

Московский государственный университет
имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математической статистики

Отчет
по технологическому практикуму

Выполнил:
студент 316 группы
Черкасов Б. Ю.
Преподаватель:
Горшенин А. К.

Москва 2024

Содержание

1. Введение

- Цели и задачи работы

2. Первый этап практикума

2.1 Подготовка данных

- Выбор датасетов и их описание
- Краткий анализ (минимумы, максимумы, средние значения, пропуски)

2.2 Аппроксимация распределений данных

- Ядерные оценки плотности распределения

2.3 Визуальный анализ данных

- Построение cdplot, dotchartm boxplot, stripchart
- Выявление выбросов и их проверка
(тесты Граббса и Дискона)

2.4 Заполнение пропусков и сравнение результатов

- Заполнение пропусков в данных
- Сравнение результатов заполнения с истинными значениями

2.5 Генерация нормального распределения и анализ.

- Генерация выборок малого(50-100 элементов) и умеренного объемов(1000-5000 наблюдений).
- Анализ с помощью графиков эмпирических функций распределения, квантилей, метода огибающих и других стандартных процедур проверки гипотез о нормальности.

2.6. Демонстрация примера

анализа данных с помощью графиков квантилей, метода огибающих, а также стандартных процедур проверки гипотез о нормальности данных на выборках малого и умеренного объемов.

3. Второй этап практикума.

Применение для проверки гипотез на различных доверительных уровнях (0.9, 0.95, 0.99) статистических критериев.

3.1 Проверка гипотез о равенстве средних и дисперсий.

- Стьюдента, включая односторонние варианты, когда проверяемая нулевая гипотеза заключается в том, что одно из сравниваемых средних значений больше (или меньше) другого.

Оценка мощности критериев при заданном объеме выборки или определения объема выборки для достижения заданной мощности;

- Уилкоксона-Манна-Уитни (ранговые);
- Фишера, Левене, Бартлетта, Флигнера-Килина (проверка гипотез об однородности дисперсий).

3.2 Исследование корреляционных взаимосвязей в данных.

- Корреляция Пирсона
- Корреляция Спирмена
- Корреляция Кендалла

3.3 Исследование значимой взаимосвязи между данными.

- Критерий согласия Пирсона (хи-квадрат)
- Точный тест Фишера
- Тест МакНемара
- Тест Кохрана-Мантеля-Хензеля

3.4 Проверка наличия мультиколлинеарности в данных.

- Корреляционная матрица
- Фактор инфляции дисперсии

3.5 Дисперсионный анализ

3.6 Подгон регрессионных (линейных и нелинейных) моделей к данным и оценка качества аппроксимации.

4. Заключение

5. Список литературы

1. Введение

В современном мире обработка и анализ данных занимают ключевое место в решении прикладных задач в различных областях науки и техники. Эффективное использование методов статистического анализа и машинного обучения позволяет выявлять закономерности, прогнозировать тенденции и принимать обоснованные решения.

В данной работе основное внимание уделено изучению и применению методов анализа данных с использованием языков программирования Python и R. Эти языки являются мощными инструментами для статистического моделирования, визуализации и работы с большими объемами данных.

Работа состоит из двух этапов. На **первом этапе** рассматриваются базовые методы анализа данных, включая аппроксимацию распределений, визуализацию данных и идентификацию выбросов.

Второй этап направлен на проверку гипотез, анализ корреляционных зависимостей и построение регрессионных моделей.

Таким образом, работа позволяет не только изучить основные методы анализа данных, но и освоить практические навыки их применения.

При реализации программного кода использовалась среда Jupyter Notebook. В работе приведена визуализация результатов в порядке: **Python, затем R**.

К данному отчету прилагаются файлы с исходным кодом и html версиями на двух языках для более подробного ознакомления.

2. Первый этап практикума

2.1 Подготовка данных

Для анализа данных и проверки различных гипотез были выбраны датасеты с сайта kaggle. Ниже приведено краткое описание каждого из датасетов, которые были использованы для проведения тестов. **Все ссылки** на датасеты можно найти в **файлах с исходным кодом формата .ipynb**.

I Датасет содержит информацию об олимпийских спортсменах за 126 лет от первых современных Олимпийских игр в 1896 году в Афинах до зимних Олимпийских игр в Пекине. Он содержит информацию о стране, росте, поле, весе и виде спорта спортсмена.

Также можно выделить другие переменные, такие как **id**, имя, описание качеств спортсмена, примечания, дата рождения. Для упрощения работы изначальный датасет был разделен на два датасета для разного пола спортсмена.

В изначальном датасете присутствует много пропусков во всех столбцах, поэтому удаляем их для дальнейшей эффективной работы.

Статистические характеристики:

- Средние значения по росту спортсменов(муж. - 180, жен. - 170), весу(муж. - 77, жен. - 63)
- Стандартные отклонения роста и веса мужчин: 9 и 12. У женщин:
- Минимальные значения и максимальные значения:
 - Мужчины: min рост - 155, max рост - 226, min вес - 48, max вес - 141
 - Женщины: min рост - 152, max рост - 201, min вес - 45, max вес - 135

Мужчины:

	height	weight	
count	705.000000	705.000000	
mean	180.248227	77.251064	
std	8.896785	12.099174	
min	155.000000	48.000000	
25%	175.000000	69.000000	
50%	180.000000	76.000000	
75%	185.000000	84.000000	
max	226.000000	141.000000	
country	height	weight	sport
Length:829	Min. :155	Min. : 48.0	Length:829
Class :character	1st Qu.:175	1st Qu.: 70.0	Class :character
Mode :character	Median :180	Median : 77.0	Mode :character
	Mean :181	Mean : 77.8	
	3rd Qu.:187	3rd Qu.: 84.0	
	Max. :226	Max. :141.0	

Женщины:

	height	weight	
count	268.000000	268.000000	
mean	170.712687	63.406716	
std	8.648110	10.879433	
min	152.000000	45.000000	
25%	165.000000	57.000000	
50%	170.000000	62.000000	
75%	175.000000	67.250000	
max	201.000000	135.000000	
country	height	weight	sport
Length:321	Min. :150.0	Min. : 45.00	Length:321
Class :character	1st Qu.:165.0	1st Qu.: 57.00	Class :character
Mode :character	Median :170.0	Median : 62.00	Mode :character
	Mean :170.8	Mean : 62.95	
	3rd Qu.:175.0	3rd Qu.: 67.00	
	Max. :201.0	Max. :135.00	

II Датасет содержит информацию о росте студентов в высшей школе для мальчиков и девочек. Особенность выбора данного датасета заключается в том, что данные были сгенерированы из нормального распределения с разными наборами средних и стандартных отклонений.

Данный датасет будет использован в пункте **2.6** при анализе распределения и проверки гипотез о нормальности.

Статистические характеристики:

- Средние значения роста мальчиков - 67, девочек - 61;
- Минимальные и максимальные значения роста:
 - Мальчики: min - 59, max - 77;
 - Девочки: min - 56, max - 69

В данных отсутствуют пропуски, что позволяет сразу приступить к анализу распределения и проверке гипотез о нормальности.

	boys	girls	boys	girls
count	1000.000000	1000.000000	Min. :59.16	Min. :56.68
mean	67.163860	61.977600	1st Qu.:65.18	1st Qu.:60.60
std	2.888716	2.116831	Median :67.13	Median :62.01
min	59.160000	56.680000	Mean :67.16	Mean :61.98
25%	65.177500	60.600000	3rd Qu.:68.95	3rd Qu.:63.38
50%	67.130000	62.005000	Max. :77.15	Max. :69.35
75%	68.952500	63.380000	NA's :500	NA's :500
max	77.150000	69.350000	Рост мальчиков. Размер - 100	

III Датасет - это огромная база данных о фильмах, предоставляющая информацию о названии фильма, рейтинге, датах релиза, доходах, жанрах и других показателей. В данных присутствуют пропуски(столбцы ‘title’, ‘release_date’, ‘homepage’, ‘overview’, ‘genres’ и др.), поэтому перед началом работы удаляем их.

Статистические характеристики представляют собой огромные значения, среди них можно выделить ключевые средние значения: бюджет порядка 2.6 млн., доход порядка 7 млн., кол-во отзывов - около 2 млн.

	id	vote_average	vote_count	revenue	runtime	budget	popularity
count	1.133847e+06	1.133847e+06	1.133847e+06	1.133847e+06	1.133847e+06	1.133847e+06	1.133847e+06
mean	7.390196e+05	1.884088e+00	1.891690e+01	6.899139e+05	4.781688e+01	2.687800e+05	1.236676e+00
std	3.949960e+05	3.023154e+00	3.189010e+02	1.788914e+07	6.167366e+01	5.076022e+06	7.596361e+00
min	2.000000e+00	0.000000e+00	0.000000e+00	-1.200000e+01	-2.800000e+01	0.000000e+00	0.000000e+00
25%	4.100815e+05	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	6.000000e-01
50%	7.438760e+05	0.000000e+00	0.000000e+00	0.000000e+00	2.300000e+01	0.000000e+00	6.000000e-01
75%	1.089520e+06	4.800000e+00	1.000000e+00	0.000000e+00	8.900000e+01	0.000000e+00	8.770000e-01
max	1.389172e+06	1.000000e+01	3.449500e+04	3.000000e+09	1.440000e+04	1.000000e+09	2.994357e+03

	id	title	vote_average	vote_count
Min. :	2	Length:10000	Min. : 2.098	Min. : 256
1st Qu.:	10164	Class :character	1st Qu.: 6.100	1st Qu.: 398
Median :	36563	Mode :character	Median : 6.649	Median : 714
Mean :	180969		Mean : 6.624	Mean : 1768
3rd Qu.:	339528		3rd Qu.: 7.201	3rd Qu.: 1707
Max. :	1140066		Max. : 9.172	Max. : 34495

		budget	popularity
revenue	runtime	Min. : 0	Min. : 0.60
Min. : 0.000e+00	Min. : 0.0	1st Qu.: 0	1st Qu.: 11.96
1st Qu.: 0.000e+00	1st Qu.: 93.0	Median : 6000000	Median : 15.94
Median : 9.498e+06	Median : 103.0	Mean : 21914052	Mean : 23.49
Mean : 6.433e+07	Mean : 105.2	3rd Qu.: 26000000	3rd Qu.: 23.09
3rd Qu.: 5.747e+07	3rd Qu.: 116.0	Max. : 460000000	Max. : 2994.36
Max. : 2.924e+09	Max. : 366.0		

IV Датасет содержит данные об успеваемости учеников в двух португальских школах, такие как оценки учащихся, их демографические, социальные и другие, относящиеся к учебе признаки. Данный датасет будет применен в пункте **3.6** для подгонки регрессионных моделей к нему.

Как будет показано, данный датасет имеет особенности для регрессионного анализа, что демонстративно покажет значимость предварительного анализа данных перед обучением модели в машинном обучении.

В качестве ключевых были отобраны следующие: ‘Hours Studied’, ‘Previous Scores’, ‘Extracurricular Activities’, ‘Sleep Hours’, ‘Sample Question Papers Practiced’, ‘Performance Index’. В данных пропуски отсутствуют.

Статистические характеристики:

	Hours Studied	Previous Scores	Sleep Hours	Sample Question Papers Practiced	Performance Index
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	4.992900	69.445700	6.530600	4.583300	55.224800
std	2.589309	17.343152	1.695863	2.867348	19.212558
min	1.000000	40.000000	4.000000	0.000000	10.000000
25%	3.000000	54.000000	5.000000	2.000000	40.000000
50%	5.000000	69.000000	7.000000	5.000000	55.000000
75%	7.000000	85.000000	8.000000	7.000000	71.000000
max	9.000000	99.000000	9.000000	9.000000	100.000000

Hours.Studied	Previous.Scores	Extracurricular.Activities	Sleep.Hours
Min. :1.000	Min. :40.00	Length:10000	Min. :4.000
1st Qu.:3.000	1st Qu.:54.00	Class :character	1st Qu.:5.000
Median :5.000	Median :69.00	Mode :character	Median :7.000
Mean :4.993	Mean :69.45		Mean :6.531
3rd Qu.:7.000	3rd Qu.:85.00		3rd Qu.:8.000
Max. :9.000	Max. :99.00		Max. :9.000
Sample.Question.Papers.Practiced	Performance.Index		
Min. :0.000	Min. : 10.00		
1st Qu.:2.000	1st Qu.: 40.00		
Median :5.000	Median : 55.00		
Mean :4.583	Mean : 55.22		
3rd Qu.:7.000	3rd Qu.: 71.00		
Max. :9.000	Max. :100.00		

2.2 Аппроксимация распределений данных с помощью ядерных оценок

Ядерная оценка плотности (KDE) используется в случаях, когда нужно сделать заключение о распределении данных, это непараметрический способ оценки плотности случайной величины, что означает, что он делает предположений о конкретной форме распределения.

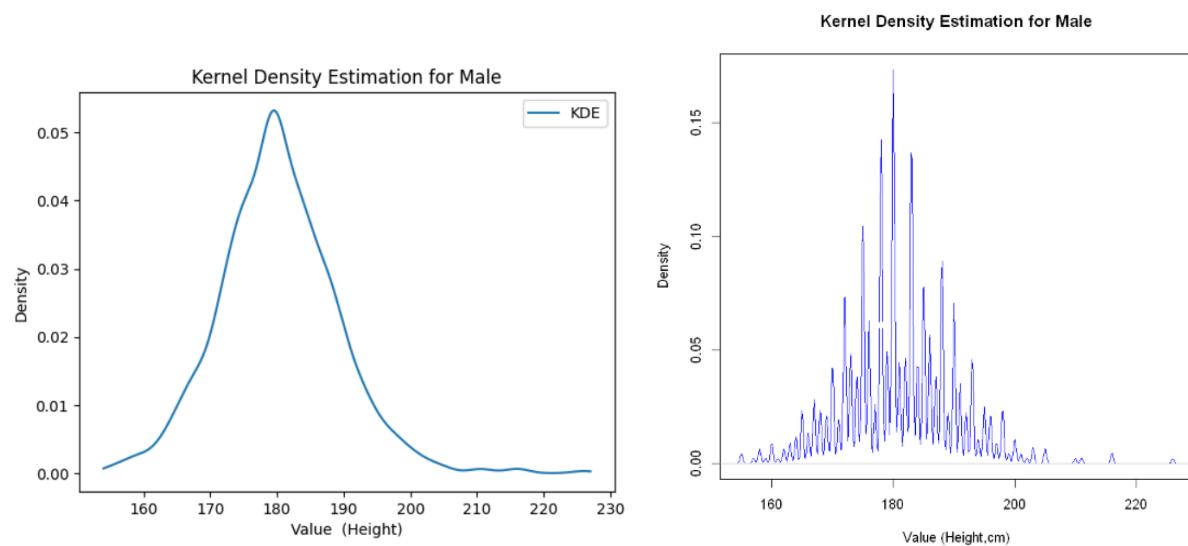
По сути, это задача сглаживания данных, когда делается заключение о совокупности, основываясь на конечной выборке данных. Она представляет данные с помощью непрерывной кривой плотности вероятности в одной или нескольких измерениях.

Основная идея заключается в том, чтобы для каждого наблюдения создать маленькую “ядровую” функцию(обычно, берется гаусс), а затем суммировать эти функции для всех точек выборки. Таким образом получается плавная оценка плотности, которая приближает истинное распределение данных.

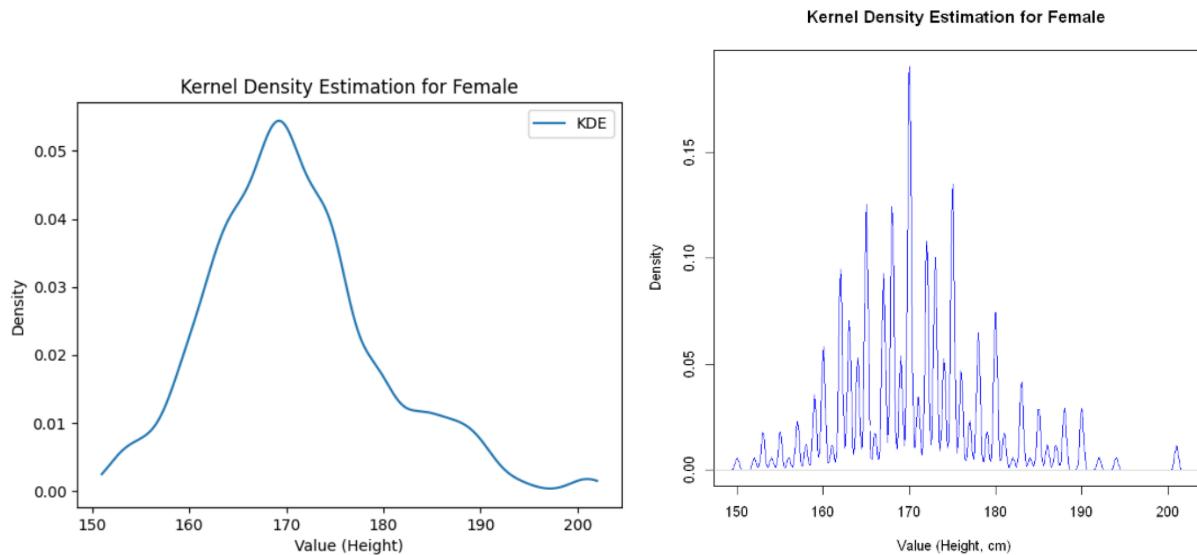
Она применяется в статистике и анализе данных для построения эмпирической плотности распределения и для визуализации данных.

Построим ядерную оценку для распределений роста спортсменов для каждого пола из **I датасета (см. пункт 2.1)**.

Мужчины:



Женщины:



Можно сделать следующие выводы:

Мужчины:

- Ядерная оценка показывает одно модальное распределение (одна вершина), что указывает на однородную группу спортсменов;
- Пик плотности распределения для мужчин находится в районе 180 см., что можно интерпретировать как наиболее часто встречающийся рост среди мужчин;
- Распределение симметричное с небольшим смещением вправо, что говорит о наличии спортсменов с ростом выше среднего;
- График справа (на R): более детализированное распределение локальных пиков указывает на влияние подгрупп спортсменов.

Женщины:

- Распределение также модальное, но пищевая плотность чуть более выражена, что указывает на меньший разброс роста среди женщин;
- Пик распределения наблюдается в районе 165-170 см, что соответствует наиболее часто встречающемуся росту у женщин;
- Распределение чуть более вытянуто вправо, что говорит о наличии высоких спортсменок, но в меньшем кол-ве;

- График справа (на R): аналогично мужчинам, наблюдаются мелкие локальные пики, которые могут быть связаны с разными спортивными дисциплинами.

В общем можно сделать вывод об однородности распределения спортсменов мужчин и женщин. Разброс у мужчин больше, чем у женщин, что может быть связано с разнообразием спортивных дисциплин, где допускаются спортсмены с разным ростом. Графики с меньшим сглаживанием показывают наличие больших локальных группировок, что может быть связано с особенностями определенных видов спорта (баскетболисты, хоккеисты, гимнасты.)

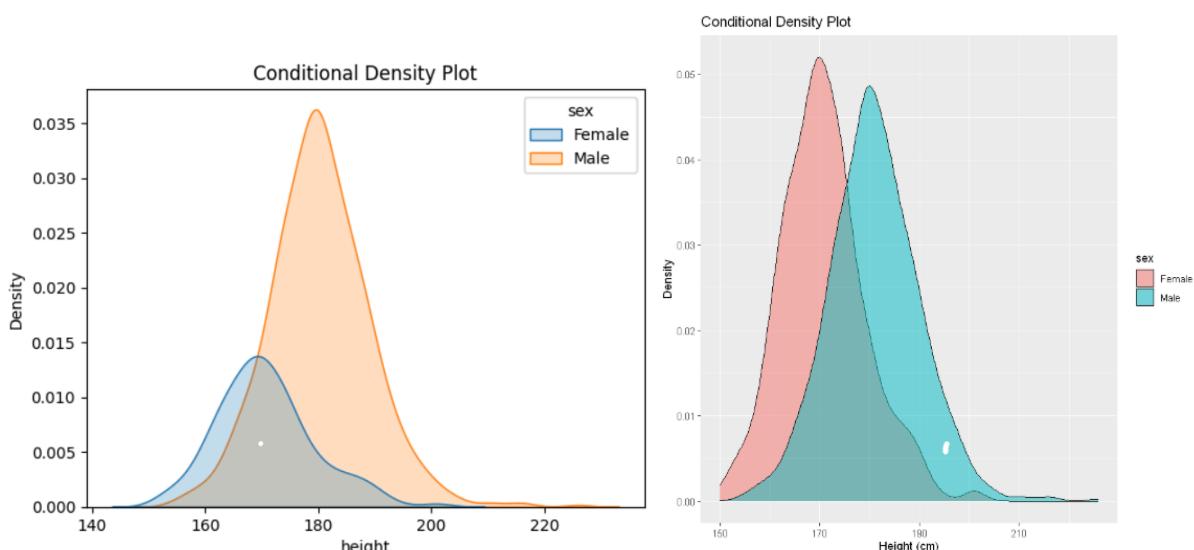
2.3 Визуальный анализ данных

- **cdplot, dotchart, boxplot, stripchart**

Везде ниже используется I датасет (см. пункт 2.1).

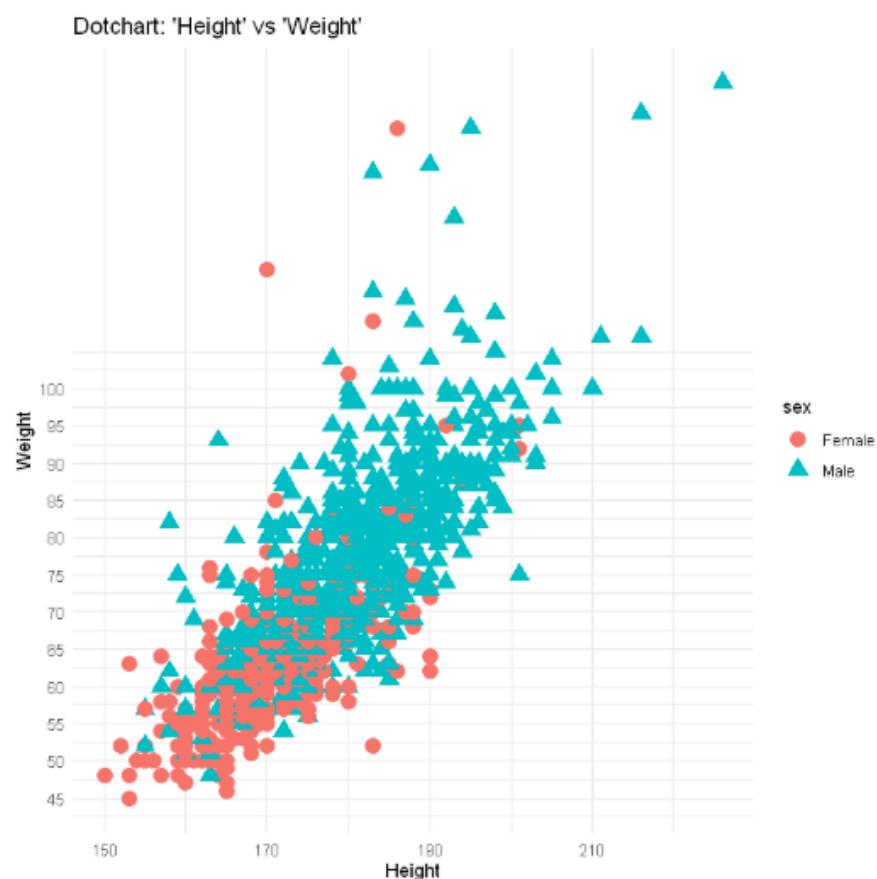
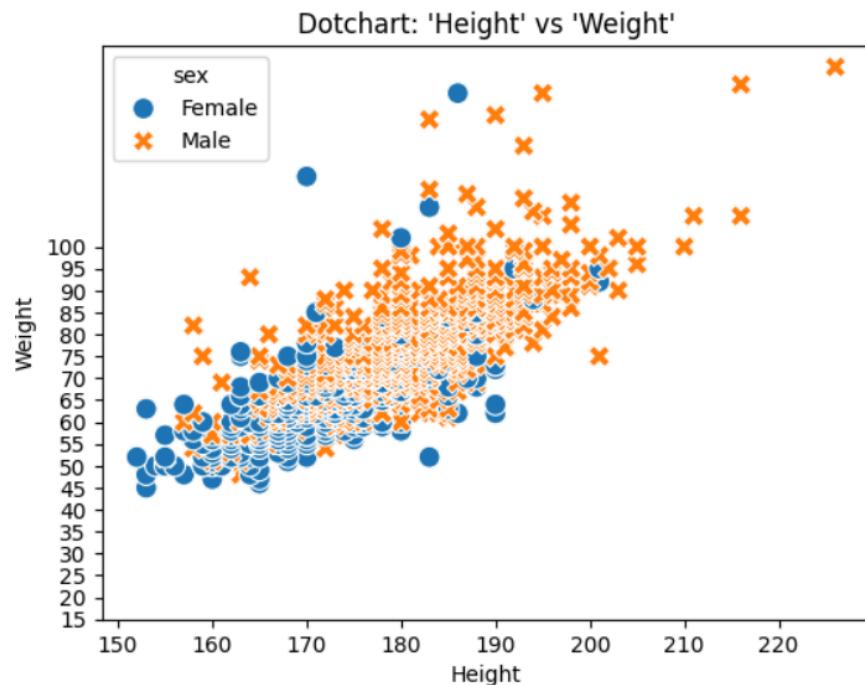
CDPlot (Cumulative Distribution Plot) - график, который отражает совокупное распределение данных. Он строится путем отображения на оси X всех значений выборки, а на оси Y - частоты этих значений(их пропорции от общего числа). Это позволяет визуализировать, как данные распределены и как быстро они накапливаются в разных диапазонах.

Изобразим этот график для роста спортсменов по их полу.



Dotchart (точечная диаграмма) - график, на котором каждая точка соответствует одному наблюдению. Точки располагаются на оси Y (или X) в зависимости от значения переменной. Он полезен для отображения распределения небольших наборов данных или для выявления выбросов.

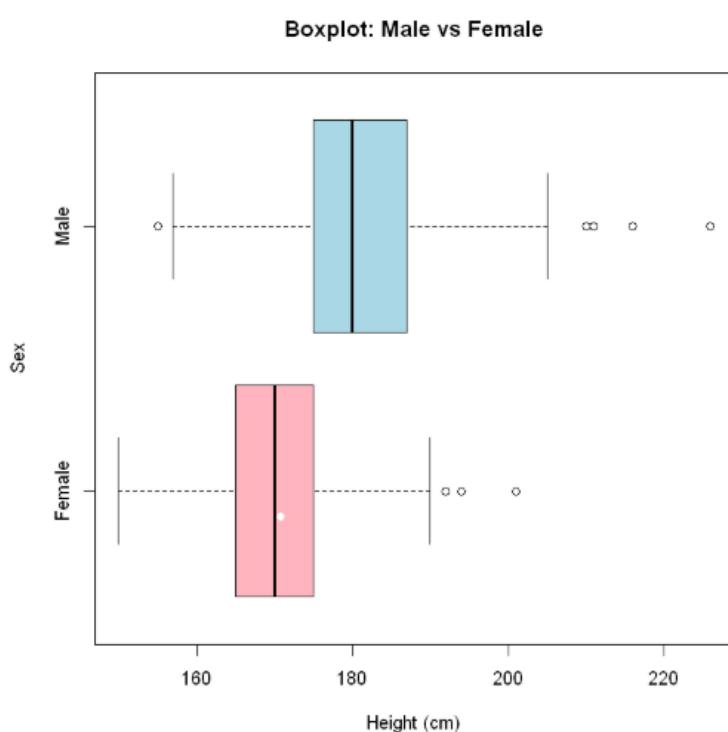
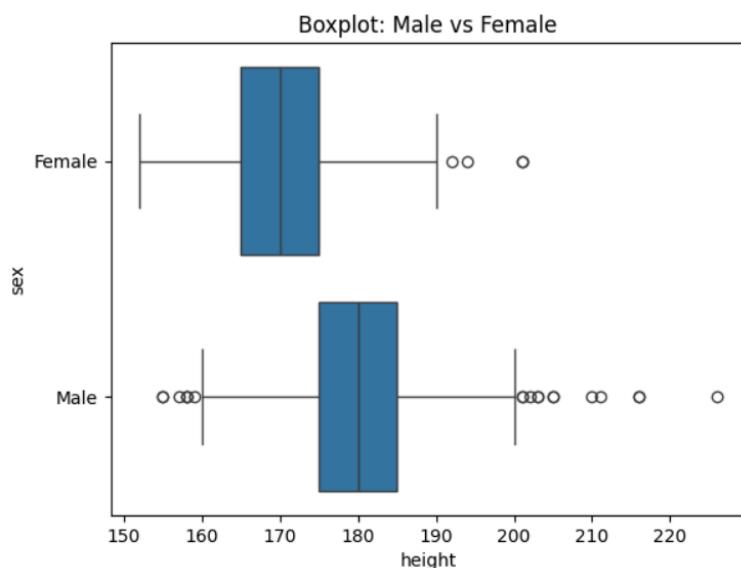
Изобразим этот график для веса и роста спортсменов:



Boxplot (коробка с усами) - эта “коробка” используется для отображения распределения данных, через их квартили(диапазоны значений, обычно их 4). Он отображает медиану, первый(до 25%) и третий квартили(до 75%), а также выбросы. “Усы”(границы коробки) показывают диапазон значений, которые не считаются выбросами.

Помогает быстро оценить распределение, медиану, выбросы, это полезно при сравнении распределений нескольких выборок или при анализе данных на наличие аномальных значений.

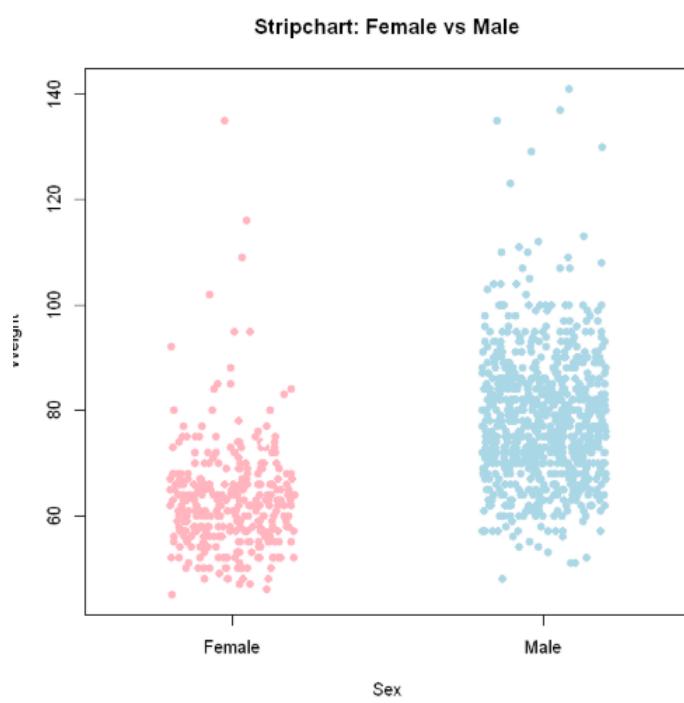
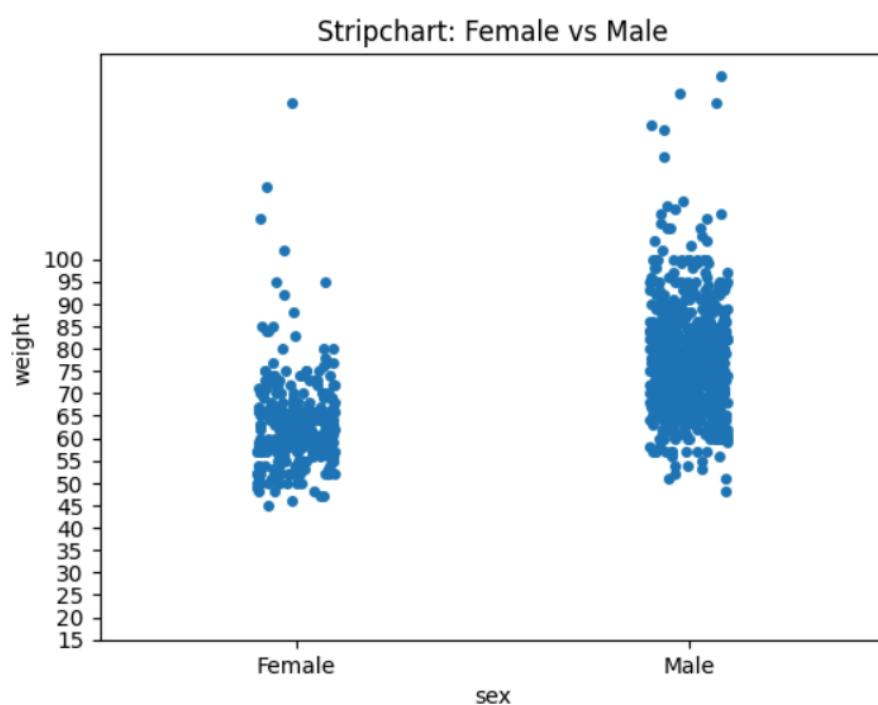
Сравним распределения роста спортсменов обоих полов:



Stripchart (точечный график с полосами) - график, представляющий собой вариацию **dotchart**, где каждый наблюдаемый элемент отображается точкой на оси, и эти точки могут быть дополнительно размечены, в виде полос или горизонтальных линий, если это необходимо для выделения групп.

Он позволяет выявить аномалии, сравнивая группы или категории.

Построим его для веса спортсменов:



Из графиков можно сделать следующие выводы:

Stripchart веса спортсменов (распределение по полу):

- Визуализация показывает, что мужчины в среднем имеют больший вес по сравнению с женщинами. Основная масса данных сосредоточена в пределах 60–90 кг для женщин и 70–100 кг для мужчин.
- У мужчин наблюдаются выбросы с весом выше 100 кг. У женщин также присутствуют выбросы, но их значительно меньше, и они менее экстремальны.

График условной плотности распределения роста (conditional density plot):

- Повторяет тенденцию, что мужчины выше женщин. Распределение плотностей также показывает, что доля высоких людей (>190 см) среди мужчин больше, чем среди женщин.
- Распределение женщин более сдвинуто влево, что подтверждает меньшую среднюю величину роста.

Dotchart зависимости веса от роста:

- Наблюдается явная положительная зависимость между ростом и весом как у мужчин, так и у женщин: более высокие спортсмены, как правило, весят больше.
- Мужчины занимают верхние границы роста и веса, а женщины — нижние.
- Некоторые выбросы (особенно у мужчин с весом больше 100 кг и ростом больше 200 см) могут быть связаны с определенными спортивными дисциплинами (например, тяжелая атлетика или баскетбол).

График "ящик с усами":

- Средний рост мужчин составляет около 180 см, а женщин — около 170 см.
- У мужчин выбросы начинаются после 200 см, а у женщин после 190 см.

- Диапазон роста у женщин более узкий, чем у мужчин, что подтверждает меньшую вариативность.

Общий вывод:

- Рост и вес: Мужчины в среднем выше и тяжелее женщин. У мужчин больше вариативность в росте и весе, что может быть связано с особенностями разных видов спорта.
- Выбросы: У мужчин больше выбросов, особенно среди спортсменов с экстремально большим ростом (>200 см) и весом (>100 кг). У женщин выбросы менее выражены.
- Зависимость роста и веса: Явная линейная зависимость между ростом и весом подтверждается для обоих полов, с заметным сдвигом вверх для мужчин.
- Особенности распределений: Мужчины обладают более широкой вариативностью параметров, тогда как у женщин распределения более компактны и смещены к меньшим значениям.

Эти данные позволяют сделать вывод о том, что различия в росте и весе между полами выражены и, скорее всего, обусловлены биологическими факторами и спецификой спортивных дисциплин.

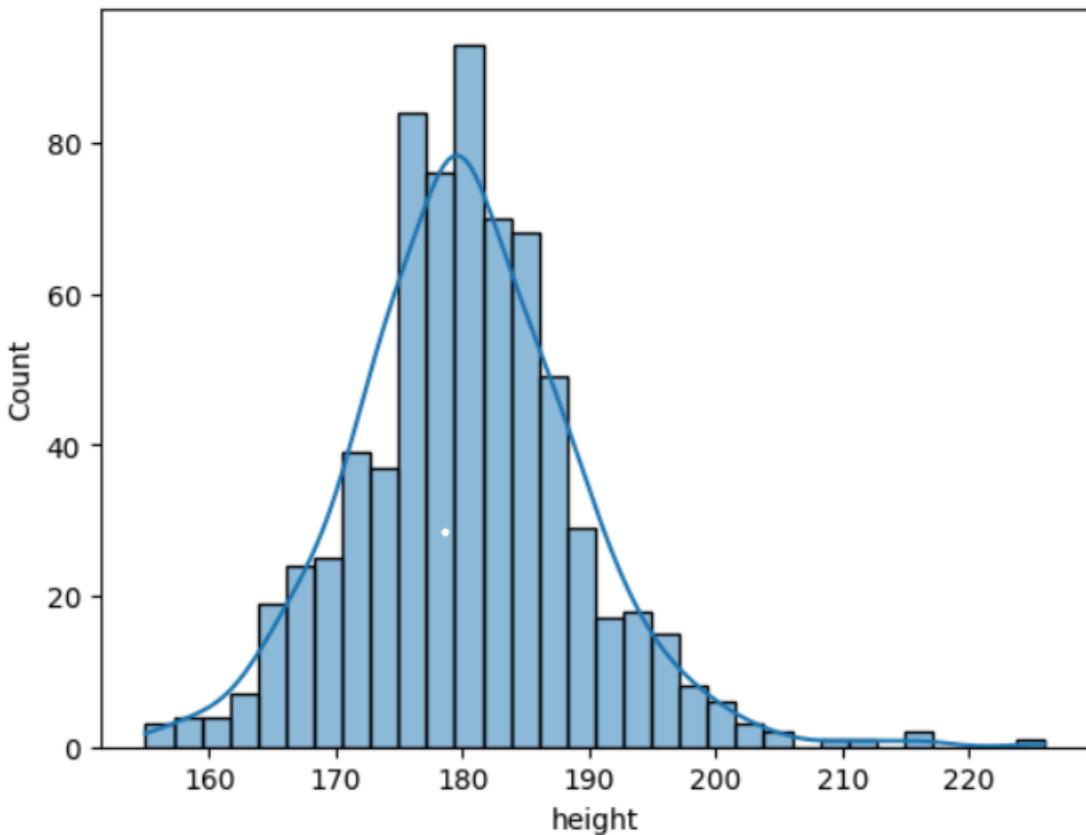
- Выявление выбросов и их проверка (тесты Граббса и Дискона)

Тест Граббса (Grubbs' Test) - статистический тест для выявления выбросов в данных. Тест основан на том, что для нормально распределенной выборки наибольшее отклонение от среднего (или медианы) скорее всего будет выбросом. Он проверяет, является ли наибольшее отклонение от среднего значением, которое значительно отличается от остальных.

Позволяет выявить одиночные выбросы в выборках, что важно в статистическом анализе, чтобы избежать искажения результатов из-за аномальных данных.

