Zhangsheng Lai
1002554

Statistics: Homework 3

10.5 Given $X_1, \ldots, X_n \sim \text{Uniform}(0, \theta)$ and $Y = \max\{X_1, \ldots, X_n\}$, we have the cdf of Y to be $F_Y(y) = (y/\theta)^n$ for $y \in [0, \theta]$.

(a) When we choose to reject H_0 when $Y > c$, the power function is $\beta(\theta) = 1 - (c/\theta)^n$, $c \in [0, \theta]$.

(b) Given size of the test to be .05, we need to solve,

$$1 - (2c)^n = .05$$

which gives us a solution of $c = 1/2(.95)^{1/n}$

(c) The size, $\alpha = \beta(1/2) = 1 - (2c)^n$, $c \in [0, 1/2]$. Thus, when $n = 20, Y = .48$, the p-value is

$$\inf\{\alpha : X^n \in R_\alpha\} = 1 - (2 \times .48)^{20} = 0.557997566$$

We would conclude that we do not reject H_0 with an approximate probability of 0.56, which does not give a strong evidence to reject H_0

(d) When $n = 20, Y = .52$, using the α formula in (c) gives us $1 - (2 \times .52)^{20} = -1.19112314$. But the given $Y = .52 > 1/2$ which is out of the defined boundaries of the size, i.e. $F_Y(0.52; \theta = 1/2) = 0$. Hence the p-value is 0. This allows us to conclude that H_0 is to be rejected as the p-value always lies in the critical region; a very strong reason to reject H_0.

10.7b Let $H_0 : F_T = F_S$ and $H_1 : F_T \neq F_S$, where the subscripts denote Twain and Snodgrass respectively. The observed value of the test statistic given by the absolute difference of their means, $|\overline{T} - \overline{S}|$ is

$$|0.231875 - 0.2097| = 0.022175$$

```
T = np.array([.225, .262, .217, .240, .230, .229, .235, .217])
S = np.array([.209, .205, .196, .210, .202, .207, .224, .223, .220, .201])
TS = np.concatenate((T, S), axis = 0)
def test_statistic(x):
    return np.absolute(np.mean(x[: 8]) - np.mean(x[8: ]))
def perm_compute(x):
    f = lambda x : np.random.shuffle(x)
    g = lambda x : test_statistic(x)
    f (x)
    return g(x)
def perm_test(x, n = 1000):
    ts_obs = test_statistic(x)
    ts = [perm_compute(x) for m in np.arange(n)]
    ts = np.array(ts)
    return np.mean(ts > ts_obs)
```

which then a p-value fo 0.0004 was obtained. Thus under this p-value, we reject the null hypothesis at both 5% and 2.5% levels.

10.8 (a) The size of this test with rejection region R is

$$\begin{aligned}
\mathbb{P}(T(X^n) > c|\theta = 0) &= \mathbb{P}(\overline{X}_n > c) \\
&= \mathbb{P}\left(Z > \sqrt{n}c\right), \ Z \text{ is the standard normal distribution} \\
&= 1 - \Phi(\sqrt{n}c), \ \Phi \text{ is the cdf of the standard normal}
\end{aligned}$$

where by Central Limit Theorem, $\overline{X}_n \sim N(0, 1/\sqrt{n})$. Thus given size α, the c is $\Phi^{-1}(1-\alpha)/\sqrt{n}$

(b) Under $H_1 : \theta = 1$, the power is $\beta(1) = \mathbb{P}(T(X^n) > c|\theta = 1) = 1 - \Phi\left(\sqrt{n}(c - 1)\right)$.

(c) Thus when $n \to \infty$,

$$c = \frac{\Phi^{-1}(1-\alpha)}{\sqrt{n}} \to 0, \text{ from the right}$$
$$c - 1 \to -1$$
$$\sqrt{n}(c-1) \to -\infty \implies \Phi^{-1}(\sqrt{n}(c-1)) \to 0$$

hence $1 - \Phi\left(\sqrt{n}(c-1)\right) \to 1$.

