Assignment_04 Final

Steven Tran

February 28, 2018

```
library(dplyr)
```

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

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

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

```
library(tidyverse)
```

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

```
## v ggplot2 2.2.1     v readr   1.1.1
## v tibble  1.4.2     v purrr   0.2.4
## v tidyr   0.8.0     v stringr 1.2.0
## v ggplot2 2.2.1     v forcats 0.2.0
```

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

R for Data Science

10.5 Exercises

5. What does tibble::enframe() do? When might you use it?

enframe() converts vectors or lists to a dataframe. The opposite to this is deframe(). I would use enframe() when I am given a vector or list to analyze.

12.6 Exercises

3. I claimed that iso2 and iso3 were redundant with country. Confirm this claim.

```
head(who)
```

```
## # A tibble: 6 x 60
##   country     iso2  iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534
##   <chr>       <chr> <chr> <int>       <int>        <int>        <int>
## 1 Afghanistan AF    AFG    1980          NA           NA           NA
## 2 Afghanistan AF    AFG    1981          NA           NA           NA
## 3 Afghanistan AF    AFG    1982          NA           NA           NA
```

```
## 4 Afghanistan AF    AFG    1983          NA           NA           NA
## 5 Afghanistan AF    AFG    1984          NA           NA           NA
## 6 Afghanistan AF    AFG    1985          NA           NA           NA
## # ... with 53 more variables: new_sp_m3544 <int>, new_sp_m4554 <int>,
## #   new_sp_m5564 <int>, new_sp_m65 <int>, new_sp_f014 <int>,
## #   new_sp_f1524 <int>, new_sp_f2534 <int>, new_sp_f3544 <int>,
## #   new_sp_f4554 <int>, new_sp_f5564 <int>, new_sp_f65 <int>,
## #   new_sn_m014 <int>, new_sn_m1524 <int>, new_sn_m2534 <int>,
## #   new_sn_m3544 <int>, new_sn_m4554 <int>, new_sn_m5564 <int>,
## #   new_sn_m65 <int>, new_sn_f014 <int>, new_sn_f1524 <int>,
## #   new_sn_f2534 <int>, new_sn_f3544 <int>, new_sn_f4554 <int>,
## #   new_sn_f5564 <int>, new_sn_f65 <int>, new_ep_m014 <int>,
## #   new_ep_m1524 <int>, new_ep_m2534 <int>, new_ep_m3544 <int>,
## #   new_ep_m4554 <int>, new_ep_m5564 <int>, new_ep_m65 <int>,
## #   new_ep_f014 <int>, new_ep_f1524 <int>, new_ep_f2534 <int>,
## #   new_ep_f3544 <int>, new_ep_f4554 <int>, new_ep_f5564 <int>,
## #   new_ep_f65 <int>, newrel_m014 <int>, newrel_m1524 <int>,
## #   newrel_m2534 <int>, newrel_m3544 <int>, newrel_m4554 <int>,
## #   newrel_m5564 <int>, newrel_m65 <int>, newrel_f014 <int>,
## #   newrel_f1524 <int>, newrel_f2534 <int>, newrel_f3544 <int>,
## #   newrel_f4554 <int>, newrel_f5564 <int>, newrel_f65 <int>
```

```
tail(who)
```

```
## # A tibble: 6 x 60
##   country  iso2  iso3   year new_sp_m014 new_sp_m1524 new_sp_m2534
##   <chr>    <chr> <chr> <int>       <int>        <int>        <int>
## 1 Zimbabwe ZW    ZWE    2008         127          614            0
## 2 Zimbabwe ZW    ZWE    2009         125          578           NA
## 3 Zimbabwe ZW    ZWE    2010         150          710         2208
## 4 Zimbabwe ZW    ZWE    2011         152          784         2467
## 5 Zimbabwe ZW    ZWE    2012         120          783         2421
## 6 Zimbabwe ZW    ZWE    2013          NA           NA           NA
## # ... with 53 more variables: new_sp_m3544 <int>, new_sp_m4554 <int>,
## #   new_sp_m5564 <int>, new_sp_m65 <int>, new_sp_f014 <int>,
## #   new_sp_f1524 <int>, new_sp_f2534 <int>, new_sp_f3544 <int>,
## #   new_sp_f4554 <int>, new_sp_f5564 <int>, new_sp_f65 <int>,
## #   new_sn_m014 <int>, new_sn_m1524 <int>, new_sn_m2534 <int>,
## #   new_sn_m3544 <int>, new_sn_m4554 <int>, new_sn_m5564 <int>,
## #   new_sn_m65 <int>, new_sn_f014 <int>, new_sn_f1524 <int>,
## #   new_sn_f2534 <int>, new_sn_f3544 <int>, new_sn_f4554 <int>,
## #   new_sn_f5564 <int>, new_sn_f65 <int>, new_ep_m014 <int>,
## #   new_ep_m1524 <int>, new_ep_m2534 <int>, new_ep_m3544 <int>,
## #   new_ep_m4554 <int>, new_ep_m5564 <int>, new_ep_m65 <int>,
## #   new_ep_f014 <int>, new_ep_f1524 <int>, new_ep_f2534 <int>,
## #   new_ep_f3544 <int>, new_ep_f4554 <int>, new_ep_f5564 <int>,
## #   new_ep_f65 <int>, newrel_m014 <int>, newrel_m1524 <int>,
## #   newrel_m2534 <int>, newrel_m3544 <int>, newrel_m4554 <int>,
## #   newrel_m5564 <int>, newrel_m65 <int>, newrel_f014 <int>,
## #   newrel_f1524 <int>, newrel_f2534 <int>, newrel_f3544 <int>,
## #   newrel_f4554 <int>, newrel_f5564 <int>, newrel_f65 <int>
```

No matter which observation one picks, iso2 and iso3 changes accordingly with country and is redundant.

4. For each country, year, and sex compute the total number of cases of TB. Make an informative visualisation of the data.

```
whoTidy <- who %>%
  gather(code, value, new_sp_m014:newrel_f65, na.rm = TRUE) %>%
  mutate(code = stringr::str_replace(code, "newrel", "new_rel")) %>%
  separate(code, c("new", "var", "sexage")) %>%
  select(-new, -iso2, -iso3) %>%
  separate(sexage, c("sex", "age"), sep = 1) %>%
  group_by(country, year, sex) %>%
  summarize(Number =n())
whoTidy
```

```
## # A tibble: 6,921 x 4
## # Groups:   country, year [?]
##    country      year sex   Number
##    <chr>       <int> <chr>  <int>
##  1 Afghanistan  1997 f          7
##  2 Afghanistan  1997 m          7
##  3 Afghanistan  1998 f          7
##  4 Afghanistan  1998 m          7
##  5 Afghanistan  1999 f          7
##  6 Afghanistan  1999 m          7
##  7 Afghanistan  2000 f          7
##  8 Afghanistan  2000 m          7
##  9 Afghanistan  2001 f          7
## 10 Afghanistan  2001 m          7
## # ... with 6,911 more rows
```

Tidy Data Article

Table4 to Table6

```
library(foreign)
library(stringr)
library(dplyr)
source("xtable.r")
pew <- read.spss("pew.sav")
```

```
## re-encoding from CP1252
```

```
## Warning in read.spss("pew.sav"): Undeclared level(s) 2, 3, 4, 9 added in
## variable: density3
```

```
## Warning in read.spss("pew.sav"): Duplicated levels in factor denom:
## Electronic ministries
```

```
## Warning in read.spss("pew.sav"): Undeclared level(s) 1, 2, 3, 4, 5, 6, 7,
## 8, 9, 10, 11, 12, 14, 16, 23, 33 added in variable: children
```

```
## Warning in read.spss("pew.sav"): Undeclared level(s) 18, 19, 20, 21, 22,
## 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
## 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
## 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
```

```
## 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96 added in
## variable: age
```

```
pew <- as.data.frame(pew)


religion <- pew[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]
religion$income <- factor(religion$income, levels = c("<$10k", "$10-20k", "$20-30k", "$30-40k", "$40-50
  "$75-100k", "$100-150k", ">150k", "Don't know/refused"))
colnames(religion) <- c("q16","religion","income")

r <- select(religion, c(religion, income))
table4 <- r %>%
  group_by_(.dots=c("religion", "income")) %>%
  summarize(Number = n()) %>%
  spread(key = income, value = Number) %>%
  arrange(religion)

table6 <- table4 %>%
  gather(key = "income",value = "freq", 2:11) %>%
  arrange(religion)

head(table4)
```

```
## # A tibble: 6 x 11
## # Groups:   religion [6]
##   religion       `<$10k` `$10-20k` `$20-30k` `$30-40k` `$40-50k` `$50-75k`
##   <chr>            <int>     <int>     <int>     <int>     <int>     <int>
## 1 Agnostic            27        34        60        81        76       137
## 2 Atheist             12        27        37        52        35        70
## 3 Buddhist            27        21        30        34        33        58
## 4 Catholic           418       617       732       670       638      1116
## 5 Don't know/re~      15        14        15        11        10        35
## 6 Evangelical P~     575       869      1064       982       881      1486
## # ... with 4 more variables: `$75-100k` <int>, `$100-150k` <int>,
## #   `>150k` <int>, `Don't know/refused` <int>
```

```
head(table6)
```

```
## # A tibble: 6 x 3
## # Groups:   religion [1]
##   religion income   freq
##   <chr>    <chr>   <int>
## 1 Agnostic <$10k      27
## 2 Agnostic $10-20k    34
## 3 Agnostic $20-30k    60
## 4 Agnostic $30-40k    81
## 5 Agnostic $40-50k    76
## 6 Agnostic $50-75k   137
```

