Caltech Department of Mathematics

Ma 3/103
Introduction to Probability and Statistics

KC Border
Winter 2020

Assignment 3

Due Tuesday, January 28 by 4:00 p.m.
in the dropbox in the lobby of Linde Hall.

Instructions:

When asked for a probability or an expectation, give both a formula and an explanation for why you used that formula, and also give a numerical value when available.

Some of these questions involve using software to create plots. You are free to ask your classmates for help with coding questions.

When asked to plot something, use informative labels (even if handwritten), so the TA knows what you are plotting, attach a copy of the plot, and, if appropriate, the commands that produced it.

No collaboration is allowed on optional exercises.

Exercise 1 (Working with some data)

In lecture I suggested that examining the empirical distribution function was a good way to look at data. Let's compare it to using histograms.

At the beginning of the term you flipped coins. This generated a long string of 0's and 1's. A segment of this string can be interpreted as a binary number, and by dividing this by the appropriate power of two, it can be interpreted as a number between 0 and 1. Moreover, if the coin tosses are independent and Heads and Tails are equally likely, then these numbers should be i.i.d. with an approximately uniform distribution. We are going

to subject this to an "eyeball test," which is one of the first things you should always do with data.

I have taken the liberty of chopping the coin toss data into strings of length 16 and 32, and converting them into numbers between 0 and 1. You can download these results from http://www.its.caltech.edu/~kcborder/Courses/Ma3/Data/Random16.txt and http://www.its.caltech.edu/~kcborder/Courses/Ma3/Data/Random32.txt. Or you can do it yourself from the raw data at http://www.its.caltech.edu/~kcborder/Courses/Ma3/Data/Flips.txt

1. (10 pts) What is the expected value of a Uniform[0,1] random variable? What is its standard deviation?

Using the program/language of your choice do the following. (I give hints for R and Mathematica below.) **You are free to ask your classmates for help with coding questions.** For each of the two files:

2. (20 pts) What is the average of the numbers in your samples? What is the sample standard deviation of each sample? (The sample standard deviation is gotten by squaring the deviation of each sample value from the sample mean and dividing by the sample size.)

3. (20 pts) Plot a histogram of these numbers, using the default method. Then plot a histogram using bins of length 0.01.

4. (20 pts) Now plot a cumulative histogram (each bin adds to the previous bin). This give you a multiple of empirical distribution, which you may want to normalize. (In Mathematica, there are a couple of way to do this is and one just an option of the `Histogram` command, but in R, there is a separate command.)

5. (10 pts) Which method makes it easier to check by eye if the data appear to be uniform? Which file looks most uniform?

Be sure to label your plots.

If you don't have a preference, there is a lot to be said for learning the R statistical programming language. It is used widely on campus, and it looks like it will be around for a while. It is also **free** and runs on the major operating systems. You can get it at http://www.r-project.org. But if you are familiar with something else, go ahead and use what you know. That's what I do. I learned to use Mathematica 2 in 1992, and so

I still use it. Even Excel can probably handle this assignment, but future ones may be trickier.

In an appendix below I offer some code for R and for Mathematica. □

Exercise 2 (Another dumb expectation trick) Pitman [pp. 168–173] describes what he calls "the method of indicators," and proves the following.

Let $A_1, \dots, A_n$ be not necessarily disjoint events, and let X be the number of events that occur. This is a random variable with $X(s) = |\{i : s \in A_i\}| = \sum_{i=1}^n \mathbf{1}_{A_i}(s)$. Then since expectation is a linear operator, $\boldsymbol{E}\, X = P(A_1) + P(A_2) + \cdots + P(A_n)$.

Use this fact to answer the following questions.

1. (15 pts) There are m urns and n balls. Each ball is dropped into an urn at random, independently of the other balls. (An urn may contain more than one ball.) What is the expected number of urns that hold at least one ball?

2. (15 pts) A seven-card stud poker hand is dealt from a well-shuffled standard deck. Let X be the random variable that tells how many Clubs are in the hand. What is $\boldsymbol{E}\, X$?

