Лабораторная работа №4

Выполнил: Величко Максим Иванович, М32061

Варианты для задания: 4.

Весь код для удобства выложен на мой GitHub(https://github.com/maksve11/ITMO\_MatStat\_sem4/tree/main/lab4)

Для каждого вопроса требуется формализовать задачу и проверить статистическую гипотезу двумя критериями, если не сказано иное. В файле song\_data.csv (взято отсюда) приведены данные о музыкальных произведениях.

№1 Разумно ли популярность песни аппроксимировать нормальным законом?

Задача: проверить, можно ли аппроксимировать распределение популярности песен в наборе данных нормальным законом.

Для этого можно использовать два критерия:

1. Критерий Колмогорова-Смирнова: сравнивает эмпирическую функцию распределения (ECDF) с теоретической функцией распределения нормального закона. Нулевая гипотеза - выборка распределена нормально.
2. Критерий Шапиро-Уилка: проверяет гипотезу о том, что выборка была взята из нормального распределения. Нулевая гипотеза - выборка распределена нормально.

Математически:

H0: выборка распределена нормально H1: выборка не распределена нормально

Если p-value меньше заданного уровня значимости (обычно 0.05), то мы отвергаем нулевую гипотезу в пользу альтернативной - выборка не распределена нормально.

Вот код, который проверяет наши гипотезы:

import pandas as pd  
from scipy.stats import shapiro, kstest, norm  
import matplotlib.pyplot as plt  
import numpy as np  
from statsmodels.graphics.gofplots import qqplot  
  
# Загрузка данных из файла song\_data.csv  
data = pd.read\_csv("song\_data.csv", delimiter=",")  
  
# Построение гистограммы распределения популярности песен  
plt.hist(data['song\_popularity'], bins=20)  
plt.xlabel('Популярность')  
plt.ylabel('Количество песен')  
plt.title('Распределение популярности песен')  
plt.show()  
  
# Критерий Колмогорова-Смирнова  
kstest\_result = kstest(data['song\_popularity'], 'norm')  
  
print("Критерий Колмогорова-Смирнова: ")  
if kstest\_result.pvalue < 0.05:  
 print("Распределение популярности песен отличается от нормального закона.")  
else:  
 print("Распределение популярности песен можно аппроксимировать нормальным законом.")  
  
# Критерий Шапиро-Уилка  
shapiro\_result = shapiro(data['song\_popularity'])  
  
print("\nКритерий Шапиро-Уилка: ")  
if shapiro\_result.pvalue < 0.05:  
 print("Распределение популярности песен отличается от нормального закона.")  
else:  
 print("Распределение популярности песен можно аппроксимировать нормальным законом.")  
  
# Построение графика Q-Q plot для проверки нормальности распределения  
qqplot(data['song\_popularity'], line='s')  
plt.title('QQ-график популярности песен')  
plt.xlabel('Квантили теоретического распределения')  
plt.ylabel('Квантили выборки')  
plt.show()

Результаты:

Изображение выглядит как диаграмма

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как диаграмма

Автоматически созданное описание

№2. Отличается ли принципиально распределение рейтинга песни в зависимости от продолжительности (разбейте условно на ”длинные” и ”короткие”, порог выбирайте сами)?

Задача: проверить, отличается ли распределение рейтинга песни в зависимости от продолжительности. Для этого условно разобьем песни на две категории: "длинные" и "короткие", используя какой-то порог по продолжительности (например, 3 минуты). Далее сравним распределение рейтинга для двух категорий песен с помощью статистических тестов.

Математический подход: пусть X - это случайная величина, представляющая рейтинг песни. Для каждой категории песен определим ее распределение вероятностей P(X) и проверим гипотезу о том, что две группы распределены одинаково:

H0: P(X\_длинные) = P(X\_короткие) (распределения рейтинга для длинных и коротких песен одинаковы) H1: P(X\_длинные) != P(X\_короткие) (распределения рейтинга для длинных и коротких песен различаются)

Для проверки данной гипотезы мы будем использовать два статистических критерия: t-критерий Стьюдента и U-критерий Манна-Уитни.

