Knowledge Quiz 1

YOUR NAME HERE

Please answer the following questions, render a pdf, and submit via moodle by 11 PM on Mon Oct 7.

Guidelines:

- No consulting with anyone else (other than clarification questions from Professor Roback)
- You may use only materials from this class (our class webpage, links on moodle, our 3 online textbooks, files posted to the RStudio server)
- No online searches or use of large language models like ChatGPT

Pledge:

I pledge my honor that on this quiz I have neither given nor received assistance not explicitly approved by the professor and that I an aware of no dishonest work.

- type your name here to acknowledge the pledge: ____
- OR
- place an X here if you intentionally are not signing the pledge: ___________

```r
library(tidyverse)
```

1. Here's is a crazy list that tells you some stuff about data science. Give code that will produce **exactly** the following outputs.

```r
data_sci <- list(
  first = c("first it must work", "then it can be" , "pretty"),
  DRY = c("Do not", "Repeat", "Yourself"),
  dont_forget = c("garbage", "in", "out"),
  our_first_tibble = mpg,
  integers = 1:25,
  doubles = sqrt(1:25),
  tidyverse = c(pack1 = "ggplot2",
                pack2 = "dplyr",
                pack3 = "lubridate",
                etc = "and more!"),
  opinion = list("SDS 264 is",
                 c("awesome!", "amazing!", "rainbows!"))
)

str(data_sci)
```

```
List of 8
 $ first           : chr [1:3] "first it must work" "then it can be" "pretty"
 $ DRY             : chr [1:3] "Do not" "Repeat" "Yourself"
 $ dont_forget     : chr [1:3] "garbage" "in" "out"
 $ our_first_tibble: tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
  ..$ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
  ..$ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
  ..$ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
  ..$ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
  ..$ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
  ..$ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
  ..$ drv         : chr [1:234] "f" "f" "f" "f" ...
  ..$ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
  ..$ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
  ..$ fl          : chr [1:234] "p" "p" "p" "p" ...
  ..$ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
 $ integers        : int [1:25] 1 2 3 4 5 6 7 8 9 10 ...
 $ doubles         : num [1:25] 1 1.41 1.73 2 2.24 ...
 $ tidyverse       : Named chr [1:4] "ggplot2" "dplyr" "lubridate" "and more!"
  ..- attr(*, "names")= chr [1:4] "pack1" "pack2" "pack3" "etc"
 $ opinion         :List of 2
  ..$ : chr "SDS 264 is"
  ..$ : chr [1:3] "awesome!" "amazing!" "rainbows!"
```

[1] “first it must work” “then it can be” “pretty”

```
data_sci$first
```

```
[1] "first it must work" "then it can be"     "pretty"
```

```r
data_sci[["first"]]
```

```
[1] "first it must work" "then it can be"     "pretty"
```

```r
data_sci[[1]]
```

```
[1] "first it must work" "then it can be"     "pretty"
```

```r
typeof(data_sci[[1]])
```

```
[1] "character"
```

$DRY [1] “Do not” “Repeat” “Yourself”

```r
data_sci[2]
```

```
$DRY
[1] "Do not"   "Repeat"   "Yourself"
```

```r
data_sci["DRY"]
```

```
$DRY
[1] "Do not"   "Repeat"   "Yourself"
```

```r
typeof(data_sci["DRY"])
```

```
[1] "list"
```

[1] 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50

```r
data_sci[["integers"]] * 2
```

```
 [1]  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
```

```
data_sci[[5]] * 2
```

```
 [1]  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
```

```
data_sci$integers * 2
```

```
 [1]  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50
```

```
typeof(data_sci$integers)
```

```
[1] "integer"
```

[1] “rainbows!”

```
data_sci[[8]][[2]][3]
```

```
[1] "rainbows!"
```

```
data_sci$opinion[[2]][3]
```

```
[1] "rainbows!"
```

```
typeof(data_sci$opinion[[2]])
```

```
[1] "character"
```

[1] “garbage” “in” “garbage” “out”

```
data_sci[[3]][c(1,2,1,3)]
```

```
[1] "garbage" "in"      "garbage" "out"
```

```
data_sci$dont_forget[c(1,2,1,3)]
```

```
[1] "garbage" "in"      "garbage" "out"
```

```r
typeof(data_sci$dont_forget[c(1,2,1,3)])
```

```
[1] "character"
```

A tibble: 234 x 2

```
 hwy   cty
```

1 29 18 2 29 21 3 31 20 4 30 21 5 26 16 6 26 18 7 27 18 8 26 18 9 25 16 10 28 20 # ... with 224 more rows

```r
typeof(data_sci[[4]])
```

```
[1] "list"
```

```r
is_tibble(data_sci[[4]])
```

```
[1] TRUE
```

```r
data_sci$our_first_tibble[c("hwy", "cty")]
```

```
# A tibble: 234 x 2
     hwy   cty
   <int> <int>
 1    29    18
 2    29    21
 3    31    20
 4    30    21
 5    26    16
 6    26    18
 7    27    18
 8    26    18
 9    25    16
10    28    20
# i 224 more rows
```

```r
data_sci[[4]][c("hwy", "cty")]
```

```
# A tibble: 234 x 2
     hwy   cty
   <int> <int>
 1    29    18
 2    29    21
 3    31    20
 4    30    21
 5    26    16
 6    26    18
 7    27    18
 8    26    18
 9    25    16
10    28    20
# i 224 more rows
```

```r
data_sci[[4]] %>%
  select(hwy, cty)
```

```
# A tibble: 234 x 2
     hwy   cty
   <int> <int>
 1    29    18
 2    29    21
 3    31    20
 4    30    21
 5    26    16
 6    26    18
 7    27    18
 8    26    18
 9    25    16
10    28    20
# i 224 more rows
```

2. Write a function called `summary_stats()` that allows a user to input a tibble, numeric variables in that tibble, and summary statistics that they would like to see for each variable. Using `across()`, the function's output should look like the example below.

```
summary_stats(mtcars,
              vars = c(mpg, hp, wt),
              stat_fcts = list(mean = mean,
                               median = median,
                               sd = sd,
                               IQR = IQR))

#  mpg_mean mpg_median   mpg_sd mpg_IQR  hp_mean hp_median    hp_sd hp_IQR wt_mean
# 1 20.09062       19.2 6.026948   7.375 146.6875       123 68.56287   83.5 3.21725
#  wt_median     wt_sd  wt_IQR  n
# 1     3.325 0.9784574 1.02875 32
```

```
summary_stats <- function(df,
                          vars = where(is.numeric),
                          stat_fcts = list(mean = mean, sd = sd)) {
  df |>
    summarize(
      across(.cols = {{ vars }},
             .fns = stat_fcts),
      n = n(),
      .groups = "drop"
    )
}

summary_stats(mtcars)
```

```
  mpg_mean    mpg_sd cyl_mean    cyl_sd disp_mean   disp_sd  hp_mean     hp_sd
1 20.09062 6.026948   6.1875 1.785922  230.7219 123.9387 146.6875 68.56287
  drat_mean   drat_sd wt_mean     wt_sd qsec_mean  qsec_sd vs_mean     vs_sd
1  3.596563 0.5346787 3.21725 0.9784574  17.84875 1.786943  0.4375 0.5040161
  am_mean     am_sd gear_mean   gear_sd carb_mean carb_sd  n
1 0.40625 0.4989909    3.6875 0.7378041    2.8125  1.6152 32
```

```
summary_stats(mtcars, vars = c(mpg, hp, wt))
```

```
  mpg_mean    mpg_sd  hp_mean    hp_sd wt_mean     wt_sd  n
1 20.09062 6.026948 146.6875 68.56287 3.21725 0.9784574 32
```

```
summary_stats(mtcars,
              vars = c(mpg, hp, wt),
              stat_fcts = list(mean = mean,
                               median = median,
                               sd = sd,
                               IQR = IQR))
```

```
  mpg_mean mpg_median   mpg_sd mpg_IQR  hp_mean hp_median    hp_sd hp_IQR
1 20.09062       19.2 6.026948   7.375 146.6875       123 68.56287   83.5
  wt_mean wt_median     wt_sd  wt_IQR  n
1 3.21725     3.325 0.9784574 1.02875 32
```

3. The Central Limit Theorem is one of the most amazing results in all of mathematics. It says that if you take random samples from any population, if the sample size is large enough, the sample means will follow a normal distribution. This is true no matter how not-normal the original population is - crazy but true!! Let's explore the CLT in two steps.

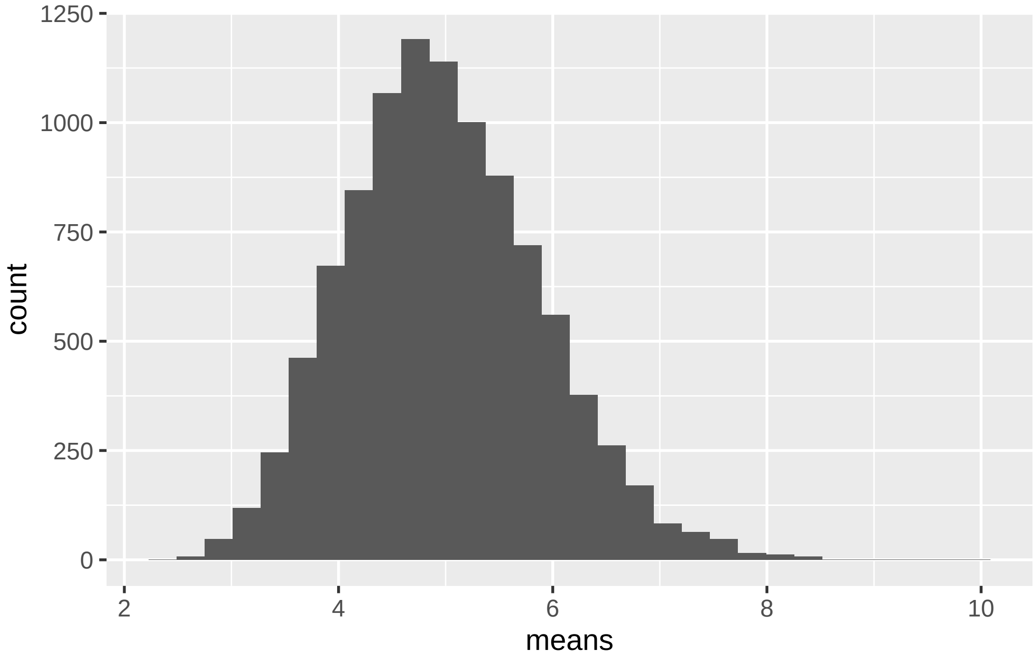
a) Write a for loop that takes 10,000 samples of size 30 from a skewed distribution and then plots the 10,000 means in a histogram to let us see if the histogram follows a normal distribution. Here are a couple of hints:

- `rexp(30, rate = 0.2)` will produce a random sample of size 30 from a skewed distribution
- `tibble(x = x)` will take a vector `x` and turn it into a column of a tibble that can be used in `ggplot`

b) Turn your for loop from (a) into a function whose attributes are `samp_size` with default of 30, and `n_means` with default of 10000. In addition, your histogram should now have a title that says "Means from samples of size 30 from a skewed population", where 30 is replaced with the user's input.

