hw5.

1. → GLM = unknown concentration ρ_0 of infections microbe.

→ progressively dilute original solution

→ after each dilution, pour (small volume) V into plate.

→ plate ↗ zero microbe → sterile
↘ any microbe → infected.

→ observing = whether infection. & ρ_0

→ ρ_t = concentration at dilution t

→ $\rho_t = \frac{\rho_0}{2^t}$

→ Y_t = observed sterile plate $\in \{0, 1\}$

→ relate ρ_0 & t & Y_t

a. → $\mu(t) := \mathbb{E}[Y_t]$ = chance of infected plate

→ plausible output dist for Y_t, using $\mu(t)$.

⇒ since Y_t can either be 0 or 1

⇒ Y_t is a discrete binary dist

⇒ possible dist = bernoulli dist

⇒ thus, $Y_t \sim \text{Bernoulli}(\mu(t))$

b. → at dilusion t, pour volum v.

$\rightarrow \mathbb{E}[\#\text{ microbe}] = \rho_t * v.$

$\rightarrow$ actual # microbes at $t \sim \text{Poisson}(\rho_t * v)$.

$\rightarrow$ expression $= \mu(t) := \mathbb{E}[Y_t]$

$\Rightarrow \mu(t) = \mathbb{P}(\text{plate is infected at dilution } t)$

$= 1 - \mathbb{P}(\text{there are 0 microbes on plate at dilution } t).$

$= 1 - \mathbb{P}(0 \sim \text{Poisson}(\rho_t * v))$

$= 1 - e^{-v\rho_t} \dfrac{(v\rho_t)^0}{0!}$

$= 1 - e^{-v\rho_t}$

$\Rightarrow$ since $\rho_t = \dfrac{\rho_0}{2^t}$

$\Rightarrow \mu(t) = 1 - e^{-v*\left(\frac{\rho_0}{2^t}\right)}$

$\Rightarrow$ thus, $\mu(t) = 1 - e^{-v*\left(\frac{\rho_0}{2^t}\right)}$.

c. $\rightarrow$ link func $g(\mu(t)) = \beta_0 + \beta_1 t$

$\rightarrow$ constant β_0 & β_1.

$\Rightarrow \mu(t) = 1 - e^{-v*\left(\frac{\rho_0}{2^t}\right)}$

$\Rightarrow$ since the only part that contain t is the second part

$\Rightarrow$ let $g_1(x) = 1 - x$

$\Rightarrow g_1(\mu(t)) = 1 - (1 - e^{-v*\left(\frac{\rho_0}{2^t}\right)}))$

$$= e^{-v*(\frac{\rho_0}{2^t})}$$

$\Rightarrow$ if we take $\ln$ for this $g(x) = g_2(x)$

$$\Rightarrow \ln[g(\mu(t))] = \log[e^{-v*(\frac{\rho_0}{2^t})}]$$

$$= -v*(\frac{\rho_0}{2^t})$$

$\Rightarrow$ we then take $\log_2$ for $g_2(x)$

$$\Rightarrow \log_2(g_2(x)) = \log_2[-v*\rho_0 * 2^{-t}]$$

$$= \log_2[-v*\rho_0] - \log_2[2^t]$$

$$= \log_2[-v*\rho_0] - t$$

$\Rightarrow$ then $\beta_0 = \log_2[-v*\rho_0]$ and $\beta_1 = -1$

$$\Rightarrow \log_2[\ln(1-\mu(t))] = \log_2[-v*\rho_0] - t$$

$\Rightarrow$ thus, $g(x) = \log_2[\ln(1-x)]$

so $g(\mu(t)) = \log_2[\ln(1-\mu(t))]$

and $\beta_0 = \log_2[-v*\rho_0] = \text{constant}$

and $\beta_1 = -1 = \text{constant}$.

d. $\rightarrow$ estimation for ρ_0

$\Rightarrow Y_t = \text{Bernoulli}(\mu(t))$

$\Rightarrow \mu(t) = 1 - e^{-v*(\frac{\rho_0}{2^t})}$.

