Problem Set 3:

Problem 2:

Lyft line gets 2 requests every 5 minutes, on average, for a particular route (request times are independent). A user requests the route and Lyft commits a car to take her. All users who request the route in the next five minutes will be added to the car as long as the car has space. The car can fit up to three users. Lyft will make $7 for each user in the car (the revenue) minus $9 (the operating cost).
a. How much does Lyft expect to make from this trip?
b. Lyft has one space left in the car and wants to wait to get another user. What is the probability that another user will make a request in the next 30 seconds?

Solution:

(a) **5**. We can find the expected requests using the Poisson distribution we are given the average requests in given time frame. After finding the expected requests we multiply with 7 to find the amount earned then subtract 9 for operating cost:

$$X \sim Poi(\lambda = 2), \quad E[X] = \lambda = 2, \implies 7 \times 2 - 9 = 5.$$

(b) **~0.164**. Using Poisson random variable:

$$X \sim Poi(\lambda = 30 \times \frac{2}{300}),\ P(X = 1) = f(1)$$

```
>>> import scipy.stats as stats
>>> stats.poisson.pmf(1, 0.2)
0.1637461506155964
```

And we can also approximate the probability with writing the binomial form, with p being $\frac{\lambda}{m}$(m is the amount of events in 5 min):

$$X \sim Bin(n, \frac{\lambda}{m}) \implies \binom{n}{1}(\frac{\lambda}{m})^1(1 - \frac{\lambda}{m})^{n-1}.$$

Since we are only waiting for one person $k = 1$ we don't need to divide the time too many times:

$$\binom{30}{1}(\frac{2}{300})^1(1-\frac{2}{300})^{30-1}$$

```
>>> import scipy.stats as stats
>>> stats.binom.pmf(1, 30, 1/150)
0.16473476739077328
```

Problem 3:

Suppose it takes at least 9 votes from a 12-member jury to convict a defendant. Suppose also that the probability that a juror votes that an actually guilty person is innocent is 0.25, whereas the probability that the juror votes that an actually innocent person is guilty is 0.15. If each juror acts independently and if 70% of defendants are actually guilty

a. Find the probability that the jury renders a correct decision.

b. Determine the percentage of defendants found guilty by the jury.

Solution:

(a) **~0.726**. Can be solved using the binomial distribution:

$$P(9 \leq X \leq 12|Guilty)P(Guilty) = \sum_{i=9}^{12}\binom{12}{i}(0.75)^i(0.25)^{12-i}(0.70),$$

$$P(9 \leq X \leq 12|Not\ Guilty)P(Not\ Guilty) = \sum_{i=9}^{12}\binom{12}{i}(0.85)^i(0.15)^{12-i}(0.30),$$

$$\implies P(9 \leq X \leq 12|Guilty)P(Guilty) + P(9 \leq X \leq 12|Not\ Guilty)P(Not\ Guilty).$$

```
>>> g = stats.binom.cdf(12, 12, 0.75) - stats.binom.cdf(8, 12, 0.75)
>>> ng = stats.binom.cdf(12, 12, 0.85) - stats.binom.cdf(8, 12, 0.85)
>>> g, ng
(0.6487786173820496, 0.9077936688387293)
>>> g*0.70 + ng*0.30
0.7264831328190535
```

(b) **~45.42%**. Similar to part (a):

$$P(9 \leq X \leq 12|Guilty)P(Guilty) = \sum_{i=9}^{12}\binom{12}{i}(0.75)^i(0.25)^{12-i}(0.70),$$

$$P(9 \leq X \leq 12|Not\ Guilty)P(Not\ Guilty) = \sum_{i=9}^{12}\binom{12}{i}(0.15)^i(0.85)^{12-i}(0.30),$$

$$\implies P(9 \leq X \leq 12|Guilty)P(Guilty) + P(9 \leq X \leq 12|Not\ Guilty)P(Not\ Guilty)$$

$$\implies p \times 100.$$

```
>>> g = stats.binom.cdf(12, 12, 0.75) - stats.binom.cdf(8, 12, 0.75)
>>> ng = stats.binom.cdf(12, 12, 0.15) - stats.binom.cdf(8, 12, 0.15)
>>> g, ng
(0.6487786173820496, 5.477914412854723e-06)
>>> percent = (g*0.70 + ng*0.30) * 100
>>> percent
45.41466755417585
```

Problem 4:

To determine whether they have measles, 1000 people have their blood tested. However, rather than testing each individual separately (1000 tests is quite costly), it is decided to use a group testing procedure:

- Phase 1: First, place people into groups of 5. The blood samples of the 5 people in each group will be pooled and analyzed together. If the test is positive (at least one person in the pool has measles), continue to Phase 2. Otherwise send the group home. 200 of these pooled tests are performed.
- Phase 2: Individually test each of the 5 people in the group. 5 of these individual tests are performed per group in Phase 2.
  Suppose that the probability that a person has measles is 5% for all people, independently of others, and that the test has a 100% true positive rate and 0% false positive rate (note that this is unrealistic). Using this strategy, compute the expected total number of blood tests (individual and pooled) that we will have to do across Phases 1 and 2.

Solution:

426.22 $\pm$ **29.58**. First we find the expected group amount having no one with measles:

$$X \sim Bin(200, 0.95^5) \implies E(X) = \sum_{k=0}^{200} \binom{200}{k} (0.95^5)^k (1 - 0.95^5)^{200-k}$$

```
>>> mean = stats.binom.mean(200, 0.95**5)
>>> mean
154.75618749999995
```

Since there we expect only ~45.25 group to contain at least one person with measles we do 45.25×5 more tests which will be 426.22 tests in total. Note that since the standard deviation is given as ~5.9 below the amount of tests done will be within 29.58 tests of the amount found 68.2% of the time.

```
>>> std = stats.binom.std(200, 0.95**5)
>>> std
5.916823442720278
```

Problem 5:

The number of times a person's computer crashes in a month is a Poisson random variable with λ = 7. Suppose that a new operating system patch is released that reduces the Poisson parameter to λ = 2 for 80% of computers, and for the other 20% of computers the patch has no effect on the rate of crashes. If a person installs the patch, and has their computer crash 4 times in the month thereafter, how likely is it that the patch has had an effect on the user's computer (i.e., it is one of the 80% of computers that the patch reduces crashes on)?

Solution:

~0.802. We can use the Bayes theorem to find the answer to the problem:

$$P(A|x=4) = \frac{P(A,x=4)}{P(x=4)},$$
$$\text{where,}$$
$$A : \text{patch worked},$$
$$A^C : \text{patch didn't work},$$
$$x = 4 : \text{amount of crashes} = 4$$

We can find $P(x=4)$ i.e. denominator like we did in problem 3:

```
>>> a = stats.poisson.pmf(4, 7)
>>> a_c = stats.poisson.pmf(4, 2)
>>> a, a_c
(0.09122619163734964, 0.09022352215774178)
>>> n = a*0.80 + a_c*0.20
>>> n
0.09102565774142808
```

And since we have already calculated the nominator which is $a \times 0.80$, we can find the answer:

$$P(A|x=4) = \frac{P(A,x=4)}{P(x=4)} = \frac{a*80}{n}.$$

```
>>> (a*0.80) / n
```

```
0.8017624384236033
```

Problem 6:

Let X be a continuous random variable with probability density function:

$$f(x) = \begin{cases} c(2-2x^2) & \text{if } -1 < x < 1, \\ 0 & \text{otherwise.} \end{cases}$$

```
a. What is the value of c?
b. What is the cumulative distribution function (CDF) of X?
c. What is $E[X]$?
```

Solution:

(a) **-0.75**. Since the pdf must sum to 1:

$$\int_{-1}^{1} c(2-2x^2)dx = c(2-\tfrac{2}{3}x^3)\,|_{-1}^{1} = c((2-\tfrac{2}{3}) - (2+\tfrac{2}{3})) = -c(\tfrac{4}{3}) = 1$$
$$\implies c = -\tfrac{3}{4}$$

(b) $F(x) = \begin{cases} \frac{1}{2}(1+x^3), & \text{for } -1 \le x \le 1, \\ 1, & \text{for } 1 < x \\ 0, & \text{for } x < -1 \end{cases}$. Since the we have a continuous random variable we integrate:

$$F(x) = P(X \le x) = \int_{-1}^{x} -\tfrac{3}{2}(1-1t^2)dt = -\tfrac{3}{2}(1-\tfrac{1}{3}t^3)\,|_{-1}^{x}$$
$$\implies -\tfrac{3}{2}((1-\tfrac{1}{3}x^3) - (1+\tfrac{1}{3})) = \tfrac{3}{2}(\tfrac{1}{3}+\tfrac{x^3}{3}) = \tfrac{1}{2}(1+x^3)$$

