Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №2**

По дисциплине: «Анализ больших данных»

Тема: «**Исследовательский анализ данных. Постановка гипотез Категориальные данные**»

|  |
| --- |
| Выполнил:  студент группы  Б.ПИН.РИС - 21.06  Иванов А.В. |
| Проверила:  старший преподаватель  кафедры ПО  Корнеева Е.И. |

Тверь 2025

# Формулировка задач, описание условий

**Общее задание на работу**

1. Ознакомьтесь с набором данных mpg из библиотеки Seaborn.

(загрузка через df = sns.load\_dataset(’mpg’))

1. Посчитайте количество строк и столбцов.
2. Проведите разведочный анализ, то есть:

(a) для каждой числовой переменной вычислите:

• Долю пропусков

• Максимальное и минимальное значение

• Среднее значение

• Медиану

• Дисперсию

• Квантиль 0.1 и 0.9

• Квартиль 1 и 3

(b) для каждой категориальной переменной вычислите:

• Долю пропусков

• Количество уникальных значений

• Моду

1. Сформулируйте и проверьте минимум 2 статистические гипотезы. Выбор критериев для проверки гипотез требуется обосновать. Сделать выводы в терминах предметной области.
2. Закодируйте категориальные переменные, необходимые для анализа, если требуется. Методом OneHotEncoding или LabelEncoding.
3. Постройте таблицу корреляции признаков и целевого столбца. Обоснуйте, какой столбец является целевым, а какие признаками.
4. Реализуйте стохастический и обычный градиентный спуск вручную, можно использовать ноутбук с лекции ссылка. Для этих данных: y = ’mpg’ и x = ’horsepower’ или ’weight’.

# Вариант и условия задач. Указать какой сложности выполняется задание.

Вариант 5.

**Сложность: Rare**

* Реализовать только общую часть задания

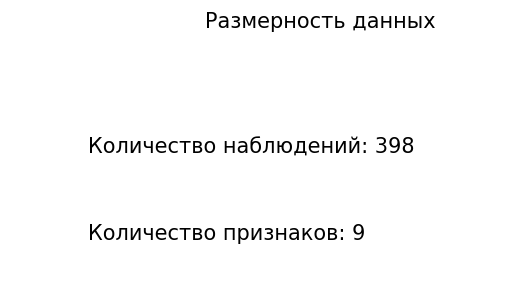
# Ссылка на репозиторий с программной реализацией

https://github.com/ArtyomI69/data\_science

# Описание проделанной работы

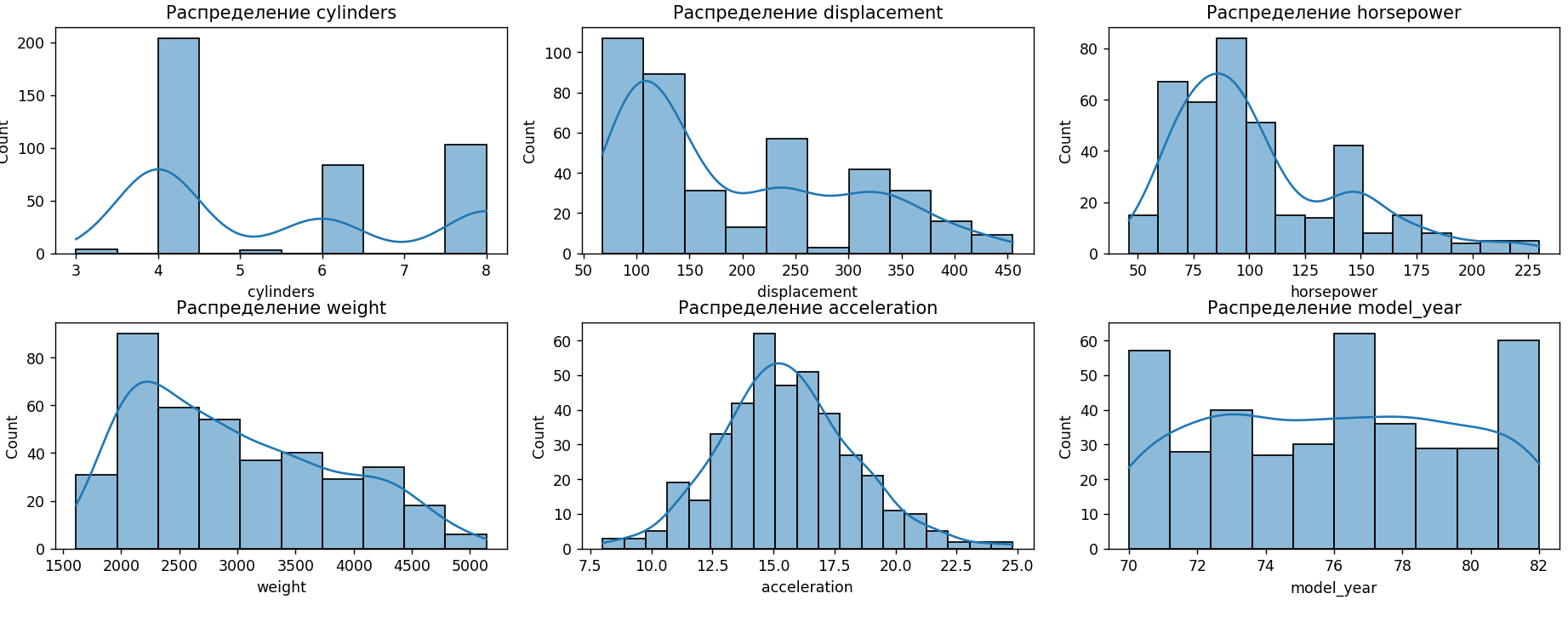
Результаты выполнения кода и анализ:

