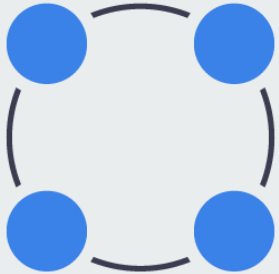
Статистика и анализ данных

Лекция 6. Непараметрические критерии

(22.10.2022)



Параметрические критерии

- Предполагают знание о виде распределения случайной величины
- t-критерий Стьюдента — данные должны быть нормально распределены
- Если данных много и они не скошены — можно использовать t-критерий Стьюдента
- А если данных мало или они специфичные?

Непараметрические критерии

Непараметрические критерии

- Не предполагают знание о виде распределения случайной величины
- Не знаем вид распределения нашей выборки — давайте перейдём к другой, но такой, чтобы мы понимали распределение

Критерий знаков

- Проверим, что медиана выборки равна некоторому числу m (одновыборочная версия)
- Не знаем про распределение выборки — не используем абсолютные значения выборки!
- Сравним значения с заданной медианой m — будем получать 0 и 1 (меньше m и больше m)
- Получаем новую бинарную выборку, ожидаем, что p = 1/2, так как сравнивали с медианой m
- Биномиальное распределение у новой выборки!

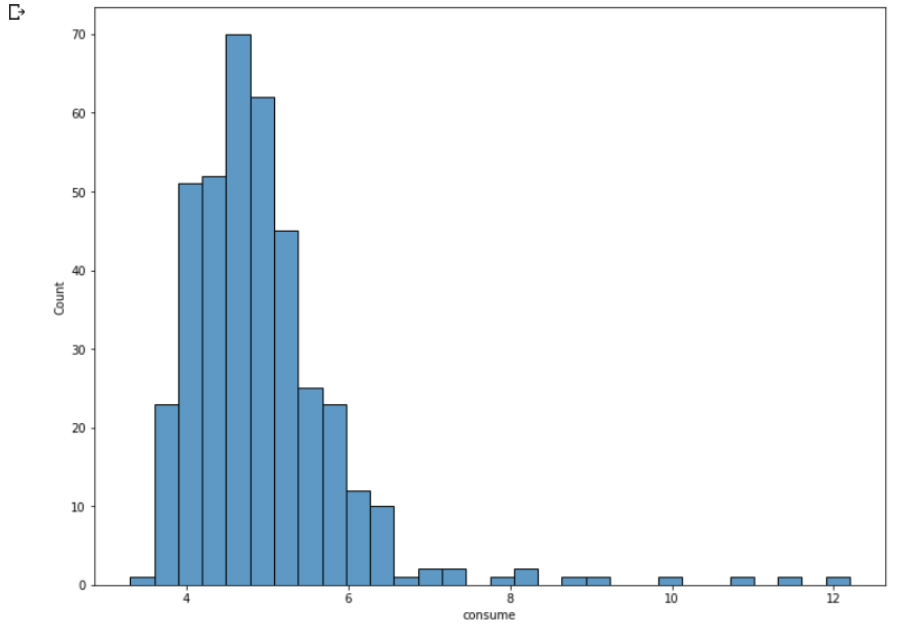
Критерий знаков

- Выборка: $X_1, \ldots, X_N \sim P$ (P — не известно)
- Нулевая гипотеза: median(X) = m
- Альтернативная гипотеза: median(X) ≠ m
- Статистика T_N
- Нулевое распределение: $T_N \sim Bin(N, 1/2)$

