دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

گزارش تمرین شماره ۱

درس یادگیری تعاملی

پاییز ۱۴۰۱

نام و نام خانوادگی

مهیار ملکی

شماره دانشجویی

۸۱۰۱۰۰۴۷۶

فهرست

چکیده

هدف از این تمرین آشنایی با یک سری مفاهیم آماری پایه و مورد نیاز برای درس یادگیری تقویتی می‌باشد. برای این منظور مسأله ای طراحی شده است که در آن محیطی دارای متغیرهای تصادفی داریم.

در این محیط مقادیری به صورت تصادفی تولید می‌شود که اثبات معنی دار بودن آن‌ها از طریق ابزار و آزمون‌های آماری امکان پذیر می‌باشد.

سوال ۱ – سوال تئوری

هدف سوال

هدف این سوال آشنایی با محیطی می‌باشد که پاداش‌های تصادفی در آن تولید می‌شود. و لزوم استفاده از روش‌های مختلف و تکنیک‌های آماری برای بررسی و مقایسه این پاداش‌ها مد نظر است.

نتایج

در این محیط دو داروی کاهش فشارخون متفاوت داریم که توسط سه پزشک با استراتژی‌های متفاوت، برای مقایسه میزان اثربخشی آنها به داوطلبان تجویز می‌شود. میزان کاهش فشار خون تا سطح مطلوب به عنوان پاداش در نظر گرفته می‌شود.

حال با توجه به شرایط جسمی متفاوت داوطلبان مانند جنسیت و سوابق بیماری و سن آنها یا منشأ افزایش فشار خون یا دلایل دیگر، هر دارو در آزمایش های مختلف پاداشی متفاوت را نتیجه دهد. از آنجایی که در نظر گرفتن تمامی این عوامل تاثیرگزار کاری دشوار و غیرممکن می‌باشد، لذا می‌توان انتظار داشت که پاداش‌ها به صورت تصادفی رخ دهند.

سوال ۲ - سوال پیاده‌سازی

هدف سوال

در این سوال خواسته شده که با استفاده از زبان برنامه‌نویسی پایتون سیاست‌های هر پزشک پیاده‌سازی شود. در ادامه برای مقایسه سیاست‌ها، با تکرار آزمایش بروی ۱۰۰ داوطلب و رسم نمودار میانگین پاداش‌ها، بررسی می‌شود که کدام سیاست پس از ۱۰۰ بار تکرار به طور میانگین پاداش بیشتری را نتیجه می‌دهد.

توضیح پیاده سازی

هر یک از سیاست‌ها به صورت جداگانه به شکل یک تابع پیاده می‌شود که در ادامه بررسی می‌شوند.

- پزشک اول:

این رویکرد طبق خواسته مسأله بر اساس استراتژی win-stay lose-shift پیاده‌سازی شده‌است. در این پیاده‌سازی تابع دریافت پاداش برای ۱۰۰ بار در یک حلقه for تکرار می‌شود. درون حلقه نیز به صورت چند جمله شرطی تو در تو پیاده سازی شده‌است، به این شکل که در صورت رخداد پاداش مثبت با احتمال ۰.۸ در تکرار بعدی همان دارو و با احتمال ۰.۲ داروی دیگر تجویز می‌شود. و در صورت رخداد پاداش منفی با احتمال ۰.۳ در تکرار بعدی همان دارو و با احتمال ۰.۷ داروی دیگر تجویز می‌شود.

```python
def doctor_A(s_id, drugs):

    drug = np.random.choice(drugs)
    rewards_mean, rewards = [], []
    rwrd = 0
    ref = 0

    for i in range(100):
        reward = get_reward(drug, s_id)
        if reward > ref:
            if drug == 1:
                drug = np.random.choice(drugs, p=[0.8, 0.2])
            else:
                drug = np.random.choice(drugs, p=[0.2, 0.8])

        else:
            if drug == 1:
                drug = np.random.choice(drugs, p=[0.3, 0.7])
            else:
                drug = np.random.choice(drugs, p=[0.7, 0.3])

        rwrd += reward
        rewards.append(reward)
        rewards_mean.append(rwrd/(i+1))

    return rewards_mean, rewards
```