(c) **0**. To find the expectation we multiply x with $f(x)$ and integrate the whole expression over R:

$$E(x) = \int_{-1}^{1} x \times -\tfrac{3}{2}(1-x^2)dx = \int_{-1}^{1} -\tfrac{3}{2}(x-x^3)dx = -\tfrac{3}{2}(\tfrac{x^2}{2} - \tfrac{x^4}{4})\,|_{-1}^{1}$$
$$\implies -\tfrac{3}{2}((\tfrac{1}{2}-\tfrac{1}{4}) - (\tfrac{1}{2}-\tfrac{1}{4})) = 0$$

Problem 7:

Say there are k buckets in a hash table. Each new string added to the table is hashed to bucket i with probability p_i , where $\sum_{i=1}^{k} p_i = 1$. If n strings are hashed into the table, find the expected number of buckets that have at least one string hashed to them. (Hint: Let X_i be a binary variable (i.e., Bernoulli random variable) that has the value 1 when

there is at least one string hashed to bucket i after the n strings are added to the table (and 0 otherwise). Compute $E[\sum_{i=1}^{k} X_i]$.)

Solution:

Since we are trying to find the expected amount of buckets that have at least one string we can turn this into a binomial random variable question where the we consider having at least one string a success and the expected value for that would be $experiment\ amount \times probability\ of\ success$:

$$For\ all\ buckets: \ X_i \sim Bin(n, p_i) \implies probability\ of\ success \implies 1 - P(X_i = 0)$$
$$\implies 1 - P(X_i = 0) = \binom{n}{0}(p_i)^0(1-p_i)^n = (1-p_i)^n$$
$$\implies P(X_i = 0) = 1 - (1-p_i)^n$$

where, n:amount of strings being hashed.
Now we find the expected amount of buckets with at least one string:

$$E[X] = k \times (1 - (1-p_i)^n).$$

Problem 8:

You are testing software and discover that your program has a non-deterministic bug that causes catastrophic failure (aka a “hindenbug”). Your program was tested for 400 hours and the bug occurred twice.
a. Each user uses your program to complete a three hour long task. If the "hindenbug" manifests they will immediately stop their work. What is the probability that the bug manifests for a given user?
b. Your program is used by 10,000 users. Use a Normal approximation to estimate the probability that more than 180 users experience the bug. Use your answer from part (a). Provide a numeric answer for this part.

Solution:

(a) **~0.0148**. Using the Poisson random variable:

$$X \sim Poi(\lambda = 3 \times 2/400) \implies P(X = 1) = \frac{\lambda^x e^{-\lambda}}{x!}$$

```
>>> stats.poisson.pmf(1, 6/400)
0.014776679094045942
```

(b) **~0.0040**. First using the binomial random variable be find the mean and the standard deviation for the sample and then we use the normal approximation to find the

probability:

$$X \sim Bin(10000, 0.0148)$$

```
>>> mean = stats.binom.mean(10000, 0.0148)
>>> std = stats.binom.std(10000, 0.0148)
>>> mean, std
(148.0, 12.075164595151488)
```

$$P(X \geq 180) = 1 - P(X < 180) \implies 1 - P(\frac{X - \mu}{\sigma} < 180) \implies 1 - \Phi(\frac{X - \mu}{\sigma})$$

```
>>> 1 - stats.norm.cdf(180, mean, std)
0.0040237857718912196
```

Problem 9:

Say the lifetimes of computer chips produced by a certain manufacturer are normally distributed with parameters μ = 1.5 × 10^6 hours and σ = 9 × 10^5 hours. The lifetime of each chip is independent of the other chips produced.
a. What is the approximate probability that a batch of 100 chips will contain at least 6 whose lifetimes are more than 3.0 × 10^6 hours?
b. What is the approximate probability that a batch of 100 chips will contain at least 65 whose lifetimes are less than 1.9 × 10^6 hours? Provide a numeric answer for this part.

