**Министерство образования и науки Российской Федерации**

**Российский государственный университет нефти и газа**

**(национальный исследовательский университет)**

**имени И.М. Губкина**

**Кафедра информатики**

Отчет по Домашнему заданию №1 («Задача регрессии»)

Дисциплины ***Средства интеллектуального анализа данных и машинное обучение***

**Выполнил:**

Студент группы АА-22-08

Сафуанов Артур

**Проверила:**

Доцент кафедры информатики, к.ф.м.н.

Вишневская Елена Александровна

Москва, 2025 г.

Оглавление

[Введение 3](#_Toc167825323)

[Ход работы 4](#_Toc167825324)

[Список используемой литературы 22](#_Toc167825325)

**Введение**

Современный этап развития науки и технологий характеризуется стремительным ростом объёмов данных, что делает актуальным применение методов анализа информации для решения прикладных задач. Регрессионный анализ, как один из ключевых инструментов машинного обучения и статистики, позволяет выявлять зависимости между переменными, прогнозировать значения целевых показателей и принимать обоснованные решения на основе данных. В контексте Data Science умение работать с данными, проводить их подготовку, исследование и строить предсказательные модели становится критически важным навыком.

Целью данной работы является освоение основных этапов исследовательского анализа данных (EDA) и применение методов регрессионного моделирования на практике.

**Ход работы**

**Задание 1. Подобрать из открытых источников или смоделировать набор данных для анализа.**

Решение:

Kaggle — это одна из самых популярных платформ для специалистов в области анализа данных, машинного обучения и искусственного интеллекта. Она предоставляет доступ к огромному количеству открытых наборов данных, а также позволяет участвовать в соревнования, изучать готовые решения и делиться своими наработками. Kaggle является отличным ресурсом для поиска данных, которые можно использовать в учебных и исследовательских целях.

Для выполнения задания 1 был выбран датасет possum.csv, доступный на платформе Kaggle. Этот набор данных содержит информацию о морфологических характеристиках опоссумов и включает 14 признаков, таких как длина головы, хвоста, возраст и пол. Датасет идеально подходит для задач регрессии и классификации, а также для отработки навыков исследовательского анализа данных.

**Признаки (переменные):**

1. **case** — идентификатор образца (уникальный номер).
2. **site** — место, где был пойман опоссум (1-7).
3. **Pop** — популяция (Vic или other).
4. **sex** — пол опоссума (m — мужской, f — женский).
5. **age** — возраст опоссума (в годах).
6. **hdlngth** — длина головы (в мм).
7. **skullw** — ширина черепа (в мм).
8. **totlngth** — общая длина тела (в см).
9. **taill** — длина хвоста (в см).
10. **footlgth** — длина ступни (в мм).
11. **earconch** — длина ушной раковины (в мм).
12. **eye** — расстояние между глазами (в мм).
13. **chest** — обхват груди (в см).
14. **belly** — обхват живота (в см).

**Задание 2. Ознакомиться с возможностями получения и загрузки данных из различных источников.**

Решение:

**Основные источники данных:**

1. **Локальные файлы:**
   * **CSV (Comma-Separated Values):** Текстовый формат, где данные разделены запятыми или другими разделителями.
   * **Excel (XLSX):** Табличный формат, поддерживающий несколько листов и сложные структуры.
   * **JSON (JavaScript Object Notation):** Формат для хранения и передачи структурированных данных, часто используемый в веб-приложениях.
   * **Текстовые файлы (TXT):** Простые файлы, содержащие неструктурированные данные.
2. **Базы данных:**
   * Данные могут храниться в реляционных базах данных (например, MySQL, PostgreSQL) или NoSQL-базах (например, MongoDB).
   * Для работы с базами данных используются специализированные библиотеки и языки запросов (SQL).
3. **Веб-источники:**
   * Данные можно загружать с веб-сайтов с помощью **веб-скрапинга** (например, с использованием библиотек BeautifulSoup или Scrapy в Python).
   * Многие сервисы предоставляют доступ к данным через **API** (Application Programming Interface), например, Twitter API, Google Maps API и др.
4. **Специализированные платформы:**
   * Платформы, такие как **Kaggle**, **UCI Machine Learning Repository** или **Google Dataset Search**, предоставляют доступ к открытым наборам данных для анализа и машинного обучения.