شکل ۱- پیاده سازی رویکرد پزشک اول

- پزشک دوم:

در این رویکرد داروی تجویزی برای هر دوره با تابع random.choice کتابخانه نامپای به صورت کاملا تصادفی انتخاب می‌شود.

```
def doctor_B(s_id, drugs):

    rewards_mean, rewards = [], []
    rwrd = 0

    for i in range(100):
        drug = np.random.choice(drugs, p=[0.5, 0.5])
        reward = get_reward(drug, s_id)
        rwrd += reward
        rewards_mean.append(rwrd/(i+1))
        rewards.append(reward)

    return rewards_mean, rewards
```

شکل ۲- پیاده سازی رویکرد پزشک دوم

- پزشک سوم:

در این رویکرد، دو مرحله ۱۰ تایی اول به صورت جداگانه با دو حلقه for و سپس مرحله ۷ تایی و ۲ تایی در قالب یک حلقه while پیاده شده‌است. پاداش متناظر با هر دارو در یک لیست ذخیره می‌شود. سپس در ادامه با مقایسه بیشترین مقدار این دو لیست دارویی که بیشترین مقدار را دارد برای مرحله بعد انتخاب می‌شود. این بدین معنی است که آن مقدار، بیشترین پاداش تا این لحظه می‌باشد.

```
def doctor_C(s_id, drugs):
    rewards, drug_1, drug_2 = [], [], []

    i = 0
    for n in range(10):
        reward = get_reward(1, s_id)
        drug_1.append(reward)
        rewards.append(reward)

    for n in range(10):
        reward = get_reward(2, s_id)
        drug_2.append(reward)
        rewards.append(reward)

    i += 20

    while i < 100:

        if max(drug_1) > max(drug_2):
            for n in range(7):
                reward = get_reward(1, s_id)
                drug_1.append(reward)
                rewards.append(reward)
        else:
            for n in range(7):
                reward = get_reward(2, s_id)
                drug_2.append(reward)
                rewards.append(reward)

        for n in range(3):
            drug = np.random.choice(drugs)
            reward = get_reward(drug, s_id)
            if drug == 1:
                drug_1.append(reward)
            else:
                drug_2.append(reward)
            rewards.append(reward)

        i += 10

    rewards_mean = []
    rwrd =0
    for i in range(100):
        rwrd += rewards[i]
        rewards_mean.append(rwrd/(i+1))

    return rewards_mean, rewards
```