$\Rightarrow g(\mu(t)) = \beta_0 + \beta_1 t$

$\Rightarrow \beta_0 = \log_2[-v * \rho_0]$ & $\beta_1 = -1$

$\Rightarrow 2^{\beta_0} = -v * \rho_0$

$\Rightarrow \rho_0 = 2^{\beta_0}/(-v)$

$\Rightarrow$ thus, if we know β_0, estimate for

$$\rho_0 = 2^{\beta_0}/(-v)$$

2. $\rightarrow$ finite markov decision process

$\rightarrow$ each state = 4 action N, S, E, W.

$\rightarrow$ each action reward == 0.

$\rightarrow$ off-grid : $R = -1$ & $A \rightarrow A'$: $R = +10$

$\rightarrow B \rightarrow B' = +5$

$\rightarrow \gamma = 0.9$ = discounted rewards

a. $\rightarrow$ center $V(s) = 0.7$ & bellman equation.

$\Rightarrow V(s) = \max_a Q(s,a)$

$\Rightarrow Q(s,a) = \sum_{s'} \mathbb{P}(s'|s,a) * (R(s',a,s) + \gamma V(s'))$
$= \sum_{s'} \mathbb{P}(s'|s,a) * (R(s',a,s) + \gamma \max_{a'} Q(s',a'))$.

s'

$\Rightarrow$ since each state have 4 possbile actions equally

$\Rightarrow R(s', a, s) = 0$ for all movement around the center

$\Rightarrow V(\text{center}) = \max_a \sum_{s'} 1 * (0 + 0.9\, V(s'))$

$\Rightarrow P(s'|s, a) =$ deterministically $1/4$.

$\Rightarrow$ deviate the optimal policy, in this we need the value

func that have equal prob to go to all adjcent state

$\Rightarrow V(\text{center}) = \sum_a P(a|s) \sum_{s'} P(s'|s,a) * (R(s',a,s) + \gamma V(s'))$.

$$\Rightarrow V(\text{center}) = \tfrac{1}{4} * (0 + 0.9(2.3)) + \tfrac{1}{4} * (0 + 0.9(0.7))$$

$$+ \tfrac{1}{4}(0 + 0.9(0.41)) + \tfrac{1}{4} * (0 + 0.9(-0.41))$$

$$= 0.675$$

≈ 0.7 (rounding up to 1 decimal point).

$\Rightarrow$ thus, the center state does have a value $+0.7$ under this policy.

b. $\rightarrow$ adding constant c to all rewards

$\rightarrow$ result $\uparrow$ in V_c to all states

$\rightarrow$ prove diff in state value = same

$\rightarrow$ V_c in terms of c and γ

$\Rightarrow V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_t, a_t) \mid s_0 = s\right]$

$\Rightarrow R(S_t, a_t) =$ each reward at time t.

$\Rightarrow$ let $R_i = R(S_i, a_i)$ & $i \in T$.

$\Rightarrow V^{\pi}(S) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{k+1} \mid S_0 = s\right]$

$\Rightarrow$ by adding a constant C to all reward.

$\Rightarrow$ new $V_{\pi}'(S) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k (R_{k+1} + c) \mid S_0 = s\right]$.

$= \mathbb{E}\left[\sum_{k=0}^{\infty} (\gamma^k R_{k+1} + \gamma^k c) \mid S_0 = s\right]$

$= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{k+1} + \sum_{k=0}^{\infty} c\gamma^k \mid S_0 = s\right]$

$= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{k+1} + c\sum_{k=0}^{\infty} \gamma^k \mid S_0 = s\right]$

$= \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{k+1} \mid S_0 = s\right] + \mathbb{E}\left[c\sum_{k=0}^{\infty} \gamma^k \mid S_0 = s\right]$

$\Rightarrow$ new $V_{\pi}'(S) = V_{old}^{\pi}(S) + c * \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k\right]$

$\Rightarrow$ since $\mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k\right]$ is independent of s

$\Rightarrow$ all state's new $V_{\pi}'(S)$ is increase by the same constant

$\Rightarrow$ let $V_c = V_{\pi}'(S) - V_{old}^{\pi}(S)$

$= c * \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k\right]$ & k = all positive int

$\Rightarrow \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k\right] = \sum_{k=0}^{\infty} \gamma^k =$ geometric series & $\gamma < 1$

$\Rightarrow V_c = c *$ geometric series that converge

$= \left[\lim_{K \to \infty} \sum_{k=0}^{K} \gamma^k\right] * c$

$$\approx \left[\frac{1}{1-\gamma}\right] * c$$

$$= \frac{c}{1-\gamma}$$

$\Rightarrow$ thus, adding a constant c does not affect the relative values of state under that policy, and

$$V_c = \frac{c}{1-\gamma}$$

c.

