

Assignment 4 updated

Maria Ren

2/23/2018

12.6.1 problems 3 and 4

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse
## √ ggplot2 2.2.1      √ purrr  0.2.4
## √ tibble  1.4.2      √ dplyr  0.7.4
## √ tidyr   0.7.2      √ stringr 1.2.0
## √ readr   1.1.1      √ forcats 0.2.0

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

```
library(dplyr)
library(tidyr)
```

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

```
# Original Data and textbook code
```

```
tidyr::who
```

```
## # A tibble: 7,240 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
## 7 Afghanistan AF    AFG   1986         NA            NA            NA
## 8 Afghanistan AF    AFG   1987         NA            NA            NA
## 9 Afghanistan AF    AFG   1988         NA            NA            NA
## 10 Afghanistan AF    AFG   1989         NA            NA            NA
## # ... with 7,230 more rows, and 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>

who1 <- who %>%
  gather(new_sp_m014:newrel_f65, key = "key", value = "cases", na.rm = TRUE)
who2 <- who1 %>%
  mutate(key = stringr::str_replace(key, "newrel", "new_rel"))
who3 <- who2 %>%
  separate(key, c("new", "type", "sexage"), sep = "_")
who4 <- who3 %>%
  select(-new, -iso2, -iso3)
who5 <- who4 %>%
  separate(sexage, c("sex", "age"), sep = 1)

a <- select(who3, country, iso2, iso3)
b <- unique.data.frame(a) %>%
  # select unique rows from the data group of who3, country, iso2 and iso3
  group_by(country) %>%
  filter(n() > 1)
b
```

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

When we group together the three columns country, iso2 and iso3, and try to find unique rows from the data, we found that none of the values in the columns have different values from each other, therefore these three columns are redundant.

