Tidyverse Problem Set

MA615

September 29, 2019

The purpose of this problem set is to provide data contexts in which to exercise the capabilitiues of the tidyverse. While some questons require specific answers, other parts of the problems have been written to be purposely ambiguous, requiring you to think through the presentation details of your answer.

HOLD THE PRESSES!
As I was preparing to post these problems yesterday, I noticed that tidyr had been updata in the last few weeks. I was looking for more exercises on gather() and spread() – which are always difficult to master. And I found that they have been superceded!! Why do I love working with R as the tidyversie is on a path of continuous improvement? Because the improvements come from developers who write things like this:

For some time, it's been obvious that there is something fundamentally wrong with the design of spread() and gather(). Many people don't find the names intuitive and find it hard to remember which direction corresponds to spreading and which to gathering. It also seems surprisingly hard to remember the arguments to these functions, meaning that many people (including me!) have to consult the documentation every time. Hadley Wickham, Pivot Vingette

So... before you do anymore tidyverse exercises, Read this tidyr 1.0.0.

Then go to the tidyr cran page and to the examples and exercies in the new vignettes.

In your solutions to the problems below, if you need to use table reshaping functions from TidyR, be sure that you use pivot_longer(), and pivot_wider().

```
library(shiny)
library(tidyr)
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 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

```
library(knitr)
library(ggplot2)
```

```
library(esquisse)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(magrittr)
```

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

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

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

```
opts_chunk$set(echo = FALSE)
```

Problem 1

Load the gapminder data from the gapminder package.

How many continents are included in the data set?

```
length(unique(gapminder$continent))
```

```
## [1] 5
```

How many countrys are included? How many countries per continent?

```
length(unique(gapminder$country))
```

```
## [1] 142
```

```
gapminder %>% group_by(continent) %>% summarise_each(n_distinct)
```

```
## # A tibble: 5 x 6
##   continent country  year lifeExp   pop gdpPercap
##   <fct>       <int> <int>   <int> <int>     <int>
## 1 Africa         52    12     619   624       624
## 2 Americas       25    12     299   300       300
## 3 Asia           33    12     393   396       396
## 4 Europe         30    12     326   360       360
## 5 Oceania         2    12      24    24        24
```

Using the gapminder data, produce a report showing the continents in the dataset, total population per continent, and GDP per capita. Be sure that the table is properly labeled and suitable for inclusion in a printed report.

```
data(gapminder)     ## load the data
pop_sum <- round(tapply(gapminder$pop,gapminder$continent,sum),2)
gdp_sum  <- round(tapply(gapminder$gdpPercap,gapminder$continent,sum),2)
tb1 <- cbind(pop_sum,gdp_sum)
cls <- rownames(tb1)
```

```
rownames(tb1) <- NULL
tb2 <- cbind(cls,tb1)
kable(tb2,digits = 2,
      col.names = c("continent", "pop", "gdpPercap"),
      caption = "Population per continent and GDP per capitaby in each continent", align = 'c') %>%
  kable_styling(latex_options = 'hold_position',font_size = 12,full_width = F,position = "center")%>%
  column_spec(1,bold = T)
```

Table 1: Population per continent and GDP per capitaby in each continent

continent	pop	gdpPercap
Africa	6187585961	1368902.86
Americas	7351438499	2140833.11
Asia	30507333901	3129251.57
Europe	6181115304	5209011.19
Oceania	212992136	446918.62

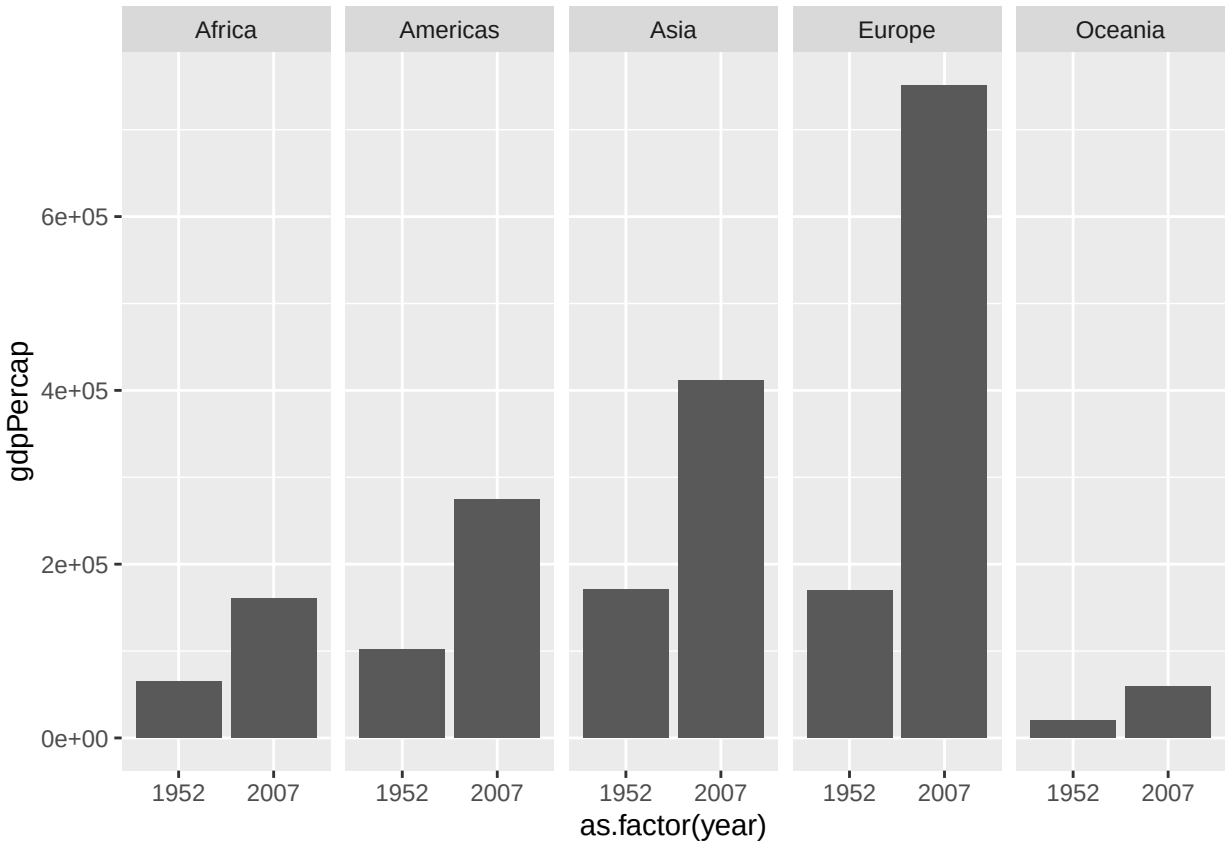
Produce a well-labeled table that summarizes GDP per capita for the countries in each continent, contrasting the years 1952 and 2007.

```
gapminder %>% filter(year %in% c(1952, 2007)) %>%
  group_by(continent,year) %>%
  summarise(GDP = sum(gdpPercap))
```

```
## # A tibble: 10 x 3
## # Groups:   continent [5]
##    continent  year     GDP
##    <fct>     <int>   <dbl>
##  1 Africa     1952  65134.
##  2 Africa     2007 160630.
##  3 Americas   1952 101977.
##  4 Americas   2007 275076.
##  5 Asia       1952 171451.
##  6 Asia       2007 411610.
##  7 Europe     1952 169832.
##  8 Europe     2007 751634.
##  9 Oceania    1952  20596.
## 10 Oceania    2007  59620.
```

Product a plot that summarizes the same data as the table. There should be two plots per continent.

```
gapminder %>% filter(year %in% c(1952, 2007)) %>%
ggplot() +
  geom_bar(mapping=aes(x=as.factor(year), y=gdpPercap),stat="identity")+
  facet_grid(.~continent)
```



Which countries in the dataset have had periods of negative population growth? Illustrate your answer with a table or plot.

```
ng <- gapminder %>% select(country,year,pop) %>%
  group_by(country) %>%
  mutate(growth = pop - lag(pop, order_by = year)) %>%
  filter(growth < 0)
unique(ng$country)
```

```
##  [1] Afghanistan            Bosnia and Herzegovina Bulgaria
##  [4] Cambodia               Croatia                Czech Republic
##  [7] Equatorial Guinea      Germany                Guinea-Bissau
## [10] Hungary                Ireland                Kuwait
## [13] Lebanon                Lesotho                Liberia
## [16] Montenegro             Poland                 Portugal
## [19] Romania                Rwanda                 Serbia
## [22] Slovenia               Somalia                South Africa
## [25] Switzerland            Trinidad and Tobago    West Bank and Gaza
## 142 Levels: Afghanistan Albania Algeria Angola Argentina ... Zimbabwe
```

Which countries in the dataset have had the highest rate of growth in per capita GDP? Illustrate your answer with a table or plot.

```
gapminder %>% select (country,year,pop) %>%
  group_by(country)  %>%
  mutate(growth = pop - lag(pop, order_by = year)) %>%
  arrange(desc(growth))
```

```
## # A tibble: 1,704 x 4
```

```
## # Groups:   country [142]
##    country  year        pop    growth
##    <fct>   <int>      <int>     <int>
##  1 China    1972  862030000 107480000
##  2 China    1967  754550000  88780000
##  3 India    1997  959000000  87000000
##  4 India    1992  872000000  84000000
##  5 China    1987 1084035000  83754000
##  6 China    1977  943455000  81425000
##  7 China    1957  637408000  81144473
##  8 China    1992 1164970000  80935000
##  9 India    1987  788000000  80000000
## 10 India    2007 1110396331  76223784
## # ... with 1,694 more rows
```

Problem 2

The data for Problem 2 is the Fertility data in the AER package. This data is from the 1980 US Census and is comprised of date on married women aged 21-35 with two or more children. The data report the gender of each woman's first and second child, the woman's race, age, number of weeks worked in 1979, and whether the woman had more than two children.

```
library(AER)
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

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

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

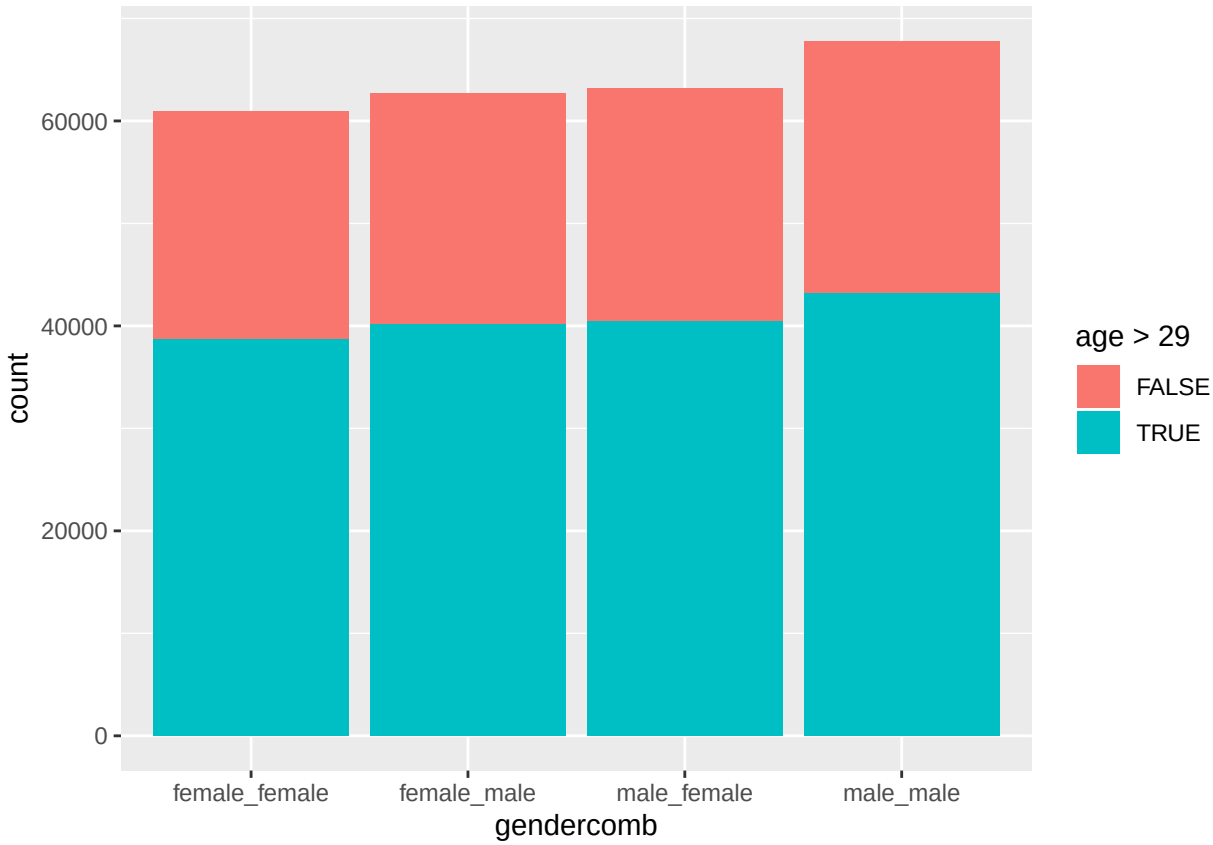
```
## Loading required package: sandwich
```

```
## Loading required package: survival
```

```
data(Fertility)
```

There are four possible gender combinations for the first two Children. Product a plot the contracts the frequency of these four combinations. Are the frequencies different for women in their 20s and wemen who are older than 29?

```
data1 <- Fertility %>% unite("gendercomb",gender1, gender2)
ggplot(data=data1, aes(x=gendercomb, fill=age>29)) +
  geom_bar()
```



Produce a plot that contrasts the frequency of having more than two children by race and ethnicity.

```
Fertility %>% filter(morekids == "yes") %>% count(afam = "yes")
```

```
## # A tibble: 1 x 2
##   afam      n
##   <chr> <int>
## 1 yes   96912
```

Problem 3

Use the mtcars and mpg datasets.

```
library(knitr)
library(ggplot2)
data(mtcars)
data(mpg)
```

How many times does the letter "e" occur in mtcars rownames?

```
cardata <- as_tibble(rownames_to_column(mtcars, var = "Model"))
cardata$number.of.e <- str_count(cardata$Model, "e")
sum(cardata$number.of.e)
```

```
## [1] 25
```

How many cars in mtcars have the brand Merc?

```
sum(str_count(cardata$Model,"Merc"))
```

```
## [1] 7
```

How many cars in mpg have the brand("manufacturer" in mpg) Merc?

```
sum(str_count(mpg$manufacturer,"mercury"))
```

```
## [1] 4
```

Contrast the mileage data for Merc cars as reported in mtcars and mpg. Use tables, plots, and a short explaination.

Problem 4

Install the babynames package. Draw a sample of 500,000 rows from the babynames data

```
library(babynames)
data(babynames)
bn <- sample_n(babynames, 500000)
```

Produce a table that displays the five most popular boy names and girl names in the years 1880,1920, 1960, 2000.

```
f1880 <- bn %>% select (year,sex,name,n) %>%
  group_by(year,sex,name) %>%
  filter(year == 1880) %>%
  filter(sex == "F") %>%
  arrange(desc(n)) %>%
  head(n = 5)
m1880 <- bn %>% select (year,sex,name,n) %>%
  group_by(year,sex,name) %>%
  filter(year == 1880) %>%
  filter(sex == "M") %>%
  arrange(desc(n)) %>%
  head(n = 5)
f1920 <- bn %>% select (year,sex,name,n) %>%
  group_by(year,sex,name) %>%
  filter(year == 1920) %>%
  filter(sex == "F") %>%
  arrange(desc(n)) %>%
  head(n = 5)
m1920 <- bn %>% select (year,sex,name,n) %>%
  group_by(year,sex,name) %>%
  filter(year == 1920) %>%
  filter(sex == "M") %>%
  arrange(desc(n)) %>%
  head(n = 5)
f1960 <- bn %>% select (year,sex,name,n) %>%
  group_by(year,sex,name) %>%
  filter(year == 1960) %>%
  filter(sex == "F") %>%
  arrange(desc(n)) %>%
  head(n = 5)
m1960 <- bn %>% select (year,sex,name,n) %>%
  group_by(year,sex,name) %>%
  filter(year == 1960) %>%
  filter(sex == "M") %>%
  arrange(desc(n)) %>%
  head(n = 5)
f2000 <- bn %>% select (year,sex,name,n) %>%
```

```
  group_by(year,sex,name) %>%
  filter(year == 2000) %>%
  filter(sex == "F") %>%
  arrange(desc(n)) %>%
  head(n = 5)
m2000 <- bn %>% select (year,sex,name,n) %>%
  group_by(year,sex,name) %>%
  filter(year == 2000) %>%
  filter(sex == "F") %>%
  arrange(desc(n)) %>%
  head(n = 5)
hot5 <- rbind(f1880, m1880, f1920, m1920, f1960, m1960, f2000, m2000)
hot5
```

```
## # A tibble: 40 x 4
## # Groups:   year, sex, name [35]
##     year sex   name          n
##    <dbl> <chr> <chr>     <int>
##  1  1880 F     Elizabeth  1939
##  2  1880 F     Ida        1472
##  3  1880 F     Annie      1258
##  4  1880 F     Clara      1226
##  5  1880 F     Grace       982
##  6  1880 M     George     5126
##  7  1880 M     Frank      3242
##  8  1880 M     Henry      2444
##  9  1880 M     Charlie     730
## 10  1880 M     Richard     728
## # ... with 30 more rows
```

What names overlap boys and girls?

```
boysn <- bn %>% filter(sex == "M")
girlsn <- bn %>% filter(sex == "F")
overlap <- intersect(boysn$name,girlsn$name)
```

What names were used in the 19th century but have not been used in the 21sth century?

```
used19 <- filter(bn, year >= 1880 & year <= 1899)
used20 <- filter(bn, year >= 2000 & year <= 2017)
only19 <- !(used20$name %in% used19$name)
```

Produce a chart that shows the relative frequency of the names “Donald”, “Hilary”, “Hillary”, “Joe”, “Barrack”, over the years 1880 through 2017.

```
from80to17 <- filter(bn, year >= 1880 & year <= 2017)
n <- length(bn$name)
from80to17 <- filter(bn, name %in% c("Donald", "Hilary", "Hillary", "Joe", "Barrack"))
rela <- from80to17 %>% group_by(name) %>% summarise(sum(n)/length(bn$name))
rela
```

```
## # A tibble: 4 x 2
##   name    `sum(n)/length(bn$name)`
##   <chr>                      <dbl>
## 1 Donald                    0.771
## 2 Hilary                    0.0159
## 3 Hillary                   0.0177
```

```
## 4 Joe                    0.237
```