سوال ۱:

بخش ۱:

۱- اتوانکودر ها می توانند به عنوان استخراج کننده ویژگی برای کارهای طبقه بندی یا رگرسیون استفاده شوند. اتوانکودر ها داده‌های بدون برچسب را می‌گیرند و کدگذاری‌های کارآمدی را در مورد ساختار داده‌ها یاد می‌گیرند که می‌توانند برای وظایف یادگیری تحت نظارت استفاده شوند. پس از آموزش یک شبکه اتوانکودر با استفاده از نمونه‌ای از داده‌های آموزشی، می‌توانیم بخش رمزگشای اتوانکودر را نادیده بگیریم و فقط از رمزگذار برای تبدیل داده‌های ورودی خام با ابعاد بالاتر به فضای رمزگذاری شده با ابعاد پایین‌تر استفاده کنیم. این بعد پایین‌تر از داده‌ها می‌تواند به عنوان یک ویژگی برای کارهای تحت نظارت استفاده شود (supervised tasks).

۲- داده‌های ورودی خام در دنیای واقعی اغلب ماهیت نویز دارند و برای آموزش یک مدل نظارت شده قوی به داده‌های تمیز و بدون نویز نیاز داریم. اتوانکودر ها می توانند برای حذف نویز داده ها استفاده شوند. حذف نویز از تصویر یکی از کاربرد هاست که در آن اتوانکودر ها سعی می کنند تصویر بدون نویز را از یک تصویر ورودی همراه با نویز بازسازی کنند. تصویر ورودی همراه با نویز به عنوان ورودی به اتوانکودر وارد می شود و خروجی بدون نویز بازسازی می شود. هنگامی که وزنه اتوانکودر (the autoencoder weights) آموزش داده شدند، می توان از آنها بیشتر برای حذف نویز تصویر خام استفاده کرد.

۳- فشرده سازی تصویر یکی دیگر از کاربردهای اتوانکودر است. تصویر ورودی خام را می توان به اتوانکودر ارسال کرد و یک بعد فشرده از داده های رمزگذاری شده به دست آورد. معمولاً اتوانکودر ها برای فشرده سازی داده ها چندان خوب نیستند.

۴- تشخیص ناهنجاری یکی دیگر از کاربردهای مفید اتوانکودر است. یک مدل تشخیص ناهنجاری می تواند برای شناسایی یک تراکنش متقلبانه یا هر کار نظارت شده بسیار نامتعادل استفاده شود. ایده این است که اتوانکودر را فقط بر روی داده های نمونه یک کلاس (کلاس اکثریت) آموزش دهیم. به این ترتیب شبکه قادر است ورودی را با تلفات خوب یا کمتر بازسازی کند. حال، اگر یک داده نمونه از کلاس هدف دیگر از طریق اتوانکودر ارسال شود، منجر به از دست دادن بازسازی نسبتاً بزرگتر می شود.

۵- از اتوانکودر حذف نویز می توان برای نسبت دادن مقادیر گمشده در مجموعه داده استفاده کرد. ایده این است که با قرار دادن تصادفی مقادیر گمشده در داده‌های ورودی و تلاش برای بازسازی داده‌های خام اصلی با به حداقل رساندن تلفات بازسازی، یک شبکه رمزگذار خودکار آموزش دهیم.

بخش ۲:

هدف generating models این است که یک توزیع داده به عنوان مثال P_D را از روی نمونه ها یاد بگیرد و نمونه های جدید را بر اساس این توزیع یاد گرفته تولید کند. مشکل VAE ها این است که علی رغم اینکه تصاویری که از روی آن آموزش دیده اند بسیار واضح هستند اما تصاویر تولیدی تار می شوند. این مشکل به دلیل فرمول مربوط به بهینه سازی است. نحوه بهینه سازی آنها این است که مقدار evidence lower bound(ELBO) را ماکزیمم می کنند. در این فرمول بهینه سازی ترم ها و پارامتر هایی وجود دارند که نمونه را به فضای پنهان مپ می کنند. زمانی که فضای پنهان از یکی از ترم ها بُعد کمتری داشته باشد، تصویر تولیدی تار خواهد بود.

$$\mathbb{E}_{z\sim q_\phi(z|x)}\, log[p_\theta(x|z)] - D_{KL}[q_\phi(z|x)||p(z)]$$

فرمول فوق همان فرمول بهینه سازی است. درصورتی که z که همان latent یا فضای پنهان است از x بُعد کمتری داشته باشد با تصاویر تار مواجه می شویم.

منبع:

ETH Library/ Explicitly Minimizing the Blur Error of Variational Autoencoders/ Author(s): Bredell, Gustav; Flouris, Kyriakos

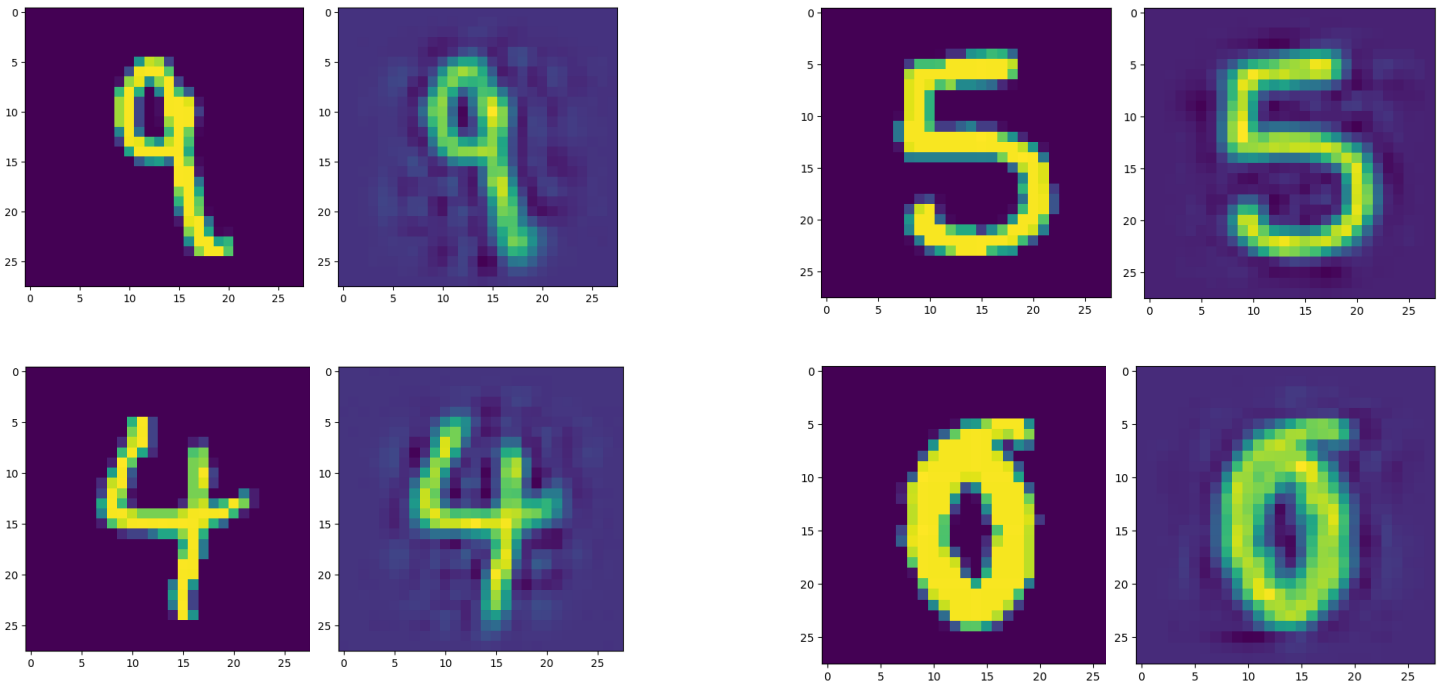
لینک مقاله:

https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwj17duAjoCEAxV3ywIHHWrAAdQQFnoECCIQAQ&url=https%3A%2F%2Fwww.research-collection.ethz.ch%2Fbitstream%2Fhandle%2F20.500.11850%2F594280%2F1%2FExplicitly_Minimizing_the_Blur_Error_of_Variational_Autoencoders.pdf&usg=AOvVaw3iTEDTiwEXax4kR_ySSenl&opi=89978449

بخش ۳:

ابتدا به کمک قطعه کد های موجود در فایل توضیحات، داده های تست را پیش پردازش و بازسازی می کنیم. برای مشاهده تصاویر اصلی و بازسازی شده به طور رندوم چهار نمونه میگیریم و به کمک توابع کتابخانه matplotlib تصاویر را می کشیم که نتیجه به شکل زیر است:

تصویر سمت راست تصویر بازسازی شده و تصویر سمت چپ تصویر اصلی است.



