TidyTuesday: Beach volleyball

Liza Bolton

2020-05-19

- This session will be recorded and put up on Past events
- Click the stacked lines at the top left of this panel to open a helfpul navigation pane
- Remeber to fill out the weekly check-in by Thursday at 11:30 pm ET.

ASA DataFest Q&A

Prof Nathan Taback will drop in to answer questions you might have.

A quick tour of the ISSC

There are three important parts of the ISSC (well 4, if you count the most important part, YOU!)

- **Slack** is where all the real-time chatting and resource sharing happens.
- **SharePoint** is an archive for the community where you can find the previous 6 Sigma Sunday newsletters, resources and recordings from past events, as well as a range of resources in the General Resources library.
- **ASA DataFest@UofT site** for registration, some suggested resoruces and more information about the competition. https://datafestuoft.github.io/

Mini-challenge

Your mission, should you choose to accept it, is to complete a mini-data visualisation challenge by the end of the day.

1. Set up GitHub

You can definitely do this challenge even if you haven't sorted out your GitHub yet, but I'd strongly recommend making this one of your ISSC goals. More information in the first 6 Sigma Sunday newsletter. You may wish to create a repository to store this mini-project in called 'ISSC' or 'TidyTuesday' folder. I have one called 'ISSC' with the files from the this AND the two previous TidyTuesday & Talks.

2. Create an R Markdown document

Or it could be an R Script, but I prefer RMDs, like what this is written in. It is perfect for when you want your code, outputs and commentary to all be together.

3. Install/load packages

If you haven't installed `tidyverse` yet, you will need that package for today. It has `dplyr` and `ggplot` in it.


```
#install.packages("tidyverse")
#install.packages("lubridate")
#install.packages("ggthemes")
library("tidyverse")
library("lubridate")
library("ggthemes")

# Notice how I am using message=FALSE in this chunk to suppress the information
# about loading tidyverse. I don't what this as part of my final document because
# it isn't very pretty. Always suppress with care though,
# and if you're running in to issues, make sure to check this.
```

4. Load the data for this week.

There is more than one way to get this data. I'm going to use the `tidytuesdayR` package becasue I installed it last week. Choose the way that is right for you from these options.

```
tuesdata <- tidytuesdayR::tt_load('2020-05-19')
```

```
## --- Downloading #TidyTuesday Information for 2020-05-19 ----

## --- Identified 1 files available for download ----

## --- Downloading files ---
```

```
## --- Download complete ---
```

```
vb_matches <- tuesdata$vb_matches
rm(tuesdata) # remove the original file because we don't need it any more
```

5. Take a look at the data

```
glimpse(vb_matches)
```

```
## Rows: 76,756
## Columns: 65
## $ circuit               <chr> "AVP", "AVP", "AVP", "AVP", "AVP", "AVP", "AV...
## $ tournament            <chr> "Huntington Beach", "Huntington Beach", "Hunt...
## $ country               <chr> "United States", "United States", "United Sta...
## $ year                  <dbl> 2002, 2002, 2002, 2002, 2002, 2002, 2002, 200...
## $ date                  <date> 2002-05-24, 2002-05-24, 2002-05-24, 2002-05-...
## $ gender                <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", ...
## $ match_num             <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14...
## $ w_player1             <chr> "Kevin Wong", "Brad Torsone", "Eduardo Bacil"...
## $ w_p1_birthdate        <date> 1972-09-12, 1975-01-14, 1971-03-11, 1970-01-...
## $ w_p1_age              <dbl> 29.69473, 27.35661, 31.20329, 32.38604, 32.05...
## $ w_p1_hgt              <dbl> 79, 78, 74, 78, 75, 75, 78, 77, 75, 79, 73, 7...
## $ w_p1_country          <chr> "United States", "United States", "Brazil", "...
## $ w_player2             <chr> "Stein Metzger", "Casey Jennings", "Fred Souz...
## $ w_p2_birthdate        <date> 1972-11-17, 1975-07-10, 1972-05-13, 1960-11-...
## $ w_p2_age              <dbl> 29.51403, 26.87201, 30.02875, 41.55236, 29.80...
## $ w_p2_hgt              <dbl> 75, 75, 79, 74, 80, 77, 78, 79, 75, 76, 76, 7...
## $ w_p2_country          <chr> "United States", "United States", "Brazil", "...
## $ w_rank                <chr> "1", "16", "24", "8", "5", "12", "13", "4", "...
## $ l_player1             <chr> "Chuck Moore", "Mark Paaluhi", "Adam Jewell",...
## $ l_p1_birthdate        <date> 1973-08-18, 1971-03-08, 1975-06-24, 1973-02-...
## $ l_p1_age              <dbl> 28.76386, 31.21150, 26.91581, 29.27036, 26.32...
## $ l_p1_hgt              <dbl> 76, 75, 77, 76, 73, NA, 75, 75, 68, 75, 77, 7...
## $ l_p1_country          <chr> "United States", "United States", "United Sta...
## $ l_player2             <chr> "Ed Ratledge", "Nick Hannemann", "Collin Smit...
## $ l_p2_birthdate        <date> 1976-12-16, 1972-01-12, 1975-05-26, 1969-10-...
## $ l_p2_age              <dbl> 25.43463, 30.36277, 26.99521, 32.63244, 24.16...
## $ l_p2_hgt              <dbl> 80, 78, 76, 80, 75, 76, 81, 77, 77, 74, 73, 7...
## $ l_p2_country          <chr> "United States", "United States", "United Sta...
## $ l_rank                <chr> "32", "17", "9", "25", "28", "21", "20", "29"...
## $ score                 <chr> "21-18, 21-12", "21-16, 17-21, 15-10", "21-18...
## $ duration              <time> 00:33:00, 00:57:00, 00:46:00, 00:44:00, 01:0...
## $ bracket               <chr> "Winner's Bracket", "Winner's Bracket", "Winn...
## $ round                 <chr> "Round 1", "Round 1", "Round 1", "Round 1", "...
## $ w_p1_tot_attacks      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p1_tot_kills        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p1_tot_errors       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p1_tot_hitpct       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p1_tot_aces         <dbl> 1, 0, 0, 0, 1, 0, 0, 0, 1, 2, 4, 0, 1, 0, 0, ...
## $ w_p1_tot_serve_errors <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p1_tot_blocks       <dbl> 7, 4, 2, 3, 0, 0, 0, 0, 2, 3, 0, 3, 4, 0, 2, ...
```

```
## $ w_p1_tot_digs         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p2_tot_attacks      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p2_tot_kills        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p2_tot_errors       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p2_tot_hitpct       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p2_tot_aces         <dbl> 2, 4, 0, 0, 0, 0, 0, 0, 0, 4, 0, 1, 2, 0, 0, ...
## $ w_p2_tot_serve_errors <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ w_p2_tot_blocks       <dbl> 0, 0, 4, 0, 6, 0, 0, 3, 3, 1, 5, 0, 0, 1, 0, ...
## $ w_p2_tot_digs         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p1_tot_attacks      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p1_tot_kills        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p1_tot_errors       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p1_tot_hitpct       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p1_tot_aces         <dbl> 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, ...
## $ l_p1_tot_serve_errors <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p1_tot_blocks       <dbl> 0, 2, 1, 2, 0, 0, 0, 0, 0, 1, 9, 1, 1, 1, 1, ...
## $ l_p1_tot_digs         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p2_tot_attacks      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p2_tot_kills        <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p2_tot_errors       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p2_tot_hitpct       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p2_tot_aces         <dbl> 0, 0, 0, 2, 0, 0, 0, 3, 0, 0, 0, 0, 1, 4, 0, ...
## $ l_p2_tot_serve_errors <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ l_p2_tot_blocks       <dbl> 1, 0, 0, 0, 1, 0, 0, 0, 0, 2, 0, 0, 0, 1, 1, ...
## $ l_p2_tot_digs         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
```

6. Wrangle the data

This data is pretty clean and tidy but we might want to play with a few things. I wanted to make seperate datasets so I could look at data by individual players across all their matches and look at general data about the players and the match.

```
vb_clean <- vb_matches %>%
  rowid_to_column(var = "match_ID")    # make an ID columns

l_player1 <- vb_clean %>%
  select(match_ID, gender, contains("l_p1"), contains("l_player1")) %>%
  rename(setNames(names(.), gsub("l_p1_", "", names(.)))) %>%
  rename(player = "l_player1") %>%
  select(match_ID, player, everything()) %>%  # try relocate() in dplyr 1.0.0
  mutate(status = "Lost", player_num = 1)

l_player2 <- vb_clean %>%
  select(match_ID, gender, contains("l_p2"), contains("l_player2")) %>%
  rename(setNames(names(.), gsub("l_p2_", "", names(.)))) %>%
  rename(player = "l_player2") %>%
  select(match_ID, player, everything()) %>%
    mutate(status = "Lost", player_num = 2)

w_player1 <- vb_clean %>%
  select(match_ID, gender, contains("w_p1"), contains("w_player1")) %>%
  rename(setNames(names(.), gsub("w_p1_", "", names(.)))) %>%
  rename(player = "w_player1") %>%
```

```
  select(match_ID, player, everything()) %>%
  mutate(status = "Won", player_num = 1)

w_player2 <- vb_clean %>%
  select(match_ID, gender, contains("w_p2"), contains("w_player2")) %>%
  rename(setNames(names(.), gsub("w_p2_", "", names(.)))) %>%
  rename(player = "w_player2") %>%
  select(match_ID, player, everything()) %>%
  mutate(status = "Won", player_num = 2)

# once I wrote this I realised it might be nice to write a function
# that does these similar steps instead...I'm not going to do that today,
# but let me know if you give it a go!

player_matches <- bind_rows(l_player1, l_player2, w_player1, w_player2)

# make a dataset with just unchaning information about each player
player_info <- player_matches %>%
  select(player, gender, birthdate, hgt, country) %>%
  unique()

# make a dataset with just information about the match
match_info <- vb_clean %>%
  select(-contains("p1"), -contains("p2"), -contains("player")) %>%
  separate(score, into=c("score_set1", "score_set2", "score_set3"), sep = ",")
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 52319 rows [1, 3,
## 4, 7, 8, 9, 10, 12, 15, 16, 17, 18, 21, 23, 24, 25, 26, 27, 28, 29, ...].
```

