

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №1  
По дисциплине: «ОМО»

Тема: «Знакомство с анализом данных: предварительная обработка и  
визуализация.»

Выполнил:  
Студент 3-го курса  
Группы АС-66  
Ляшук В.И.  
Проверил:  
Крощенко А.А

Брест 2025

Цель: Получить практические навыки работы с данными с использованием библиотек Pandas для манипуляции и Matplotlib для визуализации. Научиться выполнять основные шаги предварительной обработки данных, такие как очистка, нормализация и работа с различными типами признаков.

## Вариант 8

Выборка Pima Indians Diabetes. Содержит медицинские показатели женщин из племени Пима и информацию о наличии у них диабета. Задачи: 1. Загрузите данные и выведите их статистические характеристики. 2. Проанализируйте столбцы Glucose, BloodPressure, SkinThickness. Нулевые значения в них, скорее всего, являются ошибками. Замените их медианным значением соответствующего столбца. 3. Постройте гистограммы для признаков BMI и Age. 4. Создайте матрицу корреляции только для признаков Glucose, BMI, Age и Outcome. ОСНОВЫ МАШИННОГО ОБУЧЕНИЯ, ЛР № 1, 2025 5. Визуализируйте распределение Outcome (наличие диабета) с помощью круговой диаграммы. 6. Примените стандартизацию ко всем признакам, кроме Outcome.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler

# Настройка отображения графиков
plt.rcParams['font.size'] = 12
plt.rcParams['figure.figsize'] = (10, 6)

# 1. ЗАГРУЗКА ДАННЫХ
print("=" * 50)
print("1. ЗАГРУЗКА ДАННЫХ")
print("=" * 50)

df = pd.read_csv('../..//pima-indians-diabetes.csv', sep=',', comment='#')

column_names = [
    'Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
    'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'
]
df.columns = column_names

print("Первые 5 строк данных:")
print(df.head())
print("\nИнформация о данных:")
print(df.info())

# 2. СТАТИСТИЧЕСКИЙ АНАЛИЗ
print("\n" + "=" * 50)
print("2. СТАТИСТИЧЕСКИЕ ХАРАКТЕРИСТИКИ")
```

```

print("=" * 50)

print("Основные статистические характеристики:")
print(df.describe())

# Проверка на пропущенные значения (в данном наборе пропуски обозначены как 0 в некоторых столбцах)
print("\nКоличество нулевых значений в каждом столбце:")
for column in df.columns:
    zero_count = (df[column] == 0).sum()
    print(f"{column}: {zero_count} нулевых значений")

# 3. ОБРАБОТКА ПРОПУЩЕННЫХ ЗНАЧЕНИЙ
print("\n" + "=" * 50)
print("3. ОБРАБОТКА ПРОПУЩЕННЫХ ЗНАЧЕНИЙ")
print("=" * 50)

# Столбцы, где 0 является некорректным значением
columns_to_fix = ['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']

# Замена нулевых значений на медианные (исключая нули при расчете медианы)
for column in columns_to_fix:
    median_value = df[df[column] != 0][column].median()
    df[column] = df[column].replace(0, median_value)
    print(f"Столбец {column}: заменено {(df[column] == 0).sum()} значений на медиану {median_value:.2f}")

print("\nСтатистика после обработки пропусков:")
print(df[columns_to_fix].describe())

# 4. ВИЗУАЛИЗАЦИЯ ДАННЫХ
print("\n" + "=" * 50)
print("4. ВИЗУАЛИЗАЦИЯ ДАННЫХ")
print("=" * 50)

# Создание подграфиков
fig, axes = plt.subplots(2, 2, figsize=(15, 12))

# 4.1 Гистограмма BMI
axes[0, 0].hist(df['BMI'], bins=20, color='skyblue', edgecolor='black', alpha=0.7)
axes[0, 0].set_title('Распределение индекса массы тела (BMI)')
axes[0, 0].set_xlabel('BMI')
axes[0, 0].set_ylabel('Частота')

# 4.2 Гистограмма Age
axes[0, 1].hist(df['Age'], bins=20, color='lightgreen', edgecolor='black', alpha=0.7)
axes[0, 1].set_title('Распределение возраста (Age)')
axes[0, 1].set_xlabel('Возраст (лет)')
axes[0, 1].set_ylabel('Частота')

# 4.3 Круговая диаграмма Outcome
outcome_counts = df['Outcome'].value_counts()
labels = ['Нет диабета', 'Есть диабет']
colors = ['lightblue', 'lightcoral']
axes[1, 0].pie(outcome_counts, labels=labels, colors=colors, autopct='%.1f%%', startangle=90)

```

