MM916 Additional tidyverse questions

The following questions are based on data from the RStats community activity TidyTuesday. A new data set is released each week and the goal is to tease out interesting facts about the data. Some pretty spectacular data visualisations have been part of this…search for #tidytuesday on twitter/X.

Note that the instructions here are deliberately vague. You'll have to work out how to find out each piece of information: this is good practice for doing real-life analysis!

Question 1

a) Load the data stored in `volcano.RData`.

```
# use the load() function to read in RData files
load("volcano.RData")
```

b) Which country has had the most recent volcanic eruption?

```
# There are a couple of options to do this

# Option 1: Use volcanos - need to extract "unknown"
# Can get country directly
volcano %>%
  filter(last_eruption_year!="Unknown") %>%
  mutate(last_eruption_year=as.numeric(last_eruption_year)) %>%
  arrange(desc(last_eruption_year))
```

```
## # A tibble: 657 x 26
##    volcano_number volcano_name   primary_volcano_type last_eruption_year country
##             <dbl> <chr>          <chr>                             <dbl> <chr>
##  1         282080 Aira           Caldera                            2020 Japan
##  2         282110 Asosan         Caldera                            2020 Japan
##  3         255020 Bagana         Lava cone                          2020 Papua ~
##  4         300250 Bezymianny     Stratovolcano                      2020 Russia
##  5         357070 Chillan, Neva~ Stratovolcano                      2020 Chile
##  6         268010 Dukono         Complex                            2020 Indone~
##  7         290380 Ebeko          Stratovolcano                      2020 Russia
##  8         390020 Erebus         Stratovolcano                      2020 Antarc~
##  9         221080 Erta Ale       Shield                             2020 Ethiop~
## 10         211060 Etna           Stratovolcano(es)                  2020 Italy
## # i 647 more rows
## # i 21 more variables: region <chr>, subregion <chr>, latitude <dbl>,
## #   longitude <dbl>, elevation <dbl>, tectonic_settings <chr>,
## #   evidence_category <chr>, major_rock_1 <chr>, major_rock_2 <chr>,
## #   major_rock_3 <chr>, major_rock_4 <chr>, major_rock_5 <chr>,
## #   minor_rock_1 <chr>, minor_rock_2 <chr>, minor_rock_3 <chr>,
## #   minor_rock_4 <chr>, minor_rock_5 <chr>, population_within_5_km <dbl>, ...
```

```
# Option 2: (Better - more detail) Use eruptions and order by the three start_ columns
# Turns out it already is!
```

```r
# need to join with volcanos to get country
eruptions %>%
  arrange(desc(start_year), desc(start_month), desc(start_day)) %>%
  right_join(volcano) %>%
  # can select columns to make lookup easier include date for sanity check
  select(volcano_number, volcano_name, country, start_year:start_day)
```

```
## # A tibble: 9,828 x 6
##    volcano_number volcano_name          country start_year start_month start_day
##             <dbl> <chr>                 <chr>        <dbl>       <dbl>     <dbl>
##  1         266030 Soputan               Indone~       2020           3        23
##  2         233020 Fournaise, Piton de ~ France        2020           2        10
##  3         345020 Rincon de la Vieja    Costa ~       2020           1        31
##  4         273070 Taal                  Philip~       2020           1        12
##  5         282050 Kuchinoerabujima      Japan         2020           1        11
##  6         241040 Whakaari/White Island New Ze~       2019          12         9
##  7         311060 Semisopochnoi         United~       2019          12         7
##  8         282060 Kikai                 Japan         2019          11         2
##  9         300260 Klyuchevskoy          Russia        2019          10        24
## 10         283110 Asamayama             Japan         2019           8         7
## # i 9,818 more rows
```

```r
# Option 3: (Even better - which has been erupting most recently) Use eruptions
# and order by end_ columns
# need to join with volcanos to get country
eruptions %>%
  arrange(desc(end_year), desc(end_month), desc(end_day)) %>%
  right_join(volcano) %>%
  # can select columns to make lookup easier include date for sanity check
  select(volcano_number, volcano_name, country, end_year:end_day)
```

```
## # A tibble: 9,828 x 6
##    volcano_number volcano_name       country    end_year end_month end_day
##             <dbl> <chr>              <chr>         <dbl>     <dbl>   <dbl>
##  1         390020 Erebus             Antarctica     2020         4      18
##  2         345020 Rincon de la Vieja Costa Rica     2020         4      17
##  3         282050 Kuchinoerabujima   Japan          2020         4      17
##  4         300260 Klyuchevskoy       Russia         2020         4      17
##  5         282110 Asosan             Japan          2020         4      17
##  6         352090 Sangay             Ecuador        2020         4      17
##  7         262000 Krakatau           Indonesia      2020         4      17
##  8         263250 Merapi             Indonesia      2020         4      17
##  9         261170 Kerinci            Indonesia      2020         4      17
## 10         282080 Aira               Japan          2020         4      17
## # i 9,818 more rows
```

