EXAM_2_R

Fentaw Abitew

2022-11-18

Instructions:Do not help or accept help from anyone as you work on and complete the exam. You are at liberty to use your book, past graded exams, or any online resource that you feel will be helpful. Send all r code and associated output as an r markdown file and as a knitted word file document.

```
#install.packages("tidyverse")
```

```
#install.packages("reshape2")
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

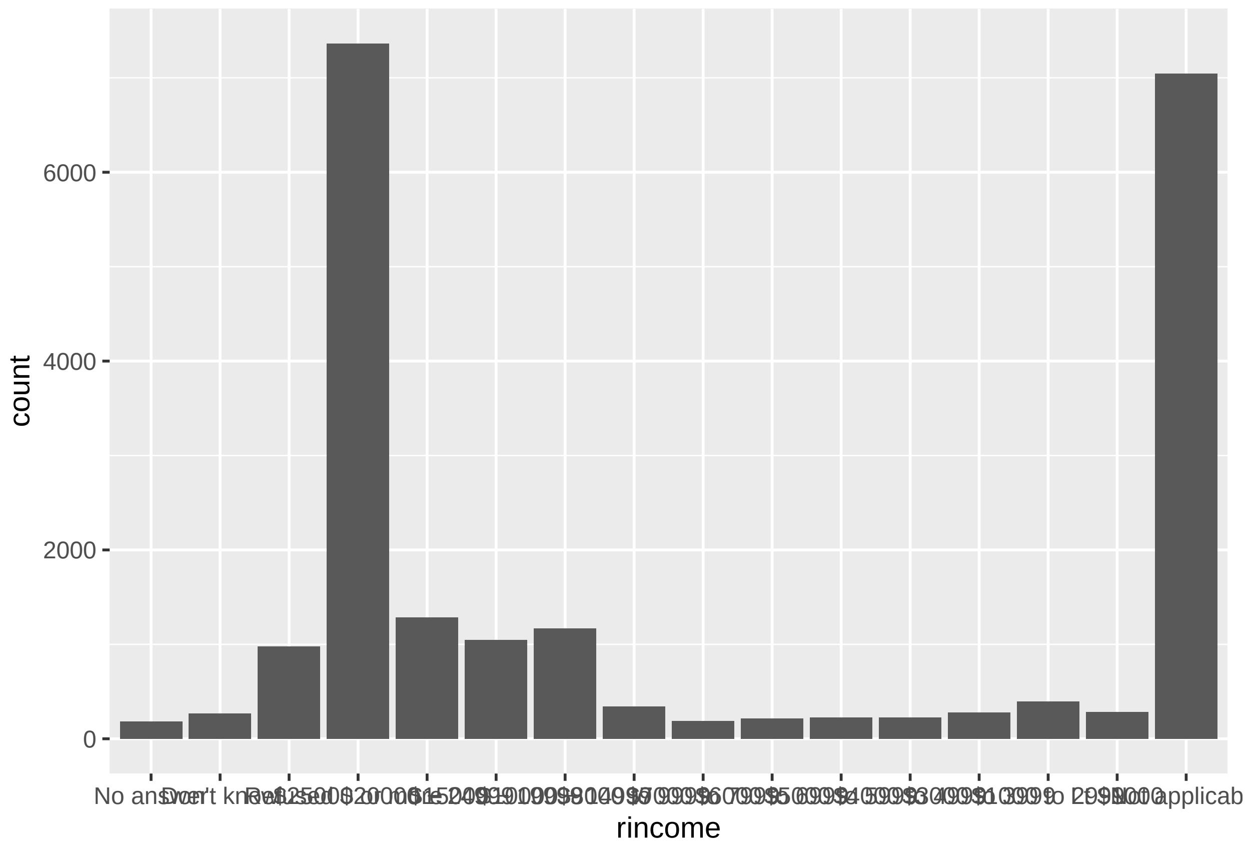
```
library(stringr)
library(dplyr)
library(ggplot2)
library(tidyr)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(readr)
library(forcats)
```

1. Using the gss_cat data frame, write r code that will produce the bar graph below. And explain in one or two sentences why the bar graph is difficult to interpret.

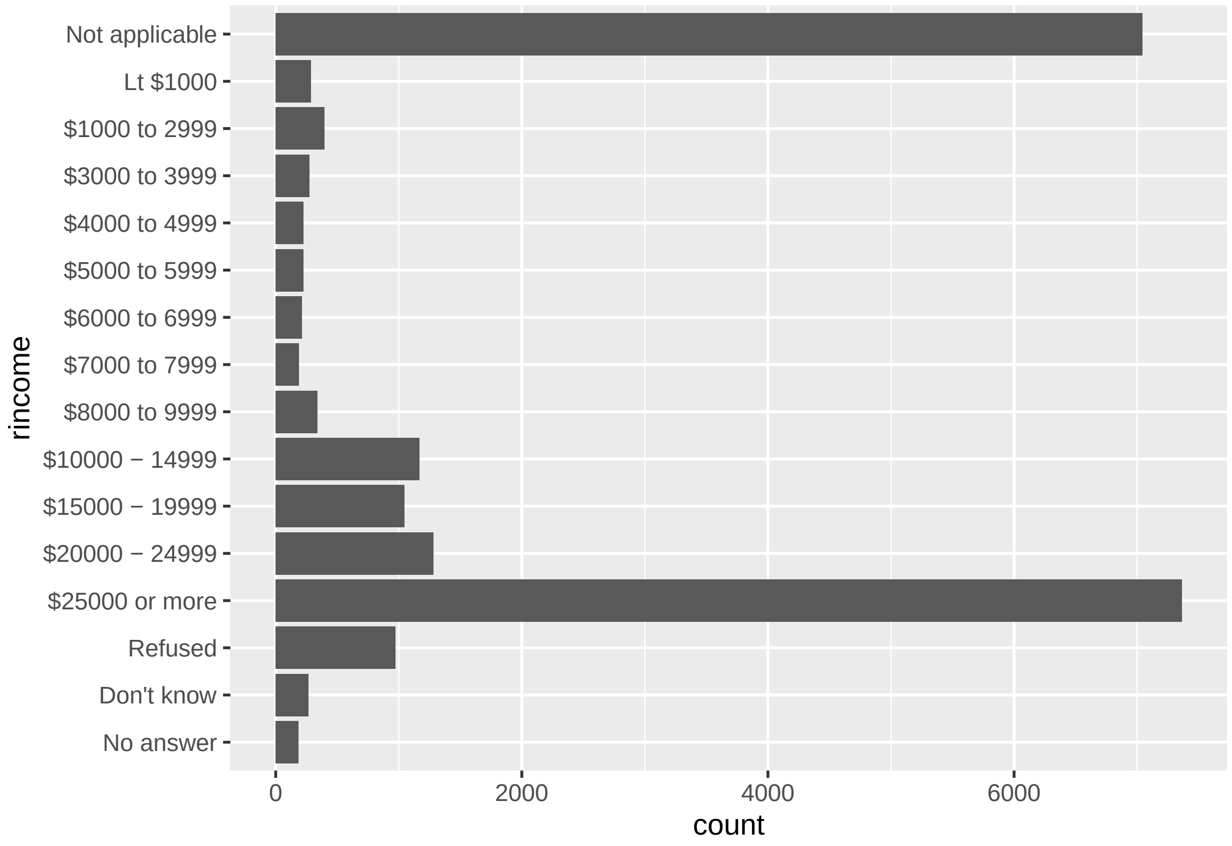
```
unclear_bar<- ggplot(gss_cat, aes(rincome)) +
  geom_bar()
unclear_bar
```



Here, we are unable to interpret the result for the obvious reason that we cannot see what the level of rincome are. The levels are overlapped and we cannot see what it is.

2. Now write r code from the same data set that produce the transformed bar graph and comment on why it is an improvement

```
flip_bar<-unclear_bar + coord_flip()
flip_bar
```



Here, it is better because we can see the levels of rincome clearly the coordinated filliped.

3. Use r code to produce the tips data frame from the reshape2 package. Name three categorical variables in the data frame.

```
names(tips) # those are the seven variables (column names) and "sex", "smoker", "day" and, "time" are c
```

```
## [1] "total_bill" "tip"        "sex"        "smoker"     "day"
## [6] "time"       "size"
```

```
tip_dataframe_sample<-head(tips, 10)
head(tip_dataframe_sample)
```

```
##   total_bill  tip    sex smoker day   time size
## 1      16.99 1.01 Female     No Sun Dinner    2
## 2      10.34 1.66   Male     No Sun Dinner    3
## 3      21.01 3.50   Male     No Sun Dinner    3
## 4      23.68 3.31   Male     No Sun Dinner    2
## 5      24.59 3.61 Female     No Sun Dinner    4
## 6      25.29 4.71   Male     No Sun Dinner    4
```

As we can see from the head data frame and obviously from the mark as fctr(factor): the categorical variables are sex, smoker, day, and time.

```
#install.packages("describer")
```

```
#also we can run class for each variable or use description to see the catagorical variables
class(tips$day)
```

```
## [1] "factor"
```

```r
describer::describe(tips)
```

```
##   .column_name .column_class .column_type .count_elements .mean_value .sd_value
## 1   total_bill       numeric       double             244   19.785943 8.9024120
## 2          tip       numeric       double             244    2.998279 1.3836382
## 3          sex        factor      integer             244          NA        NA
## 4       smoker        factor      integer             244          NA        NA
## 5          day        factor      integer             244          NA        NA
## 6         time        factor      integer             244          NA        NA
## 7         size       integer      integer             244    2.569672 0.9510998
##   .q0_value .q25_value .q50_value .q75_value .q100_value
## 1      3.07    13.3475     17.795    24.1275       50.81
## 2         1     2.0000      2.900     3.5625          10
## 3    Female         NA         NA         NA        Male
## 4        No         NA         NA         NA         Yes
## 5       Fri         NA         NA         NA        Thur
## 6    Dinner         NA         NA         NA       Lunch
## 7         1     2.0000      2.000     3.0000           6
```

4. Use r code to indicate how many levels exist for the factor day in the tips data frame and determine the frequency of each level.

```r
#One way of indicating  how many levels exist for the factor day in the tips data frame is using unique

unique(tips$day) # The day has four unique values or levels (Fri, Sat, Sun, Thur)
```

