

Лист 8

Шевкунов Даниил ПМ22-4

```
In [64]: import scipy.stats as stats
import math
import numpy as np
import pandas as pd
```

```
In [30]: def zstat(alpha):
return stats.norm.ppf(1 - (1-alpha) / 2)
```

```
In [18]: # 1
# Исходные данные
mean = 187.9
std = 32.4
n = 9

# а) Найдем 80%-ный доверительный интервал
interval_80 = stats.norm.interval(0.8, loc=mean, scale=std / n**0.5)

# б) Найдем уровень доверия для заданного интервала
Z = (210 - mean) * n**0.5 / std
confidence_level = 1 - 2 * (1 - stats.norm.cdf(Z))

print("а) 80%-ный доверительный интервал:", interval_80)
print(f"б) Уровень доверия: {confidence_level:.3%}")
```

а) 80%-ный доверительный интервал: (174.05924309211832, 201.7407569078817)
б) Уровень доверия: 95.927%

```
In [37]: # 2

# Уровень доверия
confidence_level = 0.98

# Максимальная допустимая ошибка
max_error = 0.025

# Предполагаемая доля
p = 0.5

# Вычисление объема выборки
n = math.ceil((zstat(confidence_level) ** 2 * p * (1 - p)) / max_error**2)

print("Объем выборки:", n)
```

Объем выборки: 2165

```
In [46]: # 3
sample_size = 400
```

```

man_mean = 177.8
man_std = 7.62
women_mean = 165.1
women_std = 6.35

women_gr_man = 380

# Доля пар где жена выше
wgh = women_gr_man / sample_size

Z = zstat(0.95)

error = Z * ((wgh * (1 - wgh)) / sample_size) ** 0.5

confidence_interval = (wgh - error, wgh + error)
print("а) Доверительный интервал:", confidence_interval)

# Вычисление вероятности что случайная замужняя женщина выше случайного женатого мужчины
probability_wife_above_husband = 1 - stats.norm.cdf(
    0, loc=women_mean - man_mean, scale=math.sqrt(women_std**2 + man_std**2)
)
print("б) Вероятность, что женщина выше:", probability_wife_above_husband)

```

а) Доверительный интервал: (0.9286417876460265, 0.9713582123539735)
б) Вероятность, что женщина выше: 0.10020773084642709

In [42]:

```

# 4
# Заданные данные
x = [-2, 1, 2, 3, 4, 2, 5]
n = [2, 1, 2, 2, 2, 1, 1]
sample = np.repeat(x, n)

# Значение Z для уровня доверия 95%
z_value = zstat(0.95)

# Вычисление доверительного интервала
confidence_interval = (
    sample.mean() - z_value * (sample.std() / np.sqrt(len(sample))),
    sample.mean() + z_value * (sample.std() / np.sqrt(len(sample))),
)

print("Доверительный интервал:", confidence_interval)

```

Доверительный интервал: (0.7151362464869317, 3.2848637535130685)

In [55]:

```

# 5

# Курсы доллара за последние пять дней
exchange_rates = np.array([99.25, 100.64, 103.44, 101.35, 97.65])

# Вычисление среднего значения и стандартного отклонения
mean = np.mean(exchange_rates)
std = np.std(exchange_rates, ddof=1)

```

```
# Стоимость вещи в долларах
cost_in_dollars = 719.34

# Вычисление квантиля нормального распределения
# price = stats.norm.ppf(0.99, loc=mean, scale=std)
price = mean + zstat(0.99) * std

# Вычисление минимальной суммы в рублях
print("Минимальная сумма в рублях:", price * cost_in_dollars)
```

Минимальная сумма в рублях: 76314.40116531836

In [60]:

```
# 6

# Исходные данные
n = 5 # Размер выборки
mean = 20 # средний годовой рост
std = 5 # стандартное отклонение годового роста
confidence_level = 0.97 # уровень доверия
initial_price = 100 # начальная цена актива в долларах

# Значение Z-статистики для 97% уровня доверия
z_value = zstat(confidence_level)

# Вычисление доверительного интервала
error = z_value * (std / math.sqrt(n))
confidence_interval = (mean - error, mean + error)

# Вычисление доверительного интервала для цены актива в конце следующего года
final_price_interval = (
    initial_price * (1 + confidence_interval[0] / 100),
    initial_price * (1 + confidence_interval[1] / 100),
)

print("97%-ый доверительный интервал для годового роста: ", confidence_interval)
print(
    "97%-ый доверительный интервал для цены актива в конце следующего года: ",
    final_price_interval,
)
```

97%-ый доверительный интервал для годового роста: (15.147530398402736, 24.852469601597264)

97%-ый доверительный интервал для цены актива в конце следующего года: (115.14753039840274, 124.85246960159726)

In [63]:

```
# 7

# Исходные данные
sample_size = 19
sample_std = 25
confidence_level = 0.90

# Вычисление квантилей хи-квадрат
# А руками бы искали в таблице...
chi2_lower = stats.chi2.ppf((1 - confidence_level) / 2, df=sample_size - 1)
chi2_upper = stats.chi2.ppf((1 + confidence_level) / 2, df=sample_size - 1)

# Вычисление доверительного интервала
```

```

confidence_interval = (
    ((sample_size - 1) * sample_std**2 / chi2_upper) ** 0.5,
    ((sample_size - 1) * sample_std**2 / chi2_lower) ** 0.5,
)

print(
    f"90% доверительный интервал для среднего квадратичного отклонения: {confidence_in
)

```

90% доверительный интервал для среднего квадратичного отклонения: (19.74049955262717, 34.61249782279546)

```

In [91]: # 8
data = pd.read_csv(
    r"D:\DISTR\Загрузки\iMe Desktop\sample88.csv",
    header=None,
    names=['x', 'y'],
    decimal=";",
    sep=";",
    encoding="cp1251",
)

```

```

In [117]: dif_x = data.x - data.x.mean()
dif_y = data.y - data.y.mean()

r = sum(dif_x * dif_y) / (sum(dif_x**2) * sum(dif_y**2)) ** 0.5
print(f"a) Выборочный коэффициент корреляции: {r:.5f}")

```

a) Выборочный коэффициент корреляции: 0.94304

```

In [121]: # Трансформация r в z
z = 0.5 * math.log((1 + r) / (1 - r))

# Размер выборки
n = data.shape[0]
# Уровень доверия
confidence_level = 0.92

# Квантили стандартного нормального распределения
z_alpha_2 = zstat(confidence_level)

# Стандартная ошибка z
se_z = 1 / math.sqrt(n - 3)

# Доверительный интервал для z
z_interval_lower = z - z_alpha_2 * se_z
z_interval_upper = z + z_alpha_2 * se_z

# Обратное преобразование интервала для r
r_interval_lower = (math.exp(2 * z_interval_lower) - 1) / (
    math.exp(2 * z_interval_lower) + 1
)
r_interval_upper = (math.exp(2 * z_interval_upper) - 1) / (
    math.exp(2 * z_interval_upper) + 1
)

```

```
print(  
    f"6) Доверительный интервал коэффициента корреляции: ({r_interval_lower}, {r_interval_upper})  
)
```

6) Доверительный интервал коэффициента корреляции: (0.8976670776032922, 0.9686278167868622)