```

axes[1, 0].set_title('Распределение наличия диабета')

# 4.4 Матрица корреляции
correlation_columns = ['Glucose', 'BMI', 'Age', 'Outcome']
correlation_matrix = df[correlation_columns].corr()
im = axes[1, 1].imshow(correlation_matrix, cmap='coolwarm', aspect='auto', vmin=-1, vmax=1)
axes[1, 1].set_xticks(range(len(correlation_columns)))
axes[1, 1].set_yticks(range(len(correlation_columns)))
axes[1, 1].set_xticklabels(correlation_columns)
axes[1, 1].set_yticklabels(correlation_columns)

# Добавление значений корреляции на тепловую карту
for i in range(len(correlation_columns)):
    for j in range(len(correlation_columns)):
        text = axes[1, 1].text(j, i, f'{correlation_matrix.iloc[i, j]:.2f}', ha="center", va="center", color="black")

axes[1, 1].set_title('Матрица корреляции')

plt.tight_layout()
plt.show()

# 5. ДОПОЛНИТЕЛЬНАЯ ВИЗУАЛИЗАЦИЯ
print("\n" + "=" * 50)
print("5. ДОПОЛНИТЕЛЬНАЯ ВИЗУАЛИЗАЦИЯ")
print("=" * 50)

# Диаграмма рассеяния: Glucose vs BMI с цветом по Outcome
plt.figure(figsize=(10, 6))
scatter = plt.scatter(df['Glucose'], df['BMI'], c=df['Outcome'],
                      cmap='viridis', alpha=0.6)
plt.colorbar(scatter, label='Outcome (0=Нет, 1=Да)')
plt.xlabel('Уровень глюкозы (Glucose)')
plt.ylabel('Индекс массы тела (BMI)')
plt.title('Зависимость BMI от уровня глюкозы')
plt.grid(True, alpha=0.3)
plt.show()

# 6. СТАНДАРТИЗАЦИЯ ДАННЫХ
print("\n" + "=" * 50)
print("6. СТАНДАРТИЗАЦИЯ ДАННЫХ")
print("=" * 50)

# Признаки для стандартизации (все кроме Outcome)
features_to_standardize = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',
                           'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']

df_original = df.copy()

# Стандартизация
scaler = StandardScaler()
df_standardized = df.copy()
df_standardized[features_to_standardize] = scaler.fit_transform(df[features_to_standardize])

```

```

print("Данные после стандартизации (первые 5 строк):")
print(df_standardized[features_to_standardize].head())

# Визуализация распределения до и после стандартизации
fig, axes = plt.subplots(1, 2, figsize=(15, 6))

# До стандартизации
axes[0].hist(df_original['Glucose'], bins=20, color='skyblue', edgecolor='black', alpha=0.7, label='Исходные')
axes[0].set_title('Распределение Glucose до стандартизации')
axes[0].set_xlabel('Glucose')
axes[0].set_ylabel('Частота')

# После стандартизации
axes[1].hist(df_standardized['Glucose'], bins=20, color='lightcoral', edgecolor='black', alpha=0.7,
label='Стандартизованные')
axes[1].set_title('Распределение Glucose после стандартизации')
axes[1].set_xlabel('Glucose (стандартизованный)')
axes[1].set_ylabel('Частота')

plt.tight_layout()
plt.show()

# 7. ВЫВОДЫ И АНАЛИЗ
print("\n" + "=" * 50)
print("7. ВЫВОДЫ И АНАЛИЗ")
print("=" * 50)

print("Ключевые наблюдения:")
print("1. Размер набора данных:", df.shape)
print("2. Распределение классов:")
print(f" - Без диабета: {outcome_counts[0]} случаев ({outcome_counts[0]/len(df)*100:.1f}%)")
print(f" - С диабетом: {outcome_counts[1]} случаев ({outcome_counts[1]/len(df)*100:.1f}%)")

# Анализ корреляции
correlation_with_outcome = df[correlation_columns].corr()['Outcome'].sort_values(ascending=False)
print("\n3. Корреляция признаков с Outcome:")
for feature, corr in correlation_with_outcome.items():
    if feature != 'Outcome':
        print(f" - {feature}: {corr:.3f}")

print("\n4. Статистика по возрасту:")
print(f" - Средний возраст: {df['Age'].mean():.1f} лет")
print(f" - Медианный возраст: {df['Age'].median():.1f} лет")
print(f" - Минимальный возраст: {df['Age'].min()} лет")
print(f" - Максимальный возраст: {df['Age'].max()} лет")

print("\n5. Статистика по BMI:")
print(f" - Средний BMI: {df['BMI'].mean():.1f}")
print(f" - Медианный BMI: {df['BMI'].median():.1f}")

