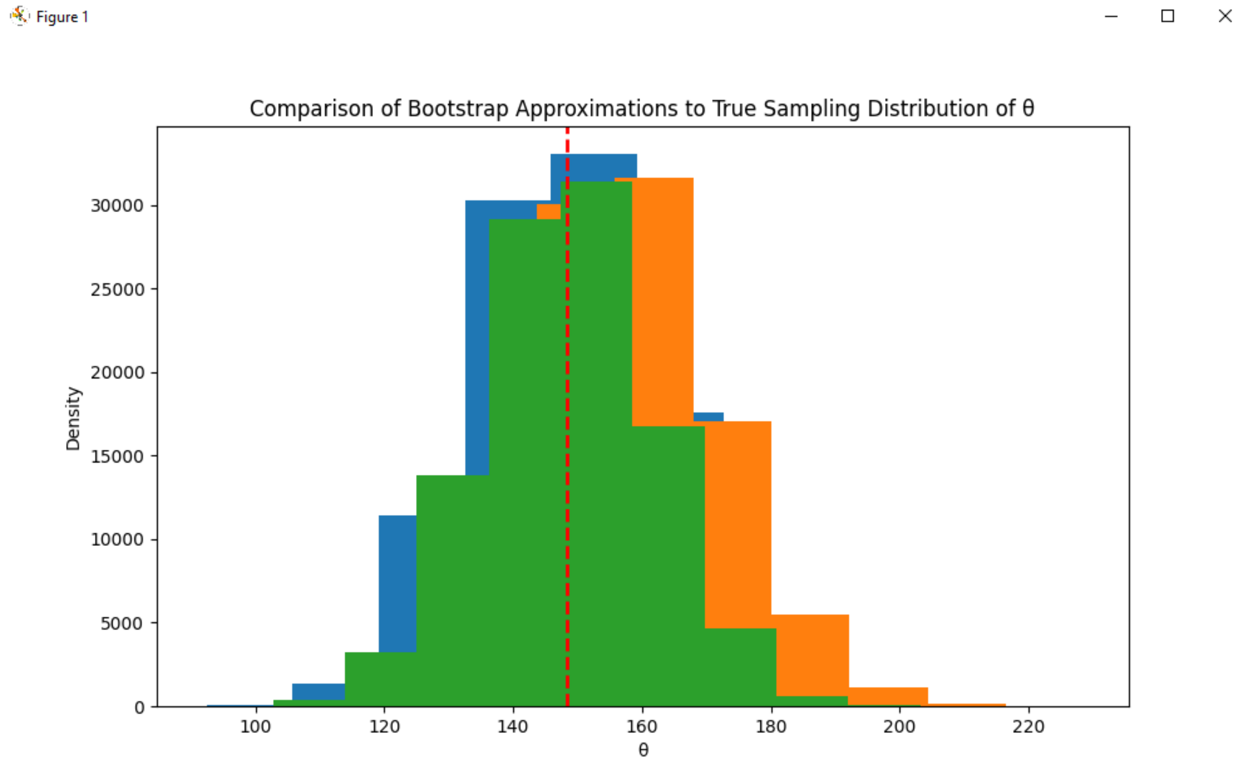
Chapter 9.14 p.146-148: ex. 9

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(1)

n = 100
mu = 5
sigma = 1

X = np.random.normal(mu, sigma, n)

theta = np.exp(mu)
theta_hat = np.exp(np.mean(X))
variance = np.var(X)

#using Delta Method
se_delta = np.abs(np.exp(mu)) * np.sqrt(variance / n)
theta_delta = np.random.normal(theta, se_delta, size=100000)
conf_interval_delta = [theta - 1.96 * se_delta, theta + 1.96 * se_delta]
print(conf_interval_delta)

#using parametric bootstrap
n_param_bootstrap = 100000
theta_param_bootstrap = np.empty(n_param_bootstrap)
for i in range(n_param_bootstrap):
    X_param_bootstrap = np.random.normal(mu, 1, n)
    theta_param_bootstrap[i] = np.exp(np.mean(X_param_bootstrap))

se_param_bootstrap = np.std(theta_param_bootstrap)
conf_interval_param_bootstrap = [theta - 1.96*se_param_bootstrap, theta +
1.96*se_param_bootstrap]
print(conf_interval_param_bootstrap)

