Slow ggplot2

Evangeline Reynolds

2018-11-03

Contents

1 Introduction **5**

2 MakeoverMonday is now in it's third year. **7**

3 Baseball, WAR, and Ethnicity **9**
3.1 . . . 11
3.2 . . . 12
3.3 . . . 13
3.4 . . . 14
3.5 . . . 15
3.6 . . . 16
3.7 . . . 17
3.8 . . . 18
3.9 . . . 19
3.10 . . . 20

4 Christmas Trees **23**
4.1 . . . 25
4.2 . . . 26
4.3 . . . 27
4.4 . . . 28
4.5 . . . 29
4.6 . . . 30
4.7 . . . 31
4.8 . . . 32
4.9 . . . 33
4.10 . . . 34
4.11 . . . 35

5 Officials' beliefs about women's representation **39**

6 Maternal Leave **41**

7 Traits **43**

8 Salarys of Trump and Obama White House Employees **45**

9 Winter Games **47**

10 Brexit **51**

11 Curry in London **55**

12 Life Expectancy Increases **59**

13 Myers Briggs **61**

14 Wine **63**

15 Arctic Ice **65**

Chapter 1

Introduction

Outline introduction:

- What is ggplot
- What is "slow" ggplotting
- What does this book contain (Makeover monday examples with slow ggplot)

Chapter 2

MakeoverMonday is now in it's third year.

January 2016, Andy Kreibel, Head Coach at The Information Lab, and Andy Cotgreave, Technical Evangelist at Tableau, have been organizing Make

The friends wanted to keep up their skills even as they had moved into more administrative jobs that didn't require regular visualizaton work. Andy and Andy were find the data for an existing data visualization in the media, and would recreate or "makeover" the visualization using Tableau, a tool that they were both expert in. They were sharing their makeovers products with each other, but also with the world on Twitter. When more people expressed interest in joining, and they two started a more organized initiative - posting the original graph and data every Sunday, so that whoever wanted to could participate in #Makeovermonday.

My first submission was late 2016, after catching wind of the exciting project via the podcast. I made a scrappy little graph about motorway casualties; sad topic, but fun graph making.

I was using base R at that time. Then in the summer of 2017 I went to a conference in Zurich, the women's summer school for political methodology. There was a session on ggplot2. I internalized some of the basics, and decided that if I wanted to learn that (powerful - as everyone kept calling it) graphing system, then I could do it via the #MakeoverMonday weekly exercises (not that I particpated weekly). Even though most folks were using Tablaeu, the administrators didn't seem to mind a few R and ggplot submission here and there. I got a little hooked.

Early this year Andy and Eva Murry sent a number of the participants a private message on Twitter. "We're writing a book: #MakeoverMonday". They were putting together a collection of a visualizations that resulted from the project, and were seeking perspectives of participants as well as permission to use some of the visualizations produced for the initiatives. Cool. I was pleased to participate. For me #MakeoverMonday allowed me to focus on the visualization task. Usually visualization comes at the end of, sometimes arguous, data cleaning — and you might already be a little spent. Having rather clean data delivered, and seeing the approaches of many other (many brilliant) data visualizers was a treat. I still need to buy my copy of the book, which contains a visualization of food prices in London as a function of how far a restaurant is from the Big Ben.

And now, using the magic of RStudio and Yihui Xie's bookdown, I'm putting together my own little collection. Of course there is a bit of curation involved — I'm not including every plot. And, I'm revising the exact code that creates the plots in many cases, to be more consistent across plots, and also, I think, to make communicating about how the plot was built easy. This involves:

- pulling out aes() from the ggplot() function
- using fewer functions; example - using labs() to add a title instead of ggtitle()
- using functions multiple times; example aes(x = var1) + aes(y = var2) rather than aes(x = var1, y = var2)

- using base R functions and tidyverse functions. For other packages, the :: style to call them
- write out arguments (no shortcuts) aes(x = gdppercap) not aes(gdppercap)
- order ggplot commands so that reactivity is obvious; scale adjustments to aesthetics might also be near the aesthetic declaration.

Chapter 3

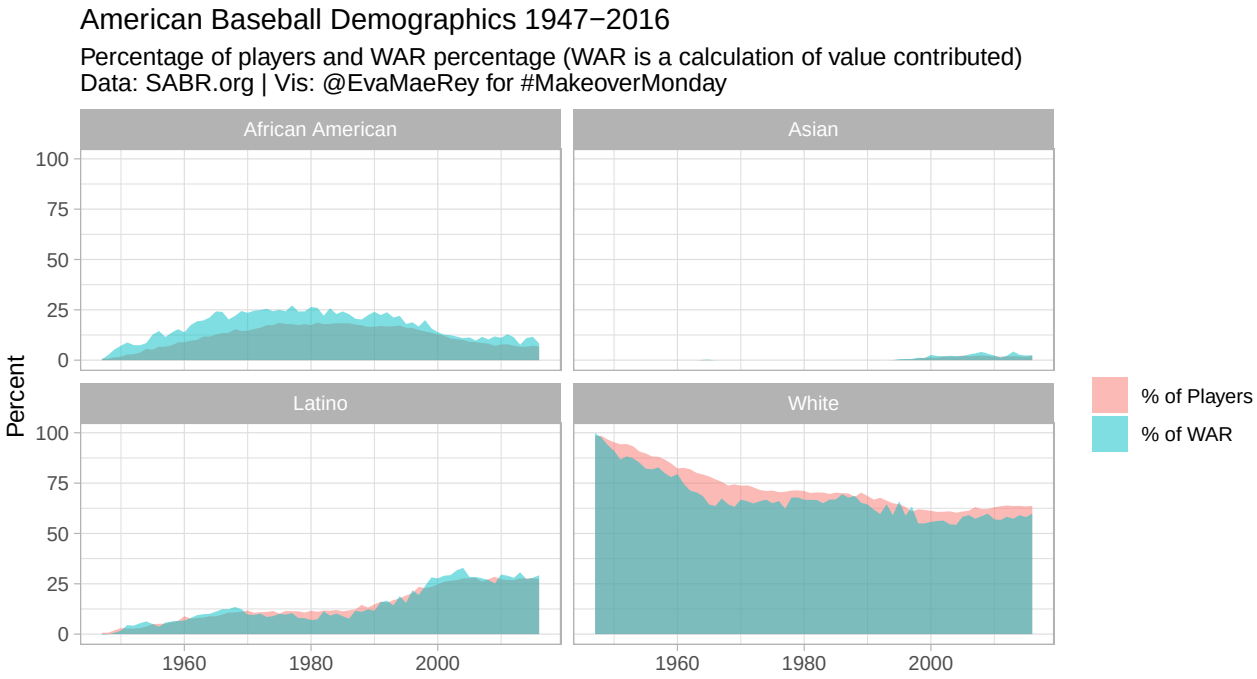
Baseball, WAR, and Ethnicity

This data visualization uses the WAR measure in baseball, a calculation based on the contributions of players. The visualizations show that new ethnicities and races started to be included in Major League baseball, the minority players that joined tended to contribute more than the expected value for players overall. For example, from 1947, when Jackie Robinson joined Major League baseball, and onward, the percent of African American players was outpaced by the percent calculated contributions (WAR) of African American players.

A random sample from the data set:

Year	Ethnicity	type	Percent
2005	African American	% of WAR	11.3
1985	Asian	% of Players	0.0
1977	Asian	% of Players	0.0
1960	Latino	% of Players	8.9
1968	African American	% of Players	15.4

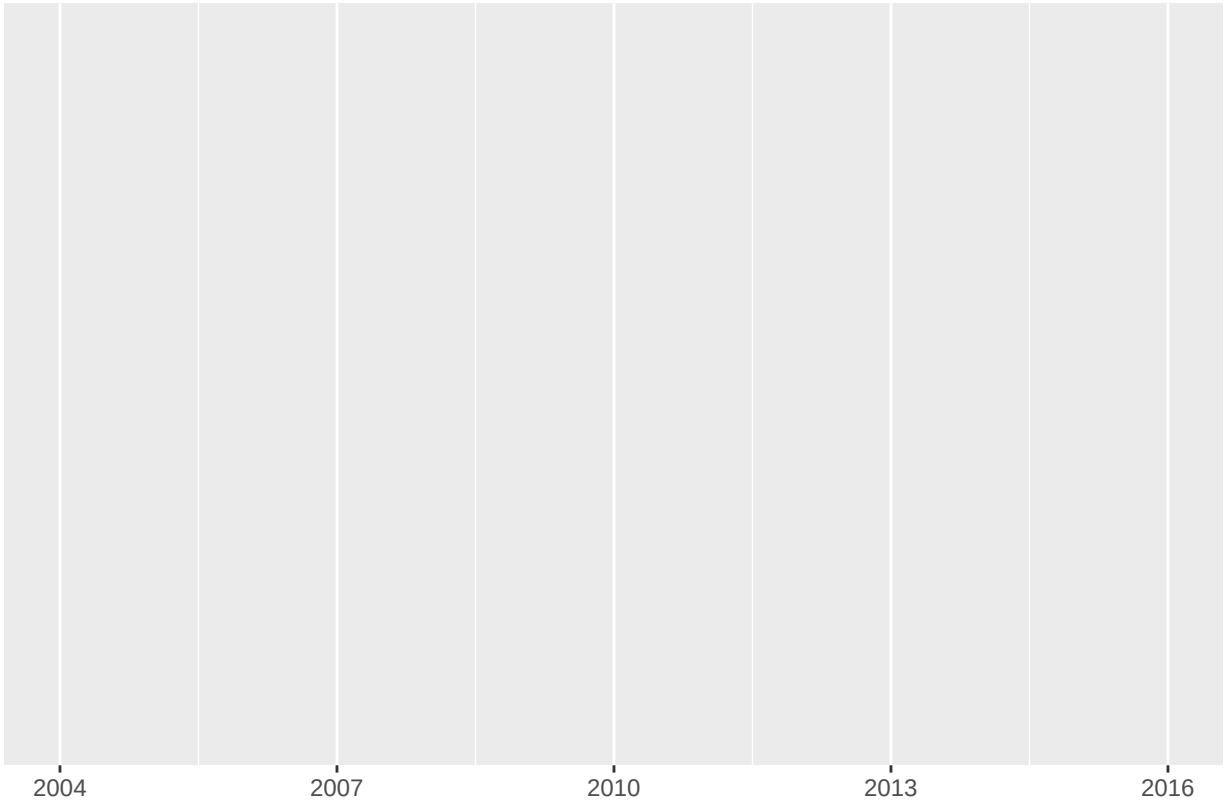
```
ggplot(df_gather) +
  aes(x = Year) +
  aes(y = Percent) +
  aes(fill = type) +
  facet_wrap(~ Ethnicity) +
  geom_area(alpha = .5, position = "dodge") +
  labs(fill = "") +
  labs(x = "") +
  labs(title = "American Baseball Demographics 1947-2016") +
  labs(subtitle = "Percentage of players and WAR percentage (WAR is a calculation of value contributed)
  theme_light()
```


American Baseball Demographics 1947−2016
Percentage of players and WAR percentage (WAR is a calculation of value contributed)
Data: SABR.org | Vis: @EvaMaeRey for #MakeoverMonday
African American
Asian
Latino
White
Percent
100
75
50
25
0
1960
1980
2000
% of Players
% of WAR

```
ggplot(df_gather)
```