Solution:

(a) **~0.344**. We first find the probability of a chip lasting at least 3.0 $\times$ 10^6 hours:

$$X \sim N(\mu, \sigma) \implies 1 - F(3.0 \times 10^6) = 1 - \Phi(\frac{3.0 \times 10^6 - \mu}{\sigma})$$

```
>>> p = 1 - stats.norm.cdf(3.0*10e6, 1.5*10e6, 9*10e5)
>>> p
0.047790352272814696
```

Then using this we make a Binomial random variable for more than 6 success out of 100 trials:

$$Y \sim Bin(100, p) \implies P(Y \geq 6) = 1 - P(Y < 6)$$

```
>>> 1 - stats.binom.cdf(5, 100, p)
0.3443388427361356
```

(b) ***~0.717***. Solution is similar to part a:

```
>>> p = stats.norm.cdf(1.9*10e6, 1.5*10e6, 9*10e5)
>>> p
0.6716393567181147
>>> 1 - stats.binom.cdf(64, 100, p)
0.7174349806102314
```

Problem 10:

A Bloom filter is a probabilistic implementation of the set data structure, an unordered collection of unique objects. In this problem we are going to look at it theoretically. Our Bloom filter uses 3 different independent hash functions H_1, H_2, H_3 that each take any string as input and each return an index into a bit-array of length n. Each index is equally likely for each hash function.
To add a string into the set, feed it to each of the 3 hash functions to get 3 array positions. Set the bits at all these positions to 1. For example, initially all values in the bit-array are zero. In this example n = 10:
Index: 0 1 2 3 4 5 6 7 8 9
Value: 0 0 0 0 0 0 0 0 0 0
After adding a string “pie”, where H_1(“pie”) = 4, H_2(“pie”) = 7, and H_3(“pie”) = 8:
Index: 0 1 2 3 4 5 6 7 8 9
Value: 0 0 0 0 1 0 0 1 1 0
Bits are never switched back to 0. Consider a Bloom filter with n = 9, 000 buckets. You have added m = 1,000 strings to the Bloom filter. Provide a numerical answer for all questions.
a. What is the (approximated) probability that the first bucket has 0 strings hashed to it?
To check whether a string is in the set, feed it to each of the 3 hash functions to get 3 array positions. If any of the bits at these positions is 0, the element is not in the set. If all bits at these positions are 1, the string may be in the set; but it could be that those bits are 1 because some of the other strings hashed to the same values. You may assume that the value of one bucket is independent of the value of all others.
b. What is the probability that a string which has not previously been added to the set will be misidentified as in the set? That is, what is the probability that the bits at all of its hash positions are already 1? Use approximations where appropriate.
c. Our Bloom filter uses three hash functions. Was that necessary? Repeat your calculation in (b) assuming that we only use a single hash function (not 3).
(Chrome uses a Bloom filter to keep track of malicious URLs. Questions such as this allow us to compute appropriate sizes for hash tables in order to get good performance with high probability in applications where we have a ballpark idea of the number of elements that will be hashed into the table.)

Solution:

(a) **~0.717**. We can imagine it using a Binomial random variable where we fail at the zeroth index for all three hash functions(binomial approximation):

$$\text{for all three hash function: } \Pi_{i=1}^{3} X_i \sim Bin(1000, 1/9000)$$
$$\implies P(X_1 = 0) \times P(X_2 = 0) \times P(X_3 = 0)$$

```
>>> stats.binom.pmf(0, 1000, 1/9000)**3
0.7165180406153465
```

(b) ***~0.00098***. We first find the probability of assigning 1 to each index and then multiply them to get the probability of getting all three of them(binomial approximation):

$$\text{for all three hash function: } \Pi_{i=1}^{3} X_i \sim Bin(1000, 1/9000)$$
$$\binom{3}{1} P(X_i = 1) \times \binom{3}{1} P(X_i = 1) \times \binom{3}{1} P(X_i = 1)$$

```
>>> (3 * stats.binom.pmf(1, 1000, 1/9000))**3
0.02654655307583188
```

(c) **~0.0994**. If we use only one hash function then our probability of getting the same index for multiple strings increase since probability of and consists of multiplying things that are between 0 and 1(binomial approximation):