#using non-parametric bootstrap
n_nonparam_bootstrap = 100000
theta_nonparam_bootstrap = np.empty(n_nonparam_bootstrap)
for i in range(n_nonparam_bootstrap):
    X_nonparam_bootstrap = np.random.choice(X, n, replace=True)
    theta_nonparam_bootstrap[i] = np.exp(np.mean(X_nonparam_bootstrap))

se_nonparam_bootstrap = np.std(theta_nonparam_bootstrap)
conf_interval_nonparam_bootstrap = [theta - 1.96*se_nonparam_bootstrap, theta +
1.96*se_nonparam_bootstrap]
print(conf_interval_nonparam_bootstrap)

# Comparing answers:
# [122.66486842374842, 174.16144978140477]
# [119.26009247003765, 177.56622573511555]
# [120.84612196202696, 175.98019624312624] answers are quite similar
```

```
#plotting histograms
plt.figure(figsize=(10, 6))
plt.hist(theta_param_bootstrap, label='Parametric Bootstrap')
plt.hist(theta_nonparam_bootstrap, label='Nonparametric Bootstrap')
plt.hist(theta_delta, label='Delta Method')
plt.axvline(x=theta, color='r', linestyle='dashed', linewidth=2, label='True
Value')
plt.xlabel('θ')
plt.ylabel('Density')
plt.title('Comparison of Bootstrap Approximations to True Sampling Distribution
of θ')
plt.show()
```



Homework 12

Chapter 10.11 p 170-173

Ex 12 Let $X_1, \dots X_n \sim \text{Poisson}(\lambda)$.

(a) Let $\lambda_0 > 0$. Find the size α Wald test for $H_0: \lambda = \lambda_0$ VS $H_1: \lambda \neq \lambda_0$

- likelihood function for Poisson distribution is given by:

$$L(\lambda) = e^{-\lambda} \cdot \frac{\lambda^1}{1!} \cdot \dots \dots \cdot e^{-\lambda} \cdot \frac{\lambda^n}{n!}$$

- log-likelihood function is

$$\log(L(\lambda)) = -n\lambda + n \cdot \log(\lambda) - \log(n!)$$

- MLE:

$$\frac{d}{d\lambda} = \frac{n-\lambda}{\lambda} = \frac{n}{\lambda} - 1 = \frac{X_i}{\lambda} - n$$

$$\frac{X_i}{\lambda} - n = 0.$$

$$\hat{\lambda} = \frac{X_i}{n} = \frac{X_1 + \dots + X_n}{n}$$

$$W = \frac{\hat{\theta} - \theta}{se} = \frac{\hat{\theta} - \theta}{\sqrt{Var(\hat{\theta})}}$$

In our problem: $W = \frac{\hat{\lambda} - \lambda_0}{\sqrt{Var(\hat{\lambda})}}$

- Second derivative of log-licelihood function

$$\frac{d^2}{d\lambda^2} = -\frac{n}{\lambda^2}$$

$$Var(\hat{\lambda}) = \frac{1}{n \cdot \lambda_0^2}$$

$$\hat{\lambda}(MLE) = \lambda_0$$

So,

$$W = \frac{\lambda_0 - \lambda_0}{\sqrt{Var(\hat{\lambda})}} = \frac{0}{\sqrt{\frac{1}{n \cdot \lambda_0^2}}} = 0$$

if $|W| > Z_{\frac{\alpha}{2}}$ reject.

if $|W| \leq Z_{\frac{\alpha}{2}}$ do not reject, cause there is not enough evidence to suggest that λ is not equal to λ_0

```python
import numpy as np
import scipy.stats

l_0 = 1
n = 20
a = 0.05
simulations = 10000000

c = scipy.stats.norm.ppf(0.975)

np.random.seed(1)
X = np.random.poisson(lam = l_0, size=[simulations, n])
W = (np.mean(X, axis = 1) - l_0) / (np.sqrt(1/n *l_0**2))
reject = np.sum(np.abs(W) > c)
ans = reject / simulations
print(ans) #0.0557798
```