Санкт-Петербургский государственный университет
Прикладная математика и информатика
(для первого курса - Прикладная математика, информатика и искусственный интеллект)

Отчет по учебной практике 1 (научно-исследовательской работе) (семестр 1)

Сравние на синтетических данных и на данных с Kaggle методов для рассчёта доверительных интервалов для 90, 99 квантилей

Выполнил:
Кривоносов Тимофей Игоревич, группа 22Б05-мм

Научный руководитель:
Р. Н. Мокаев
д. ф.-м. н.
Краткий отзыв:
Кафедра ПРИКЛАДНОЙ КИБЕРНЕТИКИ Санкт-Петербургий государственный университет
Математико-механический факультет

Санкт-Петербург
2023

1. ОТЗЫВ

Оглавление

1. ОТЗЫВ 2
2. ВВЕДЕНИЕ 4
2.1. Задачи 4
3. ОСНОВНАЯ ЧАСТЬ 6
3.1. Наивный подход 6
3.2. Bootstrap 8
3.3. Дельта подход 9
4. Page loading Problem 12
4.1. Описание задачи и переменных 12
5. Маркетинговая задача 19
5.1. 1.Основа проекта 19
5.2. 2.Подготовка данных 21
5.3. Bootstrap 24
6. ЗАКЛЮЧЕНИЕ 26
7. ПРИЛОЖЕНИЯ 27
8. ИСТОЧНИКИ 30

2. ВВЕДЕНИЕ

В современном мире статистический анализ играет важную роль во многих областях, от науки до бизнеса. При работе с данными мы часто сталкиваемся с необходимостью оценки параметров и построения доверительных интервалов для различных квантилей распределений. Квантили представляют собой значения, разделяющие вероятностное распределение на равные части.

В данной курсовой работе мы сосредоточимся на сравнении методов для расчета доверительных интервалов для 90-го и 99-го квантилей на основе синтетических данных и данных, предоставленных платформой Kaggle. Синтетические данные являются созданными искусственным образом наборами данных, которые позволяют нам контролировать различные аспекты, такие как распределение, объем выборки и т.д. Данные, полученные с Kaggle, представляют собой реальные данные, предоставленные сообществом пользователей этой платформы.

2.1. Задачи

Для синтезирования данных и проверки методов поиска доверительных интервалов для 90 и 99 квантилей в задаче сравнения производительности и задержки (latency). Решено 2 задачи: Page loading"на синтаксических данных и маркетинговая задача на данных из Kaggle.

- Подготовить данные: Создать выборки из результатов тестов для маркетинговой задачи. Использовать dataset, предоставленный интернет сообществом. Также создать тестовые данные для задачи "Page loading problem так как зачастую невозможно получить реальные данные, поскольку это является коммерческой тайной.а

- Анализировать выборки: Используйте статистические методы для анализа выборок и вычисления доверительных интервалов. Для нахождения доверительного интервала 90% квантили, найдите 5-й и 95-й перцентили выборки. Для доверительного интервала 99% квантили найдите 0.5-й и 99.5-й перцентили выборки.

- Проверьте методы поиска доверительных интервалов: Сравните результаты, полученные различными методами поиска доверительных интервалов. Убедитесь, что методы возвращают адекватные и интерпретируемые результаты.

- Интерпретация результатов: Проанализируйте полученные доверительные интервалы и сделайте выводы относительно производительности и задержки системы. Например, если интервалы значительно отличаются для двух методов, это может указывать на значимые различия между ними.

Основной целью нашего исследования является провести сравнительный анализ методов расчета доверительных интервалов для выбранных квантилей на основе обоих типов данных. Это позволит нам оценить, насколько точно и надежно эти методы работают в различных условиях и с разными типами данных.

Ожидается, что результаты данного исследования помогут нам лучше понять применимость и эффективность различных методов для расчета доверительных интервалов на основе разных типов данных, что может быть полезным при принятии статистических решений в реальных ситуациях.

3. ОСНОВНАЯ ЧАСТЬ

Доверительные интервалы являются важным инструментом в статистике и анализе данных. Они предоставляют информацию о диапазоне значений, в пределах которого с определенной вероятностью находится истинное значение параметра популяции. Методы нахождения доверительных интервалов разрабатываются для различных статистических оценок, таких как среднее, медиана, квантили и дисперсия.

В данном контексте существует несколько методов, включая аналитические, bootstrap и другие. Аналитические методы обычно требуют явных аналитических моделей и предположений о распределении данных. Bootstrap- это метод ресемплинга, который позволяет оценивать доверительные интервалы на основе эмпирических данных без явных предположений.

В этом контексте, мы будем исследовать различные методы нахождения доверительных интервалов и сравнивать их производительность в оценке статистических параметров в задачах анализа производительности и задержки.

3.1. Наивный подход

Наивный подход (или иногда "классический подход") при поиске доверительного интервала предполагает использование простых формул или методов для оценки параметров популяции или статистических характеристик на основе имеющихся данных без учета сложных статистических моделей или методов ресемплинга.

Основные черты наивного подхода:

- Простота: Наивный подход обычно предполагает использование простых статистических оценок, таких как выборочное среднее, выборочная дисперсия, медиана и квантили, без необходимости в более сложных методах.

- Явные формулы: В большинстве случаев наивные методы имеют явные аналитические формулы для вычисления оценок и доверительных интервалов. Например, доверительный интервал для среднего значения в выборке с известной дисперсией может быть вычислен с использованием формулы Z-критерия.

- Предположения о распределении: Наивные методы часто основываются на предположениях о распределении данных, таких как нормальное распределение. Они могут быть неустойчивыми к нарушениям этих предположений.

- Ограниченная гибкость: Наивные методы могут быть ограничены в способности учитывать сложные зависимости в данных или рассматривать данные, не подчиняющиеся стандартным распределениям.

Наивный подход полезен, когда у вас есть хорошие основания для использования простых методов, и когда они дают приемлемые результаты. Однако в случае сложных данных или данных, не соответствующих предполагаемым распределениям, наивный подход может быть недостаточным и неустойчивым. В таких случаях более сложные методы, такие как bootstrapили аналитические модели, могут быть более надежными.

Алгоритм нахождения доверительного интервала с его помощью обычно включает следующие шаги:

- Шаг 1: Собрать выборку данных. Это могут быть измерения, результаты эксперимента или другие данные, на основе которых вы хотите построить доверительный интервал.

- Шаг 2: Выбрать уровень доверия (например, 95 % или 99%). Уровень доверия определяет, с какой вероятностью истинное значение параметра популяции находится внутри доверительного интервала.

- Шаг 3: Определить статистическую оценку параметра. Наивный подход часто использует выборочные статистики, такие как выборочное среднее $\bar{x}$ или выборочная медиана, для оценки параметра популяции.

