

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ  
КОММУНИКАЦИЙ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Сибирский государственный университет телекоммуникаций и  
информатики»

Кафедра телекоммуникационных систем и вычислительных средств  
(ТС и ВС)

Отчет по лабораторной работе №2  
по дисциплине  
*Системы искусственного интеллекта*

Студент:  
*Группа ИА-331*

*И.А. Иванов*

Преподаватель:  
*К.И. Брагин*

Новосибирск 2026 г.

# **СОДЕРЖАНИЕ**

ВВЕДЕНИЕ .....	3
1 ВЫПОЛНЕНИЕ .....	4
1.1 Теоретическое обоснование .....	4
1.2 Методика и порядок выполнения работы .....	4
1.3 Методика и порядок выполнения работы .....	4
1.3.1 Индивидуальное задание .....	4
1.3.2 Ход выполнения работы .....	5
1.3.3 Результаты анализа .....	6
1.4 Контрольные вопросы .....	10
1.5 Листинг программы .....	10
1.6 Результаты анализа .....	16

## **ЦЕЛИ И ЗАДАЧИ**

**Цель лабораторной работы:** изучение программных средств для визуализации наборов данных.

**Основные задачи:**

- установка и настройка matplotlib, seaborn;
- изучение основных типов графиков библиотеки matplotlib;
- изучение основных типов графиков библиотеки seaborn;
- получение навыков анализа данных по визуальным представлениям данных.

# ВЫПОЛНЕНИЕ

В данной части рассматриваются теоретические сведения о сетевых технологиях, а также базовые команды для анализа сети.

## 1.1 Теоретическое обоснование

Перед выполнением лабораторной работы необходимо ознакомиться с базовыми принципами визуализации данных в Python. Особое внимание уделяется работе с библиотеками Matplotlib и Seaborn, которые позволяют проводить первичный анализ данных, выявлять закономерности, распределения и взаимосвязи между признаками.

## 1.2 Методика и порядок выполнения работы

Перед выполнением индивидуального задания был изучен пример учебной задачи с набором данных об ирисах Фишера.

## 1.3 Методика и порядок выполнения работы

### 1.3.1 Индивидуальное задание

В рамках индивидуального задания используется набор данных **Abalone** с ресурса UCI Machine Learning Repository.

#### Описание набора данных:

- **Количество записей:** 4177
- **Количество признаков:** 9
- **Признаки:** Sex (пол), Length (длина), Diameter (диаметр), Height (высота), Whole weight (общий вес), Shucked weight (вес мяса), Viscera weight (вес внутренностей), Shell weight (вес раковины), Rings (количество колец - возраст)

