

Практическое применение критерия Уилкоксона–Манна–Уитни в оценочной деятельности

**Отбор признаков в качестве ценообразующих факторов на основе
принципа несмещённости оценок**

К. А. Мурашев

14 июля 2022 г.

В своей практике оценщики часто сталкиваются с необходимостью учёта различий количественных и качественных характеристик объектов. В частности, одной из стандартных задач является установление признаков, влияющих на стоимость (т. н. ценообразующих факторов) и их отделение от признаков, влияние которых на стоимость отсутствует либо не может быть установлено.

В практике оценки широкое распространение получил субъективный отбор признаков, учитываемых при определении стоимости. При этом конкретные количественные показатели влияния этих признаков на стоимость зачастую берутся из т. н. «справочников». Не отказывая такому подходу в быстроте и невысокой стоимости его реализации, нельзя не признать, что только данные, непосредственно наблюдаемые на открытом рынке, являются надёжной основой суждения о стоимости. Приоритет таких данных над прочими, в частности, полученными путём опроса экспертов, закреплён, в том числе в Стандартах оценки RICS [23], Международных стандартах оценки 2022 [25], а также в МСФО 13 «Оценка справедливой стоимости» [18]. Поэтому можно говорить о том, что математические методы анализа данных, полученных на открытом рынке, являются наиболее надёжным средством интерпретации рыночной информации, применяемой при исследованиях рынка и предсказании стоимости конкретных объектов.

Задачей данной работы является обоснование необходимости и возможности использования строгого математического теста Уилкоксона—Манна—Уитни, позволяющего дать ответ на вопрос о необходимости учёта качественного признака в качестве ценообразующего фактора. Вместо сужденческого подхода, наиболее часто применяемого оценщиками при выборе признаков, учитываемых при проведении оценки, в данной работе предлагается идея приоритета измерительного подхода, основанного на результатах математического теста, позволяющего сделать вывод о существенности либо несущественности влияния качественного признака на стоимость.

Следует отметить, что, несмотря на то, что рассматриваемый статистический тест относится к т. н. «частотной статистике», он, посредством связи с ROC-анализом и понятием AUC, имеет отношение к современным методам машинного обучения, о чём будет рассказано далее в тексте данного материала. Наличие данной связи и элементов «байесовской статистики» представляется особенно интересным и перспективным с точки зрения внедрения методов машинного обучения и анализа данных в повседневную практику оценщиков.

Для успешного понимания всего изложенного в тексте и реализации на практике предполагается наличие у пользователей определённой общематематической подготовки, а также базовых навыков программирования на Python и R, однако отсутствие этих знаний и навыков не является препятствием для усвоения большей части материала и реализации теста в электронной таблице, прилагающейся к нему.

Материал состоит из четырёх блоков:

- описание теста Манна—Уитни—Уилкоксона (далее *U-тест*), его вероятностного смысла и связи с другими математическими методами;

- практическая реализация U-теста в электронной таблице на примере тестовых случайных данных;
- практическая реализация U-теста на реальных данных рынка жилой недвижимости Санкт-Петербургской агломерации средствами языка программирования Python, целью анализа являлась проверка существенности различия удельной стоимости между объектами, расположенными в городской и пригородной частях агломерации;
- практическая реализация U-теста на реальных данных рынка жилой недвижимости города Алматы средствами языка программирования R, целью анализа являлась проверка существенности различия удельной стоимости между объектами, продаваемыми без отдельных улучшений и объектами, продаваемыми вместе с ними.

Актуальная версия данного материала, её исходный код, скрипты на Python и R, а также электронная таблица находятся в репозитории на портале GitHub и доступны по постоянной ссылке [26].

Данный материал и все приложения к нему распространяются на условиях лицензии cc-by-sa-4.0 [30].

Оглавление

1. Технические данные	11
2. Предмет исследования	12
3. Основные сведения о тесте	14
3.1. Предпосылки и формализация гипотез	14
3.2. Реализация теста	17
3.2.1. Статистика критерия	17
3.2.2. Методы вычисления	17
3.2.3. Интерпретация результата	19
3.2.3.0.1. Показатель CLES	19
3.2.3.0.2. Рангово-бисериальная корреляция	20
3.2.4. Вычисление р-значения и итоговая проверка нулевой гипотезы	20
3.3. Соотношение с другими статистическими тестами	22
3.3.1. Сравнение U-теста Уилкоксона—Манна—Уитни с t-тестом Стью- дента	22
3.3.2. Альтернативные тесты в случае неравенства распределений	22
3.3.3. Связь между U-тестом и задачами классификации	23
3.4. Связь между U-тестом и понятиями Receiver Operating Characteris- tic (ROC), Area under curve (AUC)	24
3.4.1. Основные сведения о ROC	24
3.4.2. Понятие AUC и её вычисление	31
3.4.3. Связь между U-тестом и AUC	33
3.4.4. Практика ROC-анализа и вычисления AUC	34
3.4.4.1. Построение ROC-кривой	34
3.4.4.2. Понятие AUC и вычисление её значения	42
3.4.4.2.1. Геометрический подход к понятию AUC	44
3.4.4.2.1.1. Построение графика	44
3.4.4.2.1.2. Суммирование площадей с помощью собственной функции	46
3.4.4.2.2. Подход, основанный на сравнении рангов	49
3.4.4.2.3. Вероятностный подход к вычислению AUC	51
3.4.4.3. Вычисление AUC для больших наборов данных	52
3.4.4.4. Обобщение выводов по AUC	55

4. Практическая реализация	58
4.1. Реализация в табличном процессоре LibreOffice Calc	58
4.2. Реализация на Python	64
4.2.1. ROC-анализ, вычисление AUC	76
4.3. Реализация на R	79
4.3.1. Вычисление AUC	87
4.3.1.1. Вычисление AUC	87
4.3.1.2. Графическая интерпретация U-статистики	89
5. Выводы	97

Список таблиц

3.1	Варианты нулевой гипотезы при использовании U-теста при оценке стоимости	17
3.2	Свойства U-теста относительно t-теста	23
3.3	Таблица сопряжённости результатов работы бинарного классификатора	25
3.4	Возможные исходы применения бинарного классификатора	26
3.5	Показатели TPR, FPR, labels, scores для учебного датасета.	46
3.6	Результаты вычисления AUC четырьмя методами.	54
4.1	Нулевая и альтернативная гипотезы при анализе тестовых данных. . .	60
4.2	Нулевая и альтернативная гипотезы при анализе данных Санкт-Петербургской городской агломерации.	77
4.3	Результаты проведения тестов проверки данных по Санкт-Петербургской агломерации на нормальность ($\alpha = 0.05$).	78
4.4	Результаты проведения U-теста для данных Санкт-Петербургской агломерации ($\alpha = 0.05$).	78
4.5	Нулевая и альтернативная гипотезы при анализе данных Алматы. . .	94
4.6	Сведения о количестве наблюдений различных типов на рынке города Алматы.	94
4.7	Базовые описательные статистики наблюдений различных типов на рынке города Алматы (единица — казахстанский тенге).	95
4.8	Результаты проведения тестов проверки данных по г. Алматы на нормальность ($\alpha = 0.05$).	95
4.9	Результаты проведения U-теста для данных Алматы($\alpha = 0.05$).	95

Список диаграмм

3.1	Визуализация понятия стандартизированного значения (z-score) для нормального распределения. [68]	21
3.2	Диаграмма плотностей распределения вероятностей TPR и FPR при пороговом значении 0.	28
3.3	Диаграмма плотностей распределения вероятностей TPR и FPR при пороговом значении 1.	29
3.4	Диаграмма плотностей распределения вероятностей TPR и FPR при пороговом значении 0.	31
3.5	Диаграмма плотностей распределения вероятностей TPR и FPR при пороговом значении 1.	31
3.6	Диаграмма плотностей распределения вероятностей TPR и FPR при равном среднем.	32
3.7	Диаграмма распределения деталей относительно параметра x .	37
3.8	Сравнение порядка расположения точек для link и response.	40
3.9	Идентичные ROC-кривые, построенные с помощью библиотечной и собственной функций.	42
3.10	ROC-кривая, возникающая в случае отсутствия значения порядка оценок.	43
3.11	ROC-кривая и площадь под ней с учётом наличия повторяющихся значений.	48
3.12	Визуализация матрицы сравнений оценок (рангов).	51
3.13	ROC-кривая для оригинальных (чёрная линия) и округлённых (красная линия) данных о бракованных запчастях.	54
3.14	ROC-кривая для оригинальных данных о бракованных запчастях, построенная на основе матрицы сравнений.	55
3.15	ROC-кривая для округлённых данных о бракованных запчастях, построенная на основе матрицы сравнений.	56
4.1	Диаграмма «ящик с усами» (Boxplot) для обеих выборок.	61
4.2	Гистограмма первой выборки, совмещённая с кривой функции плотности вероятности для нормального распределения.	62
4.3	Гистограмма второй выборки, совмещённая с кривой функции плотности вероятности для нормального распределения.	63

4.4	Гистограмма плотности распределения цен за 1 кв. м квартир в Санкт-Петербургской агломерации, совмещённая с кривой функции плотности вероятности для нормального распределения.	68
4.5	Гистограмма плотности распределения цен за 1 кв. м квартир в Санкт-Петербурге, совмещённая с кривой функции плотности вероятности для нормального распределения.	71
4.6	Гистограмма плотности распределения цен за 1 кв. м квартир в Ленинградской области, расположенных в границах агломерации Санкт-Петербурга, совмещённая с кривой функции плотности вероятности для нормального распределения.	72
4.7	Диаграмма «ящик с усами» для цен предложений квартир в Санкт-Петербургской агломерации в разрезе региональной принадлежности.	76
4.8	Гистограмма цен предложения для всех объектов, совмещённая с кривой функции плотности эмпирического распределения, а также кривой функции плотности теоретического нормального распределения.	84
4.9	Гистограмма цен предложения для объектов, предлагаемых к продаже вместе с предлагаемыми к продаже совместно с отдельными улучшениями, совмещённая с кривой функции плотности эмпирического распределения, а также кривой функции плотности теоретического нормального распределения.	86
4.10	Гистограмма цен предложения для объектов, предлагаемых к продаже без отдельных улучшений, совмещённая с кривой функции плотности эмпирического распределения, а также кривой функции плотности теоретического нормального распределения.	87
4.11	Диаграмма «ящик с усами для рынка Алматы».	88
4.12	Визуализация рангов положительных и отрицательных наблюдений для рынка Алматы.	93
4.13	Визуализация рангов положительных и отрицательных наблюдений для рынка Алматы.	96

Листинги

3.1	Построение диаграммы плотностей распределения вероятностей TPR и FPR	27
3.2	Построение интерактивной диаграммы плотности распределения TPR и FPR и соответствующей ей ROC кривой для заданного порогового значения	30
3.3	Вычисление p-значения для тестовых данных	33
3.4	Создание функции для расчёта TPR и FPR	35
3.5	Создание и первичная визуализация данных о качественных и бракованных деталях	38
3.6	Сравнение предсказаний типов 'link' и 'response'	39
3.7	Построение ROC-кривой с помощью библиотечной и собственной функций	41
3.8	Построение ROC-кривой в случае отсутствия значения порядка оценок	41
3.9	Создание тестового набора данных	44
3.10	Расчёт AUC с помощью библиотеки pRoc	45
3.11	Создание функции rectangle	46
3.12	Добавление столбцов dFPR и dTPR	47
3.13	Построение пустого графа и разметка осей от 0 до 1	47
3.14	Построение ROC-кривой и прямоугольников под ней	47
3.15	Создание функции для вычисления AUC в полуавтоматическом режиме и её применение к тестовым данным	49
3.16	Создание функции для матрицы сравнений и её применение к тестовым данным	50
3.17	Создание и применение функции для вычисления AUC как вероятности	52
3.18	Построение ROC-кривой на оригинальных (чёрная линия) и округлённых (красная линия) данных о бракованных запчастях	53
3.19	Применение четырёх функций для вычисления AUC и построение ROC-кривых на основе матриц сравнения для оригинальных и округлённых оценок	53
4.1	Подключение необходимых библиотек	65
4.2	Задание применяемого уровня значимости	65
4.3	Загрузка данных и создание датафрейма	65
4.4	Создание датафрейма содержащего только необходимые переменные и выгрузка из памяти неиспользуемых данных	66

4.5	Построение гистограммы для агломерации Санкт-Петербурга	67
4.6	Создание отдельных датафреймов для Санкт-Петербурга и Ленинградской области	68
4.7	Построение гистограммы для Санкт-Петербурга	69
4.8	Построение гистограммы для Ленинградской области	70
4.9	Построение диаграммы «ящик с усами» (boxplot) для обеих подвыборок	73
4.10	Тест Шапиро-Уилка для данных по Санкт-Петербургу	74
4.11	Тест Шапиро-Уилка для данных по Ленинградской области	74
4.12	Тест K2 Агостино для данных по Санкт-Петербургу	74
4.13	Тест K2 Агостино для данных по Ленинградской области	74
4.14	Тест Андерсона-Дарлинга для данных по Санкт-Петербургу	75
4.15	Тест Андерсона-Дарлинга для данных по Ленинградской области . . .	75
4.16	Проведение теста Манна—Уитни-Уилкоксона для данных удельных цен предложения квартир в агломерации Санкт-Петербурга	75
4.17	Вычисление AUC и RBC для данных Санкт-Петербургской агломерации	76
4.18	Подключение библиотек и задание значений констант и адреса рабочего каталога	81
4.19	Создание датафрейма и его настройка	81
4.20	Подсчёт количества наблюдений	81
4.21	Создание функции для расчёта k по формуле P. W. Nowiczki	82
4.22	Расчёт k по формуле P. W. Nowiczki для наблюдений различных типов	82
4.23	Построение гистограмм для наблюдений различных типов	83
4.24	Построение базовых описательных статистик для наблюдений различных типов	83
4.25	Построение диграммы «ящик с усами» для рынка Алматы	85
4.26	Проведение тестов на нормальность для наблюдений без отделимых улучшений	85
4.27	Проведение тестов на нормальность для наблюдений с отделимыми улучшениями	85
4.28	Проведение U-теста для данных города Алматы	86
4.29	Добавление переменной с бинарными метками	88
4.30	Создание собственной функции для расчёта AUC и её применение к данным рынка Алматы	89
4.31	Создание собственной функции для вычисления U-статистики и её применение к данным рынка Алматы	90
4.32	Визуализация рангов положительных и отрицательных наблюдений для рынка Алматы	91
4.33	Визуализация рангов положительных наблюдений для рынка Алматы	92

Глава 1.

Технические данные

Данный материал, а также приложения к нему доступны по постоянной ссылке [26]. Исходный код данной работы был создан с использованием языка \TeX [51] с набором макрорасширений \LaTeX 2 ϵ [52], дистрибутива TeXLive [53] и редактора TeXstudio [70]. Расчёт в форме электронной таблицы был выполнен с помощью LibreOffice Calc [34] (Version: 7.3.4.2 / LibreOffice Community Build ID: 30(Build:2); CPU threads: 4; OS: Linux 5.11; UI render: default; VCL: kf5 (cairo+xcb) Locale: en-US (en_US.UTF-8); UI: en-US Ubuntu package version: 1:7.3.4 rc2-0ubuntu0.20.04.1 lo1; Calc: threaded). Расчёт на языке R [54] (version 4.2.1 (2022-06-23) – "Funny-Looking Kid") был выполнен с использованием IDE RStudio (RStudio 2022.02.3+492 "Prairie Trillium" Release (1db809b8, 2022-05-20) for Ubuntu Bionic; Mozilla/5.0 (X11; Linux x86_64); AppleWebKit/537.36 (KHTML, like Gecko); QtWebEngine/5.12.8; Chrome/69.0.3497.128; Safari/537.36) [50]. Расчёт на языке Python (Version 3.9.12) [33] был выполнен с использованием среды разработки Jupyter Lab (Version 3.4.2) [40] и IDE Spyder (Spyder version: 5.1.5 None* Python version: 3.9.12 64-bit * Qt version: 5.9.7 * PyQt5 version: 5.9.2 * Operating System: Linux 5.11.0-37-generic) [49]. Графические материалы, использованные в подсекции 4.1, были подготовлены с использованием Geogebra (Version 6.0.666.0-202109211234) [35]. В данном материале как и в большинстве работ цикла были использованы следующие значения:

- уровень значимости — $\alpha = 0.05$;
- доверительный интервал — $Pr = 0.95$;
- начальное положение датчика псевдослучайных чисел — $seed = 19190709$.

В качестве десятичного знака используется точка. Большинство математических обозначений записаны так, как это принято в англоязычной среде. Например, тангенс обозначается как \tan , а не tg . Результаты статистических тестов признаются значимыми в случае, когда

$$p \leq \alpha. \quad (1.1)$$

Данное решение основано, в частности на результатах дискуссии, имевшей место на портале researchgate.net [39].

Глава 2.

Предмет исследования

В случае работы с рыночными данными перед оценщиком часто встаёт задача проверки гипотезы о существенности влияния на стоимость того или иного признака, измеренного в количественной, порядковой либо номинативной шкале. Аналогичная задача возникает у аналитиков рынка недвижимости, специалистов компаний-застройщиков, риелторов, сотрудников залоговых подразделений банков, лизинговых и страховых компаний, налоговых инспекторов и иных специалистов. При этом, зачастую отсутствует возможность сбора больших массивов данных, позволяющих применить широкий спектр методов машинного обучения. В ряде случаев оценщики осознанно сужают область сбора данных до узкого сегмента рынка, в результате чего в их распоряжении оказываются лишь сверхмалые выборки объёмом менее тридцати наблюдений. При этом, ценовые данные чаще всего имеют распределение отличное от нормального. В данном случае рациональным решением является применение U-теста. Сформулируем задачу:

- предположим, что у нас существуют две выборки удельных цен коммерческих помещений, часть из которых обладает некоторым признаком (например, имеет отдельный вход), часть — нет;
- необходимо установить: оказывает ли наличие этого признака существенное влияние на удельную стоимость недвижимости данного типа или нет.

На первый взгляд, согласно сложившейся практике, оценщик может просто субъективно признать те или иные признаки значимыми, а прочие нет, после чего принять значения корректировок на различия в этих признаках из справочников. Однако, как было сказано выше, такой подход вряд ли может считаться лучшей практикой, поскольку в этом случае отсутствует какой-либо анализ рынка. Кроме того, в таком случае вряд ли можно говорить о серьёзной ценности такой работы в принципе.

Вместо этого возможно использовать случайные выборки рыночных данных и применять к ним математические методы анализа, позволяющие делать доказательные с научной точки зрения выводы о значимости влияния на стоимость со стороны того или иного признака. Данные, используемые в настоящей работе при проведении U-теста средствами Python и R, представляют собой реальные рыночные данные,

часть из которых была собрана автором путём веб-скрепинга, часть — предоставлена коллегами для анализа. Прилагаемая электронная таблица настроена таким образом, что тестовые исходные данные могут быть сгенерированы псевдослучайным образом.

Предметом исследования данной работы является непараметрический критерий Уилкоксона—Манна—Уитни, предназначенный для выборок, имеющих распределение отличное от нормального. Данное обстоятельство имеет большое значение, поскольку ценовые данные, с которыми имеют дело оценщики, чаще всего имеют именно такое распределение, что исключает возможность применения параметрического z-критерия. Кроме того, рассматриваемый тест представляет большой интерес, поскольку он имеет связь с методами машинного обучения посредством AUC, вычисление которой посредством формулы, предусмотренной в рамках теста, равно её значению, вычисляемому посредством ROC-анализа. Таким образом, изучение U-теста открывает путь к дальнейшему погружению в мир машинного обучения, который проникает во многие сферы человеческой деятельности, и в обозримом будущем существенно изменит деятельность в области оценки стоимости.

Материал содержит описание теста и инструкции по его проведению, достаточные, по мнению автора, для его доказательного использования в процессе проведения оценки.

Глава 3.

Основные сведения о тесте

3.1. Предпосылки и формализация гипотез

В первую очередь необходимо сказать, что, несмотря на заявленное общее название, правильное всё же говорить о двух тестах:

- двухвыборочный критерий Уилкоксона, разработанный Фрэнком Уилкоксоном в 1945 году [44];
- U-критерий Манна–Уитни, являющийся дальнейшим развитием вышеуказанного критерия, разработанный Генри Манном и Дональдом Уитни в 1947 году [42].

Забегая вперёд, можно сказать о том, что статистики данных критериев линейно связаны, а их p -значения почти одинаковы, что с практической точки зрения позволяет говорить о вариациях одного теста, а не о двух отдельных [44]. В данной работе по всему тексту используется общее название, а также его сокращённый вариант — U-тест, исторически относимый к критерию Манна–Уитни. Некоторые авторы [16] рекомендуют использовать двухвыборочный критерий Уилкоксона в случаях, когда нет предположений о дисперсиях, а в случае равных дисперсий применять U-критерий Манна–Уитни. Однако экспериментальные данные указывают, что p -значения критериев Уилкоксона и Манна–Уитни практически совпадают в том числе и в случае, когда дисперсии выборок существенно различаются [44]. Придерживаясь принципа KISS [60], лежащего в основе всего данного цикла публикаций, автор приходит к выводу о возможности применения единого подхода.

Также следует помнить о том, что существует Критерий Уилкоксона для связанных выборок [45], представляющий собой отдельный тест, предназначенный для анализа различий между связанными выборками, тогда как рассматриваемый в данной работе U-тест предназначен для работы с двумя независимыми выборками.

Предположим, что заданы две выборки:

$$x^m = (x_1, x_2, \dots, x_m), x_i \in \mathbb{R}; \quad y^n = (y_1, y_2, \dots, y_n), y_i \in \mathbb{R} \quad : m \leq n.$$

- Обе выборки являются простыми, объединённая выборка независима.

- Выборки взяты из неизвестных непрерывных распределений $F(x)$ и $G(y)$ соответственно.

Простая выборка — это случайная, однородная, независимая выборка. Эквивалентное определение: выборка $x^m = (x_1, x_2, \dots, x_m)$ является простой, если значения (x_1, x_2, \dots, x_m) являются реализациями m независимых одинаково распределённых случайных величин. Иными словами, отбор наблюдений является не только случайным, но и не предполагает наличие каких-либо специальных правил отбора (например, выбор каждого 10-го наблюдения).

U-тест — это непараметрический тест для проверки нулевой гипотезы, заключающейся в том, что для случайно выбранных из двух выборок наблюдений $x \in X$ и $y \in Y$ вероятность того, что x больше y , равна вероятности того, что y больше x . На математическом языке запись нулевой гипотезы выглядит следующим образом:

$$H_0 : P\{x < y\} = \frac{1}{2}. \quad (3.1)$$

Для целостности теста требуется альтернативная гипотеза, которая заключается в том, что вероятность того, что значение признака у наблюдения из выборки X превышает его у наблюдения из выборки Y , отличается (в большую или меньшую сторону) от вероятности того, что значение признака у наблюдения из Y превышает значение у наблюдения из X . На математическом языке запись альтернативной гипотезы выглядит следующим образом:

$$H_1 : P\{x < y\} \neq P\{y < x\} \vee P\{x < y\} + 0.5 \cdot P\{x = y\} \neq 0.5. \quad (3.2)$$

Согласно базовой концепции U-теста, при справедливости нулевой гипотезы распределение двух выборок непрерывно, при справедливости альтернативной гипотезы распределение одной из них стохастически больше распределения другой. При этом, можно сформулировать целый ряд нулевых и альтернативных гипотез, для которых данный тест будет давать корректный результат. Его самое широкое обобщение заключается в следующих предположениях:

- наблюдения в обеих выборках независимы;
- тип данных является как минимум ранговым, т. е. в отношении любых двух наблюдений можно сказать, какое из них больше;
- нулевая гипотеза предполагает, что распределения двух выборок равны;
- альтернативная гипотеза предполагает, что распределения двух выборок не равны.

При более строгом наборе допущений, чем приведённые выше, например, в случае допущения о том, что распределение двух выборок в случае справедливости нулевой гипотезы непрерывно, альтернативной — имеет сдвиг расположения двух

распределений, т.е. $f_1(x) = f_2(x + \sigma)$, можно сказать, что U-тест представляет собой тест на проверку гипотезы о равенстве медиан. В этом случае, U-тест можно интерпретировать как проверку того, отличается ли от нуля оценка Ходжеса—Лемана разницы значений мер центральной тенденции. В данной ситуации оценка Ходжеса—Лемана представляет собой медиану всех возможных значений различий между наблюдениями в первой и второй выборках. Вместе с тем, если и дисперсии, и формы распределения обеих выборок различаются, U-тест не может корректно проверить медианы. Можно показать примеры, когда медианы численно равны, при этом тест отвергает нулевую гипотезу вследствие малого р-значения.

Таким образом, более корректной интерпретацией U-теста является его использование для проверки именно гипотезы сдвига [43].

Гипотеза сдвига — статистическая гипотеза, часто рассматриваемая как альтернатива гипотезе о полной однородности выборок. Пусть даны две выборки данных. Пусть также даны две случайные величины X и Y , которые распределены как элементы этих выборок и имеют функции распределения $F(x)$ и $G(y)$ соответственно. В этих терминах гипотезу сдвига можно записать следующим образом:

$$H : F(x) = G(x + \sigma) : \forall x, \sigma \neq 0. \quad (3.3)$$

В этом случае U-критерий является состоятельным независимо от особенностей выборок.

Проще говоря, суть U-теста заключается в том, что он позволяет ответить на вопрос, является ли существенным различие значения количественного признака двух выборок. Применительно к оценке можно сказать, что использование данного теста помогает ответить на вопрос, является ли необходимым учёт того или иного признака в качестве ценообразующего фактора. Из сказанного выше следует, что, по умолчанию, речь идёт о двухстороннем тесте. На практике это означает, что тест не даёт прямой ответ, например на такой вопрос: «имеет ли место значимое превышение удельной стоимости помещений, имеющих отдельный вход, относительно помещений, не обладающих им». Вместо этого корректно говорить: «существует ли существенное различие в значении стоимости между помещениями двух типов: с отдельным входом и без такового». При этом существуют и односторонние реализации, позволяющие ответить на вопрос о знаке различия значения признака у двух выборок.

Условиями применения U-теста помимо вышеуказанных требований к самим выборкам являются:

- распределение значений количественного признака выборок отлично от нормального (в противном случае целесообразно использование параметрических t-критерия Стьюдента либо z-критерия для независимых выборок);
- не менее трёх значений признака в каждой выборке, допускается наличие двух значений в одной из выборок, при условии наличия в другой не менее пяти.

Подытоживая вышесказанное, можно сказать, что существуют три варианта нулевой гипотезы, в зависимости от уровня строгости, изложенные далее в таблице 3.1.

Таблица 3.1. Варианты нулевой гипотезы при использовании U-теста при оценке стоимости.

Тип гипотезы	Формулировка
Научная	Две выборки полностью однородны, т. е. принадлежат одному распределению, сдвиг отсутствует, оценка, сделанная для первой выборки, является несмещённой и для второй
Практическая	Медианы двух выборок равны между собой
Изложенная в терминах оценки	Различие признака между двумя выборками объектов-аналогов не является существенным, его учёт не требуется, данный признак не является ценообразующим фактором

3.2. Реализация теста

3.2.1. Статистика критерия

Допустим, что элементы x_1, \dots, x_n представляют собой простую независимую выборку из множества $X \in \mathbb{R}$, а элементы y_1, \dots, y_n представляют собой простую независимую выборку из множества $Y \in \mathbb{R}$, при этом выборки являются независимыми относительно друг друга. Тогда соответствующая U-статистика определяется следующим образом:

$$U = \sum_{i=1}^m \sum_{j=1}^n S(x_i, y_j),$$

при

$$S(x, y) = \begin{cases} 1, & \text{если } x > y, \\ \frac{1}{2}, & \text{если } x = y, \\ 0, & \text{если } x < y. \end{cases} \quad (3.4)$$

3.2.2. Методы вычисления

Тест предполагает вычисление статистики, обычно называемой U-статистикой, распределение которой известно в случае справедливости нулевой гипотезы. При работе со сверхмалыми выборками распределение задаётся таблично, при размерах выборки более двадцати наблюдений оно достаточно хорошо аппроксимируется нормальным распределением. Существуют два метода вычисления U-статистики: подсчёт вручную по формуле 3.4 либо применение специального алгоритма. Первый способ в силу трудоёмкости подходит только для сверхмалых выборок. Второй способ может быть формализован в виде пошагового набора инструкций и будет описан далее.

