

Homework 1

Chris Johnson

2025-02-07

```
# Homework 1
# Chris Johnson
# 2/7/2025
```

```
# Load the ggplot2 library for problem 3 and tidyverse and tidyr for problem 4.
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr      1.1.4      v readr      2.1.5
## v forcats    1.0.0      v stringr   1.5.1
## v lubridate  1.9.4      v tibble    3.2.1
## v purrr      1.0.2      v tidyr     1.3.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tidyr)
library(knitr)
```

```
# This assignment will need the following datasets: "Su_raw_matrix.txt",
# "diabetes_train.csv", and "titanic.csv".
```

```
# -----
```

```
# 1. Use "Su_raw_matrix.txt" for the following questions (30 points).
```

```
# First set the drive to where "Su_raw_matrix.txt" is saved.
```

```
setwd("C:/Users/Chris/OneDrive/Desktop/U of M/Winter25/CSC587/from_gdrive")
```

```
# 1a. Use read.delim function to read Su_raw_matrix.txt into a variable called su.
```

```
# Step 1: Read the "Su_raw_matrix.txt" file and turn it into a dataframe named "su".
```

```
# Use read.delim() with sep = "\t" to indicate tabs and header = TRUE
```

```
su = read.delim("Su_raw_matrix.txt", sep = '\t', header = TRUE)
```

```
# Step 2: Use View() with su to confirm the dataframe looks right.
```

```
# View(su)
```

```
# 1b. Use mean and sd functions to find mean and standard deviation of Liver_2.CEL column
```

```
# Step 1: Use the mean function in R on the "Liver_2.CEL" column of dataframe 'su'
```

```
# to obtain the mean value of the column.
mean(su$Liver_2.CEL)
```

```
## [1] 241.8246
```

```
# Result: Mean = 241.8246
```

```
# Step 2: Use the sd function in R on the "Liver_2.CEL" column of dataframe "su"
# to obtain the standard deviation of the column.
sd(su$Liver_2.CEL)
```

```
## [1] 1133.352
```

```
# Result: Standard Deviation = 1133.352
```

```
# 1c. Use colMeans and colSums functions to get the average and total values of each column.
# Step 1: Use the colMeans function to get the average of each column
colMeans(su)
```

```