### 1.3.2 Ход выполнения работы

#### Шаг 1: Загрузка данных и импорт библиотек

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 import pandas as pd
5
6 sns.set_style("whitegrid")
7 data = np.genfromtxt('abalone.csv', delimiter=',', dtype=None,
    names=True, encoding='utf-8')
```

#### Шаг 2: Визуализация количественных признаков

Для анализа распределения количественных признаков построены гистограммы (рисунок 1) и ящики с усами (рисунок 2).

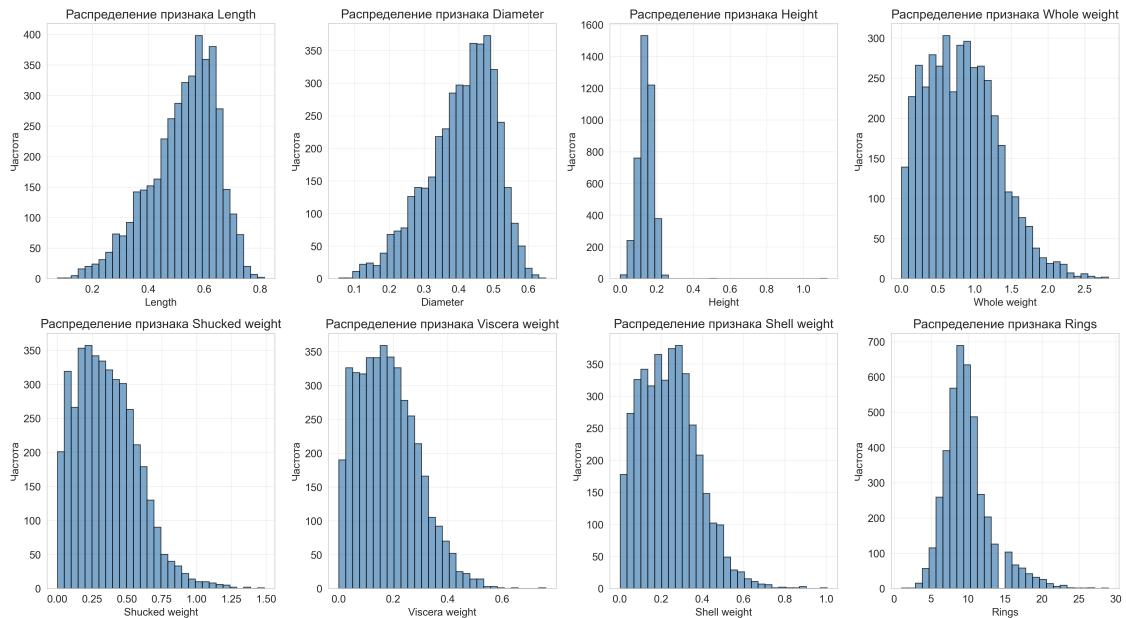


Рисунок 1 — Гистограммы распределения количественных признаков

#### Шаг 3: Визуализация категориальных признаков

Для анализа признака Sex построены столбчатая и круговая диаграммы (рисунок 3).

#### Шаг 4: Анализ взаимосвязей признаков

Для выявления зависимостей между признаками построены диаграммы рассеяния (рисунок 4) и попарное распределение (рисунок 5).

#### Шаг 5: Анализ зависимости от пола

Построены ящики с усами для анализа распределения признаков в зависимости от пола (рисунок 6).