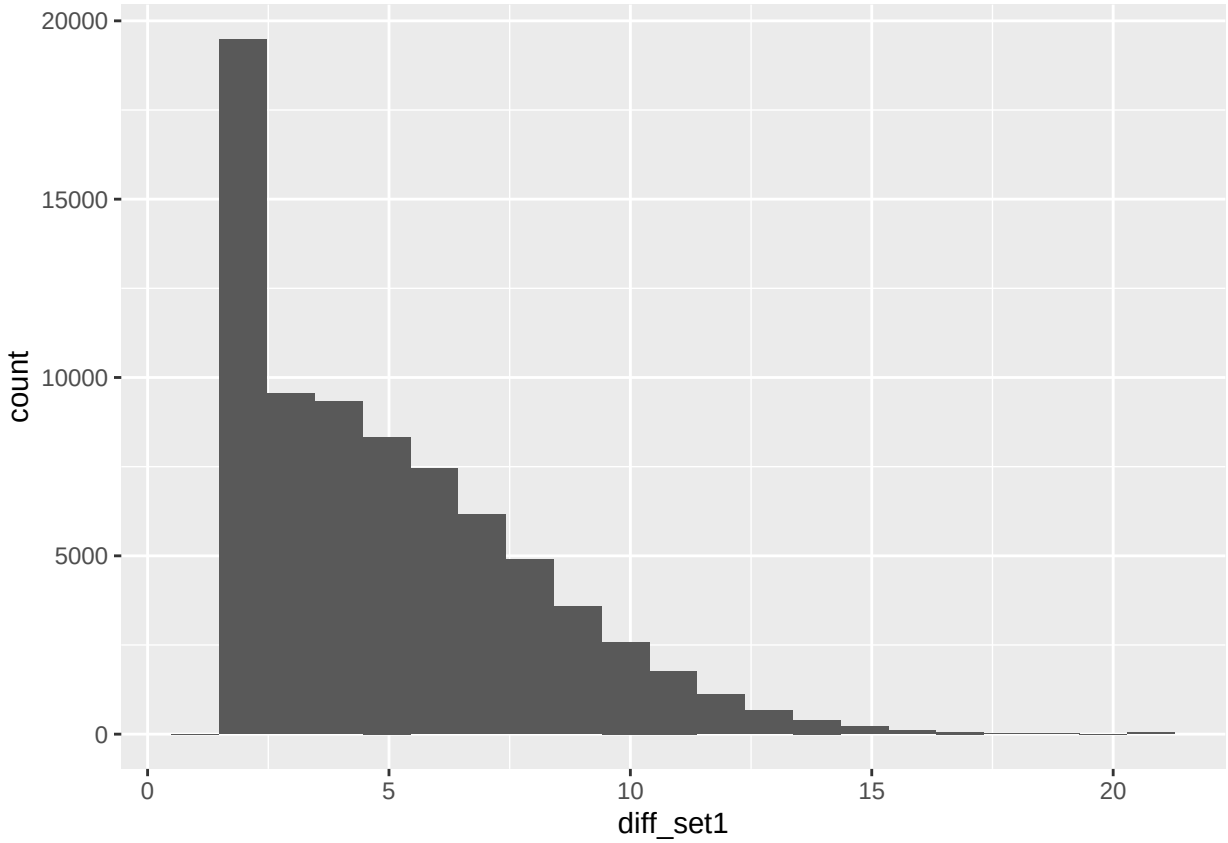
7. Create at least 3 exploratory plots/summary statistics.

You might find the Cookbook for R graphics from the BBC helpful, as well as the resources in 6 Sigma Sunday #2 on using dplyr and ggplot.

Explore!

What is the usual difference between scores in set 1 of a match?

```
match_info2 <- match_info %>%
  filter(score_set1 != "Forfeit or other") %>%
  filter(!grepl("retired", score_set1)) %>%
  rowwise() %>%
  mutate(diff_set1 = abs(eval(parse(text=score_set1))))

match_info2 %>%
  ggplot(aes(x = diff_set1)) +
  geom_histogram(binwidth = 0.99)
```



Explore!

What proportion of matches go to the third set?

```
match_info %>%
  mutate(three_sets = ifelse(is.na(score_set3), FALSE, TRUE)) %>%
  summarise(prop_3_set = mean(three_sets))
```

```
## # A tibble: 1 x 1
##   prop_3_set
##        <dbl>
## 1      0.318
```

Explore!

Win rates by players?

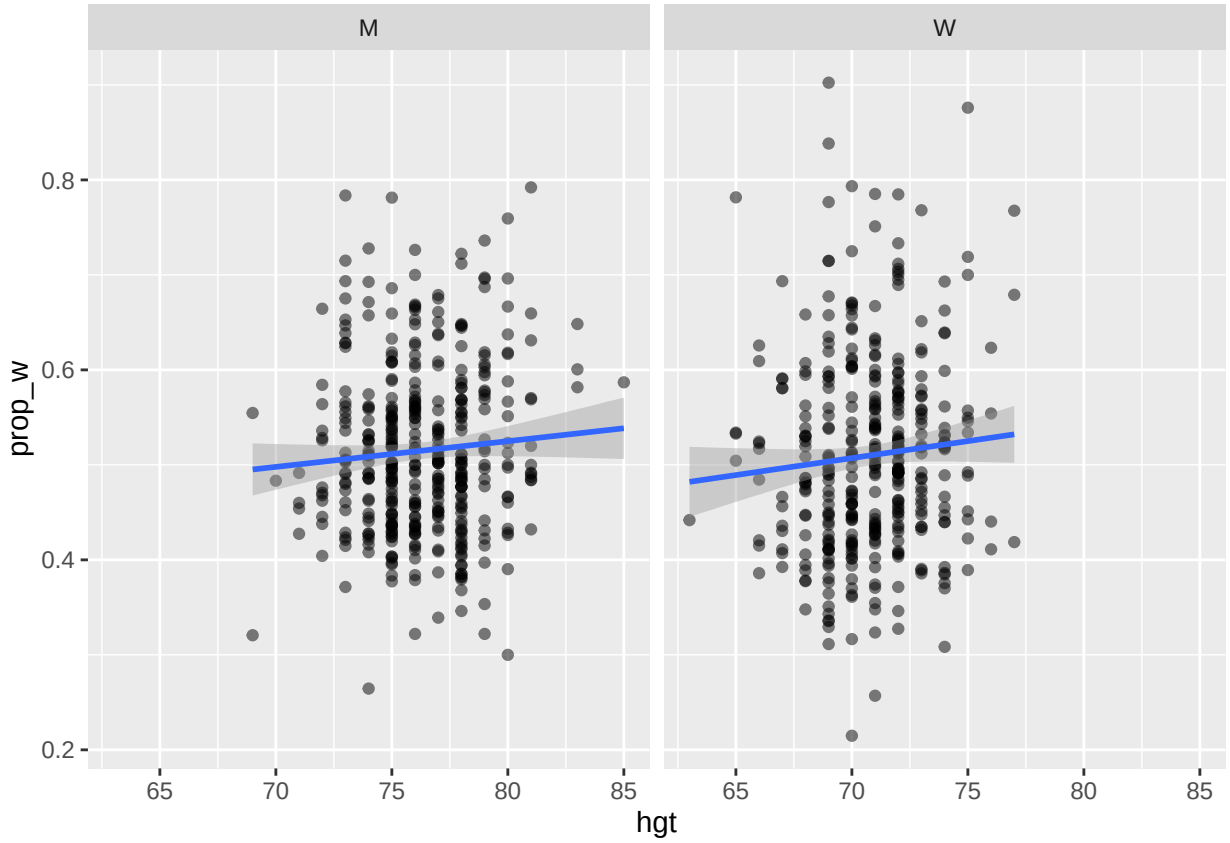
```
winrate <- player_matches %>%
  mutate(w_l = ifelse(status=="Won", 1, 0)) %>%
  group_by(player) %>%
  summarise(prop_w = mean(w_l), matches = n()) %>%
  left_join(player_info, by = "player") %>%
  filter(!is.na(hgt)) %>%
  mutate(country_top = fct_lump_n(country, n = 5))
```

```
table(winrate$country_top)
```

