

```

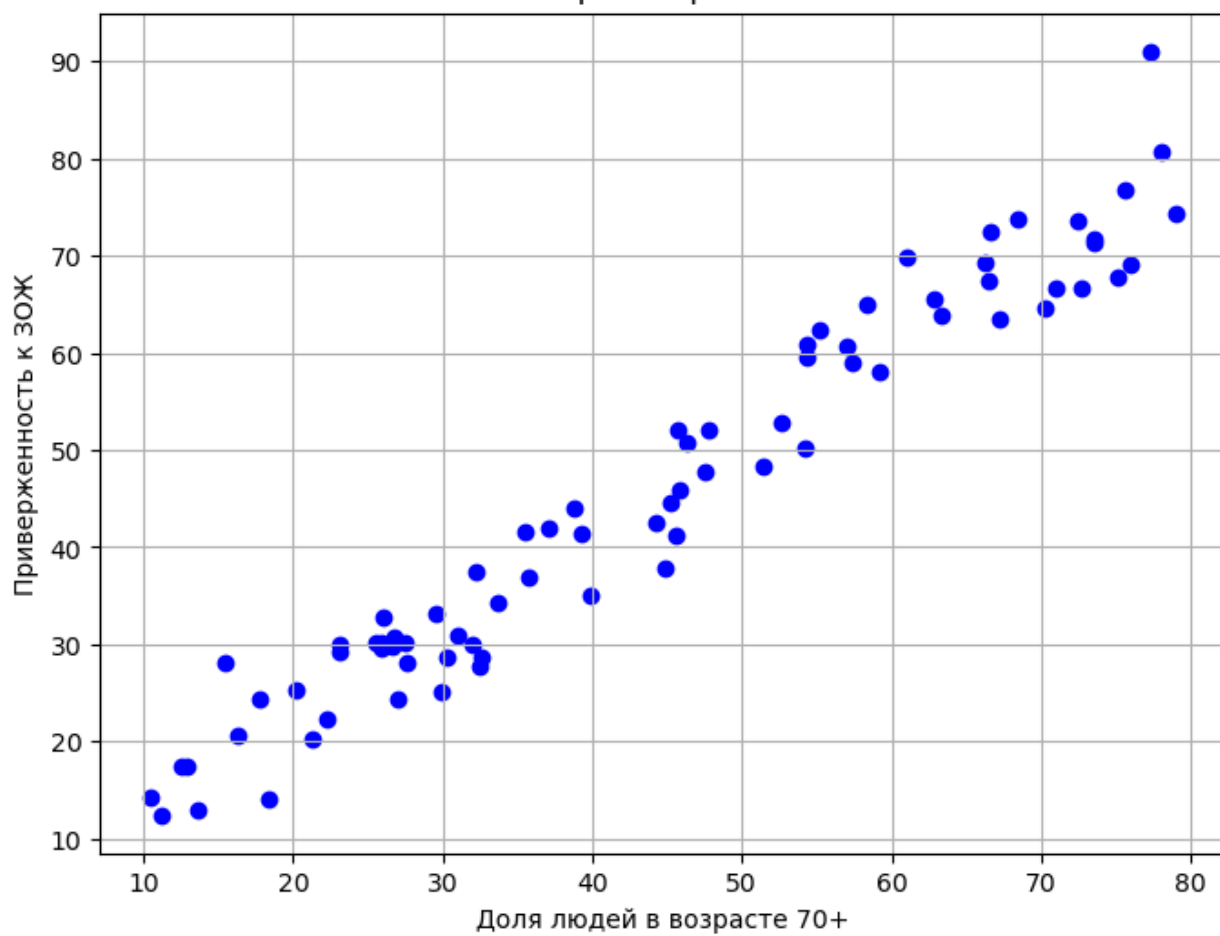
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress

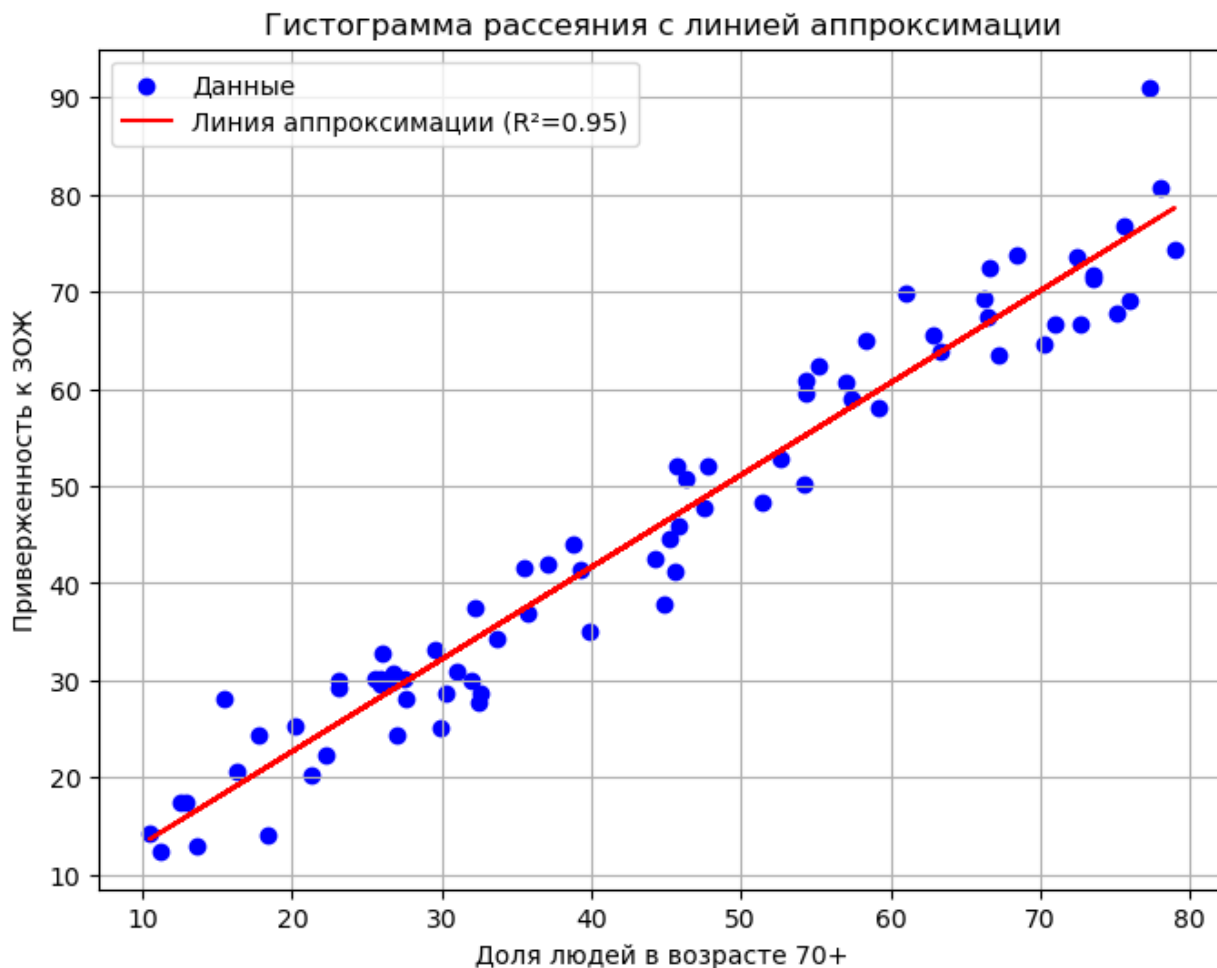
data =
pd.read_excel("C:/Users/Максим/Downloads/regions_data_updated.xlsx")
correlation = data["Доля людей в возрасте
70+"].corr(data["Приверженность к 30Ж"])
print(f"Коэффициент корреляции: {correlation}")
if correlation > 0.7:
    print("Сильная положительная связь")
elif correlation < -0.7:
    print("Сильная отрицательная связь")
else:
    print("Слабая или отсутствует линейная связь")
plt.figure(figsize=(8, 6))
plt.scatter(data["Доля людей в возрасте 70+"], data["Приверженность к
30Ж"], color='blue', label='Данные')
plt.xlabel('Доля людей в возрасте 70+')
plt.ylabel('Приверженность к 30Ж')
plt.title('Гистограмма рассеяния')
plt.grid(True)
plt.show()
slope, intercept, r_value, p_value, std_err = linregress(data["Доля
людей в возрасте 70+"], data["Приверженность к 30Ж"])
plt.figure(figsize=(8, 6))
plt.scatter(data["Доля людей в возрасте 70+"], data["Приверженность к
30Ж"], color='blue', label='Данные')
plt.plot(data["Доля людей в возрасте 70+"], slope * data["Доля людей в
возрасте 70+"] + intercept, color='red', label=f'Линия аппроксимации
(R²={r_value**2:.2f})')
plt.xlabel('Доля людей в возрасте 70+')
plt.ylabel('Приверженность к 30Ж')
plt.title('Гистограмма рассеяния с линией аппроксимации')
plt.legend()
plt.grid(True)
plt.show()
print(f"Коэффициент детерминации (R²): {r_value**2:.2f}")