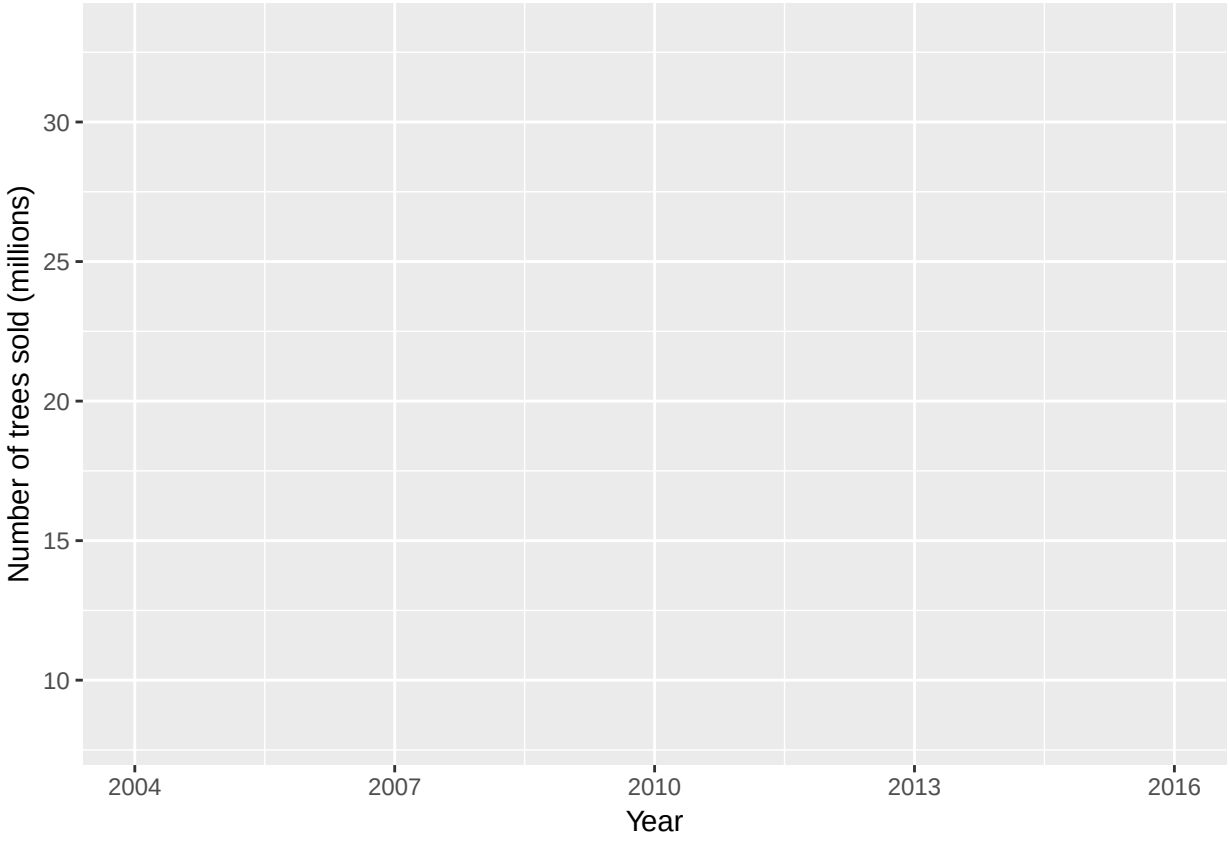

3.1

```
ggplot(df_gather) +
  aes(x = Year)
```



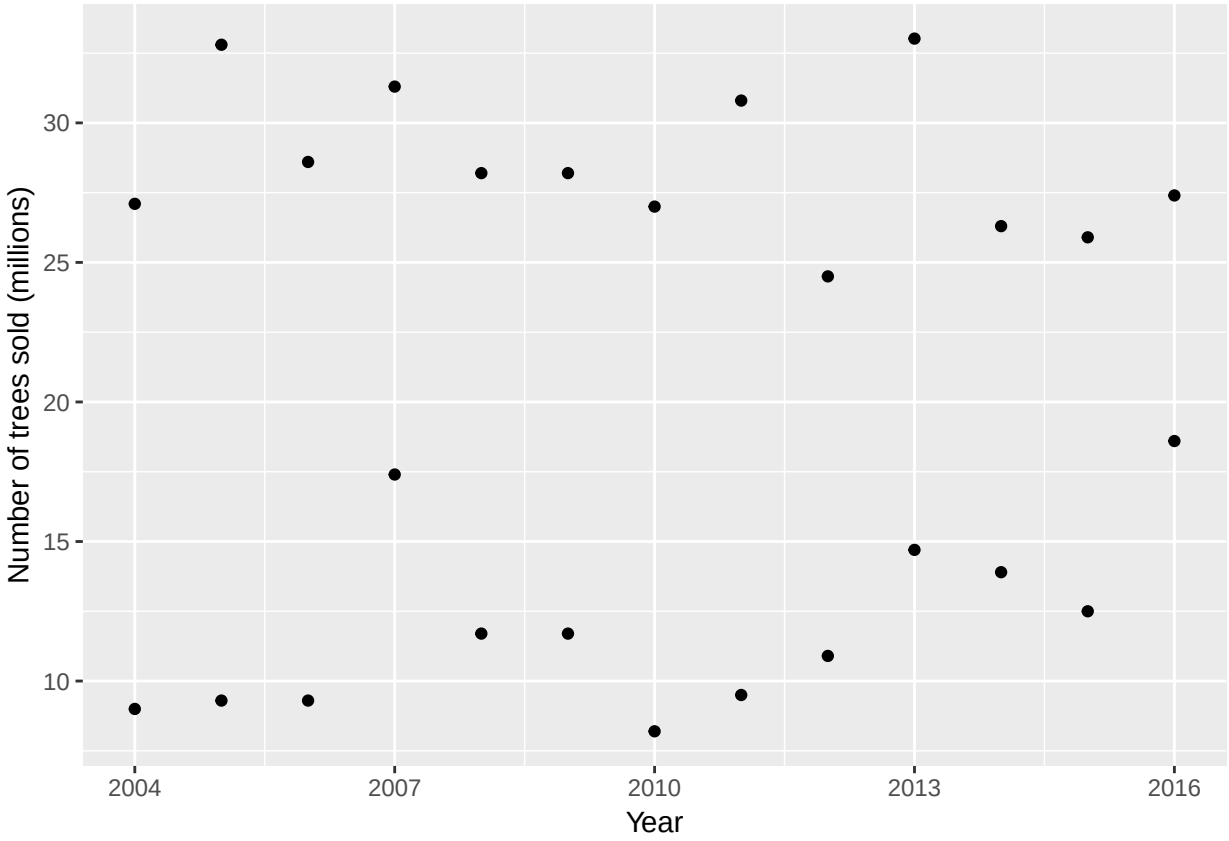
3.2

```
ggplot(df_gather) +
  aes(x = Year) +
  aes(y = Percent)
```



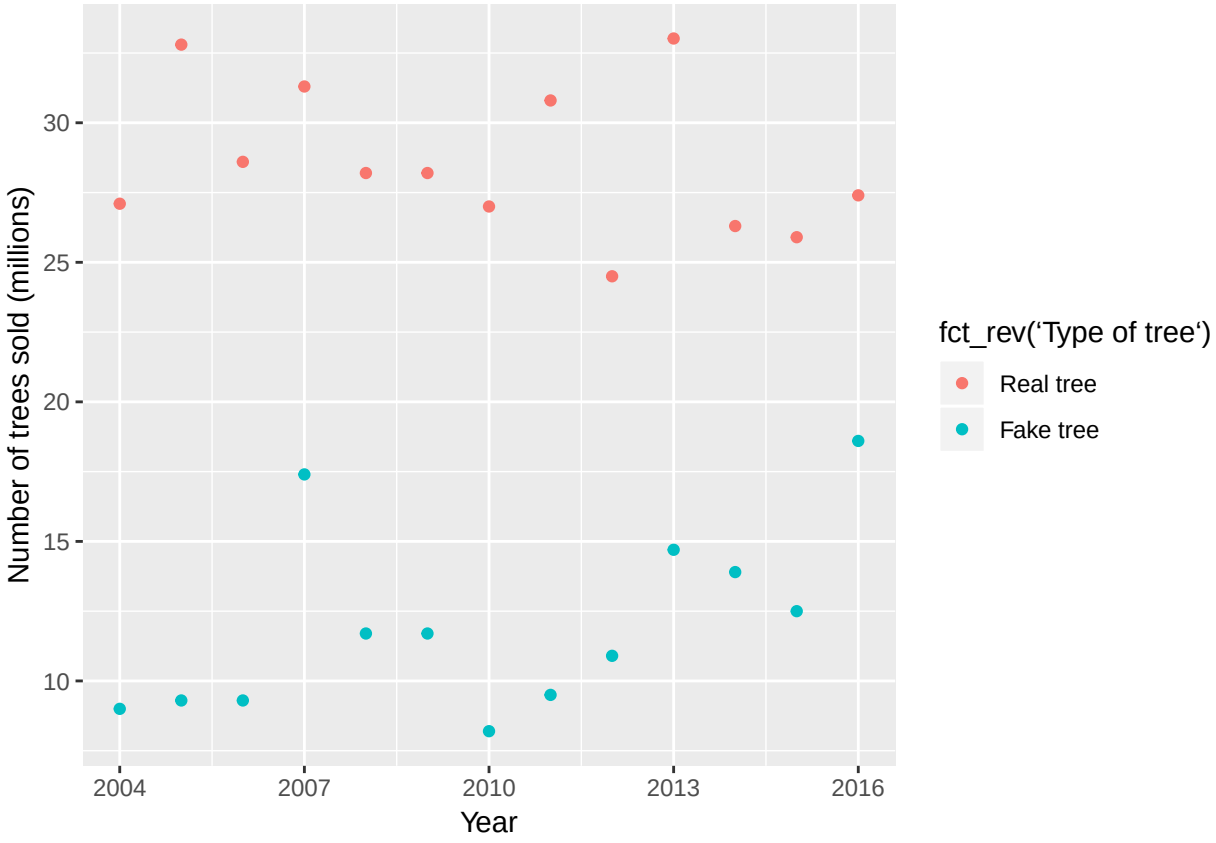
3.3

```
ggplot(df_gather) +
  aes(x = Year) +
  aes(y = Percent) +
  aes(fill = type)
```



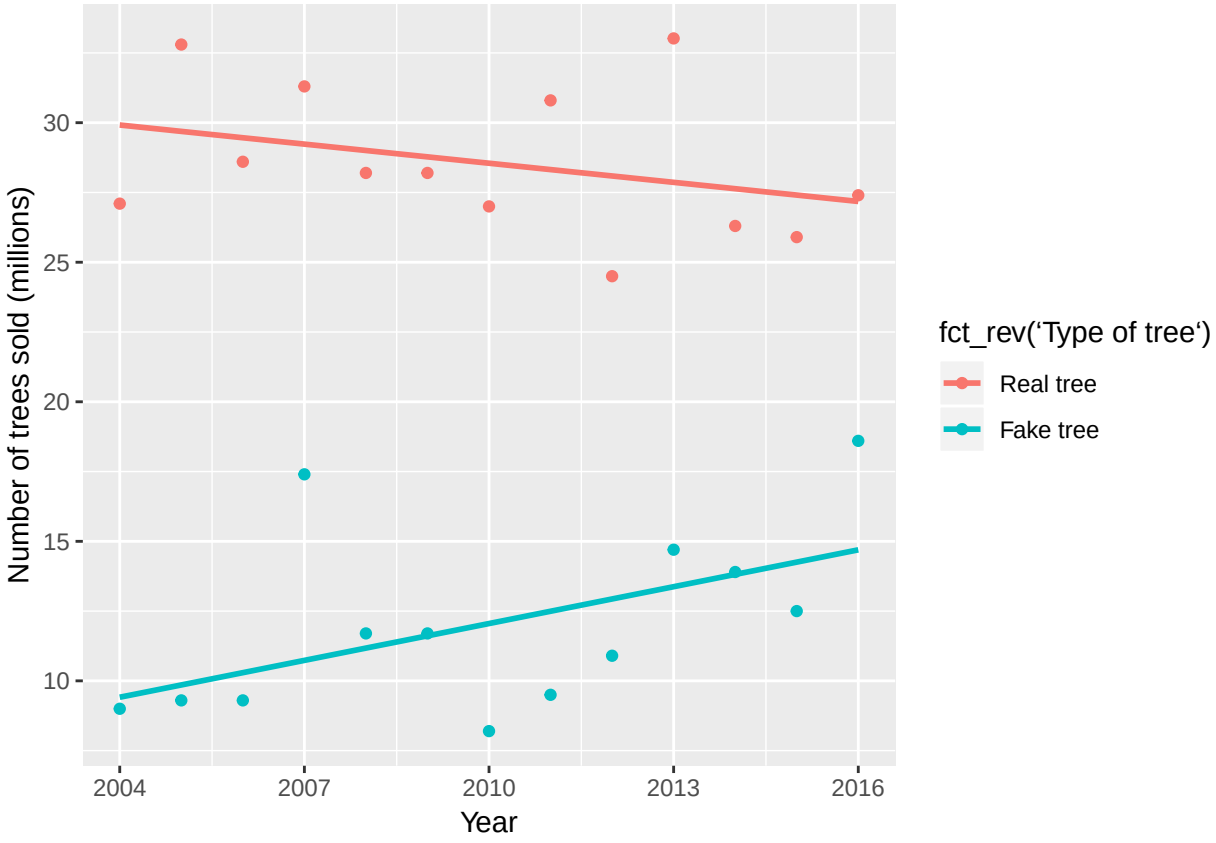
3.4

```
ggplot(df_gather) +
  aes(x = Year) +
  aes(y = Percent) +
  aes(fill = type) +
  facet_wrap(~ Ethnicity)
```



