Министерство науки и высшего образования Российской Федерации
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Математическая статистика

Расчетно-графическая работа №3

Вариант 3

Работу выполнил:
Рангулов Д.Э
Проверил:
___________ Береговенко. И.И.
«__»______________2024 г.

Санкт-Петербург 2024

Задание

Для каждой задачи необходимо провести два статистических теста, если не указано иное.

Первый критерий нужно реализовать самостоятельно (вычислить и вывести значение статистики, критическое значение и p-value). В качестве второго теста можно воспользоваться готовой реализацией. Также нужно отдельно указать, как формулируются H_0 и H_1 для выбранных тестов. Уровень значимости выбирается самостоятельно.

1. Предположите, какому вероятностному закону подчиняется распределение количества ходов. С помощью статистического теста подтвердите или опровергните это предположение.

2. Верно ли, что распределение количества ходов в рейтинговых и нерейтинговых играх одинаково?

3. Верно ли, что количество ходов уменьшается с увеличением разницы рейтинга?

Решение

Задание 1

- Для начала построим полигон ряда, чтобы понять, какое распределение рассматривать.

  Так как колонка $moves$ представлена в виде комбинации номеров клеток, то нужно ее преобразовать и создать новую колонку $move\_count$ с количеством ходов. Выполним это при помощи следующего кода:

Листинг 1: Преобразование данных

```
import pandas as pd

file_path = 'chess_games.csv'
data = pd.read_csv(file_path)
data['move_count'] = data['moves'].apply(lambda x: len(x.split()))
print(data['move_count'])
```

Сгруппируем данные по формуле Стёрджеса:

$n = 1 + \log_2(N)$

В нашем случае $N = 20058$. Следовательно $n = 16$.

Далее, рассчитаем частоты для каждого интервала и построим гистограмму, чтобы сделать предположение о распределении:

Листинг 2: Построение гистограммы

```
N = len(data)
k = int(1 + np.log2(N))
bins = np.linspace(move_counts.min(), move_counts.max(), k + 1)
plt.hist(data['move_count'], bins=bins, density=True, edgecolor='black',
    alpha=0.7, label='Hist')
```

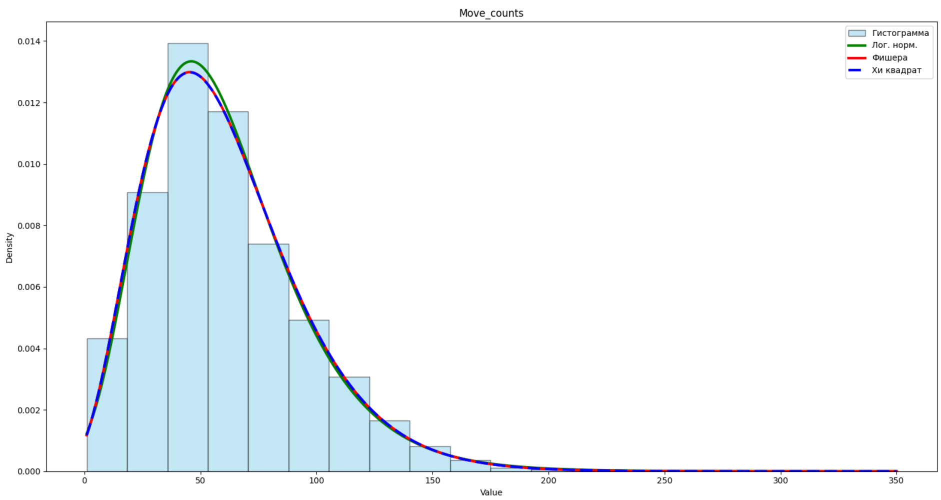

Рис. 2.1: Гистограмма

Также на графике изобразим распределения фишера, логнормальное и хи-квадрат.

Можно заметить, что есть сходства. Поэтому в качестве предположения, будем рассматривать эти два распределения.

- Найдем параметры распределения:

  Для нахождения параметров распределения, воспользуемся методом правдоподобия.

Для логнормального:

$$f = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}$$

$$L = \prod_{i=1}^{n} \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln x - \mu)^2}{2\sigma^2}}$$

Возьмем производные:

$$\frac{\delta \ln L}{\delta \mu} = \sum \frac{\ln x_i - \mu}{\sigma^2}=0$$

Следовательно

$$\mu = \frac{1}{n} \sum_{i=1}^{n} \ln x_i$$

По аналогии для производной по другому параметру, получим:

$$\sigma^2 = \frac{1}{n} \sum (\ln x_i - \mu)^2$$

В конечном итоге получаем: $\mu = 3.9$, $\sigma^2 = 0.72$

Для распределения Фишера и хи-квадрат в силу сложности вычислений будем использовать метод .fit() в Python.