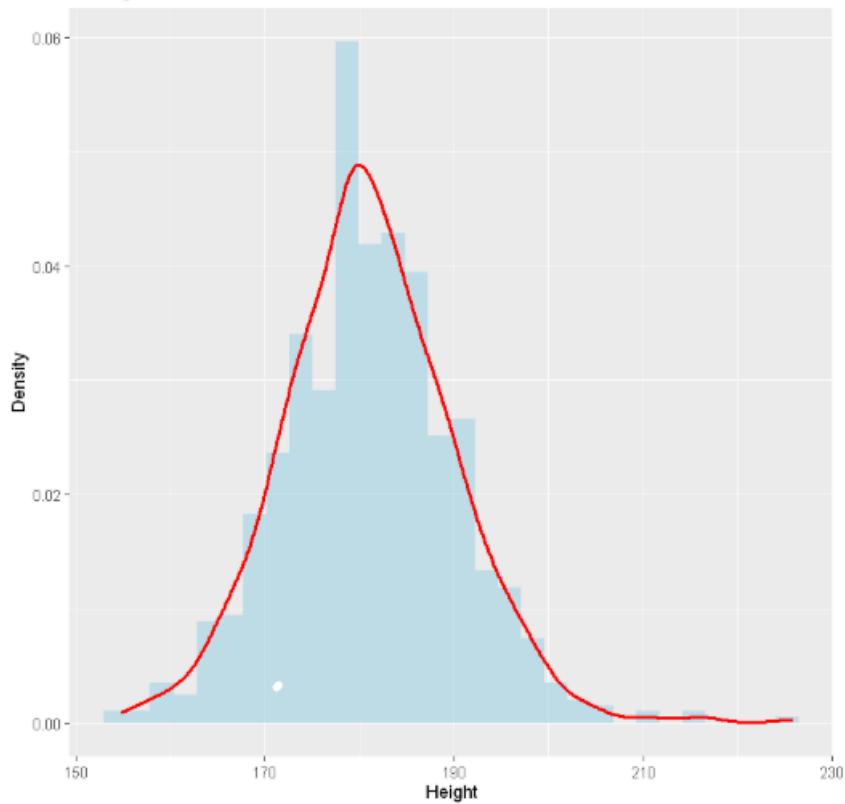
Перед тем, как проводить этот тест нужно проверить нормальность роста спортсменов. Проверим это с помощью теста Шапиро-Уилка, так как он является одним из наиболее эффективных методов для проверки гипотез о нормальности (**см. пункт 2.5**)

Мужчины:

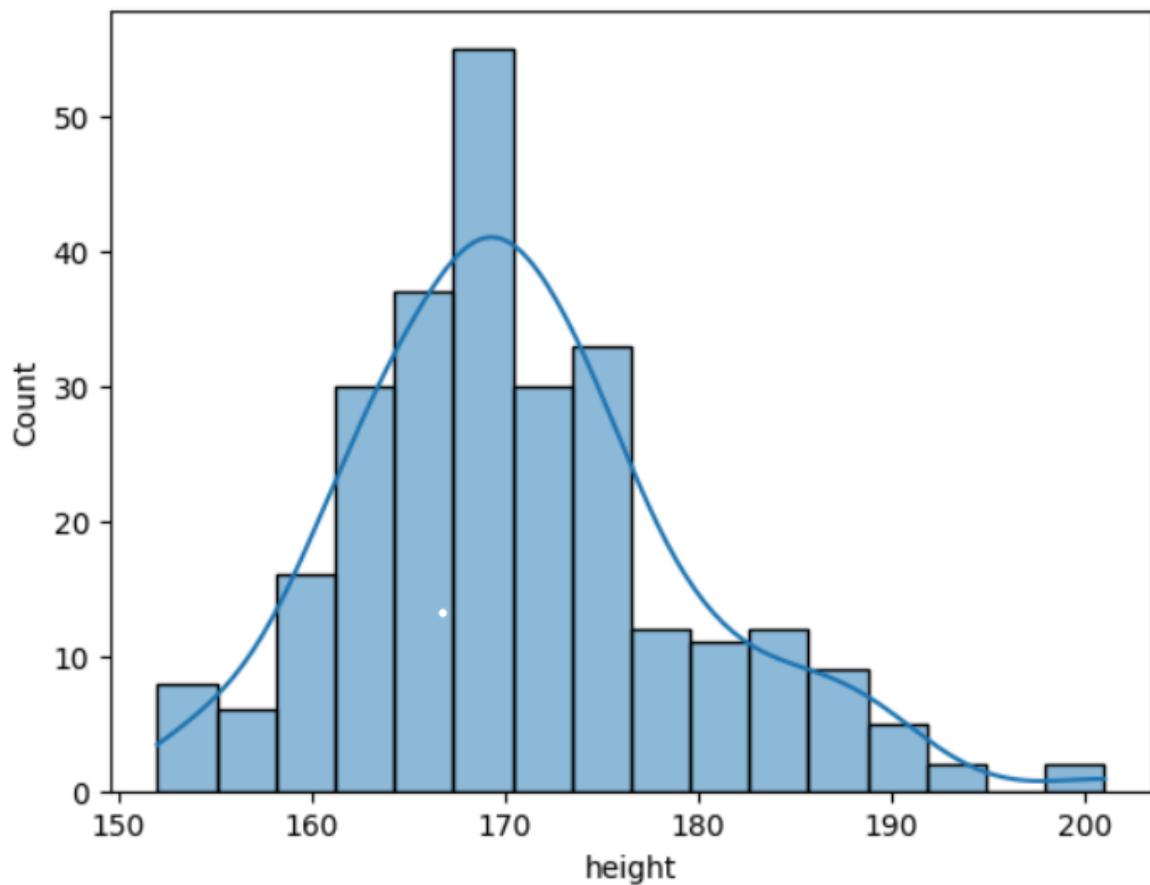


[1] "Non Normal"

Histogram with KDE

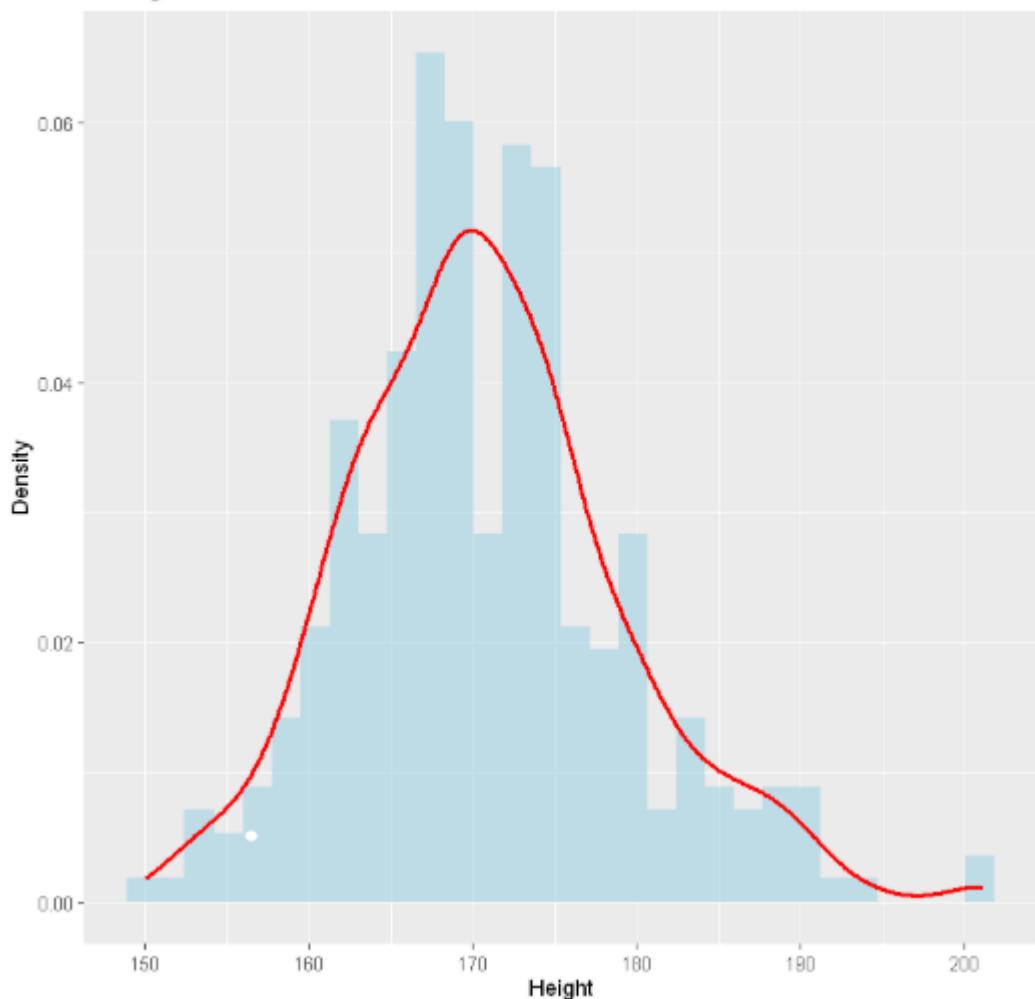


Женщины:



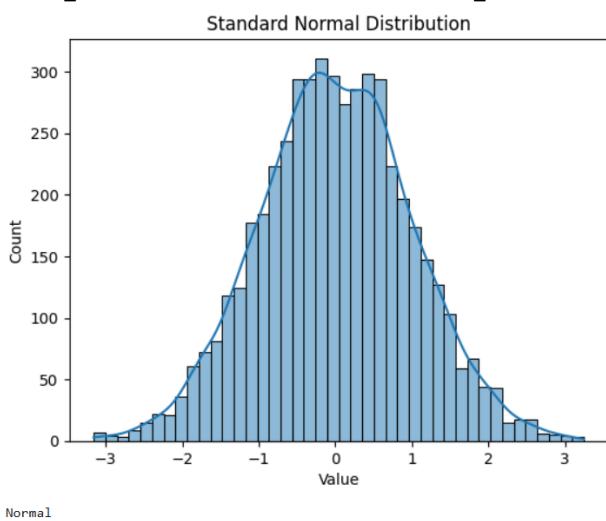
Non Normal

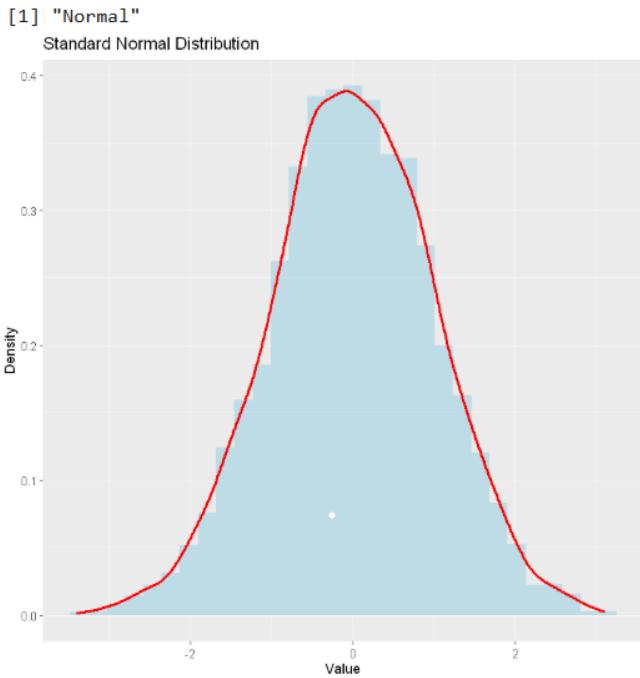
[1] "Non Normal"
Histogram with KDE



Тест показал, что гипотеза о нормальности распределения роста спортсменов мужчин и женщин отклоняется, что подтверждает и графики распределения (ядерные оценки тоже это подмечают, **см. пункт 2.2**)

Соответственно в этом случае генерируем нормальные распределения для тестирования теста Граббса.





Python:

```
: grubbs_test(standard_normal_df)
```

Grubbs Calculated Value: 3.276972078138224
Grubbs Critical Value: 4.4130862497609185
P-Value: 0.9486632602749596
H0: There are no outliers

R:

```
: grubbs_test(standard_normal_df)
```

Grubbs Calculated Value: 3.403557
Grubbs Critical Value: 4.413086
P-Value: 0.9486633
H0: Выбросов нет

Результаты тестов показали, что в выборке не присутствуют выбросы, так как данные имеют стандартное нормальное распределение, что можно проследить и на графике распределения.

Тест Диксона (Dixon's Q Test) - статистический тест, который используется для проверки наличия выброса на малых выборках. Он основывается на анализе самых больших и самых маленьких отклонений от среднего (выборка упорядочивается) и вычислении Q статистики, которая используется для оценки того, является ли значение выбросом в небольшой выборке и таким образом предотвратить возможные искажения в анализе.

Проводим тесты Диксона для стандартного нормального распределения, выбираем выборки размером 10 и 30:

```
print("Test1:\n")
print(f"Q-value: {q_dixon_test(standard_normal_df[:10])}\n")
print("Test2:\n")
print(f"Q-value: {q_dixon_test(standard_normal_df[:30])}\n")

Test1:

Q Dixon's test:

q1_crit_005:
p-value: 0.0869
With 95% confidence: -1.8715438523817087 - is an outlier
q2_crit_005:
p-value: 0.2036
With 95% confidence: -1.8715438523817087 - is an outlier
Q-value: 0.03590564764829517

Test2:

Q Dixon's test:

q1_crit_005:
p-value: 0.0092
With 95% confidence: -1.8715438523817087 - is an outlier
q2_crit_005:
p-value: 0.0756
With 95% confidence: -1.8715438523817087 - is an outlier
Q-value: 0.03590564764829517

Dixon test for outliers

data: head(standard_normal_df, 10)
Q = 0.33258, p-value = 0.3782
alternative hypothesis: lowest value -2.26969977314025 is an outlier
Dixon test for outliers

data: head(standard_normal_df, 30)
Q = 0.33235, p-value = 0.199
alternative hypothesis: highest value 2.81228285312459 is an outlier
```

Также стоит отметить, что как для теста Диксона, так и для теста Граббса существуют таблицы значений уровней значимостей, а следовательно проверку гипотез можно проводить и через значения вероятности **p-value**.

Оба теста важны для диагностики качества данных и предотвращения воздействия выбросов на результаты статистического анализа.

2.4 Заполнение пропусков и сравнение результатов

Нередко бывает так, что при сборе информации, собиратель забывает заполнить какую либо информацию или по каким-нибудь обстоятельствам не получилось собрать данные об объекте исследования из генеральной совокупности. В таком случае возникает вопрос, что делать с этими пропусками, может быть их удалить, а может быть заполнить какими то другими значениями?

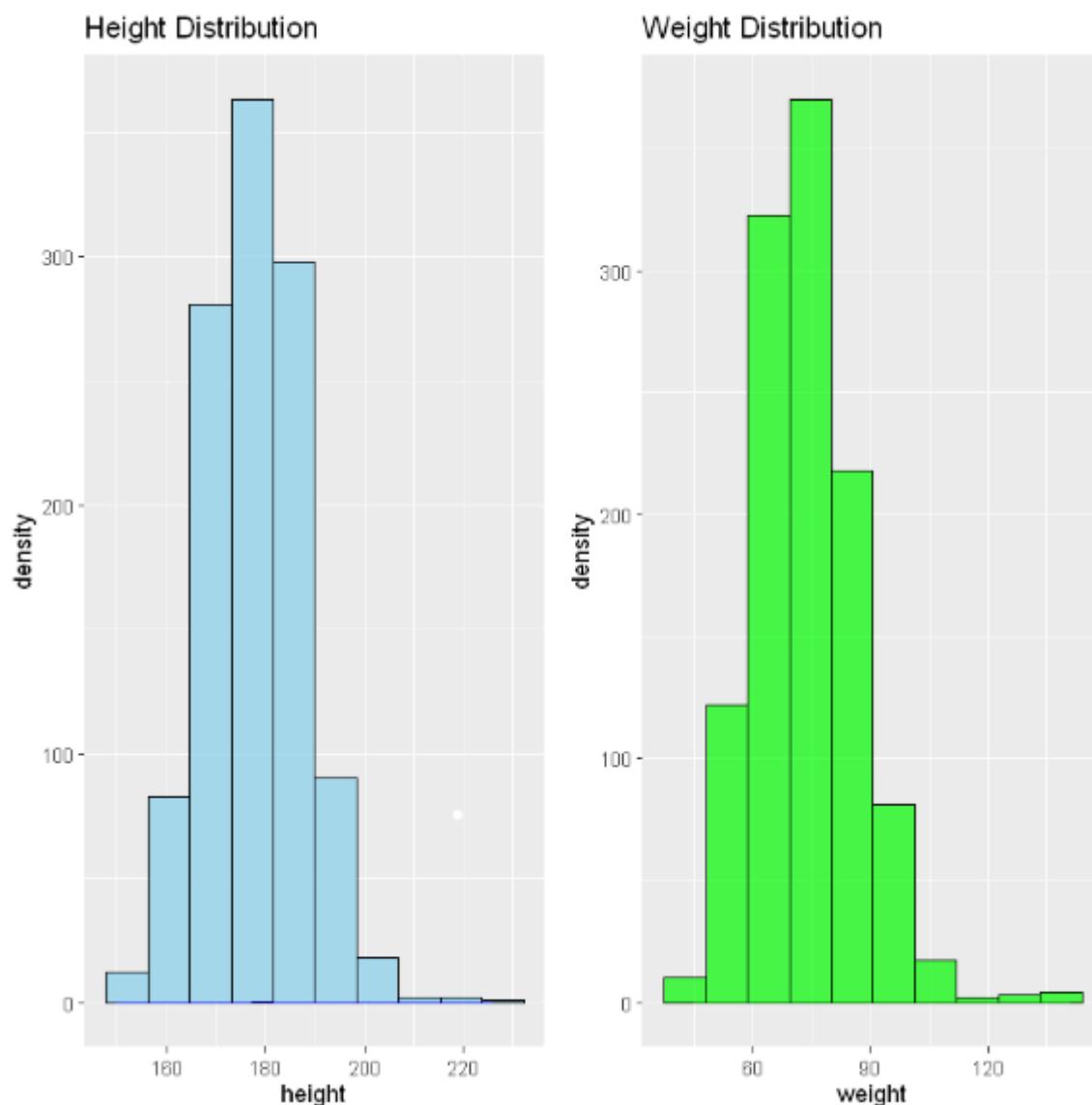
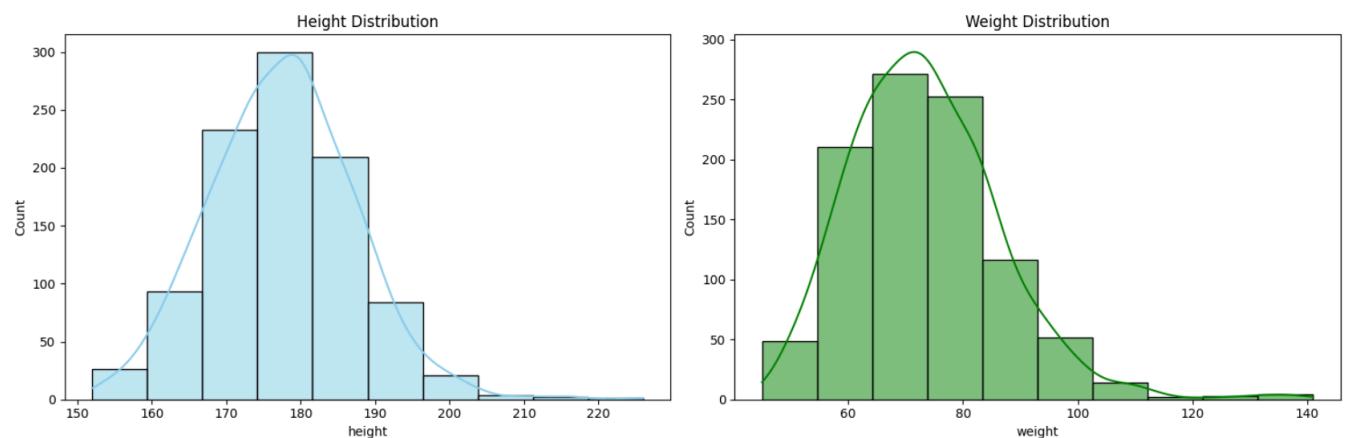
Как раз ответ на этот вопрос дает исследование заполнения значениями(средней, медианой и др.) и тестирование методами для восстановления данных.

- В начале выбирается подмножество данных, как правило случайным образом, из которых искусственно удаляются некоторые значения (добавляются пропуски);
- Эти пропуски заполняются различными методами: регрессией, медианой, интерполяцией, средними и др.
- Восстановленные значения сравниваются с “истинными”(которые были удалены). Таким образом можно оценить точность методов заполнения пропусков.

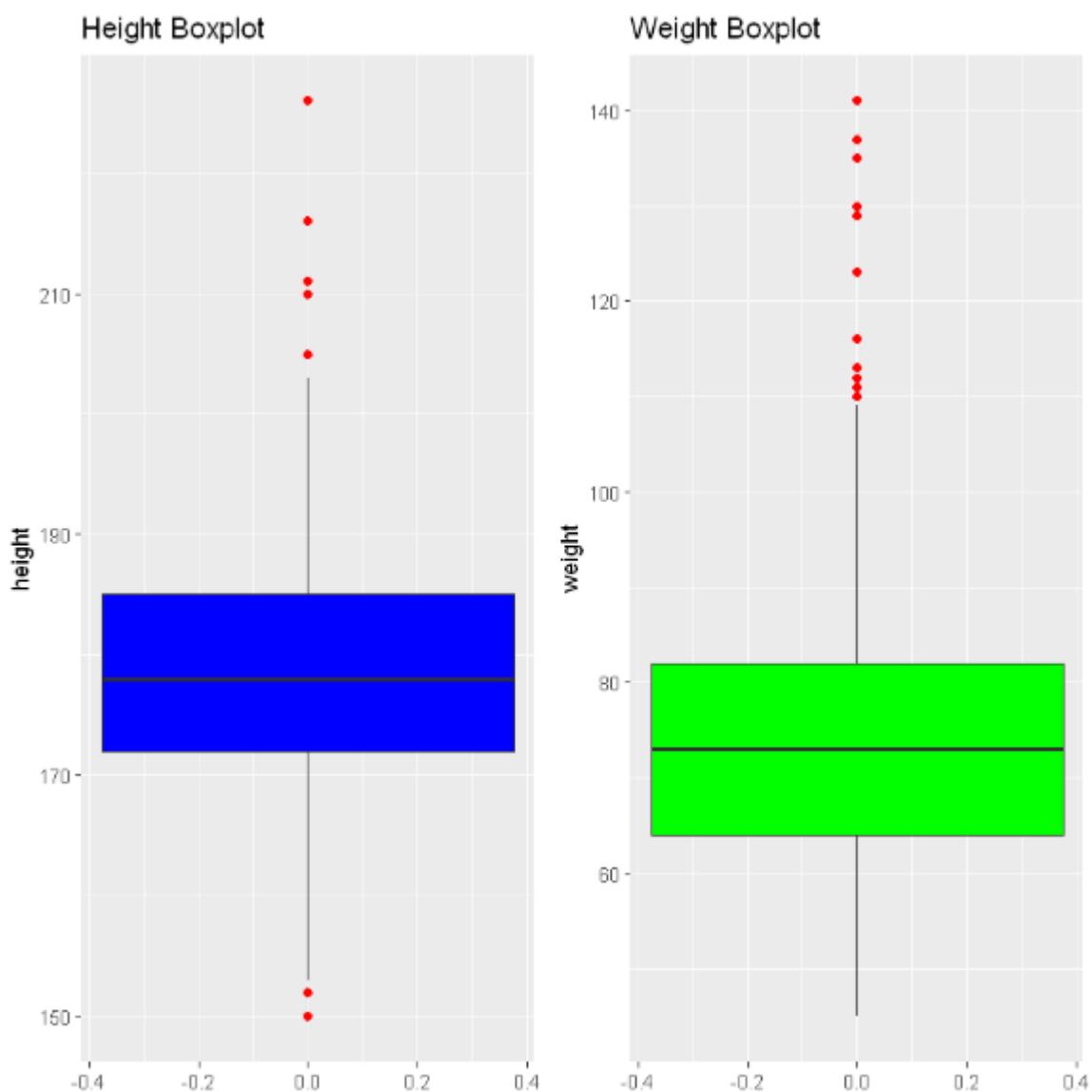
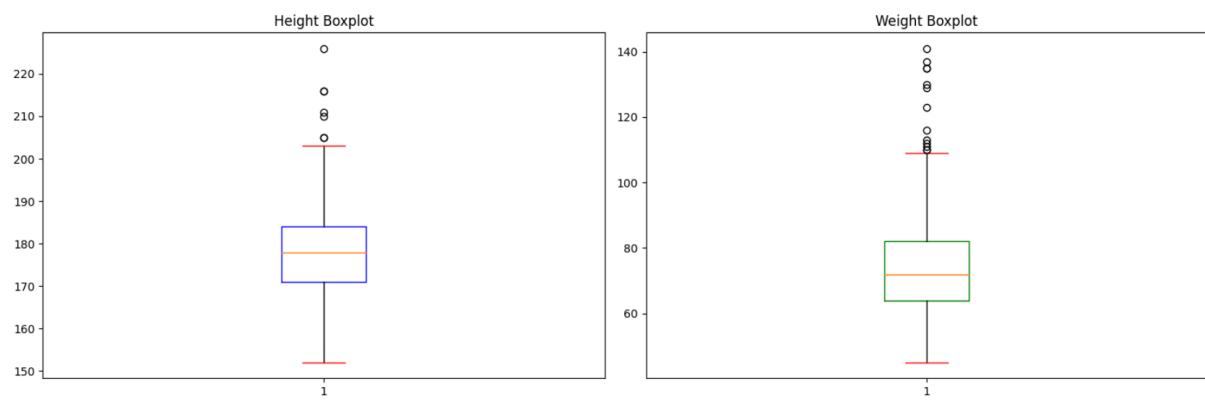
Это позволяет оценить, насколько алгоритм хорошо заполняет данные в пропусках, что может быть очень полезным в различных задачах статистики и машинного обучения.

Для **I датасета** добавим искусственно пропуски для роста и веса спортсменов, затем заполним их средним и медианой и сравним результаты с истинными значениями.

Распределение роста и веса спортсменов (мужчины и женщины)



Выбросы:



Тест 1. Среднее. Вносим случайным образом пропуски в колонку роста и веса и заполняем средним. Вначале все будет проделано для роста, затем для веса.

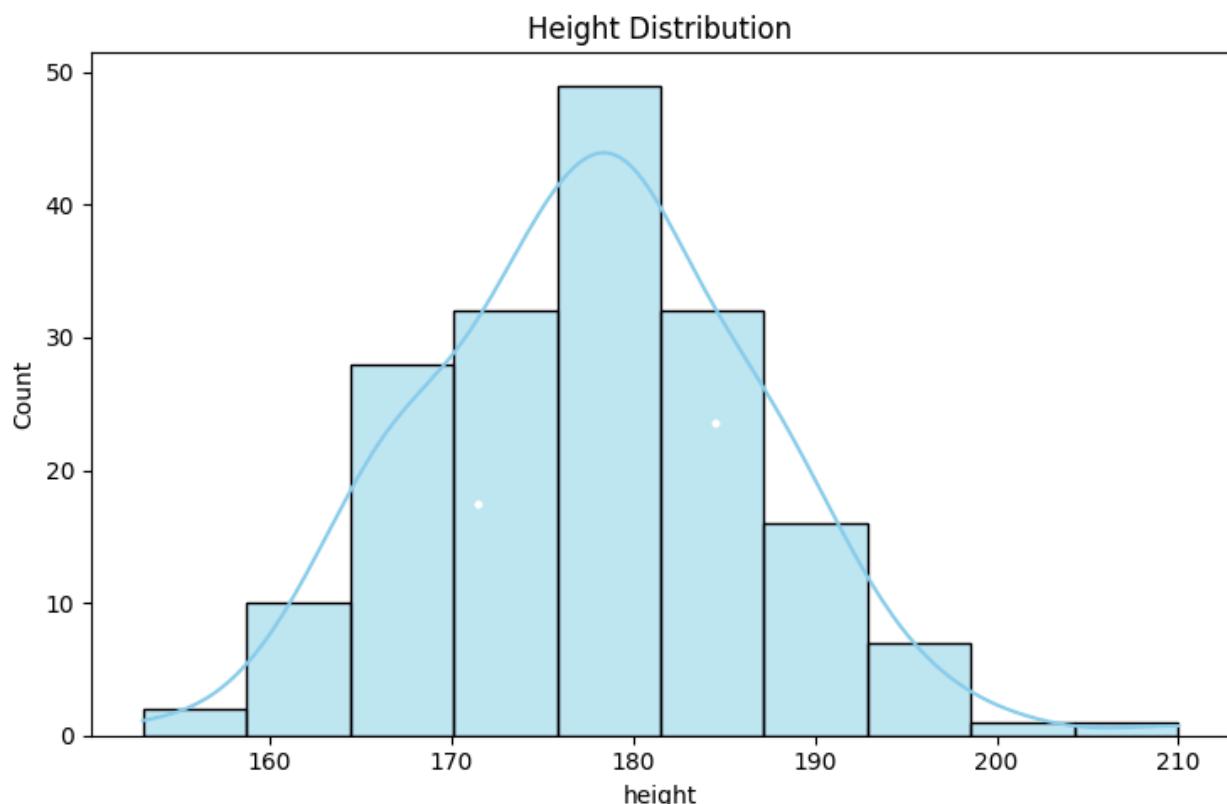
Python:

```
athlete_id      0
name            0
sex             0
born            0
height         770
weight          0
country         0
country_noc     0
description     0
special_notes   0
sport            0
dtype: int64
```

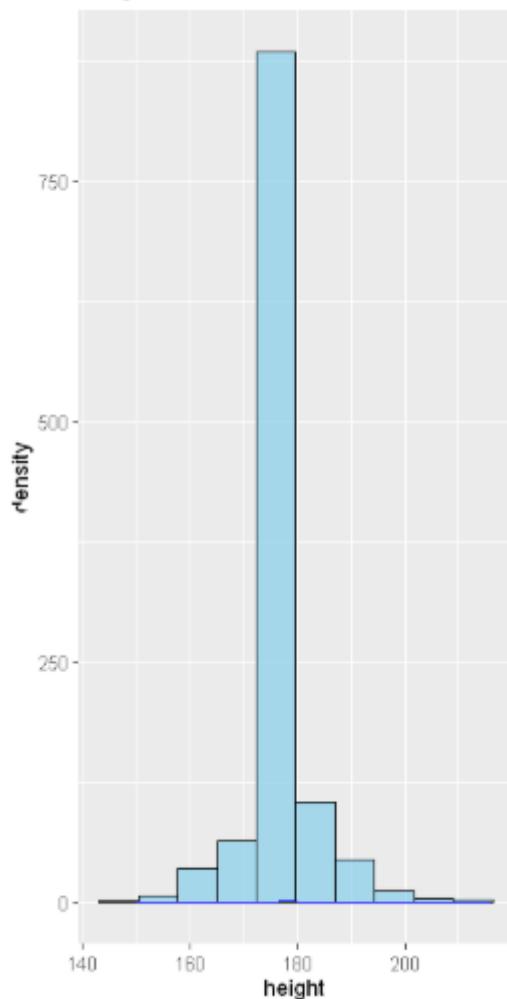
R:

```
athlete_id: 0 name: 0 sex: 0 born: 0 height: 795 weight: 0 country: 0 country_noc: 0 description: 0 special_notes: 0 sport: 0
```

Распределение роста после внесения пропусков и заполнением средним:



Height Distribution



```
-->      athlete_id      height      weight
count  9.730000e+02  178.000000  973.000000
mean   6.224677e+04  177.820225  73.437821
std    5.969695e+04   9.079762  13.297772
min    3.000000e+00  153.000000  45.000000
25%   2.651800e+04  172.000000  64.000000
50%   6.619400e+04  178.000000  72.000000
75%   8.621500e+04  183.750000  82.000000
max   1.005221e+06  210.000000  141.000000)
```

```
athlete_id      name      sex      born
Min. : 3      Length:1150      Length:1150      Length:1150
1st Qu.: 33222 Class :character  Class :character  Class :character
Median : 54888 Mode  :character  Mode  :character  Mode  :character
Mean   : 60729
3rd Qu.: 84688
Max.  : 1005221

height      weight      country      country_noc
Min. :150.0  Min. : 45.00  Length:1150      Length:1150
1st Qu.:178.1 1st Qu.: 64.00  Class :character  Class :character
Median :178.1  Median : 73.00  Mode  :character  Mode  :character
Mean   :178.1  Mean   : 73.66
3rd Qu.:178.1 3rd Qu.: 82.00
Max.  :216.0  Max.  :141.00

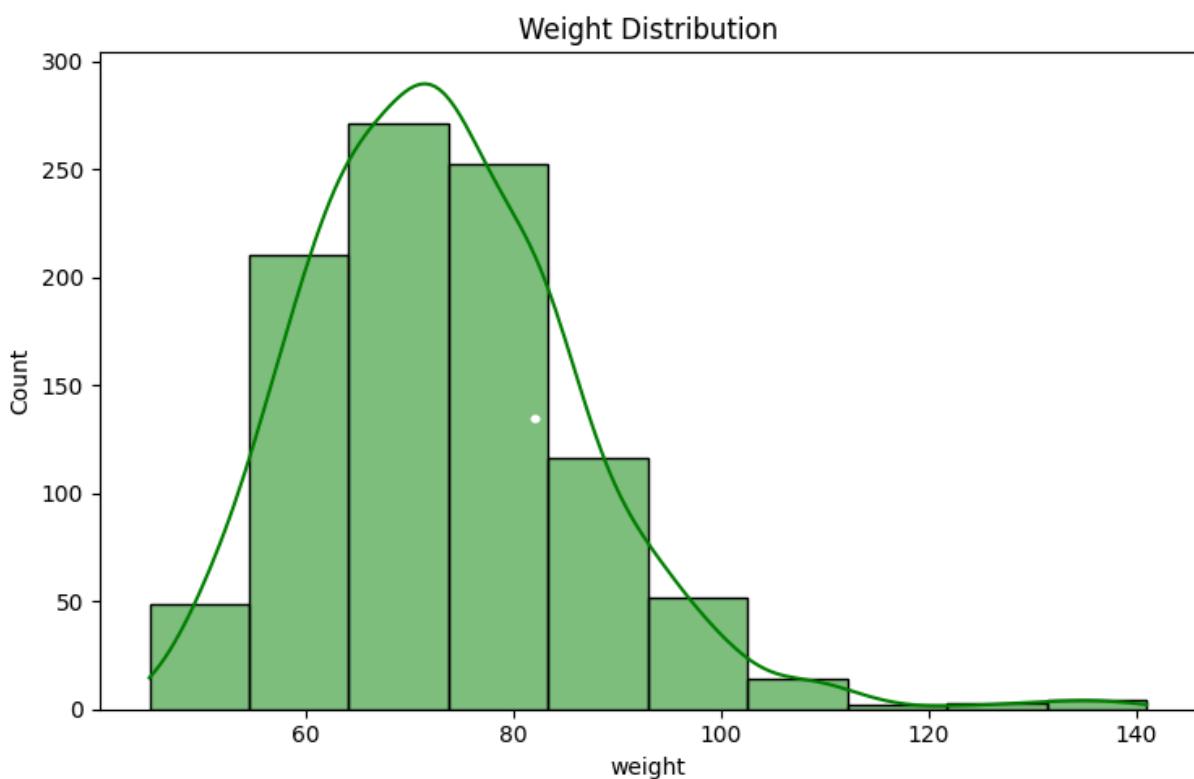
description      special_notes      sport
Length:1150      Length:1150      Length:1150
Class :character  Class :character  Class :character
Mode  :character  Mode  :character  Mode  :character
```

Заполняем теперь пропуски в столбец веса:

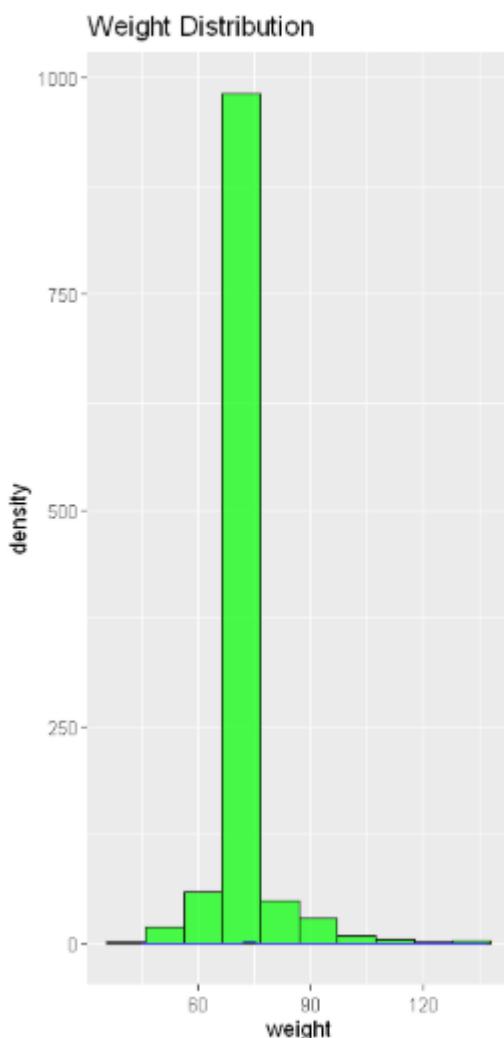
```
:  athlete_id      0
  name          0
  sex          0
  born          0
  height         0
  weight      924
  country        0
  country_noc    0
  description     0
  special_notes   0
  sport          0
dtype: int64
```

```
athlete_id: 0 name: 0 sex: 0 born: 0 height: 0 weight: 924 country: 0 country_noc: 0 description: 0 special_notes: 0 sport: 0
```

Распределение веса спортсменов после заполнения средним



	athlete_id	height	weight
count	9.730000e+02	973.000000	973.000000
mean	6.224677e+04	177.658090	73.376718
std	5.969695e+04	3.875370	2.596895
min	3.000000e+00	153.000000	50.000000
25%	2.651800e+04	177.621788	73.437821
50%	6.619400e+04	177.621788	73.437821
75%	8.621500e+04	177.621788	73.437821
max	1.005221e+06	210.000000	99.000000



```

      height          weight         country        country_noc
Min.   :150.0   Min.   :45.00   Length:1150   Length:1150
1st Qu.:178.1   1st Qu.:73.66   Class  :character  Class  :character
Median  :178.1   Median :73.66   Mode   :character  Mode   :character
Mean    :178.1   Mean   :73.78
3rd Qu.:178.1   3rd Qu.:73.66
Max.    :216.0   Max.   :137.00
description      special_notes       sport
Length:1150      Length:1150      Length:1150
Class  :character  Class  :character  Class  :character
Mode   :character  Mode   :character  Mode   :character

```

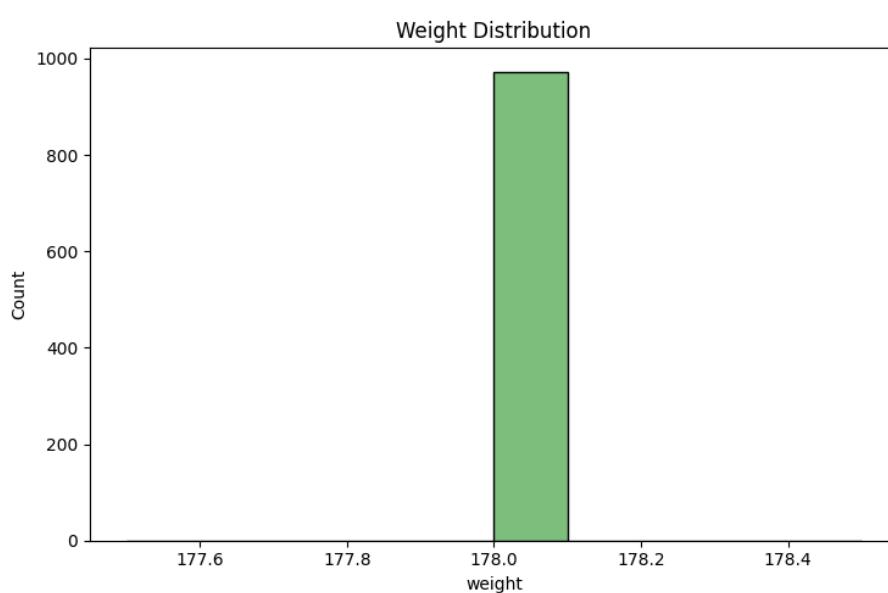
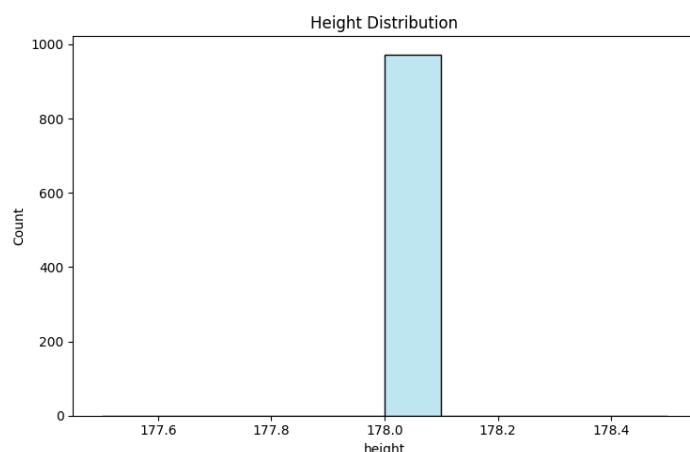
Тест 2. Заполнение медианой.

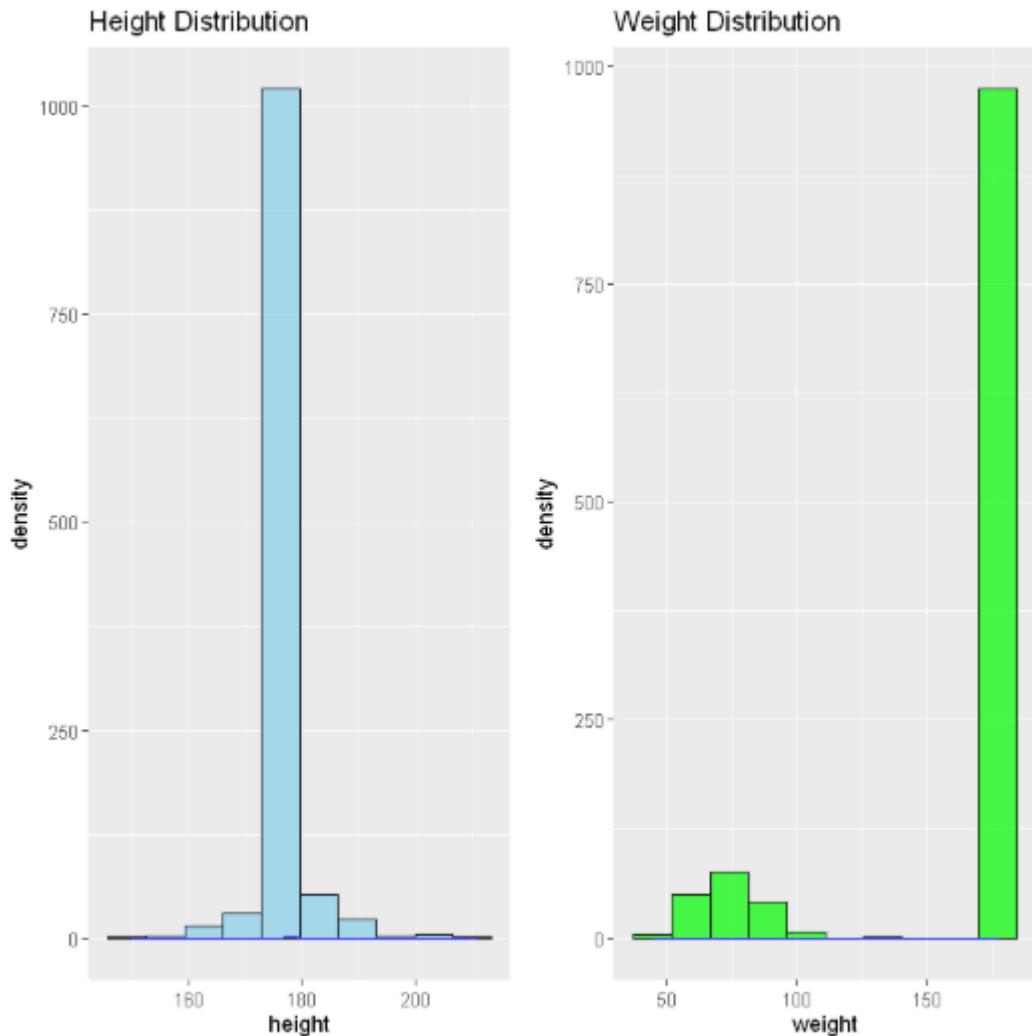
Вносим пропуски:

```
athlete_id      968
name            0
sex             0
born            0
height         973
weight          973
country          0
country_noc      0
description      0
special_notes      0
sport            0
dtype: int64
```

athlete_id: 968 name: 0 sex: 0 born: 0 height: 973 weight: 973 country: 0 country_noc: 0 description: 0 special_notes: 0 sport: 0

Распределения после заполнения медианой:





```

      athlete_id  height  weight
count    973.000000   973.0   973.0
mean     497.389517   178.0   178.0
std      4691.077143      0.0     0.0
min      178.000000   178.0   178.0
25%     178.000000   178.0   178.0
50%     178.000000   178.0   178.0
75%     178.000000   178.0   178.0
max     86035.000000   178.0   178.0)

```

```

      athlete_id       name        sex         born
Min.    : 178 Length:1150    Length:1150    Length:1150
1st Qu.: 178 Class :character Class :character Class :character
Median : 178 Mode  :character Mode  :character Mode  :character
Mean   : 9958
3rd Qu.: 178
Max.   :700998

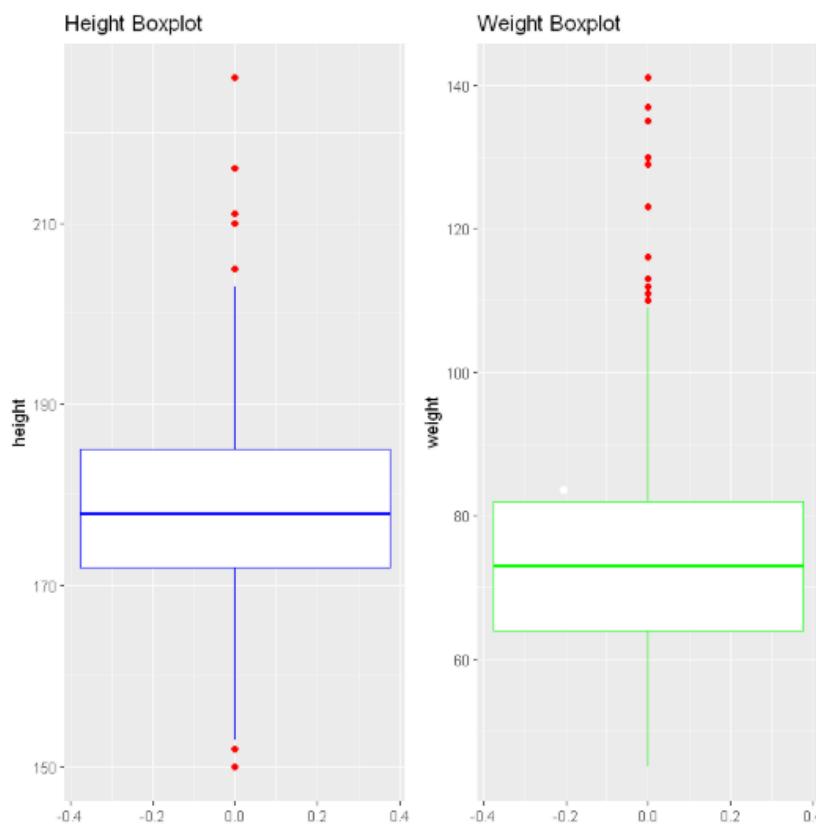
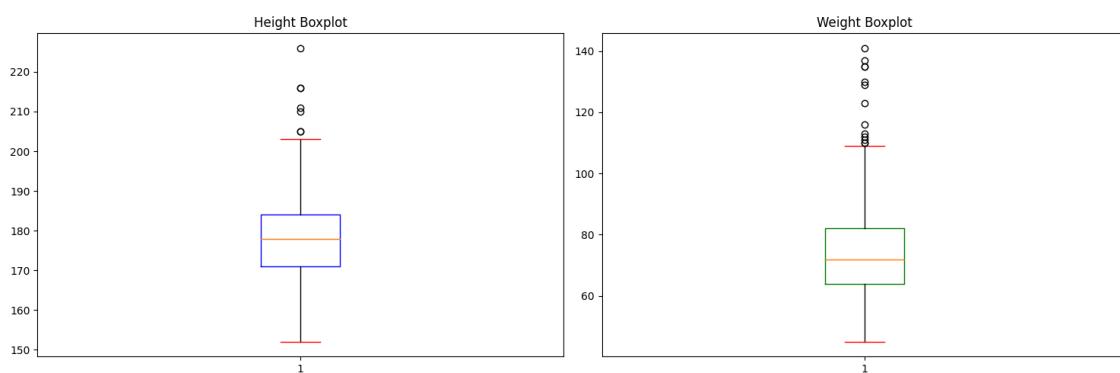
      height       weight       country   country_noc
Min.   :150.0   Min.   :45.0 Length:1150    Length:1150
1st Qu.:178.0   1st Qu.:178.0 Class :character Class :character
Median :178.0   Median :178.0 Mode  :character Mode  :character
Mean   :178.1   Mean   :161.9
3rd Qu.:178.0   3rd Qu.:178.0
Max.   :211.0   Max.   :178.0

      description   special_notes       sport
Length:1150      Length:1150 Length:1150
Class :character  Class :character Class :character
Mode  :character  Mode  :character Mode  :character

```

Исходя из проделанных процедур можно сделать вывод о том, что при внесении пропусков и заполнением различными значениями форма распределений значительно меняется. Из-за этого теряется много важной информации и соответственно качества дальнейшего анализа будет ниже.