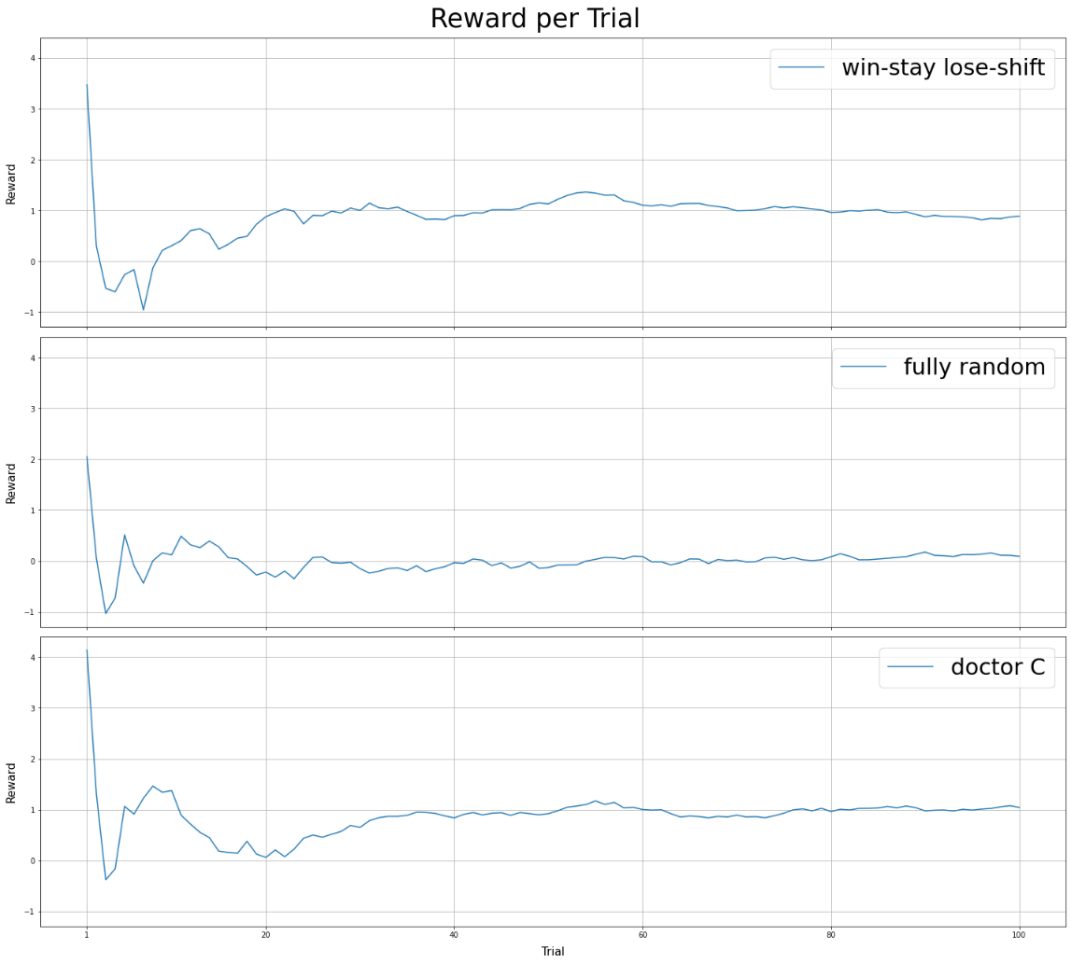
شکل ۳- پیاده سازی رویکرد پزشک سوم

نتایج

نمودار میانگین پاداش برای رویکرد هر سه پزشک در شکل ۴ قابل مشاهده است.

دو نکته از نمودارها قابل استخراج است، یک اینکه در آزمایش‌های اولیه، هر سه نمودار دارای نوسان بوده و پس از تکرار آزمایش‌ها، با میانگین‌گیری تجمعی طبق انتظار هر نمودار به مقداری مشخص همگرا می‌شود. لذا نمی‌توان با تعداد تکرار کم راجب سیاست‌ها نتیجه‌گیری کرد.

نکته دوم این که در مقایسه عملکرد این سه پزشک مشاهده می‌شود که رویکرد پزشک سوم در نهایت میانگین پاداش بیشتری را نتیجه داده و پزشک دوم کمترین میانگین پاداش را دارد. البته لازم به ذکر است که به دلیل ماهیت تصادفی پاداش‌ها که در سوال قبل راجب آن بحث شد، نتایج بدست آمده در برخی از اجراها متفاوت بودند، ولی به طور کلی و در بیشتر اجراها این نتایج یعنی بهتر بودن عملکرد پزشک سوم از نظر میانگین پاداش برقرار است.



شکل ۴- نمودارهای پاداش بر اساس آزمایشات رویکرد هر سه پزشک

روند اجرای کد پیاده‌سازی

سیاست‌های هر پزشک در فایل doctors.py پیاده‌سازی شده و این فایل در هر سوال در صورت نیاز بارگزاری می‌شود. رسم نمودارها در فایل Q2.ipynb صورت گرفته‌است.

سوال ۳ - سوال پیاده‌سازی

هدف سوال

در این سوال هدف این است که با تکرار سوال قبل میزان واریانس و ریسک را در رویکرد هر سه پزشک مقایسه و بررسی کنیم.