## 1. Общая информация о данных



## 2. Разведочный анализ

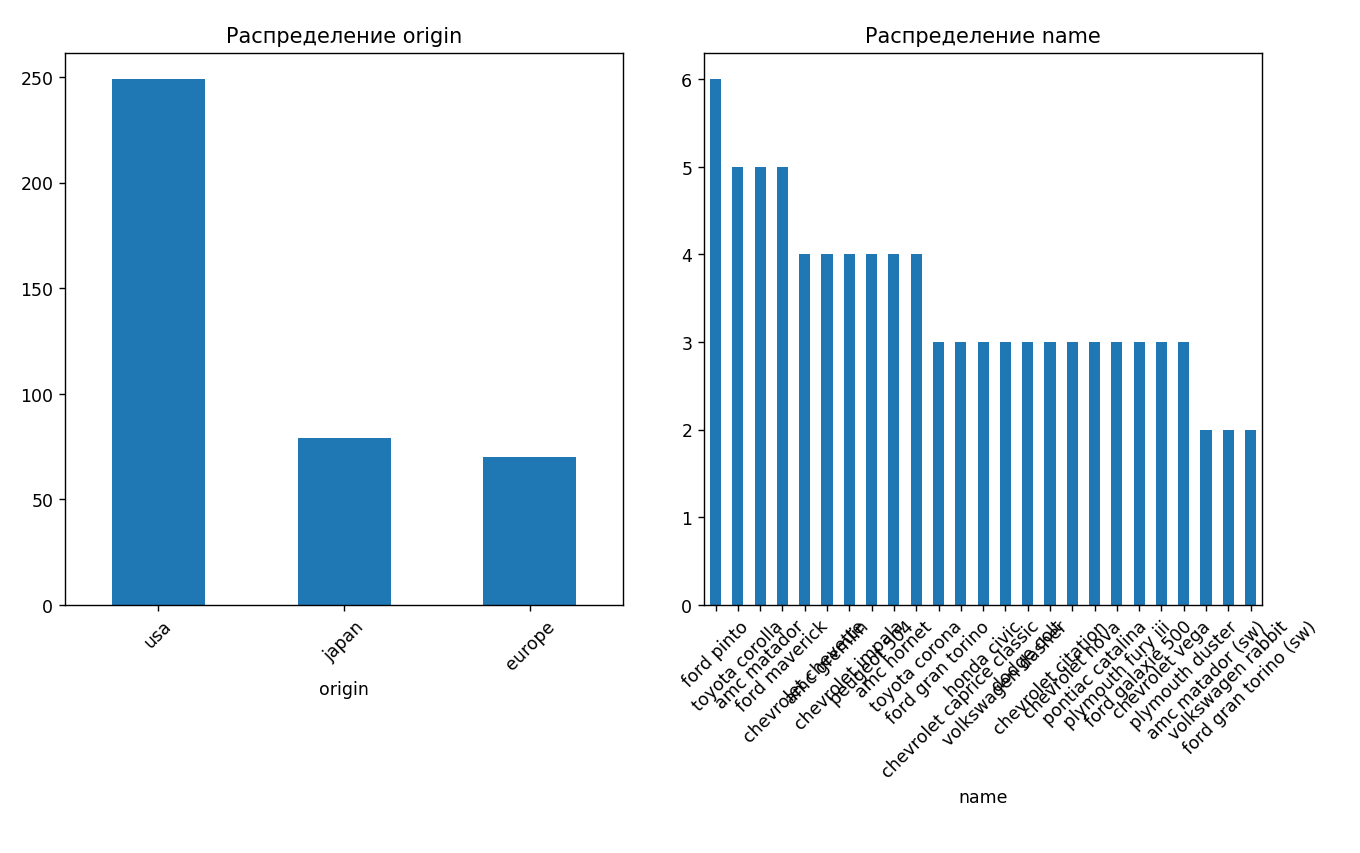
**Числовые переменные**

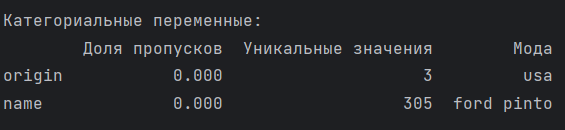




1. **mpg (милли на галлон)**
   * Среднее значение — 23.515, медиана — 23.000, что говорит о близости распределения к нормальному.
   * Разброс значений довольно велик (от 9 до 46.6), что указывает на разнообразие типов автомобилей по экономичности.
2. **cylinders (число цилиндров)**
   * Большинство автомобилей имеют от 3 до 8 цилиндров.
   * Среднее — 5.455, медиана — 4, что говорит о преобладании автомобилей с 4 цилиндрами.
3. **displacement (объем двигателя в куб. дюймах)**
   * Широкий диапазон (68–455), среднее — 193.42, медиана — 148.5.
   * Большая дисперсия (10872.199) указывает на значительное различие в мощностях двигателей.
4. **horsepower (мощность в л.с.)**
   * Встречаются как маломощные (46 л.с.), так и очень мощные (230 л.с.) автомобили.