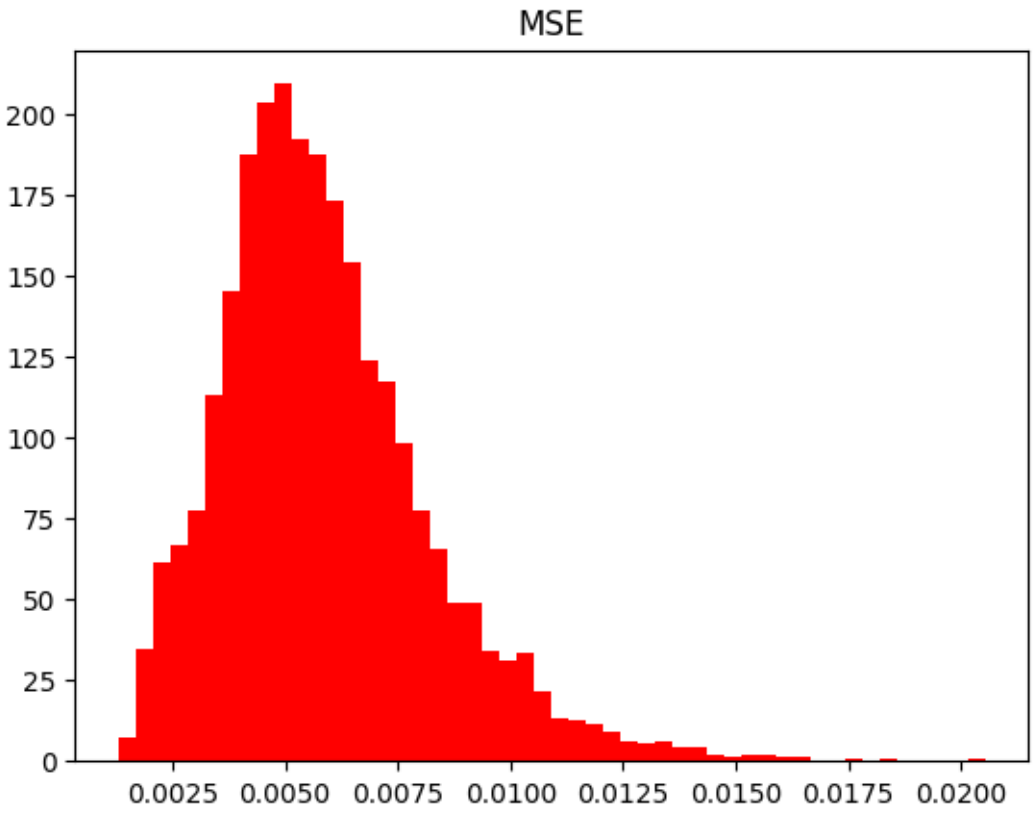
بخش ۳ د:

برای محاسبه MSE داریم:

$$\text{MSE} = \boxed{\frac{1}{n}\sum_{i=1}^{n}}\boxed{(Y_i - \hat{Y}_i)^2}$$

Mean Error Squared

به کمک این فرمول تابع مربوطه را می سازیم. در نهایت به کمک کتابخانه matplotlib هیستوگرام مربوط به داده ها را رسم می کنیم که نمودار به شکل زیر خواهد بود:



بخش ۳ ه:

پاسخ: خیر

به کمک توابع کتابخانه numpy میتوان به راحتی میانگین و انحراف معیار را محاسبه کرد. سپس به کمک قطعه کد داده شده آزمون مورد نظر را اجاره می کنیم و مقدار p-value را چاپ می کنیم که برابر مقدار زیر است:

```
p_value=  4.5387216956056 57e-43
```

با توجه به اینکه p-value از ۰.۰۵ کوچکتر است بنابراین نمی توان پذیرفت که داده های mse از توزیع نرمال برآورد شده پیروی می کنند.

سوال۲:

بخش ۱:

نقطه پرت:

در آمار، نقطه پرت داده ای است که به طور قابل توجهی با مشاهدات دیگر متفاوت است. داده پرت ممکن است به دلیل تغییر در اندازه گیری یا نتیجه خطای تجربی باشد. مورد دوم گاهی اوقات از مجموعه داده ها حذف می شوند. داده پرت می تواند مشکلات جدی در تجزیه و تحلیل های آماری ایجاد کند. نقاط پرت می توانند به طور تصادفی در هر توزیعی رخ دهند، اما می توانند رفتار یا ساختارهای جدید در مجموعه داده یا خطای اندازه گیری را نشان دهند. در مورد خطای اندازه‌گیری، فرد می‌تواند آن‌ها را کنار بگذارد یا از روشی استفاده کند که نسبت به نقاط پرت مقاوم باشد.

نقطه اهرمی:

نقاط اهرمی نقاط داده ای هستند که تأثیر قابل توجهی بر ضرایب رگرسیون برآورد شده دارند. این نقاط می توانند خط رگرسیون را مخدوش کرده و بر تناسب کلی و تفسیر مدل تأثیر بگذارند. برخلاف نقاط پرت، نقاط اهرمی ممکن است مقادیر افراطی نداشته باشند، اما در پیش بینی های خود دارای مقادیر افراطی هستند که به ماهیت تأثیرگذار آنها کمک می کند.

نقاط اهرمی نقاطی هستند که مشاهدات غیر معمول را در فضای x تشخیص رگرسیون اندازه گیری می کنند

اهرمی بالا می‌تواند بر برآورد حداقل مربعات پارامترها تأثیر بدی بگذارد. آسیب نقاط اهرمی بالا زمانی که مقادیر پرت در داده ها وجود داشته باشد بیشتر است (مقادیر به طور غیرعادی از خط رگرسیون فاصله دارند).

نقطه پرت-اهرمی:

اگر یک نقطه با اهرم بالا نیز یک نقطه پرت باشد، باعث می شود که خط کمترین مربعات بسیار کمتر دقیق باشد.

در حالی که نقاط پرت بر اساس مقادیر شدید آنها در متغیر پاسخ شناسایی می شوند، نقاط اهرمی بر اساس مقادیر شدید آنها در متغیرهای پیش بینی شناسایی می شوند. نقاط پرت می توانند تأثیر قابل توجهی بر روی برازش و برآورد ضرایب داشته باشند، در حالی که نقاط اهرمی در درجه اول بر برآورد ضرایب تأثیر می گذارد.

نقاط پرت این پتانسیل را دارند که تحلیل های آماری را تحریف کنند، که منجر به تخمین های مغرضانه و پیش بینی های نادرست می شود.

