به نام خدا

دانشگاه صنعتی شریف

دانشکده مهندسی برق

گروه دکتر کرباسی ـ آمار و احتمال مهندسی
نیم سال دوم ۱۴۰۱ ـ ۱۴۰۰

تمرین عملی سری چهارم

لطفاً به نکات زیر توجه بفرمایید: (رعایت نکردن این موارد باعث کاهش نمره می‌شود.)

۱. نتایج و پاسخ‌های خود را در یک فایل با فرمت zip به نام HW4_StudentID_Name در سایت Quera قرار دهید. همچنین فایل پایتون خود را به همان نام در قسمت مخصوص به خود آپلود کنید.

۲. کسب نمره کامل در هر سؤال مستلزم تحویل **کدها** و **توضیحات** می‌باشد.

۳. برای سؤالات، باید روشی که استفاده کرده‌اید را توضیح و نتایجی که گرفته‌اید را ارائه دهید. این توضیحات می‌تواند در یک فایل pdf. و یا در یک فایل ipynb. باشد.

۴. فایل‌های تحویلی شما دو بخش میباشند، یک بخش فایل zip. که شامل فایل ipynb. کد و گزارش شما میباشد، یک بخش هم کدهای هر سوال به شکل جداگانه میباشند که باید در فرمت py. در سامانه کوئرا در کنار فایل zip. آپلود شوند.(برای مثال اگر تمرین شامل ۳ سوال بود، باید علاوه بر فایل zip. که تحویل مصحح میشود، ۳ فایل py. در سامانه کوئرا در محل بارگذاری مشخص شده آپلود کنید.)

۵. کدهای خود را خوانا بنویسید و کامنت‌گذاری کنید. در plot های خود عنوان، label و خط‌کشی‌های مناسب را اضافه کنید.

۶. در طول ترم امکان ارسال با تاخیر پاسخ همه‌ی تمارین تا سقف پنج روز و در مجموع دوازده روز وجود دارد. پس از گذشت این مدت، پاسخ‌های ارسال‌شده پذیرفته نخواهند بود. همچنین، به ازای هر روز تأخیر غیر مجاز بیست درصد از نمره تمرین به صورت ساعتی کسر خواهد شد.

۷. کدهای شما تماماً باید توسط خودتان نوشته شده باشند. هرگونه استفاده از کد دیگران به هر شکل ممکن، تقلب محسوب می‌شود و نمره تمرین کامپیوتری جاری صفر خواهد شد. پس در هیچ صورت کدهای خود را برای دیگران ارسال نکنید.

۸. ابهام یا اشکالات خود را مي توانید از طریق smmzdr@gmail.com یا javadiamirhosein.2000@gmail.com مطرح نمایید.

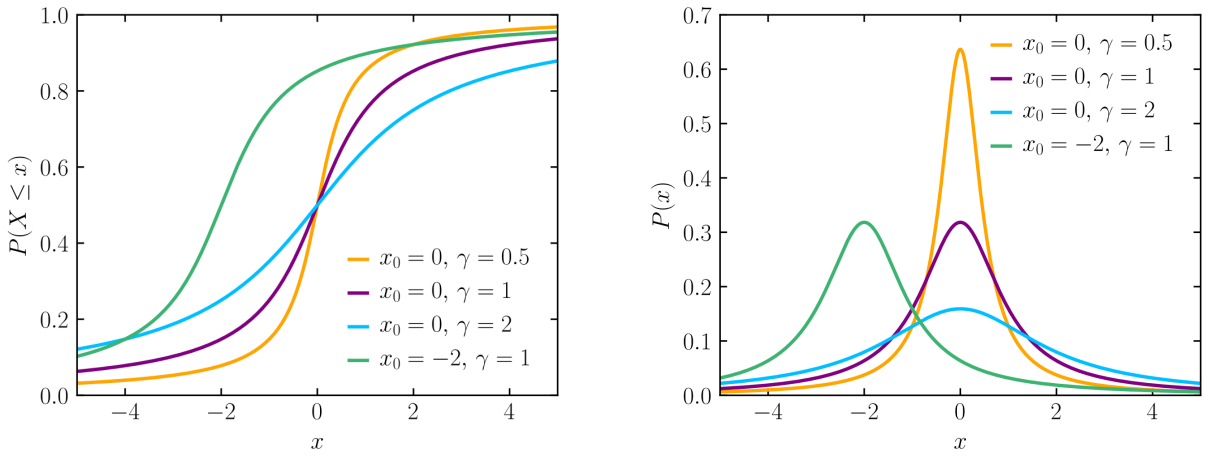
۹. مهلت تحویل: **نیمه شب ۳۱ خرداد**

---

۱ قانون اعداد بزرگ

موقعیت‌های خاصی وجود دارد که قانون اعداد بزرگ نمی‌توانند با افزایش تعداد نمونه یا تعداد آزمایش‌ها، روی مقدار مورد انتظار همگرا شوند. وقتی داده‌ها از توزیع کوشی پیروی می‌کنند، مجموعه‌ی اعداد نمی‌توانند به امید ریاضی این توزیع همگرا شوند زیرا توزیع کوشی امید ریاضی ندارد.

$$f(x, x_0, \gamma) = \frac{1}{\pi\gamma\left[1 + \left(\frac{x-x_0}{\gamma}\right)^2\right]}$$

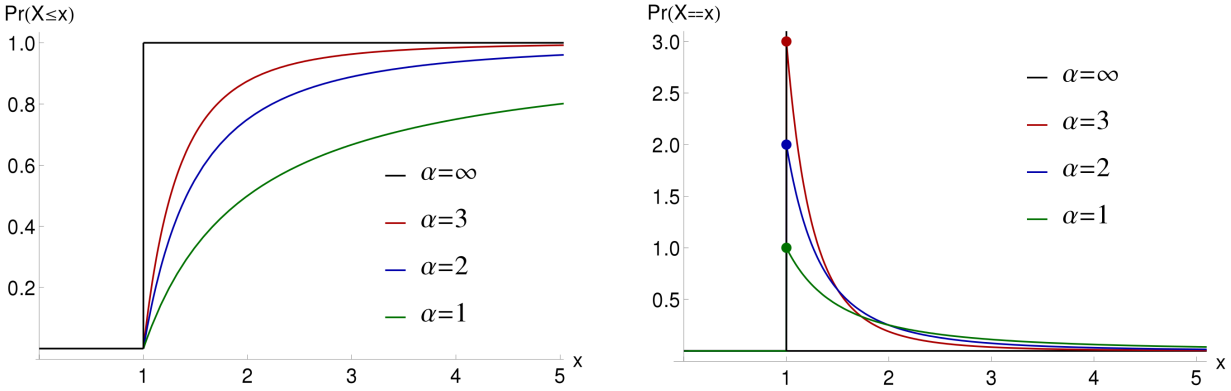


Cumulative distribution function (ب) Probability density function (الف)

شکل ۱: Cauchy distribution

به طور مشابه، قانون اعداد بزرگ برای توزیع پارتو هم کار نمی‌دهد زیرا امید ریاضی آن وقتی $\alpha \leq 1$ است نامحدود است. تابع توزیع تجمعی پارتو با پارامترهای $(x_m > 0, \alpha > 0)$ به شکل زیر است:

$$F_X(x) = \mathbb{P}(X < x) = \begin{cases} 1 - \left(\frac{x}{x_m}\right)^\alpha & x \geq x_m \\ 0 & x < x_m \end{cases}$$



Cumulative distribution function (ب) Probability density function (الف)

شکل ۲: Pareto Type I

حال میخواهیم با استفاده از نمونه برداری این ادعا را بسنجیم.

۱. تابعی به نام CauchySampling تعریف کنید که پارامترهای (x_0, γ, size) را بگیرد و از توزیع Cauchy(x_0, γ) به تعداد size نمونه برداری کند و میانگین این نمونه‌ها را به عنوان خروجی تابع برگرداند. به ازای مقادیر دلخواه ورودی مثل (x_0= 100, γ= 10, size = 100) خروجی تابع را مشاهده کنید.
برای بررسی واگرایی خروجی، این تابع را N = 1000 بار اجرا کنید و واریانس این خروجی‌ها را به دست آوردید.

۲. تابعی به نام ParetoSampling تعریف کنید که پارامترهای (x_m, α, size) را بگیرد و از توزیع Pareto(x_m, α) به تعداد size نمونه برداری کند و میانگین این نمونه‌ها را به عنوان خروجی تابع برگرداند. به ازای مقادیر دلخواه ورودی مثل (x_m= 1, α= 0.5, size = 100) خروجی تابع را مشاهده کنید. برای بررسی واگرایی خروجی، این تابع را N = 1000 بار اجرا کنید و واریانس این خروجی‌ها را به دست آوردید.

توابع پیشنهادی: cauchy و pareto از کتاب‌خانه‌ی scipy.stats

۲ قضیه‌ی حد مرکزی

قضیه حد مرکزی در نظریه احتمالات بیان می‌کند که در بیشتر مواقع، مجموع نرمالایز شده‌ی تعدادی متغیر تصادفی مستقل، که هر یک میانگین و واریانس به خوبی تعریف شده دارند، به‌طور تقریبی دارای توزیع نرمال خواهد بود. هرچه تعداد این متغیرهای مستقل افزایش یابد، این تقریب بهتر می‌شود. در این سوال میخواهیم این مسئله را به روش شبیه سازی هم بررسی کنیم.