توضیح پیاده سازی

در این قسمت چنانچه در شکل ۵ قابل مشاهده است، با استفاده از کتابخانه scipy مقدار t-score را با استفاده از ضریب آلفا داده شده و درجه آزادی که از اندازه دادگان بدست آوردیم، محاسبه می‌کنیم. لازم به ذکر است که در اینجا بدلیل کم بودن تعداد داده‌ها از توضیع t بجای توزیع نورمال استفاده می‌شود.

سپس با استفاده از کتابخانه نامپای و محاسبه میانگین و انحراف معیار داده‌ها روی محور مورد نظر، به محاسبه مقادیر بازه اطمینان می‌پردازیم. برای محاسبه بازه اطمینان از فرمول زیر استفاده می‌شود:

$$CI = \bar{x} \pm z\frac{s}{\sqrt{n}}$$

```
size = 5 # or 20
def confidence_interval(size, doctor):
    t_score = np.abs(scipy.stats.t.ppf(q=0.05/2, df=size-1))
    rwrds = np.zeros([size,100])

    for i in range(size):
        rewards_mean, _ = doctor(s_id, drugs)
        rwrds[i,:] = rewards_mean

    rwrds_mean, rwrds_std = rwrds.mean(axis=0), rwrds.std(axis=0)
    ci_up = rwrds_mean + t_score*rwrds_std/np.sqrt(size),
    ci_down = rwrds_mean - t_score*rwrds_std/np.sqrt(size)
    return ci_up, ci_down, rwrds_mean
```