```

Коэффициент корреляции: 0.9746345310326
Сильная положительная связь

Гистограмма рассеяния





Коэффициент детерминации (R^2): 0.95

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

data_user =
pd.read_excel("C:/Users/Максим/Downloads/regions_data_updated.xlsx")
r_value_user = linregress(data_user["Доля людей в возрасте 70+"],
data_user["Приверженность к 30Ж"]).rvalue
correlation_user = data_user["Доля людей в возрасте
70+"].corr(data_user["Приверженность к 30Ж"])
print(f"Коэффициент корреляции: {correlation_user}")
if correlation_user > 0.7:
    print("Сильная положительная связь")
elif correlation_user < -0.7:
    print("Сильная отрицательная связь")
else:
    print("Слабая или отсутствует линейная связь")
plt.figure(figsize=(8, 6))
```

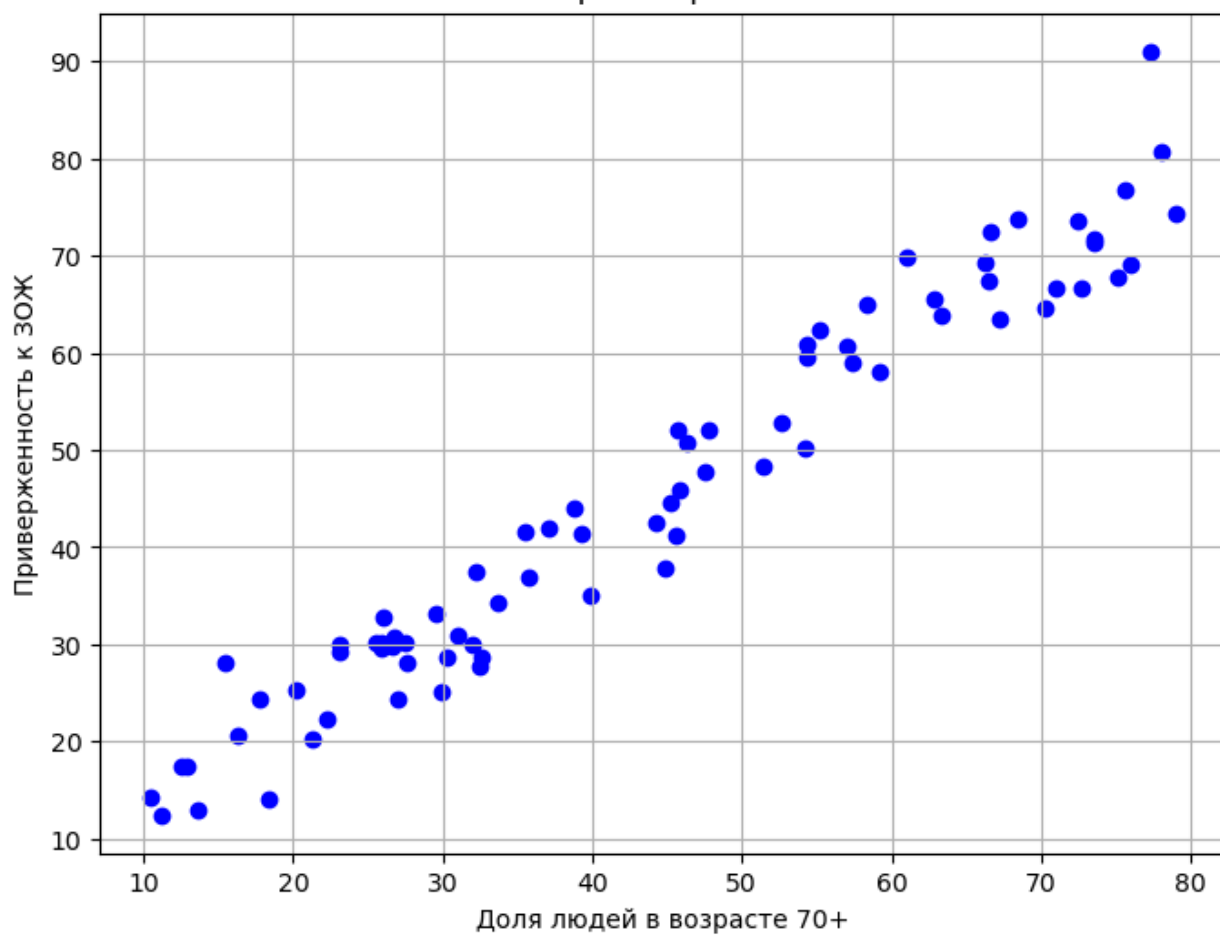
```