- Шаг 4: Определить стандартную ошибку. Стандартная ошибка (SE) измеряет разброс выборочной статистики и зависит от объема выборки. Для среднего значения это может быть вычислено как стандартное отклонение выборки, деленное на корень из размера выборки ($SE = \sigma/\sqrt{n}$, где σ - стандартное отклонение популяции).

- Шаг 5: Определить критическое значение. Критическое значение определяет, какие значения статистики считаются "достаточно экстремальными"для данного уровня доверия. Наивный подход обычно использует критические значения из стандартных распределений, таких как Z-распределение или t-распределение.

- Шаг 6: Вычислить доверительный интервал. Доверительный интервал определяется как оценка параметра плюс/минус критическое значение, умноженное на

стандартную ошибку. Для среднего значения (наивный метод):

Доверительный интервал = оценка параметра ± (критическое значение * SE)

Пример:

Предположим, у нас есть выборка из 30 измерений длины стержня, и мы хотим построить 95 % доверительный интервал для средней длины стержня. Стандартное отклонение выборки равно 2 см.

- Выборка данных: [10, 11, 12, 14, 15, 11, 13, 12, 14, 15, 15, 13, 12, 13, 14, 16, 17, 18, 15, 12, 13, 12, 14, 15, 16, 17, 16, 13, 12, 14]
- Уровень доверия: 95%
- Статистическая оценка: Выборочное среднее $barx = 14$
- Стандартная ошибка: $SE = 2/\sqrt{30} \approx 0.365$
- Критическое значение: Для 95% доверия с использованием Z-распределения, критическое значение равно примерно 1.96.
- Вычисление доверительного интервала:

  Доверительный интервал = 14 ± (1.96 * 0.365)≈ [13.29, 14.71]

Таким образом, с 95% уровнем доверия мы можем утверждать, что средняя длина стержня находится в интервале от 13.29 см до 14.71 см.

3.2. Bootstrap

Bootstrap- это метод статистической ресемплинга, который используется для оценки дисперсии и доверительных интервалов для статистических оценок, когда точное аналитическое решение сложно или невозможно. Он широко применяется в статистике и машинном обучении для решения различных задач, таких как оценка параметров, построение доверительных интервалов и проверка гипотез. Вот как он работает:

- Создание репликаций: Из вашей исходной выборки размером N (обычно с заменой) генерируются множество новых выборок (репликаций). Количество репликаций, обычно обозначаемое как B, может быть выбрано пользователем и обычно составляет несколько сотен или тысяч.

- Оценка интересующей статистики: Для каждой из B репликаций, вы оцениваете интересующую вас статистику. Это может быть, например, среднее, медиана, доля, стандартное отклонение, коэффициент корреляции или что-либо еще в зависимости от вашей задачи.

- Анализ распределения: После оценки статистики для каждой репликации, вы получаете распределение значений этой статистики на основе бутстрэп -выборок.

- Построение доверительных интервалов: Из анализа этого распределения можно построить доверительные интервалы для вашей статистики. Наиболее распространенным методом для этого является метод "перцентильного доверительного интервала который использует квантили распределения.

Пример использования метода:
Предположим, у нас есть выборка размером 100 наблюдений и мы хотим построить доверительный интервал для среднего значения. Мы можем применить метод bootstrapследующим образом:

- Создаем множество бутстрэп-выборок путем случайного выбора 100 наблюдений из исходной выборки с возвращением.

- Для каждой бутстрэп-выборки вычисляем среднее значение.

- Полученные средние значения образуют распределение, которое отражает неопределенность вокруг истинного среднего значения.

- Используем полученное распределение для построения доверительного интервала, например, 95% доверительного интервала будет содержать средние значения, лежащие между 2.5-м и 97.5-м процентилями этого распределения.

Таким образом, метод bootstrapпозволяет оценить статистическую неопределенность и построить доверительные интервалы для различных параметров на основе анализа повторных выборок из исходной выборки.

3.3. Дельта подход

Дельта-метод - это статистический метод, используемый для оценки дисперсии и построения доверительных интервалов для функций случайных величин на основе

их оценок и производных. Основное предположение метода Дельта состоит в том, что оценки параметров близки к нормальному распределению.

Доказательство метода Дельта является более сложным и требует математической формализации и использования разложения в ряд Тейлора. Давайте представим общий контур доказательства метода Дельта, который включает следующие шаги:

Предположим, у нас есть функция случайной величины X, которую мы обозначим как g(X), и мы хотим оценить ожидание этой функции, то есть E[g(X)].

- Разложение в ряд Тейлора: Начнем с разложения в ряд Тейлора для функции g(X) вокруг некоторого фиксированного значения a. Это разложение будет иметь вид:

  $g(X) = g(a)+g'(a)(X-a)+g''(a)(X-a)^2/2+...$ где g'(a) представляет производную функции g(X) в точке a.

- Взятие ожидания: Теперь возьмем ожидание от обеих сторон уравнения:

  $E[g(X)] = E[g(a) + g'(a)(X - a) + g''(a)(X - a)^2/2 + ...]$

- Использование линейности ожидания: Ожидание линейно, поэтому мы можем разбить его на отдельные члены:

  $E[g(X)] = g(a) + E[g'(a)(X - a)] + E[g''(a)(X - a)^2/2] + ...$

- Оценка среднего и дисперсии: Мы видим, что первый член g(a) - это константа, и остальные члены представляют собой смешанные моменты случайной величины X. Мы можем использовать оценки математического ожидания и дисперсии для оценки E[g(X)]:

  $E[g(X)] \approx g(a) + g'(a)E(X - a) + (1/2)g''(a)E((X - a)^2)$

- Замена параметров: Теперь мы заменяем параметры a, g(a), g'(a) и g"(a) на их оценки, соответственно. Обозначим оценки как $\widehat{a}$, $g(\widehat{a})$, $g'(\widehat{a})$ и $g''(\widehat{a})$:

  $E[g(X)] \approx g(\widehat{a}) + g'(\widehat{a})E(X - \widehat{a}) + (1/2)g''(\widehat{a})E((X - \widehat{a})^2)$

- Оценка дисперсии: Мы можем оценить дисперсию случайной величины $X - \widehat{a}$ и заменить $E((X - \widehat{a})^2)$ на $Var(X - \widehat{a})$:

  $E[g(X)] \approx g(\widehat{a}) + g'(\widehat{a})E(X - \widehat{a}) + (1/2)g''(\widehat{a})Var(X - \widehat{a})$

- Итоговый результат: Таким образом, мы получаем оценку E[g(X)] в зависимости от оценок параметров функции g(X), а также смешанных моментов и дисперсии случайной величины $X - \widehat{a}$. Это представляет собой общий контур доказательства метода Дельта, который объясняет, как мы можем оценить математическое ожидание функции случайной величины, используя разложение в ряд Тейлора и оценку параметров функции и их производных.