3.5

```
ggplot(df_gather) +
  aes(x = Year) +
  aes(y = Percent) +
  aes(fill = type) +
  facet_wrap(~ Ethnicity) +
  geom_area(alpha = .5, position = "dodge")
```

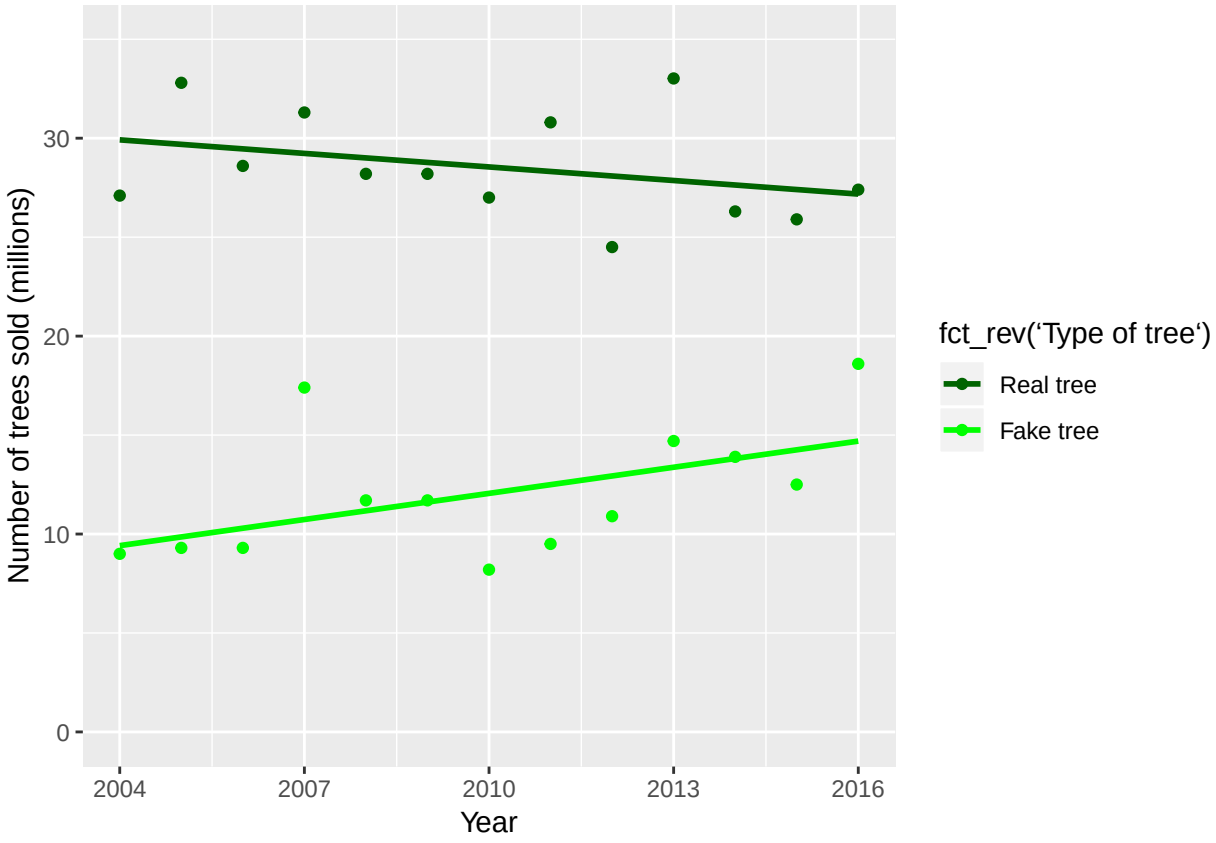


3.6

```
ggplot(df_gather) +
  aes(x = Year) +
  aes(y = Percent) +
  aes(fill = type) +
  facet_wrap(~ Ethnicity) +
  geom_area(alpha = .5, position = "dodge") +
  labs(fill = "")
```

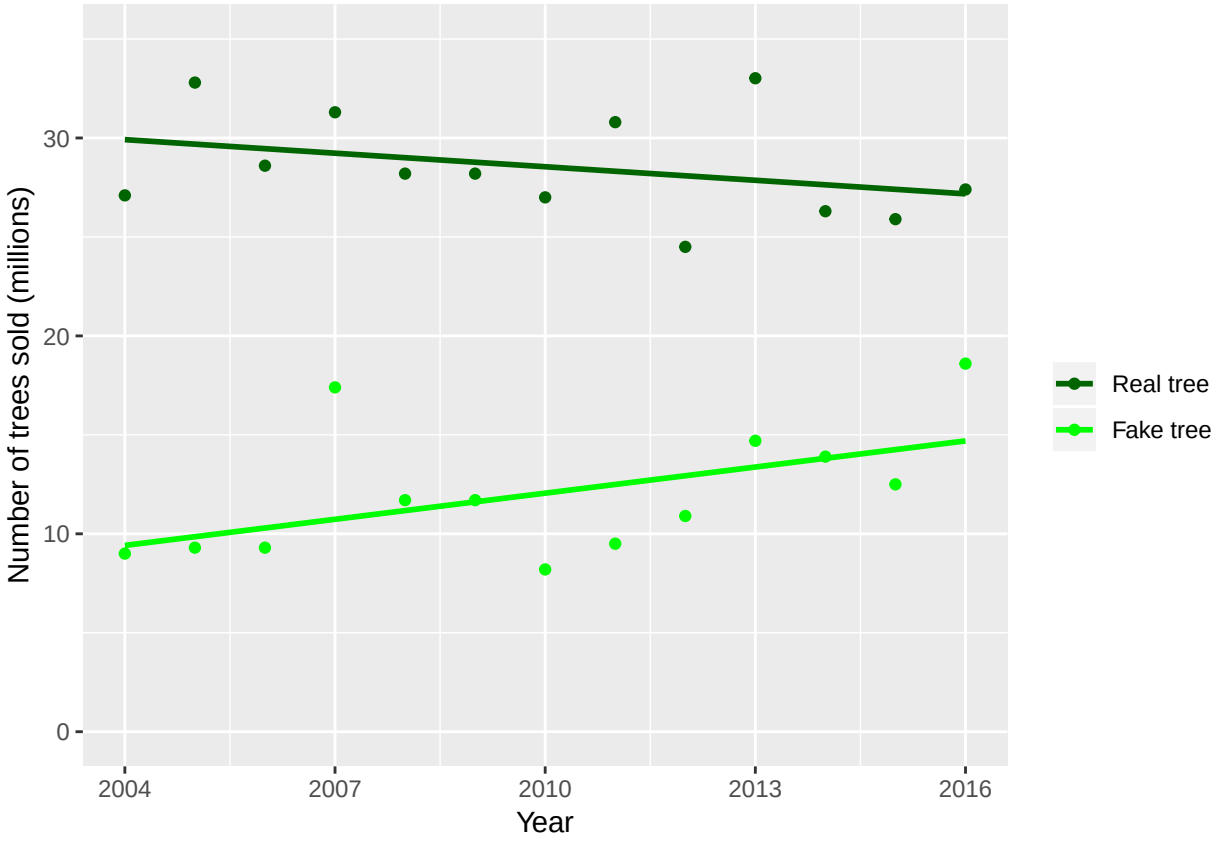
3.7

```
ggplot(df_gather) +
  aes(x = Year) +
  aes(y = Percent) +
  aes(fill = type) +
  facet_wrap(~ Ethnicity) +
  geom_area(alpha = .5, position = "dodge") +
  labs(fill = "") +
  labs(x = "")
```



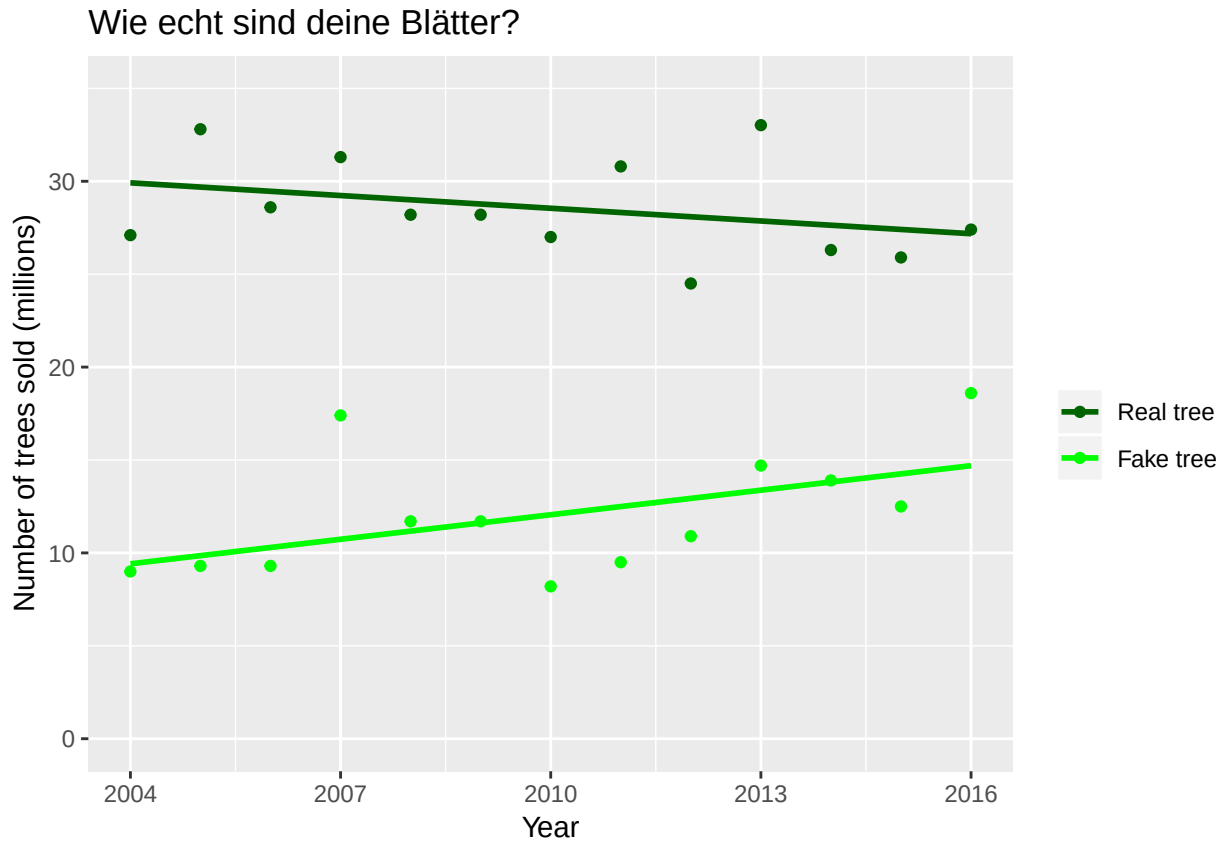
3.8

```
ggplot(df_gather) +
  aes(x = Year) +
  aes(y = Percent) +
  aes(fill = type) +
  facet_wrap(~ Ethnicity) +
  geom_area(alpha = .5, position = "dodge") +
  labs(fill = "") +
  labs(x = "") +
  labs(title = "American Baseball Demographics 1947-2016")
```