   * Среднее (104.469) выше медианы (93.5), что говорит о наличии небольшого количества очень мощных машин, смещающих среднее значение вверх.
5. **weight (вес в фунтах)**
   * Диапазон от 1613 до 5140 фунтов, что подтверждает наличие как легких, так и тяжелых автомобилей.
   * Средний вес — 2970.425 фунтов, медианный — 2803.5, что говорит о слегка правостороннем распределении.
6. **acceleration (разгон до 60 миль/ч, секунды)**
   * Значения варьируются от 8.0 до 24.8 секунд.
   * Среднее (15.568) и медиана (15.5) практически совпадают, что указывает на близкое к симметричному распределение.
7. **model\_year (год выпуска модели)**
   * Диапазон значений от 70 до 82 (вероятно, речь о 1970–1982 годах).
   * Среднее и медиана равны 76, что говорит о равномерном распределении данных по годам.

**Категориальные переменные**



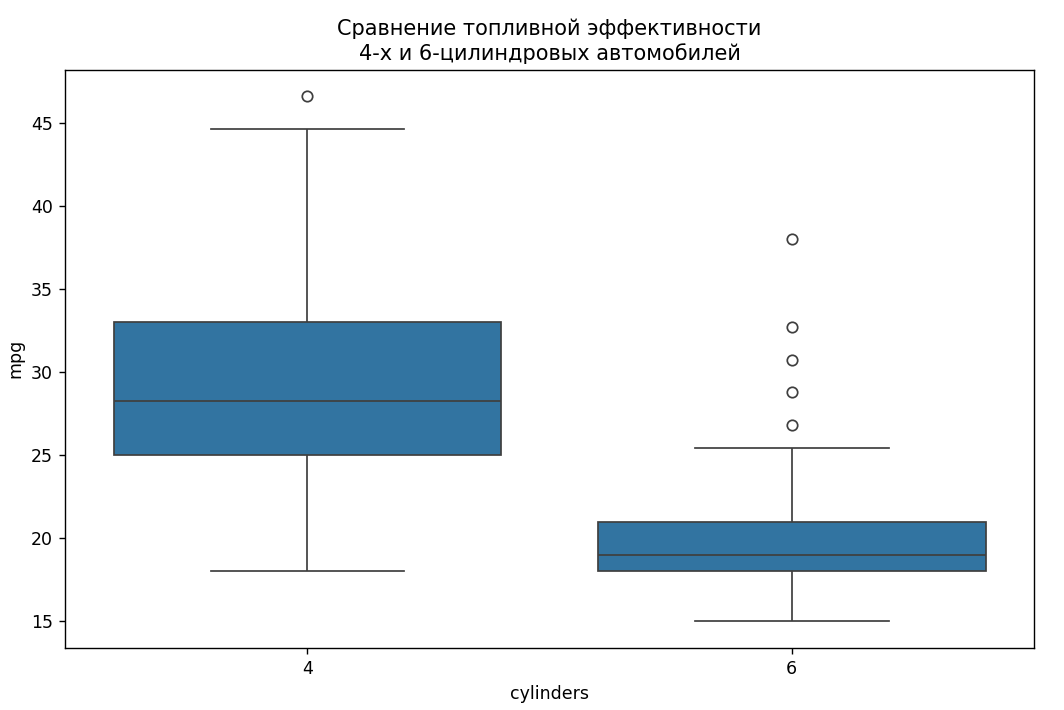


**Выводы**:

* Производители из США (usa) доминируют в данных.
* Названия автомобилей почти уникальны (305 уникальных значений из 398).