# Сохранение обработанных данных
df_standardized.to_csv('pima_indians_diabetes_processed.csv', index=False)

```

```
print("\nОбработанные данные сохранены в файл 'pima_diabetics_processed.csv'")
```

---

## 1. ЗАГРУЗКА ДАННЫХ

---

Первые 5 строк данных:

|   | Pregnancies | Glucose | BloodPressure | ... DiabetesPedigreeFunction | Age | Outcome |
|---|-------------|---------|---------------|------------------------------|-----|---------|
| 0 | 1           | 85      | 66 ...        | 0.351                        | 31  | 0       |
| 1 | 8           | 183     | 64 ...        | 0.672                        | 32  | 1       |
| 2 | 1           | 89      | 66 ...        | 0.167                        | 21  | 0       |
| 3 | 0           | 137     | 40 ...        | 2.288                        | 33  | 1       |
| 4 | 5           | 116     | 74 ...        | 0.201                        | 30  | 0       |

[5 rows x 9 columns]

Информация о данных:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 767 entries, 0 to 766
Data columns (total 9 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   Pregnancies      767 non-null   int64  
 1   Glucose          767 non-null   int64  
 2   BloodPressure    767 non-null   int64  
 3   SkinThickness    767 non-null   int64  
 4   Insulin          767 non-null   int64  
 5   BMI              767 non-null   float64 
 6   DiabetesPedigreeFunction 767 non-null   float64 
 7   Age              767 non-null   int64  
 8   Outcome          767 non-null   int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
None
```

---

## 2. СТАТИСТИЧЕСКИЕ ХАРАКТЕРИСТИКИ

---

Основные статистические характеристики:

```
Pregnancies   Glucose ...   Age   Outcome
count    767.000000 767.000000 ... 767.000000 767.000000
mean     3.842243 120.859192 ... 33.219035 0.348110
std      3.370877 31.978468 ... 11.752296 0.476682
min      0.000000 0.000000 ... 21.000000 0.000000
25%     1.000000 99.000000 ... 24.000000 0.000000
```

```
50%    3.000000 117.000000 ... 29.000000  0.000000
75%    6.000000 140.000000 ... 41.000000  1.000000
max    17.000000 199.000000 ... 81.000000  1.000000
```

[8 rows x 9 columns]

Количество нулевых значений в каждом столбце:

Pregnancies: 111 нулевых значений

Glucose: 5 нулевых значений

BloodPressure: 35 нулевых значений

SkinThickness: 227 нулевых значений

Insulin: 373 нулевых значений

BMI: 11 нулевых значений

DiabetesPedigreeFunction: 0 нулевых значений

Age: 0 нулевых значений

Outcome: 500 нулевых значений

---

### 3. ОБРАБОТКА ПРОПУЩЕННЫХ ЗНАЧЕНИЙ

---

Столбец Glucose: заменено 0 значений на медиану 117.00

Столбец BloodPressure: заменено 0 значений на медиану 72.00

Столбец SkinThickness: заменено 0 значений на медиану 29.00

Столбец Insulin: заменено 0 значений на медиану 125.00

Столбец BMI: заменено 0 значений на медиану 32.30

Статистика после обработки пропусков:

|       | Glucose    | BloodPressure | SkinThickness | Insulin    | BMI        |
|-------|------------|---------------|---------------|------------|------------|
| count | 767.000000 | 767.000000    | 767.000000    | 767.000000 | 767.000000 |
| mean  | 121.621904 | 72.387223     | 29.100391     | 140.692308 | 32.453716  |
| std   | 30.443252  | 12.104527     | 8.794378      | 86.437570  | 6.879539   |
| min   | 44.000000  | 24.000000     | 7.000000      | 14.000000  | 18.200000  |
| 25%   | 99.500000  | 64.000000     | 25.000000     | 121.000000 | 27.500000  |
| 50%   | 117.000000 | 72.000000     | 29.000000     | 125.000000 | 32.300000  |
| 75%   | 140.000000 | 80.000000     | 32.000000     | 127.500000 | 36.600000  |
| max   | 199.000000 | 122.000000    | 99.000000     | 846.000000 | 67.100000  |