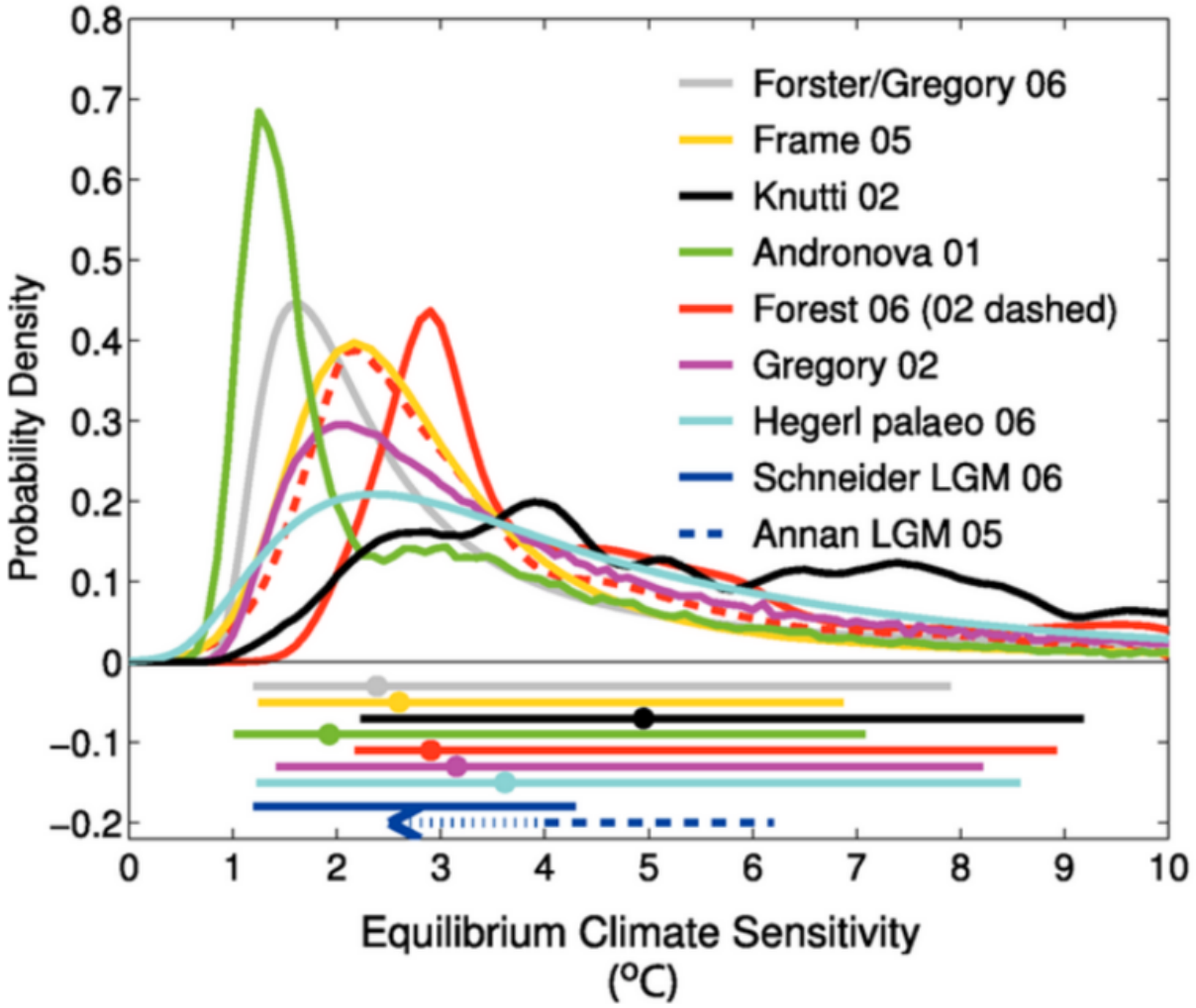
$$X \sim Bin(1000, 1/9000)$$
$$P(X = 1)$$

```
>>> stats.binom.pmf(1, 1000, 1/9000)
0.09943702552758504
```

Problem 11:

Last summer (May 2019) the concentration of CO_2 in the atmosphere was 414 parts per million (ppm) which is substantially higher than the pre-industrial concentration: 275 ppm. CO_2 is a greenhouse gas and as such increased CO_2 corresponds to a warmer planet.
Absent some pretty significant policy changes, we will reach a point within the next 50 years (i.e., well within your lifetime) where the CO_2 in the atmosphere will be double the pre-industrial level. In this problem we are going to explore the following question: What will happen to the global temperature if atmospheric CO_2 doubles?
The measure, in degrees Celsius, of how much the global average surface temperature will change (at the point of equilibrium) after a doubling of atmospheric CO_2 is called "Climate Sensitivity." Since the earth is a complicated ecosystem climate scientists model Climate Sensitivity as a random variable, S. The IPPC Fourth Assessment

Report had a summary of 10 scientific studies that estimated the PDF of S:



In this problem we are going to treat S as part-discrete and part-continuous. For values of S less than 7.5, we are going to model sensitivity as a discrete random variable with PMF based on the average of estimates from the studies in the IPCC report. Here is the PMF for S in the range 0 through 7.5:

Sensitivity, S (degrees C) 0 1 2 3 4 5 6 7

Expert Probability 0.00 0.11 0.26 0.22 0.16 0.09 0.06 0.04

The IPCC fifth assessment report notes that there is a non-negligible chance of S being greater than 7.5 degrees but didn't go into detail about probabilities. In the paper "Fat-Tailed Uncertainty in the Economics of Catastrophic Climate Change" Martin Weitzman discusses how different models for the PDF of Climate Sensitivity (S) for large values of S have wildly different policy implications.