c) How many eruptions are recorded as having a severity index greater than 4?

```r
# You will need to filter out the missing values
eruptions %>% filter(!is.na(vei)) %>%
  summarise(severe_eruptions=sum(vei>4))
```

```
## # A tibble: 1 x 1
##   severe_eruptions
##              <int>
## 1              237
```

```r
#Note that another way to do the same thing is
eruptions %>% summarise(severe_eruptions=sum(vei>4,na.rm=TRUE))
```

```
## # A tibble: 1 x 1
##   severe_eruptions
##              <int>
## 1              237
```

d) Which type of volcano has produced the most severe eruption in the data set?

```r
# If you interpret this literally, as the most severe single eruption, then:
eruptions %>% filter(vei==max(vei,na.rm=TRUE)) %>%
# this gives several different eruptions which are all tied for maximum VEI.
# then look up the type
left_join(volcano) %>% select(primary_volcano_type,vei)
```

```
## # A tibble: 8 x 2
##   primary_volcano_type   vei
##   <chr>                <dbl>
## 1 Stratovolcano            7
## 2 Stratovolcano            7
## 3 Stratovolcano            7
## 4 Shield(s)                7
## 5 Caldera                  7
## 6 Caldera                  7
## 7 Caldera                  7
## 8 Caldera                  7
```

```r
# Alternatively, if you interpret it as the volcano type with the most severe
# eruptions *on average*:
# associate type with each individual eruption
eruptions %>% left_join(volcano) %>%
# find the mean severity for each type (as in 3c, have to eliminate NA values)
group_by(primary_volcano_type) %>% summarise(mean_vei = mean(vei,na.rm=TRUE)) %>%
# finally, sort by mean VEI so that the most severe pop to the top of the data frame
arrange(desc(mean_vei))
```

```
## # A tibble: 24 x 2
##    primary_volcano_type mean_vei
##    <chr>                   <dbl>
##  1 Caldera(s)               3.38
##  2 Subglacial               2.82
##  3 Maar(s)                  2.67
##  4 Pyroclastic cone         2.6
##  5 Compound                 2.57
##  6 Pyroclastic shield       2.46
##  7 Shield(s)                2.42
##  8 Stratovolcano?           2.38
##  9 Lava dome(s)             2.35
## 10 Fissure vent(s)          2.27
## # i 14 more rows
```

e) On average, does there appear to be an effect of major rock type 1 on the severity of an eruption?

```r
# Need to join the two data sets
# use left join since eruptions is bigger
eruptions %>%
```

```r
  left_join(volcano) %>%
  # Want to find the average severity based on major rock type 1
  group_by(major_rock_1) %>%
  # I've used median as the average since it is more robust
  summarise(median(vei, na.rm=TRUE))
```

```
## # A tibble: 11 x 2
##    major_rock_1                             `median(vei, na.rm = TRUE)`
##    <chr>                                                          <dbl>
##  1 Andesite / Basaltic Andesite                                       2
##  2 Basalt / Picro-Basalt                                              2
##  3 Dacite                                                             2
##  4 Foidite                                                            1
##  5 Phono-tephrite /  Tephri-phonolite                                 3
##  6 Phonolite                                                          2
##  7 Rhyolite                                                           2
##  8 Trachyandesite / Basaltic Trachyandesite                           2
##  9 Trachybasalt / Tephrite Basanite                                   2
## 10 Trachyte / Trachydacite                                            3
## 11 <NA>                                                               2
```

```r
# certain rock types do appear to have different average severity indices.
# Not enough information here to say with any certainty but trachyte and
# phono-tephrite appear to have slightly higher severity and foidite has lower.
# Majority seem to have a median severity index of 2.`
```

Question 2

a) Load the data stored in `plants1.RData`

```r
load("plants1.RData")
```

b) Find out how many species are extinct versus extinct in the wild.

```r
plants %>%
  group_by(red_list_category) %>%
  # The function n() takes a count
  summarise(count=n())
```

```
## # A tibble: 2 x 2
##   red_list_category   count
##   <chr>               <int>
## 1 Extinct               435
## 2 Extinct in the Wild    65
```

c) Of the species that are threatened, how many have had action taken in 2 or fewer areas?

```r
# Option 1: Simply add up the columns
plants %>%
  # Get the sum of the action columns
  mutate(ActionsTaken = action_LWP + action_SM + action_LP + action_RM + action_EA + action_NA) %>%
  summarise(sum(ActionsTaken<=2))
```

```
## # A tibble: 1 x 1
##   `sum(ActionsTaken <= 2)`
##                      <int>
```

```
## 1                       485
```

```r
# Option 2: Not covered in class but an additional option is c_across()
plants %>%
  rowwise() %>%
  # Get the sum of the action columns
  mutate(ActionsTaken=sum(c_across(action_LWP:action_NA))) %>%
  ungroup() %>%
  summarise(sum(ActionsTaken<=2))
```

```
## # A tibble: 1 x 1
##   `sum(ActionsTaken <= 2)`
##                      <int>
## 1                      485
```

```r
# In option 2 rowwise() indicates that we want to do operations row-wise rather
# than column-wise - by default sum() would want to take the sum of the columns.
# then c_across allows us to specify column names in the same way as normal.
# rowwise() is a s pecial type of grouping - to go back to standard use use ungroup()`
```