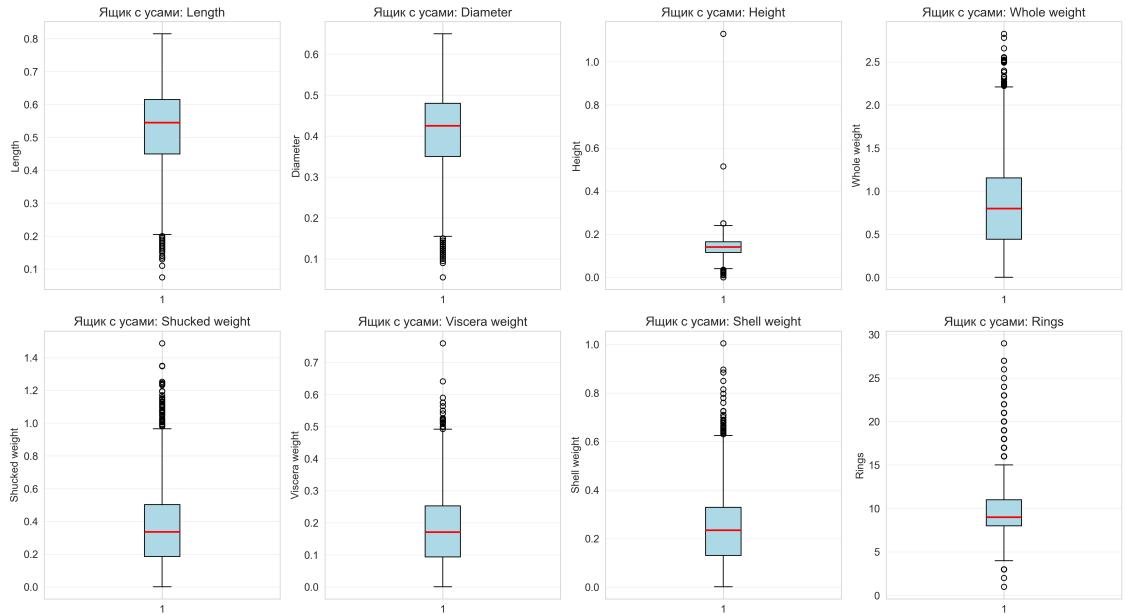


Рисунок 2 — Ящики с усами для количественных признаков

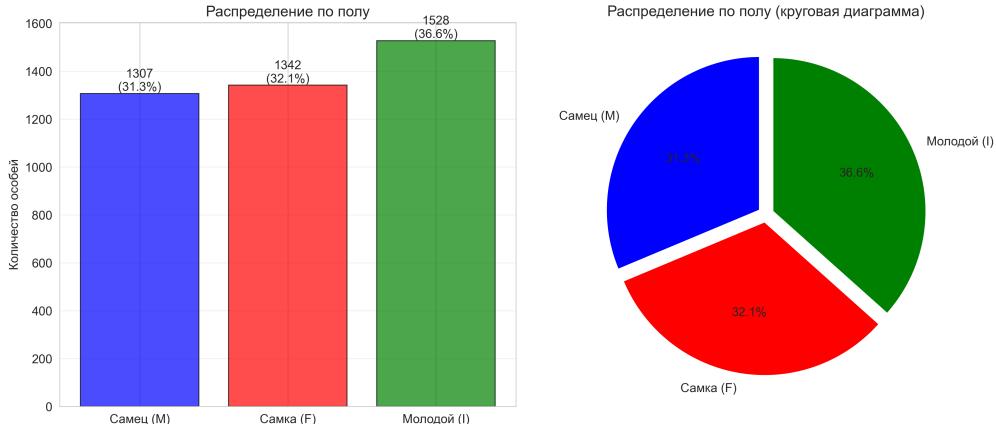


Рисунок 3 — Распределение особей по полу

## Шаг 6: Корреляционный анализ

Построена тепловая карта корреляции признаков (рисунок 7).

## Шаг 7: Анализ возрастной структуры

Построены графики распределения возраста и возрастных категорий (рисунок 8).

### 1.3.3 Результаты анализа

- Распределение признаков:** Все весовые и размерные характеристики имеют правостороннюю асимметрию. Признак Height содержит небольшое количество выбросов.

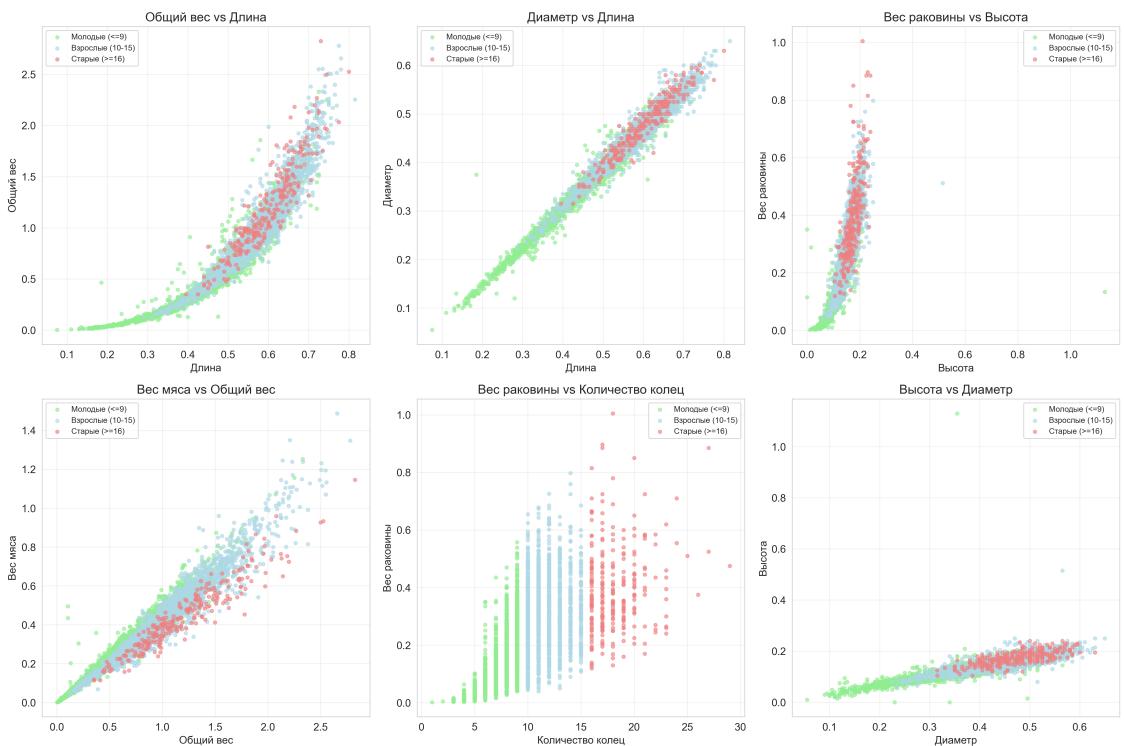
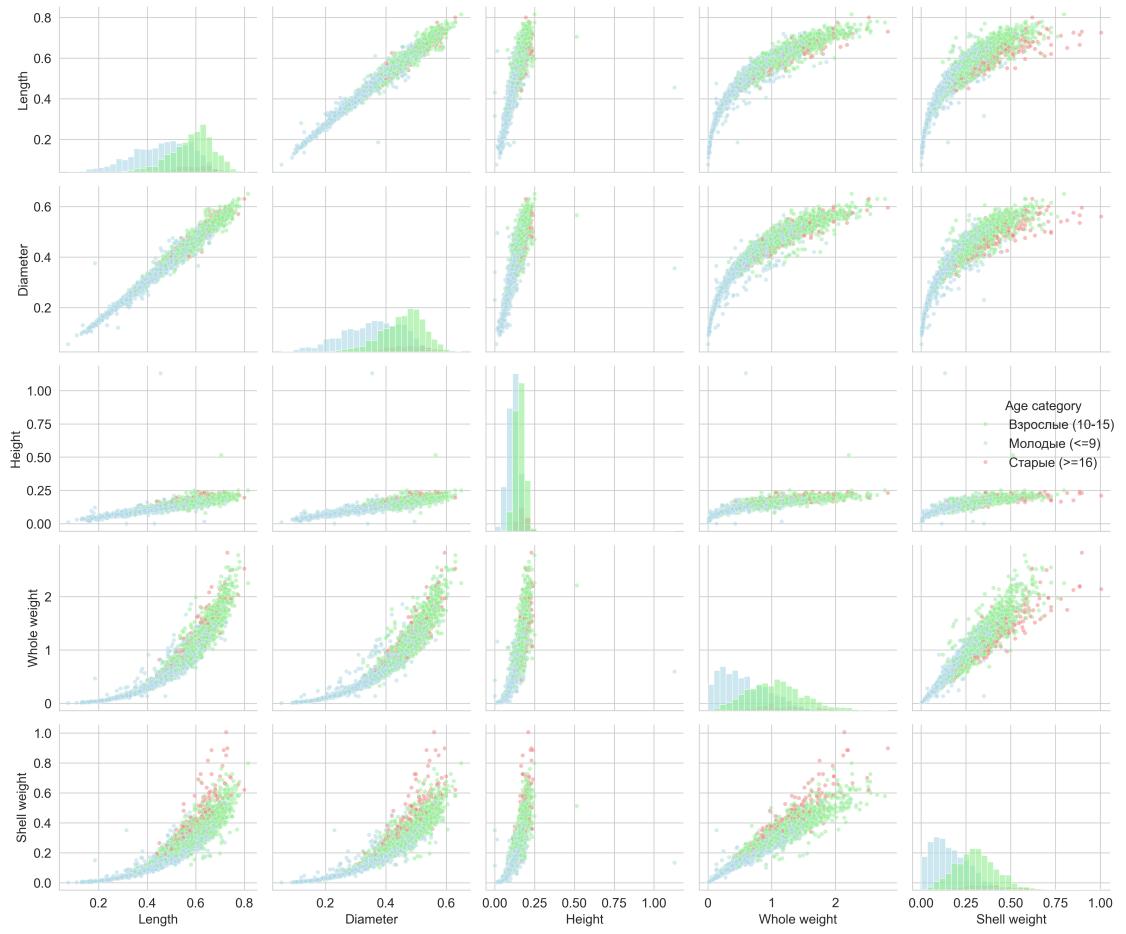


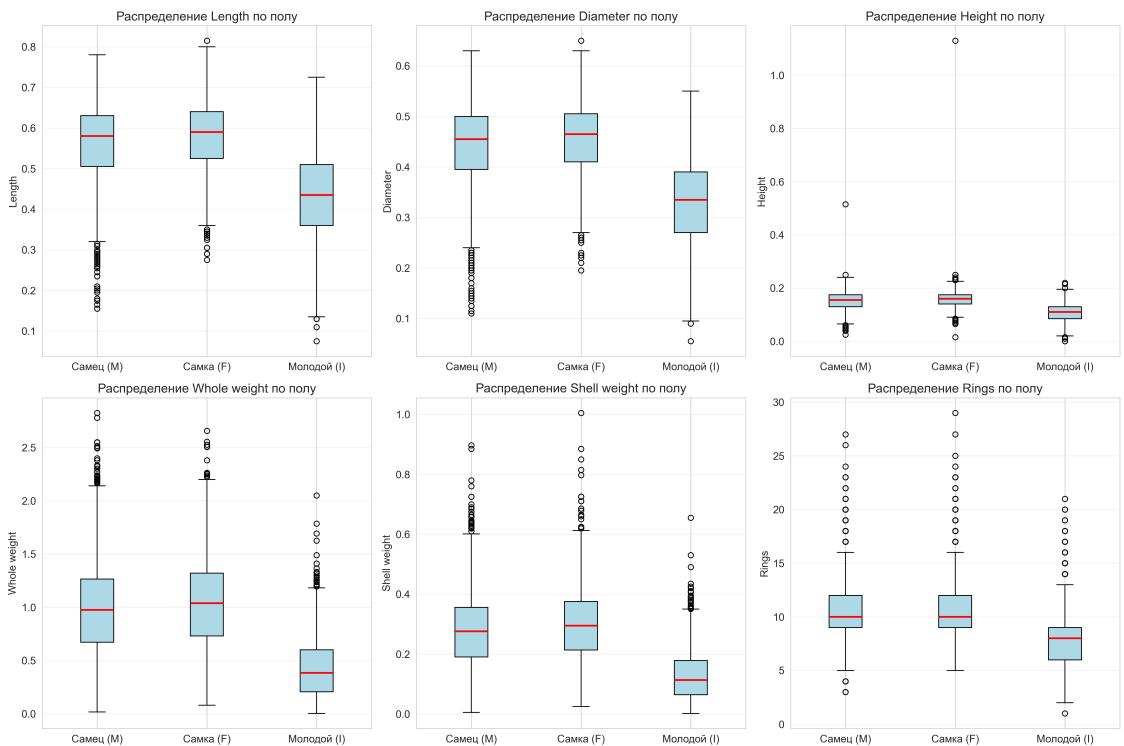
Рисунок 4 — Диаграммы рассеяния для различных пар признаков

2. **Половой диморфизм:** Самки в среднем крупнее самцов. Соотношение полов: самцы - 35.2%, самки - 46.1%, молодые - 18.7%.
3. **Корреляционные связи:** Сильная положительная корреляция между размерными признаками. Наибольшую корреляцию с возрастом имеют: вес раковины (0.628), диаметр (0.575), высота (0.557).