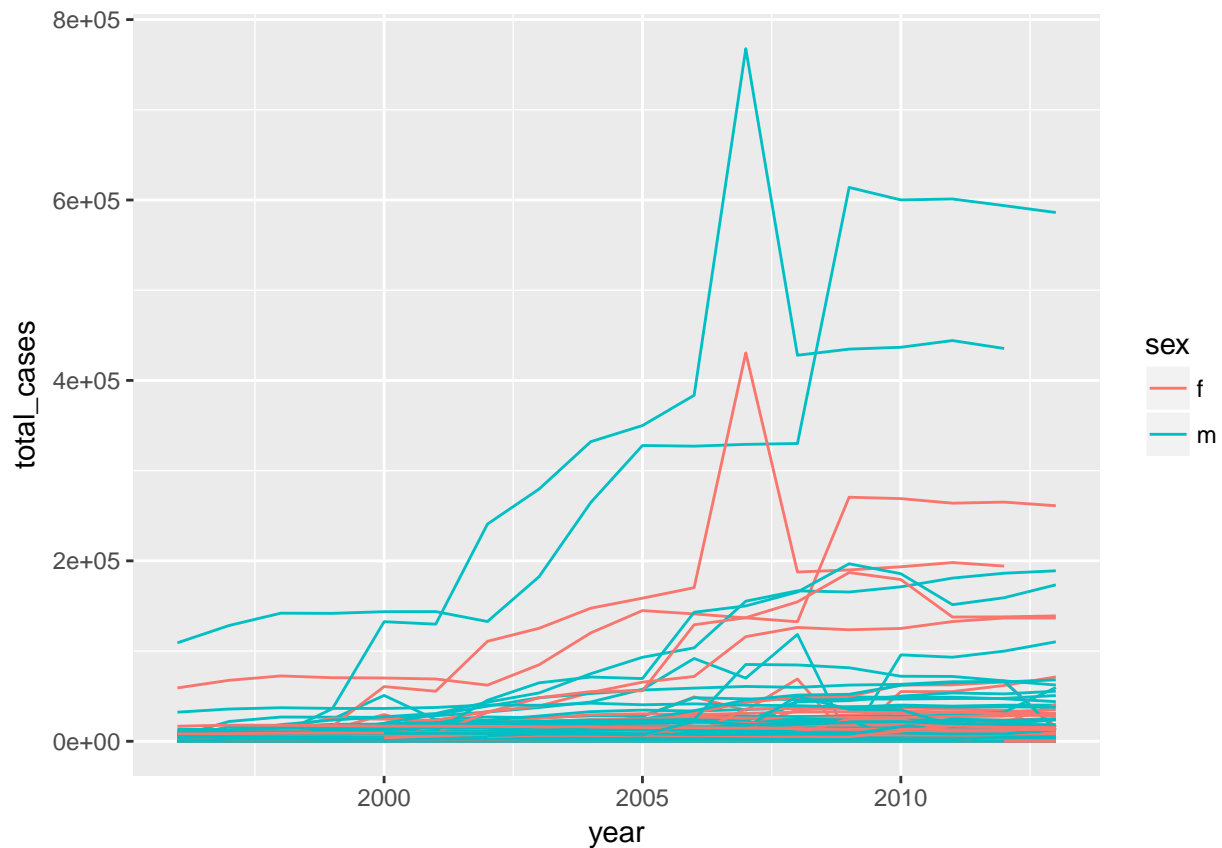
- (4) For each country, year, and sex compute the total number of cases of TB. Make an informative visualisation of the data.

```
who5 %>%
  group_by(country,sex,year) %>%
  summarize(total_cases=sum(cases))

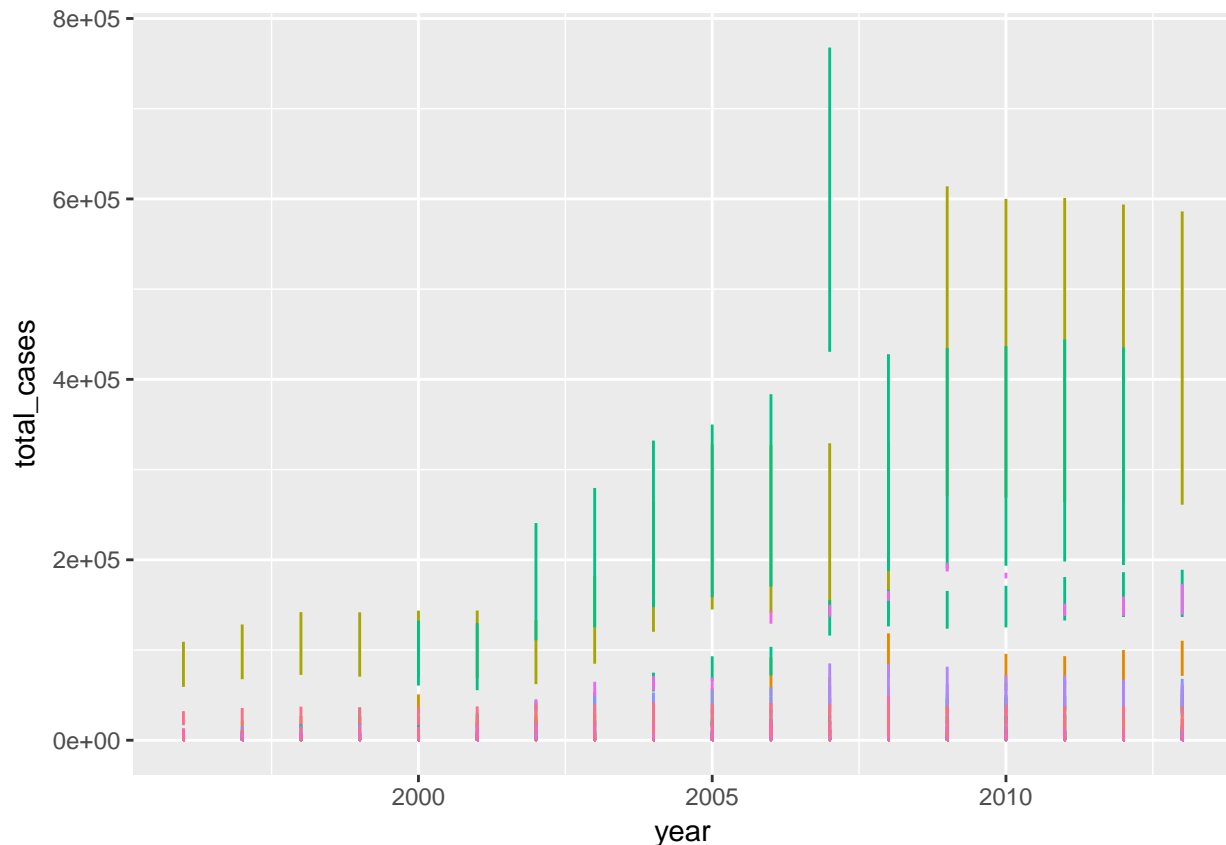
## # A tibble: 6,921 x 4
## # Groups:   country, sex [?]
##   country    sex  year total_cases
##   <chr>      <chr> <int>      <int>
## 1 Afghanistan f    1997        102
## 2 Afghanistan f    1998       1207
## 3 Afghanistan f    1999        517
## 4 Afghanistan f    2000       1751
## 5 Afghanistan f    2001       3062
## 6 Afghanistan f    2002       4418
## 7 Afghanistan f    2003       4423
## 8 Afghanistan f    2004       5587
## 9 Afghanistan f    2005       6818
## 10 Afghanistan f    2006       8520
## # ... with 6,911 more rows
```

Total_cases gives the total number of cases of TB for each country, each sex in each year

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



```
who5 %>%
  group_by(country, year, sex) %>%
  filter(year > 1995) %>%
  summarise(total_cases=sum(cases)) %>%
  unite(year_country, year, country, remove = FALSE) %>%
  ggplot(aes(x = year, y = total_cases, group = year_country, color = country)) +
  geom_line(show.legend = FALSE)
```



Looking at both graph, the first one shows that amount of cases of TB found in male seems to be more than female. From the second graph, we can see that the number of TB cases is especially high around the year of 2007.

10.5 : problem 5

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

```
x <- c(m=2,n=9)
tibble::enframe(x)
```

```
## # A tibble: 2 x 2
##   name  value
##   <chr> <dbl>
## 1 m      2.00
## 2 n      9.00
```

`tibble::enframe()` turns a vector with names into a tibble with two columns, as shown from above. You can use it when you have a named data vector, and you want to add that to another data frame.

Tidy Data Article :

3) table 4 to table 6

```
# Get data from github for table 4
```

```
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)

##
## Attaching package: 'reshape2'

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

source("xtable.r")

# Data from http://pewforum.org/Datasets/Dataset-Download.aspx

# Load data -----

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"))

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

xtable(counts[1:10, ], file = "pew-clean.tex")

# Convert into the form in which I originally saw it -----

raw <- dcast(counts, religion ~ income)

## Using freq as value column: use value.var to override.
xtable(raw[1:10, 1:7], file = "pew-row.tex")

### HW added line

saveRDS(raw, "rawdat.RDS")
table4 <- raw
table4

```

	religion	<\$10k	\$10-20k	\$20-30k	\$30-40k	\$40-50k	\$50-75k
## 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/refused	15	14	15	11	10	35
## 6	Evangelical Prot	575	869	1064	982	881	1486
## 7	Hindu	1	9	7	9	11	34
## 8	Historically Black Prot	228	244	236	238	197	223
## 9	Jehovah's Witness	20	27	24	24	21	30
## 10	Jewish	19	19	25	25	30	95
## 11	Mainline Prot	289	495	619	655	651	1107

```
## 12          Mormon      29      40      48      51      56      112
## 13          Muslim       6       7       9      10       9       23
## 14          Orthodox    13      17      23      32      32      47
## 15      Other Christian   9       7      11      13      13      14
## 16          Other Faiths 20      33      40      46      49      63
## 17 Other World Religions  5       2       3       4       2       7
## 18          Unaffiliated 217     299     374     365     341     528
##      $75-100k $100-150k >150k Don't know/refused
## 1          122      109      84          96
## 2           73       59      74          76
## 3           62       39      53          54
## 4          949      792     633         1489
## 5           21       17      18          116
## 6          949      723     414         1529
## 7           47       48      54           37
## 8          131       81      78          339
## 9           15       11       6           37
## 10          69       87     151          162
## 11         939      753     634         1328
## 12          85       49      42           69
## 13          16        8       6           22
## 14          38       42      46           73
## 15          18       14      12           18
## 16          46       40      41           71
## 17           3        4       4            8
## 18         407      321     258          597
```

Turning table 4 to table 6

```
a <- melt(data=table4)
```

```
## Using religion as id variables
```

```
a <- a[order(a["religion"]),]
colnames(a)[colnames(a)=="variable"] <- "income"
colnames(a)[colnames(a)=="value"] <- "freq"
arrange(a,religion,income,freq)
```

```
##      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
## 11     Atheist     <$10k   12
## 12     Atheist     $10-20k  27
## 13     Atheist     $20-30k  37
## 14     Atheist     $30-40k  52
## 15     Atheist     $40-50k  35
## 16     Atheist     $50-75k  70
## 17     Atheist     $75-100k  73
```