- 1) Необходимо построить общий вариационный ряд для двух выборок, а затем присвоить каждому наблюдению ранг, начиная с единицы для наименьшего из них. В случае наличия связей, т. е. групп повторяющихся значений (такой

группой могут являться в т.ч. и только два равных значения), каждому наблюдению из такой группы присваивается значение, равное медиане значений рангов группы до корректировки (например, в случае вариационного ряда (3, 5, 5, 5, 5, 8) ранги до корректировки имеют вид (1, 2, 3, 4, 5, 6) после — (1, 3.5, 3.5, 3.5, 3.5, 6)).

- 2) Необходимо провести подсчёт сумм рангов наблюдений каждой из выборок, обозначаемых как R_1 , R_2 соответственно. При этом, общая сумма рангов R может быть вычислена по формуле

$$R = \frac{N(N+1)}{2}, \quad (3.5)$$

где N — общее число наблюдений в обеих выборках.

- 3) Далее вычисляем U -значение для первой выборки:

$$U_1 = R_1 - \frac{n_1(n_1+1)}{2}, \quad (3.6)$$

где R_1 — сумма рангов первой выборки, n_1 — число наблюдений в первой выборке.

Аналогичным образом вычисляется U -значение для второй выборки:

$$U_2 = R_2 - \frac{n_2(n_2+1)}{2}, \quad (3.7)$$

где R_2 — сумма рангов второй выборки, n_2 — число наблюдений во второй выборке.

Из вышеприведённых формул следует, что

$$U_1 + U_2 = R_1 - \frac{n_1(n_1+1)}{2} + R_2 - \frac{n_2(n_2+1)}{2}. \quad (3.8)$$

Также известно, что

$$\begin{cases} R_1 + R_2 = \frac{N(N+1)}{2} \\ N = n_1 + n_2. \end{cases} \quad (3.9)$$

Тогда

$$U_1 + U_2 = n_1 n_2. \quad (3.10)$$

Использование данной формулы в качестве контрольного соотношения может быть полезно для проверки корректности вычислений при расчёте в табличном процессоре.

- 4) Из двух значений U_1 , U_2 во всех случаях выбираем меньшее, которое и будет являться U -статистикой и использоваться в дальнейших расчётах. Обозначим его как U .

3.2.3. Интерпретация результата

Для корректной интерпретации результата теста необходимо указать:

- размеры выборок;
- значения меры центральной тенденции для каждой выборки (с учётом непараметрического характера теста, подходящей мерой центральной тенденции представляется медиана);
- значение самой U-статистики;
- показатель CLES [58];
- рангово-бисериальный коэффициент корреляции (RBC) [64];
- принятый уровень значимости (как правило 0.05);
- расчётное р-значение.

Понятие U-статистики было рассмотрено ранее, большинство других показателей широко известны и не требуют какого-либо отдельного рассмотрения. Остановимся на показателях CLES и RBC.

3.2.3.0.1. Показатель CLES

Common language effect size (CLES) — вероятность того, что значение случайно выбранного наблюдения из первой группы больше значения случайно выбранного наблюдения из второй группы. Данный показатель вычисляется по формуле

$$CLES = \frac{U_1}{n_1 n_2}. \quad (3.11)$$

Вместо обозначения *CLES* часто используется обозначение *f* (*favorable*). Данное выборочное значение является несмещённой оценкой значения для всей совокупности объектов, принадлежащих множеству.

Следует отметить, что значение и смысл данного показателя эквивалентны значению и смыслу показателя AUC[65]. Таким образом, можно говорить о том, что данный показатель характеризует качество U-теста как бинарного классификатора.

$$CLES = f = AUC_1 = \frac{U_1}{n_1 n_2}. \quad (3.12)$$

Вопросы связи между U-статистикой и показателем (AUC) рассмотрены в 3.4.

3.2.3.0.2. Рангово-бисериальная корреляция Метод представления степени влияния для U-теста заключается в использовании меры ранговой корреляции, известной как рангово-бисериальная корреляция. Как и в случае с иными мерами корреляции значение коэффициента рангово-бисериальной корреляции имеет область значений $[-1; 1]$, при этом нулевое значение означает отсутствие какой-либо связи. Коэффициент рангово-бисериальной корреляции обычно обозначается как r . Для его вычисления используется простая формула, основанная на значении CLES. Выдвинем гипотезу о том, что в паре случайных наблюдений, одно из которых взято из первой выборки, другое — из второй, значение первого больше. Запишем её на математическом языке:

$$H : x_i > y_j, \quad x \in X, y \in Y. \quad (3.13)$$

Тогда значение коэффициента рангово-бисериальной корреляции представляет собой разницу между долей случайных пар наблюдений, удовлетворяющей (favorable) гипотезе — f , и комплементарной ей доле случайных пар, не удовлетворяющих (unfavorable) гипотезе — u . Таким образом, данная формула представляет собой формулу разности между показателями CLES для каждой из групп.

$$r = f - u = CLES_1 - CLES_2 = f - (1 - f) \quad (3.14)$$

Существует также ряд альтернативных формул, дающих идентичный результат:

$$r = 2f - 1 = \frac{2U_1}{n_1 n_2} - 1 = 1 - \frac{2U_2}{n_1 n_2}. \quad (3.15)$$

3.2.4. Вычисление р-значения и итоговая проверка нулевой гипотезы

При достаточном большом числе наблюдений в каждой выборке значение U-статистики имеет приблизительно нормальное распределение. Тогда её стандартизированное значение (z-метка, z-score) [68] может быть вычислено по формуле

$$z = \frac{U - m_U}{\sigma_U}, \quad (3.16)$$

где m_U — среднее арифметическое U , σ_U — её стандартное отклонение. Визуализация понятия *стандартизированное значения для нормального распределения* приведена на рисунке 3.1. Среднее для U вычисляется по формуле:

$$m_U = \frac{n_1 n_2}{2}. \quad (3.17)$$

Формула стандартного отклонения в случае отсутствия связей выглядит следующим образом:

$$\sigma_U = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}. \quad (3.18)$$

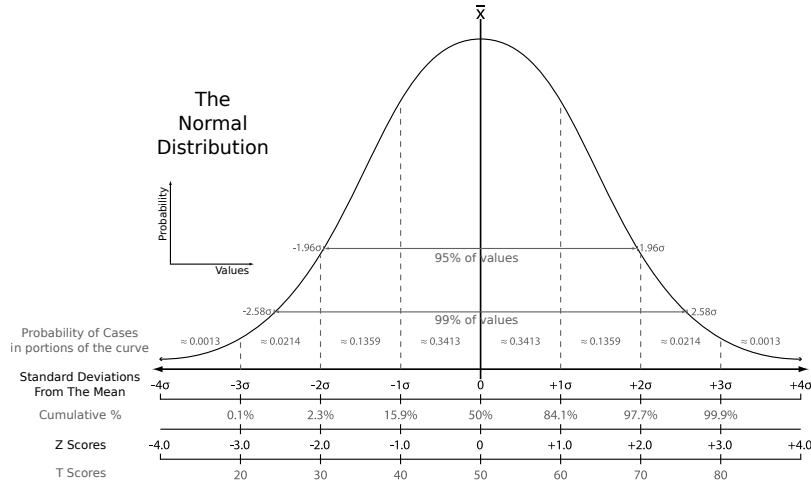


Рис. 3.1. Визуализация понятия стандартизированного значения (z-score) для нормального распределения. [68]

В случае наличия связей используется другая формула:

$$\sigma_{U_{ties}} = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12} - \frac{n_1 n_2 \sum_{k=1}^K (t_k^3 - t_k)}{12n(n-1)}} = \sqrt{\frac{n_1 n_2}{12} \left((n+1) - \frac{\sum_{k=1}^K (t_k^3 - t_k)}{n(n-1)} \right)}, \quad (3.19)$$

где t_k — количество наблюдений, имеющих ранг k , K — общее число рангов, имеющих связи. Далее, получив стандартизированное значение (z-score), и используя аппроксимацию стандартного нормального распределения, вычисляется p -значение для заданного уровня значимости (как правило 0.05). Интерпретация результата осуществляется следующим образом:

$$\begin{aligned} p \leq 0.05 &\Rightarrow \text{нулевая гипотеза отклоняется} \\ p > 0.05 &\Rightarrow \text{нулевая гипотеза не может быть отклонена.} \end{aligned} \quad (3.20)$$

При этом существует и альтернативный вариант интерпретации:

$$\begin{aligned} p < 0.05 &\Rightarrow \text{нулевая гипотеза отклоняется} \\ p \geq 0.05 &\Rightarrow \text{нулевая гипотеза не может быть отклонена.} \end{aligned} \quad (3.21)$$

На сегодняшний день нет однозначной позиции, как следует интерпретировать ситуацию, когда $p = \alpha$. В данной работе используется вариант, описанный в 3.20.

3.3. Соотношение с другими статистическими тестами

3.3.1. Сравнение U-теста Уилкоксона—Манна—Уитни с t-тестом Стьюдента

Часто можно услышать о том, что U-тест является непараметрическим аналогом t-теста, предназначенным для данных, чьё распределение отличается от нормального. С чисто практической точки зрения действительно можно сказать, что в случае нормального распределения определение наличия существенной разницы между двумя выборками целесообразно проводить посредством t-теста, в случае отличия распределения от нормального — U-теста. Таким образом, можно сказать, что эти тесты используются для одной и той же цели.

При этом, математический смысл U-теста и t-теста существенно отличается. Как уже было сказано ранее U-тест предназначен для проверки нулевой гипотезы, заключающейся в том, что для случайно выбранных из двух выборок наблюдений $x \in X$ и $y \in Y$ вероятность того, что x больше y , равна вероятности того, что y больше x , альтернативная гипотеза несёт утверждение о том, что эти вероятности не равны. В то же время t-тест предназначен для проверки нулевой гипотезы о равенстве средних двух выборок, при этом альтернативная гипотеза заключается в том, что средние двух выборок не равны. В связи с этим, при сравнении этих тестов следует иметь ввиду, что в общем случае, U-тест и t-тест проверяют разные нулевые гипотезы, хотя и имеющие отчасти схожий практический смысл. Результат U-теста чаще всего очень близок к результату двухвыборочного t-теста для ранжированных данных. Далее в таблице 3.2 проводится общее сравнение U-теста с t-тестом.

3.3.2. Альтернативные тесты в случае неравенства распределений

В случае необходимости проверки стохастического упорядочивания двух выборок (т.е. альтернативной гипотезы $P(Y > X) + 0.5P(Y = X) \neq 0.5$) без предположения о равенстве их распределений (т.е. когда нулевая гипотеза имеет вид: $P(Y > X) + 0.5P(Y = X) = 0.5$, но не $F(X) = G(Y)$), следует использовать более подходящие тесты. К ним относятся, в т.ч. тест Брунера-Мунцеля [19], представляющий собой устойчивый к гетероскедастичности аналог U-теста, и тест Флигнера-Поличелло [32], представляющий собой тест на равенство медиан. В частности, в случае использовании более общей нулевой гипотезы $P(Y > X) + 0.5P(Y = X) = 0.5$ U-тест может достаточно часто приводить к возникновению ошибки первого рода даже в случае работы с большими выборками (в особенности в случае неравенства дисперсий и существенно различающегося объёма выборок), вследствие чего в таких случаях использование альтернативных тестов будет предпочтительным [21]. Таким образом, в случае отсутствия предположения о равенстве распределений в случае справедливости нулевой гипотезы, использование альтернативных тестов будет являться предпочтительным.

В случае проверки гипотезы сдвига при существенно отличающихся распределениях U-тест может дать ошибочную интерпретацию значимости [12], вследствие

Таблица 3.2. Свойства U-теста относительно t-теста.

Свойство	Описание
Применимость к порядковому данным	В случае работы с порядковыми (ранговыми), а не количественными данными применение U-теста является предпочтительным относительно применения t-теста, при этом следует помнить, что расстояние между соседними значениями вариационного ряда нельзя считать постоянным.
Робастность	Поскольку U-тест работает с суммой рангов, а не значений признаков, он реже, чем t-тест ошибочно указывает на значимость вследствие наличия выбросов. Однако, в целом U-тест больше подвержен ошибке первого рода в случае, когда данные одновременно обладают свойством гетероскедастичности и имеют распределение отличное от нормального.
Эффективность	В случае нормального распределения асимптотическая эффективность U-теста составляет $\frac{3}{4}\pi \approx 0.95$ от показателя t-теста [11]. В случае существенного отличия распределения от нормального и достаточно большого числа наблюдений эффективность U-теста существенно превышает эффективность t-теста [7]. Однако такое сравнение эффективности следует интерпретировать с осторожностью, поскольку U-тест и t-тест проверяют разные гипотезы и оценивают разные величины. В случае, например, потребности в сравнении средних значений применение U-теста не является оправданным в принципе.

чего в таких условиях предпочтительным будет использование варианта t-теста [56], предназначенного для случаев неравных дисперсий [12].

В ряде случаев может быть оправданным преобразование количественных данных в ранги и последующее проведение t-теста в том или ином его варианте в зависимости от предположений о равенстве дисперсий. При преобразовании количественных данных в порядковые исходные дисперсии не будут сохранены, их следует пересчитать для самих рангов. В случае равенства дисперсий подходящей непараметрической заменой F-теста [31] может являться тест Брауна-Форсайта.

3.3.3. Связь между U-тестом и задачами классификации

U-тест представляет собой частный случай модели упорядоченного выбора (ordered logit model) [46].

3.4. Связь между U-тестом и понятиями Receiver Operating Characteristic (ROC), Area under curve (AUC)

3.4.1. Основные сведения о ROC

Основываясь на сказанном в 3.3.3, можно сделать вывод о том, что U-тест является не только тестом для проверки гипотезы сдвига (либо иной аналогичной по смыслу), но и представляет собой некий классификатор. Забегая вперёд можно сказать, что смысл U-теста как классификатора заключается в следующем:

- существует «позитивный» исход сравнения двух случайных наблюдений, заключающийся в том, что наблюдение из X больше наблюдения из Y ;
- проводится оценка доли суммы рангов «позитивных» элементов;
- как и в целом с ROC, в случае, если значение доли «позитивных» элементов превышает 0.5, это говорит о том, что классификатор в целом выполняет свою функцию, в случае равенства 0.5 — его эффективность равнозначна угадыванию с помощью подбрасывания монеты, в случае значения менее 0.5 — использование такого классификатора даёт обратный результат.

На первый взгляд, связь между U-тестом и ROC не выглядит очевидной. В данной секции будет предпринята попытка разобраться в том, почему эти понятия всё же имеют связь, и в чём заключается суть U-теста как классификатора.

Сам ROC-анализ не входит в периметр данного материала. Поэтому рассмотрим лишь его основные моменты.

ROC-кривая (ROC-curve: Receiver Operating Characteristic) — график, позволяющий оценить качество бинарной классификации, отображает соотношение между долей объектов от общего количества носителей признака, верно классифицированных как несущие признак (true positive rate (TPR), называемой *чувствительностью алгоритма классификации*), и долей объектов от общего количества объектов, не несущих признака, ошибочно классифицированных как несущие признак (false positive rate (FPR), величина $1-FPR$ называется *специфичностью алгоритма классификации*) при варьировании порога решающего правила. Также известна как **кривая ошибок**. Анализ классификаций с применением ROC-кривых называется **ROC-анализом**.

Количественная интерпретация ROC даёт показатель AUC (Area Under Curve, площадь под кривой). AUC — это площадь, ограниченная ROC-кривой и осью доли ложных положительных классификаций (ось абсцисс). Чем выше показатель AUC, тем качественнее классификатор, при этом значение 0.5 демонстрирует непригодность выбранного метода классификации (соответствует случайному угадыванию с помощью монеты). Значение менее 0.5 говорит, что классификатор действует с точностью до наоборот: если положительные результаты назвать отрицательными и наоборот, классификатор будет работать лучше [65].

Введём некоторые термины.

- P — количество объектов в выборке, обладающих некоторым признаком (Condition positive).
- N — количество объектов в выборке, не обладающих некоторым признаком (Condition negative).
- TP — результат теста, корректно определивший наличие существующего в действительности (Positive) признака (True positive, истинно положительный).
- TN — результат теста, корректно определивший отсутствие несуществующего в действительности (Negative) признака (True negative, истинно отрицательный).
- FP — результата теста, ошибочно определивший наличие несуществующего в действительности (Negative) признака (False positive, ложно положительный).
- FN — результат теста, ошибочно определивший отсутствие существующего в действительности (Positive) признака (False negative, ложно отрицательный).

На основании вышесказанного можно создать таблицу 3.3 сопряжённости результатов применения бинарного классификатора. Строки содержат данные о фактическом наличии либо отсутствии признака, столбцы — предсказанном (predicted) с помощью классификатора. Как видно из таблицы 3.4 бинарный классификатор может приво-

Таблица 3.3. Таблица сопряжённости результатов работы бинарного классификатора.

Всего $P + N$	Predicted Positive (PP)	Predicted negative (PN)
Positive (P)	TP	FN, ошибка второго рода [69]
Negative (N)	FP, ошибка первого рода [69]	TN

дить к возникновению ошибок двух типов. Введём ещё несколько определений и определим формулы для расчёта вероятностей исходов его работы. Вероятность TPR может быть записана как

$$P_{TPR} = \mathbb{P}(1, x \in C_1), \quad (3.26)$$

что означает, что если объект x принадлежит классу C_1 , то данный показатель оценивает вероятность того, что бинарный классификатор отнесёт объект x к этому классу. Вероятность FPR записывается как

$$P_{FPR} = \mathbb{P}(1, x \in C_0), \quad (3.27)$$

что означает вероятность того, что объект, принадлежащий классу C_0 будет ошибочно отнесён к классу C_1 .

Как правило, принцип работы бинарного классификатора основан на сравнения измерения x с некоторым фиксированным порогом c . Из этого следует, что два предыдущих выражения можно переписать и объединить в систему.

$$\begin{cases} P_{TPR} = \mathbb{P}(x > c, x \in C_1) \\ P_{FPR} = \mathbb{P}(x > c, x \in C_0) \end{cases} \quad (3.28)$$

Таблица 3.4. Возможные исходы применения бинарного классификатора.

Показатель	Формула	Альтернативные названия
TPR (SEN)	$TPR = \frac{TP}{P} = 1 - FNR \quad (3.22)$	True positive rate, Recall, Sensitivity, Probability of detection, hit rate, power, чувствительность
FPR	$FPR = \frac{FP}{N} = 1 - TNR \quad (3.23)$	False positive rate, probability of false alarm, fall-alarm
FNR	$FNR = \frac{FN}{P} = 1 - TPR \quad (3.24)$	False negative rate, miss rate
TNR (SPC)	$TNR = \frac{TN}{N} = 1 - FPR \quad (3.25)$	True negative rate, specifity, selectivity, специфичность

Из этого следует, что ROC-кривая представляет собой диаграмму

$$P_{FPR}(c), P_{TPR}(c), \quad (3.29)$$

таким образом, построение кривой означает изменение значения порога c .

Рассмотрим пример [29]. Возьмём $f(x \in C_0) = \mathcal{N}(0,1)$ и $f(x \in C_1) = \mathcal{N}(2,1)$ в качестве функций плотности вероятностей C_0 и C_1 соответственно. Далее поэтапно построим ROC-кривую средствами языка Python. Далее рассмотрим диаграмму 3.2, построенную с помощью кода, приведённого в скрипте 3.1. Область, закрашенная синим цветом, показывает вероятность FPR, т. е. ложно-положительного обнаружения значимости, тогда как область, закрашенная зелёным цветом — плотность вероятности TPR, т. е. корректного обнаружения значимости. ROC-кривая показывает значения именно этих показателей. Вертикальная прерывистая линия — порог чувствительности c . В данной ситуации он находится на отметке 0 по оси абсцисс. В случае его смещения на отметку 1 площадь под кривой FPR (синий цвет) существенно уменьшится, т. е. снизится вероятность ложно-положительного обнаружения признака, однако, вместе с этим уменьшится и площадь TPR (зелёный цвет), что означает увеличение ложно-отрицательных результатов. Данная ситуация рассмотрена на диаграмме 3.3.

Листинг 3.1. Построение диаграммы плотностей распределения вероятностей TPR и FPR

```
# Import Libraries
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

# Plot
f0 = stats.norm(0, 1)
f1 = stats.norm(2, 1)
fig, ax = plt.subplots()
xi = np.linspace(-2, 5, 100)
ax.plot(xi, f0.pdf(xi), label=r'$f(x|C_0)$')
ax.plot(xi, f1.pdf(xi), label=r'$f(x|C_1)$')
ax.legend(fontsize=16, loc=(1, 0))
ax.set_xlabel(r'$x$', fontsize=18)
ax.vlines(0, 0, ax.axis()[-1] * 1.1, linestyle='--', lw=3.)
ax.fill_between(xi, f1.pdf(xi), where=xi > 0, alpha=.3, color='g')
ax.fill_between(xi, f0.pdf(xi), where=xi > 0, alpha=.3, color='b')

# Save to .pdf
plt.savefig('Plot-ROC-step-1.pdf', bbox_inches='tight')
```

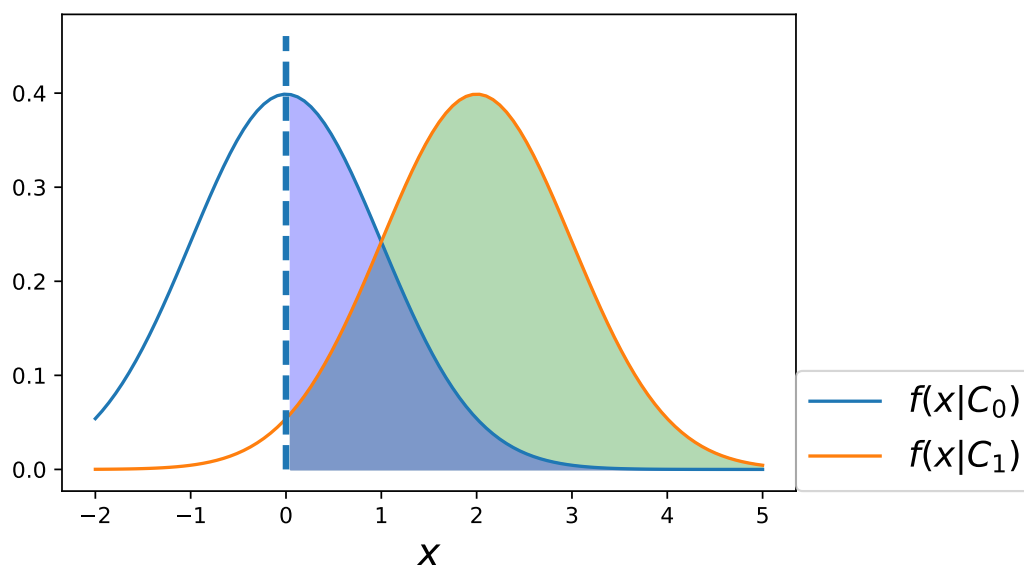


Рис. 3.2. Диаграмма плотностей распределения вероятностей TPR и FPR при пороговом значении 0.

Как видно из приведённых диаграмм, повышение порога приводит к потере части как истинно-положительных, так и ложно-положительных результатов, уменьшение — к увеличению числа фиксаций наличия признака (как верных так и ошибочных). В предельных случаях, слишком низкое пороговое значение приведёт к тому, что все результаты будут интерпретированы как положительные, слишком высокое — к нулевому количеству наблюдений, у которых был обнаружен признак. Задача ROC-анализа заключается в выборе рационального порогового значения.

Добавим к уже имеющимся диаграммам ROC-кривые, соответствующие пороговым значениям 0 и 1. А также создадим интерактивную диаграмму с помощью кода, представленного в скрипте 3.2. Формат PDF не позволяет добавлять подобные интерактивные элементы, поэтому рассмотрим случаи с фиксированными значениями 0 и 1, представленные на диаграммах 3.4, 3.5 соответственно. В левой части каждой из них показаны уже знакомые графики функций плотности вероятности распределений TPR, FPR. В правой части показана ROC-кривая и точка, соответствующая заданному пороговому значению. Несложно догадаться, что координата x точки соответствует площади под кривой FPR, координата y — площади под кривой TPR. Увеличение порогового значения влечёт за собой смещение точки влево, уменьшение — вправо.

Чем лучше сам бинарный классификатор, тем ближе к левому верхнему углу будет проходить соответствующая ему ROC-кривая, поскольку в этом случае высокое значение TPR будет сочетаться с низким значением FPR. Бинарный классификатор, работающий также хорошо (на самом деле плохо) как алгоритм угадывания с помощью подбрасывания монеты (в том случае, если монета является «честной»),

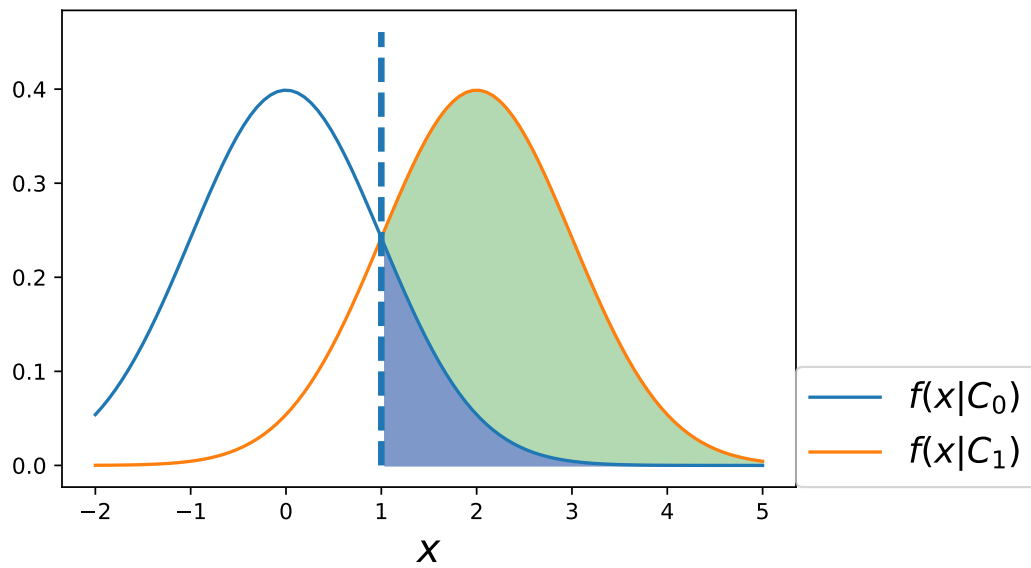


Рис. 3.3. Диаграмма плотностей распределения вероятностей TPR и FPR при пороговом значении 1.

даёт ROC-кривую, представляющую собой прямой отрезок между точками (0,0) и (1,1). В таком случае левая часть диаграммы покажет полное совпадение кривых функций плотности вероятностей TPR, FPR. Такой случай показан на диаграмме 3.4. Для самостоятельной практики можно использовать скрипт 3.2, запуская его в среде Jupyter Lab, позволяющей использовать интерактивные возможности браузера.