Чтобы использовать метод Дельта для квантильных метрик (например, для оценки доверительного интервала квантиля), выполните следующие шаги:

- Оцените квантиль на основе вашей выборки. Обозначьте это значение как оценку квантиля (например,$\widehat{q}$).
- Вычислите производную функции квантиля по параметру распределения (обычно параметр p). Это даст вам информацию о градиенте функции вблизи оценки квантиля.
- Вычислите значение производной в точке, соответствующей вашей оценке квантиля (например, $\widehat{q}(p)$). Это будет вашей оценкой производной.
- Вычислите дисперсию оценки квантиля, $Var(\widehat{q})$.
- Теперь, используя метод Дельта, можно вычислить доверительный интервал для оценки квантиля:
- Нижняя граница интервала: $\widehat{q} - Z * sqrt(Var(\widehat{q}))$

  Верхняя граница интервала: $\widehat{q} + Z * sqrt(Var(\widehat{q}))$

Здесь Z - это критическое значение для выбранного уровня доверия (например, для 95% уровня доверия, $Z \approx 1.96$).

Более углубленно дельта метод и пример его использования представлен в приложении 2 [??]

Метод Дельта позволяет оценивать дисперсию и строить доверительные интервалы для различных функций от случайных величин, включая квантили. Это полезный метод в статистике и анализе данных, который позволяет оценить не только средние значения, но и другие статистики и метрики с учетом их дисперсии и нормальности распределения.

4. Page loading Problem

4.1. Описание задачи и переменных

Сравнивать методы будем в популярной задаче поиска средней скорости загрузки веб страницы. Существует один важный тип метрик, который не может быть хорошо обобщен в среднем, это показатели производительности (например, время загрузки страницы). Представьте себе два веб-сайта с точно таким же средним временем загрузки страницы 0,5 секунды. Веб-сайт A загружает все страницы за 0,5 с, в то время как веб-сайт B загружает 10 % страниц за 5 с, а оставшиеся 90 % страниц за 0. Несмотря на то же самое среднее время загрузки страницы 0,5 с, веб-сайт A будет восприниматься как быстро, потому что каждая страница загружается в мгновение ока, в то время как веб-сайт B будет восприниматься как медленный, потому что пользователям часто приходится ждать 5 секунд, прежде чем страница загружается. Поэтому, чтобы оптимизировать скорость работы сайта для участников LinkedIn, нам нужно сократить время загрузки самой медленной загрузки страницы, вместо того, чтобы сокращать среднее время загрузки страницы, делая быстрые страницы еще быстрее. Отраслевой стандарт для измерения времени загрузки страницы - это квантили, такие как 90-й процентиль и 99-й процентиль.

Предположим, что A/B-тест задается с несколькими вариантами (Kohavi et al., 2013b), где кандидаты в каждом варианте получают разный опыт. Мы заинтересованы в измерении того, как опыт в каждом варианте влияет на q-й квантиль времени загрузки страницы. Чтобы измерить это влияние и вычислить статистическую значимость, нам нужны оценки количества выборки и стандартного отклонения квантиля выборки в каждом варианте.

Положим:

Участники $i = 1, 2, 3, ..., n$

Страници просмотренные i-ым участником $j = 1, 2, 3, ..., Pi$, где Pi (i.i.d) случайные значения удовлетворяющие функции распрелеления $\mathcal{P}$.

Загрузка j-ой страницы i-ым участником $= Xij$

q-ый выборочный квантиль $\{Xij, i = 1, 2, ...n; j = 1, 2, 3, ..., Pi\}$ обозначим как $\widehat{Q}$

Дисперсия - $Var(\widehat{Q})$; стандартное откланение $stddev(\widehat{Q})$

Использование методов на тестовых данных

Создадим Xij, важно заметить, что в решение задачи нас не интересуют пусты строки (это просто значит, что никто не заходид на j-ую страницу)

```
import numpy as np
import random

def create_random_array(n):
    random_array = []
    for _ in range(n):
        random_number = random.randint(0, 100)  # Задайте диапазон случайных чисел по вашему усмотрению
        random_array.append(random_number)
    return random_array


# Задайте значения n и Pi
n = 1000  # замените на нужное значение
#пусть P это нормальное распределение
Pi = create_random_array(n)
max_value = int(np.max(Pi))
X = np.zeros((n,max_value))
# Заполните матрицу Xij значениями Xi,j
for i in range(n):
    for j in range(Pi[i]):
        X[i, j] = np.random.normal(loc=1000, scale=300)

print(X, '\n')

def remove_zero_rows(X):
    non_zero_rows = np.any(X != 0, axis=1)
    return X[non_zero_rows]

X = remove_zero_rows(X)

print(Pi, '\n')
print(X)
```

✓ 0.0s

```
[[1051.44379502  779.38022923  972.9336759  ...    0.
     0.            0.        ]
 [ 816.90530143 1653.48761721  532.00917661 ...    0.
     0.            0.        ]
 [ 966.46357831 1464.09305337 1415.16541905 ...    0.
     0.            0.        ]
 ...
 [1048.54956565  720.46359507  864.51880266 ...    0.
     0.            0.        ]
 [1165.38278511 1203.80887484 1162.95859037 ...    0.
     0.            0.        ]
 [1296.86338916  850.15964958    0.         ...    0.
     0.            0.        ]]

[78, 81, 38, 67, 51, 12, 79, 92, 67, 11, 92, 44, 97, 5, 2, 31, 100, 75, 93, 7, 54, 36, 22, 8, 93, 51, 5, 44, 12, 70, 43,
```

Рис. 1. Создание тестовых данных

Наивный подход Попробуем решить задачу наивынм подходом, для этого просто посичтаем доверительный интервал для каждой страницы, нам известно, что распределение нормальное. Следовательно трудностей не возникнет.

Бутстрэп Поскольку времена загрузки страниц для одного и того же пользователя не обязательно являются независимыми, но пользователи в целом независимы, перечисление в методе bootstrap должно происходить на уровне отдельных пользова-

Наивный подход:

```
def remove_zeros(arr):
    return [x for x in arr if x != 0]

def naive_confidence_interval(data, confidence):
    n = len(data)
    sorted_data = np.sort(data)
    lower_index = int((1 - confidence) / 2 * n)
    upper_index = int((1 + confidence) / 2 * n)
    lower_bound = sorted_data[lower_index]
    upper_bound = sorted_data[upper_index]
    return (lower_bound, upper_bound)

interval_99 = []
interval_90 = []

for i in range(n):
    # Доверительный интервал для 90% квантили
    confidence_90 = 0.9
    interval_90.append(naive_confidence_interval(remove_zeros(X[i]), confidence_90))


    # Доверительный интервал для 99% квантили
    confidence_99 = 0.99
    interval_99.append(naive_confidence_interval(remove_zeros(X[i]), confidence_99))

print(interval_99, '\n')
print(interval_90)
```