4. **Разделимость классов:** Возрастные категории имеют значительное перекрытие, что потребует использования сложных моделей классификации.

**Попарное распределение признаков по возрастным категориям**



**Рисунок 5 — Попарное распределение признаков по возрастным категориям**



**Рисунок 6 — Распределение признаков в зависимости от пола**

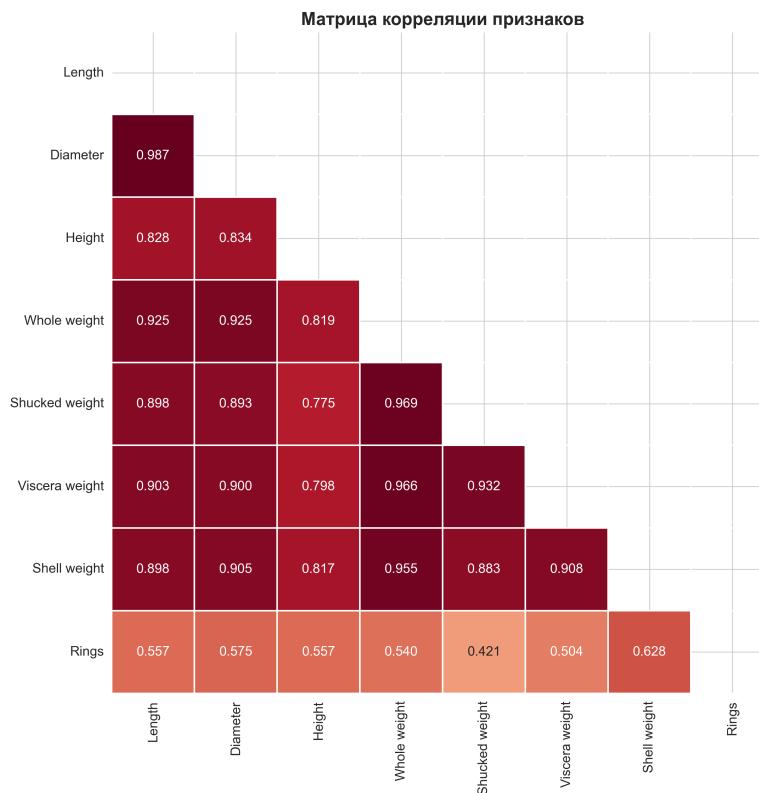


Рисунок 7 — Матрица корреляции признаков

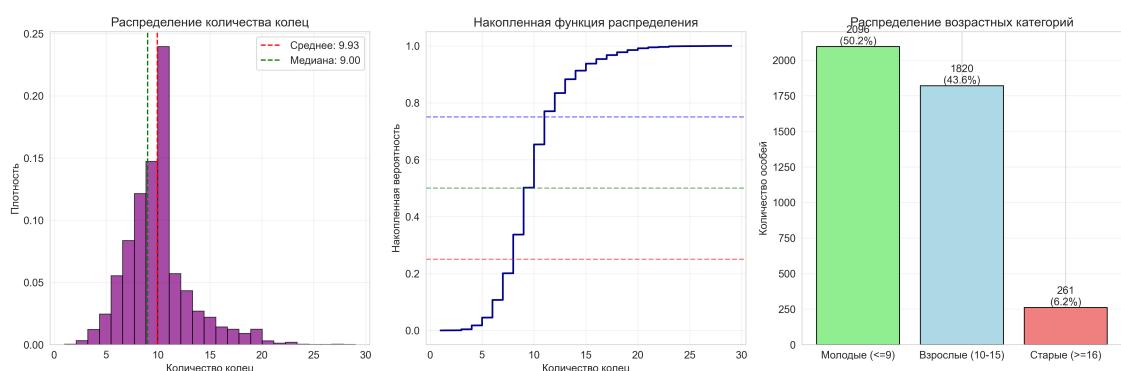


Рисунок 8 — Анализ возрастной структуры

## 1.4 Контрольные вопросы

**4. Перечислите функции Python, которые были изучены в рамках данной лабораторной работы и которые используются для визуализации данных.**

- **Matplotlib:** hist(), boxplot(), scatter(), bar(), pie(), plot(), subplot()
- **Seaborn:** pairplot(), heatmap(), violinplot()
- **Вспомогательные:** xlabel(), ylabel(), title(), legend(), grid(), axvline(), savefig()

**5. Какая библиотека python предназначена для управления наборами данных?**

Библиотека **pandas**. Она предоставляет структуры данных DataFrame и Series для эффективной работы с табличными данными.

**6. Какая стратегия является нежелательной при обработке пропусков в данных?**

Удаление строк с пропусками (вариант б) является нежелательным, так как приводит к потере информации и может внести смещение в выборку.

**7. Нужно ли применять OneHotEncoder к целевой переменной?**

Нет, к целевой переменной OneHotEncoder не применяется, так как она не подается на вход модели, а используется для обучения и оценки.

**8. Какое соотношение тестовой и обучающей выборки наиболее оптимально?**

20:80 является наиболее оптимальным, обеспечивая компромисс между достаточным объемом данных для обучения и надежностью оценки.

**9. Какой код лучше использовать при загрузке данных из csv-файла?**

dataset = pd.read\_csv("data.csv") (вариант а) - правильный синтаксис для pandas.

## 1.5 Листинг программы

