# Math 152 - Statistical Theory - Homework 7

write your name here

Due: Friday, Oct 9, 2020, midnight PDT

## Important Note:

You should work to turn in assignments that are clear, communicative, and concise. Part of what you need to do is not print pages and pages of output. Additionally, you should remove these exact sentences and the information about HW scoring below.

Click on the *Knit to PDF* icon at the top of R Studio to run the R code and create a PDF document simultaneously. [PDF will only work if either (1) you are using R on the network, or (2) you have LaTeX installed on your computer. Lightweight LaTeX installation here: https://yihui.name/tinytex/]

> Either use the college's RStudio server (https://rstudio.pomona.edu/) or install R and R Studio on to your personal computer. See: https://research.pomona.edu/johardin/math152f20/ for resources.

#### Assignment

#### 1: PodQ

Describe one thing you learned from someone in your pod this week (it could be: content, logistical help, background material, R information, etc.) 1-3 sentences.

#### 2: 8.5.6

Suppose that $X_1, \dots, X_n$ form a random sample from the exponential distribution with unknown mean $\mu$. Describe a method for constructing a confidence interval for $\mu$ with a specified confidence coefficient $\gamma$ ($0 < \gamma$ $<1$).

Hint: Determine constants $c_1$ and $c_2$ such that $P(c_1 < (1/\mu)\sum_{i=1}^n X_i < c_2) = \gamma$.

Also, see Theorem 5.7.7 and Definition 8.2.1. And Exercise 5.7.1 could be helpful.

#### 3: 8.5.7

In the June 1986 issue of Consumer Reports, some data on the calorie content of beef hot dogs is given. Here are the numbers of calories (kcal) in 20 different hot dog brands:

```
hotdogs <- c(186, 181, 176, 149, 184, 190, 158, 139, 175, 148,
             152, 111, 141, 153, 190, 157, 131, 149, 135, 132)
```

Assume that these numbers are the observed values from a random sample of twenty independent normal random variables with mean $\mu$ and variance $\sigma^2$, both unknown. Find and interpret a 90% confidence interval for the mean number of calories $\mu$.

#### 4: 8.6.5

Suppose that two random variables $\mu$ and $\tau$ have the joint normal-gamma distribution such that $E(\mu) = -5$. $\text{Var}(\mu) = 1$, $E(\tau) = 1/2$, and $\text{Var}(\tau) = 1/8$. Find the prior hyperparameters $\mu_0$, $\lambda_0$, $\alpha_0$, and $\beta_0$ that specify the normal-gamma distribution. Use the theorems in the text.

### 5: 8.6.8

Suppose that two random variables $\mu$ and $\tau$ have the joint normal-gamma distribution with hyperparameters $\mu_0 = 4, \lambda_0 = 0.5, \alpha_0 = 1, and \beta_0 = 8$. Find the values of

a. $P(\mu > 0)$
b. $P(0.736 < \mu < 15.680)$

### 6: 8.6.9

Using the prior and data in the numerical example on nursing homes in New Mexico in this section, find

a. the shortest possible interval such that the **posterior** probability that $\mu$ lies in the interval is 0.90, and b.the shortest possible **confidence** interval for $\mu$ for which the confidence coefficient is 0.90.

### 7: R - confidence interval coverage

**Note** Because of this week's exam, I have included all of the R code for this problem. There is *no* R code for you to write! However, you do need to interpret the results in a few sentences for each question.

How well do frequentist confidence intervals actually capture the parameter of interest? What happens when we forget to use a t-multiplier and use a standard normal multiplier instead? First, let's see what happens when we correctly use the t-multiplier. Remember, we're talking about sampling distributions which means we'll have to take LOTS OF SAMPLES and look at many different confidence intervals.