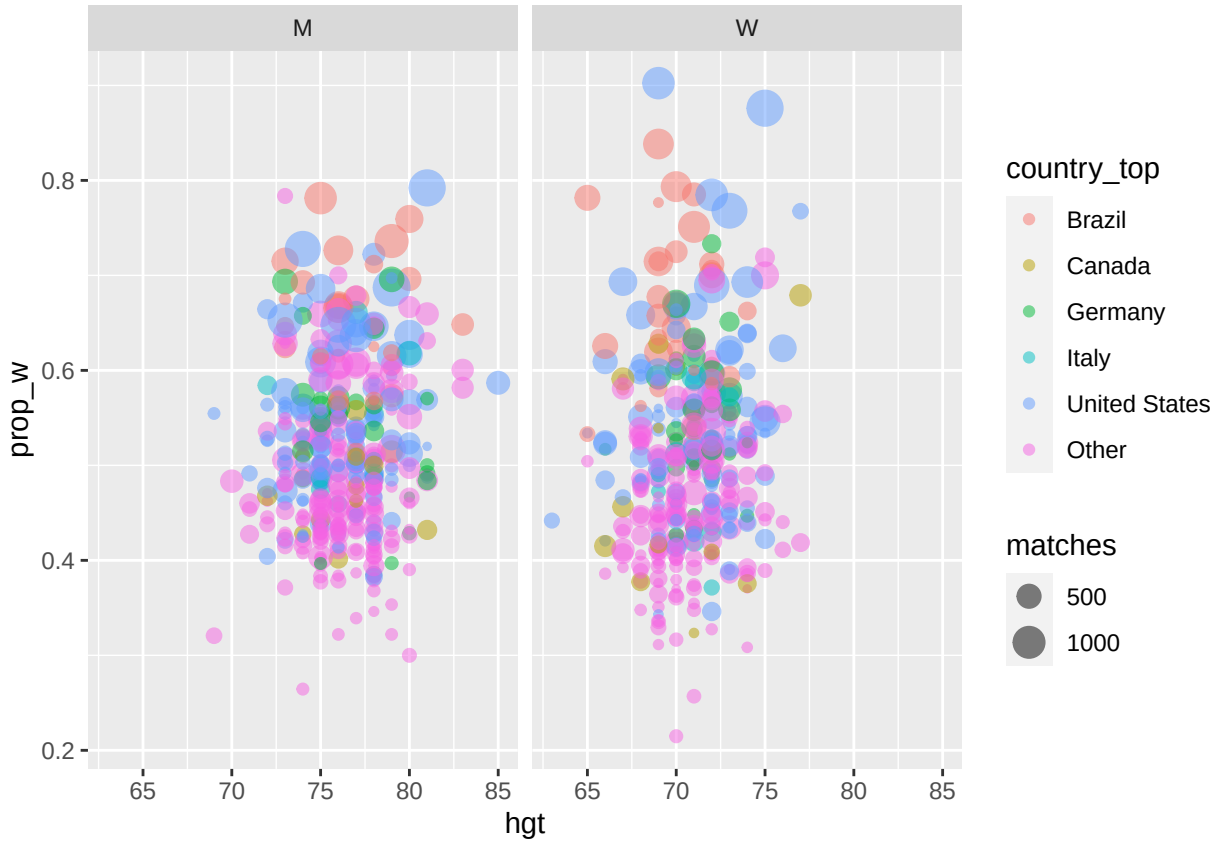
```
##
##        Brazil        Canada       Germany         Italy United States
##           156           138           137           140          2001
##         Other
##          2741
```

```
winrate %>%
  filter(matches >= 100) %>%
  ggplot(aes(x = hgt, y = prop_w)) +
  geom_point(alpha = 0.5) +
  facet_wrap(~gender) +
  geom_smooth(method="lm")
```

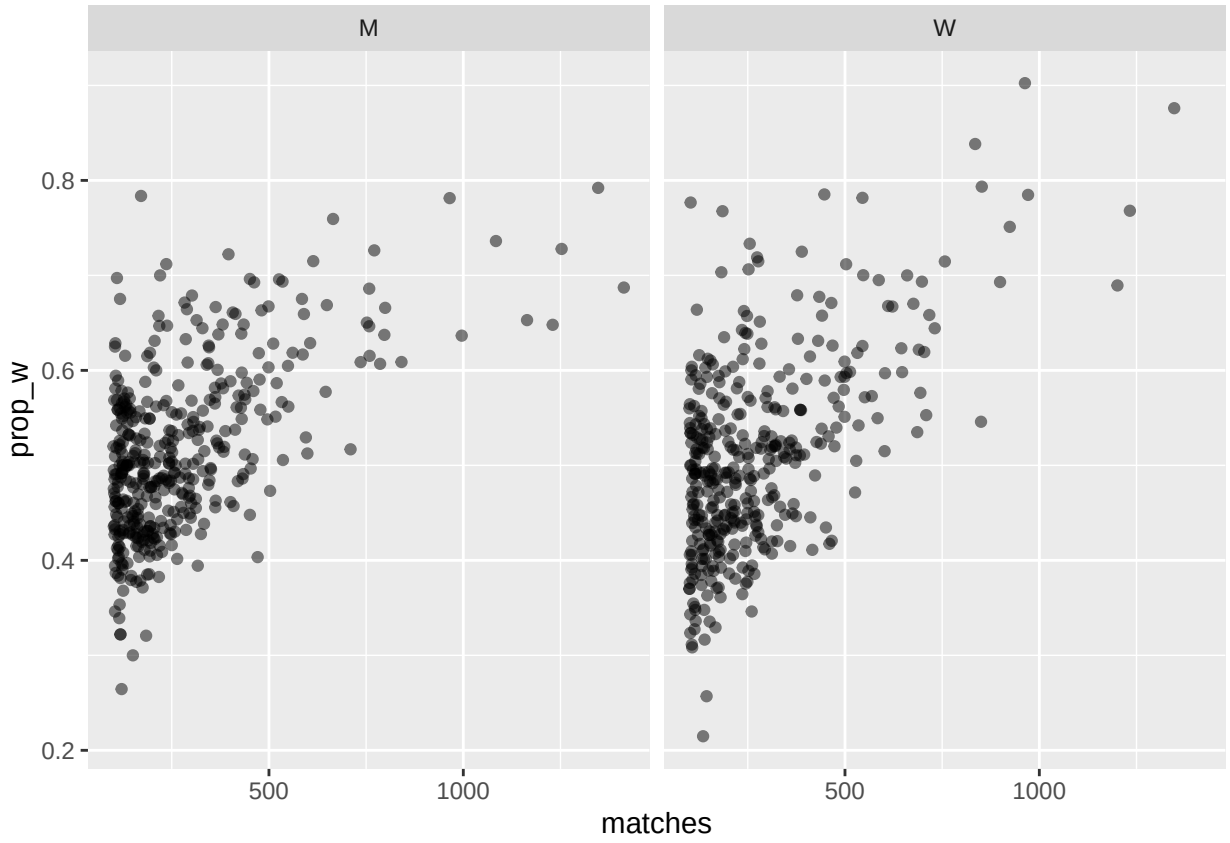
```
## `geom_smooth()` using formula 'y ~ x'
```




```
winrate %>%
  filter(matches >= 100) %>%
  ggplot(aes(x = hgt, y = prop_w, colour = country_top, size = matches)) +
  geom_point(alpha = 0.5) +
  facet_wrap(~gender)
```




```
winrate %>%
  filter(matches >= 100) %>%
  ggplot(aes(x = matches, y = prop_w)) +
  geom_point(alpha = 0.5) +
  facet_wrap(~gender)
```

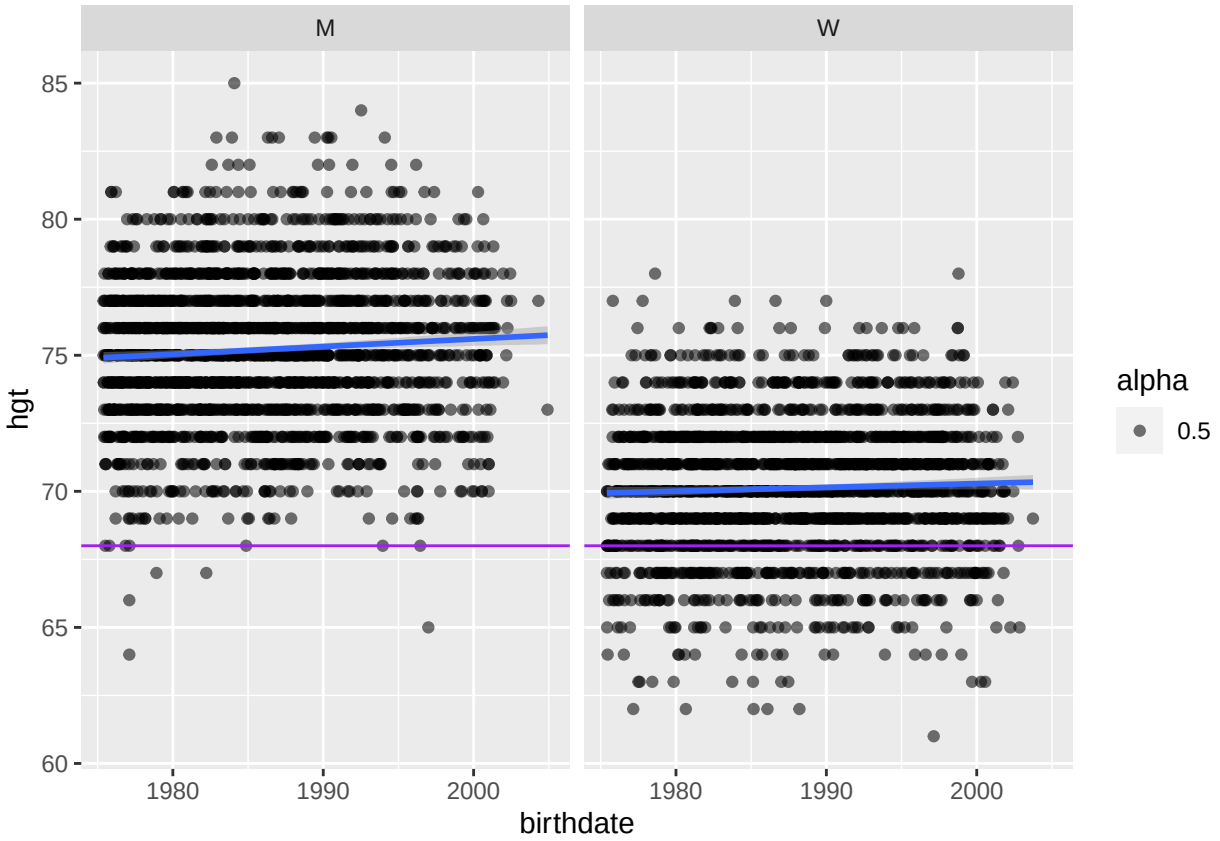


Explore!

