|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИИТ) | |
| Прикладная Математика (ПМ) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4** | | | |
| **по дисциплине «Технологии и инструментарий анализа больших данных»** | | | |
|  | | | |
| Выполнил студент группы ИКБО-13-20 | | Вакуленко А. И. | |
|  | |  | |
| Принял | | Горячев А. А. | |
| Практические работы выполнены | «\_\_\_»\_\_\_\_\_\_\_2023 г. | | (подпись студента) | |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_2023 г. | | (подпись руководителя) | |
|  |  | |  | |

Москва 2023

**Задание 1.**

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

Таблица 1 – данные о хранении автомобилей

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

|  |
| --- |
| Листинг 1 - Задание 1 |
| import numpy as np  outdoor\_storage = np.array([80, 98, 75, 91, 78])  garage\_storage = np.array([100, 82, 105, 89, 102])  print('outdoor\_storage', outdoor\_storage)  print('garage\_storage', garage\_storage) |

Результат работы программы представлены на рисунке 1.

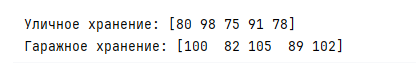


Рисунок 1 - Результат работы

**Задание 2**

Найти и интерпретировать корреляцию между переменными «Улица» и «Гараж» (подсчитать корреляцию по Пирсону) (Листинг 2).

Результат работы программы представлены на рисунке 2.

|  |
| --- |
| Листинг 2 - Задание 2 |
| print(f'Матрица коэффициентов корреляции: \n{np.corrcoef(outdoor\_storage, garage\_storage)}') print(f'Коэффициент корреляции между двумя переменными: {np.corrcoef(outdoor\_storage, garage\_storage)[0, 1]}') |

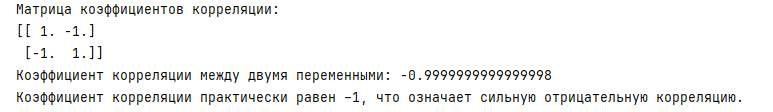


Рисунок 2 - Результат работы

**Задание 3.**

Построить диаграмму рассеяния для вышеупомянутых переменных. (Листинг 3)

|  |
| --- |
| Листинг 3 - Задание 3 |
| import matplotlib.pyplot as plt  plt.grid()  plt.scatter(outdoor\_storage, garage\_storage, marker='o')  plt.title('Диаграмма рассеивания данных о хранении автомобилей', fontsize='16')  plt.xlabel('Стоянка на улице')  plt.ylabel('Стоянка в гараже')  print('На получившимся графике мы видим сильную отрицательную корреляцию') |

Результат работы программы представлены на рисунке 3.

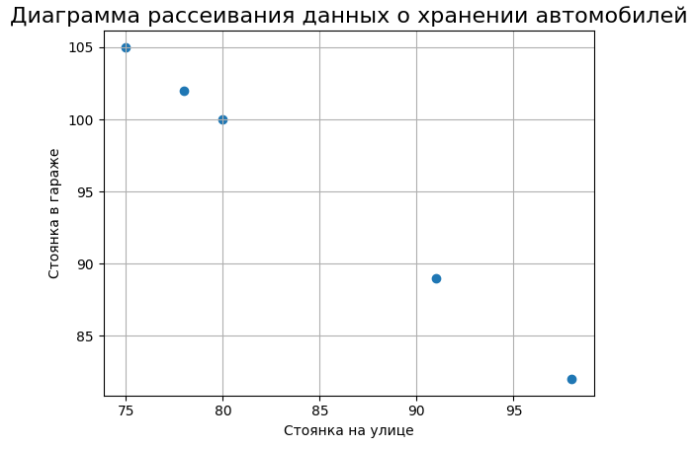


Рисунок *3* – Гистограммы числовых показателей

**Задание 4.**

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

О наборе данных:

* phone\_name:название
* brand:марка телефона
* os:операционная система телефона
* inches size of the phone screen as inches: размер экрана телефона в дюймах
* resolution: разрешение экрана телефона
* battery: емкость батареи телефона
* battery\_type: тип батареи телефона
* ram (GB): оперативная память телефона в ГБ
* announcement\_date: дата анонса телефона
* weight(g): вес телефона в граммах
* storage (GB): объем памяти телефона в ГБ
* video\_720p: поддерживает ли камера телефона функцию 720p
* video\_1080p: поддерживает ли камера телефона функцию
* video\_4K: поддерживает ли камера телефона функцию
* video\_8K: поддерживает ли камера телефона функцию 8K
* video\_30fps: поддерживает ли камера телефона функцию 30 кадров в секунду
* video\_60fps: поддерживает ли камера телефона функцию 60 кадров в секунду
* video\_120fps: поддерживает ли камера телефона функцию 120 кадров в секунду
* video\_240fps: поддерживает ли камера телефона функцию 240 кадров в секунду
* video\_480fps: поддерживает ли камера телефона функцию 480 кадров в секунду
* video\_960fps: поддерживает ли камера телефона функцию 960 кадров в секунду
* price(USD):цена телефона в долларах США

Для решения задачи был написан код (Листинг 4)

|  |
| --- |
| Листинг 4 - Задание 4 |
| import pandas as pd  data = pd.read\_csv('cleaned\_all\_phones.csv')  data = data[data.apply(lambda row: not any(row == 'None'), axis=1)]  data.isnull().sum() |

Результат работы программы представлены на рисунках 3, 4.

