HW4

Ozair Meghani

4/27/2018

12.6.1

problem 3

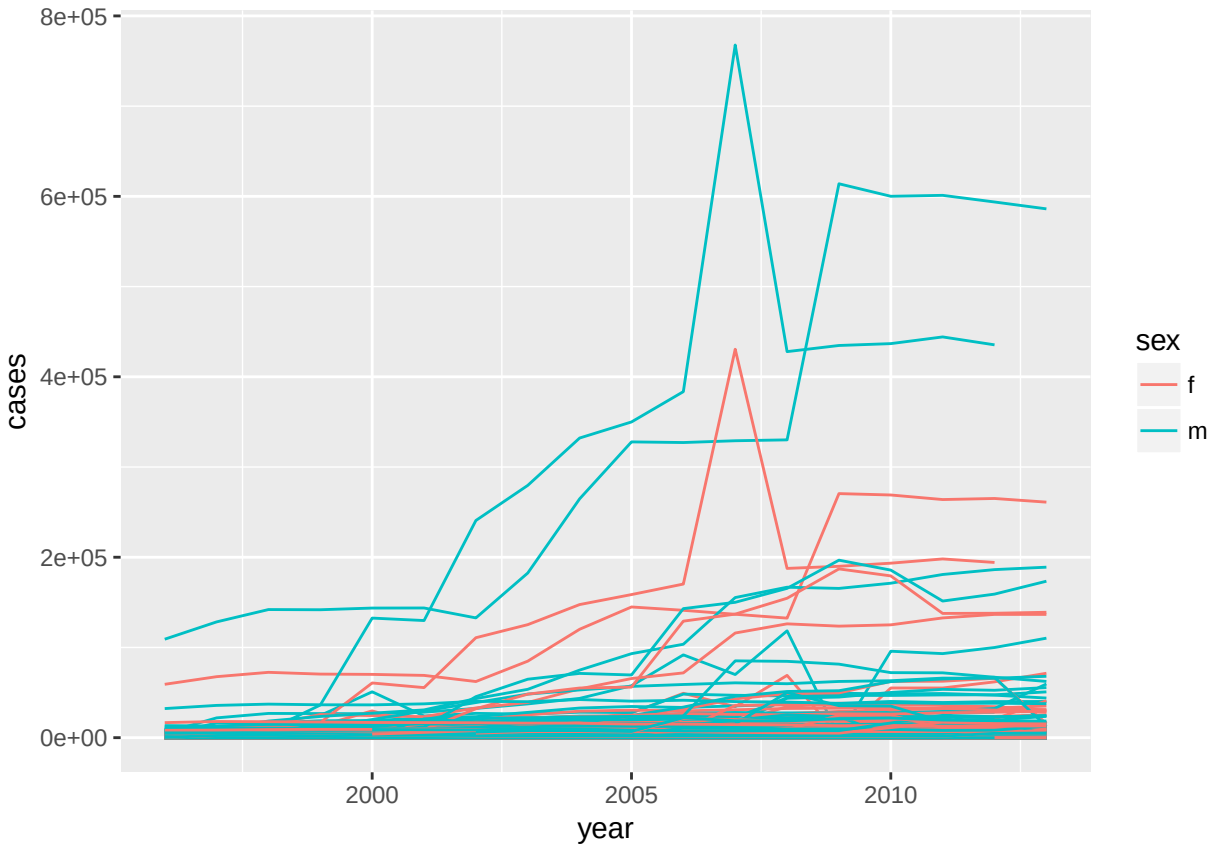
Those columns are abreviations of the countries, so it is repetitive to have the same information under three different columns.

```
## # A tibble: 219 x 3
## # Groups:   country [219]
##    country             iso2  iso3
##    <chr>               <chr> <chr>
##  1 Afghanistan         AF    AFG
##  2 Albania             AL    ALB
##  3 Algeria             DZ    DZA
##  4 American Samoa      AS    ASM
##  5 Andorra             AD    AND
##  6 Angola              AO    AGO
##  7 Anguilla            AI    AIA
##  8 Antigua and Barbuda AG    ATG
##  9 Argentina           AR    ARG
## 10 Armenia             AM    ARM
## # ... with 209 more rows
```

12.6.1

Problem 4

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



10.5.

Problem 5

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

converts named atomic vectors or lists to two-column data frames. For unnamed vectors, the natural sequence is used as name column.

```
x <- c(1,2,3,5,6)
enframe(x, name = "name", value = "value")
enframe(c(a = 5, b = 7))
```

Problem 3 - converting table 4 to table 6

```
## 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
```

reltrad	income	count
Agnostic	<$10k	27
Agnostic	$10-20k	34
Agnostic	$20-30k	60
Agnostic	$30-40k	81
Agnostic	$40-50k	76
Agnostic	$50-75k	137
Agnostic	$75-100k	122
Agnostic	$100-150k	109
Agnostic	>150k	84
Agnostic	Don't know/refused	96

4.convert table 7 to table 8

```
## 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.
```

```
## Warning in format.POSIXlt(as.POSIXlt(x), ...): unknown timezone 'zone/tz/
## 2018c.1.0/zoneinfo/America/New_York'
```

year	artist	time	track	date	week	rank
2000	3 Doors Down	03:53:00	Kryptonite	2000-04-08	1	51
2000	3 Doors Down	03:53:00	Kryptonite	2000-04-08	1	51
2000	3 Doors Down	03:53:00	Kryptonite	2000-04-08	1	51
2000	3 Doors Down	03:53:00	Kryptonite	2000-04-08	1	47
2000	3 Doors Down	03:53:00	Kryptonite	2000-04-08	1	44
2000	3 Doors Down	03:53:00	Kryptonite	2000-04-08	1	38
2000	3 Doors Down	03:53:00	Kryptonite	2000-04-08	1	28
2000	3 Doors Down	03:53:00	Kryptonite	2000-04-08	1	22
2000	3 Doors Down	03:53:00	Kryptonite	2000-04-08	1	18
2000	3 Doors Down	03:53:00	Kryptonite	2000-04-08	1	18