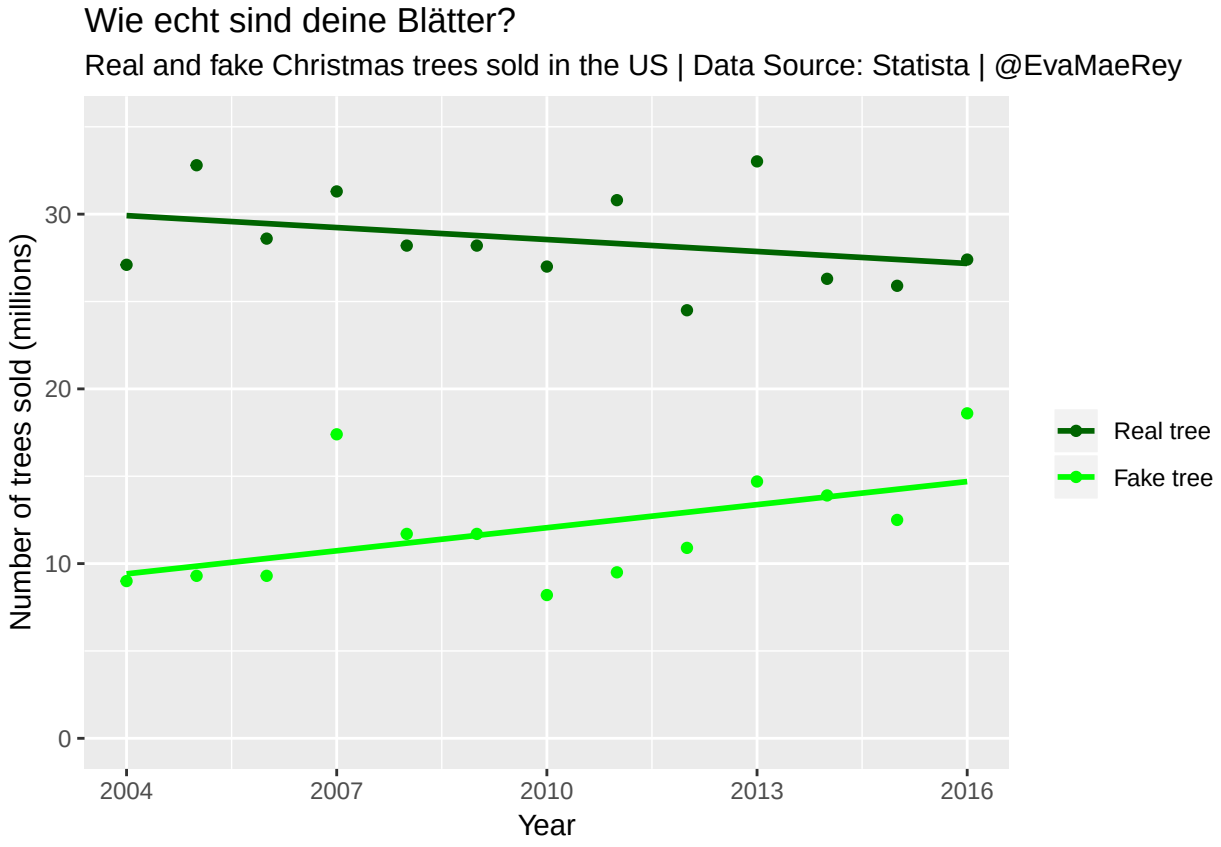
3.9

```
ggplot(df_gather) +
  aes(x = Year) +
  aes(y = Percent) +
  aes(fill = type) +
  facet_wrap(~ Ethnicity) +
  geom_area(alpha = .5, position = "dodge") +
  labs(fill = "") +
  labs(x = "") +
  labs(title = "American Baseball Demographics 1947-2016") +
  labs(subtitle = "Percentage of players and WAR percentage (WAR is a calculation of value contributed)"
```



3.10

```
ggplot(df_gather) +
  aes(x = Year) +
  aes(y = Percent) +
  aes(fill = type) +
  facet_wrap(~ Ethnicity) +
  geom_area(alpha = .5, position = "dodge") +
  labs(fill = "") +
  labs(x = "") +
  labs(title = "American Baseball Demographics 1947-2016") +
  labs(subtitle = "Percentage of players and WAR percentage (WAR is a calculation of value contributed)
  theme_light()
```



Chapter 4

Christmas Trees

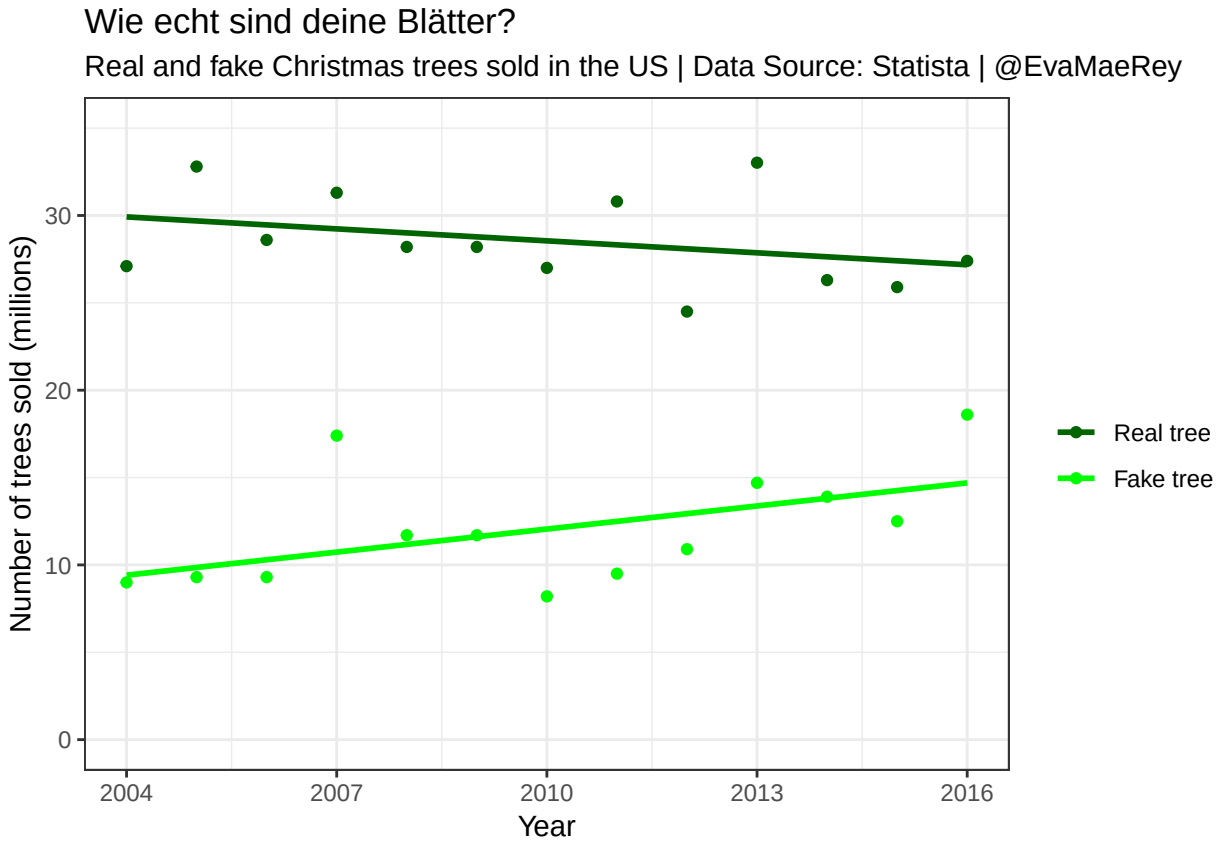
Here is a simple plot of Christmas Tree Sales in the U.S. The plot shows that artificial tree sales are on the rise, contrasting with declines in real trees. The title plays on the German Christmas Carol "O Tannenbaum", "Oh Christmas Tree" in English. "Wie echt sind deine Blätter?" means "how real are your leaves"; the original text from the carol is "Wie treu sind deine Blätter!" which means "How true your leaves are!"

I also plot the cumulative number of trees purchased of each type, artificial and real, from 2004 to 2014, comparing that to the 2016 U.S. population. Almost one real tree per person was bought over the course of 10 years!

A random sample from the data set:

Year	Number of trees sold	Type of tree	Number of trees sold (millions)
2016	27400000	Real tree	27.4
2011	9500000	Fake tree	9.5
2011	30800000	Real tree	30.8
2009	28200000	Real tree	28.2
2012	24500000	Real tree	24.5

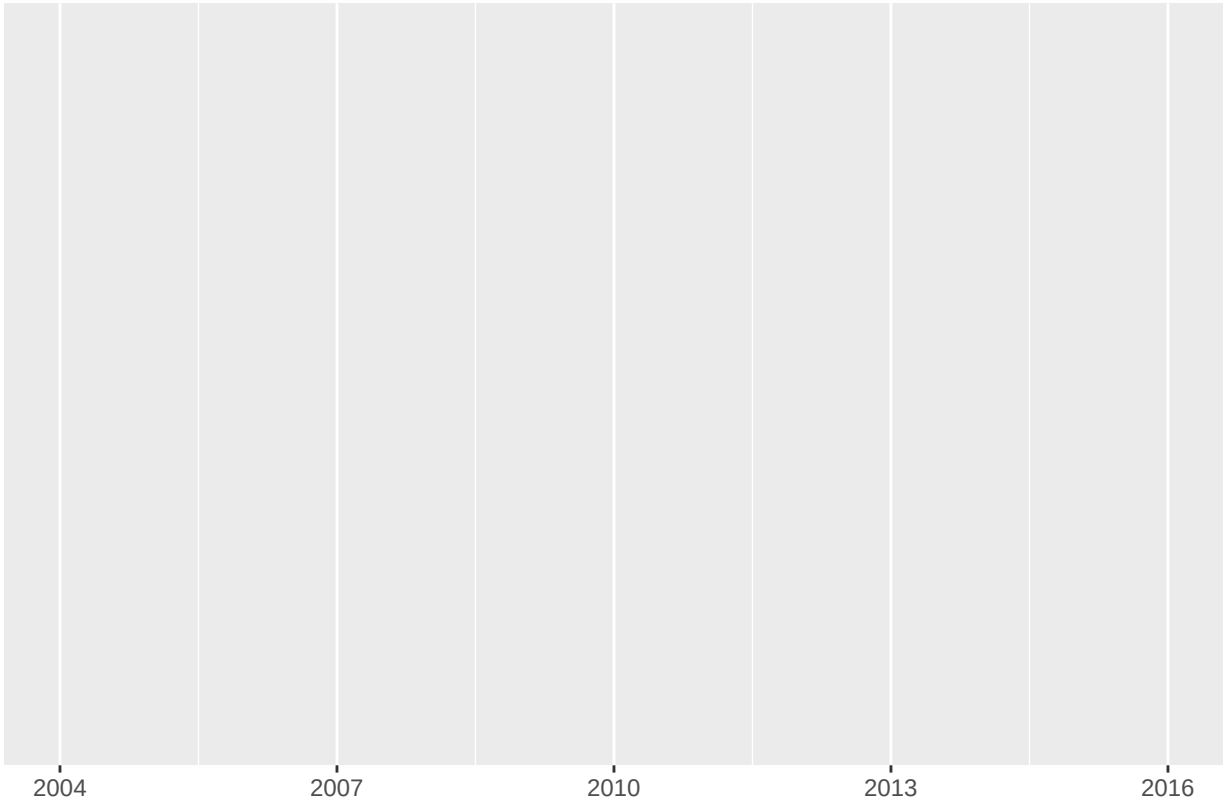
```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`) +
  geom_point() +
  aes(col = fct_rev(`Type of tree`)) +
  geom_smooth(method = "lm", se = F) +
  scale_color_manual(values = c("darkgreen", "green")) +
  ylim(c(0, 35)) +
  labs(col = "") +
  labs(title = "Wie echt sind deine Blätter?") +
  labs(subtitle = "Real and fake Christmas trees sold in the US | Data Source: Statista | @EvaMaeRey ")
  theme_bw()
```


Wie echt sind deine Blätter?
Real and fake Christmas trees sold in the US | Data Source: Statista | @EvaMaeRey
Number of trees sold (millions)
0
10
20
30
2004
2007
2010
2013
2016
Year
Real tree
Fake tree

```
ggplot(data = dta)
```

