Homework 4

Prepare your answers as a **single PDF file**.
Group work: You may work in groups of 1-3. Include all group member names in the PDF file.
Only **one person** in the group should submit to Canvas.
Due: check on Canvas.

1. Consider the two tables shown below called **population** and **countyseats**.

population:

	state	county	year	population
1	California	Orange	2000	2846289
2	California	Orange	2010	3010232
3	California	Los Angeles	2000	3694820
4	California	Los Angeles	2010	3792621

countyseats:

	statename	countyname	countyseat
1	California	Orange	Santa Ana
2	California	Los Angeles	Los Angeles
3	California	San Diego	San Diego
4	Oregon	Wasco	The Dalles

You should be able to calculate the output by hand though you may use R to check your answer.
Draw the output table from the following operations (you should be able to calculate the output by hand though you may use R to check your answers).

a) `population %>% inner_join(countyseats)`

	state	county	year	population	countyname	countyseat
1	California	Orange	2000	2846289	Orange	Santa Ana
2	California	Orange	2010	3010232	Orange	Santa Ana
3	California	Los Angeles	2000	3694820	Los Angeles	Los Angeles
4	California	Los Angeles	2010	3792621	Los Angeles	Los Angeles

b) `population %>% inner_join(countyseats, by=c(state="statename"))`

	state	county	year	population	countyname	countyseat
1	California	Orange	2000	2846289	Orange	Santa Ana
3	California	Los Angeles	2000	3694820	Los Angeles	Los Angeles

c) population %>% inner_join(countyseats, by=c(state="statename", county="countyname"))

	state	county	year	population	countyname	countyseat
1	California	Orange	2000	2846289	Orange	Santa Ana

d) population %>% inner_join(countyseats, by=c(state="statename", county="countyname", year="countyseat"))

There are no matches

2. Consider the `billboard` dataset that is supplied with the tidyverse which shows the Billboard top 100 song rankings in the year 2000. Apply the tidyverse's data wrangling verbs to answer these questions. For each question, **give only the code**.

a) Show for each track, how many weeks it spent on the chart

```
> question1 <- billboard %>% pivot_longer(cols = starts_with("wk"), names_to = "Week", values_to = "Rank") %>% group_by(track) %>% summarize(WeeksOnChart = sum(!is.na(Rank)))
```

b) List tracks in decreasing order of number of weeks spent on the chart

```
> question1 <- billboard %>% pivot_longer(cols = starts_with("wk"), names_to = "Week", values_to = "Rank") %>% group_by(track) %>% summarize(WeeksOnChart = sum(!is.na(Rank))) %>% arrange(desc(WeeksOnChart))
```

c) Show for each track, its top rank

```
> question2 <- billboard %>% pivot_longer(cols = starts_with("wk"), names_to = "Week", values_to = "Rank") %>% group_by(track) %>% summarize(TopRank = max(Rank, na.rm = TRUE))
```

d) List tracks in increasing order of its top rank

```
> question2 <- billboard %>% pivot_longer(cols = starts_with("wk"), names_to = "Week", values_to = "Rank") %>% group_by(track) %>% summarize(TopRank = max(Rank, na.rm = TRUE)) %>% arrange(TopRank)
```

e) Show for each artist, their top rank

```
> question3 <- billboard %>% pivot_longer(cols = starts_with("wk"), names_to = "Week", values_to = "Rank") %>% group_by(artist) %>% summarize(TopRank = max(Rank, na.rm = TRUE))
```

f) List artists in increasing order of their top rank

```
> question3 <- billboard %>% pivot_longer(cols = starts_with("wk"), names_to = "Week", values_to = "Rank") %>% group_by(artist) %>% summarize(TopRank = max(Rank, na.rm = TRUE))
```

g) List tracks that spent more than 35 weeks in the charts

```
question4 <- billboard %>% pivot_longer(cols = starts_with("wk"), names_to = "Week", values_to = "Rank") %>% group_by(track) %>% summarize(WeeksOnChart = sum(!is.na(Rank))) %>% arrange(desc(WeeksOnChart)) %>% filter(WeeksOnChart > 35)
```

h) List tracks that spent more than 35 weeks in the charts along with their artists

```
> question5 <- billboard %>% pivot_longer(cols = starts_with("wk"), names_to = "Week", values_to = "Rank") %>% group_by(track, artist) %>% summarize(WeeksOnChart = sum(!is.na(Rank))) %>% arrange(desc(WeeksOnChart)) %>% filter(WeeksOnChart > 35)
```

Hint: ***First***, **convert to a tidy table**. Show code first for this step. All the above questions can then be answered with a single data pipeline.

3. The `demographics.csv`[1] file (available in the Datasets module on Canvas) gives the proportion of a country's population in different age groups and some other demographic data such as mortality rates and expected lifetime. You can read a CSV file into a tibble using tidyverse's `read_csv()`, like so: `demo <- read_csv("demographics.csv")`

(a) The data is not "tidy". In 2-3 sentences, explain why.

Tidy data should have one observation per row, one variable per column, and one value per cell. In the provided data, information about different demographic indicators, such as population by age group, sex, and other demographic variables, is spread across multiple rows per country.

(b) Transform the table to tidy data with one country per row. [Give code]

```
> demo_tidy <- demo %>% select(-`Series Name`)
> demo_tidy <- demo_tidy %>% pivot_wider(names_from = `Series Code`, values_from = YR2015)
```

(c) Add the male/female population numbers together (i.e., ignore sex-related differences). [Hint: You will have to mutate for every pair of columns, e.g., `mutate(SP.POP.0014.IN=SP.POP.0014.MA.IN+SP.POP.0014.FE.IN`] [Give code]

Note: the Series Name column is included only to show the meaning of the Series Code values. This column can be dropped. At the end, the data should look as below.

Country Name	Country Code	SP.DYN.LE00.IN	SP.URB.TOTL	SP.POP.TOTL	SP.POP.80UP	SP.POP.1564.IN	SP.POP.0014.IN	SP.DYN.AMRT	SP.POP.TOTL.IN	SP.POP.65UP.IN
Afghanistan	AFG	63.37700	8535606	34413603	85552	18116800	15443807	455.4700	34413603	852996
Albania	ALB	78.02500	1654503	2880703	66965	1979175	537788	150.4100	2880703	363740
Algeria	DZA	76.09000	28146511	39728025	453741	25993589	11404930	191.6310	39728025	2329506
American Samoa	ASM	NA	48689	55812	NA	NA	NA	NA	NA	NA
Andorra	AND	NA	68919	78011	NA	NA	NA	NA	NA	NA
Angola	AGO	59.39800	17691524	27884381	69363	14113726	13136043	485.9310	27884381	634612

[1] Original dataset: https://databank.worldbank.org/source/population-estimates-and-projections/Type/TABLE/preview/on#

```
> demo_tidy_2 <- demo_tidy %>% mutate(SP.POP.80UP=SP.POP.80UP.FE+SP.POP.80UP.MA,
.after=SP.POP.TOTL) %>% mutate(SP.POP.1564=SP.POP.1564.MA.IN+SP.POP.1564.FE.IN,
.after=SP.POP.80UP) %>% mutate(SP.POP.0014.IN=SP.POP.0014.MA.IN+SP.POP.0014.FE.IN,
.after=SP.POP.1564) %>% mutate(SP.DYNM.AMRT=SP.DYN.AMRT.FE+SP.DYN.AMRT.MA,
.after=SP.POP.0014.IN) %>% mutate(SP.POP.TOTL.IN=SP.POP.TOTL.FE.IN+SP.POP.TOTL.MA.IN,
.after=SP.DYNM.AMRT) %>% mutate(SP.POP.65UP=SP.POP.65UP.FE.IN+SP.POP.65UP.MA.IN,
.after=SP.POP.TOTL.IN)
```

(d) Write code to show the top 5 countries with the lowest proportion of the population below 14 years old (i.e., SP.POP.0014.IN/SP.POP.TOTL) [Code, and list of 5 countries]

```
> demo_tidy_3 <- demo_tidy_2 %>% mutate(Proportion_Below_14 = `SP.POP.0014.IN` /
`SP.POP.TOTL`)
> top_5_lowest <- demo_tidy_3 %>% select(`Country Name`, Proportion_Below_14) %>%
arrange(Proportion_Below_14) %>% head(5)
```

	Country Name	Proportion_Below_14
1	Hong Kong SAR, China	0.1116757
2	Macao SAR, China	0.1261549
3	Singapore	0.1263214
4	Japan	0.1298938
5	Germany	0.1321694