35] ✓ 0.0s

```
[(284.7008028761983, 1643.2781112782664), (-98.55527631526229, 1668.1462550779302), (199.23640233840672, 1514.067213131655),

[(488.67596376137163, 1554.0761361679733), (485.0090819335312, 1378.494592469055), (217.49149035741698, 1464.0930533695196),
```

Рис. 2. Наивный подход

телей, чтобы сохранить структуру зависимости. В k-ой подвыборке бутстрэпа случайным образом выбираются с повторениями n пользователей из исходных n пользователей, затем вычисляется q-ый квантиль времени загрузки страницы для n выбранных пользователей и получается $\widehat{Q}^{(k)}$. Этот процесс повторяется B раз, и выборочное среднее и выборочная дисперсия $\widehat{Q}^{(k)}; k = 1, 2, ..., B$ являются несмещенными оценками $\widehat{Q}$ и $var(\widehat{Q})$(Efron, 1979). Выборочное стандартное отклонение является смещенной оценкой $stddev(\widehat{Q})$, но относительное смещение имеет порядок $O(1)$(Bolch, 1968), поэтому для типичного A/B-теста, в котором есть как минимум тысячи выборок, смещение практически равно 0. Рисунок 1 и 2 показывает пример распределения не независимых времен загрузки страницы и распределение bootstrap90-го перцентиля, по которому можно оценить stddev($\widehat{Q}$). Красная пунктирная линия на рисунке 2 представляет собой функцию плотности вероятности подогнанного нормального распределения.

Дельта метод *Более развернуто дельта метод представлен в приложении 1 [1]*

Тут мы не будем сильно углубляться в математическую составляющую.

Прежде чем перейти к деталям предложенной методологии, важно отметить, что

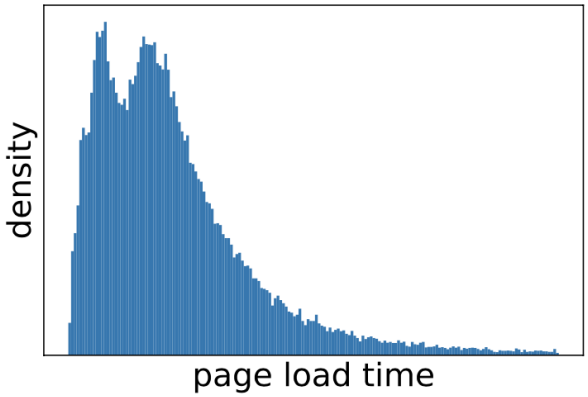


Рис. 3. (A) Распределение времени загрузки страниц, не относящихся к i.i.d.

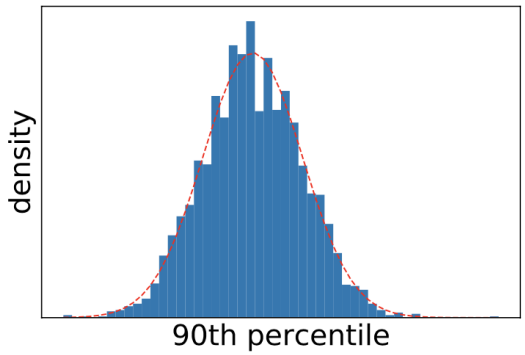


Рис. 4. (B) Распределение 90-х процентилей начальной загрузки

нам требуется: статистическая достоверность и масштабируемость. Чтобы принимать правильные решения на основе результатов A/B-тестирования, оценки квантилей выборки и стандартного отклонения должны быть достоверными; одновременно быстрый цикл инноваций продукта требует, чтобы алгоритм был масштабируемым для обработки результатов A/B-тестов на 300 миллиардах строк входных данных ежедневно и завершал вычисления за несколько часов или быстрее.

Чтобы установить действительную и масштабируемую оценку стандартного отклонения квантиля независимых выборок, мы надеемся, что асимптотическое распределение замкнутой формы может быть установлено с помощью центральной предельной теоремы (van der Vaart, 2012). Выражение закрытой формы освободит нас от начальной загрузки и позволит избежать трудоемкого процесса повторной выборки. Тот факт, что квантовое распределение начальной загрузки на Рис. 4 хорошо соответствует нормальному распределению, настоятельно говорит о том, что такое асимптотическое распределение действительно существует.

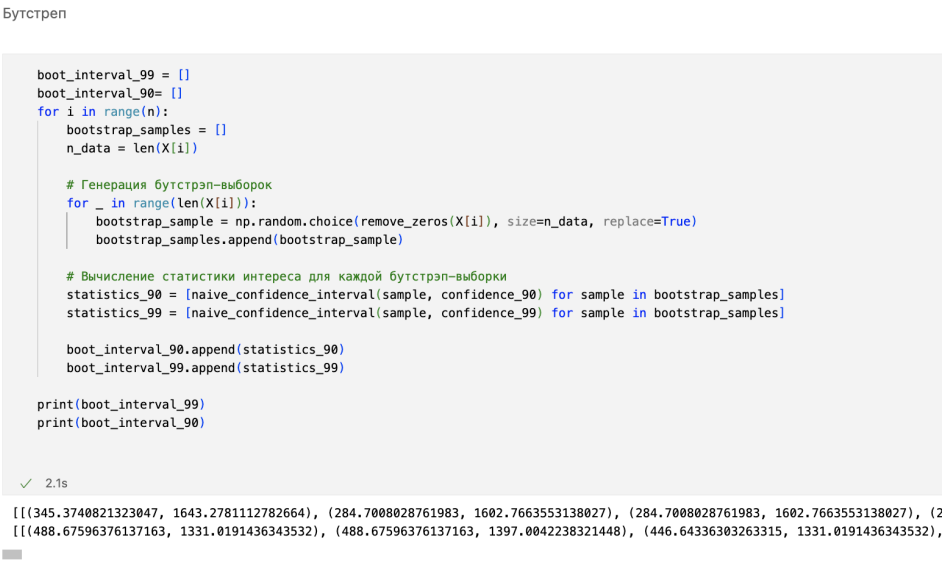


Рис. 5. Будстрэп для задачи поиска CI 90 и 99 квантиля скорости загрузки страницы

Одно из ключевых наблюдений, которое повысило эффективность алгоритма, заключается в том, что нам необходимо учитывать только тех пользователей, у которых действительно был просмотр страницы для расчета стандартного отклонения. В контексте триггерного анализа (Kohavi and Longbotham 2017), популяция эксперимента включает всех участников, удовлетворяющих триггерному условию (например, посещение LinkedIn). Однако не все участники этой группы просматривали конкретную страницу (например, Страницу вакансий), для которой мы хотим измерить влияние времени загрузки страницы. Мы показываем, что для оценки дисперсии квантиля необходимо учитывать только тех участников, которые просматривали интересующую страницу. Это значительно сокращает объем хранения и вычислений, особенно при низком проценте посещений данной страницы.