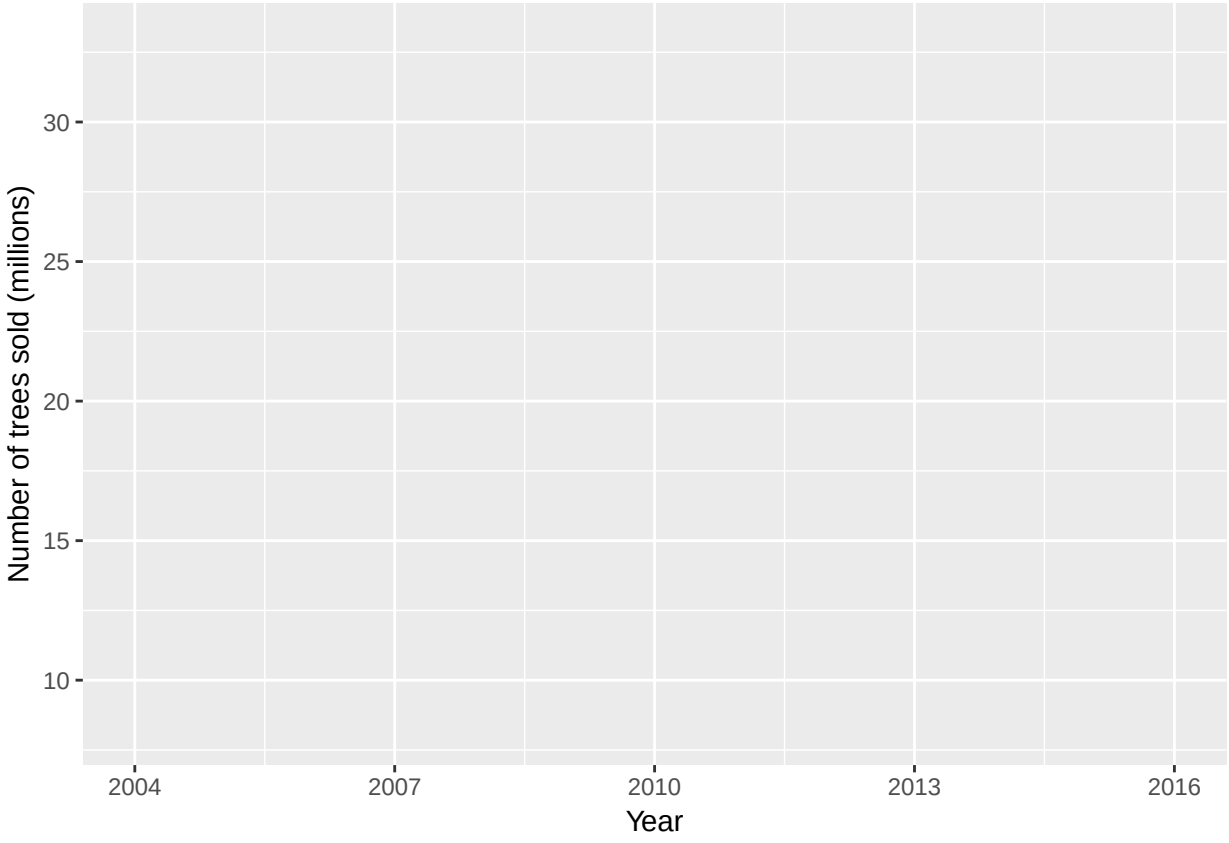

4.1

```
ggplot(data = dta) +
  aes(Year)
```



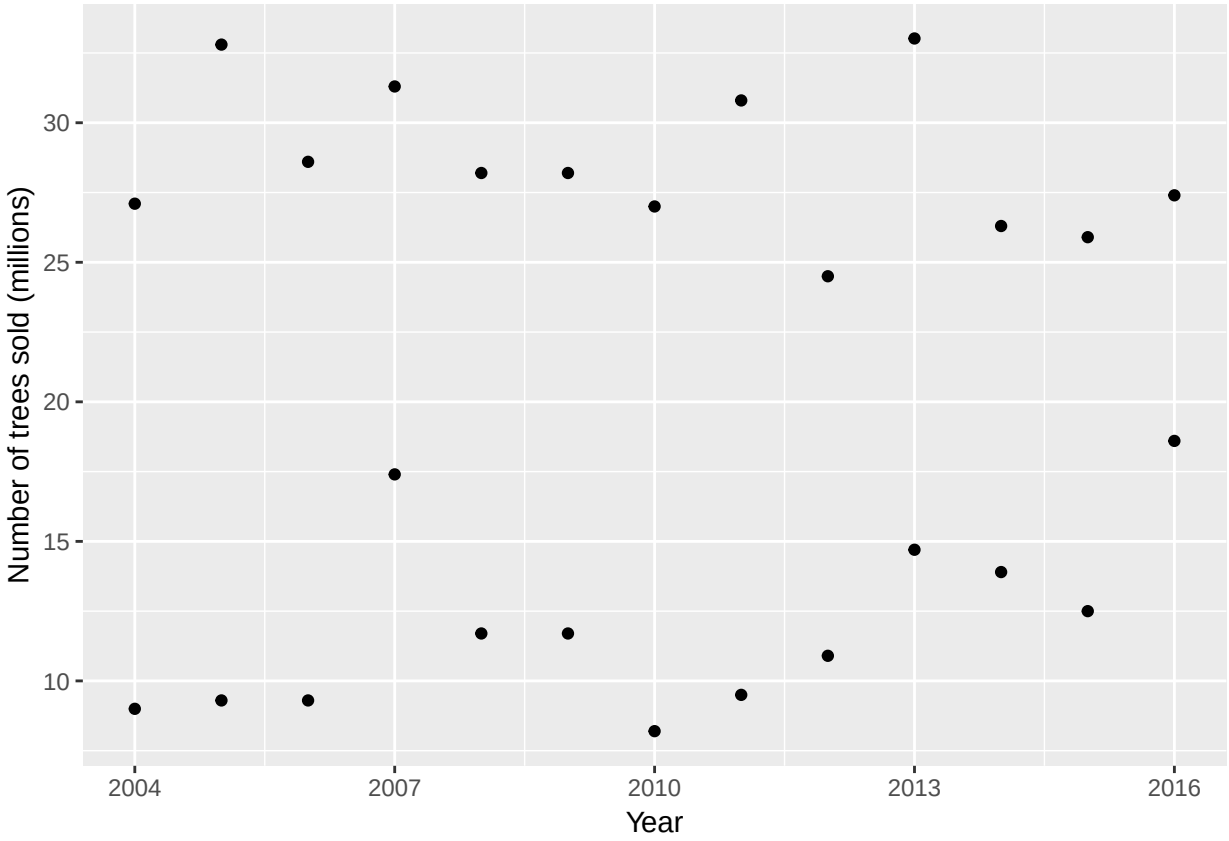
4.2

```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`)
```



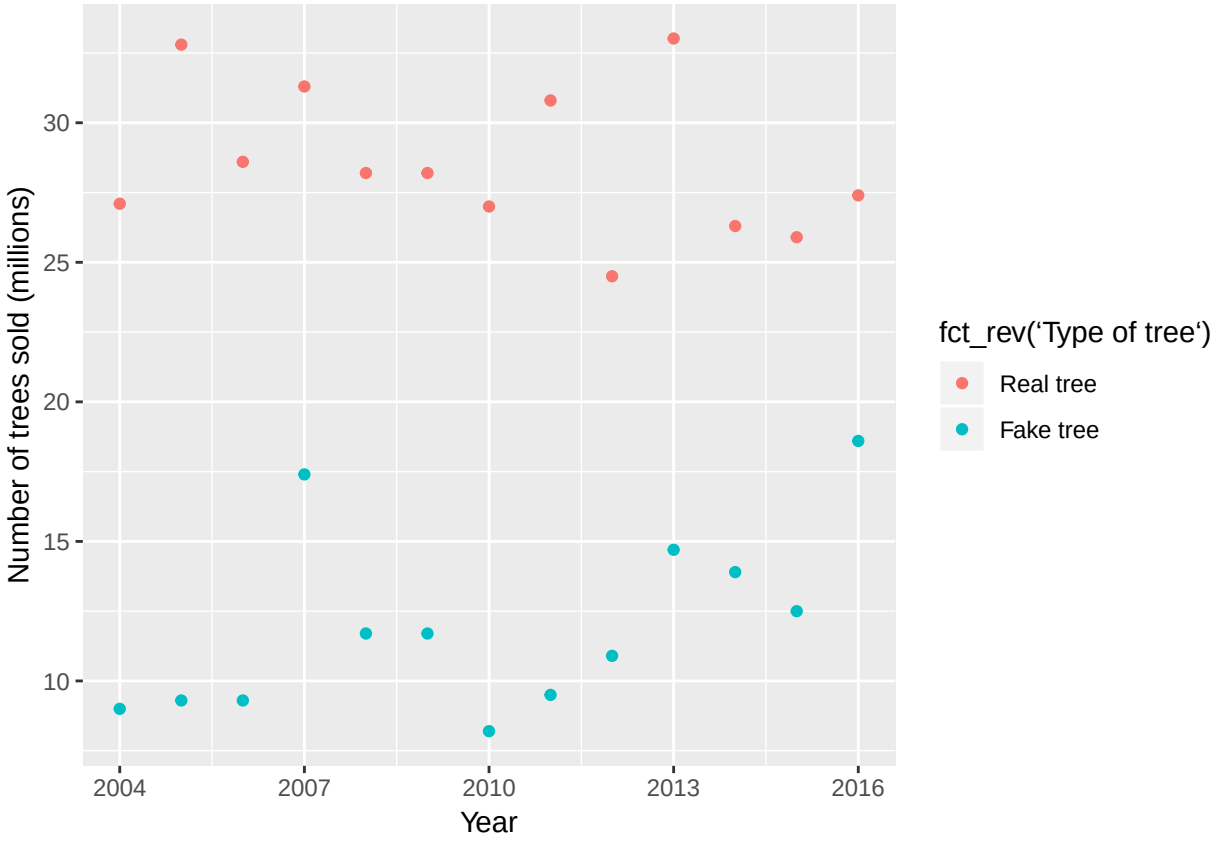
4.3

```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`) +
  geom_point()
```



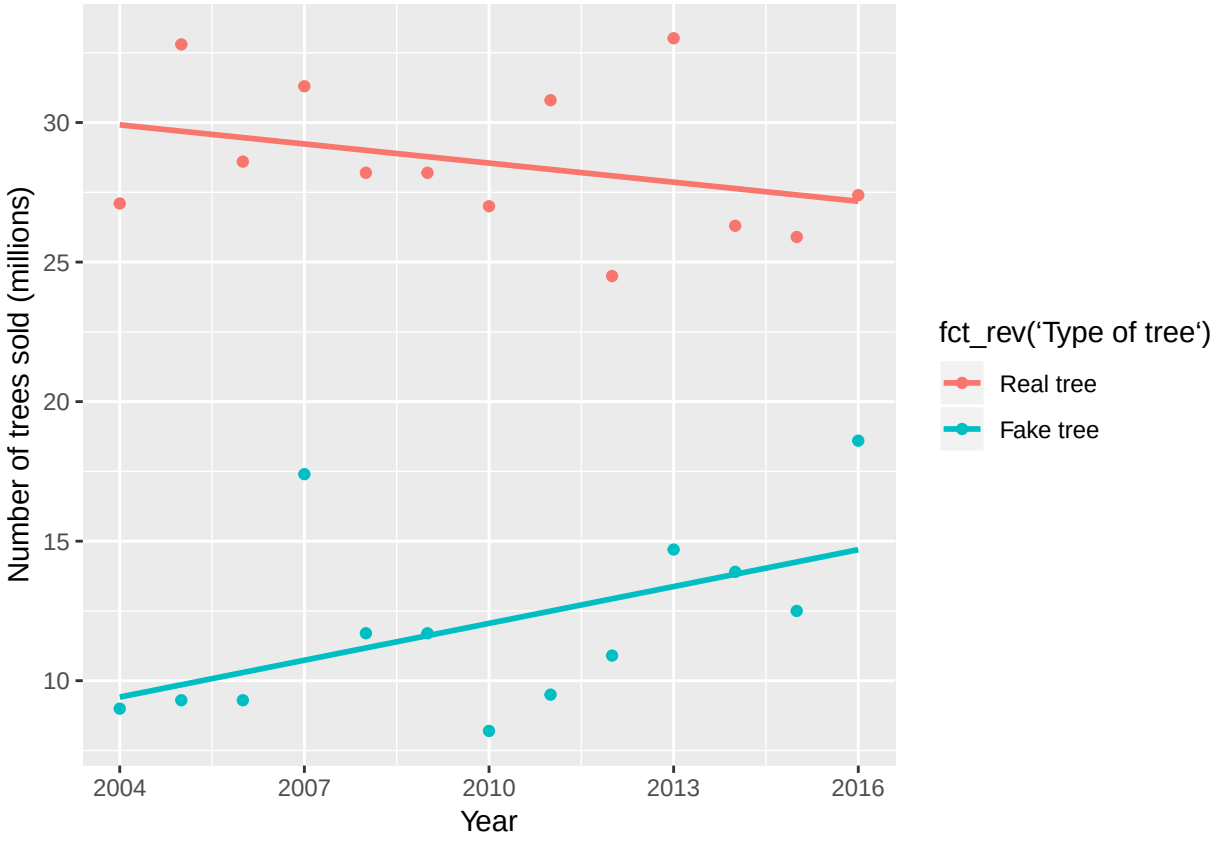
4.4

```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`) +
  geom_point() +
  aes(col = fct_rev(`Type of tree`))
```