Листинг 3.2. Построение интерактивной диаграммы плотности распределения TPR и FPR и соответствующей ей ROC кривой для заданного порогового значения

```

# Import Libraries
%matplotlib inline
from ipywidgets import interact
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

# Plot
f0 = stats.norm(0, 1)
f1 = stats.norm(2, 1)
fig, ax = plt.subplots()
xi = np.linspace(-2, 5, 100)
ax.plot(xi, f0.pdf(xi), label=r'$f(x|C_0)$')
ax.plot(xi, f1.pdf(xi), label=r'$f(x|C_1)$')
ax.legend(fontsize=16, loc=(1, 0))
ax.set_xlabel(r'$x$', fontsize=18)
ax.vlines(0, 0, ax.axis()[-1] * 1.1, linestyle='--', lw=3.)
ax.fill_between(xi, f1.pdf(xi), where=xi > 0, alpha=.3, color='g')
ax.fill_between(xi, f0.pdf(xi), where=xi > 0, alpha=.3, color='b')

# Plot ROC-curve and make all interactive
def plot_roc_interact(c=0):
    xi = np.linspace(-3,5,100)
    fig,axs = plt.subplots(1,2)
    fig.set_size_inches((10,3))
    ax = axs[0]
    ax.plot(xi,f0.pdf(xi),label=r'$f(x|C_0)$')
    ax.plot(xi,f1.pdf(xi),label=r'$f(x|C_1)$')
    ax.set_xlabel(r'$x$',fontsize=18)
    ax.vlines(c,0,ax.axis()[-1]*1.1,linestyle='--',lw=3.)
    ax.fill_between(xi,f1.pdf(xi),where=xi>c,alpha=.3,color='g')
    ax.fill_between(xi,f0.pdf(xi),where=xi>c,alpha=.3,color='b')
    ax.axis(xmin=-3,xmax=5)
    crange = np.linspace(-3,5,50)
    ax=axs[1]
    ax.plot(1-f0.cdf(crange),1-f1.cdf(crange))
    ax.plot(1-f0.cdf(c),1-f1.cdf(c),'o',ms=15.)
    ax.set_xlabel('False-alarm probability')
    ax.set_ylabel('Detection probability')

interact(plot_roc_interact,c=(-3,5,.05))
%
```

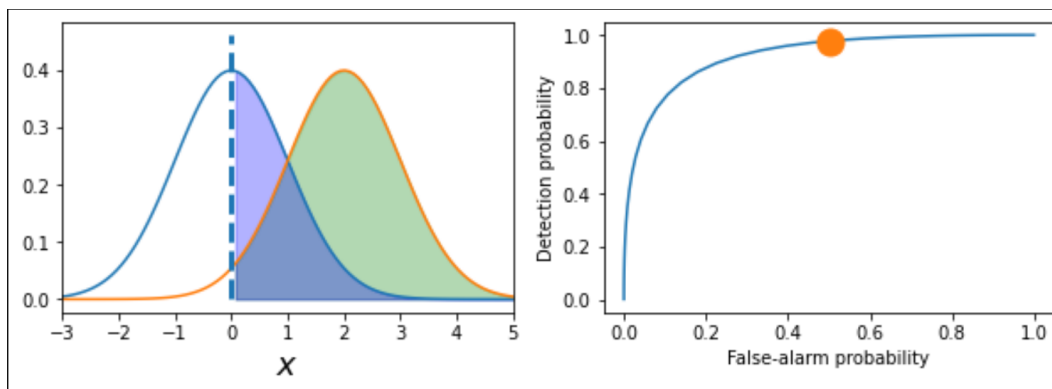


Рис. 3.4. Диаграмма плотностей распределения вероятностей TPR и FPR при пороговом значении 0.

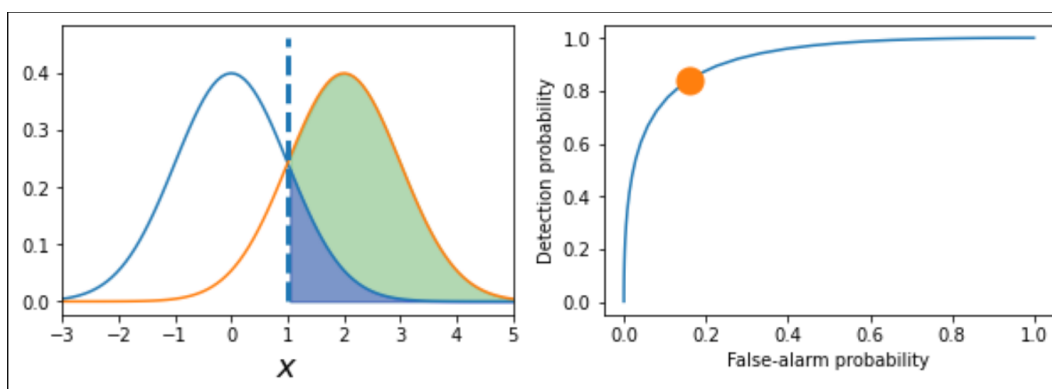


Рис. 3.5. Диаграмма плотностей распределения вероятностей TPR и FPR при пороговом значении 1.

3.4.2. Понятие AUC и её вычисление

Как следует из названия, AUC представляет собой площадь под ROC-кривой, ограниченную точкой, соответствующую заданному пороговому значению. В нормированном пространстве, в котором обычно и строится ROC-кривая, значение AUC эквивалентно вероятности того, что классификатор присвоит больший вес случайно выбранной положительной сущности, чем случайно выбранной отрицательной. AUC не зависит от конкретного порогового значения, поскольку ROC-кривая строится путем его перебора. Это означает, что AUC вычисляется путём интегрирования по пороговым значениям. AUC задаётся выражением:

$$AUC = \int P_{TPR}(P_{FPR})dP_{FPR}. \quad (3.30)$$

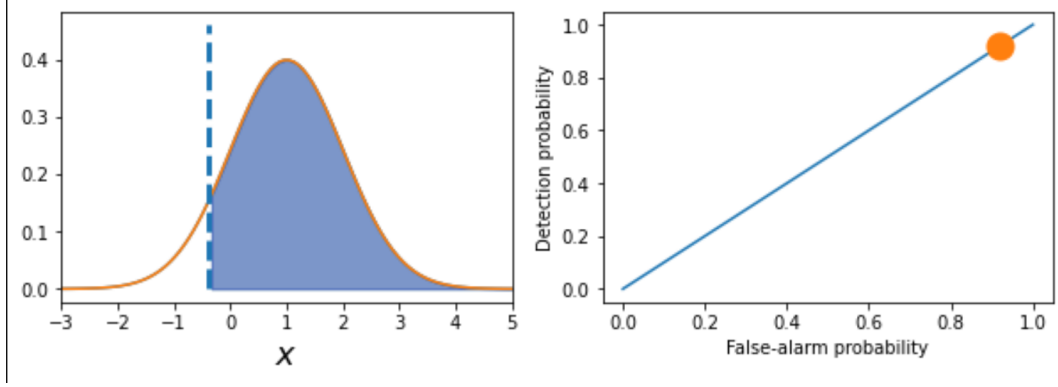


Рис. 3.6. Диаграмма плотностей распределения вероятностей TPR и FPR при равном среднем.

Пошаговый расчёт AUC выполняется следующим образом.

$$P_{TPR}(c) = 1 - F_1(c), \quad (3.31)$$

где F_1 — кумулятивная функция плотности для C_1 . Аналогичным образом вычисляется

$$P_{FPR}(c) = 1 - F_0(c), \quad (3.32)$$

где F_0 — кумулятивная функция плотности для C_0 .

Возьмём некоторое конкретное значение c^* , которому соответствует определённая $P_{FPR}(c^*)$. Иными словами, ему соответствует вероятность того, что случайный элемент x_0 , принадлежащий классу C_0 больше порогового значения c^* , т. е.

$$P_{FPR}(c^*) = \mathbb{P}(x_0 > c^* | x_0 \in C_0). \quad (3.33)$$

Тогда, рассуждая аналогичным образом относительно TPR, получим

$$P_{TPR}(c^*) = \mathbb{P}(x_1 > c^* | x_1 \in C_1). \quad (3.34)$$

Далее, опираясь на то, что AUC реализуется через интеграл, подбираем его значение таким образом, что распределение c^* соответствует распределению F_0 . В таком случае P_{TPR} является самостоятельной случайной величиной, с соответствующим ожиданием в виде

$$\mathbb{E}(P_{TPR}) = \int P_{TPR} dP_{FPR} = AUC. \quad (3.35)$$

Теперь возможно сформулировать определение для AUC.

AUC — ожидаемая вероятность того, что элемент $x_1 \in C_1$ будет отнесён к C_1 с большей вероятностью, чем элемент $x_0 \in C_0$. Таким образом,

$$1 - F_1(t) > 1 - F_0(t) \forall t. \quad (3.36)$$

Формулировка «для любых t » означает, что $1 - F_1(t)$ *стохастически* больше $1 - F_0(t)$. Последнее обстоятельство является ключевым с точки зрения связи AUC с U-тестом, которая будет показана далее.

Листинг 3.3. Вычисление р-значения для тестовых данных

```
print('p-value:', stats.wilcoxon(f1.rvs(30), f0.rvs(30))[1])
```

3.4.3. Связь между U-тестом и AUC

Ранее было приведено достаточно подробное описание U-теста. Данный параграф содержит только краткие сведения о нём, имеющие непосредственное отношение к вопросу его связи с AUC.

U-тест представляет собой непараметрический тест, позволяющий проверить принадлежность двух выборок одному распределению. Его основная идея заключается в том, что если между двумя классами отсутствует различие, то их объединение в один больший класс (множество) и последующее вычисление статистики (любой) для нового большего класса даст несмещённую оценку для любого из начальных классов. Иными словами, в случае отсутствия разницы в распределении у двух выборок, их объединение и предположение о том, что реально наблюдаемые данные двух выборок представляют собой лишь один из равнозначных вариантов перемещения наблюдений, означают отсутствие различия любой статистической оценки для любого варианта перемещения относительно другого, а также относительно объединённого множества.

Предположим, что нам нужно сравнить выборки посредством медианы, среднего или какой-либо иной меры центральной тенденции. С точки зрения кумулятивных функций распределения для двух популяций, в случае H_0 мы имеем следующее:

$$H_0 : F_X(t) = F_Y(t), \quad \forall t, \quad (3.37)$$

что указывает на то, что все наблюдения принадлежат одному распределению. Тогда альтернативная гипотеза заключается в том, что

$$H_1 : F_X(t) < F_Y(t), \quad \forall t, \quad (3.38)$$

что возможно, в частности, в случае существования *сдвига* одного распределения относительно другого. В этом случае выборки $X_{i=1}^n, X_{j=1}^m$ представляют собой независимые группы наблюдений. При этом размер выборок может отличаться.

Методика теста заключается в объединении двух выборок в одно множество и присвоения рангов каждому элементу внутри него. U-статистика представляет собой сумму рангов для множества X . Если значение статистики достаточно мало, это означает, что распределение множества X стохастически смещено влево относительно распределения множества Y , т.е. $F_X t < F_Y t$.

Поскольку при достаточно большом числе наблюдений (20 и более) распределение U-статистики хорошо аппроксимируется нормальным распределением, для оценки значимости подходит р-значение. Вычислим его с помощью языка Python согласно скрипту 3.3. р-значение составило $1.9729484515803686e - 05$, что меньше уровня значимости 0.05, вследствие чего мы можем отклонить нулевую гипотезу 3.37. Поскольку данные были сгенерированы случайным образом, в случае повторения

эксперимента, конкретное p -значение будет отличаться от полученного при написании данной работы. Однако оно всегда будет ниже порогового значения в силу заданных в алгоритме параметров.

U-статистика, в частности, может быть записана следующим образом:

$$U = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \mathbb{1}(Y_j > X_i), \quad (3.39)$$

где $\mathbb{1}(Y_j > X_i)$ — характеристическая функция, показывающая, что статистика (для дискретного случая) оценивает вероятность того, что Y стохастически больше X . Таким образом, это соответствие означает, что её значение равно AUC. Связь между AUC и U-тестом заключается в схожем смысле: проверке стохастического превышения значения наблюдений, принадлежащих одной выборке, относительно наблюдений, принадлежащих другой выборке.

3.4.4. Практика ROC-анализа и вычисления AUC

Данная подсекция не является обязательной к прочтению в случае, если целью является только практическое проведение самого U-теста. Однако она даёт представление о методах машинного обучения, не относящихся к т. н. *частотной статистике*, к которой относится сам U-тест, и показывает взаимосвязь между этими направлениями анализа данных. Кроме того, в ней будут даны достаточные знания для проведения ROC-анализа как такового, что может быть полезно в других ситуациях, с которыми может столкнуться оценщик в своей практике.

3.4.4.1. Построение ROC-кривой

ROC-анализ и в частности построение ROC-кривых широко используются для нахождения компромисса между *чувствительностью* (*sensitivity*) и *специфичностью* (*specificity*) бинарного классификатора. Большинство классификаторов, используемых в машинном обучении, дают результат в виде количественной оценки того, что данный объект имеет «положительное» значение признака. Для преобразования такой количественной оценки в конкретное предсказание «да» либо «нет» необходимо некоторое *пороговое значение* (*threshold*). В таком случае наблюдения, имеющие оценку выше этого порога, будут классифицированы «как положительные», ниже — как «отрицательные». Разные пороговые значения обеспечивают разные уровни чувствительности и специфичности. Установление относительно высокого порогового значения обеспечивает консервативный подход в вопросе отнесения конкретного случая к «положительным», что уменьшает вероятность возникновения ложно положительных результатов, вместе с тем увеличивает риск пропуска наблюдаемых положительных значений, т. е. снижает уровень истинно положительных результатов классификации. Относительно низкое пороговое значение обеспечивает более либеральный подход к отнесению наблюдений к «положительным», что снижает специфичность (увеличивает число ложно отрицательных результатов) и повышает

Листинг 3.4. Создание функции для расчёта TPR и FPR

```
# create own function for ROC
appraiserRoc <- function(labels, scores){
  labels <- labels[order(scores, decreasing=TRUE)]
  data.frame(TPR=cumsum(labels)/sum(labels),
    FPR=cumsum(!labels)/sum(!labels), labels)
}
```

чувствительность (увеличивает число истинно положительных результатов). ROC-кривая показывает соотношение истинно положительных и ложно положительных результатов, давая представление обо всём спектре таких «компромиссов». Существует множество библиотек языка R, строящих ROC-кривые и вычисляющих метрики для ROC-анализа. В данном случае, для лучшего понимания сути ROC-анализа некоторые действия будут выполнены с помощью написания собственных функций. Далее будет показан алгоритм построения ROC-кривой на основе набора реальных исходов и соответствующих им оценок. Расчёт включает два этапа:

- сортировка по убыванию наблюдаемых исходов по их предсказанным оценкам;
- расчёт суммарных показателей истинно положительных (*TPR*) и истинно отрицательных (*TNR*) результатов для упорядоченных наблюдаемых исходов.

Создадим соответствующую функцию (скрипт 3.4) Данная функция принимает два входных параметра:

- *labels* — булевой вектор, содержащий данные о фактической классификации, и
- *scores* — вектор вещественных чисел, содержащий данные об оценках, предсказанных некоторым классификатором.

Поскольку возможны только два исхода классификации, вектор *labels* может содержать только значения *TRUE* и *FALSE* (либо 1 и 0 в зависимости от предпочтения аналитика). Последовательность таких бинарных значений можно интерпретировать как набор инструкций для черепашьей графики [55] с учётом одной важной особенности: в данном случае черепаха имеет компас и получает инструкции абсолютных направлений движения: «на север» либо «на восток» вместо относительных «направо» и «налево». Черепаха начинает своё движение из начальной точки с координатами (0;0) и прокладывает путь на плоскости в соответствии с последовательностью инструкций. При получении команды *TRUE* (1) она делает один шаг на север, т. е. в положительном направлении по оси *y*, при получении команды *FALSE* (0) — на восток, т. е. в положительном направлении по оси *x*. Длина шагов подбирается таким образом, что в случае последовательного получения всех команд *TRUE* (1), черепаха окажется в точке с координатами (0;1), всех команд *FALSE* (0) — в точке с координатами (1;0), таким образом, длина шага «на север» может отличаться

от длины шага «на восток». Путь на плоскости определяется порядком следования команд *TRUE* (1) и *FALSE* (0) и всегда заканчивается в точке с координатами (1;1).

Продвижение черепашки по битам строки инструкций представляет собой настройку порога классификации на всё менее и менее строгий. Как только черепаха прошла бит, это означает, что она решила классифицировать этот бит как «положительный». Если этот бит на самом деле был «положительным», это означает истинно положительный результат, если он на самом деле был «отрицательным» — ложно положительный. Ось *y* показывает уровень истинно положительных результатов (TPR), вычисляемый как отношение числа выявленных на данный момент положительных результатов к общему числу фактических положительных результатов. Ось *x* показывает уровень ложно положительных результатов (FPR), вычисляемый как отношение числа выявленных на данный момент положительных результатов к общему числу фактических отрицательных результатов. Векторизованная реализация этой логики использует кумулятивные суммы (функция **cumsum**) вместо того, чтобы перебирать значения по одному, хотя именно это и делает компьютер на более низком уровне.

Вычисленная таким образом «кривая» ROC на самом деле является ступенчатой функцией. При наличии очень большого числа положительных и отрицательных случаев, эти шаги очень малы, и кривая выглядит гладкой. При этом, при действительно большом числе наблюдений, построение каждой точки является затруднительным, вследствие чего на практике большинство функций, предназначенных для построение ROC-кривых, используемых в практических целях, содержат дополнительные шаги и часто используют те или иные способы аппроксимации.

В качестве примера рассмотрим ситуацию, при которой оценщик осуществляет оценку деталей, изготовленных предприятием. Известно, что часть деталей сделана качественно, часть — является браком. Оценка качественных деталей осуществляется на основе затратного и сравнительного подходов в обычном порядке, бракованных — по скраповой стоимости. В этом случае необходимо отнести каждую деталь к той или иной категории. Существует некоторый измеряемый признак «*x*», а также некоторый неизмеряемый силами оценщика признак «*y*», значение которого позволяет отнести детали к качественным либо бракованным. Также известно, что между признаками *x* и *y* существует некоторая функция связи. Таким образом, зная значение *x* с некоторой вероятностью можно сделать вывод о значении *y*. В этом случае целесообразно провести выборочное исследование вместе со специалистами предприятия-заказчика с целью измерения параметров *x* и *y* для некоторой выборки.

Для рассмотрения примера используем смоделированные данные. Существует некий входной признак *x*, линейно связанный с неявным результатом *y*, при этом эта связь предполагает наличие некоторой случайности. Значение *y* показывает превышает ли деталь требования по допуску. Если да, её следует отнести к браку. Используемый алгоритм предполагает следующие шаги:

- создание функции «**sim_parts_data**», генерирующей данные по определённым правилам, и устанавливающей пороговое значение $y > 100$ для отнесения деталей к бракованным;

- создание датафрейма «**parts_data**» с помощью данной функции;
- создание правила «**test_set_idx**», согласно которому, 80 % данных случайным образом относятся к обучающей выборке, 20 % — к тестовой;
- применение правила «**test_set_idx**» к данным «**parts_data**»;
- создание тестовой «**test_set**» и обучающей «**training_set**» подвыборок.
- построение диаграммы, показывающей расположение наблюдений из обучающей выборки.

Для реализации вышеприведённого алгоритма был использован код из скрипта 3.5.

В результате была получена диаграмма 3.7. Как видно, если значение параметра x меньше 15, все точки красные, что означает брак деталей, выше 96 — зелёные, что означает исправность деталей. Между этими значениями находится область неопределённости, в правой части которой преобладают зелёные точки, в левой — красные.



Рис. 3.7. Диаграмма распределения деталей относительно параметра x .

Обучающая подвыборка будет использована для построения модели логистической регрессии, основанной на признаке x , позволяющей отнести конкретную деталь к качественным либо бракованным. Данная модель будет использоваться для присвоения оценок (scores) наблюдениям, попавшим в обучающую выборку, в дальнейшем

Листинг 3.5. Создание и первичная визуализация данных о качественных и бракованных деталях

```
# Sample of ROC-analysis

# enable libraries
library(ggplot2)
library(dplyr)
library(pROC)

#set seed
set.seed(19190709)

# create own function for ROC
appraiserRoc <- function(labels, scores){
  labels <- labels[order(scores, decreasing=TRUE)]
  data.frame(TPR=cumsum(labels)/sum(labels),
    FPR=cumsum(!labels)/sum(!labels), labels)
}

# create function
sim_parts_data <- function(N, noise=100){
  x <- runif(N, min=0, max=100)
  y <- 122 - x/2 + rnorm(N, sd=noise)
  bad_parts <- factor(y > 100)
  data.frame(x, y, bad_parts)
}

# create dataset
parts_data <- sim_parts_data(2000, 10)

# create rule for test subset
test_set_idx <- sample(1:nrow(parts_data),
  size=floor(nrow(parts_data)/4))

# create training and test subsets
test_set <- parts_data[test_set_idx,]
training_set <- parts_data[-test_set_idx,]

# plot graph
test_set %>%
  ggplot(aes(x=x, y=y, col=bad_parts)) +
  scale_color_manual(values=c("green", "red")) +
  geom_point() +
  ggtitle("Bad parts related to x")
```

Листинг 3.6. Сравнение предсказаний типов 'link' и 'response'

```

fit_glm <- glm(bad_parts ~ x, training_set,
family=binomial(link="logit"))

glm_link_scores <- predict(fit_glm, test_set, type="link")

glm_response_scores <- predict(fit_glm, test_set, type="response")

score_data <- data.frame(link=glm_link_scores,
response=glm_response_scores,
bad_parts=test_set$bad_parts,
stringsAsFactors=FALSE)

score_data %>%
ggplot(aes(x=link, y=response, col=bad_parts)) +
scale_color_manual(values=c("green", "red")) +
geom_point() +
geom_rug() +
ggtitle("Both link and response scores put cases in the same order")

```

эти оценки будут использованы для построения ROC-кривой совместно с истинными метками (labels) (напомним, что построение ROC-кривой осуществляется для наблюдений с заведомо известными значениями параметров x и y), используемой в дальнейшем для всего множества объектов, для которых известны значения x , но неизвестны значения y . Сами оценки (scores), как и значения x , y не отображаются на графике и используются только для сортировки меток. Два разных классификатора, сортирующих метки в одинаковом порядке дадут идентичные ROC-кривые независимо от абсолютных значений. В этом можно убедиться, построив ROC-кривую, основанную на предсказаниях «ответа» либо «связи» из модели логистической регрессии. Оценки «ответа» были отображены в шкалу (0, 1) с помощью сигмоиды[66], оценки «связи» оставлены без преобразований. При этом точки, показывающие конкретные наблюдения, расположены в одинаковом порядке. Для реализации проверки данной гипотезы используем код 3.6. Как видно из рисунка 3.8, порядок следования точек одинаков для 'link' и 'response'.

Перейдём непосредственно к построению ROC-кривой. Используем для этого как готовую функцию из пакета 'pROC', так и ранее созданную нами функцию **appraiserRoc** (см. скрипт 3.7) Результат работы первой представлен в виде оранжевой кривой, второй — в виде кружков красного (для бракованных деталей) и чёрного (для качественных деталей) цветов (см. диаграмму 3.9). Несложно догадаться, что красная точка соответствовала команде «на север», чёрная — «на восток». Поскольку библиотечная и собственная функции выполняют одни и те же действия, две кривые идентичны.

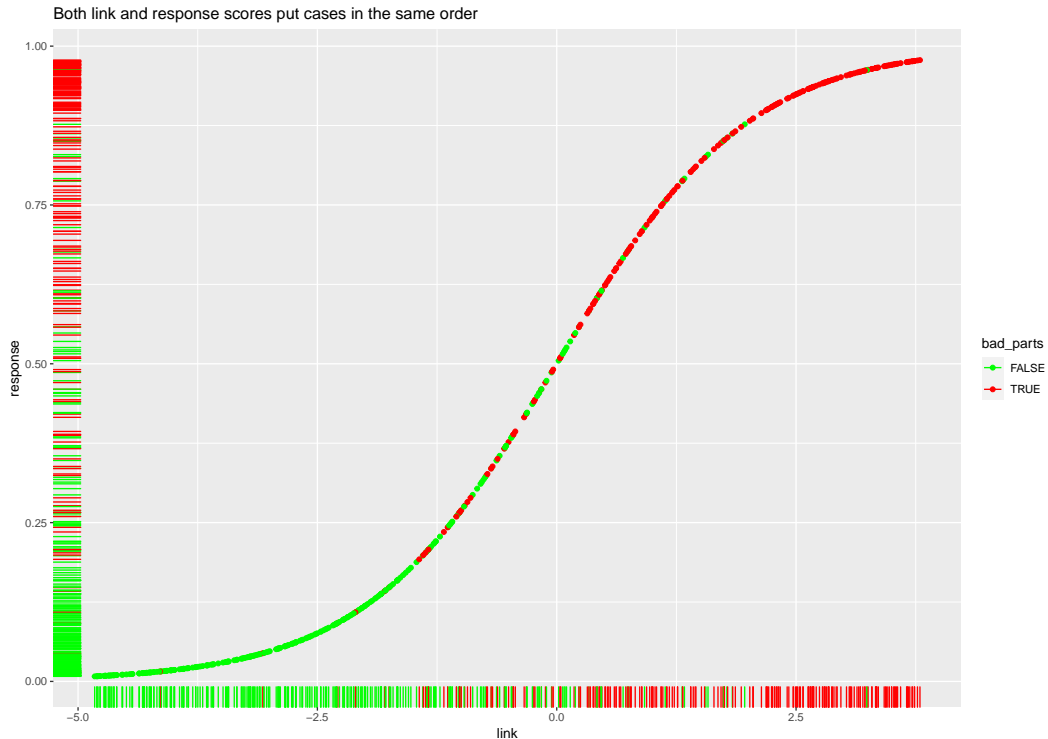


Рис. 3.8. Сравнение порядка расположения точек для link и response.

Следует обратить внимание, что по оси абсцисс отложена шкала 'Specificity', а не FPR, в результате чего значения инвертированы. Поскольку, согласно таблице 3.4 $\text{Specificity} = 1 - \text{FPR}$, можно говорить о взаимной однозначности этих показателей, а также о возможности использования любого из них при построении ROC-кривой. Данный вариант отображения шкалы был самостоятельно выбран функцией **roc** из библиотеки **pRoc**. В процессе своей работы, если пользователь не установит свои настройки, функция, основываясь на том, у какой группы (качественные детали, бракованные детали) выше медианный балл, выбирает отображение шкалы таким образом, чтобы значение AUC всегда было больше 0.5. Для возможности построения совместной диаграммы в процессе использования функции **appraiserRoc**, которая конечно же не настолько умная, было проведено простое вычитание.

Данный подход имеет одно ограничение, основываясь на прогностическом характере порядка расположения исходов, он не позволяет корректно обработать информацию в случае, если последовательность состоит из одинаковых оценок. «Черепаша» предполагает, что порядок меток имеет значение, но в ситуации одинаковых оценок нет никакого значимого порядка. Данные участки следует отображать диагональной линией, но не традиционными «шагами».

Рассмотрим пример, когда диагональ является единственно адекватным способ построения ROC-кривой. Для этого создадим крайне несбалансированный набор

Листинг 3.7. Построение ROC-кривой с помощью библиотечной и собственной функций

```
# plot ROC
plot(roc(test_set$bad_parts, glm_response_scores, direction='<'),
col='orange', lwd=3, main='The turtle finds its way', xlim = c(1, 0))
glm_simple_roc <- appraiser_roc(test_set$bad_parts=='TRUE',
glm_link_scores)
with(glm_simple_roc, points(1 - FPR, TPR, col=1 + labels))
```

Листинг 3.8. Построение ROC-кривой в случае отсутствия значения порядка оценок

```
# plot ROC for 99% negative cases
N <- 2000
P <- 0.01
rare_success <- sample(c(TRUE, FALSE), N, replace=TRUE, prob=c(P, 1-P))
guess_not <- rep(0, N)
plot(roc(rare_success, guess_not), print.auc=TRUE)
appr_roc <- appraiserRoc(rare_success, guess_not)
with(appr_roc, lines(1 - FPR, TPR, col='blue', lty=2))
```

данных, в котором только 1 % наблюдений является «положительным». В этом случае результат предсказания всегда будет отрицательным. Поскольку все оценки будут одинаковы, нет необходимости в каком-либо их упорядочивании, функция из пакета **pRoc** корректно распознаёт такие ситуации и рисует диагональную линию (1,0; 0,1). При этом «черепашка» предполагает, что порядок оценок имеет какое-то значение и перемещается между этими точками по случайной траектории, чередуя направления «на север» и «на восток». Код, вызывающий построение такой ROC-кривой, приведён в скрипте 3.8. На диаграмме 3.10 чёрная диагональная линия была построена библиотечной функцией, голубая прерывистая — собственной ранее написанной функцией **appraiserRoc**. Как видно, библиотечная функция корректно определила случай одинаковости оценок, собственная привела к случайным блужданиям «черепахи».

Чем больше значение N, тем ближе к диагонали будет блуждание черепахи. Бóльшая несбалансированность требует большего числа точек для того, чтобы путь проходил приблизительно рядом с диагональю.

В менее экстремальных случаях возникновение диагональных участков возможно, в частности, в случае округления оценок, приводящего к равенству некоторых из них.

Для дальнейшего ознакомления с темой построения ROC-кривых можно порекомендовать изучение данного теоретического материала [15], а также практику на онлайн-тренажёре [47].

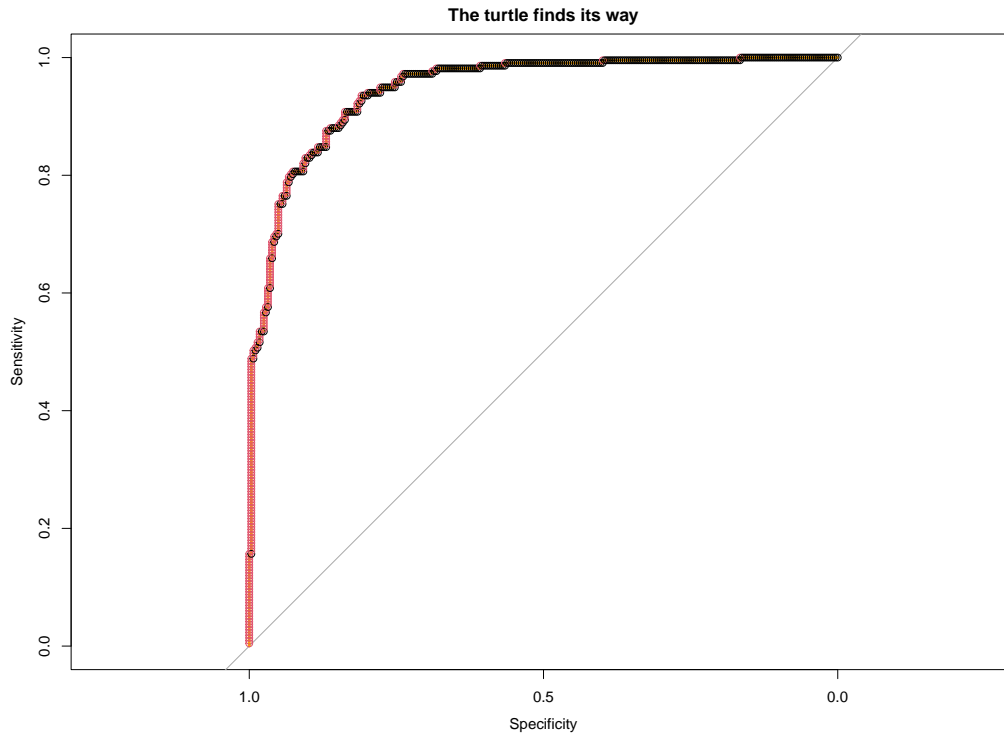


Рис. 3.9. Идентичные ROC-кривые, построенные с помощью библиотечной и собственной функций.

3.4.4.2. Понятие AUC и вычисление её значения

ROC-кривая является популярным средством визуализации компромисса между чувствительностью и специфичностью бинарного классификатора. Ранее в 3.4.4.1 вопрос построения ROC-кривой был рассмотрен с точки зрения шагов «черепахи», которая принимает вектор с инструкциями в виде шагов «на север», т. е. в положительном направлении по оси y и «на восток», т. е. в положительном направлении по оси x . При этом последовательность оценок, на основе которых формируется вектор битовых инструкций, упорядочена таким образом, что в начале идут случаи, наиболее вероятно являющиеся «положительными». При этом «черепаха» предполагает, что все случаи являются «положительными». Далее для обучающей выборки (в которой заведомо известны истинные значения) проводится определение был ли случай истинно (TP) либо ложно (FP) положительным. Размер шага по оси y обратно пропорционален числу положительных наблюдений, по оси x — отрицательных, таким образом, путь построения кривой всегда заканчивается в точке (1,1). В результате получается график зависимости частоты истинно положительных результатов (TPR или чувствительность) от частоты ложно положительных результатов (FPR или «1—специфичность»), являющийся ROC-кривой. При этом график сам по себе не даёт каких-либо количественных оценок качества бинарного

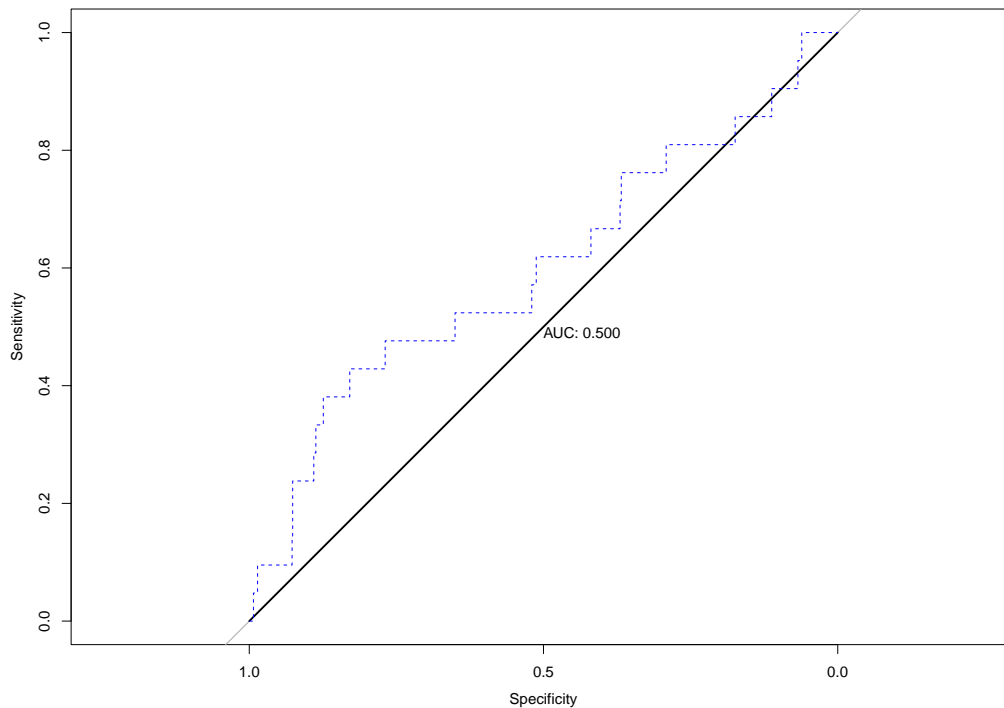


Рис. 3.10. ROC-кривая, возникающая в случае отсутствия значения порядка оценок.

классификатора.

Вычисление площади под ROC-кривой — это один из способов обобщения количественной оценки качества классификатора. Данная метрика настолько распространена, что в контексте области анализа данных термины «площадь под кривой» или «AUC» подразумевают именно площадь под ROC-кривой, если прямо не указано иное.

На первый взгляд наиболее простой и интуитивно понятной метрикой производительности классификатора является его точность. К сожалению, в ряде случаев такая метрика просто не будет работать. Например, в случае заболевания, встречающегося у одного человека на миллион, полностью бесполезный тест, который всегда показывает «отрицательный» результат, даст точность 99.9999%. В отличие от показателя точности ROC-кривые нечувствительны к несбалансированности классов, такой бесполезный тест будет иметь значение $AUC = 0.5$, что равносильно отсутствию теста вообще (см. диаграмму 3.10).

В данном разделе сначала будет рассмотрен геометрический подход к понятию AUC и разработана функция, вычисляющая её значение, далее мы перейдём к другой — вероятностной — интерпретации понятия AUC.

Листинг 3.9. Создание тестового набора данных

```
# Geometric interpretation of AUC

# activate libraries
library(pROC)

# create dataset
category <- c(1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0,
0)
prediction <- rev(seq_along(category))
prediction[9:10] <- mean(prediction[9:10])
```

3.4.4.2.1. Геометрический подход к понятию AUC Для начала создадим учебный набор данных с помощью кода, показанного в 3.9. Вектор «**prediction**» содержит псевдооценки, которые на практике присваиваются классификатором. В данной учебной задаче они представляют собой убывающую последовательность, что в целом соответствует меткам «category». Оценки для наблюдений с номерами 9 и 10, одна из которых представляет положительный случай, а другая — отрицательный, заменяются их средними значениями для создания эффекта связей.

Для построения ROC-кривой необходимо рассчитать TPR и FPR. В 3.4.4.1 было показано как это можно сделать в полуавтоматическом режиме посредством подсчёта кумулятивных сумм для положительных и отрицательных меток. В данном разделе будет использоваться библиотека **pRoc**, выполняющая все расчёты автоматически на низком уровне. Рассчитаем значения TPR, FPR и AUC с помощью кода 3.10. Также будет создан датафрейм, содержащий данные по каждому наблюдению, и показанный в таблице 3.5. Функция **roc** способа возвращать значения многих показателей, но на данном этапе нам требуются только TPR и FPR. Напомним, что TPR означает чувствительность, а FPR эквивалентен выражению «1 – специфичность». По умолчанию, функция **roc** возвращает значения в порядке возрастания, вследствие чего они были инвертированы для того, чтобы начальная точка имела координаты (0,0). Значение AUC, возвращённое функцией, составило 0.825. В дальнейшем мы сравним его с тем, которое будет получено в ходе его самостоятельного полуавтоматического расчёта.

3.4.4.2.1.1. Построение графика Если бы ROC-кривая была идеальной ступенчатой функцией, площадь под ней могла бы быть найдена путём добавления набора вертикальных полос шириной равной промежуткам между точками на оси абсцисс (FPR), и высотой равной высоте шага на оси ординат (TPR). Реальные ROC-кривые могут включать участки, соответствующие повторяющимся значениям, образующие сегменты, отличные от ступеней, необходим соответствующий учёт таких повторов. На рисунке 3.11 площадь обычных «шагов» показана зелёным цветом, случаи повторов обозначены синими прямоугольниками, разделёнными пополам

Листинг 3.10. Расчёт AUC с помощью библиотеки pRoc

```
# create ROC object&dataframe and calculate AUC
roc_obj <- roc(category, prediction)
auc(roc_obj)
roc_df <- data.frame(
  TPR=rev(roc_obj$sensitivities),
  FPR=rev(1 - roc_obj$specificities),
  labels=roc_obj$response,
  scores=roc_obj$predictor)

```

наклонными отрезками ROC-кривой. Таким образом, половина площади данных «шагов» включается в общую площадь под кривой.

Для построения диаграммы 3.11 были выполнены следующие шаги.

1) Определим функцию **rectangle**, принимающую в качестве аргументов:

- начальные координаты x , y ;
- ширину и высоту шага;
- угол поворота после каждого шага;
- плотность закрашки.

Предложенный в скрипте 3.11 алгоритм строит прямоугольник следующим образом:

- функция принимает от пользователя начальные координаты x , y ;
- далее она получает значение длины шага «на восток» (вправо по оси x), вычисляет новое значение координаты x , сохраняя значение координаты y ;
- после выполнения шага происходит поворот на 45° против часовой стрелки;
- затем она получает значение длины шага «на север» (вверх по оси y), вычисляет новое значение координаты y , сохраняя значение координаты x ;
- далее следует новый поворот влево на 45° и новые шаги в обратных направлениях;
- таким образом, выполнив по одному шагу «на восток», «на север», «на запад» и «на юг», а также три поворота на 45° против часовой стрелки функция возвращается к начальной точке, завершив построение прямоугольника.

2) Для вычисления шага «на восток» и «на запад» необходимо вычислить разность между соседними значениями FPR, «на север» и «на юг» — между соседними значениями TPR. Для этого добавим два столбца dFPR и dTPR соответственно с помощью кода, показанного в 3.12. Поскольку число пар, для которых вычисляется разница меньше числа наблюдений на единицу, в конце (поскольку данные датфрейма отсортированы по убыванию) следует добавить ноль.

Таблица 3.5. Показатели TPR, FPR, labels, scores для учебного датасета.

	TPR	FPR	labels	scores
1	0.00	0.00	1.00	20.00
2	0.10	0.00	1.00	19.00
3	0.20	0.00	1.00	18.00
4	0.30	0.00	1.00	17.00
5	0.40	0.00	0.00	16.00
6	0.40	0.10	1.00	15.00
7	0.50	0.10	1.00	14.00
8	0.60	0.10	0.00	13.00
9	0.60	0.20	1.00	11.50
10	0.70	0.30	0.00	11.50
11	0.80	0.30	1.00	10.00
12	0.80	0.40	0.00	9.00
13	0.90	0.40	1.00	8.00
14	0.90	0.50	0.00	7.00
15	0.90	0.60	0.00	6.00
16	1.00	0.60	1.00	5.00
17	1.00	0.70	0.00	4.00
18	1.00	0.80	0.00	3.00
19	1.00	0.90	0.00	2.00
20	1.00	1.00	0.00	1.00

Листинг 3.11. Создание функции **rectangle**

```
# create function for plotting rectangles
rectangle <- function(x, y, width, height, density=12, angle=45, ...)
  polygon(c(x,x+width,x+width,x), c(y,y,y+height,y+height),
    density=density, angle=angle, ...)
```

- 3) Далее осуществляется разметка координатной сетки от нуля до единицы по каждой оси с помощью кода 3.13.
- 4) Для случая повторяющихся значений предусмотрен особый вид шага «на северо-восток» — диагональная линия.
- 5) На заключительном этапе осуществляется построение самой ROC-кривой, а также прямоугольников, образующих площадь под ней, согласно скрипту 3.14. Функция **mapply** позволяет применить функцию **rectangle** последовательно к каждой строке датафрейма.

3.4.4.2.1.2. Суммирование площадей с помощью собственной функции Площадь под кривой (AUC) (выделена красным цветом) представляет собой сумму

Листинг 3.12. Добавление столбцов dFPR и dTPR

```
# add dFPR and dTPR columns
roc_df <- transform(roc_df,
dFPR = c(diff(FPR), 0),
dTPR = c(diff(TPR), 0))
```

Листинг 3.13. Построение пустого графа и разметка осей от 0 до 1

```
# plot empty graph from 0 to 1 for each axis
plot(0:10/10, 0:10/10, type='n', xlab='FPR', ylab='TPR',
     main = 'ROC-curve and rectangles under it')
abline(h=0:10/10, col='lightblue')
abline(v=0:10/10, col='lightblue')
```

Листинг 3.14. Построение ROC-кривой и прямоугольников под ней

```
# plot ROC-curve and rectangles under it
with(roc_df, {
  mapply(rectangle, x=FPR, y=0,
width=dFPR, height=TPR, col='green', lwd=2)
  mapply(rectangle, x=FPR, y=TPR,
width=dFPR, height=dTPR, col='blue', lwd=2)
  lines(FPR, TPR, type='b', lwd=3, col='red')
})
```

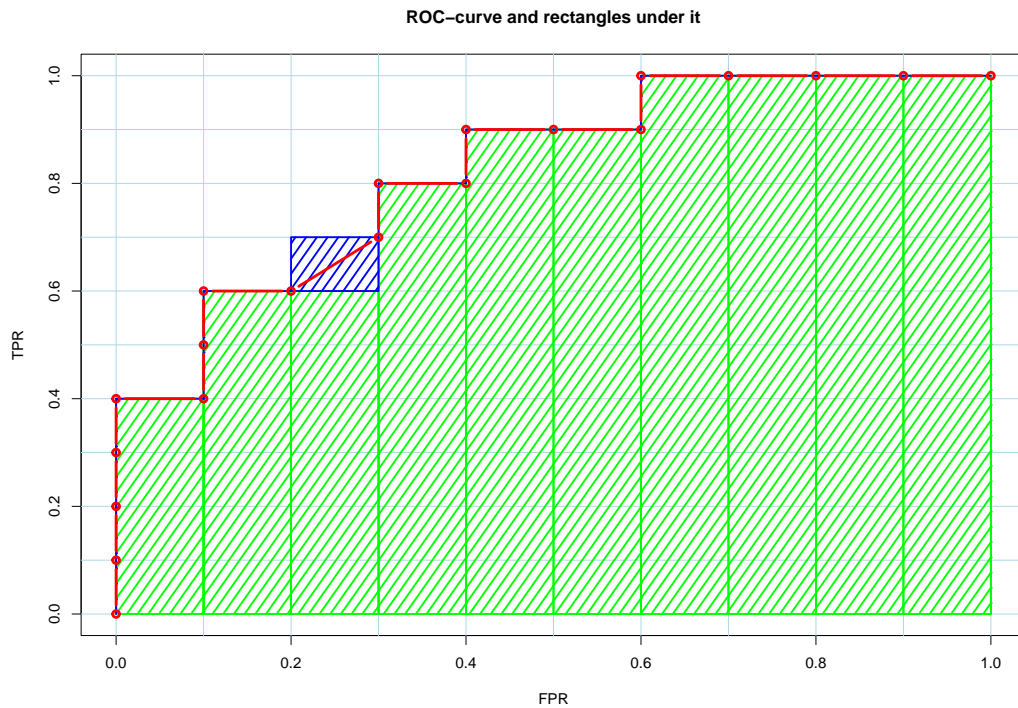


Рис. 3.11. ROC-кривая и площадь под ней с учётом наличия повторяющихся значений.

площадей всех зелёных прямоугольников и половины площади синего. Для вычисления площади каждого из прямоугольников нет необходимости знать абсолютные координаты его вершин, достаточно его ширины и высоты. Поскольку одна из сторон каждого прямоугольника лежит на оси x , высота любого из них определяется значением TPR, ширина — dFPR. Тогда общая площадь всех зелёных прямоугольников равна скалярному произведению [59] TPR и dFPR. Такой векторный подход вычисляет площадь для каждой точки данных, даже если ширина или высота равны нулю (в этом случае их дальнейшее включение в расчёт не имеет какого-либо значения). Площадь синих прямоугольников (при их наличии) определяется значениями dFPR и dTPR и также представляет собой их скалярное произведение как векторов. Для областей графика, содержащих шаги «на север» либо «на восток», одно из этих значений (dTPR, dFPR) будет равно нулю, вследствие чего возникновение синих прямоугольников возможно только в случае одновременного изменения TPR и FPR. При этом под кривой находится только половина такого прямоугольника.

Напомним, что ранее вычисленное значением AUC составило 0.825. Теперь рассчитаем его в полуавтоматическом режиме согласно алгоритму, описанному в предыдущем абзаце. Для этого сначала создадим функцию **appraiser_auc**, а затем применим её к тестовым данным (см. скрипт 3.15). Возвращённое значение составит 0.825, что говорит о корректности логики и алгоритма.

Листинг 3.15. Создание функции для вычисления AUC в полуавтоматическом режиме и её применение к тестовым данным

```
# create function for AUC calculation
appraiser_auc <- function(TPR, FPR){
# inputs already sorted, best scores first
dFPR <- c(diff(FPR), 0)
dTPR <- c(diff(TPR), 0)
sum(TPR * dFPR) + sum(dTPR * dFPR)/2
}

# apply function to data
with(roc_df, appraiser_auc(TPR, FPR))
```

3.4.4.2.2. Подход, основанный на сравнении рангов Для вычисления AUC также возможно использование принципиально иного подхода. Для его реализации необходимо создать матрицу, содержащую все возможные комбинации позитивных и негативных случаев. Каждая строка представляет собой позитивный случай, строки упорядочены таким образом, что верхняя строка содержит случай с наименьшей оценкой, нижняя — с наибольшей. Аналогичным образом, столбцы содержат негативные случаи, отсортированные таким образом, что левый столбец содержит наибольшие оценки. Тогда каждая ячейка представляет собой сравнение конкретного положительного случая с конкретным отрицательным. В случае, если оценка либо ранг положительного случая выше чем у отрицательного такая ячейка принимает значение TRUE. В случае достаточно хорошего классификатора, большинство положительных случаев будут иметь более высокие оценки (ранги), чем отрицательные, а все исключения будут сосредоточены в левом верхнем углу, где расположены положительные случаи с низкими оценками и отрицательные с высокими. Скрипт 3.16 содержит код для реализации данного алгоритма и его визуализации, показанной на диаграмме 3.12.

Построение ROC-кривой при этом выполняется обычным образом, единственное отличие заключается в том, что она немного сдвигается и растягивается для того, чтобы координаты совпадали с углами ячеек матрицы. Данный способ построения ROC-кривой делает очевидным следующий факт: ROC-кривая представляет собой границу области, где положительные случаи имеют более высокие оценки (ранги) чем отрицательные. Таким образом, AUC может быть вычислена путём замены значений в матрице таким образом, что

- ячейки, в которых оценки (ранги) положительных случаев превосходят оценки (ранги) отрицательных, принимают значение 1;
- ячейки, в которых оценки (ранги) отрицательных случаев превосходят оценки (ранги) положительных, принимают значение 0;

Листинг 3.16. Создание функции для матрицы сравнений и её применение к тестовым данным

```
# create function for rank comparison
rank_comparison_auc <- function(labels, scores, plot_image=TRUE, ...){
  score_order <- order(scores, decreasing=TRUE)
  labels <- as.logical(labels[score_order])
  scores <- scores[score_order]
  pos_scores <- scores[labels]
  neg_scores <- scores[!labels]
  n_pos <- sum(labels)
  n_neg <- sum(!labels)
  M <- outer(sum(labels):1, 1:sum(!labels),
  function(i, j) (1 + sign(pos_scores[i] - neg_scores[j]))/2)

  AUC <- mean (M)
  if (plot_image){
    image(t(M[nrow(M):1,]), ...)
    library(pROC)
    with( roc(labels, scores),
    lines(((1 + 1/n_neg)*((1 - specificities) - 0.5/n_neg),
    (1 + 1/n_pos)*sensitivities - 0.5/n_pos,
    col="blue", lwd=2, type='b'))
    text(0.5, 0.5, sprintf("AUC = %0.4f", AUC))
  }

  return(AUC)
}

# apply function to data
rank_comparison_auc(labels=as.logical(category), scores=prediction)
```

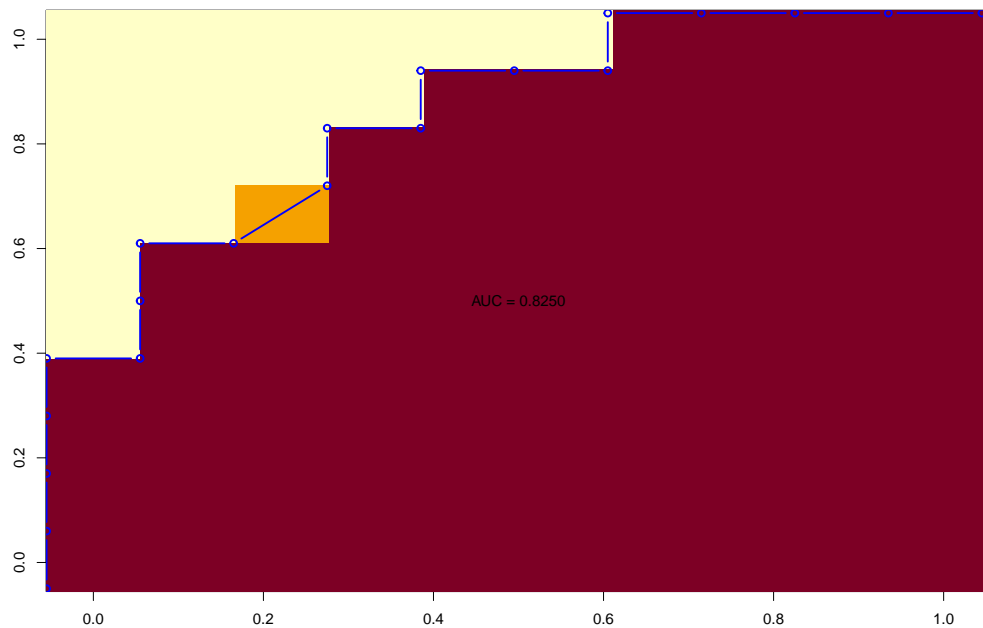


Рис. 3.12. Визуализация матрицы сравнений оценок (рангов).

- ячейки, в которых оценки (ранги) равны, принимают значение 0.5.

Поскольку применение функции знака **sign** приводит к получению одного из трёх возможных значений: -1, 0, 1, для помещения значений в требуемый диапазон используется следующий приём: мы прибавляем к таким значениям единицу и делим на два. Итоговый расчёт AUC осуществляется путём вычисления среднего значения.

3.4.4.2.3. Вероятностный подход к вычислению AUC Вероятностная интерпретация заключается в том, что если случайным образом выбрать положительный и отрицательный случай, то вероятность того, что значение оценки (ранга) положительного случая будет больше, чем у отрицательного, определяется AUC и равна ей. Это, в частности, следует из диаграммы 3.12, на которой общая площадь графика нормирована к единице, ячейки матрицы содержат сведения обо всех возможных комбинациях положительных и отрицательных случаев, а площадь под кривой состоит из ячеек, в которых оценки (ранги) положительных случаев превосходят таковые у отрицательных. Для аппроксимации AUC при вероятностном подходе создадим и применим функцию согласно скрипту 3.17. Возвращённое значение AUC составит 0.8248116, что приблизительно соответствует её точному значению, вычисленному ранее.

Листинг 3.17. Создание и применение функции для вычисления AUC как вероятности

```

# create function for calculation AUC as probability
auc_probability <- function(labels, scores, N=1e7){
  pos <- sample(scores[labels], N, replace=TRUE)
  neg <- sample(scores[!labels], N, replace=TRUE)
  # sum( (1 + sign(pos - neg))/2)/N # does the same thing
  (sum(pos > neg) + sum(pos == neg)/2) / N # give partial credit for ties
}

# apply function to data
auc_probability(as.logical(category), prediction)

```

3.4.4.3. Вычисление AUC для больших наборов данных

В предыдущих секциях мы работали с очень маленьким датасетом, позволяющим в практически ручном режиме перебрать данные. В данной секции мы будем работать с ранее созданным набором данных, содержащих сведения о качественных и бракованных деталях, требующих разных методов оценки. Следует напомнить, что в данном датасете вектор оценок содержится в переменной *glm_responce_scores*, а вектор меток — в переменной *bad_parts*.

Ранее созданный набор данных не содержит повторяющихся значений (связок), поэтому для более глубокого изучения вопроса эффект связок будет добавлен искусственно. После чего построим ROC-кривую на оригинальных данных, а затем на округлённых. Код, выполняющий вышеуказанные действия, приведён в скрипте 3.18. Для создания эффекта связок был создан новый вектор оценок, содержащий их значения с округлением до одного знака после точки. Результат построения двух ROC-кривых представлен на диаграмме 3.13. Чёрная линия соответствует ROC-кривой, построенной на оригинальных данных, содержащих уникальные оценки для каждого наблюдения, красная — построенной на округлённых данных. Поскольку в качестве входного вектора использовали данные переменной *glm_response_scores*, принимающие значение от 0 до 1, оценки, округлённые до одного знака после точки, имеют всего одиннадцать возможных значений от 0.0 до 1.0. Значение AUC для обоих случаев выведено прямо на диаграмму. Как видно, значения приблизительно равны вследствие достаточно большого числа наблюдений.

Далее попробуем применить стандартную, а также три собственные функции для вычисления AUC. А затем построим две ROC-кривые на основе матриц сравнения рангов для оригинальных и округлённых оценок. Код для применения четырёх функций для вычисления AUC, а также построения ROC-кривых на основе матриц сравнения, приведён в скрипте 3.19.