بخش ۲:

ضریب تعیین:

در آمار، ضریب تعیین که R^2 یا r^2 نشان داده می شود و "R مربع" تلفظ می شود، نسبت تغییرات متغیر وابسته است که از متغیر(های) مستقل قابل پیش بینی است. آماره ای است که در چارچوب مدل‌های آماری استفاده می‌شود که هدف اصلی آن یا پیش‌بینی نتایج آینده یا آزمون فرضیه‌ها بر اساس سایر اطلاعات مرتبط است. این معیار بر اساس نسبت تغییرات کل نتایج توضیح داده شده توسط مدل، معیاری از چگونگی تکرار نتایج مشاهده شده توسط مدل ارائه می‌کند.

نحوه محاسبه به شکل زیر است:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad SS_{\text{res}} = \sum_i (y_i - f_i)^2 = \sum_i e_i^2$$

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \qquad SS_{\text{tot}} = \sum_i (y_i - \bar{y})^2$$

بخش ۳:

رگرسیون خطی یک روش آماری است که برای پیش‌بینی یک متغیر وابسته پیوسته (متغیر هدف) بر اساس یک یا چند متغیر مستقل (متغیرهای پیش‌بینی‌کننده) استفاده می‌شود. این تکنیک یک رابطه خطی بین متغیرهای وابسته و مستقل را فرض می‌کند که به این معنی است که متغیر وابسته متناسب با تغییرات متغیرهای مستقل تغییر می‌کند. از رگرسیون خطی برای تعیین میزانی استفاده می شود که یک یا چند متغیر می توانند مقدار متغیر وابسته را پیش بینی کنند.

در واقع بر اساس داده هایی که داریم تلاش می کنیم تا معادله خطی را پیدا کنیم که داده ها روی آن یا نزدیک به آن باشند بنابراین اینگونه عمل می کنیم.

فرض می کنیم معادله خط رگرسیونی به شکل زیر است:

$$h(x_i) = \beta_0 + \beta_1 x_i$$

که خروجی آن مقدار حدسی ما برای ورودی x_i می باشد. β0 و β1 ضرایب معادله رگرسیونی هستند. با توجه به اینکه از least square error استفاده می کنیم، فرمول هایی که در کد استفاده شده به فرم زیر هستند:

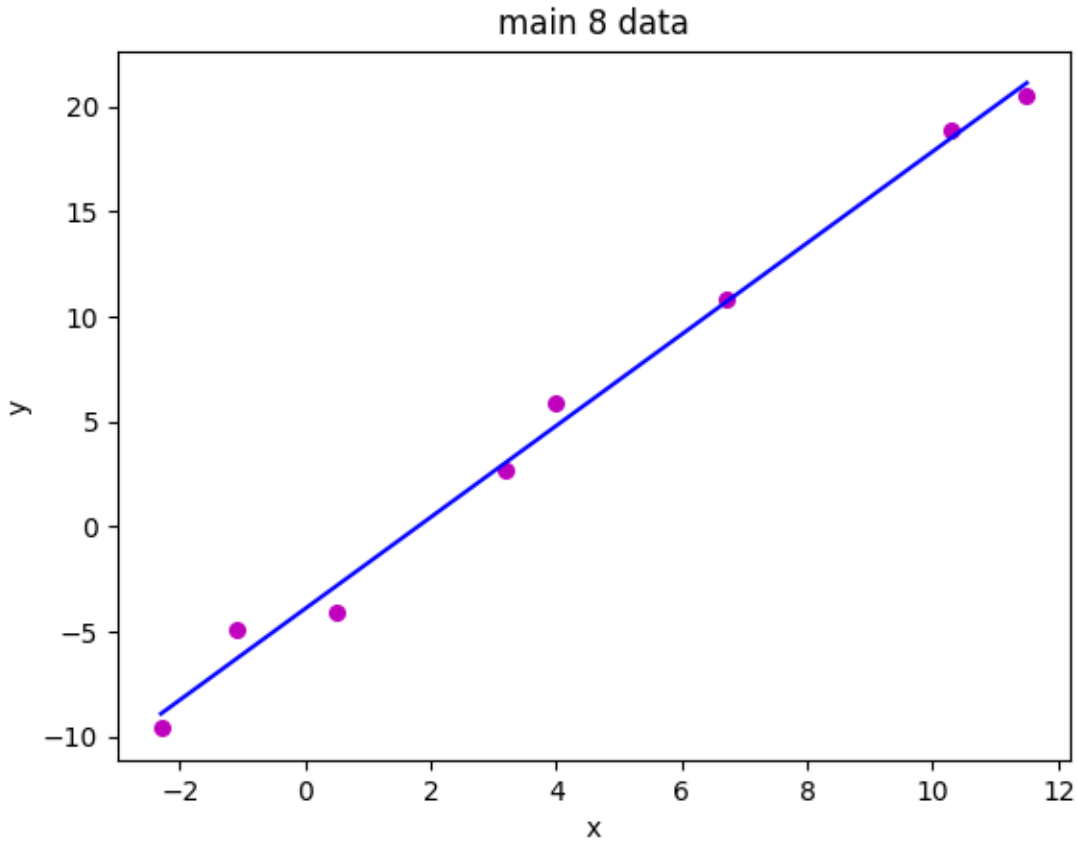
$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

