Data 602 - Assignment One

Due Friday, September 23, 2022 @ 11:59pm

```
library(binom)
library(car)
```

```
## Loading required package: carData
```

```
library(collapsibleTree)
library(dbplyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:dbplyr':
##
##     ident, sql
```

```
## The following object is masked from 'package:car':
##
##     recode
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(EnvStats)
```

```
##
## Attaching package: 'EnvStats'
```

```
## The following object is masked from 'package:car':
##
##     qqPlot
```

```
## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
## The following object is masked from ’package:base’:
##
##     print.default
```

```
library(ggformula)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: ggstance
```

```
##
## Attaching package: ’ggstance’
```

```
## The following objects are masked from ’package:ggplot2’:
##
##     geom_errorbarh, GeomErrorbarh
```

```
## Loading required package: scales
```

```
## Loading required package: ggridges
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
library(ggplot2)
library(gmodels)
library(htmltools)
library(ISLR)
library(knitr)
library(lawstat)
```

```
##
## Attaching package: ’lawstat’
```

```
## The following object is masked from ’package:car’:
##
##     levene.test
```

```
library(markdown)
library(mosaic)
```

```
## Registered S3 method overwritten by ’mosaic’:
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The ’mosaic’ package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
```

```
##
## Attaching package: ’mosaic’

## The following object is masked from ’package:Matrix’:
##
##     mean

## The following object is masked from ’package:scales’:
##
##     rescale

## The following object is masked from ’package:ggplot2’:
##
##     stat

## The following object is masked from ’package:EnvStats’:
##
##     iqr

## The following objects are masked from ’package:dplyr’:
##
##     count, do, tally

## The following objects are masked from ’package:car’:
##
##     deltaMethod, logit

## The following objects are masked from ’package:stats’:
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from ’package:base’:
##
##     max, mean, min, prod, range, sample, sum
```

```
library(mdsr)
library(mosaicData)
library(nycflights13)
library(olsrr)
```

```
##
## Attaching package: ’olsrr’

## The following object is masked from ’package:datasets’:
##
##     rivers
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## ------------------------------------------------------------------------------

##
## Attaching package: ’plyr’

## The following object is masked from ’package:mosaic’:
##
##     count

## The following objects are masked from ’package:dplyr’:
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
library(purrr)
```

```
##
## Attaching package: ’purrr’

## The following object is masked from ’package:plyr’:
##
##     compact

## The following object is masked from ’package:mosaic’:
##
##     cross

## The following object is masked from ’package:scales’:
##
##     discard

## The following object is masked from ’package:car’:
##
##     some
```

```
library(plotly)
```

```
##
## Attaching package: ’plotly’

## The following objects are masked from ’package:plyr’:
##
##     arrange, mutate, rename, summarise

## The following object is masked from ’package:mosaic’:
##
##     do
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(resampledata)
```

```
##
## Attaching package: 'resampledata'
```

```
## The following object is masked from 'package:carData':
##
##     Salaries
```

```
## The following object is masked from 'package:datasets':
##
##     Titanic
```

```
library(rmarkdown)
library(rpart)
library(rpart.plot)
library(rvest)
library(SDaA)
```

```
##
## Attaching package: 'SDaA'
```

```
## The following object is masked from 'package:plyr':
##
##     ozone
```

```
## The following object is masked from 'package:ggplot2':
##
##     seals
```

```
library(shiny)
library(stringi)
library(tibble)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'
## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```
library(tidyselect)
library(tinytex)
library(yaml)
library(shiny)
```

Attempt all problems below. You will be required to complete this assignment in R Notebook or R Markdown, and submit then submit a .pdf file (Preview -> Knit to PDF) to the Assignment One in Gradescope. Ensure you show all relevant work, either with a series of steps in LaTeX or with R-code/code in R chunks. Use at least four decimal places in your probability computations.

1. A recent study[1] of college graduates in the United states discovered that approximately 60% of degree holders would "change their majors if they could go back to school" and re-do their undergraduate degree. Let's presume this proportion also holds for Canadian undergraduate university degree holders.

One randomly selects two Canadians who hold an undergraduate degree. Compute the probability that

Compute the probability that

(a) both would change their undergraduate major (if they had the ability for a re-do).

(b) neither would change their undergraduate major (if they had the ability for a re-do).

(c) at least one of the two would change their undergraduate major (if they had the ability for a re-do).

(d) Suppose you are to randomly pick n-Canadians with undergraduate degrees in such a way that the probability of *at least one of them* would change their undergraduate degree is 0.95. Compute the minimum number of Canadians with undergraduate degrees you would have to randomly select. In other words, compute the *sample size* n. (Hint: $ln(a^b) = b * ln(a) \ldots$)

```
#Question 1

pchange = 0.6
pnochange = 0.4
#a
#P(X=2)
pbothchange = dbinom(2,2,pchange)
pbothchange
```

```
## [1] 0.36
```

```
#b
#P(X=0)
pbothnochange = dbinom(0,2,pchange)
pbothnochange
```

```
## [1] 0.16
```

[1] (https://www.bestcolleges.com/blog/college-graduate-majors-survey/)

```
#c
#P(X>=1) = 1 - P(X<1) = 1 - P(X=0)
#1 - dbinom(0,2,pchange)
patleastonechange = 1 - pbinom(0,2,pchange)
patleastonechange
```

```
## [1] 0.84
```

```
#d
#with sample size = n
#P(X>=1) = 0.95
#P(X>=1) = 1 - P(X=0)
#P(X=0) = 0.05 = 0.4^n
n = log(0.05)/log(0.4)
n
```

```
## [1] 3.269412
```

2. For Question 2, you are asked to create the following simulation: Toss a fair-die 1000 times then compute the sum of the 1000 tosses. For example, $\mathcal{S} = \{Toss1, Toss2, \cdots, Toss1000\}$. Then $\sum_{i=1}^{1000} Toss_i =$?

Step 1: Create a series of vectors to hold output Create an R chunk and type the R code appearing below. You can exclude the documentation.

```
#nsims = 1000  # the number of simulations
#outcome = numeric(nsims) #create empty vector that will be filled with numeric outcomes
nsims = 3000
outcome = numeric(nsims)
fivesix = numeric(nsims)
```

Step 2: Run the Simulation This step requires that we create a *loop*. There are many types of loops in R, a *while* loop, and a *for* loop. We will be using a for-loop here. Create another R chunk and type the following text (again, you do not need to type the documentation). **ENSURE** the body of the for-loop is wrapped with a open curly bracket { and a closed curly-bracet }!

```
for(i in 1:nsims){ #we are going to perform the body of the loop 1000 times
  outcome[i] = sample(c(1,2,3,4,5,6), 1, replace=FALSE)
  fivesix[i] = if (outcome[i] == 5 || outcome[i] == 6) 1 else 0#stores the outcome of the ith toss in p
  }  #close the for-loop
simresult = data.frame(outcome) #creates a data frame with two columns
head(simresult,3)  #shows the first three rows of the data frame, outcome of each die toss in column 2
```

```
##   outcome
## 1       4
## 2       6
## 3       2
```

```
tail(simresult,3)  #shows the last three rows of the data frame, die toss outcome in column 2
```

```
##      outcome
## 2998       1
## 2999       3
## 3000       2
```

```
fivesixsult = data.frame(fivesix)
head(fivesixsult,5)
```

```
##   fivesix
## 1       0
## 2       1
## 3       0
## 4       0
## 5       0
```

Step 3: Visualize the Simulation with `ggplot2`

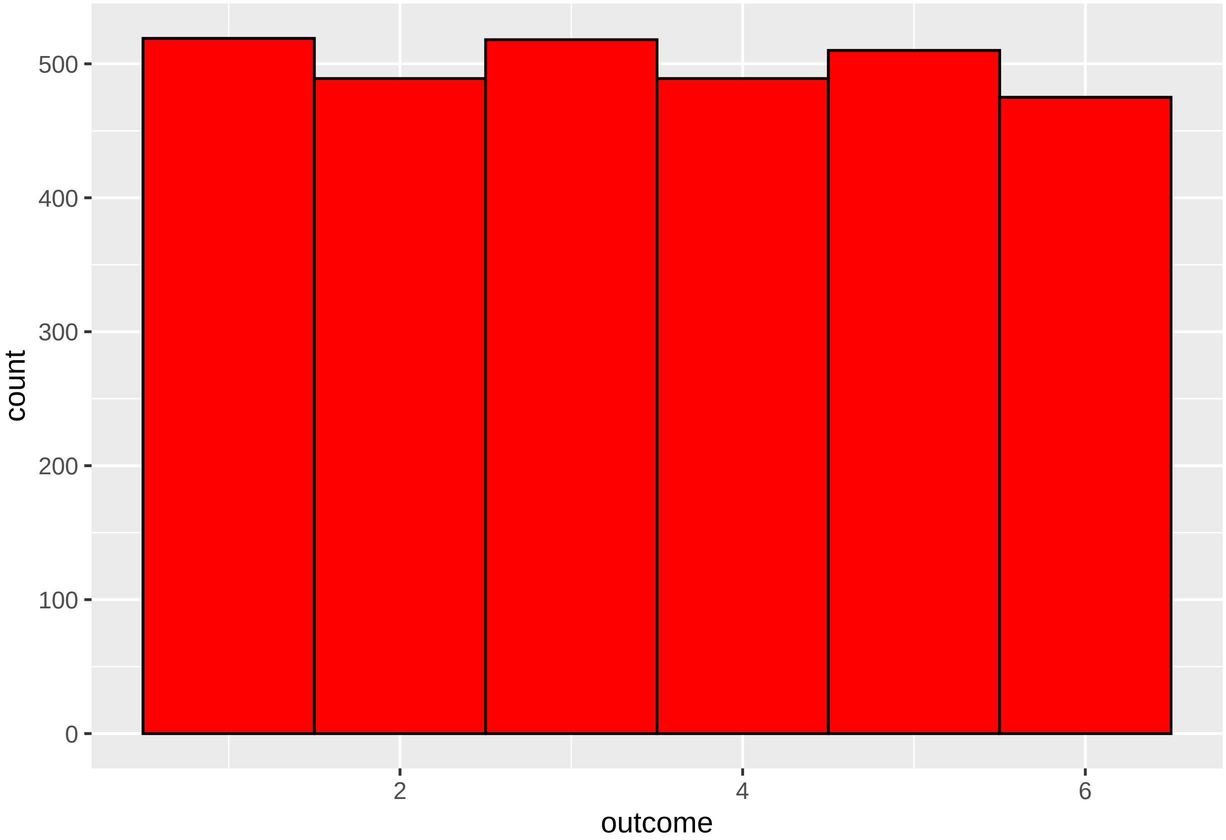
You will need the `ggplot2` package for Step 3. Just in case..

Below you will create histogram that displays the result of your 1000 simulated tosses of the fair die. After you create this data visuaulization, take particular notice of the distribution of the outcome of each number 1 through 6.

```
library(ggplot2)

sum(simresult$outcome)  #computes the sum of the 1000 tosses appearing in the outcome variable of simre
```

```
## [1] 10407
```

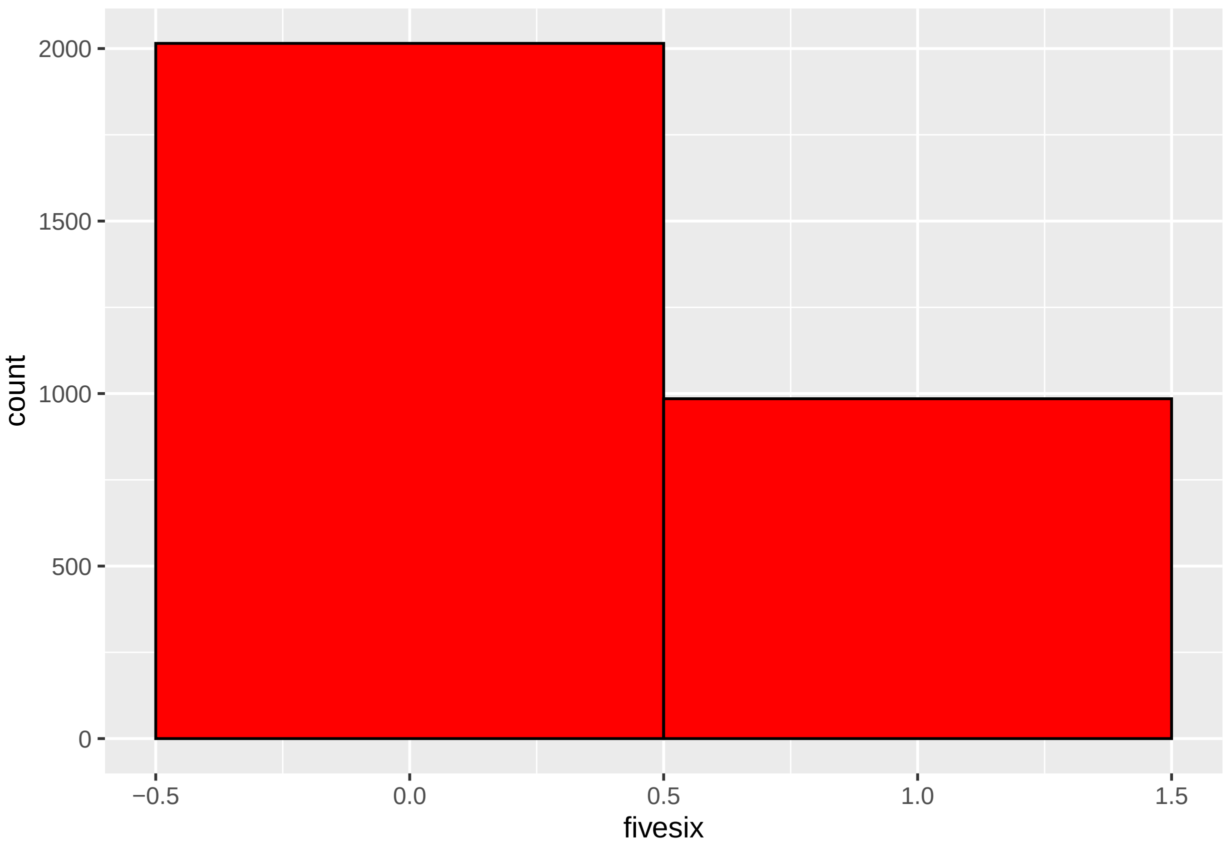
```
ggplot(data=simresult, aes(x = outcome)) + geom_histogram(binwidth=1, fill='red', col='black') #creates
```



```
sum(fivesixsult$fivesix)
```

```
## [1] 985
```

```
ggplot(data=fivesixsult, aes(x = fivesix)) + geom_histogram(binwidth=1, fill='red', col='black')
```



Step 4: Using Simulation to Compute a Probability In this step, you are going to take the simulation of a die toss done *3000* times and use the results to compute the probability that the outcome of the die toss is either a 5 or a 6. Revisit your code in your last R chunk, and *add* the following to the pre-amble of the for-loop

nsims = 3000 outcome = numeric(nsims) fivesix = numeric(nsims)

Now, within the body of the for-loop, add the code *after* the line that consists of outcome[i]:

fivesix[i] = if (outcome[i] == 5 || outcome[i] == 6) 1 else 0

This code instructs R to look at each outcome. *If* (if) the simulated die roll is equal to a 5 *or* a 6 (or is indicated by ||), then the roll is assigned a value of 1, otherwise (*else*) the roll is not counted (or counted as 0).

So, after you have finished a few simulations, here is the inquiry for Question 2: Rather than a single trial result from the outcome of a simulated die toss, suppose a trial consisted of the three die tosses. An element in the sample space $o_i = (toss1, toss2, toss3)$. Moreover, on each trial you wish to observe if the sum of the three tosses is 14 or more. For example, a $(5, 6, 3)$ outcome sums to 14 and satisifed the condition $sum \geq 14$. You wish to estimate the probability of observing a sum of 14 or more when three fair die are tossed. Run 3000 simulations. Make appropriate changes to the your last chunk of R code to compute $P(Sum \geq 14)$. You are not required to include the commands **head(dataframe, no. rows to diplay)** nor **tail(dataframe, no.rows to display)**. This is for your sake, so you get a chance to see if the output is what you want - AKA your code works.

```r
#Question 2

nsims = 3000
```

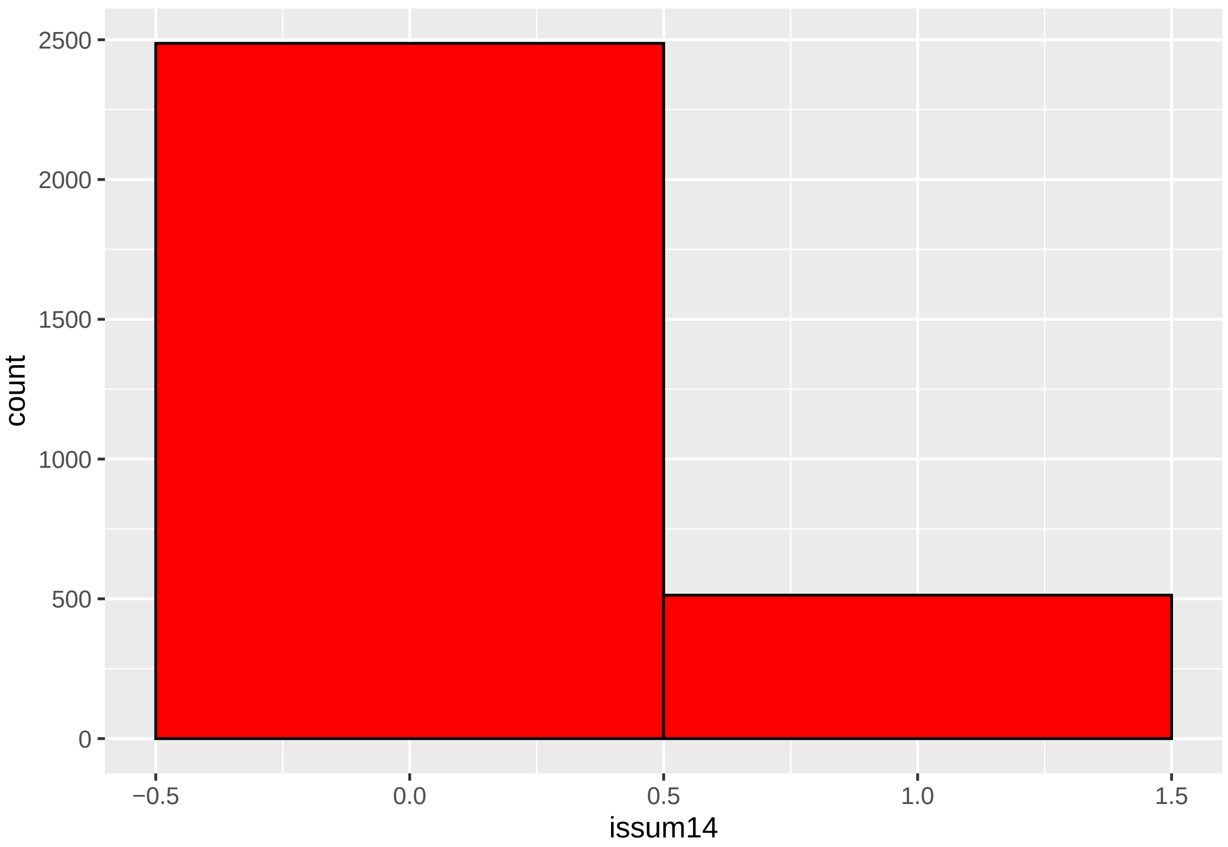
```
outcome2 = numeric(nsims)
issum14 = numeric(nsims)

for(i in 1:nsims){ #we are going to perform the body of the loop 3000 times
  outcome[i] = sum(sample(c(1,2,3,4,5,6), 3, replace=T))
  issum14[i] = if (outcome[i] >= 14) 1 else 0
  }  #close the for-loop
simresult = data.frame(outcome) #creates a data frame with two columns
head(simresult,5)
```

```
##   outcome
## 1       9
## 2       9
## 3       8
## 4      10
## 5      11
```

```
issum14result = data.frame(issum14)
head(issum14result,10)
```

```
##    issum14
## 1        0
## 2        0
## 3        0
## 4        0
## 5        0
## 6        0
## 7        0
## 8        0
## 9        1
## 10       0
```

```
ggplot(data=issum14result, aes(x = issum14)) + geom_histogram(binwidth=1, fill='red', col='black')
```



```
t <- table(issum14)
t
```

```
## issum14
##    0    1
## 2487  513
```

```
numofbi <- t[names(t) == 1]
numofsm <- t[names(t) == 0]
numofbi
```

```
##   1
## 513
```

```
numofsm
```

```
##    0
## 2487
```

```
psumbi14 = numofbi/nsims
psumbi14
```

```
##     1
## 0.171
```

```
#probability of observing a sum of 14 or more when three fair die are tossed is 0.1637
```

3. An abbreviated deck of 20 cards consists of four suits ($\heartsuit, \diamondsuit, \spadesuit, \clubsuit$) and the following denominations (10, Jack, Queen, King, Ace). You pick at random five cards, or a ‘hand’, without replacement.

(a) Compute the probability your hand will consist of a 10, Jack, Queen, King, and Ace of the *same suit*. One example of such a hand is: $(10\heartsuit, J\heartsuit, Q\heartsuit, K\heartsuit, Ace\heartsuit)$.

```
#Question 3
phandofsamesuit = (choose(4,1) * choose(5,5)) / choose(20,5)
phandofsamesuit
```

```
## [1] 0.0002579979
```

(b) Compute the probability that you get a three-of-a-kind. For example, (10♡, 10♢, 10♠, J♣, K♡).

```
p3ofkind = (choose(5,1)*choose(4,2)*choose(4,3)*choose(4,1)*choose(4,1))/choose(20,5)
p3ofkind
```

```
## [1] 0.123839
```

(c) Compute probability that one observes two Aces and two ♢’s.

```
#P( A = 2 non-diamondsuit Aces and 2 non-Aces diamondsuits and 1 non-Ace non-diamondsuit card)
pa = (choose(3,2)*choose(4,2)*choose(12,1))/choose(20,5)

#P( B = 1 diamond ace, 1 non-diamondsuit ace, 1 non-ace diamondsuit, 3 non diamondsuit non aces cards)

pb = (choose(1,1)* choose(3,1) * choose(4,1) *choose(12,3))/ choose(20,5)

# P( two aces and two diamondsuit from a hand) = P(A) + P(B)
p2a2d = pa + pb
p2a2d
```

```
## [1] 0.1842105
```

4. An oil and gas executive needs to fly from Calgary, Alberta (airport code YYC) to Washington-Dulles (airport code IAD) to attend a meeting with lobbyists about the building of a certain pipeline. Because there is no direct flight from YYC to IAD, this traveler has fly from YYC to a different city, then connect with a flight to IAD. The traveler has airline options. Airline AA will connect through Dallas, Airline UA will connect through Chicago, or Airline D which connects through Minneapolis-St.Paul. Taking into their past experiences with flying with the three airlines in question, this executive hints that the probability of flying with Airline AA is 0.15. The probability they will fly with Airline D is three times more than the probability of flying with Airline UA. Historical data has shown that 15% of passengers who fly with Airline AA miss their connecting flights in Dallas. Similarly, 10% of Airline D passengers and 30% of Airline UA passengers miss their connecting flights.

The executive has called the office of the lobby-group to say they have missed their connecting flight. Compute the probability that the executive called from Chicago (or is flying Airline UA).

```
#Question 4
#P(AA) = 0.15, P(D) = 3*P(UA)
#P(M|AA)= 0.15, P(M|D)= 0.1, P(M|UA)= 0.3
#P(UA|M) = ?
#P(AA) + P(D) + P(UA) = 1
#P(D) + P(UA) = 0.85
paa = 0.15
pua = 0.85 / 4
pd = 3*pua
pua
```

```
## [1] 0.2125
```

```
pd
```

```
## [1] 0.6375
```

```
#Events AA, D, UA are mutually exclusive

#P(UA|M) = (P(M|UA) * P (UA)) / P(M)

airlineprobs = c(paa, pua, pd) #create a vector of airline probabilities for AA, UA, and D respectively
missprobs = c(0.15, 0.3, 0.1) #create a vector of conditional probability miss connecting
probmiss = sum(airlineprobs*missprobs) #compute P(Miss connecting)
probmiss #return P(M)
```

```
## [1] 0.15
```

```
probuagivenmiss = (0.3 * pua)/probmiss  #compute P(UA|M) = (P(M|UA) * P (UA)) / P(M)
probuagivenmiss #return the answer
```

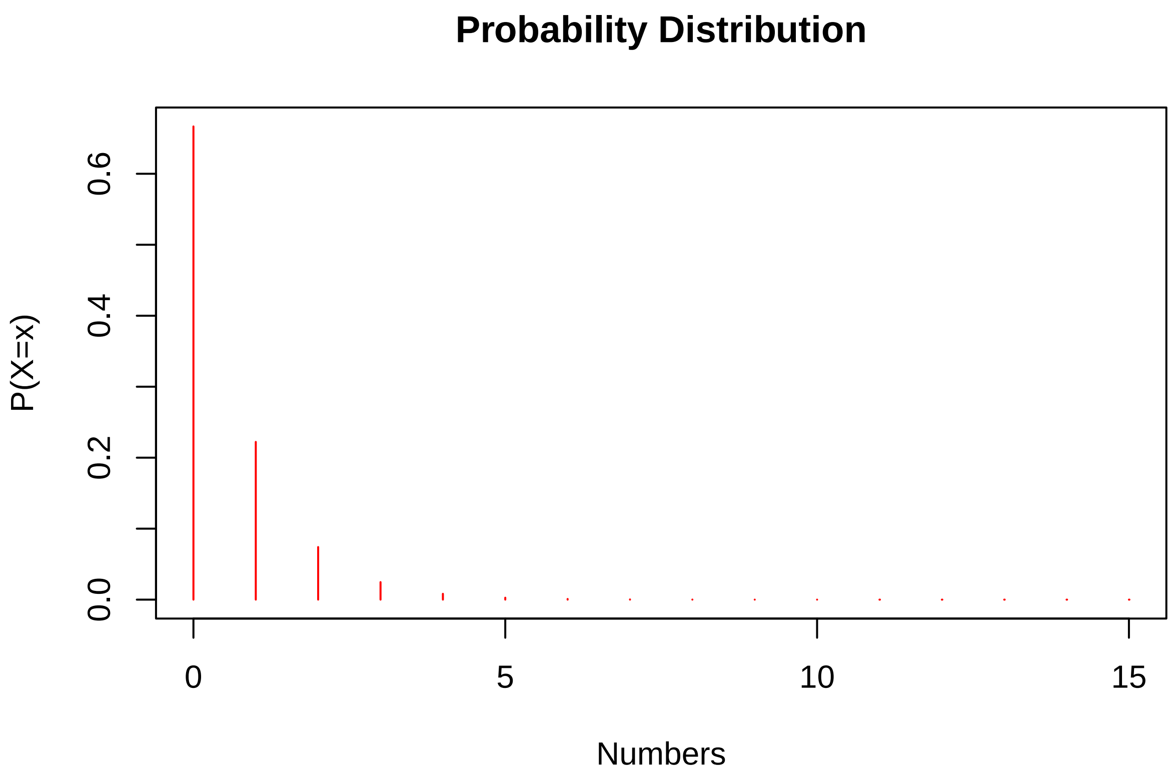
```
## [1] 0.425
```

5. A random variable X has the following probability distribution function

$$P(X = x) = \frac{2}{3^{x+1}} \qquad x = 0, 1, 2, \cdots$$

(a) Using R Studio, create a display that shows the probability distribution of this particular random variable X. Refer to the various code provided for examples appearing in both Probability Module 4 and Review Exercise 5 from Thursday, September 8th. For values of x, use **xvalues = 0:15**.

```
#Question 5
q5pdf = function(x){

  2/(3**(x+1))
}
xvalues = 0:15
options(scipen=999)  #included so the P(X=x)s are not in scienfic notation
probx = q5pdf(xvalues)  #computes P(X=x) for each value
plot(xvalues, probx , xlab="Numbers", ylab="P(X=x)", main="Probability Distribution", type="h", col='re
```



(b) How likely is it to observe values beyond 3? Compute.

```
#P(X>3) = 1 - P(X<=3) = 1 -(P(X=0) + P(X=1) + P(X=2) + P(X=3))
probbeyond3 = 1 - sum(q5pdf(0:3))
probbeyond3
```

```
## [1] 0.01234568
```

```
#the probability of observe valuses beyond 3 is 0.0123
```

(c) Compute the mean or expected value of X, $E(X)$ or μ_X. (Hint: In computing $E(X)$, change the upper limit on xvalues from 15 to 100…)

```
xq5 = c(0:100)
pxq5 = q5pdf(xq5)
expectxq5 = sum(xq5*pxq5)
expectxq5
```

```
## [1] 0.5
```

```
#the expected value of X is 0.5
```

(d) Compute the standard deviation of X, $SD(X)$ or σ_X. (use the same values of x as you did in part (c))

```
#SD(X) = sqrt(Var(X))
#Var(X) = E(X^2) - E(X)^2
varxq5 = sum(xq5^{2} * pxq5) - expectxq5^{2}
varxq5
```

```
## [1] 0.75
```

```
sdq5 = sqrt(varxq5)
sdq5
```

```
## [1] 0.8660254
```

```
#standard deviation of X is 0.8660
```

(e) Consider the interval $(\mu_X - \sigma_X, \mu_X + \sigma_X)$. Compute $P(\mu_X - \sigma_X < X < \mu_X + \sigma_X)$.

```
#mux - sigmax, mux + sigmax

q5a = expectxq5 - sdq5 #mux - sigmax
q5b = expectxq5 + sdq5 #mux + sigmax
q5a
```

```
## [1] -0.3660254
```

```
q5b
```

```
## [1] 1.366025
```

```
#P(mux - sigmax < X < mux + sigmax) = P(X = 0) + P (X = 1)
probe = sum(q5pdf(0:1))
probe
```

```
## [1] 0.8888889
```

```
#P(mux - sigmax < X < mux + sigmax) = 0.8888889
```

6. Ipsos-Reid[2] released the results of a poll taken during the first week of February 2022 and found that 40% of Canadians between the ages of 18 and 34 “believe the people involved in the trucker conveys/protests were doing was wrong and do not deserve any sympathy.”

You are to randomly pick $n = 50$ Canadians between the ages of 18 and 34 (inclusive), and ask each their sentiments. You observe that 35 of them believe that people involved in the trucker convey/protests “deserve no sympathies’.

(a) How likely is this outcome? Compute.

[2] (https://www.ipsos.com/en-ca/news-polls/nearly-half-say-they-may-not-agree-with-trucker-convoy)

```
probq6a = dbinom(35, 50, 0.4)
probq6a
```

```
## [1] 0.00001249428
```

```
#choose(50,35) * 0.4^35 * 0.6^15
#the probability of this outcome is 0.00001249428
```

(b) From your computation in part (a), what do you think of the “40% statistic” quoted above is accurate? Ensure you use probability theory in your explanation.

```
#I think the "40% statistic" quoted above is accurate.
#The sample size of part a is relatively small which increases the margin of error due.
#In Ipsos-Reid's survey, a sample of 1,000 Canadians aged 18+ was interviewed.
#I could say that the sample size between the ages of 18 and 34 in the survey would be much bigger than
#A large sample size will avoid misleading statistics.
```

(c) Suppose you are to randomly inspect n-Canadians aged 18 to 34 on this issue until you find the 10th to say “no sympathies” for people involved in the trucker conveys/protests. Compute the probability that $n = 30$.

```
pq6c = choose(29,9) * 0.4^10 * 0.6^20
pq6c
```

```
## [1] 0.03839513
```

7. You and four friends decide to play “Odd Person Out”. In this game, the five of you each toss a fair coin. The person who throws the *odd outcome* has to pay for the next round of drinks/coffee/kombutcha/whatever-you-all-fancy. For example, if one person flips a head while the other four flips tails, then the person who flipped the head has to pay for all five, and vice versa. Should such an outcome not occur, everyone flips again *until* the “odd person out” occurs. Presuming all five toss a fair coin, the random variable X that counts the number of tosses needed to observe “odd person out” is given by

$$P(X = x) = (0.6875)^{x-1}(0.3125) \qquad x = 1, 2, 3, 4, \cdots$$

It has taken 10 rounds to observe “odd person out”, or $X = 10$. Did it take more trials than expected to observe “odd person out” or less? Ensure you incorporate course content in your explanation.

```
q7 = function(x){
  0.6875^(x-1) * 0.3125
}

xq7 = c(1:100)
pxq7 = q7(xq7)
expectxq7 = sum(xq7*pxq7)
expectxq7
```

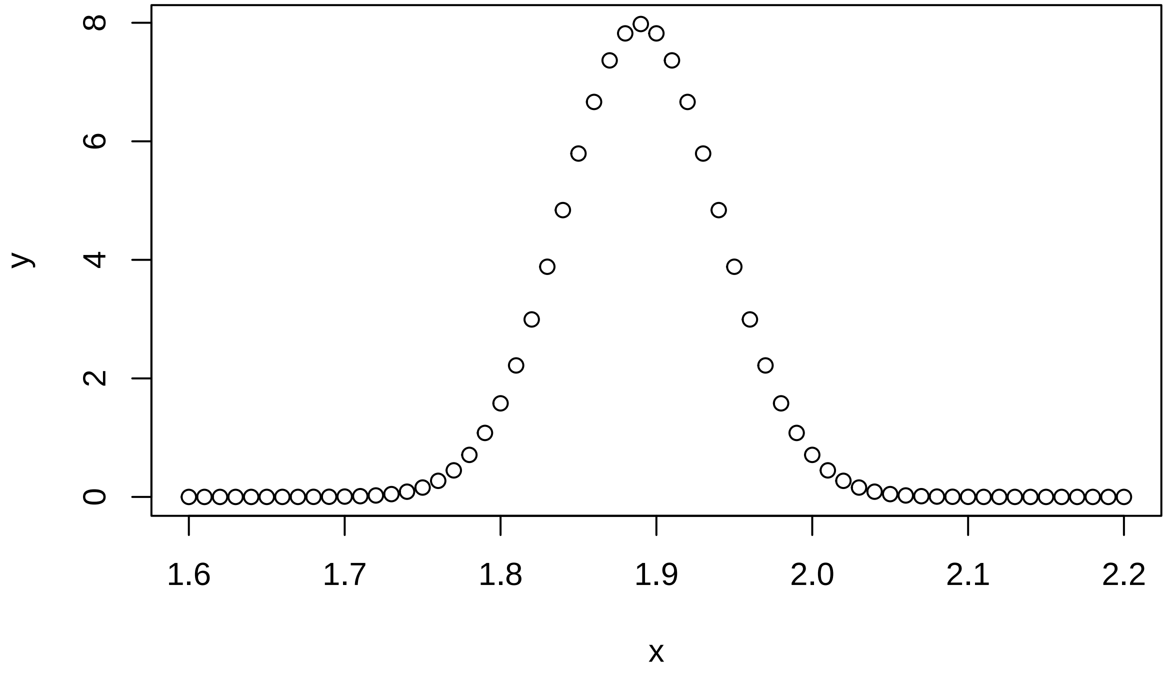
```
## [1] 3.2
```

```
# The Expected Value of the given probability mass function is 3.2,
#which means that we should expect the to observe "odd person out" in 4th toss.
#In the given scenario, It has taken 10 rounds to observe "odd person out".
#Thus, I would say it take more trials than expected to observe "odd person out"
```

8. In a certain beverage manufacturer's factory, an automated soft-drink filling machine is to fill 2-litre bottles with product, the amount of soft-drink slightly varying from one 2-litre bottle to the next in according with a Normal probability model with a mean of $\mu = 1.89$ litres and a standard deviation of $\sigma = 0.05$ litres.

(a) You are to randomly pick a 2-litre bottle off the production line and measure its contents. Compute the probability that the amount of soft-drink dispensed into this bottle is between 1.83 and 1.91 litres.

```
x = seq(1.6, 2.2, by=0.01)

y = dnorm(x, mean = 1.89 , sd = 0.05)

# Plot the graph.
plot(x, y)
```



```
#P(1.83 < X < 1.91) = P(X < 1.91) - P(X < 1.83)

probq8a = pnorm(1.91, mean = 1.89 , sd = 0.05) - pnorm(1.83, mean = 1.89 , sd = 0.05)
probq8a
```

```
## [1] 0.5403521
```

(b) Find the 90th-percentile and interpret is meaning in the context of these data.

```
percentile90th = qnorm(0.90, mean = 1.89 , sd = 0.05)
percentile90th
```

```
## [1] 1.954078
```

```
#The 90th-percentile means that there is 90% of observations below it.
#In this case, the 90th-percentile is 1.95.
#It indicates that 90% of observation fall below 1.954078. i,e smaller than 1.954078 liter.
```

(c) What proportion of *all* 2-litre bottles will be filled to overflow?

```
#P(X>2) = 1 - P(X<=2)
proportionoverflow = 1 - pnorm(2, mean = 1.89 , sd = 0.05)
proportionoverflow
```

```
## [1] 0.01390345
```

```
#pnorm(2, mean = 1.89 , sd = 0.05, lower.tail = F)

#1.39% of all 2-litre bottles will be filled to overflow.
```

(d) You are to randomly pick 50 2-litre bottles for inspection, measuring the amount of product dispensed into each of the bottles. Compute the probability that between 5 and 10 of these bottles will have less than 1.85 litres of soft-drink.

```
#Calculate the probability of having less than 1.85 litres.
#P(X<1.85)
probless = pnorm(1.85, mean = 1.89 , sd = 0.05)
probless
```

```
## [1] 0.2118554
```

```
#Calcualate the probability that between 5 and 10 of these bottles will have less than 1.85 litres for
plessberween = sum(dbinom(5:10, 50, probless))
plessberween
```

```
## [1] 0.4892489
```

```
#the probability that between 5 and 10 of these bottles will have less than 1.85 litres of soft-drink i
```

9. The data file GSS2002.csv consists of data resulting from the General Social Suvery (GSS) that tracks various demographic, characteristics, and views on social and political issues since the early 1970s. This file can be imported into R with the following command

```
gss = read.csv("http://people.ucalgary.ca/~jbstall/DataFiles/GSS2002.csv")
head(gss,5)
```

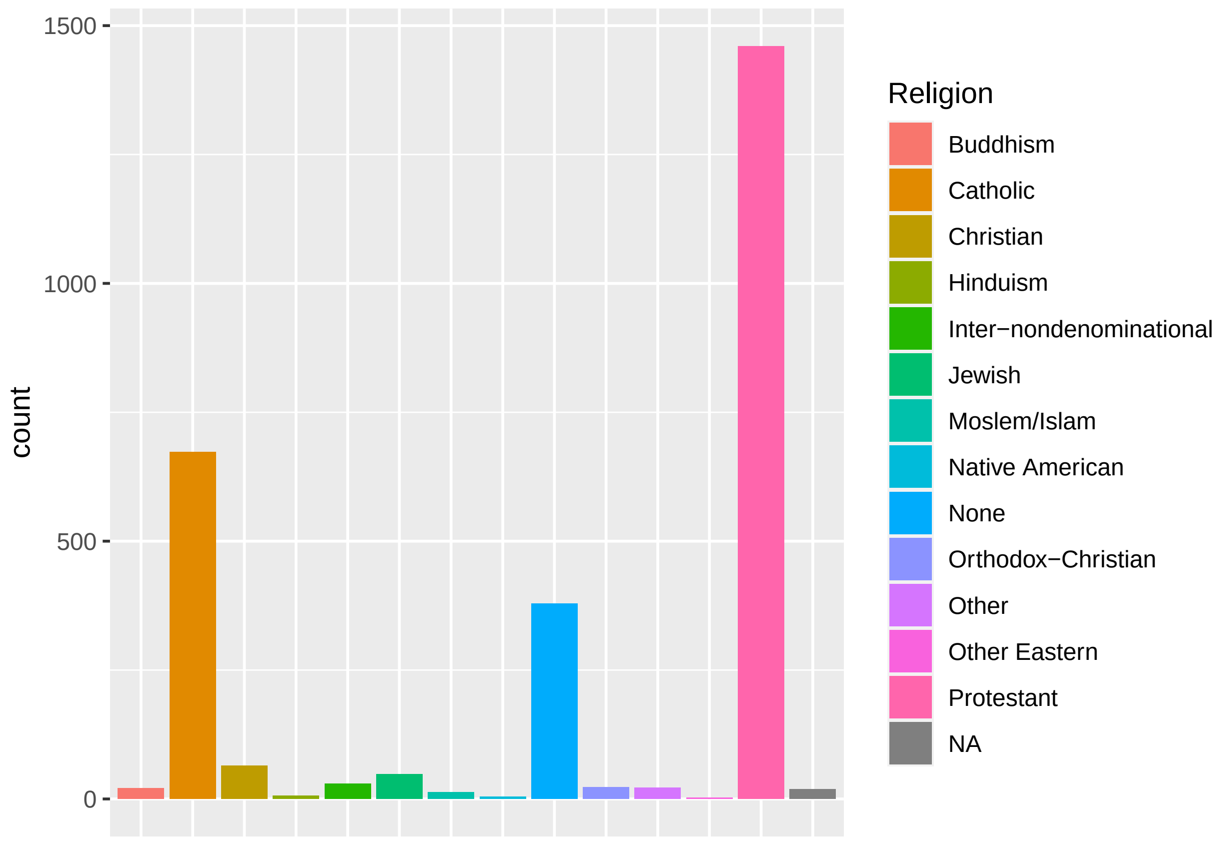
```
##   ID        Region Gender  Race Education   Marital                Religion
## 1  1 South Central Female White        HS  Divorced Inter-nondenominational
## 2  2 South Central   Male White Bachelors   Married              Protestant
## 3  3 South Central Female White        HS Separated              Protestant
## 4  4 South Central Female White   Left HS  Divorced              Protestant
## 5  5 South Central   Male White   Left HS  Divorced              Protestant
##          Happy      Income      PolParty     Politics Marijuana DeathPenalty
## 1 Pretty happy 30000-34999    Strong Rep Conservative      <NA>        Favor
## 2 Pretty happy 75000-89999   Not Str Rep Conservative Not legal        Favor
## 3         <NA> 35000-39999    Strong Rep         <NA>      <NA>         <NA>
## 4         <NA> 50000-59999 Ind, Near Dem         <NA>      <NA>         <NA>
## 5         <NA> 40000-49999           Ind         <NA>      <NA>         <NA>
##   OwnGun GunLaw SpendMilitary  SpendEduc    SpendEnv    SpendSci Pres00
## 1     No  Favor    Too little Too little About right About right   Bush
## 2    Yes Oppose   About right Too little About right About right   Bush
## 3   <NA>   <NA>          <NA>       <NA>        <NA>        <NA>   Bush
## 4   <NA>   <NA>   About right Too little  Too little  Too little   <NA>
## 5   <NA>   <NA>          <NA>       <NA>        <NA>        <NA>   <NA>
##   Postlife
## 1      Yes
## 2      Yes
## 3     <NA>
## 4     <NA>
## 5     <NA>
```

This data set consists of a various categorical variables. You can run the **head(gss, # of rows to see)** command to inspect the different variable names. Or, the command **columnnames(gss)** will return the names of the different columns/variables. To determine the different values that are possible on a certain categorical variable, the command **levels(dataframename$variablename)** will return the different values.

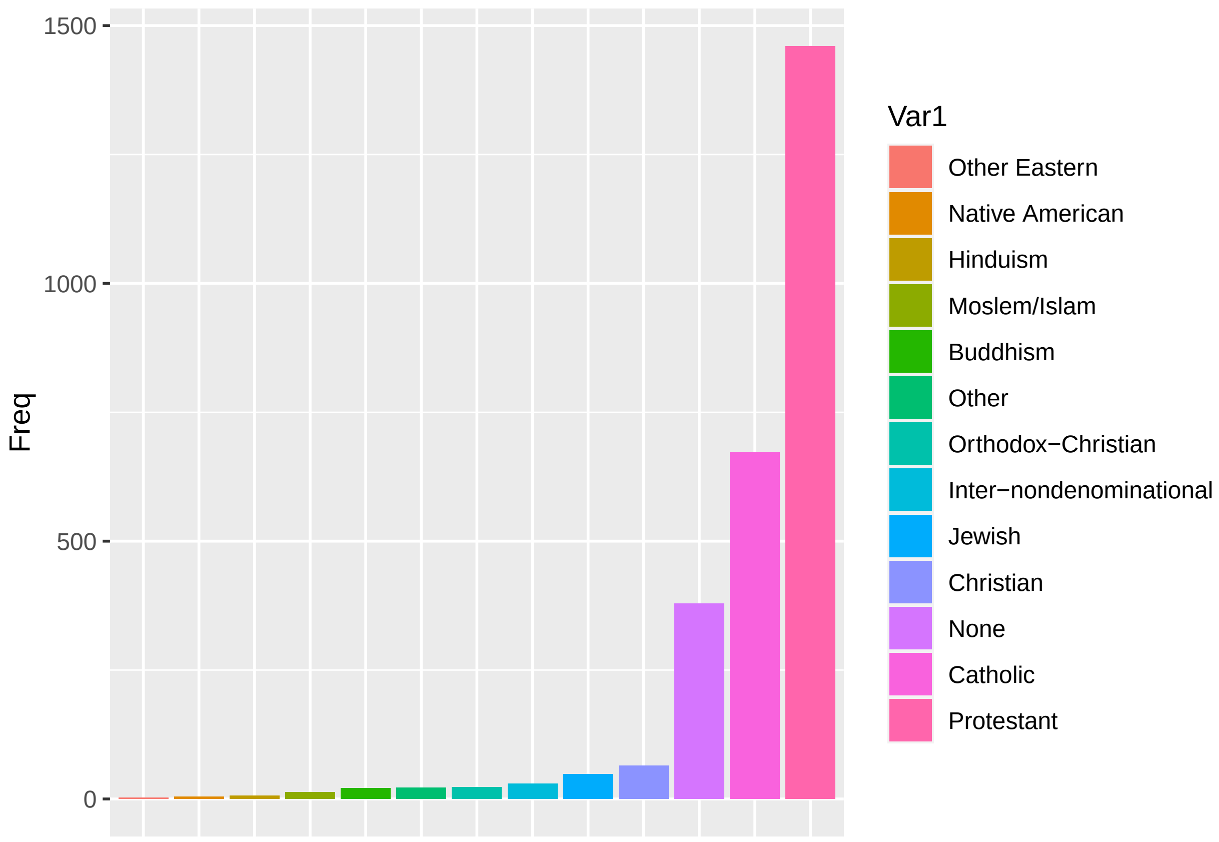
A Brief Tutorial on Bar Graphs

For example, consider the variable "Religion" which provides income classes that each respondent falls into. A bar-graph of this variable is created below using the code presented below. I have included a "non-sorted" bar graph as a comparater.

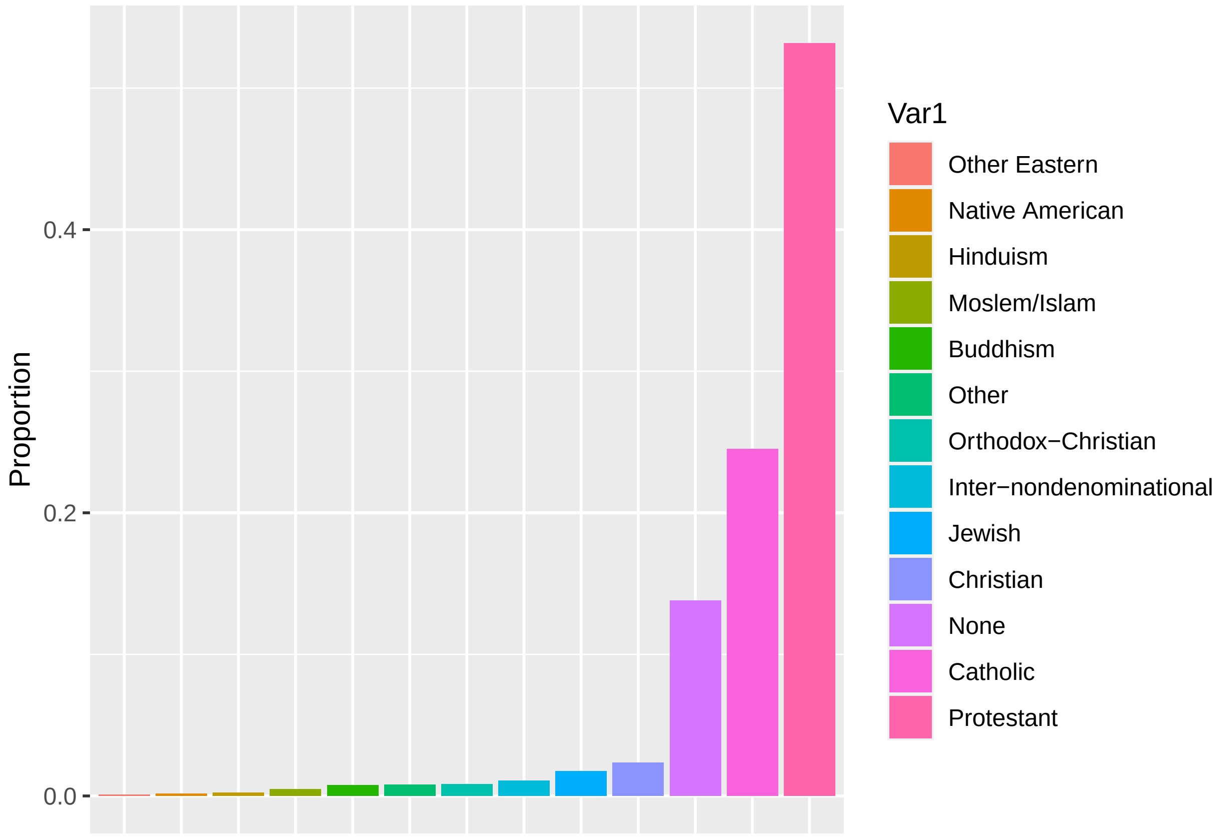
```
ggplot(data=gss, aes(x = Religion, fill=Religion)) + geom_bar(position="dodge", na.rm=TRUE) + theme(axi
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```



```
counts = as.data.frame(sort(table(gss$Religion)))
# head(counts, 4)
ggplot(data=counts, aes(x=Var1, y=Freq, fill=Var1)) + geom_bar(stat="identity") + theme(axis.title.x=ele
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```

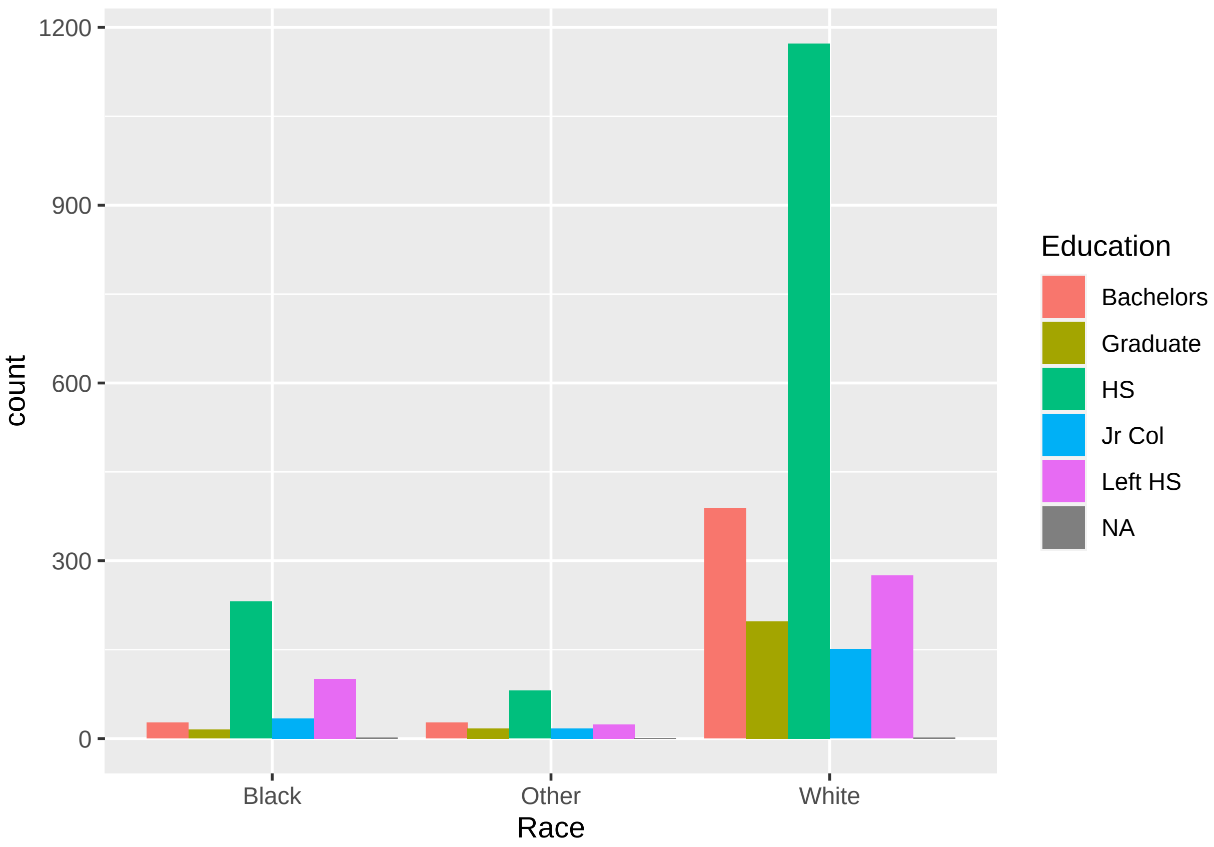


```
reliprop = counts$Freq/sum(counts$Freq) #converts counts to proportions
counts1 = data.frame(counts, reliprop) #create a new data frame adding reliprob variable to counts
# head(counts1, 4)
ggplot(data=counts1, aes(x=Var1, y=reliprop, fill=Var1)) + geom_bar(stat="identity") + ylab("Proportion
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```



Another type of bar graph shows a distribution of responses for a certain level of a categorical variable. Consider the bar graph below, which was produced with the code:

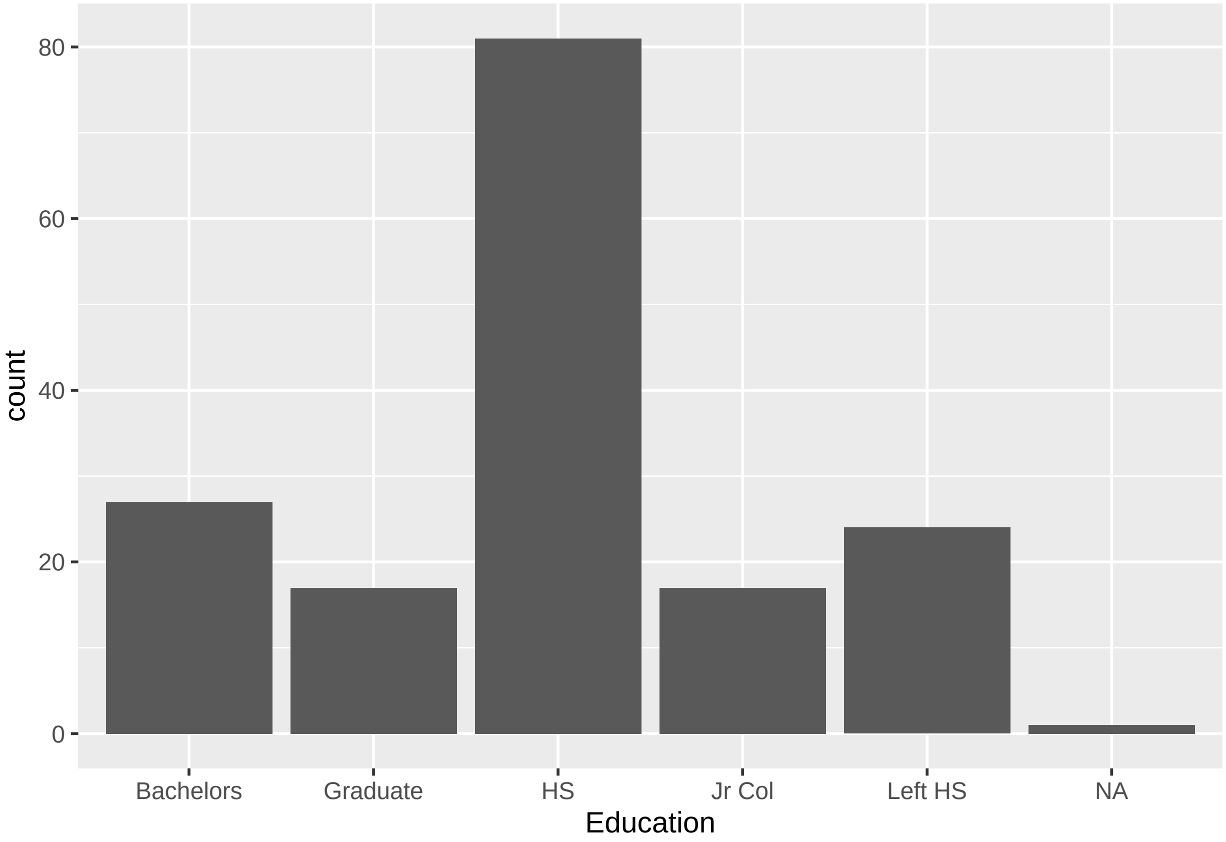
```
ggplot(data=gss, aes(x = Race, fill=Education)) + geom_bar(position = "dodge", na.rm=TRUE)
```



From above one can see the distribution of education level amongst the three different types of races data was collected on in the 2002 General Social Survey.

We can also parse out the data and look at bargraph for a certain race:

```
ggplot(data=filter(gss, Race == "Other"), aes(x = Education)) + geom_bar(na.rm=TRUE)
```

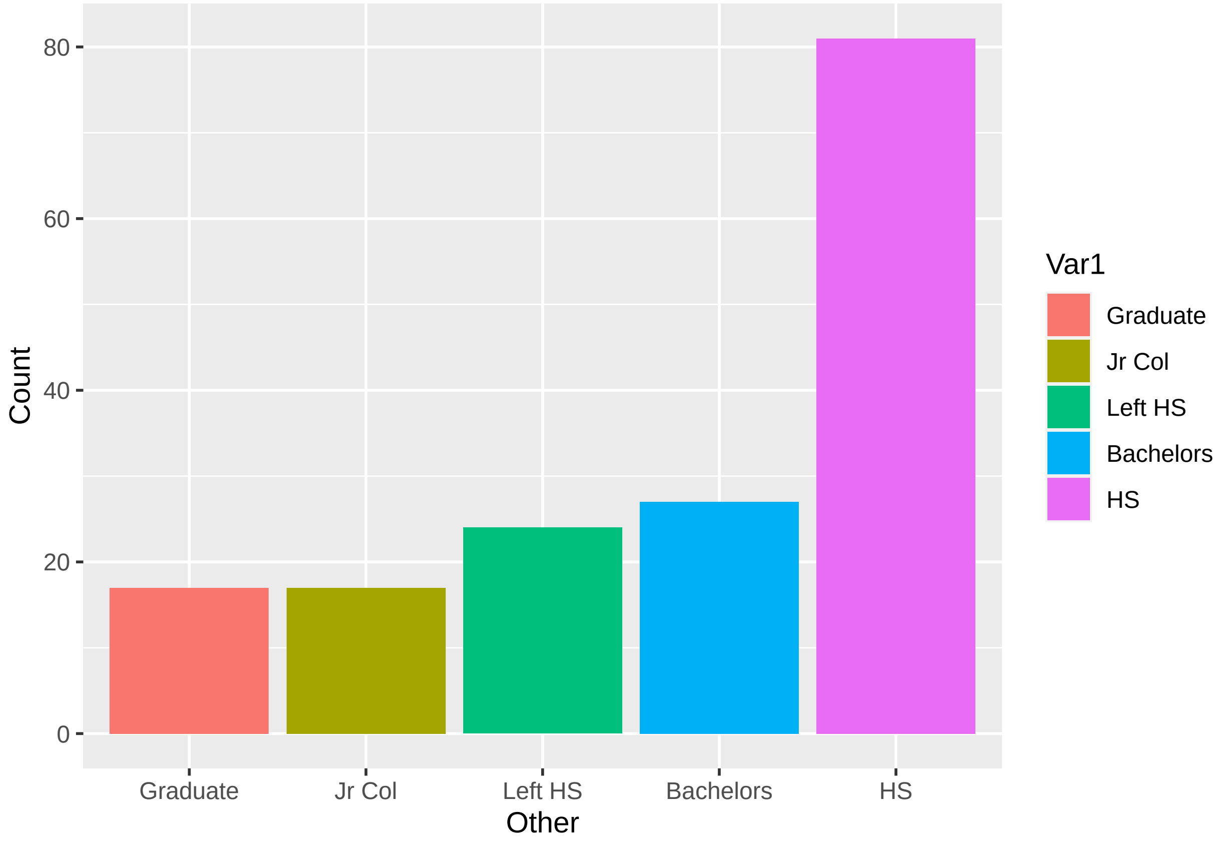


This can be converted into a sorted type of bar graph with

```
newdat = filter(gss, Race == "Other")
counts = as.data.frame(sort(table(newdat$Education)))
counts
```

```
##        Var1 Freq
## 1  Graduate   17
## 2    Jr Col   17
## 3   Left HS   24
## 4 Bachelors   27
## 5        HS   81
```

```
ggplot(data=counts, aes(x=Var1, y=Freq, fill=Var1)) + geom_bar(stat="identity") + xlab("Other") + ylab(
```



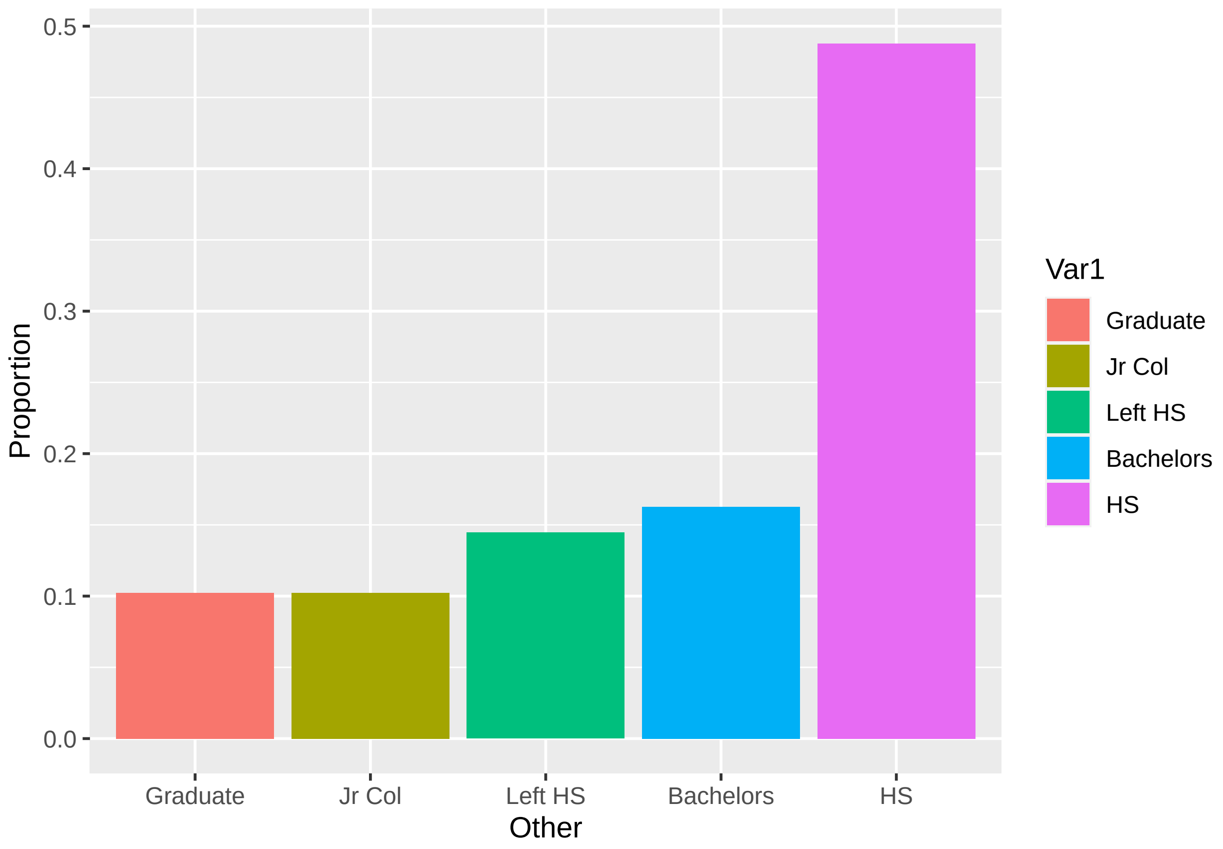
```
props = counts$Freq/sum(counts$Freq)
props
```

```
## [1] 0.1024096 0.1024096 0.1445783 0.1626506 0.4879518
```

```
newdat2 = data.frame(counts, props)
head(newdat2, 5)
```

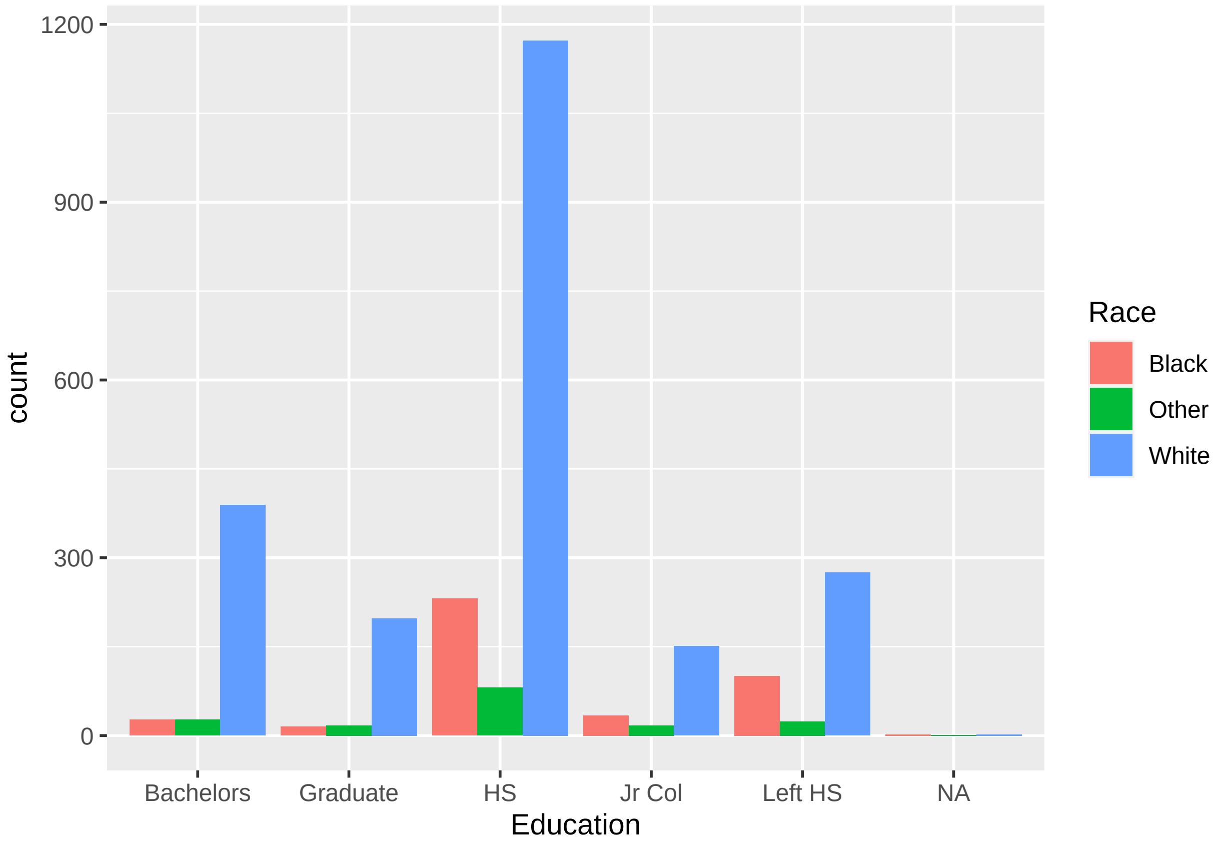
```
##        Var1 Freq     props
## 1  Graduate   17 0.1024096
## 2    Jr Col   17 0.1024096
## 3   Left HS   24 0.1445783
## 4 Bachelors   27 0.1626506
## 5        HS   81 0.4879518
```

```
ggplot(newdat2, aes(x=Var1, y=props, fill=Var1)) + geom_bar(stat="identity") + xlab("Other") + ylab("Pr
```



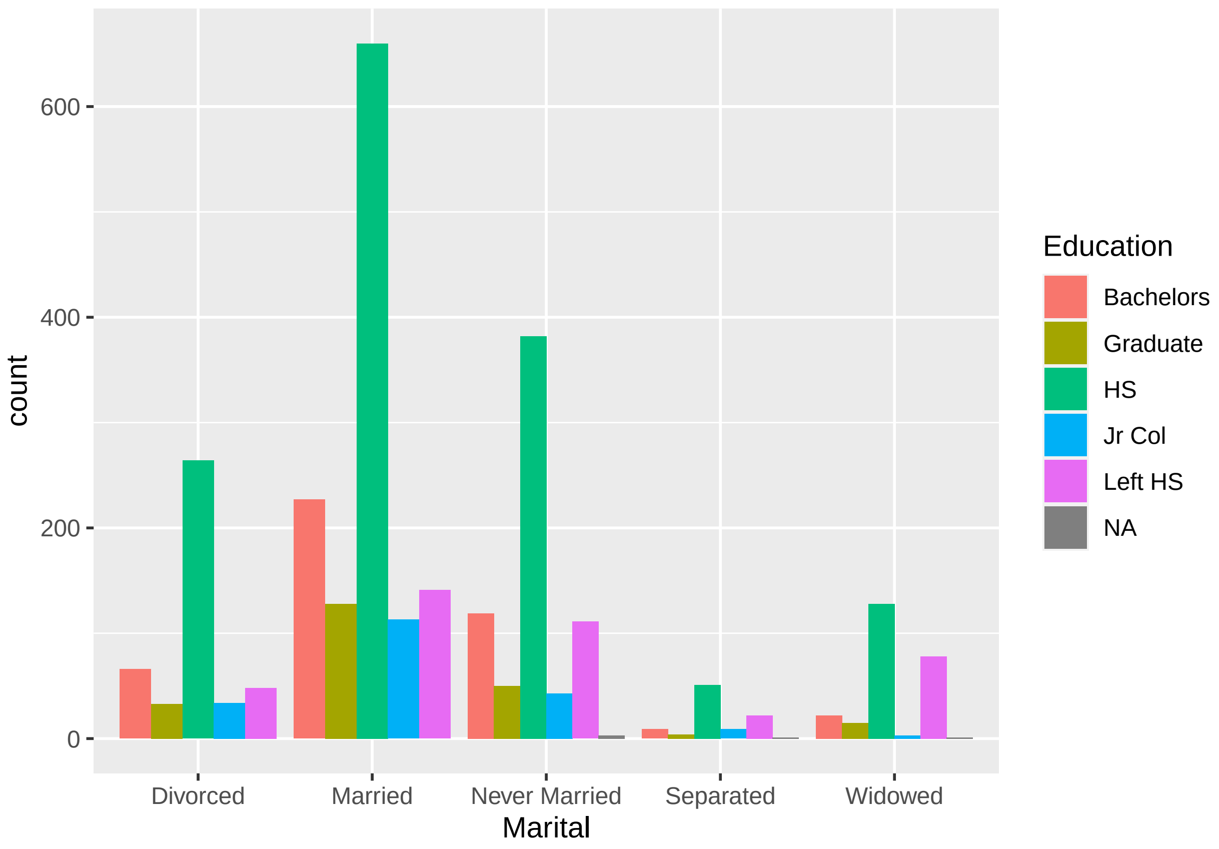
(a) Create a bar graph that demonstrates the distribution of race within each level of education. What can you infer from this bar graph?

```
ggplot(data=gss, aes(x = Education, fill=Race)) + geom_bar(position = "dodge", na.rm=TRUE)
```



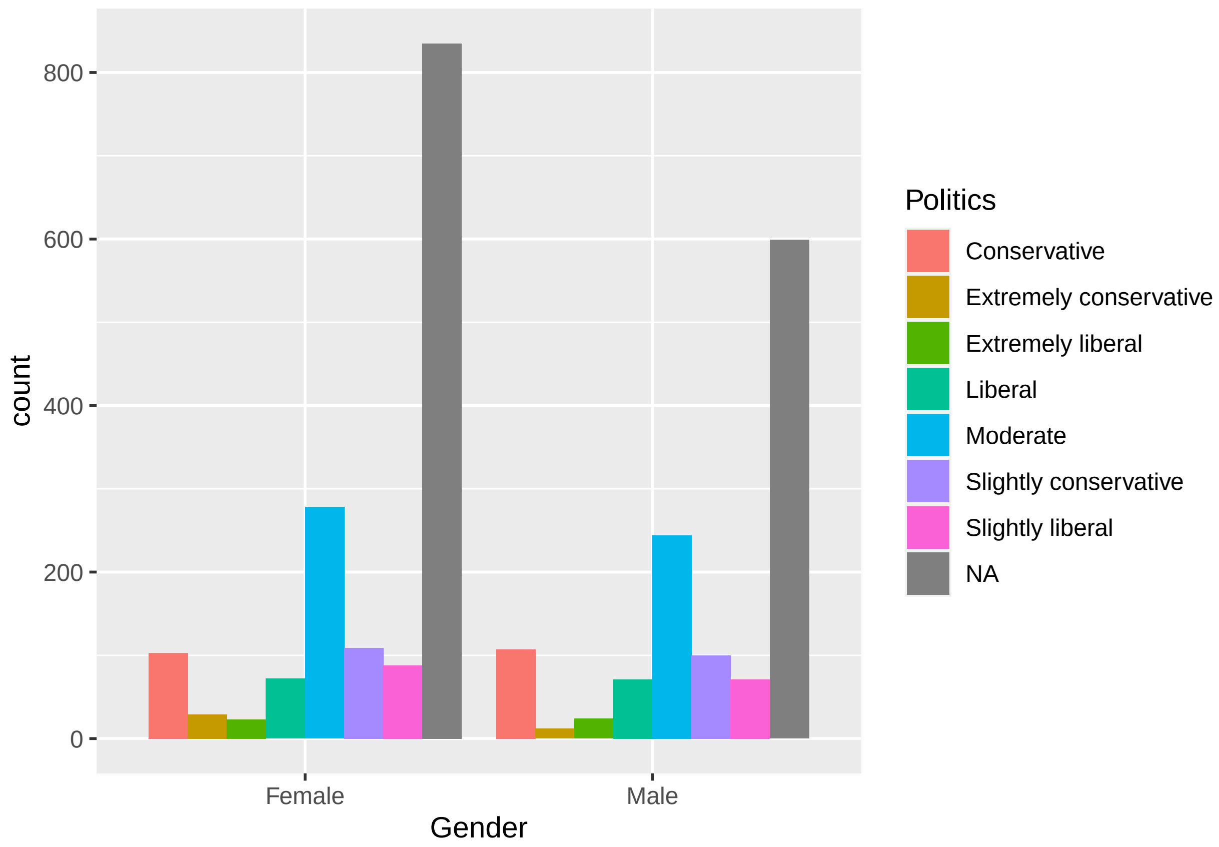
(b) Create a data visualization that can be used to demonstrate if there is a relationship between one's marital status (Marital) and their education level.

```
ggplot(data=gss, aes(x = Marital, fill=Education)) + geom_bar(position = "dodge", na.rm=TRUE)
```



(c) Create a data visualization that can be used to demonstrate if there is a relationship between one's Gender and their Politics.

```
ggplot(data=gss, aes(x = Gender, fill=Politics)) + geom_bar(position = "dodge", na.rm=TRUE)
```



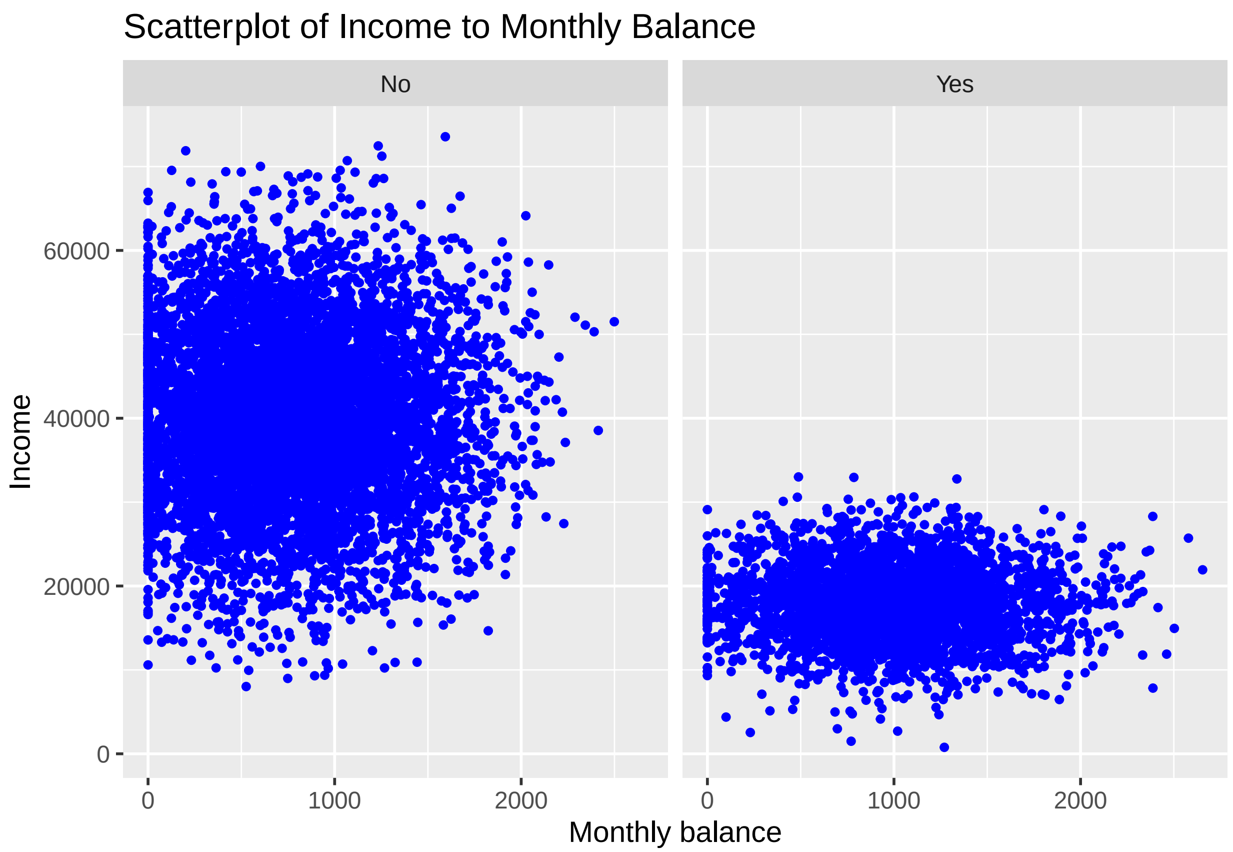
10. Refer to the **Default** data set in the *ISLR* package. This data set consists of 10000 cases. There are four different variables in this data set. "default" is a categorical variable that indicates if a person has defaulted on their credit card debt (Yes) or has not (No); the variable "student" flags a respondent as a student (Yes) or not (No); the third variable is the person's credit card balancing they are carrying, and the last variable "income" is the person's annual income.