شکل ۵- پیاده سازی محاسبات بازه اطمینان

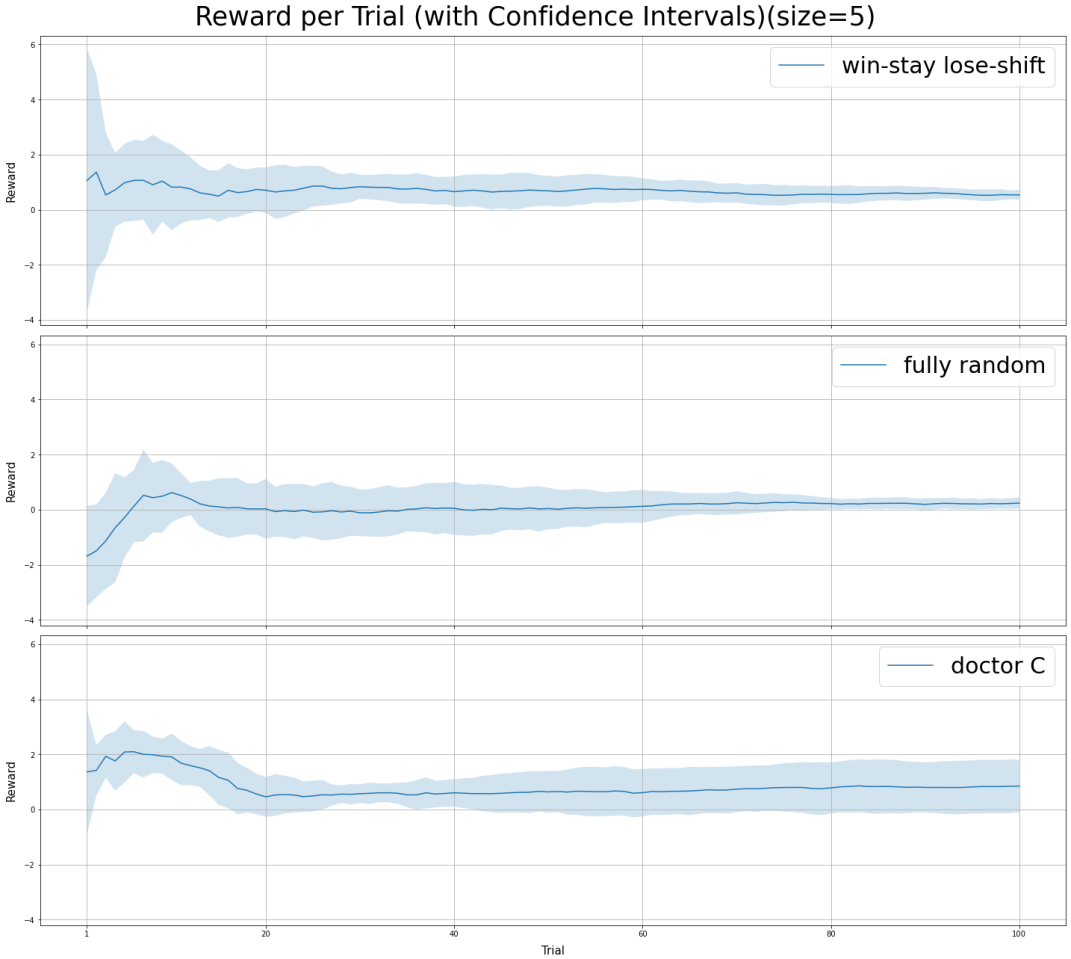
نتایج

دلیل تکرار سوال ۲: به دلیل ماهیت تصادفی پاداش‌ها هر بار اجرای سیاست‌ها ممکن است پاسخ متفاوتی را نتیجه بدهد (حتی در میانگین پاداش نهایی که نمودار به آن همگرا می‌شود) لذا لازم است با تکرار چند باره آزمایش این نایقینی را کاهش داد.

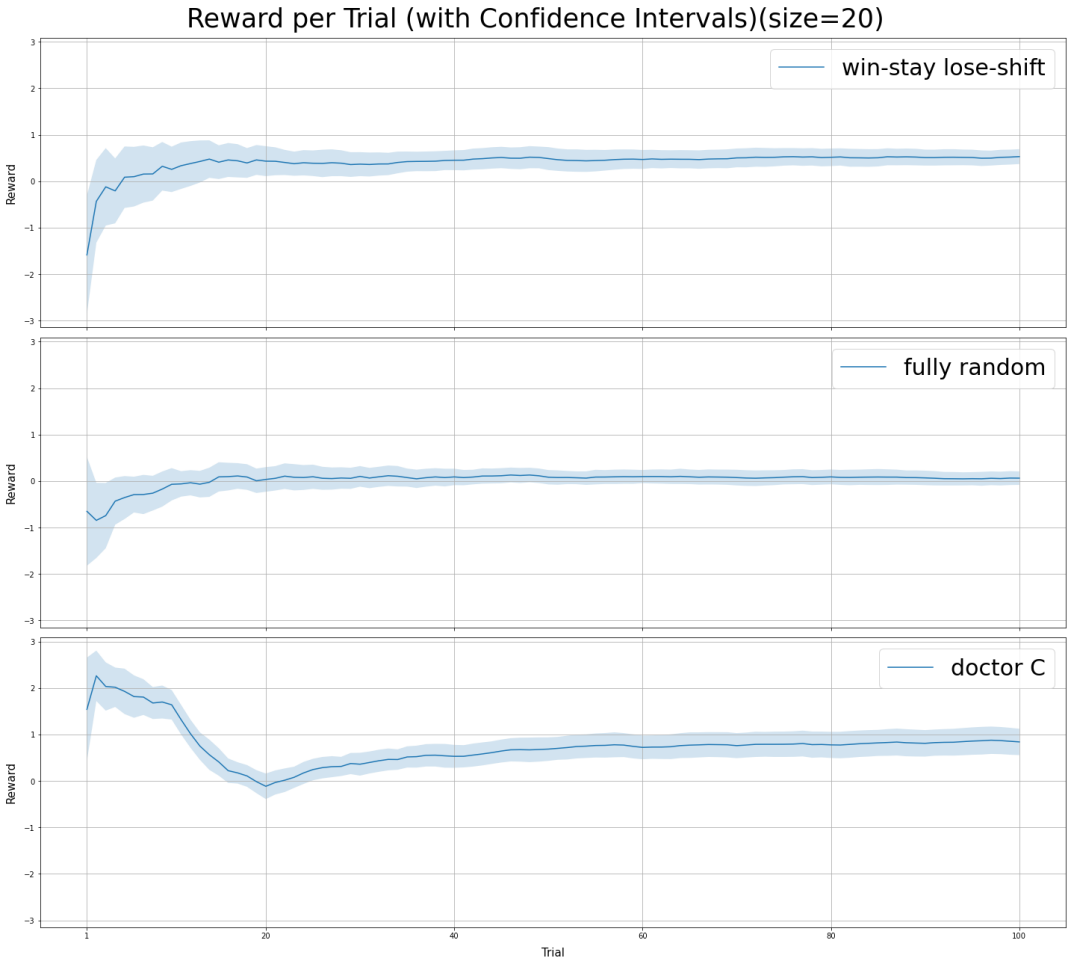
نمودارهای به دست آمده به همراه بازه‌های اطمینان برای هر سه پزشک در شکل ۶ قابل مشاهده است. چهار نکته از نمودارها قابل استخراج است:

- نکته اول اینکه با مقایسه بازه اطمینان پاداش رویکردهای مختلف پزشک‌ها می‌توان دریافت که رویکرد پزشک سوم نایقینی بیشتری داشته و واریانس تغییرات آن بیشتر است زیرا طول بازه اطمینان این پزشک بیشتر شده‌است.
- نکته دوم آن است که طبق انتظار با افزایش تعداد دفعات آزمایش از ۵ به ۲۰ بازه‌های اطمینان به طور کلی کوچکتر شده زیرا تکرار بیشتر آزمایش‌ها باعث اطمینان بیشتری نسبت به نتایج می‌شود.

- نکته سوم اینکه همانند سوال قبل همچنان پزشک سوم از لحاظ میانگین پاداش عملکرد بهتری دارد، این امر در اینجا با تکرار بیشتری بدست آمده و قابل اطمینان تر می‌باشد.
- نکته چهارم اینکه چنانچه در نمودارها پیداست در آزمایش‌های ابتدایی بازه اطمینان بسیار بزرگتر بوده که نشان‌دهنده نایقینی است. طبق انتظار با تکرار بیشتر آزمایش، این بازه اطمینان کوچکتر شده و اطمینان بیشتری به همراه دارد.



شکل ۶- نمودارهای پاداش براساس آزمایش رویکرد هر سه پزشک به همراه بازه های اطمینان با ۵ بار تکرار



شکل ۷–نمودارهای پاداش براساس آزمایش رویکرد هر سه پزشک به همراه بازه های اطمینان با ۲۰ بار تکرار

روند اجرای کد پیاده‌سازی

سیاست‌های هر پزشک در فایل doctors.py پیاده‌سازی شده‌است، همچنین محاسبه بازه اطمینان و رسم نمودارها در فایل Q3.ipynb صورت گرفته‌است.

سوال ۴ - سوال پیاده‌سازی

هدف سوال

در این سوال خواسته شده است که با ۱۰ بار تکرار رویکرد پزشک‌ها، نمودار boxplot مربوطه را برای هر سه پزشک رسم کنیم. با استفاده از نمودار باکسپلات می‌توان پاداش میانگین هر رویکرد و واریانس آن را به خوبی مقایسه کرد.