##      Brain_1.CEL      Brain_2.CEL Fetal_brain_1.CEL Fetal_brain_2.CEL
##      204.9763      315.0924      198.3439      267.6551
## Fetal_liver_1.CEL Fetal_liver_2.CEL      Liver_1.CEL      Liver_2.CEL
##      209.8722      399.1482      160.8558      241.8246
```

```
# Results: Brain_1.CEL      Brain_2.CEL Fetal_brain_1.CEL Fetal_brain_2.CEL Fetal_liver_1.CEL
#           204.9763      315.0924      198.3439      267.6551      209.8722
#           Fetal_liver_2.CEL      Liver_1.CEL      Liver_2.CEL
#           399.1482      160.8558      241.8246
```

```
# Step 2: Use the colSums function to get the sum of each column
colSums(su)
```

```
##      Brain_1.CEL      Brain_2.CEL Fetal_brain_1.CEL Fetal_brain_2.CEL
##      2588031      3978357      2504290      3379413
## Fetal_liver_1.CEL Fetal_liver_2.CEL      Liver_1.CEL      Liver_2.CEL
##      2649846      5039645      2030966      3053278
```

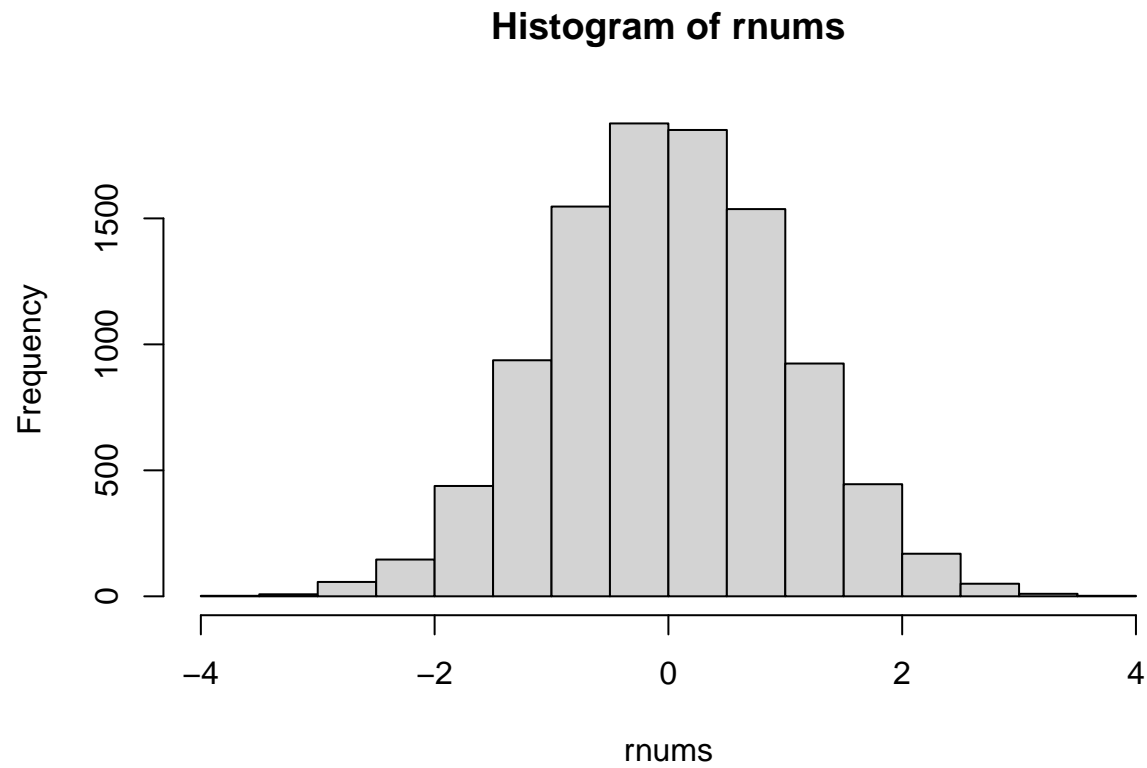
```
# Results: Brain_1.CEL      Brain_2.CEL Fetal_brain_1.CEL Fetal_brain_2.CEL Fetal_liver_1.CEL
#           2588031      3978357      2504290      3379413      2649846
#           Fetal_liver_2.CEL      Liver_1.CEL      Liver_2.CEL
#           5039645      2030966      3053278
```

```
# -----
```

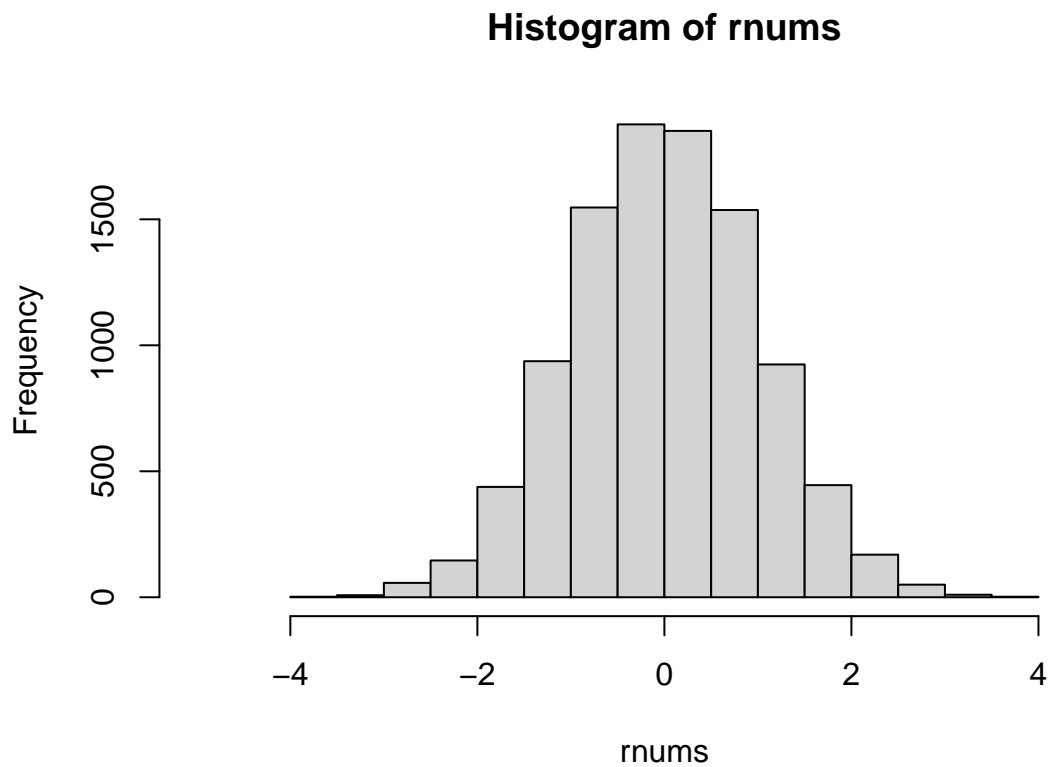
```
# 2. Use rnorm(n, mean = 0, sd = 1) function in R to generate 10000 numbers for the following
# (mean, sigma) pairs and plot histogram for each, meaning you need to change the function
# parameter accordingly. Then comment on how these histograms are different from each other
# and state the reason. (20 points)
```

```
# Step 1: Generate a set of 10,000 random numbers with a mean of 0 and standard deviation of 1.
rnums = rnorm(10000, mean = 0, sd = 1)
```

```
# Step 2: Run it to confirm it works.  
# rnums  
  
# Step 3: Create a histogram for rnums.  
hist(rnums)
```



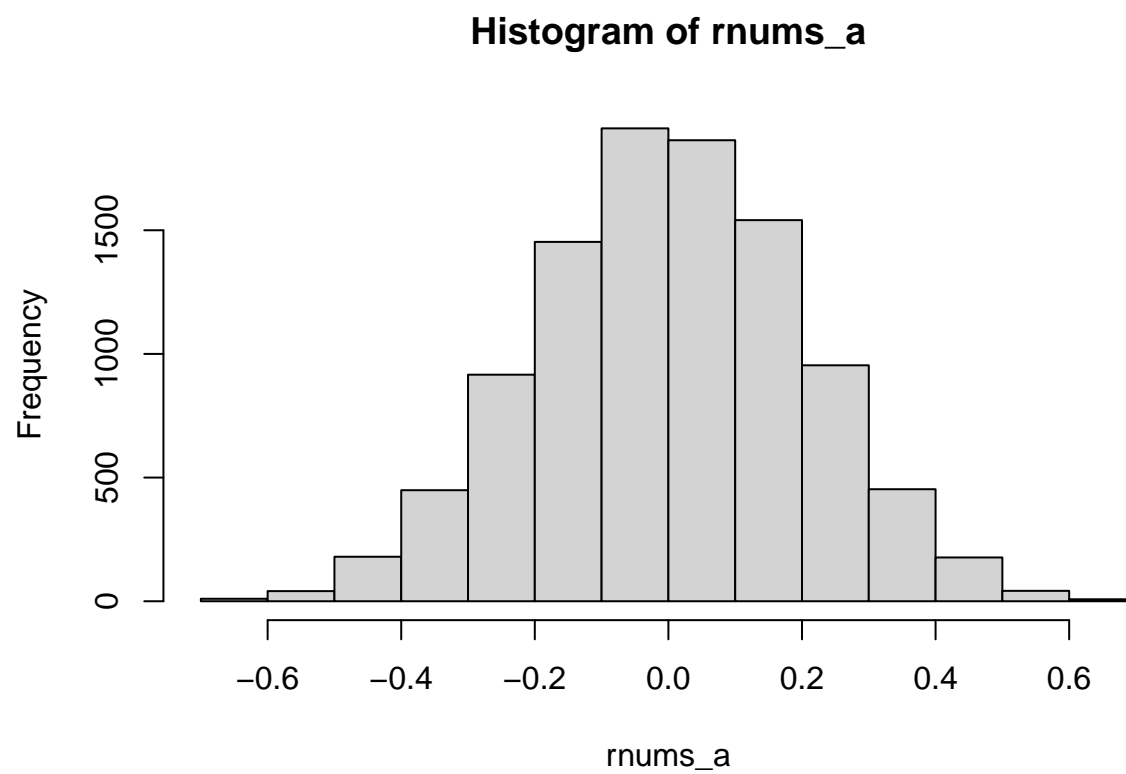
```
# Step 4: To better see the plot differences set xlim to c(-5, 5) and plot.  
hist(rnums, xlim = c(-5, 5))
```



```
# 2a. mean = 0, sigma = 0.2
# Step 1: Use rnorm function with n = 10000, mean = 0, sigma = 0.2 to create values rnums_a.
rnums_a = rnorm(10000, mean = 0, sd = 0.2)