10.12 (a) We known that the MLE for λ is $\overline{X}_n = n^{-1}\sum_{i=1}^n X_i$. The Fisher information $I_n(\lambda)$ is

$$I_n(\lambda) = nI(\lambda) = -n\mathbb{E}_\lambda\left(\frac{\partial^2 f_X(X;\lambda)}{\partial\lambda^2}\right) = -n\mathbb{E}_\lambda\left(-\frac{X}{\lambda^2}\right) = \frac{n}{\lambda}$$

thus by the property of MLE,

$$\frac{\overline{X}_n - \lambda}{\hat{\mathsf{se}}} \rightsquigarrow N(0,1)$$

We reject the null hypothesis if $\left|\frac{\overline{X}_n - \lambda_0}{\sqrt{\overline{X}_n/n}}\right| > z_{\alpha/2}$ and do not reject otherwise.

(b)

```
import numpy as np
from scipy.stats import norm
def poisson_sample(l, n):
    """
    Generates n Poisson distributed samples with parameter l.
    """
    return np.random.poisson(lam = 1, size = n)
def wald_test(sample, n = 20, alpha = .05, null_lambda = 1):
    """
    Perfoms Wald test and returns p-value.
    """
    xbar = np.mean(sample)
    test_statistic = np.absolute((xbar - null_lambda)/ (null_lambda / n) ** 0.5)
    return 2 * (1 - norm.cdf(test_statistic))
def multwald(l = 1, n = 20, alpha = .05, null_lambda = 1, B = 10000):
    """
    Performs Wald test B times and return proportion of test where null hypothesis is  rejected.
    """
    count = 0
    for i in np.arange(B):
        sample = poisson_sample(l, n)
        if wald_test(sample) < alpha:
            count += 1

    return count/B
multwald()
```

From performing the simulation of Wald 10000 times, the proportion of null rejected is 0.0564 which is very close to the type I error rate of α.

11.3 The posterior density

$$f(\theta|x^n) \propto \mathcal{L}_n(\theta)f(\theta)$$
$$f(\theta|x^n) \propto \prod_{i=1}^{n}\left(\frac{\mathbb{1}_{X_i\le\theta}}{\theta}\right)(1/\theta)$$

Evaluating the partition function:

$$\int_0^\infty \left(\prod_{i=1}^{n}\mathbb{1}_{X_i\le\theta}\right)(1/\theta)^{n+1}\,d\theta = \int_{X_{max}}^\infty \left(\prod_{i=1}^{n}\mathbb{1}_{X_i\le\theta}\right)(1/\theta)^{n+1}\,d\theta + \int_0^{X_{max}} \left(\prod_{i=1}^{n}\mathbb{1}_{X_i\le\theta}\right)(1/\theta)^{n+1}\,d\theta$$
$$= \int_{X_{max}}^\infty \left(\prod_{i=1}^{n}\mathbb{1}_{X_i\le\theta}\right)(1/\theta)^{n+1}\,d\theta$$

where $X_{max} = \max\{X_1,\dots,X_n\}$ and the second term goes to zero due to the indictor function $\mathbb{1}_{X_i\le\theta}$ which is zero for values of theta smaller than any of the X_i's. Hence

$$f(\theta|x^n) = \left(\prod_{i=1}^{n}\mathbb{1}_{X_i\le\theta}\right)(1/\theta)^{n+1}$$

11.4 (a) The likelihood function where $\theta = (p_1, p_2)$, $X_i \sim \text{Bernoulli}(p_1)$ and $Y_i \sim \text{Bernoulli}(p_2)$ is

$$\mathcal{L}(\theta) = p_1^{\sum_{i=1}^n X_i}(1-p_1)^{n-\sum_{i=1}^n X_i}p_2^{\sum_{i=1}^n Y_i}(1-p_2)^{n-\sum_{i=1}^n Y_i}$$

with log-likelihood, $\ell(\theta) = \sum_{i=1}^{n} X_i \log p_1 + \left(n - \sum_{i=1}^{n} X_i\right)\log(1-p_1) + \sum_{i=1}^{n} Y_i \log p_2 + \left(n - \sum_{i=1}^{n} Y_i\right)\log(1-p_2)$

