بنام خدا

گزارش تمرین کامپیوتری ۴

آرشام لولوهری

۹۹۱۰۲۱۵۶

۱:

۱) تابع مذکور با استفاده از Cauchy.rvs ساخته شده است و به اندازه size نمونه میگیرد. Loc در تابع، همان x0 است و scale نیز همان گاما است. در انتهای CauchySampling ، میانگین آرایه نمونه ها به طول size را برمیگردانیم. برای تست کردن ها، مقادیر loc=100,scale=10,size=100 را انتخاب میکنیم. ابتدا یکبار خروجی این تابع را در test ریخته و چاپ میکنیم که در این تست مقدار حدودی ۱۱۰ برای میانگین نمونه ها بدست آمده است. این عدد، یک نمونه احتمالی برای میانگین توزیع ماست. برای بررسی دقیقتر، به تعداد N=1000 بار این عمل را انجام میدهیم (به اندازه size از توزیعمان نمونه میگیریم و میانگینشان را حساب میکنیم). بدین ترتیب در y، یک آرایه به طول N از میانگین های محاسبه شده داریم. با چاپ کردن واریانس y، به عدد بسیار بزرگی (در کد نوشته شده، در حدود ۷۰۰۰۰۰) میرسیم. این نشان میدهد که پراکندگی این ۱۰۰۰ تا میانگین، بسیار زیاد است و در نتیجه توزیع کوشی نمیتواند یک میانگین مشخص داشته باشد. در نتیجه قانون اعداد بزرگ برایش قابل استفاده نیست.

۲) به روش مشابه قبل، تابع ParetoSampling را میسازیم. توجه داریم در پایتون، b همان آلفا و scale همان xm است. Loc نیز شیفت مکانی است که در این مسئله نیازی نیست و برابر با صفر میگذاریم. برای تست، آلفا را 0.5 (حتما باید کمتر از یک باشد تا امید ریاضی نداشته باشیم) و xm را ۱ قرار میدهیم و تعداد نمونه ها در هر تست را ۱۰۰ میگیریم. ابتدا یک تست از تابع میگیریم و به میانگین حدود 19.1 میرسیم. سپس ۱۰۰۰ بار ParetoSampling را اجرا کرده و تمام میانگین های حاصله را در n میریزیم. واریانس n بسیار بزرگ و در تست ما در اوردر 10^18 است که نشان میدهد میانگین ها پراکندگی بسیار دارند، همگرا نیستند و در نتیجه قانون اعداد بزرگ قابل استفاده نیست.

۲:

۱) تابع مذکور با استفاده از random.binomial و نمونه گیری به اندازه size ساخته شده است. برای اطمینان از درستی کد نیز در y یک بردار نمونه به ازای p=0.5,n=20 ریخته ایم. واریانس این نمونه ها به واریانس توزیع دو جمله ای (که برابر با np(1-p)=5 است) و میانگینشان به امید ریاضی این توزیع(که برابر با np=10 است) همگرا میشود.

۲) در این روش، بردار کل نمونه های دو جمله ای را میگیریم و احتمال را برابر با نسبت تعداد مقادیری که در بازه مورد نظر هستند، به کل نمونه ها میگیریم. برای اینکار با استفاده از np.where، یکبار مشخص میکنیم که کدام اندیس های آرایه نمونه ها، از l بیشترند و یکبار مشخص میکنیم کدام اندیس ها از u کمترند. اشتراک این اندیس ها، اندیس نمونه هایی است که ویژگی موردنظر را دارند و آرایه ی اشتراک این دو را با intersect1d بدست می آوریم. طول این آرایه، تعداد نمونه هایی است که در بازه مذکور قرار دارند و نسبت پیدا میشود. در زیر آن، به ازای l=8,u=10 تابع را تست میکنیم و به مقدار حدودی 0.45 یا 0.46 میرسیم.