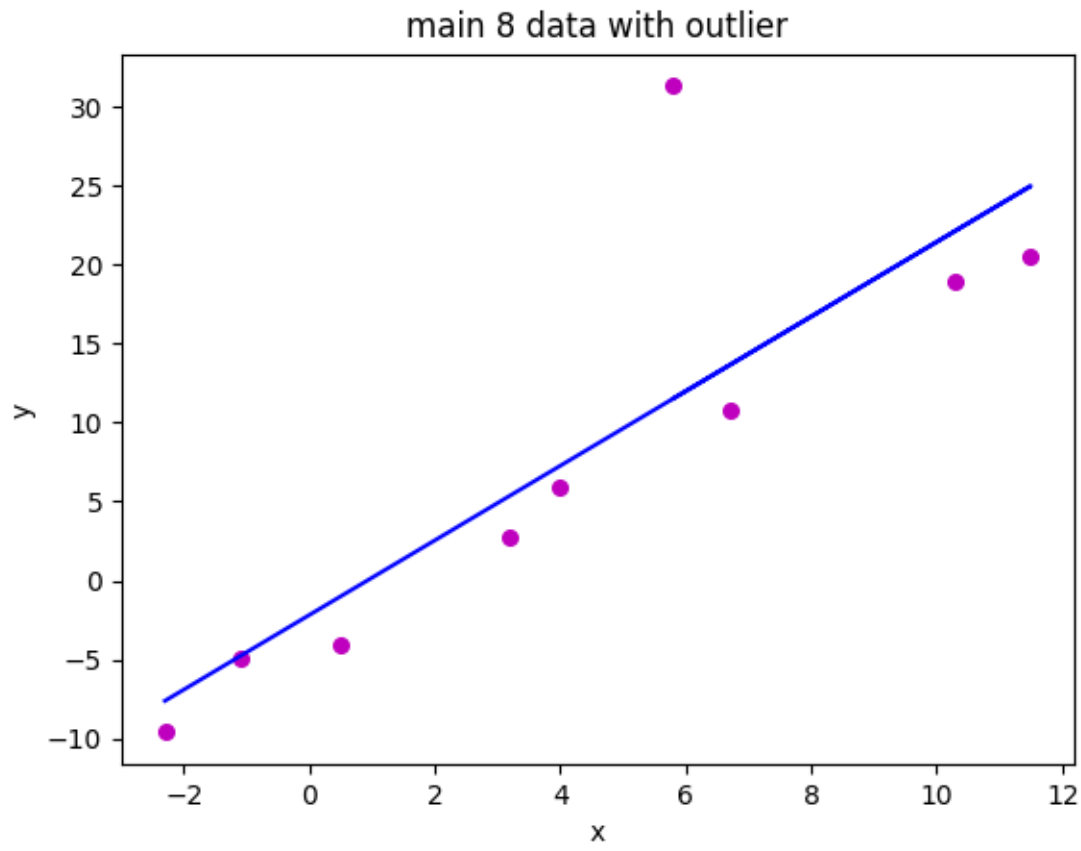
$$\beta_1 = \frac{SS_{xy}}{SS_{xx}}$$

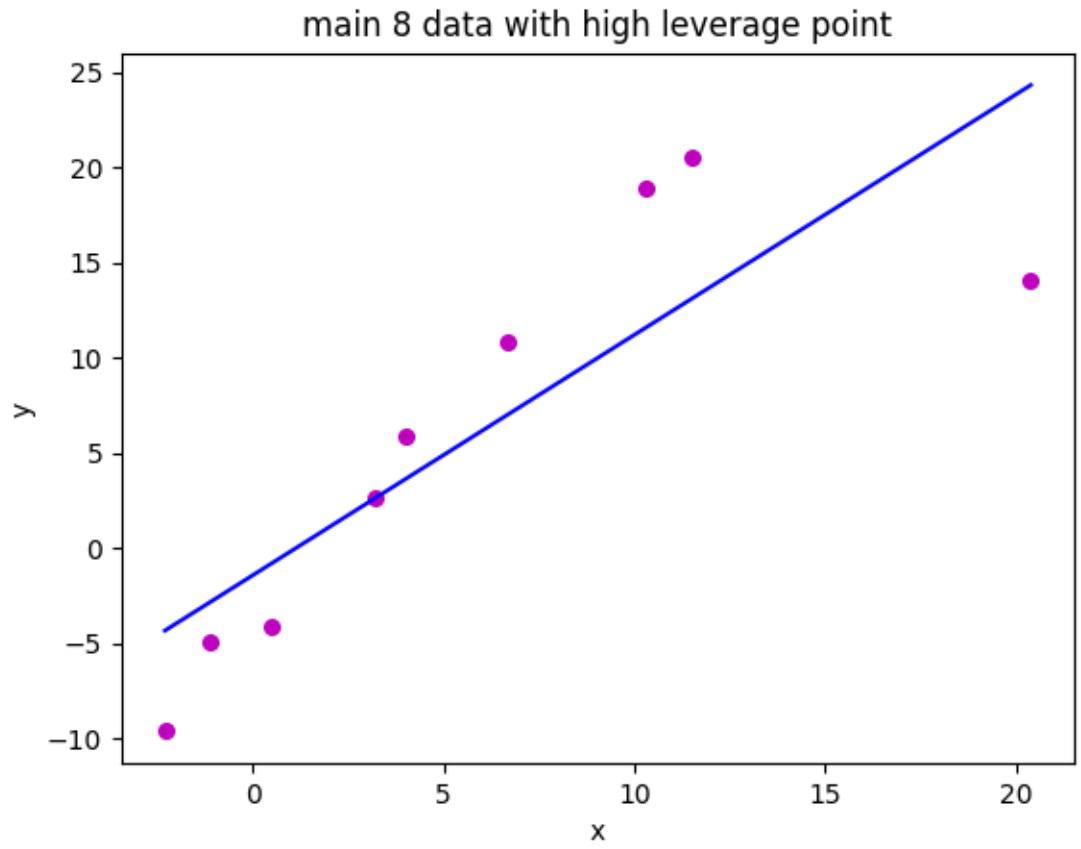
$$SS_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^{n} y_i x_i - n\bar{x}\bar{y}$$

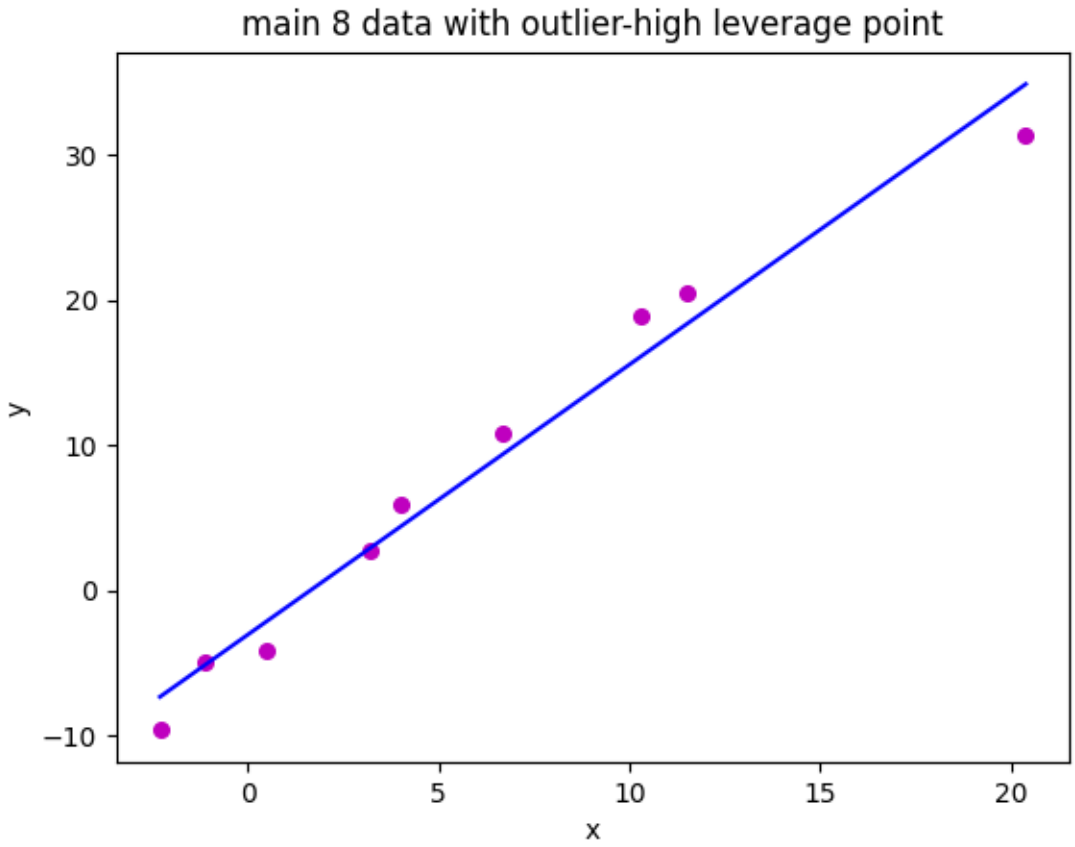
$$SS_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - n(\bar{x})^2$$

پس از رسم نمودار داده ها و خط رگرسیونی نمودار های زیر به دست می آید:




main 8 data with outlier
y
x
30
25
20
15
10
5
0
−5
−10
−2
0
2
4
6
8
10
12


main 8 data with high leverage point
y
x
25
20
15
10
5
0
−5
−10
0
5
10
15
20




```
r^2 =  0.9931142293628097
r^2 =  0.6943381680789323
r^2 =  0.7069879724740986
r^2 =  0.9738367949787371
```

مقادیر ضریب تعیین به ترتیب در بالا آمده است.

همان طور که مشخص است، معادله رگرسیونی زمانی که برای داده های اصلی محاسبه شده تا حد خیلی خوبی بر داده ها منطبق است. یعنی یا داده ها روی خط رگرسیونی قرار گرفته اند یا فاصله کمی با آن دارند.

اما زمانی که داده پرت وارد محاسبات می شود، طبق نمودار دوم مشاهده می شود که خط رگرسیونی محاسبه شده با داده های اصلی فاصله زیادی دارد و بیشتر به سمت نقطه پرت که مقدار زیادی دارد متمایل شده است.