Также можно дополнительно отметить, что выбросы практически не меняются:



что говорит о том, что заполнение разными значениями ухудшает качество, соответственно в дальнейшем стоит работать с истинными значениями данного датасета.

2.5 Генерация нормального распределения и анализ.

Анализ нормальности данных является важным этапом статистического анализа, так как многие методы (регрессия, дисперсионный анализ и др.) предполагают нормальное распределение данных. Цель предлагаемой задачи - не только проверить данные на соответствие нормальному распределению, но и научиться интерпретировать результаты различных методов анализа.

Генерация данных с разными параметрами (среднее, стандартное отклонение) и их анализ через графики и тесты позволяет понять:

1. Как распределение данных влияет на результаты проверок.
2. Какие тесты лучше подходят для выборок разных размеров.
3. Как графические методы могут помочь визуальной оценке нормальности.

Рассмотрим подробнее каждый из методов.

График эмпирической функции распределения (ECDF).

ECDF показывает, какая доля наблюдений выборки меньше или равна определенному значению. Она строится для визуального сравнения с теоретической функцией распределения.

Если данные следуют нормальному распределению, то эмпирическая функция будет близка к теоретической функции нормального распределения (кривой распределения). Соответственно чем сильнее отклонения от этой кривой, тем сильнее выводы о ненормальности распределения.

Это помогает визуально оценить нормальность распределения данных.

Генерируем выборки разного объема (100 и 5000) с нормальным распределением и изобразим график.

Генерация нормального распределения:

Python:

```
# Генерируем две выборки: малого (50-100) и умеренного (1000-5000) объемов стандартного нормального распределения

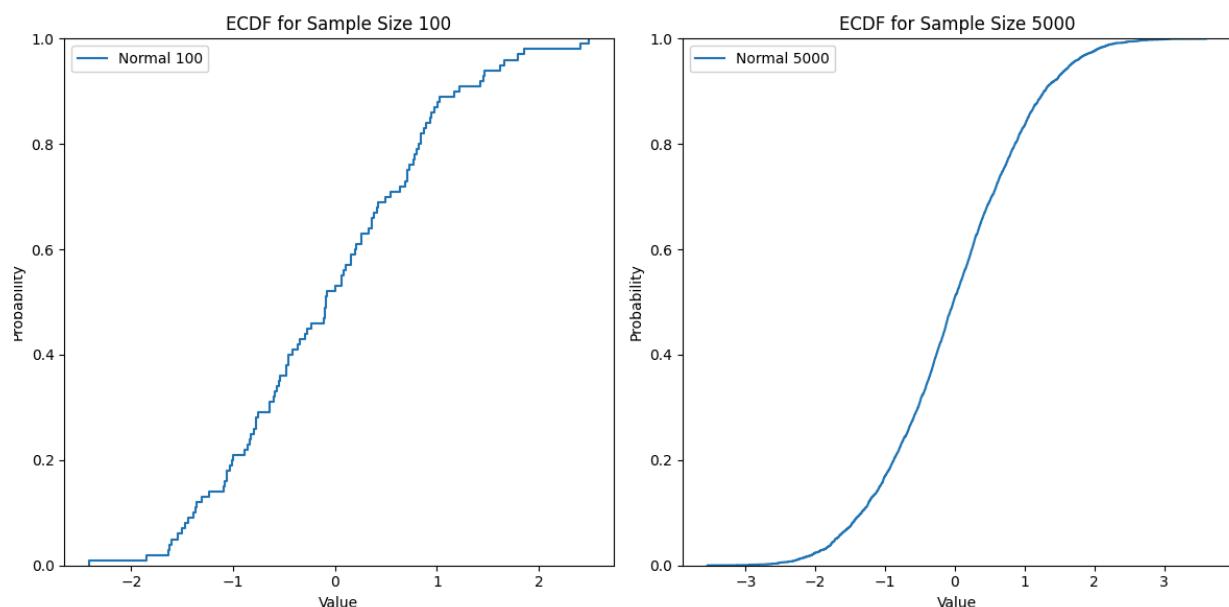
mu, sigma = 0, 1 # параметры нормального распределения: среднее и дисперсия
small_normal_df_1 = np.random.normal(mu, sigma, 100)
mode_normal_df_2 = np.random.normal(mu, sigma, 5000)
```

R:

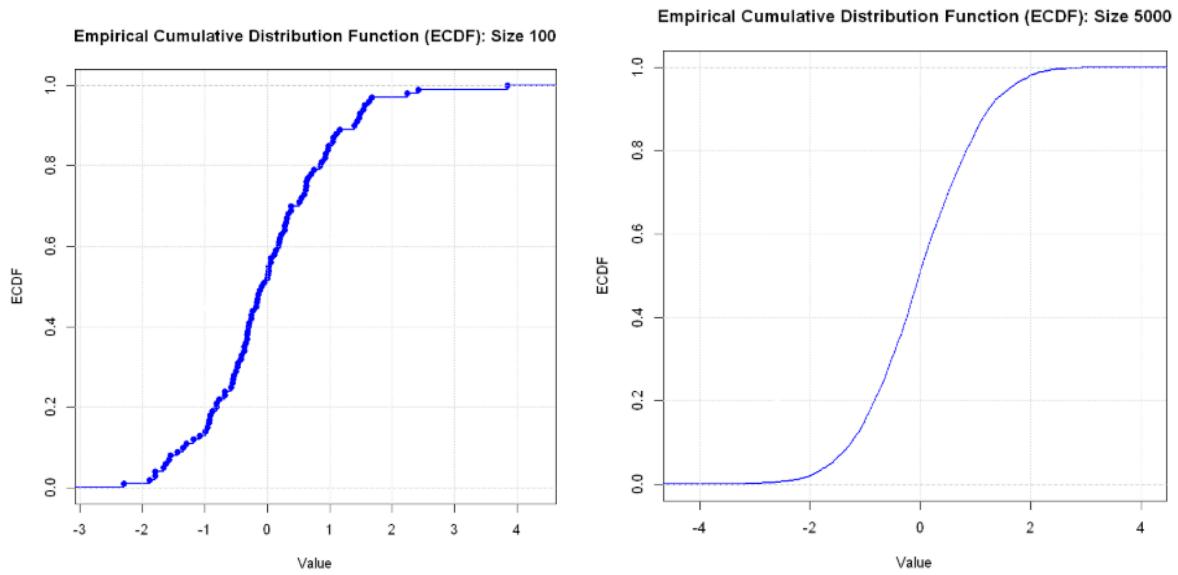
```
# Генерируем две выборки: малого (50-100) и умеренного (1000-5000) объемов стандартного нормального распределения

mu <- 0
sigma <- 1 # параметры нормального распределения: среднее и дисперсия
small_normal_df_1 <- rnorm(100, mu, sigma)
mode_normal_df_2 <- rnorm(5000, mu, sigma)
```

Python:



R:

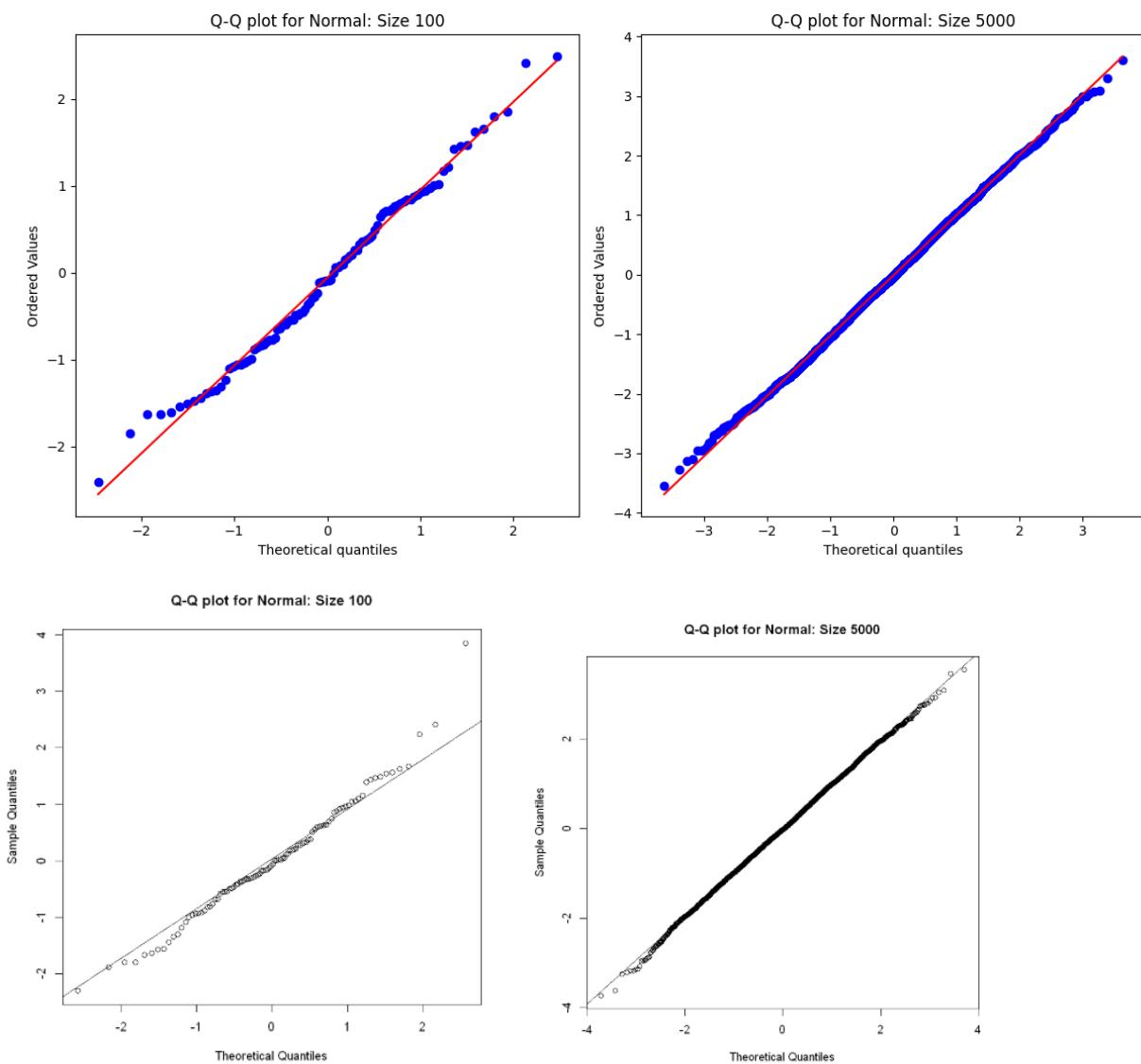


Видно, что чем больше размер выборки, тем плавнее эмпирическая функция распределения, а значит тем более лучше данные подходят к нормальному распределению.

Графики квантилей (QQ-plot) сравнивает квантильные значения(значения, которая не превышает случайная величина с фиксированной вероятностью) данных с квантилями теоретического распределения. Если точки лежат близко к диагонали, то распределение данных соответствует теоретическому.

Это позволяет подчеркнуть отклонения в хвостах распределения (асимметрия или наличие выбросов) и центр распределения.

Строим QQ-plot с теми же данными, что сгенерировали для ECDF:

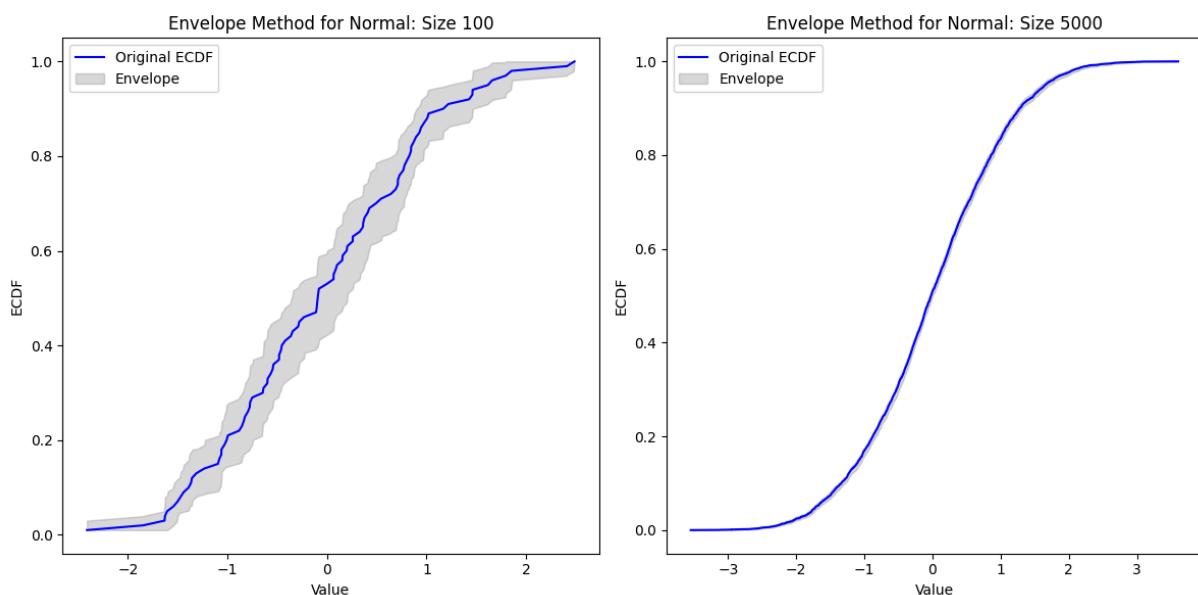


Метод огибающих (Envelope method) - этот метод основан на построении доверительных интервалов вокруг теоретической линии (например, на QQ-plot). Если эмпирические данные лежат внутри этих огибающих, то распределение можно считать нормальным.

Это добавляет более строгий критерий к визуальному анализу и требует расчета доверительных интервалов, которые зависят от объема выборки.

Он позволяет повысить объективность в визуальном анализе и оценить небольшие отклонения от нормальности.

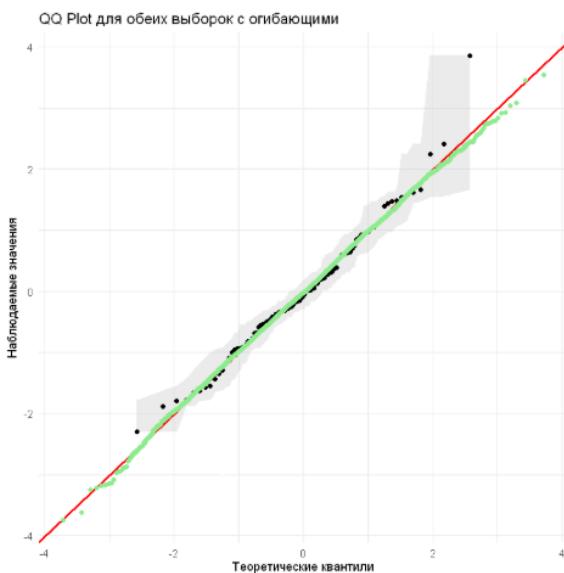
Построим график с помощью метода огибающего для начальных выборок:



На графиках серыми областями помечены доверительные интервалы, видно, что чем больше выборка, тем все более ближе она подходит к нормальному распределению.

При построении с помощью языка R выборка с 100 и 5000 элементами была отображена на одном графике ниже:

- **Черные точки** - выборка размера 100;
- **Зеленые точки** - выборка из 5000 элементов;
- **Серая область** - доверительный интервал;
- **Красная линия** - теоретическая линия квантилей нормального распределения.



Это были визуальные методы исследования нормальности распределения, теперь стоит рассмотреть методы исследования с помощью расчета статистик и значений на уровнях значимости, которые позволяют на основе оценки и проверки значений на определенном уровне значимости сделать вывод о нормальности выборки, они являются подтверждающими критериями.

В общем случае при проверках гипотез используется значение **вероятности** p-value на определенном уровне значимости, которое можно найти в таблице значений тестов, однако во многих тестах есть проверка гипотез, связанных со статистиками тестов.

Ниже приведены результаты тестов для выборок **small_normal_df_1** и **mode_normal_df_2**, которые были сгенерированы в начале.

Критерий Колмогорова-Смирнова проверят насколько эмпирическая функция распределения отличается от теоретической (нормальной). Он чувствителен к отклонениям в центре распределения, менее чувствителен к хвостам. Является общей проверкой соответствия данных нормальному распределению.

```
(KstestResult(statistic=0.07520052032445412, pvalue=0.5969916749808757, statistic_location=-0.4543193937540292, statistic_sign=1),  
KstestResult(statistic=0.012773007156689747, pvalue=0.3851310800216007, statistic_location=-1.205386875614058, statistic_sign=1))
```

Тест 1:

Принимаем нулевую гипотезу: данные распределены нормально

Тест 2:

Принимаем нулевую гипотезу: данные распределены нормально

Asymptotic one-sample Kolmogorov-Smirnov test

```
data: small_normal_df_1  
D = 0.049635, p-value = 0.9662  
alternative hypothesis: two-sided
```

Asymptotic one-sample Kolmogorov-Smirnov test

```
data: mode_normal_df_2  
D = 0.015758, p-value = 0.1668  
alternative hypothesis: two-sided
```

Тест 1:

Н0: данные нормально распределены

Тест2:

Н0: данные нормально распределены

Критерий Шапиро-Уилка оценивает, насколько значение статистики близок к ожидаемому значению при нормальному распределении. Является одним из наиболее эффективных методов проверки нормальности. Основан на линейной несмешенной оценке дисперсии к ее обычной оценке методом максимального правдоподобия. Чаще всего используется для выборок до 2000 элементов.

Критерий Андерсона-Дарлинга - это модификация критерия Колмогорова-Смирнова, учитывающая хвосты распределений. Является непараметрическим критерием, подходит для более строгих проверок, когда хвосты распределений важны.

Критерий Крамера-фон-Мизеса оценивает суммарные отклонения эмпирической функции распределения от теоретической. Устойчив к отклонениям в разных частях распределения. Он полезен для выборок разного объема, особенно если важны общие отклонения.

Критерий Колмогорова-Смирнова в модификации Лиллифорса - эта модификация применима при неизвестных параметрах нормального распределения. Он менее строгий чем тест К-С.

Сначала оцениваются выборочное среднее и дисперсия, затем как и в критерии Колмогорова-Смирнова находится максимальное отклонение выборочной функции распределения от теоретической, после чего принимается решение о статистической значимости наблюдаемого отклонения выборочной функции распределения от теоретической.

Основное допущение в этой модификации: параметры теоретического распределения оцениваются по тем же данным, что и при проверке на соответствие распределений.

Критерий Шапиро-Франсия является адаптацией критерия Шапиро-Уилка для проверки гипотезы на больших выборках. Он более чувствителен к отклонениям в хвостах и для проверки требуется выборка из более чем 5000 элементов.

Для малых выборок до **100** элементов некоторые тесты (например, Шапиро-Уилка) особенно эффективны, так как они чувствительны к мелким отклонениям.

Для больших выборок (**1000+**) большинство тестов могут давать значимые результаты, даже при минимальных отклонениях, которые могут быть несущественными для анализа.

Приведем результаты тестов:

Python:

Результаты тестов для Выборка из 100 элементов:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9891, p-value=5.9269e-01
res: True

Андерсон-Дарлинг: Статистика=0.3174, p-value=0.5382479217380227
res: True

Крамер фон Мизес: Статистика=0.0977, p-value=5.9737e-01
res: True

Колмогоров-Смирнов (Лиллиефорс): Статистика=0.0752, p-value=5.9699e-01
res: True

Шапиро-Франсия: Статистика=0.9891, p-value=5.9269e-01
res: True

Результаты тестов для Выборка из 5000 элементов:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9994, p-value=1.3162e-01
res: True

Андерсон-Дарлинг: Статистика=0.5516, p-value=0.15528729415522974
res: True

Крамер фон Мизес: Статистика=0.1674, p-value=3.4059e-01
res: True

Колмогоров-Смирнов (Лиллиефорс): Статистика=0.0128, p-value=3.8513e-01
res: True

Шапиро-Франсия: Статистика=0.9994, p-value=1.3162e-01
res: True

R:

Результаты тестов для Sample of 100 elements:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9798, p-value=1.2713e-01
res: TRUE

Андерсон-Дарлинг: Статистика=0.3503, p-value=4.6548e-01
res: TRUE

Крамер фон Мизес: Статистика=0.0575, p-value=4.0475e-01
res: TRUE

Колмогоров-Смирнов (Лиллиефорс): Статистика=0.0496, p-value=9.6623e-01
res: TRUE

Шапиро-Франция: Статистика=0.9798, p-value=1.2713e-01
res: TRUE

Результаты тестов для Sample of 5000 elements:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9997, p-value=6.4122e-01
res: TRUE

Андерсон-Дарлинг: Статистика=0.3671, p-value=4.3194e-01
res: TRUE

Крамер фон Мизес: Статистика=0.0669, p-value=3.0727e-01
res: TRUE

Колмогоров-Смирнов (Лиллиефорс): Статистика=0.0158, p-value=1.6685e-01
res: TRUE

Шапиро-Франция: Статистика=0.9997, p-value=6.4122e-01
res: TRUE

В файлах с кодом приведены примеры тестов на нормальном распределении с шумом, но для упрощения, здесь оставим только тесты на стандартном нормальном распределении.

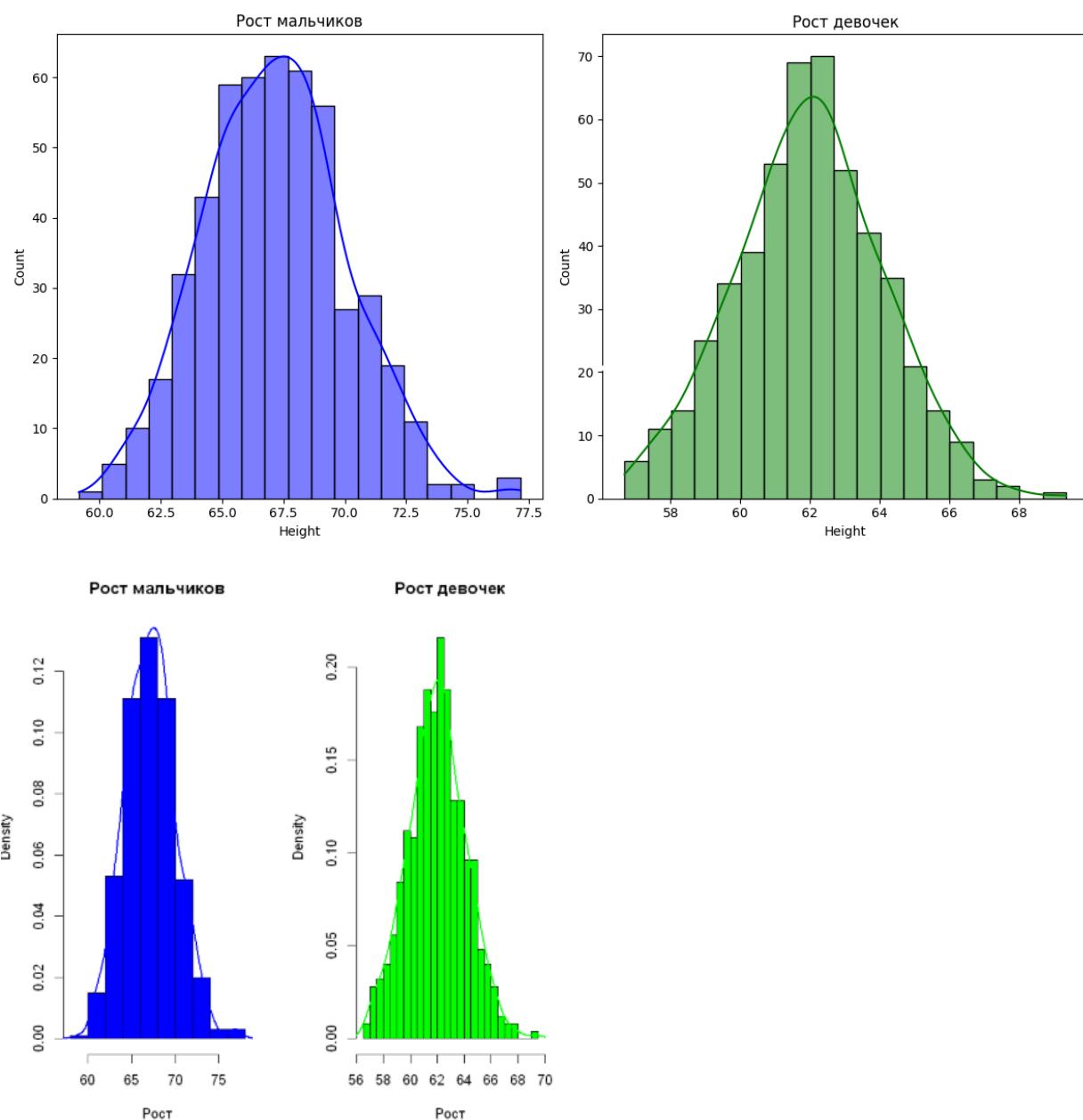
Дальше предлагается разобрать данные методы на **IV датасете (см. пункт 2.1).**

2.6. Демонстрация примера анализа данных с помощью методов и графиков из пункта 2.5

На IV датасете будут разобраны методы из пункта 2.5.

Вкратце, данный датасет содержит данные о росте студентов из высшей школы, которые были сгенерированы из нормальных распределений с различными параметрами (среднее, отклонение).

Перед тем как проверять гипотезы, надо удостовериться, что данные **распределены нормально и независимы**.



Проверяем нормальность и независимость с помощью тестов Шапиро-Уилка и коэффициента корреляции Пирсона.

Python:

Шапиро-Уилк: Статистика=0.9953, p-value=1.3945e-01

Шапиро-Уилк: Статистика=0.9976, p-value=6.8459e-01

H0: Данные - нормальные

Корреляция Пирсона: -0.013530877850919405, p-value: 0.7627922008265289

Данные независимы

R:

Шапиро-Уилк: Статистика=0.9953, p-value=1.3945e-01

Шапиро-Уилк: Статистика=0.9976, p-value=6.8459e-01

H0: Данные - нормальные

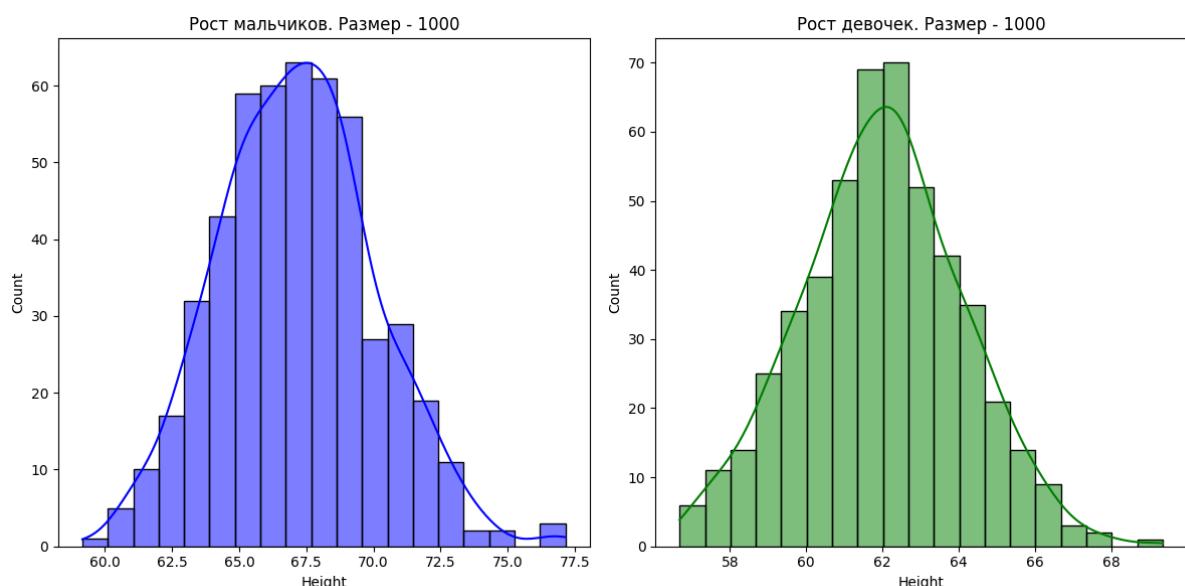
Корреляция Пирсона: -0.01353088 , p-value: 0.7627922

Данные независимы

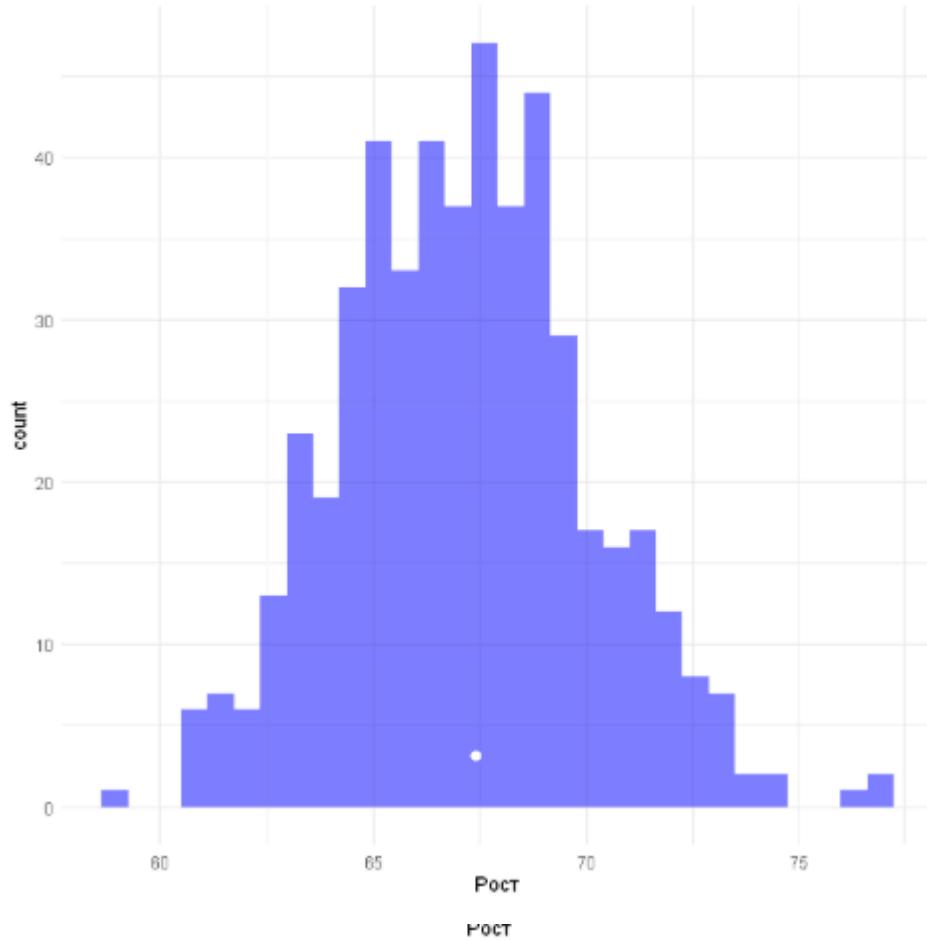
Можно проверять гипотезы.

Специально добавляем в выборку еще 500 элементов, чтобы можно было провести тесты на большом размере выборки.

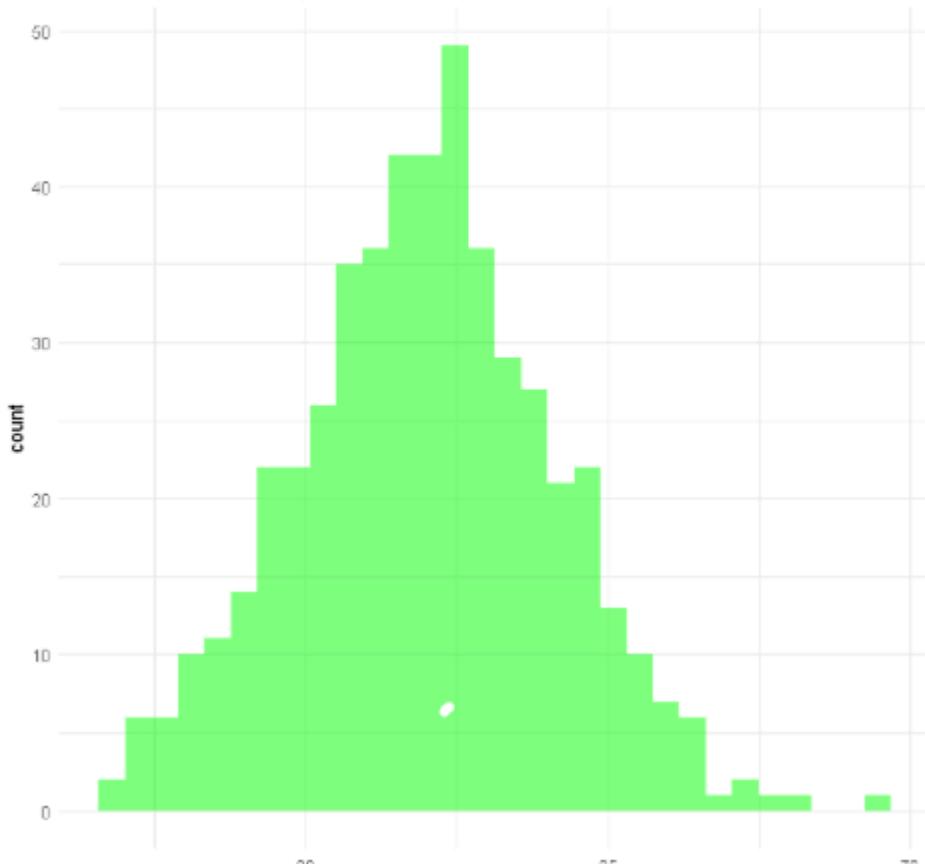
Распределение с объемом 1000 элементов:



Рост мальчиков. Размер - 1000

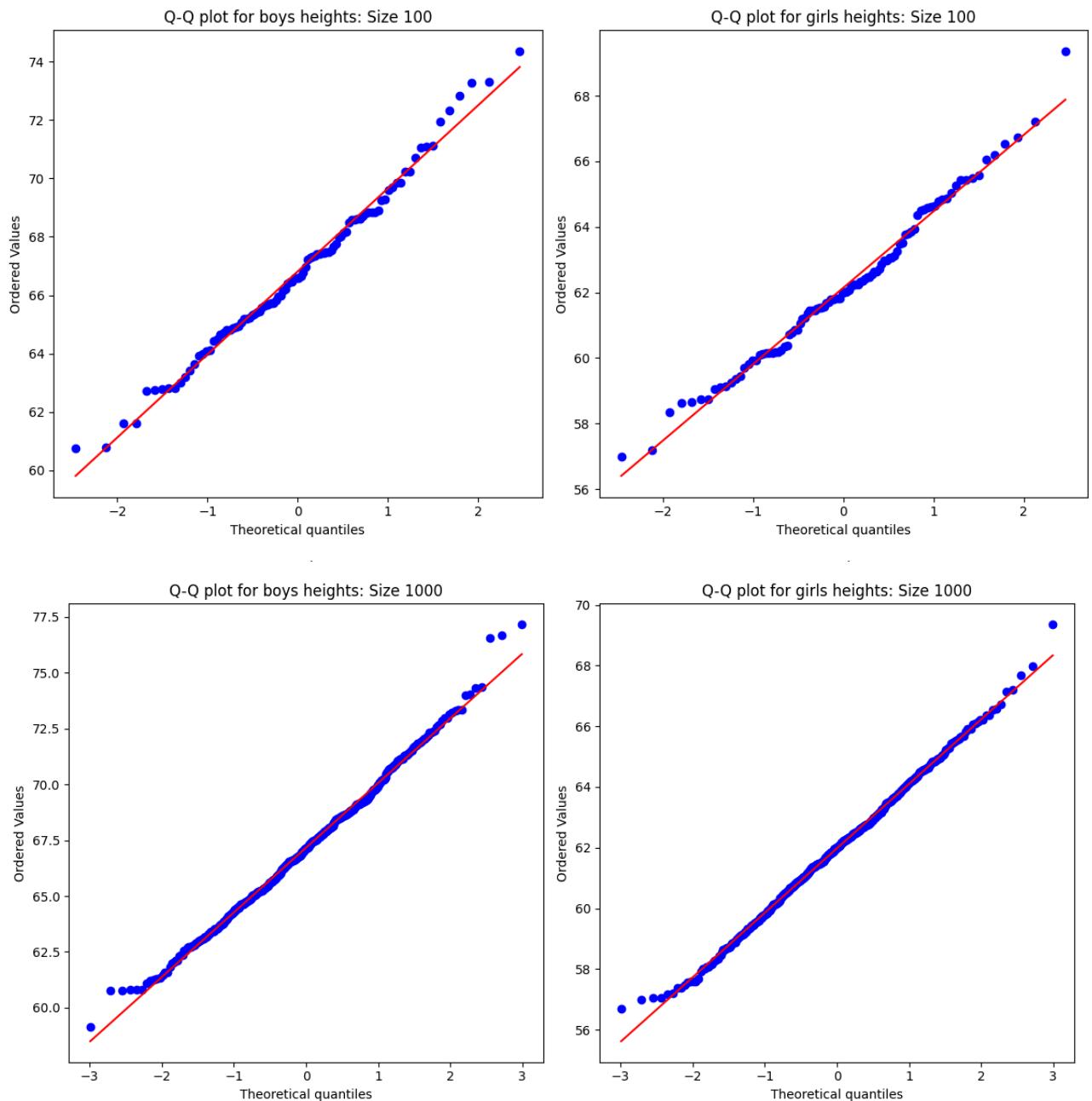


Рост девочек. Размер - 1000

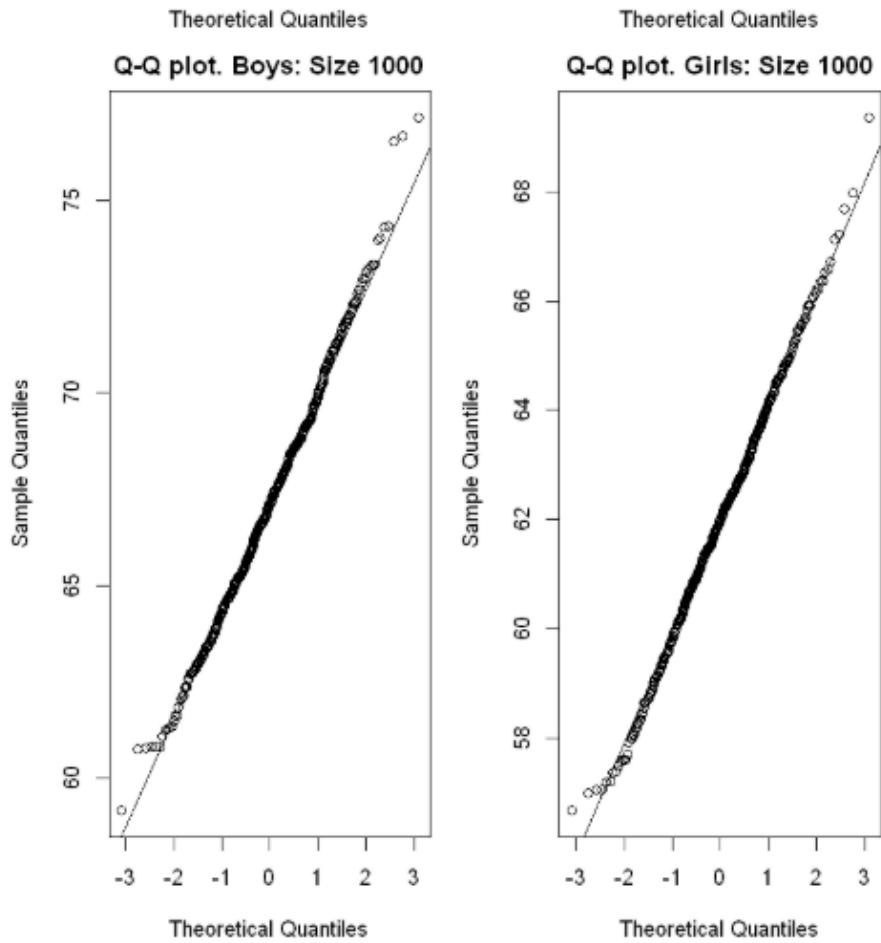
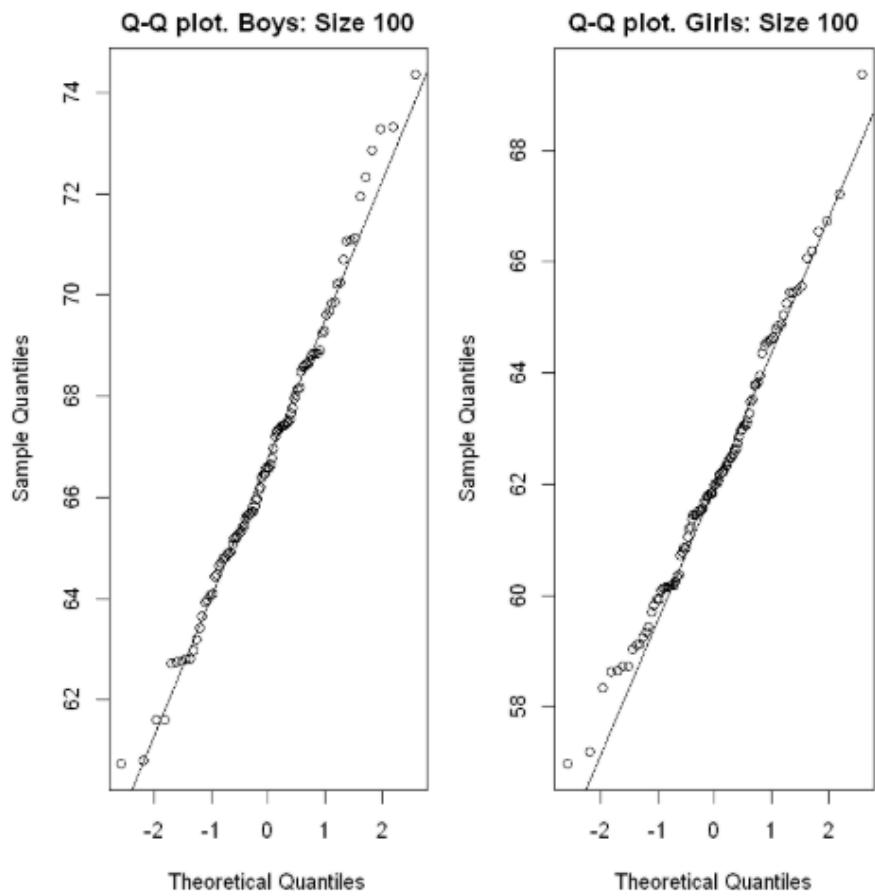