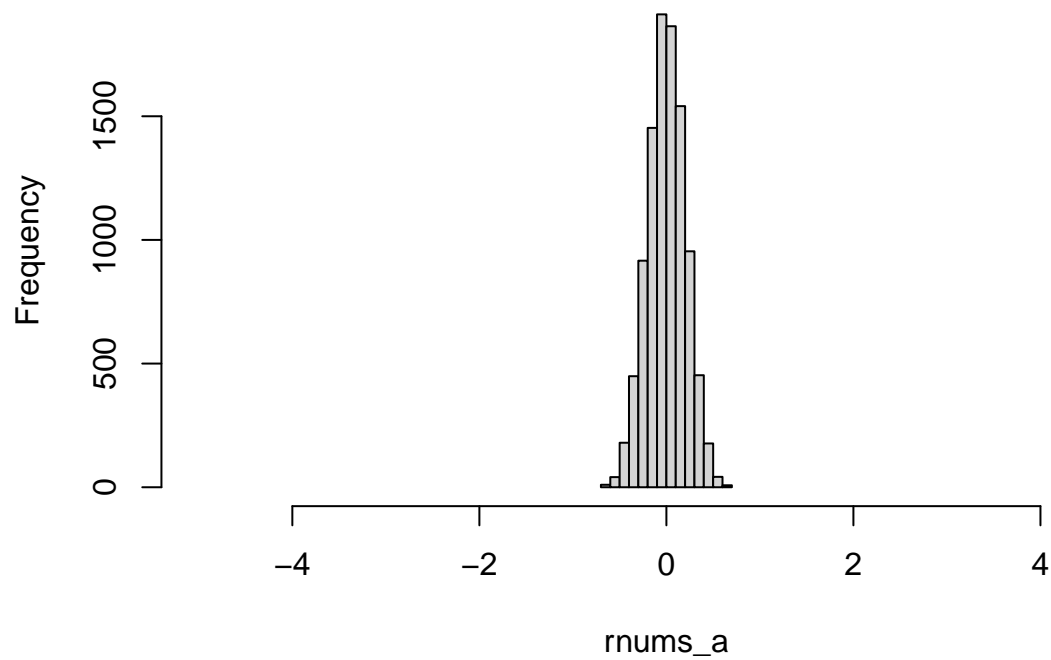
# Step 2: Run it to confirm it works.
# rnums_a

# Step 3: Create a histogram for rnums_a using the hist function.
hist(rnums_a)
```



```
# Step 4: To better see the plot differences set xlim to c(-5, 5) and plot.  
hist(rnums_a, xlim = c(-5, 5))
```

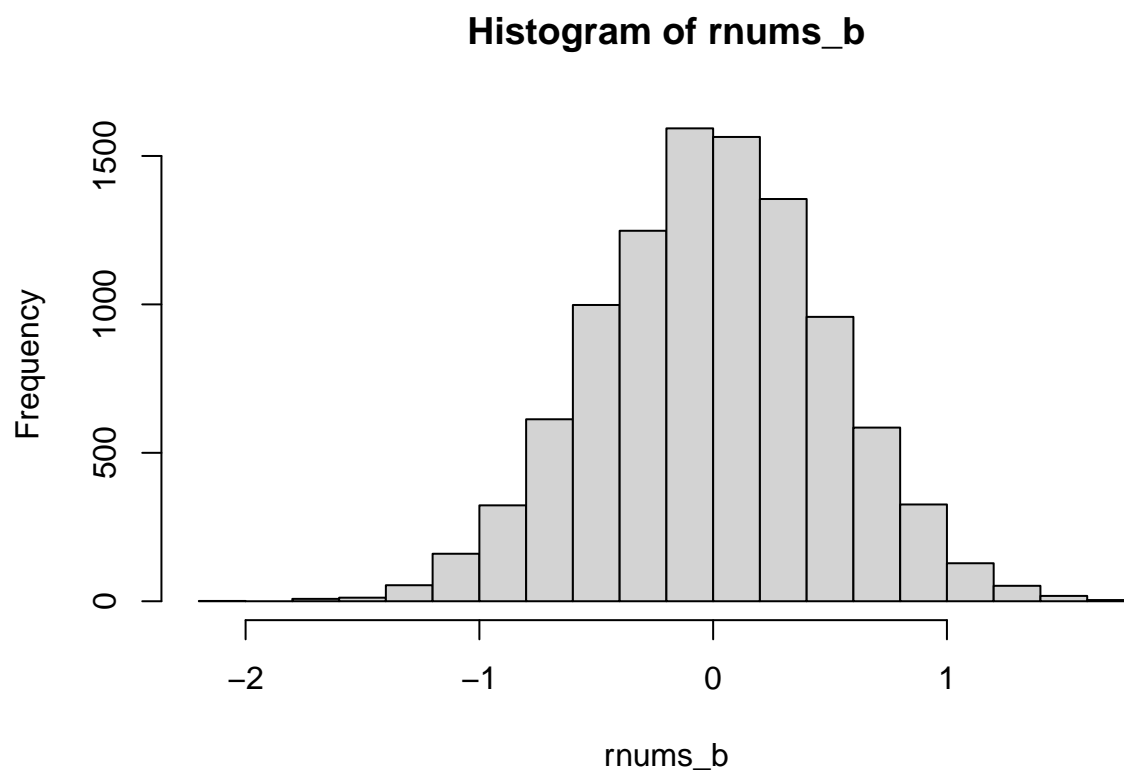
Histogram of rnums_a



```
# 2b. mean = 0, sigma = 0.5
# Step 1: Use rnorm function with n = 10000, mean = 0, sigma = 0.5 to create values rnums_b.
rnums_b = rnorm(10000, mean = 0, sd = 0.5)