Are players getting any taller?

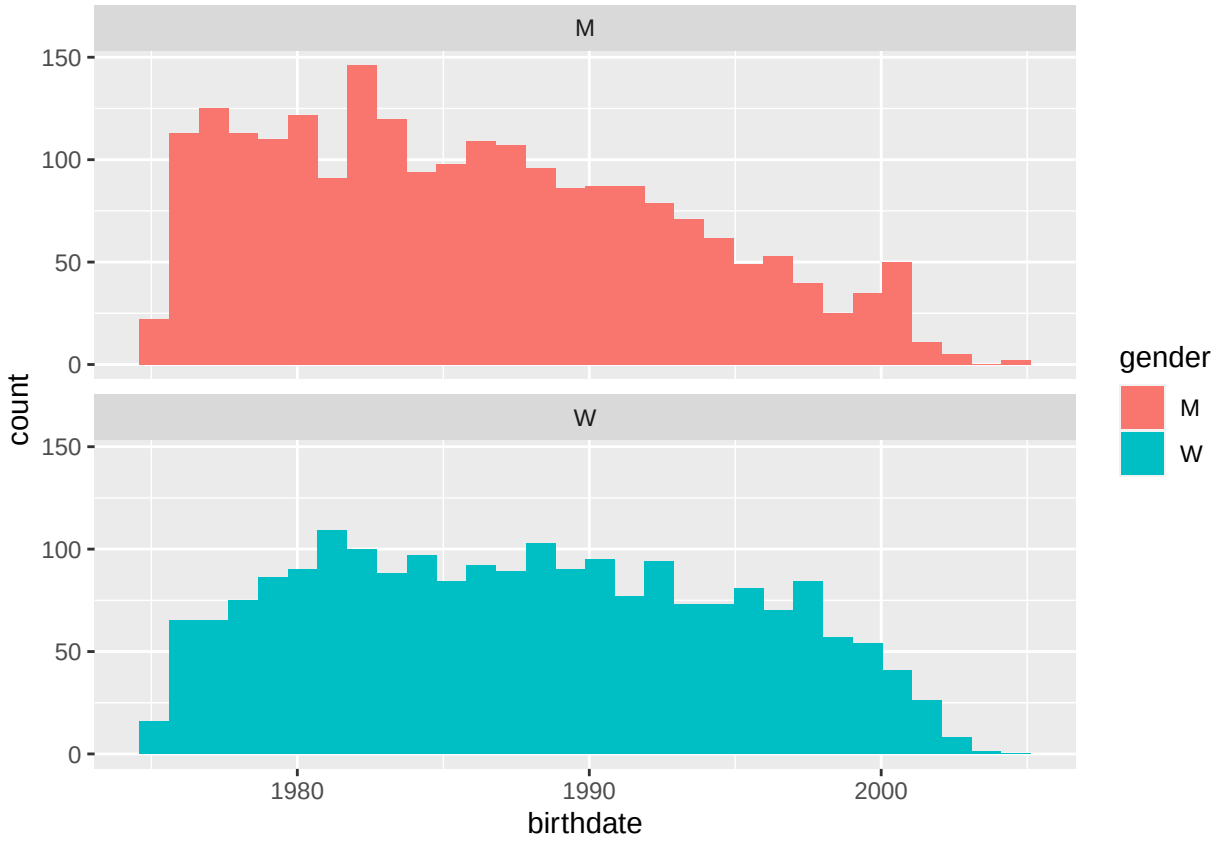
```
player_info %>%
  filter(birthdate>1960) %>%
  filter(!is.na(hgt)) %>%
  ggplot(aes(x = birthdate, y = hgt)) +
  geom_point(aes(alpha = 0.5)) +
  geom_smooth(method="lm") +
  facet_wrap(~gender) +
  geom_hline(aes(yintercept = 68), color = "purple")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

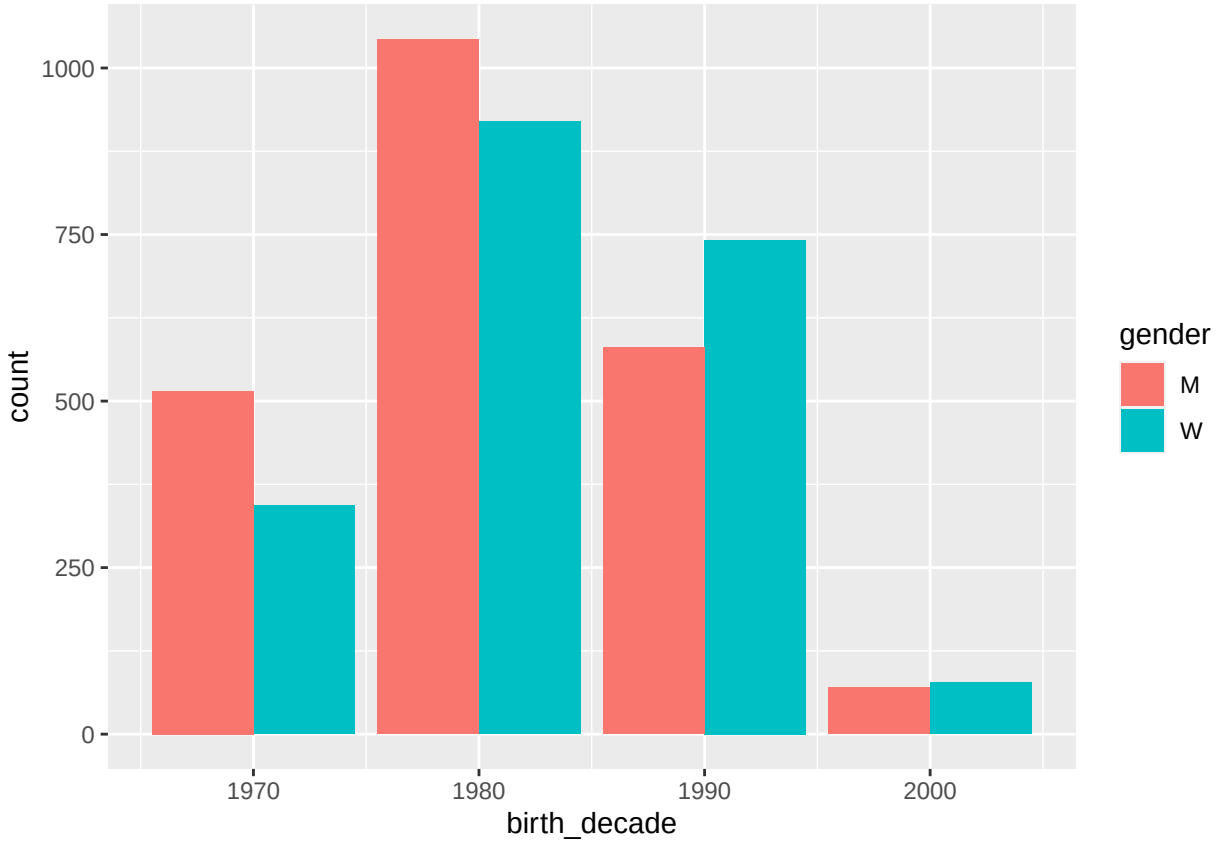



```
player_info %>%
  filter(birthdate>1960) %>%
  filter(!is.na(hgt)) %>%
  ggplot(aes(x = birthdate, fill=gender)) +
  geom_histogram() +
  facet_wrap(~gender, nrow=2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
player_info %>%
  filter(birthdate>1960) %>%
  filter(!is.na(hgt)) %>%
  mutate(birth_decade = floor_date(birthdate, years(10))) %>%
  group_by(birth_decade, gender) %>%
  mutate(count = n()) %>%
  ggplot(aes(x = birth_decade, y = count, fill = gender)) +
    geom_bar(stat="identity", position = "dodge")
```

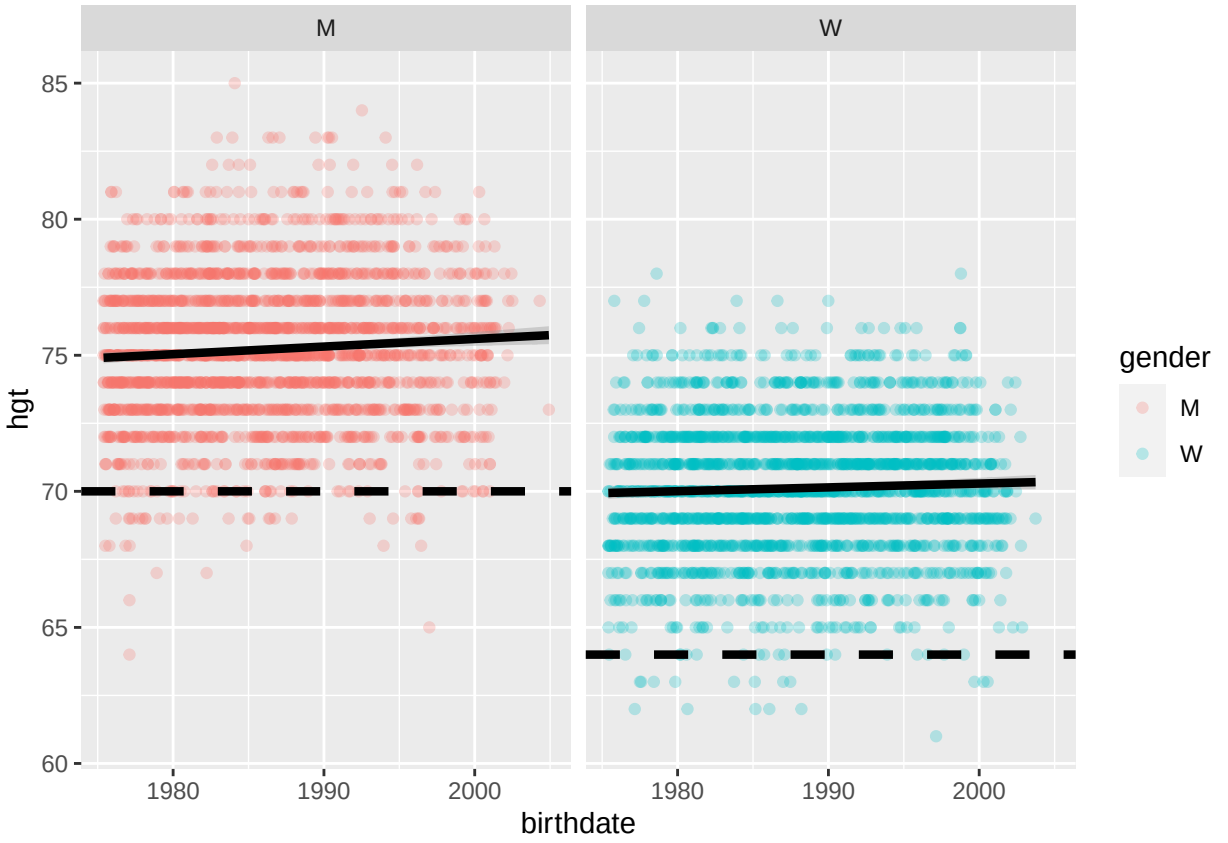


8. Choose one plot to improve and use/include the following:

I've chosen the height/gender/age.

The average height of Canadian men is 5' 10" (70 inches) and the average height of Canadian women is 5' 4" (64 inches). Source: https://www.cbc.ca/news/health/height-growth-canada-1.3695398

```
can_pop <- tibble(hgt = c(70, 64), gender = c("M", "W"))

winrate_filter <- winrate %>%
  filter(birthdate>1960) %>%
  filter(!is.na(hgt))

base_plot <- winrate_filter %>%
  ggplot(aes(x = birthdate, y = hgt, color = gender)) +
  geom_point(alpha = 0.25) +
  geom_smooth(method="lm", formula = y ~ x, colour = "black", size = 1.5) +
  facet_wrap(~gender) +
  geom_hline(aes(yintercept = hgt), can_pop, color = "black", size = 1.5, lty = "dashed")
base_plot
```




```
## model for all
summary(lm(hgt~birthdate, data = winrate_filter))
```

```
##
## Call:
## lm(formula = hgt ~ birthdate, data = winrate_filter)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4572  -2.7180   0.0068   2.5331  12.1875
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.320e+01  1.448e-01  505.43  < 2e-16 ***
## birthdate   -7.452e-05  2.166e-05   -3.44 0.000588 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.632 on 4289 degrees of freedom
## Multiple R-squared:  0.002751,   Adjusted R-squared:  0.002519
## F-statistic: 11.83 on 1 and 4289 DF,  p-value: 0.0005878
```

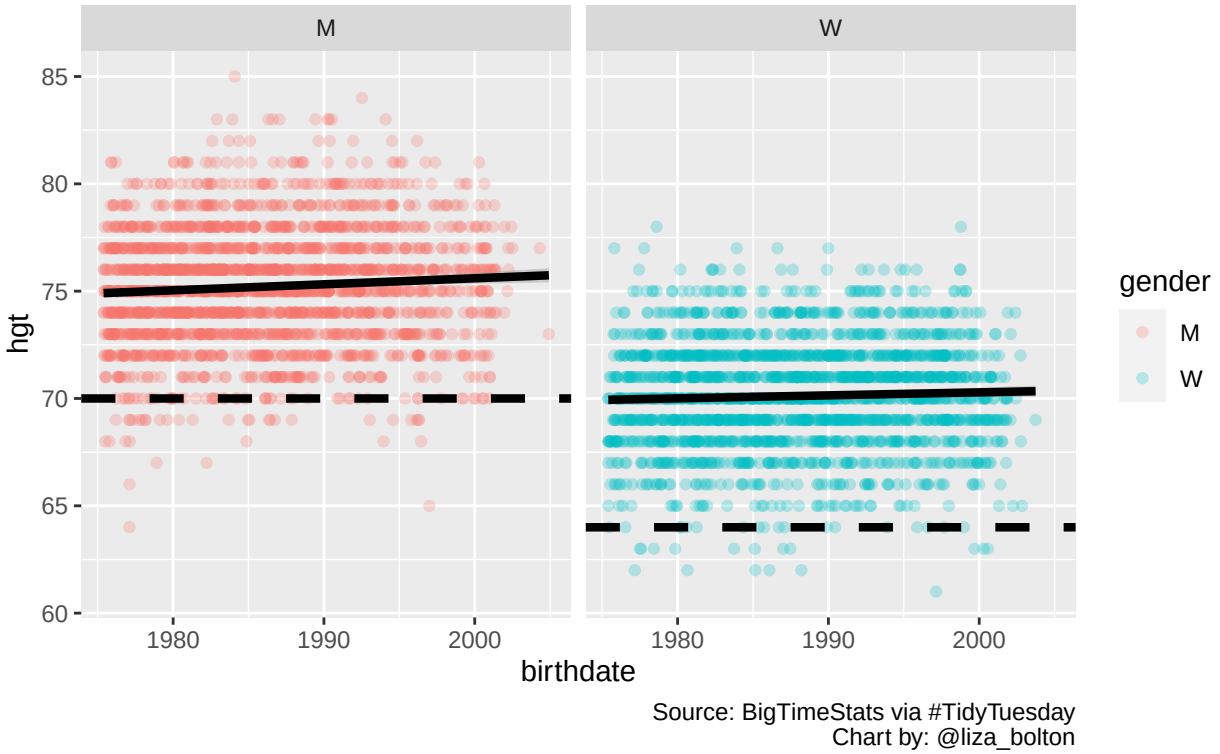
```
## model with gender interaction
summary(lm(hgt~birthdate*gender, data = winrate_filter))
```