Диаграмма 3.14 содержит ROC-кривую, построенную на оригинальных данных, диаграмма 3.15 содержит ROC-кривую, построенную на округлённых. Как видно, каждый её участок представляет собой диагональный отрезок, направленный

Листинг 3.18. Построение ROC-кривой на оригинальных (чёрная линия) и округлённых (красная линия) данных о бракованных запчастях

```
# create function for calculation AUC as probability
auc_probability <- function(labels, scores, N=1e7){
  pos <- sample(scores[labels], N, replace=TRUE)
  neg <- sample(scores[!labels], N, replace=TRUE)
  # sum( (1 + sign(pos - neg))/2)/N # does the same thing
  (sum(pos > neg) + sum(pos == neg)/2) / N # give partial credit for ties
}

# apply function to data
auc_probability(as.logical(category), prediction)
```

Листинг 3.19. Применение четырёх функций для вычисления AUC и построение ROC-кривых на основе матриц сравнения для оригинальных и округлённых оценок

```
# apply all functions to data
results <- data.frame(
  'Full Resolution' = c(
    auc = as.numeric(auc(roc_full_resolution)),
    appraiser_auc = appraiser_auc(rev(roc_full_resolution$sensitivities),
      rev(1 - roc_full_resolution$specificities)),
    rank_comparison_auc = rank_comparison_auc(test_set$bad_parts,
      glm_response_scores,
    main='Full-resolution scores (no ties)'),
    auc_probability = auc_probability(as.logical(test_set$bad_parts),
      glm_response_scores)
  ),
  'Rounded Scores' = c(
    auc = as.numeric(auc(roc_rounded)),
    appraiser_auc = appraiser_auc(rev(roc_rounded$sensitivities),
      rev(1 - roc_rounded$specificities)),
    rank_comparison_auc = rank_comparison_auc(test_set$bad_parts,
      rounded_scores,
    main='Rounded scores (ties in all segments)'),
    auc_probability = auc_probability(as.logical(test_set$bad_parts),
      rounded_scores)
  )
)
```

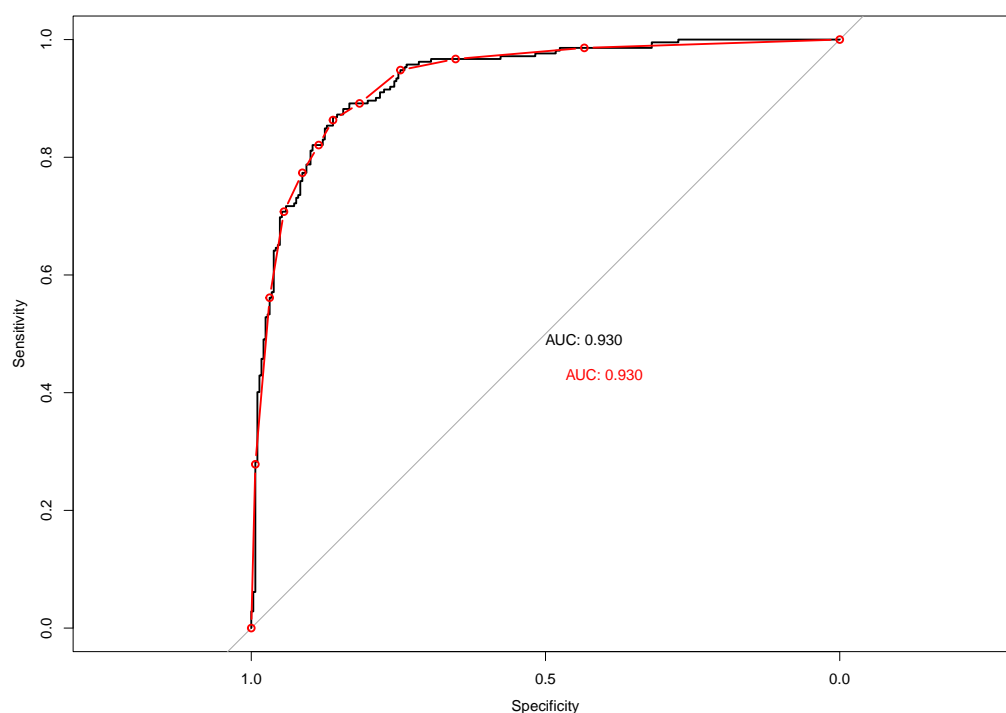


Рис. 3.13. ROC-кривая для оригинальных (чёрная линия) и округлённых (красная линия) данных о бракованных запчастях.

на «северо-восток», что, как было показано ранее, соответствует повторяющимся значениям.

Итоговые результаты расчёта AUC четырьмя методами показаны в таблице 3.6. Как видно, все три собственные функции достаточно хорошо вычисляют AUC и дают близкие результаты относительно функции из пакета **pROC**. Естественно, данные функции служат больше для учебных целей. В практической деятельности следует использовать штатные функции **pROC** либо **ROCR**.

Таблица 3.6. Результаты вычисления AUC четырьмя методами.

	Full.Resolution	Rounded.Scores
auc	0.9302279874213836841079	0.9301379061844864404307
appraiser_auc	0.9302279874213836841079	0.9301379061844863294084
rank_comparison_auc	0.9302279874213835730856	0.9301379061844864404307
auc_probability	0.930719999999999917577	0.9291749999999999731770

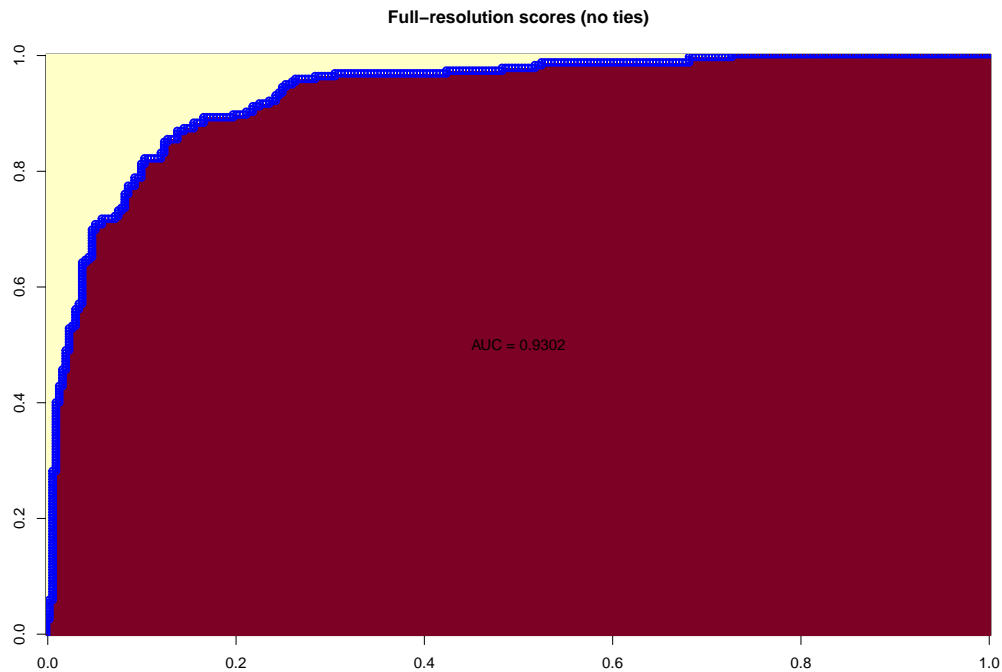


Рис. 3.14. ROC-кривая для оригинальных данных о бракованных запчастях, построенная на основе матрицы сравнений.

3.4.4.4. Обобщение выводов по AUC

В предыдущих секциях был проведён анализ понятия AUC, приведены алгоритмы её расчёта, в т. ч. основанные на вероятностном подходе. В целом можно сказать, что глубокое изучение вопросов, связанных с ROC и AUC, не является строго обязательным для успешного применения U-теста в практической деятельности. Если геометрический подход к вычислению AUC ещё достаточно интуитивен и прост, то интерпретация AUC как вероятности представляется несколько заумной и требует определённого уровня общей математической подготовки.

Однако в контексте оценочной деятельности концепция AUC как вероятности действительно достаточно полезна, поскольку оценщики часто имеют дело с относительно небольшими выборками и им бывает необходимо понимать, в какой степени они могут быть уверены, что стоимость объекта обладающего некоторым признаком, превосходит стоимость объекта, не обладающего им. Таким образом, применение AUC отсылает нас к *байесовскому подходу* к вероятности, оперирующему в т. ч. таким понятием как *степень уверенности*, что, в свою очередь, наводит мост между традиционной *частотной статистикой*, в рамках которой когда-то и был разработан U-тест, и современным машинным обучением, основанным по большей части на *байесовском подходе*. Одно это обстоятельство, по мнению автора, слу-

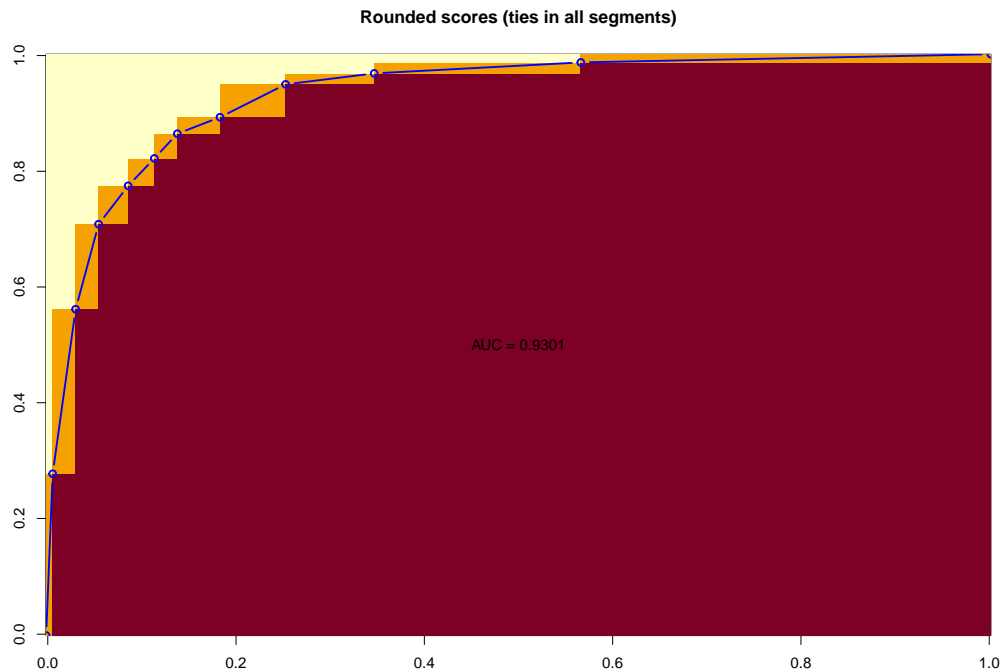


Рис. 3.15. ROC-кривая для округлённых данных о бракованных запчастях, построенная на основе матрицы сравнений.

жит достаточным основанием для того, чтобы погрузиться в темы ROC и AUC при изучении практических вопросов применения U-теста оценщиками.

Кроме того, задачи оценщиков не сводятся только к определению самой стоимости. Зачастую на более раннем этапе проведения оценки, например на стадии установления свойств оцениваемых объектов, применение классификаторов может иметь самостоятельную ценность. При этом достаточно часто общее поведение классификатора на всём диапазоне возможных пороговых значений представляет меньший интерес, чем его поведение в определённом относительно узком диапазоне. Например, при решении рассмотренной выше задачи отделения бракованных деталей от качественных, решающее значение имеет качество работы классификатора в пограничной зоне пороговых значений. При этом, конкретная настройка такого значения может зависеть от целей проведения оценки. Например, с точки зрения залоговой службы банка, кредитующего предприятие, либо Центрального Банка, осуществляющего мониторинг достаточности залогов, в определённом смысле чувствительность классификатора важнее его специфичности, поскольку отнесение бракованных деталей к качественным несёт больший риск, чем обратная ситуация. В то же время, с точки зрения, например, налоговой службы отнесение основных средств к неисправным с последующим их списанием, приводящим к снижению налогооблагаемой базы,

представляет более серьёзную проблему чем ошибочное отнесение неисправных к исправным, т. е. возникает ситуация, когда специфичность важнее чувствительности. Таким образом, возможны ситуации, когда цена ошибки первого и второго рода не одинакова. Такие ситуации представляют определённую проблему, поскольку концепция AUC предполагает одинаковую «цену» таких ошибок. Однако, данная тематика является достаточно специфической и выходит за рамки данного материала, темой которого является U-тест. Кратко можно сказать, что существуют иные, отличные от AUC метрики, позволяющие учесть разную «цену» ошибок первого и второго рода. Для дальнейшего самостоятельного изучения темы AUC можно порекомендовать данный материал [41].

Глава 4.

Практическая реализация

4.1. Реализация в табличном процессоре LibreOffice Calc

На данный момент можно с уверенностью сказать, что табличные процессоры являются стандартом для расчётов оценщиков. Проникновение средств разработки в профессиональную деятельность, например языков программирования Python и R, идёт достаточно медленно. Кроме того, самостоятельный поэтапный расчёт позволяет лучше понять методику U-теста. Поэтому было принято решение создать пошаговую инструкцию для проведения U-теста в электронной таблице. Для этого был использован программный продукт LibreOffice Calc (Version: 7.3.4.2 / LibreOffice Community Build ID: 30(Build:2); CPU threads: 4; OS: Linux 5.11; UI render: default; VCL: kf5 (cairo+xcb) Locale: en-US (en_US.UTF-8); UI: en-US Ubuntu package version: 1:7.3.4 rc2-0ubuntu0.20.04.1 lo1; Calc: threaded), существенная часть функционала которого имеется также и в наиболее распространённом приложении такого рода — Microsoft Excel. Отсутствуют основания полагать, что сделанные расчёты не будут корректно работать в приложениях, отличных от LibreOffice Calc. Однако гарантировать это также невозможно. Для однозначно корректного проведения теста рекомендуется использовать именно данное приложение, имеющее версии для всех основных операционных систем. Актуальная версия файла U-test.ods находится в репозитории вместе с остальными материалами данной работы.

Данные, рассматриваемые в данной подсекции, являются вымышленными и были созданы алгоритмом генерации псевдослучайных чисел LibreOffice Calc. Для повторной генерации необходимо использовать сочетание клавиш *ctrl+shift+F9*.

Рассмотрим учебную задачу. В ячейках I3:J30 содержатся данные значений некоторого количественного признака для двух выборок, взятых из множеств I и J соответственно. Различие между элементами этих множеств заключается в наличии некоторого признака у элементов множества I и его отсутствии у элементов множества J . Задача заключается в проверке гипотезы о том, что различие в данном признаке следует признать существенным, а сам признак является ценообразующим фактором. Выдвинем нулевую гипотезу, сформулировав её в трёх вариантах, соответствующих трём уровням строгости, описанным ранее в таблице 3.1. Следует

отметить, что U-тест входит в периметр т. н. *частотного подхода к вероятности* (о различиях между *частотным* и *байесовским* подходами к вероятности применительно к оценке стоимости можно прочесть, в частности в [22]). Как известно, частотный подход базируется на предпосылке о том, что случайность является следствием объективной неопределённости, которая может быть уменьшена только путём проведения серии экспериментов. В частотном подходе существует чёткое разделение на случайные и неслучайные параметры. Типичной задачей является оценка тех или иных параметров генеральной совокупности, представляющей собой набор случайных величин, на основе детерминированных параметров выборки, например: среднее, мода, дисперсия и т. д. Последние представляют собой конкретные значения, в которых уже нет никакой случайности. Таким образом, принимая фундаментальное предположение о случайном характере изучаемых величин, мы применяем те или иные методы математической статистики, позволяющие получить конкретные значения оценок параметров. Из этого следует, что нулевая гипотеза чаще всего «пессимистична», т. е. несёт утверждение, что в основе исследуемого явления или процесса лежит случайность, вследствие чего мы не имеем возможность делать надёжные выводы. С учётом всего вышесказанного, сформулируем нулевую и альтернативную гипотезы (таблица 4.1) в трёх вариантах, согласно уровням строгости, показанным в таблице 3.1. Ячейки C2:C19 содержат некоторые описательные статистики. Для удобства первичного анализа бывает полезно показать свойства выборок графически. На рисунке 4.1 изображена диаграмма «ящик с усами» (Box-plot), позволяющая сделать некоторые выводы на основе одного взгляда. Как видно, значения средних и медиан двух выборок различны. При этом также отличаются минимальные значения. При этом максимальное значение одинаково. Также следует обратить внимание, что несмотря на то, что среднее и медиана первой выборки превышают аналогичные показатели второй, минимальное значение первой меньше чем у второй. В таких условиях ещё сложнее сделать вывод о том, является ли различие в признаке существенным, или же разница в показателе стоимости носит случайный характер. Следующим подготовительным этапом является проверка нормальности распределения значений количественного признака (в данном случае условного показателя удельной стоимости). Существует ряд строгих тестов, позволяющих провести такую проверку численными методами. В подсекциях 4.2 и 4.3 будут показаны соответствующие способы проведения такого теста. В данном разделе ограничимся графическим способом. На рисунках 4.2, 4.3 изображены гистограммы распределения частот для первой и второй выборок соответственно, совмещённые с кривыми функции плотности вероятности для нормального распределения.

Как видно на диаграммах, форма распределения обеих выборок существенно отличается от формы кривой функции плотности вероятности нормального распределения. При работе с реальными данными в любом случае необходимо проводить количественные тесты проверки на нормальность распределений, однако на данном этапе остановимся на интерпретации диаграмм и сделаем вывод о том, что распределения обеих выборок отличаются от нормального, что позволяет сделать вывод о неприменимости параметрических методов статистического оценивания и необходимости использования непараметрических, к числу которых относится и U-тест.

Таблица 4.1. Нулевая и альтернативная гипотезы при анализе тестовых данных.

Тип гипотезы	Нулевая гипотеза (H0)	Альтернативная гипотеза (H1)
Научная	Распределение удельных показателей стоимости одинаково для объектов-аналогов, обладающих признаком «X» (множество объектов I), и не обладающих им (множество объектов J), сдвиг между ними отсутствует, статистические оценки, сделанные для одного множества объектов-аналогов, являются несмещёнными для другого.	Распределение удельных показателей стоимости для объектов из множества I отличается от распределения, имеющего место у множества J , существует сдвиг, оценка, сделанная для объектов, принадлежащих одному множеству, будет смещённой для объектов, принадлежащих другому.
Практическая	Медианное значение удельного показателя стоимости объектов, обладающих признаком «X», не отличается от медианного значения удельного показателя стоимости объектов, не обладающих признаком «X», — их медианы равны.	Медианное значение удельного показателя стоимости объектов, обладающих признаком «X», отличается от медианного значения удельного показателя стоимости объектов, не обладающих признаком «X», — их медианы не равны.
Изложенная в терминах оценки	Наличие или отсутствие признака «X» не оказывает сколько-нибудь заметного влияния на стоимость — признак «X» не является ценообразующим фактором.	Наличие или отсутствие признака «X» оказывает влияние на стоимость — признак «X» является ценообразующим фактором.

При работе с электронной таблицей отсутствует потребность в отдельном построении общего вариационного ряда для двух выборок. Вместо этого можно сразу перейти к вычислению рангов наблюдений. С учётом возможного наличия связок (повторяющихся значений) следует использовать функцию `RANK.AVG`, последовательно указав при этом три аргумента: наблюдение, для которого вычисляется ранг, диапазон всех значений общего вариационного ряда, тип сортировки: 0 — по убыванию, 1 — по возрастанию, в нашем случае необходимо указать 1. Столбцы L, N содержат дублирующие значения, столбцы M и O — ранги соответствующих наблюдений.

После этого проведём подсчёт сумм рангов для каждой из выборок в ячейках C20:C21. В ячейке C22 проведём подсчёт общей суммы рангов обеих выборок. Для проверки рассчитаем тот же показатель согласно формуле 3.5.

Далее в ячейках C25, C26 по формулам 3.6, 3.7 вычислим соответственно значения U_1 , U_2 . После чего проверим корректность контрольного соотношения 3.10 в ячейке D27. В C28 выбираем меньшее значение, которое и будет использоваться в дальнейшем в качестве U-статистики. В нашем случае меньшее значения U-статистики

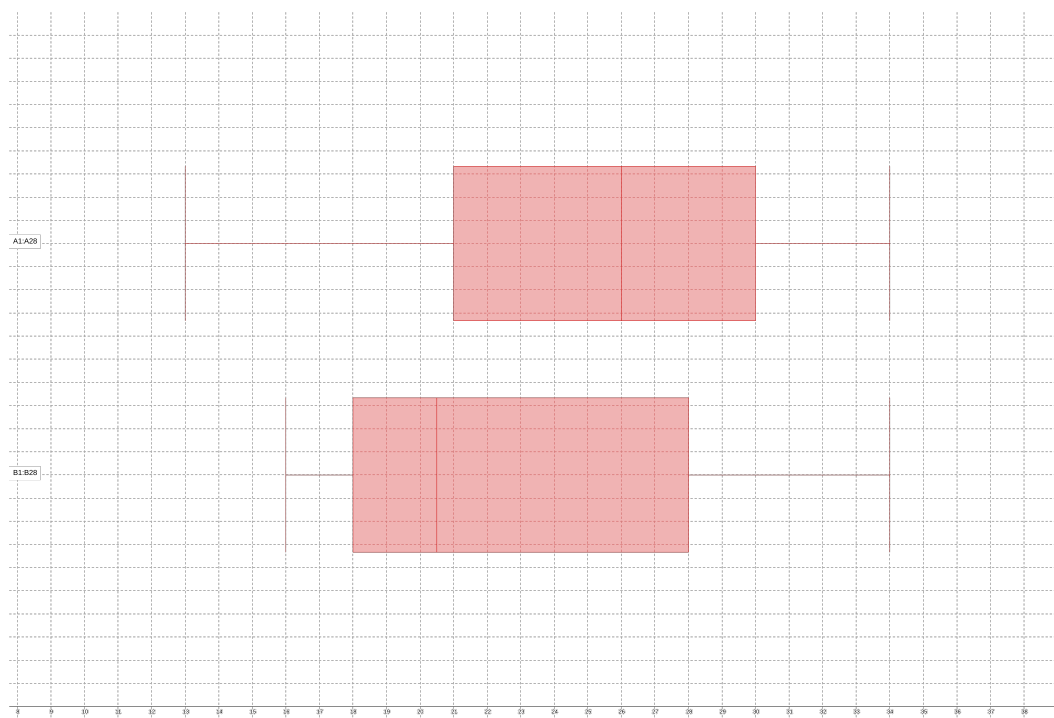


Рис. 4.1. Диаграмма «ящик с усами» (Boxplot) для обеих выборок.

у выборки из множества J .

Рассчитаем показатель CLES. Для этого используем формулу 3.11. Результат содержится в С29. В рассматриваемом примере значение показателя составляет 0.61, что следует интерпретировать следующим образом: «вероятность того, что значение показателя удельной стоимости случайно выбранного наблюдения из множества I превышает аналогичный показатель случайно выбранного наблюдения из множества J составляет 0.61 (61 %)».

Далее рассчитаем значение коэффициента рангово-бисериальной корреляции по формулам 3.14, 3.15, разместив его в ячейке С36. В рассматриваемом случае значение составило 0.21, что говорит о том, что сила корреляционной связи между наличием у объекта признака «Х» и удельным показателем его стоимости составляет 0.21.

После этого перейдём к расчёту стандартизированного значения согласно формуле 3.16. Для этого в ячейке С37 рассчитаем среднее по формуле 3.17, а затем перейдём к вопросу расчёта стандартного отклонения. Следует отметить, что для этого существуют две формулы: одна (3.18) применяется в случае отсутствия связей (ячейка С38), вторая (3.19) — при их наличии (ячейка С39). В рассматриваемом случае связи имели место. Их обработка осуществлялась в столбцах Р и Q, а также в ячейках Е35:Е49. В результате было получено два значения, отличие между которыми составило менее одного процента. Учёт фактора связей необходим с точки

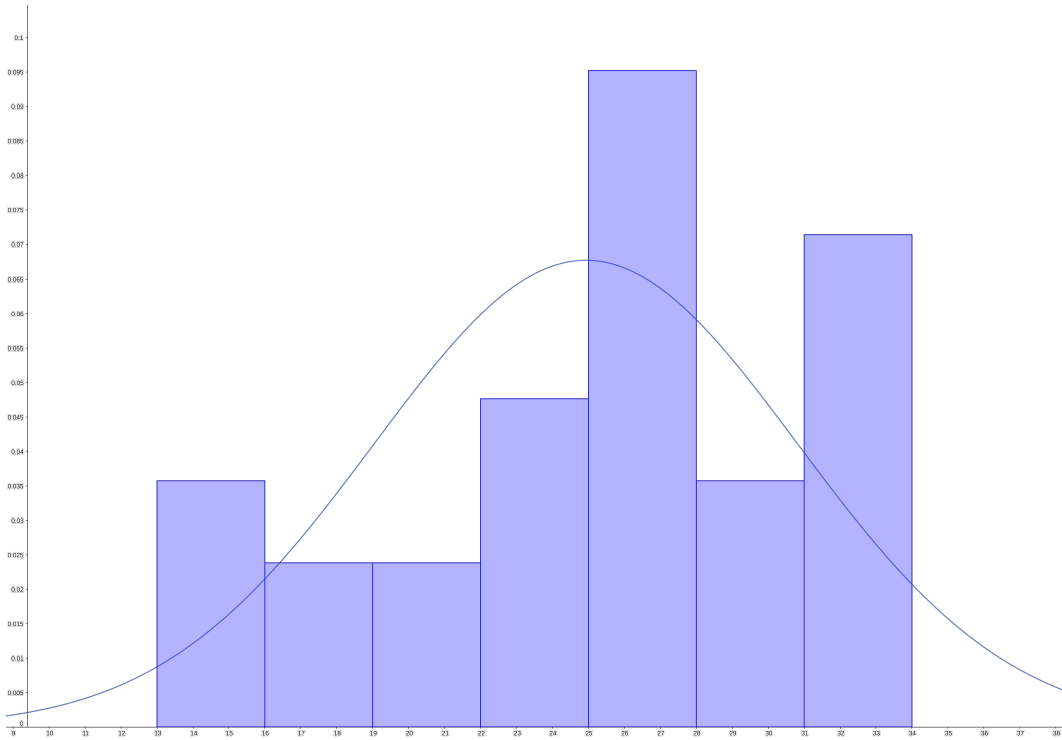


Рис. 4.2. Гистограмма первой выборки, совмещённая с кривой функции плотности вероятности для нормального распределения.

зрения максимальной научной корректности результата, однако в повседневной практической деятельности некоторые оценщики могут столкнуться со сложностями с корректным учётом фактора связей, а также не иметь достаточно времени для дополнительных расчётов. Практический опыт говорит о том, что сколько-нибудь существенное отличие значений стандартного отклонения, полученных с помощью формулы 3.19, от значений, полученных согласно 3.18, бывает в случаях большого числа связей, а также наличия крупных групп. В остальных ситуациях более простая формула, автоматически вычисляющая показатель σ , даёт корректный результат, достаточный для практического применения в оценке. В любом случае, решение об использовании строгих либо простых методов принимает сам оценщик. В рассматриваемом примере учёт фактора связей был осуществлён.

Зная среднее арифметическое и стандартное отклонение, вычисляем z-метку в ячейке С44 (поскольку одной из предпосылок U-теста является непрерывность распределения, а эмпирические данные имеют дискретное, при вычислении z-метки используется поправка), а затем, используя аппроксимацию стандартного нормального распределения, — p-значение. В рассматриваемом примере оно составило 0.173. Используя правило 3.20, приходим к выводу о невозможности отклонить нулевую гипотезу. Таким образом, используя формулировку, наиболее близкую к оценочной деятельности (см. таблицу 4.1), можно прийти к следующему выводу: наличие

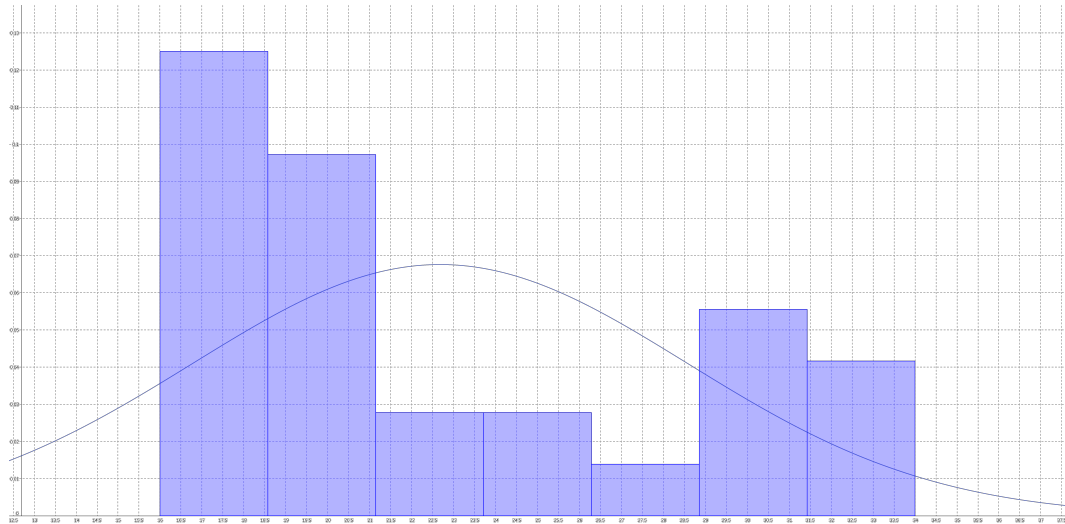


Рис. 4.3. Гистограмма второй выборки, совмещённая с кривой функции плотности вероятности для нормального распределения.

или отсутствие признака «X» не оказывает сколько-нибудь заметного влияния на стоимость — признак «X» не является ценообразующим фактором.

В данной подсекции мы рассмотрели пошаговый расчёт статистики критерия, а также осуществили интерпретацию результата. Следует отметить, что, несмотря на возможность и даже относительное удобство такого варианта проведения U-теста, предпочтение всё же следует отдавать профессиональным средствам разработки в области машинного обучения и статистического вывода, например, языкам программирования Python или R, о которых и пойдёт речь ниже.

4.2. Реализация на Python

В сфере машинного обучения и, в особенности в ряде областей таких как *deep learning*, язык Python уже стал де факто стандартом. Кроме того, он универсален и прекрасно подходит для разработки тех или иных экспертных систем. Его популярность означает в т. ч. наличие огромного количества обучающих материалов по всем аспектам разработки в области анализа данных, предназначенных для пользователей любого уровня подготовки. При этом, большая часть необходимых оценщику вычислений можно провести путём вызова готовых функций из подключаемых библиотек, предназначенных для анализа данных, без необходимости написания большого объёма кода и без глубоких знаний в области программирования. По мнению автора данной работы, будущее оценки заключается именно в применении экспертных систем, основанных на обучении моделей на основе наборов данных открытых рынков. Как будет показано ниже, применение Python существенно сокращает время проведения U-теста, а также позволяет создавать визуализации исследуемого рынка, не прибегая к сторонним средствам. Кроме того, использование готовых функций практически исключает вероятность возникновения ошибок в расчётах. При написании кода была использована версия языка Python 3.9.12, а также IDE Spyder (5.1.5). Код в формате скрипта доступен по ссылке [27], код в формате Python Notebook доступен по ссылке [28].

Рассмотрим реальный набор данных, содержащий сведения об удельных показателях стоимости квартир в Санкт-Петербургской агломерации. Данные были собраны 28 сентября 2021 года с сайта *ciap.ru* и доступны по ссылке [24]. Рассматриваемый набор данных содержит 34821 наблюдение. Как известно, Санкт-Петербургская агломерация включает в себя как территории, входящие в состав города федерального значения, так и те, которые формально относятся к Ленинградской области. При этом, разделение на город и область носит чисто юридический характер. С социально-экономической точки зрения, ближайшие территории Ленинградской области неразрывно связаны с Санкт-Петербургом и являются частью одной агломерации, к слову, крупнейшей в мире на такой широте. При формировании запросов, использованных в процессе скрепинга, южная граница агломерации была установлена примерно по оси автодороги А-120, северная — автодороги 41А-189. При этом в её состав были включены некоторые населённые пункты за пределами этих границ, например, города Кировск и Шлиссельбург.

Сформулируем задачу. Необходимо установить наличие либо отсутствие статистически значимого различия в ценах объектов, расположенных в границах самого Санкт-Петербурга, и объектов, формально расположенных в Ленинградской области. Аналогично предыдущему случаю, сформулируем нулевую и альтернативную гипотезы, имеющие на этот раз практический смысл (см. таблицу 4.2).

Язык Python изначально не был создан специально для анализа данных. Поэтому в его базовой версии могут отсутствовать многие функции, необходимые для проведения расчётов. К счастью, для решения задач в области машинного обучения и анализа данных существует ряд подключаемых библиотек, содержащих множество необходимых функций. Их количество и широта решаемых задач не столь вели-

Листинг 4.1. Подключение необходимых библиотек

```
# import libraries
import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
import scipy.stats as stats
from scipy.stats import norm
from scipy.stats import normaltest
from scipy.stats import shapiro
from scipy.stats import anderson
from scipy.stats import mannwhitneyu
```

Листинг 4.2. Задание применяемого уровня значимости

```
# set significance level
alpha = 0.05
```

ки как, например, у языка R, однако они являются исчерпывающими для более чем 95 % задач, стоящих перед оценщиками. Для решения задач, рассматриваемых в данном материале, потребуются следующие библиотеки: `numpy`, `pandas`, `math`, `matplotlib.pyplot`, `scipy.stats`. Для их подключения используем код, представленный в скрипте 4.1.

Установим уровень значимости α , принимаемый для всей дальнейшей работы. Выбор его значения остаётся за исследователем, однако в работах по эконометрике и исследованию операций чаще всего встречается значение *0.05*, которое и будет использовано. Для задания значения уровня значимости используется код 4.2.

После этого всё готово для начала работы. Создадим датафрейм на основе текстового файла, содержащего изучаемый набор данных (скрипт 4.3). Датафрейм в точности повторяет содержимое исходного файла и содержит 34821 наблюдения и 4 переменные: порядковый номер, ссылку на объявление, показатель стоимости 1 кв. м, а также код местоположения, состоящий из четырёх букв: первая из которых означает регион (s — Санкт-Петербург, l — Ленинградская область), вторая

Листинг 4.3. Загрузка данных и создание датафрейма

```
# import dataset
df = pd.read_csv('spba-flats-210928.csv')
print(df)
type(df['price_m'])
```

Листинг 4.4. Создание датафрейма содержащего только необходимые переменные и выгрузка из памяти неиспользуемых данных

```
# get only prices and counties, release RAM
df1 = df[['price_m', 'county']]
del [df]
```

и третья — административный район, три последних — муниципальное образование либо территорию. При этом Python добавил собственную переменную, содержащую номера наблюдений. Следует обратить внимание на то, что нумерация в Python, как и в большинстве языков программирования, начинается не с единицы, а с нуля. Поскольку переменные, содержащие номера наблюдений и ссылки на объявления из исходного файла, не будут использоваться в дальнейшем, создадим новый датафрейм, содержащий только необходимые переменные, а также выгрузим из виртуальной памяти первый датафрейм для оптимизации ресурсов компьютера (листинг 4.4). В рассматриваемом случае такая микрооптимизация не играет большой роли, однако в целях выработки навыков написания хорошего кода, лучше всё же написать одну дополнительную строку.

Теперь в распоряжении оценщика в удобном виде есть рабочий датафрейм, содержащий данные о рынке квартир всей агломерации Санкт-Петербурга. Для формирования первого представления о распределении построим гистограмму, совмещённую с кривой плотности для нормального распределения. Для определения рационального числа интервалов (столбцов гистограммы) k используем формулу Heinhold-Gaede [3]:

$$k = \sqrt{n}, \quad (4.1)$$

где n — число наблюдений. Для построения гистограммы используем скрипт 4.5.

Рассмотрим полученную гистограмму 4.4. Ось x содержит значения цен за 1 кв. м, ось y — значения вероятностей интервалов. Обе оси представлены в стандартном виде. Также показаны значения матожидания и стандартного отклонения для теоретического нормального распределения. Как видно, распределение имеет тяжёлый правый хвост, что позволяет сделать предварительный вывод о том, что оно отличается от нормального. В дальнейшем будет проведён строгий тест на нормальность, пока же можно ограничиться первичной субъективной интерпретацией гистограммы. Поскольку предметом исследования является различие между объектами, расположенными в двух частях агломерации, а исходный набор данных содержит сведения о наблюдениях из обеих, потребуется создание двух отдельных датафреймов. Здесь следует сделать небольшое отступление: практический опыт говорит о том, что сам анализ данных и построение моделей занимают только 20 % времени, тогда как 80 % уходит на сбор и предобработку данных. Одним из важных элементов этих процессов является правильная разметка данных. В случае с рассматриваемым набором данных их анализ в разрезе отдельных территорий вплоть до уровня муниципалитетов был предусмотрен изначально путём указания индекса территории

Листинг 4.5. Построение гистограммы для агломерации Санкт-Петербурга

```
# calculate the number of observations on data frame
spbaLenR = round(math.sqrt(len(df1.index)))

# fit a normal distribution to the data: mean and standard deviation
mu, std = norm.fit(df1['price_m'])

# plot the histogram
plt.hist(df1['price_m'], bins=spbaLenR, density=True)

# plot the PDF
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)

plt.plot(x, p, 'k', linewidth=2)
title = 'Fit Values: {:.2f} and {:.2f}'.format(mu, std)
plt.title(title)

# save to .pdf
plt.savefig('spba-price-histogram-py.pdf')
```

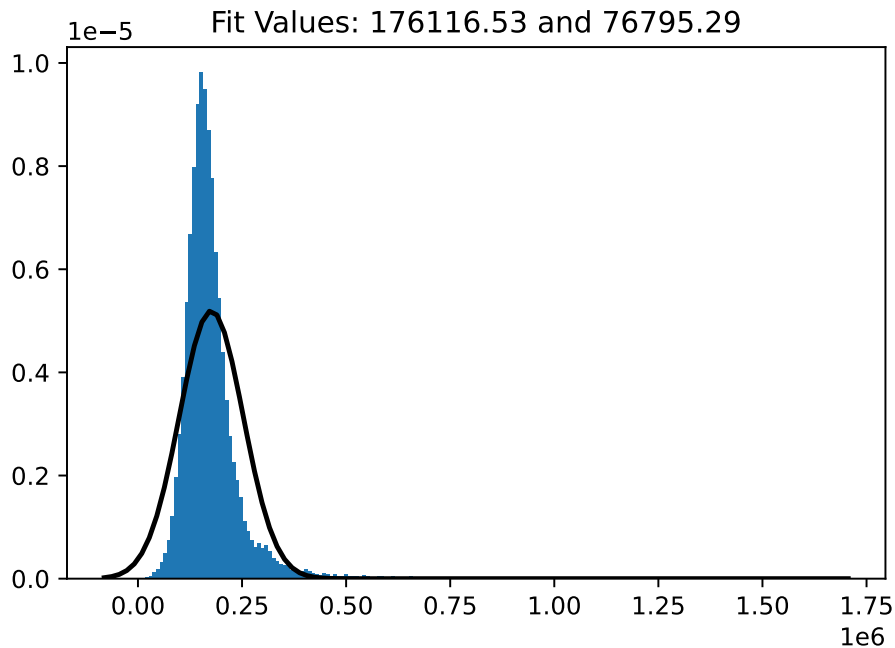


Рис. 4.4. Гистограмма плотности распределения цен за 1 кв. м квартир в Санкт-Петербургской агломерации, совмещённая с кривой функции плотности вероятности для нормального распределения.

для каждого наблюдения. Как было сказано выше, первая буква индекса содержит указание на то, в какой части агломерации расположено наблюдение. В этом случае, для создания двух отдельных датафреймов достаточно двух строк, содержащих несложные регулярные выражения. Датафрейм *dfs* содержит данные для наблюдений из Санкт-Петербурга (28643 наблюдения), *dfl* — Ленинградской области (6178 наблюдений). Построим гистограммы для обеих частей агломерации (скрипты 4.7, 4.8).

Гистограмму иногда путают со столбчатой диаграммой. Следует напомнить, что правильно построенная гистограмма является отображением вероятностных свойств данных, сумма площадей всех её прямоугольников равна единице, а по оси *y* отложены значения вероятностей диапазонов (столбцов гистограммы), а не число

Листинг 4.6. Создание отдельных датафреймов для Санкт-Петербурга и Ленинградской области

```
# create separate dataframes for city and suburbs
dfs = df1[df1['county'].str.startswith('s')] # Saint-Petersburg
dfl = df1[df1['county'].str.startswith('l')] # Leningradskaja oblastq
```

Листинг 4.7. Построение гистограммы для Санкт-Петербурга

```
# Saint-Petersburg
# calculate the number of observations on data frame
spbLenR = round(math.sqrt(len(dfs.index)))

# fit a normal distribution to the data: mean and standard deviation
muS, stdS = norm.fit(dfs['price_m'])

# plot the histogram
plt.hist(dfs['price_m'], bins=spbLenR, density=True)

# plot the PDF
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
ps = norm.pdf(x, muS, stdS)

plt.plot(x, ps, 'k', linewidth=2)
title = 'S-Pb. Fit Values: {:.2f} and {:.2f}'.format(muS, stdS)
plt.title(title)

# save to .pdf
plt.savefig('spb-price-histogram-py.pdf')
```

Листинг 4.8. Построение гистограммы для Ленинградской области

```
# L0
# calculate the number of observations on data frame
loLenR = round(math.sqrt(len(dfl.index)))

# fit a normal distribution to the data: mean and standard deviation
muL, stdL = norm.fit(dfl['price_m'])

# plot the histogram
plt.hist(dfl['price_m'], bins=loLenR, density=True)

# plot the PDF
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
pl = norm.pdf(x, muL, stdL)

plt.plot(x, pl, 'k', linewidth=2)
title = 'L0. Fit Values: {:.2f} and {:.2f}'.format(muL, stdL)
plt.title(title)

# save to .pdf
plt.savefig('lo-price-histogram-py.pdf')
```

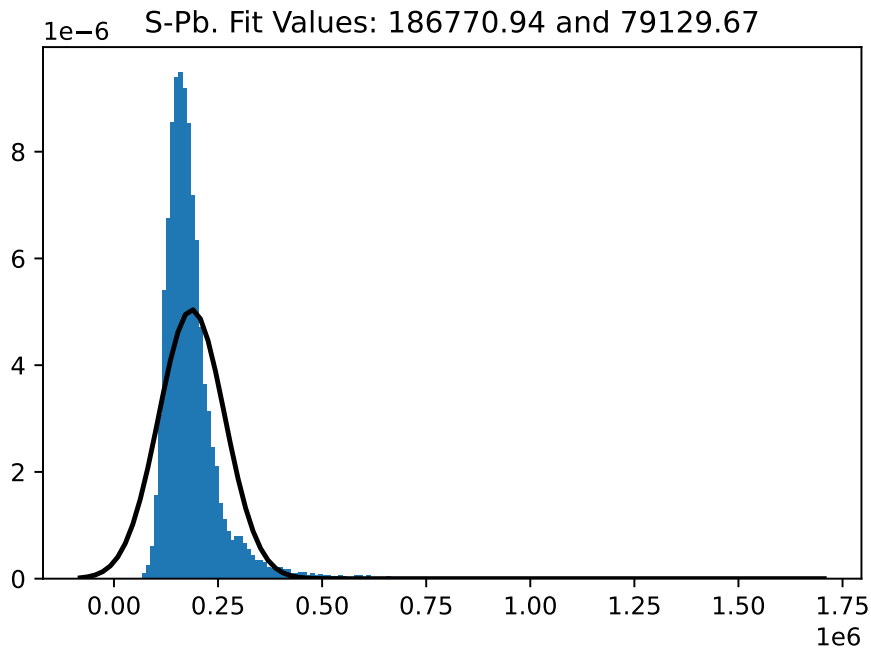


Рис. 4.5. Гистограмма плотности распределения цен за 1 кв. м квартир в Санкт-Петербурге, совмещённая с кривой функции плотности вероятности для нормального распределения.

наблюдений в каждом диапазоне. Как видно из гистограммы 4.5, распределение удельных цен в Санкт-Петербурге, также как и в случае с распределением цен для всей агломерации, имеет тяжёлый правый хвост. При этом распределение цен для объектов агломерации, находящихся за пределами границ Санкт-Петербурга, показанное на гистограмме 4.6 выглядит относительно симметрично.

Также построим график «ящик с усами» для обоих датафреймов с помощью скрипта 4.9 (см. диаграмму 4.7). Как видно, значение медианы цен объектов, расположенных в Санкт-Петербурге, выше значения третьего квартиля цен объектов, расположенных на прилегающих территориях Ленинградской области.

Данные обстоятельства позволяют сделать субъективное предположение о том, что нулевую гипотезу следует отклонить. Однако графические методы анализа подходят только для быстрой первичной интерпретации, а также для презентационных целей. Для формирования объективного доказательного суждения потребуется проведение самого U-теста.

Для проверки применимости U-теста следует провести тест на нормальность распределения для обоих датафреймов (dfs , dft). Существует множество критериев для проверки гипотезы о нормальности распределения выборки. В данном случае были использованы три теста:

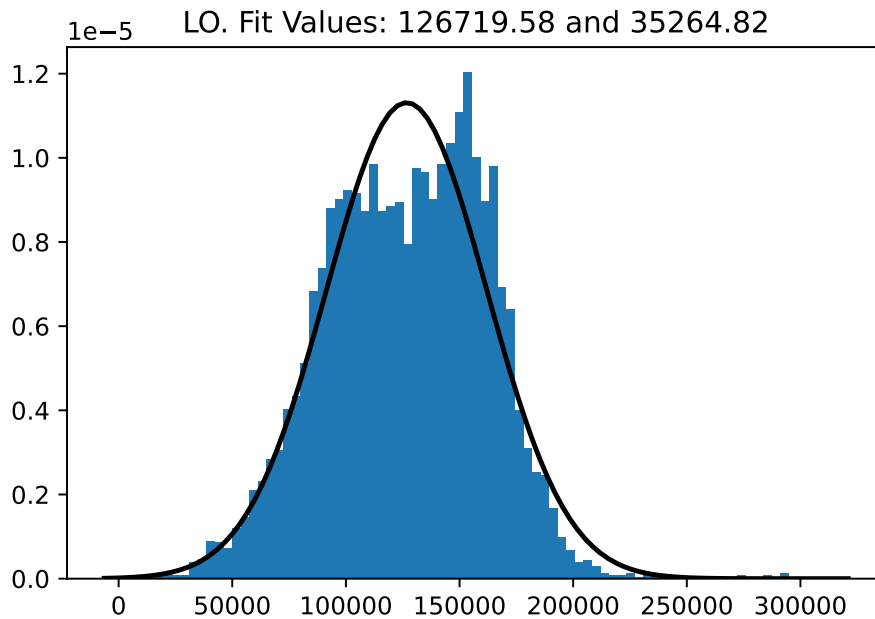


Рис. 4.6. Гистограмма плотности распределения цен за 1 кв. м квартир в Ленинградской области, расположенных в границах агломерации Санкт-Петербурга, совмещённая с кривой функции плотности вероятности для нормального распределения.

- тест Шапиро—Франчия [4];
- тест K^2 Д'Агостино [9];
- тест Андерсона—Дарлинга [2].

Тест Шапиро—Франчия оценивает выборку данных и вычисляет, насколько вероятно, что она была взята из генеральной совокупности, имеющей нормальное распределение. Данный тест считается одним из наиболее мощных тестов проверки на нормальность [16]. При этом существуют некоторые предпосылки, указывающие на то, что он хорошо работает на выборках среднего размера, не превышающих пяти тысяч наблюдений (минимальное количество должно быть не менее пяти).

Тест K^2 Д'Агостино основывается на анализе показателей асимметрии [67] и эксцесса [63], представляющих собой третий и четвёртый центральные моменты [57] соответственно. Данный тест также считается одним из наиболее мощных и не имеет ограничений по максимальному числу наблюдений.

Тест Андерсона—Дарлинга представляет собой модифицированную версию критерия согласия Колмогорова—Смирнова [61] и используется для проверки гипотезы о том, что эмпирическое распределение согласуется с одним из известных теоретических. В отличие от двух предыдущих тестов, его результатом является не p -значение,

Листинг 4.9. Построение диаграммы «ящик с усами» (boxplot) для обеих подвыборок

```
# add labels to data
dfs['region'] = 'SPb'
dfl['region'] = 'LO'

# plot boxplot
prices = [dfs, dfl]
allPrices = pd.concat(prices)
plt.figure()
allPrices.boxplot(by='region')

# save to .pdf
plt.savefig('spb-lo-boxplot-py.pdf')
```

а статистика критерия, что требует более сложной интерпретации результата, которая однако легко автоматизируется.

Сформулируем нулевые гипотезы:

- $H_0(\text{SPb})$: распределение значений удельных цен предложений квартир в Санкт-Петербурге не отличается от нормального;
- $H_0(\text{LO})$: распределение значений удельных цен предложений квартир на территориях Ленинградской области, входящих в агломерацию Санкт-Петербурга, не отличается от нормального.

Таким образом всего будет выполнено 6 тестов, результаты которых сведены в таблицу 4.3. В отношении данных по Санкт-Петербургу все три теста позволили отклонить $H_0(\text{SPb})$, два из трёх тестов также позволили отклонить $H_0(\text{LO})$. На основании данных результатов можно сделать вывод о том, что распределение одной из выборок однозначно отличается от нормального, второй — отличается от нормального с высокой вероятностью. В связи с этим, применение параметрических тестов для сравнения двух выборок является неуместным, вследствие чего следует использовать рассмотренный выше U-тест. Теперь остаётся только провести сам U-тест. Для этого используем скрипт 4.16. Его результаты представлены в таблице 4.4. Поскольку p -значение меньше заданного уровня значимости, можно сделать практический вывод о том, что различия в показателях стоимости объектов, расположенных в границах Санкт-Петербурга, и объектов, расположенных на территориях его агломерации, расположенных в Ленинградской области, являются существенными и требуют соответствующий учёт. Другие интерпретации результата могут быть получены из столбца «Альтернативная гипотеза (H_1)» таблицы 4.2.

Листинг 4.10. Тест Шапиро-Уилка для данных по Санкт-Петербургу

```
stat, p = shapiro(dfs['price_m'])
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
if p <= alpha:
print('Sample does not look Gaussian (reject H0)')
else:
print('Sample looks Gaussian (fail to reject H0)')
```

Листинг 4.11. Тест Шапиро-Уилка для данных по Ленинградской области

```
stat, p = shapiro(df1['price_m'])
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
if p <= alpha:
print('Sample does not look Gaussian (reject H0)')
else:
print('Sample looks Gaussian (fail to reject H0)')
```

Листинг 4.12. Тест K2 Агостино для данных по Санкт-Петербургу

```
stat, p = normaltest(dfs['price_m'])
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
if p <= alpha:
print('Sample does not look Gaussian (reject H0)')
else:
print('Sample looks Gaussian (fail to reject H0)')
```

Листинг 4.13. Тест K2 Агостино для данных по Ленинградской области

```
stat, p = normaltest(df1['price_m'])
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
if p <= alpha:
print('Sample does not look Gaussian (reject H0)')
else:
print('Sample looks Gaussian (fail to reject H0)')
```

Листинг 4.14. Тест Андерсона-Дарлинга для данных по Санкт-Петербургу

```
result = anderson(dfs['price_m'])
print('Statistic: %.3f' % result.statistic)
p = 0
for i in range(len(result.critical_values)):
    sl, cv = result.significance_level[i], result.critical_values[i]
    if result.statistic < result.critical_values[i]:
        print('%.3f: %.3f, data looks normal (fail to reject H0)' % (sl, cv))
    else:
        print('%.3f: %.3f, data does not look normal (reject H0)' % (sl, cv))
```

Листинг 4.15. Тест Андерсона-Дарлинга для данных по Ленинградской области

```
result = anderson(dfl['price_m'])
print('Statistic: %.3f' % result.statistic)
p = 0
for i in range(len(result.critical_values)):
    sl, cv = result.significance_level[i], result.critical_values[i]
    if result.statistic < result.critical_values[i]:
        print('%.3f: %.3f, data looks normal (fail to reject H0)' % (sl, cv))
    else:
        print('%.3f: %.3f, data does not look normal (reject H0)' % (sl, cv))
```

Листинг 4.16. Проведение теста Манна–Уитни–Уилкоксона для данных удельных цен предложения квартир в агломерации Санкт-Петербурга

```
stat, p = mannwhitneyu(dfs['price_m'], dfl['price_m'])
print('stat=%.3f, p=%.3f' % (stat, p))
if p <= 0.05:
    print('Probably different distributions')
else:
    print('Probably the same distribution')
```

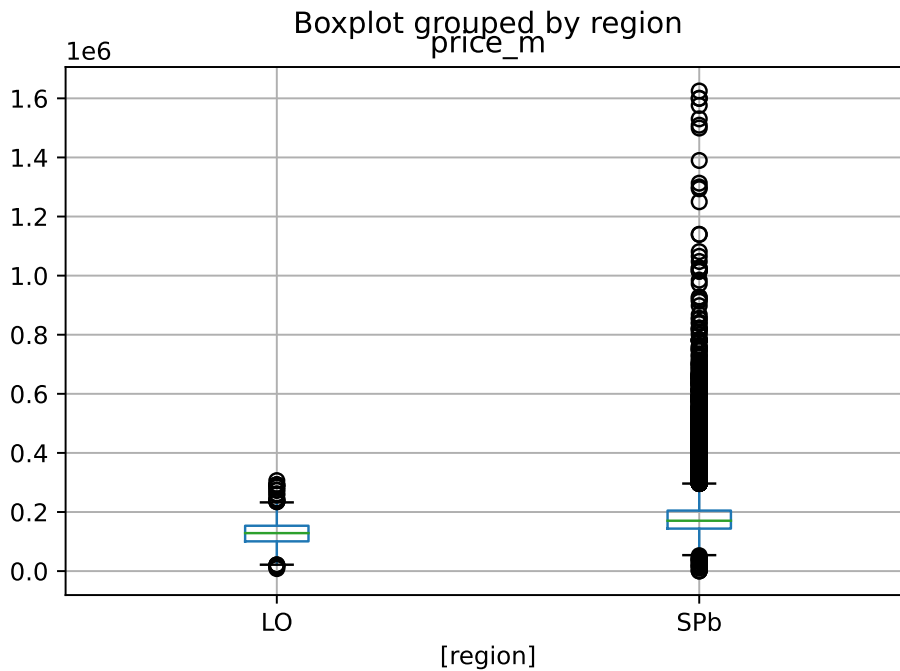


Рис. 4.7. Диаграмма «ящик с усами» для цен предложений квартир в Санкт-Петербургской агломерации в разрезе региональной принадлежности.

4.2.1. ROC-анализ, вычисление AUC

Теоретические вопросы связи U-теста и понятия AUC были рассмотрены ранее в 3.4, детальный разбор практических вопросов вычисления AUC для реальных данных приведён в 4.3.1.1. В данном подразделе будет показан код для вычисления AUC и RBC. Код приведён в скрипте 4.17. Полученные значения вместе с другими результатами показаны в таблице 4.4.

Листинг 4.17. Вычисление AUC и RBC для данных Санкт-Петербургской агломерации

```
# calculate AUC&RBC
n1n2 = len(dfs.index) * len(dfl.index)
auc = stat/n1n2
rbc = auc-(1-auc)
```

Таблица 4.2. Нулевая и альтернативная гипотезы при анализе данных Санкт-Петербургской городской агломерации.

Тип гипотезы	Нулевая гипотеза (H0)	Альтернативная гипотеза (H1)
Научная	Распределение удельных показателей стоимости квартир, расположенных в границах Санкт-Петербурга, и квартир, расположенных на прилегающих к нему территориях Ленинградской области, одинаково, сдвиг между ними отсутствует, статистические оценки, сделанные для множества объектов-аналогов, расположенных в одной части агломерации, являются несмещёнными для объектов, расположенных в другой.	Распределение удельных показателей стоимости квартир, расположенных в границах Санкт-Петербурга, отличается от распределения удельных показателей стоимости квартир, расположенных на прилегающих к нему территориях Ленинградской области, существует сдвиг, оценка, сделанная для объектов, расположенных в одной части агломерации, будет смещённой для объектов, расположенных в другой её части.
Практическая	Медиана удельного показателя стоимости квартир, расположенных в границах Санкт-Петербурга, равна медиане удельного показателя стоимости квартир, расположенных на прилегающих территориях Ленинградской области.	Медиана удельного показателя стоимости квартир, расположенных в границах Санкт-Петербурга не равна медиане удельного показателя стоимости квартир, расположенных на прилегающих территориях Ленинградской области.
Изложенная в терминах оценки	Расположение квартиры в границах Санкт-Петербурга либо на прилегающих к нему территориях Ленинградской области не является существенным различием и не требует какого-либо специального учёта.	Расположение квартиры в границах Санкт-Петербурга либо на прилегающих к нему территориях Ленинградской области является существенным различием и требует отдельный учёт.

Таблица 4.3. Результаты проведения тестов проверки данных по Санкт-Петербургской агломерации на нормальность ($\alpha = 0.05$).

Тест	Санкт-Петербург	Ленинградская область
Шapiro—Уилка:	4.10	4.11
статистика критерия (W)	0.689	0.991
p-значение	0.000	0.000
H ₀	отклоняется	отклоняется
K^2 Д'Агостино:	4.12	4.13
статистика критерия (K^2)	28166.251	4.067
p-значение	0.000	0.131
H ₀	отклоняется	не может быть отклонена
Андерсона—Дарлинга:	4.14	4.15
статистика критерия (A^2)	1688.671	15.795
H ₀ :	отклоняется	отклоняется
Итоговый вывод:		
H ₀	отклоняется	отклоняется

Таблица 4.4. Результаты проведения U-теста для данных Санкт-Петербургской агломерации ($\alpha = 0.05$).

Показатель	Значение
Статистика критерия	142555441.000
p-значение	0.000
Нулевая гипотеза (см. таблицу 4.2)	отклоняется
AUC	0.806
RBC	0.611

4.3. Реализация на R

Язык программирования R не столь распространён как Python, хотя и пользуется достаточной популярностью в развитых странах. В Северной Евразии область его применения является достаточно нишевой и, чаще всего, он используется в научной деятельности, в особенности в области биологии и химии. Для специалиста по машинному обучению знание данного языка является скорее бонусом, но не основным навыком. Тем не менее, следует отметить достоинства R, к которым можно отнести:

- большой набор библиотек и функций, существенно превосходящий набор средств Python;
- очень хорошие средства визуализации результата;
- удобные инструменты разработки веб-приложений, например Shiny;
- язык является не компилируемым, а интерпретируемым, что иногда бывает удобнее в случае решения конкретных небольших задач.

Второе обстоятельство является, пожалуй, главным аргументом в пользу включения языка R в цикл публикаций по искусственному интеллекту для оценщиков. Если Python как язык общего назначения изначально предназначен для создания компилируемых исполняемых приложений, R разработан для пошагового анализа данных и представления всех промежуточных результатов.

Выбор основного языка программирования, используемого оценщиком, зависит от конкретной задачи: в случае разработки крупных комплексных решений предпочтительнее использование Python. В ситуациях, когда целью является решение частной задачи, в особенности требующей серьёзной визуализации результата, есть смысл обратить внимание на R. В любом случае оба этих языка обладают достаточным набором средств для решения всего спектра задач по анализу данных, возникающих в процессе оценки стоимости.

При написании кода на R была использована его версия 4.2.1 (2022-06-23) "Funny-Looking Kid", а также IDE RStudio (RStudio RStudio 2022.02.3+492 "Prairie Trillium" Release, 2022-05-20 for Ubuntu Bionic).

Рассмотрим ещё одну практическую задачу на примере набора данных о рынке жилья города Алматы, предоставленный профессором университета «Нархоз» G. Shoulenbaeva. Файл с данными доступен по ссылке[20]. Рассматриваемый набор данных содержит 2355 наблюдений, а также 12 переменных, содержащих сведения о значениях признаков наблюдений. Одна из переменных содержит сведения о том, предлагается ли квартира к продаже вместе с мебелью и бытовой техникой или без них. Возможны три варианта значения переменной:

- продажа квартиры без мебели и техники;
- продажа квартиры с частичным оснащением предметами интерьера и техники;

- продажа полностью оснащённой квартиры.

Сформулируем задачу: необходимо установить наличие либо отсутствие влияния оснащения квартиры предметами движимого имущества на её стоимость. Данная задача, по мнению автора, представляет определённый теоретический и практический интерес. Во-первых, теория оценки гласит, что при определении стоимости объекта недвижимости следует учитывать стоимость только неотделимых улучшений объекта, тогда как стоимость элементов, являющихся движимым имуществом, следует исключать из стоимости самого объекта. При этом, на практике зачастую невозможно точно определить принадлежность того или иного элемента к отдельным либо неотделимым улучшениям, а также определить их наличие у объектов-аналогов. Математический анализ данных рынка позволит ответить на вопрос, существует ли данная проблема в принципе, либо влияние фактора наличия улучшений, имеющих признаки отдельных, слишком несущественно и в любом случае не может быть корректно учтено при проведении оценки. Во вторых, решение данной задачи даст новые знания о конкретном рынке недвижимого имущества. Для дальнейшего анализа будем считать, что существуют только два варианта:

- продажа без потенциально отдельных улучшений и движимого имущества;
- продажа вместе с потенциально отдельными улучшениями и движимым имуществом.

Решение объединить две категории в одну продиктовано, во-первых, математическими ограничениями U-теста, предназначенного для сравнения только двух выборок (для анализа более чем двух выборок существует непараметрический тест Краскела– Уоллиса также известный как односторонний ранговый ANOVA [62]), во-вторых, с точки зрения обозначенной выше теоретической проблемы, важно понять, оказывает ли влияние на стоимость факт наличия каких-либо отдельных улучшений как таковых, в третьих, деление объектов на частично и полностью оснащённые могло носить несколько субъективный характер. Варианты нулевой и альтернативной гипотез приведены в таблице 4.5.

При написании кода на R автор использовал его версию 4.2.0, а также IDE RStudio (version 2022.02.2 Build 485). При начале работы следует подключить необходимые библиотеки, задать некоторые константы, а также установить адрес рабочего каталога, например так, как это показано в скрипте 4.18.

Далее необходимо создать датафрейм на основе существующего текстового файла с данными. Затем в целях оптимизации использования ресурсов желательно оставить только необходимые переменные 'price.m' и 'furniture', а затем преобразовать датафрейм в более удобный и современный формат 'tibble' (скрипт 4.19). После этого необходимо рассчитать общее число наблюдений, а также каждого типа в зависимости от наличия отдельных улучшений (скрипт 4.20). Результаты подсчёта представлены в таблице 4.6.

В целях первичной визуализации данных построим гистограммы для всех наблюдений. Число столбцов на этот раз будет определено по формуле, разработанной

Листинг 4.18. Подключение библиотек и задание значений констант и адреса рабочего каталога

```
# activate libraries
library(tidyverse)
library(moments)
library(ggplot2)
library(gamlss)
library(normtest)
library(nortest)

# set constants
options('scipen'=2, 'digits'=3)
set.seed(19190709)

# set work catalog
setwd('~/.Mann-Whitney-Wilcoxon/')
```

Листинг 4.19. Создание датафрейма и его настройка

```
# create data set from file, create subset with needed variables,
# change the type of object to a more convenient and modern one
almatyFlats <- read.csv('almaty-aps-2019-1.csv', header = TRUE, sep =
'', dec = '.')
myvars <- c('price.m', 'furniture')
almatyFlats <- almatyFlats[myvars]
as_tibble(almatyFlats)
```

Листинг 4.20. Подсчёт количества наблюдений

```
# calculation of the total number of observations,
# as well as depending on the equipment
n.total <- nrow(almatyFlats)
n.non.equip <- NROW(almatyFlats$furniture[ which(almatyFlats$furniture
== 0)])
n.equip <- NROW(almatyFlats$furniture[ which(almatyFlats$furniture >
0)]))
```

Листинг 4.21. Создание функции для расчёта k по формуле P. W. Nowiczki

```
# create function for second Nowiczki formula
kHistNowiczki2 <- function(x, na.omit = FALSE){ # create function,
  ignore missed values
  n <- NROW(x) # calculate n
  kurt = kurtosis(x) # calculate kurtosis
  kn2 = (((kurt^4)*(n^2))^(1/5))*(1/3) # calculate k
  return(kn2) # return k
} # end of function
```

Листинг 4.22. Расчёт k по формуле P. W. Nowiczki для наблюдений различных типов

```
# calculation numbers of k for different types of observations
k.all.data <- kHistNowiczki2(almatyFlats$price.m)
k.non.equip <- kHistNowiczki2(almatyFlats$price.m[
  which(almatyFlats$furniture == 0)])
k.equip <- kHistNowiczki2(almatyFlats$price.m[
  which(almatyFlats$furniture > 0)])
```

в 1991 году P. W. Nowiczki [10]:

$$k = \frac{1}{3} \sqrt[5]{\varepsilon^4 n^2} \equiv \frac{1}{3} \sqrt[5]{\frac{n^2}{\xi^8}}, \quad (4.2)$$

где ε — коэффициент эксцесса, ξ — коэффициент контрэксцесса. Для удобства, сначала создадим соответствующую функцию (скрипт 4.21). а затем рассчитаем рациональное число интервалов для всего набора данных (скрипт 4.22), а затем отдельно для квартир, предлагаемых к продаже без отдельных улучшений и с ними. Результаты расчёта приведены в таблице 4.6.

Построим гистограммы для всех наблюдений (диаграмма 4.8), тех, которые имеют оснащение 4.9, и тех, которые продаются без каких-либо отдельных улучшений 4.10. Из диаграмм следует, что распределения имеют тяжёлые правые хвосты, что косвенно указывает на то, что распределение во всех случаях отличается от нормального. Для обоснованного суждения в дальнейшем будут выполнены количественные тесты проверки нормальности. Код для построения гистограмм приведён в скрипте 4.23. В таблице 4.7 содержатся базовые описательные статистики для каждого из трёх типов наблюдений, построенные при помощи кода 4.24. Для лучшего восприятия с помощью скрипта также была построена диаграмма «ящик с усами» (boxplot) (скрипт 4.25), приведённая на рисунке 4.11. Как видно, медиана цен объектов, предлагаемых к продаже вместе с отдельными улучшениями, выше медианы объектов, предлагаемых к продаже без них.

Листинг 4.23. Построение гистограмм для наблюдений различных типов

```
# plot the histogram, combined with the density curve of the theoretical
# normal distribution for all observations
histDist(almatyFlats$price.m,
  density = TRUE,
  nbins = kHistNowiczki2(almatyFlats$price.m),
  xlab = 'price per meter, kaz tenge',
  ylab = 'probability',
  main = 'Price per meter histogram, all observations')

# plot the histogram, combined with the density curve of the theoretical
# normal distribution for observations without equipment
histDist(almatyFlats$price.m[ which(almatyFlats$furniture == 0)],
  density = TRUE,
  nbins = kHistNowiczki2(almatyFlats$price.m[
  which(almatyFlats$furniture == 0)]),
  xlab = 'price per meter, kaz tenge',
  ylab = 'probability',
  main = 'Price per meter histogram, observations without equipment')

# plot the histogram, combined with the density curve of the theoretical
# normal distribution for observations with equipment
histDist(almatyFlats$price.m[ which(almatyFlats$furniture > 0)],
  density = TRUE,
  nbins = kHistNowiczki2(almatyFlats$price.m[
  which(almatyFlats$furniture > 0)]),
  xlab = 'price per meter, kaz tenge',
  ylab = 'probability',
  main = 'Price per meter histogram, observations without equipment')
```

Листинг 4.24. Построение базовых описательных статистик для наблюдений различных типов

```
# summaries
summary(almatyFlats$price.m)
summary(almatyFlats$price.m[ which(almatyFlats$furniture == 0)])
summary(almatyFlats$price.m[ which(almatyFlats$furniture > 0)])
```

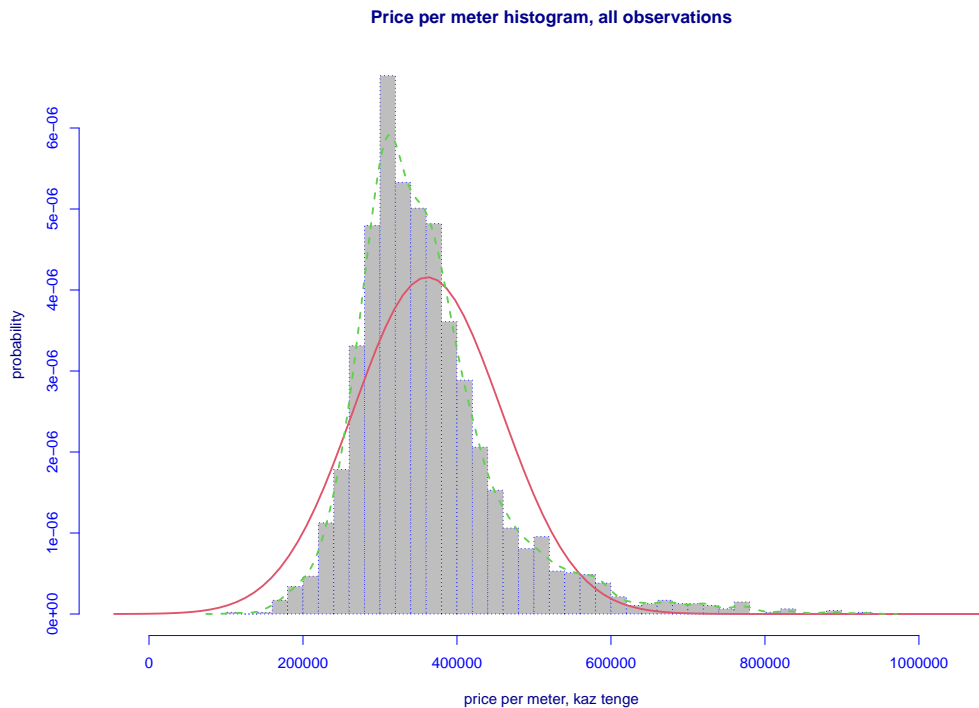


Рис. 4.8. Гистограмма цен предложения для всех объектов, совмещённая с кривой функции плотности эмпирического распределения, а также кривой функции плотности теоретического нормального распределения.

Следующий этап анализа заключается в проверке нормальности распределения удельных цен предложений. Как уже было сказано выше, язык R обладает очень богатым набором инструментов. Существует множество библиотек, предлагающих в общей сложности несколько десятков тестов. В данной работе были применены следующие тесты:

- критерий Шапиро—Уилка [4];
- критерий Шапиро—Франчия [6];
- критерий Андерсона—Дарлинга [2];
- скорректированный критерий Харка—Бера [8];
- критерий Колмогорова—Смирнова с поправкой Лилиефорса [5].

Код для проведения тестов для объектов, продаваемых без отдельных улучшений, приведён в скрипте 4.26, вместе с ними — 4.27. Результаты тестов для обеих групп сведены в таблицу 4.8. Результаты всех тестов позволяют сделать однозначный

Листинг 4.25. Построение диграммы «ящик с усами» для рынка Алматы

```
# plot boxplots
nequiped <- subset(almatyFlats, furniture == 0)
equiped <- subset(almatyFlats, furniture > 0)
boxplot(nequiped$price.m, equiped$price.m,
ylab = 'price per meter',
names =c('not equiped', 'equiped'))
rm(nequiped)
rm(equiped)
```

Листинг 4.26. Проведение тестов на нормальность для наблюдений без отдельных улучшений

```
# normality tests for non equipped observations
# Shapiro-Wilk test for normality
shapiro.test(almatyFlats$price.m[ which(almatyFlats$furniture == 0)])
# Shapiro-Francia test for normality
sf.test(almatyFlats$price.m[ which(almatyFlats$furniture == 0)])
# Anderson-Darling test for normality
ad.test(almatyFlats$price.m[ which(almatyFlats$furniture == 0)])
# Adjusted Jarque-Bera test for normality
ajb.norm.test(almatyFlats$price.m[ which(almatyFlats$furniture == 0)])
# Lilliefors (Kolmogorov-Smirnov) test for normality
lillie.test(almatyFlats$price.m[ which(almatyFlats$furniture == 0)])
```

Листинг 4.27. Проведение тестов на нормальность для наблюдений с отдельными улучшениями

```
# normality tests for non equipped observations
# Shapiro-Wilk test for normality
shapiro.test(almatyFlats$price.m[ which(almatyFlats$furniture > 0)])
# Shapiro-Francia test for normality
sf.test(almatyFlats$price.m[ which(almatyFlats$furniture > 0)])
# Anderson-Darling test for normality
ad.test(almatyFlats$price.m[ which(almatyFlats$furniture > 0)])
# Adjusted Jarque-Bera test for normality
ajb.norm.test(almatyFlats$price.m[ which(almatyFlats$furniture > 0)])
# Lilliefors (Kolmogorov-Smirnov) test for normality
lillie.test(almatyFlats$price.m[ which(almatyFlats$furniture > 0)])
```

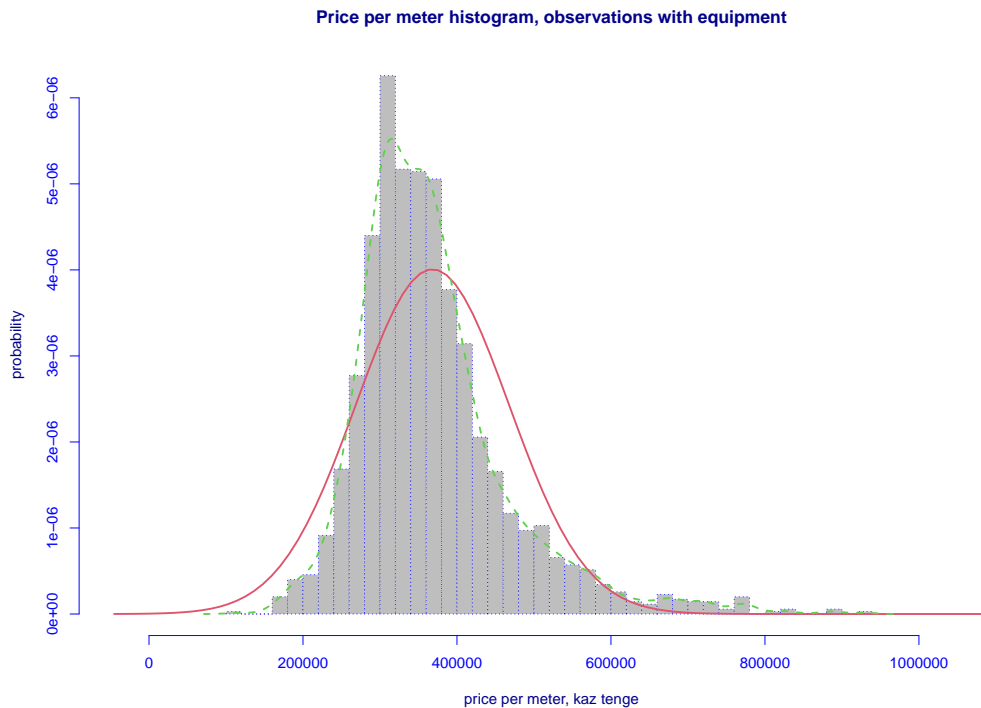


Рис. 4.9. Гистограмма цен предложения для объектов, предлагаемых к продаже предлагаемых к продаже совместно с отдельными улучшениями, совмещённая с кривой функции плотности эмпирического распределения, а также кривой функции плотности теоретического нормального распределения.

вывод: распределения обеих подвыборок отличаются от нормального, что указывает на необходимость применения непараметрических критериев.

Для проведения U-теста достаточно выполнить код, содержащийся в скрипте 4.28. Результаты теста приведены в таблице 4.9. На основании полученного результата можно сделать вывод о том, что учёт фактора наличия неотделимых улучшений и движимого имущества должен быть произведён.

Листинг 4.28. Проведение U-теста для данных города Алматы

```
# perform Mann-Whitney U-test
wilcox.test(almatyFlats$price.m[ which(almatyFlats$furniture == 0)],
almatyFlats$price.m[ which(almatyFlats$furniture > 0)])
```

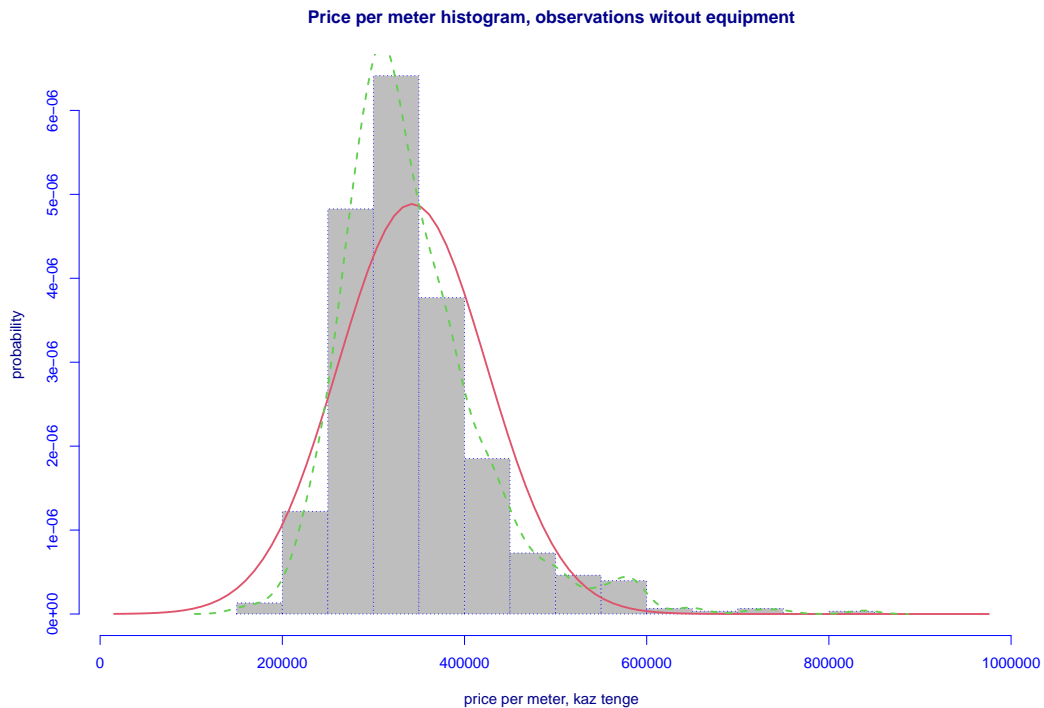


Рис. 4.10. Гистограмма цен предложения для объектов, предлагаемых к продаже без отдельных улучшений, совмещённая с кривой функции плотности эмпирического распределения, а также кривой функции плотности теоретического нормального распределения.

4.3.1. Вычисление AUC

4.3.1.1. Вычисление AUC

Теоретические вопросы связи U-теста и понятия AUC были рассмотрены ранее в 3.4. Отметим только, что AUC, представляющая собой количественную оценку площади под ROC-кривой, является широко распространённой характеристикой качества бинарного классификатора. Удивительно, но AUC напрямую связана с U-статистикой. В данном подразделе будет осуществлён расчёт AUC, а также показана связь между ним и U-тестом.

Добавим к имеющемуся датафрейму вектор *labels*, содержащий бинарные метки, характеризующие наличие либо отсутствие движимого имущества у объекта (см. скрипт 4.29). Таким образом, переменная *price.m* является вектором оценок, *labels* — меток.

Создадим собственную функцию для расчёта AUC на основе формулы 3.12, в которой U_1 — статистика теста, а затем применим её к данным. Для этого напишем скрипт 4.30. Возвращённое значение AUC составило **0.583**, что можно интерпрети-

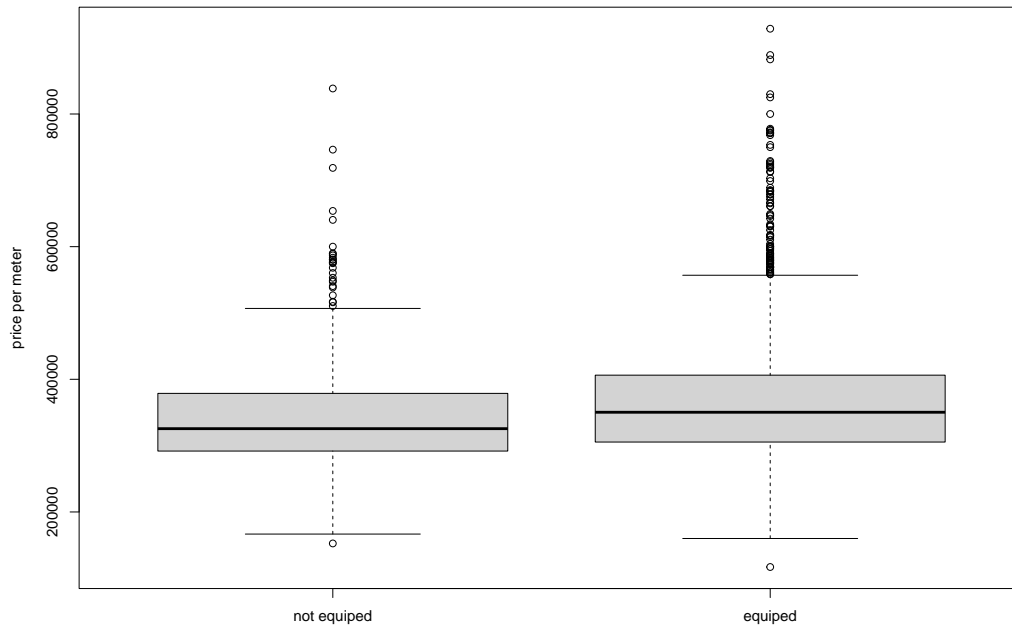


Рис. 4.11. Диаграмма «ящик с усами для рынка Алматы».

ровать следующим образом: «вероятность того, что удельная стоимость случайного наблюдения (объекта-аналога), предлагаемого к продаже вместе с отдельными улучшениями и движимым имуществом выше аналогичного показателя случайного наблюдения, предлагаемого к продаже без отдельных улучшений и движимого имущества».

Также для лучшего понимания дальнейших действий создадим собственную функцию для вычисления U-статистики в соответствии с формулой 3.6, приняв за R_1, n_1 значения «положительных случаев». Естественно, также возможно вычисление U-статистики и для «отрицательных» случаев (R_2, n_2), являющейся комплементарной относительно U_1 , поскольку эти статистики в сумме всегда дают $n_1 \times n_2$. Данное

Листинг 4.29. Добавление переменной с бинарными метками

```
# create vector for labels
almatyFlats$labels <- rep(0, length(almatyFlats$price.m))

# set values by condition
almatyFlats$labels[almatyFlats$furniture > 0] <- 1
```

Листинг 4.30. Создание собственной функции для расчёта AUC и её применение к данным рынка Алматы

```
# create function to calculate AUC
auc_wmw <- function(labels, scores){
  labels <- as.logical(labels)
  pos <- scores[labels]
  neg <- scores[!labels]
  U <- as.numeric(wilcox.test(pos, neg)$statistic)
  U/(length(pos) * length(neg))
}

# apply auc_wmw to data
auc_wmw(almatyFlats$labels, almatyFlats$price.m)
```

действие при построении ROC-кривой приведёт к её переворачиванию, что даёт AUC_2 равную $1 - AUC_1$. Поскольку, как было доказано ранее, U-тест связан с байесовским подходом к вероятности, нет необходимости определять что является положительным и отрицательным случаем с помощью каких-либо специальных методов. Достаточно априорных знаний о том, что объекты, предлагаемые к продаже вместе с отдельными улучшениями и движимым имуществом, стоят дороже чем объекты, предлагаемые к продаже без них. Код для создания такой функции приведён в скрипте 4.31. Возвращённое значение AUC составило **0.583**. Функция **rank**, используемая внутри нашей функции **auc_wmw**, работает следующим образом: наблюдению, имеющему наименьшее значение, присваивается ранг 1 (если нет связей), далее ранги присваиваются по мере возрастания значения. В случае связей таким наблюдениям присваивается медианное значение для всей группы.

В дальнейшем будет приведено обоснование эквивалентности значения, получаемого в результате применения формулы 3.6 и понятия AUC.

4.3.1.2. Графическая интерпретация U-статистики

Построим диаграмму, на которой горизонтальными полосами будут отложены ранги наблюдений, при этом «положительные» наблюдения будут отмечены зелёным цветом, «отрицательные» — красным. Используем для этого код 4.32. В результате была получена диаграмма 4.12.

Рассмотрим теперь только положительные случаи, показанные зелёными полосами. Последовательно уложим их друга на друга, сдвигая вправо по мере встречи отрицательных случаев, получив таким образом последовательность ступенчатых шагов. Используем код 4.33. Рассмотрим диаграмму 4.13. Общая площадь зелёных полос эквивалентна значению R_1 , вычисляемому согласно логике формулы 3.5. Красная рамка справа представляет собой ту часть, которую мы вычтем. Следует обратить внимание на то, что общая площадь зелёных полос внутри заштрихованного

Листинг 4.31. Создание собственной функции для вычисления U-статистики и её применение к данным рынка Алматы

```

create function to calculate AUC in different way
auc_wmw2 <- function(labels, scores){
labels <- as.logical(labels)
n1 <- sum(labels)
n2 <- sum(!labels)
R1 <- sum(rank(scores)[labels])
U1 <- R1 - n1 * (n1 + 1)/2
U1/(n1 * n2)
}

# apply auc_wmw2 to data
auc_wmw2(almatyFlats$labels, almatyFlats$price.m)

```

красного квадрата равна

$$S_{\text{green-bars-right-side}} = \frac{n_1(n_1 + 1)}{2}, \quad (4.3)$$

что представляет собой сумму целых чисел от 1 до n_1 . Прямоугольник в левой части, выделенный чёрным пунктиром, представляет собой ту часть, которую мы оставим. Зелёная области внутри него (сумма зелёных полос) представляет собой значение статистики U_1 .

В рассматриваемом случае, n_1 представляет собой число шагов по оси y , очевидно равное числу зелёных полос. Менее очевидным является то, что n_2 — число шагов «на восток» внутри чёрного прямоугольника, соответствующее числу пропущенных негативных наблюдений. В целом можно сказать, что данный график представляет собой графическое отображение формулы для расчёта U_1 .

Таким образом, вся сетка чёрного прямоугольника имеет площадь равную $n_1 \times n_2$, а U_1 представляет собой абсолютное значение площади под кривой. Для нормирования значения следует разделить U_1 на $n_1 \times n_2$, что полностью совпадает с академической формулой 3.12 для вычисления AUC при выполнении U-теста. Практика нормирования значений следует из правил ROC-анализа, что дополнительно указывает на связь U-теста с методами машинного обучения.