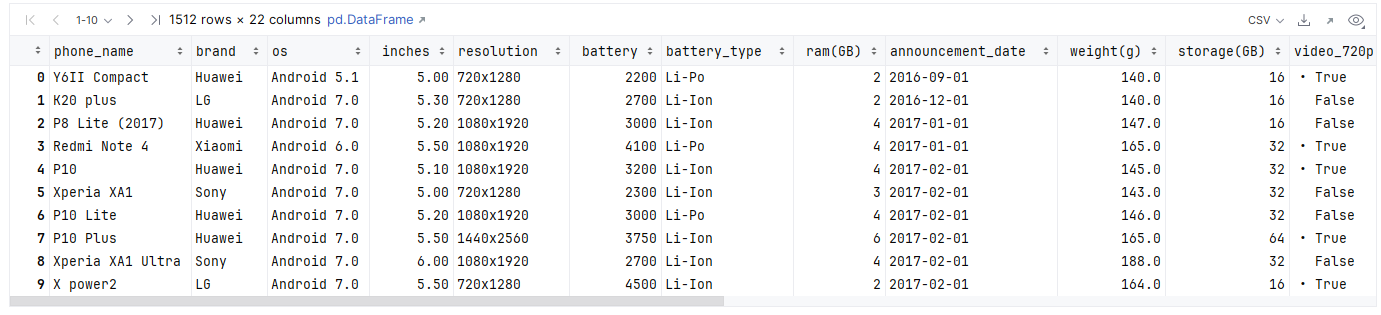
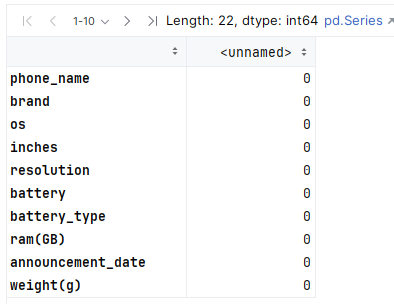


Рисунок 3 – Датасет



**Вывод**

По полученным значениям можно сделать вывод, что BMI содержит не очень много выбросов, из-за чего значение моды приблизительно равно значениям среднего и медианы, тогда как в Charges содержится очень много выбросов, из-за чего медиана и тем более среднее не верно характеризуют центральную тенденцию и сильно отличаются от моды.

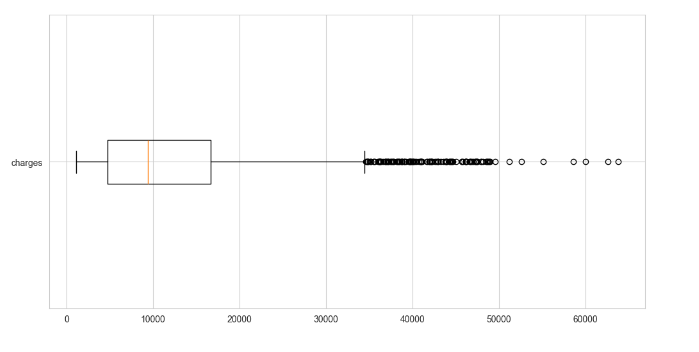
**Задание 5.**

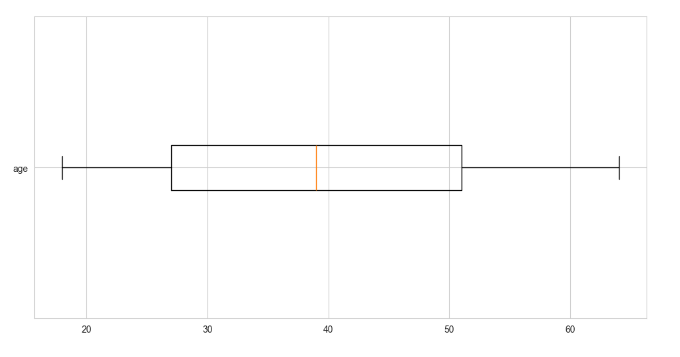
Построить box-plot для числовых показателей. Названия графиков должны соответствовать названиям признаков. Сделать выводы.

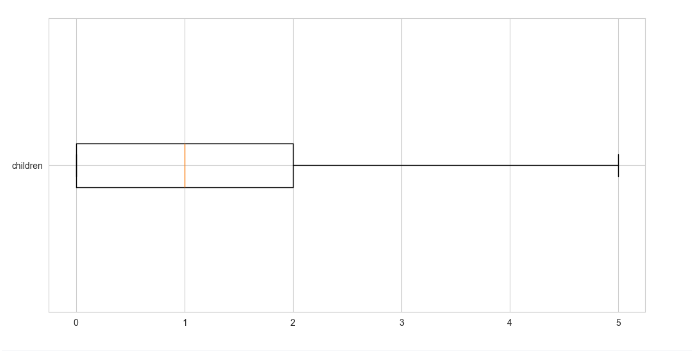
Для решения задачи был написан код (Листинг 5)

|  |
| --- |
| Листинг 5 - Задание 5 |
| plt.figure(figsize=(12,6)) plt.boxplot([dataset.charges], labels=['charges'], vert=False) plt.show()  plt.figure(figsize=(12,6)) plt.boxplot([dataset.age], labels=['age'], vert=False) plt.show()  plt.figure(figsize=(12,6)) plt.boxplot([dataset.children], labels=['children'], vert=False) plt.show()  plt.figure(figsize=(12,6)) plt.boxplot([dataset.bmi], labels=['bmi'], vert=False) plt.show() |

Результат работы программы представлены на рисунке 5.







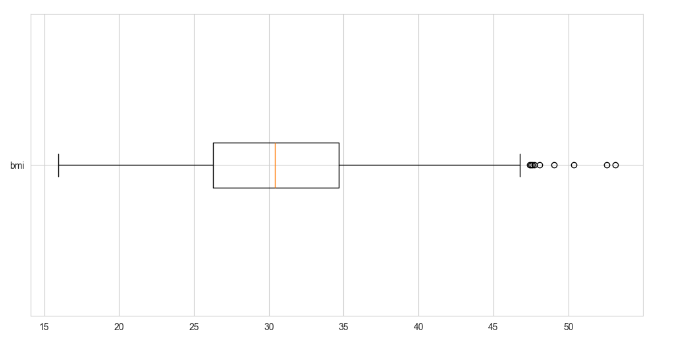


Рисунок 5 – Результат работы

**Вывод**

По графикам видно, что т.к. прямоугольники и усы симметричны, то данные распределяются симметрично, а также на графиках BMI и Charges видны выбросы, причем на втором их гораздо больше.