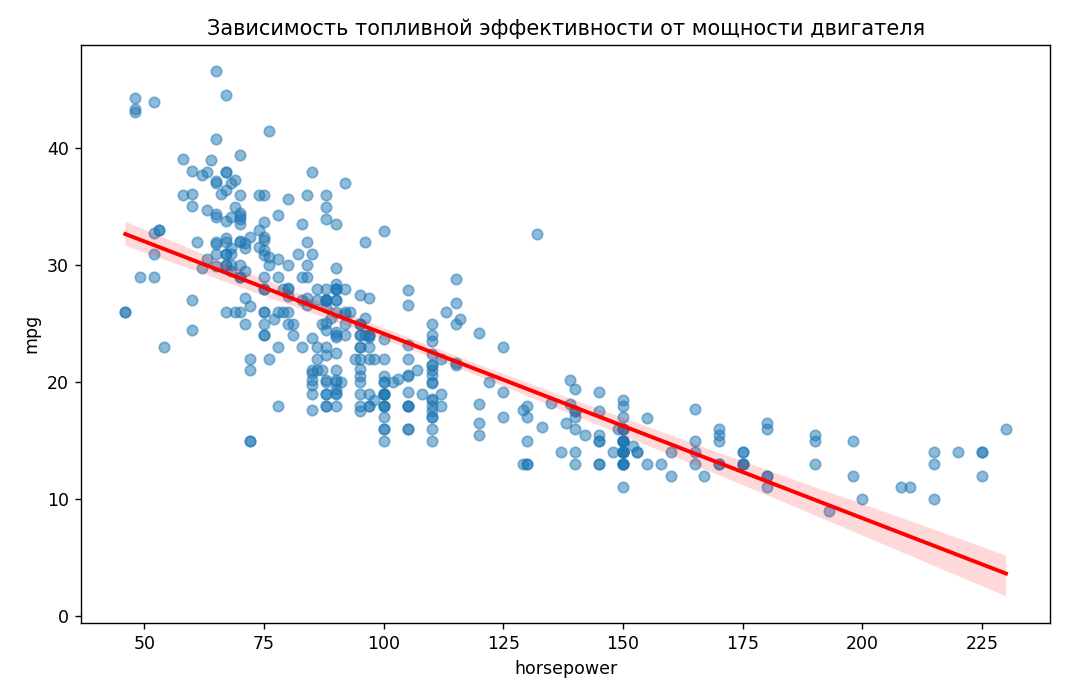
## ****3. Проверка гипотез****

**Гипотеза 1**: Средний расход топлива (mpg) для 4-х и 6-цилиндровых автомобилей различается.



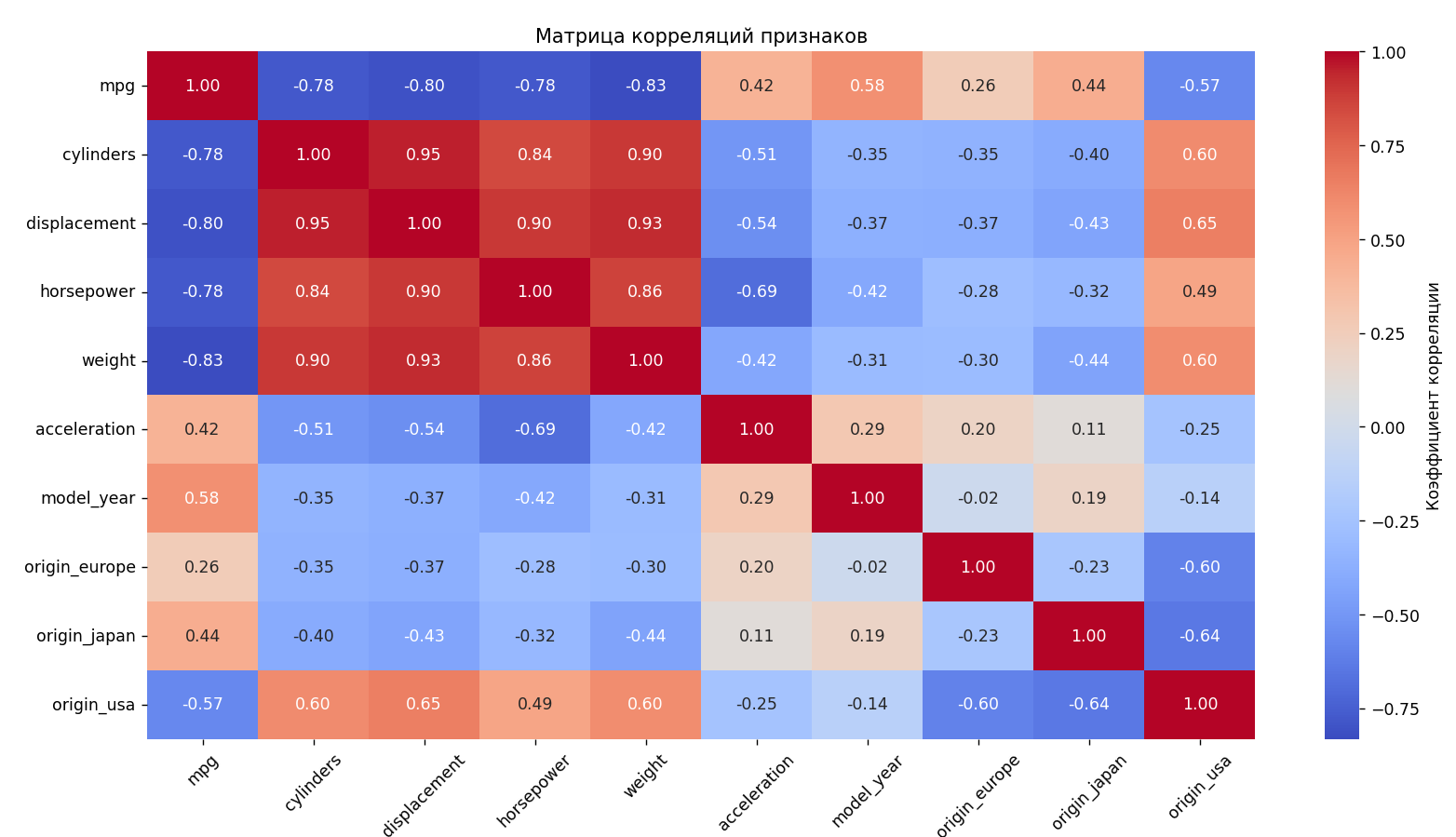
*  p-value = 0.0000 означает, что различие статистически значимо.
* Это подтверждает, что автомобили с 4 и 6 цилиндрами имеют разные средние показатели mpg.
* Из диаграммы размаха видно, что автомобили с 4 цилиндрами в среднем более экономичны по расходу топлива, чем автомобили с 6 цилиндрами.