توضیح پیاده سازی

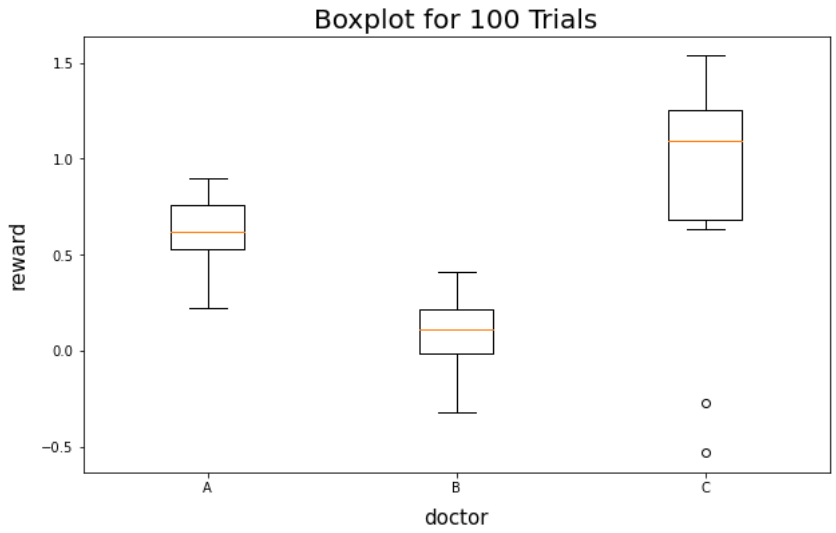
در این قسمت چنانچه در شکل ۸ قابل مشاهده است، پیاده سازی مانند قبل بوده فقط از یک حلقه for استفاده شده تا ۱۰ بار تکرار را پیاده سازی کرده و آزمایش صدم را در یک آرایه ذخیره کنیم.

```
size = 10
rwrds = np.zeros([size, 3])

for i in range(size):
    rewards_mean, _ = doctor_A(s_id, drugs)
    rwrds[i, 0] = rewards_mean[-1]

    rewards_mean, _ = doctor_B(s_id, drugs)
    rwrds[i, 1] = rewards_mean[-1]

    rewards_mean, _ = doctor_C(s_id, drugs)
    rwrds[i, 2] = rewards_mean[-1]

plt.boxplot(rwrds)
```

شکل ۸ - پیاده سازی ترسیم نمودار باکسپلات

نتایج

چنانچه در شکل ۹ قابل مشاهده است، نتایج بدست آمده در قسمت‌های قبلی در اینجا نیز مشاهده و تایید می‌شود. رویکرد پزشک سوم بیشترین میانگین پاداش را داشته ولی واریانس تغییرات آن نیز بیشتر بوده و باعث کاهش اطمینان می‌شود.



شکل ۹ - نمودار باکسپلات آزمایش صدم

روند اجرای کد پیاده‌سازی

سیاست‌های هر پزشک در فایل doctors.py پیاده‌سازی شده و رسم نمودارها در فایل Q4.ipynb صورت گرفته‌است.

سوال ۵ - سوال پیاده‌سازی

هدف سوال

در این سوال انتظار می‌رود با استفاده از تست‌های آماری مانند z-test یا t-test ، ادعای اثربخشی بهتر داروی اول را تحقیق کرد و بررسی کرد که این ادعا از نظر آماری معنی‌دار هست یا خیر.

توضیح پیاده سازی

در این قسمت، فرمول زیر در شکل ۱۰ پیاده شده تا مقدار z-value بدست آید. سپس با استفاده از کتابخانه scipy و تابع norm.cdf احتمال یک طرفه سمت چپ و با تفریق این مقدار از ۱ احتمال یک طرفه سمت راست بدست می‌آید.

$$z = \frac{\overline{X1} - \overline{X2}}{\sqrt{\sigma^2 \left(\frac{1}{n1} + \frac{1}{n2}\right)}}$$

```
drug_A, drug_B = [], []

for i in range(50):
    drug_A.append(get_reward(1, s_id))
    drug_B.append(get_reward(2, s_id))

z_stat = ( np.mean(drug_A) - np.mean(drug_B) ) / np.sqrt((
np.std(drug_A)**2 + np.std(drug_B)**2 ) / 100)
p_value = 1 - scipy.stats.norm.cdf(z_stat) # one-sided ( right_tail )
```

شکل ۱۰ - پیاده سازی تست آماری **z-test**

نتایج

طبق صورت سوال فرض صفر و فرض جایگزین به شرح زیر است:

- فرض صفر: داروی دوم اثربخشی بهتری از داروی اول دارد ($\mu_1 \leq \mu_2$)
- فرض جایگزین: داروی اول اثربخشی بهتری از داروی دوم دارد ($\mu_1 > \mu_2$)

بعد از اجرای کد، مقدار 4e-11 برای p-value بدست می‌آید. این مقدار بسیار کوچکتر از مقدار آلفا یعنی ۰.۰۵ بوده، در نتیجه با اطمینان خوبی فرض صفر رد شده و فرض جایگزین تایید می‌شود. لذا از دید آماری داروی A به طور معناداری پاداش بیشتری را نسبت به داروی B نتیجه می‌دهد.

روند اجرای کد پیاده‌سازی

سیاست‌های هر پزشک در فایل doctors.py پیاده‌سازی شده و محاسبه آماره‌ها در فایل Q5.ipynb صورت گرفته‌است.

سوال ۶ – سوال تئوری

هدف سوال

هدف از این سوال آشنایی با روش A/B testing و مقایسه آن با رویکردهای گفته شده در تمرین می‌باشد.

نتایج

چنانچه می‌دانیم روش A/B testing این گونه عمل می‌کند: به عنوان مثال برای آزمایش دو تمپلیت برای یک اپلیکیشن موبایل، این دو تمپلیت به صورت تصادفی و به طور یکنواخت به کاربران نمایش داده می‌شود(نیمی از کاربران تمپلیت A و نیمی دیگر تمپلیت B)، سپس بازدهی و عملکرد برنامه طی زمان مشخصی مانند ۱۰ روز سنجیده می‌شود تا مشخص شود کدام تمپلیت عملکرد بهتری داشته است. بنابراین **به نظر می‌رسد این روش به رویکرد پزشک دوم شبیه است زیرا در هر دو، نمونه ها به صورت تصادفی به داوطلبین ارایه می‌شوند.**

مقایسه با رویکرد سایر پزشک‌ها:

در رویکرد پزشک سوم به دلیل انتخاب بیشینه پاداش بین کل آزمایش‌های قبلی و حلقه های ۱۰ تایی ابتدایی آزمون، تا انتها با احتمال زیادی فقط دارویی که در ۱۰ آزمایش اول خود بیشینه پاداش بیشتری داشته انتخاب خواهد شد، همچنین در رویکرد پزشک اول داروی انتخابی برای آزمایش بعد با توجه به آزمایش قبلی آن انتخاب می‌شود، به این صورت که اگر منجر به پاداش مثبت شود با احتمال زیادی همان دارو و اگر منجر به پاداش منفی شود با احتمال زیادی داروی دیگر انتخاب می‌شود. در صورتی که در A/B testing انتخاب داروها به صورت کاملا تصادفی است و به عملکرد قبلی داروها توجهی نمی‌شود، لذا در روش A/B testing هر دو دارو تا انتهای آزمون شانس یکسانی برای آزمایش خواهندداشت.

معایب و مزایای روش A/B testing :

در این روش همانطور که گفته شد تا انتهای آزمایش هر دو دارو به تعداد مساوی تست خواهندشد، لذا ممکن است افراد بیشتری تحت عوارض داروی بدتر قرار بگیرند، بنابراین این روش باعث اتلاف زمان و هزینه و سلامت افراد می‌شود. در حالی که در روشهای دیگر دارویی که پاداش کمتری دارد در تکرارهای بعدی با احتمال کمتری تجویز می‌شود.

از طرف دیگر در روش‌های RL به دلیل این که تعداد داده‌ها به سمت یکی از داروها بایاس می‌شود(مثلا در اینجا داروی اول به تعداد بیشتری مورد آزمایش قرار می‌گیرد)، لذا نمی‌توان به خوبی از نظر آماری نتایج را بررسی کرده به یک جواب معنادار از نظر آماری رسید. در حالی که در روش A/B testing هر دو دارو به تعداد برابر مورد آزمایش قرار می‌گیرند و می‌توان نتایج آن از نظر آماری تحلیل کرد.

منابع