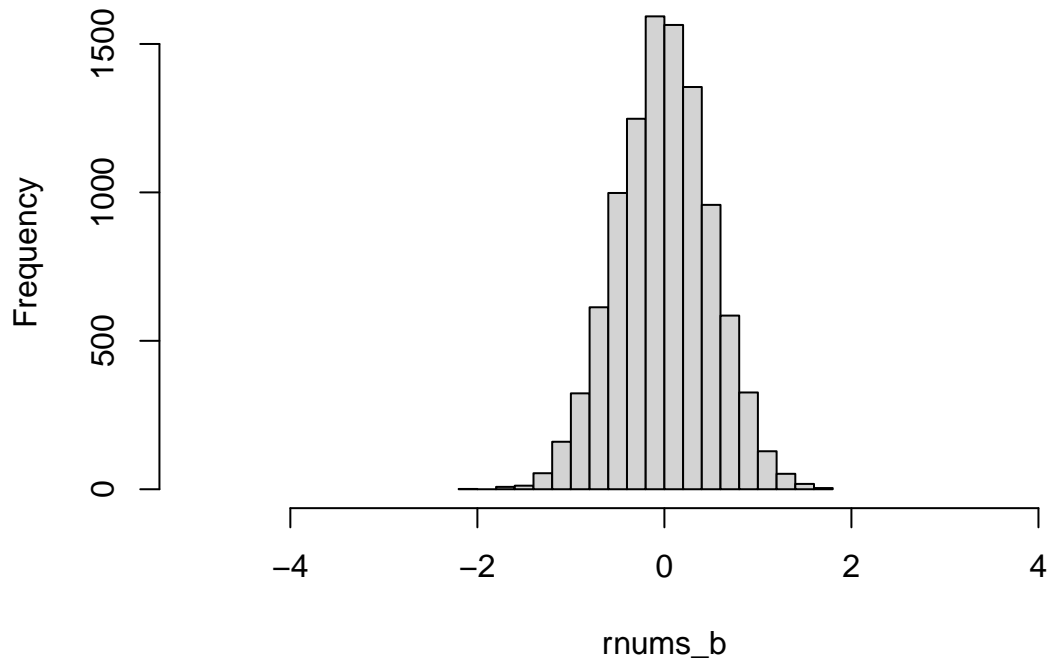
# Step 2: Run it to confirm it works.
# rnums_b

# Step 3: Create a histogram for rnums_b using the hist function.
hist(rnums_b)
```

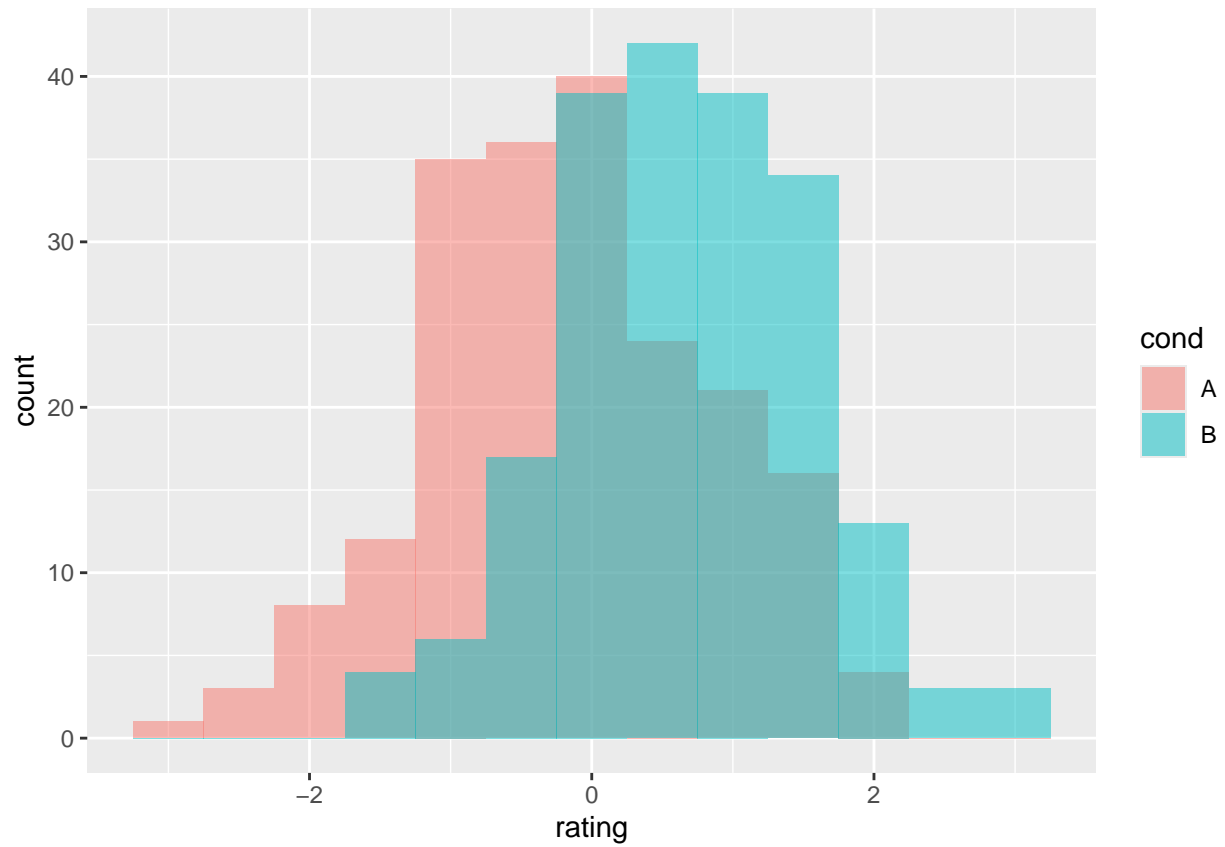


```
# Step 4: To better see the plot differences set xlim to c(-5, 5) and plot.  
hist(rnums_b, xlim = c(-5, 5))
```

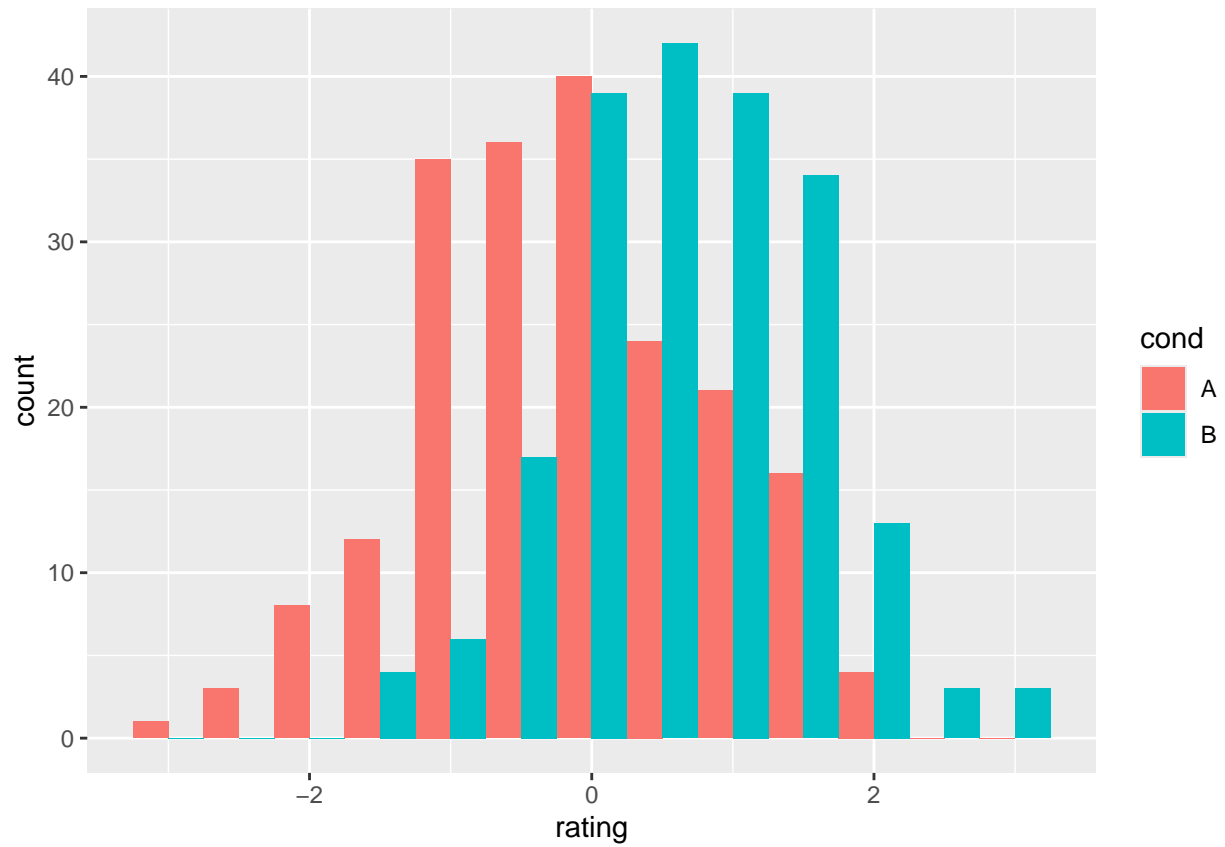
Histogram of rnums_b



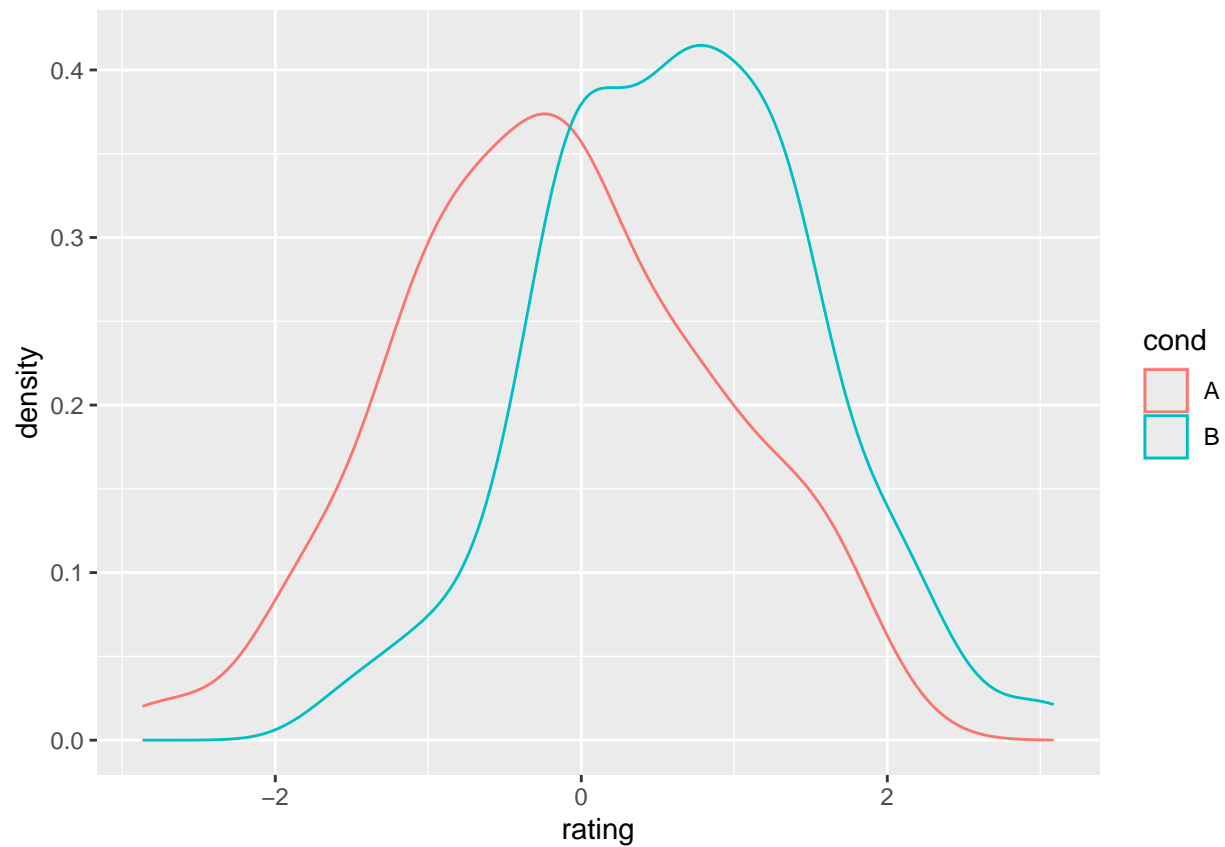
```
# -----  
# 3. Perform the steps below with "dat" dataframe which is just a sample data for you  
# to observe how each plot function ( 3b through 3e ) works. Notice that you need to  
# have ggplot2 library installed on your system. Please refer slides how to install  
# and import a library. Installation is done only once, but you need to import the  
# library every time you need it by saying library(ggplot2). Then run the following  
# commands for questions from 3a through 3e and observe how the plots are generated  
# first. (20 points)  
  
# 3a.  
# Step 1: Create a dataframe named dat.  
dat = data.frame(cond = factor(rep(c("A", "B"), each = 200)),  
                  rating = c(rnorm(200), rnorm(200, mean = .8)))  
# Step 2: View it to confirm it looks right.  
# View(dat)  
  
# 3b.  
# Create overlaid histograms for dat.  
ggplot(dat, aes(x = rating, fill = cond)) +  
  geom_histogram(binwidth = .5, alpha = .5, position = "identity")
```

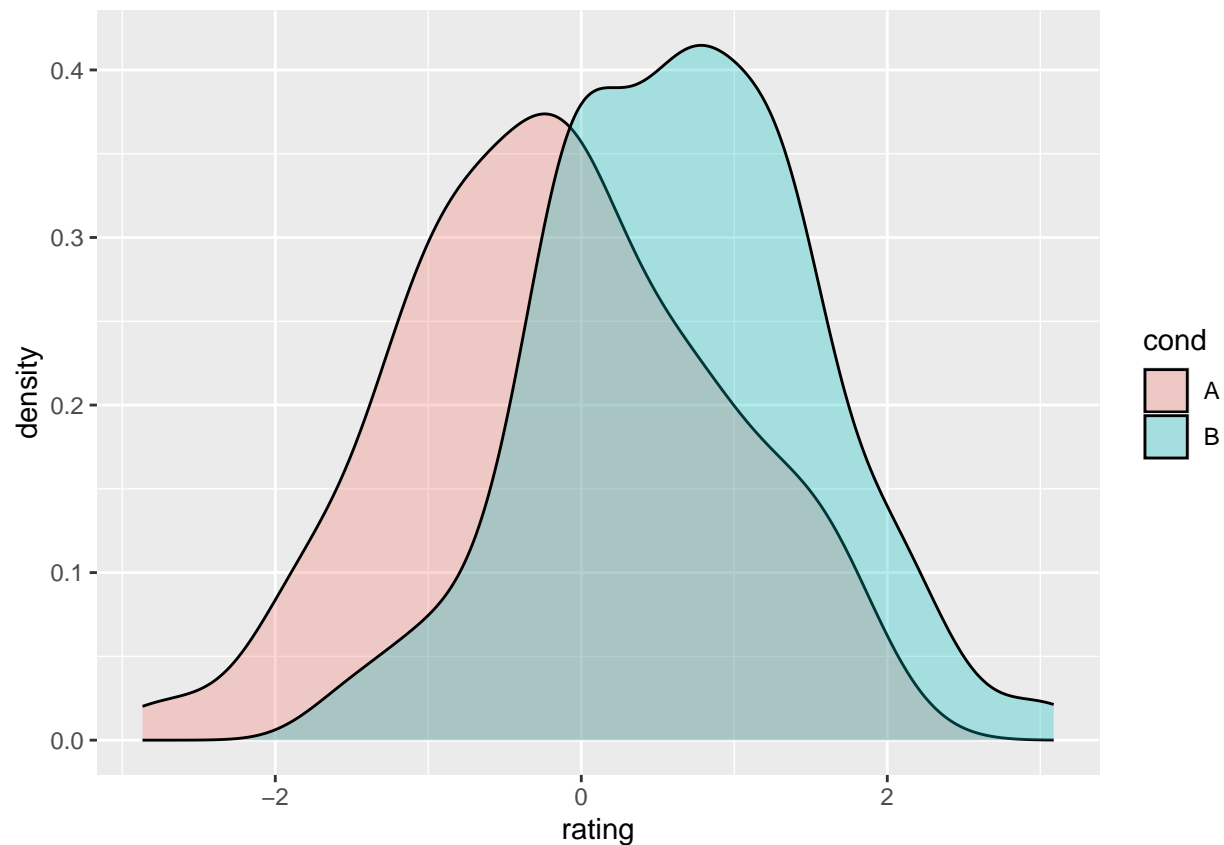
```
# 3c.  
# Create an interleaved histogram for dat.  
ggplot(dat, aes(x = rating, fill = cond)) + geom_histogram(binwidth = .5, position = "dodge")
```



```
# 3d.  
# Create density plots for dat.  
ggplot(dat, aes(x = rating, colour = cond)) + geom_density()
```



```
# 3e.  
# Create density plots with semitransparent fill for dat.  
ggplot(dat, aes(x = rating, fill = cond)) + geom_density(alpha = .3)
```

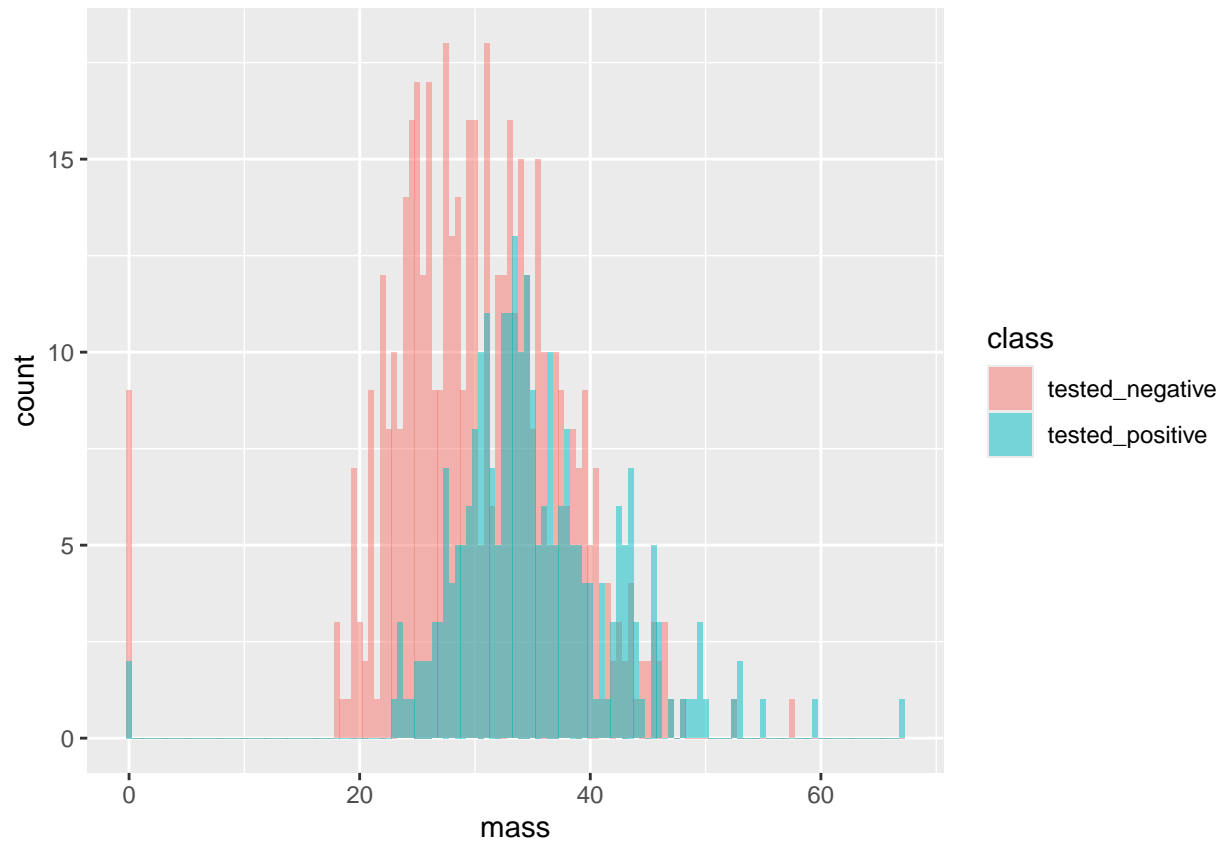


```
# 3f.
# Set the drive to where "diabetes_train.csv" is saved.
setwd("C:/Users/Chris/OneDrive/Desktop/U of M/Winter25/CSC587/from_gdrive")

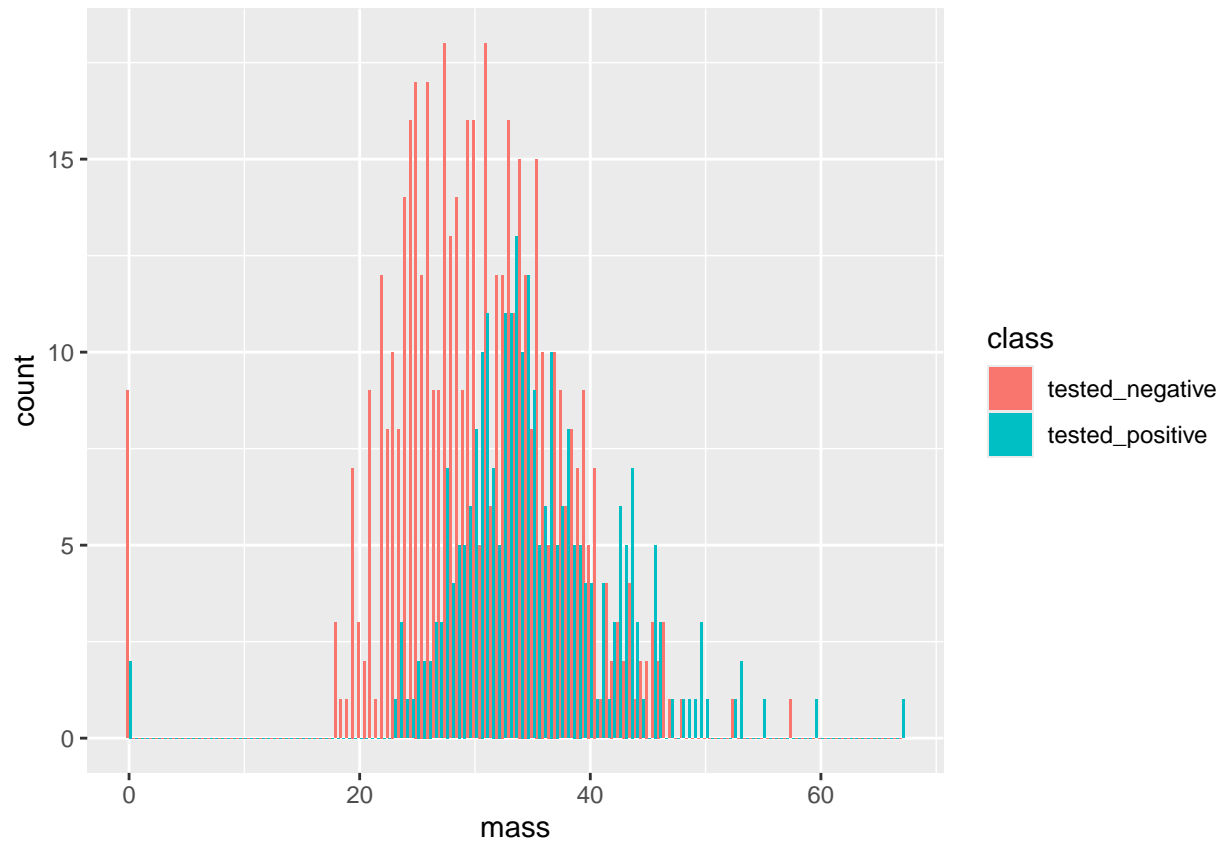
# Step 1: Read the "diabetes_train.csv" file and create a dataframe named 'diabetes'
# use commas for sep and true for header.
diabetes = read.csv("diabetes_train.csv", sep = ',', header = TRUE)