**Задание 6.**

Используя признак charges или imb, проверить, выполняется ли центральная предельная теорема. Использовать различные длины выборок n. Количество выборок = 300. Вывести результат в виде гистограмм. Найти стандартное отклонение и среднее для полученных распределений. Сделать выводы.

Для решения задачи был написан код (Листинг 6)

|  |
| --- |
| Листинг 6 - Задание 6 |
| num\_samples = 300  sample\_sizes = [30, 50, 70, 100, 120, 150, 200, 250]  fig, axes = plt.subplots(len(sample\_sizes), 1, figsize=(8, 12), sharex=True)  for i, n in enumerate(sample\_sizes):  sample\_means = []  for k in range(num\_samples):  sample = np.random.choice(dataset['charges'], size=n, replace=True)  sample\_mean = np.mean(sample)  sample\_means.append(sample\_mean)  axes[i].hist(sample\_means, bins=30, alpha=0.5)  axes[i].set\_title(f'n = {n}')  axes[i].set\_ylabel('Частота')  plt.xlabel('Средние значения')  means = [np.mean(sample\_means[i:i + num\_samples]) for i in range(0, len(sample\_means), num\_samples)] stds = [np.std(sample\_means[i:i + num\_samples]) for i in range(0, len(sample\_means), num\_samples)]  print(f"Средние значения средних значений выборок (charges):") print(means) print(f"Стандартные отклонения средних значений выборок charges):") print(stds)  plt.tight\_layout() plt.show() |

Результат работы программы представлены на рисунке 6.

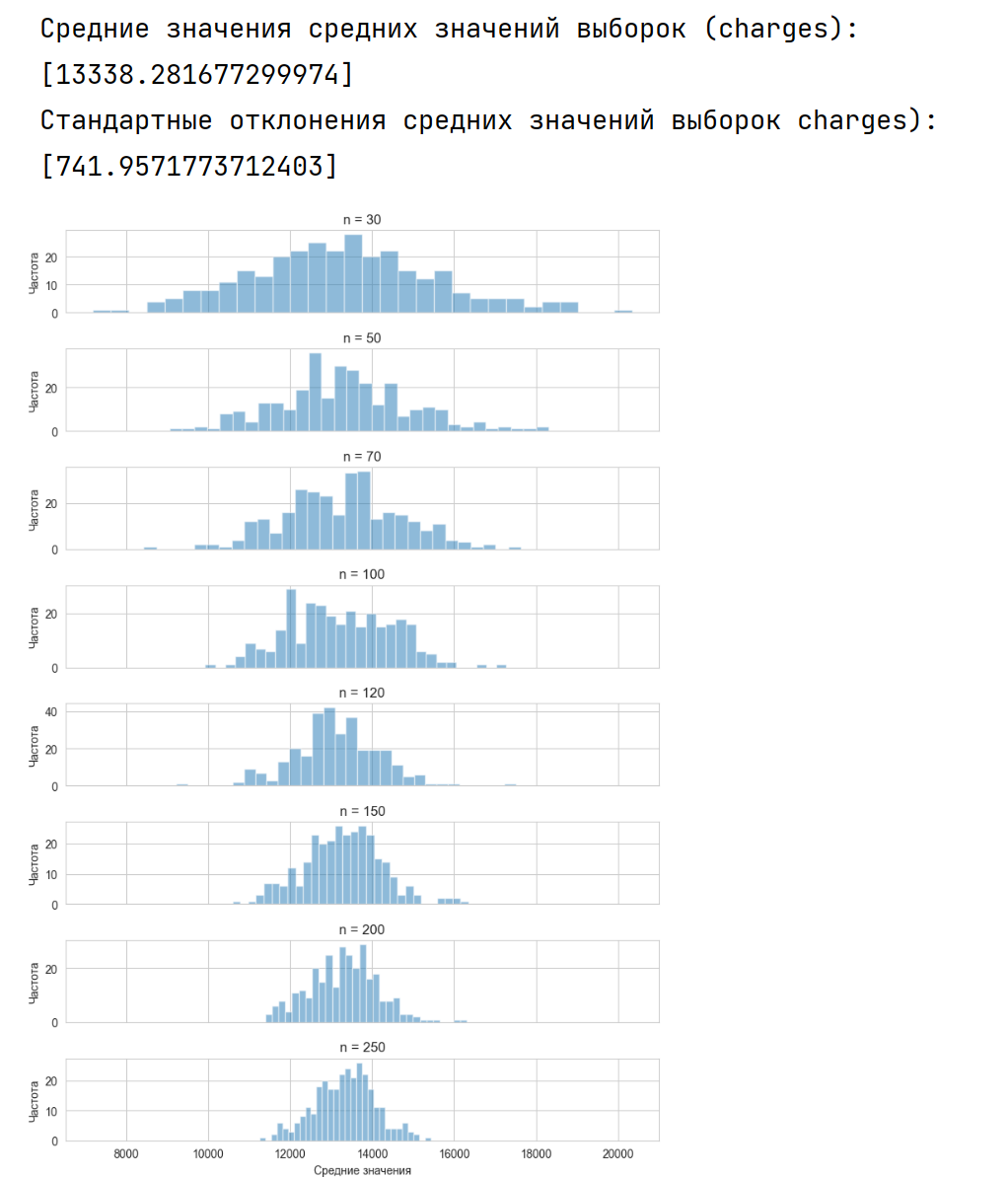


Рисунок 6 – Гистограммы с длинами выборок с 30 по 250

**Вывод**

Из данных гистограмм можно сделать вывод, что для признака IMB выполняется центральная предельная теорема и при увеличении длины выборок n, график всё больше аппроксимируется нормальным распределением, а также среднее таких выборок всё ближе к среднему генеральной совокупности.