نقاط اهرمی بالا در رگرسیون خطی، نقاطی هستند که دارای مقادیر متغیر مستقل بسیار غیرمعمول در هر جهت از میانگین (بزرگ یا کوچک) هستند. همان طور که از نمودار سوم مشخص است، این نقطه مثل یک اهرم عمل کرده و خط رگرسیونی را به سمت خودش کشیده.

زمانی که نقطه ای را اضافه کردیم که هم اهرم بالا هم پرت است طبق نمودار آخر، این دو تا حد خوبی رو معادله تاثیر گذاشته اند و نسبت به خط رگرسیونی اصلی انحراف داشته است.

بخش ۴:

رگرسیون قوی(robust regression) جایگزینی برای رگرسیون least square است. زمانی که داده ها با موارد پرت یا مشاهدات تأثیرگذار آلوده شده باشند می توان از آن استفاده کرد.

رگرسیون خطی قوی نسبت به رگرسیون خطی استاندارد حساسیت کمتری به نقاط پرت دارد. رگرسیون خطی استاندارد از برازش حداقل مربعات معمولی برای محاسبه پارامترهای مدل استفاده می کند که داده های پاسخ را به داده های پیش بینی با یک یا چند ضریب مرتبط می کند.

تعداد زیادی روش به عنوان روش های قوی رگرسیونی پیشنهاد شده اند که تلاش آنها بر این بوده که میزان اثرگذاری نقاط پرت و نقاط اهرمی را کاهش دهند. به عنوان مثال یک از آنها M-estimation است. این روش نسبت به مشاهدات غیر معمول روی y مقاوم است اما همچنان بر نقاط اهرمی روی x حساس است.

یک دیگر از روش ها R-estimation است که در تلاش است تا مجموع مربعات ranked residuals را مینیمم کند.

روش های دیگری مانند LMS، LTS و... وجود دارند. بر اساس اینکه مشاهدات پرت یا اهرمی چگونه هستند و هرکدام از این مدل ها نسبت به کدام یک حساسیت کمتری دارند می توان ار آنها به least square استفاده کرد. گاهی اوقات می توان داده های پرت را از محاسبات حذف کرد.

منبع:

Robust Linear Regression: A Review and Comparison/Chun Yu and Weixin Yao

لینک مقاله:

https://escholarship.org/content/qt8h14g71j/qt8h14g71j_noSplash_dfc95f7415c4f3d46bd17116a68ef4ff.pdf?t=ode1qj

---

سوال ۳:

بخش۱:

برای داده های کمی میتوان مقادیر نامعلوم را با میانگین داده ها جایگزین کرد. به طور کلی میتوان گفت دو راه برای حل مشکل داده های نامعلوم داریم: ۱- حذف آنها از dataset ۲- جایگزین کردن آنها با مقدار مناسب

روش اول توصیه نمی شود. چراکه برخی جاها لازم است این موارد بررسی شوند و در برخی شرایط حذف کردن آنها تحلیل های آماری را دچار مشکل می کند.

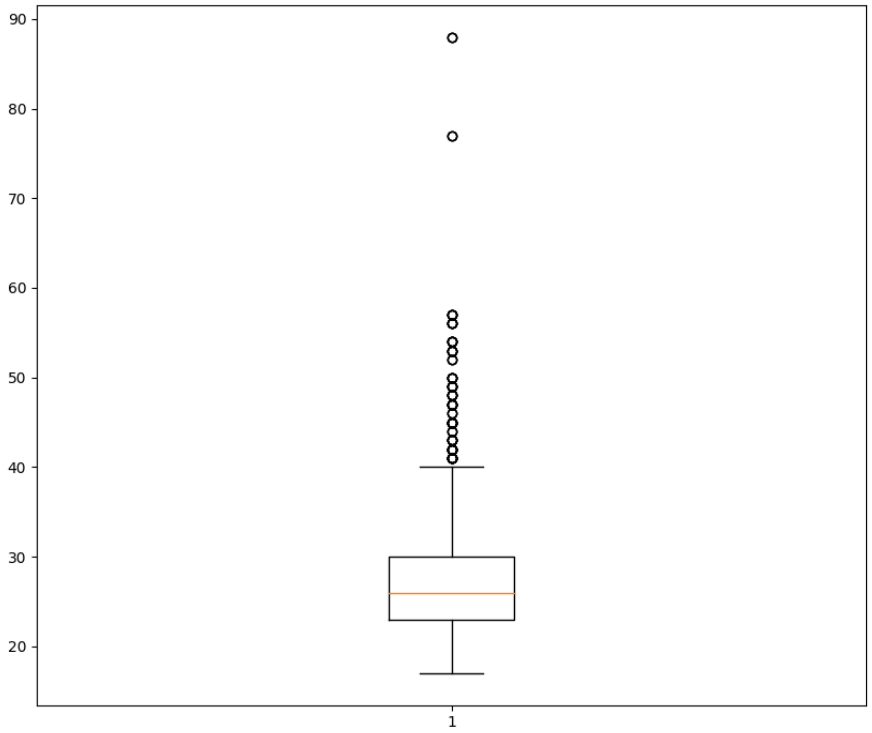
مورد دوم کارآمد تر است زیرا چیزی از دست نمی رود. اما مهم است که به جای داده هایی که نداریم چه مقداری قرار دهیم تا تحلیل آماری را تحت تاثیر قرار ندهد. به عنوان مثال می توان با مقادیر دلخواه، میانگین، مُد، میانه و... جایگزین کرد.

مرسوم ترین روش جایگزینی با میانگین است. البته درصورتی که داده های پرت داشته باشیم این روش مناسب نیست.

به کمک توابع آماده کتابخانه pandas میانگین داده های معتبر ستون های pace و dribbling را محاسبه می کنیم. با گشتن روی cell های مربوط به این دو ستون، هر کدام که خروجی isnull() برای آنها درست باشد، مقدار آن را با مقدار میانگین محاسبه شده در قسمت قبلی جایگزین می کنیم.

بخش ۲:

به کمک کتابخانه matplotlib و تابع boxplot داده های ستون age را به شکل نمودار جعبه ای رسم می کنیم که به شکل زیر است:



سایر داده ها برابر مقادیر زیر هستند:

```
min =  17
max =  88
Q1 =  23.0
Q2 =  26.682853299939705
Q3 =  30.0
```

چارک ها به کمک تابع percentile که برای کتابخانه numpy است قابل محاسبه هستند. از آنجایی که چارک دوم همان میانگین است میتوان به جای این تابع از تابع mean استفاده کرد. درصورتی که از تابع percentile استفاده کنیم و صدک ۵۰ام را محاسبه کنیم خروجی به شکل زیر خواهد بود:

```
min =  17
max =  88
Q1 =  23.0
Q2 =  26.0
Q3 =  30.0
```

چارک اول یا صدک ۲۵ام:

مشاهده ای است که یک چهارم داده ها از آن کوچکتر باشند. برای محاسبه آن، ابتدا میانه را محاسبه می کنیم سپس از داده های بین مینیمم و میانه دوباره میانه محاسبه می گیریم که آن را با Q1 نمایش می دهیم.

چارک دوم یا صدک ۵۰ام یا میانگین:

همان میانه داده هاست که با Q2 نمایش می دهیم. در واقع نیمی از داده ها از این مقدار کمتر هستند.

چارک سوم یا صدک ۷۵ام:

مشاهده ای است که سه چهارم داده ها از آن کوچکتر باشند. برای محاسبه، ابتدا میانه را پیدا می کنیم و میانه داده های بین میانه و ماکزیمم را پیدا می کنیم که با Q3 نمایش داده می شود.

