Homework #3

Answer the following questions in a single R script called `hw03.R`. Answers must be given by R commands. You cannot simply look at the data set and answer the question via direct inspection. Use comments (#) to indicate which portion of your code answers which question. Be sure that you obtain the correct solutions to each question when you execute your script one line at a time from top to bottom.

Each question will be graded out of 4 points according to the following criteria:

- 0 points: No attempts is made to answer the question.
- 1 point: An attempt is made that, although unsuccessful, revealed some understanding of what the question was asking.
- 2 points: Solution is incorrect, but with some modifications, could be corrected.
- 3 points: Solution is incorrect, but easily resolved with minor modifications **OR** solution is correct, but obtained via convoluted reasoning or by avoiding standard approaches.
- 4 points: Solution is correct and uses standard approaches.

For the following problems, you will use the data contained in `ws03_gun_violence.csv`. Included with this assignment is a script called `hw03_start.R`. The code in this script was taken from Worksheet #3. Copy this script to `hw03.R` when starting your homework. You will need to run this portion of the script before starting the exercises below. This script makes important changes needed for this assignment.

#1) Give a bar chart for the number of 'defensive' incidents by year.

#2) Give a stacked bar chart for the number killed by year with sub-bars colored according to whether the weapon was an 'assault' vs. 'non-assault' weapon.

#3) Give a stacked bar chart for the number killed in a mass shootings (four or more killed) by year with sub-bars colored according to whether the weapon was an 'assault' vs 'non-assault' weapon.

#4) Give a dodge bar chart by year for the number of incidents for each incident category we're considering. This means that for each year, there should be a bar chart that counts the number of incidents in the following categories: officer involved, felon, accidental, defensive, assault weapon, terrorism, gang, drugs.

#5) In #4, the plot was a bit deceptive in that it implicitly assumed there was no overlap between these different types of incidents. Count the number of incidents, by year, that fell within 2 or more incident categories.

#6) Give a dodge bar chart by year for the number killed in incidents that were either officer involved, felon, accidental, defensive, assault weapon, terrorism, gang, drugs. Each such incident bar should give the number killed in such incidents. This problem is similar to #4, except of simply counting how many incidents occurred, we count how many were killed in such incidents.

#7) For lack of a better term, let's say that an incident is 'of interest' if it falls into one of the following categories: officer involved, felon, accidental, defensive, assault weapon, terrorism, gang or drugs. Give a dodge bar chart by year for the number killed in 'of interest' incidents versus those that are 'not of interest'.

#8) Give a dodge bar chart for the number of 'defensive' incidents by year along with the total number killed (by year) by mass shootings (4 or more killed).

#9) Earlier, we disaggregated the `incident_characteristics` column. In this exercise, we will do the same for the `participant_status` and `participant_type` columns. The information in these two columns are linked via an indexing system. For instance, in the row with `incident_id = 92383`, we find that the `participant_status` column contains:

```
0::Killed||1::Unharmed
```

and the `participant_type` column contains:

```
0::Victim||1::Subject-Suspect
```

This means that person 0 was a killed victim and person 1 is an unharmed subject-suspect. For this exercise, create a tibble that has one and only entry under `participant_status` and `participant_type`. This will require duplicating existing rows. For instance the row with `incident_id = 92383` will be duplicated with one row containing:

```
participant_status:  Killed
participant_type:  Victim
```

and another row containing:

```
participant_status:  Unharmed
participant_type:  Subject-Suspect
```

To perform this type of disaggregation, I used the function `separate_rows`, `substr`, `strsplit`, (from the `stringr` library) and `rowwise` (from `dplyr`).