```
## [1] Sun  Sat  Thur Fri 
## Levels: Fri Sat Sun Thur
```

```r
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 objects are masked from 'package:dplyr':
## 
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:purrr':
## 
##     compact
```

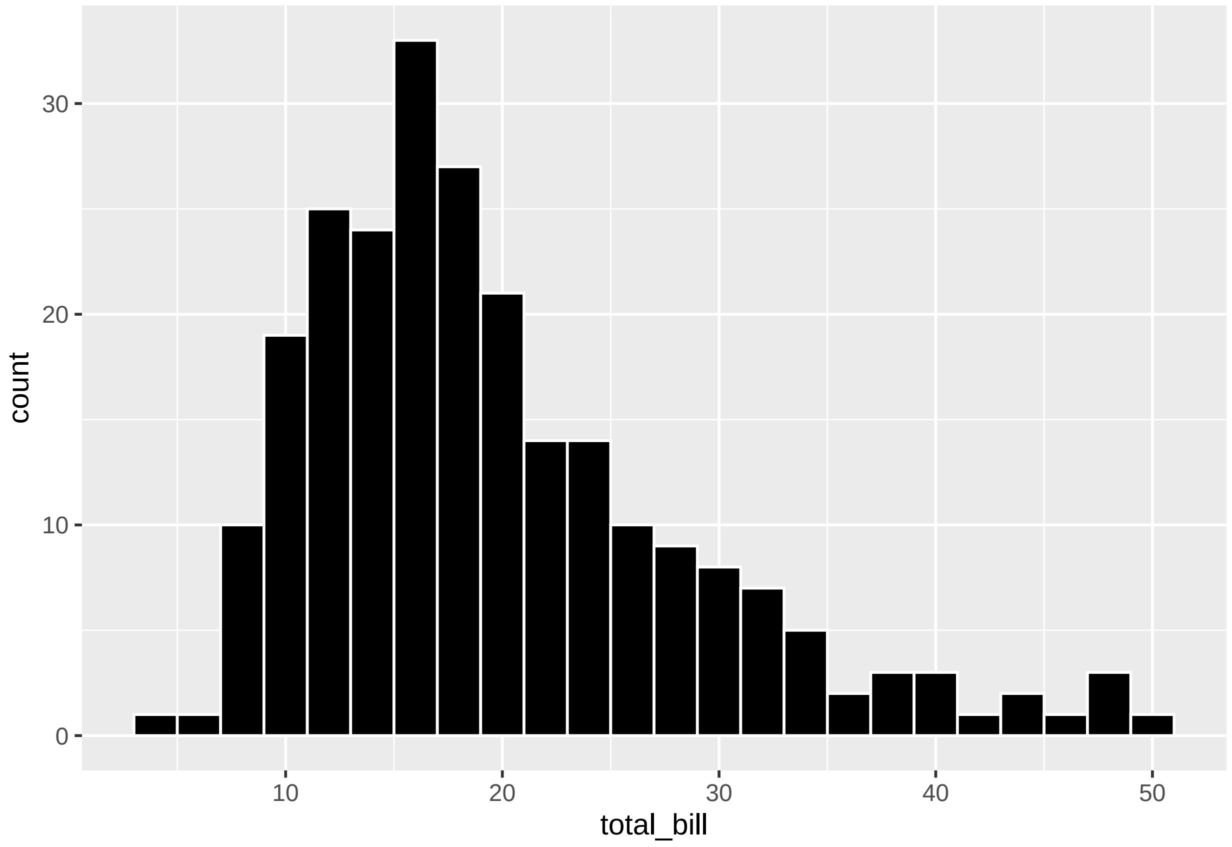
```r
count(tips$day) # each level of day with the freq
```

```
##      x freq
## 1  Fri   19
## 2  Sat   87
## 3  Sun   76
## 4 Thur   62
```

5. Produce r code that will produce the following histogram from the tips data frame

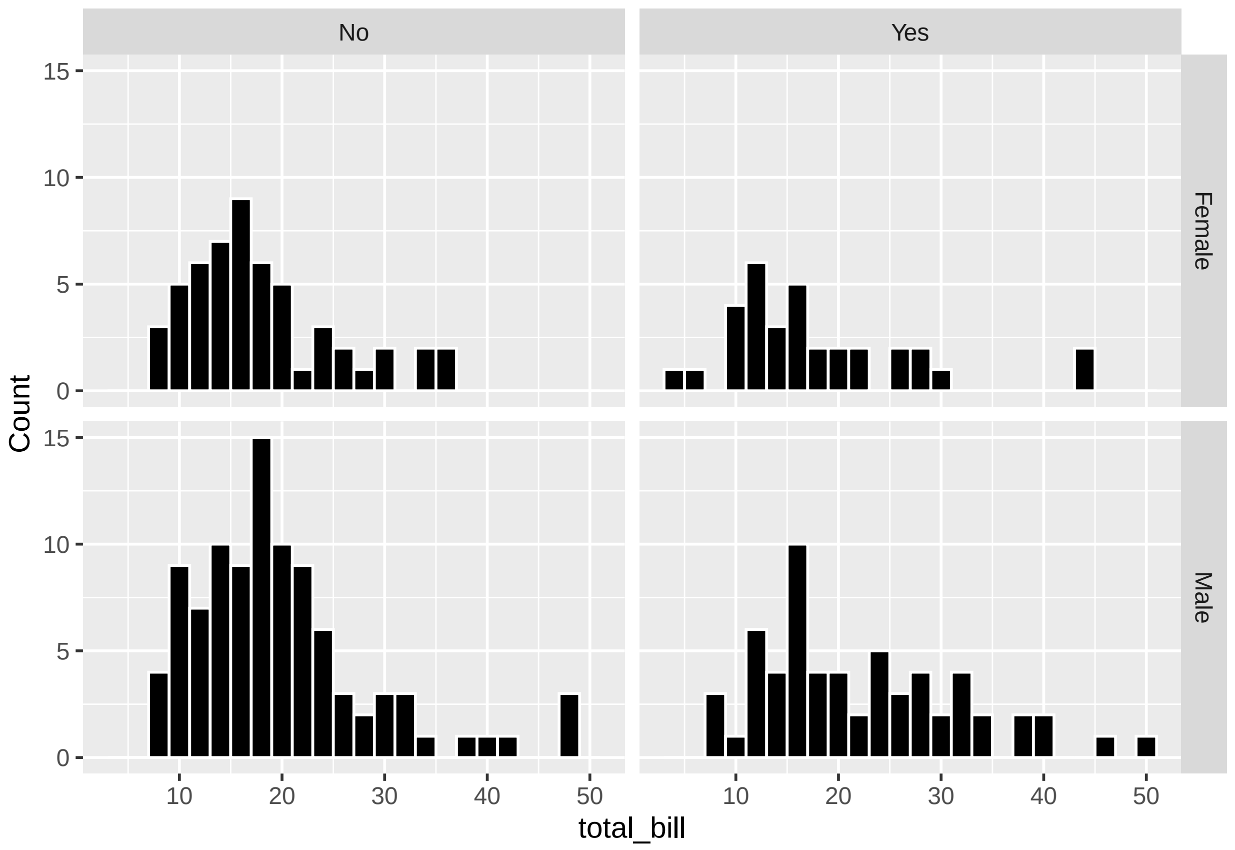
```r
barlines<-"white"
barfill <-"black"
ggplot(data = tips, aes(x=total_bill)) +
  geom_histogram(aes(y = ..count..), binwidth = 2,
                       colour = barlines, fill = barfill)
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
```



6. Write r code that will produce the following histograms from the tips data frame

```r
barlines<-"white"
ggplot(data = tips, aes(x=total_bill)) +
 scale_y_continuous(name = "Count")+
geom_histogram(aes(y = ..count..), binwidth = 2,
                       colour = barlines, fill = barfill)  +
facet_grid(sex ~smoker)
```



7.

Using the stringr::words data set along with str_subset code, produce R code that will show a 9 letter word that has the letter a in the middle.

```
#stringr::words
letter9_a<- str_subset(stringr::words, "....a....") # C-H-A-R-a-C-T-E-R
letter9_a
```

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

```
writeLines(letter9_a)
```

```
## character
```

8. Produce a string that will force a match for the regular expression \”” Use and show the R code command writelines to confirm your answer

```
p<-  '\\""\ '
writeLines(p)
```

```
## \""
```

‘

```
v<- 'abebe beso \\""\ bela '
writeLines(v)
```

```
## abebe beso \"" bela
```

9. Describe in words (two or three sentences) what the following regular expression will match ˆ.*e$

```
# ^ start
# $ end
# . and * these characters have special meaning in regex
```

```r
### all in all,  ^.*$ will match any string. example,

x= c("dog", "$1#23", "FENTAW")
str_view(x, "^.*$")
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, pleas
```

10.Using the methods demonstrated in class regarding Factors and Forcats, use and show R code to create a factor that will enable you to sort the string vector (“eight”, “four”, “ten”, “two”) according to quantity, not alphabetical order.

```r
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
# Create a vector

string_vector <- c("eight","four","ten","two")

# I create level from one up to ten in terms of quantity

string_level<-c("one","two", "three", "four", "five", "six", "seven", "eight", "nine", "ten") # I creat

# using factor and level create another variable

vec<-factor(string_vector, levels = string_level)

#sort it according level
sort(vec)
```

```
## [1] two   four  eight ten
## Levels: one two three four five six seven eight nine ten
```