$$T_N = \sum_{i=1}^{N} [X_i > m]$$

Критерий знаков

```
sns.histplot(data["consume"], bins=30);
```



Критерий знаков

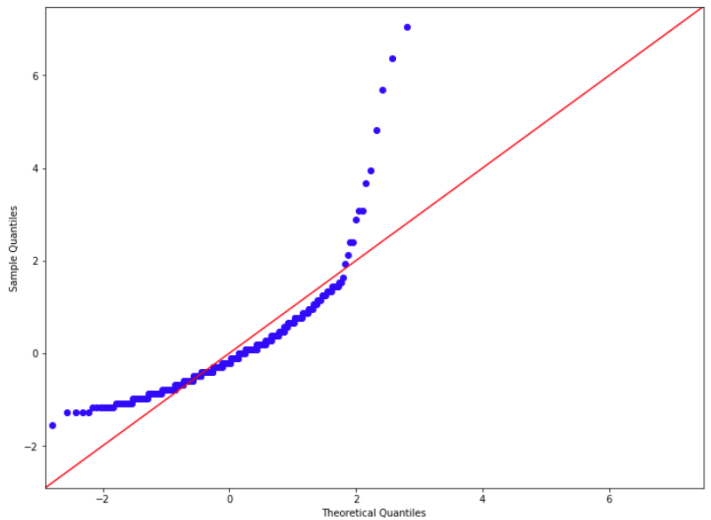
```
[23] from scipy.stats import shapiro

     shapiro(data["consume"])
```

```
ShapiroResult(statistic=0.7749733328819275, pvalue=1.0203466473862174e-22)
```

```
import statsmodels.api as sm

values = (data["consume"] - data["consume"].mean()) / data["consume"].std()
sm.qqplot(values, line="45");
```



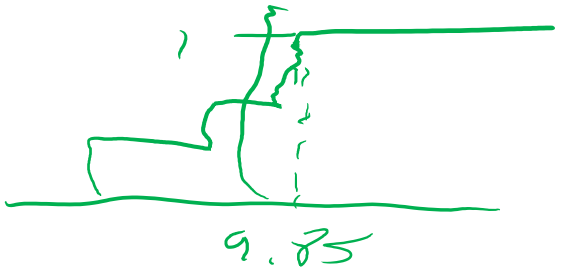
Критерий знаков

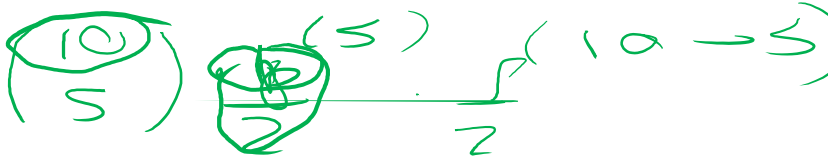
```
[31] from scipy.stats import binom

     m = 4.85
     N = len(data["consume"])
     tN = (data["consume"] > m).sum()

     binom(n=N, p=0.5).cdf(tN) * 2
```

```
0.028905970266677763
```



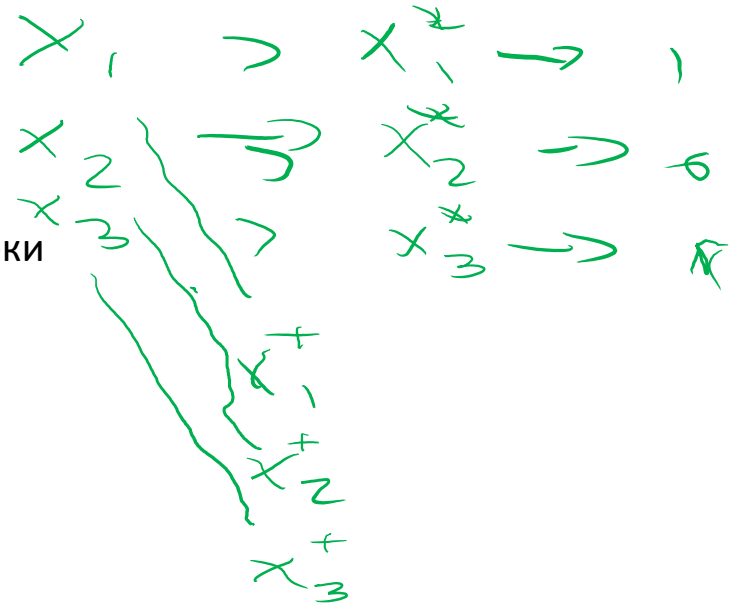


```
[32] from statsmodels.stats.descriptivestats import sign_test

     sign_test(data["consume"], mu0=m)
```

```
(-22.0, 0.028905970266677763)
```

Критерий знаков



- Выборка: $X_{11}, \ldots, X_{1N}, X_{21}, \ldots, X_{2N}$ — связанные выборки
- Нулевая гипотеза: $P(X_1 > X_2) = 1/2$
- Альтернативная гипотеза: $P(X_1 > X_2) \neq 1/2$
- Статистика T_N
- Нулевое распределение: $T_N \sim Bin(N, 1/2)$

$$T_N = \sum_{i=1}^{N} [X_{1i} > X_{2i}]$$

Критерий рангов

- Мы превратили выборку в выборку бинарных величин — потеряли часть информации и получили более слабый по мощности критерий
- Промежуточный вариант — отказаться от абсолютных значений, но сохранить порядок в выборке

