ДЗ 2

Дмитрий Фунштейн гр. БИТ212

14 июня 2023 г.

Задача 1. *Дан файл, в котором указан рост 150 студентов. Построить гистограмму с разбиением на 12 ячеек в промежутке от 160 до 190 см. Найти выборочное среднее, выборочную дисперсию, построить интервальную оценку для математического ожидания и дисперсии. Предположим, что рост студентов подчинен нормальному распределению с найденным средним и дисперсией. Построить данную функцию плотности вероятности на том же графике, что и гистограмма. Проверить гипотезу о том, что рост студентов распределен в соответствии с данным законом распределения, с помощью критерия согласия "хи-квадрат".*

Выполним задание на ЯП Python с использованием библиотек Pandas, SciPy и NumPy. Графики будем рисовать с помощью библиотеки matplotlib через pyplot и средства Pandas.

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

plt.figure()

# Читаем данные из дано и делаем выборку по значениям,
# которые входят в промежуток от 160 до 190
df = pd.read_csv("data.dat")
sample = df[(df["Height"] >= 160) & (df["Height"] <= 190)]
n = sample["Height"].size
print(f"Sample size: {n}")

# Задаем уровень доверительного интервала
alpha = 0.05

# Строим гистограмму роста
bin_edges = np.linspace(160, 190, 13)

ax1 = sample["Height"].plot.hist(density=True, bins=bin_edges, ec="black", grid=True)
ax1.set_xlabel("Height (cm)")
ax1.set_ylabel("Density")
ax1.set_title("Height histogram")
```

```python
# Ищем выборочное среднее и выборочную дисперсию
# с помощью методов mean() и var() соответственно
# Также ищем стандартную ошибку среднего
mean = sample["Height"].mean()
var = sample["Height"].var()
std = sample["Height"].std()
std_error = std / np.sqrt(n)

print(f"Sample mean: {mean}")
print(f"Sample variance: {var}")

# Ищем критическое значение t для уровня доверительного интервала
# и числа степеней свободы
t_crit = stats.t.ppf(1 - alpha / 2, n-1)

# Ищем критические значения хи-квадрат распределения для уровня
# доверительного инвервала и числа степеней свободы
chi2_lower = stats.chi2.ppf(alpha / 2, n-1)
chi2_upper = stats.chi2.ppf(1 - alpha / 2, n-1)

# Ищем интервальную оценку для мат. ожидания
lower_mean = mean - t_crit * std_error
upper_mean = mean + t_crit * std_error
print(f"Interval estimation (mean): ({lower_mean}, {upper_mean})")

# Ищем интервальную оценку для дисперсии
lower_var = (n-1) * var / chi2_upper
upper_var = (n-1) * var / chi2_lower
print(f"Interval estimation (var): ({lower_var}, {upper_var})")

# Строим функцию плотности вероятности на том же графике,
# что и гистограмма. Сначала вычисляем значения плотности вероятности
# для каждого бина, а после - наносим на график
x = np.linspace(160, 190, 1000)
pdf = stats.norm.pdf(x, loc=mean, scale=np.sqrt(var))
ax1.plot(x, pdf, color="red")

plt.savefig("hist.png")
plt.show()

# Вычисляем наблюдаемые частоты
# и ищем ожидаемые частоты для каждого инвервала
observed, _ = np.histogram(sample["Height"], bins=bin_edges)
expected = n * np.diff(stats.norm.cdf(bin_edges, loc=mean, scale=std))

# Ищем статистику критерия хи-квадрат
chisq_statistic = np.sum((observed - expected)**2 / expected)
print(f"Chi-squared statistic: {chisq_statistic}")
```

```
# Ищем критическое значение хи-квадрат для заданного уровня значимости
df = len(observed) - 1
critical_value = stats.chi2.ppf(q=1-alpha, df=df)
print(f"Chi-squared critical value: {critical_value}")

# Определяем, принимаем ли мы или отвергаем гипотезу
if chisq_statistic > critical_value:
    result = "Reject"
else:
    result = "Accept"

print(f"Chi-squared test: {result}")
```

В результате исполнения написанного кода получились следующие результаты Оценка для выборочного среднего:

$$\bar{X} = 174.35241891891891$$

Оценка для выборочной дисперсии:

$$S^2 = 30.89405296617027$$

Интервальные оценки для математического ожидания и для дисперсии:

Математическое ожидание	$(173.4495082169535, 175.25532962088434)$
Дисперсия	$(24.890141243109706, 39.37900475029282)$

Статистика критерия хи-квадрат:

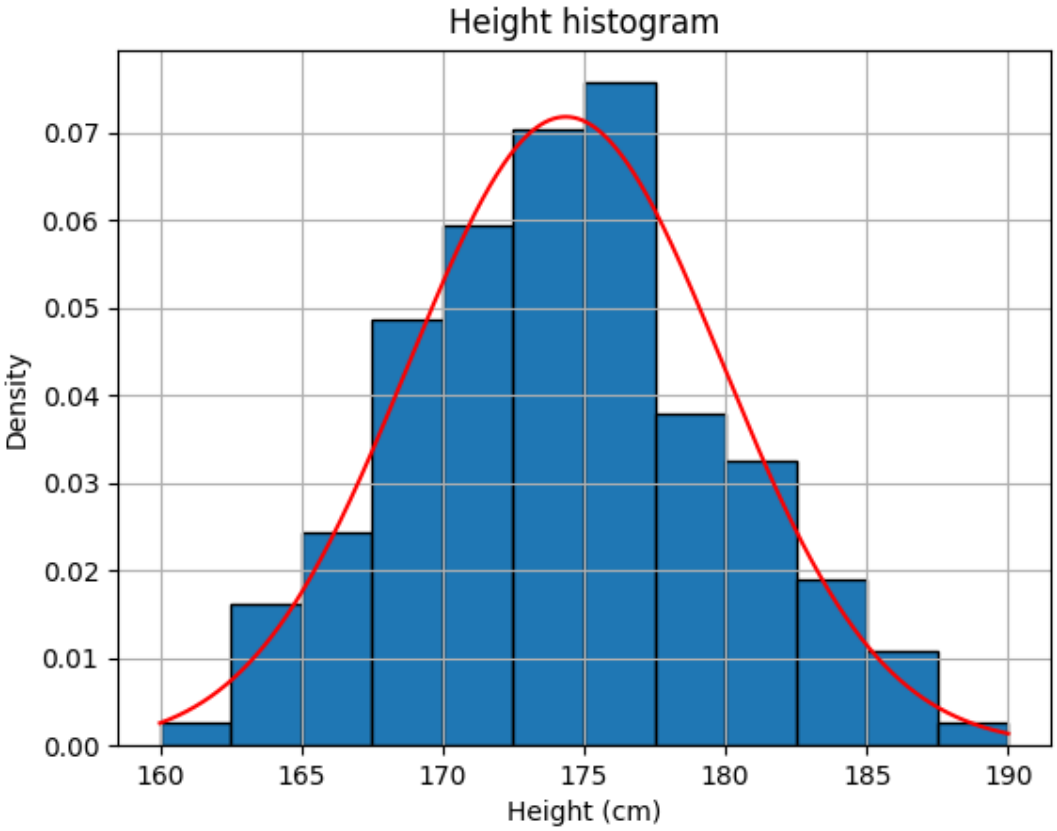
$$\chi^2 = 3.640111044623316$$

Критическое значение хи-квадрат:

$$\chi^2_k = 19.67513757268249$$

На основании полученных данных мы принимаем гипотезу о том, что рост студентов распределен в соответствии с данным законом распределения.

Полученная гистограмма путем исполнения программы:



□