Вот код, который проверяет наши гипотезы:

import pandas as pd  
import scipy.stats as stats  
  
# Загрузка данных  
data = pd.read\_csv("song\_data.csv", delimiter=',')  
  
# Задаем порог продолжительности для разбиения песен на две категории  
duration\_threshold = 180000 # 3 минуты в милисекундах  
  
# Создаем две выборки: длинные и короткие песни  
long\_songs = data[data["song\_duration\_ms"] >= duration\_threshold]["song\_popularity"]  
short\_songs = data[data["song\_duration\_ms"] < duration\_threshold]["song\_popularity"]  
  
# Проверка гипотезы с помощью t-критерия Стьюдента  
t\_stat, p\_value = stats.ttest\_ind(long\_songs, short\_songs, equal\_var=False)  
print("t-критерий Стьюдента:")  
print("t-statistic =", t\_stat)  
print("p-value =", p\_value)  
if p\_value < 0.05:  
 print("Различие в распределении рейтинга для длинных и коротких песен статистически значимо.")  
else:  
 print("Различие в распределении рейтинга для длинных и коротких песен статистически незначимо.")  
  
# Проверка гипотезы с помощью U-критерия Манна-Уитни  
u\_stat, p\_value = stats.mannwhitneyu(long\_songs, short\_songs, alternative="two-sided")  
print("\nU-критерий Манна-Уитни:")  
print("U-statistic =", u\_stat)  
print("p-value =", p\_value)  
if p\_value < 0.05:  
 print("Различие в распределении рейтинга для длинных и коротких песен статистически значимо.")  
else:  
 print("Различие в распределении рейтинга для длинных и коротких песен статистически незначимо.")

Результаты:

Изображение выглядит как текст

Автоматически созданное описание

№3. Зависит ли популярность песни от продолжительности?

Задача: проверить гипотезу о том, зависит ли популярность песни от ее продолжительности.

Формально:

H0: средняя популярность длинных и коротких песен не отличается.

H1: средняя популярность длинных и коротких песен отличается.

Для решения этой задачи используются два критерия: t-критерий Стьюдента и U-критерий Манна-Уитни.

Вот код, который проверяет наши гипотезы:

import pandas as pd  
import scipy.stats as stats  
import seaborn as sns  
from matplotlib import pyplot as plt  
  
# Загрузка данных  
data = pd.read\_csv("song\_data.csv", delimiter=',')  
  
# Задаем порог продолжительности для разбиения песен на две категории  
duration\_threshold = 180000 # 3 минуты в милисекундах  
  
# Создаем две выборки: длинные и короткие песни  
long\_songs = data[data["song\_duration\_ms"] >= duration\_threshold]["song\_popularity"]  
short\_songs = data[data["song\_duration\_ms"] < duration\_threshold]["song\_popularity"]  
  
# Проверка гипотезы с помощью t-критерия Стьюдента  
t\_stat, p\_value = stats.ttest\_ind(long\_songs, short\_songs, equal\_var=False)  
print("t-критерий Стьюдента:")  
print("t-statistic =", t\_stat)  
print("p-value =", p\_value)  
if p\_value < 0.05:  
 print("Различие в распределении рейтинга для длинных и коротких песен статистически значимо.")  
else:  
 print("Различие в распределении рейтинга для длинных и коротких песен статистически незначимо.")  
  
# Проверка гипотезы с помощью U-критерия Манна-Уитни  
u\_stat, p\_value = stats.mannwhitneyu(long\_songs, short\_songs, alternative="two-sided")  
print("\nU-критерий Манна-Уитни:")  
print("U-statistic =", u\_stat)  
print("p-value =", p\_value)  
if p\_value < 0.05:  
 print("Различие в распределении рейтинга для длинных и коротких песен статистически значимо.")  
else:  
 print("Различие в распределении рейтинга для длинных и коротких песен статистически незначимо.")  
  
# Построение гистограмм  
sns.histplot(long\_songs, kde=False, label="Длинные песни")  
sns.histplot(short\_songs, kde=False, label="Короткие песни")  
plt.legend()  
plt.title("Распределение рейтинга для длинных и коротких песен")  
plt.xlabel("Рейтинг")  
plt.ylabel("Частота")  
plt.show()  
  
# Построение Boxplot  
sns.boxplot(x="song\_duration\_ms", y="song\_popularity", data=data, whis=1.5)  
plt.title("Boxplot для длинных и коротких песен")  
plt.xlabel("Продолжительность (мс)")  
plt.ylabel("Рейтинг")  
plt.show()

Вот результаты:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как диаграмма

Автоматически созданное описание

Изображение выглядит как диаграмма

Автоматически созданное описаниеСлишком много данных, поэтому в Boxplot продолжительность отображается некорректно, можно зазумить, но увы, производительности не хватает.