Ранги

- Вариационный ряд — отсортированная по возрастанию выборка

  $X_1, \ldots, X_N \Rightarrow X_{(1)} \le \ldots < X_{(k1)} = \ldots = X_{(k2)} < \ldots \le X_{(N)}$

- Группы равных элементов — связки
- Если X_i не в связке, то $\text{rank}(X_i) = r: X_i = X_{(r)}$
- Если X_i в связке от k_1 до k_2, то $\text{rank}(X_i) = (k_1 + k_2) / 2$

1 2 4 4 5

Rank: 1 2 3.5 3.5 5

$\frac{3+4}{2} = 3.5$

Ранги

```
sample = data.head(10).copy()
sample["consume_rank"] = sample["consume"].rank()
sample[["consume", "consume_rank"]]
```

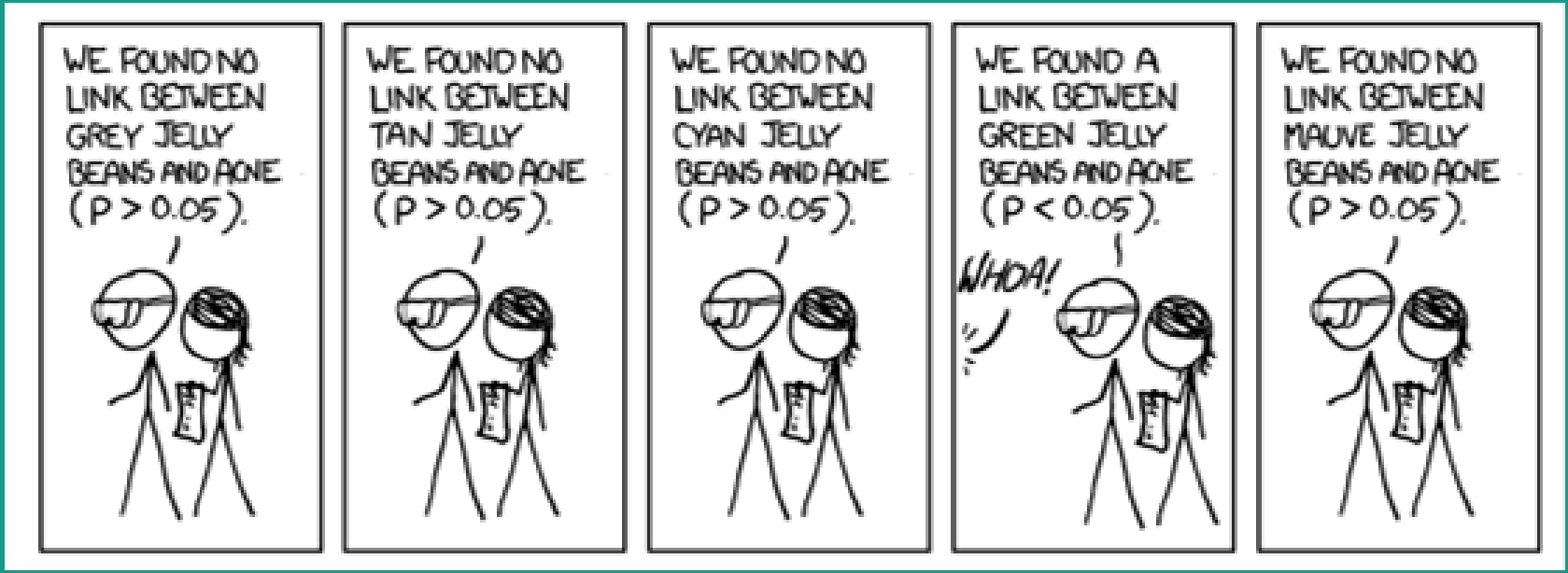
	consume	consume_rank
0	5.0	5.5
1	4.2	2.0
2	5.5	8.0
3	3.9	1.0
4	4.5	4.0
5	6.4	9.5
6	4.4	3.0
7	5.0	5.5
8	6.4	9.5
9	5.3	7.0

Критерий Манна-Уитни-Уилкоксона

- Есть 2 выборки X и Y, измеренных хотя бы в ранговой шкале.
- Выборки должны быть независимы
- H0: P(X > Y) = P(X < Y)
- Для этого вычисляется специальная U-статистика:
  - Честно считаем для всех возможных пар количество случаев, когда $x_i > y_j$, ситуации равенства считаем за 0.5, получим U1
  - Аналогично, перевернув знак, считаем U2
  - В качестве U берем минимум из этих величин
- Есть и другие методы подсчета U, менее вычислительно громоздкие

Для U-статистики есть специальные таблицы, дающие p-value по n_1 и n_2, однако для больших объемов выборок статистика имеет близкое к нормальному распределение.