QQ-plot:

Python:

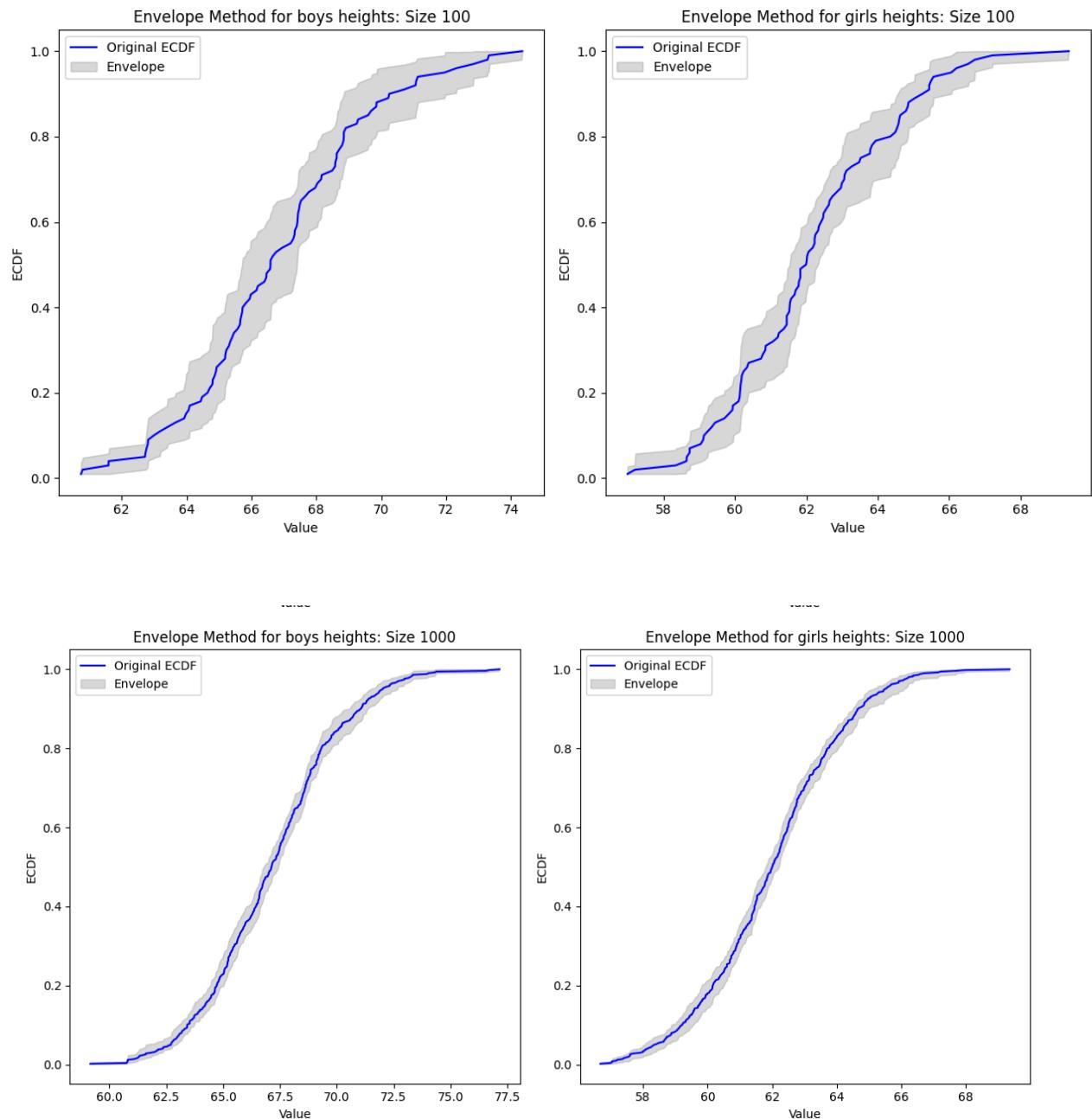


R:

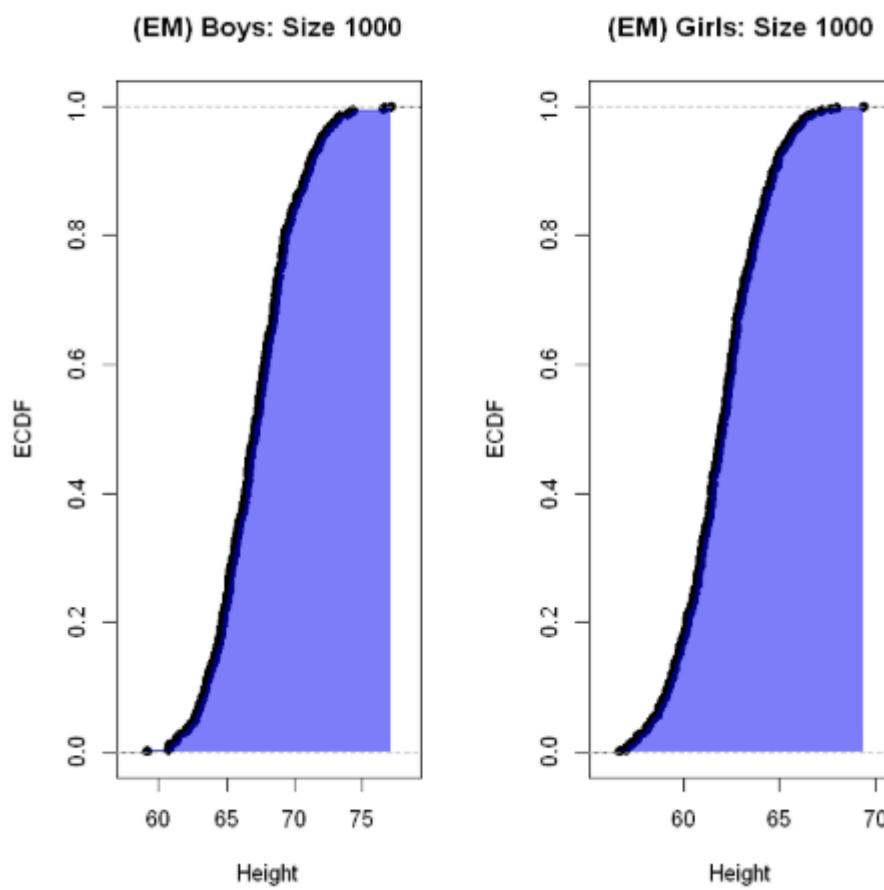
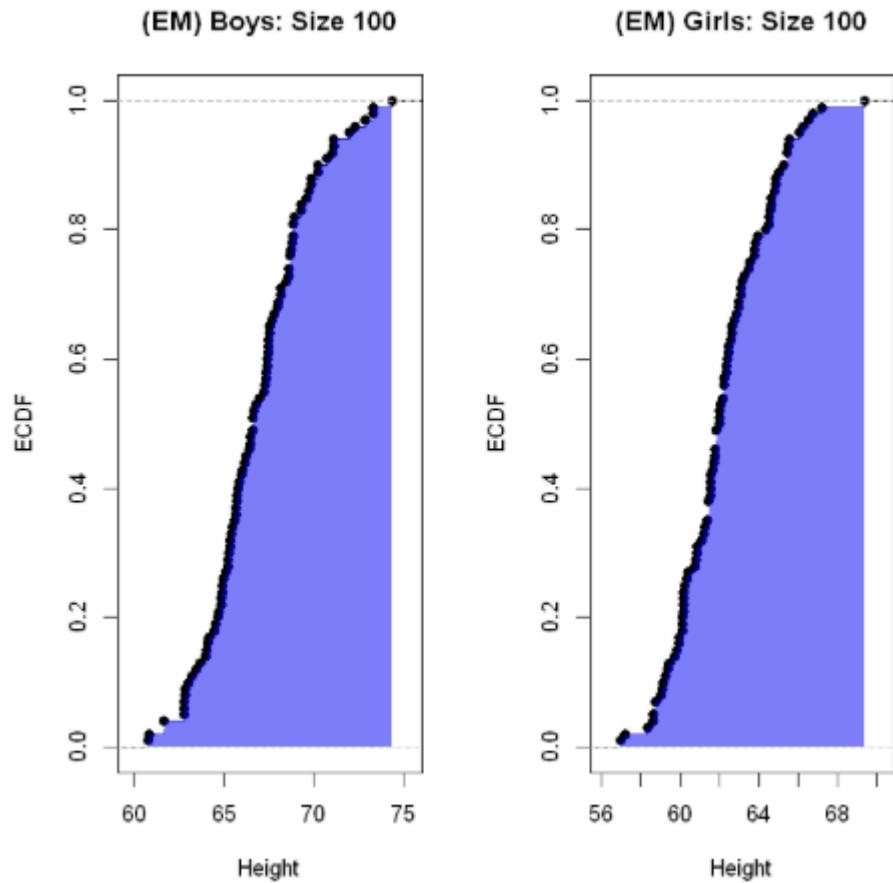


Метод огибающих (Envelope Method):

Python:

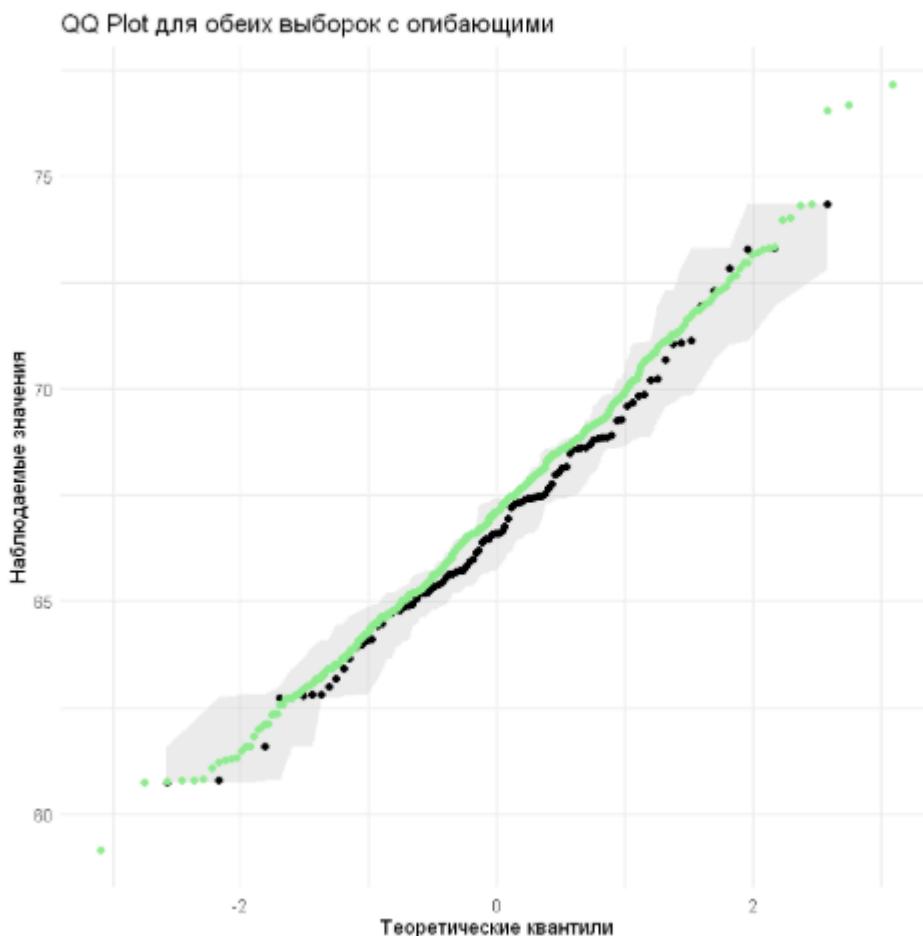


R: (ручная реализация)

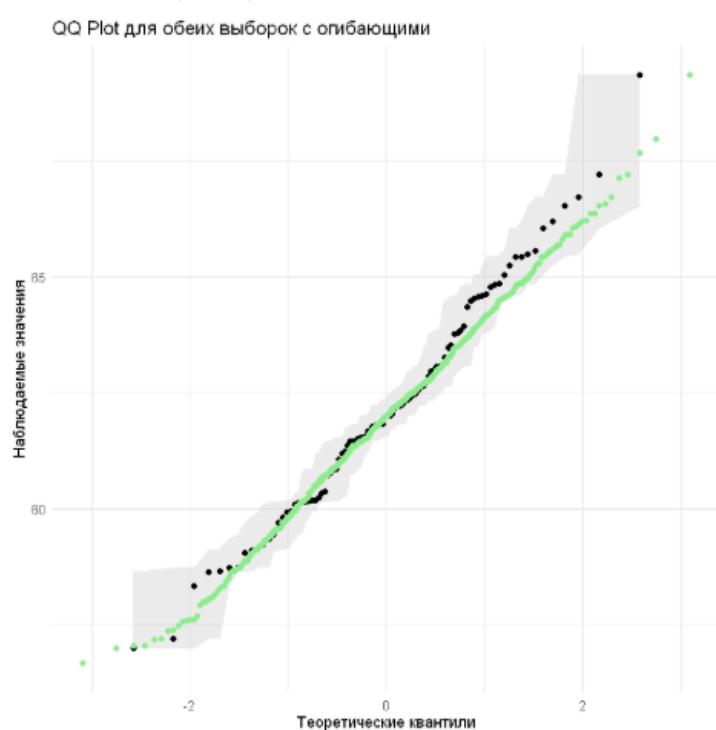


С доверительными интервалами:

[1] "Для выборок роста студентов (**мальчиков**) умеренного(1000) и **малого**(100) объемов":



[2] "Для выборок роста студентов (**девочек**) умеренного(1000) и **малого**(100) объемов":



Критерии и тесты проверки гипотез о нормальности.

Критерий Колмогорова-Смирнова.

Python:

Размер выборки: 100

```
KstestResult(statistic=1.0, pvalue=0.0, statistic_location=60.75, statistic_sign=-1)
KstestResult(statistic=1.0, pvalue=0.0, statistic_location=56.99, statistic_sign=-1)
```

Размер выборки: 1000

```
KstestResult(statistic=1.0, pvalue=0.0, statistic_location=59.16, statistic_sign=-1)
KstestResult(statistic=1.0, pvalue=0.0, statistic_location=56.68, statistic_sign=-1)
```

K-S Test.

Тест 1:

Принимаем альтернативную гипотезу: данные не распределены нормально

Тест 2:

Принимаем альтернативную гипотезу: данные не распределены нормально

Тест 3:

Принимаем альтернативную гипотезу: данные не распределены нормально

Тест 4:

Принимаем альтернативную гипотезу: данные не распределены нормально

R:

Размер выборки: 100

```
Asymptotic one-sample Kolmogorov-Smirnov test

data: head(hs_heights_df$boys, 100)
D = 1, p-value < 2.2e-16
alternative hypothesis: two-sided
```

```
Asymptotic one-sample Kolmogorov-Smirnov test

data: head(hs_heights_df$girls, 100)
D = 1, p-value < 2.2e-16
alternative hypothesis: two-sided
```

Размер выборки: 1000

```
Asymptotic one-sample Kolmogorov-Smirnov test

data: hs_heights_df$boys
D = 1, p-value < 2.2e-16
alternative hypothesis: two-sided
```

```
Asymptotic one-sample Kolmogorov-Smirnov test

data: hs_heights_df$girls
D = 1, p-value < 2.2e-16
alternative hypothesis: two-sided
```

K-S Test.

Тест 1: Принимаем альтернативную гипотезу: данные не распределены нормально
Тест 2: Принимаем альтернативную гипотезу: данные не распределены нормально
Тест 3: Принимаем альтернативную гипотезу: данные не распределены нормально
Тест 4: Принимаем альтернативную гипотезу: данные не распределены нормально

Python:

Значение alpha: 0.05

Результаты тестов для Выборка из 100 элементов - мальчики:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9882, p-value=5.2571e-01
res: True

Андерсон-Дарлинг: Статистика=0.2969, p-value=0.591685433740818
res: True

Крамер фон Мизес: Статистика=33.3333, p-value=4.0719e-09
res: False

Колмогоров-Смирнов (Лиллиефорс): Статистика=1.0000, p-value=0.0000e+00
res: False

Шапиро-Франсия: Статистика=0.9882, p-value=5.2571e-01
res: True

Результаты тестов для Выборка из 100 элементов - девочки:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9875, p-value=4.6927e-01
res: True

Андерсон-Дарлинг: Статистика=0.4094, p-value=0.3446076611712207
res: True

Крамер фон Мизес: Статистика=33.3333, p-value=4.0719e-09
res: False

Колмогоров-Смирнов (Лиллиефорс): Статистика=1.0000, p-value=0.0000e+00
res: False

Шапиро-Франсия: Статистика=0.9875, p-value=4.6927e-01
res: True

Результаты тестов для Выборка из 1000 элементов - мальчики:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9953, p-value=1.3945e-01
res: True

Андерсон-Дарлинг: Статистика=0.3637, p-value=0.43992864330840115
res: True

Крамер фон Мизес: Статистика=166.6667, p-value=0.0000e+00
res: False

Колмогоров-Смирнов (Лиллиефорс): Статистика=1.0000, p-value=0.0000e+00
res: False

Шапиро-Франсия: Статистика=0.9953, p-value=1.3945e-01
res: True

Результаты тестов для Выборка из 1000 элементов - девочки:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9976, p-value=6.8459e-01
res: True

Андерсон-Дарлинг: Статистика=0.1716, p-value=0.9306129241386756
res: True

Крамер фон Мизес: Статистика=166.6667, p-value=0.0000e+00
res: False

Колмогоров-Смирнов (Лиллиефорс): Статистика=1.0000, p-value=0.0000e+00
res: False

Шапиро-Франсия: Статистика=0.9976, p-value=6.8459e-01
res: True

R:

Результаты тестов для Выборка из 100 элементов - мальчики:
True - принимаем гипотезу о нормальности распределения / False - иначе

Шапиро-Уилк: Статистика=0.9882, p-value=5.2571e-01
res: TRUE

Андерсон-Дарлинг: Статистика=0.2969, p-value=5.8501e-01
res: TRUE

Крамер фон Мизес: Статистика=0.0434, p-value=6.1582e-01
res: TRUE

Колмогоров-Смирнов (Лиллиефорс): Статистика=1.0000, p-value=2.7678e-87
res: FALSE

Шапиро-Франция: Статистика=0.9882, p-value=5.2571e-01
res: TRUE

- - - - -
Результаты тестов для Выборка из 100 элементов - девочки:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9875, p-value=4.6927e-01
res: TRUE

Андерсон-Дарлинг: Статистика=0.4094, p-value=3.3876e-01
res: TRUE

Крамер фон Мизес: Статистика=0.0721, p-value=2.6009e-01
res: TRUE

Колмогоров-Смирнов (Лиллиефорс): Статистика=1.0000, p-value=2.7678e-87
res: FALSE

Шапиро-Франция: Статистика=0.9875, p-value=4.6927e-01
res: TRUE

Результаты тестов для Выборка из 1000 элементов - мальчики:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9953, p-value=1.3945e-01
res: TRUE

Андерсон-Дарлинг: Статистика=0.3637, p-value=4.3865e-01
res: TRUE

Крамер фон Мизес: Статистика=0.0512, p-value=4.9389e-01
res: TRUE

Колмогоров-Смирнов (Лиллиефорс): Статистика=1.0000, p-value=0.0000e+00
res: FALSE

Шапиро-Франция: Статистика=0.9953, p-value=1.3945e-01
res: TRUE

- - - - -
Результаты тестов для Выборка из 1000 элементов - девочки:
True - принимаем гипотезу о нормальности распределения / False - иначе.

Шапиро-Уилк: Статистика=0.9953, p-value=1.3945e-01
res: TRUE

Андерсон-Дарлинг: Статистика=0.3637, p-value=4.3865e-01
res: TRUE

Крамер фон Мизес: Статистика=0.0512, p-value=4.9389e-01
res: TRUE

Колмогоров-Смирнов (Лиллиефорс): Статистика=1.0000, p-value=0.0000e+00
res: FALSE

Шапиро-Франция: Статистика=0.9953, p-value=1.3945e-01
res: TRUE

Вывод: Тесты **Шапиро-Уилка, Андерсона-Дарлинга, Шапиро-Франсия** показали, что распределения роста студентов (мальчиков, девочек) имеет нормальное распределение на умеренных и малых выборках.

В то время, как критерий **Коломгорова-Смирнова, его модификация Лиллиефорс и Крамер фон Мизеса** показали, что данные не имеют нормальное распределения на всех объемах выборки.

3. Второй этап практикума.

Применение для проверки гипотез на различных доверительных уровнях (0.9, 0.95, 0.99) статистических критериев.

На этом этапе рассматривается применение статистических методов для проверки различных гипотез, анализа взаимосвязей и исследования структуры данных. Такие задачи возникают при анализе данных в различных областях. Использование статистических критериев помогает принимать обоснованные решения и проверять предположения о свойствах данных, основываясь на определенных уровнях значимости.

Далее будут рассмотрены методы проверки гипотез о средних значениях и дисперсиях (Стьюдента, Уилкоксона-Манна-Уитни, Фишера, Левене и др.), корреляционные взаимосвязи между переменными (Пирсона, Спирмена, Кендалла), мультиколлинеарность, дисперсионный анализ и регрессионные модели.

3.1 Проверка гипотез о равенстве средних и дисперсий.

Проверка гипотез о равенстве средних - одна из основных задач в статистике. Она позволяет сравнивать группы, чтобы выяснить, отличаются ли они по интересующему нас показателю. Методы делятся на параметрические (t-тест) и непараметрические (критерий Уилкоксона -Манна-Уитни), что позволяет учитывать как свойства данных, так и их ограничения.

Критерий Стьюдента (t-тест) - классический метод для проверки гипотез о равенстве средних значений двух выборок. Основан на распределении Стьюдента. Существует двусторонний (равенство средних) и односторонний (проверка больше или меньше одно среднее по сравнению с другим) варианты.

Предполагает нормальность распределения данных.

Для двух выборок предполагает равенство дисперсий (в случае неравенства дисперсий применяется другой метод: t-тест Уэлча - модификация t-теста без учета равенства дисперсий)

Позволяет оценить мощность критерия, т.е. оценить вероятность обнаружения различий, если они существуют. Мощность зависит от объема выборки, уровня значимости и величины эффекта. Соответственно позволяет также найти необходимый объем выборки для заданной мощности.

Сгенерируем выборки с нормальным распределением и проведем на них односторонний и двухсторонний тесты Стьюдента.

```
import numpy as np
import pandas as pd
from scipy import stats
import warnings
warnings.filterwarnings('ignore')

# размер выборки
n = 70

# Генерируем данные
np.random.seed(42)
samples = [
    np.random.normal(50, 11, n),
    np.random.normal(55, 10, n),
    np.random.normal(46, 7, n),
    np.random.normal(49, 7, n),
]
```

```
# Подключаем библиотеки
set.seed(42)

# Размер выборки
n <- 70

# Генерируем данные
samples <- list(
  rnorm(n, mean = 50, sd = 11),
  rnorm(n, mean = 55, sd = 10),
  rnorm(n, mean = 46, sd = 7),
  rnorm(n, mean = 49, sd = 7)
)
```

Односторонний вариант t-теста:

Пусть задано среднее значение: 52

Проверим нулевую гипотезу о равенстве (+ больше, либо меньше) среднего значения выборки 52 на разных уровнях значимости.

Python:

```
Односторонний t-test: sample_mean > mu

alpha=0.1: t_stat=-2.944, p_value=0.998
Вывод: не отклоняем H0

alpha=0.05: t_stat=-2.944, p_value=0.998
Вывод: не отклоняем H0

alpha=0.01: t_stat=-2.944, p_value=0.998
Вывод: не отклоняем H0
```

Во всех трех случаях не отклоняем H_0 , поскольку **p_value** $> \alpha = (0.1, 0.05, 0.01)$ во всех случаях

R:

```
One Sample t-test

data: sample1
t = -0.9607, df = 69, p-value = 0.3401
alternative hypothesis: true mean is not equal to 52
90 percent confidence interval:
48.15874 53.03276
sample estimates:
mean of x
50.59575

One Sample t-test

data: sample1
t = -0.9607, df = 69, p-value = 0.3401
alternative hypothesis: true mean is not equal to 52
95 percent confidence interval:
47.67973 53.51177
sample estimates:
mean of x
50.59575

One Sample t-test

data: sample1
t = -0.9607, df = 69, p-value = 0.3401
alternative hypothesis: true mean is not equal to 52
99 percent confidence interval:
46.72373 54.46777
sample estimates:
mean of x
50.59575
```

0.3401 > 0.1, следовательно не отклоняем нулевую гипотезу о том, что среднее не равно 52.

Python:

```
Односторонний t-test: sample_mean < mu

alpha=0.1: t_stat=-2.944, p_value=0.002
Вывод: отклоняем H0

alpha=0.05: t_stat=-2.944, p_value=0.002
Вывод: отклоняем H0

alpha=0.01: t_stat=-2.944, p_value=0.002
Вывод: отклоняем H0
```

Во всех трех случаях отклоняем H_0 , поскольку $p\text{-value} < \alpha = (0.1, 0.05, 0.01)$ во всех случаях

R:

```
One Sample t-test

data: sample1
t = -0.9607, df = 69, p-value = 0.83
alternative hypothesis: true mean is greater than 52
90 percent confidence interval:
48.70439      Inf
sample estimates:
mean of x
50.59575
One Sample t-test

data: sample1
t = -0.9607, df = 69, p-value = 0.83
alternative hypothesis: true mean is greater than 52
95 percent confidence interval:
48.15874      Inf
sample estimates:
mean of x
50.59575
One Sample t-test

data: sample1
t = -0.9607, df = 69, p-value = 0.83
alternative hypothesis: true mean is greater than 52
99 percent confidence interval:
47.11453      Inf
sample estimates:
mean of x
50.59575
```

Во всех трех случаях не отклоняем H_0 (среднее не больше), поскольку $p\text{-value} > \alpha = (0.1, 0.05, 0.01)$ во всех случаях

R:

```
One Sample t-test

data: sample1
t = -0.9607, df = 69, p-value = 0.17
alternative hypothesis: true mean is less than 52
90 percent confidence interval:
-Inf 52.48711
sample estimates:
mean of x
50.59575

One Sample t-test

data: sample1
t = -0.9607, df = 69, p-value = 0.17
alternative hypothesis: true mean is less than 52
95 percent confidence interval:
-Inf 53.03276
sample estimates:
mean of x
50.59575

One Sample t-test

data: sample1
t = -0.9607, df = 69, p-value = 0.17
alternative hypothesis: true mean is less than 52
99 percent confidence interval:
-Inf 54.07696
sample estimates:
mean of x
50.59575
```

Во всех трех случаях отклоняем H_0 , поскольку $p\text{-value} < \alpha = (0.1, 0.05, 0.99)$ во всех случаях

Теперь реализуем оценку мощности критерия для одновыборочного теста:

Python:

Мощность критерия при $\alpha=0.1$: 0.994

Мощность критерия при $\alpha=0.05$: 0.985

Мощность критерия при $\alpha=0.01$: 0.934

R:

```
One-sample t test power calculation

      n = 70
      delta = 1
      sd = 2
      sig.level = 0.1
      power = 0.9937367
alternative = two.sided
One-sample t test power calculation

      n = 70
      delta = 1
      sd = 2
      sig.level = 0.05
      power = 0.9847848
alternative = two.sided
One-sample t test power calculation

      n = 70
      delta = 1
      sd = 2
      sig.level = 0.01
      power = 0.93397
alternative = two.sided
```

*Определим **необходимый объем** выборки для достижения заданной точности **0.9**:*

Python:

Необходимый объем выборки для достижения мощности 0.9 при alpha=0.1: 71

Необходимый объем выборки для достижения мощности 0.9 при alpha=0.05: 87

Необходимый объем выборки для достижения мощности 0.9 при alpha=0.01: 124

R:

```
One-sample t test power calculation

n = 35.65268
delta = 1
sd = 2
sig.level = 0.1
power = 0.9
alternative = two.sided
One-sample t test power calculation

n = 43.99552
delta = 1
sd = 2
sig.level = 0.05
power = 0.9
alternative = two.sided
One-sample t test power calculation

n = 62.87024
delta = 1
sd = 2
sig.level = 0.01
power = 0.9
alternative = two.sided
```

Теперь проведем двусторонний вариант критерия Стьюдента для двух независимых выборок.

Проведем его для 3 и 4 выборки, так как у них равные дисперсии.

```
# Выделяем 3 и 4 выборки
sample3 = samples[2]
sample4 = samples[3]
```

```
# Выделяем 3 и 4 выборки
sample3 <- samples[[2]]
sample4 <- samples[[3]]
```

Python:

Двусторонний t-test:

```
alpha=0.1: t_stat=-1.425, p_value=0.156
```

Вывод: не отклоняем H_0

```
alpha=0.05: t_stat=-1.425, p_value=0.156
```

Вывод: не отклоняем H_0

```
alpha=0.01: t_stat=-1.425, p_value=0.156
```

Вывод: не отклоняем H_0

***Вывод:** во всех случаях $p_value^* > \alpha = (0.1, 0.05, 0.01)$ \implies не отклоняем H_0 : не приходится говорить о разнице средних

R:

```
Welch Two Sample t-test

data: sample3 and sample4
t = 6.5127, df = 121.72, p-value = 1.74e-09
alternative hypothesis: true difference in means is not equal to 0
90 percent confidence interval:
 6.566148 11.049250
sample estimates:
mean of x mean of y
 54.13141 45.32371
Welch Two Sample t-test

data: sample3 and sample4
t = 6.5127, df = 121.72, p-value = 1.74e-09
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 6.130439 11.484960
sample estimates:
mean of x mean of y
 54.13141 45.32371
Welch Two Sample t-test

data: sample3 and sample4
t = 6.5127, df = 121.72, p-value = 1.74e-09
alternative hypothesis: true difference in means is not equal to 0
99 percent confidence interval:
 5.268722 12.346677
sample estimates:
mean of x mean of y
 54.13141 45.32371
```

***Вывод:** во всех случаях $p_value^* > \alpha = (0.1, 0.05, 0.01)$ \implies не отклоняем H_0 : не приходится говорить о разнице средних

Реализация оценки мощности критерия для двухвыборочного теста.

Python:

Мощность критерия при $\alpha=0.1$: 0.642

Мощность критерия при $\alpha=0.05$: 0.517

Мощность критерия при $\alpha=0.01$: 0.276

R:

```
Two-sample t test power calculation
```

```
  n = 70
  d = 1.100842
sig.level = 0.1
power = 0.9999993
alternative = two.sided
```

NOTE: n is number in *each* group

```
Two-sample t test power calculation
```

```
  n = 70
  d = 1.100842
sig.level = 0.05
power = 0.9999967
alternative = two.sided
```

NOTE: n is number in *each* group

```
Two-sample t test power calculation
```

```
  n = 70
  d = 1.100842
sig.level = 0.01
power = 0.9999428
alternative = two.sided
```

Пусть теперь задана мощность критерия **0.9**. Определим необходимый объем выборки для нее.

Python:

Необходимый объем выборки для достижения мощности 0.9 при $\alpha=0.1$: 147

Необходимый объем выборки для достижения мощности 0.9 при $\alpha=0.05$: 181

Необходимый объем выборки для достижения мощности 0.9 при $\alpha=0.01$: 257

R:

```
Two-sample t test power calculation
```

```
  n = 14.86199
  d = 1.100842
sig.level = 0.1
power = 0.9
alternative = two.sided
```

NOTE: n is number in *each* group

```
Two-sample t test power calculation
```

```
  n = 18.35689
  d = 1.100842
sig.level = 0.05
power = 0.9
alternative = two.sided
```

NOTE: n is number in *each* group

```
Two-sample t test power calculation
```

```
  n = 26.26899
  d = 1.100842
sig.level = 0.01
power = 0.9
alternative = two.sided
```

NOTE: n is number in *each* group

Критерий Уилкоксона-Манна-Уитни - непараметрический аналог критерия Стьюдента, сравнивающий медианы двух выборок, упорядочивая значения по рангам (порядку), поэтому он относится к ранговым критериям.

Он не требует нормальности данных и более устойчив к выбросу.

Применяется, когда данные не соответствуют t-тесту.

Для проведения теста воспользуемся **I датасетом (см. пункт 2.1)**.

Проверим гипотезу о равенстве средних рангов спортсменов из Италии и Великобритании. Нулевая гипотеза состоит в равенстве средних рангов весов спортсменов.

```
country
United States          580
Canada                  204
Germany                164
Great Britain           154
Italy                   154
...
Latvia                 1
Australia  Russian Federation   1
Guinea Bissau          1
Cyprus     Greece          1
Independent Olympic Athletes Timor-Leste  1
Name: count, Length: 191, dtype: int64
```

```
# Сформируем выборки
IT_group = df[df['country'] == 'Italy']['weight']
GB_group = df[df['country'] == 'Great Britain']['weight']
```

```
# Формируем выборки
IT_group <- df$weight[df$country == "Italy"]
GB_group <- df$weight[df$country == "Great Britain"]
```

Python:

Тест Уилкоксона-Манна-Уитни (две независимые выборки):

alpha=0.1: U-stat=12521.5, p_value=0.396

Вывод: не отклоняем H0

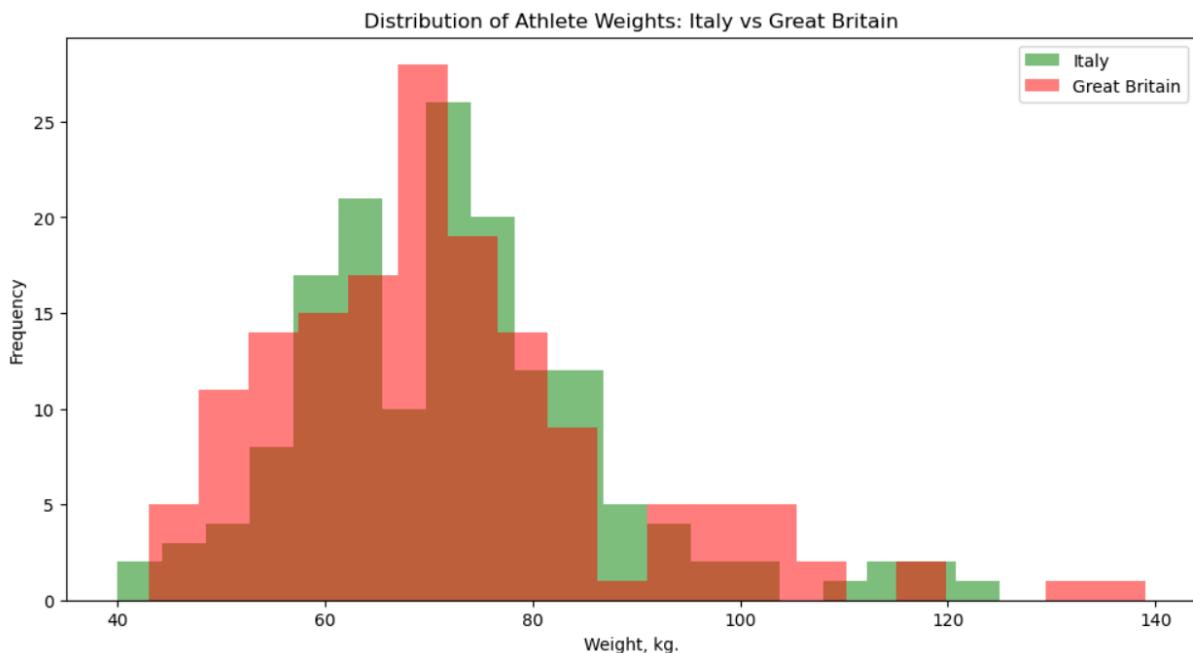
alpha=0.05: U-stat=12521.5, p_value=0.396

Вывод: не отклоняем H0

alpha=0.01: U-stat=12521.5, p_value=0.396

Вывод: не отклоняем H0

Вывод: по результату теста *p_value* > $\alpha = (0.1, 0.05, 0.01)$ \Rightarrow > не отклоняем нулевую гипотезу: не можем говорить о том, что распределения весов из Италии и Великобритании отличаются, т.е. средние рангов не отличаются. Да, на гистограмме можно заметить, что есть различия между значениями весов, но смещения центров распределений не такое значительное, соответственно можно сделать вывод, что статистически данные не сильно различаются.



R:

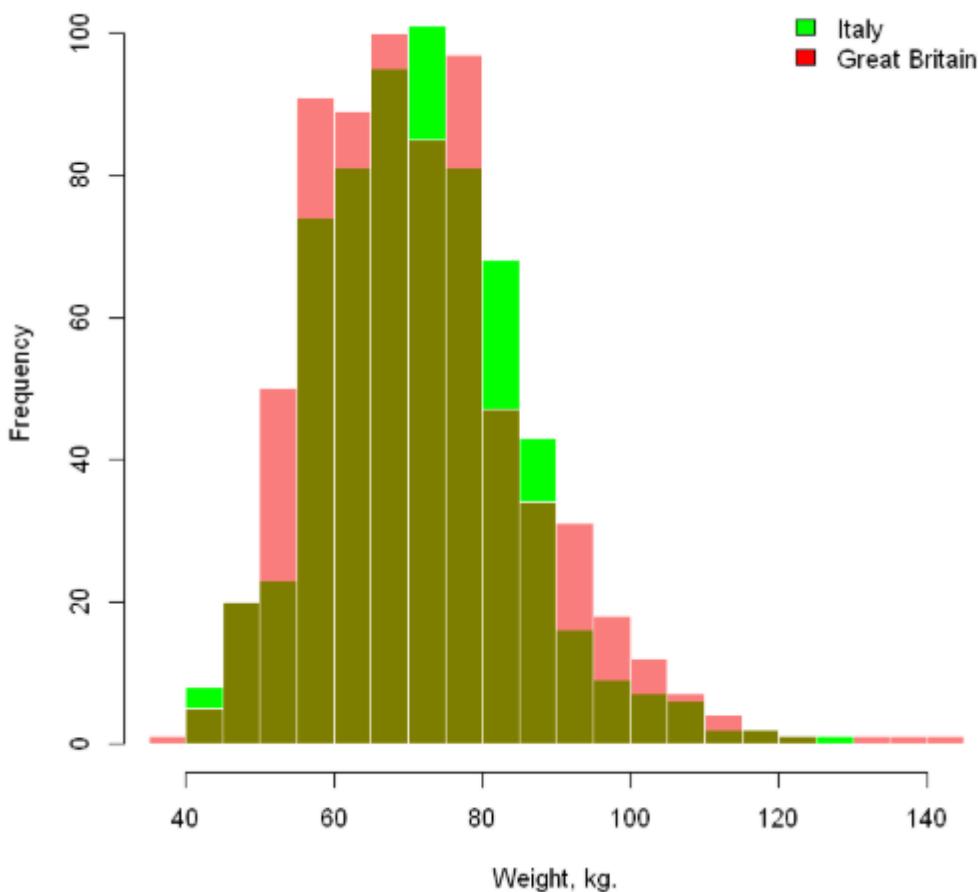
Wilcoxon rank sum test with continuity correction

```
data: IT_group and GB_group
W = 231290, p-value = 0.2034
alternative hypothesis: true location shift is not equal to 0
Wilcoxon rank sum test with continuity correction

data: IT_group and GB_group
W = 231290, p-value = 0.2034
alternative hypothesis: true location shift is not equal to 0
Wilcoxon rank sum test with continuity correction

data: IT_group and GB_group
W = 231290, p-value = 0.2034
alternative hypothesis: true location shift is not equal to 0
```

Distribution of Athlete Weights: Italy vs Great Britain



Проверка об однородности дисперсий.

Критерий Фишера - параметрический метод для сравнения дисперсий двух групп. Рассчитывается как отношение дисперсий двух групп. Статистика теста следует распределению Фишера.

Требует нормальности распределений в обоих выборках и независимости. Чувствителен к выбросам и гетероскедастичности.

Применяется зачастую перед использованием критерия Стьюдента. В регрессионном анализе позволяет оценить значимость линейных регрессионных моделей. В дисперсионном анализе позволяет оценивать значимость факторов и их взаимодействия.

Позволяет сделать выводы о равномерном распределении данных. Как правило, если гипотеза с использованием Критерия Фишера верна, то для сравнения средних можно воспользоваться более мощным критерием.

Возьмем начальные группы, которые были сгенерированы для критерия Стьюдента и проверим гипотезу нулевую гипотезу о равенстве дисперсий.

Python:

Критерий Фишера:

```
alpha=0.1: F_stat=2.025, p_value=0.004
```

Вывод: Отвергаем нулевую гипотезу (дисперсии различаются).

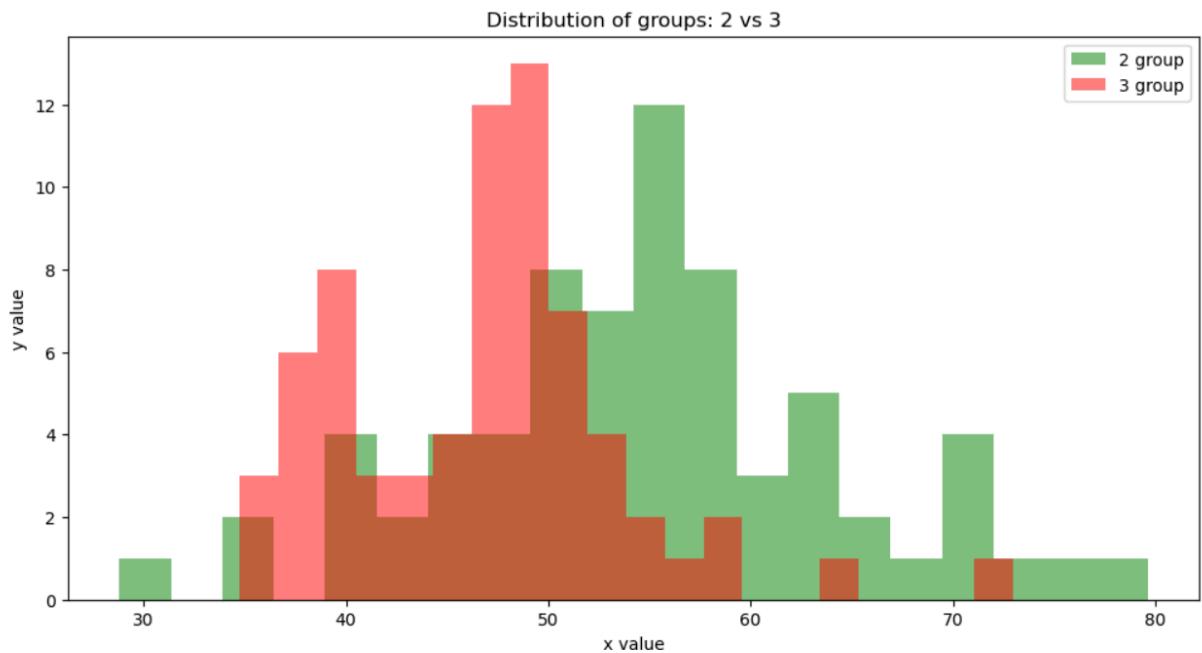
```
alpha=0.05: F_stat=2.025, p_value=0.004
```

Вывод: Отвергаем нулевую гипотезу (дисперсии различаются).

```
alpha=0.01: F_stat=2.025, p_value=0.004
```

Вывод: Отвергаем нулевую гипотезу (дисперсии различаются).

