CSCI3022 S22

Homework 4: Working with RVs

Due Friday, March 4 at 11:59 pm to Canvas and Gradescope

Name: Matthew Su

Your solutions to computational questions should include any specified Python code and results as well as written commentary on your conclusions. Remember that you are encouraged to discuss the problems with your classmates, but **you must write all code and solutions on your own**.

NOTES:

- Any relevant data sets should be available on Canvas. To make life easier on the graders if they need to run your code, do not change the relative path names here. Instead, move the files around on your computer.
- If you're not familiar with typesetting math directly into Markdown then by all means, do your work on paper first and then typeset it later. Here is a reference guide linked on Canvas on writing math in Markdown. **All** of your written commentary, justifications and mathematical work should be in Markdown. I also recommend the wikibook for LaTex.
- Because you can technically evaluate notebook cells is a non-linear order, it's a good idea to do **Kernel** $\rightarrow$ **Restart & Run All** as a check before submitting your solutions. That way if we need to run your code you will know that it will work as expected.
- It is **bad form** to make your reader interpret numerical output from your code. If a question asks you to compute some value from the data you should show your code output **AND** write a summary of the results in Markdown directly below your code.
- 45 points of this assignment are in problems. The remaining 5 are for neatness, style, and overall exposition of both code and text.
- This probably goes without saying, but... For any question that asks you to calculate something, you **must show all work and justify your answers to receive credit**. Sparse or nonexistent work will receive sparse or nonexistent credit.
- There is *not a prescribed API* for these problems. You may answer coding questions with whatever syntax or object typing you deem fit. Your evaluation will primarily live in the clarity of how well you present your final results, so don't skip over any interpretations! Your code should still be commented and readable to ensure you followed the given course algorithm.
- There are two ways to quickly make a .pdf out of this notebook for Gradescope submission. Either:

- Use File -> Download as PDF via LaTeX. This will require your system path find a working install of a TeX compiler
- Easier: Use File -> Print Preview, and then Right-Click -> Print using your default browser and "Print to PDF"

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pylab as plt
import scipy.stats as stats
%matplotlib inline
```

Shortcuts: Problem 1 | Problem 2 |

Back to top

(25 pts) Problem 1 (Theory): Working with Distributions

When you made a histogram on homework 2 of the precipitation column for Boulder, you might have noticed that one particular point was a **massive** outlier. Specifically, the 2013 flood where pretty devastating to the community. One takeaway a data analyst might derive from the data could be the statement that "Boulder has storm that generates over 12 inches of rain **once per 60 years**." Let's explore what that probability statement might imply!

Part A: Suppose that attempt to *count* occurrences of "massive rainfall" with a Poisson process. For a *fixed* interval of 60 years, we would assert that the rate λ of massive rainfalls is $\lambda = 1 \frac{heavy\ rain}{60\ yr\ interval.}$

Using `stats.poisson` and the related `scipy.stats`, functions, print a list of the probabilities of observing exactly k such storms in a 60-year interval for $k = 0, 1, 2, \ldots 10$.

In [2]:
```
for i in range(11):
    print(stats.poisson.pmf(k = i, mu = 1))
```

```
0.36787944117144233
0.36787944117144233
0.18393972058572114
0.06131324019524039
0.015328310048810101
0.0030656620097620 2
0.0005109436682936698
7.299195261338139e-05
9.123994076672672e-06
1.013777119630298e-06
1.0137771196302987e-07
```

Part B: We could instead consider describing the process with the model that "each year is independently and identically likely to *contain* a heavy storm with probability $1/60$." This would

suggest a different random variable to count the number of heavy storms in a 60 year period. Which one?

Using the appropriate random variable and its implementation in `scipy.stats`, again print a list of the probabilities of observing exactly k years with a storm in a 60-year interval (again, do all of $k = 0, 1, 2, \ldots 10$).

In [3]:
```
# for i in range(11):
#     print(stats.poisson.pm(k=i, mu = 1-stats.geom.cdf(60,1/60)))