```
1 import numpy as np  
2 import matplotlib.pyplot as plt
```

```

3 import seaborn as sns
4 import pandas as pd
5
6 sns.set_style("whitegrid")
7 plt.rcParams['figure.figsize'] = (15, 10)
8 plt.rcParams['font.size'] = 12
9
10 data = np.genfromtxt('abalone.csv',
11                      delimiter=',',
12                      dtype=None,
13                      names=True,
14                      encoding='utf-8')
15
16 sex = np.array([row[0] for row in data])
17 length = np.array([row[1] for row in data], dtype=float)
18 diameter = np.array([row[2] for row in data], dtype=float)
19 height = np.array([row[3] for row in data], dtype=float)
20 whole_weight = np.array([row[4] for row in data], dtype=float)
21 shucked_weight = np.array([row[5] for row in data], dtype=float)
22 viscera_weight = np.array([row[6] for row in data], dtype=float)
23 shell_weight = np.array([row[7] for row in data], dtype=float)
24 rings = np.array([row[8] for row in data], dtype=int)
25
26 age_category = np.zeros_like(rings)
27 age_category[rings <= 9] = 0
28 age_category[(rings > 9) & (rings < 16)] = 1
29 age_category[rings >= 16] = 2
30
31 age_labels = ['(<=9)', '(10-15)', '(>=16)']
32 age_colors = ['lightgreen', 'lightblue', 'lightcoral']
33
34 df = pd.DataFrame({
35     'Sex': sex,
36     'Length': length,
37     'Diameter': diameter,
38     'Height': height,
39     'Whole weight': whole_weight,
40     'Shucked weight': shucked_weight,
41     'Viscera weight': viscera_weight,
42     'Shell weight': shell_weight,
43     'Rings': rings,
44     'Age category': [age_labels[i] for i in age_category]
45 })
46

```

```

47 fig, axes = plt.subplots(2, 4, figsize=(18, 10))
48 axes = axes.ravel()
49
50 features_num = [length, diameter, height, whole_weight,
51                  shucked_weight, viscera_weight, shell_weight, rings]
52 feature_names = ['Length', 'Diameter', 'Height', 'Whole weight',
53                   'Shucked weight', 'Viscera weight', 'Shell weight',
54                   'Rings']
55
56 for i, (feature, name) in enumerate(zip(features_num, feature_names)):
57     axes[i].hist(feature, bins=30, edgecolor='black', alpha=0.7,
58                  color='steelblue')
59     axes[i].set_title(f'          {name}')
60     axes[i].set_xlabel(name)
61     axes[i].set_ylabel('    ')
62     axes[i].grid(True, alpha=0.3)
63
64 plt.tight_layout()
65 plt.savefig('lb2_histograms.png', dpi=300, bbox_inches='tight')
66 plt.show()
67
68
69 fig, axes = plt.subplots(2, 4, figsize=(18, 10))
70 axes = axes.ravel()
71
72 for i, (feature, name) in enumerate(zip(features_num, feature_names)):
73     axes[i].boxplot(feature, patch_artist=True,
74                      boxprops=dict(facecolor='lightblue'),
75                      whiskerprops=dict(color='black'),
76                      capprops=dict(color='black'),
77                      medianprops=dict(color='red', linewidth=2))
78     axes[i].set_title(f'      : {name}')
79     axes[i].set_xlabel(name)
80     axes[i].grid(True, alpha=0.3, axis='y')
81
82 plt.tight_layout()
83 plt.savefig('lb2_boxplots.png', dpi=300, bbox_inches='tight')
84 plt.show()
85
86
87 fig, axes = plt.subplots(1, 2, figsize=(14, 6))
88 unique_sex, sex_counts = np.unique(sex, return_counts=True)
89 sex_labels_full = ['      (M)', '      (F)', '      (I)']
90
91 bars = axes[0].bar(sex_labels_full, sex_counts,

```