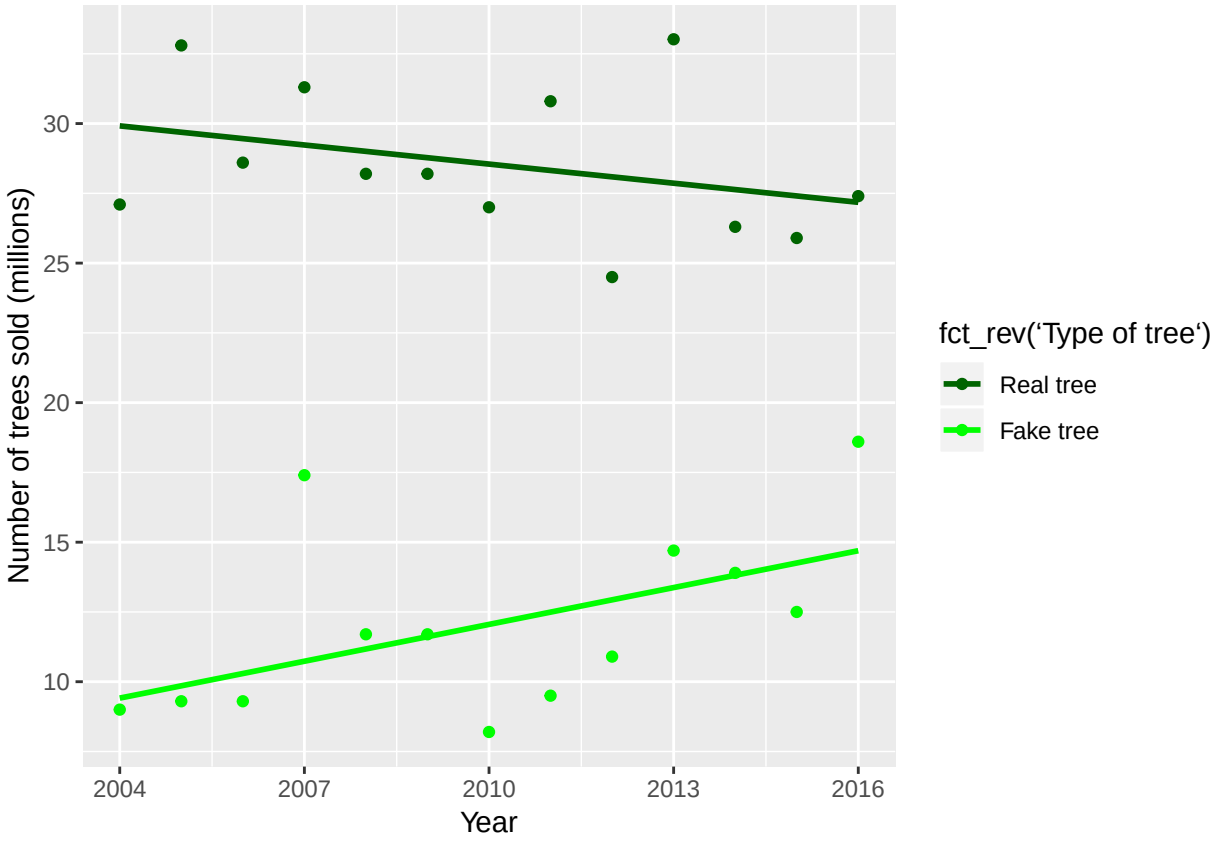
4.5

```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`) +
  geom_point() +
  aes(col = fct_rev(`Type of tree`)) +
  geom_smooth(method = "lm", se = F)
```



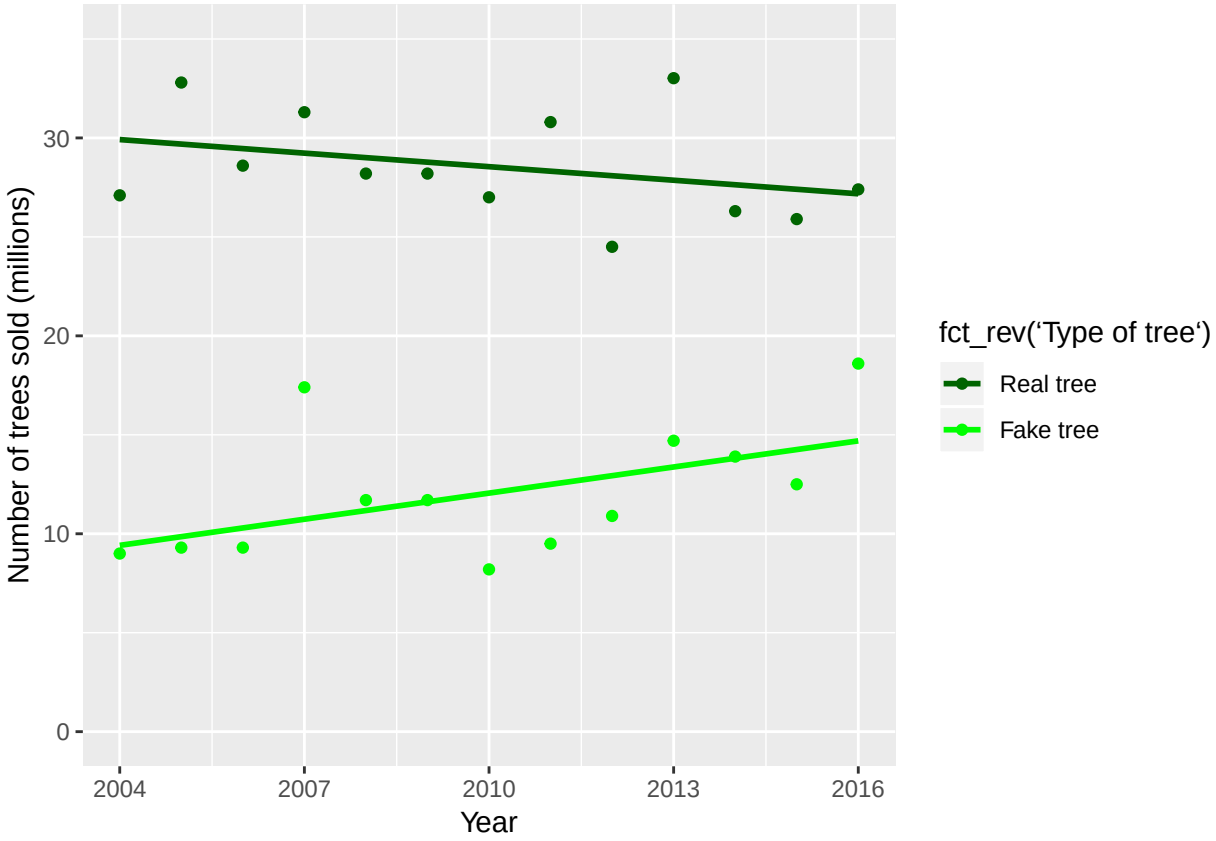
4.6

```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`) +
  geom_point() +
  aes(col = fct_rev(`Type of tree`)) +
  geom_smooth(method = "lm", se = F) +
  scale_color_manual(values = c("darkgreen", "green"))
```



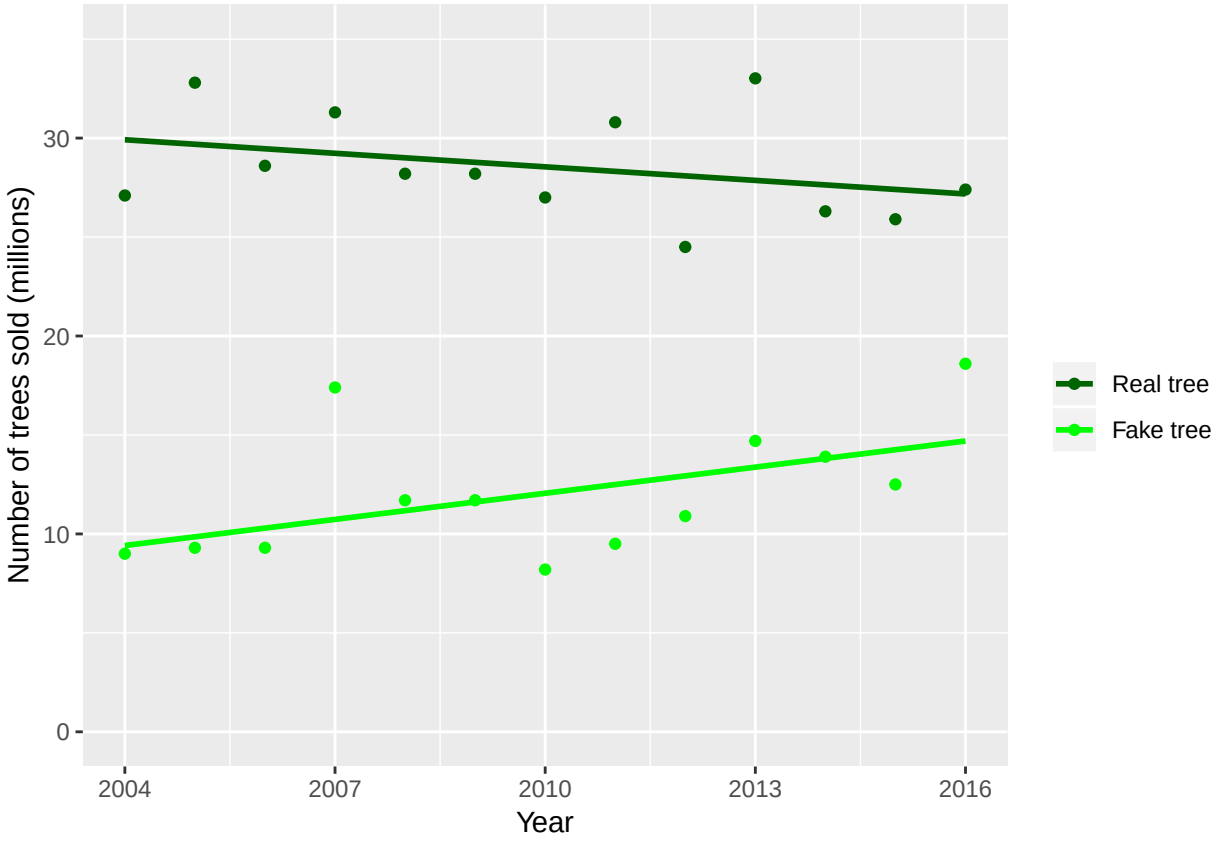
4.7

```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`) +
  geom_point() +
  aes(col = fct_rev(`Type of tree`)) +
  geom_smooth(method = "lm", se = F) +
  scale_color_manual(values = c("darkgreen", "green")) +
  ylim(c(0, 35))
```