بخش ۳:

آ) به صورت رندوم ۱۰۰ عدد از بین تعداد سطر های دیتافریم می گیریم و داده های این خانه ها را که مربوط به ستون وزن هستند در یک آرایه نامپای ذخیره می کنیم و به کمک توابع آماده نامپای واریانس و میانگین و انحراف معیار را محاسبه می کنیم و نتایج به شکل زیر خواهد بود:

```
variance =  48.84510000000001
standard deviation =  6.988926956264 46
mean =  77.93
```

ب) نمودار q-q نموداری از چارک های مجموعه داده اول در برابر چارک های مجموعه داده دوم است. منظور ما از یک چارک، کسر (یا درصد) نقاط زیر مقدار داده شده است.

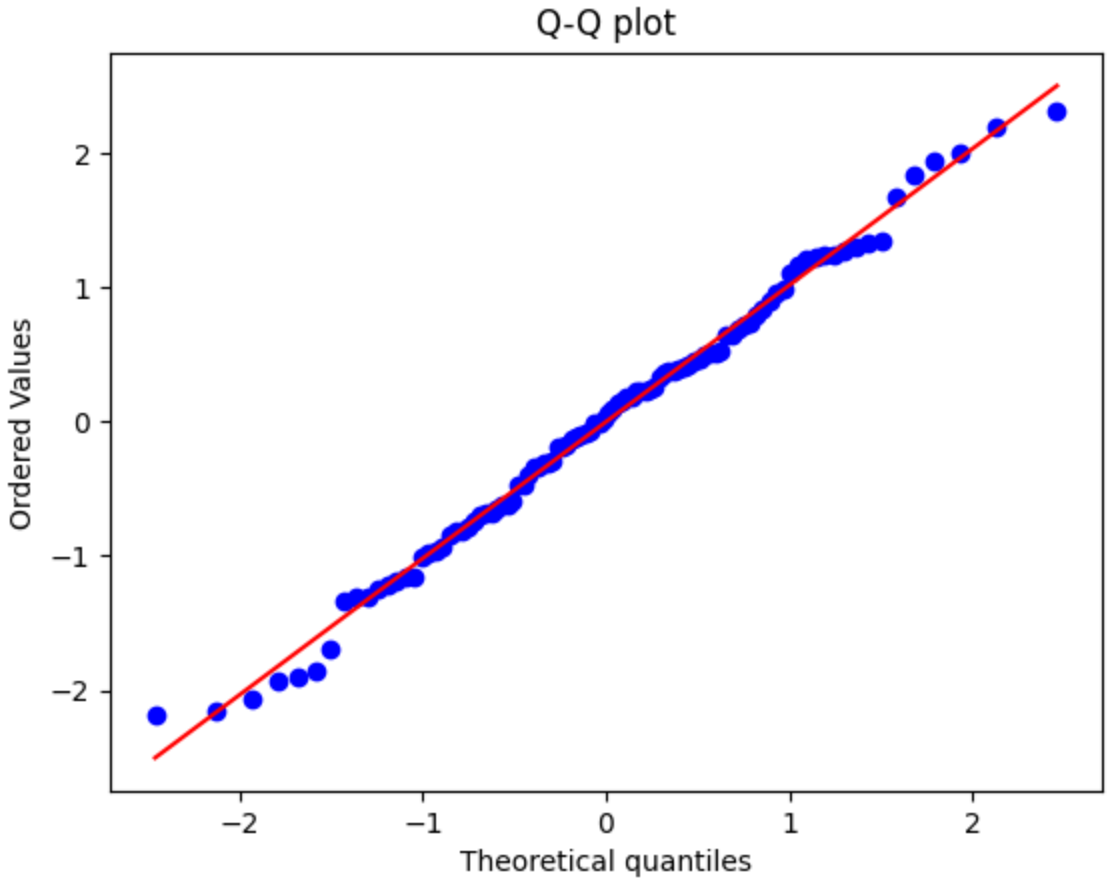
یک خط ۴۵٪ ترسیم می شود، اگر نمونه ها از یک جامعه باشند، نقاط در امتداد این خط قرار دارند.

ج)

ابتدا به کمک تابع random.normal و مقادیری که در قسمت آ محاسبه کردیم، ۱۰۰ نمونه از توزیع نرمال برآورد شده می گیریم. سپس مراحل زیر را طی می کنیم تا نمودار Q-Q را رسم کنیم:

ابتدا داده بر میداریم، سورت می کنیم، نمودار توزیع نرمال را می کشیم، z-value را محاسبه می کنیم، نمودار را رسم می کنیم.

نتیجه نمودار به شکل زیر خواهد شد:



با توجه به اینکه داده ها حول خط ۴۵ درصد هستند، می توان گفت که توزیع نرمال است.

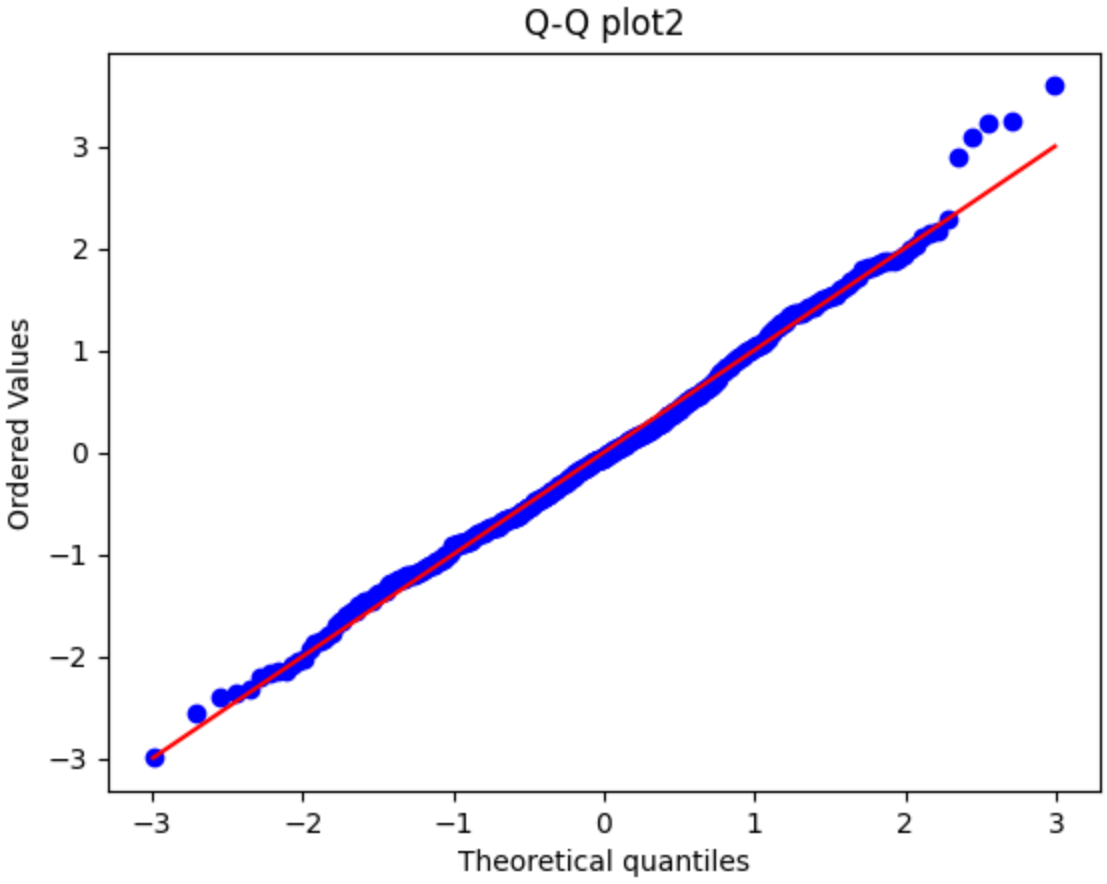
د)

پس از محاسبه p value و انجام آزمون مورد نظر با توجه به اینکه مقدار p value از ۰.۰۵ بیشتر شده می توان گفت که توزیع نرمال دارد.

```
p value =  0.05646196007728577
```