For values of S greater than or equal to 7.5 degrees Celsius, we are going to model S as a continuous random variable. Consider two different assumptions for S when it is at least 7.5 degrees Celsius: a fat tailed distribution (f_1) and a thin tailed distribution (f_2):

$$f_1(x) = \frac{K}{x} \; s.t. \; 7.5 \leq x < 30$$

$$f_2(x) = \frac{K}{x^3} \; s.t. \; 7.5 \leq x < 30$$

For this problem assume that the probability that S is greater than 30 degrees Celsius is 0.

a. Compute the probability that Climate Sensitivity is at least 7.5 degrees Celsius.
b. Calculate the value of K for both f_1 and f_2.
c. It is estimated that if temperatures rise more than 10 degrees Celsius, all the ice on Greenland will melt. Estimate the probability that S is greater than 10 under both the f_1 and f_2 assumptions.
d. Calculate the expectation of S under both the f_1 and f_2 assumptions.
e. Let $R = S^2$ be a crude approximation of the cost to society that results from S. Calculate $E[R]$ under both the f_1 and f_2 assumptions.
Notes: (1) Both f_1 and f_2 are "power law distributions". (2) Calculating expectations for a variable that is part discrete and part continuous is as simple as: use the discrete formula for the discrete part and the continuous formula for the continuous part.

Solution:

(a) **0.06**. Since it is discrete for S we can find the answer easily:

$$P(S > 7.5) = 1 - P(S \leq 7.5) = 1 - \sum_{x=0}^{7} P(S = x) = 1 - 0.94 = 0.06$$

(b) **~0.043 and 1.8**. Integral of the bounded area for both functions should be equal to 1:

$$0.06 = \int_{7.5}^{30} \frac{K}{x} dx = K \ln|x| \,|_{7.5}^{30} = K(\ln(\frac{30}{7.5})) \implies K = \frac{0.06}{\ln 4} = 0.0432808512266689.$$

$$0.06 = \int_{7.5}^{30} \frac{K}{x^3} dx = \frac{K}{-2} x^{-2} \,|_{7.5}^{30} = \frac{K}{-2}(30^{-2} - 7.5^{-2}) \implies K = 1.8.$$

(c) **~0.048 and 0.008**. We integrate both equations for the interval $[10, 30]$:

$$\int_{10}^{30} \frac{K}{x} dx = K \ln|x| \,|_{10}^{30} = K(\ln(\tfrac{30}{10})) \implies 0.051 \ln 3 = 0.04754887502163469.$$

$$\int_{10}^{30} \tfrac{K}{x^3} dx = \tfrac{K}{-2} x^{-2} \,|_{10}^{30} = \tfrac{K}{-2}(30^{-2} - 10^{-2}) = 0.008.$$

d) **~7.543 and 7.515**. We multiply the pdf with x and take the integral over the bounded interval:

$$\int_{7.5}^{30} x \frac{K}{x} dx = K = 0.0432808512266689.$$

$$\int_{7.5}^{30} x \frac{K}{x^3} dx = \frac{K}{-1} x^{-1} \,|_{7.5}^{30} = \frac{1.8}{-1}(30^{-2} - 7.5^{-2}) = 0.015.$$

We add 7.5 to both cases since f_1 and f_2 are for when S is over 7.5.
(e) **~18.26 and ~2.495**. We find it similarly to part (d) using law of the unconscious statistician:

$$\int_{7.5}^{30} x^2 \frac{K}{x} dx = \frac{K}{2} x^2 \mid_{7.5}^{30} = 18.259109111250943.$$

$$\int_{7.5}^{30} x^2 \frac{K}{x^3} dx = K \ln |x| \mid_{7.5}^{30} = 1.8 \times (\ln(\frac{30}{7.5})) = 2.495329850015803.$$