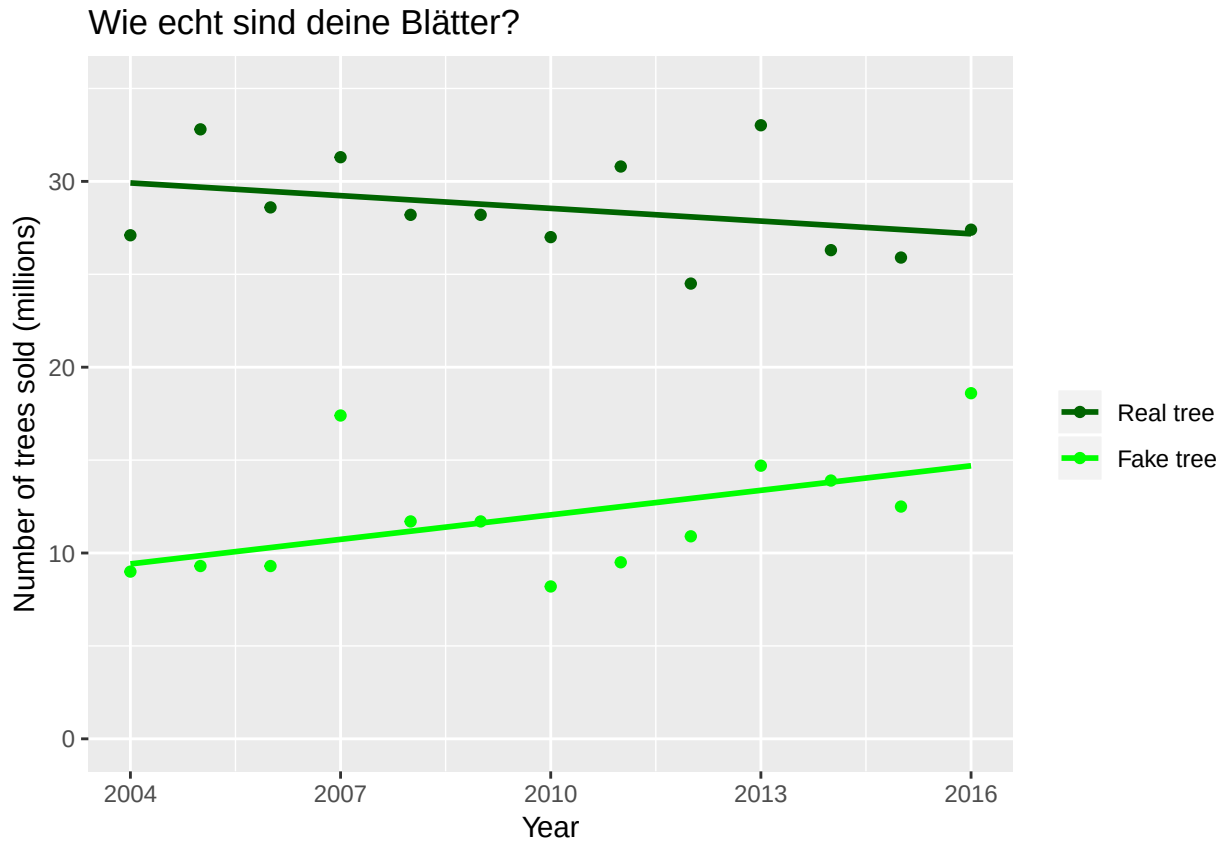
4.8

```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`) +
  geom_point() +
  aes(col = fct_rev(`Type of tree`)) +
  geom_smooth(method = "lm", se = F) +
  scale_color_manual(values = c("darkgreen", "green")) +
  ylim(c(0, 35)) +
  labs(col = "")
```



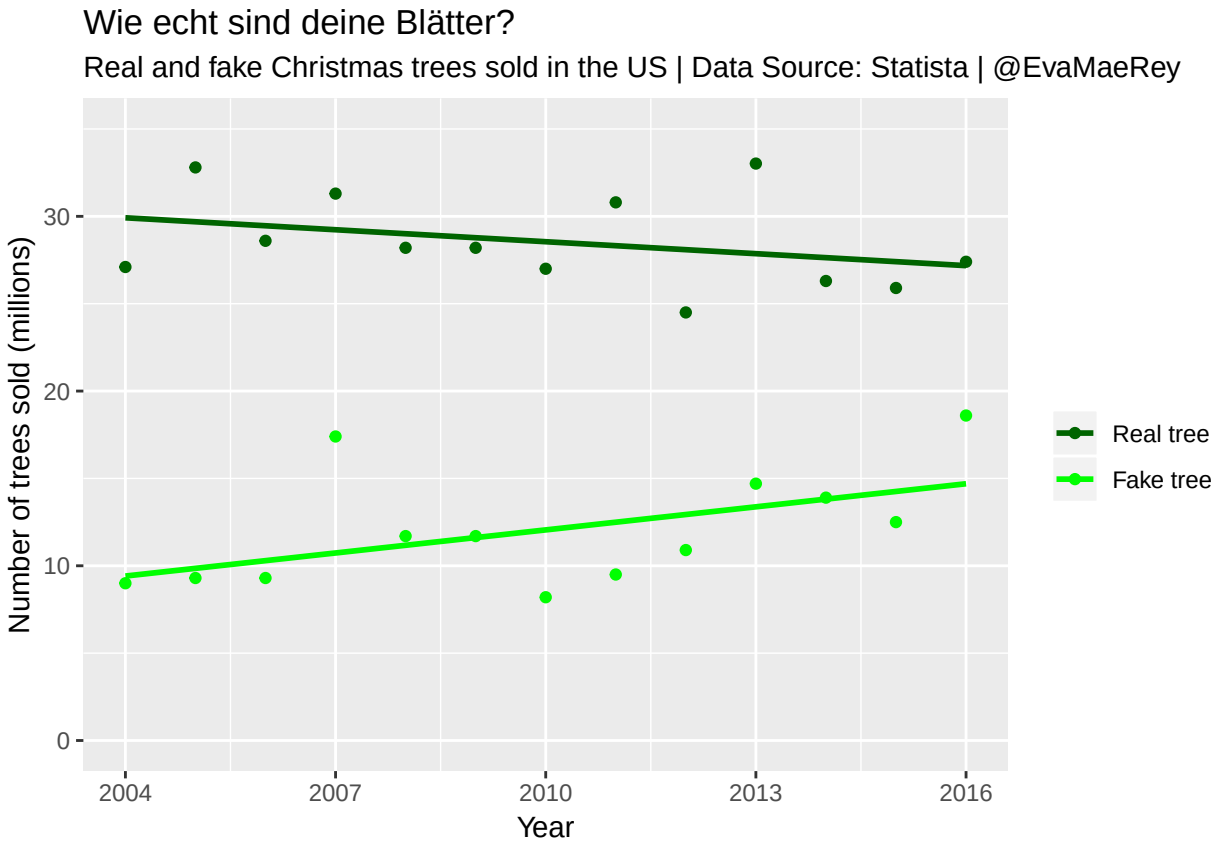
4.9

```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`) +
  geom_point() +
  aes(col = fct_rev(`Type of tree`)) +
  geom_smooth(method = "lm", se = F) +
  scale_color_manual(values = c("darkgreen", "green")) +
  ylim(c(0, 35)) +
  labs(col = "") +
  labs(title = "Wie echt sind deine Blätter?")
```



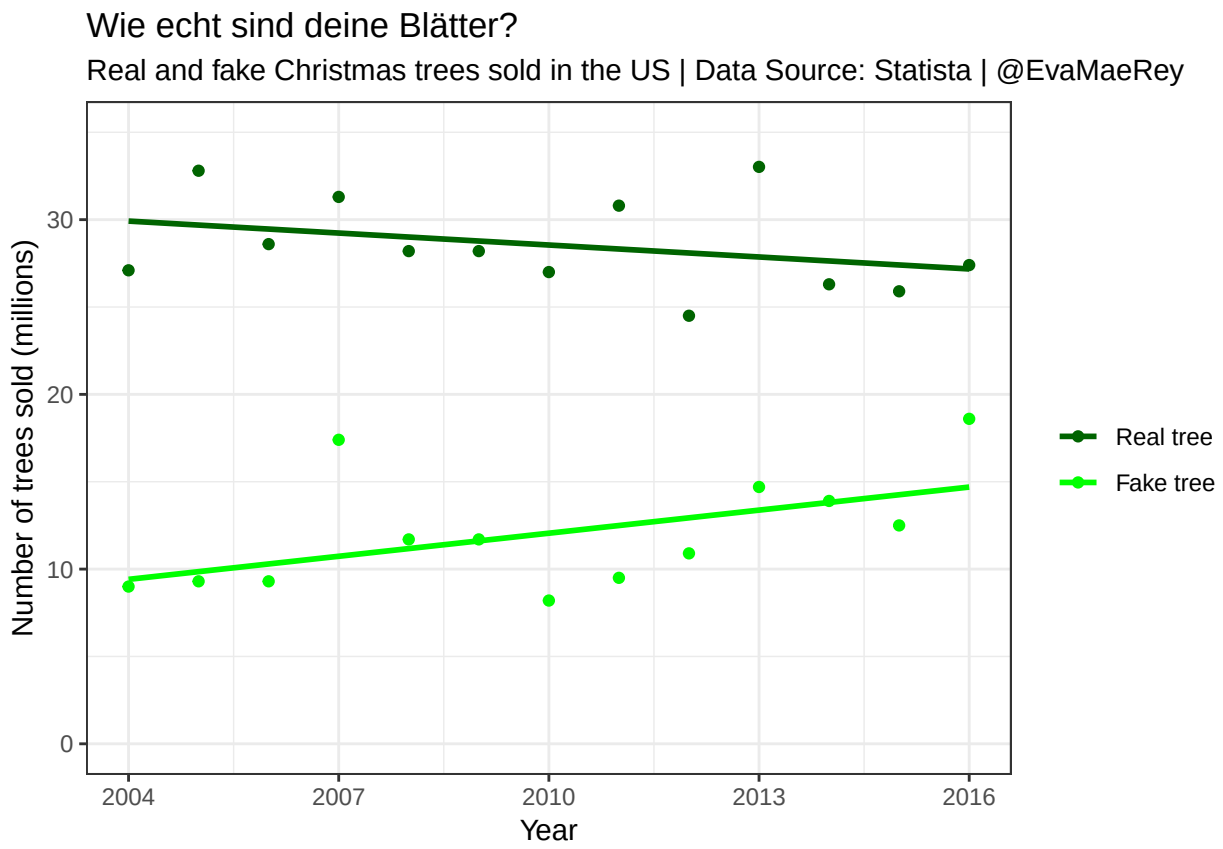
4.10

```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`) +
  geom_point() +
  aes(col = fct_rev(`Type of tree`)) +
  geom_smooth(method = "lm", se = F) +
  scale_color_manual(values = c("darkgreen", "green")) +
  ylim(c(0, 35)) +
  labs(col = "") +
  labs(title = "Wie echt sind deine Blätter?") +
  labs(subtitle = "Real and fake Christmas trees sold in the US | Data Source: Statista | @EvaMaeRey ")
```



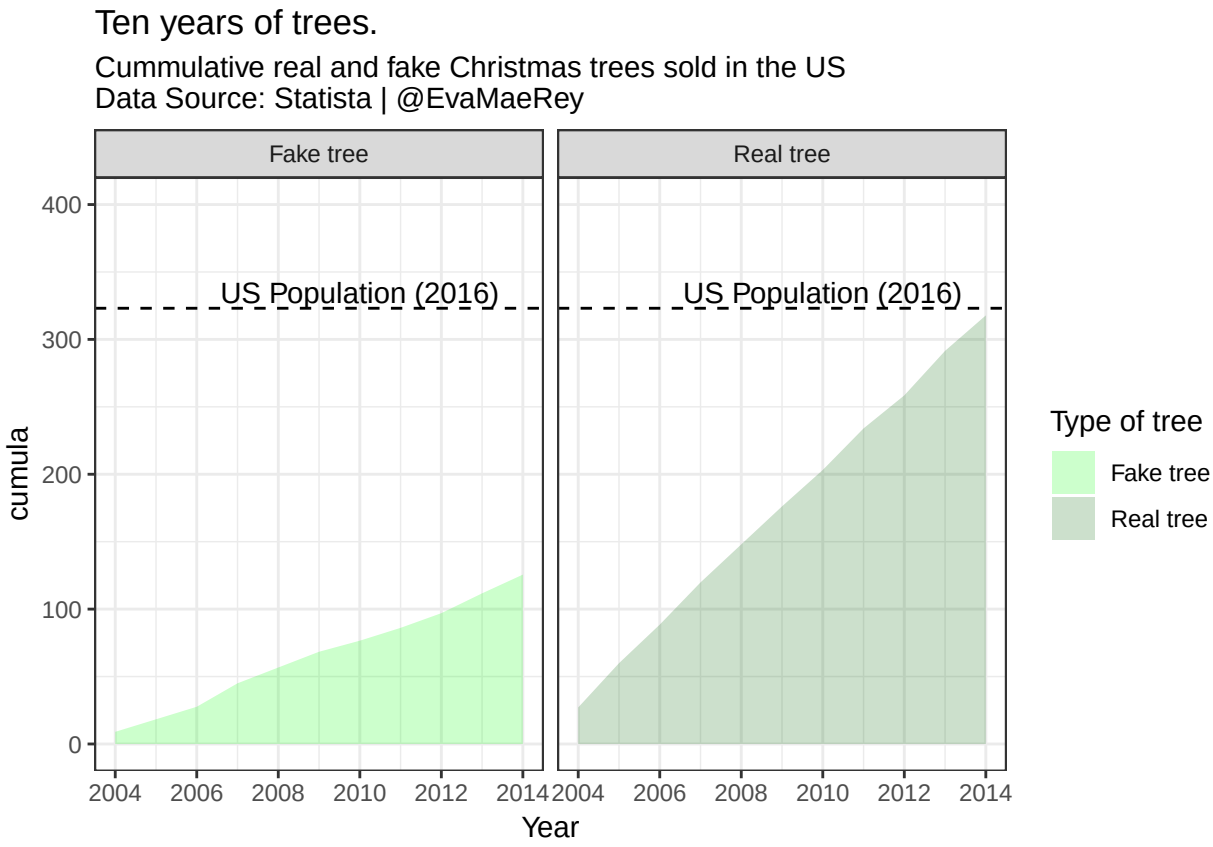
4.11

```
ggplot(data = dta) +
  aes(Year) +
  aes(y = `Number of trees sold (millions)`) +
  geom_point() +
  aes(col = fct_rev(`Type of tree`)) +
  geom_smooth(method = "lm", se = F) +
  scale_color_manual(values = c("darkgreen", "green")) +
  ylim(c(0, 35)) +
  labs(col = "") +
  labs(title = "Wie echt sind deine Blätter?") +
  labs(subtitle = "Real and fake Christmas trees sold in the US | Data Source: Statista | @EvaMaeRey ")
  theme_bw()
```




```
dta <- dta %>%
  group_by(`Type of tree`) %>%
  mutate(cumula = cumsum(`Number of trees sold (millions)`))

ggplot(dta %>% filter(Year <= 2014)) +
  aes(Year) +
  aes(y = cumula) +
  aes(fill = `Type of tree`) +
  geom_hline(yintercept = 323.1, lty = 2) +
  geom_area(alpha = .2) + facet_wrap(~ `Type of tree`) +
  annotate(geom = "text", x = 2010, y = 335, label = "US Population (2016)") +
  labs(title = "Ten years of trees.") +
  labs(subtitle = "Cummulative real and fake Christmas trees sold in the US\nData Source: Statista | @E
  scale_fill_manual(values = c("green", "darkgreen")) +
```

```
  theme_bw() +
  ylim(c(0, 400))
```