**Задание 3. Изучить методы импорта данных из файлов различных форматов, способы обработки некорректных значений.**

Решение:

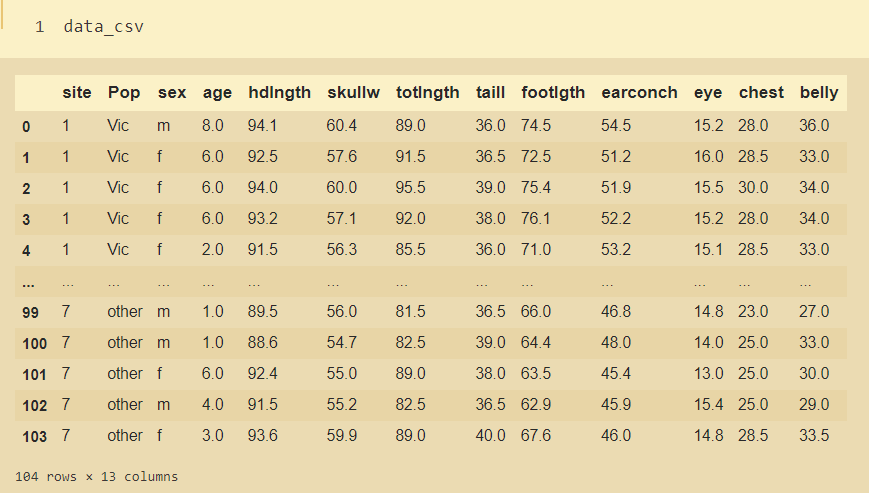
data\_csv = pd.read\_csv('possum.csv')

data\_excel = pd.read\_excel('possum.xls')

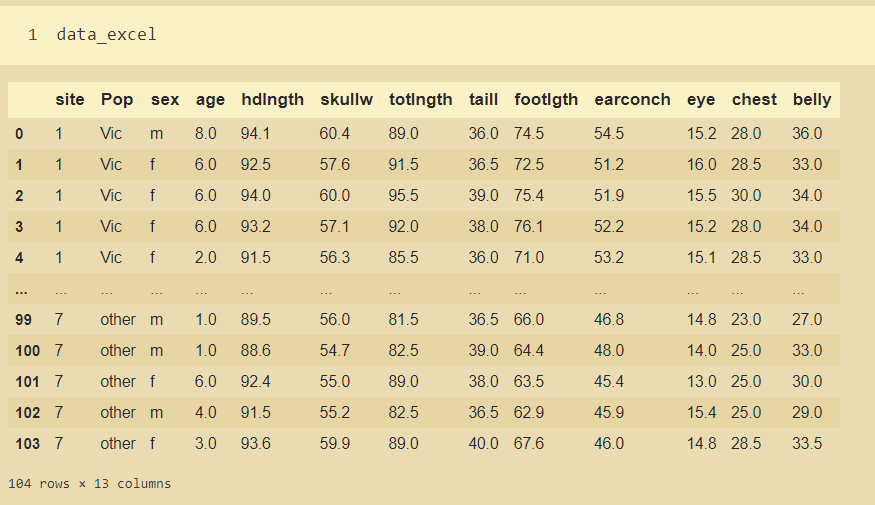
data\_csv, data\_excel = data\_csv.drop('case', axis=1), data\_excel.drop('case', axis=1)

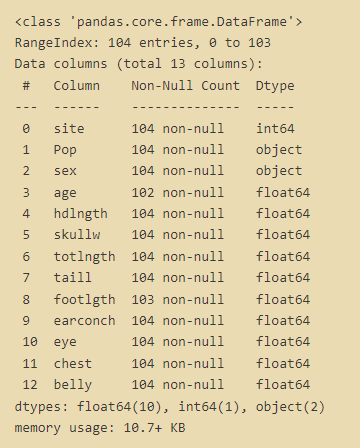
data\_csv.info()

Сначала импортируем в csv формате:



Затем выводится такая же таблица с эксель файла:



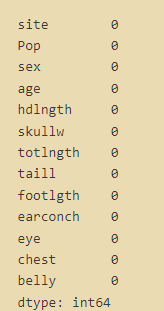


Методы считывания данных с файлов различных форматов изучены.

