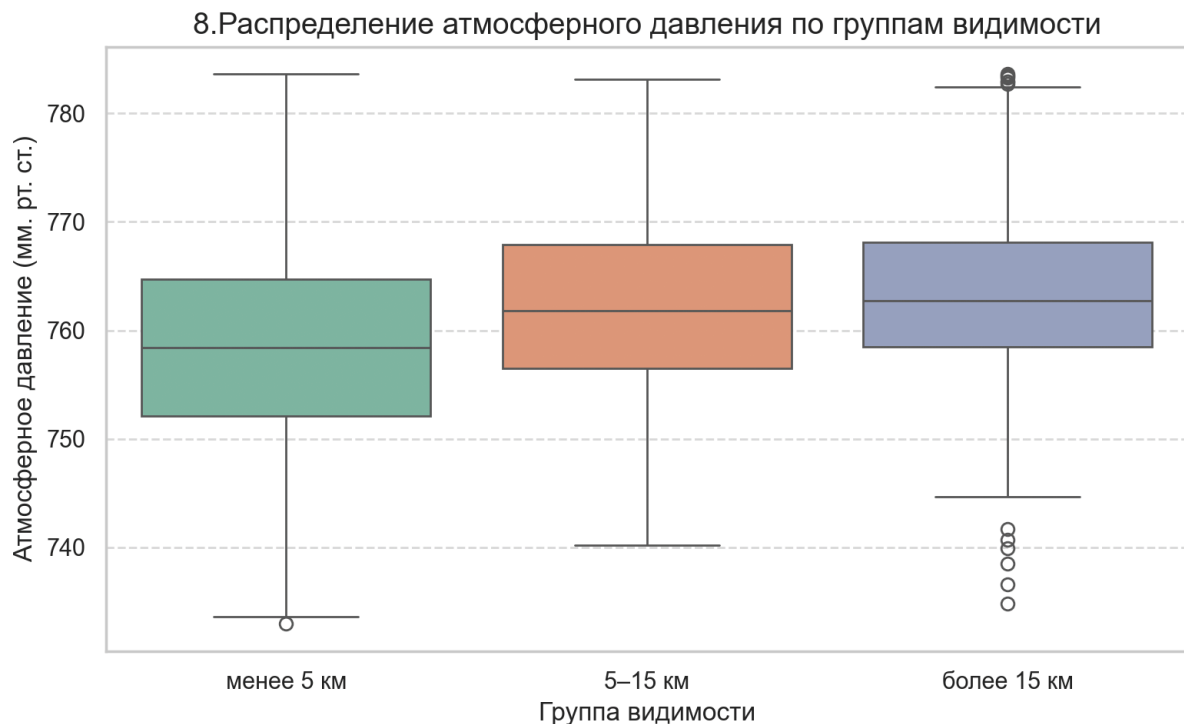
```
# 8
# Boxplot атмосферного давления по группам видимости
# создаем группы по VV
def visibility_group(v):
    if v < 5:
        return "менее 5 км"
    elif 5 <= v <= 15:
        return "5-15 км"
    else:
        return "более 15 км"

df["Группа видимости"] = df["VV"].apply(visibility_group)

# проверка распределения по группам
print(df["Группа видимости"].value_counts())

# строим boxplot
plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x="Группа видимости", y="P", palette="Set2")
plt.title("8.Распределение атмосферного давления по группам видимости", fontsize=14)
plt.xlabel("Группа видимости")
plt.ylabel("Атмосферное давление (мм. рт. ст.)")
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.tight_layout()
plt.show()
```

## Результаты работы программы:



Эти кружочки на boxplot — это выбросы (outliers), то есть значения, которые выходят за “нормальный” диапазон данных.

Когда Seaborn или Matplotlib строят boxplot:

- прямоугольник показывает межквартильный размах (IQR) — область от 25% до 75% данных;
- горизонтальная линия внутри прямоугольника — это медиана;
- “усы” тянутся до крайних значений, не выходящих за границы:  
[Q1–1.5×IQR, Q3+1.5×IQR][Q1–1.5×IQR,Q3+1.5×IQR]
- все, что выше или ниже этих границ, считается выбросом и отображается кружочком (o).

9. (0.5 балла) Постройте круговую диаграмму для признака «высота основания облаков».

**Решение:**

```
# 9
# круговая диаграмма по высоте основания облаков (H)

# группируем данные
h_counts = df["H"].value_counts().reset_index()
h_counts.columns = ["Высота основания облаков (м)", "Количество"]

# сортируем по высоте (для логичного порядка)
h_counts = h_counts.sort_values(by="Высота основания облаков (м)")

# построение красивой диаграммы Matplotlib
plt.figure(figsize=(9, 7))
wedges, texts, autotexts = plt.pie(
    h_counts["Количество"],
    autopct="%1.1f%%",
    startangle=120,
    colors=plt.cm.Set3.colors,
    wedgeprops={"edgecolor": "white"},
    pctdistance=0.8
)
```

```
# добавляем легенду справа, чтобы не перекрывала диаграмму
plt.legend(
    wedges,
    h_counts["Высота основания облаков (м)"],
    title="Высота основания облаков (м)",
    loc="center left",
    bbox_to_anchor=(1, 0.5),
    fontsize=10
)

plt.title("9.Распределение по высоте основания облаков", fontsize=14)
plt.tight_layout()
plt.show()
```

## Результаты работы программы:

9.Распределение по высоте основания облаков