differentiating with respect to p_1 and p_2 to get the MLE,

$$\frac{\partial \ell}{\partial p_1} = \frac{\sum_{i=1}^n X_i}{p_1} - \frac{(n-\sum_{i=1}^n X_i)}{1-p_1}$$
$$\frac{\partial \ell}{\partial p_2} = \frac{\sum_{i=1}^n Y_i}{p_2} - \frac{(n-\sum_{i=1}^n Y_i)}{1-p_2}$$

we get $\hat{p}_1 = \sum X_i/n$ and $\hat{p}_2 = \sum Y_i/n$ when we solve for the above to be equal to 0. Using the multiparameter delta method, with $\tau = g(\theta) = p_2 - p_1$, we have $\hat{\tau} = \hat{p}_2 - \hat{p}_1$. We then require $\bigtriangledown \hat{g}$ and $J_n(\hat{\theta})$ to evaluate $\hat{\mathsf{se}}(\hat{\tau})$. It is easy to see that $\bigtriangledown \hat{g} = \begin{pmatrix}-1 & 1\end{pmatrix}^T$ and

$$I_n(\theta) = \begin{pmatrix} \mathbb{E}_\theta\left(\frac{\sum X_i}{p_1^2} + \frac{(n-\sum X_i)}{(1-p_1)^2}\right) & 0 \\ 0 & \mathbb{E}_\theta\left(\frac{\sum Y_i}{p_2^2} + \frac{(n-\sum Y_i)}{(1-p_2)^2}\right)\end{pmatrix}$$
$$= \begin{pmatrix} \frac{n}{p_1} + \frac{n}{(1-p_1)} & 0 \\ 0 & \frac{n}{p_2} + \frac{n}{1-p_2}\end{pmatrix}$$
$$J_n(\theta) = \begin{pmatrix} \frac{p_1(1-p_1)}{n} & 0 \\ 0 & \frac{p_2(1-p_2)}{n}\end{pmatrix}$$

and thus

$$\hat{\mathsf{se}}(\theta)^2 = (\bigtriangledown \hat{g})^T J_n(\hat{\theta})(\bigtriangledown \hat{g}) = \begin{pmatrix}-1 & 1\end{pmatrix}\begin{pmatrix} \frac{p_1(1-p_1)}{n} & 0 \\ 0 & \frac{p_2(1-p_2)}{n}\end{pmatrix}\begin{pmatrix}-1 \\ 1\end{pmatrix} = \frac{p_1(1-p_1)}{n} + \frac{p_2(1-p_2)}{n}$$

thus $\hat{\mathsf{se}}(\hat{\theta}) = \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n} + \frac{\hat{p}_2(1-\hat{p}_2)}{n}} = 0.0894427191$ for $n = 50$ and the $\hat{p}_1$ and $\hat{p}_2$ obtained earlier. A 90% confidence interval is 0.2 ± 0.147580487

(b) Using parametric bootstrap, we have MLE of p_1 and p_2 to be $\hat{p}_1 = 3/5$ and $\hat{p}_2 = 4/5$ respectively and thus MLE of τ to be $1/5$. The parametric bootstrap requires sampling from $X_P \sim \text{Bernoulli}(3/5)$ and $X_T \sim \text{Bernoulli}(4/5)$, where the subscripts denote placebo and treatment respectively. Using 1000 simulations, we get a standard error of 0.0895209919516.

```
import numpy as np

mle_p1 = 3/5
mle_p2 = 4/5
mle_tau = mle_p2 - mle_p1
n = 100000

se2_boot = 0

for i in np.arange(n):
    p1_mean = np.mean(np.random.binomial(1, mle_p1, size = 50))
    p2_mean = np.mean(np.random.binomial(1, mle_p2, size = 50))
    se2_boot += ((p2_mean - p1_mean) - mle_tau) ** 2
se_boot = np.sqrt(se2_boot/n)
print (se_boot)
```

A 90% confidence interval will then be 0.2 ± 0.148

(c) With the prior $f(p_1, p_2) = 1$,

$$f(p_1, p_2|x^n, y^n) \propto \mathcal{L}(p_1, p_2) = p_1^{\sum_{i=1}^n X_i}(1-p_1)^{n-\sum_{i=1}^n X_i} p_2^{\sum_{i=1}^n Y_i}(1-p_2)^{n-\sum_{i=1}^n Y_i}$$

and since

$$\begin{aligned} f(p_1, p_2|x^n, y^n) &= f(p_1|x^n)f(p_2|y^n) \\ \text{and } f(p_1|x^n) &\propto p_1^{\sum_{i=1}^n X_i}(1-p_1)^{n-\sum_{i=1}^n X_i} \\ f(p_2|y^n) &\propto p_2^{\sum_{i=1}^n Y_i}(1-p_2)^{n-\sum_{i=1}^n Y_i} \end{aligned}$$

the simulation is by drawing samples from $p_1|x^n \sim \text{Beta}(31, 21)$ and $p_2|y^n \sim \text{Beta}(41, 11)$ which gives a posterior mean estimate of τ to be 0.19313 with the code below:

```
n = 1000

p1 = np.random.beta(31, 21, size = n)
p2 = np.random.beta(41, 11, size = n)

np.mean(p2 - p1)
```

we then plot a histogram with the code below

```
n = 1000

p1 = np.random.beta(31, 21, size = n)
p2 = np.random.beta(41, 11, size = n)

tau = p2 - p1

plt.hist(tau, cumulative = True, normed = True, bins = 20)
plt.axhline(y = 0.05, color = 'r', linewidth = 0.5)
plt.axhline(y = 0.95, color = 'r', linewidth = 0.5)
```

and see that a 90% confidence interval by simulation is approximately $(0.023248, 0.36918)$.

(d) The MLE of ψ is $\log\left(\frac{3/5}{2/5} \div \frac{4/5}{1/5}\right) = \log 3/8$. Using the multiparameter delta method, $\triangledown g = \begin{pmatrix} \frac{1}{p_1(1-p_1)} & -\frac{1}{p_2(1-p_2)} \end{pmatrix}$ and with $J_n(\theta)$ from earlier

$$\begin{aligned} \hat{\text{se}}(\theta)^2 = (\triangledown\hat{g})^T J_n(\hat{\theta})(\triangledown\hat{g}) &= \begin{pmatrix} \frac{1}{p_1(1-p_1)} & -\frac{1}{p_2(1-p_2)} \end{pmatrix} \begin{pmatrix} \frac{p_1(1-p_1)}{n} & 0 \\ 0 & \frac{p_2(1-p_2)}{n} \end{pmatrix} \begin{pmatrix} \frac{1}{p_1(1-p_1)} \\ -\frac{1}{p_2(1-p_2)} \end{pmatrix} \\ &= \frac{1}{np_1(1-p_1)} + \frac{1}{np_2(1-p_2)} \end{aligned}$$

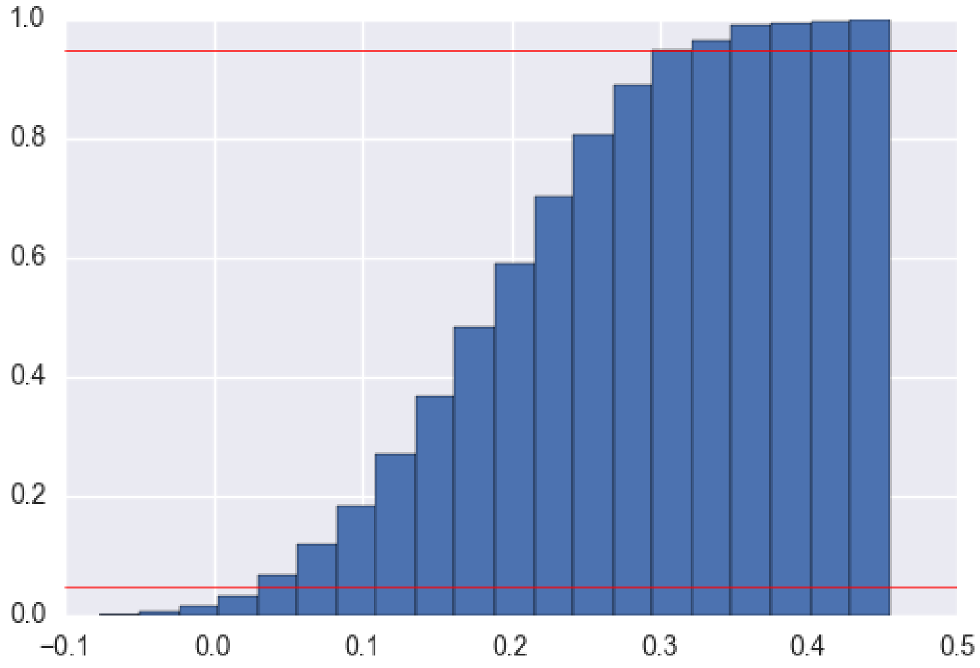

Figure 1: Cumulative distribution to obtain posterior confidence interval

thus $\hat{\text{se}}(\hat{\theta}) = \sqrt{\frac{1}{n\hat{p}_1(1-\hat{p}_1)} + \frac{1}{n\hat{p}_2(1-\hat{p}_2)}} = 0.456435465$. A 90% confidence interval would be $\log 3/8 \pm 0.753118517$

(e) The posterior estimate of ψ is 0.94397 and the posterior 90% interval for ψ is $(-1.68, -0.416)$.

```
n = 1000

p1 = np.random.beta(31, 21, size = n)
p2 = np.random.beta(41, 11, size = n)

psi_distribution = np.log((p1 / (1 - p1)) / (p2 / (1 - p2)))

psi_estimate = np.mean(psi_distribution)

print (psi_estimate)

plt.axhline(y = 0.05, color = 'r', linewidth = 0.5)
plt.axhline(y = 0.95, color = 'r', linewidth = 0.5)

plt.hist(psi_distribution, cumulative = True, normed = True, bins = 20)
```

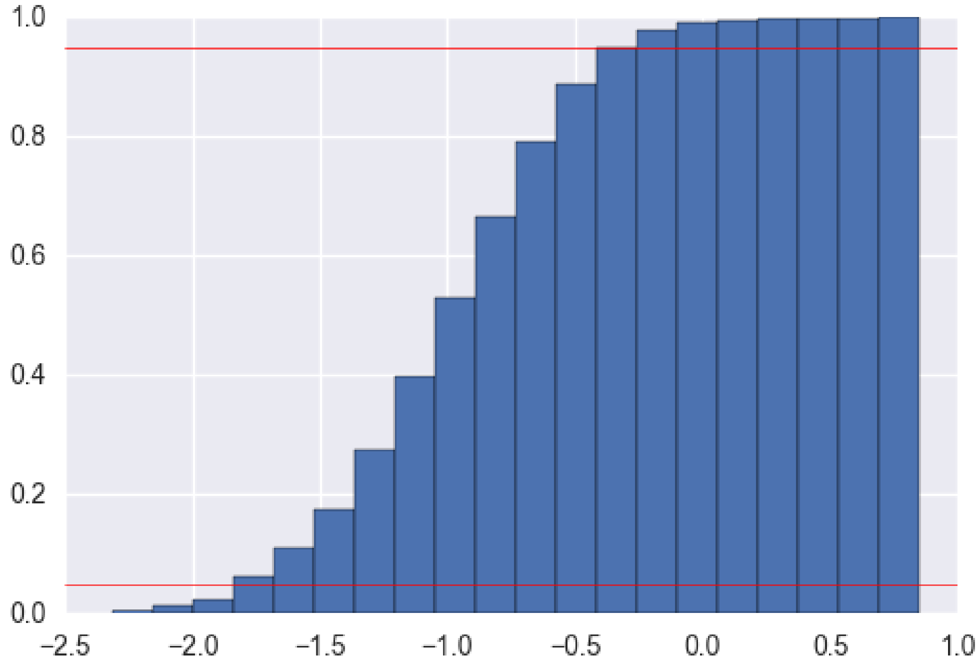

Figure 2: Cumulative distribution to obtain ψ posterior confidence interval