ه)

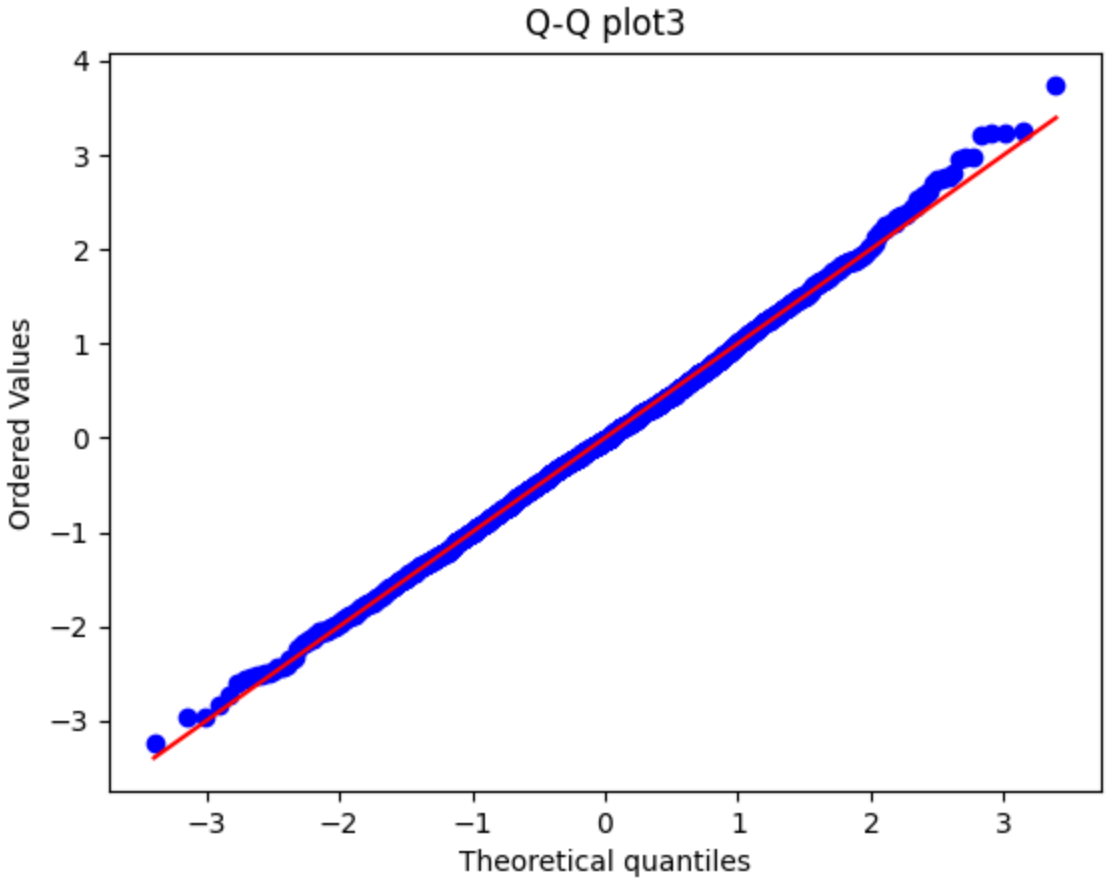
برای ۵۰۰ نمونه خروجی ها به فرم زیر خواهند بود:




```
variance2 =  55.815116
standard deviation2 =  7.470951478894773
mean2 =  77.378
p value =  0.022484665736556053
```

با توجه به نمودار می توان گفت نرمال است اما بر اساس آزمون چون p value از ۰.۰۵ کمتر است توزیع نرمال نیست.

برای ۲۰۰۰ نمونه داریم:



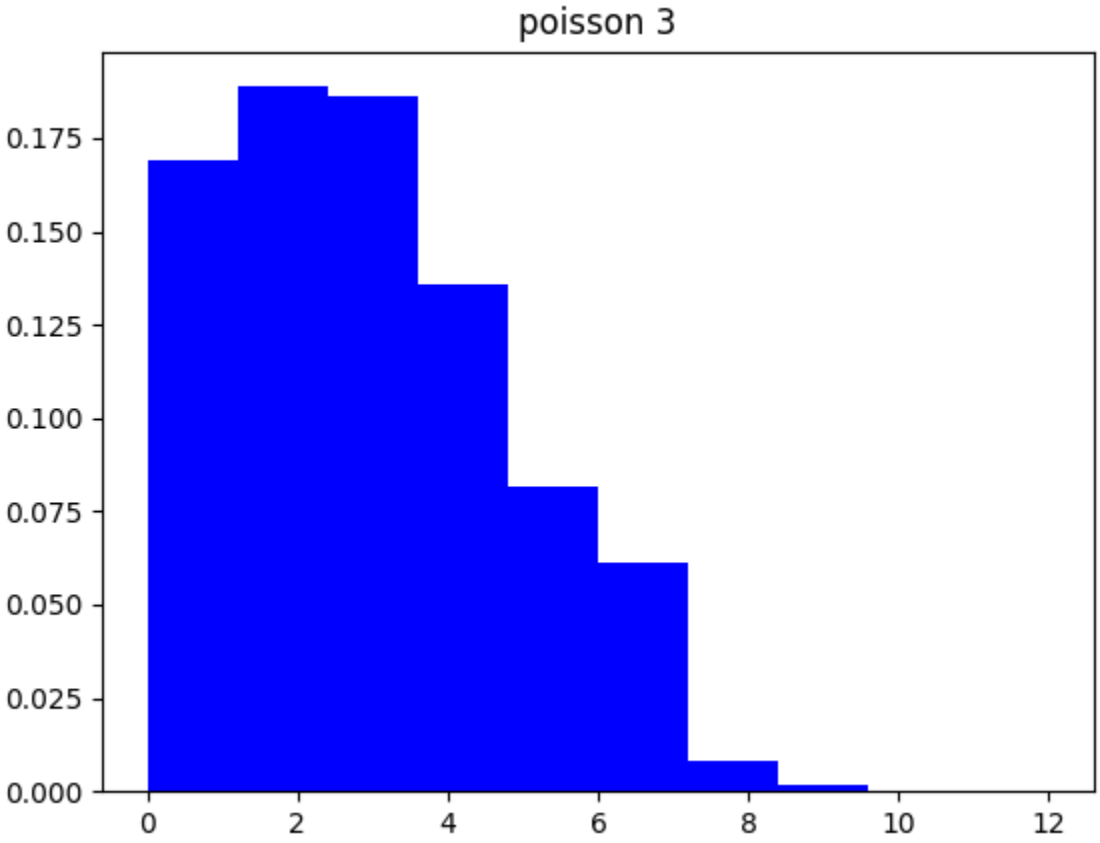

```
variance3 =  50.58416975000001
standard deviation3 =  7.112254899116032
mean3 =  76.6945
p value3 =  9.47383523453027e-06
```

باز هم نمودار تا حد خوبی منطبق است اما طبق آزمون نرمال نیست.

اگر توجه کنیم هرچه نمونه بیشتری برداریم بالای نمودار از خط ۴۵٪ دورتر می شود و مقدار p value هم در حال کاهش است بنابراین هر چه نمونه بیشتری برداریم دقت بیشتر خواهد بود.