**Задание 7.**

Построить 95% и 99% доверительный интервал для среднего значения расходов и среднего значения индекса массы тела.

Для решения задачи был написан код (Листинг 7)

|  |
| --- |
| Листинг 7 - Задание 7 |
| selected\_features = ['charges', 'bmi']  confidence\_levels = [0.95, 0.99]  confidence\_intervals = {}  for feature in selected\_features:  confidence\_intervals[feature] = {}  for confidence\_level in confidence\_levels:  data\_feature = dataset[feature]  n = len(data\_feature)  mean = np.mean(data\_feature)  std = np.std(data\_feature)   z\_score = stats.norm.ppf((1 + confidence\_level) / 2)   margin\_of\_error = z\_score \* (std / np.sqrt(n))   lower\_bound = mean - margin\_of\_error  upper\_bound = mean + margin\_of\_error   confidence\_intervals[feature][confidence\_level] = (lower\_bound, upper\_bound)  *# Вывод результатов* for feature, intervals in confidence\_intervals.items():  print(f"Доверительные интервалы для {feature}:")  for confidence\_level, interval in intervals.items():  print(f"Для {confidence\_level \* 100}% доверительный интервал: ({interval[0]}, {interval[1]})")  print() |

Результат работы программы представлены на рисунке 7.

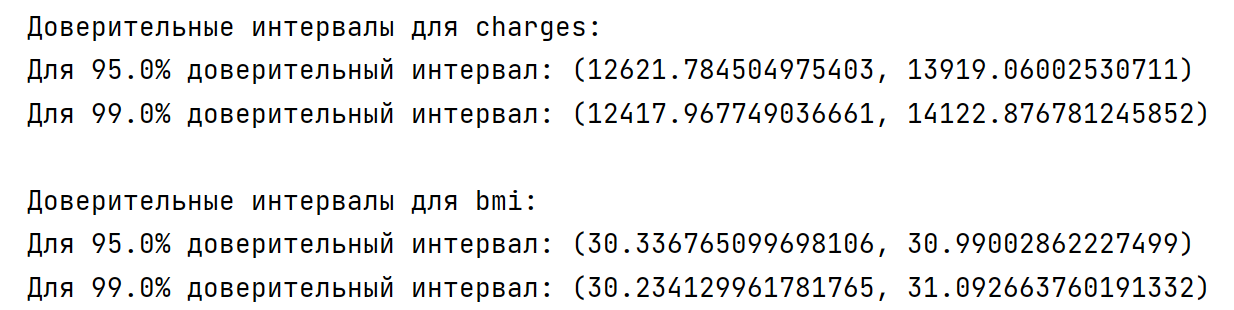


Рисунок 7 – Доверительные интервалы для BMI и Charges

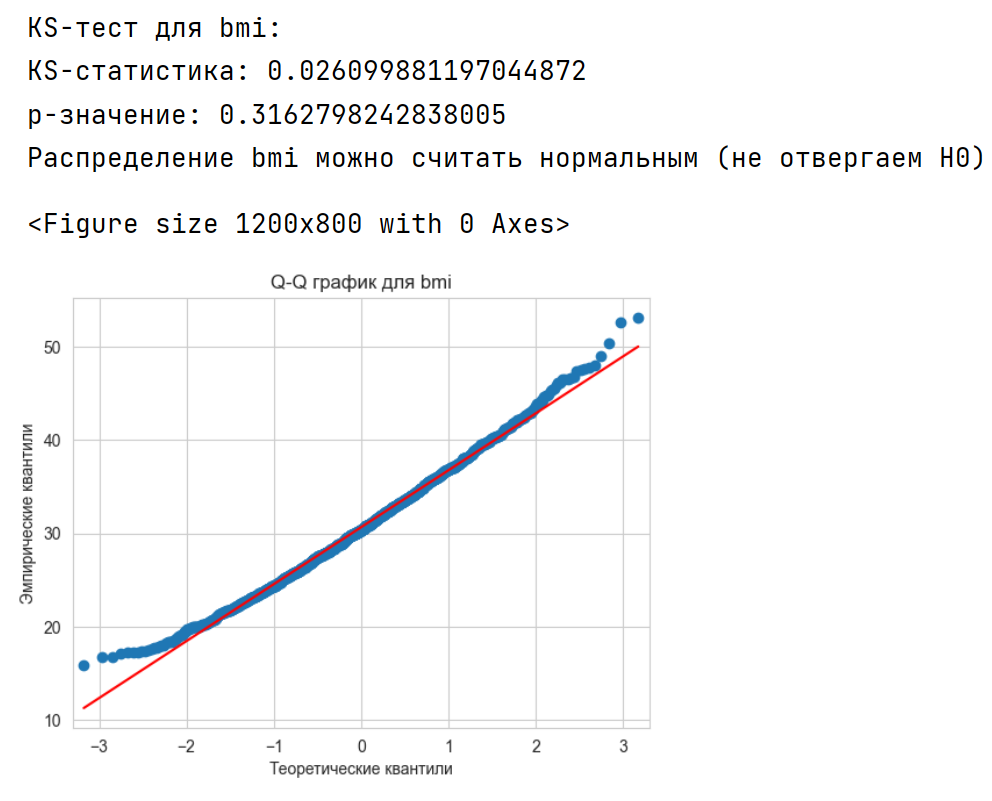
**Задание 8.**

Проверить распределения следующих признаков на нормальность: индекс массы тела, расходы. Сформулировать нулевую и альтернативную гипотезы. Для каждого признака использовать KS-тест и q-q plot. Сделать выводы на основе полученных p-значений.

Для решения задачи был написан код (Листинг 8)

|  |
| --- |
| Листинг 8 - Задание 8 |
| ksBMI = sts.kstest(data.bmi, 'norm')  selected\_features = ['bmi', 'charges']  alpha = 0.05  for feature in selected\_features:  ks\_statistic, ks\_p\_value = stats.kstest(dataset[feature], 'norm', (dataset[feature].mean(), dataset[feature].std()))   plt.figure(figsize=(12, 8))  sm.qqplot(dataset[feature], line='s')  plt.title(f'Q-Q график для {feature}')  plt.xlabel('Теоретические квантили')  plt.ylabel('Эмпирические квантили')   print(f'KS-тест для {feature}:')  print(f'KS-статистика: {ks\_statistic}')  print(f'p-значение: {ks\_p\_value}')   if ks\_p\_value > alpha:  print(f'Распределение {feature} можно считать нормальным (не отвергаем H0)')  else:  print(f'Распределение {feature} не является нормальным (отвергаем H0)')   plt.show() |

Результат работы программы представлены на рисунке 9.



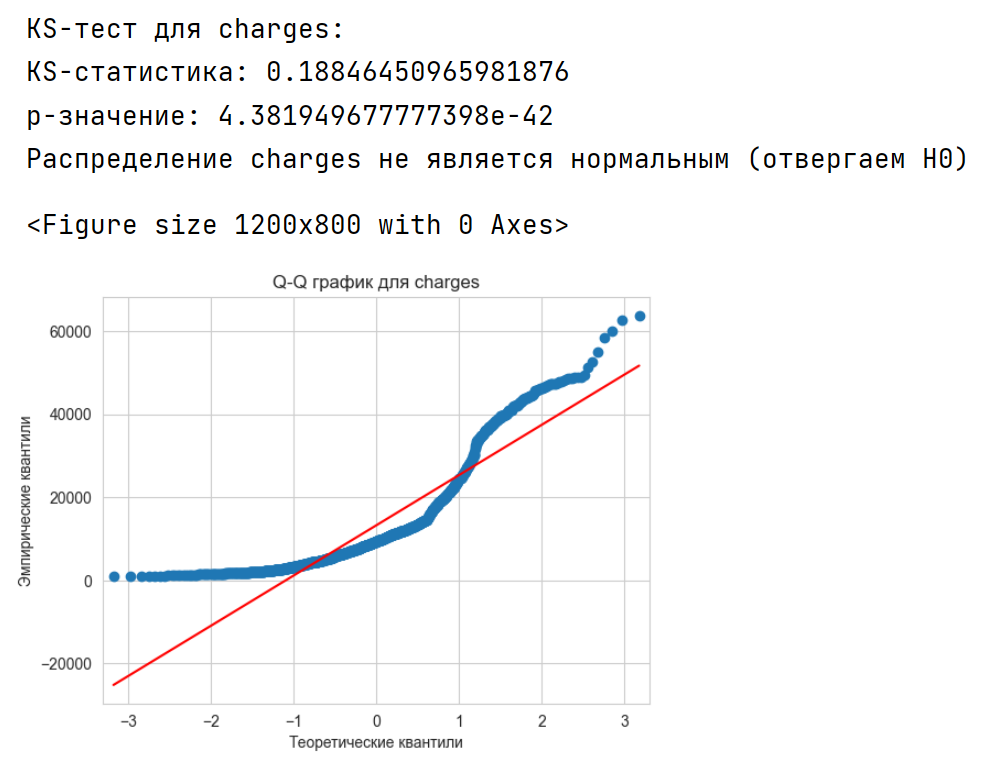


Рисунок 9 – Распределения BMI и Charges

**Вывод**

Если принять, что:нулевая гипотеза – распределение для этих признаков является нормальным, а альтернативная гипотеза – распределения для этих признаков не является нормальным, то по ks тестированию можно сделать однозначный вывод, что в обоих случаях можно отбросить нулевую гипотезу (p-значение <0.05), однако, по графику Q-Q plot для признака BMI может показаться, что данные практически идеально распределяются на прямой, однако, на концах исходные значения выше чем нормальные, что означает что данные также не распределяются ассиметрично, что означает, что нулевую гипотезу также можно откинуть.

**Задание 9.**

Загрузить данные из файла “ECDCCases.csv”.

Для решения задачи был написан код (Листинг 9)

|  |
| --- |
| Листинг 9 - Задание 9 |
| data\_set = pd.read\_csv('ECDCCases.csv') data\_set |

Результат работы программы представлены на рисунке 10.



Рисунок 10 – Подключение файла ECDCCases.csv

**Задание 10.**

Проверить в данных наличие пропущенных значений. Вывести количество пропущенных значений в процентах. Удалить два признака, в которых больше всех пропущенных значений. Для оставшихся признаков обработать пропуски: для категориального признака использовать заполнение значением по умолчанию (например, «other»), для числового признака использовать заполнение медианным значением. Показать, что пропусков больше в данных нет.

Для решения задачи был написан код (Листинг 10)

|  |
| --- |
| Листинг 10 - Задание 10 |
| *# Проверить в данных наличие пропущенных значений. # Вывести количество пропущенных значений в процентах.* missing\_percentage = (data\_set.isnull().sum() / len(data\_set)) \* 100 print("Процент пропущенных значений по каждому признаку:") missing\_percentage |

Результат работы программы представлены на рисунке 11.