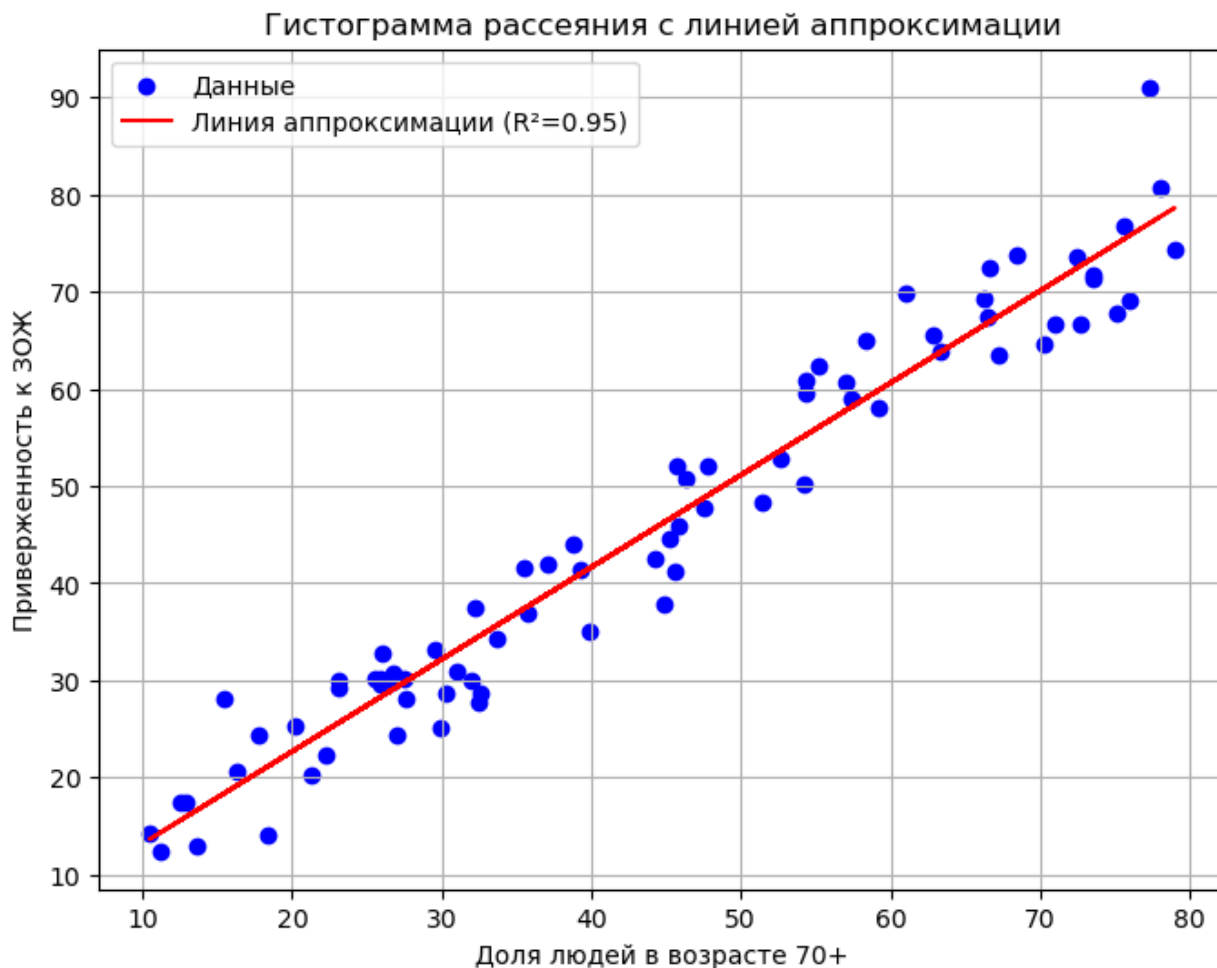
plt.scatter(data_user["Доля людей в возрасте 70+"],
data_user["Приверженность к 30Ж"], color='blue', label='Данные')
plt.xlabel('Доля людей в возрасте 70+')
plt.ylabel('Приверженность к 30Ж')
plt.title('Гистограмма рассеяния')
plt.grid(True)
plt.show()
slope_user, intercept_user, r_value_user, p_value_user, std_err_user =
linregress(data_user["Доля людей в возрасте 70+"],
data_user["Приверженность к 30Ж"])
plt.figure(figsize=(8, 6))
plt.scatter(data_user["Доля людей в возрасте 70+"],
data_user["Приверженность к 30Ж"], color='blue', label='Данные')
plt.plot(data_user["Доля людей в возрасте 70+"], slope_user *
data_user["Доля людей в возрасте 70+"] + intercept_user, color='red',
label=f'Линия аппроксимации ( $R^2={r\_value\_user**2:.2f}$ )')
plt.xlabel('Доля людей в возрасте 70+')
plt.ylabel('Приверженность к 30Ж')
plt.title('Гистограмма рассеяния с линией аппроксимации')
plt.legend()
plt.grid(True)
plt.show()
r_squared = r_value_user**2
print(f"Коэффициент детерминации ( $R^2$ ): {r_squared:.2f}")

```

Коэффициент корреляции: 0.9746345310326
Сильная положительная связь

Гистограмма рассеяния





Коэффициент детерминации (R^2): 0.95