## 18	Atheist	\$100-150k	59
## 19	Atheist	>150k	74
## 20	Atheist	Don't know/refused	76
## 21	Buddhist	<\$10k	27
## 22	Buddhist	\$10-20k	21
## 23	Buddhist	\$20-30k	30
## 24	Buddhist	\$30-40k	34
## 25	Buddhist	\$40-50k	33
## 26	Buddhist	\$50-75k	58
## 27	Buddhist	\$75-100k	62
## 28	Buddhist	\$100-150k	39
## 29	Buddhist	>150k	53
## 30	Buddhist	Don't know/refused	54
## 31	Catholic	<\$10k	418
## 32	Catholic	\$10-20k	617
## 33	Catholic	\$20-30k	732
## 34	Catholic	\$30-40k	670
## 35	Catholic	\$40-50k	638
## 36	Catholic	\$50-75k	1116
## 37	Catholic	\$75-100k	949
## 38	Catholic	\$100-150k	792
## 39	Catholic	>150k	633
## 40	Catholic	Don't know/refused	1489
## 41	Don't know/refused	<\$10k	15
## 42	Don't know/refused	\$10-20k	14
## 43	Don't know/refused	\$20-30k	15
## 44	Don't know/refused	\$30-40k	11
## 45	Don't know/refused	\$40-50k	10
## 46	Don't know/refused	\$50-75k	35
## 47	Don't know/refused	\$75-100k	21
## 48	Don't know/refused	\$100-150k	17
## 49	Don't know/refused	>150k	18
## 50	Don't know/refused	Don't know/refused	116
## 51	Evangelical Prot	<\$10k	575
## 52	Evangelical Prot	\$10-20k	869
## 53	Evangelical Prot	\$20-30k	1064
## 54	Evangelical Prot	\$30-40k	982
## 55	Evangelical Prot	\$40-50k	881
## 56	Evangelical Prot	\$50-75k	1486
## 57	Evangelical Prot	\$75-100k	949
## 58	Evangelical Prot	\$100-150k	723
## 59	Evangelical Prot	>150k	414
## 60	Evangelical Prot	Don't know/refused	1529
## 61	Hindu	<\$10k	1
## 62	Hindu	\$10-20k	9
## 63	Hindu	\$20-30k	7
## 64	Hindu	\$30-40k	9
## 65	Hindu	\$40-50k	11
## 66	Hindu	\$50-75k	34
## 67	Hindu	\$75-100k	47
## 68	Hindu	\$100-150k	48
## 69	Hindu	>150k	54
## 70	Hindu	Don't know/refused	37
## 71	Historically Black Prot	<\$10k	228

## 72	Historically Black Prot	\$10-20k	244
## 73	Historically Black Prot	\$20-30k	236
## 74	Historically Black Prot	\$30-40k	238
## 75	Historically Black Prot	\$40-50k	197
## 76	Historically Black Prot	\$50-75k	223
## 77	Historically Black Prot	\$75-100k	131
## 78	Historically Black Prot	\$100-150k	81
## 79	Historically Black Prot	>150k	78
## 80	Historically Black Prot	Don't know/refused	339
## 81	Jehovah's Witness	<\$10k	20
## 82	Jehovah's Witness	\$10-20k	27
## 83	Jehovah's Witness	\$20-30k	24
## 84	Jehovah's Witness	\$30-40k	24
## 85	Jehovah's Witness	\$40-50k	21
## 86	Jehovah's Witness	\$50-75k	30
## 87	Jehovah's Witness	\$75-100k	15
## 88	Jehovah's Witness	\$100-150k	11
## 89	Jehovah's Witness	>150k	6
## 90	Jehovah's Witness	Don't know/refused	37
## 91	Jewish	<\$10k	19
## 92	Jewish	\$10-20k	19
## 93	Jewish	\$20-30k	25
## 94	Jewish	\$30-40k	25
## 95	Jewish	\$40-50k	30
## 96	Jewish	\$50-75k	95
## 97	Jewish	\$75-100k	69
## 98	Jewish	\$100-150k	87
## 99	Jewish	>150k	151
## 100	Jewish	Don't know/refused	162
## 101	Mainline Prot	<\$10k	289
## 102	Mainline Prot	\$10-20k	495
## 103	Mainline Prot	\$20-30k	619
## 104	Mainline Prot	\$30-40k	655
## 105	Mainline Prot	\$40-50k	651
## 106	Mainline Prot	\$50-75k	1107
## 107	Mainline Prot	\$75-100k	939
## 108	Mainline Prot	\$100-150k	753
## 109	Mainline Prot	>150k	634
## 110	Mainline Prot	Don't know/refused	1328
## 111	Mormon	<\$10k	29
## 112	Mormon	\$10-20k	40
## 113	Mormon	\$20-30k	48
## 114	Mormon	\$30-40k	51
## 115	Mormon	\$40-50k	56
## 116	Mormon	\$50-75k	112
## 117	Mormon	\$75-100k	85
## 118	Mormon	\$100-150k	49
## 119	Mormon	>150k	42
## 120	Mormon	Don't know/refused	69
## 121	Muslim	<\$10k	6
## 122	Muslim	\$10-20k	7
## 123	Muslim	\$20-30k	9
## 124	Muslim	\$30-40k	10
## 125	Muslim	\$40-50k	9

## 126	Muslim	\$50-75k	23
## 127	Muslim	\$75-100k	16
## 128	Muslim	\$100-150k	8
## 129	Muslim	>150k	6
## 130	Muslim	Don't know/refused	22
## 131	Orthodox	<\$10k	13
## 132	Orthodox	\$10-20k	17
## 133	Orthodox	\$20-30k	23
## 134	Orthodox	\$30-40k	32
## 135	Orthodox	\$40-50k	32
## 136	Orthodox	\$50-75k	47
## 137	Orthodox	\$75-100k	38
## 138	Orthodox	\$100-150k	42
## 139	Orthodox	>150k	46
## 140	Orthodox	Don't know/refused	73
## 141	Other Christian	<\$10k	9
## 142	Other Christian	\$10-20k	7
## 143	Other Christian	\$20-30k	11
## 144	Other Christian	\$30-40k	13
## 145	Other Christian	\$40-50k	13
## 146	Other Christian	\$50-75k	14
## 147	Other Christian	\$75-100k	18
## 148	Other Christian	\$100-150k	14
## 149	Other Christian	>150k	12
## 150	Other Christian	Don't know/refused	18
## 151	Other Faiths	<\$10k	20
## 152	Other Faiths	\$10-20k	33
## 153	Other Faiths	\$20-30k	40
## 154	Other Faiths	\$30-40k	46
## 155	Other Faiths	\$40-50k	49
## 156	Other Faiths	\$50-75k	63
## 157	Other Faiths	\$75-100k	46
## 158	Other Faiths	\$100-150k	40
## 159	Other Faiths	>150k	41
## 160	Other Faiths	Don't know/refused	71
## 161	Other World Religions	<\$10k	5
## 162	Other World Religions	\$10-20k	2
## 163	Other World Religions	\$20-30k	3
## 164	Other World Religions	\$30-40k	4
## 165	Other World Religions	\$40-50k	2
## 166	Other World Religions	\$50-75k	7
## 167	Other World Religions	\$75-100k	3
## 168	Other World Religions	\$100-150k	4
## 169	Other World Religions	>150k	4
## 170	Other World Religions	Don't know/refused	8
## 171	Unaffiliated	<\$10k	217
## 172	Unaffiliated	\$10-20k	299
## 173	Unaffiliated	\$20-30k	374
## 174	Unaffiliated	\$30-40k	365
## 175	Unaffiliated	\$40-50k	341
## 176	Unaffiliated	\$50-75k	528
## 177	Unaffiliated	\$75-100k	407
## 178	Unaffiliated	\$100-150k	321
## 179	Unaffiliated	>150k	258

4) table 7 to table 8

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

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

```
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)
```

[illegible]

```
## 6      84      93      94      NA      NA      NA      NA      NA      NA      NA      NA      NA      NA      NA
##      wk32 wk33 wk34 wk35 wk36 wk37 wk38 wk39 wk40 wk41 wk42 wk43 wk44 wk45
## 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       3       4       5       5       9       9      15      14      13      14      16      17
## 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
##      wk46 wk47 wk48 wk49 wk50 wk51 wk52 wk53 wk54 wk55 wk56 wk57 wk58 wk59
## 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      21      22      24      28      33      42      42      49      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
##      wk60 wk61 wk62 wk63 wk64 wk65 wk66 wk67 wk68 wk69 wk70 wk71 wk72 wk73
## 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
##      wk74 wk75 wk76
## 1      NA      NA      NA
## 2      NA      NA      NA
## 3      NA      NA      NA
## 4      NA      NA      NA
## 5      NA      NA      NA
## 6      NA      NA      NA
```

```
# Data only shows the first couple of rows
```

```
# Table 8
```

```
library(tidyverse)
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)
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))
bb.5 <- bb.4 %>% separate(week, into=c("A", "B", "C"), sep=c(1, -8), convert=TRUE)
bb.6 <- bb.5 %>% select(-A, -C)
bb.7 <- bb.6 %>% dplyr::rename(week = B)
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.10 <- bb.9 %>% mutate(time = as.character(time))
bb.10$time <- gsub(':00', '', bb.10$time)
head(bb.10)

## # A tibble: 6 x 7
##   year artist time track date week rank
##   <int> <chr> <chr> <chr> <date> <int> <chr>
## 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

# Data only shows the first couple of rows

```