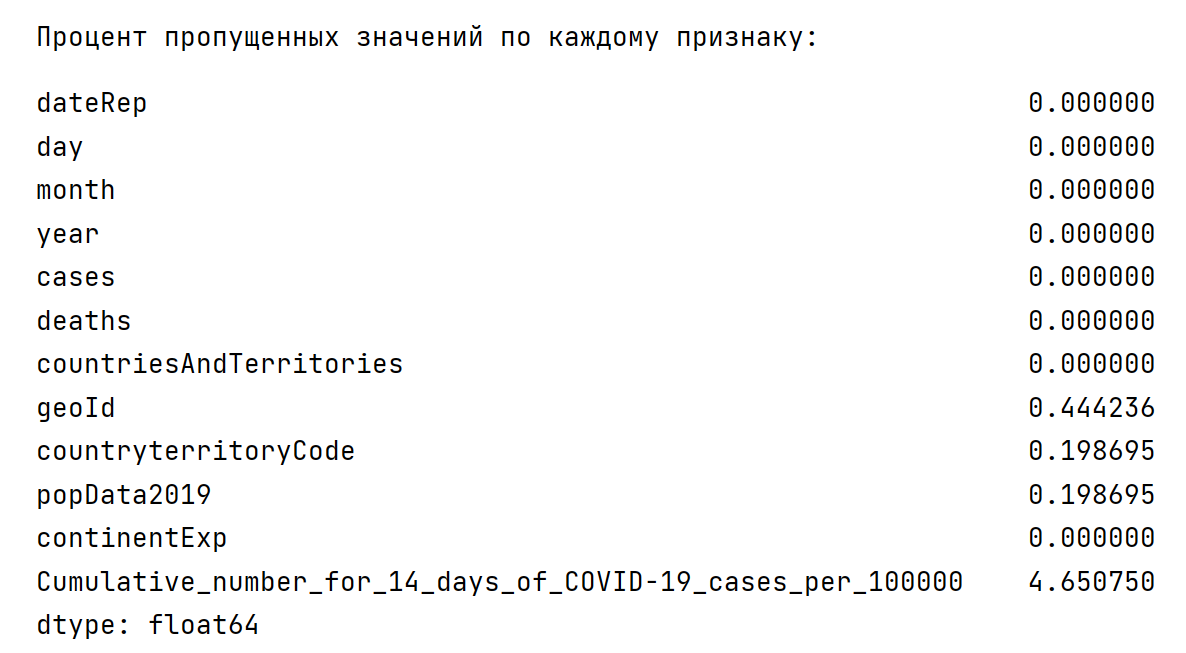


Рисунок 11 – Результат кода

|  |
| --- |
| Листинг 11 - Задание 10 |
| *# Удалить два признака, в которых больше всех пропущенных значений* columns\_to\_drop = missing\_percentage.nlargest(2).index  data\_set.drop(columns=columns\_to\_drop, inplace=True) columns\_to\_drop |

Результат работы программы представлены на рисунке 12.

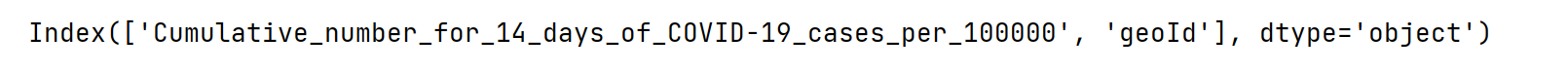


Рисунок 12 – Результат кода

|  |
| --- |
| Листинг 12 - Задание 10 |
| *# Для оставшихся признаков обработать пропуски: для категориального признака использовать заполнение значением по умолчанию # (например, «other»), для числового признака использовать заполнение медианным значением. Показать, что пропусков больше в данных нет* categorical\_columns = data\_set.select\_dtypes(include=['object']).columns data\_set[categorical\_columns] = data\_set[categorical\_columns].fillna('other')  numeric\_columns = data\_set.select\_dtypes(include=['int', 'float']).columns data\_set[numeric\_columns] = data\_set[numeric\_columns].fillna(data\_set[numeric\_columns].median()) data\_set |

Результат работы программы представлены на рисунке 13.



Рисунок 13 – Результат кода

|  |
| --- |
| Листинг 13 - Задание 10 |
| *# Показать, что пропусков больше в данных нет.* missing\_percentage = (data\_set.isnull().sum() / len(data\_set)) \* 100 missing\_percentage |

Результат работы программы представлены на рисунке 14.

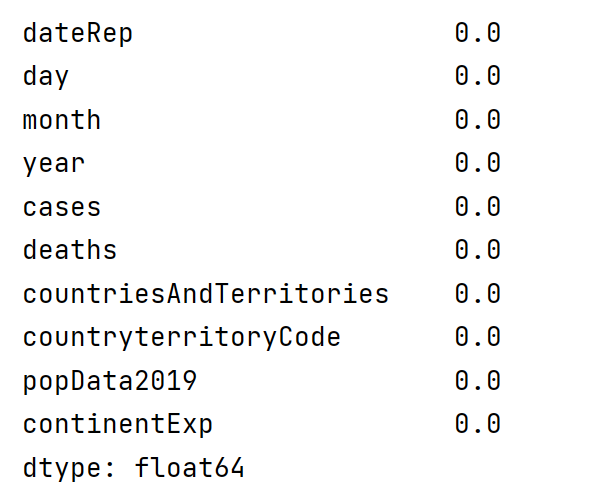


Рисунок 14 – Результат кода

**Задание 11.**

Посмотреть статистику по данным, используя describe(). Сделать выводы о том, какие признаки содержат выбросы. Посмотреть, для каких стран количество смертей в день превысило 3000 и сколько таких дней было.

Для решения задачи был написан код (Листинг 11-12)

|  |
| --- |
| Листинг 14 - Задание 11 |
| data\_description = data\_set.describe() data\_description |

Результат работы программы представлены на рисунке 15.