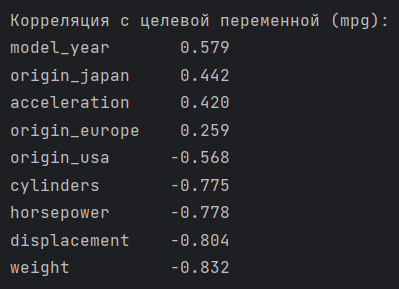
 **Гипотеза 2**: Корреляция между мощностью (horsepower) и расходом топлива (mpg).



* Коэффициент Спирмена = -0.85 указывает на сильную отрицательную корреляцию.
* p-value = 0.0000 подтверждает статистическую значимость связи.
* Это означает, что по мере увеличения мощности двигателя (horsepower) топливная эффективность (mpg) снижается.
* График рассеяния с линией тренда подтверждает эту тенденцию.

## 4. Таблица корреляции





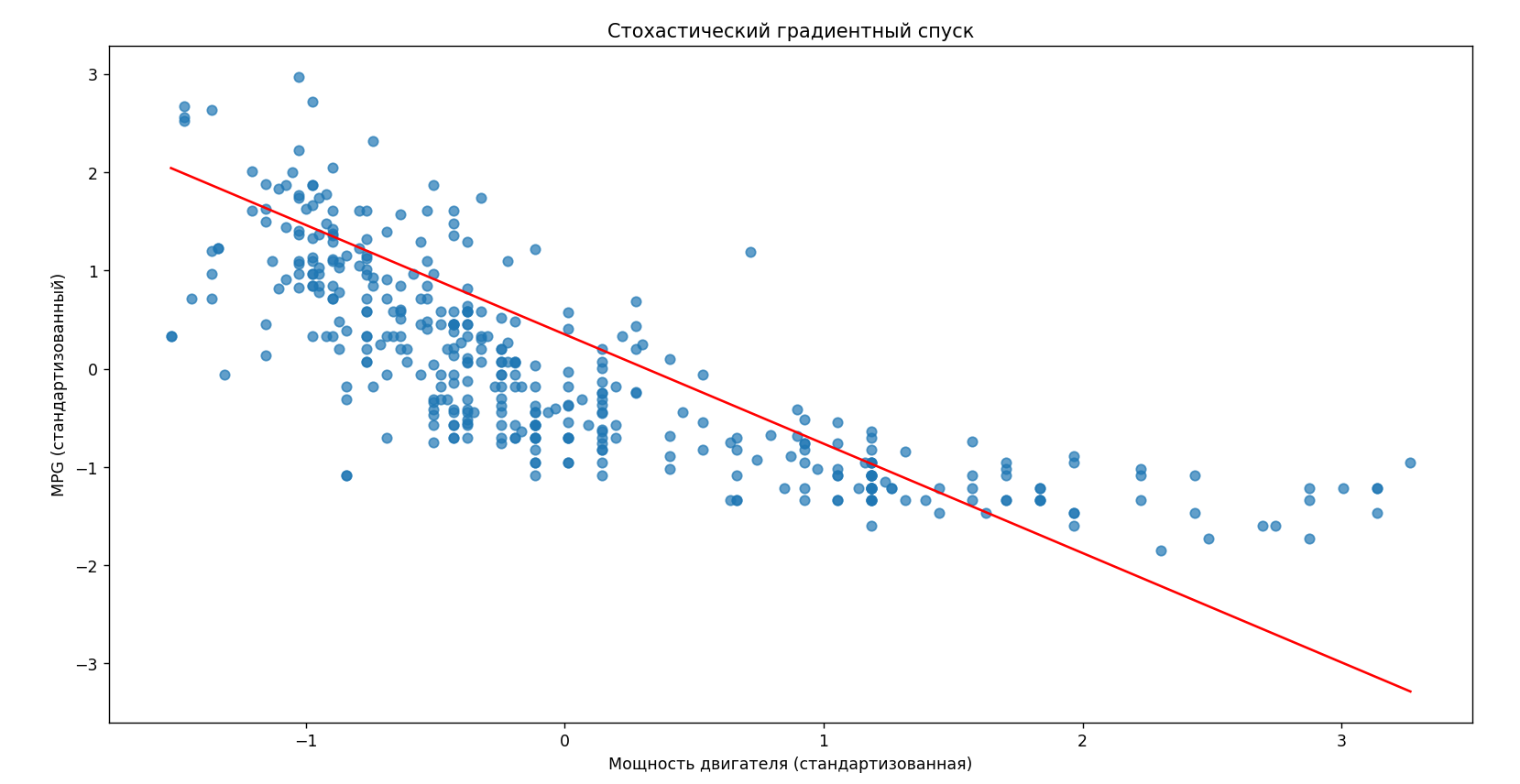