Теперь приступим к обработке некорректных значений с помощью встроенных методов pandas:

data\_csv.dropna(inplace=True)

data\_csv.isnull().sum()



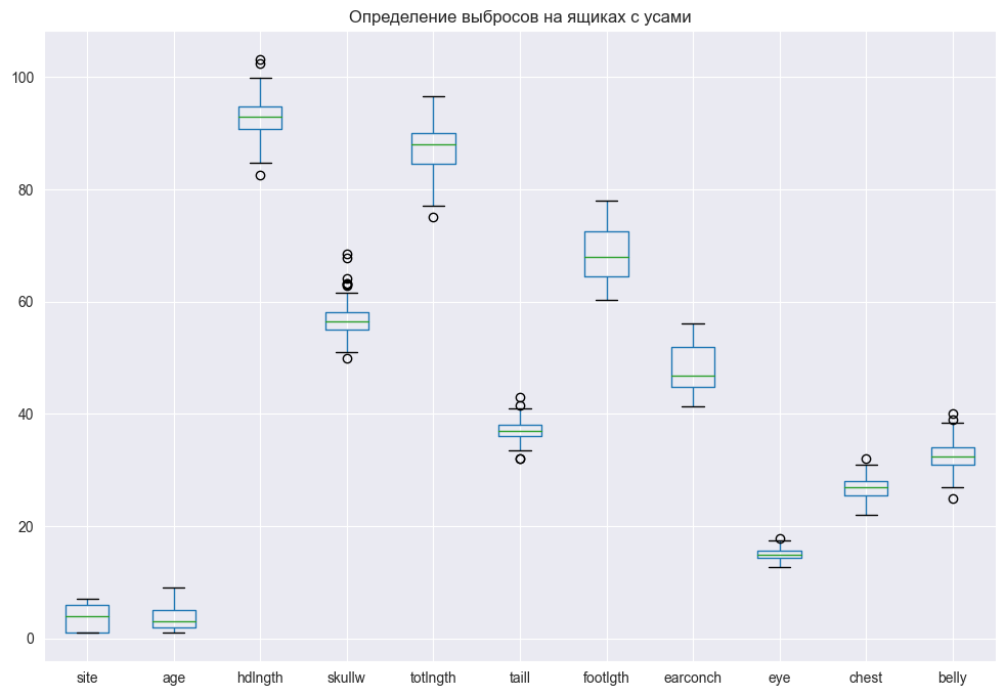
Таким образом все пропущенные значения были исключены из датасета.

Сейчас воспользуемся методом IQR, чтобы избавиться от выбросов в данных, но для начала мы рассмотрим распределения каждой величины с помощью boxplots:

data\_csv.boxplot(figsize=(12,8))

plt.title('Определение выбросов на ящиках с усами')

plt.show()



Далее пропишем код для удаления выбросов:

for col in data\_csv.columns:

if col not in ['Pop', 'sex']:

print(f'Обработали выбросы в {col}')

Q1 = np.percentile(data\_csv[col], 25)

Q3 = np.percentile(data\_csv[col], 75)

IQR = Q3 - Q1

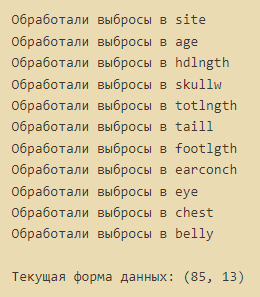
high = Q3 + 1.5 \* IQR

low = Q1 - 1.5 \* IQR

data\_csv = data\_csv[(data\_csv[col] >= low) & (data\_csv[col] <= high)]

print()

print(f'Текущая форма данных: {data\_csv.shape}')



Как видим выбросы были обработаны, так как датасет уменьшился в размере.

**Задание 4 – 5. Провести одномерный и двумерный описательный анализ. Исследовать данные, провести одномерный описательный анализ.**

Решение:

Начнём с одномерного анализа данных и приведём соответствующие описательные характеристики и графики:

print('Для каждого столбца указывается среднее арифметическое, стандартное отклонение, ' \

'минимум и максимум, процентили:')

data\_csv.describe()



В таблице указаны основные показатели, учитывающиеся при одномерном анализе данных.

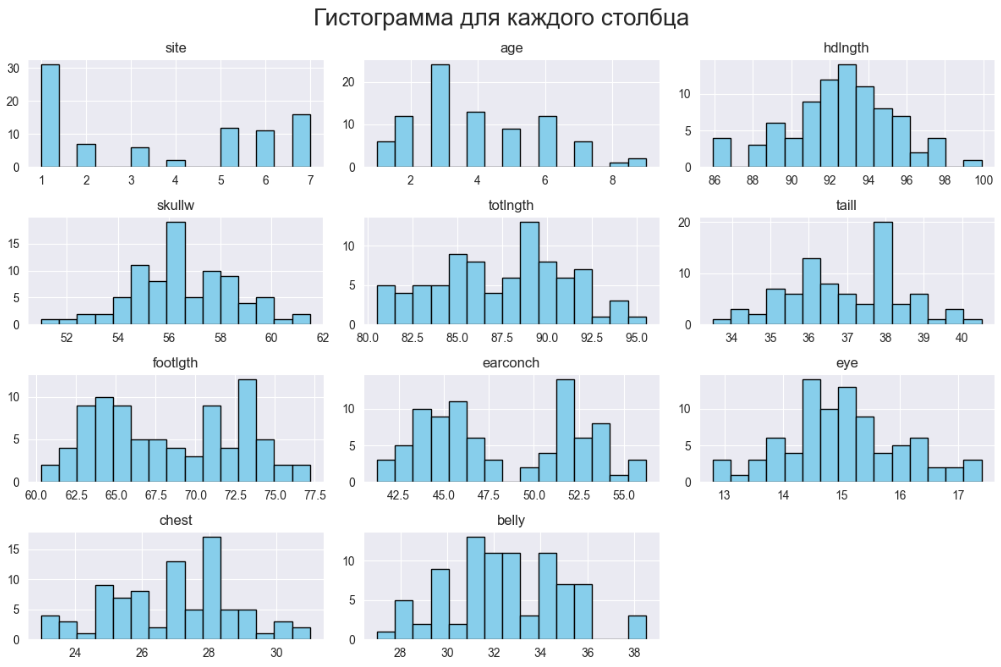
Перейдём к визуализации столбцов:

data\_csv.hist(figsize=(12, 8), edgecolor='black', bins=15, color='skyblue')

plt.suptitle('Гистограмма для каждого столбца', fontsize=20)

plt.tight\_layout()

plt.show()



Здесь можем видеть, что в датасете присутствуют как дискретные данные, так и непрерывные.

Посмотрим на графики категориальных признаков:

plt.figure(figsize=(12,8),dpi=200)

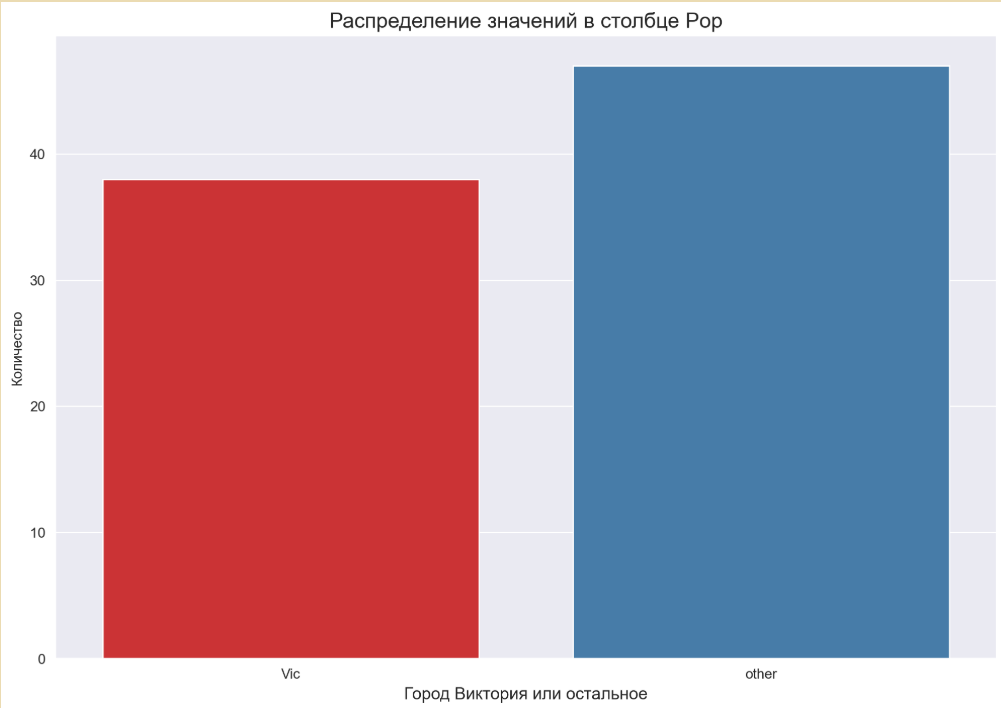
sns.countplot(data=data\_csv, x='Pop', palette='Set1')

plt.title('Распределение значений в столбце Pop', fontsize=15)

plt.ylabel('Количество')

plt.xlabel('Город Виктория или остальное', fontsize=12)

plt.show()



Видим, что опоссумов с других территорий больше, чем с города Виктория.