# Step 2: View it to confirm it looks right.
# View(diabetes)

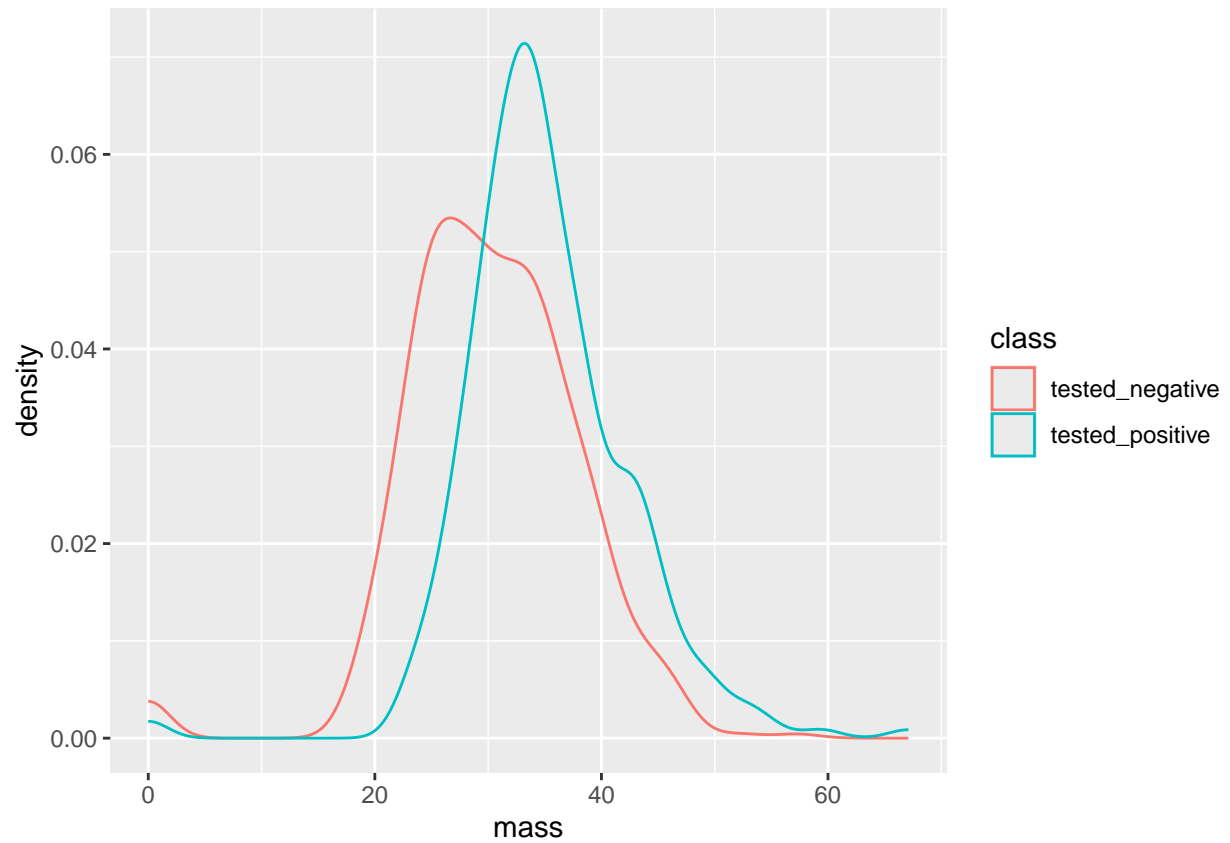
# 3f(b).
# Create overlaid histograms for diabetes.
ggplot(diabetes, aes(x = mass, fill = class)) +
  geom_histogram(binwidth = .5, alpha = .5, position = "identity")
```