---

---

### 6. СТАНДАРТИЗАЦИЯ ДАННЫХ

---

Данные после стандартизации (первые 5 строк):

|   | Pregnancies | Glucose   | ... | DiabetesPedigreeFunction | Age       |
|---|-------------|-----------|-----|--------------------------|-----------|
| 0 | -0.843726   | -1.203741 | ... | -0.364265                | -0.188940 |
| 1 | 1.234240    | 2.017463  | ... | 0.604701                 | -0.103795 |

|   |           |           |     |           |           |
|---|-----------|-----------|-----|-----------|-----------|
| 2 | -0.843726 | -1.072264 | ... | -0.919684 | -1.040393 |
| 3 | -1.140579 | 0.505469  | ... | 5.482732  | -0.018650 |
| 4 | 0.343683  | -0.184789 | ... | -0.817052 | -0.274086 |

## ===== 7. ВЫВОДЫ И АНАЛИЗ =====

Ключевые наблюдения:

1. Размер набора данных: (767, 9)

2. Распределение классов:

- Без диабета: 500 случаев (65.2%)
- С диабетом: 267 случаев (34.8%)

3. Корреляция признаков с Outcome:

- Glucose: 0.492
- BMI: 0.312
- Age: 0.236

4. Статистика по возрасту:

- Средний возраст: 33.2 лет
- Медианный возраст: 29.0 лет
- Минимальный возраст: 21 лет
- Максимальный возраст: 81 лет

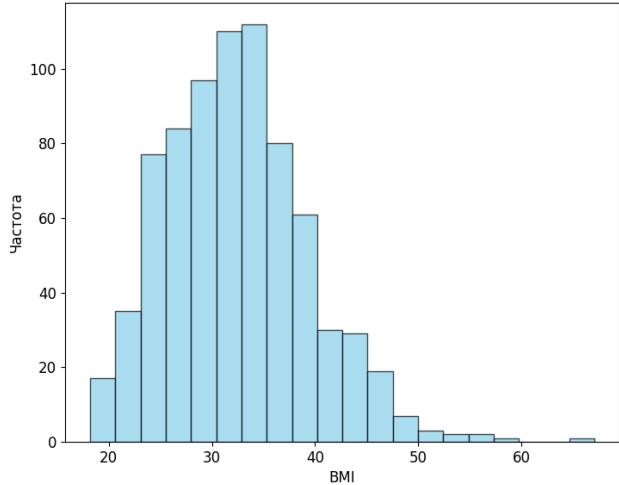
5. Статистика по BMI:

- Средний BMI: 32.5
- Медианный BMI: 32.3

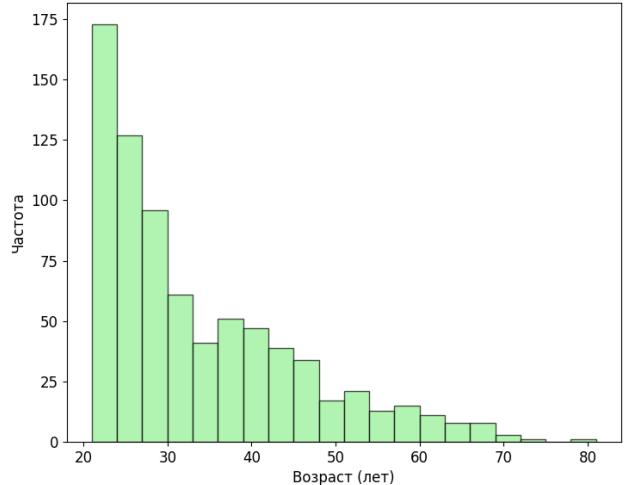
Обработанные данные сохранены в файл 'pima\_indians\_diabetes\_processed.csv'

## Графики:

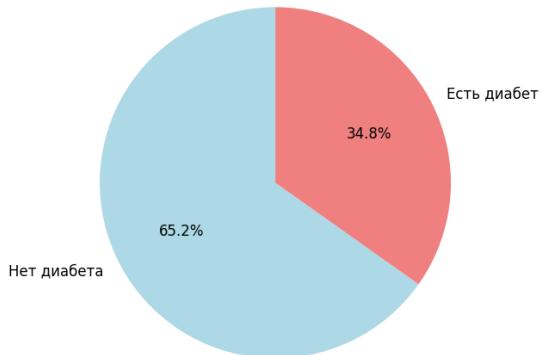
Распределение индекса массы тела (BMI)



Распределение возраста (Age)