۳)در این روش، برای پیدا کردن $P(l \leq Y \leq u)$ که در آن $Y \sim Binomial(n,p)$ ، توجه داریم توزیع دو جمله ای، جمع n توزیع برنولی iid است. اگر n به اندازه کافی بزرگ باشد، طبق قضیه حد مرکزی، متغیر تصادفی

$$\frac{Y - n\mu}{\sigma\sqrt{n}}$$

به توزیع نرمال استاندارد همگرا میشود. در نتیجه میتوان نوشت:

$$P(l \le Y \le u) = P\left(\frac{l-n\mu}{\sigma\sqrt{n}} \le \frac{Y-n\mu}{\sigma\sqrt{n}} \le \frac{u-n\mu}{\sigma\sqrt{n}}\right)$$
$$\approx \Phi\left(\frac{u-n\mu}{\sigma\sqrt{n}}\right) - \Phi(\frac{l-n\mu}{\sigma\sqrt{n}})$$

اینکار در تابع EstProb انجام شده است. ابتدا امید و واریانس توزیع برنولی با پارامتر p محاسبه شده است. تابع فی در بالا را با استفاده از norm.cdf در کد استفاده کرده ایم و عین فرمول بالا را به عنوان خروجی تابع گرفته ایم. دوباره به ازای پارامتر های بخش های قبل، احتمال را محاسبه میکنیم و چاپ میکنیم.

۴) Continuity correction بیان میکند که در حالتی که متغیر تصادفی Y گسسته باشد، با فرض اینکه l,u صحیح باشند، برای احتمال مورد نظر میتوان نوشت:

$$P(l \le Y \le u) = P(l - 0.5 \le Y \le u + 0.5)$$

بدلیل گسسته بودن مقادیر Y، طرف راست تساوی با طرف چپ برابر است اما در برخی موارد و به ویژه زمانی که Y توزیع دوجمله ای دارد، در استفاده از قضیه حد مرکزی، طرف راست تساوی به مقدار دقیقتری منجر میشود. یعنی کافیست مشابه روش قبل عمل کنیم و خواهیم داشت:

$$P(l - 0.5 \le Y \le u + 0.5)$$
$$\approx \Phi\left(\frac{u+0.5-n\mu}{\sigma\sqrt{n}}\right) - \Phi(\frac{l-0.5-n\mu}{\sigma\sqrt{n}})$$

بدین ترتیب در کد نیز تنها بخش return اندکی متفاوت میشود. دوباره به ازای پارامترهای قبلی، احتمال را با این تابع محاسبه و چاپ میکنیم.

۵) تمام مقادیر به ازای پارامترهای مذکور در بخش های قبل چاپ شده اند و مجموعه نتایج به صورت زیر است:

```
samples mean= 9.9612
samples Var= 5.0684945599999995
P(8<=Y<=10)=  0.4632            (using samples)
P(8<=Y<=10)=  0.3144533152386512                (using CLT)
P(8<=Y<=10)=  0.45669212447945295               (using continue correction)
```

حال مقدار دقیق را نیز توسط تعریف توزیع دو جمله ای حساب میکنیم:

$$P(8 \le Y \le 10) = \sum_{k=8}^{10} \binom{n}{k} p^k (1-p)^{n-k}$$

$$= \left[\binom{20}{8} + \binom{20}{9} + \binom{20}{10}\right] \left(\frac{1}{2}\right)^{20}$$

$$= 0.4565$$

میبینیم بیشترین دقت را در روش continuity correction با قضیه حد مرکزی داریم که همانطور که ذکر شد، دقت بسیار بیشتری نسبت به حد مرکزی عادی دارد. پس از آن نیز روش اول و با استفاده از نمونه های گرفته شده، دقت بیشتری دارد.