Вывод: Отклоняем H_0 во всех случаях, дисперсии отличаются



R:

F test to compare two variances

```

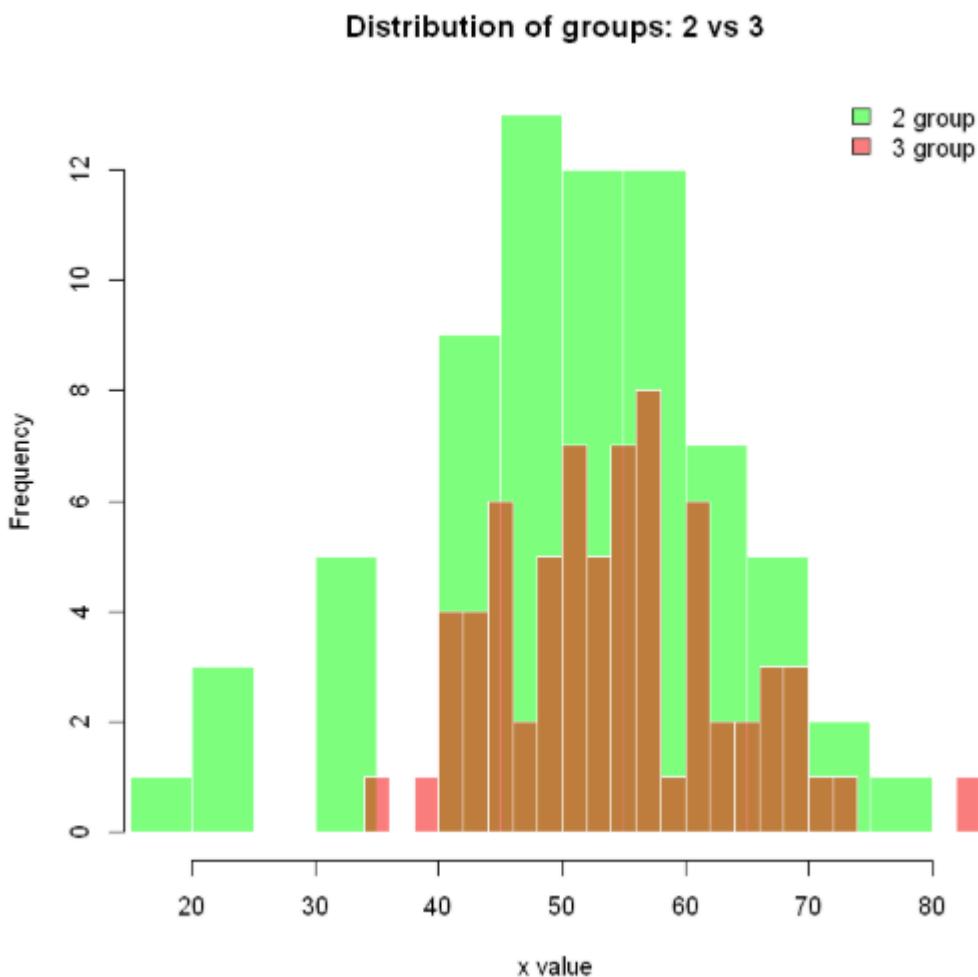
data: sample2 and sample3
F = 1.7108, num df = 69, denom df = 69, p-value = 0.02718
alternative hypothesis: true ratio of variances is not equal to 1
90 percent confidence interval:
1.148175 2.549065
sample estimates:
ratio of variances
1.710782
F test to compare two variances

data: sample2 and sample3
F = 1.7108, num df = 69, denom df = 69, p-value = 0.02718
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
1.063032 2.753232
sample estimates:
ratio of variances
1.710782
F test to compare two variances

data: sample2 and sample3
F = 1.7108, num df = 69, denom df = 69, p-value = 0.02718
alternative hypothesis: true ratio of variances is not equal to 1
99 percent confidence interval:
0.9134702 3.2040175
sample estimates:
ratio of variances
1.710782

```

Вывод: Отклоняем H_0 во всех случаях, дисперсии отличаются



Критерий Левене - непараметрический тест для проверки равенства дисперсий. Основан на сравнении средних абсолютных отклонений (либо от медианы, либо от среднего) в группах. Менее чувствителен к отклонениям от нормальности. Обеспечивает корректность предпосылок для методов, требующих равенства дисперсий.

Используется часто, так как балансирует между устойчивостью к нарушениям и мощностью.

Возьмем **I датасет** и проверим гипотезу о равенстве дисперсий сразу нескольких стран.

Python:

```
# Возьмем для разных стран из датасета спортсменов (см. выше) и проверим разные группы весов спортсменов на равенство дисперсий
IT_sample = df[df['country'] == 'Italy']['weight'] # Италия
GB_sample = df[df['country'] == 'Great Britain']['weight'] # Великобритания
AUS_sample = df[df['country'] == 'Australia']['weight'] # Австралия
USA_sample = df[df['country'] == 'United States']['weight'] # Штаты
HUN_sample = df[df['country'] == 'Hungary']['weight'] # Венгрия
GR_sample = df[df['country'] == 'Germany']['weight'] # Германия
```

R:

```
# Формируем выборки для каждой страны
IT_sample <- df[df$country == 'Italy', 'weight'] # Италия
GB_sample <- df[df$country == 'Great Britain', 'weight'] # Великобритания
AUS_sample <- df[df$country == 'Australia', 'weight'] # Австралия
USA_sample <- df[df$country == 'United States', 'weight'] # Штаты
HUN_sample <- df[df$country == 'Hungary', 'weight'] # Венгрия
GR_sample <- df[df$country == 'Germany', 'weight'] # Германия
```

Проводим тест для критерия Левене:

Python:

Критерий Левене:

alpha=0.1: F_stat=2.025, p_value=0.628

Вывод: не отклоняем H0

alpha=0.05: F_stat=2.025, p_value=0.628

Вывод: не отклоняем H0

alpha=0.01: F_stat=2.025, p_value=0.628

Вывод: не отклоняем H0

R:

A anova: 2 × 3		
	Df	F value
	<int>	<dbl>
group	5	4.12533
	4412	NA
		NA

0.0009698544 << 0.1, соответственно отклоняем **нулевую гипотезу**. Веса спортсменов разных стран статистически различаются.

Критерий Бартлетта проверяет гипотезу о равенстве дисперсий в нескольких группах. Он основан на предположении, что данные распределены нормально, он чувствителен к нарушению данного предположения. Если распределение сильно отклоняется от нормального, то результаты критерия могут быть ненадежными.

Проверим распределения весов спортсменов по странам на нормальность.

Python:

```
Тест Шапиро-Уилка на нормальность распределений:  
ShapiroResult(statistic=0.9454338788854248, pvalue=1.0822028461405066e-05)  
ShapiroResult(statistic=0.9251457568286688, pvalue=3.4370398767905684e-07)  
ShapiroResult(statistic=0.9066105794874437, pvalue=8.095811611134984e-06)  
ShapiroResult(statistic=0.9659126426120617, pvalue=2.3760128265702485e-10)  
ShapiroResult(statistic=0.9529930757799643, pvalue=0.009041020551677431)  
ShapiroResult(statistic=0.9624393455056247, pvalue=0.00020485087328055596)
```

Видно, что распределения весов спортсменов по странам не удовлетворяют нормальному распределению, а значит результат критерия может быть недостоверным.

Критерий Бартлетта:

```
alpha=0.1: F_stat=2.025, p_value=0.515  
Вывод: не отклоняем H0
```

```
alpha=0.05: F_stat=2.025, p_value=0.515  
Вывод: не отклоняем H0
```

```
alpha=0.01: F_stat=2.025, p_value=0.515  
Вывод: не отклоняем H0
```

R:

```
[1] "Тест Шапиро-Уилка на нормальность распределений
```

```
Shapiro-Wilk normality test
```

```
data: IT_sample  
W = 0.98008, p-value = 1.255e-07
```

```
Shapiro-Wilk normality test
```

```
data: GB_sample  
W = 0.96064, p-value = 1.006e-12
```

```
Shapiro-Wilk normality test
```

```
data: AUS_sample  
W = 0.96789, p-value = 1.118e-10
```

```
Shapiro-Wilk normality test
```

```
data: USA_sample  
W = 0.96765, p-value < 2.2e-16
```

```
Shapiro-Wilk normality test
```

```
data: HUN_sample  
W = 0.9602, p-value = 9.694e-09
```

```
Shapiro-Wilk normality test
```

```
data: GR_sample  
W = 0.96248, p-value = 6.052e-11
```

```
Bartlett test of homogeneity of variances
```

```
data: weight by country  
Bartlett's K-squared = 26.109, df = 5, p-value = 8.498e-05
```

8.498e-05 << 0.1, следовательно отклоняем нулевую гипотезу.

Проведем также тест критерия Бартлетта на нормально-распределенных данных. Для наглядности возьмем начальные 1 и 2 группы для теста.

Python:

Критерий Бартлетта: нормальные данные

alpha=0.1: F_stat=2.025, p_value=0.994

Вывод: не отклоняем H0

alpha=0.05: F_stat=2.025, p_value=0.994

Вывод: не отклоняем H0

alpha=0.01: F_stat=2.025, p_value=0.994

Вывод: не отклоняем H0

R:

```
Bartlett test of homogeneity of variances
```

```
data: list(samples[[1]], samples[[2]])
Bartlett's K-squared = 4.8795, df = 1, p-value = 0.02718
```

P_value < 0.1, 0.05, но больше 0.01, значит на уровне значимости 0.01 мы не отвергаем нулевую гипотезу о гомогенности дисперсий.

Критерий Флигнера-Килина - непараметрический аналог критерия Бартлетта для проверки равенства дисперсий в нескольких выборках. Использует ранги абсолютных отклонений от медиан, вместо дисперсий, что делает его ранговым критерием. Является аналогом однофакторного дисперсионного анализа (**см. пункт 3.5**), но на рангах.

Не требует нормальности данных, устойчив к выбросам и асимметрии распределения. Подходит для данных с выбросами или с распределением, сильно отличным от нормального.

Также проверим дисперсии стран.

Python:

Критерий Флигнера-Килина:

alpha=0.1: F_stat=2.025, p_value=0.354

Вывод: не отклоняем H0

alpha=0.05: F_stat=2.025, p_value=0.354

Вывод: не отклоняем H0

alpha=0.01: F_stat=2.025, p_value=0.354

Вывод: не отклоняем H0

R:

Fligner-Killeen test of homogeneity of variances

```
data: weight by country
Fligner-Killeen:med chi-squared = 22.032, df = 5, p-value = 0.0005162
0.0005162 << 0.1 -> отклоняем нулевую гипотезу.
```

3.2 Исследование корреляционных взаимосвязей в данных.

Корреляционный анализ используется для оценки силы и направления взаимосвязи между переменными. Важно определить, насколько изменения одной переменной связаны с изменением другой. Например, как возраст влияет на доход или как температура связана с объемом продаж мороженого. Выбор метода (Пирсон, Спирмен, Кендалл) зависит от линейности и распределения данных. Исследование корреляций позволяет понять структуру данных и выявить потенциальные зависимости для дальнейшего анализа.

Корреляция Пирсона измеряет линейную связь между данными. Она рассчитывается как:

$$r_{XY} = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \cdot \sum(Y_i - \bar{Y})^2}} = \frac{cov(X, Y)}{\sqrt{s_X^2 s_Y^2}},$$

где X, Y - выборки с наблюдениями, \bar{X}, \bar{Y} - выборочные средние выборок X, Y ,

$s_X^2 s_Y^2$ - выборочные дисперсии выборок X, Y .

$r_{XY} \in [-1, 1]$:

- $r_{XY} = 1$: идеальная положительная линейная связь.
- $r_{XY} = -1$: идеальная отрицательная линейная связь.
- $r_{XY} = 0$: отсутствие линейной связи.

Корреляция Пирсона допускает следующие предположения:

- Данные должны быть нормально распределены;
- Связь между переменными линейная.

Проверим взаимосвязь между 1 и 2 начальными группами (**см. критерий Стьюдента**).

Python:

Коэффициент корреляции Пирсона:

```
alpha=0.1: corr_coeff: 0.0746, p_value=0.5394
```

Вывод: не отклоняем H_0

```
alpha=0.05: corr_coeff: 0.0746, p_value=0.5394
```

Вывод: не отклоняем H_0

```
alpha=0.01: corr_coeff: 0.0746, p_value=0.5394
```

Вывод: не отклоняем H_0

R:

```
Pearson's product-moment correlation
```

```
data: sample1 and sample2
```

```
t = 1.306, df = 68, p-value = 0.196
```

```
alternative hypothesis: true correlation is not equal to 0
```

```
95 percent confidence interval:
```

```
-0.08154699  0.37752229
```

```
sample estimates:
```

```
cor
```

```
0.1564243
```

Значение **p_value** > 0.1, соответственно не отклоняем нулевую гипотезу, коэффициент получился статистически значимым.

Корреляция Спирмена измеряет монотонную связь (не обязательно линейную) между переменными.

Рассчитывается на основе рангов переменных:

$$\rho = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)},$$

где $d_i = R_i - S_i$ - разница между рангами x_i и y_i в рядах X и Y соответственно.

Диапазон значений такой же как и у Пирсона (-1 до 1).

Не требует нормальности данных и работает с нелинейными зависимостями (если они монотонные).

Возьмем также **I датасет** и проверим связь роста и веса спортсменов. Нулевая гипотеза: нет монотонной зависимости.

Python:

Коэффициент корреляции Спирмена (`weight vs height`):

`alpha=0.1: corr_coeff: 0.8093, p_value=0.0000`

Вывод: отклоняем H_0

`alpha=0.05: corr_coeff: 0.8093, p_value=0.0000`

Вывод: отклоняем H_0

`alpha=0.01: corr_coeff: 0.8093, p_value=0.0000`

Вывод: отклоняем H_0

R:

`Spearman's rank correlation rho`

```
data: df_weights and df_heights
S = 2.4839e+11, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
0.8138474
```

Отвергаем нулевую гипотезу.

Проверим связь идентификатора атлета с его ростом:

Python:

Коэффициент корреляции Спирмена (`athlete_id vs height`)

`alpha=0.1: corr_coeff: -0.0444, p_value=0.0077`

Вывод: отклоняем H_0

`alpha=0.05: corr_coeff: -0.0444, p_value=0.0077`

Вывод: отклоняем H_0

`alpha=0.01: corr_coeff: -0.0444, p_value=0.0077`

Вывод: отклоняем H_0

R:

```
Spearman's rank correlation rho

data: df$athlete_id and df_heights
S = 1.3503e+12, p-value = 0.1186
alternative hypothesis: true rho is not equal to 0
sample estimates:
rho
-0.01103274
```

Коэффициент корреляции статистически значимый. Не отклоняем нулевую гипотезу.

Корреляция Кендалла измеряет монотонную связь по аналогии со Спирменом, но еще и учитывает порядок наблюдений.

Сравниваются парные наблюдения:

$$\tau = \frac{C - D}{\frac{1}{2}n(n - 1)},$$

где C - количество согласованных пар и D - количество несогласованных пар.

Диапазон значений аналогичен диапазону Пирсона и Спирмена.

Более устойчив к выбросам. Медленнее по вычислениям на больших данных.

Стоит отметить, что корреляция указывает только на связь, но не доказывает, что одна переменная влияет на другую. Высокая корреляция в большинстве случаев указывает на мультиколлинеарность в данных.

При нормальном распределении данных все три метода (Пирсон, Спирмен, Кендалл) дают схожие результаты.

Проверим также связь роста и веса и разберем пример на выборке с нормальным распределением.

Python:

Коэффициент корреляции Кендалла (height vs weight):

```
alpha=0.1: corr_coeff: 0.6319, p_value=0.0000  
Вывод: отклоняем H0
```

```
alpha=0.05: corr_coeff: 0.6319, p_value=0.0000  
Вывод: отклоняем H0
```

```
alpha=0.01: corr_coeff: 0.6319, p_value=0.0000  
Вывод: отклоняем H0
```

R:

```
Kendall's rank correlation tau
```

```
data: df_heights and df_weights  
z = 131.77, p-value < 2.2e-16  
alternative hypothesis: true tau is not equal to 0  
sample estimates:  
tau  
0.6375541
```

Отклоняем нулевую гипотезу

Пример на начальных выборках с нормальным распределением.

Python:

Коэффициент корреляции Кендалла (Нормальное распределение):

```
alpha=0.1: corr_coeff: 0.0435, p_value=0.5945  
Вывод: не отклоняем H0
```

```
alpha=0.05: corr_coeff: 0.0435, p_value=0.5945  
Вывод: не отклоняем H0
```

```
alpha=0.01: corr_coeff: 0.0435, p_value=0.5945  
Вывод: не отклоняем H0
```

R:

```
Kendall's rank correlation tau
```

```
data: sample1 and sample2  
z = 1.1407, p-value = 0.254  
alternative hypothesis: true tau is not equal to 0  
sample estimates:  
tau  
0.0931677
```

Не отклоняем нулевую гипотезу.

3.3 Исследование значимой взаимосвязи между данными.

Взаимосвязь категориальных данных играет ключевую роль в анализе, когда нужно проверить гипотезы о связи переменных. Например, существует ли связь между уровнем образования и выбором профессии или влияет ли реклама на принятие решений.

Статистические методы, такие как хи-квадрат, точный тест Фишера, тест МакНемара и другие, помогают проверить наличие этой связи с учетом различных условий (объем выборки, структура данных). Эти инструменты широко используются в социологических, маркетинговых и медицинских исследованиях.

Критерий согласия Пирсона или хи-квадрат

используется для проверки гипотезы о независимости двух категориальных переменных (анализ таблицы сопряженности) или для проверки, соответствуют ли данные ожидаемому распределению.

1. Сначала строится таблица сопряженности, где фиксируются наблюдаемые частоты (O_{ij}) для каждой категории;
2. Вычисляются ожидаемые частоты (E_{ij}): ожидаемое количество наблюдений в ячейке, рассчитанное на основе независимости переменных;
3. Статистика хи-квадрат:

$$\chi^2 = \sum \frac{(O_{ij} - E_{ij})^2}{E_{ij}},$$

это значение сравнивается с критическим значением из таблицы распределения хи-квадрат.

Этот метод предполагает достаточно большой размер выборки. Ожидаемые частоты не должны быть слишком малыми (рекомендуется $E_{ij} \geq 5$).

Проверим связь между полом спортсмена и их страной.
Нулевая гипотеза: нет статистически значимой зависимости между переменными.

Python:

```
Хи-квадрат тест:  
Статистика: 370.87  
p_value: 0.0000  
Число степеней свободы: 190
```

Вывод: alpha=0.1: отклоняем H0

Вывод: alpha=0.05: отклоняем H0

Вывод: alpha=0.01: отклоняем H0

R:

```
Pearson's Chi-squared test
```

```
data: contingency_table  
X-squared = 1071.3, df = 347, p-value < 2.2e-16
```

Вывод получился статистически значимым, соответственно можно отклонить нулевую гипотезу об отсутствии связи между полом спортсмена и его страной. В итоге между полом и страной атлета есть статистическая значимая зависимость.

Возьмем **III датасет** про фильмы (**см. пункт 2.1**) и проверим статистическую связь между названием фильма и его жанрами.

Python:

```
Хи-квадрат тест:  
Статистика: 18943915.17  
p_value: 0.9703  
Число степеней свободы: 18955524
```

Вывод: alpha=0.1: не отклоняем H0

Вывод: alpha=0.05: не отклоняем H0

Вывод: alpha=0.01: не отклоняем H0

Между названием фильма и его жанрами нет статистически значимой связи.

R:

Pearson's Chi-squared test

```
data: contingency_table_movies
X-squared = 20540024, df = 20615639, p-value = 1
```

Точный тест Фишера применяется для проверки независимости двух категориальных переменных в таблице **2x2**, особенно когда выборка мала и критерий хи-квадрат неприменим. Он основан на точном расчете вероятности наблюдаемого распределения (и более экстремальных случаев), исходя из гипотезы независимости, поэтому он и называется точным.

Вычисляется вероятность для каждого возможного расположения значений с фиксированными суммами строк и столбцов.

Тест предполагает маленький размер выборки и таблицу сопряженности признаков размера **2x2**.

Пример таблицы:

	Юноши	Девушки	Всего
На диете	a	b	a + b
Не на диете	c	d	c + d
Всего	a + c	b + d	n

Фишер показал, что вероятность получения любого такого набора величин задается гипергеометрическим распределением:

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{n!a!b!c!d!}$$

Эта формула дает точную вероятность наблюдения любого специфического набора данных, при условии заданных маргинальных итогов, общего итога и нулевой гипотезе об

одинаковой предрасположенности к диете независимо от пола (соотношение между диетиками и людьми не находящимися на диете для юношей такое же, как для девушек).

*Возьмем такой пример: Пусть у нас есть два фильма: (*Interstellar* и *The Avengers*) и языками (*Russian* и *English*), где значения - это сколько раз был упомянут фильм на том или ином языке, их смоделируем случайным образом.*

H_0 : нет статистической зависимости между фильмом и языком, упоминавшем его.

Python:

```
from scipy.stats import fisher_exact

# Формируем таблицу сопряженности
movies_by_language = pd.DataFrame({
    'English': [450, 520],
    'Russian': [380, 400]
}, index=['Interstellar', 'The Avengers'])

Odds Ratio: 0.9109311740890689, p_value: 0.3359130284681005
Вывод: alpha=0.1: не отклоняем H0

Вывод: alpha=0.05: не отклоняем H0

Вывод: alpha=0.01: не отклоняем H0
```

R:

```
# Тест Фишера
movies_by_language <- matrix(c(450, 520, 380, 400), nrow = 2,
                               dimnames = list(c("Interstellar", "The Avengers"),
                                              c("English", "Russian")))

Fisher's Exact Test for Count Data

data: movies_by_language
p-value = 0.3359
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
0.750828 1.105232
sample estimates:
odds ratio
0.9109807
```

Тест МакНемара используется для сравнения зависимых выборок (парных данных) с двумя категориями, например “до” и “после” эксперимента.

Сначала анализируется таблица сопряженности **2x2** для парных наблюдений, например такая:

	До: Успех	До: Неудача
После: Успех	a	b
После: Неудача	c	d

далее вычисляется статистика:

$$\chi^2 = \frac{(b-c)^2}{b+c},$$

где b и c - несовпадения между “до” и “после”.

Тест МакНемара предполагает, что наблюдения являются парными и каждый исход для каждого объекта является бинарным.

Нулевая гипотеза утверждает, что маргинальные распределения для всех исходов совпадают:

$$p_a + p_b = p_a + p_c$$

$$p_c + p_d = p_b + p_d$$

То есть, $H_0: p_b = p_c$

Смоделируем ситуацию: У нас есть новый диагностический тест (Тест B), который выявляет заболевание заболевание, и мы хотим проверить, насколько он это улучшает по сравнению со старым тестом (Тест A). Возьмем выборку из

100 пациентов и проводим два теста, сравнивая результаты с истинным диагнозом(известен заранее).

Нулевая гипотеза: нет статистической зависимости(разницы) между тестами A и B, т.е. кол-во случаев, когда тест A верен, а тест B ошибается и наоборот, одинаково.

Python:

```
import random
from scipy.stats import chi2

random.seed(42)
n_patients = 100

data = {
    "True Diagnosis": [random.choice([True, False]) for _ in range(n_patients)],
    "Test A Result": [random.choice([True, False]) for _ in range(n_patients)],
    "Test B Result": [random.choice([True, False]) for _ in range(n_patients)],
}
```

Таблица сопряженности:

	Test B True	Test B False
Test A True	35	22
Test A False	20	23

Хи-квадрат статистика: 0.10

p-value: 0.7576

Вывод: alpha=0.1: не отклоняем H0

Вывод: alpha=0.05: не отклоняем H0

Вывод: alpha=0.01: не отклоняем H0

R:

```
# Тест Макнелара
n_patients <- 100
set.seed(42)
data <- data.frame(
  True_Diagnosis = sample(c(TRUE, FALSE), n_patients, replace = TRUE),
  Test_A_Result = sample(c(TRUE, FALSE), n_patients, replace = TRUE),
  Test_B_Result = sample(c(TRUE, FALSE), n_patients, replace = TRUE)
)
```

```

Test B Result
Test A Result TRUE FALSE
    TRUE      26     19
    FALSE      29     26
McNemar's Chi-squared test with continuity correction

data: mcnemar_table
McNemar's chi-squared = 1.6875, df = 1, p-value = 0.1939

```

Тест Кохрана-Мантеля-Хензеля используется для анализа зависимости двух категориальных переменных с учетом влияния третьей переменной (стратифицированных данных).

Данные разделяются на несколько таблиц **2x2**, каждая из которых соответствует уровню контрольной переменной. Затем рассчитывается статистика

$$\chi^2_{CMH} = \frac{(\sum O_{ij} - \sum E_{ij})^2}{\sum Var}$$

где обозначения в числителе идентичны обозначениям в методе хи-квадрат, а в знаменателе берется суммарная дисперсия всех признаков в таблице.

Этот тест проверяет, что отношение шансов одинаково для всех таблиц. Он допускает предположения, что все таблицы по каждой категории стратифицированной переменной и что есть независимость наблюдений между стратифицированными группами.

*Пусть у нас есть следующая ситуация: Собраны данные об очередности рождаемости ребенка, с наличием синдрома Дауна и без него. А также сведения о возрасте матери. Проверим с помощью теста статистическую зависимость между **очередностью рождения и наличием синдрома Дауна**, с учетом **возраста матери**. Нулевая гипотеза все также про независимость переменных.*

Python:

```
# Пример данных: возраст матери, очередность рождения и наличие синдрома Дауна
data = {
    'Mother_Age_Group': ['<30', '<30', '<30', '<30', '30-40', '30-40', '30-40', '30-40', '40+', '40+'],
    'Order_of_Birth': ['1st', '2nd', '1st', '3rd', '2nd', '1st', '2nd', '3rd', '1st', '2nd'],
    'Down_Syndrome': ['Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No']
}
```

Группа возраста: <30

Таблица сопряженности:

Down_Syndrome	0	1
Order_of_Birth		
1	0	2
2	1	0
3	1	0

Статистика хи-квадрат: 4.00, p_value=0.1353

Вывод: alpha=0.1: не отклоняем H0

Вывод: alpha=0.05: не отклоняем H0

Вывод: alpha=0.01: не отклоняем H0

Группа возраста: 30-40

Таблица сопряженности:

Down_Syndrome	0	1
Order_of_Birth		
1	1	0
2	0	2
3	1	0

Статистика хи-квадрат: 4.00, p_value=0.1353

Вывод: alpha=0.1: не отклоняем H0

Вывод: alpha=0.05: не отклоняем H0

Вывод: alpha=0.01: не отклоняем H0

```
Группа возраста: 40+
Таблица сопряженности:
Down_Syndrome  0  1
Order_of_Birth
1              0  1
2              1  0
```

Статистика хи-квадрат: 2.00, p_value=0.1573

Вывод: alpha=0.1: не отклоняем H0

Вывод: alpha=0.05: не отклоняем H0

Вывод: alpha=0.01: не отклоняем H0

R:

```
# Тест Кохрана-Мантеля-Хензеля для возраста матери, очередности рождения и синдрома Дауна
data <- data.frame(
  Mother_Age_Group = factor(c('<30', '<30', '<30', '<30', '30-40', '30-40', '30-40', '30-40', '40+', '40+'),
                             levels = c('<30', '30-40', '40+')),
  Order_of_Birth = factor(c('1st', '2nd', '1st', '3rd', '2nd', '1st', '2nd', '3rd', '1st', '2nd'),
                          levels = c('1st', '2nd', '3rd')),
  Down_Syndrome = c(1, 0, 1, 0, 1, 0, 1, 0, 1, 0)
)
```

Cochran-Mantel-Haenszel test

```
data: mantel_table
Cochran-Mantel-Haenszel M^2 = 2.3529, df = 2, p-value = 0.3084
```

3.4 Проверка наличия мультиколлинеарности в данных.

Мультиколлинеарность, что это за зверь?

Это статистическая проблема, возникающая, когда независимые переменные сильно коррелируют друг с другом. Это означает, что одна переменная может быть выражена через линейную комбинацию других.

Она ухудшает интерпретируемость модели и делает результаты анализа менее надежными, поэтому очень важно уметь с ней бороться.

Чтобы выявлять мультиколлинеарность используются два инструмента: корреляционная матрица и фактор инфляции дисперсии.

Корреляционная матрица - матрица, состоящая из коэффициентов корреляции признаков, которые располагаются на пересечении строк и столбцов. Все диагональные элементы равны 1, так как это пересечение одних и тех же элементов. Эта матрица симметрична, так как коэффициент корреляции между X и Y равен коэффициенту корреляции Y и X. Все элементы матрицы лежат в диапазоне от -1 до 1:

- +1: Полная положительная корреляция.
- 0: Нет линейной связи.
- -1: Полная отрицательная корреляция.

Она помогает понять, как связаны числовые переменные в наборе данных. Высокие коэффициенты корреляции между независимыми переменными могут указывать на мультиколлинеарность.

Она помогает строить регрессионный анализ, а также выявлять различные закономерности.

Берем **III датасет** про фильмы и проверим *следующую гипотезу: отличаются ли средние рейтинги фильмов(vote average) для двух языков(original language): английский и французский.*

Проверим средние обеих групп:

Python:

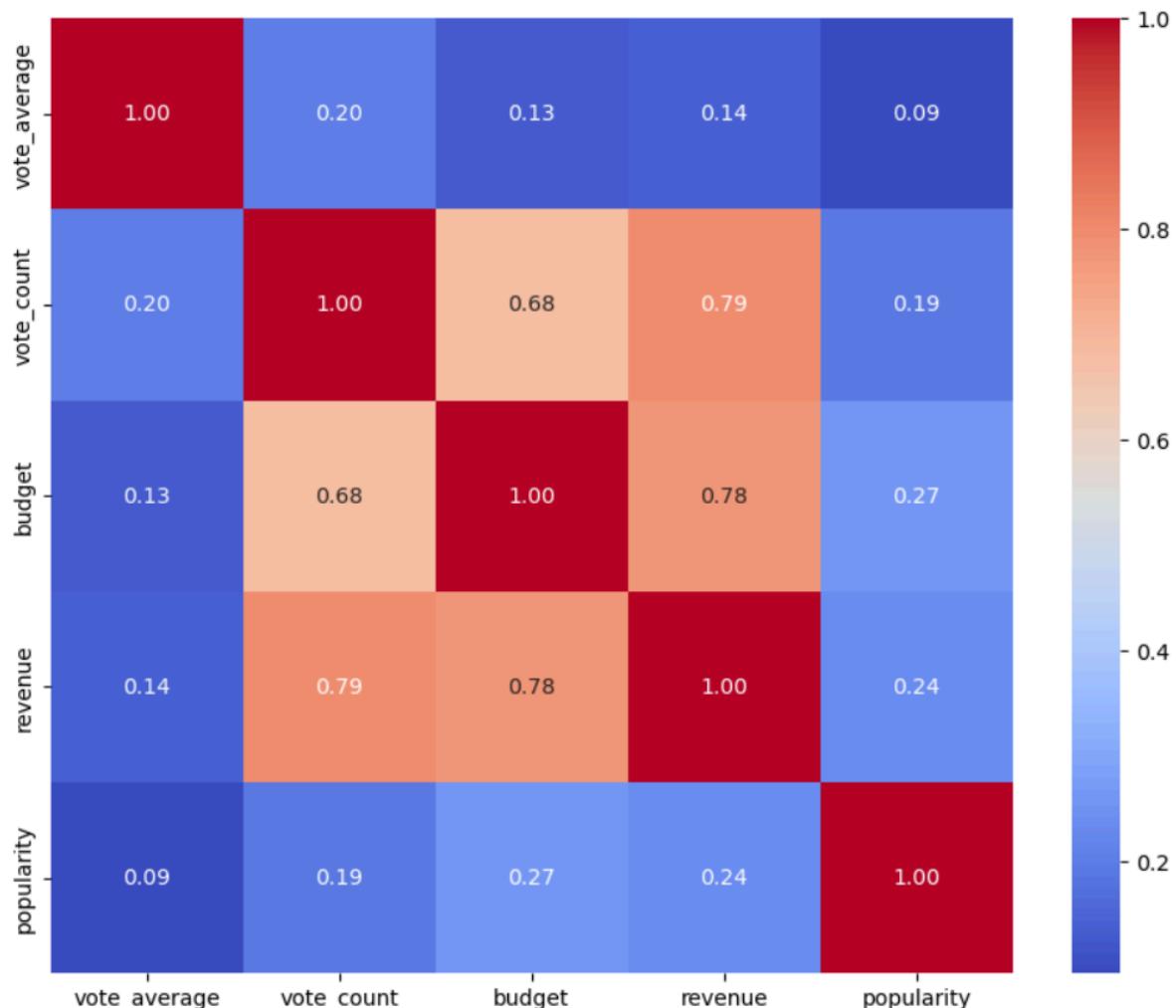
```
from scipy.stats import ttest_ind

# Формируем выборки фильмов на английском и французском языках
en_movies = movie_df[movie_df['original_language'] == "en"]['vote_average']
fr_movies = movie_df[movie_df['original_language'] == "fr"]['vote_average']

Тест Стьюдента.
t_stat=-1.417, p_value=0.158

Вывод: alpha=0.1: не отклоняем H0
Вывод: alpha=0.05: не отклоняем H0
Вывод: alpha=0.01: не отклоняем H0
```

Построим корреляционную матрицу для числовых переменных: *vote_average*, *vote_count*, *budget*, *revenue*, *popularity*



- По корреляционной матрице видно, что сильная корреляция между *budget* и *revenue*:
 - Коэффициент корреляции равен примерно **0.78**.
 - Это может говорить о том, что фильмы с высоким бюджетом, как правило обычно имеют более высокий доход.
- Сильная корреляция между *vote_count* и *revenue*:
 - Корреляция около **0.79**.
 - Это говорит о том, что фильмы с высоким доходом, как правило получают больше голосов, что логично, поскольку популярные фильмы чаще привлекают внимание зрителей
- Умеренная корреляция между *budget* и *vote_count*:
 - Коэффициент равен **0.68**.
 - Следовательно, фильмы с большим бюджетом чаще имеют больше голосов, что может быть связано с их маркетингом или широкой аудиторией.
- Слабая корреляция между *vote_average* и другими параметрами:
 - Наибольшая корреляция у *vote_average* с *vote_count*. (**0.20**) -> средний рейтинг не сильно зависит от бюджета, популярности, дохода или числа голосов.
- Слабая корреляция между *popularity* и другими параметрами:

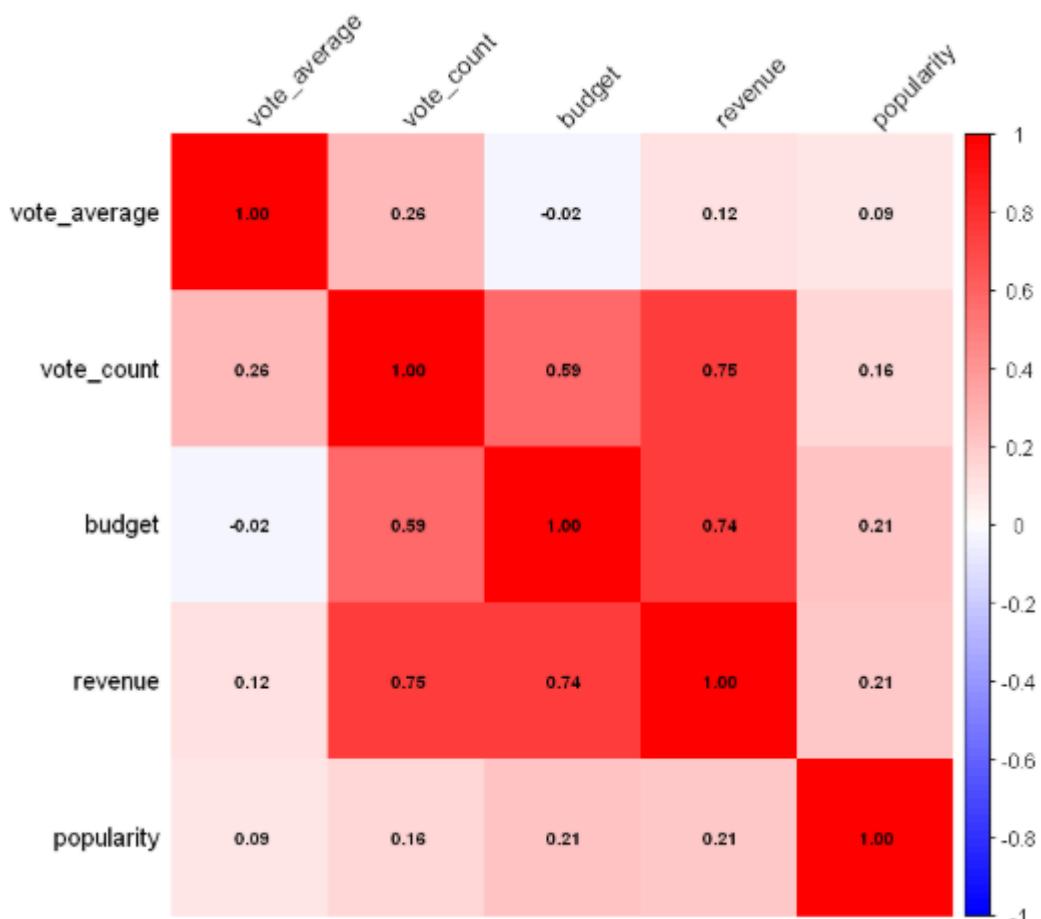
- корреляция между *popularity* и *revenue* составляет около **0.24**
- Популярность имеет слабую связь с доходом, бюджетом или числом голосов, что может быть связано с тем, что она отражает текущий интерес к фильму, а не его абсолютный успех.

R:

```
# Тест Стьюдента для сравнения средних
en_movies <- movie_df %>% filter(original_language == "en") %>% pull(vote_average)
fr_movies <- movie_df %>% filter(original_language == "fr") %>% pull(vote_average)
```

Welch Two Sample t-test

```
data: en_movies and fr_movies
t = -2.9492, df = 762, p-value = 0.003283
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.16419493 -0.03296188
sample estimates:
mean of x mean of y
6.539707 6.638285
```



1. По корреляционной матрице видно, что сильная корреляция между *budget* и *revenue*:
 - Коэффициент корреляции равен примерно *_0.74_*.

- Это может говорить о том, что фильмы с высоким бюджетом, как правило обычно имеют более высокий доход.
- 2. Сильная корреляция между *vote_count* и *revenue*:**
- Корреляция около 0.75.
 - Это говорит о том, что фильмы с высоким доходом, как правило получают больше голосов, что логично, поскольку популярные фильмы чаще привлекают внимание зрителей
- 3. Умеренная корреляция между *budget* и *vote_count*:**
- Коэффициент равен 0.59.
 - Следовательно, фильмы с большим бюджетом чаще имеют большие голоса, что может быть связано с их маркетингом или широкой аудиторией.
- 4. Слабая корреляция между *vote_average* и другими параметрами:**
- Наибольшая корреляция у *vote_average* с *vote_count*. (0.26) -> средний рейтинг не сильно зависит от бюджета, популярности, дохода или числа голосов.
- 5. Слабая корреляция между *popularity* и другими параметрами:**
- корреляция между *popularity* и *revenue* составляет около 0.21
 - Популярность имеет слабую связь с доходом, бюджетом или числом голосов, что может быть связано с тем, что она отражает текущий интерес к фильму, а не его абсолютный успех.

Фактор инфляции дисперсии (VIF) - статистический показатель, который измеряет степень мультиколлинеарности среди независимых переменных в регрессионной модели (мера мультиколлинеарности). Он показывает, насколько дисперсия коэффициента регрессии увеличивается из-за линейной зависимости между предикторами.

Он рассчитывается как:

$$VIF_j = \frac{1}{1 - R_j^2},$$

где R_j^2 - коэффициент детерминации регрессии

переменной X_j на все остальные независимые переменные.

Коэффициент детерминации показывает, насколько хорошо переменная X_j объясняется другими независимыми

переменными. Если R_j^2 близко к 1, то *VIF* становится большим, указывая на сильную мультиколлинеарность .

В пункте **3.6** с помощью коэффициента детерминации будет оцениваться обобщающая способность регрессионной модели,

поэтому можно проследить взаимосвязь между дисперсией и качеством регрессионной модели.

Стандартно полагают следующую **интерпретацию VIF**:

- $VIF = 1$: Полное отсутствие мультиколлинеарности.
Переменная никак не связана с другими независимыми переменными (предикторами).
- $1 < VIF \leq 5$: Умеренная мультиколлинеарность.
Переменную можно будет оставить в модели.
- $VIF > 5 (VIF \gg 5)$: Высокая мультиколлинеарность.
Следует рассмотреть удаление переменной или использование методов снижения мультиколлинеарности.

Этот показатель рассчитывается для каждой независимой переменной отдельно, поэтому он помогает понять, какая именно переменная вызывает проблему и соответственно играет важную роль в выборе предикторов в регрессионной модели.

Рассчитаем его для числовых переменных:

Python:

Фактор инфляции дисперсии (VIF):

	Feature	VIF
0	vote_average	1.261263
1	vote_count	3.238341
2	budget	3.040922
3	revenue	4.140714
4	popularity	1.149178

- `vote_average` : $VIF \approx 1.26$
 - Мультиколлинеарности практически нет.
- `vote_count` : $VIF \approx 3.24$
 - Умеренная мультиколлинеарность, это не вызывает серьезных проблем.
- `budget` : $VIF \approx 3.04$
 - Так же умеренная мультиколлинеарность.
- `revenue` : $VIF \approx 4.14$
 - Уровень мультиколлинеарности немного выше, но не критический.
- `popularity` : $VIF \approx 1.15$
 - Мультиколлинеарность практически отсутствует

Вывод: Значения VIF для всех переменных меньше 5, что указывает на отсутствие серьезной мультиколлинеарности в данных. Зависимость между предикторами слабые и не создают значимых искажений.

R:

```
[1] "Фактор инфляции дисперсии (VIF):"  
      Feature      VIF  
vote_count vote_count 2.288036  
budget       budget    2.268152  
revenue      revenue   3.365644  
popularity  popularity 1.052466
```

- `vote_average` : $VIF \approx 1.26$
 - Мультиколлинеарности практически нет.
- `vote_count` : $VIF \approx 2.29$
 - Умеренная мультиколлинеарность, это не вызывает серьезных проблем.
- `budget` : $VIF \approx 2.26$
 - Так же умеренная мультиколлинеарность.
- `revenue` : $VIF \approx 3.36$
 - Уровень мультиколлинеарности немного выше, но не критический.
- `popularity` : $VIF \approx 1.05$
 - Мультиколлинеарность практически отсутствует

3.5 Дисперсионный анализ

Дисперсионный анализ (ANOVA) - метод статистического анализа, который используется для проверки гипотез о равенстве средних значений в нескольких группах. Его основная цель - определить, существует ли статистически значимая разница между средними значениями сразу нескольких выборок.

Этот метод базируется на анализе разброса данных: он разделяет общую вариабельность данных на компоненты, которые объясняются влиянием разных факторов или случайными ошибками.

ANOVA сравнивает разброс данных между группами и внутри групп:

- Дисперсия между группами отражает влияние фактора, который изучается;
- Дисперсия внутри групп показывает естественную вариацию, вызванную случайными факторами.

Если дисперсия между группами значительно больше дисперсии внутри группы, то можно сделать вывод о влиянии изучаемого фактора на средние значения.