Множественная проверка гипотез



Множественная проверка гипотез

- Научились проверять одну гипотезу
- А если гипотез не 1, а 100?

Множественная проверка гипотез

- Выбранный уровень значимости ($\alpha = 0.05$)
  - в теории - вероятность ложно отвергнуть нулевую гипотезу
  - на практике - шансы посчитать, что ваш препарат работает, когда это не так
- Что происходит, если гипотез 10?

Множественная проверка гипотез

- Оценим шансы ложно принять хотя бы 1 препарат за работающий
- Вероятность правильно не принять 1 препарат: $(1 - \alpha)$
- Вероятность правильно не принять 10 препаратов: $(1 - \alpha)^{10}$
- Вероятность ложно принять хотя бы 1 из 10 препаратов: $1 - (1 - \alpha)^{10}$

$1 - (0.95)^{10}$

- Для 10 препаратов, $\alpha = 0.05$: $P = 0.40126$

Множественная проверка гипотез

- Для 10 препаратов, $\alpha = 0.05$: $P = 0.40126$
- То есть вероятность ошибки первого рода 0.4!

- Если $N = 100$: $P = 0.99408$
- То есть найдется хотя бы 1 препарат, который будет лучше плацебо (но просто случайно)

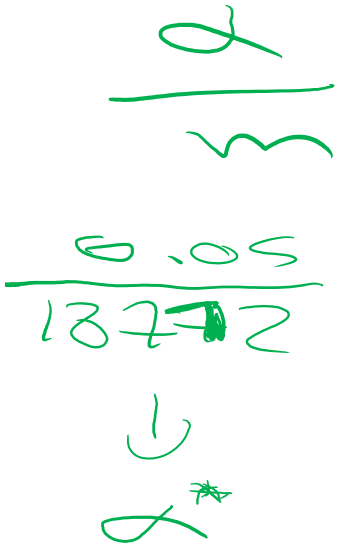
Ошибка первого рода

- Ранее хотели ошибку первого рода α
- Теперь хотим того же, но для группы экспериментов

- Групповая ошибка первого рода: $FWER(V > 0)$
- V – число ложноотвергнутых гипотез
- FamilyWise Error Rate

- Цель: $FWER(V > 0) \leq \alpha$
- Как добиться? Подобрать α_i для проверки гипотезы H_i

Поправка Бонферрони



- Возьмём новые уровни значимости: $\alpha_1 = ... = \alpha_m = \alpha / m$
- Вспомним вероятность отвергнуть хотя бы 1 из N гипотез:
  $N = 1 - (1 - \alpha/N) * N$
- Для N=10: 0.04889 ≈ 0.05
- Для N=100: 0.04878 ≈ 0.05

- Однако сильно уменьшает мощность тестов (то есть возможность детектировать эффект при его наличии)

Нисходящие методы проверки

- Посчитали уровни значимости для гипотез $H_1, ..., H_m$
- Отсортируем их и соответствующие им гипотезы:
- $p_{(1)} \le ... \le p_{(m)}$ $H_{(1)}, ..., H_{(m)}$
- Проверять будем по следующему алгоритму:
  - Если $p_{(1)} > \alpha_{(1)}$, то принимаем все нулевые гипотезы $H_{(1)}, ..., H_{(m)}$, иначе отвергаем $H_{(1)}$ и проверяем дальше
  - Если $p_{(2)} > \alpha_{(2)}$, то принимаем все нулевые гипотезы $H_{(2)}, ..., H_{(m)}$, иначе отвергаем $H_{(2)}$ и проверяем дальше
  - ...

Метод Холма

- Обеспечивает FWER на уровне α

$$\alpha_1 = \frac{\alpha}{m}$$

$$\alpha_2 = \frac{\alpha}{m-1}$$

$$\alpha_i = \frac{\alpha}{m-i+1}$$

$$\alpha_m = \alpha$$