Table7 to Table8

```
library(reshape2)
```

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

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(stringr)
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:lubridate':
##
##     here
```

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

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

```
##     summarize
```

```
raw <- read.csv("billboard.csv")
raw <- raw[, c("year", "artist.inverted", "track", "time", "date.entered", "x1st.week", "x2nd.week", "x3
names(raw)[2] <- "artist"
raw$artist <- iconv(raw$artist, "MAC", "ASCII//translit")
raw$track <- str_replace(raw$track, " \\(.*?\\)", "")
names(raw)[-(1:5)] <- str_c("wk", 1:76)
raw <- arrange(raw, year, artist, track)
long_name <- nchar(raw$track) > 20
raw$track[long_name] <- paste0(substr(raw$track[long_name], 0, 20), "...")
table7 <- raw

clean <- melt(raw, id = 1:5, na.rm = T)
clean$week <- as.integer(str_replace_all(clean$variable, "[^0-9]+", ""))
clean$variable <- NULL
clean$date.entered <- ymd(clean$date.entered)
clean$date <- clean$date.entered + weeks(clean$week - 1)
clean$date.entered <- NULL
clean <- rename(clean, c("value" = "rank"))
clean <- arrange(clean, year, artist, track, time, week)
clean <- clean[c("year", "artist", "time", "track", "date", "week", "rank")]
table8 <- clean

head(table7)
```

```
##   year       artist                    track time date.entered wk1 wk2 wk3
## 1 2000        2 Pac           Baby Don't Cry 4:22   2000-02-26  87  82  72
## 2 2000      2Ge+her The Hardest Part Of ... 3:15   2000-09-02  91  87  92
## 3 2000 3 Doors Down               Kryptonite 3:53   2000-04-08  81  70  68
## 4 2000 3 Doors Down                    Loser 4:24   2000-10-21  76  76  72
## 5 2000     504 Boyz            Wobble Wobble 3:35   2000-04-15  57  34  25
## 6 2000          98? Give Me Just One Nig... 3:24   2000-08-19  51  39  34
##   wk4 wk5 wk6 wk7 wk8 wk9 wk10 wk11 wk12 wk13 wk14 wk15 wk16 wk17 wk18
## 1  77  87  94  99  NA  NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 2  NA  NA  NA  NA  NA  NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 3  67  66  57  54  53  51   51   51   51   47   44   38   28   22   18
## 4  69  67  65  55  59  62   61   61   59   61   66   72   76   75   67
## 5  17  17  31  36  49  53   57   64   70   75   76   78   85   92   96
## 6  26  26  19   2   2   3    6    7   22   29   36   47   67   66   84
##   wk19 wk20 wk21 wk22 wk23 wk24 wk25 wk26 wk27 wk28 wk29 wk30 wk31 wk32
## 1   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 2   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 3   18   14   12    7    6    6    6    5    5    4    4    4    4    3
## 4   73   70   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 5   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 6   93   94   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   wk33 wk34 wk35 wk36 wk37 wk38 wk39 wk40 wk41 wk42 wk43 wk44 wk45 wk46
## 1   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 2   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 3    3    3    4    5    5    9    9   15   14   13   14   16   17   21
## 4   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 5   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 6   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   wk47 wk48 wk49 wk50 wk51 wk52 wk53 wk54 wk55 wk56 wk57 wk58 wk59 wk60
```

```
## 1   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 2   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 3   22   24   28   33   42   42   49   NA   NA   NA   NA   NA   NA   NA
## 4   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 5   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 6   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   wk61 wk62 wk63 wk64 wk65 wk66 wk67 wk68 wk69 wk70 wk71 wk72 wk73 wk74
## 1   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 2   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 3   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 4   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 5   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
## 6   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA   NA
##   wk75 wk76
## 1   NA   NA
## 2   NA   NA
## 3   NA   NA
## 4   NA   NA
## 5   NA   NA
## 6   NA   NA
```

```
head(table8)
```

```
##   year artist time          track       date week rank
## 1 2000  2 Pac 4:22 Baby Don't Cry 2000-02-26    1   87
## 2 2000  2 Pac 4:22 Baby Don't Cry 2000-03-04    2   82
## 3 2000  2 Pac 4:22 Baby Don't Cry 2000-03-11    3   72
## 4 2000  2 Pac 4:22 Baby Don't Cry 2000-03-18    4   77
## 5 2000  2 Pac 4:22 Baby Don't Cry 2000-03-25    5   87
## 6 2000  2 Pac 4:22 Baby Don't Cry 2000-04-01    6   94
```