Выделяют следующие виды дисперсионного анализа:

- **Однофакторный дисперсионный анализ:**
Используется для сравнения средних значений одного фактора (например, влияние разных типов удобрений на рост растений);
- **Многофакторный дисперсионный анализ:**
Анализирует влияние двух и более факторов (например, влияние удобрений и типа почвы на рост растений);
- **ANOVA с повторными изменениями:**
Применяется, когда измерения проводятся на одних и тех же объектах в разные моменты времени или при разных условиях.
- **Дисперсионный анализ смешанного типа (смешанные модели):**
Учитывает как фиксированные факторы (интересующие исследователя), так и случайные факторы (например, различные партии продукции).

Дисперсионный анализ допускает предположения о нормальности распределения данных, равенство дисперсий в группах (гомоскедастичность) и независимость наблюдений.

Нарушение хотя бы одного предположения может привести к ошибочным выводам. В таких случаях используют модифицированные методы для ненормальных данных.

Так как он позволяет сравнивать более двух групп, то это делает его более удобным, чем тест Стьюдента.

Метод широко используется в различных сферах деятельности, экономит время и ресурсы, исключая необходимость проведения множественных парных тестов и одновременно является мощным инструментом для проверки гипотез.

Возьмем датасет про олимпийских спортсменов (**I датасет**) и выдвинем следующую гипотезу: *страна не влияет на рост спортсмена*. Возьмем и сформируем следующие группы спортсменов из следующих стран: **United States, Canada, Germany, Great Britain, Italy**

Однофакторный + одномерный.

Python:

```
# Формируем выборку для спортсменов из заданных стран
selected_countries = ["United States", "Canada", "Germany", "Great Britain", "Italy"]
filtered_df = olymp_sport_df[olymp_sport_df['country'].isin(selected_countries)]
```

Однофакторный ANOVA: Влияние страны на рост спортсмена

	sum_sq	df	F	PR(>F)
C(country)	1242.702478	4.0	2.844129	0.023055
Residual	136651.768063	1251.0	NaN	NaN

Вывод: $\alpha=0.1$: отклоняем H_0

Вывод: $\alpha=0.05$: отклоняем H_0

Вывод: $\alpha=0.01$: не отклоняем H_0

*P.S. То, что в residual значения равно NaN это нормально, потому что для остатков не проводится гипотезное тестирование, т.к. они не являются факторами. Заметим, что в данном случае был реализован и одномерный дисперсионный анализ(**height** - одна зависимая переменная и остальные факторы действуют на нее).*

По результатам анализа видно, что при $\alpha = (0.1, 0.05)$ наблюдается влияние страны на рост, в то время при $\alpha=0.01$ нулевая гипотеза о том, что страна не влияет на рост спортсмена. не отклоняется.

R:

```
# Однофакторный ANOVA: влияние страны на рост
selected_countries <- c("United States", "Canada", "Germany", "Great Britain", "Italy")
filtered_df <- olymp_sport_df %>% filter(country %in% selected_countries)
```

```

          Df Sum Sq Mean Sq F value    Pr(>F)
country        4   2171   542.7   4.863 0.000653 ***
Residuals    4156 463798   111.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Многофакторный.

Для многофакторного добавим дополнительный факторы в группы из предыдущего однофакторного варианта с той же гипотезой.

Дополнительные факторы: длина описания(*description_length*) и длина примечаний(*special_notes_description*), которая может свидетельствовать о какой-нибудь важной информации про спортсмена.

Python:

```

ANOVA: Влияние страны, пола, описания и примечаний на рост спортсмена:
            sum_sq      df       F    PR(>F)
C(country)    2112.398018    4.0   6.815326 1.998541e-05
C(sex)         39104.576180   1.0  504.659500 4.355631e-94
C(country):C(sex)  51.506227   4.0   0.166177 9.555663e-01
description_length  227.092050   1.0   2.930710 8.715852e-02
special_notes_length  295.248616   1.0   3.810296 5.116254e-02
Residual        96393.890924 1244.0      NaN      NaN

```

R:

```

          Df Sum Sq Mean Sq F value    Pr(>F)
country        4   2171   543    6.894 1.58e-05 ***
sex             1 135085 135085 1716.056 < 2e-16 ***
description_length  1     500     500    6.350  0.0118 *
special_notes_length  1    1447    1447   18.376 1.85e-05 ***
country:sex      4    164      41    0.519  0.7214
Residuals       4149 326603      79
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Многомерный.

*Добавим еще одну зависимую переменную *weight*, дополнительные факторы оставляем те же, что и в многофакторном варианте.*

Python:

MANOVA: Влияние факторов на рост и вес спортсмена:

:

	Intercept	Value	Num DF	Den DF	F Value	Pr > F
	Wilks' lambda	0.0191	2.0000	1247.0000	31965.7535	0.0000
	Pillai's trace	0.9809	2.0000	1247.0000	31965.7535	0.0000
	Hotelling-Lawley trace	51.2682	2.0000	1247.0000	31965.7535	0.0000
	Roy's greatest root	51.2682	2.0000	1247.0000	31965.7535	0.0000

	C(country)	Value	Num DF	Den DF	F Value	Pr > F
	Wilks' lambda	0.9750	8.0000	2494.0000	3.9650	0.0001
	Pillai's trace	0.0250	8.0000	2496.0000	3.9543	0.0001
	Hotelling-Lawley trace	0.0255	8.0000	1779.1037	3.9769	0.0001
	Roy's greatest root	0.0222	4.0000	1248.0000	6.9413	0.0000

	C(sex)	Value	Num DF	Den DF	F Value	Pr > F
	Wilks' lambda	0.6748	2.0000	1247.0000	300.4919	0.0000
	Pillai's trace	0.3252	2.0000	1247.0000	300.4919	0.0000
	Hotelling-Lawley trace	0.4819	2.0000	1247.0000	300.4919	0.0000
	Roy's greatest root	0.4819	2.0000	1247.0000	300.4919	0.0000

	description_length	Value	Num DF	Den DF	F Value	Pr > F
	Wilks' lambda	0.9973	2.0000	1247.0000	1.7086	0.1815
	Pillai's trace	0.0027	2.0000	1247.0000	1.7086	0.1815
	Hotelling-Lawley trace	0.0027	2.0000	1247.0000	1.7086	0.1815
	Roy's greatest root	0.0027	2.0000	1247.0000	1.7086	0.1815

special_notes_length	Value	Num DF	Den DF	F Value	Pr > F
Wilks' lambda	0.9960	2.0000	1247.0000	2.5267	0.0803
Pillai's trace	0.0040	2.0000	1247.0000	2.5267	0.0803
Hotelling-Lawley trace	0.0041	2.0000	1247.0000	2.5267	0.0803
Roy's greatest root	0.0041	2.0000	1247.0000	2.5267	0.0803

R:

```
Df    Wilks approx F num Df den Df Pr(>F)
country          4 0.99058     4.92      8   8300 4.374e-06 ***
sex              1 0.67844    983.50      2   4150 < 2.2e-16 ***
description_length 1 0.99646     7.38      2   4150 0.0006318 ***
special_notes_length 1 0.99554     9.30      2   4150 9.326e-05 ***
Residuals        4151
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Вывод: По результатам анализа видно, что рост и вес спортсмена зависят от независимых переменных, следовательно мы отклоняем нулевую гипотезу о том, что нет статистически значимой связи между ними.

Дисперсионный анализ с повторными изменениями, постоянными факторами, случайными факторами, и смешанные модели с факторами обоих типов.

Дисперсионный анализ с повторными изменениями:

Исследуются изменения одной и той же группы объектов во времени или при различных условиях. Дисперсионный анализ с постоянными факторами, случайными факторами, и смешанные модели с факторами обоих типов:

- **С постоянными факторами:** Уровни фактора фиксированы, исследование только их влияния.
- **Случайные факторы:** Уровни выбираются случайно, выводы распространяются на всю генеральную совокупность.

- **Смешанные модели:** Включаются как постоянные, так и случайные факторы.

Возьмем следующий пример: Предположим, что у нас есть данные об эффективности двух методов тренировок на выносливость, где замеры проводились у тех же спортсменов в разные моменты времени (до тренировки и через месяц). Проверим: есть ли различия в результатах между методами тренировок с учетом повторных изменений?

Но: различий нет, нет влияния методов и временных этапов на результаты.

1. Постоянные факторы:

- Метод тренировки - 2 уровня, фиксированный.

2. Случайные факторы:

- Спортсмен (случайный фактор), т.к. разные спортсмены могут иметь разные выносливости, независимо друг от друга.

3. Повторные изменения:

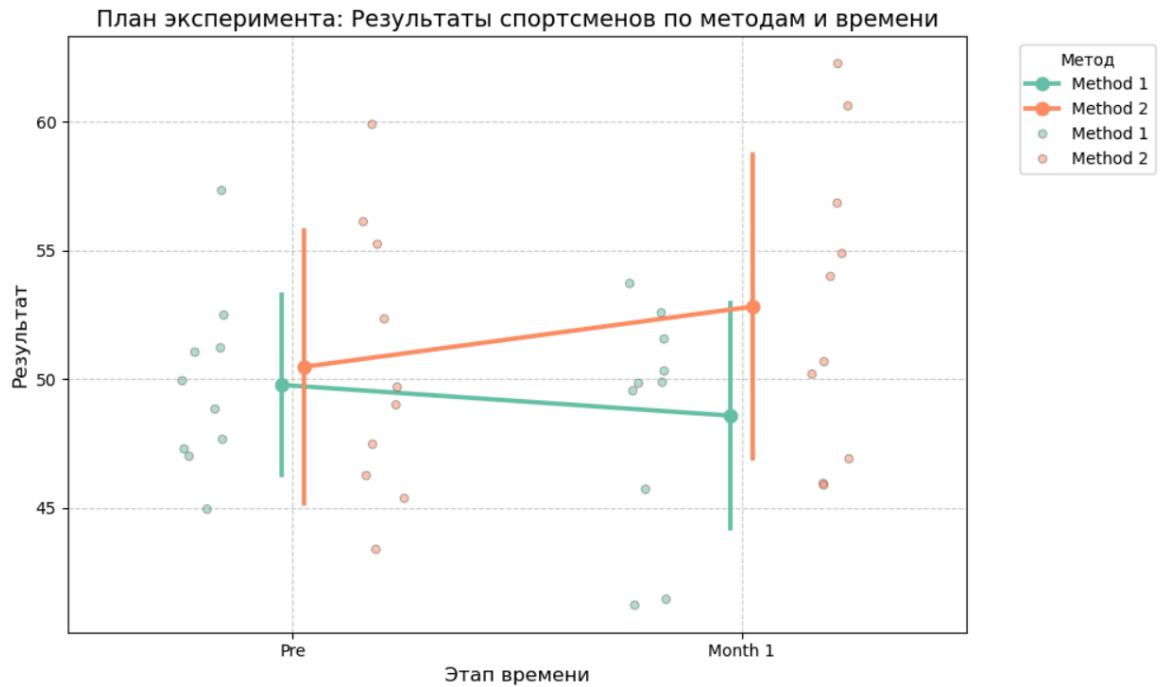
- Время (измерения повторяются на одном и том же спортсмене с течением времени.)

Возьмем смешанную модель, которая включает все эти факторы.

Python:

```
Mixed Linear Model Regression Results
=====
Model: MixedLM      Dependent Variable: Result
No. Observations: 40          Method: REML
No. Groups: 10           Scale: 22.5710
Min. group size: 4          Log-Likelihood: -115.2245
Max. group size: 4          Converged: Yes
Mean group size: 4.0

Coef. Std.Err. z P>|z| [0.025 0.975]
-----
Intercept 49.459 1.349 36.655 0.000 46.814 52.103
Method[T.Method 2] 2.471 1.502 1.645 0.100 -0.474 5.416
Time[T.Pre] -0.575 1.502 -0.383 0.702 -3.519 2.370
Group Var 1.278 0.789
```



Линии соединяют средние значения для каждого метода на двух временных этапах. Индивидуальные точки для спортсменов показывают разброс данных внутри групп. Доверительные интервалы на основе стандартного отклонения.

Вывод: Для 2 метода $P > |z| = 0.1$, при $\alpha = 0.1$ мы можем отвергнуть нулевую гипотезу, т.е. есть статистическая значимая связь между методами и временными этапами и результатами.

При $\alpha = (0.05, 0.01)$ можем не отвергать нулевую гипотезу, т.к. $P > |z| > \alpha$, следовательно нет влияния на результаты со стороны методов и временными этапами.

В рамках данной модели можно сделать вывод о том, что 2 метод тренировки может давать более высокие результаты, но временные этапы не оказывают существенного влияния на результат.

R:

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [lmerModLmerTest]
Formula: Result ~ Method + Time + (1 | Athlete)
Data: df
```

REML criterion at convergence: 246.5

Scaled residuals:

Min	1Q	Median	3Q	Max
-2.1483	-0.4075	-0.1021	0.7853	1.9192

Random effects:

Groups	Name	Variance	Std.Dev.
Athlete	(Intercept)	0.2472	0.4972
Residual		36.3288	6.0273

Number of obs: 40, groups: Athlete, 10

Fixed effects:

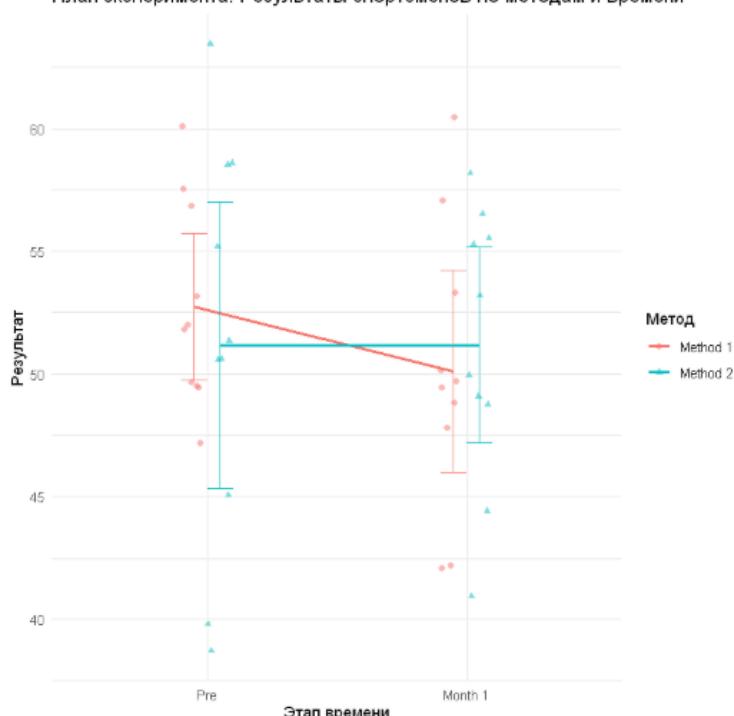
	Estimate	Std. Error	df	t value	Pr(> t)
(Intercept)	52.0803	1.6581	35.2329	31.409	<2e-16 ***
MethodMethod 2	-0.2414	1.9060	27.9999	-0.127	0.900
TimeMonth 1	-1.3146	1.9060	27.9999	-0.690	0.496

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

(Intr)	MthdM2
MethodMthd2	-0.575
TimeMonth 1	-0.575 0.000

План эксперимента: Результаты спортсменов по методам и времени



3.6 Подгонка регрессионных (линейных и нелинейных) моделей к данным и оценка качества аппроксимации

Подгонка регрессионной модели - это процесс построения уравнения, которое максимально точно описывает взаимосвязь между зависимой переменной (откликом) и одной или несколькими независимыми переменными (предикторами). Цель состоит в том, чтобы найти математическую функцию, которая отражает эту связь, минимизируя ошибки между наблюдаемыми и предсказанными значениями.

Этот процесс используется для прогнозирования, анализа влияния факторов и изучения структуры данных.

Регрессия определяет, как изменяется зависимая переменная при изменении одной или нескольких независимых переменных.

Это достигается с помощью линейных (линейная, логистическая, регуляризованная и др.) или нелинейных (полиномиальная (хотя иногда полиномиальная может выступать как частный случай линейной), деревья решений, логарифмическая, степенная и т.д.)) моделей.

Основная цель - минимизация отклонения между наблюдаемыми и предсказанными значениями.

Линейные модели проще, но менее гибкие. Нелинейные модели подходят для сложных зависимостей, но риск переобучения в этом случае повышается. Соответственно линейные модели легче интерпретировать, поскольку параметры напрямую показывают влияние предикторов. Нелинейные модели требуют более сложных методов оценки параметров и больше вычислительных ресурсов.

Методы оценки качества аппроксимации

- R^2 (коэффициент детерминации):

Показывает долю объясненной вариации в данных. Чем ближе R^2 к 1, тем лучше модель описывает данные.

Обычно его определяют таким образом для случайной величины y от факторов x :

$$R^2 = 1 - \frac{D[y|x]}{D[y]} = 1 - \frac{\sigma^2}{\sigma_y^2},$$

где $D[y] = \sigma_y^2$ - дисперсия случайной величины y ,

$D[y|x] = \sigma^2$ - условная (по факторам x) дисперсия зависимой переменной (дисперсия ошибки модели).

- Среднеквадратичная ошибка (MSE):

Среднее значение квадратов отклонений между предсказанными и реальными значениями.

- Средняя абсолютная ошибка (MAE):

Среднее абсолютное значение ошибок. Удобна для интерпретации в тех же единицах, что и данные.

- Графики остатков:

Используются для проверки предположений модели, таких как нормальность и равномерность распределения ошибок.

Стоит также отметить, что линейная регрессия допускает следующие предположения:

1. **Линейная зависимость** между целевой переменной y и предиктором или предикторами X ;

Проверка: построение графиков зависимости остатков против предсказанных значений.

2. Нормальность остатков: остатки модели должны быть нормально распределенными с нулевым средним значением. Это нужно для корректного использования критериев значимости (F-тесты, t-тесты).

Проверяется с помощью графика квантилей (QQ-plot).

3. Гомоскедастичность (равномерность дисперсии остатков): дисперсия остатков одинакова для всех значений предикторов.

Это нужно для того, чтобы избежать смещения стандартных ошибок коэффициентов.

Это проверяется с помощью графика остатков против предсказанных значений.

4. Независимость остатков: коэффициент корреляции остатков равен нулю. Это нужно, чтобы избежать ложных результатов из-за автокорреляции.

5. Отсутствие мультиколлинеарности: независимые переменные X не должны быть сильно коррелированы друг с другом, чтобы получить стабильные и интерпретируемые коэффициенты регрессии.

6. Независимость наблюдений.

Нарушение хотя бы одного допущения может дать некорректный результат, поэтому рекомендуется предварительно проверить все допущения, прежде чем делать вывод о качестве модели. Если допущения не выполнены, значит остается либо выбрать другую модель, либо использовать определенные методы для коррекций данных, чтобы допущения регрессии выполнялись.

Воспользуемся **IV датасетом** (см. пункт 2.1) про успеваемость учеников в двух португальских школах, содержащий оценки учащихся, их демографические, социальные и другие, относящиеся к учебе признаки.

В начале построим линейную модель регрессии и не забудем учесть ее допущения.

Python:

```
Data columns (total 6 columns):
 #   Column           Non-Null Count   Dtype  
 ---  --  
 0   Hours Studied    10000 non-null    int64  
 1   Previous Scores  10000 non-null    int64  
 2   Extracurricular Activities 10000 non-null    object  
 3   Sleep Hours      10000 non-null    int64  
 4   Sample Question Papers Practiced 10000 non-null    int64  
 5   Performance Index 10000 non-null    float64 
 dtypes: float64(1), int64(4), object(1)
 memory usage: 468.9+ KB
```

R:

```
$ Hours.Studied          : int  7 4 8 5 7 3 7 8 5 4 ...
$ Previous.Scores        : int  99 82 51 52 75 78 73 45 77 89 ...
$ Extracurricular.Activities : chr "Yes" "No" "Yes" "Yes" ...
$ Sleep.Hours            : int  9 4 7 5 8 9 5 4 8 4 ...
$ Sample.Question.Papers.Practiced: int  1 2 2 2 5 6 6 6 2 0 ...
$ Performance.Index       : num  91 65 45 36 66 61 63 42 61 69 ...
```

Тест 1. Подгоним простую модель линейной регрессии, используя в качестве целевой переменной: **Performance Index**(оценка успеваемости студента) и предиктора **Hours Studied**(часы, затраченные на учебу).

Python:

OLS Regression Results			
Dep. Variable:	Performance Index	R-squared:	0.140
Model:	OLS	Adj. R-squared:	0.140
Method:	Least Squares	F-statistic:	1623.
Date:	Mon, 25 Nov 2024	Prob (F-statistic):	0.00
Time:	12:09:15	Log-Likelihood:	-42992.
No. Observations:	10000	AIC:	8.599e+04
Df Residuals:	9998	BIC:	8.600e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	41.3792	0.387	106.890	0.000	40.620	42.138
Hours Studied	2.7731	0.069	40.289	0.000	2.638	2.908
Omnibus:	6751.810	Durbin-Watson:		1.996		
Prob(Omnibus):	0.000	Jarque-Bera (JB):		570.766		
Skew:	0.002		Prob(JB):	1.15e-124		
Kurtosis:	1.830		Cond. No.		12.5	

R:

Call:

```
lm(formula = Performance.Index ~ Hours.Studied, data = student_df)
```

Residuals:

Min	1Q	Median	3Q	Max
-37.564	-15.244	-0.152	15.529	35.756

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	41.37917	0.38712	106.89	<2e-16 ***
Hours.Studied	2.77306	0.06883	40.29	<2e-16 ***

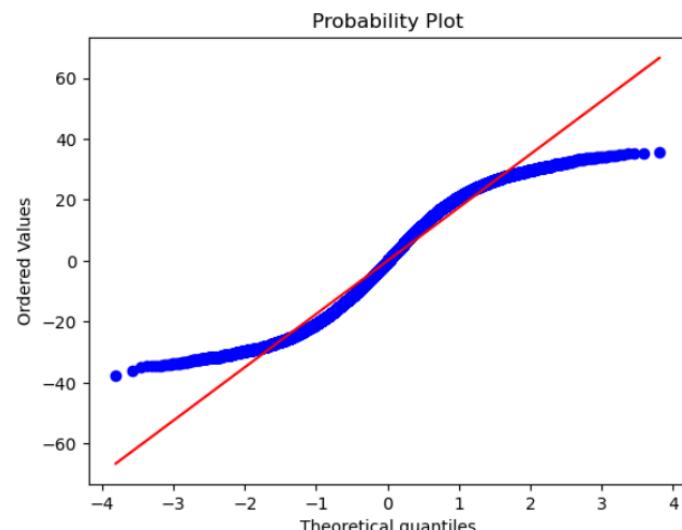
Signif. codes:	0 ****	0.001 **	0.01 *	0.05 .
	0.1	'	'	1

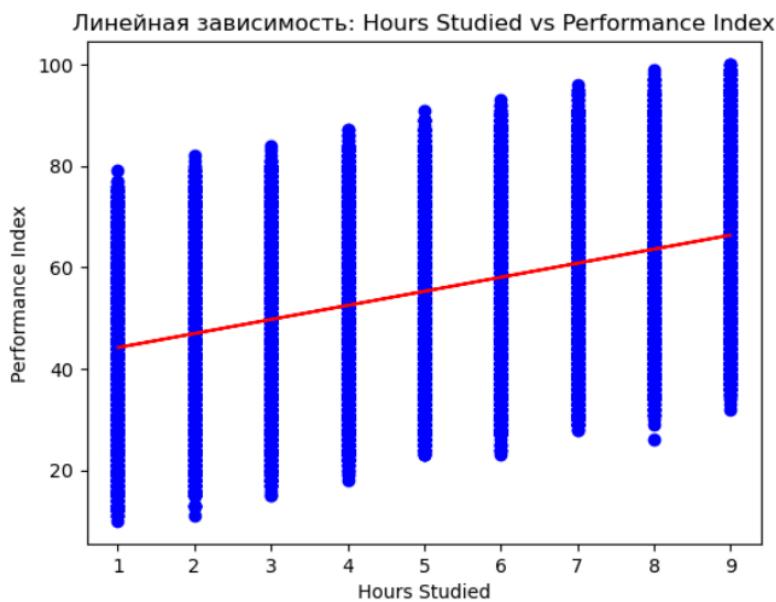
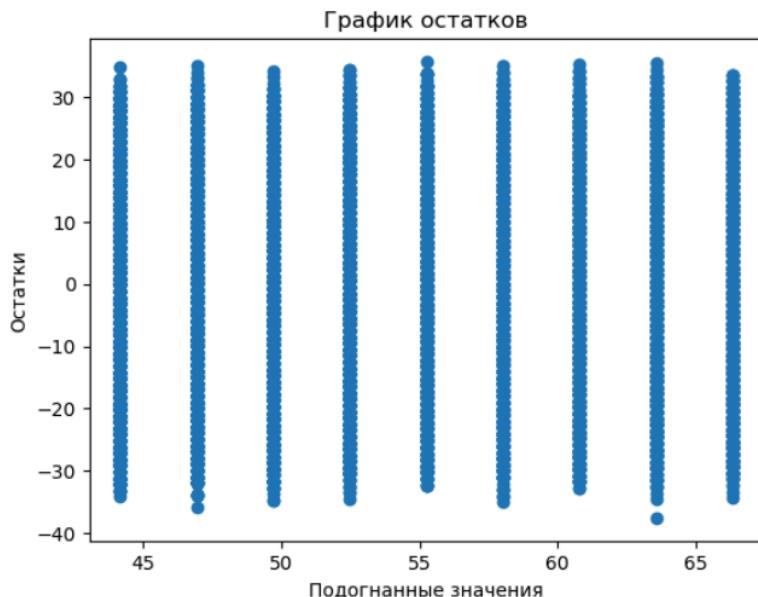
Residual standard error: 17.82 on 9998 degrees of freedom

Multiple R-squared: 0.1397, Adjusted R-squared: 0.1396

F-statistic: 1623 on 1 and 9998 DF, p-value: < 2.2e-16

Проверяем допущения линейной регрессии:





QQ-график показывает отклонение распределения остатков от нормального распределения на хвостах, в то время как в центре графика точки лежат близки к красной линии, что указывает на нормальность остатков в этой части.

Также график остатков явно сигнализирует о том, что у нас распределение очень похоже на равномерное, что указывает на возможное упущение важных предикторов и нелинейную зависимость между целевой переменной и предиктором.

Соответственно, линейная модель регрессии может недостаточно хорошо описывать зависимость.

Mean Squared Error: 317.5336833078034

R-squared: 0.1396743750333912

Среднеквадратичная ошибка предсказания, равная 317,5 означает, что модель плохо справляется с точным прогнозом.

Коэффициент детерминации R-squared (около 14%) означает, что модель объясняет только, 14% вариации целевой переменной, это низкое значение говорит о том, что зависимость между целевой переменной и предиктором слабая и модель не описывает данные достаточно хорошо.

Наблюдается **положительная корреляция** между **часами учебы и индексом успеваемости**, что подтверждается красной линией. Но также вариативность данных(точки вертикально расположены) указывает, что **другие переменные тоже значительно влияют** на целевую переменную. Линия регрессии плохо описывает вариации целевой переменной, так как данные сильно сосредоточены вокруг нее. Соответственно, можно сделать вывод о том, что **линейная зависимость одного предиктора недостаточна** для точного описания данных.

```
# Проверка p-values  
model.pvalues
```

```
const          0.0  
Hours Studied 0.0  
dtype: float64
```

P_value <0.05, что говорит о том, что между индексом успеваемости и часами, потраченными на учебу существует статистически значимая связь, но по результатам подгонки выяснилось, что она слабая и незначительная и объясняет только 14% вариации, то есть этой связи может быть недостаточно для точного прогнозирования в дальнейшем.

R:

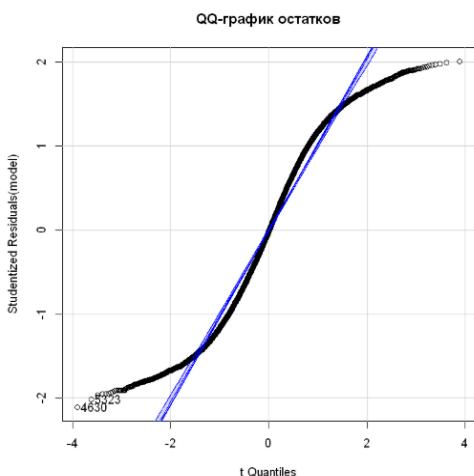
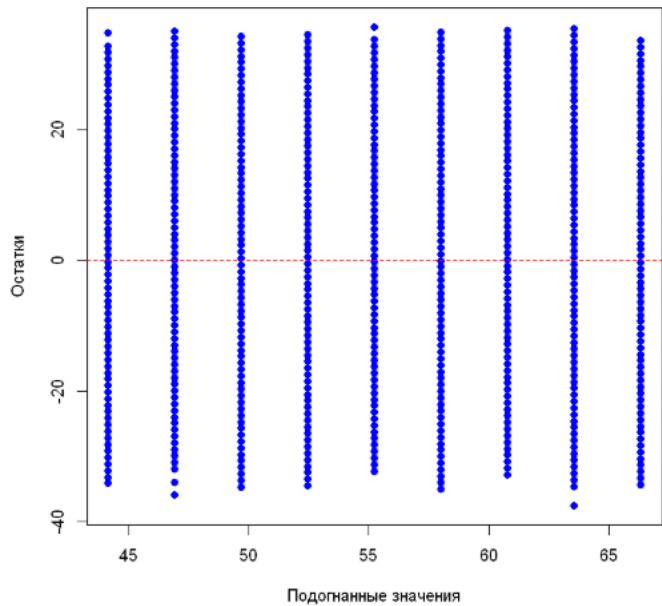
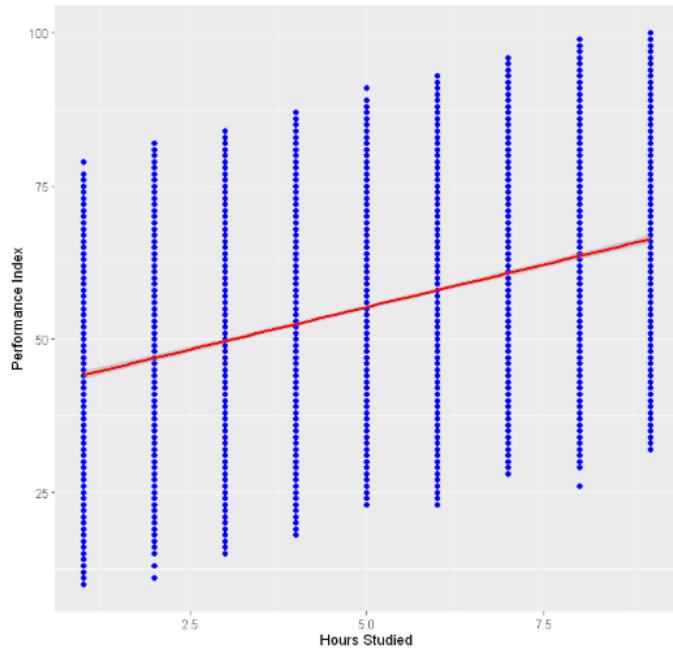


График остатков vs Подогнанные значения



Линейная зависимость: Hours Studied vs Performance Index

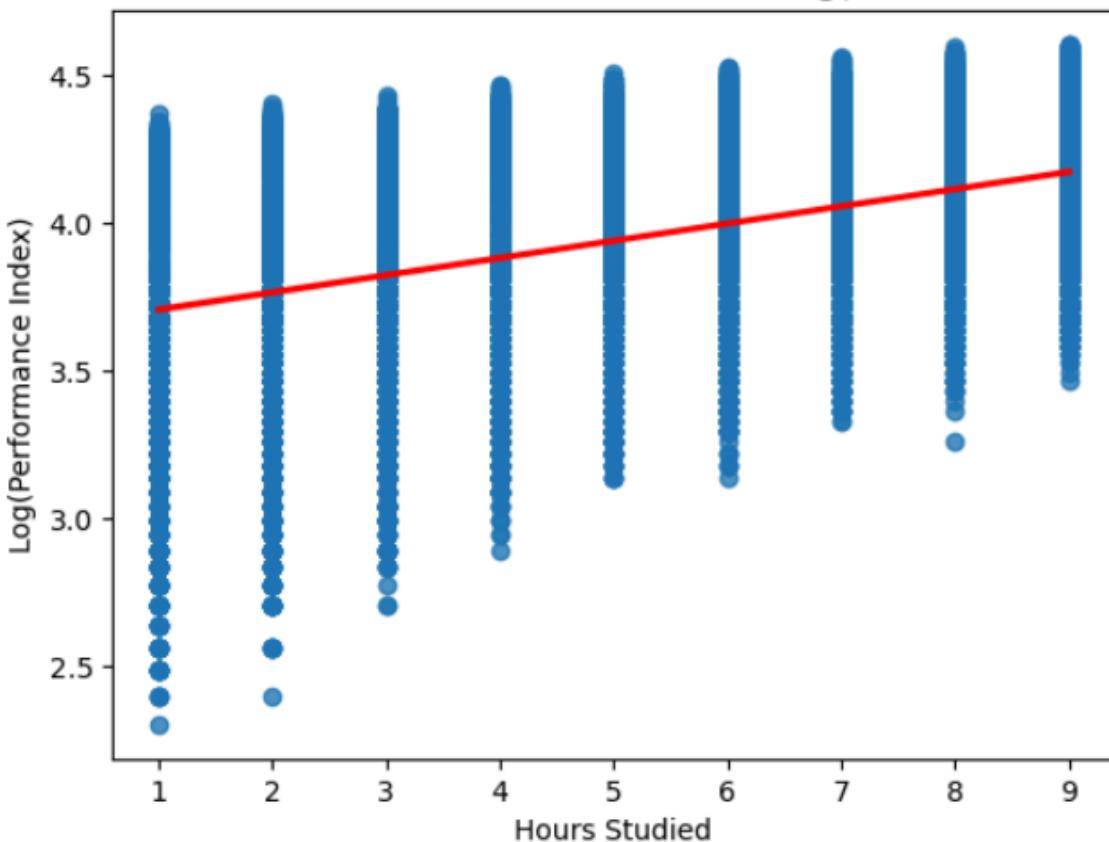


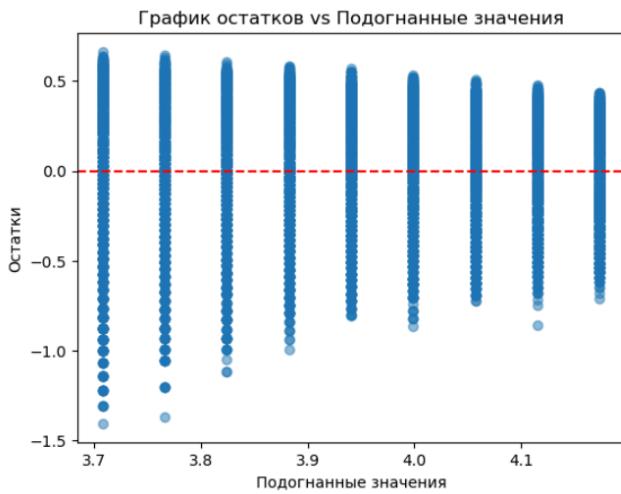
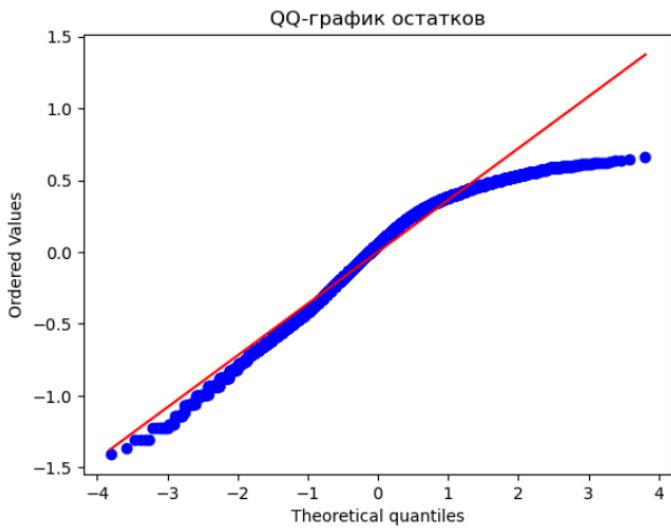
Дополнительно: хвосты на концах распределения остатков могут говорить о выбросах на концах распределения, можно попробовать применить логарифмическое преобразование, чтобы убрать их, посмотрим как поведет себя модель при этом.

Python:

```
OLS Regression Results
=====
Dep. Variable: Log_Performance_Index R-squared:          0.144
Model:                 OLS   Adj. R-squared:        0.144
Method:                Least Squares F-statistic:      1687.
Date: Mon, 25 Nov 2024 Prob (F-statistic):    0.00
Time: 12:09:20 Log-Likelihood:     -4183.4
No. Observations: 10000   AIC:             8371.
Df Residuals:    9998   BIC:            8385.
Df Model:           1
Covariance Type: nonrobust
=====
              coef    std err      t      P>|t|      [ 0.025   0.975]
-----
const      3.6491    0.008  456.867    0.000     3.633    3.665
Hours Studied  0.0583    0.001   41.071    0.000     0.056    0.061
-----
Omnibus:            558.168 Durbin-Watson:       1.994
Prob(Omnibus):      0.000  Jarque-Bera (JB):  595.455
Skew:               -0.567  Prob(JB):        4.99e-130
Kurtosis:            2.624  Cond. No.         12.5
=====
```

Линейная зависимость: Hours Studied vs Log(Performance Index)





Mean Squared Error: 0.13517359257340703

R-squared: 0.14435900929158885

В результате преобразования, модель стала лучше в плане минимизации ошибок, оно уменьшило влияние выбросов, что позволило модели лучше предсказывать данные, о чем свидетельствует низкое значений среднеквадратичной ошибки.

В то время, как R-squared возрос незначительно, что продолжает говорить о слабой зависимости между переменными. Один хвост убрался, но второй в верхней части распределения говорит о том, что присутствуют выбросы с которыми модель не может справиться.

R:

Call:
lm(formula = Log_Performance_Index ~ Hours.Studied, data = student_df)

Residuals:

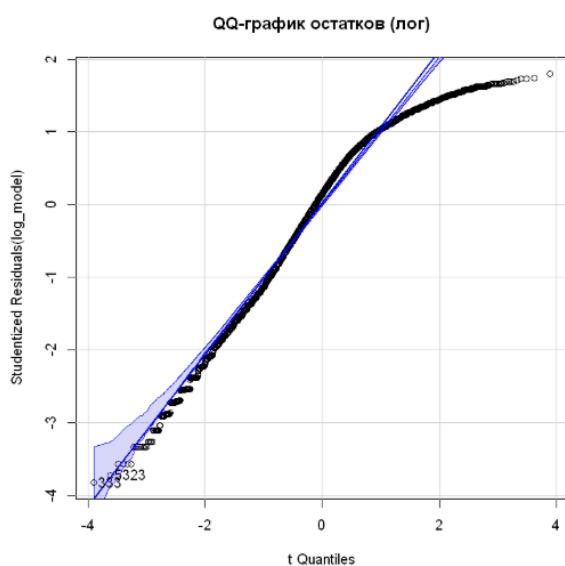
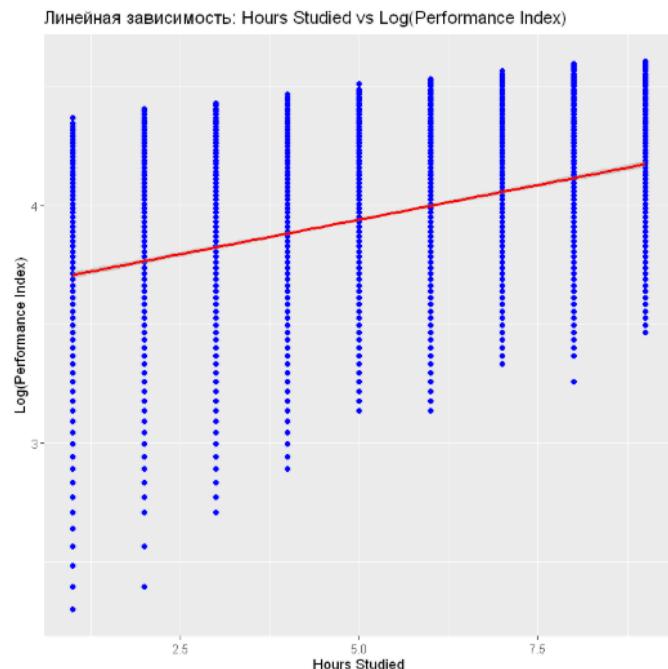
Min	1Q	Median	3Q	Max
-1.40485	-0.26202	0.05349	0.30724	0.66201

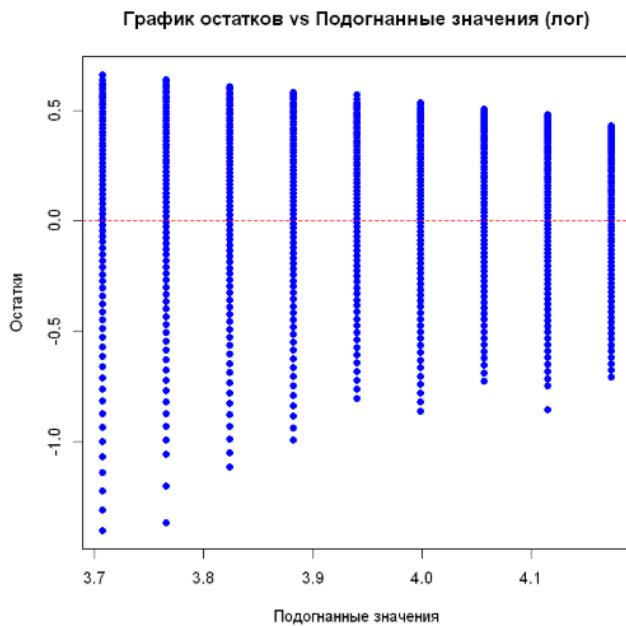
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.649108	0.007987	456.87	<2e-16 ***
Hours.Studied	0.058326	0.001420	41.07	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3677 on 9998 degrees of freedom
Multiple R-squared: 0.1444, Adjusted R-squared: 0.1443
F-statistic: 1687 on 1 and 9998 DF, p-value: < 2.2e-16



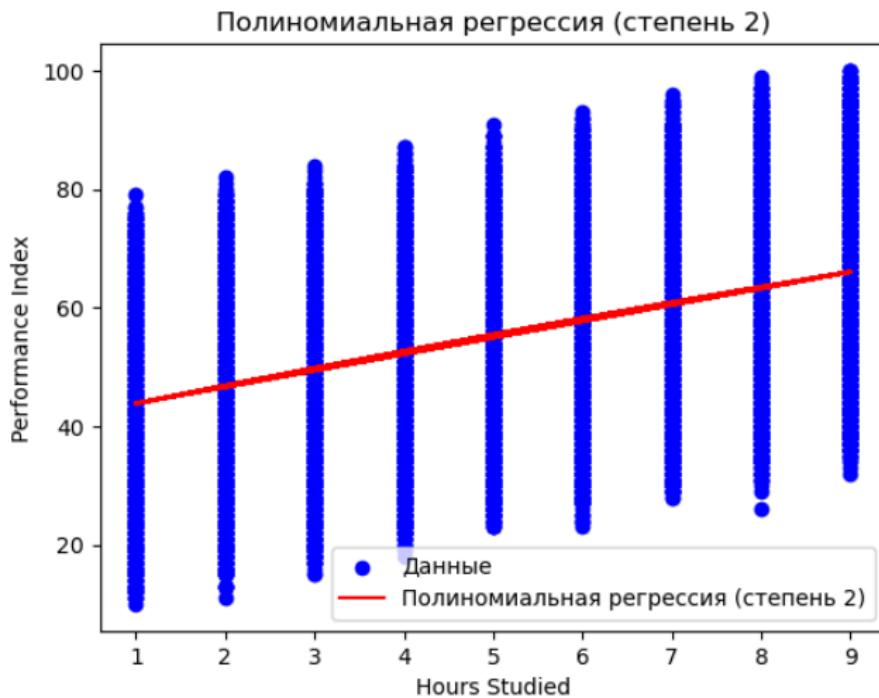


Возьмем теперь полиномиальную регрессию и проверим качество предсказания на ней.

