Assignment 4

Nathan Mokhtarzadeh

April 26, 2018

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------
------------------------------------------ tidyverse 1.2.1 --

## v ggplot2 2.2.1     v purrr   0.2.4
## v tibble  1.4.2     v dplyr   0.7.4
## v tidyr   0.8.0     v stringr 1.3.0
## v readr   1.1.1     v forcats 0.3.0

## Warning: package 'ggplot2' was built under R version 3.4.4

## Warning: package 'stringr' was built under R version 3.4.4

## -- Conflicts ------------------------------------------------------------
------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(foreign)
library(dplyr)
library(xtable)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'

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

```
library(stringr)
```

12.6.1

```
who1 <- who %>%
  gather(new_sp_m014:newrel_f65, key = "key", value = "cases", na.rm = TRUE)
#who1
who1 %>%
  count(key)
```

```
## # A tibble: 56 x 2
##    key              n
##    <chr>        <int>
```

```
##  1 new_ep_f014   1032
##  2 new_ep_f1524  1021
##  3 new_ep_f2534  1021
##  4 new_ep_f3544  1021
##  5 new_ep_f4554  1017
##  6 new_ep_f5564  1017
##  7 new_ep_f65    1014
##  8 new_ep_m014   1038
##  9 new_ep_m1524  1026
## 10 new_ep_m2534  1020
## # ... with 46 more rows
```

```
who2 <- who1 %>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel"))
who2
```

```
## # A tibble: 76,046 x 6
##    country     iso2  iso3   year key         cases
##    <chr>       <chr> <chr> <int> <chr>       <int>
##  1 Afghanistan AF    AFG    1997 new_sp_m014     0
##  2 Afghanistan AF    AFG    1998 new_sp_m014    30
##  3 Afghanistan AF    AFG    1999 new_sp_m014     8
##  4 Afghanistan AF    AFG    2000 new_sp_m014    52
##  5 Afghanistan AF    AFG    2001 new_sp_m014   129
##  6 Afghanistan AF    AFG    2002 new_sp_m014    90
##  7 Afghanistan AF    AFG    2003 new_sp_m014   127
##  8 Afghanistan AF    AFG    2004 new_sp_m014   139
##  9 Afghanistan AF    AFG    2005 new_sp_m014   151
## 10 Afghanistan AF    AFG    2006 new_sp_m014   193
## # ... with 76,036 more rows
```

1

I think it's reasonable, but it could have some bad effects depending on what you want

an NA value can prevent certain computations from being made while having a zero

may result in some inaccurate computations like means

2

If you neglect the step we won't be able to separate into var, sexage, and new

```
who3 <- who2 %>%
  separate(key, c("new", "type", "sexage"), sep = "_")
#who3

who3 %>%
  count(new)
```

```
## # A tibble: 1 x 2
##   new       n
##   <chr> <int>
## 1 new   76046
```

```
who4 <- who3 %>%
  select(-new, -iso2, -iso3)

who5 <- who4 %>%
  separate(sexage, c("sex", "age"), sep = 1)
who5
```

```
## # A tibble: 76,046 x 6
##    country      year type  sex   age   cases
##    <chr>       <int> <chr> <chr> <chr> <int>
##  1 Afghanistan  1997 sp    m     014       0
##  2 Afghanistan  1998 sp    m     014      30
##  3 Afghanistan  1999 sp    m     014       8
##  4 Afghanistan  2000 sp    m     014      52
##  5 Afghanistan  2001 sp    m     014     129
```

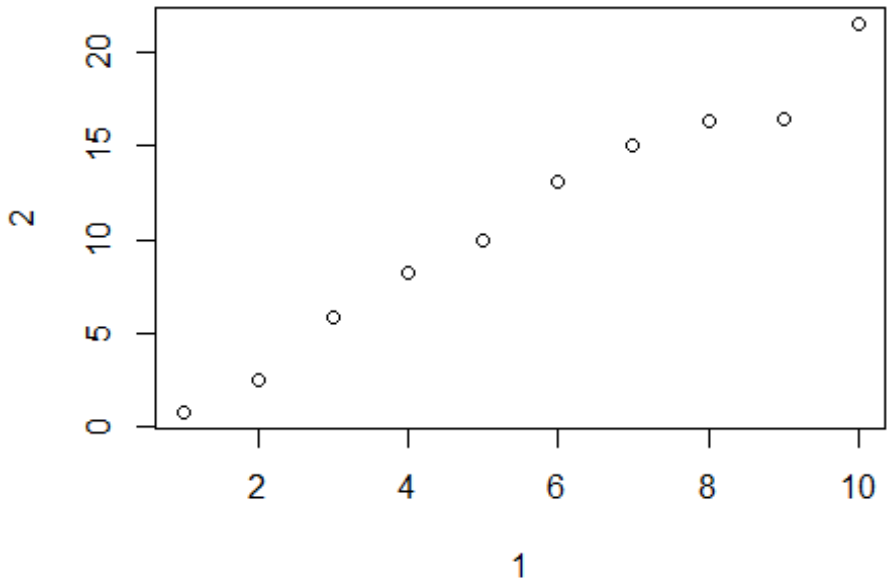
```
##  6 Afghanistan  2002 sp    m     014      90
##  7 Afghanistan  2003 sp    m     014     127
##  8 Afghanistan  2004 sp    m     014     139
##  9 Afghanistan  2005 sp    m     014     151
## 10 Afghanistan  2006 sp    m     014     193
## # ... with 76,036 more rows
```

Problem 5

```
annoying <- tibble(
  `1` = 1:10,
  `2` = `1` * 2 + rnorm(length(`1`))
)
annoying$`1`[1]
```

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

```
#4.2
plot(annoying)
```




```
#points(annoying[2], col = 'blue')
#points(annoying[1], col = 'red')
#4.3
toAdd <- annoying[[1]]/annoying[[2]]
annoying <- add_column(annoying,`3` = toAdd)
#4.4
names(annoying)[names(annoying) == '3'] <- 'three'
names(annoying)[names(annoying) == '2'] <- 'two'
```

```
names(annoying)[names(annoying) == '1'] <- 'one'

#5 using data from previous questions
#run in script file, the data displays there
tibble::enframe(annoying[1:3,])
```

```
## # A tibble: 3 x 2
##   name  value
##   <chr> <tibble>
## 1 one   1:3
## 2 two   c(0.778539913202733, 2.46374542068442, 5.80127118786662)
## 3 three c(1.28445566250577, 0.811772183606701, 0.517128040191348)
```

Table 4->6

```
pew <- read.spss('pew.sav')

pewDf <- as_data_frame(pew)

religion <- pewDf[c('q16', 'reltrad', 'income')]
religion$reltrad <- as.character(religion$reltrad)
religion$reltrad <- str_replace(religion$reltrad, 'Churches', "")
religion$reltrad <- str_replace(religion$reltrad, 'Protestant', 'Prot')
religion$reltrad[religion$q16 == 'Atheist(do not believe in God)'] <-
'Atheist'
religion$reltrad[religion$q16 == 'Agnostic (not sure if there is a God)'] <-
'Agnostic'
religion$reltrad <- str_trim(religion$reltrad)
religion$reltrad <- str_replace_all(religion$reltrad, ' \\(.*?\\)', '')

religion$income <- c('Less than $10,000' = '<$10k',
                     '10 to under $20,000' = '$10-20k',
                     '20 to under $30,000' = '$20-30k',
                     '30 to under $40,000' = '$30-40k',
                     '40 to under $50,000' = '$40-50k',
                     '50 to under $75,000' = '$50-75k',
                     '75 to under $100,000' = '$75-100k',
                     '100 to under $150,000' = '$100-150k',
                     '$150,000 or more' = '>150k',
                     "Don't know/Refused (VOL)" = "Don't
know/refused")[religion$income]

lens <- dplyr::count(religion, 'reltrad', 'income')
names(lens)[1] <- 'religion'

first <- religion %>%
  dplyr::group_by(reltrad, q16, income) %>%
  count(income)

second <- first[c('reltrad', 'income', 'n')]
```

```
third <- second %>%
  dplyr::rename(freq = n)
#just showing the first ten elements, for whatever reason when I knit the
entire dataset gets put onto the document and it becomes 300 pages
third[1:10,]
```

```
## # A tibble: 10 x 3
##    reltrad  income              freq
##    <chr>    <chr>              <int>
##  1 Buddhist $10-20k               21
##  2 Buddhist $100-150k             39
##  3 Buddhist $20-30k               30
##  4 Buddhist $30-40k               34
##  5 Buddhist $40-50k               33
##  6 Buddhist $50-75k               58
##  7 Buddhist $75-100k              62
##  8 Buddhist <$10k                 27
##  9 Buddhist >150k                 53
## 10 Buddhist Don't know/refused    54
```

Table 7 -> 8

```
bb <- read.csv('billboard.csv')

clean <- bb %>%
  gather(key = 'week', value = 'rank',  -time, -genre, -date.entered, -
date.peaked, -year, -artist.inverted, -track) %>%
  select(year, artist=artist.inverted, time, track, date = date.entered,
week, rank ) %>%
  filter(!is.na(rank)) %>%
  arrange(track) %>%
  separate(week, into=c('x', 'y', 'z'), convert = TRUE, sep=c(1, -7)) %>%
  dplyr::rename(week = y) %>%
  select(-x, -z) %>%
  arrange(artist, track)
#When I knit, it shows the entire tibble data, just showing the first ten
elements.
clean[1:10,]
```

```
##    year  artist time
## 1  2000   2 Pac 4:22
## 2  2000   2 Pac 4:22
## 3  2000   2 Pac 4:22
## 4  2000   2 Pac 4:22
## 5  2000   2 Pac 4:22
## 6  2000   2 Pac 4:22
## 7  2000   2 Pac 4:22
## 8  2000 2Ge+her 3:15
## 9  2000 2Ge+her 3:15
## 10 2000 2Ge+her 3:15
##                                                    track       date
```

```
## 1                            Baby Don't Cry (Keep Ya Head Up II) 2000-02-26
## 2                            Baby Don't Cry (Keep Ya Head Up II) 2000-02-26
## 3                            Baby Don't Cry (Keep Ya Head Up II) 2000-02-26
## 4                            Baby Don't Cry (Keep Ya Head Up II) 2000-02-26
## 5                            Baby Don't Cry (Keep Ya Head Up II) 2000-02-26
## 6                            Baby Don't Cry (Keep Ya Head Up II) 2000-02-26
## 7                            Baby Don't Cry (Keep Ya Head Up II) 2000-02-26
## 8  The Hardest Part Of Breaking Up (Is Getting Back Your Stuff) 2000-09-02
## 9  The Hardest Part Of Breaking Up (Is Getting Back Your Stuff) 2000-09-02
## 10 The Hardest Part Of Breaking Up (Is Getting Back Your Stuff) 2000-09-02
##    week rank
## 1     1   87
## 2     2   82
## 3     3   72
## 4     4   77
## 5     5   87
## 6     6   94
## 7     7   99
## 8     1   91
## 9     2   87
## 10    3   92
```