

## Homework # 4:

(Do not use R for problems 1 - 4)

1) Find the following probabilities:

- a)  $\Pr\{Z < -0.64\}$                       b)  $\Pr\{Z \geq 0.64\}$                       c)  $\Pr\{-1.12 < Z < 1.12\}$   
d)  $\Pr\{-1.12 < Z < 0.0\}$                       e)  $\Pr\{-4.00 < Z < -1.12\}$                       f)  $\Pr\{-4.00 < Z < 0.0\}$   
g) Add (d) and (e) and compare to the answer to (f). Are you surprised? *Why or why not?*  
h)  $\Pr\{Z < -.22 \text{ or } Z > .22\}$  (you want the probability that  $Z$  is outside the range  $-.22$  to  $.22$ )  
i)  $\Pr\{-.22 < Z < .22\}$                       j) Add (h) and (i). Are you surprised? *Why or why not?*

2) Let's investigate the caracal (*Caracal caracal*, a cat native to Africa). Somehow we know that their true average weight ( $\mu$ ) is 13.5 kg ( $\mu = 13.5$  kg) and the true standard deviation ( $\sigma$ ) is 2.25 kg ( $\sigma = 2.25$  kg), and that their weight follows a normal distribution. Figure out the following probabilities.

- a)  $\Pr\{Y < 17\}$                       b)  $\Pr\{Y > 17\}$                       c)  $\Pr\{Y < 15\}$                       d)  $\Pr\{13 < Y < 17\}$

3) Refer to problem (2). Give the following percentiles:

- a) 17                      b) 33.3                      c) 50                      d) 90                      e) 95                      f) 99

4) The average number of white blood cells per cubic mm in humans is about 7,250 (this is highly variable). We'll assume that  $\mu = 7,250$ , and  $\sigma = 1,600$ . Let's also assume that the number of white blood cells follows a normal distribution.

- a) Give the values for the number of white blood cells for the middle 90% of people.  
(Hint/comment: if, for example, you want values for the middle 60%, you need to figure out many percent go in each tail. In other words, if you want the middle 60% that implies that you have 40% left that you need to divide equally into each tail. You will need two numbers in your answer. Draw a picture to help you.)
- b) Give the values for the number of red blood cells for the middle 95% of people.
- c) Give the 97.5<sup>th</sup> percentile.
- d) Are you surprised by the answers to (b) and (c)? *Explain*. If you're not sure what's going on, draw some pictures of the normal curves.
- e) Calculate the 99.9<sup>th</sup> percentile.
- f) Suppose you come across a person with a white blood cell count of 14,250 per cubic mm. Would you think anything was unusual about this person? *Why or why not (make sure you explain this!!)*.

Now let's do some R (on next page).

5) We want to explore the effect of parameters on a binomial distribution (this is a bit similar to last week's assignment).

Start up R/RStudio

a) First we'll do our coin example from lecture but use 15 tosses instead of 10. Let's calculate those probabilities and plot a graph so we know what our "distribution" looks like:

*In the script window or on the command line do the following:*

<code>y &lt;- c(0:15)</code>	we're giving y the numbers 0 through 15.
<code>y</code>	if you want to see "y", you can do this step (not necessary).
<code>pr &lt;- dbinom(y,15,.5)</code>	this gives us all the probabilities for the binomial. ".5" is the probability of success, "15" is the number of trials, and y is the number we're interested in (in this case, all the values from 0 to 15). Make sure you put in the numbers in the right order.
<code>pr</code>	this should give you all the probabilities.
<code>barplot(pr,names.arg = y)</code>	this generates a plot of "pr" (our probabilities) and tells it to use the contents of "y" to label our x axis.

***What does the graph look like?***

*Make sure you present your graph as well.* Instructions for cutting and pasting graphs were given in the last homework assignment.

b) Let's try some different "parameters". In the instructions for (a), let's change .5 to .75 (the third step above) and see what happens. This is kind of like having a coin that comes up heads 75% of the time.