```
import pandas as pd
import numpy as np
regions = {
    "Центральный федеральный округ": [
        "Белгородская область", "Брянская область", "Владимирская
        область", "Воронежская область",
        "Ивановская область", "Калужская область", "Костромская
        область", "Курская область",
        "Липецкая область", "Московская область", "Орловская область",
        "Рязанская область",
        "Смоленская область", "Тамбовская область", "Тверская
        область", "Тульская область",
        "Ярославская область", "Город федерального значения Москва"
    ],
    "Южный федеральный округ": [
        "Республика Адыгея", "Республика Калмыкия", "Республика Крым",
        "Краснодарский край",
        "Астраханская область", "Волгоградская область", "Ростовская
```

```

область", "Город федерального значения Севастополь"
],
"Северо-Западный федеральный округ": [
    "Республика Карелия", "Республика Коми", "Архангельская
область", "Вологодская область",
    "Калининградская область", "Ленинградская область",
"Мурманская область", "Новгородская область",
    "Псковская область", "Ненецкий автономный округ", "Город
федерального значения Санкт-Петербург"
],
"Дальневосточный федеральный округ": [
    "Республика Саха (Якутия)", "Камчатский край", "Приморский
край", "Хабаровский край",
    "Амурская область", "Магаданская область", "Сахалинская
область", "Еврейская автономная область",
    "Чукотский автономный округ"
],
"Сибирский федеральный округ": [
    "Республика Алтай", "Республика Бурятия", "Республика Тыва",
"Республика Хакасия",
    "Алтайский край", "Забайкальский край", "Красноярский край",
"Иркутская область",
    "Кемеровская область", "Новосибирская область", "Омская
область", "Томская область"
],
"Уральский федеральный округ": [
    "Курганская область", "Свердловская область", "Тюменская
область", "Челябинская область",
    "Ханты-Мансийский автономный округ – Югра", "Ямало-Ненецкий
автономный округ"
],
"Приволжский федеральный округ": [
    "Республика Башкортостан", "Республика Марий Эл", "Республика
Мордовия", "Республика Татарстан",
    "Удмуртская Республика", "Чувашская Республика", "Кировская
область", "Нижегородская область",
    "Оренбургская область", "Пензенская область", "Ульяновская
область", "Самарская область",
    "Саратовская область", "Пермский край"
],
"Северо-Кавказский федеральный округ": [
    "Республика Дагестан", "Республика Ингушетия", "Кабардино-
Балкарская Республика",
    "Карачаево-Черкесская Республика", "Республика Северная Осетия
– Алания", "Чеченская Республика",
    "Ставропольский край"
]
}
num_observations = 100

```