Проверим качество при разных степенях полиномов (2, 4).

Python:

2 степень.



Mean Squared Error (полиномиальная, степень 2): 317.4958
R-squared (полиномиальная, степень 2): 0.1397771

4 степень.

```
Коэффициенты модели: [[-7.20069075e-07 7.71012986e-05 2.80103544e+01 -1.73544469e+01  
6.03548282e-07 2.80103559e+01 -1.73544446e+01 -1.73544441e+01  
1.40260903e+01 -3.88510627e+00 0.00000000e+00 2.80103564e+01  
-1.73544451e+01 -1.73544451e+01 1.40260895e+01 -3.88510627e+00  
1.40260895e+01 -3.88510618e+00 1.54158301e+00 -1.79974844e-01  
0.00000000e+00 2.80103564e+01 -1.73544451e+01 -1.73544451e+01  
1.40260895e+01 -3.88510621e+00 1.40260895e+01 -3.88510621e+00  
1.54158318e+00 -1.79974966e-01 -3.88510621e+00 1.54158318e+00  
-1.79974733e-01 3.41628687e-02 -9.02229654e-04]]  
Свободный член модели: [2.61803911]
```

Mean Squared Error: 317.34410074886733

R-squared: 0.1401880299370235

R:

2 степень 2

Call:

```
lm(formula = Performance.Index ~ Hours.Studied + Hours_Studied_2,  
    data = student_df)
```

Residuals:

Min	1Q	Median	3Q	Max
-37.485	-15.206	-0.116	15.515	35.533

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)							
(Intercept)	40.77694	0.67359	60.536	<2e-16	***						
Hours.Studied	3.10366	0.31034	10.001	<2e-16	***						
Hours_Studied_2	-0.03314	0.03034	-1.093	0.275							

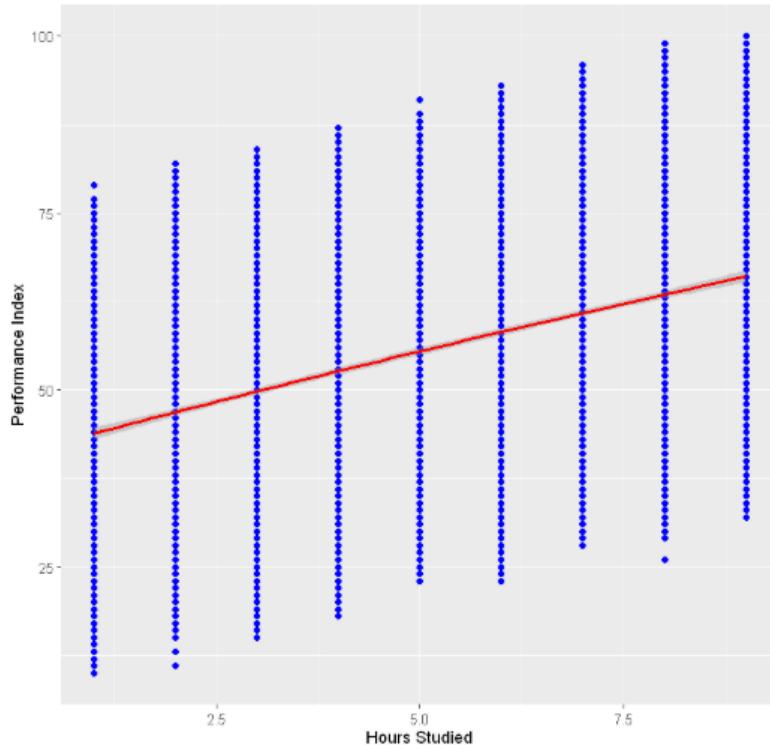
Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	.	0.1	' '	1

Residual standard error: 17.82 on 9997 degrees of freedom

Multiple R-squared: 0.1398, Adjusted R-squared: 0.1396

F-statistic: 812.2 on 2 and 9997 DF, p-value: < 2.2e-16

Полиномиальная регрессия (степень 2)



Mean Squared Error (полиномиальная, степень 2): 317.4958

R-squared (полиномиальная, степень 2): 0.1397771

4 степень.

Call:

```
lm(formula = Performance.Index ~ Hours.Studied + Hours_Studied_2 +
    Hours_Studied_3 + Hours_Studied_4, data = student_df)
```

Residuals:

Min	1Q	Median	3Q	Max
-37.503	-15.314	-0.163	15.497	35.497

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	41.640150	2.048109	20.331	<2e-16 ***	
Hours.Studied	2.134394	2.521877	0.846	0.397	
Hours_Studied_2	0.271172	0.957032	0.283	0.777	
Hours_Studied_3	-0.035525	0.140555	-0.253	0.800	
Hours_Studied_4	0.001373	0.006996	0.196	0.844	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.82 on 9995 degrees of freedom

Multiple R-squared: 0.1398, Adjusted R-squared: 0.1395

F-statistic: 406.1 on 4 and 9995 DF, p-value: < 2.2e-16

Mean Squared Error (полиномиальная, степень 4): 317.4841
R-squared (полиномиальная, степень 4): 0.1398087

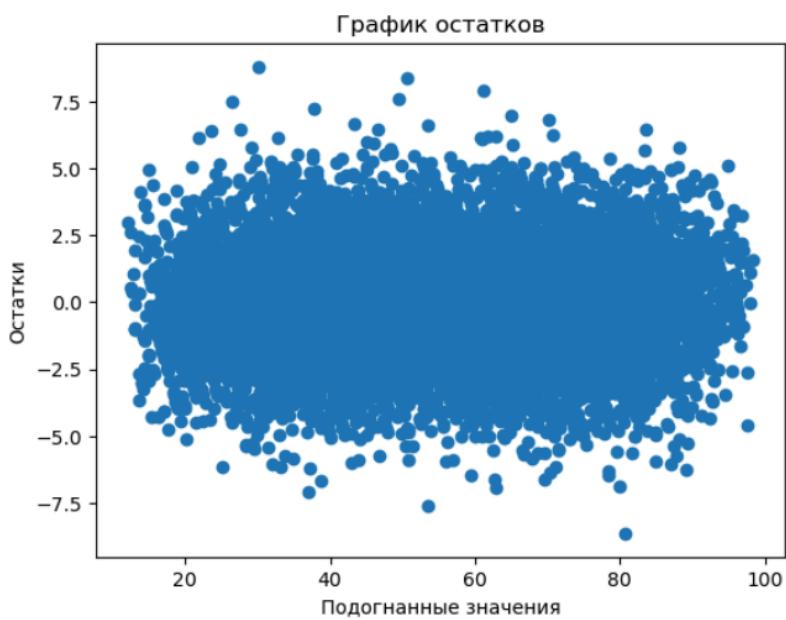
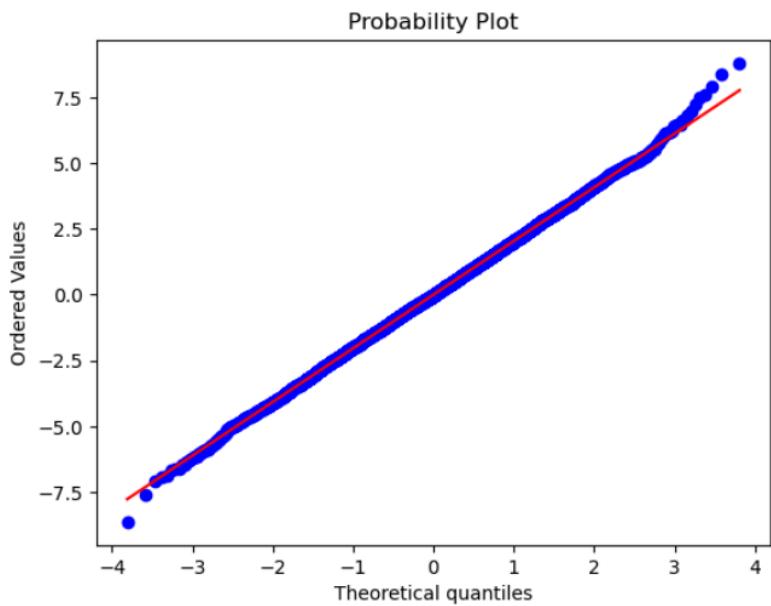
Можно заметить, что качество предсказания разных степеней полиномов мало отличаются. Также показатели полиномиальной регрессии и построенной ранее линейной регрессии не отличаются.

Тест 2. Предыдущая подгонка показала слабую зависимость одного предиктора (часы учебы) на индекс успеваемости. Добавим теперь **несколько предикторов** и подгоним **модель линейной регрессии с несколькими предикторами** и посмотрим на **качество данной аппроксимации**.

Python:

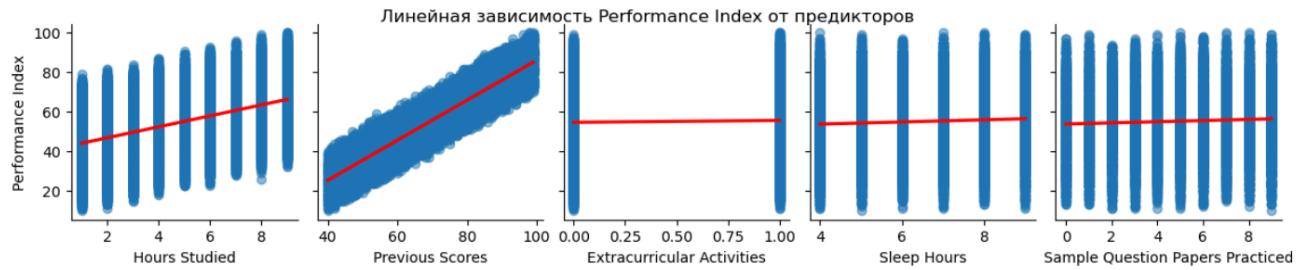
```
# Выбираем независимые переменные и целевую
X = student_df[['Hours Studied', 'Previous Scores', 'Extracurricular Activities', 'Sleep Hours', 'Sample Question Papers Practiced']]
y = student_df['Performance Index']
```

OLS Regression Results													
Dep. Variable: Performance Index		R-squared:		0.989									
Model:		OLS		Adj. R-squared:		0.989							
Method:		Least Squares		F-statistic:		1.757e+05							
Date: Mon, 25 Nov 2024		Prob (F-statistic):		0.00									
Time: 12:09:22		Log-Likelihood:		-21307.									
No. Observations:		10000		AIC:		4.263e+04							
Df Residuals:		9994		BIC:		4.267e+04							
Df Model:		5											
Covariance Type: nonrobust													
	coef	std err	t	P> t	[0.025	0.975]							
const	-34.0756	0.127	-268.010	0.000	-34.325	-33.826							
Hours Studied	2.8530	0.008	362.353	0.000	2.838	2.868							
Previous Scores	1.0184	0.001	866.450	0.000	1.016	1.021							
Extracurricular Activities	0.6129	0.041	15.029	0.000	0.533	0.693							
Sleep Hours	0.4806	0.012	39.972	0.000	0.457	0.504							
Sample Question Papers Practiced	0.1938	0.007	27.257	0.000	0.180	0.208							
Omnibus:	3.851	Durbin-Watson:	2.001										
Prob(Omnibus):	0.146	Jarque-Bera (JB):	4.036										
Skew:	0.013	Prob(JB):	0.133										
Kurtosis:	3.095	Cond. No.	452.										

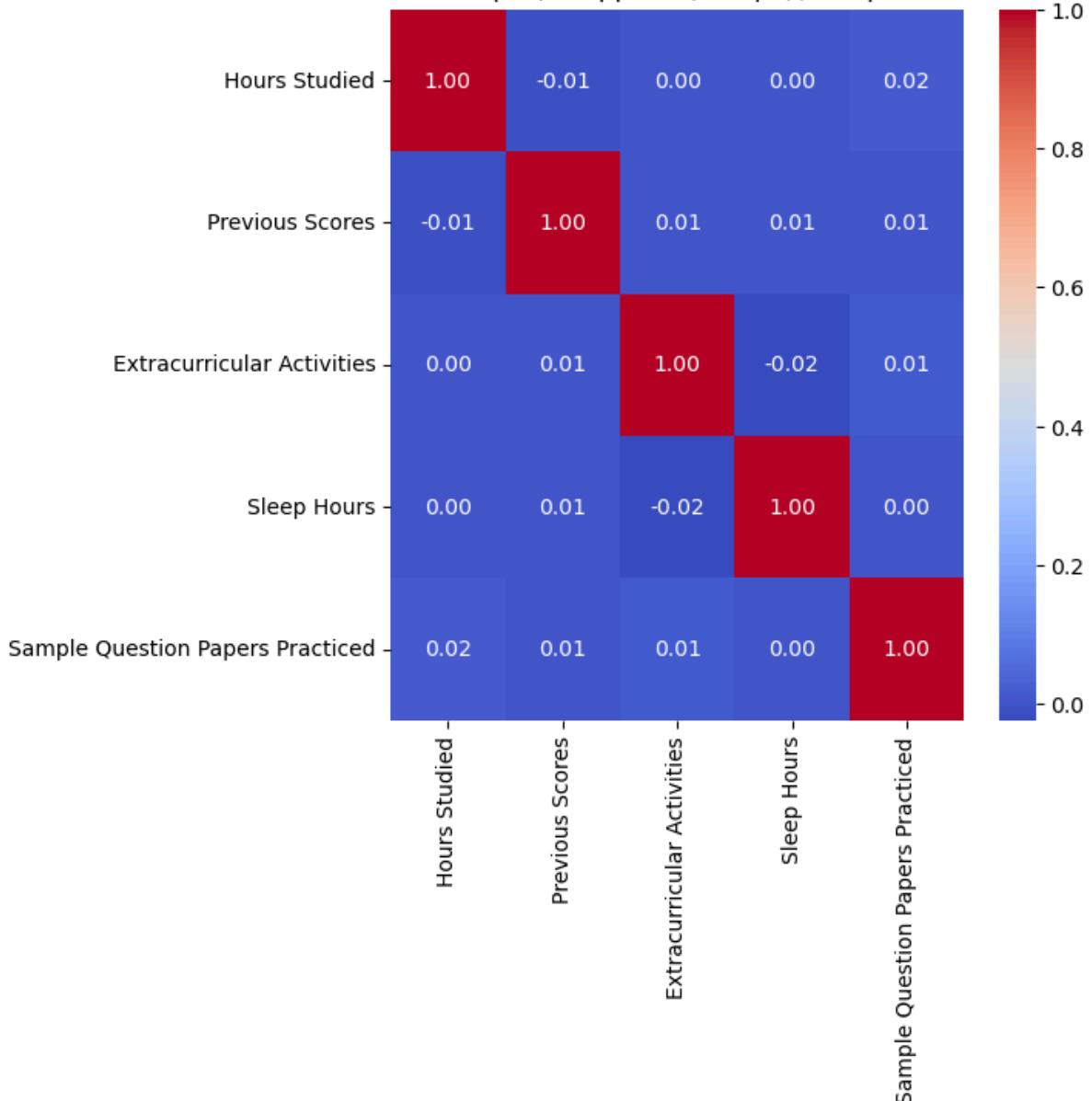


Mean Squared Error: 4.151350633946038

R-squared: 0.9887523323780958



Матрица корреляции предикторов



	Feature	VIF
0	const	38.916475
1	Hours Studied	1.000478
2	Previous Scores	1.000326
3	Extracurricular Activities	1.000802
4	Sleep Hours	1.000600
5	Sample Question Papers Practiced	1.000557

```
# Проверка p-values
model.pvalues
```

```
const                      0.000000e+00
Hours Studied              0.000000e+00
Previous Scores             0.000000e+00
Extracurricular Activities 1.681087e-50
Sleep Hours                 0.000000e+00
Sample Question Papers Practiced 7.461573e-158
dtype: float64
```

По результатам подгонки многомерной линейной регрессии(несколько предикторов) можно сделать вывод о том, что влияние нескольких предикторов одновременно на целевую переменную оказывает существенный эффект, чем их влияние по отдельности, что можно увидеть из каждого отдельных графиков.

Наиболее значимая линейная зависимость от предиктора **Previous Scores** подчеркивает сильную связь между предыдущими результатами учеников с их индексом успеваемости.

Значения p_value продолжают говорить о статистической значимой зависимости целевой переменной от предикторов.

- Коэффициент детерминации R-squared (около 0.99 -> 99%) означает, что модель хорошо описывает и прогнозирует 99% всей вариации целевой переменной, что значительно улучшает качество модели.
- Среднеквадратичная ошибка около 4.15 указывает, что модель прогнозирует с некоторой погрешностью, но это погрешность незначительна(Как в предыдущей подгонке линейной регрессии после логарифмического преобразования подозреваю, что она также снизится, если его применить ко всем предикторам).
- По графикам видно (QQ-график) - распределение остатков совпадает с нормальным распределением, есть незначительные отклонения на концах, но они незначительные. А также случайный разброс остатков подтверждает, что допущение гомоскедастичности не нарушено. Мультиколлинеарность отсутствует, и есть линейная зависимость целевой переменной от предикторов, что означает надежность вывода модели.

R:

Call:
lm(formula = Performance.Index ~ Hours.Studied + Previous.Scores +
Extracurricular.Activities + Sleep.Hours + Sample.Question.Papers.Practiced,
data = student_df)

Residuals:

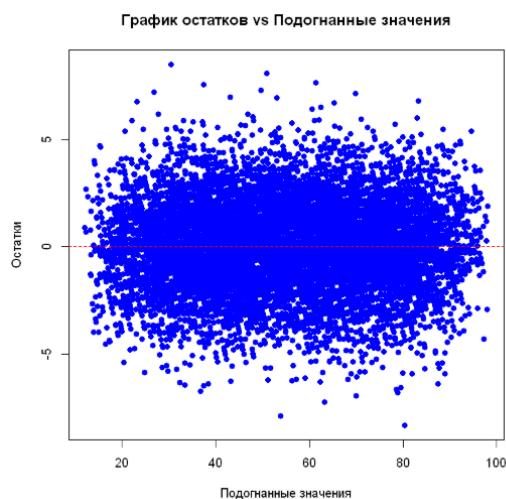
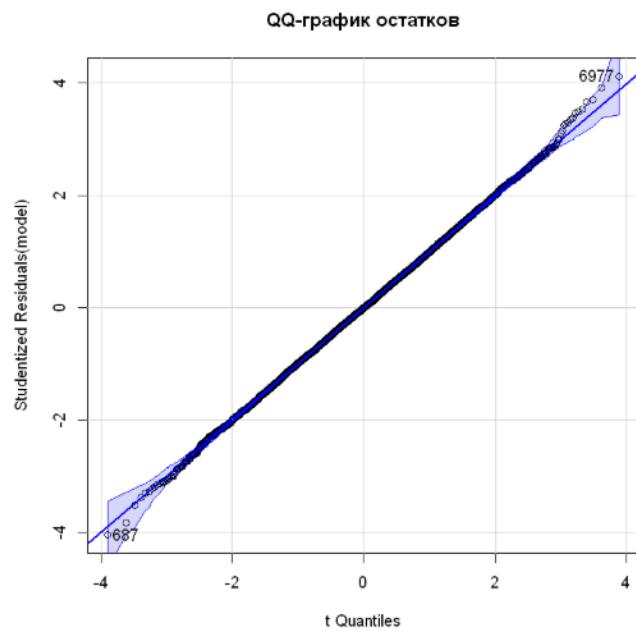
Min	1Q	Median	3Q	Max
-8.3299	-1.3831	-0.0062	1.3701	8.4864

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-33.763726	0.126841	-266.19	<2e-16 ***
Hours.Studied	2.853429	0.007962	358.40	<2e-16 ***
Previous.Scores	1.018584	0.001189	857.02	<2e-16 ***
Extracurricular.Activities	NA	NA	NA	NA
Sleep.Hours	0.476333	0.012153	39.19	<2e-16 ***
Sample.Question.Papers.Practiced	0.195198	0.007189	27.15	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

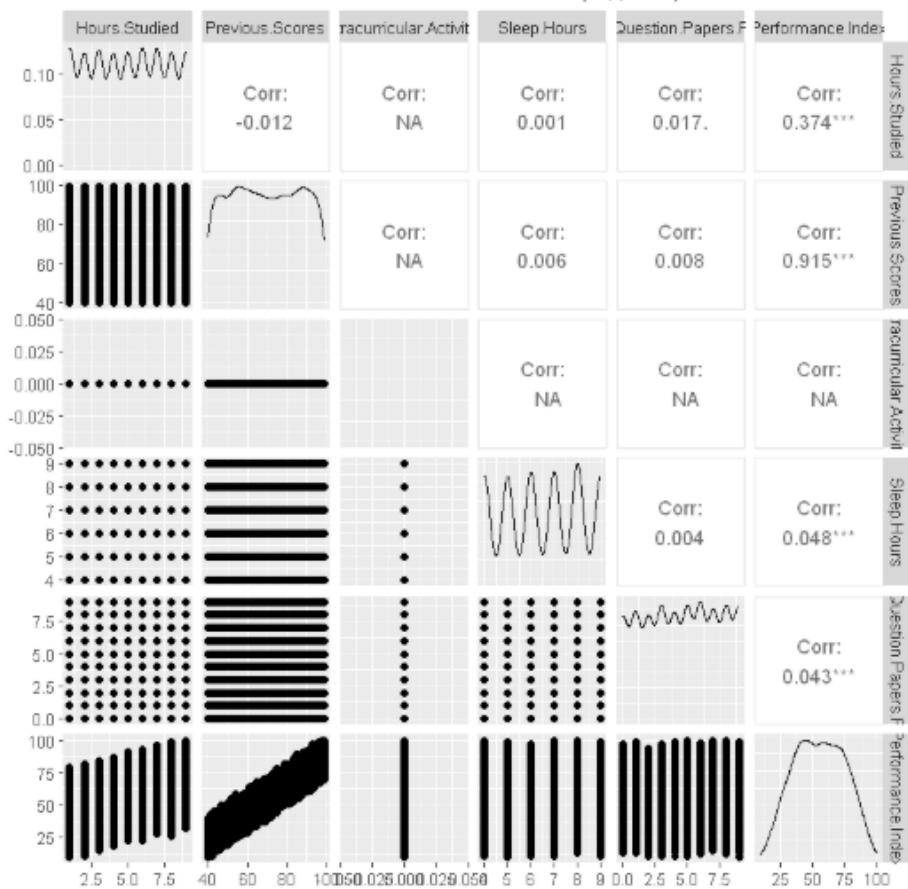
Residual standard error: 2.061 on 9995 degrees of freedom
Multiple R-squared: 0.9885, Adjusted R-squared: 0.9885
F-statistic: 2.147e+05 on 4 and 9995 DF, p-value: < 2.2e-16



Mean Squared Error: 4.245176

R-squared: 0.9884981

Линейная зависимость Performance Index от предикторов



	Feature	VIF
Hours.Studied	Hours.Studied	1.000464
Previous.Scores	Previous.Scores	1.000254
Sleep.Hours	Sleep.Hours	1.000052
Sample.Question.Papers.Practiced	Sample.Question.Papers.Practiced	1.000386

Тест 3. Подгоним линейную регрессионную модель, используя предиктор **Previous Scores** к целевой переменной **Performance Index**.

Python:

```
# Выбираем предиктор и целевую переменную
X = student_df[['Previous Scores']]
y = student_df[['Performance Index']]
```

R:

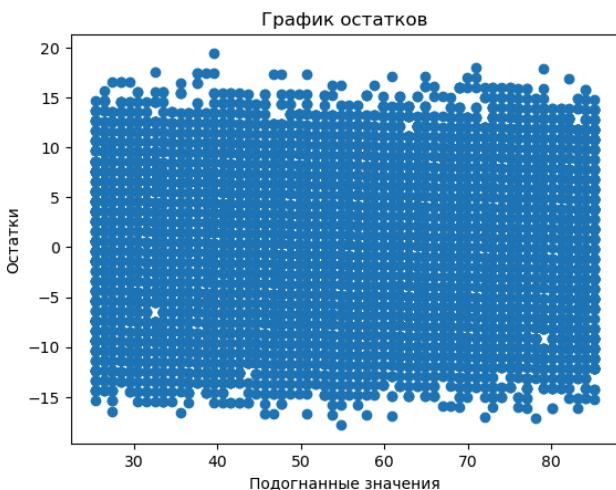
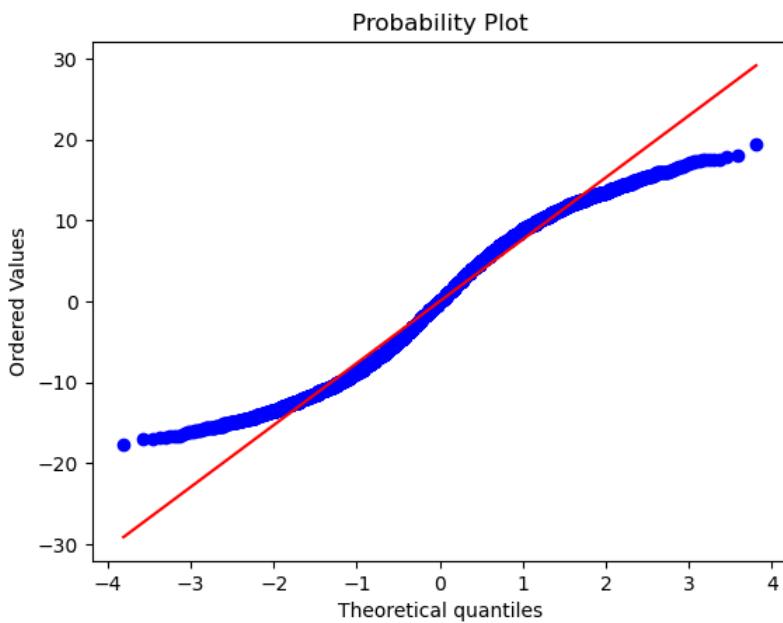
```
# Выбираем предиктор и целевую переменную
X <- student_df[, "Previous.Scores", drop = FALSE]
y <- student_df$Performance.Index
```

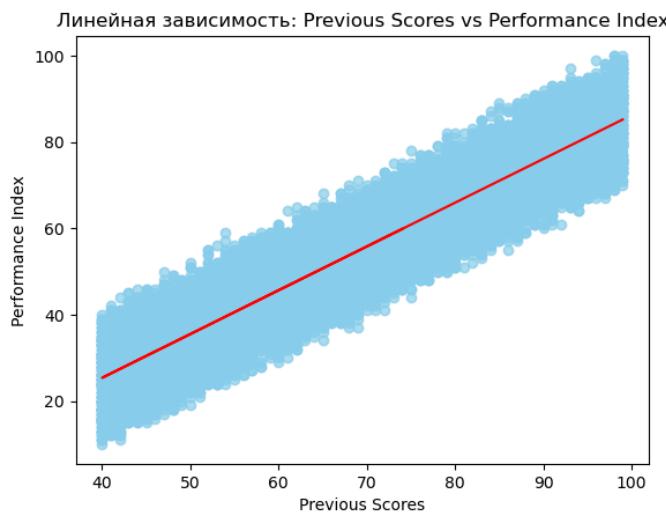
Python:

```
OLS Regression Results
Dep. Variable: Performance Index      R-squared:      0.838
Model: OLS                            Adj. R-squared:  0.838
Method: Least Squares                F-statistic:   5.156e+04
Date: Mon, 25 Nov 2024              Prob (F-statistic): 0.00
Time: 12:09:35                      Log-Likelihood: -34657.
No. Observations: 10000               AIC: 6.932e+04
Df Residuals: 9998                  BIC: 6.933e+04
Df Model: 1
Covariance Type: nonrobust

            coef  std err      t    P>|t|   [0.025   0.975]
const     -15.1818    0.320   -47.502  0.000  -15.808  -14.555
Previous Scores  1.0138    0.004   227.058  0.000    1.005    1.023

Omnibus: 2510.679  Durbin-Watson: 2.027
Prob(Omnibus): 0.000  Jarque-Bera (JB): 431.609
Skew: 0.011  Prob(JB): 1.89e-94
Kurtosis: 1.982  Cond. No. 295.
```





Mean Squared Error: 59.95012237721492

R-squared: 0.8375711642188021

- По итогу видна линейная зависимость между предиктором и целевой переменной, распределение имеет тяжелые хвосты, что говорит о том, что некоторые остатки являются аномалиями.
- График остаточных значений тоже демонстрирует симметрию остатков, что наталкивает на мысль о равномерном распределении.
- Хороший коэффициент детерминации, но огромная среднеквадратичная ошибка, говорящая о том, что модель допускает серьезные ошибки в прогнозе дают повод понять, что модель допускает серьезные ошибки при прогнозах.

Вывод модели может быть не надежным, но по сравнению с другими предикторами он является наиболее достоверным.

R:

Call:

```
lm(formula = Performance.Index ~ Previous.Scores, data = student_df)
```

Residuals:

Min	1Q	Median	3Q	Max
-17.7729	-6.5239	-0.0082	6.3689	19.4346

Coefficients:

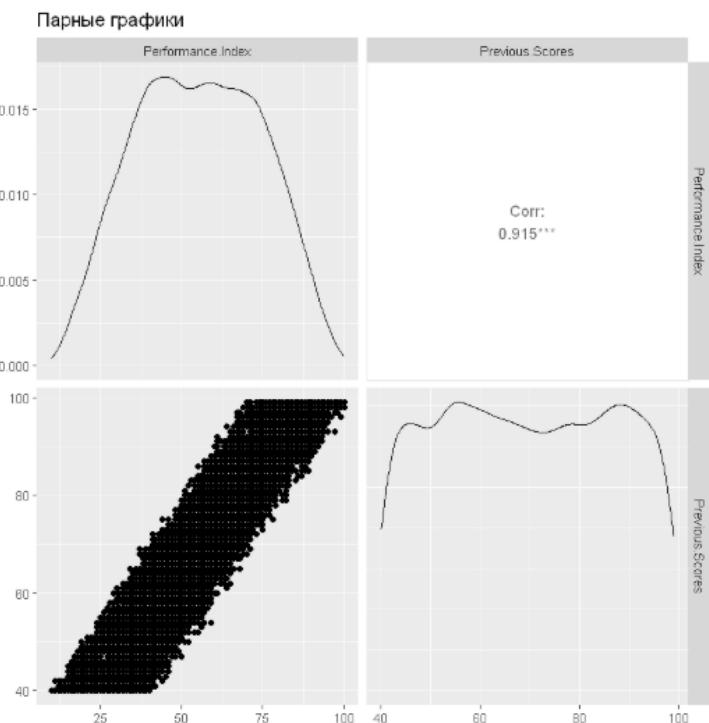
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-15.181799	0.319605	-47.5	<2e-16 ***
Previous.Scores	1.013837	0.004465	227.1	<2e-16 ***

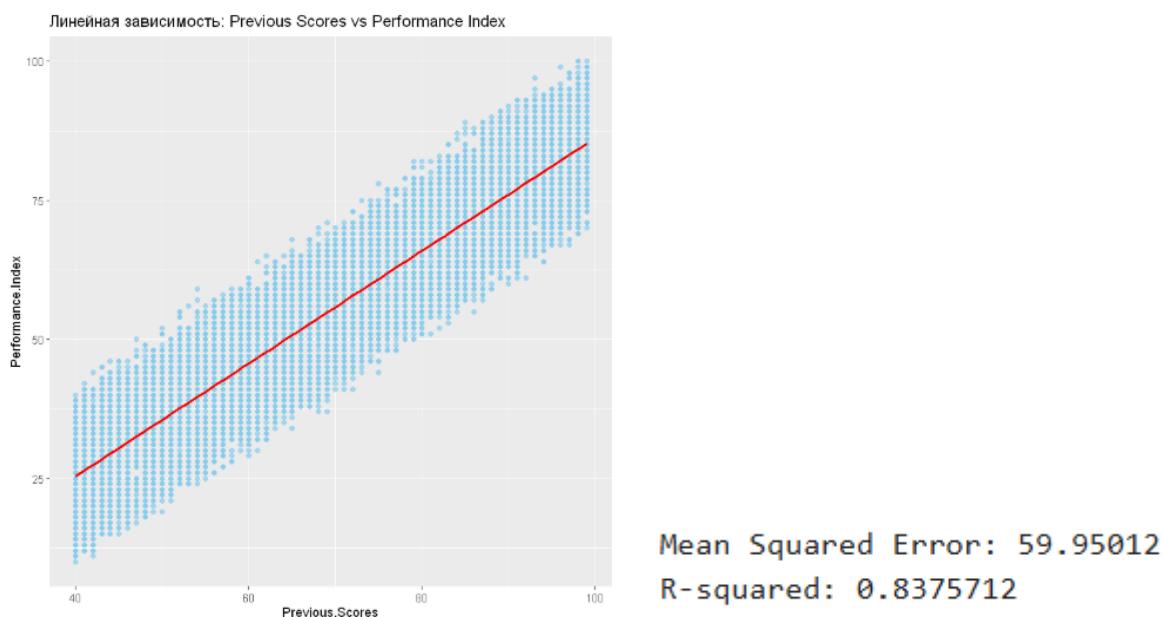
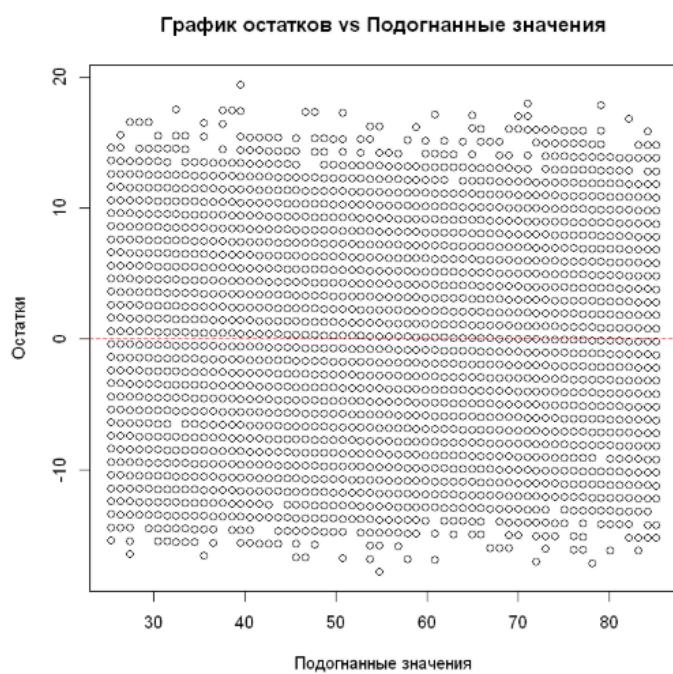
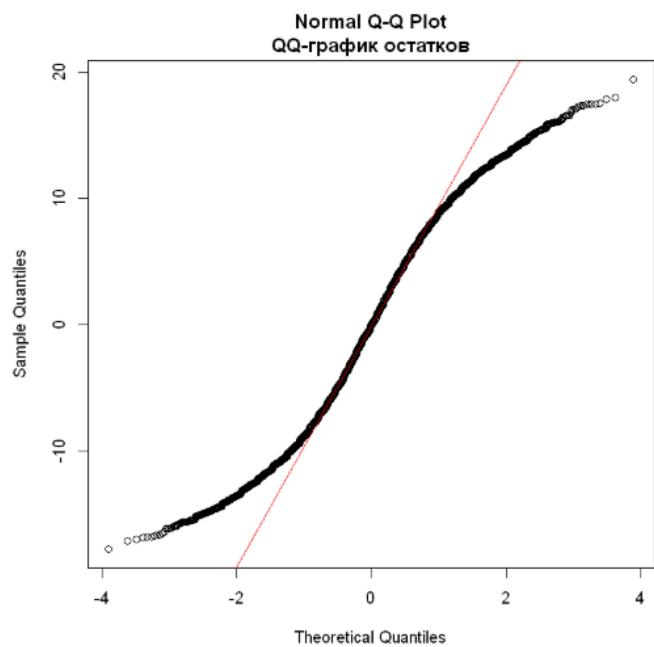
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.744 on 9998 degrees of freedom

Multiple R-squared: 0.8376, Adjusted R-squared: 0.8376

F-statistic: 5.156e+04 on 1 and 9998 DF, p-value: < 2.2e-16

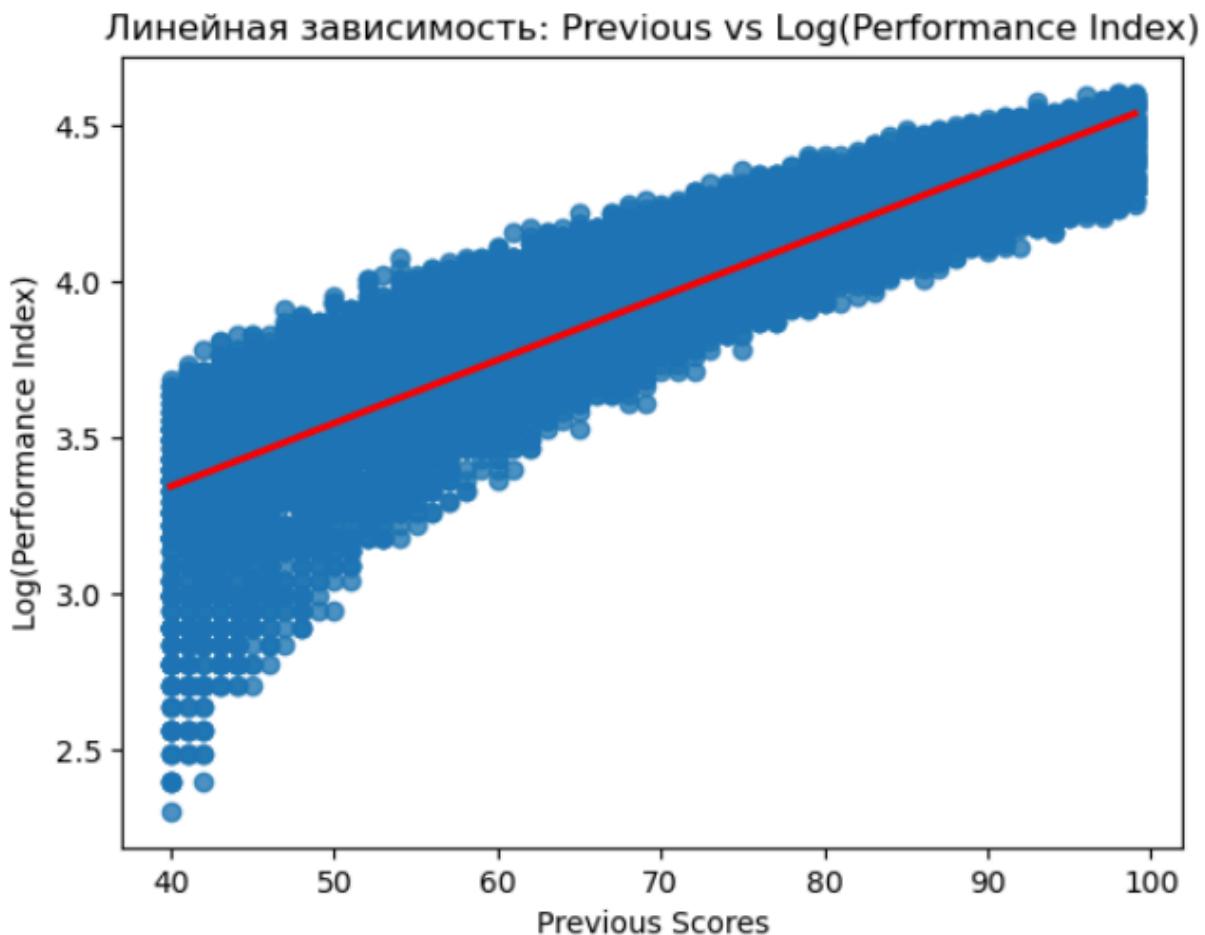


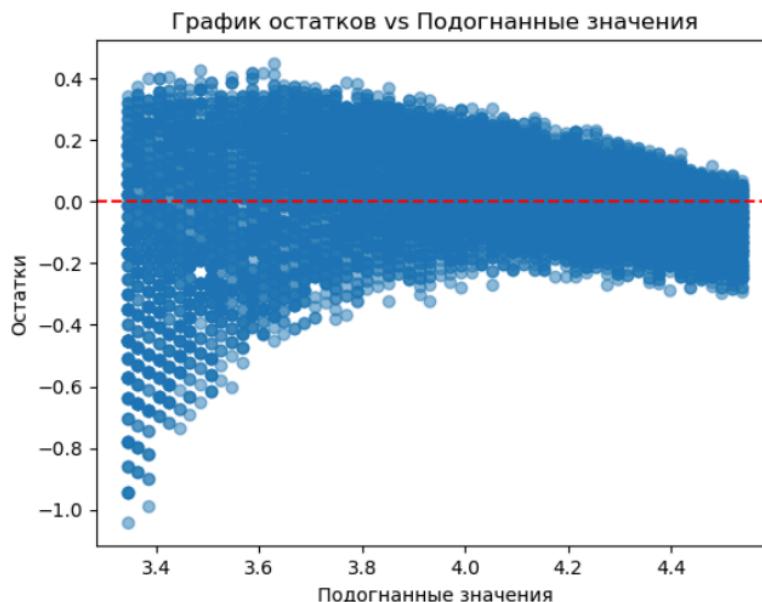
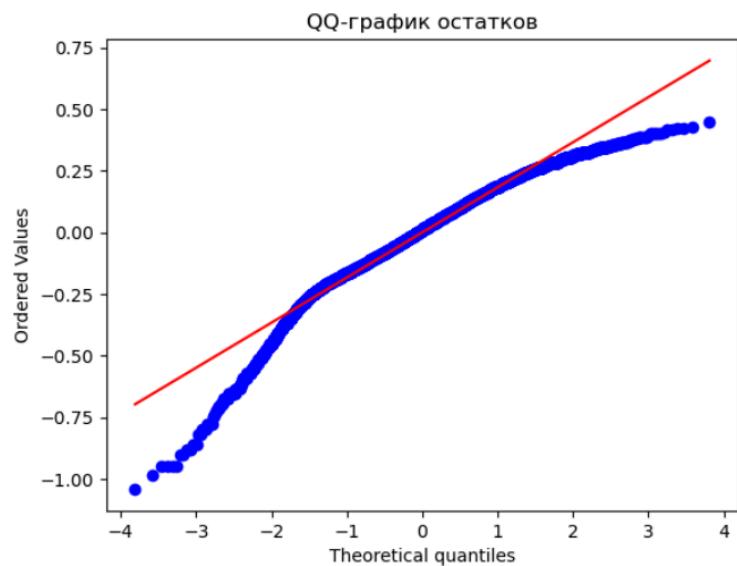


Дополнительно: Логарифмическое преобразование.

