homework

Yuehan Xiao

2/26/2018

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.4.2

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

## <U+221A> ggplot2 2.2.1     <U+221A> purrr   0.2.4
## <U+221A> tibble  1.4.2     <U+221A> dplyr   0.7.4
## <U+221A> tidyr   0.8.0     <U+221A> stringr 1.3.0
## <U+221A> readr   1.1.1     <U+221A> forcats 0.2.0

## Warning: package 'tibble' was built under R version 3.4.3

## Warning: package 'tidyr' was built under R version 3.4.3

## Warning: package 'purrr' was built under R version 3.4.2

## Warning: package 'dplyr' was built under R version 3.4.2

## Warning: package 'stringr' was built under R version 3.4.3

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

```
library(tibble)
```

10.5.5 it converts named atomic vectors or lists to two-column data frames.For unnamed vectors, the natural sequence is used as name column. If the data is having a named list that you need to convert into dataframe, then you can use this code.

12.6

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

```
## Observations: 76,046
## Variables: 6
## $ country <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanis...
## $ iso2    <chr> "AF", "AF", "AF", "AF", "AF", "AF", "AF", "AF", "AF", ...
## $ iso3    <chr> "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG"...
## $ year    <int> 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, ...
## $ key     <chr> "new_sp_m014", "new_sp_m014", "new_sp_m014", "new_sp_m...
## $ cases   <int> 0, 30, 8, 52, 129, 90, 127, 139, 151, 193, 186, 187, 2...
```

```
who2 <- who1 %>%
 mutate(key = stringr::str_replace(key, "newrel", "new_rel"))
who3 <- who2 %>%
  separate(key, c("new", "type", "sexage"), sep = "_")
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)
```

12.6.1.3

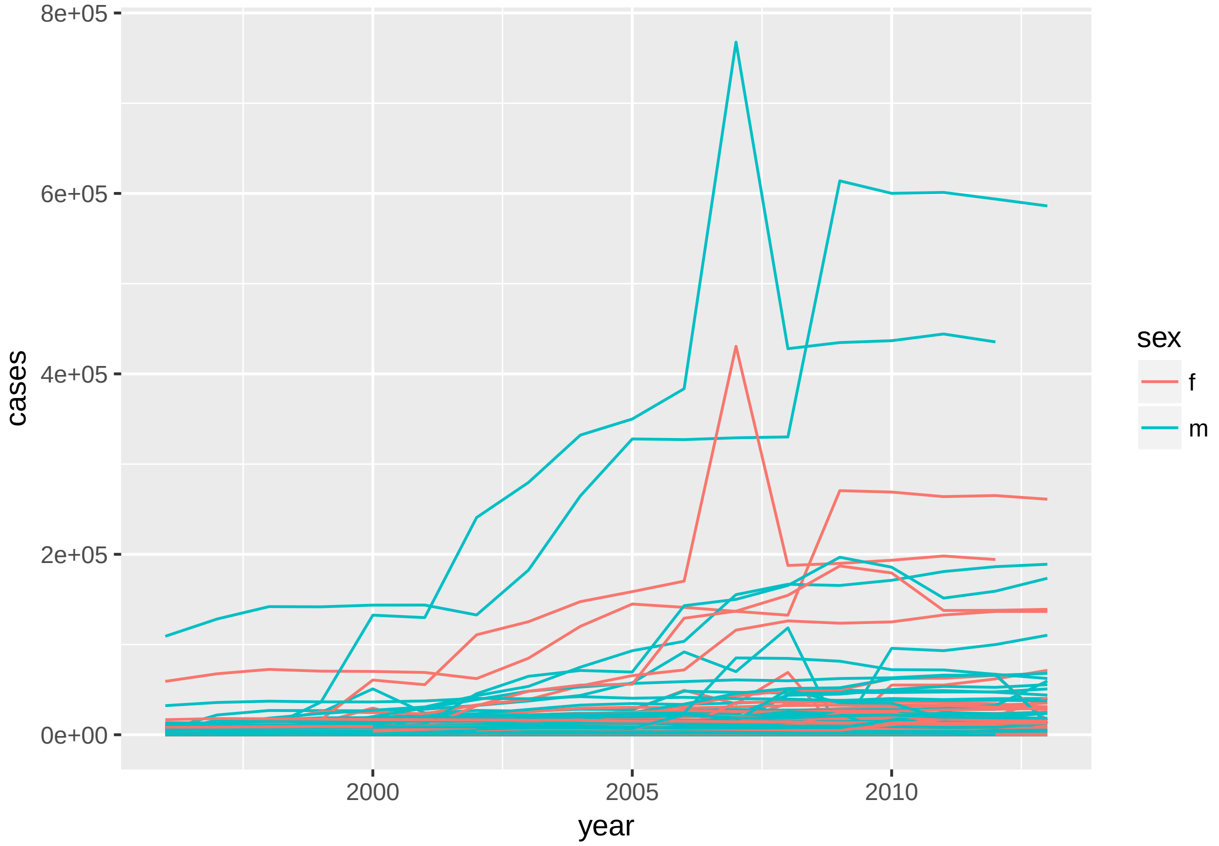
```
select(who3, country, iso2, iso3) %>%
  distinct() %>%
  group_by(country) %>%
  filter(n() > 1)
```

```
## # A tibble: 0 x 3
## # Groups:   country [0]
## # ... with 3 variables: country <chr>, iso2 <chr>, iso3 <chr>
```