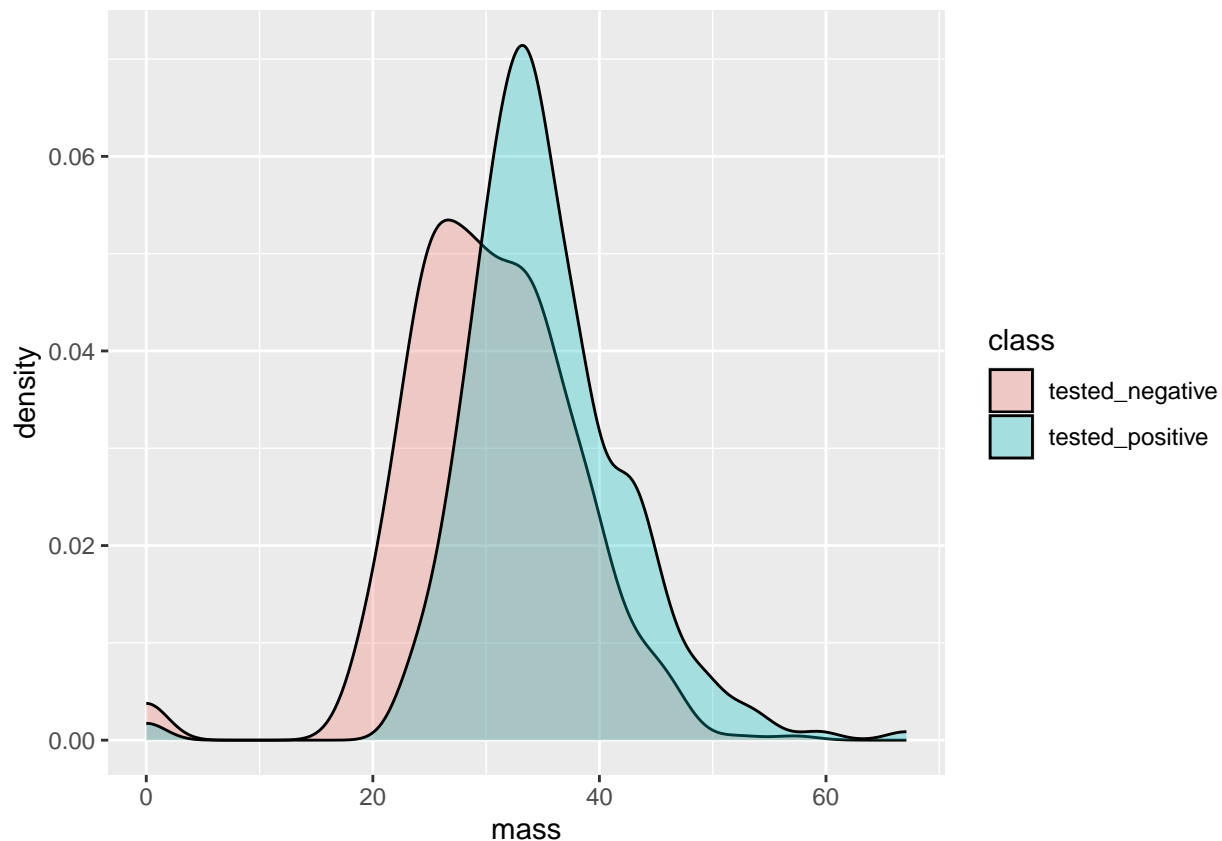
```
# 3f(c).  
# Create an interleaved histogram for diabetes.  
ggplot(diabetes, aes(x = mass, fill = class)) + geom_histogram(binwidth = .5, position = "dodge")
```



```
# 3f(d).  
# Create density plots for diabetes.  
ggplot(diabetes, aes(x = mass, colour = class)) + geom_density()
```



```
# 3f(e).  
# Create density plots with semitransparent fill for diabetes.  
ggplot(diabetes, aes(x = mass, fill = class)) + geom_density(alpha = .3)
```



```
# -----
# 4. Read the titanic.csv file from DATA folderto a variable named passengers and
# perform the following steps and explain the operation very briefly. Please make
# sure you have tidyverse installed on your system and you may specifically need
# to import the tidyr library. Otherwise, the chain of operations through "piping"
# won't work. (20 points):

# Set the drive to where "titanic.csv" is saved.
setwd("C:/Users/Chris/OneDrive/Desktop/U of M/Winter25/CSC587/from_gdrive")

# Step 1: Read the "titanic.csv" file and create a dataframe named passengers.
passengers = read.csv("titanic.csv", sep = ',', header = TRUE)
# Step 2: View it to confirm it looks right.
# View(passengers)

# 4a.
# Pass the passengers values and drop the rows with NA values and then summarize
# the Min, 1st Qu., Median, Mean, 3rd Qu., and Max of each column. .
passengers %>% drop_na() %>% summary()
```

```
##      X      PassengerId      Survived      Pclass
## Min.   : 0.0   Min.   : 1.0   Min.   :0.0000   Length:714
## 1st Qu.:221.2  1st Qu.:222.2  1st Qu.:0.0000   Class :character
## Median :444.0  Median :445.0  Median :0.0000   Mode  :character
## Mean   :447.6  Mean   :448.6  Mean   :0.4062
## 3rd Qu.:676.8  3rd Qu.:677.8  3rd Qu.:1.0000
```



```
## Max. :890.0 Max. :891.0 Max. :1.0000
## Name Sex Age SibSp
## Length:714 Length:714 Min. : 0.42 Min. :0.0000
## Class :character Class :character 1st Qu.:20.12 1st Qu.:0.0000
## Mode :character Mode :character Median :28.00 Median :0.0000
## Mean :29.70 Mean :0.5126
## 3rd Qu.:38.00 3rd Qu.:1.0000
## Max. :80.00 Max. :5.0000
## Parch Ticket Fare Cabin
## Min. :0.0000 Length:714 Min. : 0.00 Length:714
## 1st Qu.:0.0000 Class :character 1st Qu.: 8.05 Class :character
## Median :0.0000 Mode :character Median : 15.74 Mode :character
## Mean :0.4314 Mean : 34.69
## 3rd Qu.:1.0000 3rd Qu.: 33.38
## Max. :6.0000 Max. :512.33
## Embarked
## Length:714
## Class :character
## Mode :character
##
##
##
```

```
# 4b.
# Pass the passengers values and filter the values so it only displays the rows with
# male passengers.
```

```
head(passengers %>% filter(Sex == "male"))
```

```
## X PassengerId Survived Pclass Name Sex Age SibSp
## 1 0 1 0 3 Braund, Mr. Owen Harris male 22 1
## 2 4 5 0 3 Allen, Mr. William Henry male 35 0
## 3 5 6 0 3 Moran, Mr. James male NA 0
## 4 6 7 0 1 McCarthy, Mr. Timothy J male 54 0
## 5 7 8 0 3 Palsson, Master. Gosta Leonard male 2 3
## 6 12 13 0 3 Saunderson, Mr. William Henry male 20 0
## Parch Ticket Fare Cabin Embarked
## 1 0 A/5 21171 7.2500 S
## 2 0 373450 8.0500 S
## 3 0 330877 8.4583 Q
## 4 0 17463 51.8625 E46 S
## 5 1 349909 21.0750 S
## 6 0 A/5. 2151 8.0500 S
```

```
# 4c.
# Pass the passengers values and arrange them by Fare price in descending order.
```

```
head(passengers %>% arrange(desc(Fare)))
```

```
## X PassengerId Survived Pclass Name Sex Age
## 1 258 259 1 1 Ward, Miss. Anna female 35
## 2 679 680 1 1 Cardeza, Mr. Thomas Drake Martinez male 36
## 3 737 738 1 1 Lesurer, Mr. Gustave J male 35
## 4 27 28 0 1 Fortune, Mr. Charles Alexander male 19
## 5 88 89 1 1 Fortune, Miss. Mabel Helen female 23
```

```
## 6 341      342      1      1      Fortune, Miss. Alice Elizabeth female 24
##   SibSp Parch   Ticket    Fare      Cabin Embarked
## 1     0     0 PC 17755 512.3292          C
## 2     0     1 PC 17755 512.3292 B51 B53 B55          C
## 3     0     0 PC 17755 512.3292          B101          C
## 4     3     2  19950 263.0000 C23 C25 C27          S
## 5     3     2  19950 263.0000 C23 C25 C27          S
## 6     3     2  19950 263.0000 C23 C25 C27          S
```

```
# 4d.
# Pass the passengers values and create a column named FamSize which is the sum
# of the columns Parch and SibSp.
```

```
head(passengers %>% mutate(FamSize = Parch + SibSp))
```

```
##   X PassengerId Survived Pclass
## 1 0           1         0       3
## 2 1           2         1       1
## 3 2           3         1       3
## 4 3           4         1       1
## 5 4           5         0       3
## 6 5           6         0       3
```

```
##                                     Name    Sex Age SibSp Parch
## 1                                     Braund, Mr. Owen Harris   male  22     1     0
## 2 Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                                     Heikkinen, Miss. Laina female  26     0     0
## 4 Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 5                                     Allen, Mr. William Henry   male  35     0     0
## 6                                     Moran, Mr. James         male  NA     0     0
##   Ticket    Fare Cabin Embarked FamSize
## 1   A/5 21171  7.2500          S      1
## 2   PC 17599 71.2833    C85          C      1
## 3 STON/O2. 3101282  7.9250          S      0
## 4   113803 53.1000   C123          S      1
## 5   373450  8.0500          S      0
## 6   330877  8.4583          Q      0
```

```
# 4e.
# Pass the passengers values and create a column for mean Fare amount and another
# for the sum of survivors from Survived and grouped by the sex of the passengers.
```

```
head(passengers%>% group_by(Sex) %>% summarise(meanFare = mean(Fare), numSurv = sum(Survived)))
```

```
## # A tibble: 2 x 3
##   Sex    meanFare numSurv
##   <chr>    <dbl>    <int>
## 1 female    44.5      233
## 2 male     25.5      109
```

```
# -----
# 5. By using quantile(), calculate 10th, 30th, 50th, 60th percentiles of skin
# attribute of diabetes data. (10 points)

# Use the quantile function with x = diabetes$skin and c(10, 30, 50 , 60 / 100)
```

```
# to see the 10th, 30th, 50th, and 60th percentiles.  
quantile(x = diabetes$skin, probs = c(10, 30, 50, 60)/100)
```

```
## 10% 30% 50% 60%  
##    0  10  23  27
```

```
# Results: 10% 30% 50% 60%  
#           0  10  23  27
```