$\Rightarrow$ the expected return $V^{\pi}(s) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{k+1} \mid S_0 = s\right]$

$\Rightarrow$ we can tell from the graph that A' have a negative value while B' have a positive value

$\Rightarrow$ this means that to get to A' state, the penalty for running into edge is larger than possibly reach state A then to A'.

$\Rightarrow$ vice, versa, since state B' have a positive value, the penalty for running into edge is smaller than possibly reach state B then to B'.

$\Rightarrow$ we know that once we get to state A, we have to get to A'. bc the A' have a negative value, we can tell that it's more likely to go into the penalty state (running to the edge of the grid), thus, the expected return A smaller than the immediate reward.

$\Rightarrow$ we can analyze the same thing for B, once we get to state B we have to get to B', bc the B' have a negative value, we can tell that it's less likely to go into the penalty state thus, the expected return B larger than the immediate reward.

3. $\rightarrow$ robot A: starting position K_1 & $K \in \{1, 4\}$.

$\rightarrow$ from K_1 to K_3 = $\nearrow$ shoot (S), $\rightarrow$ dribble (D) & P_4 = shoot.

$\rightarrow$ if S $\nearrow$ score (G), $\rightarrow$ miss (M). & D $\nearrow$ advce P_{t+1}, $\rightarrow$ miss (M).

$\rightarrow$ terminate states = G & M.

$\rightarrow$ transitional model (y) = $\mathbb{P}$(dribble & advance).

$\rightarrow$ discount $\gamma = 1$

$\rightarrow \mathbb{P}(G|k,S) = \frac{k}{6}$ & $\mathbb{P}(M|k,S) = 1 - \frac{k}{6}$

$\rightarrow \mathbb{P}(k+1|k,D) = y$ & $\mathbb{P}(M|k,D) = y$.

$\rightarrow \mathbb{R}(k, S, G) = 1$.

a. $\rightarrow$ always shoot $V^{\pi}(1)$.

$\Rightarrow V^{\pi}(S) = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k \mathbb{R}(S_t, a_t) | S_0 = S]$ & V(terminal state) = 0

$\Rightarrow V^{\pi}(1) = \mathbb{E}[\sum_{k=1}^{4} \gamma^k \mathbb{R}(S_t, S) | S_0 = 1] = \sum_{p'} \mathbb{P}(p'|1) * (\mathbb{R}(1,S,s') + \gamma V(p'))$

$= \mathbb{E}[R(1,S)]$

$= \mathbb{P}(G|1,S) * 1 + \mathbb{P}(M|1,S) * 0$

$= 1/6 * 1 + 5/6 * 0 = 1/6.$

$\Rightarrow V^{\pi}(1) = 1/6$

$\Rightarrow$ thus, $V^{\pi}(1)$ for the policy π that shoot is $1/6$.

b. $\rightarrow Q^*(s,a)$ = value of a q-state (s,a).

$\rightarrow$ expected utility = starting with action a at state s & optimally acting after.

$\rightarrow Q^*(3,D)$ & y.

$$\Rightarrow Q^*(s,D) = \sum_{s'} \mathbb{P}(s'|s,a) * (R(s',a,s) + \gamma V(s'))$$

$$= \sum_{s'} \mathbb{P}(s'|s,a) * (R(s',a,s) + \gamma \max_{a'} Q(s',a')).$$

$$\Rightarrow Q^*(3,D) = \sum_{p'} \mathbb{P}(p'|3,D) * (R(p',D,3) + \gamma V(p'))$$