***What does the graph look like now? What changed? Why? Which probabilities are now higher? Why?***

*Again, make sure you present your graph.*

6) Now let's try a normal distribution. The first plot will be easy, but the second will be more difficult to do, so follow the instructions carefully. First we'll plot a standard normal, then we'll see what happens if you change the parameters.

```
x <- seq (-5, 5, length.out=350)
```

This gives us 300 equally spaced values of  $x$  between -3.5 and 3.5. “seq” let's us generate a sequence of numbers, and “length.out” tells us how many numbers we want in the sequence.

```
plot(x, dnorm(x, mean = 0, sd = 1), col = "red", type = "l", ylim = c(0,0.425),  
ylab = "density", xlab = "y", main = "Several normal curves with different parameters")
```

“plot” is a command to generate graphs (it's quite versatile, and we can't explain everything here). “dnorm” is essentially the equation for the normal distribution. This command will plot  $x$  and the value of  $x$  at each of the 350 points using the normal equation. The “mean”, “sd” and “col” functions should be self-explanatory. type = “l” (that's lower case “L”, not “one”) generates lines instead of points (try it without type = “l” to see the difference). “ylim” sets the limits on the y-axis; we do this so that the curve doesn't touch the top of the box that's around the graph. “ylab” and “xlab” were covered in a previous homework but should be fairly obvious. “main” gives the main title over the graph.

You can hold off on copying and pasting your graph until you've done all of the steps below (you only need to hand in one graph for problem 6). See also the optional step at the end of problem 6 if you want to make your plot look nicer.

At this point you should have a standard normal curve. Let's see what happens when we change the parameters.

(a) First, let's change the mean from 0 to 1:

```
lines(x, dnorm(x, mean = 1, sd = 1), col = "blue")
```

“lines” will plot lines, but without erasing the previous plot. In other words, it'll plot the lines on the same graph as before. We changed the color to blue so you can see which plot is which.

***What happened to the normal distribution? How is it different from the first distribution you plotted?***

(b) Now let's adjust the standard deviation:

```
lines(x, dnorm(x, mean = 0, sd = 2), col = "green")
```

You should know how this works at this point.

***Again, what happened to the normal distribution? How is it different from the previous normal distributions you plotted?***

(c) Finally, let's adjust both the mean and standard deviation.

```
lines(x, dnorm(x, mean = 1.5, sd = 1.7), col = "purple")
```

***How is this different from the above normal distributions?***

**(d) Can you understand how important the parameters of a distribution are? What do the parameters of a distribution tell us about the distribution?**

**(f)** At this point your graph should have four curves plotted on top of each other in your final graph. Copy this graph and paste it into a word processor or whatever you want to use so that you can print a copy.

**Optional:** if you want to make your graph look even nicer, you can add the following:

```
legend (-4.5,0.39, c("mean = 0, sd = 1","mean = 1, sd = 1",  
"mean = 0, sd = 2", "mean = 1.5, sd = 1.7"),  
col = c("red","blue","green","purple"), lty = c(1,1,1,1))
```

The `legend` command can get pretty complicated. Here's a breakdown of what's going on:

The first two numbers (-4.5, 0.39) give the x and y coordinates of the top left corner of the legend box. These coordinates are the same ones used in the plot.

The first “`c(...)`” contains the actual labels for the legend. In this case we want to combine four “labels”, one for each line in our plot (“`c`” means combine or concatenate).

The “`col = c(...)`” part gives the four colors we want to use (we need to make sure they match the ones used in the plot and are in the same order as the first `c(...)` which has the labels).

Finally, we need to tell R what we used for plotting - we used lines, and “`lty(...)`” tells R to draw lines of the appropriate colors in our legend. The ones (1's) inside the `lty` command tell R to draw simple lines (R can draw many different kinds of lines - try using 2's or 3's here and you'll see what this means).

\*\*\*\*\*

### **Copying and pasting graphs from R:**

*See instructions from the previous set of homework.*

\*\*\*\*\*

***Problems are due in recitation the week of February 24<sup>th</sup>.***