```
library(ISLR)
head(Default, 4) #shows the first four lines of the Default data set
```

```
##   default student   balance   income
## 1      No      No  729.5265 44361.63
## 2      No     Yes  817.1804 12106.13
## 3      No      No 1073.5492 31767.14
## 4      No      No  529.2506 35704.49
```

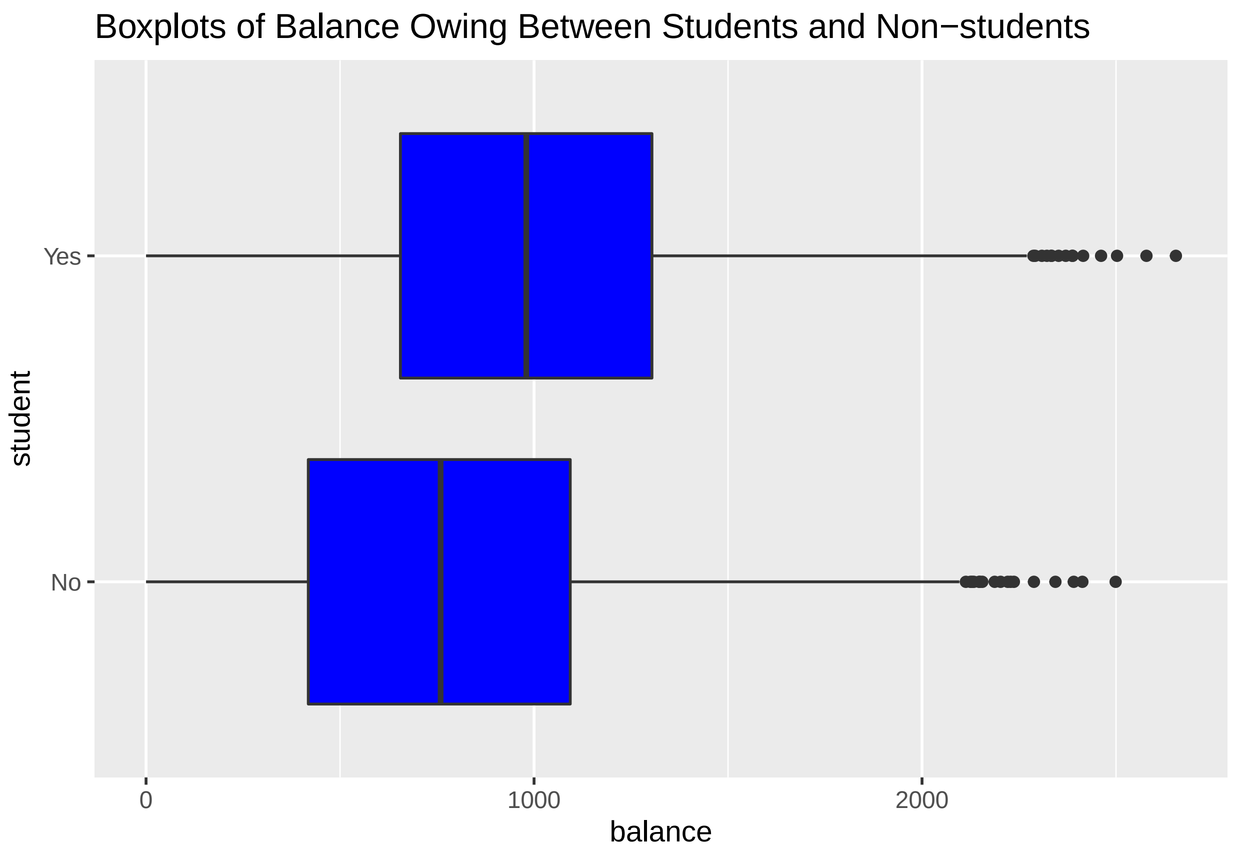
(a) Create a scatterplot that demonstrates the relationship between a person's income and their monthly balance they carry on their credit cards. Place the "income" variable as the y-axis and the "balance" variable as the x-axis. Within this visualization, differentiate between those who are students and those who are not.

```
ggplot(data=Default, aes(x = balance, y = income)) + geom_point(col='blue', size=1) + xlab("Monthly bal
```



(b) Create side-by-side boxplots that will compare the distributions of balance owing between students and non-students.

```
ggplot(data=filter(Default, (student == "Yes" | student=="No")), aes(x = student, y=balance)) + geom_bo
```



(c) Compute the means, medians, standard deviations, x_5, x_{95} (the 5th and 95th percentiles, respectively) for the data you visually summarized in part (b).

```
mean(~balance, data= Default)
```

```
## [1] 835.3749
```

```
aggregate(x = Default$balance,
          by = list(Default$student),
          FUN = mean)
```

```
##   Group.1        x
## 1      No 771.7704
## 2     Yes 987.8182
```

```
aggregate(x = Default$balance,
          by = list(Default$student),
          FUN = median)
```

```
##   Group.1        x
## 1      No 759.1891
## 2     Yes 979.9894
```

```
aggregate(x = Default$balance,
          by = list(Default$student),
          FUN = sd)
```

```
##   Group.1        x
## 1      No 469.6749
## 2     Yes 482.9097
```

```
percentile5thYes = qnorm(0.05, mean = 987.8182   , sd = 482.9097)
percentile5thNo = qnorm(0.05, mean = 771.7704        , sd = 469.6749)
percentile95thYes = qnorm(0.95, mean = 987.8182  , sd = 482.9097)
percentile95thNo = qnorm(0.95, mean = 771.7704       , sd = 469.6749)


percentile5thYes #193.5024
```

```
## [1] 193.5024
```

```
percentile5thYes #193.5024
```

```
## [1] 193.5024
```

```
percentile95thYes #1782.134
```

```
## [1] 1782.134
```

```
percentile95thNo #1544.317
```

```
## [1] 1544.317
```

```
#The mean of balance of studnet is 987.8182
#The mean of balance of non-studnet is 771.7704
#The median of balance of studnet is 979.9894
#The median of balance of non-studnet is 759.1891
#The standard deviations of balance of studnet is 482.9097
#The standard deviations of balance of studnet is 469.6749
#The 95th percentile of balance of studnet is 1782.134
#The 95th percentile of balance of non-studnet is 1544.317
#The 5th percentile of balance of studnet is 193.5024
#The 5th percentile of balance of non-studnet is 193.5024
```

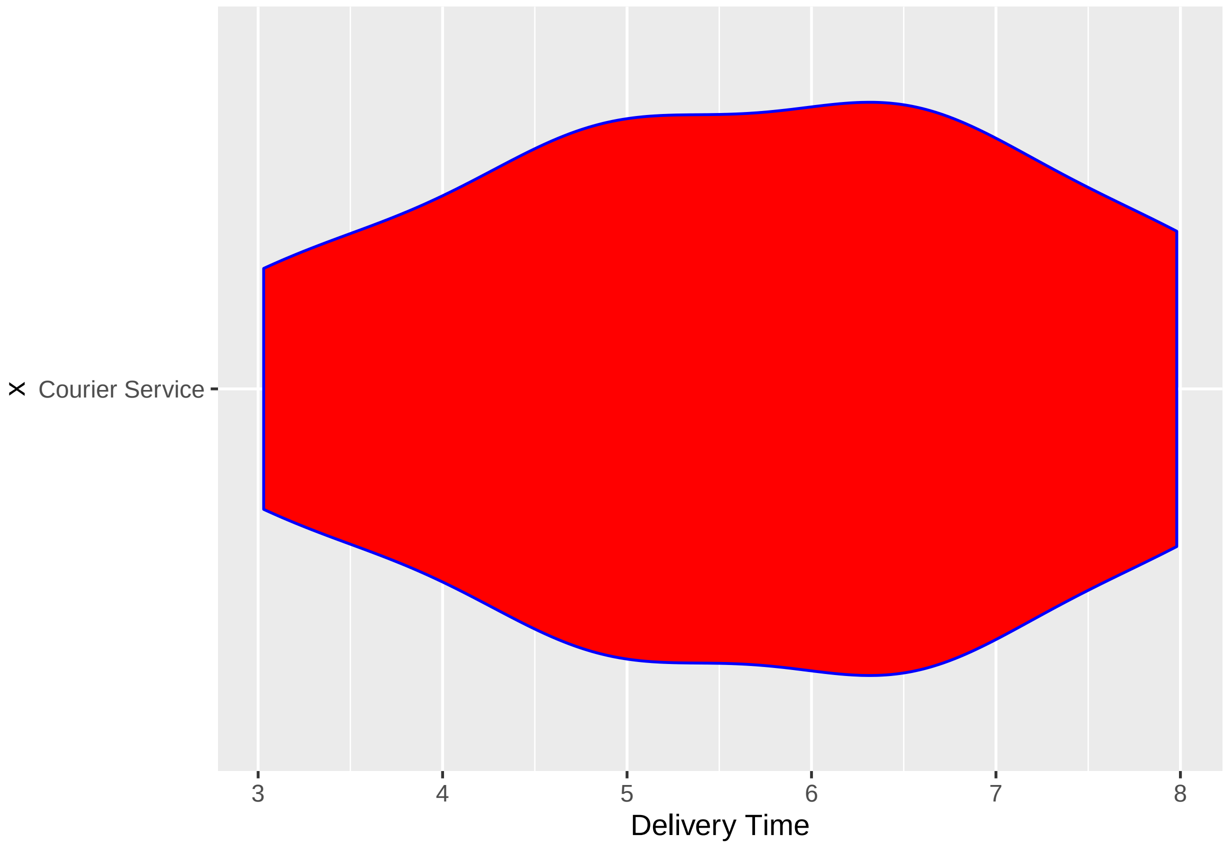
11. A local courier service advertises that the amount of time they take to deliver a package can be modelled by the Normal distribution with a mean delivery time of 5.0 hours and a standard deviation of 1.5 hours. A random sample of $n = 12$ deliveries was taken, and the number of hours it took each to be delivered was recorded. The data appears in csv file.

(a) Read the data in this file into a data frame. Create a violin plot of these data.

```r
mean11 = 5.0
sd11 =1.5
q11data <- read.csv("http://people.ucalgary.ca/~jbstall/DataFiles/Data602Assignment1Question11.csv")
head(q11data,20)
```

```
##    Delivery_time
## 1           3.03
## 2           6.33
## 3           6.50
## 4           5.22
## 5           3.56
## 6           6.76
## 7           7.98
## 8           4.82
## 9           7.96
## 10          4.54
## 11          5.09
## 12          6.46
```

```r
ggplot(data = q11data, aes(x = "Courier Service", y = Delivery_time)) + geom_violin(col="blue", fill="re
```



(b) From this data, compute the sample mean, the sample median, the sample standard deviation, the first and third quartiles, and the 99th percentile.

```
mean(~Delivery_time, data=q11data)
```

```
## [1] 5.6875
```

```
median(~Delivery_time, data=q11data)
```

```
## [1] 5.775
```

```
sd(~Delivery_time, data=q11data)
```

```
## [1] 1.580369
```

```
quantile(~Delivery_time, data=q11data)
```

```
##    0%   25%   50%   75%  100%
## 3.030 4.750 5.775 6.565 7.980
```

```
percentile99thNo = qnorm(0.99, mean = 5.6875         , sd = 1.580369)
percentile99thNo
```

```
## [1] 9.363988
```

```
#The sample mean is 5.6875
#The sample median is 5.775
#The sample standard deviation is 1.580369
#The first quartiles is 4.750
#The third quartiles is 6.565
#The 99th percentile is 9.363988
```

(c) Suppose you were part of a marketing campaign to promote the efficiency of delivery times, as a part of the campaign there was a promise of delivery within a certain number of hours, beyond which there would be a refund for 1% of all deliveries. Provide the point of refund.

```
#Base on the result of part b, the point of refund should be over 6 hours.
```