Листинг 3: Метод правдоподобия

```
params_lognorm = lognorm.fit(data)
params_f = f.fit(data)
params_chi2 = stats.chi2.fit(data)
```

- Статистические тесты:

  Пусть $\alpha = 0.05$

  $H_0 : F_{\text{Э}}(x) = F_{\text{Т}}(x)$;

$H_1 : F_Э(x) \neq F_Т(x)$,

где Э - эмпирическая, Т - теоретическая функции распределения.

1. Воспользуемся критерием согласия Колмогорова:

   $D_n = sup|F_Э(x) - F_Т(x)|$,

   По теореме Колмогорова, при справедливости проверяемой гипотезы: $\lim(\sqrt{n}D_n \leq t) = K(t) = \sum_{j=-\infty}^{+\infty}(-1)^j e^{-2j^2t^2}$. Гипотеза H_0 отвергается, если статистика $\sqrt{n}D_n$ превышает квантиль распределения K_α, и принимается в обратном случае.

   Реализуем этот критерий в python.

Листинг 4: Построение функций распределения

```
sorted_data = np.sort(data)
efr = np.arange(1, n + 1) / n
params = lognorm.fit(data)
cdf_theoretical = lognorm.cdf(sorted_data, *params)
```

Таким образом efr - эмпирическая функция распределения, cdf_theoretical - теоретическая

Листинг 5: Реализация критерия Колмогорова

```
def ktest(efr, cdf_theoretical):
    K = np.max(np.abs(efr - cdf_theoretical))
    return K
```

Таким образом, получаем

$K_{\text{Лог.норм.}} = 0.0156$

$K_{\text{Фишер}} = 0.02$

$K_{\text{Хи-квадрат}} = 0.02$

Сравним с готовой реализацией kstest в Python:

Листинг 6: kstest

```
stat, p_value = stats.kstest(data, "lognorm", args=params)
```

Получили те же самые значения.

Посчитаем p_value:

$p\_value = 0.014$ - для Логнормального

$p\_value = 0.0005$ - для Фишера

$p\_value = 0.0005$ - для Хи квадрат.

Так как p_value во всех трех случаях меньше $\alpha = 0.05$, мы отклоняем гипотезу H_0 о том, что выборка соответствует какому-либо распределению из рассмотренных.

2. Рассмотрим критерий согласия Пирсона:

   Критерий согласия Пирсона выглядит следующим образом:
   $\chi^2 = \sum_{i=1}^{n} \frac{(n_i - n'_i)^2}{n'_i} \approx \chi_{k-1}$, где n_i - эмпирическая частота, n'_i - теоретическая.

   Статистика подчиняется распределению χ^2_r с $r = k - 1$ степенями свободы, если верна проверяемая гипотеза H_0

   Таким образом, наша задача состоит в том, чтобы посчитать эмпирические и теоретические частоты и затем определить наблюдаемое значение статистики.

   Помимо этого, для корректного применения критерия Пирсона, нужно обьединить интервалы, в которых значения частот меньше 5.

   Для начала нужно подсчитать частоты в интервалах.

   Количество и размер определим из формулы Стерджесса

   $n = 1 + \log_2(N)$

   В нашем случае $N = 20058$. Следовательно $n = 16$.

   Далее в коде ниже рассчитаем теоретические и эмпирические частоты и применим критерий:

Листинг 7: Подготовка данных к применению критерия

```
counts, bin_edges = np.histogram(data, bins=16)
shape, loc, scale = lognorm.fit(data, floc=0)
observed = counts
cdf_values = lognorm.cdf(bin_edges, shape, loc, scale)
expected = len(data) * np.diff(cdf_values)
chi2_stat, p_value = chisquare(f_obs=observed, f_exp=expected)
```

По итогам работы программы получаем:

stat = 2071.2, p_value = 0 для логнормального распределения.

stat = 399.58, p_value = 8.4e-76 для распределения Фишера.

stat = 399.49, p_value = 8.81e-76 для распределения хи-квадрат.

Таким образом, мы отклоняем гипотезу H_0 во всех случаях

Задание 2

Построим гистограммы двух выборок:

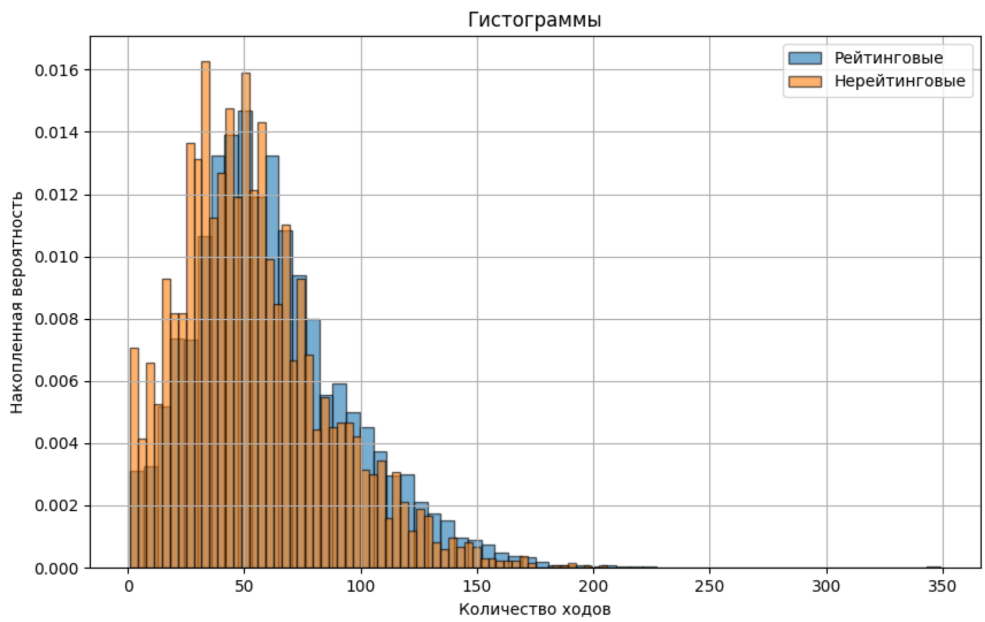

Рис. 3.1: Гистограмма

По графику можно сказать, что они имеют некоторые различия. Рейтинговые игры имеют более стандартизированные ходы, в отличие от нерейтинговых. Скорее всего, это связано, с человеческим фактором серьезности в рейт/нерейт играх.

1. Рассмотрим Критерий однородности Колмогорова-Смирнова:

   H_0: $F_1 = F_2$, то есть выборки распределены одинаково

   H_1: $F_1 \neq F_2$

   $\lambda_{\text{н}} = \sqrt{\frac{n_1 \cdot n_2}{n_1 + n_2}} \cdot sup|F_1(x) - F_2(x)|$, где F_1 - эмпирическая функция распределения по первой выборке объема n_1, F_2 - эмпирическая функция распределения по второй выборке объема n_2

   Тогда при $n_1, n_2 \to \infty$, и $F_1 = F_2$ величина

   $\sqrt{\frac{n_1 \cdot n_2}{n_1 + n_2}} \cdot sup|F_1(x) - F_2(x)| \to \lambda$, где λ имеет закон распределения, определенный функцией $K(\lambda)$

   При помощи кода ниже, разделим нашу data на две выборки в соответствии с заданием:

   Листинг 8: Подготовка данных к применению критерия

```
rated_games = data[data['rated'] == True]['move_count'].values
unrated_games = data[data['rated'] == False]['move_count'].values
```

   Используем готовую реализацию $stats.ks\_2samp(rated\_games, unrated\_games)$ из $scipy$

   Получаем,

   $stat = 0.10$

   $p\_value = 2.73e - 30$

   Таким образом, мы отклоняем гипотезу H_0 об однородности распределений.

2. Критерий однородности хи-квадрат

   Критерий однородности является универсальным и используется для проверки гипотезы о различии распределений двух и более совокупностей.

H_0: Распределение количества ходов одинаково для рейтинговых и нерейтинговых игр.

H_1: Распределение количества ходов неодинаково для рейтинговых и нерейтинговых игр.

$\chi^2 = \sum_{i=1}^{K}\sum_{j=1}^{N}\frac{(v_{i,j} - n_i p_j)^2}{n_i p_j}$, где где N - общее количество возможных значений, k - количество независимых выборок, n_i – объем i-ой выборки, $v_{i,j}$ - количество значений типа j в i-ой выборке

Поскольку вероятности p_j неизвестны, их можно заменить оценками, вычисленными при условии истинности H_0.

$p_j = \frac{\sum_{i=1}^{k} v_{ij}}{n}$

Для начала, составим таблицу сопряженности:

Листинг 9: Подготовка данных к применению критерия

```
nu = np.zeros((2, len(unique_move_counts)), dtype=int)

for i, move_count in enumerate(unique_move_counts):
    nu[0, i] = np.sum(rated_games == move_count)

for i, move_count in enumerate(unique_move_counts):
    nu[1, i] = np.sum(unrated_games == move_count)
```

Теперь можем применить критерий. Для этого, реализуем вручную формулы, представленные выше, посчитаем статистику и p_value

Листинг 10: Реализация критерия на Python

```
hat_p = np.sum(nu, axis=0) / np.sum(nu)
chi2_score = 0
for i in range(2):
  for j in range(4):
    chi2_score += np.power(nu[i,j] - np.sum(nu, axis=1)[i] * hat_p[j], 2) /
        (np.sum(nu, axis=1)[i] * hat_p[j])
sign_level = 0.05
print(f"chi2_score = {chi2_score}")
print(f"significance level = {sign_level}")
p_value = 1 - chi2.cdf(chi2_score, df=1 * 3)
print(f"p_value = {p_value}")
```

Имеем, $\chi^2_{\text{набл}} = 44.39$, $p\_value = 1.24e - 09$. Гипотеза H_0 отклоняется.

