

# STAT 521: Assignment 4

Make sure to show your computation and/or attach appropriate output.

## Problem 1

In a random sample of 17 members of a health club, resting pulse was measured:

54, 63, 88, 72, 49, 92, 70, 73, 69, 101, 48, 66, 80, 64, 77, 83, 91

- Calculate the sample mean and standard deviation. Use SAS, R, or Python.
- Based on part (a), manually calculate a 90% confidence interval for the true mean resting pulse. Assume that resting pulse is approximately normally distributed. Give your interpretation of the confidence interval.
- Based on part (a), manually calculate a 95% confidence interval for the true mean resting pulse. How does the 95% confidence interval compare to the 90% confidence interval?
- Use SAS, R or Python to verify your answer in part (b) and (c). Attach your output.

## Problem 2 (Computer exercise)

The data set **lowbwt** (see **Assignment4.sas/R/py**) contains information recorded for a sample of 100 low birth weight infants born in two teaching hospitals in Boston, Massachusetts. Measurements of systolic blood pressure are saved under the variable name **sbp**, while indicators of gender – where 1 represents a male and 0 a female – are saved under **sex**.

- Use SAS, R or Python to compute the mean and its 95% confidence interval for systolic blood pressure, broken down by gender of low birth weight infants. Attach your output.
- Compute a 95% confidence interval for the difference of mean systolic blood pressure between male and female low birth weight infants. Assume the population variances are equal. Attach your output.
- You wish to report your results for publication. Report the results in part (a) and (b) in the following format:

Table 1: Mean systolic blood pressure (in mmHg) and 95% confidence interval of 100 low birth-weight infants by gender

	Mean SBP	95% CI
Male	##.##	(##.##, ##.##)
Female	##.##	(##.##, ##.##)
Difference	##.##	( ##.##, ##.##)

- Do you think it is possible that males and females have the same mean systolic blood pressure? Explain briefly.

### Problem 3

Hemoglobin levels in 11-year-old boys vary according to a normal distribution with  $\sigma = 1.2$  g/dL.

- How large a sample is needed to estimate mean  $\mu$  with 95% confidence so the margin of error is 0.4 g/dL? Do this by hand and verify your answer using SAS, R or Python (see page 2-3 of Lecture 6 note).
- Suppose that you wish to estimate mean  $\mu$  with more precision. Change the margin of error to 0.1, 0.2, and 0.3 g/dL in part (a) above. Use SAS, R or Python and observe how the necessary sample size changes. Plot the required sample size as a function of the margin of error. Describe your findings.

### Problem 4

An epidemiologist wishes to know what proportion of adults living in a large metropolitan area have hepatitis B virus of subtype **ayr**.

- Determine the sample size that would be required to estimate the true proportion to within  $\pm 0.05$  with 95% confidence. In a similar metropolitan area the proportion of adults with the characteristic is reported to be 0.20. Do the calculation by hand and verify your answer using SAS, R or Python (see page 5 of Lecture 6 note).
- If data from another metropolitan area were not available and a pilot sample could not be drawn, what sample size would be required?

### Problem 5 (Biostats students only)

*Monte Carlo simulation in R:*

Let's do a very simple simulation using R. If you have not installed R in your computer, please download the R installer at <http://www.r-project.org>. I would also recommend using R studio (<http://www.rstudio.com/products/rstudio/download>) as a programming editor.

Once you install R (and R studio), start the program. In the R console, type:

```
# Generate 5 random values from  $Z \sim N(0, 1)$ 
z <- rnorm(5)
z
```

```
## [1] -0.56047565 -0.23017749  1.55870831  0.07050839  0.12928774
```

You just produced 5 random values from the standard normal distribution  $N(0,1)$ . Due to randomness, your result may be different from what I have above. Something you should know:

- A line that starts with # is a comment.
- The function `rnorm(n)` produces a sequence (vector) of  $n$  values from  $N(0,1)$ . This vector of 5 random values was assigned to an object `z` (`<-` is an assignment)
- To see the values assigned to `z`, just type `z`.
- You don't have to type commands directly in the R console. You can write your commands in the editor window (from the menu, File -> New Files -> R Script) and then run your code in there.

Now suppose you want to see if *each of the 5 values is less than 1.28*. You can evaluate this by running the following code:

```
# Are they smaller than 1.28?  
z < 1.28
```

```
## [1] TRUE TRUE FALSE TRUE TRUE
```

The answer is returned as logical (either **TRUE** or **FALSE**). In this case, 4 out of the 5 values are less than 1.28. Only the third value was greater than or equal to 1.28 and thus it was returned **FALSE**. The following code actually counts how many of them are **TRUE**.

```
# Count them  
sum(z < 1.28)
```

```
## [1] 4
```

If you apply the `sum()` function on an object with logical data type, **TRUE** is considered as 1 and **FALSE** as 0. Thus, this returns the number of **TRUE** which is 4 in this case.

If we want to know the proportion of  $Z$  taking less than 1.28, or the probability  $P(Z < 1.28)$ , then use the `mean()` function:

```
# Get the proportion of z < 1.28  
mean(z < 1.28)
```

```
## [1] 0.8
```

In this case, 80% of all values (4 out of 5) are less than 1.28.

Now if we can produce so many random numbers from the standard normal distribution  $N(0, 1)$  and get the proportion of  $Z < 1.28$ , then we should be able to approximate the CDF of  $Z$  up to 1.28, i.e.,  $P(Z < 1.28) = F_Z(1.28)$ . Okay then, let's produce 1 million random numbers from  $N(0, 1)$  and then approximate  $P(Z < 1.28)$ :

```
# Generate 1 million random values from Z ~ N(0, 1)  
z <- rnorm(10 ^ 6)
```

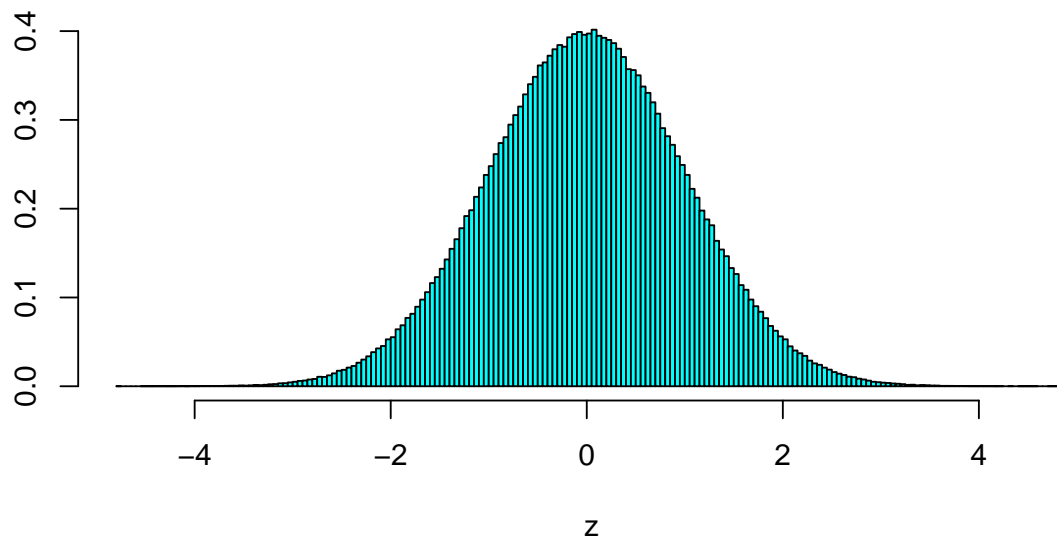
```
# Approx P(z < 1.28)  
mean(z < 1.28)
```

```
## [1] 0.899728
```

The answer I get is 0.899728. This value is actually very close to the true value of  $P(Z < 1.28) = 0.8997274 \approx 0.9$  (Look up the z-table by yourself).

You can also look at the histogram of 1 million  $z$  values that you just created. Run the following commands:

```
# Look at the histogram  
library(MASS)  
truehist(z)
```



That's a nice looking normal distribution! The `MASS` library is required to use `truehist()` command that display a histogram (the first line of the code above).

**Here's some exercise for you.** Please submit your R code as well as your answers.

- a) Produce one million random numbers from  $N(0, 1)$  and approximate  $P(Z < 1.96)$
- b) Similarly, approximate  $P(Z < -1.28 \text{ or } Z > 1.28)$ . You will need to use the “or” operator `|` (vertical bar or “pipe”)
- c) Approximate  $P(-1.96 < Z < 1.96)$ . You will need to use the “and” operator `&` (ampersand)

You might want to check Quick-R: Logical operators: <https://www.statmethods.net/management/operators.html>