```
##
```

```
## Call:
## lm(formula = hgt ~ birthdate * gender, data = winrate_filter)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.958  -1.638  -0.057   1.815   9.846
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.476e+01  1.405e-01 532.202  < 2e-16 ***
## birthdate          7.673e-05  2.209e-05   3.474 0.000518 ***
## genderW           -4.894e+00  2.089e-01 -23.422  < 2e-16 ***
## birthdate:genderW -3.841e-05  3.121e-05  -1.230 0.218593
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.594 on 4287 degrees of freedom
## Multiple R-squared:  0.4916, Adjusted R-squared:  0.4913
## F-statistic:  1382 on 3 and 4287 DF,  p-value: < 2.2e-16
```

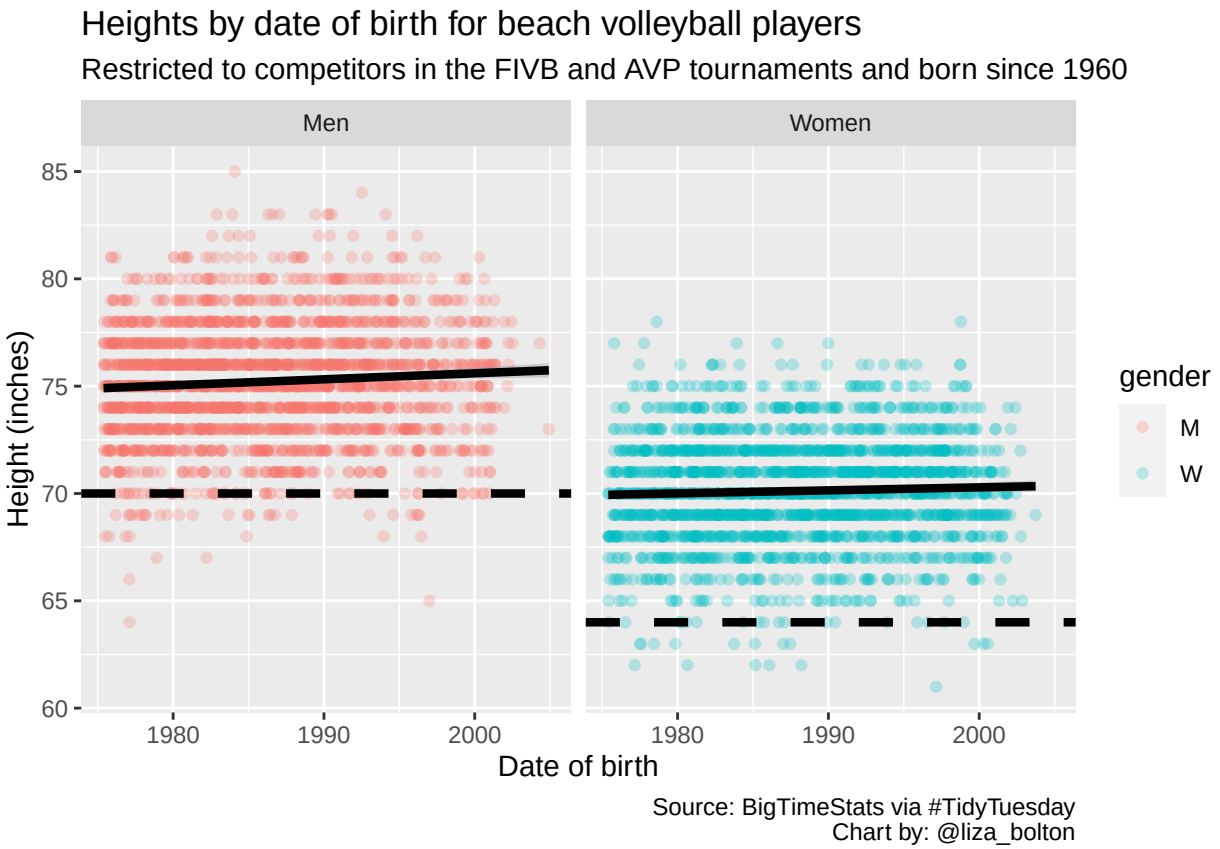
A title and subtitle AND caption acknowledging the data source + your name

```
p1 <- base_plot +
  labs(title = "Heights by date of birth for beach volleyball players",
         subtitle = "Restricted to competitors in the FIVB and AVP tournaments and born since 1960",
       caption = "Source: BigTimeStats via #TidyTuesday\n Chart by: @liza_bolton")

p1
```



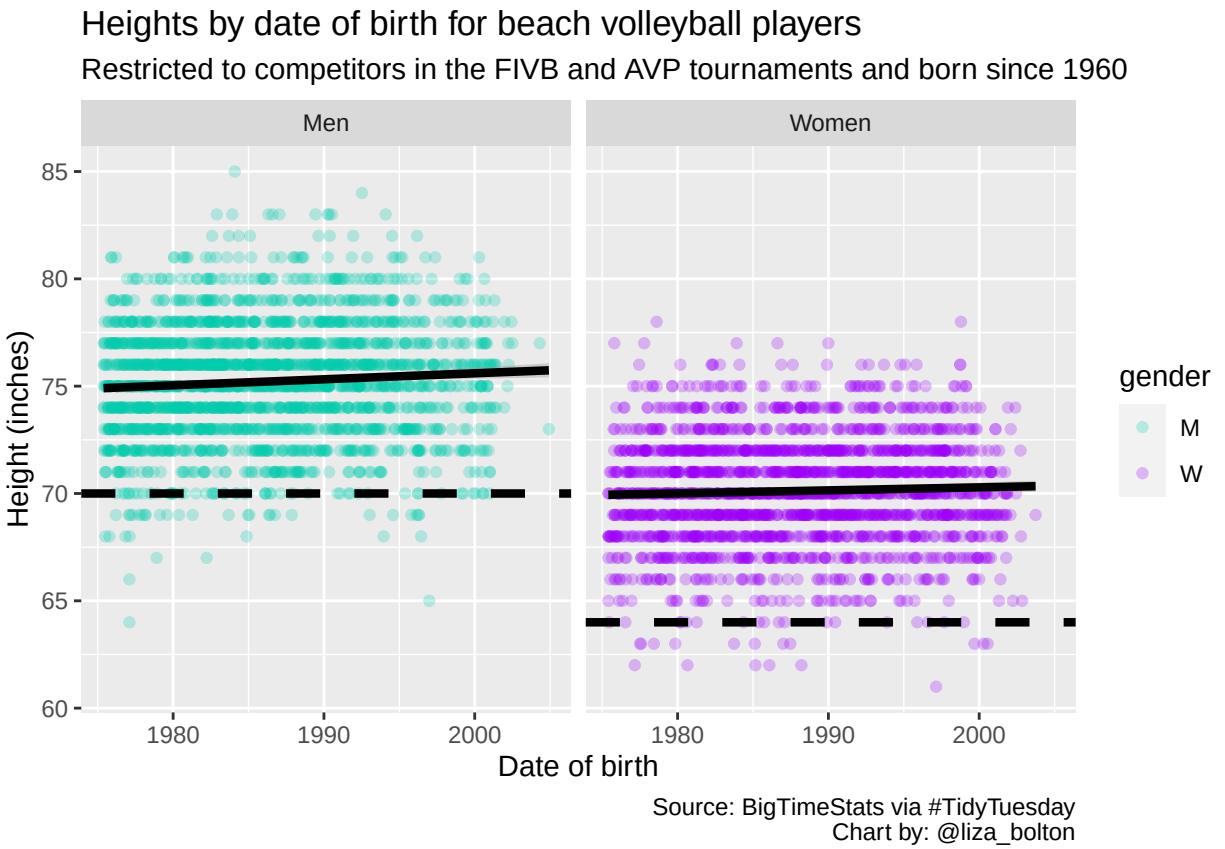
Labelled axes

```
# New facet label names for gender (don't want just letters)
gender.labs <- c("Men", "Women")
names(gender.labs) <- c("M", "W")

p2 <- p1 +
  facet_grid(~gender, labeller = labeller(gender = gender.labs)) +
  xlab("Date of birth") +
  ylab("Height (inches)")
p2
```



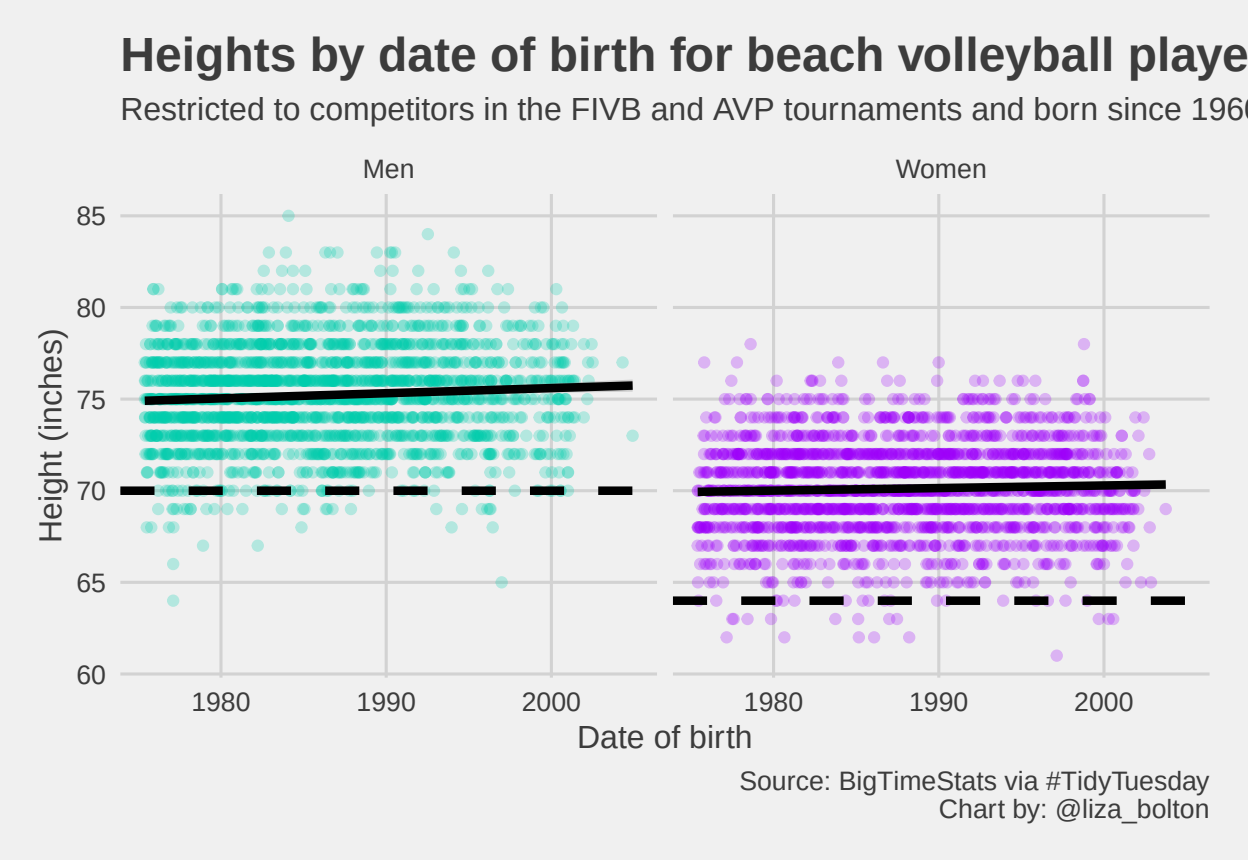
An appropriate colour palette