۳:

۱) کد همانطور که ذکر شده، نوشته شده است. ابتدا با استفاده از پکیج pandas و read_csv، فایل اکسل با نام heart.csv که در محل فایل پایتون قرار دارد، خوانده میشود. دستور head پنج سطر اول فایل را نمایش میدهد و دستور info برخی اطلاعات را (شامل ستون ها و تعدادشان، جنس داده های هر ستون یا همان datatype ها، میزان مصرف مموری و...) نمایش میدهد:

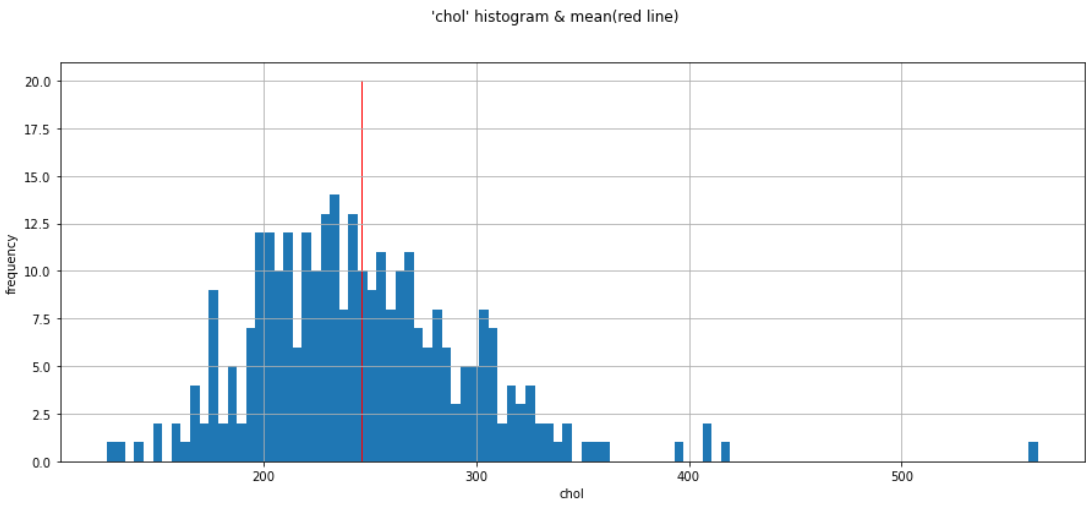
```
   age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  \
0   63    1   3       145   233    1        0      150      0      2.3      0
1   37    1   2       130   250    0        1      187      0      3.5      0
2   41    0   1       130   204    0        0      172      0      1.4      2
3   56    1   1       120   236    0        1      178      0      0.8      2
4   57    0   0       120   354    0        1      163      1      0.6      2

   ca  thal  target
0   0     1       1
1   0     2       1
2   0     2       1
3   0     2       1
4   0     2       1
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
None
```

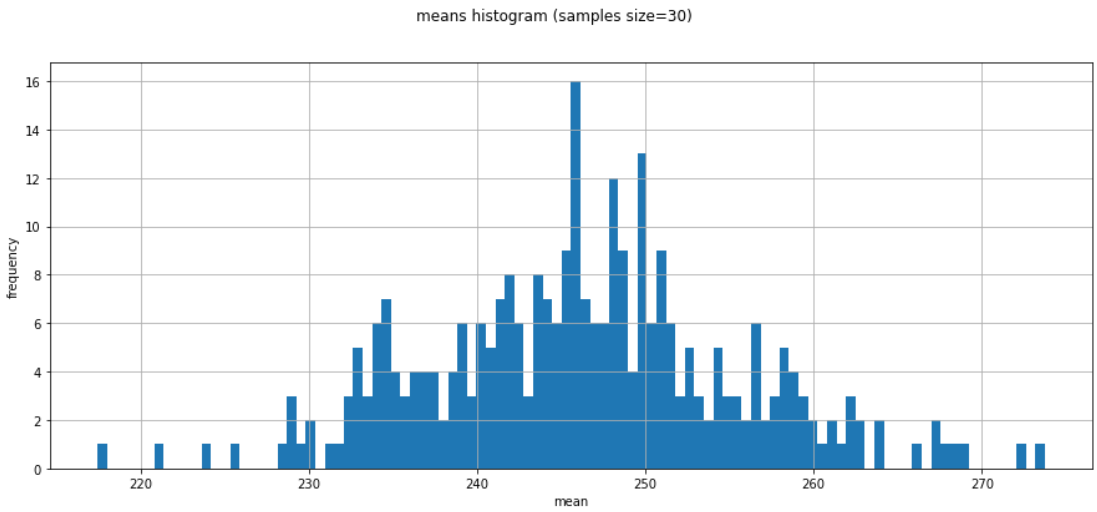
۲)این کد در دومین cell نوشته شده است. در خط اول، ستون مربوط به تیتر ’chol‘ که همان کلسترول است، جدا شده و در df_chol ریخته شده است. سپس با دستور describe، اطلاعات زیر نمایش داده میشود (شامل تعداد داده ها، میانگین، مینیمم و ماکزیمم، میانه و چارک ها یا percentile ها):

```
count    303.000000
mean     246.264026
std       51.830751
min      126.000000
25%      211.000000
50%      240.000000
75%      274.500000
max      564.000000
Name: chol, dtype: float64
```