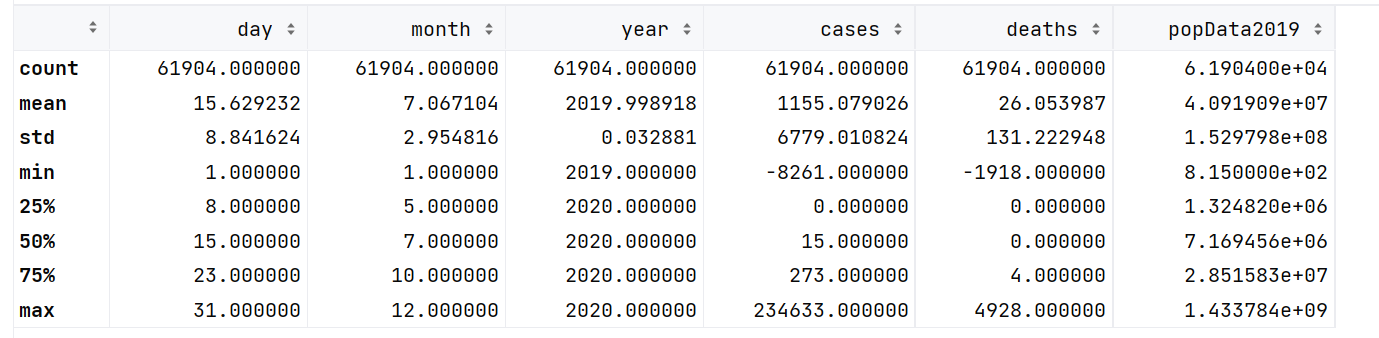


Рисунок 15 – Статистика данных через describe().

**Вывод**

Метод describe(), позволяет увидеть наиболее выделяющиеся выбросы, посредством проверки минимума и максимума и по данной таблице можно сделать вывод, что признаки «cases» и «death» содержать выбросы, т.к. их минимумы равны отрицательным числам, что не может быть действительностью в контексте смысла данных признаков.

|  |
| --- |
| Листинг 15 - Задание 11 |
| data\_set[data\_set['deaths'] > 3000]['countriesAndTerritories'].unique() |

Результат работы программы представлены на рисунке 16.

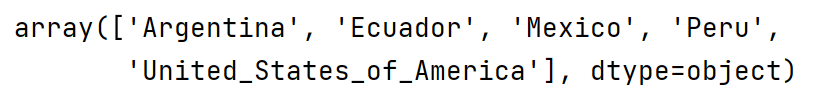


Рисунок 16 – Страны с кол-вом смертей >3000 и кол-во таких дней

**Задание 12.**

Найти дублирование данных. Удалить дубликаты.

Для решения задачи был написан код (Листинг 13-14)

|  |
| --- |
| Листинг 16 - Задание 12 |
| data\_set = data\_set.drop\_duplicates() data\_set |

Результат работы программы представлены на рисунке 17.



Рисунок 17 – Результат кода

**Задание 13.**

Загрузить данные из файла “bmi.csv”. Взять оттуда две выборки. Одна выборка – это индекс массы тела людей c региона northwest, вторая выборка – это индекс массы тела людей с региона southwest. Сравнить средние значения этих выборок, используя t-критерий Стьюдента. Предварительно проверить выборки на нормальность (критерий ШопироУилка) и на гомогенность дисперсии (критерий Бартлетта).

Для решения задачи был написан код (Листинг 17)

|  |
| --- |
| Листинг 17 - Задание 18 |
| df = pd.read\_csv('bmi.csv') df |

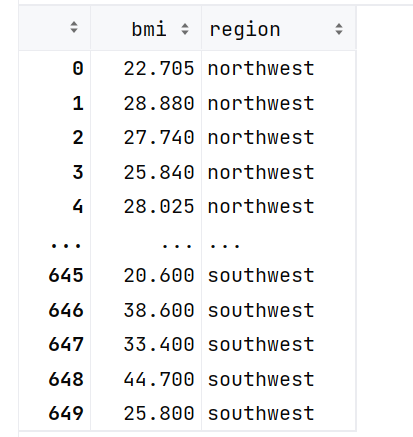


Рисунок 18 – Результат кода

|  |
| --- |
| Листинг 18 - Задание 18 |
| northwest\_bmi = df[df['region'] == 'northwest']['bmi'] southwest\_bmi = df[df['region'] == 'southwest']['bmi']  northwest\_bmi |