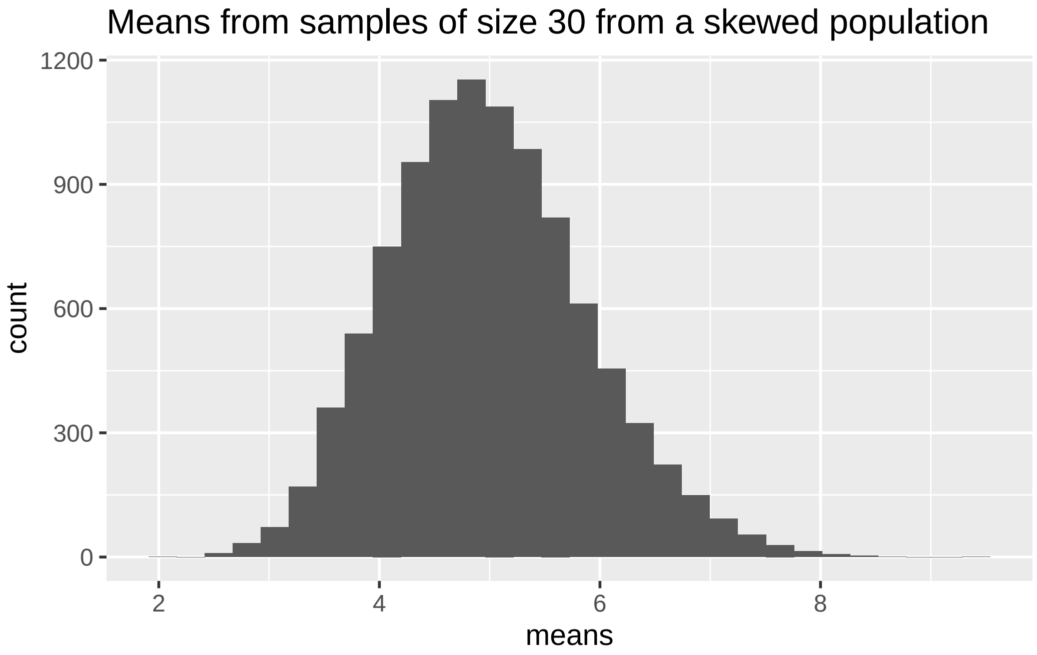
```
means <- vector("double", 10000)
for (i in 1:10000) {
  x <- rexp(30, rate = 0.2)
  means[[i]] <- mean(x)
}
temp <- tibble(means = means)
ggplot(temp, aes(x = means)) +
  geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



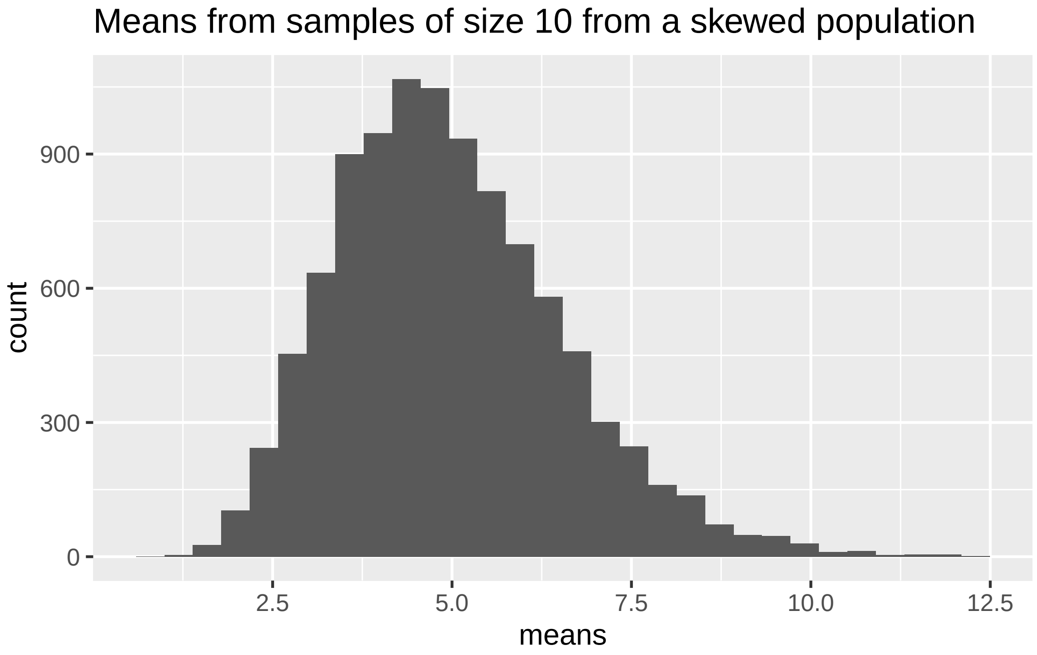
```r
clt <- function(samp_size= 30, n_means = 10000) {
  means <- vector("double", n_means)
  for (i in 1:n_means) {
    x <- rexp(samp_size, rate = 0.2)
    means[[i]] <- mean(x)
  }

  label <- rlang::englue("Means from samples of size {samp_size} from a skewed population")

  temp <- tibble(means = means)
  ggplot(temp, aes(x = means)) +
    geom_histogram() +
    labs(title = label)
}

clt()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
clt(10)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Means from samples of size 10 from a skewed population
count
means