Python:

```
OLS Regression Results
=====
Dep. Variable: Log_Performance_Index   R-squared:          0.781
Model:                          OLS   Adj. R-squared:      0.781
Method:                         Least Squares   F-statistic:     3.570e+04
Date: Mon, 25 Nov 2024   Prob (F-statistic):    0.00
Time: 12:09:42           Log-Likelihood:       2635.2
No. Observations:            10000   AIC:             -5266.
Df Residuals:                 9998   BIC:             -5252.
Df Model:                      1
Covariance Type:            nonrobust
=====
            coef    std err        t    P>|t|    [0.025    0.975]
-----
const      2.5336    0.008  330.137    0.000     2.519    2.549
Previous Scores  0.0203    0.000  188.939    0.000     0.020    0.020
=====
Omnibus:           1152.005   Durbin-Watson:      2.018
Prob(Omnibus):      0.000    Jarque-Bera (JB): 2061.617
Skew:              -0.775    Prob(JB):         0.00
Kurtosis:           4.596    Cond. No.        295.
=====
```





Mean Squared Error: 0.03456479366663104

R-squared: 0.7812068634597616

R:

Call:

```
lm(formula = Log_Performance_Index ~ Previous.Scores, data = student_df)
```

Residuals:

Min	1Q	Median	3Q	Max
-1.04125	-0.11149	0.01237	0.13380	0.45010

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.5335558	0.0076742	330.1	<2e-16 ***
Previous.Scores	0.0202571	0.0001072	188.9	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

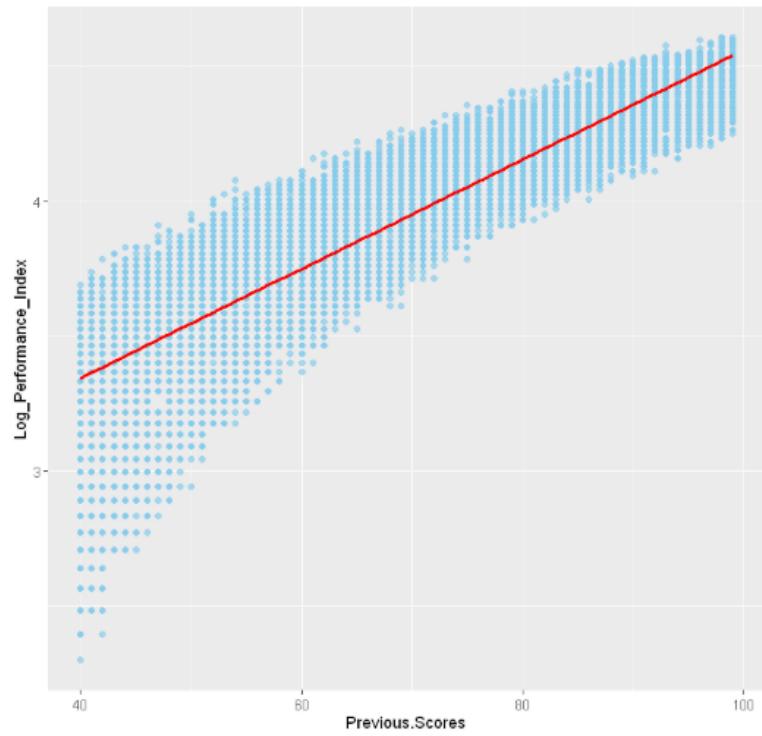
Residual standard error: 0.1859 on 9998 degrees of freedom

Multiple R-squared: 0.7812, Adjusted R-squared: 0.7812

F-statistic: 3.57e+04 on 1 and 9998 DF, p-value: < 2.2e-16

Линейная зависимость: Previous vs Log(Performance Index)

Линейная зависимость: Previous vs Log(Performance Index)



Normal Q-Q Plot
QQ-график остатков (лог-преобразование)

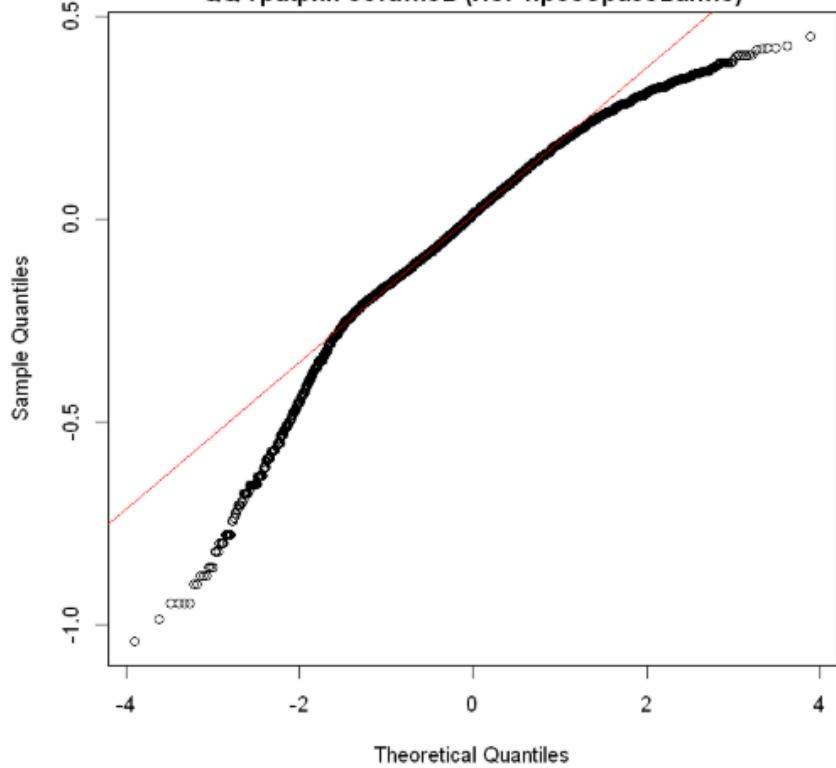
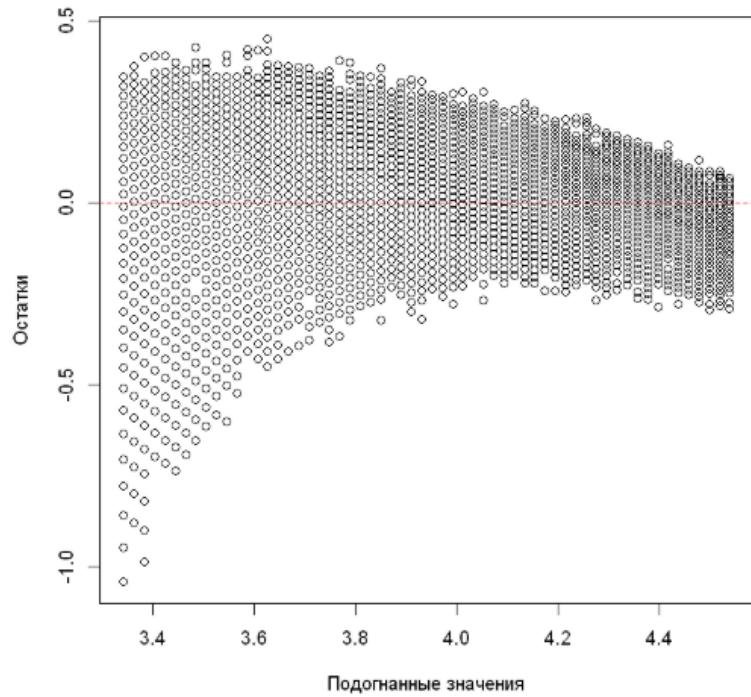


График остатков vs Подогнанные значения



Mean Squared Error (log): 0.03456479

R-squared (log): 0.7812069

Преобразования улучшили вывод модели, нельзя утверждать о гомоскедастичности и нормальности распределения, исходя из графиков, но по итогам всех моделей можно смело утверждать, что вывод данной модели более надежен, чем вывод, связанный с другими предикторами.

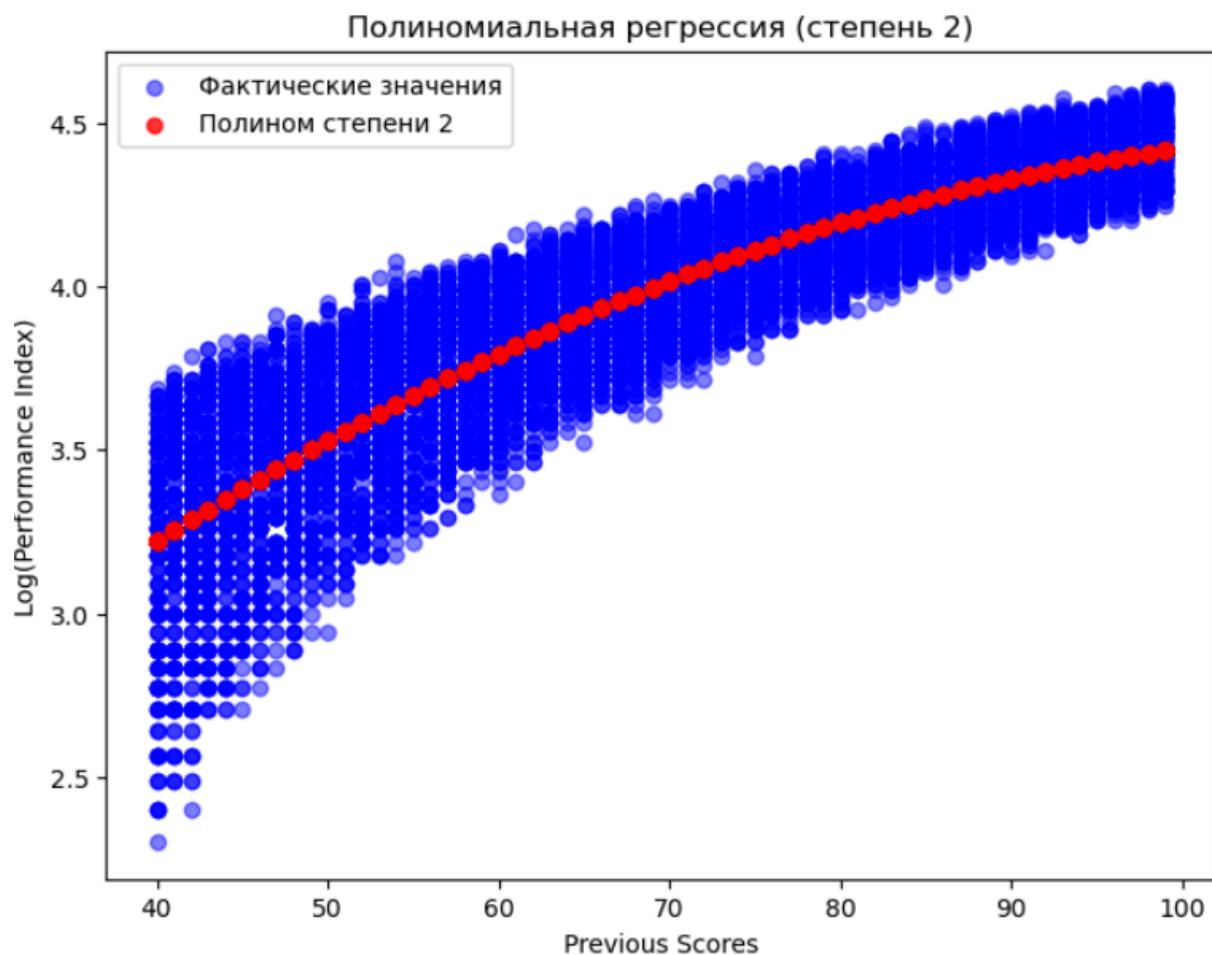
По итогу мы отклоняем нулевую гипотезу об отсутствии влияния предиктора на целевую переменную.

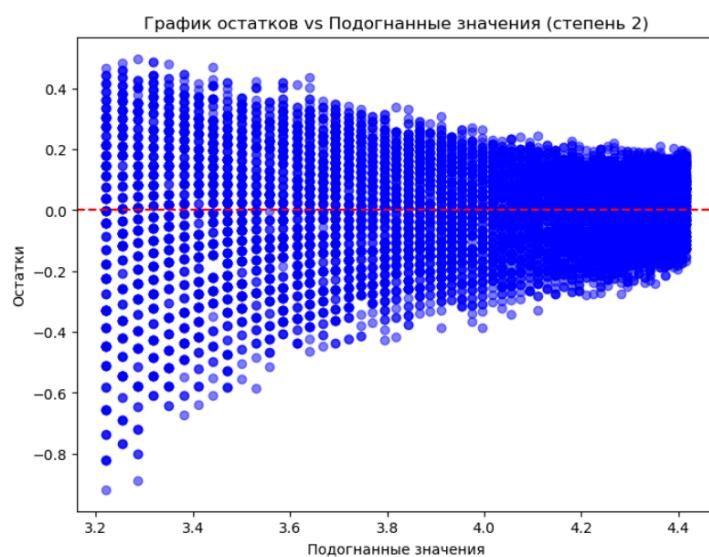
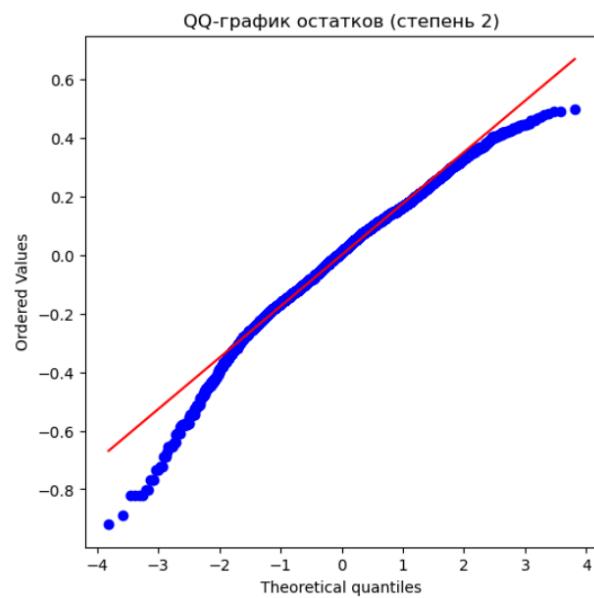
Подгоним теперь полиномиальную регрессию.

Python:

```
Полиномиальная регрессия: степень 2
MSE: 0.031224399569298275
R2-squared: 0.8023513641006408
```

```
OLS Regression Results
=====
Dep. Variable: Log_Performance_Index R-squared: 0.802
Model: OLS Adj. R-squared: 0.802
Method: Least Squares F-statistic: 2.029e+04
Date: Mon, 25 Nov 2024 Prob (F-statistic): 0.00
Time: 12:09:43 Log-Likelihood: 3143.4
No. Observations: 10000 AIC: -6281.
Df Residuals: 9997 BIC: -6259.
Df Model: 2
Covariance Type: nonrobust
=====
            coef    std err          t      P>|t|      [0.025]      [0.975]
-----
const    1.5527    0.031      50.301      0.000      1.492      1.613
x1       0.0504    0.001      54.374      0.000      0.049      0.052
x2      -0.0002   6.62e-06     -32.703      0.000     -0.000     -0.000
=====
Omnibus: 537.154 Durbin-Watson: 2.028
Prob(Omnibus): 0.000 Jarque-Bera (JB): 759.917
Skew: -0.492 Prob(JB): 9.68e-166
Kurtosis: 3.926 Cond. No. 9.91e+04
=====
```





Полиномиальная регрессия: степень 4

MSE: 0.031017684702724257

R2-squared: 0.8036598572009739

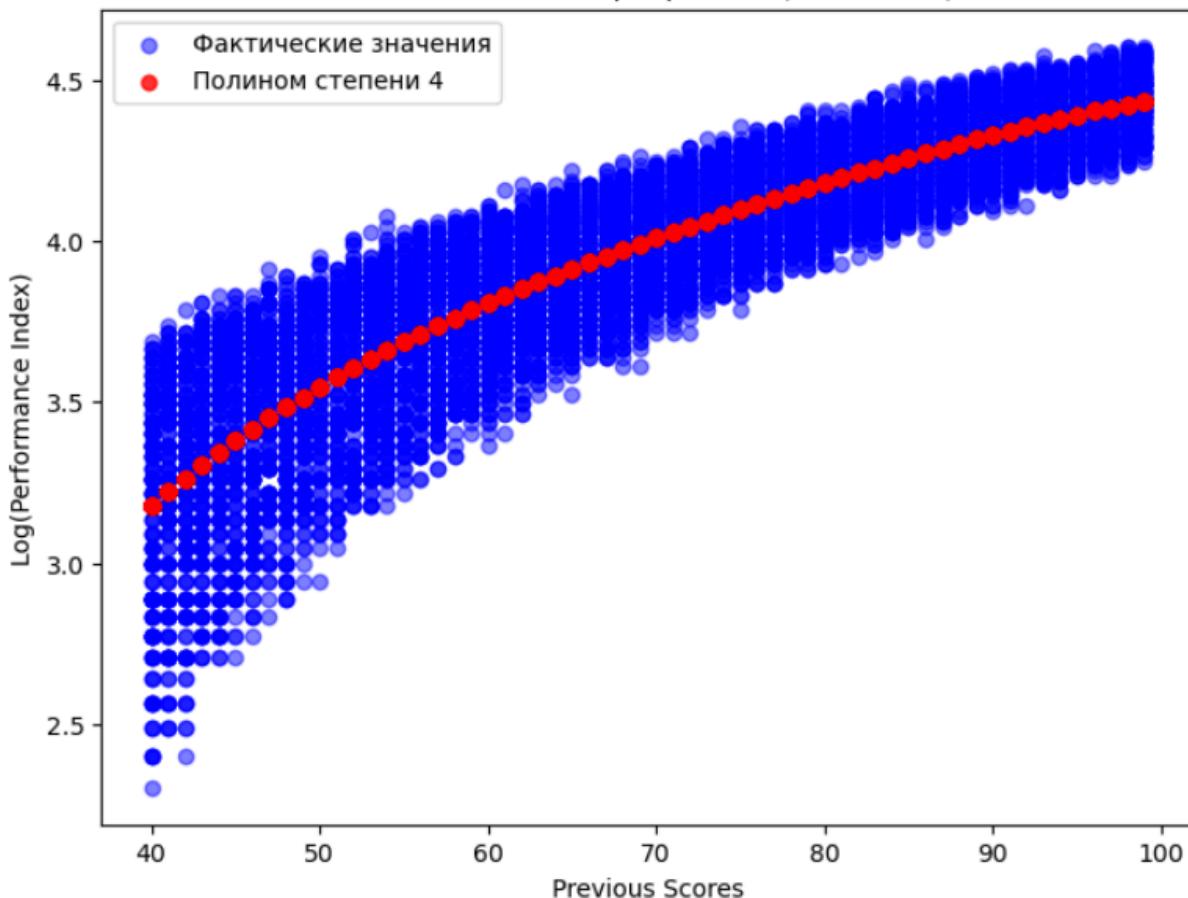
OLS Regression Results

Dep. Variable:	Log_Performance_Index	R-squared:	0.804
Model:	OLS	Adj. R-squared:	0.804
Method:	Least Squares	F-statistic:	1.023e+04
Date:	Mon, 25 Nov 2024	Prob (F-statistic):	0.00
Time:	12:09:44	Log-Likelihood:	3176.6
No. Observations:	10000	AIC:	-6343.
Df Residuals:	9995	BIC:	-6307.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	-1.1693	0.579	-2.020	0.043	-2.304	-0.035
x1	0.2060	0.036	5.732	0.000	0.136	0.276
x2	-0.0034	0.001	-4.196	0.000	-0.005	-0.002
x3	2.812e-05	7.99e-06	3.520	0.000	1.25e-05	4.38e-05
x4	-8.949e-08	2.87e-08	-3.118	0.002	-1.46e-07	-3.32e-08

Omnibus:	455.270	Durbin-Watson:	2.029
Prob(Omnibus):	0.000	Jarque-Bera (JB):	613.109
Skew:	-0.453	Prob(JB):	7.33e-134
Kurtosis:	3.807	Cond. No.	1.38e+10

Полиномиальная регрессия (степень 4)



QQ-график остатков (степень 4)

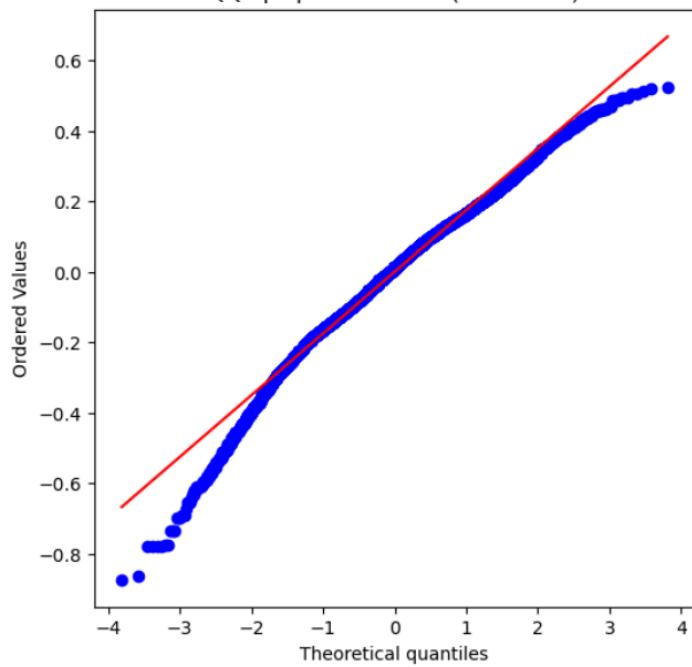
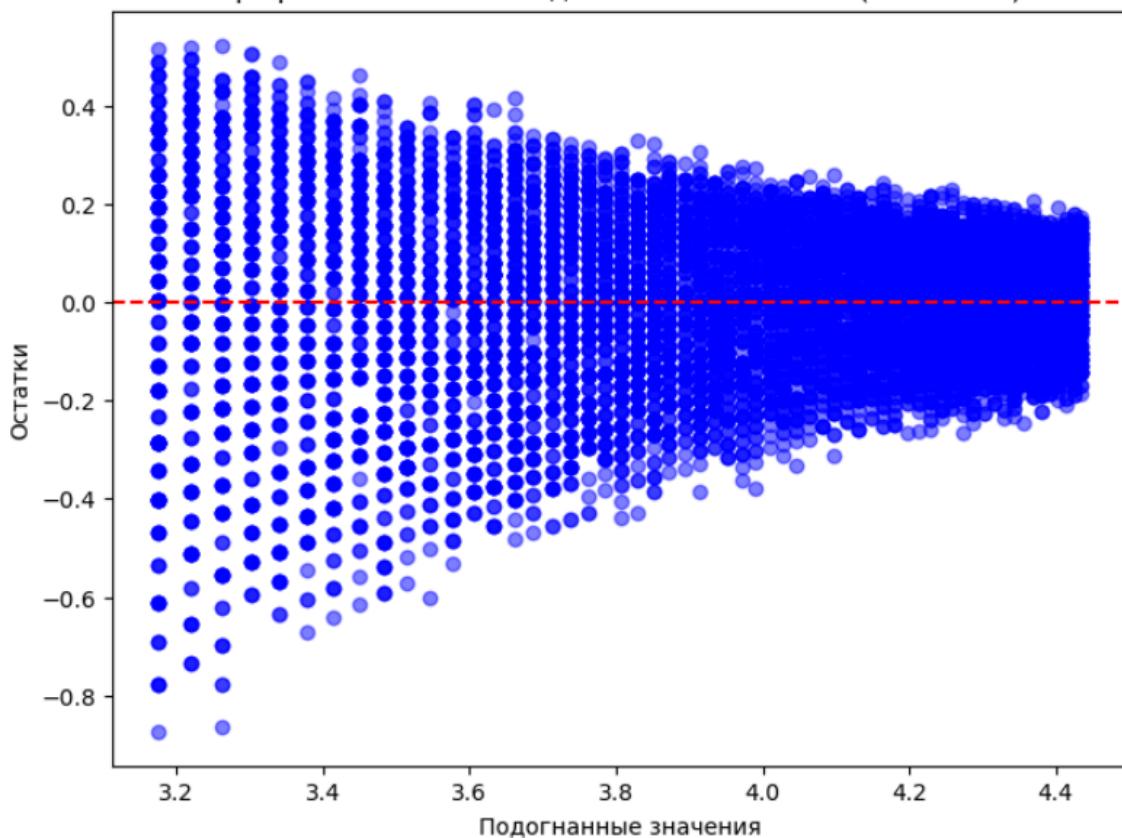


График остатков vs Подогнанные значения (степень 4)



Полиномиальная регрессия: степень 6

MSE: 0.03101648145636343

R2-squared: 0.8036674736805587

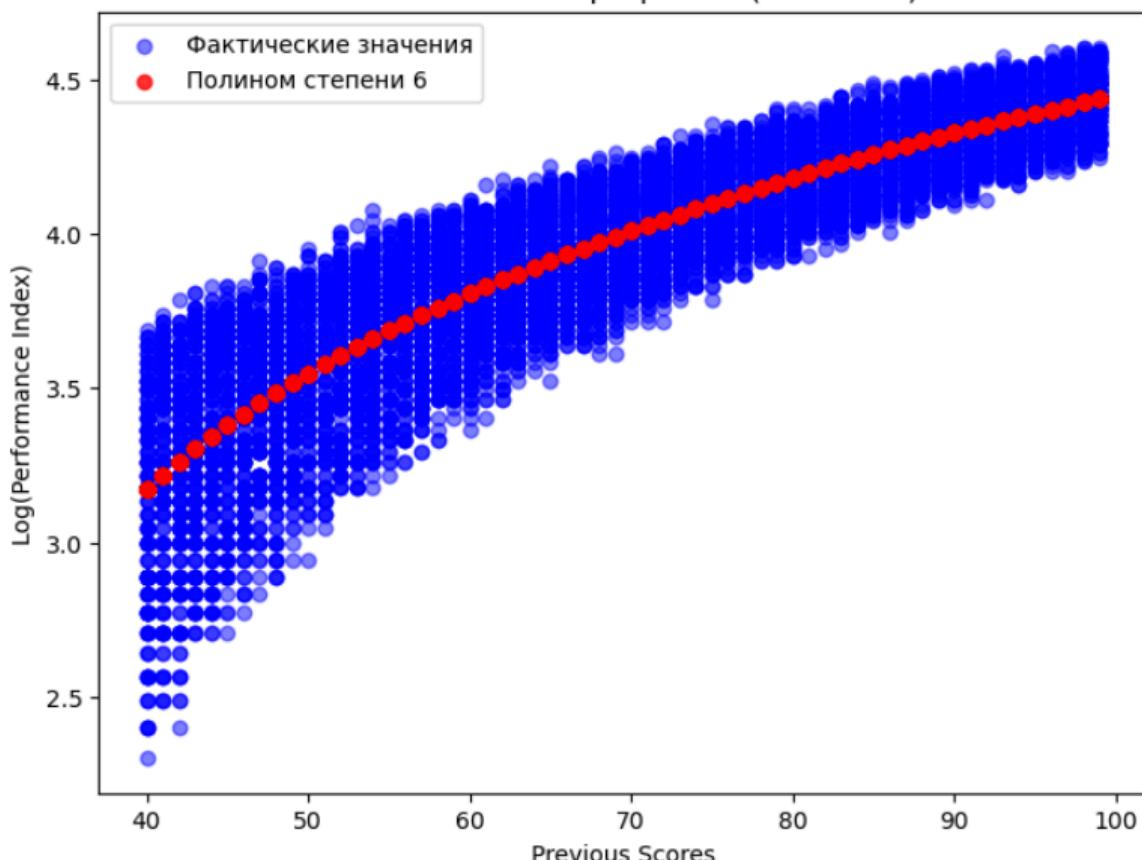
OLS Regression Results

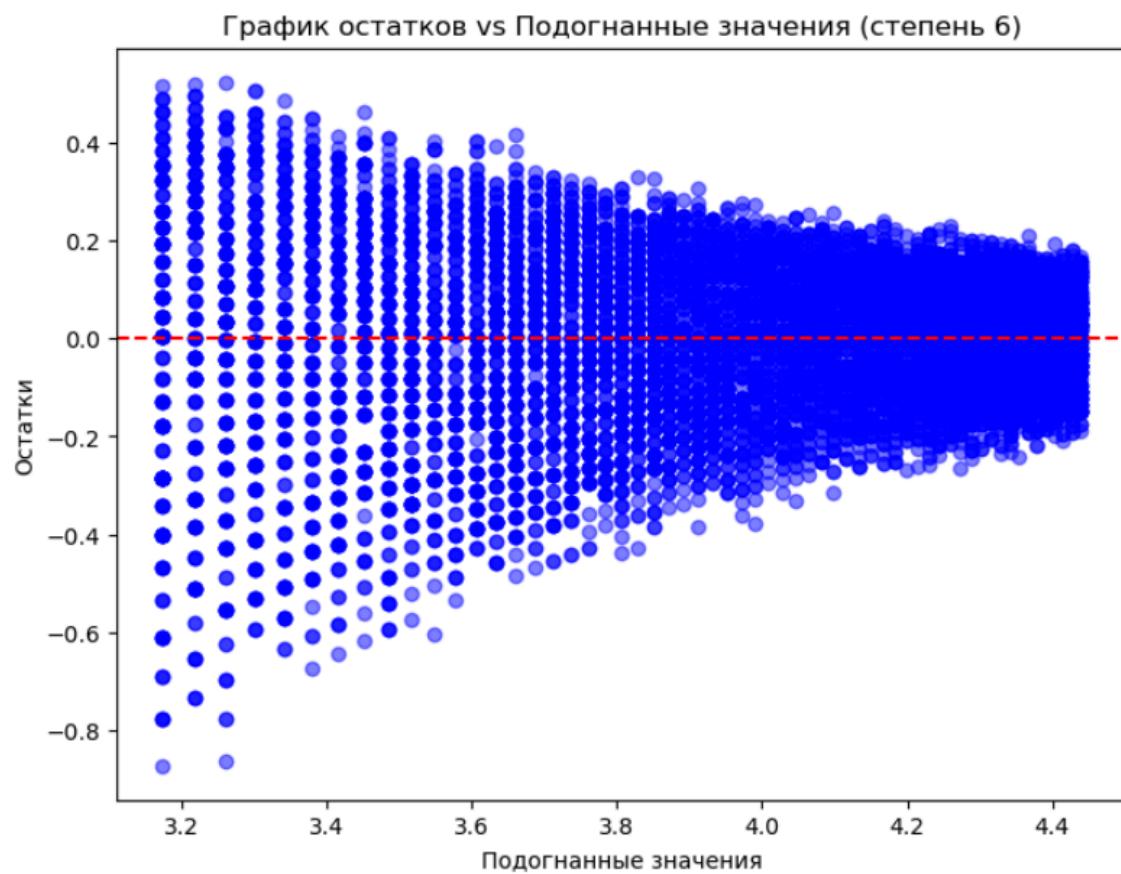
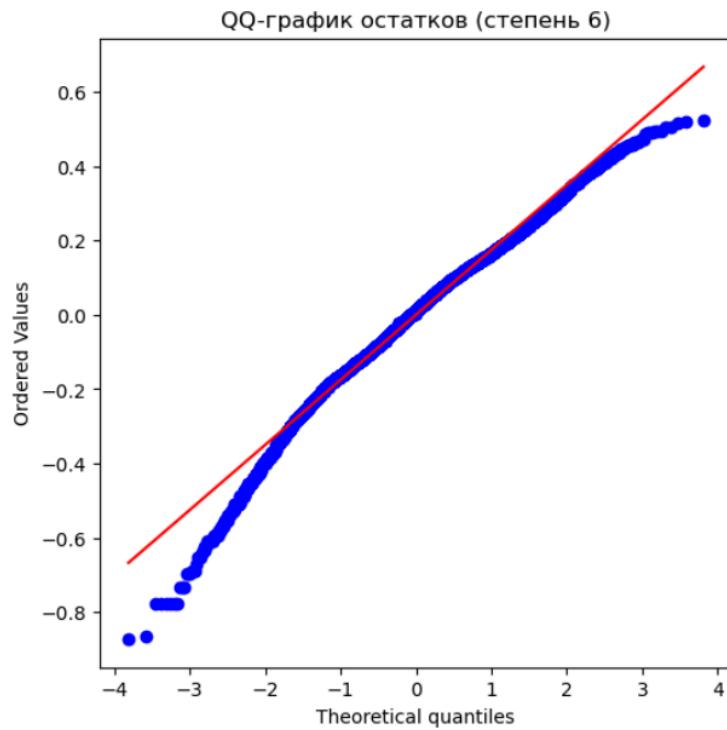
Dep. Variable:	Log_Performance_Index	R-squared:	0.804
Model:	OLS	Adj. R-squared:	0.804
Method:	Least Squares	F-statistic:	8182.
Date:	Mon, 25 Nov 2024	Prob (F-statistic):	0.00
Time:	12:09:44	Log-Likelihood:	3176.8
No. Observations:	10000	AIC:	-6342.
Df Residuals:	9994	BIC:	-6298.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	0.0064	0.004	1.586	0.113	-0.002	0.014
x1	0.0679	0.043	1.593	0.111	-0.016	0.152
x2	0.0030	0.003	0.912	0.362	-0.003	0.009
x3	-0.0001	9.77e-05	-1.257	0.209	-0.000	6.87e-05
x4	1.834e-06	1.43e-06	1.282	0.200	-9.7e-07	4.64e-06
x5	-1.262e-08	1.03e-08	-1.229	0.219	-3.28e-08	7.51e-09
x6	3.345e-11	2.9e-11	1.154	0.248	-2.34e-11	9.02e-11

Omnibus:	454.571	Durbin-Watson:	2.029
Prob(Omnibus):	0.000	Jarque-Bera (JB):	611.772
Skew:	-0.452	Prob(JB):	1.43e-133
Kurtosis:	3.806	Cond. No.	2.21e+15

Полиномиальная регрессия (степень 6)





R:

Полиномиальная регрессия: степень 2

MSE: 0.0312243995692983

R2-squared: 0.802351364100641

Call:

```
lm(formula = y ~ poly_features)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.91807	-0.11153	0.01025	0.12175	0.49834

Coefficients:

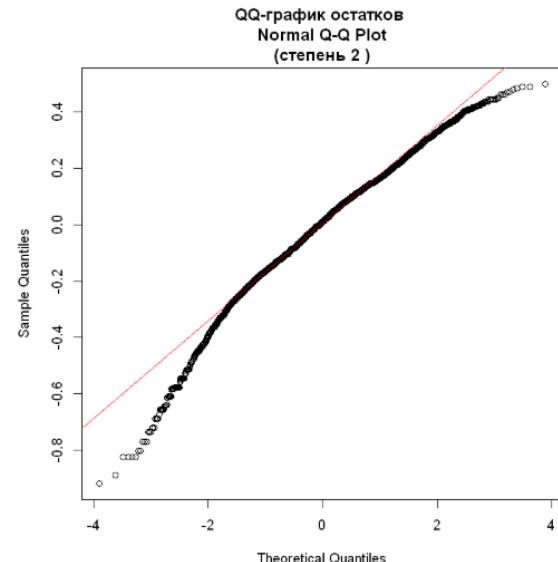
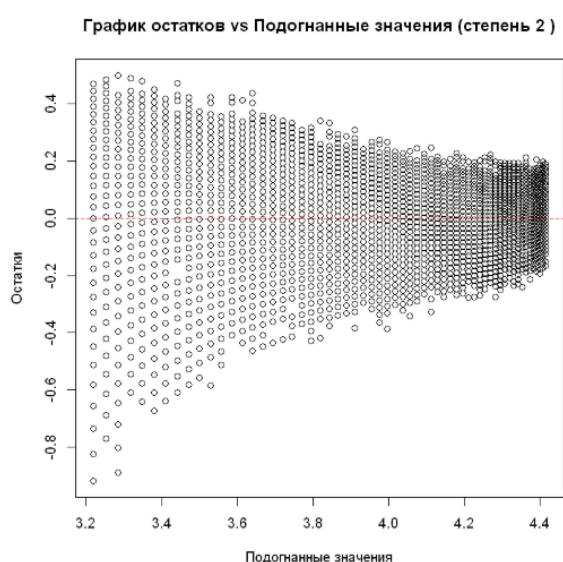
	Estimate	Std. Error	t value	Pr(> t)							
(Intercept)	1.553e+00	3.087e-02	50.30	<2e-16 ***							
poly_features1	5.037e-02	9.263e-04	54.37	<2e-16 ***							
poly_features2	-2.167e-04	6.625e-06	-32.70	<2e-16 ***							

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	.	0.1	' '	1

Residual standard error: 0.1767 on 9997 degrees of freedom

Multiple R-squared: 0.8024, Adjusted R-squared: 0.8023

F-statistic: 2.029e+04 on 2 and 9997 DF, p-value: < 2.2e-16



Полиномиальная регрессия: степень 4

MSE: 0.0310176847027242

R2-squared: 0.803659857200974

Call:

lm(formula = y ~ poly_features)

Residuals:

Min	1Q	Median	3Q	Max
-0.87309	-0.11176	0.01241	0.12253	0.52242

Coefficients:

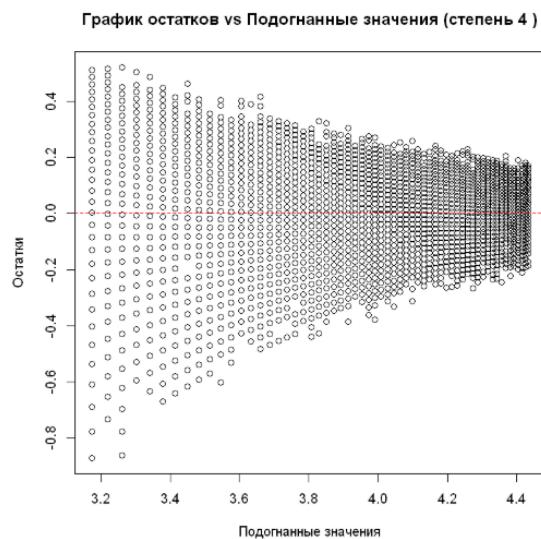
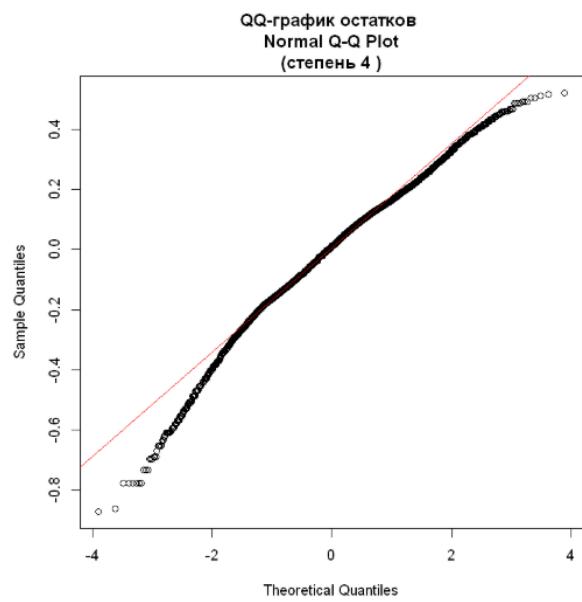
	Estimate	Std. Error	t value	Pr(> t)							
(Intercept)	-1.169e+00	5.788e-01	-2.020	0.043385 *							
poly_features1	2.060e-01	3.594e-02	5.732	1.02e-08 ***							
poly_features2	-3.416e-03	8.141e-04	-4.196	2.74e-05 ***							
poly_features3	2.812e-05	7.988e-06	3.520	0.000433 ***							
poly_features4	-8.949e-08	2.870e-08	-3.118	0.001826 **							

Signif. codes:	0	'***'	0.001	'**'	0.01	'*'	0.05	'..'	0.1	' '	1

Residual standard error: 0.1762 on 9995 degrees of freedom

Multiple R-squared: 0.8037, Adjusted R-squared: 0.8036

F-statistic: 1.023e+04 on 4 and 9995 DF, p-value: < 2.2e-16



Полиномиальная регрессия: степень 6

MSE: 0.03101477098798

R2-squared: 0.803678300846089

Call:

lm(formula = y ~ poly_features)

Residuals:

Min	1Q	Median	3Q	Max
-0.86724	-0.11188	0.01064	0.12242	0.52239

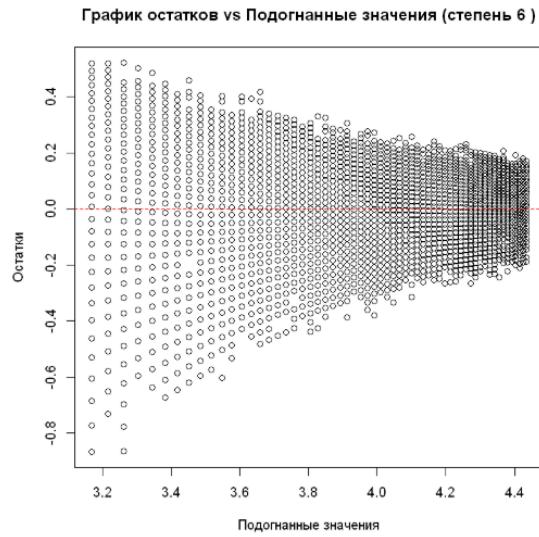
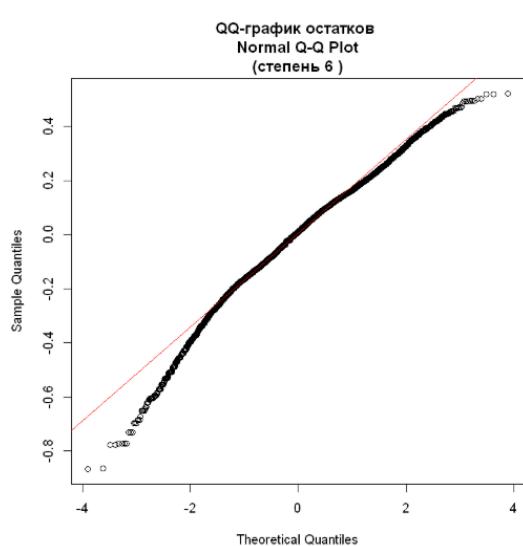
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-8.318e+00	1.121e+01	-0.742	0.458
poly_features1	8.496e-01	1.054e+00	0.806	0.420
poly_features2	-2.701e-02	4.053e-02	-0.666	0.505
poly_features3	4.791e-04	8.166e-04	0.587	0.557
poly_features4	-4.837e-06	9.099e-06	-0.532	0.595
poly_features5	2.612e-08	5.319e-08	0.491	0.623
poly_features6	-5.877e-11	1.275e-10	-0.461	0.645

Residual standard error: 0.1762 on 9993 degrees of freedom

Multiple R-squared: 0.8037, Adjusted R-squared: 0.8036

F-statistic: 6818 on 6 and 9993 DF, p-value: < 2.2e-16



Видно, что полиномы несколько улучшают коэффициент детерминации, но MSE остается примерно той же. В целом сильного отличия от вывода с прошлой моделью с логарифмическим преобразованием нет.

4. Заключение

Проведенный практикум включает в себя широкий спектр статистических и аналитических методов, применяемых для изучения свойств данных, проверки гипотез и построения аналитических методов.

Его цель заключалась в изучении различных подходов анализа данных и его особенностей, что позволило получить важные навыки для прикладных исследований:

- полный цикл анализа от визуализации до построения и проверки моделей;
- анализ реальных статистических методов и критериев, их особенности и применение в разных задачах;
- оценка качества моделей и проверка предпосылок методов.

Изученные методы помогают не только выявлять закономерности, но и строить обоснованные выводы на основе данных.

Эти навыки находят применение в исследованиях, экономике, медицине, инженерии и других областях, где требуется глубокий анализ данных.

Это были основные результаты, полученные при тестировании методов на различных данных. Более подробно ознакомиться с реализацией кода можно в файлах, приложенных к этому отчету.

5. Список литературы

1. Брюс П., Брюс Э., Гедек П. Практическая статистика для специалистов Data Science. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2021. – 352 с.
2. Мастицкий С.Э., Шитиков В.К. Статистический анализ и визуализация данных с помощью R. – М.: ДМК Пресс, 2015. – 496 с.
3. Нильсен Э. Практический анализ временных рядов: прогнозирование со статистикой и машинное обучение. – СПб.: ООО «Диалектика», 2021. – 544 с.
4. [Проверка статистических гипотез](#) Смирнова З.М., Крейнина М.В.
5. [Критерии нормальности](#)
6. [Работа с пропусками в наборе данных](#)
7. [VIF](#)
8. [Регрессионный анализ](#)
9. [Дисперсионный анализ](#)
10. [Анализ и визуализация реальных табличных данных в R](#)