a. Comment on the coverage rate of a standard t-interval for the population mean, $\mu$. [Bigger n.samps will probably give you more information.]

```r
set.seed(47)
n.samps = 10000  # you might get more info by taking more samples
n.obs = 10  # what happens if you increase the sample size?
mymean = c()  # place holder
myvar = c()  # place holder
conf.level = 0.95
mu = 47
sigma = 4

for (i in 1:n.samps) {
  mysample = rnorm(n.obs, mu, sigma)  #note, mean is mu, sd is sigma
  mymean = c(mymean, mean(mysample))
  myvar = c(myvar, var(mysample))

}

upper.CI = mymean - qt((1 - conf.level)/2, n.obs - 1) * sqrt(myvar)/sqrt(n.obs)
lower.CI = mymean + qt((1 - conf.level)/2, n.obs - 1) * sqrt(myvar)/sqrt(n.obs)

sum(upper.CI < mu)
```

```
## [1] 244
```

```r
sum(lower.CI > mu)
```

```
## [1] 249
```

b. Repeat a. above but change qt to use the quantile (multiplier) for a normal distribution instead of a t distribution. What is the new coverage rate? Why does that make sense?

```
set.seed(7474)
n.samps = 10000  # you might get more info by taking more samples
n.obs = 10  # what happens if you increase the sample size?
mymean = c()  # place holder
myvar = c()  # place holder
conf.level = 0.95
mu = 47
sigma = 4

for (i in 1:n.samps) {
  mysample = rnorm(n.obs, mu, sigma)  #note, mean is mu, sd is sigma
  mymean = c(mymean, mean(mysample))
  myvar = c(myvar, var(mysample))

}

upper.CI = mymean - qnorm((1 - conf.level)/2, 0, 1) * sqrt(myvar)/sqrt(n.obs)
lower.CI = mymean + qnorm((1 - conf.level)/2, 0, 1) * sqrt(myvar)/sqrt(n.obs)

sum(upper.CI < mu)
```

```
## [1] 389
```

```
sum(lower.CI > mu)
```

```
## [1] 460
```

c. Repeat a. and b. for a sample of size 100 (n.obs=100). Also, report the actual multipliers (the output of `qt()` and `qnorm()`). How does sample size play a role in coverage rate?

```
set.seed(4774)
n.samps = 10000  # you might get more info by taking more samples
n.obs = 100  # what happens if you increase the sample size?
mymean = c()  # place holder
myvar = c()  # place holder
conf.level = 0.95
mu = 47
sigma = 4

for (i in 1:n.samps) {
  mysample = rnorm(n.obs, mu, sigma)  #note, mean is mu, sd is sigma
  mymean = c(mymean, mean(mysample))
  myvar = c(myvar, var(mysample))

}

upper.CI = mymean - qt((1 - conf.level)/2, n.obs - 1) * sqrt(myvar)/sqrt(n.obs)
lower.CI = mymean + qt((1 - conf.level)/2, n.obs - 1) * sqrt(myvar)/sqrt(n.obs)

sum(upper.CI < mu)
```

```
## [1] 240
```

```r
sum(lower.CI > mu)
```

```
## [1] 236
```

```r
(n.samps - sum(upper.CI < mu) - sum(lower.CI > mu) )/ n.samps
```

```
## [1] 0.9524
```

```r
n.samps = 10000  # you might get more info by taking more samples
n.obs = 100  # what happens if you increase the sample size?
mymean = c()  # place holder
myvar = c()  # place holder
conf.level = 0.947
mu = 47
sigma = 4

for (i in 1:n.samps) {
  mysample = rnorm(n.obs, mu, sigma)  #note, mean is mu, sd is sigma
  mymean = c(mymean, mean(mysample))
  myvar = c(myvar, var(mysample))

}

upper.CI = mymean - qnorm((1 - conf.level)/2, 0, 1) * sqrt(myvar)/sqrt(n.obs)
lower.CI = mymean + qnorm((1 - conf.level)/2, 0, 1) * sqrt(myvar)/sqrt(n.obs)

sum(upper.CI < mu)
```

```
## [1] 282
```

```r
sum(lower.CI > mu)
```

```
## [1] 264
```

```r
(n.samps - sum(upper.CI < mu) - sum(lower.CI > mu) )/ n.samps
```

```
## [1] 0.9454
```