۱. تابعی به نام SampleBinomial تعریف کنید که پارامترهای (p, n, size) را بگیرد و به تعداد خواسته‌شده (size) نمونه از توزیع Binomial(p, n) تولید کند و این نمونه‌ها را به عنوان خروجی تابع مشخص کنید.
برای مطمئن شدن از کارایی درست تابعتان میانگین و واریانس نمونه‌های تولید شده را برای ورودی دلخواه مثل (p=0.5, n=20, size=10000) با مقدار تئوری این مقادیر مقایسه کنید.

۲. تابعی با نام FindProb تعریف کنید که پارامترهای (samples, l, u) را بگیرد و نسبت داده‌هایی که در بازه‌ی [l, u] هستند را به کل داده به دست بیاورد و به عنوان خروجی برگرداند.

۳. تابعی به نام EstProb تعیین کنید که پارامترهای (p, n, l, u) را بگیرد و احتمال خواسته‌شده را با استفاده از قضیه حد مرکزی تخمین بزند.

۴. برای تخمین این احتمال میتوان از ورژن continue correction این تخمین‌گر استفاده کنیم. تابعی به نام CorEstProb تعریف کنید که پارامترهای (p, n, l, u) را بگیرد و احتمال خواسته‌شده را تخمین بزند.

۵. توابع گفته‌شده را برای رودی دلخواه مثل (p=0.5, n=20, size=10000, l=8, u=10) حساب کنید. کدام یک از تخمین‌ها به احتمال تجربی نزدیک‌تر بود؟ نتایج خود را تحلیل کنید.

تمام قسمت‌های قبل را بدون استفاده از حلقه‌های لوپ مثل for و به شکل ماتریسی حل کنید. همچنین توابع شما باید به ازای همه ورودی‌های ممکن درست کار بدهند.
توابع پیشنهادی: np.random.binomial از کتاب‌خانه‌ی numpy و norm.cdf از کتاب‌خانه‌ی scipy.stats

۳ دیتاست بیماری قلب

در این تمرین با کمک دیتاست مرتبط با بیماری‌های قلبی، تلاش می‌کنیم که با این مفهوم بهتر آشنا شویم.

(لینک دانلود دیتاست)

۱. دیتاست مربوطه را بخوانید. در این دیتاست، هر سطر نشان‌دهنده‌ی ویژگی‌های یک فرد می‌باشد. با کمک دستور

()head. پنج سطر اول آن را مشاهده کنید. به کمک دستور ()info. نیز میتوان بعضی ویژگی‌های دیتاست را مشاهده کرد. خروجی این دو دستور را چاپ کنید و در گزارش کار بیاورید.

```
import pandas as pd
df = pd.read_csv("heart.csv")
print(df.head())
print(df.info())
```

۲. ستون chol نمایانگر مقدار کلسترول در خون فرد می‌باشد. با دستور ()describe. میتوانید ویژگی‌های آماری این ستون را مشاهده کنید. خروجی این دستور را چاپ کنید و در گزارش کار بیاورید.

```
df_chol = df['chol']
print(df_chol.describe())
```

۳. نمودار هیستوگرام را برای این ویژگی رسم کنید. (تعداد bin ها را ۱۰۰ قرار دهید) میانگین کل داده‌ها را با یک خط قرمز عمودی روی نمودار مشخص کنید. توزیعی که نمایش دادید نرمال می‌باشد؟

۴. حال از تکنیک جدیدی برای محاسبه‌ی میانگین استفاده می‌کنیم. به این صورت که در هر تلاش، تعدادی نمونه از کل نمونه‌ها را انتخاب میکنیم و میانگین آن‌ها را ذخیره می‌کنیم. سپس نمودار هیستوگرام را برای این مقادیر (میانگین‌ها) رسم می‌کنیم. توجه کنید که سمپل‌هایی که از داده‌ی اصلی برداشته می‌شوند تا میانگینشان محاسبه شود، به صورت تصادفی انتخاب می‌شوند. تعداد نمونه‌هایی از داده‌ی اصلی که هر بار انتخاب می‌کنیم تا به کمک آن‌ها میانگین را حساب کنیم، ۳۰ قرار دهید. (حدودا ۱۰ درصد کل داده‌ها) به این ترتیب ۳۰۰ بار این کار را انجام دهید و میانگین‌های به دست آمده را در یک لیست ذخیره کنید.(در این مرحله یک لیست ۳۰۰ تایی از میانگین‌ها خواهید داشت که هر خانه ی آن با میانگین گیری بین ۳۰ داده که به طور رندم از مجموعه‌ی کل داده‌ها انتخاب شده‌اند، به دست آمده‌است.) نمودار هیستوگرام را برای این ۳۰۰ مقدار رسم کنید.

۵. این کار را با نمونه‌های با اندازه‌ی ۲۰ و ۶۰ و ۱۰۰ نیز انجام دهید و نمودار هیستوگرام را برای هرکدام از آن‌ها رسم کنید. کدام یک از نمودارها بیشترین شباهت را به توزیع نرمال پیدا کرده است؟