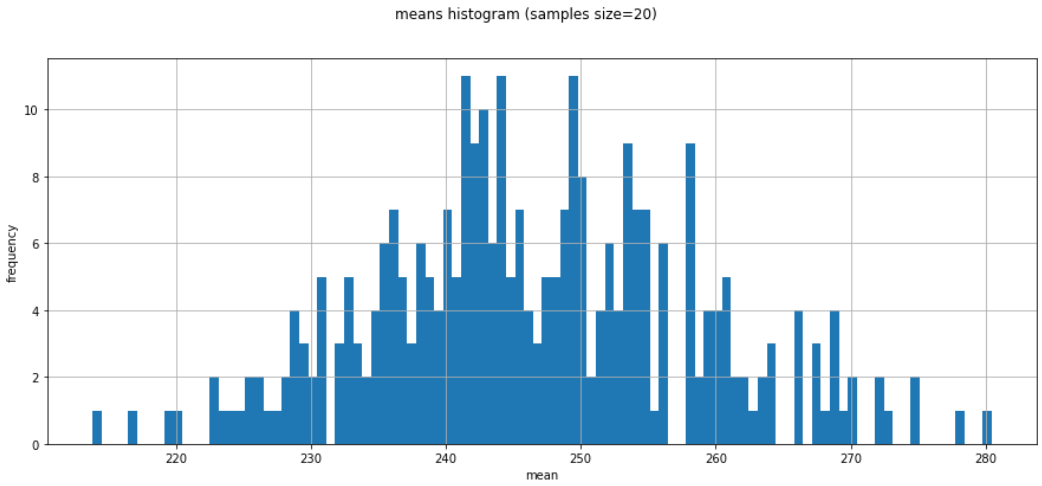
۳)در سومین cell اینکار انجام شده است. ابتدا figure با نام و سایز مشخص ساخته شده، و سپس با hist، نمودار هیستوگرام رسم شده است. تعداد دسته ها برابر ۱۰۰ تنظیم شده و محدوده ی رسم نمودار نیز از کوچکترین تا بزرگترین داده در ستون کلسترول قرار داده شده است. سپس لیبل های محورها و خط کشی ها تنظیم شده اند و در نهایت با bar، روی میانگین داده ها، یک خط قرمز رنگ رسم شده است. این میانگین حدود 246.26 است. نمودار رسم شده در زیر، تا حدی به صورت نرمال و حول میانگین ۲۴۶ توزیع شده است:

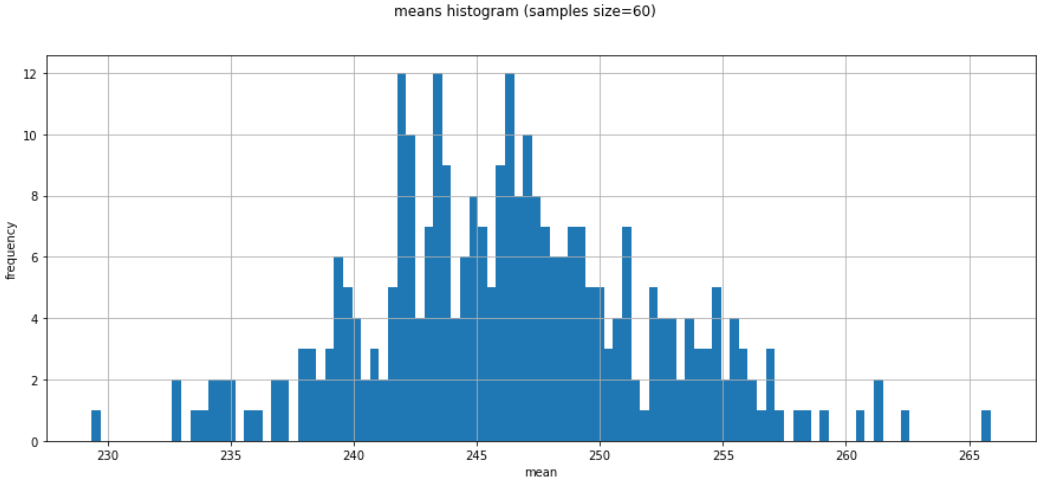


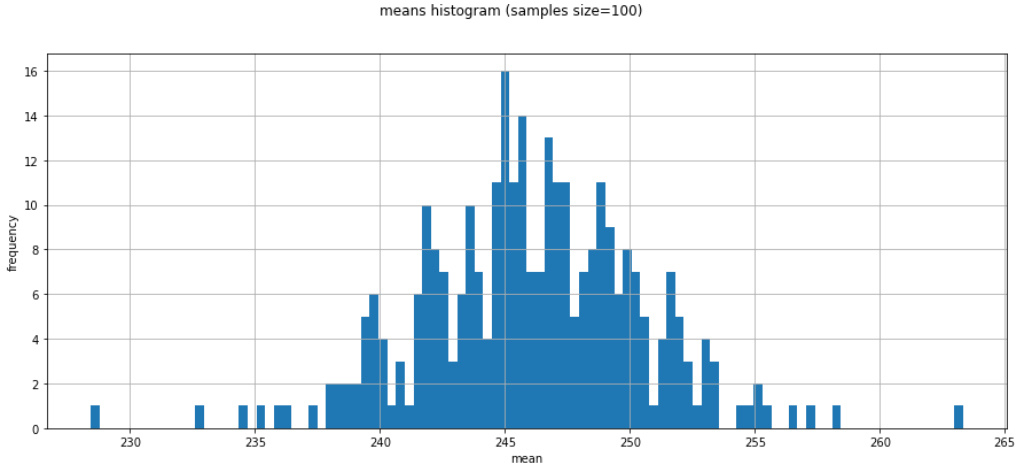
۴)در cell بعدی این کار انجام شده است. ابتدا یک تابع SamplesMean تعریف شده که با گرفتن کل داده ها، به تعداد size از آن آرایه را با استفاده از sample و به صورت رندوم نمونه برداری میکند و میانگین آنها را برمیگرداند. این تابع باید N=300 بار اجرا شود و داده های بدست آمده در means ریخته میشود. سپس مشابه قبل هیستوگرام را برای means رسم میکنیم. تعداد دسته ها را همان ۱۰۰ میگیریم. نمودار حاصله به صورت زیر است:



۵) در سه تا cell آخر، اینکار انجام شده. نمودار های حاصله به صورت زیر اند که به ترتیب برای سایز ۲۰ و ۶۰ و ۱۰۰ هستند (در کد، صرفا size به ۲۰ و ۶۰ و ۱۰۰ تغییر میکند):


means histogram (samples size=20)
frequency
mean
0
2
4
6
8
10
220
230
240
250
260
270
280


means histogram (samples size=60)
frequency
mean
0
2
4
6
8
10
12
230
235
240
245
250
255
260
265


means histogram (samples size=100)
frequency
mean
0
2
4
6
8
10
12
14
16
230
235
240
245
250
255
260
265

مشاهده میشود تمامی نمودارها تا حدی به یک توزیع نرمال حول میانگین در حدود ۲۴۶ شبیه هستند اما هرچه سایز نمونه ها را بزرگتر میکنیم دقت بیشتری داریم چرا که میانگین ها در سایز های بزرگتر، مقادیر منطقی تری خواهند بود. مثلا در سایز نمونه های ۲۰، نوسانات بیشتری نسبت به ۱۰۰ (که بیشترین شباهت به نرمال را دارد) داراست.