$\Rightarrow p'$ can either be $\nearrow M$, $\searrow$ advance to $K=4$

$\Rightarrow$ ① $p' =$ miss

$\Rightarrow$ value $= \mathbb{P}(M|3,D) * (R(M,D,3) + \gamma V(M))$

$= (1-y) * (0+0) = 0$

$\Rightarrow$ ② $p' =$ advance to $K=4$

$\Rightarrow$ value $= \mathbb{P}(\text{advance}|3,D) * (R(K=4,D,3) + \gamma V(K=4))$

$= y * [0 + V(k=4)]$

$\Rightarrow V(k=4) = \max_a Q(4,a)$

$= \max_a \sum_{p'} P(p'|4,a) * (R(p',a,4) + \gamma V(p'))$

$\Rightarrow$ at $K=4$, action $= S$ & $p' = \begin{matrix} \nearrow G \\ \rightarrow M \end{matrix}$

$\Rightarrow V(k=4) = \sum_{p'} P(p'|4,S) * [R(p',S,4) + \gamma V(p')]$

$= P(G|4,S) * [R(G,S,4) + V(G)] + P(M|4,S) * [R(M,S,4) + V(M)]$

$= 4/6 * (1+0) + (1 - 4/6) * (0+0)$

$= 4/6$

$\Rightarrow Q^*(3,D) = [\text{value} = \text{miss}] + [\text{value} = \text{pass}]$

$= [\text{value} = \text{miss}] + [\text{value} = V(k=4)]$

$= [\text{value} = \text{miss}] + y * [V(k=4)]$

$= 0 + 4/6\, y = 2/3\, y$

$\Rightarrow$ thus, $Q^*(3,D) = \frac{2}{3} y$.

c. $\rightarrow V_t^*(s)$ = value of a state at iteration t.

$\rightarrow$ expected utility starting s & acting optimally.

$\rightarrow y = 3/4$ & first two $(t=1,2)$ value iteration.

$\rightarrow t=0$ = value 0 in everystate $V_0^*(1) = \ldots\ldots = V_0^*(4) = 0$.

$\Rightarrow V_{t+1}^{*}(s) = \max_{a \in A} \sum_{s'} \mathbb{P}(s'|s,a)(\mathbb{R}(s,a,s') + V_t^{*}(s'))$.

$\Rightarrow V_0^{*}(1) = V_0^{*}(2) = V_0^{*}(3) = V_0^{*}(4) = G = M = 0$.

$\Rightarrow V_{0+1}^{*}(1) = \max_{a \in A} \sum_{p'} \mathbb{P}(p'|1,a)(\mathbb{R}(1,a,p') + 0))$.

$\Rightarrow$ ① $a = S$ & $p' = \begin{cases} \nearrow M \\ \searrow G \end{cases}$

$\Rightarrow$ value $= 1/6$

$\Rightarrow$ ② $a = D$ & $p' = \begin{cases} \nearrow M \\ \searrow k = 2 \end{cases}$

$\Rightarrow$ value $= \frac{1}{4} * 0 + \frac{3}{4} * 0 = 0$

value[S] > value[D]

$\Rightarrow V_1^{*}(1) = 1/6$

$\Rightarrow$ since all $V_0^{*}(K) = 0$, all the rest $V_1^{*}(K)$ will choos shoot

$\Rightarrow V_1^{*}(K) = \mathbb{P}(G|K,S)$

$\Rightarrow V_1^{*}(1) = 1/6$ & $V_1^{*}(2) = 2/6$ & $V_1^{*}(3) = 3/6$ & $V_1^{*} = 4/6$

$\Rightarrow V_{1+1}^{*}(1) = \max_{a \in A} \sum_{p'} \mathbb{P}(p'|1,a)(\mathbb{R}(1,a,p') - V_1^{*}(p'))$.

$\Rightarrow$ ① $a = S$ & $p' = \begin{cases} \nearrow M \\ \searrow G \end{cases}$

$\Rightarrow$ value $= \frac{5}{6} * (0+0) + \frac{1}{6} * (1+0) = 1/6$

$\Rightarrow$ ② $a = D$ & $p' = \begin{cases} \nearrow M \\ \searrow k = 2 \end{cases}$

$\Rightarrow$ value $= \frac{1}{4}(0-0) + \frac{3}{4}(0 + \frac{2}{6}) = 1/4$

value[S] < value[D]