d) Which country contains the most threatened or extinct species of plant?

```r
plants %>%
  group_by(country) %>%
  summarise(Count=n()) %>%
  arrange(desc(Count))
```

```
## # A tibble: 72 x 2
##    country       Count
##    <chr>         <int>
##  1 Madagascar       98
##  2 United States    66
##  3 Ecuador          52
##  4 Tanzania         25
##  5 Malaysia         18
##  6 Burundi          17
##  7 Guinea           14
##  8 Indonesia        12
##  9 New Caledonia    11
## 10 South Africa     11
## # i 62 more rows
```

```r
# Madagascar has the most followed by the US and Ecuador`
```

e) Which plant type is the most threatened?

```r
# if "threatened" means "on this list but not yet fully extinct"
plants %>% filter(red_list_category != "Extinct") %>%
  group_by(group) %>%
  summarise(Count=n()) %>%
  arrange(desc(Count))
```

```
## # A tibble: 3 x 2
##   group            Count
##   <chr>            <int>
## 1 Flowering Plant     60
## 2 Cycad                4
## 3 Ferns and Allies     1
```

```
# flowering plants are by far the most commonly observed in the data.
# The next question, however, is: What is the proportion of known flowering plant
# species relative to all plants? If there are more flowering plants than any other
# kind then this result might be expected...if not then that suggests the flowering
# plants are more easily threatened than other types of plant.`
```

f) During which time period did most plants go extinct? (Exclude plants that went extinct pre-1900 and those with no registered date.)

```
plants %>%
  # exclude data that we don't need
  filter(!year_last_seen%in%c("Before 1900",NA)) %>%
  group_by(year_last_seen) %>%
  summarise(Count=n()) %>%
  arrange(desc(Count))
```

```
## # A tibble: 6 x 2
##   year_last_seen Count
##   <chr>          <int>
## 1 1940-1959         74
## 2 1900-1919         70
## 3 1920-1939         70
## 4 1960-1979         60
## 5 2000-2020         52
## 6 1980-1999         44
```

```
#1940-1959 had the most extinctions. The trend seems to be largely decreasing
# up until the most recent 20 year period.`
```

Question 3

Read in the data stored in `rap rankings.RData`

The algorithm used to rank the rap artists is described at *https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-04-14/readme.md*

Have a go at implementing the algorithm described.

```
load("rap rankings.RData")

# this is a hard one! The sample solution below uses tidyverse functions exclusively,
# but another approach would be to go through the rows with a for loop and use if-else
# statements to keep a tally of rankings and points.

polls %>%
  # Group by all of the variables that appear in the final data set
  group_by(title, artist, gender, year,rank)  %>%
  summarise(n=n()) %>%
  # we're done with the groups so we should ungroup
  ungroup() %>%
  # now have a count of how many times each song got each rank.
  # but at this point, each row = a song getting a certain ranking;
  # next, pivot_wider so that each row = a song.
  # specify values_fill to fill in zeros
  pivot_wider(names_from="rank", values_from="n", names_prefix="n", values_fill=0) %>%
```

```r
  # calculate the number of points and number of ranks given
  mutate(points=10*n1+8*n2+6*n3+4*n4+2*n5,
         n=n1+n2+n3+n4+n5) %>%
  # reorder columns
  select(title:gender, points, n, n1, n2, n3, n4, n5) %>%
  # Order rows: points is most important, followed by total rankings then each ranking
  arrange(desc(points),desc(n),
          desc(n1),desc(n2),
          desc(n3),desc(n4),desc(n5)) %>%
  rowid_to_column(var="ID")
```

```
## # A tibble: 311 x 11
##       ID title          artist gender points     n    n1    n2    n3    n4    n5
##    <int> <chr>          <chr>  <chr>   <dbl> <int> <int> <int> <int> <int> <int>
##  1     1 Juicy          The N~ male      140    18     9     3     3     1     2
##  2     2 Fight The Pow~ Publi~ male      100    11     7     3     1     0     0
##  3     3 Shook Ones (P~ Mobb ~ male       94    13     4     5     1     1     2
##  4     4 The Message    Grand~ male       90    14     5     3     1     0     5
##  5     5 Nuthin' But A~ Dr Dr~ male       84    14     2     4     2     4     2
##  6     6 C.R.E.A.M.     Wu-Ta~ male       62    10     3     1     1     4     1
##  7     7 93 'Til Infin~ Souls~ male       50     7     2     2     2     0     1
##  8     8 Passin' Me By  The P~ male       48     6     3     2     0     0     1
##  9     9 N.Y. State Of~ Nas    male       46     7     1     3     1     1     1
## 10    10 Dear Mama      2Pac   male       42     6     2     1     1     2     0
## # i 301 more rows
```