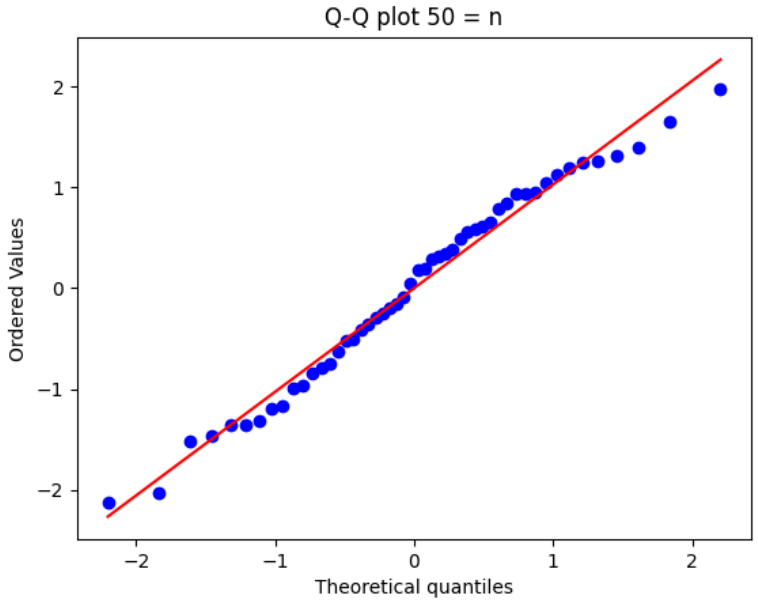
بخش ۴:

آ) به کمک کتابخانه های mstplotlib و numpy از توزیع پواسون نمونه می گیریم و هیستوگرام داده ها به فرم زیر خواهد بود:



ب)

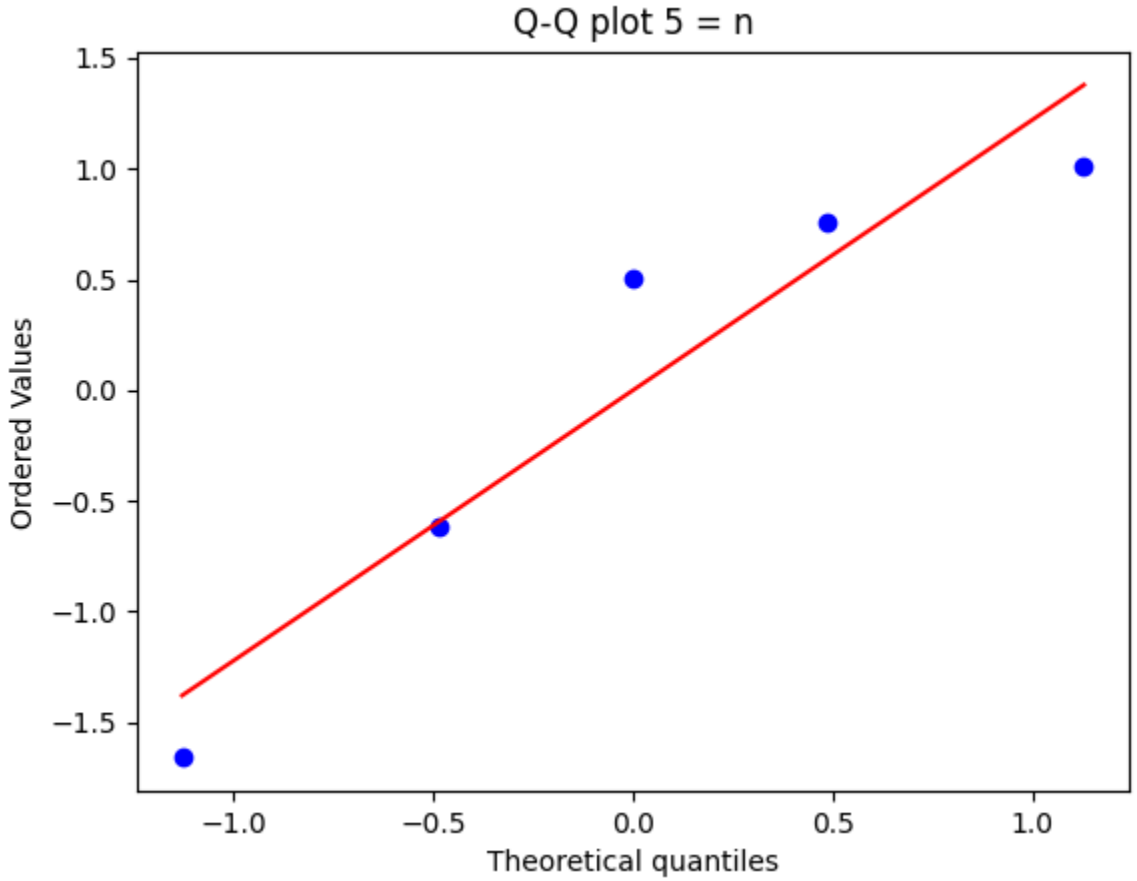
برای n = 50



```
standard deviation2 =  1.5844241856270687
mean2 =  2.64
p value2 =  0.024038864299654 96
```

هم با توجه به نمودار که از خط ۴۵٪ انحراف دارد و مقدار p value که از ۰.۰۵ کمتر است توزیع نرمال نیست.

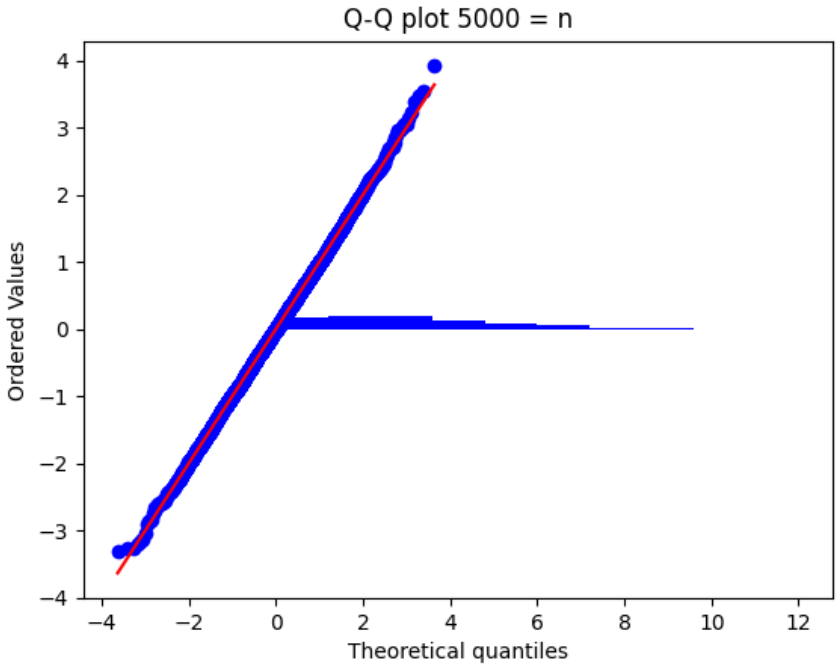
برای n = 5 داریم:




```
standard deviation3 =  2.3151673805580453
mean3 =  4.2
p value3 =  0.0413324311375618
```

با توجه به نمودار نمی توان گفت که توزیع نرمال است داده ها از نمودار خیلی دور هستند. بر اساس p value چون از ۰.۰۵ کمتر است پس توزیع نرمال نیست.

برای n = 5000 داریم:




```
standard deviation1 =  1.7406045386589108
mean1 =  2.9828
p value1 =  1.7393666233878043e-38
```

با توجه به نمودار میتوان گفت که توزیع نرمال است چون به خط ۴۵٪ نزدیک است اما با توجه به p value نمیتوان پذیرفت چون از ۰.۰۵ کمتر است.

طبق قضیه حد مرکزی برای تعداد نمونه های بزرگتر از ۳۰ باید توزیع به نرمال میل کند یا توجه به نمودار ها میتوان به این نتیجه رسید چون با بزرگتر شدن اندازه نمونه، داده ها به خط ۴۵٪ بیشتر نزدیک می شوند. در حالی که در آزمون شاپیرو، با بزرگتر شدن اندازه نمونه مقدار p value هم کمتر می شود.