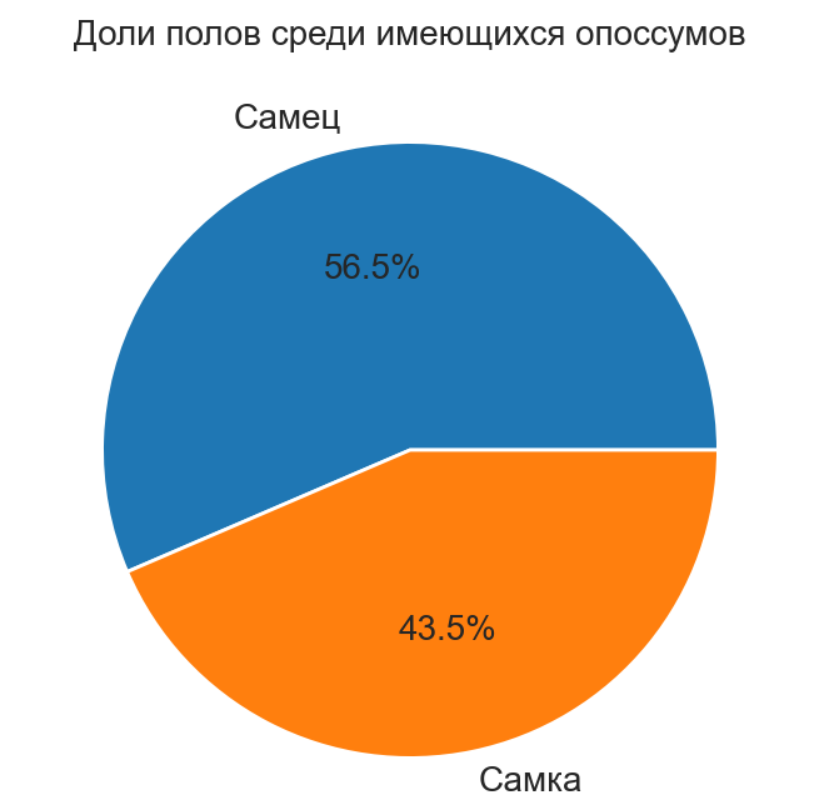
Далее построим pieplot по полу опоссумов:

pie\_data = data\_csv['sex'].value\_counts()

plt.figure(figsize=(8,4),dpi=200)

plt.pie(pie\_data.values, labels=['Самец', 'Самка'], autopct='%1.1f%%');

plt.title('Доли полов среди имеющихся опоссумов', fontsize=10);



Можем сделать вывод, что в датасете самцов больше, чем самок.

Приступим к двумерному описательному анализу и посмотрим на матрицу корреляций:

print('Матрица корреляций:')

data\_csv.corr()



Также рассмотрим диаграммы рассеяния для лучшего понимания зависимостей в данных:

plt.figure(figsize=(12,8), dpi=200)

legend\_map = {'m': 'Самец',

'f': 'Самка'}

sns.scatterplot(data=data\_csv, x='hdlngth', y='totlngth', hue=data\_csv['sex'].map(legend\_map))

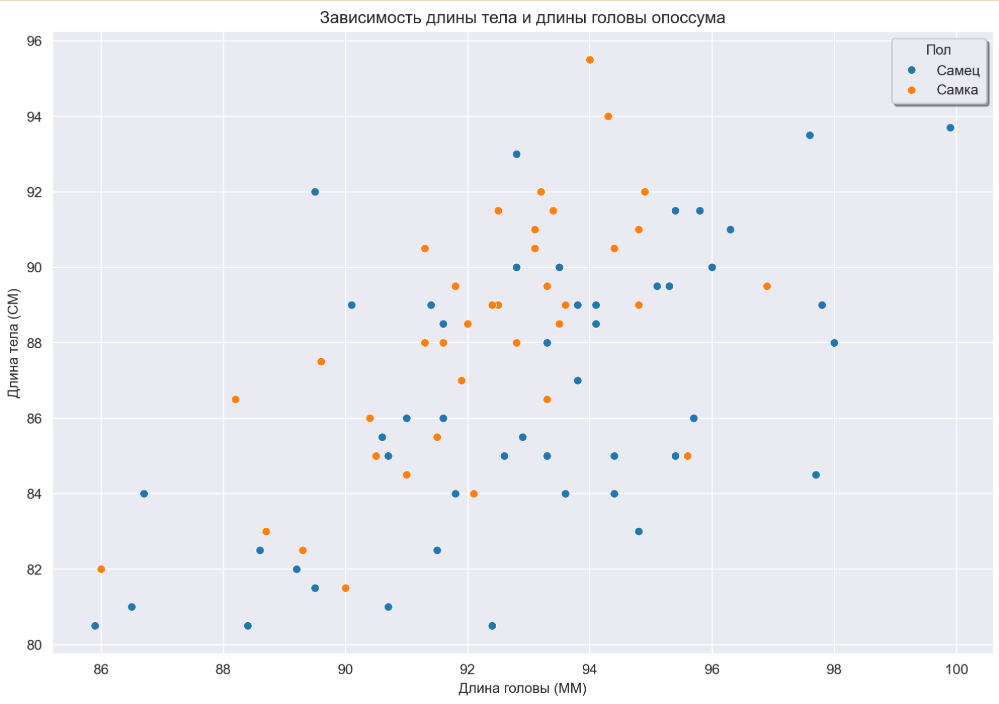
plt.title('Зависимость длины тела и длины головы опоссума')

plt.xlabel('Длина головы (ММ)')

plt.ylabel('Длина тела (СМ)')

plt.legend(title='Пол', shadow=True, frameon=True)

plt.show()



Видим, что, например, длина головы и тела имеют положительную корреляцию. Другими словами, чем больше длина головы, тем больше длина тела в целом.

Рассмотрим ещё одну диаграмму:

plt.figure(figsize=(12,8), dpi=200)

legend\_map = {'m': 'Самец',

'f': 'Самка'}

sns.scatterplot(data=data\_csv, x='taill', y='earconch', hue=data\_csv['sex'].map(legend\_map),

palette='viridis')

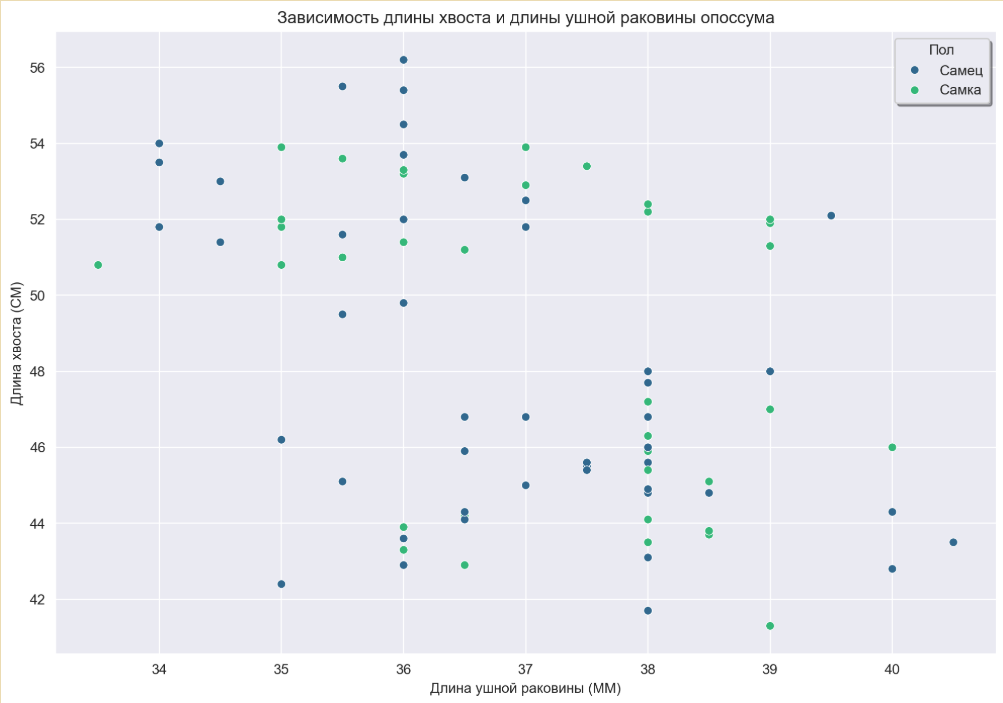
plt.title('Зависимость длины хвоста и длины ушной раковины опоссума')

plt.xlabel('Длина ушной раковины (ММ)')

plt.ylabel('Длина хвоста (СМ)')

plt.legend(title='Пол', shadow=True, frameon=True)

plt.show()



Здесь уже мы имеем дело с отрицательной корреляцией. И как это ни странно, с уменьшением длины ушной раковины, длина хвоста увеличивается.

**Задание 6. Провести разведочный двумерный анализ данных с целью выявления зависимостей.**

Решение:

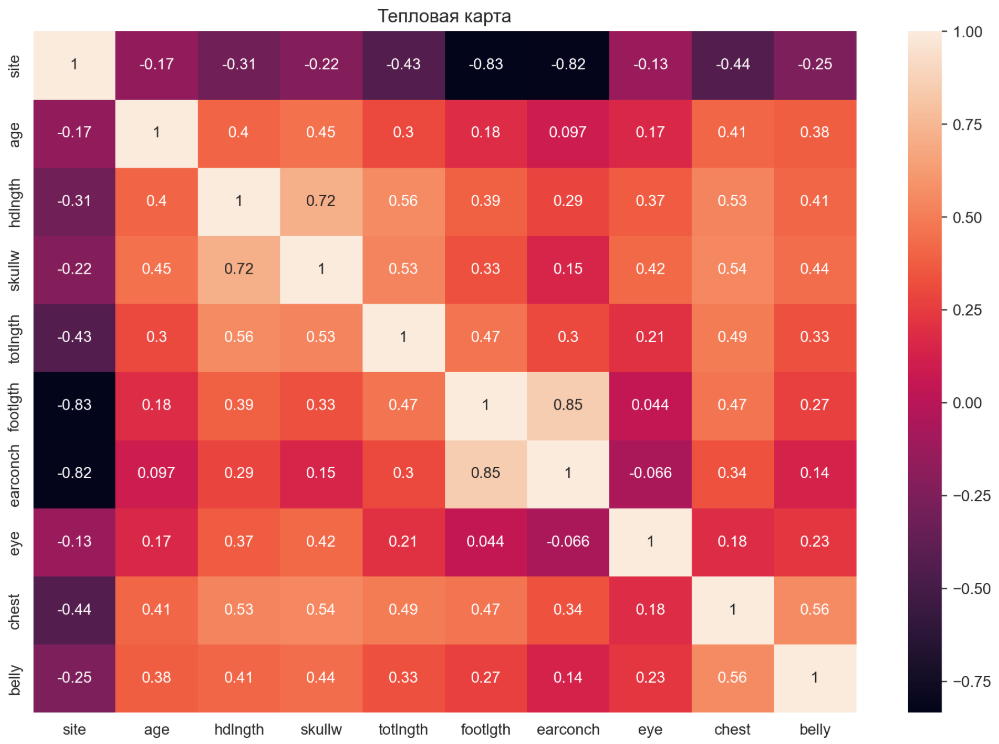
Для определения зависимостей между признаками построим heatmap и удалим мультиколлинеарные признаки:

data\_corr = data\_csv.drop(['taill', 'Pop', 'sex'], axis=1).corr()

plt.figure(figsize=(12,8), dpi=200)

sns.heatmap(data=data\_corr, annot=True, cmap='rocket')

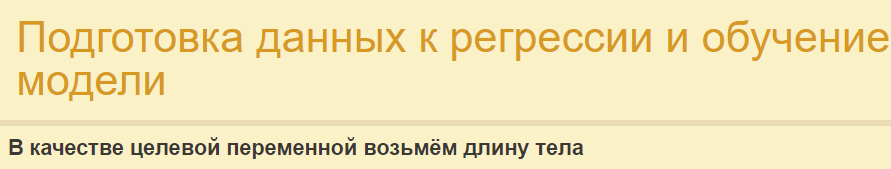
plt.show()

data\_csv = data\_csv.drop('earconch', axis=1)

**Задание 7. Подготовить данные и построить линейную регрессионную модель, оценить точность.**

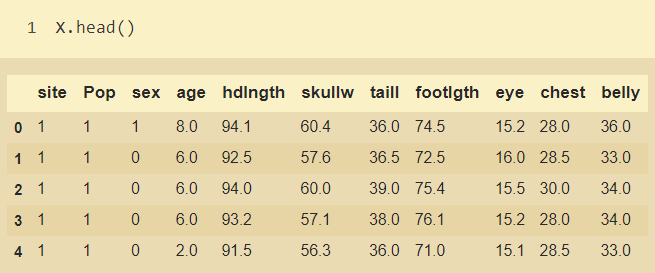
Решение:



Предобработаем категориальные признаки путём присваивания им числовых значений: 0 и 1.

X['Pop'] = X['Pop'].map({'Vic': 1, 'other': 0})

X['sex'] = X['sex'].map({'m': 1, 'f': 0})



Теперь распишем пайплайн, в который включены сразу масштабирование и линейная регрессия. Таким образом, мы не будем отдельно обрабатывать данные:

pipe = Pipeline([

('scale', StandardScaler()),

('model', LinearRegression())

])

Теперь разделим выборку на тренировочную и тестовую:

X = data\_csv.drop('totlngth', axis=1)

y = data\_csv[['totlngth']]

Так как данных для обучения достаточно мало, то необходимо выполнить кросс-валидацию:

mse = cross\_val\_score(pipe, X, y, cv=5, scoring='neg\_mean\_squared\_error')

mae = cross\_val\_score(pipe, X, y, cv=5, scoring='neg\_mean\_absolute\_error')

r2 = cross\_val\_score(pipe, X, y, cv=5, scoring='r2')