$\Rightarrow V_2^{*}(1) = 1/4$ & action $= D$

$\Rightarrow V_{t+1}^{*}(2) = \max_{a \in A} \sum_{p'} \mathbb{P}(p' | 2, a)\left(\mathbb{R}(2, a, p') - V_1^{*}(p')\right).$

$\Rightarrow$ ① $a = S$ & $p' = \begin{matrix} \nearrow M \\ \to G \end{matrix}$

$\Rightarrow$ value $= \frac{4}{6} * (0+0) + \frac{2}{6} * (1+0) = 1/3$

value[S] < value[D]

$\Rightarrow$ ② $a = D$ & $p' = \begin{matrix} \nearrow M \\ \to k=3 \end{matrix}$

$\Rightarrow$ value $= \frac{1}{4}(0-0) + \frac{3}{4}(0 + \frac{3}{6}) = 3/8$

$\Rightarrow V_2^{*}(2) = 3/8$ & action $= D$

$\Rightarrow V_{t+1}^{*}(3) = \max_{a \in A} \sum_{p'} \mathbb{P}(p' | 3, a)\left(\mathbb{R}(3, a, p') - V_1^{*}(p')\right).$

$\Rightarrow$ ① $a = S$ & $p' = \begin{matrix} \nearrow M \\ \to G \end{matrix}$

$\Rightarrow$ value $= \frac{3}{6} * (0+0) + \frac{3}{6} * (1+0) = 1/2$

value[S] = value[D]

$\Rightarrow$ ② $a = D$ & $p' = \begin{matrix} \nearrow M \\ \to k=4 \end{matrix}$

$\Rightarrow$ value $= \frac{1}{4}(0-0) + \frac{3}{4}(0 + \frac{4}{6}) = 1/2$

$\Rightarrow V_2^{*}(3) = 1/2$ & action $= D$ or S.

$\Rightarrow V_{t+1}^{*}(4) = \max_{a \in A} \sum_{p'} \mathbb{P}(p' | 4, a)\left(\mathbb{R}(4, a, p') - V_1^{*}(p')\right).$

$\Rightarrow$ ① $a = S$ & $p' = \begin{matrix} \nearrow M \\ \to G \end{matrix}$

$\Rightarrow$ value $= \frac{2}{6} * (0+0) + \frac{4}{6} * (1+0) = 2/3$

$\Rightarrow V_2^{*}(4) = 2/3$. & action $= S$

$\Rightarrow$ thus, value iteration for $t = 1, 2$ is

k/t	k=1	k=2	k=3	k=4
t=0	$V_0^*(1)=0$	$V_0^*(2)=0$	$V_0^*(3)=0$	$V_0^*(4)=0$
t=1	$V_1^*(1)=1/6$	$V_1^*(2)=2/6$	$V_1^*(3)=3/6$	$V_1^*(4)=4/6$
t=2	$V_2^*(1)=1/4$	$V_2^*(2)=3/8$	$V_2^*(3)=1/2$	$V_2^*(4)=2/3$

d. $\rightarrow$ range y & $Q^*(3,S) \geq Q^*(3,D)$.

$\Rightarrow Q^*(3,D) = \frac{2}{3}y.$ $\leftarrow$ part b.

$\Rightarrow Q^*(3,S) = \sum_{p'} P(p'|3,S) * (R(p',S,3) + \gamma V(p'))$

$\Rightarrow p'$ can either be $\nearrow M$, $\searrow G$

$\Rightarrow Q^*(3,S) = 3/6 * (0+0) + 3/6 * (1+0) = 1/2$

$\Rightarrow Q^*(3,D) = 2/3\, y$ & $Q^*(3,S) = 1/2$

$\Rightarrow$ for $Q^*(3,S) \geq Q^*(3,D)$

$\Rightarrow 1/2 \geq 2/3\, y$

$\Rightarrow 2/3\, y \leq 1/2$

$\Rightarrow y \leq 3/4$ & $y \geq 0$

$\Rightarrow$ thus, for $Q^*(3,S) \geq Q^*(3,D)$, $0 \leq y \leq 3/4$.

4. $\rightarrow$ FDA: new drug = bioequivalent to current drug.

→ bioequivalent = effect not substiantially diff.

→ drug effect = # hormone in blood after drug.