```
# Get my gender colours if you want them
source("https://gist.githubusercontent.com/elb0/ae55809dbc610a50fba7bb5377497cd6/raw/1b17ddb92d45f5caee

p3 <- p2 +
  scale_color_manual(values = rev(suffrage_cols))
p3
```



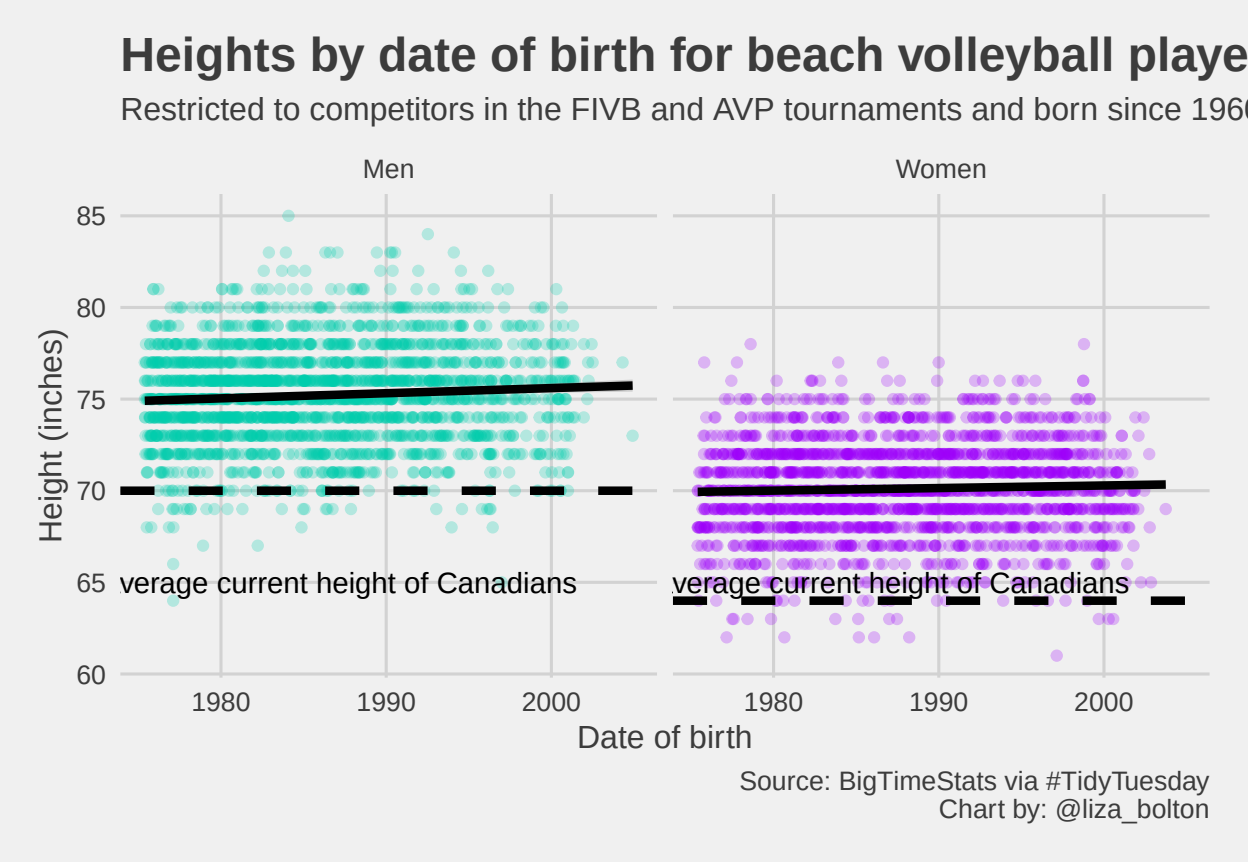
Explicitly use a theme (check out this list of defaults included with ggplot or get the ggtheme package)

```
p4 <- p3 +
  theme_fivethirtyeight() +
  theme(legend.position = "none", axis.title = element_text()) +
  xlab("Date of birth") +
  ylab("Height (inches)")
p4
```

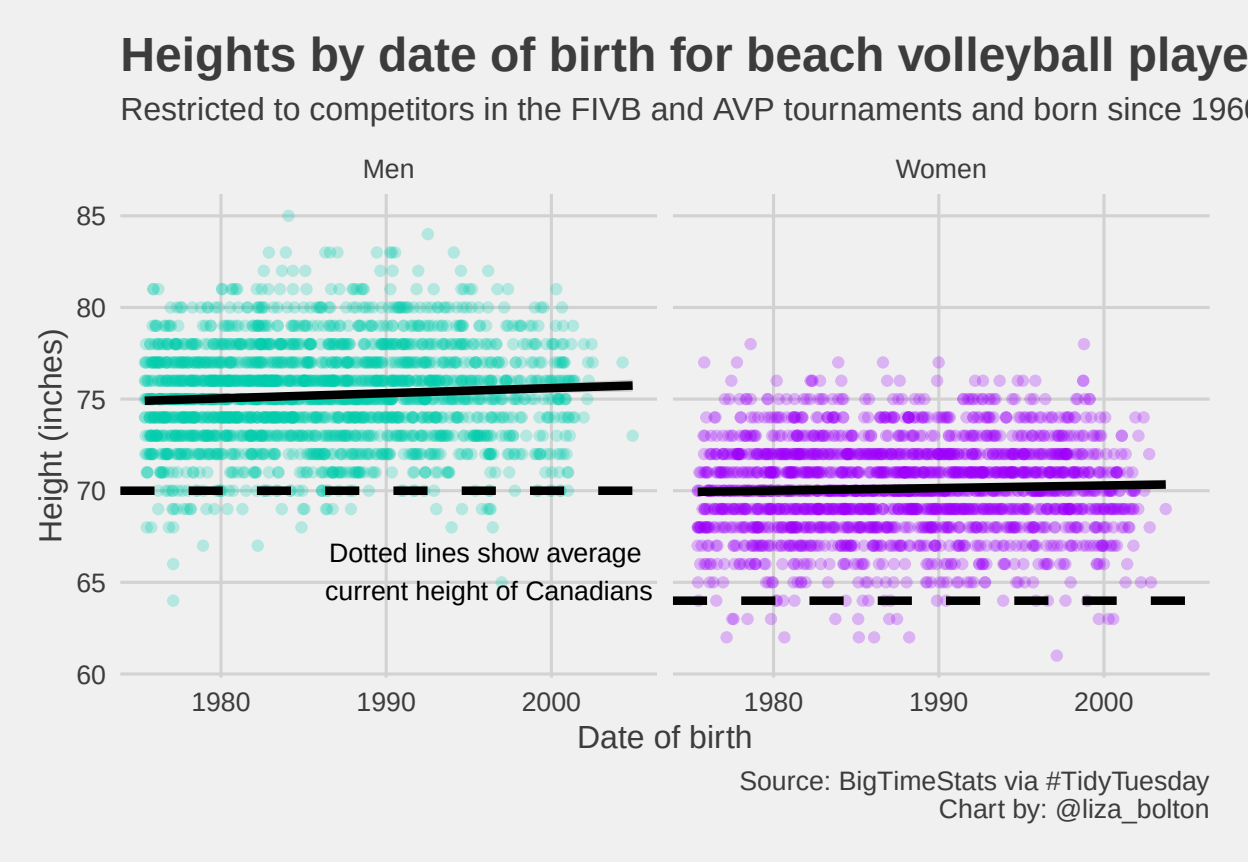


BONUS: Add an annotation

```
p5 <- p4 +
  annotate("text", label = "Dotted lines show average current height of Canadians", x = as.Date("1980-0
p5
```




```
anno <- tibble(x1 = as.Date("1996-01-01"),
                   y1 = 65.6,
                   gender = "M")

p5_2 <- p4 +
  geom_text(data = anno, aes(x = x1,  y = y1, label = "Dotted lines show average\n current height of Ca

p5_2
```



9. Save the plot using ggsave().

If you run `?ggsave`, it will tell you that "`ggsave()` is a convenient function for saving a plot. It defaults to saving the last plot that you displayed, using the size of the current graphics device. It also guesses the type of graphics device from the extension."

```
ggsave("vb_heights_birthyear_gender.png")
```

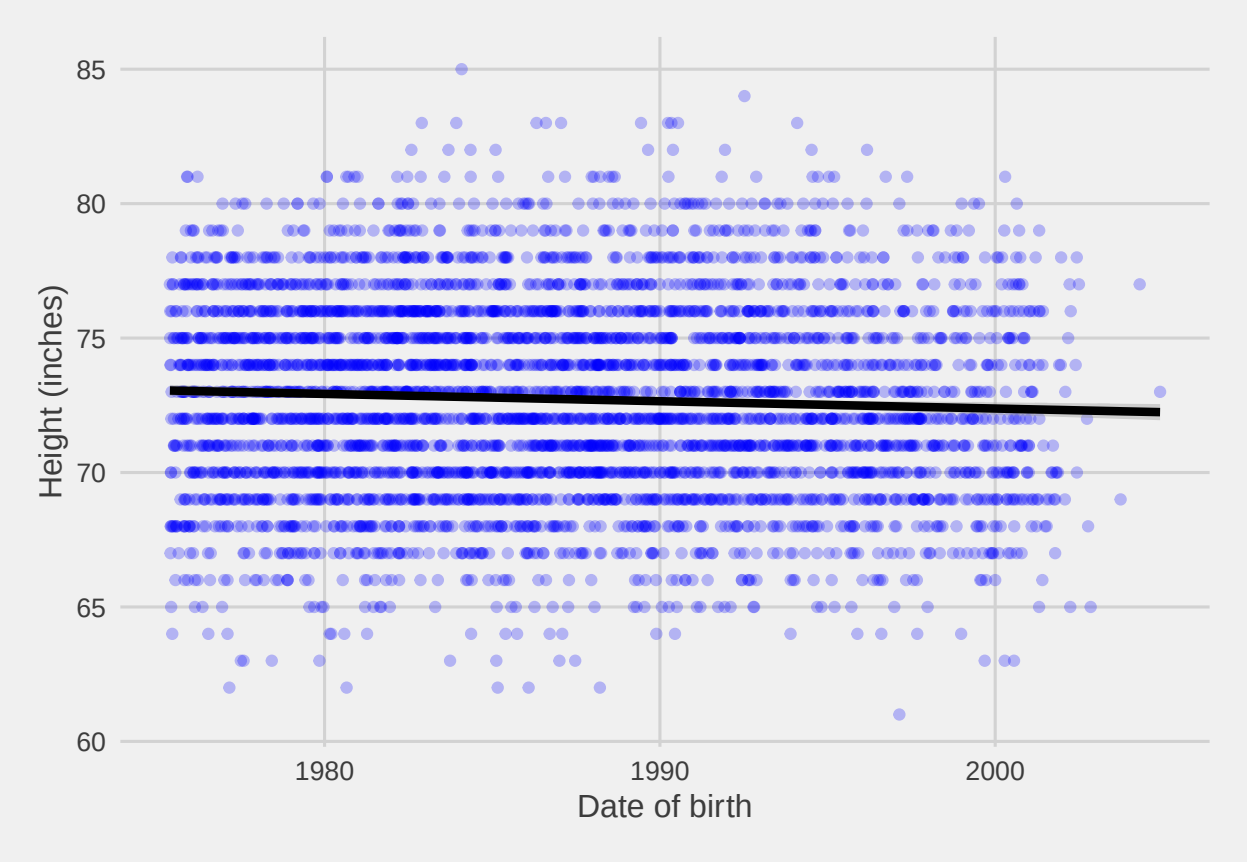
```
## Saving 6.5 x 4.5 in image
```

```
ggsave("vb_heights_birthyear_gender.png", width = 8, height = 4.5)
```

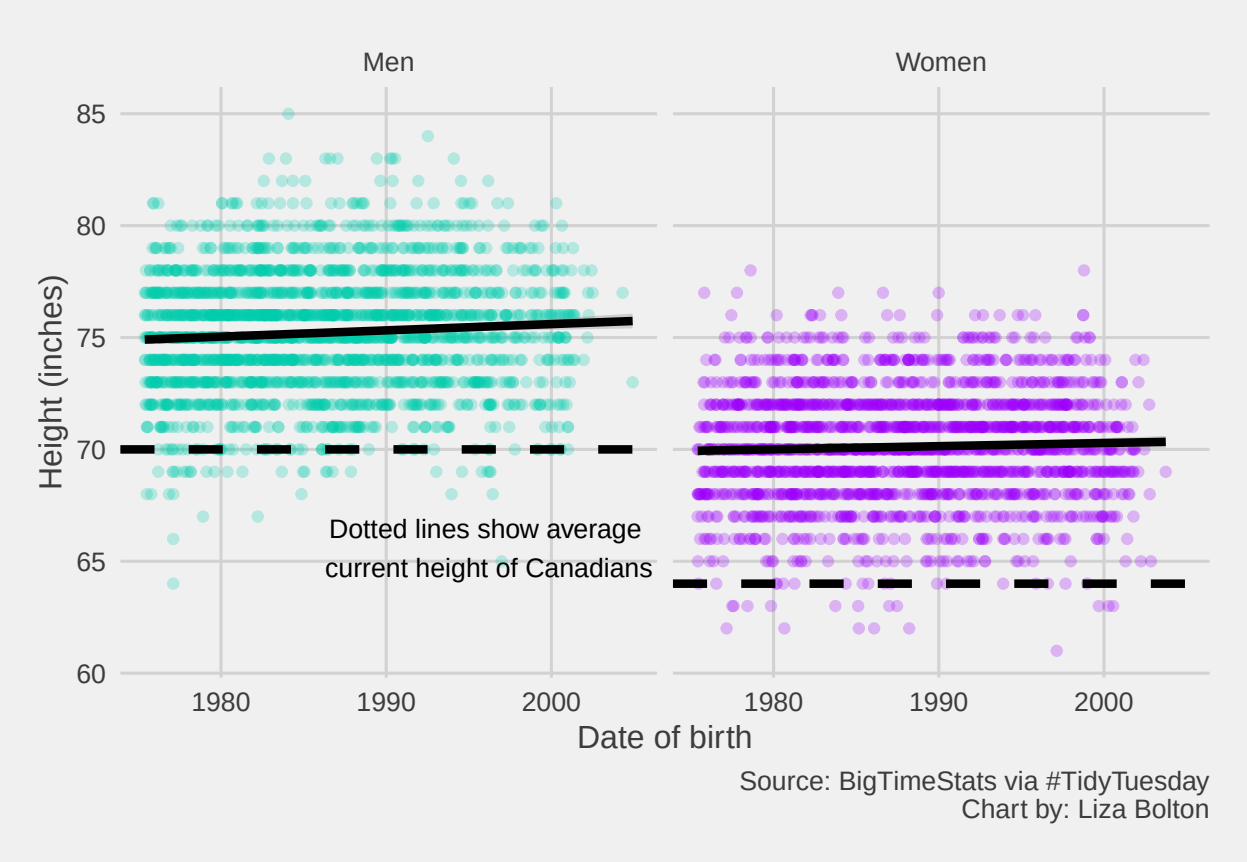
BONUS BONUS!: Cowplot

```
no_gender <- winrate_filter %>%
  ggplot(aes(x = birthdate, y = hgt)) +
  geom_point(alpha = 0.25, color = "blue") +
  geom_smooth(method="lm", formula = y ~ x, colour = "black", size = 1.5) +
  #labs(title = "Heights by date of birth for beach volleyball players",
  #        subtitle = "Restricted to competitors in the FIVB and AVP tournaments and born since 1960") +
  theme_fivethirtyeight() +
  theme(legend.position = "none", axis.title = element_text()) +
  xlab("Date of birth") +
```

```
  ylab("Height (inches)")

no_gender
```




```
base_plot_2 <- winrate_filter %>%
  ggplot(aes(x = birthdate, y = hgt, color = gender)) +
  geom_point(alpha = 0.25) +
  geom_smooth(method="lm", formula = y ~ x, colour = "black", size = 1.5) +
  facet_wrap(~gender) +
  geom_hline(aes(yintercept = hgt), can_pop, color = "black", size = 1.5, lty = "dashed") +
  labs(caption = "Source: BigTimeStats via #TidyTuesday\n Chart by: Liza Bolton") +
  theme_fivethirtyeight() +
  theme(legend.position = "none", axis.title = element_text()) +
  facet_grid(~gender, labeller = labeller(gender = gender.labs)) +
  xlab("Date of birth") +
  ylab("Height (inches)") +
  scale_color_manual(values = rev(suffrage_cols)) +
  geom_text(data = anno, aes(x = x1,  y = y1, label = "Dotted lines show average\n current height of Ca


base_plot_2
```




```
blank <- ggplot() +
  theme_fivethirtyeight()

library(cowplot)
```

```
##
## ************************************************************

## Note: As of version 1.0.0, cowplot does not change the

##   default ggplot2 theme anymore. To recover the previous

##   behavior, execute:
##   theme_set(theme_cowplot())

## ************************************************************

##
## Attaching package: 'cowplot'

## The following object is masked from 'package:ggthemes':
##
##     theme_map
```

```
## The following object is masked from 'package:lubridate':
##
##     stamp
```

```
title <- ggdraw(blank) +
  draw_label(
    "   Are volley ball players getting shorter? Simpson's Paradox on the beach.",
    fontface = 'bold',
    x = 0,
    hjust = 0
  )


step1 <- cowplot::plot_grid(no_gender, blank)
final <- cowplot::plot_grid(title, step1, base_plot_2, rel_widths = c(1,1,2), nrow=3, rel_heights = c(1


save_plot("combo_vb_plot.png", final, base_width = 10, base_height = 8)
```

10. Share the plot!

Share the plot and link to your commented code with all your working in #portfolio-building with a 1–2 sentence explanation by the end of Tuesday May 19 (bonus if you share it on Twitter with #TidyTuesday). Our ISSC Tweeps are on this list. Message me if you want to be added!

Thanks everyone!

Please make sure you fill out the weekly check-in by Thursday at 11:30 pm ET.