Выведем все регрессионные метрики в виде датасета:

pd.DataFrame(data=[-mse, -mae, r2, np.sqrt(-mse)], index=['MSE', 'MAE', 'R2', 'RMSE']).transpose()



Кросс-валидация проводилась, используя 5 циклов, поэтому и получилось 5 значений. Наконец, усредним значения, чтобы получить истинные результаты:

pd.DataFrame(data=[(-mse).mean(), (-mae).mean(), r2.mean(), (np.sqrt(-mse)).mean()],

index=['MSE', 'MAE', 'R2', 'RMSE'],

columns=['Среднее значение'])

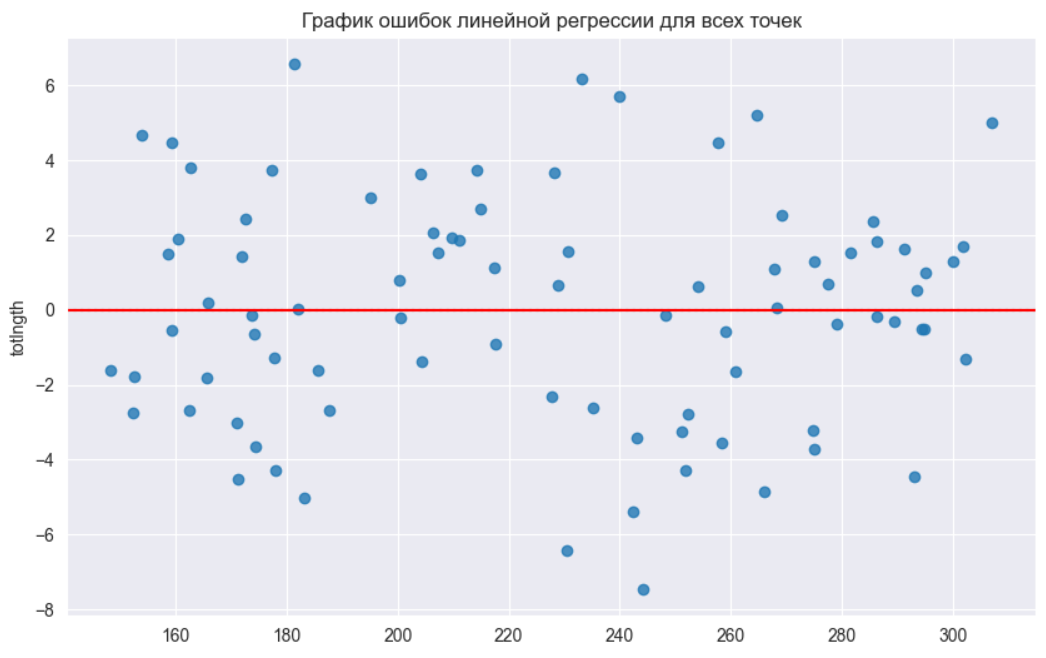


Таким образом, мы получили RMSE = 2.187, что означает небольшое отклонение относительно общей длины тела. В пересчёте на реальные цифры, модель ошибается в среднем на 2 см в предсказании, когда опоссумы имеют среднюю длину примерно 1 м.

Также выведем коэффициенты множественной регрессии и коэффициент пересечения в таблице:



И добавлю график ошибок, чтобы лучше интерпретировать обучение:



**Список используемой литературы**

1. **Pandas:**
   * Официальная документация Pandas.
   * McKinney, W. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython. — 2nd Edition. — O'Reilly Media, 2017. — 544 p.
2. **Scikit-learn:**
   * Официальная документация Scikit-learn.
   * Pedregosa, F., et al. Scikit-learn: Machine Learning in Python // Journal of Machine Learning Research. — 2011. — Vol. 12. — P. 2825–2830.
3. **Seaborn:**
   * Официальная документация Seaborn.
   * Waskom, M. L. seaborn: statistical data visualization // Journal of Open Source Software. — 2021. — Vol. 6, № 60. — P. 3021.
4. **Matplotlib:**
   * Официальная документация Matplotlib.
   * Hunter, J. D. Matplotlib: A 2D Graphics Environment // Computing in Science & Engineering. — 2007. — Vol. 9, № 3. — P. 90–95.
5. **Общие ресурсы по Python и анализу данных:**
   * VanderPlas, J. Python Data Science Handbook: Essential Tools for Working with Data. — O'Reilly Media, 2016. — 548 p.