→ O = old drug, N = new drug & P = placebo.

→ bioequivalent $|\theta| \leq 0.2$ & $\theta = \dfrac{\mathbb{E}[N-O]}{\mathbb{E}[O-P]}$.

→ θ from dataset & boostrap confidence interval.

a. → data for #8 subjects.

→ estimate $\hat{\theta}$ of θ.

$$\Rightarrow \theta = \frac{\mathbb{E}[N-O]}{\mathbb{E}[O-P]}.$$

$$= \frac{\mathbb{E}[N] - \mathbb{E}[O]}{\mathbb{E}[O] - \mathbb{E}[P]}$$

$$\Rightarrow \theta = \mathbb{E}[\hat{\theta}]$$

b. → (i) func bootstrap_bioequivalence (N, O, P, B)

→ N, O, P = n length array

→ B = # bootstrap replica

→ return = length B array with $\hat{\theta}$ replica

→ (ii) bootstrap_bioequivalence (N, O, P, 1000)

→ plot hist & x, y axis label

(iii) 95% confidence interval for θ

→ np.percentile.

c. → 95% interval within FDA requirement.

Untitled

April 20, 2020

[33]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import scipy.stats
import numpy.random as rnd
%matplotlib inline
```

[34]:
```
dataframe = pd.read_csv("bioequivalence.csv")
dataframe
```

[34]:

	Patient	New Drug	Old Drug	Placebo
0	0	16449	17649	9243
1	1	14614	12013	9671
2	2	17274	19979	11792
3	3	23798	21816	13357
4	4	12560	13850	9055
5	5	10157	9806	6290
6	6	16570	17208	12412
7	7	26325	29044	18806

[35]:
```
#np.asarray(dataframe["Old Drug"])
O_expect = np.mean(np.asarray(dataframe["Old Drug"]))
N_expect = np.mean(np.asarray(dataframe["New Drug"]))
P_expect = np.mean(np.asarray(dataframe["Placebo"]))
theta_hat = (N_expect - O_expect) / (O_expect - P_expect)
theta_hat
```

[35]:
```
-0.07130609590256017
```

theta_hat is -0.07130609590256017

[36]:
```
O_data = np.asarray(dataframe["Old Drug"])
N_data = np.asarray(dataframe["New Drug"])
P_data = np.asarray(dataframe["Placebo"])
```

[37]:
```
def bootstrap_bioequivalence(N,O,P,B):
    indices = np.arange(0,8)
    pop_para = []
    for i in np.arange(0,B):
        boostrap_indice = rnd.choice(indices,B)
```

```
        boostrap_O = np.mean([O[i] for i in boostrap_indice])
        boostrap_N = np.mean([N[i] for i in boostrap_indice])
        boostrap_P = np.mean([P[i] for i in boostrap_indice])
        boostrap_theta = (boostrap_N - boostrap_O) / (boostrap_O - boostrap_P)
        pop_para.append(boostrap_theta)
    return pop_para
```

[ ]:
```
repeat_10000 = bootstrap_bioequivalence(N_data,O_data,P_data,10000)
```

[ ]:
```
plt.hist(repeat_10000)
plt.xlabel("theta_hat")
plt.ylabel("frequency (amount)")
plt.title("boostrap theta_hat distribution")
```

[ ]:
```
lower_bound = np.percentile(repeat_10000,2.5)
upper_bound = np.percentile(repeat_10000,97.5)
print("the 95% confidence interval for theta is: ",lower_bound, "< theta <",␣
 ↪upper_bound)
print("the 95% confidence interval for theta is: [",lower_bound,",",␣
 ↪upper_bound,"]")
```

Base on part B, we can conclude that the new drug and old drug are bioequivalent, at the 95% confidence interval level. The 95% confidence interval level fall within the FDA requirement for bioequivalent. The bioequivalence is define as |theta|<= 0.2, which mean -0.2<=theta<=0.2. And from part B, we can conclude that the 95% confidence interval for this new drug is -0.076690<=theta<=-0.0066.Thus, every value in that interval satisfy the definition so the entire 95% confidence interval satisfy the definition. We can then conclude that the new drug and old drug are bioequivalent at the 95% confidence interval level.

[ ]: