Tidy Tuesday Himalayan Climbing Expeditions

The DataViz

9/22/2020

Tidy Tuesday Week of Sept 22 2020

Tidy Tuesday Data Background

The data this week comes from The Himalayan Database.

The Himalayan Database is a compilation of records for all expeditions that have climbed in the Nepal Himalaya. The database is based on the expedition archives of Elizabeth Hawley, a longtime journalist based in Kathmandu, and it is supplemented by information gathered from books, alpine journals and correspondence with Himalayan climbers.

The data cover all expeditions from 1905 through Spring 2019 to more than 465 significant peaks in Nepal. Also included are expeditions to both sides of border peaks such as Everest, Cho Oyu, Makalu and Kangchenjunga as well as to some smaller border peaks. Data on expeditions to trekking peaks are included for early attempts, first ascents and major accidents.

h/t to Alex Cookson for sharing and cleaning this data!

This blog post by Alex Cookson explores the data in greater detail.

I don't want to underplay that there are some positives and some awful negatives for native Sherpa climbers. One-third of Everest deaths are Sherpa Climbers.

Also National Geographic has 5 Ways to help the Sherpas of Everest.

Load Libraries

```
library(tidyverse)
# Library for Ridge Plots
library(ggridges)
# Library to easily reorder factors
library(forcats)
# Library to wrap long Strings in ggplot
library(scales)
# Library to plot images in ggplot
library(ggimage)
# Library with good dark ggplot themes
library(ggdark)
# Library that makes file path handling easier
library(here)
# Library to create color maps
library(RColorBrewer)
```

Read in Data

```
members <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   year = col_double(),
##   age = col_double(),
##   hired = col_logical(),
##   highpoint_metres = col_double(),
##   success = col_logical(),
##   solo = col_logical(),
##   oxygen_used = col_logical(),
##   died = col_logical(),
##   death_height_metres = col_double(),
##   injured = col_logical(),
##   injury_height_metres = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
expeditions <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/da
```

```
## Parsed with column specification:
## cols(
##   expedition_id = col_character(),
##   peak_id = col_character(),
##   peak_name = col_character(),
##   year = col_double(),
##   season = col_character(),
##   basecamp_date = col_date(format = ""),
##   highpoint_date = col_date(format = ""),
##   termination_date = col_date(format = ""),
##   termination_reason = col_character(),
##   highpoint_metres = col_double(),
##   members = col_double(),
##   member_deaths = col_double(),
##   hired_staff = col_double(),
##   hired_staff_deaths = col_double(),
##   oxygen_used = col_logical(),
##   trekking_agency = col_character()
## )
```

```
peaks <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/202
```

```
## Parsed with column specification:
## cols(
##   peak_id = col_character(),
##   peak_name = col_character(),
##   peak_alternative_name = col_character(),
##   height_metres = col_double(),
##   climbing_status = col_character(),
##   first_ascent_year = col_double(),
##   first_ascent_country = col_character(),
##   first_ascent_expedition_id = col_character()
```

```
## )
```

Analysis of peaks.csv

```
peaks %>% arrange(first_ascent_year)
```

```
## # A tibble: 468 x 8
##    peak_id peak_name peak_alternativ~ height_metres climbing_status
##    <chr>   <chr>     <chr>                   <dbl> <chr>
##  1 SPH2    Sharphu ~ Tanga II                 6328 Climbed
##  2 LNPO    Langpo    Longpo                   6965 Climbed
##  3 JONG    Jongsang  Jonsong, Jhinsa~         7462 Climbed
##  4 NEPA    Nepal Pe~ <NA>                     7177 Climbed
##  5 RAMT    Ramtang   <NA>                     6601 Climbed
##  6 DOMO    Domo      Jongsang SE Peak         7447 Climbed
##  7 KABN    Kabru No~ <NA>                     7338 Climbed
##  8 LING    Lingtren  <NA>                     6713 Climbed
##  9 SPHN    Sphinx    Pathibhara Phur~         6825 Climbed
## 10 KIRA    Kirat Ch~ Tent Peak                7362 Climbed
## # ... with 458 more rows, and 3 more variables: first_ascent_year <dbl>,
## #   first_ascent_country <chr>, first_ascent_expedition_id <chr>
```

Sharphu II has a typo in the year climbed: the data says 201 but in this article it details 2018. (https://gripped.com/news/first-ascent-of-sharphu-ii-in-nepals-kangchenjunga/)

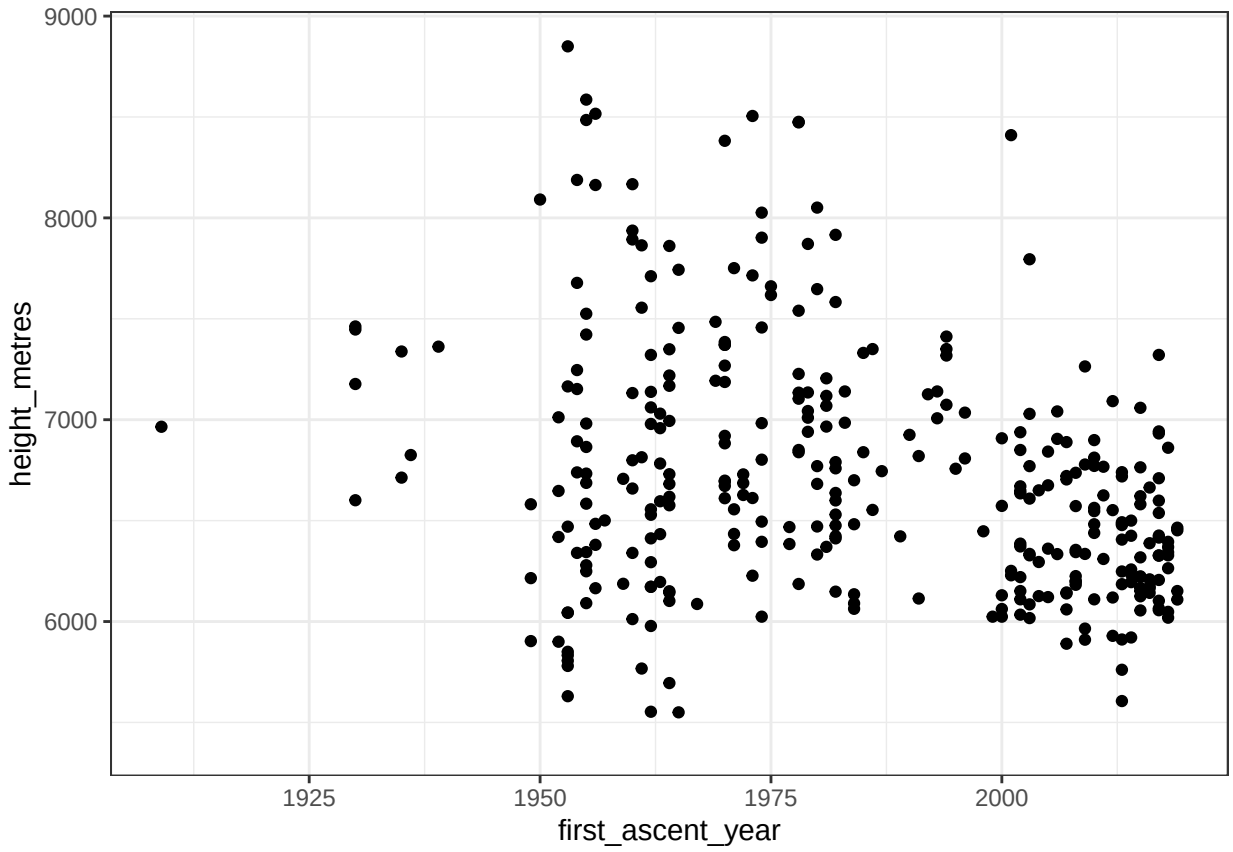
```
peaks[peaks$peak_id=="SPH2","first_ascent_year"] <- 2018
```

```
peaks %>% filter(peak_name=="Everest")
```

```
## # A tibble: 1 x 8
##   peak_id peak_name peak_alternativ~ height_metres climbing_status
##   <chr>   <chr>     <chr>                   <dbl> <chr>
## 1 EVER    Everest   Sagarmatha, Cho~         8850 Climbed
## # ... with 3 more variables: first_ascent_year <dbl>,
## #   first_ascent_country <chr>, first_ascent_expedition_id <chr>
```

```
ggplot(
  data = peaks,
  aes(x = first_ascent_year,y = height_metres)
) + geom_point(color='black') + theme_bw()
```

```
## Warning: Removed 132 rows containing missing values (geom_point).
```



Get a number of peaks left to climb by year

```
count_of_peaks <- nrow(peaks)
peaks_years <- peaks %>% filter(!is.na(first_ascent_year)) %>%
                         group_by(first_ascent_year) %>%
                         summarize(count = n(),
                                   max_height = max(height_metres)
                                   ) %>% ungroup() %>%
                         mutate(cumulative_peaks = cumsum(count),
                                peaks_left = count_of_peaks-cumulative_peaks)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

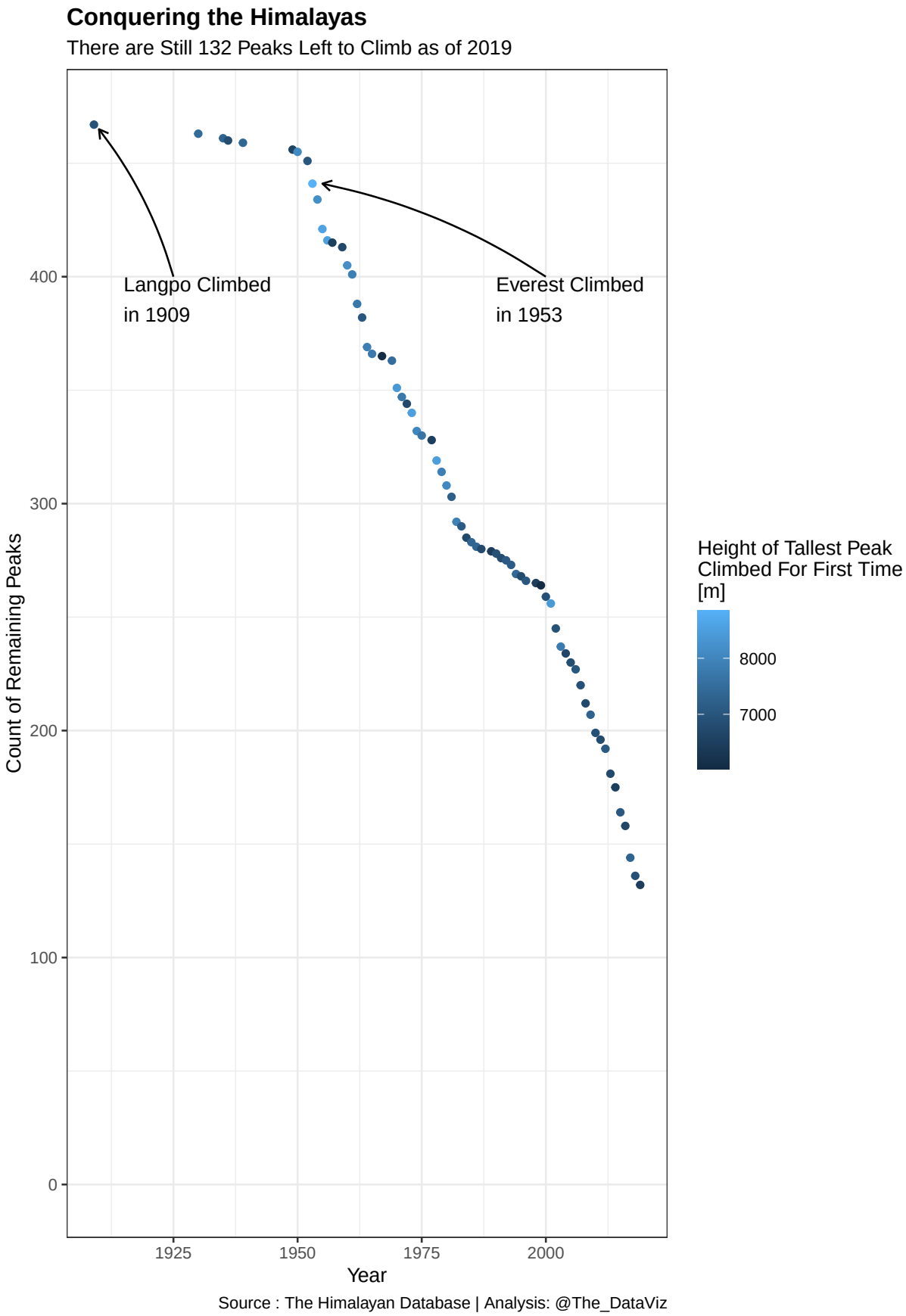
Average height of unclimbed peaks

```
peaks %>% group_by(climbing_status) %>%
                         summarize(mean_height = mean(height_metres,na.rm = T))
```

```
## # A tibble: 2 x 2
##   climbing_status mean_height
##   <chr>                 <dbl>
## 1 Climbed               6706.
## 2 Unclimbed             6523.
```

```
ggplot(
  data = peaks_years,
  aes(x = first_ascent_year,y = peaks_left, color = max_height)
```

```
) + geom_point() +
  theme_bw()+
  theme(plot.title = element_text(face="bold"))+
  # First point Description
  annotate(
    geom = "curve", x = 1925, y = 400, xend = 1910, yend = 465,
    curvature = .1, arrow = arrow(length = unit(2, "mm")),color="black"
  ) +
  annotate(geom = "text", x = 1915, y = 390, size = 4,
           label = "Langpo Climbed\nin 1909",
           hjust = "left",color="black")+
  # Everest Description
  annotate(
    geom = "curve", x = 2000, y = 400, xend = 1955, yend = 441,
    curvature = .1, arrow = arrow(length = unit(2, "mm")),color="black"
  ) +
  annotate(geom = "text", x = 1990, y = 390, size = 4,
           label = "Everest Climbed\nin 1953",
           hjust = "left",color="black")+
  scale_y_continuous(limits = c(0,count_of_peaks)) +
  ylab("Count of Remaining Peaks")+
  xlab("Year")+
  labs(title="Conquering the Himalayas",
       subtitle="There are Still 132 Peaks Left to Climb as of 2019",
       color = "Height of Tallest Peak\nClimbed For First Time\n[m]",
       caption="Source : The Himalayan Database | Analysis: @The_DataViz"
       )
```


Conquering the Himalayas
There are Still 132 Peaks Left to Climb as of 2019
Langpo Climbed
in 1909
Everest Climbed
in 1953
Count of Remaining Peaks
400
300
200
100
0
1925
1950
1975
2000
Year
Height of Tallest Peak
Climbed For First Time
[m]
8000
7000
Source : The Himalayan Database | Analysis: @The_DataViz

```
ggsave("Peaks_Climbed.png")
```

Analysis of expeditions.csv

```
expeditions %>% arrange(year)
```

```
## # A tibble: 10,364 x 16
##    expedition_id peak_id peak_name  year season basecamp_date highpoint_date
##    <chr>         <chr>   <chr>     <dbl> <chr>  <date>        <date>
##  1 KANG05201     KANG    Kangchen~  1905 Summer NA            1905-09-01
##  2 KABN07301     KABN    Kabru No~  1907 Autumn NA            NA
##  3 JONG09301     JONG    Jongsang   1909 Autumn NA            NA
##  4 LNPO09301     LNPO    Langpo     1909 Autumn NA            1909-09-14
##  5 KANG10101     KANG    Kangchen~  1910 Spring NA            NA
##  6 KIRA10101     KIRA    Kirat Ch~  1910 Spring NA            NA
##  7 LNPO10101     LNPO    Langpo     1910 Spring NA            NA
##  8 KABN20401     KABN    Kabru No~  1920 Winter NA            NA
##  9 KANG20301     KANG    Kangchen~  1920 Autumn NA            NA
## 10 TLNG20301     TLNG    Talung     1920 Autumn NA            NA
## # ... with 10,354 more rows, and 9 more variables: termination_date <date>,
## #   termination_reason <chr>, highpoint_metres <dbl>, members <dbl>,
## #   member_deaths <dbl>, hired_staff <dbl>, hired_staff_deaths <dbl>,
## #   oxygen_used <lgl>, trekking_agency <chr>
```

Visual of Success Rate of Expeditions per year

```
expedition_success_by_year <- expeditions %>%
                mutate(success = ifelse(grepl("Success",termination_reason),
                                        "Succeeded","Failed")) %>%
                group_by(year, success) %>%
                summarize(count = n()) %>% ungroup() %>%
                spread(success,count) %>% replace(is.na(.), 0) %>%
                mutate(total_expeditions = Failed+Succeeded,
                       success_rate = Succeeded/total_expeditions
                       )
```

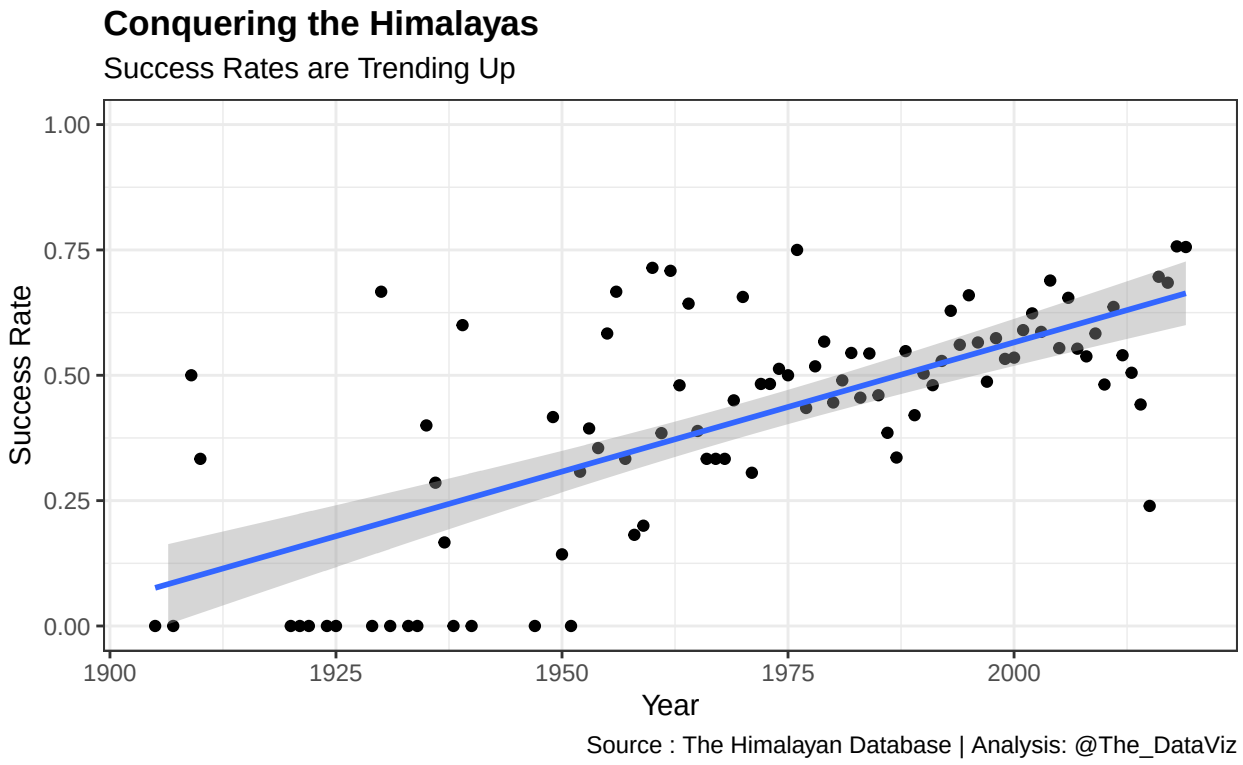
```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
```

```
expedition_success_by_year
```

```
## # A tibble: 92 x 5
##     year Failed Succeeded total_expeditions success_rate
##    <dbl>  <int>     <int>             <int>        <dbl>
##  1  1905      1         0                 1        0
##  2  1907      1         0                 1        0
##  3  1909      1         1                 2        0.5
##  4  1910      2         1                 3        0.333
##  5  1920      3         0                 3        0
##  6  1921      1         0                 1        0
##  7  1922      1         0                 1        0
##  8  1924      1         0                 1        0
##  9  1925      2         0                 2        0
```

```
## 10  1929      2         0                 2        0
## # ... with 82 more rows
```

```
ggplot(
  data = expedition_success_by_year,
  aes(x=year,y=success_rate)
) + geom_point(color='black') +
  geom_smooth(method = "lm",formula='y ~ x') +
  ylim(c(0,1)) +
  theme_bw()+
  theme(plot.title = element_text(face="bold"))+
  ylab("Success Rate")+
  xlab("Year")+
  labs(title="Conquering the Himalayas",
       subtitle="Success Rates are Trending Up",
       color = "Height of Tallest Peak\nClimbed For First Time\n[m]",
       caption="Source : The Himalayan Database | Analysis: @The_DataViz"
       )
```




```
ggsave("Success_Rates.png")
```

Visualization of success by season

```
expedition_success_by_season <- expeditions %>%
                mutate(success = ifelse(grepl("Success",termination_reason),
                                        "Succeeded","Failed")) %>%
                group_by(season, success) %>%
                summarize(count = n()) %>% ungroup() %>%
                spread(success,count) %>% replace(is.na(.), 0) %>%
```

```
                mutate(total_expeditions = Failed+Succeeded,
                       success_rate = Succeeded/total_expeditions
                       )
```

```
## `summarise()` regrouping output by 'season' (override with `.groups` argument)
```

```
expedition_success_by_season
```

```
## # A tibble: 5 x 5
##   season  Failed Succeeded total_expeditions success_rate
##   <chr>    <int>     <int>             <int>        <dbl>
## 1 Autumn    2298      2766              5064        0.546
## 2 Spring    2099      2776              4875        0.569
## 3 Summer      67        41               108        0.380
## 4 Unknown      1         1                 2        0.5
## 5 Winter     172       143               315        0.454
```

Visualization of success by group size

There are some weird groups of size 0. For my analysis on group size I dont want them.

```
expeditions %>% filter(members == 0) %>% arrange(members)
```

```
## # A tibble: 48 x 16
##    expedition_id peak_id peak_name  year season basecamp_date highpoint_date
##    <chr>         <chr>   <chr>     <dbl> <chr>  <date>        <date>
##  1 HIME80302     HIME    Himalchu~  1980 Autumn NA            NA
##  2 ANN186305     ANN1    Annapurn~  1986 Autumn NA            NA
##  3 ANNS71301     ANNS    Annapurn~  1971 Autumn NA            1971-10-15
##  4 CHEO87301     CHEO    Cheo Him~  1987 Autumn NA            NA
##  5 DHA149001     DHA1    Dhaulagi~  1949 Unkno~ NA            NA
##  6 DHA262301     DHA2    Dhaulagi~  1962 Autumn NA            NA
##  7 DHA471101     DHA4    Dhaulagi~  1971 Spring NA            1971-05-22
##  8 DHA571301     DHA5    Dhaulagi~  1971 Autumn NA            1971-10-05
##  9 EVER68101     EVER    Everest    1968 Spring NA            NA
## 10 EVER74102     EVER    Everest    1974 Spring NA            NA
## # ... with 38 more rows, and 9 more variables: termination_date <date>,
## #   termination_reason <chr>, highpoint_metres <dbl>, members <dbl>,
## #   member_deaths <dbl>, hired_staff <dbl>, hired_staff_deaths <dbl>,
## #   oxygen_used <lgl>, trekking_agency <chr>
```

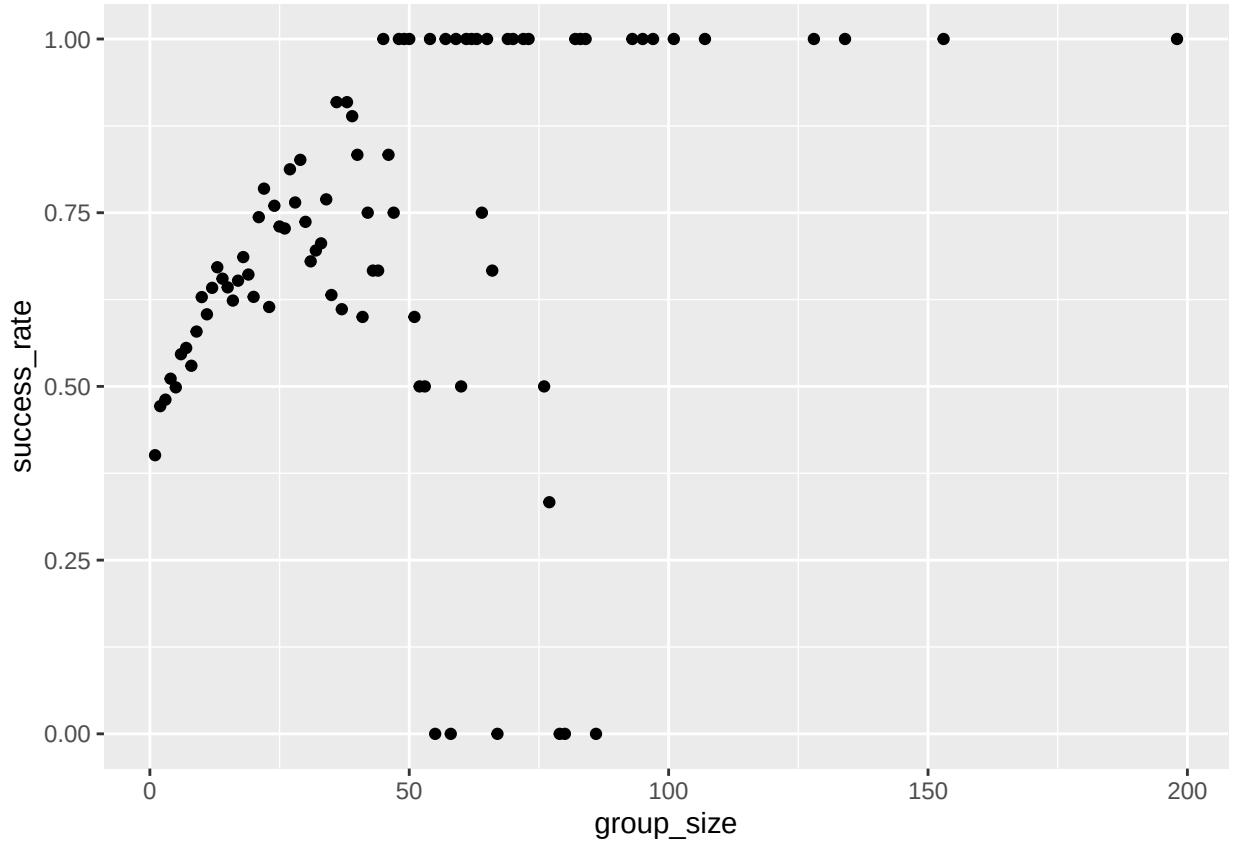
```
expedition_success_by_expedition_size <- expeditions %>%
                mutate(group_size = members + hired_staff,
                       success = ifelse(grepl("Success",termination_reason),
                                        "Succeeded","Failed")) %>%
                filter(group_size > 0) %>%
                group_by(group_size, success) %>%
                summarize(count = n()) %>% ungroup() %>%
                spread(success,count) %>% replace(is.na(.), 0) %>%
                mutate(total_expeditions = Failed+Succeeded,
                       success_rate = Succeeded/total_expeditions
                       )
```

```
## `summarise()` regrouping output by 'group_size' (override with `.groups` argument)
```

```
expedition_success_by_expedition_size
```

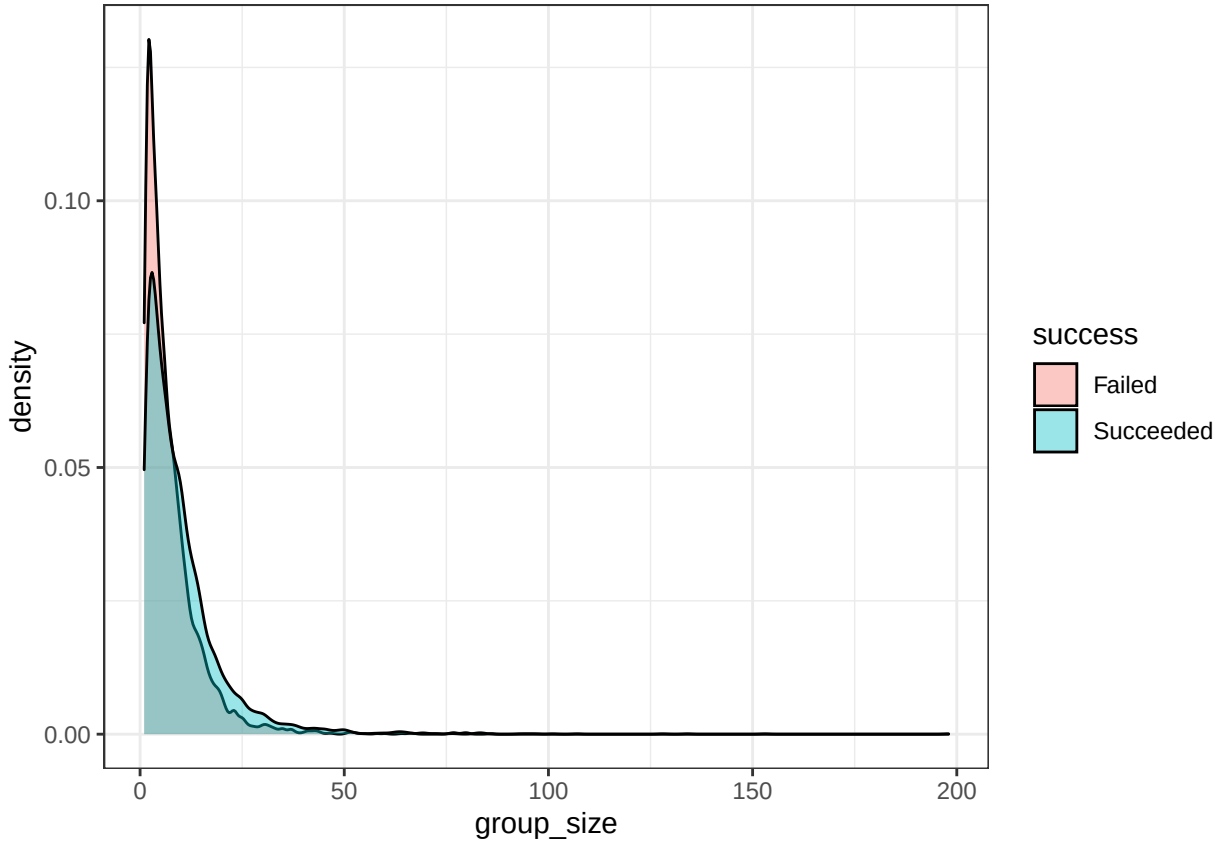
```
## # A tibble: 87 x 5
##    group_size Failed Succeeded total_expeditions success_rate
##         <dbl>  <int>     <int>             <int>        <dbl>
##  1          1    354       237               591        0.401
##  2          2    744       664              1408        0.472
##  3          3    517       479               996        0.481
##  4          4    463       484               947        0.511
##  5          5    370       368               738        0.499
##  6          6    323       389               712        0.546
##  7          7    265       331               596        0.555
##  8          8    245       276               521        0.530
##  9          9    216       297               513        0.579
## 10         10    169       286               455        0.629
## # ... with 77 more rows
```

```
ggplot(
  data = expedition_success_by_expedition_size,
  aes(x=group_size,y=success_rate)
) + geom_point(color='black')
```




```
group_size_successes <- expeditions %>%
                mutate(group_size = members + hired_staff,
                       success = ifelse(grepl("Success",termination_reason),
                                        "Succeeded","Failed")) %>%
                filter(group_size > 0)
```

```
ggplot(
  data = group_size_successes,
  aes(x = group_size,fill = success)
) + geom_density(alpha=0.4) +
  theme_bw()
```

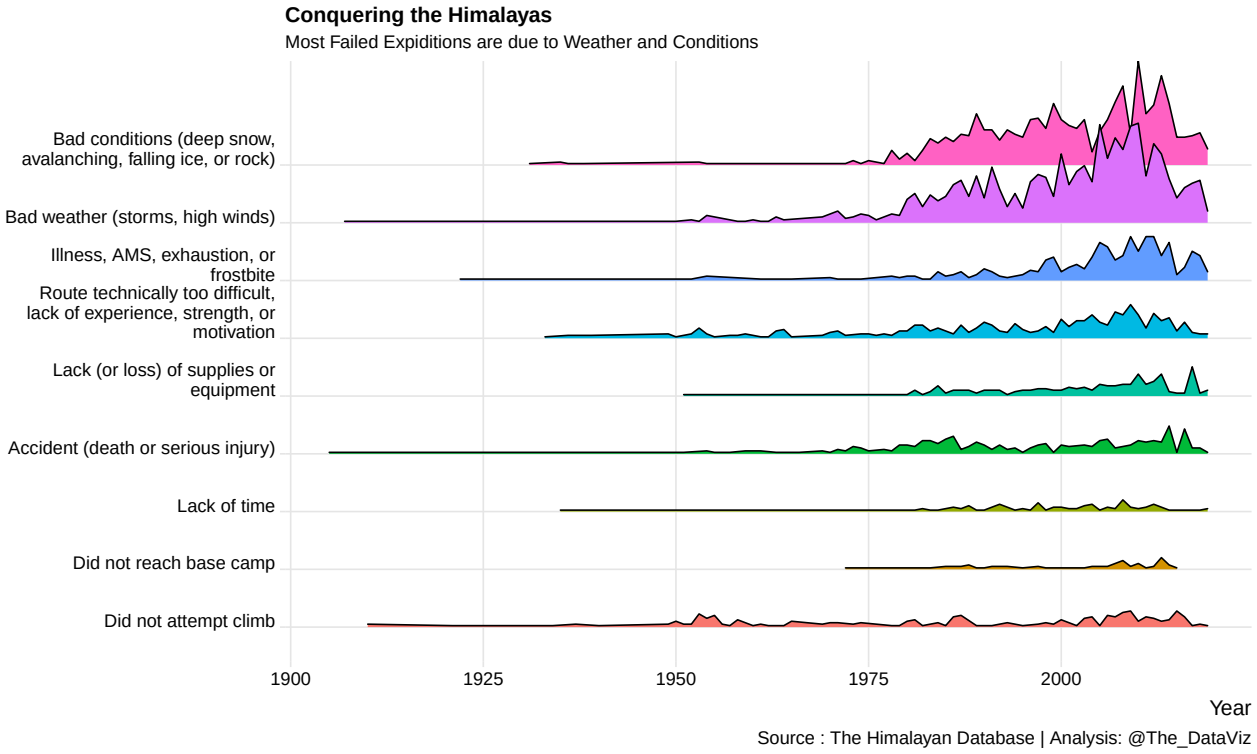



```
expedition_terminations_by_year <- expeditions %>%
    filter(!grepl("Success",termination_reason),
           !grepl("Attempt rumoured",termination_reason),
           !grepl("Unknown",termination_reason),
           !grepl("Other",termination_reason)
           ) %>%
    group_by(year,termination_reason) %>%
    count() %>% ungroup() %>%
    mutate(termination_reason = fct_reorder(termination_reason, n))
expedition_terminations_by_year
```

```
## # A tibble: 495 x 3
##     year termination_reason                          n
##    <dbl> <fct>                                   <int>
##  1  1905 Accident (death or serious injury)          1
##  2  1907 Bad weather (storms, high winds)            1
##  3  1910 Did not attempt climb                       2
##  4  1921 Did not attempt climb                       1
##  5  1922 Illness, AMS, exhaustion, or frostbite      1
##  6  1924 Accident (death or serious injury)          1
##  7  1925 Did not attempt climb                       1
```

```
##  8  1929 Accident (death or serious injury)         1
##  9  1929 Bad weather (storms, high winds)           1
## 10  1930 Bad weather (storms, high winds)           1
## # ... with 485 more rows
```

```
ggplot(
  data = expedition_terminations_by_year,
  aes(x=year,y=termination_reason,height=n,fill=termination_reason)
) + geom_density_ridges(stat="identity") +
  theme_ridges() +
  theme(legend.position = "none") +
  scale_y_discrete(labels = wrap_format(35))+
  ylab("")+
  xlab("Year")+
  labs(title="Conquering the Himalayas",
       subtitle="Most Failed Expiditions are due to Weather and Conditions",
       caption="Source : The Himalayan Database | Analysis: @The_DataViz"
       )
```




```
ggsave("Failed_expeditions.png")
```

Analysis of members.csv

```
members
```

```
## # A tibble: 76,519 x 21
##    expedition_id member_id peak_id peak_name  year season sex     age
##    <chr>         <chr>     <chr>   <chr>     <dbl> <chr>  <chr> <dbl>
##  1 AMAD78301     AMAD7830~ AMAD    Ama Dabl~  1978 Autumn M        40
##  2 AMAD78301     AMAD7830~ AMAD    Ama Dabl~  1978 Autumn M        41
```

```
##  3 AMAD78301     AMAD7830~ AMAD    Ama Dabl~  1978 Autumn M        27
##  4 AMAD78301     AMAD7830~ AMAD    Ama Dabl~  1978 Autumn M        40
##  5 AMAD78301     AMAD7830~ AMAD    Ama Dabl~  1978 Autumn M        34
##  6 AMAD78301     AMAD7830~ AMAD    Ama Dabl~  1978 Autumn M        25
##  7 AMAD78301     AMAD7830~ AMAD    Ama Dabl~  1978 Autumn M        41
##  8 AMAD78301     AMAD7830~ AMAD    Ama Dabl~  1978 Autumn M        29
##  9 AMAD79101     AMAD7910~ AMAD    Ama Dabl~  1979 Spring M        35
## 10 AMAD79101     AMAD7910~ AMAD    Ama Dabl~  1979 Spring M        37
## # ... with 76,509 more rows, and 13 more variables: citizenship <chr>,
## #   expedition_role <chr>, hired <lgl>, highpoint_metres <dbl>, success <lgl>,
## #   solo <lgl>, oxygen_used <lgl>, died <lgl>, death_cause <chr>,
## #   death_height_metres <dbl>, injured <lgl>, injury_type <chr>,
## #   injury_height_metres <dbl>
```

Peak Popularity over time for years with at least 25 climbers

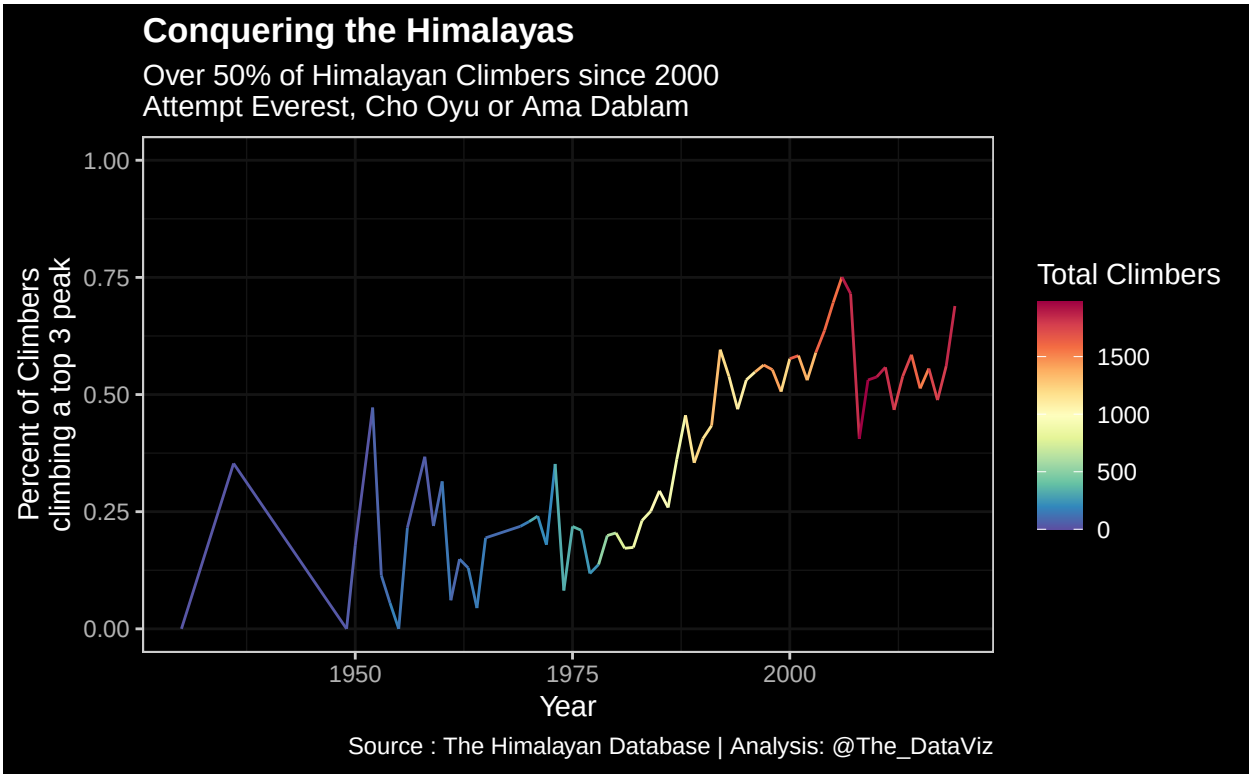
What share of all climbers climb the top 10 peaks

```
top_3_peaks <- members %>%
  filter(hired==FALSE) %>%
  group_by(peak_name) %>%
  count() %>%
  ungroup() %>%
  arrange(desc(n)) %>%
  top_n(3,n)

peak_popularity <- members %>%
  filter(hired==FALSE) %>%
  mutate(top3 = peak_name %in% top_3_peaks$peak_name) %>%
  group_by(top3,year) %>%
  count() %>%
  ungroup() %>%
  spread(top3,n) %>%
  replace(is.na(.), 0) %>%
  mutate(total = `FALSE`+`TRUE`,
         perc_top3 = `TRUE`/total,
         perc_rest = 1-perc_top3) %>%
  filter(total>25)

peak_popularity
```

```
## # A tibble: 68 x 6
##     year `FALSE` `TRUE` total perc_top3 perc_rest
##    <dbl>   <int>  <int> <int>     <dbl>     <dbl>
##  1  1930      26      0    26    0          1
##  2  1936      22     12    34    0.353      0.647
##  3  1949      45      0    45    0          1
##  4  1950      23      5    28    0.179      0.821
##  5  1952      38     34    72    0.472      0.528
##  6  1953     101     13   114    0.114      0.886
##  7  1954     171     10   181    0.0552     0.945
##  8  1955     136      0   136    0          1
##  9  1956      40     11    51    0.216      0.784
## 10  1958      31     18    49    0.367      0.633
```

```
## # ... with 58 more rows
```

```
myPalette <- colorRampPalette(rev(brewer.pal(11, "Spectral")))

ggplot(
  data = peak_popularity,
  aes(x=year,y=perc_top3,color=total)
) + geom_line() +
  ylim(c(0,1)) +
  ylab("Percent of Climbers\nclimbing a top 3 peak")+
  xlab("Year")+
  labs(title="Conquering the Himalayas",
       subtitle="Over 50% of Himalayan Climbers since 2000\nAttempt Everest, Cho Oyu or Ama Dablam",
       color="Total Climbers",
       caption="Source : The Himalayan Database | Analysis: @The_DataViz"
       ) +
  scale_color_gradientn(colors = myPalette(100),
                        limits=c(0, max(peak_popularity$total))) +
  dark_theme_bw()+
  theme(plot.title = element_text(face="bold"))
```




```
ggsave("Top3_Peak_Climber_Share.png")
```

Members Citizenship

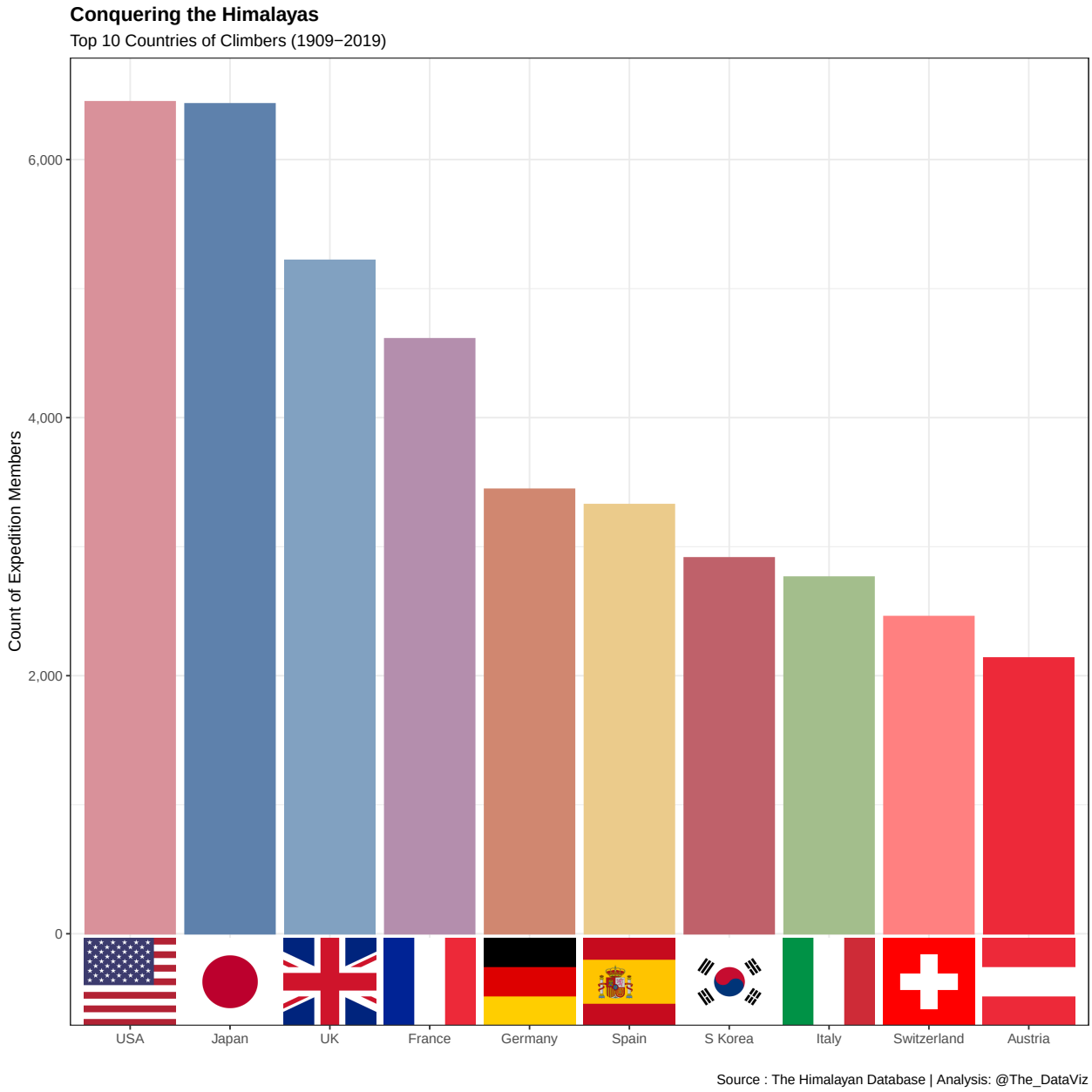
Build Data frame with image links from Flags Folder (ordered by country name)

```
citizenship_counts <- members %>% filter(hired==FALSE) %>%
  mutate(citizenship = str_replace(citizenship,"W Germany","Germany")) %>%
```

```
  group_by(citizenship) %>%
  count() %>%
  ungroup() %>%
  arrange(desc(n)) %>%
  top_n(10,n) %>%
  arrange(citizenship) %>%
  mutate(img = list.files(here::here('9-22-2020 Conquering The Himilayas',
                                     'Flags'), full.names = TRUE)) %>%
  arrange(desc(n)) %>% mutate(rank = 1:10)
citizenship_counts
```

```
## # A tibble: 10 x 4
##    citizenship     n img                                                  rank
##    <chr>       <int> <chr>                                               <int>
##  1 USA          6448 /Users/hunterkempf/Documents/GitHub/Tidy-Tuesday/9-2~     1
##  2 Japan        6432 /Users/hunterkempf/Documents/GitHub/Tidy-Tuesday/9-2~     2
##  3 UK           5219 /Users/hunterkempf/Documents/GitHub/Tidy-Tuesday/9-2~     3
##  4 France       4611 /Users/hunterkempf/Documents/GitHub/Tidy-Tuesday/9-2~     4
##  5 Germany      3445 /Users/hunterkempf/Documents/GitHub/Tidy-Tuesday/9-2~     5
##  6 Spain        3326 /Users/hunterkempf/Documents/GitHub/Tidy-Tuesday/9-2~     6
##  7 S Korea      2913 /Users/hunterkempf/Documents/GitHub/Tidy-Tuesday/9-2~     7
##  8 Italy        2764 /Users/hunterkempf/Documents/GitHub/Tidy-Tuesday/9-2~     8
##  9 Switzerland  2458 /Users/hunterkempf/Documents/GitHub/Tidy-Tuesday/9-2~     9
## 10 Austria      2137 /Users/hunterkempf/Documents/GitHub/Tidy-Tuesday/9-2~    10
```

```
colors <- c("#D9919A", "#5E81AC", "#81A1C1", "#B48EAD", "#D08770",
            "#EBCB8B", "#BF616A", "#A3BE8C", "#FF8080",  "#ED2939")

ggplot(citizenship_counts, aes(reorder(citizenship, desc(n)), n)) +
  geom_col(
    aes(fill = as.factor(rank),
        color = as.factor(rank)),
    show.legend = FALSE) +
  geom_image(y = -370,
             aes(image = img),
             size = rep(0.09, 10),  # Play with this a bit
             by = "width") +
  scale_fill_manual(values = colors) +
  scale_color_manual(values = colors) +
  scale_y_continuous(labels = scales::comma) +
  expand_limits(y = -370) +
  labs(x = "", y="Count of Expedition Members",
       title="Conquering the Himalayas",
       subtitle = 'Top 10 Countries of Climbers (1909-2019)',
       caption="Source : The Himalayan Database | Analysis: @The_DataViz") +
  theme_bw()+
  theme(plot.title = element_text(face="bold"))
```




```
ggsave("Climber_citizenship.png")
```

Look at hired help

```
members %>% filter(hired==TRUE) %>% mutate(citizenship = str_replace(citizenship,
                                                                     "W Germany",
                                                                     "Germany")) %>%
  group_by(citizenship) %>%
  count() %>%
  ungroup() %>%
  arrange(desc(n))
```

```
## # A tibble: 10 x 2
##    citizenship         n
##    <chr>           <int>
```

```
##  1 Nepal           15071
##  2 China             563
##  3 India              76
##  4 <NA>                7
##  5 India/Nepal         4
##  6 Nepal/India?        4
##  7 India?              3
##  8 Belgium             1
##  9 Nepal/Australia     1
## 10 Nepal/India         1
```

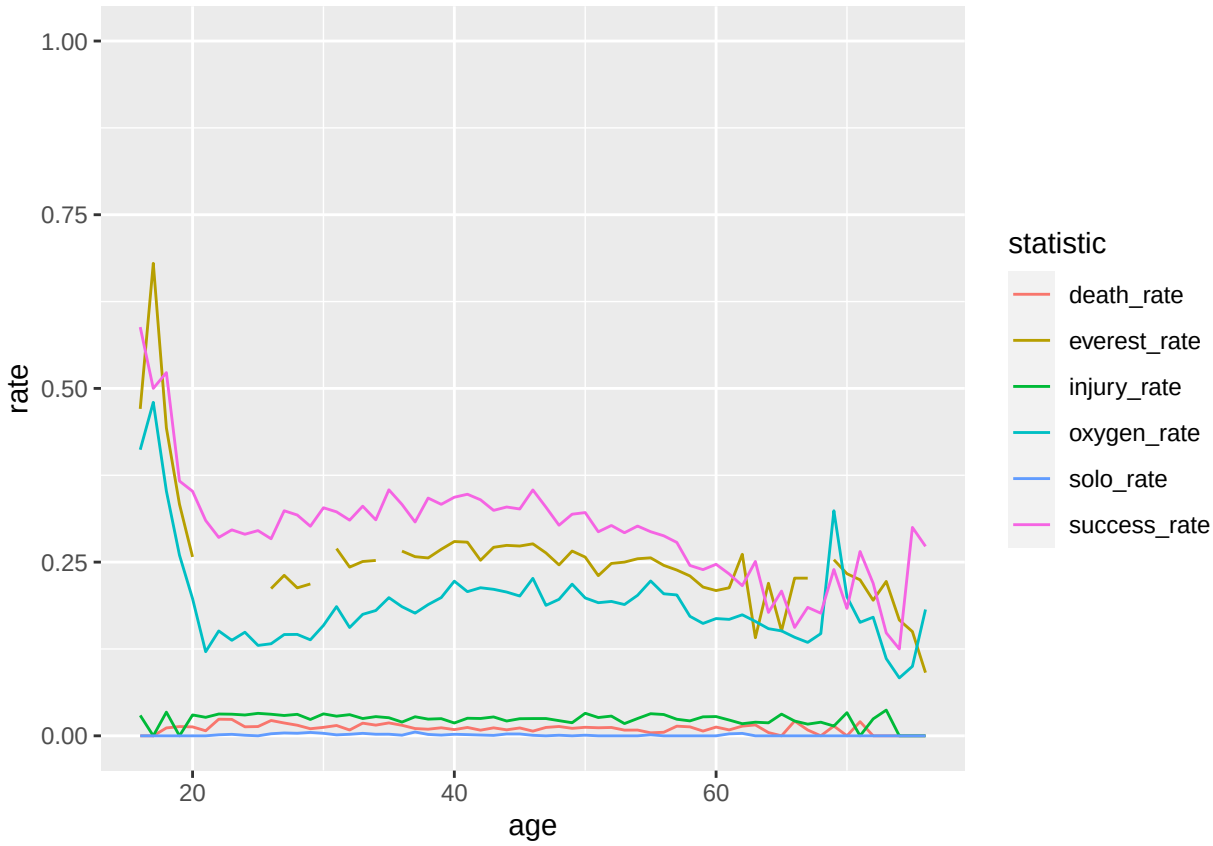
Average Age of Climbers

I will keep values with a count of at least 10

```
age_statistics <- members %>% filter(hired==F) %>% group_by(age) %>%
                              summarize( count = n(),
                                         everest_rate = mean(peak_name=="Everest"),
                                        success_rate = mean(success),
                                        oxygen_rate = mean(oxygen_used),
                                        injury_rate = mean(injured),
                                        death_rate = mean(died),
                                        solo_rate = mean(solo)) %>%
                              filter(count>=10) %>%
                              select(-count) %>%
                              gather("statistic","rate",-age)
```

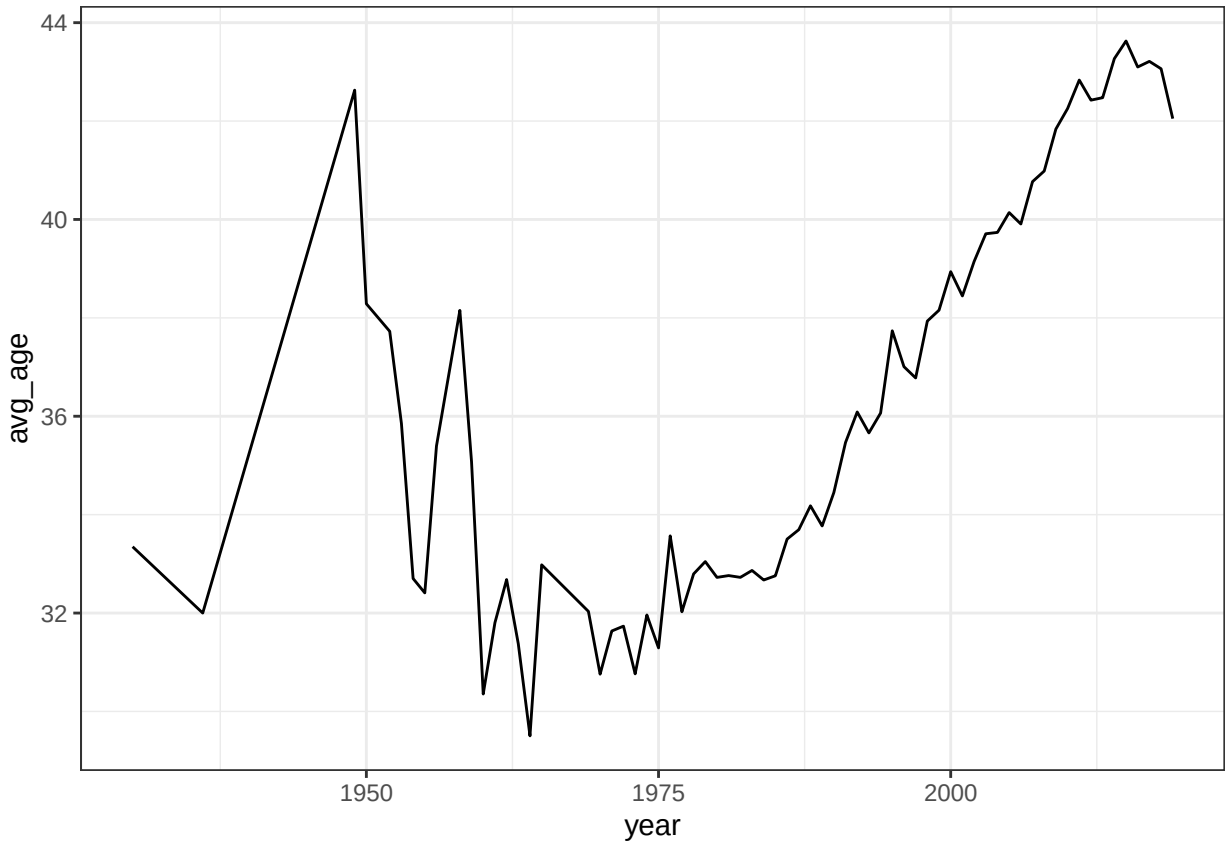
```
ggplot(
  data = age_statistics,
  aes(x= age,y=rate,color=statistic)
) + geom_line() + ylim(c(0,1))
```

```
## Warning: Removed 6 row(s) containing missing values (geom_path).
```




```
climber_age_by_year <- members %>% filter(hired==F) %>% group_by(year) %>%
                              summarize( count = n(),
                                         avg_age = mean(age,na.rm=T)
                              ) %>%
                              filter(count>25)
```

```
ggplot(
  data = climber_age_by_year,
  aes(x= year,y=avg_age)
) + geom_line(color="black") + theme_bw()
```


44
40
36
32
avg_age
1950
1975
2000
year