 **Положительная корреляция:**

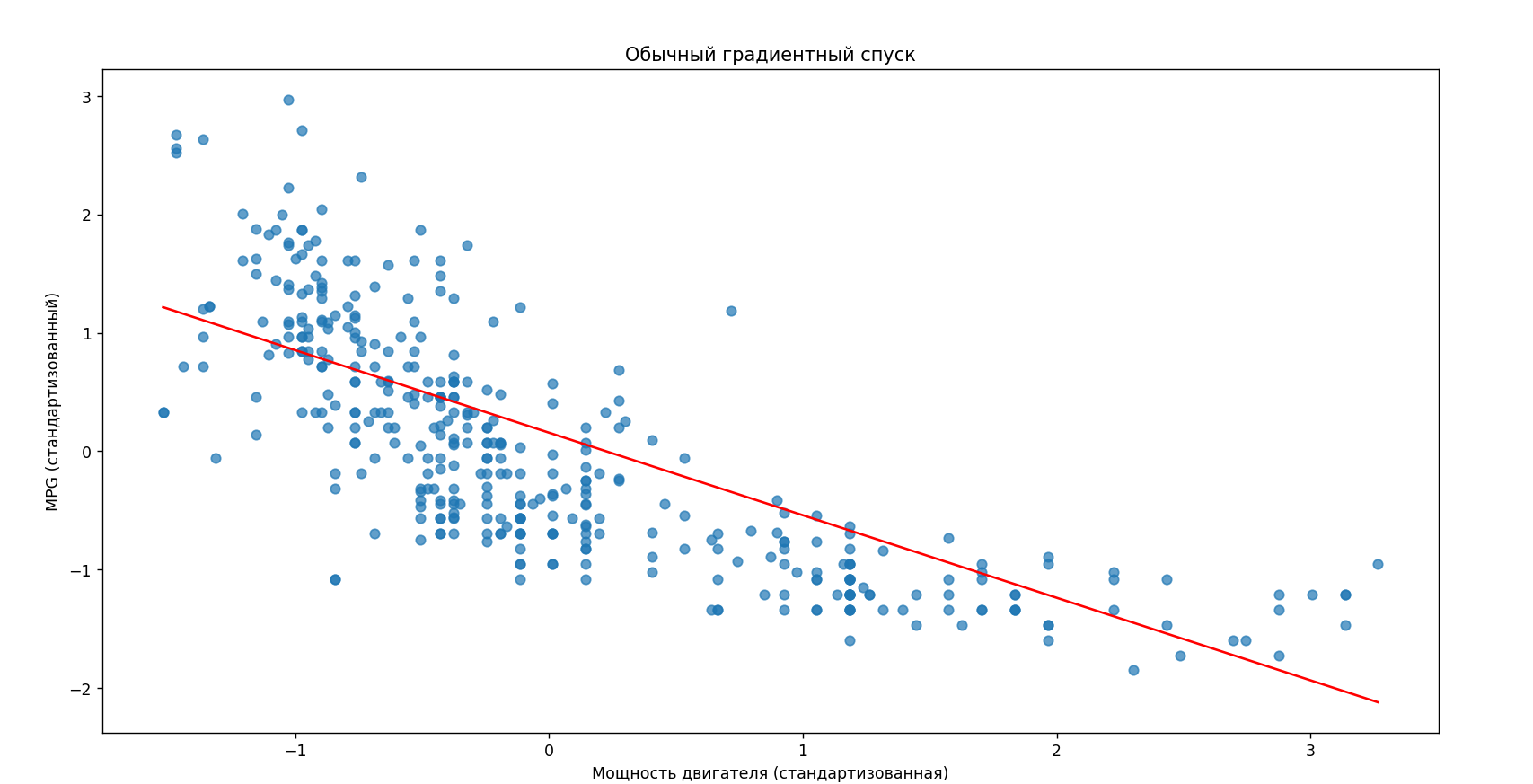
* **model\_year (0.579)** – более новые автомобили имеют более высокий mpg (экономичность топлива).
* **origin\_japan (0.442)** – автомобили японского производства, как правило, более экономичны.
* **acceleration (0.420)** – ускорение связано с более высокой экономичностью топлива.
* **origin\_europe (0.259)** – автомобили европейского производства также демонстрируют некоторую склонность к высокой экономичности.