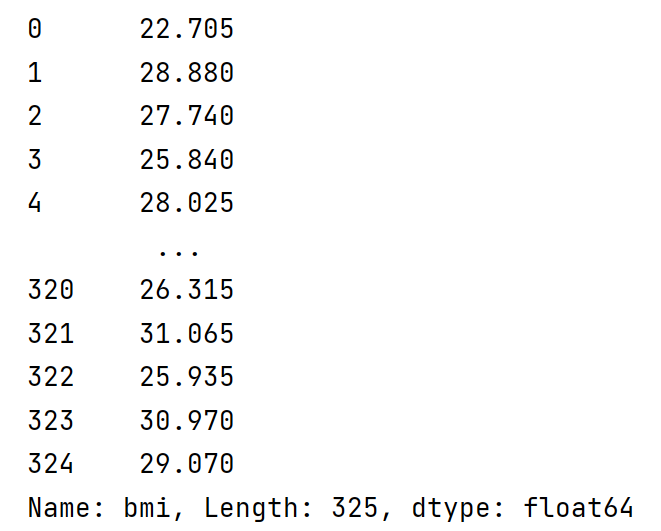


Рисунок 19 – Результат кода

|  |
| --- |
| Листинг 19 - Задание 18 |
| southwest\_bmi |

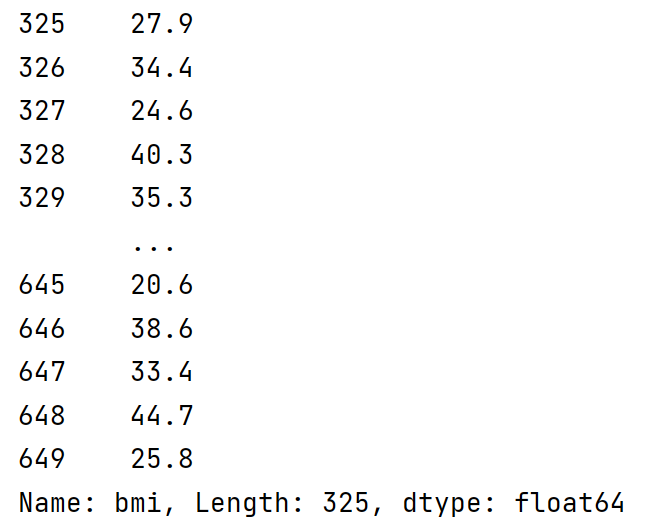


Рисунок 20 – Результат кода

**Задание 14.**

Кубик бросили 600 раз, получили следующие результаты: N Количество выпадений

|  |  |
| --- | --- |
| № | Количество выпадений |
| 1 | 97 |
| 2 | 98 |
| 3 | 109 |
| 4 | 95 |
| 5 | 97 |
| 6 | 104 |

С помощью критерия Хи-квадрат проверить, является ли полученное распределение равномерным. Использовать функцию scipy.stats.chisquare().

Для решения задачи был написан код (Листинг 16)

|  |
| --- |
| Листинг 20 - Задание 14 |
| p\_value\_nw = shapiro(northwest\_bmi) p\_value\_sw = shapiro(southwest\_bmi)  print(f'p-value для выборки из northwest: {p\_value\_nw}') print(f'p-value для выборки из southwest: {p\_value\_sw}')  *#Если значение p-value для обеих выборок больше 0.05, то можно предположить, что данные распределены нормально.* from scipy.stats import bartlett  p\_value\_bartlett = bartlett(northwest\_bmi, southwest\_bmi)  print(f'p-value критерия Бартлетта: {p\_value\_bartlett}')  *#Если значение p-value критерия Бартлетта больше 0.05, то можно предположить, что дисперсии в обеих выборках схожи (гомогенность дисперсии).* from scipy.stats import ttest\_ind  t\_stat, p\_value = ttest\_ind(northwest\_bmi, southwest\_bmi)  print(f'Значение t-статистики: {t\_stat}') print(f'p-value t-теста: {p\_value}')  *#Если значение p-value меньше выбранного уровня значимости (например, 0.05), то можно сделать вывод о статистической разнице между средними значениями двух выборок.  #Обратите внимание, что для корректного применения t-критерия Стьюдента, данные должны соответствовать условиям нормального распределения и гомогенности дисперсии.* from scipy.stats import chisquare  expected\_counts = [100] \* 6  observed\_counts = [97, 98, 109, 95, 97, 104]  chi\_square\_stat, p\_value = chisquare(f\_obs=observed\_counts, f\_exp=expected\_counts)  print(f'Значение статистики Хи-квадрат: {chi\_square\_stat}') print(f'p-value: {p\_value}') |

Результат работы программы представлены на рисунке 21.

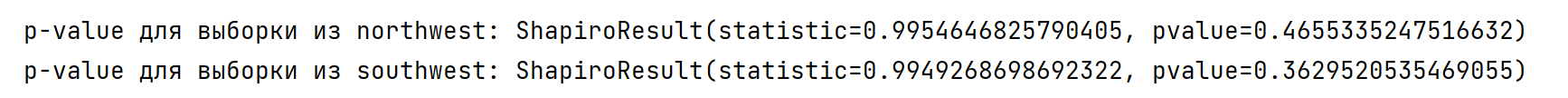








Рисунок 21 – Создание фрейма и проверка его критерием Хи-квадрат

**Вывод**

Исходя из полученного p-значения, т.к. оно больше 0.05, можно сделать вывод, что данное распределение является равномерным.

**Задание 15.**

С помощью критерия Хи-квадрат проверить, являются ли переменные зависимыми. Создать датафрейм, используя следующий код:

data = pd.DataFrame({'Женат': [89,17,11,43,22,1],

'Гражданский брак': [80,22,20,35,6,4],

'Не состоит в отношениях': [35,44,35,6,8,22]})

data.index = ['Полный рабочий день','Частичная занятость','Временно не работает','На домохозяйстве','На пенсии','Учёба']

Использовать функцию scipy.stats.chi2\_contingency(). Влияет ли семейное положение на занятость?.

Для решения задачи был написан код (Листинг 17-18)

|  |
| --- |
| Листинг 21 - Задание 15 |
| data = pd.DataFrame({'Женат': [89, 17, 11, 43, 22, 1],  'Гражданский брак': [80, 22, 20, 35, 6, 4],  'Не состоит в отношениях': [35, 44, 35, 6, 8, 22]}) data.index = ['Полный рабочий день', 'Частичная занятость', 'Временно не работает', 'На домохозяйстве', 'На пенсии', 'Учёба']  chi2\_stat, p\_value, dof, expected = chi2\_contingency(data)  print(f'Значение статистики Хи-квадрат: {chi2\_stat}') print(f'p-value: {p\_value}') print(f'Степени свободы: {dof}') |

Результат работы программы представлены на рисунке 22.

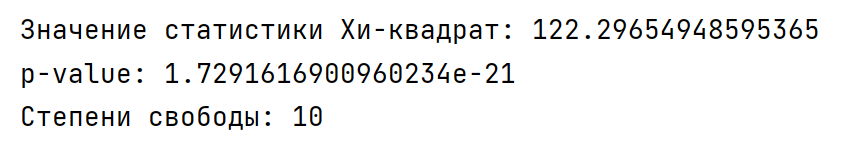


Рисунок 22 – Созданная таблица

**Вывод**

Исходя из полученного p-значения, т.к. оно больше 0.05, можно сделать вывод, что семейное положение влияет на занятость.