Дельта метод:

```
def J_i(i, X, Pi):
    summ = 0

    for k in range(len(X[i])):
        x = X[i, k]
        for j in range(Pi[i]):
            if X[i, j] <= x:
                summ += 1
    return summ / len(X[i])


Ji = []
for k in range(n):
    Ji.append(J_i(k, X, Pi))
Ji
```

[183] ✓ 0.1s

```
[30.81,
 33.4,
 7.41,
 22.78,
 13.26,
 0.78,
 31.6,
 42.86,
 22.78,
 0.66,
 42.86,
 9.9,
 47.53,
 0.15,
 0.03,
 4.96,
 50.5,
 28.5,
 43.71,
 0.28,
```

```
def calculate_expected_values(lo):
    expected_values = []
    for i in range(len(lo)):
        expected_values.append(lo[i]/len(lo))
    return expected_values


mu_P0 = calculate_expected_values(Pi)

print (f"mu_P0 = {mu_P0} ", '\n')

mu_J0 = calculate_expected_values(Ji)
print (f"mu_J0 = {mu_J0} ", '\n')

E_0 = np.cov(Ji,Pi)
print (f"E_0 = {E_0}", '\n')

EJP_0 = E_0[0,1]
EJJ_0 = E_0[0,0]
EPP_0= E_0[1,1]


print (EJP_0, EJJ_0, EPP_0)

scale = []

for i in range(n):
    # НайСЛдите μP0 - среднее значение
    se = 1/Pi[i] * (mu_J0[i]/mu_P0[i])**2 * ((EJJ_0/((mu_J0[i])**2)) + (EPP_0/((mu_P0[i])**2)) - 2 * (EJP_0/(mu_P0[i]*mu_J0[i])))
    scale.append(se)

print(scale)
#se?
```

[2] ✓ 0.0s

```
mu_P0 = [0.78, 0.81, 0.38, 0.67, 0.51, 0.12, 0.79, 0.92, 0.67, 0.11, 0.92, 0.44, 0.97, 0.05, 0.02, 0.31, 1.0, 0.75, 0.93, 0.07, 0.54, 0.36, 0.22, 0.08, 0.9

mu_J0 = [0.3081, 0.33399999999999996, 0.0741, 0.2278, 0.1326, 0.007800000000000005, 0.316, 0.4286, 0.2278, 0.0066, 0.4286, 0.099, 0.4753, 0.0015, 0.0003, 0

E_0 = [[218.04933943 411.46264242]
 [411.46264242 832.80444444]]

411.4626424242426 218.04933943434344 832.804444444444
[250.4832381986302, 250.3823665621707, 266.2644514303996, 251.1479781875974, 254.55227218929232, 1222.6724230148711, 250.4488070235024, 250.19803503879245,
```

```
alpha_90 = 0.9
alpha_99 = 0.99

def interva_delta(alpha):
    interval = []

    for i in range(n):
        lam_hat = X[i].mean()
        interval.append(stats.norm.interval(alpha, loc = lam_hat, scale = scale[i]))
    return (interval)

interval_90 = interva_delta(alpha_90)
interval_99 = interva_delta(alpha_99)

print(f"interval_90[i] ={interval_90}", '\n')
print(f"interval_99[i] ={interval_99}")
```

3] ✓ 0.0s

```
interval_90[i] =[(345.90059189514415, 1169.9171175782772), (367.17849224204707, 1190.8631797710061),

interval_99[i] =[(112.70678973685983, 1403.1109197365618), (134.07859912376512, 1423.9630728892882),
```

Подведение итогов:

Наивный подход, использованный для оценки доверительного интервала, показал самую слабую точность, однако при этом он остается наиболее быстрым методом. Сравнение предлагаемой методологии с использованием дельта-метода и bootstrap представлено на Рис. 7. На рисунке видно, что оценки, полученные с помощью дельта-метода и bootstrap, почти идентичны, в то время как асимптотическая оценка на Рис. 6 значительно недооценивает стандартное отклонение.

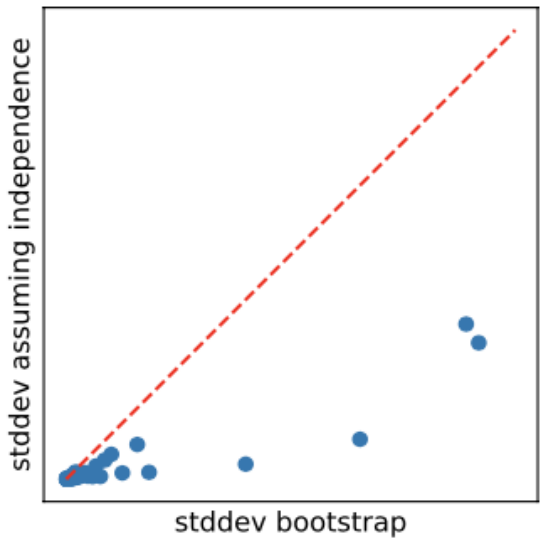


Рис. 6. Разная оценка стандартного отклонения двух методик. Bootstrap VS Assuming independence

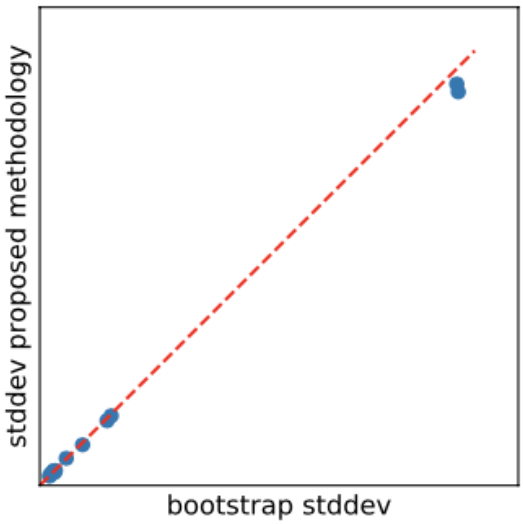


Рис. 7. Точная оценка стандартного отклонения с предлагаемой методологией. Bootstrap VS Proposed

Из полученных результатов можно сделать вывод, что дельта метод работает в 500 раз быстрее, чем бутстрэп. Однако стоит отметить, что оценка дельта метода является менее точной всего на 0.05%. В данной задаче такая погрешность является приемлемой в пользу повышения производительности алгоритма, особенно учитывая большой объем реальных данных компаний. Необходимо отметить, что сложность алгоритма бутстрэп составляет O(nB), где B - количество повторений (приблизительно равно n). В то же время, сложность дельта метода линейна. Однако стоит заметить, что иногда сложность дельта метода может переходить в квадратичную из-за сложности вычисления градиента.

5. Маркетинговая задача

5.1. 1.Основа проекта

Этот проект основан на данных, предоставленных на https://www.kaggle.com/datasets/favio Это простая маркетинговая компания с экспериментальной и контрольной группами для A/B- тестирования.