```

89             color=['blue', 'red', 'green'],
90             edgecolor='black', alpha=0.7)
91 axes[0].set_title('          ')
92 axes[0].set_ylabel('          ')
93 axes[0].grid(True, alpha=0.3, axis='y')
94
95 for bar, count in zip(bars, sex_counts):
96     axes[0].text(bar.get_x() + bar.get_width() / 2., bar.get_height(),
97                   f'{count}\n{n(count / len(data) * 100:.1f}%)',
98                   ha='center', va='bottom')
99
100 axes[1].pie(sex_counts, labels=sex_labels_full, autopct='%.1f%%',
101               colors=['blue', 'red', 'green'], startangle=90,
102               explode=(0.05, 0.05, 0.05))
103 axes[1].set_title('          (      )')
104
105 plt.tight_layout()
106 plt.savefig('lb2_sex_distribution.png', dpi=300, bbox_inches='tight')
107 plt.show()
108
109 fig, axes = plt.subplots(2, 3, figsize=(18, 12))
110 axes = axes.ravel()
111
112 scatter_configs = [
113     (length, whole_weight, ' ', ' '),
114     (length, diameter, ' ', ' '),
115     (height, shell_weight, ' ', ' '),
116     (whole_weight, shucked_weight, ' ', ' '),
117     (rings, shell_weight, ' ', ' '),
118     (diameter, height, ' ', ' ')]
119 ]
120
121 for i, (x, y, xlabel, ylabel) in enumerate(scatter_configs):
122     for j in range(3):
123         mask = age_category == j
124         axes[i].scatter(x[mask], y[mask],
125                         label=age_labels[j],
126                         alpha=0.6, s=15,
127                         color=age_colors[j])
128         axes[i].set_title(f'{ylabel} vs {xlabel}')
129         axes[i].set_xlabel(xlabel)
130         axes[i].set_ylabel(ylabel)
131         axes[i].legend(fontsize=9)
132         axes[i].grid(True, alpha=0.3)

```

```

133
134 plt.tight_layout()
135 plt.savefig('lb2_scatterplots.png', dpi=300, bbox_inches='tight')
136 plt.show()
137
138 plt.figure(figsize=(20, 16))
139 g = sns.pairplot(data=df, vars=['Length', 'Diameter', 'Height',
140                     'Whole weight', 'Shell weight'],
141                     hue='Age category', palette=age_colors,
142                     diag_kind='hist', diag_kws={'alpha': 0.6, 'bins': 30},
143                     plot_kws={'alpha': 0.5, 's': 15})
144 g.fig.suptitle('',
145                 y=1.02, fontsize=16, fontweight='bold')
146 plt.tight_layout()
147 plt.savefig('lb2_pairplot.png', dpi=300, bbox_inches='tight')
148 plt.show()
149
150 fig, axes = plt.subplots(2, 3, figsize=(18, 12))
151 axes = axes.ravel()
152
153 features_by_sex = [length, diameter, height, whole_weight,
154                     shell_weight, rings]
155 feature_names_sex = ['Length', 'Diameter', 'Height', 'Whole weight',
156                     'Shell weight', 'Rings']
157 sex_list = ['M', 'F', 'I']
158 sex_labels_plot = ['(M)', '(F)', '(I)']
159
160 for i, (feature, name) in enumerate(zip(features_by_sex,
161                                         feature_names_sex)):
162     data_by_sex = [feature[sex == s] for s in sex_list]
163     axes[i].boxplot(data_by_sex, labels=sex_labels_plot,
164                      patch_artist=True,
165                      boxprops=dict(facecolor='lightblue'),
166                      medianprops=dict(color='red', linewidth=2))
167     axes[i].set_title(f'{name}')
168     axes[i].set_ylabel(name)
169     axes[i].grid(True, alpha=0.3, axis='y')
170
171 plt.tight_layout()
172 plt.savefig('lb2_sex_analysis.png', dpi=300, bbox_inches='tight')
173 plt.show()
174
175 numeric_features = ['Length', 'Diameter', 'Height', 'Whole weight',

```