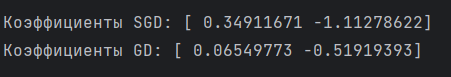
 **Отрицательная корреляция:**

* **origin\_usa (-0.568)** – американские автомобили менее экономичны.
* **cylinders (-0.775)**, **horsepower (-0.778)**, **displacement (-0.804)**, **weight (-0.832)** – большее количество цилиндров, большая мощность, объем двигателя и масса автомобиля связаны с более низким mpg (большим расходом топлива).

## 5. Градиентный спуск







Выводы по стохастическому градиентному спуску (SGD) и обычному градиентному спуску (GD):

1. **Различие в коэффициентах:**
   * Стохастический градиентный спуск (SGD) получил коэффициенты [0.34911671, -1.11278622], что указывает на более крутой наклон линии регрессии.
   * Обычный градиентный спуск (GD) привёл к коэффициентам [-0.16534121, -0.80025795], что означает меньший наклон.
2. **Графическая интерпретация:**
   * Визуально линия SGD (первая диаграмма) имеет более выраженный отрицательный наклон, чем линия GD (вторая диаграмма). Это подтверждает, что модель SGD нашла более сильную связь между мощностью двигателя и расходом топлива (MPG).
   * Разброс точек относительно линии регрессии схож в обоих случаях, но модель SGD лучше подстроилась под данные.
3. **Особенности методов:**
   * **SGD** работает быстрее и может быстрее адаптироваться к данным, но его результаты могут быть менее стабильны из-за высокой вариативности градиентов.
   * **GD** более стабилен, так как использует всю выборку для вычисления градиента, но может сходиться медленнее.

# Краткий вывод по работе. Описание реализованной программы и её функций.

Проведен комплексный анализ данных об автомобилях (mpg):

1. **Ключевые зависимости**: Обнаружена сильная отрицательная связь между расходом топлива (mpg) и мощностью/весом авто (коэф. Спирмена ≈ -0.9).
2. **Стат. гипотезы**:
   * 4-цилиндровые авто имеют значительно лучшую топливную эффективность vs 6-цилиндровые (p < 0.0001)
   * Современные модели (↑ model\_year) показывают ↑ mpg (r = 0.58)
3. **Моделирование**: Реализация градиентного спуска подтвердила линейную зависимость mpg от мощности (коэф. ≈ -0.9).

# Ссылки на используемые материалы. Документация

* [Seaborn: Datasets](https://seaborn.pydata.org/generated/seaborn.load_dataset.html)
* [Pandas: Data Analysis](https://pandas.pydata.org/docs/)
* [NumPy: Mathematical Operations](https://numpy.org/doc/stable/)
* [SciPy: Statistical Functions](https://docs.scipy.org/doc/scipy/reference/stats.html)
* [Scikit-learn: Preprocessing](https://scikit-learn.org/stable/modules/preprocessing.html)