for i in range(11):
    print(stats.binom.pmf(k=i, n = 60, p = 1/60))
```

```
0.36479231075334484
0.37097523127460547
0.18548761563729835
0.06078125258171482
0.014680217784566326
0.002786753206561736
0.0004329701309629879 3
5.66111067118668e-05
6.356755626544394e-06
6.225071423357996e-07
5.380993942224899e-08
```

Part C: Are your results in Part A and Part B the same? Why or why not? What fundamental assumptions are different in using a Poisson model to describe this process instead of the model in Part B?

The results from Part A and Part B are remarkably similar up until k = 7, where then they start to devulge. The key reason why the results overall are different is because a poisson model is deployed for an infinite sample set whereas the binomial deployment is for a finite set. Therefore the probability that is calculated for the poisson model reflects an infinte scale wereas the binomal model is relative to its finite limits.

Part D: We could also use the *Geometric* random variable to track an event like "what is the probability that we *don't* observe a single heavy storm in the 60 year interval." Would this calculation agree with the corresponding probability (observing $k = 0$ storms) from the models in Parts A and B? Why or why not? You may use code to justify your answer.

Our resulting answer be closer to the binom result from part B since it has a fixed number, with binom being a fixed number of trials done finding the probabilty of a number of sucesses and geom being a fixed number of sucesses counting the number of trials required to reach that count of sucessess.. The way the code below sets up results in a similar result to the binom as it is essentially a conversion between the fixed natures of the two.

In [36]:
```
1 - stats.geom.cdf(k = 60, p = 1/60)
```

Out[36]:
```
0.36479231075334484
```

Back to top

(25 pts) Problem 2 (Computation): Working with Continuous Distributions

---

Suppose you are assigned to study a new found species of rodent, and discover that they have variable tail lengths. A colleague comes back to you and proposes that the tail length of an individual rodent is given by the random variable X with pdf of the form:

$$f(x) = \begin{cases} \frac{a}{\sqrt{x}} & 1 \leq x < 4 \\ 0 & \text{otherwise} \end{cases}$$

where a is some normalizing constant.

Part A: Determine the value of a such that $f(x)$ is a valid probability density function. Use that value for the rest of the problem.

$$1 = \int_1^4 \frac{a}{\sqrt{x}} dx$$

$$1 = a * \int_1^4 x^{-\frac{1}{2}} dx$$

$$1 = a * \frac{x^{-\frac{1}{2}+1}}{-\frac{1}{2}+1} = 2a * (\sqrt{4} - \sqrt{1})$$

$$a = 0.5$$

Part B: Compute *by hand* the cumulative density (cdf) function $F(x)$ for X.

$$\int_1^x \frac{0.5}{\sqrt{t}} dx$$

$$0.5 * \int_1^x t^{-\frac{1}{2}} dx$$

$$0.5 * [\frac{t^{-\frac{1}{2}+1}}{-\frac{1}{2}+1}]\Big|_1^x = (\sqrt{x} - 1) = cdf$$

Part C: Use the cdf you derived in **Part B** to calculate the median ($\tilde{x}$) tail length for a rodent.

In order to calculate median, set cdf to equal 0.5. Since we have already simplified down the integral, start from there:

$$\sqrt{x} - 1 = 0.5$$

$$x = 1.5^2 = 2.25$$

Part D: Compute by hand both the mean $E[X]$ and variance $E[(X - E[X])^2]$ of the tail length distribution.

Mean:

$$\int_1^4 \frac{0.5x}{\sqrt{x}} dx$$

$$0.5 * \int_1^4 \sqrt{x} dx$$

$$0.5 * [\frac{2}{3} x^{\frac{3}{2}}]\Big|_1^4 = \frac{16}{3} - \frac{2}{3} * \frac{1}{2} = \frac{7}{3}$$

Variance:

$$\int_1^4 \frac{0.5x^2}{\sqrt{x}} dx$$

$$0.5 * \int_1^4 \frac{2}{5} x^{\frac{5}{2}} dx$$

$$0.5 * [\frac{2}{5} x^{\frac{5}{2}}]\Big|_1^4 = \frac{64}{5} - \frac{2}{5} * \frac{1}{2} = \frac{31}{5}$$

$$\frac{31}{5} - \frac{49}{9} = \frac{34}{45}$$

Part E: Create a plot of the pdf f of X. Clearly mark (via vertical lines) where both $E[X]$ and the median of X are on the function. Which is larger? Could you have known that directly from the plot before completing parts **B-D**?

In [20]:

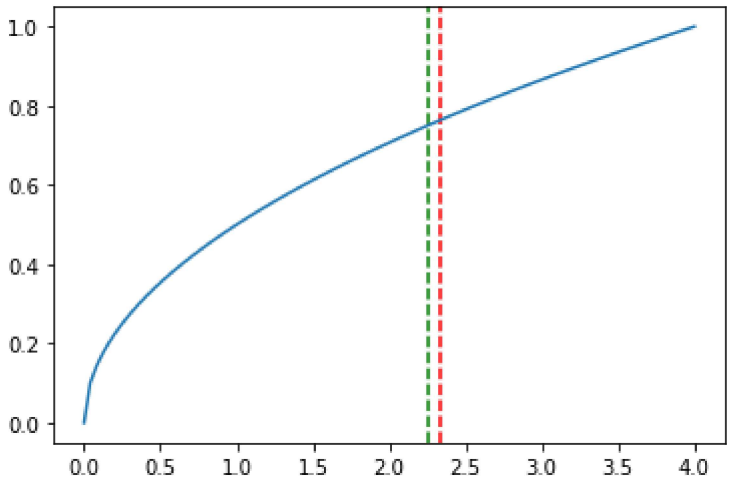
```
from matplotlib.pyplot import plot
import math
def f(x):
    if x >=0 and x <=4:
        return 1/2 * math.sqrt(x)
    else:
        return 0