```

data = []
for district, regions_list in regions.items():
    for region in regions_list:
        for _ in range(num_observations // len(regions)):
            data.append({
                "Регион РФ": region,
                "Федеральный округ": district,
                "Доля людей в возрасте 70+": np.random.uniform(10,
80),
                "Приверженность к 30Ж": np.random.uniform(10, 80)
            })
df = pd.DataFrame(data)
random_sample = df.sample(frac=1/3, random_state=42)
stratified_sample = df.groupby('Федеральный округ',
group_keys=False).apply(lambda x: x.sample(frac=1/3, random_state=42))
random_sample_path = r"C:\Users\Максим\Downloads\
random_sample_federal.xlsx"
stratified_sample_path = r"C:\Users\Максим\Downloads\
stratified_sample_federal.xlsx"
random_sample.to_excel(random_sample_path, index=False)
stratified_sample.to_excel(stratified_sample_path, index=False)
print(f"Выборки успешно сохранены в папку Загрузки:\n{
random_sample_path}\n{stratified_sample_path}")

```

Выборки успешно сохранены в папку Загрузки:
C:\Users\Максим\Downloads\random_sample_federal.xlsx
C:\Users\Максим\Downloads\stratified_sample_federal.xlsx

```

import pandas as pd
import numpy as np
from scipy import stats
random_sample = pd.read_excel(r"C:\Users\Максим\Downloads\
random_sample_federal.xlsx")
stratified_sample = pd.read_excel(r"C:\Users\Максим\Downloads\
stratified_sample_federal.xlsx")
mean_random = random_sample["Доля людей в возрасте 70+"].mean()
mean_stratified = stratified_sample["Доля людей в возрасте
70+"].mean()
df = pd.read_excel(r"C:\Users\Максим\Downloads\
regions_data_updated.xlsx")
mean_population = df["Доля людей в возрасте 70+"].mean()
def confidence_interval(data, confidence=0.95):
    # Рассчитываем стандартную ошибку
    se = stats.sem(data)
    # Рассчитываем границы доверительного интервала
    margin = se * stats.t.ppf((1 + confidence) / 2., len(data)-1)
    return (data.mean() - margin, data.mean() + margin)
conf_interval_90_random = confidence_interval(random_sample["Доля
людей в возрасте 70+"], confidence=0.90)
conf_interval_95_random = confidence_interval(random_sample["Доля

```



```

людей в возрасте 70+"], confidence=0.95)
conf_interval_99_random = confidence_interval(random_sample["Доля
людей в возрасте 70+"], confidence=0.99)
conf_interval_90_stratified =
confidence_interval(stratified_sample["Доля людей в возрасте 70+"],
confidence=0.90)
conf_interval_95_stratified =
confidence_interval(stratified_sample["Доля людей в возрасте 70+"],
confidence=0.95)
conf_interval_99_stratified =
confidence_interval(stratified_sample["Доля людей в возрасте 70+"],
confidence=0.99)
print(f"Среднее генеральной выборки: {mean_population:.2f}")
print(f"Среднее случайной выборки: {mean_random:.2f}")
print(f"Среднее стратифицированной выборки: {mean_stratified:.2f}")
print("\nДоверительные интервалы для случайной выборки:")
print(f"90%: {conf_interval_90_random}")
print(f"95%: {conf_interval_95_random}")
print(f"99%: {conf_interval_99_random}")
print("\nДоверительные интервалы для стратифицированной выборки:")
print(f"90%: {conf_interval_90_stratified}")
print(f"95%: {conf_interval_95_stratified}")
print(f"99%: {conf_interval_99_stratified}")

```

Среднее генеральной выборки: 43.78
 Среднее случайной выборки: 44.91
 Среднее стратифицированной выборки: 43.29

Доверительные интервалы для случайной выборки:
 90%: (43.06457676707644, 46.754853290384055)
 95%: (42.709249400457985, 47.11018065700251)
 99%: (42.01182662703991, 47.807603430420585)

Доверительные интервалы для стратифицированной выборки:
 90%: (41.45056383552035, 45.13002521167909)
 95%: (41.09627783198994, 45.4843112152095)
 99%: (40.40089900565058, 46.17969004154886)