Задание 3

Рассмотрим два коэффициента корреляции:

Коэффициент корреляции Пирсона:

$$r_{xy} = \frac{\overline{XY} - \overline{X} \cdot \overline{Y}}{\sigma_x \sigma_y},$$

где $\overline{XY} = \dfrac{\sum xy}{n}$, $\overline{X} = \dfrac{\sum x}{n}$, $\overline{Y} = \dfrac{\sum y}{n}$, $\sigma_x = \sqrt{\overline{X^2} - (\overline{Y})^2}$, $\sigma_y = \sqrt{\overline{Y^2} - (\overline{X})^2}$, $\overline{X^2} = \dfrac{\sum x^2}{n}$, $\overline{Y^2} = \dfrac{\sum y^2}{n}$

Коэффициент ранговой корреляции Спирмена:

$$r_s = 10\frac{6\sum_i d_i^2}{n(n^2-1)},$$

где d_i - квадрат разности рангов $d_i = x_i - y_i$.

Рассмотрим гипотезы:

H_0: $r = 0$, т.е. коэффициент не является статистическим значимым/

$H_1 : r \neq 0$, коэффициент существенно отличен от нуля.

Проверка гипотезы осуществляется при помощи t-критерия Стьюдента:

$$t_{\text{набл}} = |r|\sqrt{\frac{n-2}{1-r^2}} \approx T_{n-2}$$

1. Рассмотрим коэффициент корреляции Пирсона и реализуем его на $Python$.

   Разделим данные по заданию, составив два массива $move\_counts$ и $rating\_diff$:

Листинг 11: Подготовка данных

```
data['move_counts'] = data['moves'].apply(lambda x: len(x.split()))
data['rating_diff'] = (data['white_rating'] + data['black_rating'])
```

Рассчитаем все величины:

Листинг 12: Расчет величин

```
n = len(data)
hat_XY = np.sum(data['move_counts'] * data['rating_diff']) / n
hat_X = np.sum(data['move_counts']) / n
hat_Y = np.sum(data['rating_diff']) / n
sigma_x = np.sqrt(np.mean(data['move_counts']**2) - np.mean(data['
    rating_diff'])**2)
sigma_y = np.sqrt(np.mean(data['rating_diff']**2) - np.mean(data['
    move_counts'])**2)
```

Наконец, посчитаем коэффициент:

Листинг 13: Расчет коэффициента

```
r_xy = (hat_XY - hat_Y * hat_X) / (sigma_y * sigma_x)
```

Получаем $r_xy = 0.1605$

Рассчитаем $t_{\text{набл}}$:

Листинг 14: Расчет статистики

```
t_nabl = abs(r_xy) * np.sqrt((n-2) / (1 - r_xy**2))
```

Получаем $t_{\text{набл}} = 23.03$. Так как $n = 20058, \alpha = 0.05$, то $t_{\text{крит}} = 1.96$.

Область одностороняя, имеем, что $t_{\text{набл}} > t_{\text{крит}}$. Следовательно зависимость между величинами является статистически значимой(принимаем гипотезу H_1).

Проанализируем получившееся значение $r_{xy} = 0.16$. Оно меньше, чем 0.3, значит зависимость есть, но очень слабая. Помимо этого, так как коэффициент положительный(положительная корреляция), то зависимость выглядит так: с увеличением разницы в рейтинге, количество ходов увеличивается.

2. Критерий Спирмена:

   Для критерия Спирмена, воспользуемся готовой реализацией в Python:

Листинг 15: Рассчет коэффициента

```
r = data[['rating_diff', 'move_counts']].corr(method='spearman')
print(r)
```

Получаем $r = 0.165$.

По аналогии с прошлым пунктом рассчитаем статистику:

Листинг 16: Расчет статистики

```
t_nabl = abs(r_xy) * np.sqrt((n-2) / (1 - r_xy**2))
```

Получаем, что $t_{\text{набл}} = 23.69$. Как и в прошлом пункте$t_{\text{набл}} > t_{\text{крит}}$. Выводы остаются теми же.

Вывод по работе

В РГР были рассмотрены статистические тесты на реальных датасетах о шахматах. Были применены различные критерии в зависимости от требуемого задания.