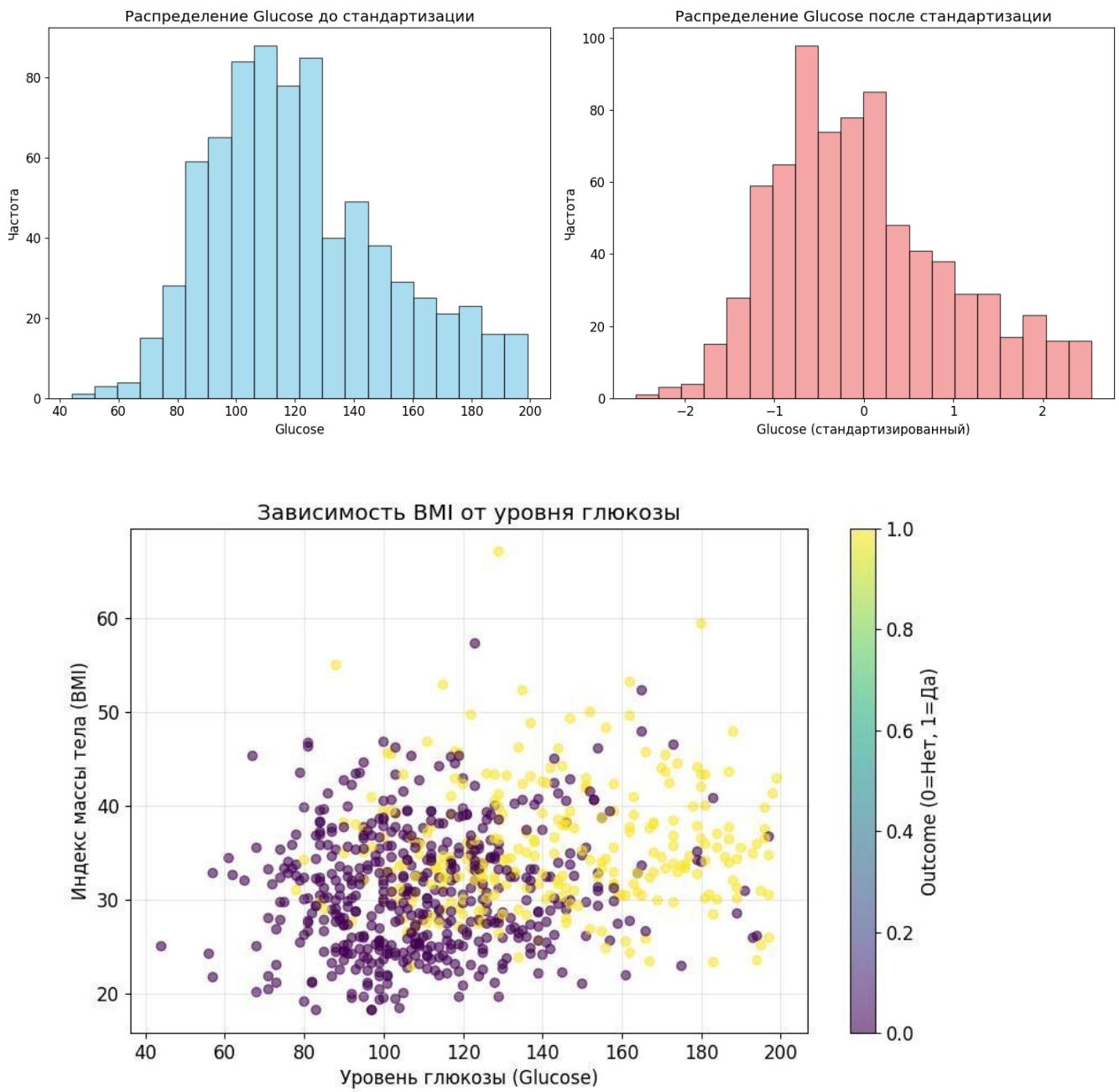


Распределение наличия диабета



Матрица корреляции

|         | Glucose | BMI  | Age  | Outcome |
|---------|---------|------|------|---------|
| Glucose | 1.00    | 0.23 | 0.27 | 0.49    |
| BMI     | 0.23    | 1.00 | 0.03 | 0.31    |
| Age     | 0.27    | 0.03 | 1.00 | 0.24    |
| Outcome | 0.49    | 0.31 | 0.24 | 1.00    |



**Вывод:** в результате выполнения данной лабораторной работы получили практические навыки работы с данными с использованием библиотек Pandas для манипуляции и Matplotlib для визуализации. Научились выполнять основные шаги предварительной обработки данных, такие как очистка, нормализация и работа с различными типами признаков.