# Листинг кода

import seaborn as sns  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
from scipy import stats  
from sklearn.preprocessing import OneHotEncoder  
  
  
def load\_data():  
 return sns.load\_dataset('mpg')  
  
  
def visualize\_data\_shape(df):  
 plt.figure(figsize=(6, 3))  
 plt.text(0.1, 0.6, f"Количество наблюдений: {df.shape[0]}", fontsize=12)  
 plt.text(0.1, 0.3, f"Количество признаков: {df.shape[1]}", fontsize=12)  
 plt.axis('off')  
 plt.title("Размерность данных")  
 plt.show()  
  
  
def plot\_numeric\_distributions(df):  
 numeric\_cols = df.select\_dtypes(include=['int', 'float']).columns.drop('mpg')  
 plt.figure(figsize=(15, 10))  
 for i, col in enumerate(numeric\_cols, 1):  
 plt.subplot(3, 3, i)  
 sns.histplot(df[col], kde=True)  
 plt.title(f"Распределение {col}")  
 plt.tight\_layout()  
 plt.show()  
  
  
def plot\_categorical\_distributions(df):  
 categorical\_cols = df.select\_dtypes(include=['object', 'category']).columns  
 plt.figure(figsize=(15, 5))  
 for i, col in enumerate(categorical\_cols, 1):  
 plt.subplot(1, 3, i)  
 df[col].value\_counts().nlargest(25).plot(kind='bar')  
 plt.title(f"Распределение {col}")  
 plt.xticks(rotation=45)  
 plt.tight\_layout()  
 plt.show()  
  
  
def run\_hypothesis(df):  
 plt.figure(figsize=(10, 6))  
 sns.boxplot(x='cylinders', y='mpg', data=df[df['cylinders'].isin([4, 6])])  
 plt.title("Сравнение топливной эффективности\n4-х и 6-цилиндровых автомобилей")  
 plt.show()  
  
  
def plot\_correlation(df):  
 df\_clean = df[['horsepower', 'mpg']].dropna()  
 plt.figure(figsize=(10, 6))  
 sns.regplot(x='horsepower', y='mpg', data=df\_clean, scatter\_kws={'alpha': 0.5}, line\_kws={'color': 'red'})  
 plt.title("Зависимость топливной эффективности от мощности двигателя")  
 plt.show()  
  
  
def encode\_categorical(df):  
 encoder = OneHotEncoder()  
 encoded\_origin = encoder.fit\_transform(df[['origin']]).toarray()  
 return pd.concat([df, pd.DataFrame(encoded\_origin, columns=encoder.get\_feature\_names\_out(['origin']))], axis=1)  
  
  
def plot\_correlation\_matrix(df):  
 plt.figure(figsize=(12, 8))  
 corr\_matrix = df.select\_dtypes(include=['number']).corr()  
 sns.heatmap(corr\_matrix, annot=True, fmt=".2f", cmap='coolwarm', cbar\_kws={'label': 'Коэффициент корреляции'})  
 plt.title("Матрица корреляций признаков")  
 plt.xticks(rotation=45)  
 plt.yticks(rotation=0)  
 plt.show()  
  
  
def stochastic\_gradient\_descent(X, y, lr=0.01, epochs=100):  
 theta = np.random.randn(2)  
 X\_b = np.c\_[np.ones((len(X), 1)), X]  
 for epoch in range(epochs):  
 for i in range(len(X\_b)):  
 xi = X\_b[i:i + 1]  
 yi = y[i:i + 1]  
 gradients = 2 \* xi.T.dot(xi.dot(theta) - yi)  
 theta -= lr \* gradients  
 return theta  
  
  
def gradient\_descent(X, y, lr=0.01, epochs=100):  
 theta = np.random.randn(2)  
 X\_b = np.c\_[np.ones((len(X), 1)), X]  
 for epoch in range(epochs):  
 gradients = 2 / len(X) \* X\_b.T.dot(X\_b.dot(theta) - y)  
 theta -= lr \* gradients  
 return theta  
  
  
def plot\_gradient\_descent(X, y, theta, title):  
 plt.figure(figsize=(10, 6))  
 plt.scatter(X, y, alpha=0.7)  
 x\_values = np.array([X.min(), X.max()])  
 y\_values = theta[0] + theta[1] \* x\_values  
 plt.plot(x\_values, y\_values, color='red')  
 plt.xlabel('Мощность двигателя (стандартизованная)')  
 plt.ylabel('MPG (стандартизованный)')  
 plt.title(title)  
 plt.show()  
  
  
def main():  
 df = load\_data()  
 visualize\_data\_shape(df)  
 plot\_numeric\_distributions(df)  
 plot\_categorical\_distributions(df)  
 run\_hypothesis(df)  
 plot\_correlation(df)  
 df\_encoded = encode\_categorical(df)  
 plot\_correlation\_matrix(df\_encoded)  
  
 data = df[['horsepower', 'mpg']].dropna()  
 X = data[['horsepower']].values  
 y = data['mpg'].values  
 X\_normalized = (X - X.mean()) / X.std()  
 y\_normalized = (y - y.mean()) / y.std()  
  
 theta\_stochastic = stochastic\_gradient\_descent(X\_normalized, y\_normalized)  
 theta\_standard = gradient\_descent(X\_normalized, y\_normalized)  
  
 plot\_gradient\_descent(X\_normalized, y\_normalized, theta\_stochastic, "Стохастический градиентный спуск")  
 plot\_gradient\_descent(X\_normalized, y\_normalized, theta\_standard, "Обычный градиентный спуск")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()