So iso2 and iso3 are redudant with country.

12.6.4

```
who5 %>%
  group_by(country, year, sex) %>%
  filter(year > 1995) %>%
  summarise(cases = sum(cases)) %>%
  unite(country_sex, country, sex, remove = FALSE) %>%
  ggplot(aes(x = year, y = cases, group = country_sex, colour = sex)) +
  geom_line()
```



3.convert table 4 to table 6

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

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

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.4.3

##
```

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

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

```r
source("xtable.r")

pew<-read.spss("pew.sav", header=TRUE,stringsAsFactors = FALSE )
```

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

```
## Warning in read.spss("pew.sav", header = TRUE, stringsAsFactors = FALSE):
## Duplicated levels in factor denom: Electronic ministries
```

```
## Warning in read.spss("pew.sav", header = TRUE, stringsAsFactors = FALSE):
## 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", header = TRUE, stringsAsFactors = FALSE):
## 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
```

```r
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"))
counts <- count(religion, c("reltrad", "income"))
names(counts)[1] <- "religion"
head(counts,10)
```

```
##    religion             income freq
## 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
## 7  Agnostic           $75-100k  122
## 8  Agnostic          $100-150k  109
## 9  Agnostic              >150k   84
## 10 Agnostic Don't know/refused   96
```

4. convert table 7 to table 8

```
options(stringsAsFactors = FALSE)
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.4.2

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:plyr':
##
##     here

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

```
library(reshape2)
library(stringr)
library(plyr)
source("xtable.r")
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':
##
##     set_names

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

```
library(tidyr)

bb <-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.
```

```
bb.1 <- bb %>% gather(key="week", value ="rank", -year, -artist.inverted, -track, -time, -genre, -date.e
bb.2 <- bb.1 %>% select(year, artist=artist.inverted, time, track, date=date.entered, week, rank)
bb.3 <- bb.2 %>% arrange(track)
bb.4 <- bb.3 %>% filter(!is.na(rank))
#keep the one is not in na
bb.5 <-bb.4 %>% separate(week, into=c("A", "week", "C"), sep=c(1, -7), convert=TRUE)
bb.6 <-bb.5 %>% select(-A, -C)
bb.7 <-bb.6%>% filter(!is.na(week))
#must to specified rename in the dplyr
bb.8 <- bb.7 %>% arrange(artist, track)
bb.9 <-bb.8 %>% mutate(date = date+(week-1)*7)
bb.10 <- bb.9 %>% mutate(rank = as.integer(rank))
bb.11 <- as.data.frame(bb.10)
head(bb.11,10)
```

```
##    year  artist     time
## 1  2000   2 Pac 04:22:00
## 2  2000   2 Pac 04:22:00
## 3  2000   2 Pac 04:22:00
## 4  2000   2 Pac 04:22:00
## 5  2000   2 Pac 04:22:00
## 6  2000   2 Pac 04:22:00
## 7  2000   2 Pac 04:22:00
## 8  2000 2Ge+her 03:15:00
## 9  2000 2Ge+her 03:15:00
## 10 2000 2Ge+her 03:15:00
##                                                            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) 2000-03-04
## 3                            Baby Don't Cry (Keep Ya Head Up II) 2000-03-11
## 4                            Baby Don't Cry (Keep Ya Head Up II) 2000-03-18
## 5                            Baby Don't Cry (Keep Ya Head Up II) 2000-03-25
## 6                            Baby Don't Cry (Keep Ya Head Up II) 2000-04-01
## 7                            Baby Don't Cry (Keep Ya Head Up II) 2000-04-08
## 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) 2000-09-09
## 10 The Hardest Part Of Breaking Up (Is Getting Back Your Stuff) 2000-09-16
##    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
```