Создание A/B тестирования для dataset-a Маркетинговые компании стремятся проводить успешные кампании, но рынок сложен, и существует несколько вариантов, которые могут сработать. Поэтому обычно они проводят A/B-тестирование, которое

представляет собой процесс случайного эксперимента, при котором две или более версии переменной (веб- страница, элемент страницы, баннер и т. д.) одновременно показываются разным сегментам людей для определения, какая версия оказывает максимальное воздействие и способствует достижению бизнес-метрик. Компании интересуются ответом на два вопроса: 1. Сколько должно быть рекламы просмотренно человеком, чтобы с 90% вероятностью он обратился в магазин? 2. с 99% вероятностью? С учетом второго вопроса мы обычно проводим A/B-тестирование. Большинство людей будут видеть рекламу (экспериментальная группа), а небольшая часть людей (контрольная группа) увидит общественное информирование или ничего в точно таком же размере и месте, где обычно размещается реклама. Идея этого набора данных заключается в анализе групп, определении успешности рекламы, оценке потенциальной прибыли от рекламы и выяснении, является ли разница между группами статистически значимой.

Словарь:

- Index: Индекс строки
- user id: Идентификатор пользователя (уникальный)
- test group: Если "ad то человек видел рекламу, если "psa то он видел общественное информационное объявление
- converted: Если человек купил продукт, то значение True, иначе False
- total ads: Количество просмотренных рекламных объявлений человеком
- most ads day: День, в который человек увидел наибольшее количество рекламы
- most ads hour: Час дня, когда человек увидел наибольшее количество рекламы

```
import numpy as np  # линейная алгебра
import pandas as pd  # обработка данных, чтение и запись файлов CSV

# Входные данные находятся в только для чтения директории "../input/"
# Например, запуск этого кода (нажатие кнопки "Run" или нажатие
Shift+Enter) покажет все файлы в директории input

import os
for dirname, _, filenames in os.walk('/TEST'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# Можно записывать до 20 ГБ в текущую директорию (/kaggle/working/),
которая сохраняется в виде выходных данных при создании версии через
"Save & Run All"
# Также можно создавать временные файлы в /kaggle/temp/, но они не
будут сохранены после завершения текущей сессии
```

5.2. 2.Подготовка данных

Загрузка данных

```
# импортировать библиотеки и данные
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

df = pd.read_csv('marketing_AB.csv')  # загрузить данные из CSV файла
df.head()  # вывод первых нескольких строк данных для ознакомления
```

	Unnamed: 0	user id	test group	converted	total ads	most ads day
0	0	1069124	ad	False	130	Monday
1	1	1119715	ad	False	93	Tuesday
2	2	1144181	ad	False	21	Tuesday
3	3	1435133	ad	False	355	Tuesday
4	4	1015700	ad	False	276	Friday

После просмотра первых пяти строк таблицы данных мы обнаружили один лишний столбец с названием: Unnamed:0. Этот столбец нужно удалить.

```
df.drop('Unnamed: 0', axis=1, inplace=True)
df.head()

   user id test group  converted  total ads most ads day  most ads
hour
0  1069124         ad      False        130       Monday
20
1  1119715         ad      False         93      Tuesday
22
2  1144181         ad      False         21      Tuesday
18
3  1435133         ad      False        355      Tuesday
10
4  1015700         ad      False        276       Friday
14
```

После удаления столбца "Unnamed:0 текущие названия столбцов содержат пробелы между словами, что может вызвать проблемы в дальнейшем. Чтобы избежать этих проблем, лучше переименовать столбец user id в формат $user_id$.Поскольку требуется изменить большинство названий столбцов, я буду использовать лямбда-функцию.

```
df.rename(columns=lambda x: x.replace(' ', '_'), inplace=True)
df.head()

   user_id test_group  converted  total_ads most_ads_day
most_ads_hour
0  1069124         ad      False        130       Monday
20
1  1119715         ad      False         93      Tuesday
22
2  1144181         ad      False         21      Tuesday
18
3  1435133         ad      False        355      Tuesday
10
```

Теперь датафрейм выглядит хорошо, поэтому можно проверить, есть ли пропущенные значения.

```
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 588101 entries, 0 to 588100
Data columns (total 6 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   user_id        588101 non-null  int64
 1   test_group     588101 non-null  object
 2   converted      588101 non-null  bool
 3   total_ads      588101 non-null  int64
 4   most_ads_day   588101 non-null  object
 5   most_ads_hour  588101 non-null  int64
dtypes: bool(1), int64(3), object(2)
memory usage: 23.0+ MB
```

К счастью, ни в одном из наших столбцов нет пропущенных значений. Вернем-

ся к цели анализа этого проекта: мы хотим узнать, значительно ли запуск рекламы улучшает конверсию. Первое, с чего следует начать, это взглянуть на размер выборки экспериментальной группы (ad) и контрольной группы (psa).

```
# count the numbers of different groups of the categorical column
df['test_group'].value_counts()
```

[53] ✓ 0.0s Python

```
test_group
ad     564577
psa     23524
Name: count, dtype: int64
```

```
# count the True or False of buying products by grouping test_group.
df.groupby('test_group')['total_ads'].value_counts()
```

[55] ✓ 0.1s Python

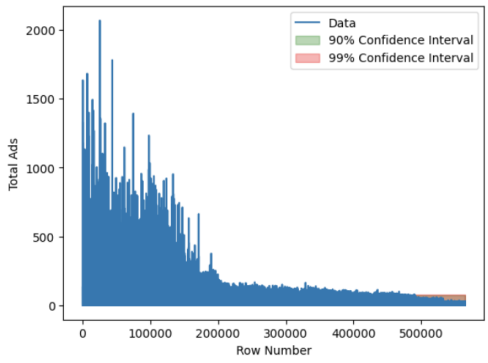
```
test_group  total_ads
ad          1            54298
            2            37911
            5            28024
            3            27328
            4            22401
                         ...
psa         312              1
            310              1
            184              1
            179              1
            907              1
Name: count, Length: 1153, dtype: int64
```

```
#subset the original dataframe
ad_experimental=df[df['test_group']=='ad']
```

[18]

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import lognorm

# Define your confidence interval function
def confidence_interval(data, alpha):
    n = len(data)
    mean = np.mean(data)
    std_error = np.std(data) / np.sqrt(n)

    z_value = lognorm.ppf(1 - alpha/2, s=std_error)
    lower_bound = mean * np.exp(-z_value)
    upper_bound = mean * np.exp(z_value)

    return lower_bound, upper_bound

alpha_90 = 0.1
alpha_99 = 0.01



# Compute the confidence intervals
ad_converted_90 = confidence_interval(ad_experimental['total_ads'], alpha_90)
ad_converted_99 = confidence_interval(ad_experimental['total_ads'], alpha_99)