xvalues = np.linspace(0,4,100)
yvalues = list(map(f,xvalues))
# Median = green
plt.axvline(x = 2.25, color = 'green', linestyle='--')

#E[X] (Mean) = red
plt.axvline(x = 7/3, color = 'red', linestyle='--')

plot(xvalues,yvalues)
```

Out[20]: [<matplotlib.lines.Line2D at 0x2d8b205ff10>]



We can see that E[X] is the larger of the two. We could not have know that I believe since they were so close, at least without the graph. With the graph we can see that the logarithmic nature means that there was a leveling off of the probability of larger and larger tails, contributing to a larger sample pool of values that was reflected in the larger mean. Still, it's an educated guess at best in this case without understanding how rat tails are expect to be for that particular scenerio and thus requires this sort of calculation for a better understannding.

Part F: You head down to the lab to check out these new rodents, and find a beautiful litter of 5 running around. You decide to take this random sample and think about measuring it, but before that you think about the underlying random variable you've spent so much work describing. Assuming your model in this problem was correct, what are:

- The probability that at least 4 out of the 5 rodents have tails longer than the **median** tail?
- The probability that at least 2 out of the 5 rodents have tails longer than the **average** tail?

In [29]:

```
# integral from median = 2.25 to 4 of f(x) = 0.5 = p
# 4 rats, 4 rats
print(stats.binom.pmf(k=4, n = 5, p = 0.5) + stats.binom.pmf(k=5, n = 5, p = 0.5))
count = 0;
```

```
# integral from E[X] = 7/3 to 4 of f(x) = 2-sqrt(7/3) ~ 0.48 = p
# 2 rats, 3 rats, 4 rats, 5 rats
for i in range(2,5):
    count += stats.binom.pmf(k=i, n = 5, p = 2-math.sqrt(7/3))
print (count)
```

```
0.18749999999999994
0.7526576008797043
```

In [ ]: