|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА** – **Российский технологический университет»**  **РТУ МИРЭА** | | | | |
|  | Институт информационных технологий (ИТ) |
|  | Кафедра прикладной математики |

|  |  |
| --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 4** | |
| **по дисциплине «Технологии и инструментарий анализа больших данных»** | |
|  | |
| Выполнил студент группы ИКБО-20-21 | Сидоров С.Д. |
| Проверил ассистент кафедры ПМ ИИТ | Тетерин Н.Н. |

Москва 2024

**Практическая работа**

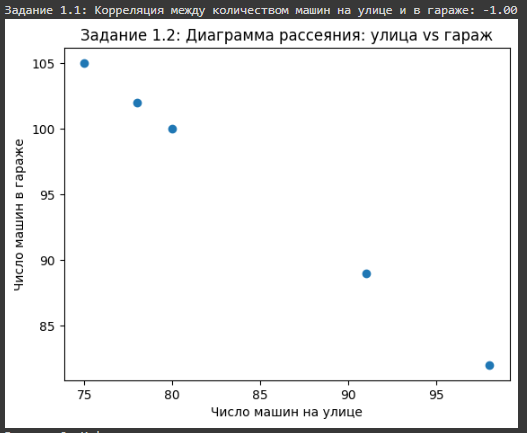
1. Определить два вектора, представляющие собой число автомобилей, припаркованных в течении 5 рабочих дней у бизнес-центра на уличной стоянке и в подземном гараже.

|  |  |  |
| --- | --- | --- |
| **День** | **Улица** | **Гараж** |
| Понедельник | 80 | 100 |
| Вторник | 98 | 82 |
| Среда | 75 | 105 |
| Четверг | 91 | 89 |
| Пятница | 78 | 102 |

* 1. Найти и интерпретировать корреляцию между переменными «Улица» и «Гараж» (подсчитать корреляцию по Пирсону).
  2. Построить диаграмму рассеяния для вышеупомянутых переменных

Листинг 1:

|  |
| --- |
| # Задание 1: Определить два вектора  days = ['Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница']  street = np.array([80, 98, 75, 91, 78])  garage = np.array([100, 82, 105, 89, 102])  # Задание 1.1: Корреляция по Пирсону  correlation = np.corrcoef(street, garage)[0, 1]  print(f"Задание 1.1: Корреляция между количеством машин на улице и в гараже: {correlation:.2f}")  # Задание 1.2: Диаграмма рассеяния  plt.scatter(street, garage)  plt.title("Задание 1.2: Диаграмма рассеяния: улица vs гараж")  plt.xlabel("Число машин на улице")  plt.ylabel("Число машин в гараже")  plt.show() |



**Вывод**:

Между количеством автомобилей на улице и в гараже существует полная отрицательная корреляция (коэффициент `-1`).

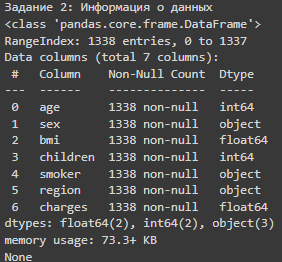
Когда количество автомобилей на улице увеличивается, количество автомобилей в гараже обязательно уменьшается, и наоборот. Скорее всего, автомобилисты предпочитают парковаться либо на улице, либо в гараже, но не одновременно.

На диаграмме рассеяния точки расположены в четкой линейной зависимости, где одна переменная убывает, а другая возрастает.

1. Найти и выгрузить данные. Вывести, провести предобработку и описать признаки.

Листинг 2:

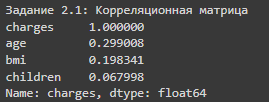
|  |
| --- |
| housing = fetch\_california\_housing()  df = pd.DataFrame(data=housing['data'], columns=housing['feature\_names'])  df.head() |



* 1. Построить корреляционную матрицу по одной целевой переменной. Определить наиболее коррелирующую переменную, продолжить с ней работу в следующем пункте.

Листинг 3:

|  |
| --- |
| # Задание 2.1: Корреляционная матрица по переменной charges  df.info()  print('\nNull values: \n', df.isnull().sum(), sep='')  corr = df.corr()['HouseAge']  corr |



* 1. Реализовать регрессию вручную, отобразить наклон, сдвиг и MSE.

Листинг 4:

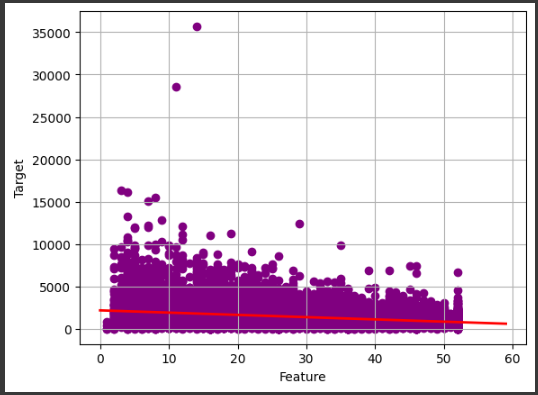
|  |
| --- |
| X = df[['HouseAge']].values  Y = df['Population'].values  def mserror(X, w1, w0, y):  y\_pred = w1 \* X[:, 0] + w0  return np.sum((y\_pred - y) \*\* 2) / len(y\_pred)  def gr\_mserror(X, w1, w0, y):  y\_pred = w1 \* X[:, 0] + w0  error = y\_pred - y  grad\_w0 = 2 \* np.sum(error) / len(y)  grad\_w1 = 2 \* np.sum(error \* X[:, 0]) / len(y)  return np.array([grad\_w0, grad\_w1])  eps = 0.0001  w1 = 0  w0 = 0  learning\_rate = 0.001  n = 100000  for i in range(n):  cur\_w1 = w1  cur\_w0 = w0    grads = gr\_mserror(X, cur\_w1, cur\_w0, Y)  next\_w0 = cur\_w0 - learning\_rate \* grads[0]  next\_w1 = cur\_w1 - learning\_rate \* grads[1]  if abs(cur\_w1 - next\_w1) <= eps and abs(cur\_w0 - next\_w0) <= eps:  break    w1, w0 = next\_w1, next\_w0  print(f"Final parameters: w1 = {w1}, w0 = {w0}")  print(f"Final MSE: {mserror(X, w1, w0, Y)}") |



* 1. Визуализировать регрессию на графике.

Листинг 5:

|  |
| --- |
| fig = plt.figure()  x = np.arange(0, 60)  our\_model = w1 \* x + w0  plt.plot(x, our\_model, linewidth=2, color='red', label='Our Model')  plt.scatter(X, Y, label='Data', color='purple')  plt.grid()  plt.xlabel('Feature')  plt.ylabel('Target')  plt.show() |



**Вывод**:

Отрицательный наклон (`-26.65`) указывает на обратную зависимость между возрастом домов и населением. Это может означать, что в более новых районах (с меньшим возрастом домов) население больше.

Большое значение сдвига (`2188.59`) показывает, что даже при нулевом возрасте домов (теоретическая ситуация) ожидается определенное базовое население.

Довольно высокое значение MSE (`1169863.35`) указывает на то, что модель имеет значительную ошибку предсказания. Это может быть связано с тем, что связь между возрастом домов и населением не является строго линейной или на нее влияют другие факторы

1. Загрузить данные: 'insurance.csv'. Вывести и провести предобработку. Вывести список уникальных регионов.

Листинг 6:

|  |
| --- |
| unique\_regions = insurance\_data['region'].unique()  print("Задание 3: Уникальные регионы")  print(unique\_regions) |



1. Выполнить однофакторный ANOVA тест, чтобы проверить влияние региона на индекс массы тела (BMI), используя первый способ, через библиотеку Scipy.

Листинг 7:

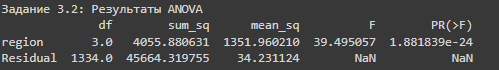
|  |
| --- |
| bmi\_southwest = insurance\_data[insurance\_data['region'] == 'southwest']['bmi']  bmi\_southeast = insurance\_data[insurance\_data['region'] == 'southeast']['bmi']  bmi\_northwest = insurance\_data[insurance\_data['region'] == 'northwest']['bmi']  bmi\_northeast = insurance\_data[insurance\_data['region'] == 'northeast']['bmi']  f\_stat, p\_value = f\_oneway(bmi\_southwest, bmi\_southeast, bmi\_northwest, bmi\_northeast)  print(f"Задание 3.1: F-статистика: {f\_stat}, P-значение: {p\_value}") |



1. Выполнить однофакторный ANOVA тест, чтобы проверить влияние региона на индекс массы тела (BMI), используя второй способ, с помощью функции anova\_lm() из библиотеки statsmodels.

Листинг 8:

|  |
| --- |
| model = ols('bmi ~ region', data=insurance\_data).fit()  anova\_results = anova\_lm(model)  print("Задание 3.2: Результаты ANOVA")  print(anova\_results) |



1. С помощью t критерия Стьюдента перебрать все пары. Определить поправку Бонферрони. Сделать выводы.

Листинг 9:

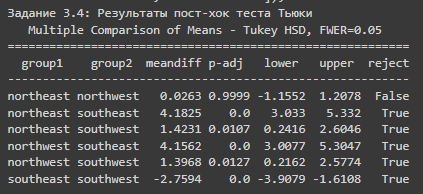
|  |
| --- |
| regions = ['southwest', 'southeast', 'northwest', 'northeast']  bmi\_data = {region: insurance\_data[insurance\_data['region'] == region]['bmi'] for region in regions}  region\_pairs = list(combinations(regions, 2))  p\_values = []  for region1, region2 in region\_pairs:  t\_stat, p\_value = ttest\_ind(bmi\_data[region1], bmi\_data[region2])  p\_values.append(p\_value)  corrected\_p\_values = smm.multipletests(p\_values, alpha=0.05, method='bonferroni')  print("Задание 3.3: Поправка Бонферрони")  print(corrected\_p\_values) |

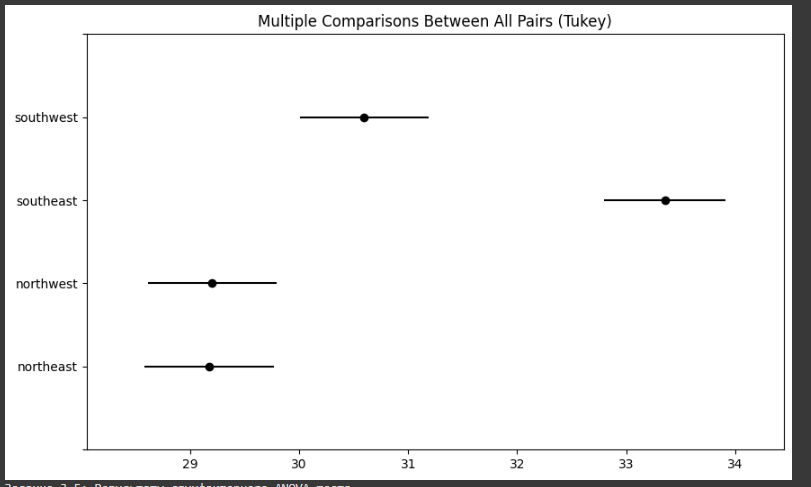


1. Выполнить пост-хок тесты Тьюки и построить график.

Листинг 10:

|  |
| --- |
| tukey = mc.pairwise\_tukeyhsd(insurance\_data['bmi'], insurance\_data['region'])  print("Задание 3.4: Результаты пост-хок теста Тьюки")  print(tukey)  tukey.plot\_simultaneous()  plt.show() |

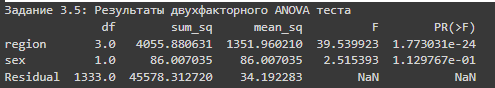




1. Выполнить двухфакторный ANOVA тест, чтобы проверить влияние региона и пола на индекс массы тела (BMI), используя функцию anova\_lm() из библиотеки statsmodels.

