Assignment_04 Final

Steven Tran

February 28, 2018

```r
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
```

```r
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.

```r
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>
```

```r
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", "$40-50k
  "$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>
```

```r
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

```r
table7 <- read_csv("billboard.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   artist.inverted = col_character(),
##   track = col_character(),
##   time = col_time(format = ""),
##   genre = col_character(),
##   date.entered = col_date(format = ""),
##   date.peaked = col_date(format = ""),
##   x66th.week = col_character(),
##   x67th.week = col_character(),
##   x68th.week = col_character(),
##   x69th.week = col_character(),
##   x70th.week = col_character(),
##   x71st.week = col_character(),
##   x72nd.week = col_character(),
##   x73rd.week = col_character(),
##   x74th.week = col_character(),
##   x75th.week = col_character(),
##   x76th.week = col_character()
## )

## See spec(...) for full column specifications.
```

```r
table8 <- table7 %>%
  gather(key="week", value = "rank", -year, -artist.inverted, -track, -time, -genre, -date.entered, -dat
  select(year, artist=artist.inverted, time, track, date = date.entered, week, rank ) %>%            ar
  filter(!is.na(rank)) %>%
  separate(week, into=c("A", "B", "C"), sep=c(1, -7), convert=TRUE) %>%
  select(-A, -C) %>%
  dplyr::rename(week = B) %>%
  arrange(artist, track) %>%
  mutate(date = date + (week-1)*7 ) %>%
  mutate(rank = as.integer(rank))

head(table7)
```

```
## # A tibble: 6 x 83
##    year artist.inverted  track        time  genre date.entered date.peaked
##   <int> <chr>            <chr>        <tim> <chr> <date>       <date>
## 1  2000 Destiny's Child  Independent~ 03:38 Rock  2000-09-23   2000-11-18
## 2  2000 Santana          Maria, Maria 04:18 Rock  2000-02-12   2000-04-08
## 3  2000 Savage Garden    I Knew I Lo~ 04:07 Rock  1999-10-23   2000-01-29
## 4  2000 Madonna          Music        03:45 Rock  2000-08-12   2000-09-16
## 5  2000 Aguilera, Chris~ Come On Ove~ 03:38 Rock  2000-08-05   2000-10-14
## 6  2000 Janet            Doesn't Rea~ 04:17 Rock  2000-06-17   2000-08-26
## # ... with 76 more variables: x1st.week <int>, x2nd.week <int>,
## #   x3rd.week <int>, x4th.week <int>, x5th.week <int>, x6th.week <int>,
## #   x7th.week <int>, x8th.week <int>, x9th.week <int>, x10th.week <int>,
## #   x11th.week <int>, x12th.week <int>, x13th.week <int>,
## #   x14th.week <int>, x15th.week <int>, x16th.week <int>,
## #   x17th.week <int>, x18th.week <int>, x19th.week <int>,
## #   x20th.week <int>, x21st.week <int>, x22nd.week <int>,
## #   x23rd.week <int>, x24th.week <int>, x25th.week <int>,
## #   x26th.week <int>, x27th.week <int>, x28th.week <int>,
## #   x29th.week <int>, x30th.week <int>, x31st.week <int>,
## #   x32nd.week <int>, x33rd.week <int>, x34th.week <int>,
## #   x35th.week <int>, x36th.week <int>, x37th.week <int>,
## #   x38th.week <int>, x39th.week <int>, x40th.week <int>,
## #   x41st.week <int>, x42nd.week <int>, x43rd.week <int>,
## #   x44th.week <int>, x45th.week <int>, x46th.week <int>,
## #   x47th.week <int>, x48th.week <int>, x49th.week <int>,
## #   x50th.week <int>, x51st.week <int>, x52nd.week <int>,
## #   x53rd.week <int>, x54th.week <int>, x55th.week <int>,
## #   x56th.week <int>, x57th.week <int>, x58th.week <int>,
## #   x59th.week <int>, x60th.week <int>, x61st.week <int>,
## #   x62nd.week <int>, x63rd.week <int>, x64th.week <int>,
## #   x65th.week <int>, x66th.week <chr>, x67th.week <chr>,
## #   x68th.week <chr>, x69th.week <chr>, x70th.week <chr>,
## #   x71st.week <chr>, x72nd.week <chr>, x73rd.week <chr>,
## #   x74th.week <chr>, x75th.week <chr>, x76th.week <chr>
```

```r
head(table8)
```

```
## # A tibble: 6 x 7
##    year artist time   track                         date        week  rank
##   <int> <chr>  <time> <chr>                         <date>     <int> <int>
## 1  2000 2 Pac  04:22  Baby Don't Cry (Keep Ya Head~ 2000-02-26     1    87
## 2  2000 2 Pac  04:22  Baby Don't Cry (Keep Ya Head~ 2000-03-04     2    82
## 3  2000 2 Pac  04:22  Baby Don't Cry (Keep Ya Head~ 2000-03-11     3    72
## 4  2000 2 Pac  04:22  Baby Don't Cry (Keep Ya Head~ 2000-03-18     4    77
## 5  2000 2 Pac  04:22  Baby Don't Cry (Keep Ya Head~ 2000-03-25     5    87
## 6  2000 2 Pac  04:22  Baby Don't Cry (Keep Ya Head~ 2000-04-01     6    94
```