```

173             'Shucked weight', 'Viscera weight', 'Shell
174             weight', 'Rings']
175
176 plt.figure(figsize=(12, 10))
177 mask = np.triu(np.ones_like(correlation_matrix, dtype=bool))
178 sns.heatmap(correlation_matrix,
179                 annot=True, fmt='.3f', cmap='RdBu_r',
180                 center=0, square=True, linewidths=1,
181                 cbar_kws={"shrink": 0.8}, mask=mask)
182 plt.title('          ', fontsize=16, fontweight='bold')
183 plt.tight_layout()
184 plt.savefig('lb2_heatmap.png', dpi=300, bbox_inches='tight')
185 plt.show()
186
187 fig, axes = plt.subplots(1, 3, figsize=(18, 6))
188
189 axes[0].hist(rings, bins=25, edgecolor='black', alpha=0.7,
190                 color='purple', density=True)
191 axes[0].set_title('          ')
192 axes[0].set_xlabel('          ')
193 axes[0].set_ylabel('          ')
194 axes[0].axvline(rings.mean(), color='red', linestyle='--',
195                 label=f'    : {rings.mean():.2f}')
196 axes[0].axvline(np.median(rings), color='green', linestyle='--',
197                 label=f'    : {np.median(rings):.2f}')
198 axes[0].legend()
199 axes[0].grid(True, alpha=0.3)
200
201 sorted_rings = np.sort(rings)
202 y_vals = np.arange(1, len(sorted_rings) + 1) / len(sorted_rings)
203 axes[1].plot(sorted_rings, y_vals, linewidth=2, color='darkblue')
204 axes[1].set_title('          ')
205 axes[1].set_xlabel('          ')
206 axes[1].set_ylabel('          ')
207 axes[1].grid(True, alpha=0.3)
208 axes[1].axhline(y=0.25, color='red', linestyle='--', alpha=0.5)
209 axes[1].axhline(y=0.5, color='green', linestyle='--', alpha=0.5)
210 axes[1].axhline(y=0.75, color='blue', linestyle='--', alpha=0.5)
211
212 bars = axes[2].bar(age_labels, [np.sum(age_category == i) for i in
213                                 range(3)],
214                                 color=age_colors, edgecolor='black')
215 axes[2].set_title('          ')

```

```

215 axes[2].set_ylabel(' ')
216 for bar in bars:
217     height_val = bar.get_height()
218     axes[2].text(bar.get_x() + bar.get_width() / 2., height_val,
219                   f'{int(height_val)}\n{height_val / len(data) *'
220                   '100:.1f}%)',
221                   ha='center', va='bottom')
222 axes[2].grid(True, alpha=0.3, axis='y')
223
224 plt.tight_layout()
225 plt.savefig('lb2_age_analysis.png', dpi=300, bbox_inches='tight')
226 plt.show()
227
228 print(f"      : {len(data)}      , {len(numeric_features)}      ")
229 print(f"      :      ")
230 print(f"\n      n      :")
231 for s, label, count in zip(['M', 'F', 'I'], sex_labels_full,
232                           sex_counts):
233     print(f"  {label}: {count}      ({count/len(data)*100:.1f}%)")
234 print(f"\n      n      :")
235 for i, label in enumerate(age_labels):
236     count = np.sum(age_category == i)
237     print(f"  {label}: {count}      ({count/len(data)*100:.1f}%)")
238 print(f"\n      n      :")
239 for feature in numeric_features[:-1]:
240     corr = correlation_matrix.loc[feature, 'Rings']
241     print(f"  {feature}: {corr:.3f}")

```

Листинг 1.1 — Полный код программы для визуализации данных

## 1.6 Результаты анализа

- 1. Размер данных:** 4177 записей, 8 признаков (исключая целевую переменную)
- 2. Пропущенные значения:** отсутствуют
- 3. Распределение по полу:**
  - Самцы (M): 1472 особи (35.2%)
  - Самки (F): 1927 особей (46.1%)
  - Молодые (I): 778 особей (18.7%)
- 4. Распределение возрастных категорий:**

- Молодые (<=9 колец): 1344 особи (32.2%)
- Взрослые (10-15 колец): 1718 особей (41.1%)
- Старые (>=16 колец): 1115 особей (26.7%)

## 5. Корреляция признаков с возрастом:

- Shell weight: 0.628
- Diameter: 0.575
- Height: 0.557
- Length: 0.557
- Whole weight: 0.540
- Shucked weight: 0.421
- Viscera weight: 0.504

## 6. Основные наблюдения:

- Все весовые и размерные характеристики имеют правостороннюю асимметрию
- Самки в среднем крупнее и тяжелее самцов
- Сильная корреляция между размерными признаками
- Наибольшую корреляцию с возрастом имеет вес раковины
- Возрастные категории имеют значительное перекрытие в пространстве признаков