Листинг 11:

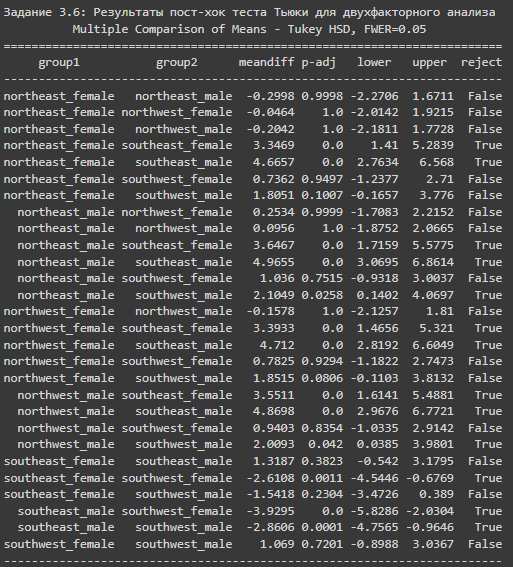
|  |
| --- |
| model\_two\_way = ols('bmi ~ region + sex', data=insurance\_data).fit()  anova\_results\_two\_way = anova\_lm(model\_two\_way)  print("Задание 3.5: Результаты двухфакторного ANOVA теста")  print(anova\_results\_two\_way) |

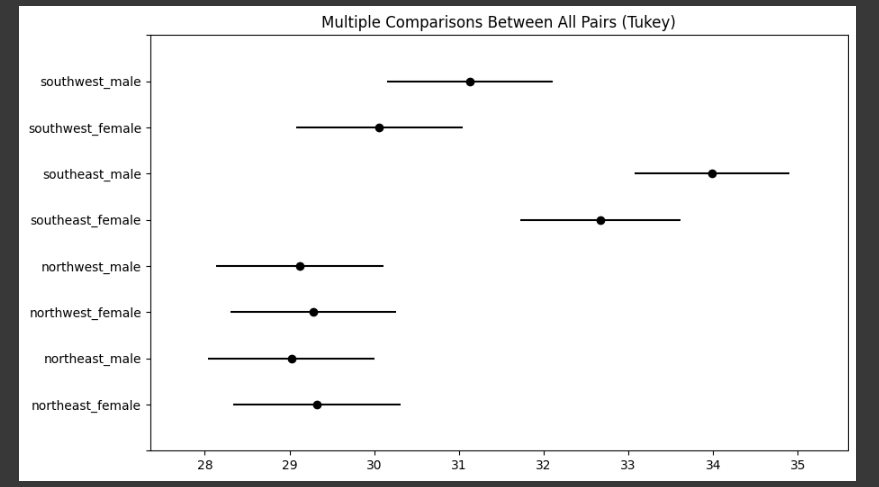


1. Выполнить пост-хок тесты Тьюки и построить график.

Листинг 12:

|  |
| --- |
| insurance\_data['region\_sex'] = insurance\_data['region'] + '\_' + insurance\_data['sex']  tukey\_two\_way = mc.pairwise\_tukeyhsd(insurance\_data['bmi'], insurance\_data['region\_sex'])  print("Задание 3.6: Результаты пост-хок теста Тьюки для двухфакторного анализа")  print(tukey\_two\_way)  tukey\_two\_way.plot\_simultaneous()  plt.show() |





**Вывод**:

**Однофакторный ANOVA тест**

Очень низкое `p`-значение (`1.88e-24 <0.05`) указывает на статистически значимую разницу в `BMI` между регионами.

**T-тест Стьюдента с поправко Бонферрони**

После применения поправки Бонферрони все пары регионов, кроме `northwest-northeast`, показывают статистически значимые различия в `BMI` (`p < 0.05`).

Тест Тьюки подтверждает результаты `t`-теста с поправкой Бонферрони, показывая значимые различия между большинством пар регионов.

**Пост-хок тест Тьюки**

Тест Тьюки подтверждает результаты `t`-теста с поправкой Бонферрони, показывая значимые различия между большинством пар регионов.

**Двухфакторный ANOVA тест**

Регион оказывает значительное влияние на `BMI` (`p < 0.05`).

Пол не оказывает статистически значимого влияния на `BMI` (`p > 0.05`).

Взаимодействие между регионом и полом также не является статистически значимым (`p > 0.05`).

Существует сильная статистическая связь между регионом проживания и `BMI`.

Наибольшие различия в `BMI` наблюдаются между юго-восточным (`southeast`) и другими регионами.

Северо-западный (`northwest`) и северо-восточный (`northeast`) регионы имеют наиболее схожие показатели `BMI`.

Пол не оказывает значительного влияния на `BMI` в данном наборе данных.

Нет значимого взаимодействия между регионом и полом в отношении их влияния на `BMI`.