Синяя линия представляет собой ROC-кривую, построенную с помощью штатной функции **roc** из библиотеки **pROC**. Она масштабирована на n_2 по оси x (FPR) и на n_1 по оси y (TPR). В случае полос, представляющих уникальные ранги, она совпадает с их левым краем, в случае связок — разделяет полосу вертикально, представляя собой диагональ «на северо-восток».

Листинг 4.32. Визуализация рангов положительных и отрицательных наблюдений для рынка Алматы

```
# create function to prepare data for plotting
u_illustration_part1 <- function(labels, scores){
  # put cases in order by score
  sort_order <- order(scores)
  labels <- labels[sort_order]
  scores <- scores[sort_order]

  # count the cases
  n <- length(labels)

  # find overall rank for each case by score
  ranks <- rank(scores)

  # start with an empty plot
  plot(c(0, n), c(0, n), type='n', xlim = c(0, 2500),
       xlab='rank', ylab='case', asp=1)

  # plot bars representing ranks of all cases
  mapply(rectangle, x=0, y=(n - 1):0, # starting from the top
         width=ranks, height=1,
         density=NA, lwd=2, col=c('red', 'green')[1 + labels])
  # set labels
  legend('topright', legend=c('negative cases (no furniture &
  equipment)', 'positive cases (with furniture & equipment)'),
  text.col=c('red', 'green'), bty='o', box.lwd=1, inset=0.1)
}

# apply function to data
u_illustration_part1(labels=almatyFlats$labels,
  scores=almatyFlats$price.m)
```

Листинг 4.33. Визуализация рангов положительных наблюдений для рынка Алматы

```
# create function to stack positive cases
u_illustration_part2 <- function(labels, scores){
  # sort the cases
  sort_order <- order(scores)
  labels <- labels[sort_order]
  scores <- scores[sort_order]

  # count positive and negative cases
  n1 <- sum(labels) # number of positive cases
  n2 <- sum(!labels) # number of negative cases

  # find the overall ranks for the positive cases
  ranks <- rank(scores)
  pos_ranks <- ranks[as.logical(labels)]

  # how far to slide each bar to make stairsteps on the right hand edge
  x_offset <- n2 + (1:n1) - pos_ranks

  # start with an empty plot
  plot(c(0, n2 + n1), c(0, n1), type='n', asp=1,
       xlab='n2 + n1 divisions', ylab='n1 divisions')

  # plot bars for ranks of positive cases
  mapply(rectangle, x=x_offset, y=(n1 - 1):0,
         width=pos_ranks, height=1,
         density=NA, border='darkgreen', lwd=2, col='green')

  # mark the area we remove, and the area we keep
  rectangle(n2, 0, n1, n1, density=10, col='red', lwd=1)
  rectangle(0, 0, n2, n1, density=0, col='black', lty=2, lwd=3)

  # draw a scaled version of the "official" ROC curve on top
  roc_obj <- roc(labels, scores)
  roc_df <- with(roc_obj, data.frame(FPR=rev(1 - specificities),
                                    TPR=rev(sensitivities)))
  with(roc_df, lines(n2*FPR, n1*TPR, type='l', lwd=4, col='blue'))
}

# apply function to data
u_illustration_part2(labels=almatyFlats$labels,
  scores=almatyFlats$price.m)
```

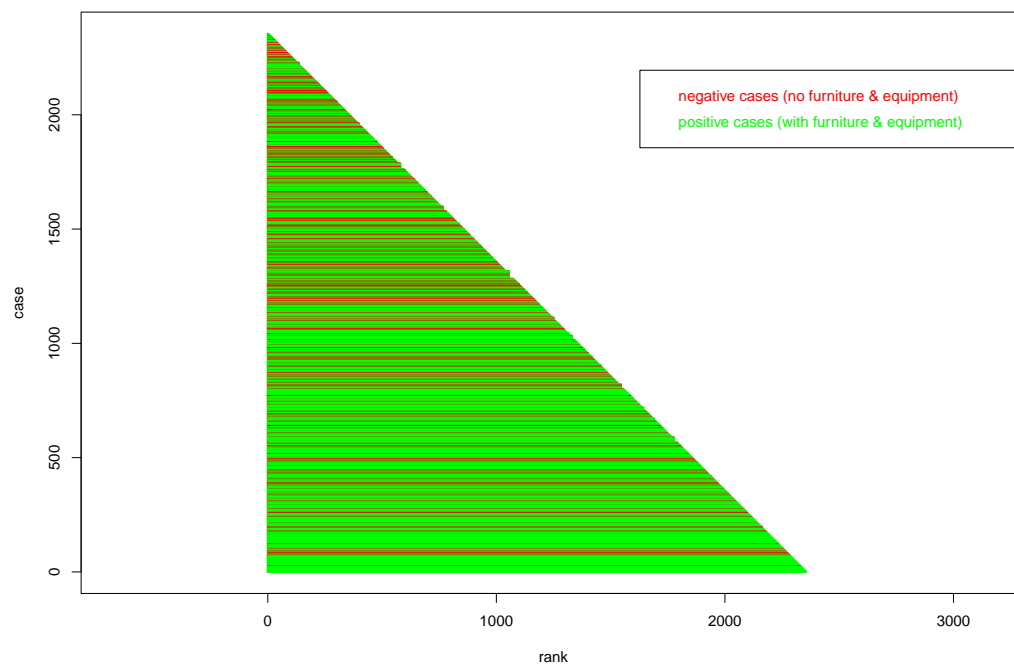


Рис. 4.12. Визуализация рангов положительных и отрицательных наблюдений для рынка Алматы.

Таблица 4.5. Нулевая и альтернативная гипотезы при анализе данных Алматы.

Тип гипотезы	Нулевая гипотеза (H0)	Альтернативная гипотеза (H1)
Научная	Распределения удельных показателей стоимости квартир, предлагаемых к продаже вместе с отдельными улучшениями и движимым имуществом, и квартир, продаваемых без них, одинаковы, сдвиг между ними отсутствует, статистические оценки, сделанные для множества объектов-аналогов, продаваемых вместе с отдельными улучшениями и движимым имуществом, являются несмещёнными для объектов, продаваемых без них, (справедливо и обратное утверждение).	Распределение удельных показателей стоимости квартир, предлагаемых к продаже вместе с отдельными улучшениями и движимым имуществом, и квартир, продаваемых без них, различается, существует сдвиг, оценка, сделанная для объектов, предлагаемых к продаже вместе с отдельными улучшениями и движимым имуществом, будет смещённой для объектов, предлагаемых к продаже без них, (справедливо и обратное утверждение).
Практическая	Медиана удельного показателя стоимости квартир, предлагаемых к продаже вместе с отдельными улучшениями и движимым имуществом, равна медиане удельного показателя стоимости квартир, предлагаемых к продаже без них.	Медиана удельного показателя стоимости квартир, предлагаемых к продаже вместе с отдельными улучшениями и движимым имуществом, не равна медиане удельного показателя стоимости квартир, предлагаемых к продаже без них.
Изложенная в терминах оценки	Наличие либо отсутствие отдельных улучшений и движимого имущества в составе продаваемой квартиры не является существенным различием и не требует какой-либо специальный учёт, т. е. не является ценообразующим фактором.	Наличие либо отсутствие отдельных улучшений и движимого имущества в составе продаваемой квартиры является существенным различием и требует специальный учёт, т. е. является ценообразующим фактором.

Таблица 4.6. Сведения о количестве наблюдений различных типов на рынке города Алматы.

Тип наблюдений	Количество	Рациональное число интервалов (k)
Все наблюдения	2355	36
Наблюдения без отдельных улучшений	605	22
Наблюдения с отдельными улучшениями	1750	31

Таблица 4.7. Базовые описательные статистики наблюдений различных типов на рынке города Алматы (единица — казахстанский тенге).

Тип наблюдений	Min	1Q	Медиана	Среднее	3Q	Max
Все наблюдения	117000	300000	344432	361554	400000	928571
Наблюдения без отделимых улучшений	152542	291803	325581	342581	378788	838462
Наблюдения с отделимыми улучшениями	117000	305446	350331	368113	406183	928571

Таблица 4.8. Результаты проведения тестов проверки данных по г. Алматы на нормальность ($\alpha = 0.05$).

Тест	Без отделимых улучшений	С отделимыми улучшениями
Шапиро—Уилка:	4.26	4.27
статистика критерия (W)	0.9	0.9
р-значение	<2e-16	<2e-16
H ₀	отклоняется	отклоняется
Шапиро—Франция:	4.26	4.27
статистика критерия (W)	0.9	0.9
р-значение	<2e-16	<2e-16
H ₀	отклоняется	отклоняется
Андерсона—Дарлинга	4.26	4.27
статистика критерия (A)	13	42
р-значение	<2e-16	<2e-16
H ₀	отклоняется	отклоняется
Жарка—Бера (скорр.)	4.26	4.27
статистика критерия (AJB)	757	1720
р-значение	<2e-16	<2e-16
H ₀	отклоняется	отклоняется
Лиллиефорса (K-C)	4.26	4.27
статистика критерия (D)	0.1	0.1
р-значение	<2e-16	<2e-16
H ₀	отклоняется	отклоняется
Итоговый вывод:		
H ₀	отклоняется	отклоняется

Таблица 4.9. Результаты проведения U-теста для данных Алматы ($\alpha = 0.05$).

Показатель	Значение
Статистика критерия	441360
р-значение	1e-09
Нулевая гипотеза (см. таблицу 4.5)	отклоняется
AUC	0.583
RBC	0.166

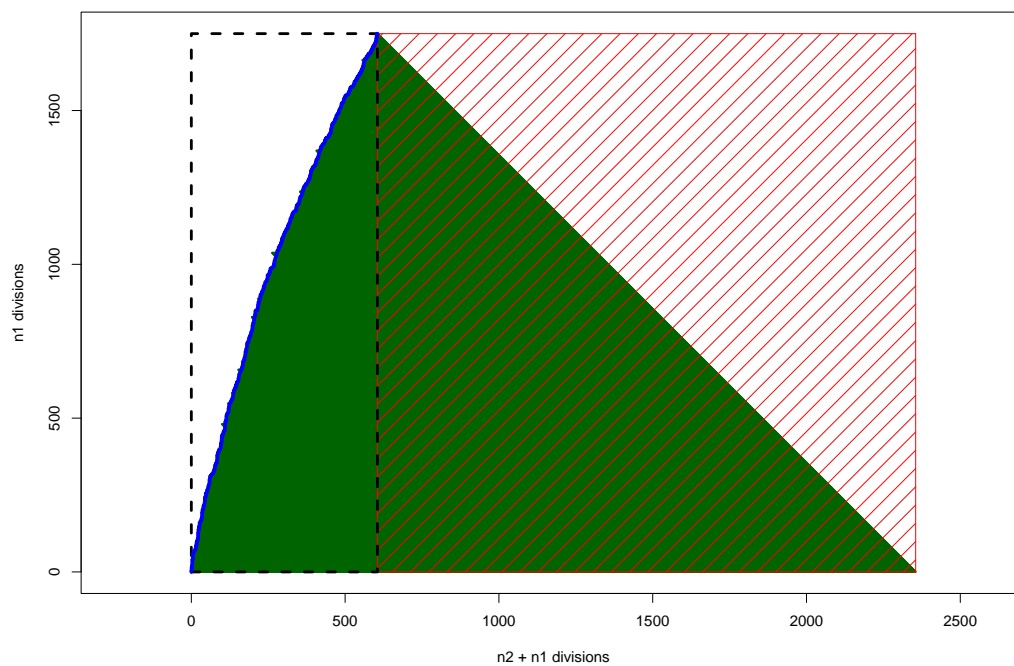


Рис. 4.13. Визуализация рангов положительных и отрицательных наблюдений для рынка Алматы.

Глава 5.

Выводы

В данной работе были рассмотрены теоретические и практические аспекты применения критерия Уилкоксона—Манна—Уитни в повседневной практике оценщика. Данный тест может стать надёжным критерием проверки значимости тех или иных признаков объектов-аналогов, наблюдаемых на открытом рынке. Самостоятельный анализ рыночных данных является лучшим и, вероятно, единственным подлинно доказательным инструментом оценщика, стремящегося создавать ценность для заказчика, а также новые знания о рынках. Несмотря на некоторый начальный порог входа, анализ данных средствами языков программирования является достаточно простым. Строго говоря, часть сделанных шагов является избыточной и была сделана с целью продемонстрировать малую часть возможностей языков Python и R, а также показать важность визуализации данных. Минимальный набор действий, необходимых для проведения U-теста включает следующие этапы:

- загрузка данных и создание датафрейма;
- создание двух подвыборок на основе значения изучаемого признака;
- проведение тестов на нормальность распределений (в большинстве случаев достаточно проведения теста Шапиро—Франчия либо K2 Д’Агостино);
- проведение U-теста.

С учётом наличия готовых скриптов проведение теста занимает всего несколько минут.

В завершение материала хочется дать следующую рекомендацию: современному оценщику в первую очередь необходимо изучать Python: скорее всего, его инструменты исчерпывающе закроют все потребности по анализу данных. В случае стремления к совершенству и наличия желания быть на уровень выше, знание R может помочь в решении особенно сложных и нетривиальных задач. Впрочем, существуют и альтернативные мнения [48]. Эпоха массового применения методов машинного обучения и математической статистики только начинается, и сейчас сложно предсказать, какие средства станут стандартом оценщика через 5 или 10 лет.

Источники информации

- [1] Donald R. Mann Henry B. and Whitney. «On a test of whether one of two random variables is stochastically larger than the other». B: *The annals of mathematical statistics* (1947), с. 50—60.
- [2] T. W. Anderson; D. A. Darling. «Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes». English. B: *Annals of Mathematical Statistics* 23.2 (1952), с. 193—212. DOI: 10.1214/aoms/1177729437. URL: <https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-23/issue-2/Asymptotic-Theory-of-Certain-Goodness-of-Fit-Criteria-Based-on/10.1214/aoms/1177729437.full> (дата обр. 29.05.2022).
- [3] J. Heinhold и K. W. Gaede. *Ingenieur-Statistik*. 1965, с. 327.
- [4] S. S. Shapiro и M. B. Wilk. «An analysis of variance test for normality (complete samples)». English. B: *Biometrika* 52 (3-4 1965-12-01), с. 591—611.
- [5] Hubert W Lilliefors. «On the Kolmogorov-Smirnov Test for Normality with Mean and Variance Unknown». B: *Journal of the American Statistical Association* 62.318 (1967-06-01), с. 399—402. ISSN: 0162-1459. DOI: 10.1080/01621459.1967.10482916. URL: https://en.wikipedia.org/wiki/Lilliefors_test (дата обр. 04.06.2022).
- [6] S. S. Shapiro и R. S. Francia. «An Approximate Analysis of Variance Test for Normality». B: *Journal of the American Statistical Association* 67.337 (1972), с. 215—216. DOI: 10.1080/01621459.1972.10481232. URL: https://en.wikipedia.org/wiki/Shapiro%E2%80%93Francia_test (дата обр. 04.06.2022).
- [7] William J. Conover. *Practical Nonparametric Statistics*. John Wiley и Sons, 1980.
- [8] Carlos M. Jarque и Anil K. Bera. «Efficient tests for normality, homoscedasticity and serial independence of regression residuals». B: *Economics Letters* 6.3 (1980), с. 255—259. URL: https://en.wikipedia.org/wiki/Jarque%E2%80%93Bera_test (дата обр. 04.06.2022).
- [9] Ralph B. Agostino, Albert Belanger и Jr. Ralph B. Agostino. «A suggestion for using powerful and informative tests of normality». English. B: *The American Statistician* 44.4 (1990), с. 316—321. DOI: 10.2307/2684359. URL: <https://web.archive.org/web/20120325140006/http://www.cee.mtu.edu/~vgriffis/CE%205620%20materials/CE5620%20Reading/Dagostino%20et%20al%20-%20normality%20tests.pdf> (дата обр. 29.05.2022).

- [10] П. В. Новицкий и И. А. Зограф. *Оценка погрешностей результатов измерений*. 2-е изд. Ленинград: Энергоатомиздат, 1991, с. 304. ISBN: 5283045137. URL: <http://kepstr.eltech.ru/tor/mri/Literatura/Novitzkij%201991.pdf> (дата обр. 15.10.2021).
- [11] Erich L Lehmann. «Elements of Large Sample Theory». В: *Springer* (1999), с. 176.
- [12] Eiiti Kasuya. «Mann–Whitney U test when variances are unequal». В: *Animal Behaviour* 6.61 (2001), с. 1247—1249. DOI: 10.1006/anbe.2001.1691. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0003347201916914> (дата обр. 06.06.2022).
- [13] Michael C Mozer. *Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic*. English. 2003.
- [14] Corinna Cortes и Mehryar Mohri. «AUC optimization vs. error rate minimization». В: *Advances in neural information processing systems* 16.16 (2004), с. 313—320.
- [15] Tom Fawcett. «An introduction to ROC analysis». В: *Pattern Recognition Letters* 27 (2005-12-19), с. 861—874. URL: <file:///home/kaarlahti/Downloads/ROCintro.pdf> (дата обр. 25.06.2022).
- [16] А. И. Кобзарь. *Прикладная математическая статистика*. 2006.
- [17] Dennis D. Boos и L. A. Stefanski. *Essential Statistical Inference*. English. Springer, 2013.
- [18] Министерство финансов России. *Международный стандарт финансовой отчётности (IFRS) 13 «Оценка справедливой стоимости»*. с изменениями на 11 июля 2016 г. Russian. Russia, Moscow: Минфин России, 2015-12-28. URL: <https://normativ.kontur.ru/document?moduleId=1&documentId=326168#10> (дата обр. 10.06.2020).
- [19] E. Brunner и Arne C Bathke. *Rank and pseudo-rank procedures for independent observations in factorial designs: Using R and SAS*. English. Springer International Publishing, 2018. ISBN: 978-3-030-02912-8.
- [20] G. Shulenbaeva. *almaty-aps-2019-1*. Под ред. К. А. Murashev. 2019. URL: https://github.com/Kirill-Murashev/AI_for_valuers_R_source/tree/main/datasets/almaty_aps_2019_1.csv.
- [21] Julian D. Karch. «Psychologists Should use Brunner-Munzel’s instead of Mann-Whitney’s U-test as the Default Nonparametric Procedure». В: *Advances in Methods and Practices in Psychological Science* 2.4 (2021). ISSN: 2515-2459. DOI: 10.1177/2515245921999602. URL: <https://journals.sagepub.com/doi/10.1177/2515245921999602> (дата обр. 06.06.2022).
- [22] К. А. Murashev. «Short Introduction to the differences between Frequentist and Bayesian approaches to probability in valuation». В: (2021-10-10). URL: https://github.com/Kirill-Murashev/AI_for_valuers_book/tree/main/Parts-Chapters/Frequentist-and-Bayesian-probability (дата обр. 23.05.2022).

- [23] Royal Institution of Chartered Surveyors (RICS). *RICS Valuation — Global Standards*. English. UK, London: RICS, 2021-11-30. URL: <https://www.rics.org/uk/upholding-professional-standards/sector-standards/valuation/red-book/red-book-global/> (дата обр. 11.05.2022).
- [24] K. A. Мырашев. *spba-flats-210928*. 2021-09-28. URL: https://github.com/Kirill-Murashev/datasets/blob/main/Saint-Petersburg/flats/spba_flats_210928.csv.
- [25] International Valuation Standards Council. *International Valuation Standards*. 2022-01-31. URL: <https://www.rics.org/uk/upholding-professional-standards/sector-standards/valuation/red-book/international-valuation-standards/>.
- [26] K. A. Murashev. «Practical application of the Mann-Whitney-Wilcoxon test (U-test) in valuation». English, Spanish, Russian, Interslavic. B: (2022-05-15). URL: https://github.com/Kirill-Murashev/AI_for_valuers_book/tree/main/Parts&Chapters/Mann-Whitney-Wilcoxon (дата обр. 15.05.2022).
- [27] URL: https://github.com/Kirill-Murashev/AI_for_valuers_book/blob/main/Parts-Chapters/Mann-Whitney-Wilcoxon/U-test.py.
- [28] URL: https://github.com/Kirill-Murashev/AI_for_valuers_book/blob/main/Parts-Chapters/Mann-Whitney-Wilcoxon/U-test.ipynb.
- [29] *AUC-Derivation.ipynb*. URL: https://colab.research.google.com/github/unpingco/Python-for-Signal-Processing/blob/master/AUC_Derivation.ipynb#scrollTo=BgrH5C49LqMx (дата обр. 14.06.2022).
- [30] Creative Commons. *cc-by-sa-4.0*. URL: <https://creativecommons.org/licenses/by-sa/4.0/> (дата обр. 27.01.2021).
- [31] *F-test*. URL: <https://en.wikipedia.org/wiki/F-test> (дата обр. 06.06.2022).
- [32] *Fligner-Policello test in R*. URL: <https://search.r-project.org/CRAN/refmans/RVAideMemoire/html/fp.test.html> (дата обр. 06.06.2022).
- [33] Python Software Foundation. *Python site*. Английский. Python Software Foundation. URL: <https://www.python.org/> (дата обр. 17.08.2021).
- [34] The Document Foundation. *LibreOffice Calc*. Английский. URL: <https://www.libreoffice.org/discover/calc/> (дата обр. 20.08.2021).
- [35] *GeoGebra official site*. URL: <https://www.geogebra.org/> (дата обр. 26.08.2021).
- [36] Bob Horton. *AUC Meets the Wilcoxon-Mann-Whitney U-Statistic*. URL: <https://blog.revolutionanalytics.com/2017/03/auc-meets-u-stat.html> (дата обр. 14.07.2022).
- [37] Bob Horton. *Calculating AUC: the area under a ROC Curve*. URL: <https://blog.revolutionanalytics.com/2016/11/calculating-auc.html> (дата обр. 14.07.2022).

- [38] Bob Horton. *ROC Curves in Two Lines of R Code*. URL: <https://blog.revolutionanalytics.com/2016/08/roc-curves-in-two-lines-of-code.html> (дата обр. 14.07.2022).
- [39] *If p-value is exactly equal to 0.05, is that significant or insignificant?* URL: https://www.researchgate.net/post/If_p-value_is_exactly_equal_to_005_is_that_significant_or_insignificant.
- [40] *Jupyter site*. URL: <https://jupyter.org> (дата обр. 13.05.2022).
- [41] Nicolas Kruchten. *Machine Learning Meets Economics*. URL: <http://nicolas.kruchten.com/content/2016/01/ml-meets-economics/> (дата обр. 11.07.2022).
- [42] Machinelearning.ru. *U-критерий Манна-Уитни*. Russian. URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%9A%D1%80%D0%B8%D1%82%D0%B5%D1%80%D0%B8%D0%B9_%D0%A3%D0%B8%D0%BB%D0%BA%D0%BE%D0%BA%D1%81%D0%BE%D0%BD%D0%B0-%D0%9C%D0%B0%D0%BD%D0%BD%D0%B0-%D0%A3%D0%B8%D1%82%D0%BD%D0%B8 (дата обр. 14.05.2022).
- [43] Machinelearning.ru. *Гипотеза сдвига*. URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%93%D0%B8%D0%BF%D0%BE%D1%82%D0%B5%D0%B7%D0%B0_%D1%81%D0%B4%D0%B2%D0%B8%D0%B3%D0%B0 (дата обр. 15.05.2022).
- [44] Machinelearning.ru. *Критерий Уилкоксона двухвыборочный*. Russian. URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%9A%D1%80%D0%B8%D1%82%D0%B5%D1%80%D0%B8%D0%B9_%D0%A3%D0%B8%D0%BB%D0%BA%D0%BE%D0%BA%D1%81%D0%BE%D0%BD%D0%B0_%D0%B4%D0%B2%D1%83%D1%85%D0%B2%D1%8B%D0%B1%D0%BE%D1%80%D0%BE%D1%87%D0%BD%D1%8B%D0%B9 (дата обр. 14.05.2022).
- [45] Machinelearning.ru. *Критерий Уилкоксона для связанных выборок*. Russian. URL: http://www.machinelearning.ru/wiki/index.php?title=%D0%9A%D1%80%D0%B8%D1%82%D0%B5%D1%80%D0%B8%D0%B9_%D0%A3%D0%B8%D0%BB%D0%BA%D0%BE%D0%BA%D1%81%D0%BE%D0%BD%D0%B0_%D0%B4%D0%BB%D1%8F_%D1%81%D0%B2%D1%8F%D0%B7%D0%BD%D1%8B%D1%85_%D0%B2%D1%8B%D0%B1%D0%BE%D1%80%D0%BE%D0%BA (дата обр. 14.05.2022).
- [46] *Ordered logit*. URL: https://en.wikipedia.org/wiki/Ordered_logit (дата обр. 06.06.2022).
- [47] Kennis Research. *Receiver Operating Characteristic (ROC) Curves*. URL: <https://kennis-research.shinyapps.io/ROC-Curves/> (дата обр. 25.06.2022).
- [48] I. Shutov. *Who decided for everyone that Python is convenient for commercial analytics?* Russian. URL: <https://habr.com/ru/post/670250/> (дата обр. 08.06.2022).
- [49] *Spyder IDE site*. URL: <https://www.spyder-ide.org/>.
- [50] PBC Studio. *RStudio official site*. Английский. URL: <https://www.rstudio.com/> (дата обр. 19.08.2021).
- [51] CTAN team. *TeX official site*. English. CTAN Team. URL: <https://www.ctan.org/> (дата обр. 15.11.2020).

- [52] LaTeX team. *LaTeX official site*. English. URL: <https://www.latex-project.org/> (дата обр. 15.11.2020).
- [53] *TeXLive official site*. URL: <https://www.tug.org/texlive/> (дата обр. 15.11.2020).
- [54] The R Foundation. *The R Project for Statistical Computing*. Английский. The R Foundation. URL: <https://www.r-project.org/> (дата обр. 17.08.2021).
- [55] *Turtle graphics*. URL: https://en.wikipedia.org/wiki/Turtle_graphics (дата обр. 22.06.2022).
- [56] *Welch-t-test*. URL: https://en.wikipedia.org/wiki/Welch's_t-test (дата обр. 06.06.2022).
- [57] Wikipedia. *Central Moment*. URL: https://en.wikipedia.org/wiki/Central_moment (дата обр. 29.05.2022).
- [58] Wikipedia. *Common Language Effect Size*. English. URL: https://en.wikipedia.org/wiki/Effect_size#Common_language_effect_size (дата обр. 16.05.2022).
- [59] Wikipedia. *Dot product*. URL: https://en.wikipedia.org/wiki/Dot_product (дата обр. 14.07.2022).
- [60] Wikipedia. *KISS principle*. URL: https://en.wikipedia.org/wiki/KISS_principle (дата обр. 06.11.2020).
- [61] Wikipedia. *Kolmogorov–Smirnov test*. URL: https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test (дата обр. 29.05.2022).
- [62] Wikipedia. *Kruskal–Wallis one-way analysis of variance*. URL: https://en.wikipedia.org/wiki/Kruskal%E2%80%93Wallis_one-way_analysis_of_variance (дата обр. 31.05.2022).
- [63] Wikipedia. *Kurtosis*. URL: <https://en.wikipedia.org/wiki/Kurtosis> (дата обр. 29.05.2022).
- [64] Wikipedia. *Rank-biserial correlation*. English. URL: https://en.wikipedia.org/wiki/Effect_size#Rank-biserial_correlation (дата обр. 16.05.2022).
- [65] Wikipedia. *Receiver operating characteristic*. URL: https://en.wikipedia.org/wiki/Receiver_operating_characteristic (дата обр. 17.05.2022).
- [66] Wikipedia. *Sigmoid function*. URL: https://en.wikipedia.org/wiki/Sigmoid_function (дата обр. 24.06.2022).
- [67] Wikipedia. *Skewness*. URL: <https://en.wikipedia.org/wiki/Skewness> (дата обр. 29.05.2022).
- [68] Wikipedia. *Standard score*. URL: https://en.wikipedia.org/wiki/Standard_score (дата обр. 17.05.2022).
- [69] Wikipedia. *Type I and type II errors*. URL: https://en.wikipedia.org/wiki/Type_I_and_type_II_errors (дата обр. 08.06.2022).

- [70] Benito van der Zander. *TeXstudio official site*. URL: <https://www.texstudio.org/> (дата обр. 15.11.2020).