Hint: Write down the sample space first. □

Exercise 3 (A gentle introduction to conditional expectations)

An expectation computed using a conditional probability is called a conditional expectation. That is, for a simple random variable $X = \sum_{i=1}^n x_i \mathbf{1}_{E_i}$, we define

$$\boldsymbol{E}(X \,\big|\, A) = \sum_{i=1}^n x_i P(E_i \,\big|\, A).$$

For example:

- (50 pts) You roll a single die until you get a six. What is the expected number of rolls (including the roll giving six) conditioned on the event that all throws gave even numbers. □

Exercise 4 (10 pts) How much time did you spend on the previous exercises? □

Exercise 5 (Optional Exercise) (50 pts) Select n points on a circle (not a disc) independently according to a uniform distribution. What is the probability that there is a semicircle containing all of them?

Hint: Think before you calculate. □

Appendix: Sample code for Exercise 1

If you don't have a preference, there is a lot to be said for learning the R statistical programming language. It is used widely on campus, and it looks like it will be around for a while. It is also **free** and runs on the major operating systems. You can get it at http://www.r-project.org. You may like to use a graphical interface with it, such as RStudio (https://rstudio.com). But if you are familiar with something else, go ahead and use what you know. I have heard good things about NumPy and SciPy, as well as Matlab. I use Mathematica since I started using Mathematica 2 in 1992. Caltech has a site license for Mathematica, so students can use it free. (Mathematica and Matlab were both developed by Caltech alumni.)

R is the open source version of S, which was developed at Bell Labs. These are the same folks who brought you C and UNIX, so the command names tend to be short and somewhat cryptic. Mathematica, on the other hand, is verbose with really long function names, which all start with uppercase letters. Mathematica uses brackets `[ ]` instead of parentheses `( )` to delimit function arguments. Both use double quotes `" "` to delimit strings such as file names.

The hints following are for the 32-bit numbers. Don't forget to also do the same for the 16-bit numbers.

Hint: Badly documented sample R code:

Warning: I am not an R programer, and I am sure there are probably better ways to do things. For instance, I use `=` to assign to variables, as most other languages do. Real R programmers use `<-` for assignment. Most of what I know I got by Googling various questions. Also—typing `?command` will bring up help on the command `command`. But this only works if you know the command name.

First, use

```
setwd("your_data_pathname")
```

to change your working directory to the folder where the data file is. Note the quotation marks! (Or be prepared to use a full path name.) You can use `getwd()` and `list.files()` check that you are in the right place. Note the empty parentheses!

Read the data from the file into an array. Check the length, it should be 800 for the file Random32.txt. (# is a comment character.)

```
a = as.matrix(read.table("Random32.txt"))  # the as.matrix is important!
length(a)
```

The mean and standard deviation of the sample are calculated by the functions

```
mean(a)
sd(a)
```

Now try a default histogram:

```
hist(a)
```

Now try a histogram with bins of size 0.01. Also instead of actual counts, use relative frequencies (density):

```
bins=seq(0.0,1.0,by=0.01)
hist(a, breaks=bins, freq=FALSE)   # freq=FALSE uses relative frequencies ?!
```

Now let's examine the empirical cdf.

```
c=ecdf(a)
plot(c)
```

How do you save these plots? Say you want to save the plot above to a `png` file named `Hist.png`. Here you go:

```
png("Hist.png")     # open the file for writing
plot(c)             # plot to the file
dev.off()           # close the file.  This is crucial.
```

To save to a `pdf` file use `pdf("Hist.pdf")` for the first line. I found this at http://wiki.stdout.org/rcookbook/Graphs/Output%20to%20a%20file/, but that URL seems to have disappeared.

Hint: Undocumented sample Mathematica code:

```
SetDirectory["Your path goes here"]
a = Flatten[ Import["Random32.txt", "Table"] ];

Mean[a]
StandardDeviation[a]

g1 = Histogram[a]
Export["File name 1.pdf", g1]
g2 = Histogram[a, 100 ]
Export["File name 2.pdf", g2]
```