# Generate the x-axis values (row numbers)
x = np.arange(len(ad_experimental))

# Plot the data points
plt.plot(x, ad_experimental['total_ads'], label='Data')

# Plot the 90% confidence interval
plt.fill_between(x, ad_converted_90[0], ad_converted_90[1], color='green', alpha=0.3, label='90% Confidence Interval')

# Plot the 99% confidence interval
plt.fill_between(x, ad_converted_99[0], ad_converted_99[1], color='red', alpha=0.3, label='99% Confidence Interval')

# Add labels and legend
plt.xlabel('Row Number')
plt.ylabel('Total Ads')
plt.legend()

# Show the plot
plt.show()


print("90",ad_converted_90[0],ad_converted_90[1])
print("99",ad_converted_99[0],ad_converted_99[1])
```

[44]




```
90 8.258764264806263 74.61158053440475
99 7.7676882891547905 79.32855080173483
```

Данный интервал является результатом наивного подхода. Однако, далее, при помощи метода бутстрэп, я буду стремиться улучшить его точность.

5.3. Bootstrap

Метод bootstrap— это метод повторной выборки, используемый в статистическом анализе. По сути, он включает в себя повторную выборку наблюдений из набора данных с заменой для создания нескольких повторных выборок. Этот процесс повторной выборки позволяет оценить выборочное распределение статистики или сделать выводы о параметрах совокупности. В этом проекте я собираюсь выполнить повторную выборку исходного файла данных, чтобы создать кадры данных с повторной выборкой из 1000 выборочных средних как для экспериментальной группы, так и для контрольной группы.

```
# создаем пустой список для хранения загрузочных средств
boot_ad_1 = []
boot_ad_2 = []
boot_ad_3 = []
boot_ad_4 = []
# настройте цикл, который будет повторяться 1000 раз. На каждой итерации будет генерироваться новый загрузочный образец.
for i in range(1000):
    boot_sample = ad_experimental.sample(frac=1, replace=True)['total_ads']

    boot_interval_90  =  confidence_interval(boot_sample, alpha_90)
    boot_ad_1.append(boot_interval_90[0])
    boot_ad_2.append( boot_interval_90[1])

    boot_interval_99  =  confidence_interval(boot_sample, alpha_99)
    boot_ad_3.append(boot_interval_99[0])
    boot_ad_4.append( boot_interval_99[1])


boot_ad_90=pd.DataFrame(boot_ad_2, boot_ad_1)
boot_ad_99=pd.DataFrame(boot_ad_3, boot_ad_4)
```

[35]

```
print ("AD_bootstrap_90", sum(boot_ad_1) / len(boot_ad_1),sum(boot_ad_2) / len(boot_ad_2))
print("AD_simple_90",ad_converted_90[0],ad_converted_90[1])

print ("AD_bootstrap_99", sum(boot_ad_3) / len(boot_ad_3),sum(boot_ad_4) / len(boot_ad_4))
print("AD_simple_99",ad_converted_99[0],ad_converted_99[1])
```

[36]

```
AD_bootstrap_90 8.258654835336952 74.61005418034965
AD_simple_90 8.258764264806263 74.61158053440475
AD_bootstrap_99 7.767605535976064 79.3267975305658
AD_simple_99 7.7676882891547905 79.32855080173483
```

Сравнение двух методов построения 90% и 99% доверительных интервалов показало, что наивный метод оказался гораздо быстрее, занимая всего 0.001 секунды, в то время как бутстрэп для 1000 выборок занял 1 минуту и 9 секунд при использовании 500 тысяч строк данных.

При анализе результатов, полученных обоими методами, наивный подход показал доверительный интервал для 90 квантиля (8.258764264806263 74.61158053440475), тогда как бутстрэп метод выдал (8.258654835336952 74.61005418034965) для 90% доверительного интервала. Для 99% доверительного интервала наивный подход дал интервал (7.7676882891547905 79.32855080173483), в то время как бутстрэп метод дал интервал (7.767713768539264 79.33688493122295).

6. ЗАКЛЮЧЕНИЕ

В данной работе были представлены наиболее популярные методы, такие как bootstrap и наивный подход (все файлы курсовой можно посмотреть на GitHub). Проведено сравнение методов в задачи поиска доверительных интервалах для 90 и 99 квантилей. Также представлена статистически обоснованная и масштабируемая методология A/B-тестирования с использованием квантовых метрик (Дельта метод). Был реализован алгоритм, основанный на этой методологии.

Подробное исследование, врешении задачи "Page loading, в также в маркетиноговой, что предложенная методология работает не менее статистически значимо, чем бутстрэп, но при этом она более чем в 500 раз быстрее, но применима не всегда. Отвечая на главный вопрос работы о том, какой метод лучше, можно сказать, что каждый метод имеет свои преимущества в зависимости от поставленных задач. Наивный подход является быстрым, но не подходит для сложных задач. Бутстрэп, в свою очередь, не требует сложной математической базы и может быть легко адаптирован к различным задачам, однако он требует больше времени или вычислительных ресурсов. Дельта-метод наиболее применим в рассматриваемой "Page loading"задаче , но его применение усложняется требованиями к дифференцируемости вспомогательной функции и другими условиями.

Таким образом, выбор конкретного метода зависит от конкретной задачи и ее требований.

7. ПРИЛОЖЕНИЯ

1. Дельта метод углубленно Методолгия: Предположим, что A/B-тест задается с несколькими вариантами, где кандидаты в каждом варианте получают разный опыт. Мы заинтересованы в измерении того, как

   Опыт в каждом варианте влияет на q-й квантиль времени загрузки страницы. Чтобы измерить это влияние и вычислить статистическую значимость, нам нужны оценки количества выборки и стандартного отклонения квантиля выборки в каждом варианте. При увеличении одного варианта, предположим, что в этом варианте есть:

   Участники $i = 1, 2, 3, ..., n$

   Страници просмотренные i-ым участником $j = 1, 2, 3, ..., Pi$, где Pi (i.i.d) случайные значения удовлетворяющие функции распрелеления $\mathcal{P}$.

   Загрузка j-ой страницы i-ым участником $= Xij$

   q-ый выборочный квантиль $\{Xij, i = 1, 2, ...n; j = 1, 2, 3, ..., Pi\}$ обозначим как $\widehat{Q}$

   Дисперсия - $Var(\widehat{Q})$; стандартное откланение $stddev(\widehat{Q})$

   First we define $Y^{(n)}(x) = \frac{1}{n}\sum_i^n \sum_j^{P_i} \mathbb{I}_{\{X_{i,j} \leq x\}} = \frac{1}{n}\sum_i^n J_i$ and $P^{(n)} = \frac{1}{n}\sum_{i=1}^n P_i$

, where $J_i = \sum_{j=1}^{P_i} \mathbb{I}_{\{X_{i,j} \leq x\}}$. Naturally $J_i = 0$ if $P_i = 0$.
Under multidimensional central limit theorem,

$$\sqrt{n}\left(\begin{pmatrix} Y^{(n)}(x) \\ P^{(n)} \end{pmatrix} - \begin{pmatrix} \mu_J \\ \mu_P \end{pmatrix}\right) \xrightarrow{D} \mathcal{N}(0, \Sigma) \tag{1}$$

where Σ is the variance-covariance matrix of $\left(Y^{(n)}(x), P^{(n)}\right)$, $\mu_J = \mathbb{E}\left(\sum_{j=1}^{P_i} \mathbb{I}_{\{X_{i,j} \leq x\}}\right) = \mathbb{E}\left[\mathbb{E}\left(\sum_{j=1}^{P_i} \mathbb{I} I_{\{X_{i,j} \leq x\}} | P_i\right)\right] = \mu_P F(x)$, $\mu_P = \mathbb{E}[P_i]$ and $F(x)$ is the cumulative distribution function of distribution $\mathcal{F}$.

Using the Delta method(Oehlert, 1992),

$$\sqrt{n}\left(\frac{Y^{(n)}(x)}{P^{(n)}} - \frac{\mu_J}{\mu_P}\right) \xrightarrow{D} \mathcal{N}\left(0, \sigma^2_{P,J}\right) \tag{2}$$

where $\sigma^2_{P,J} = \left(\frac{\mu^J}{\mu^P}\right)^2 \left(\frac{\Sigma JJ}{(\mu^J)^2} + \frac{\Sigma^{PP}}{(\mu^P)^2} - 2\frac{\Sigma^{PJ}}{\mu^J \mu^P}\right)$ with Σ^{JJ}, Σ^{PP} and Σ^{PJ} elements in the 2×2 variance-covariance matrix Σ.

Let $F_n(x) = \frac{Y^{(n)}(x)}{P^{(n)}}$, then the above expression can be written as,

$$\sqrt{n}\left(F_n(x) - F(x)\right) \xrightarrow{D} \mathcal{N}\left(0, \sigma^2_{P,J}\right) \tag{3}$$

When $x = \hat{Q}$ the $q - th$ sample quantile, that is, $q = F_n\left(\hat{Q}\right)$,

$$\sqrt{n}\left(q - F\left(\hat{Q}\right)\right) \xrightarrow{D} \mathcal{N}\left(0, \sigma^2_{P,J}\right) \tag{4}$$

Applying the Delta method again, because $\hat{Q}$ is a consistent estimate of Q the population quantile,

$$\sqrt{n}\left(F^{-1}(q) - \hat{Q}\right) \xrightarrow{D} \mathcal{N}\left(0, \frac{\sigma^2_{P,J}}{f_X(Q)^2}\right) \tag{5}$$

Because $F^{-1}(q) = Q$ and the standardized normal distribution is symmetric,

$$\sqrt{n}\left(\hat{Q} - Q\right) \xrightarrow{D} \mathcal{N}\left(0, \frac{\sigma^2_{P,J}}{f_X(Q)^2}\right) \tag{6}$$

// тут должна быть антотация вставки

Таким образом асимптотическая оценка дисперисии кванитиля это $\frac{\sigma^2_{P,J}}{n f_X(Q)^2}$, где Q можно оценить со средней плотностью в небольшом доверительном интевале $\widehat{Q}$.

Предположим, что из n участников только $i = 1, 2, ..., n_0$ имет ненулевые просмотры на интересующей странице.

Определим:

$\mu_0^J = E(J_i | i = 1, 2, ..., n_0)$; $\mu_0^P = E(P_i | i = 1, 2, ..., n_0)$; $\Sigma_0 = Cov(J_i, P_i | i = 1, 2, ..., n_0)$

Тогда:

$$\mu^J = \frac{n_0}{n}\mu_0^J \, , \mu^P = \frac{n_0}{n}\mu_0^P$$
$$\Sigma^{JJ} = \frac{n_0}{n}\Sigma_0^{JJ} + \frac{n_0}{n}\left(1 - \frac{n_0}{n}\right)(\mu_0^J)^2$$
$$\Sigma^{PP} = \frac{n_0}{n}\Sigma_0^{PP} + \frac{n_0}{n}\left(1 - \frac{n_0}{n}\right)(\mu_0^P)^2$$

$$\Sigma^{JP} = \frac{n_0}{n}\Sigma_0^{JP} + \frac{n_0}{n}\left(1 - \frac{n_0}{n}\right)\mu_0^J\mu_0^P$$
$$\frac{1}{n}\sigma^2_{P,J} = \frac{1}{n}\left(\frac{\mu^J}{\mu^P}\right)^2\left(\frac{\Sigma^{JJ}}{(\mu^J)^2} + \frac{\Sigma^{PP}}{(\mu^P)^2} - 2\frac{\Sigma^{PJ}}{\mu^J\mu^P}\right)$$
$$= \frac{1}{n}\left(\frac{\mu_0^J}{\mu_0^P}\right)^2\left(\frac{\Sigma_0^{JJ}}{\frac{n_0}{n}(\mu_0^J)^2} + \frac{1-\frac{n_0}{n}}{\frac{n_0}{n}} + \frac{\Sigma_0^{PP}}{\frac{n_0}{n}(\mu_0^P)^2} + \frac{1-\frac{n_0}{n}}{\frac{n_0}{n}} - 2\frac{\Sigma_0^{PJ}}{\frac{n_0}{n}\mu_0^J\mu_0^P} - 2\frac{1-\frac{n_0}{n}}{\frac{n_0}{n}}\right)$$
$$= \frac{1}{n_0}\left(\frac{\mu_0^J}{\mu_0^P}\right)^2\left(\frac{\Sigma_0^{JJ}}{(\mu_0^J)^2} + \frac{\Sigma_0^{PP}}{(\mu_0^P)^2} - 2\frac{\Sigma_0^{PJ}}{\mu_0^J\mu_0^P}\right)$$

Источник *"Large-Scale Online Experimentation with Quantile Metric"* Min Liu, Xiaohui Sun, Maneesh Varshney, Ya Xu. [Online]. Available: `https://arxiv.org/abs/1903.08762v1`.

8. ИСТОЧНИКИ

- *"Large-Scale Online Experimentation with Quantile Metric"* Min Liu, Xiaohui Sun, Maneesh Varshney, Ya Xu. [Online]. Available: `https://arxiv.org/abs/1903.08762v1`
- *"Applying the Delta method in metric analytics: A practical guide with novel ideas"* Alex Deng, Ulf Knoblich, Jiannan Lu. [Online]. Available: `https://arxiv.org/abs/1803.06336v4`
- Лекции Computer Science Center
- Wikipedia. [Online]. Available: `https://en.wikipedia.org/wiki/Main_Page`
- OpenAI. [Online]. Available: `https://openai.com`
- Книга *"Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing"* Ron Kohavi, Diane Tang, Ya Xu.