Chapter 5

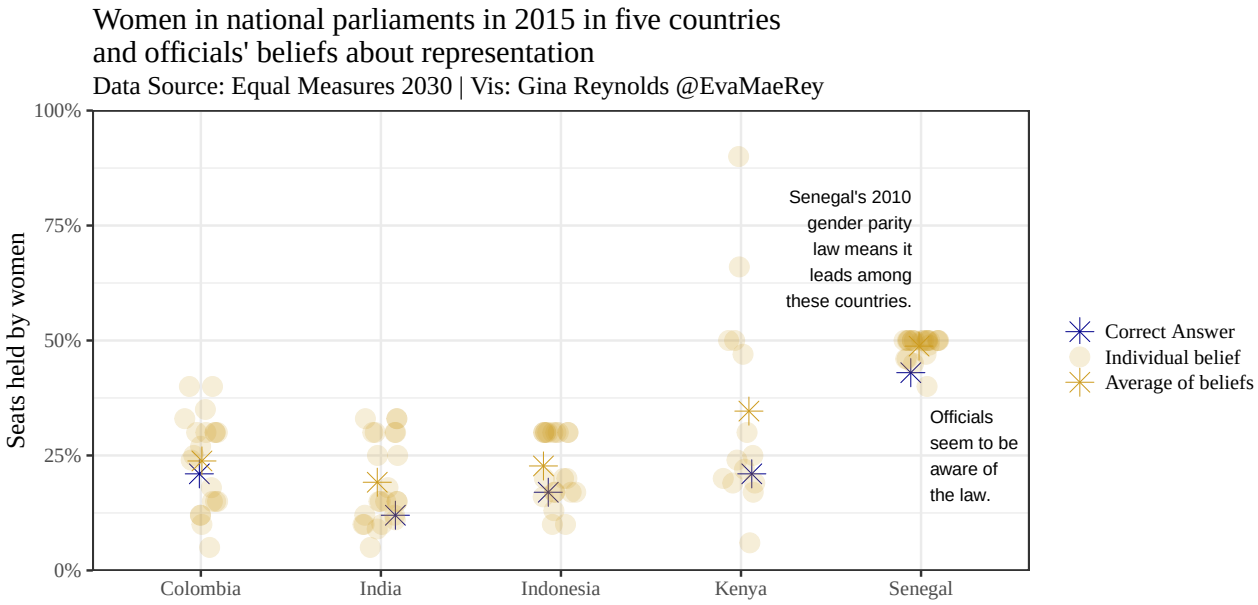
Officials' beliefs about women's representation

The data provided is based on a small survey of elite officials in five less developed countries. The question that arrises from the data is: How well do elites know the conditions in their countries. In general, the elites overestimate women's representation. But this is not the case in Senegal, where there are gender quotas in the Parliament. Most elites therefore estimate that the representation is about equal with men. I jitter the responses of the elites horizontally to avoid overplotting.

A random sample from the data set:

Country	Topic	value	value_type	alpha
Senegal	Share of seats held by women	0.4500000	Individual belief	0.3
Senegal	Share of seats held by women	0.5000000	Individual belief	0.3
India	Share of seats held by women	0.0900000	Individual belief	0.3
Indonesia	Share of seats held by women	0.1000000	Individual belief	0.3
India	Share of seats held by women	0.1917391	Average of beliefs	0.7

```
ggplot(data = df_all) +
  aes(x = Country) +
  aes(y = value) +
  aes(col = fct_inorder(value_type)) +
  aes(alpha = fct_inorder(value_type)) +
  aes(shape = fct_inorder(value_type)) +
  geom_jitter(width = .1, height = 0, size = 7) +
  geom_hline(yintercept = c(0, 100), col = "grey") +
  geom_hline(yintercept = c(50), lty = 2, col = "grey") +
  theme_bw(base_size = 20, base_family = "Times") +
  scale_y_continuous(limits = c(0, 1), expand = c(0, 0), labels = scales::percent) +
  scale_colour_manual(name = "", values = c("darkblue", "goldenrod3", "goldenrod3")) +
  scale_alpha_manual(name = "", values = c(1, .17, 1)) +
  scale_shape_manual(name = "", values = c(8, 19, 8)) +
  annotate(geom = "text", x = 4.95, y = .70, label = str_wrap("Senegal's 2010 gender parity law means i
  annotate(geom = "text", x = 5.05, y = .250, label = str_wrap("Officials seem to be aware of the law."
  labs(x = "") +
  labs(y = "Seats held by women") +
  labs(title = "Women in national parliaments in 2015 in five countries \nand officials' beliefs about
  labs(subtitle = "Data Source: Equal Measures 2030 | Vis: Gina Reynolds @EvaMaeRey")
```


Women in national parliaments in 2015 in five countries
and officials' beliefs about representation
Data Source: Equal Measures 2030 | Vis: Gina Reynolds @EvaMaeRey
Seats held by women
100%
75%
50%
25%
0%
Colombia
India
Indonesia
Kenya
Senegal
Senegal's 2010 gender parity law means it leads among these countries.
Officials seem to be aware of the law.
Correct Answer
Individual belief
Average of beliefs

Chapter 6

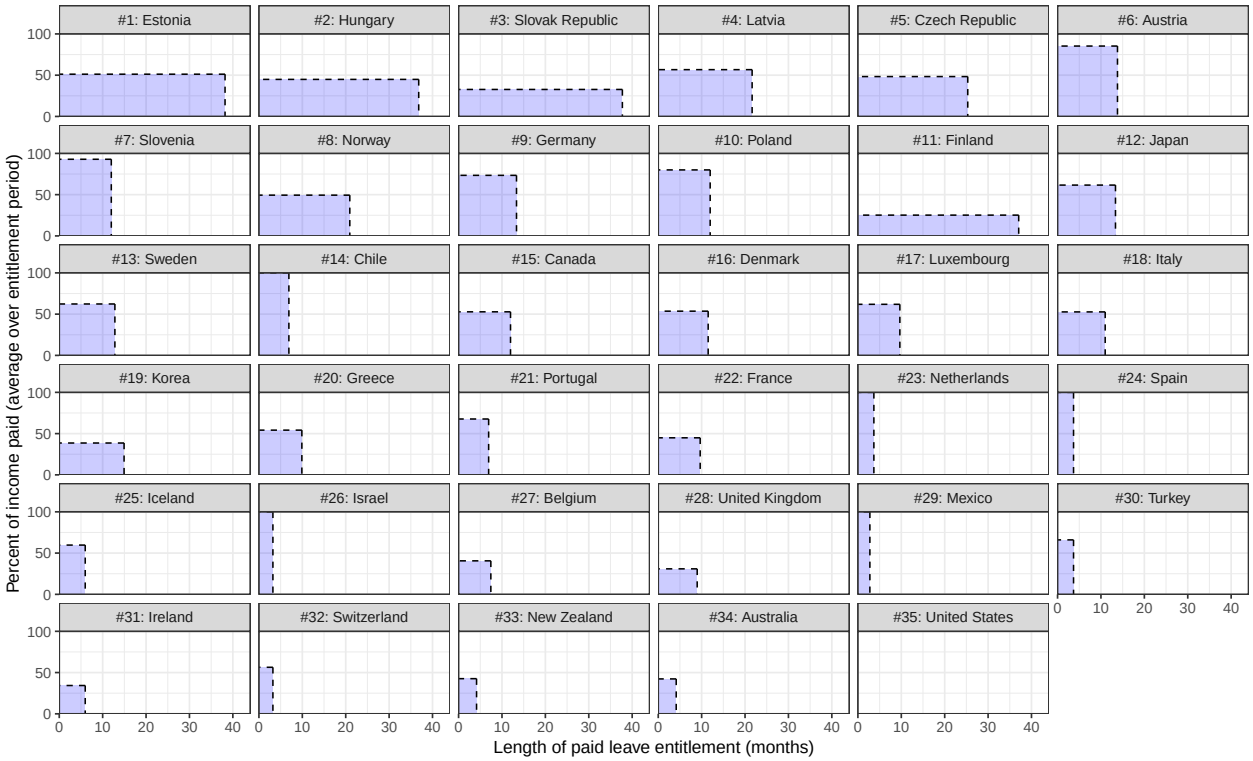
Maternal Leave

The OECD provides a comparative report on how much paid leave women are entitled to after childbirth. But leave takes different forms. In some places, the allowed leave is longer, but sometimes that means that the pay out compared to the regular salary is lower. To emphasize the different forms that law around paid leave take, I plotted the total payout available to mothers as areas of rectangles, where one side is the length of leave allowed, and the other side is the proportion of salary paid to the new mom.

A random sample from the data set:

Country	Paid maternity leave avg payment rate (%)	Paid maternity leave full rate equivalent in weeks	Paid materni
Turkey	66.0	10.6	
Finland	74.4	13.0	
Norway	97.9	12.7	
Slovenia	100.0	15.0	
Israel	100.0	14.0	

```
ggplot(df) +
  aes(x = paid_leave_months) +
  aes(y = `Total paid leave avg payment rate (%)`) +
  aes(xmin = 0) +
  aes(xmax = paid_leave_months) +
  aes(ymin = 0) +
  aes(ymax = `Total paid leave avg payment rate (%)`) +
  facet_wrap(fct_inorder(rank_name) ~ .) +
  geom_rect(fill = "blue", alpha = .2) +
  aes(yend = 0) +
  aes(xend = 0) +
  geom_segment(aes(yend = `Total paid leave avg payment rate (%)`), lty = "dashed") +
  geom_segment(aes(xend = paid_leave_months), lty = "dashed") +
  scale_y_continuous(limits = c(0, 100), expand = c(0, 0), breaks = c(0, 50, 100)) +
  scale_x_continuous(limits = c(0, 44), expand = c(0, 0)) +
  labs(x = "Length of paid leave entitlement (months)") +
  labs(y = "Percent of income paid (average over entitlement period)") +
  labs(title = "Total paid leave available to mothers in the OECD") +
  labs(subtitle = "Countries rank ordered by paid leave full rate equivalent (blue rectangular area)\nV
  theme_bw(base_size = 12)
```


Total paid leave available to mothers in the OECD
Countries rank ordered by paid leave full rate equivalent (blue rectangular area)
Visualization: Gina Reynolds | Data source: OECD.org
#1: Estonia
#2: Hungary
#3: Slovak Republic
#4: Latvia
#5: Czech Republic
#6: Austria
#7: Slovenia
#8: Norway
#9: Germany
#10: Poland
#11: Finland
#12: Japan
#13: Sweden
#14: Chile
#15: Canada
#16: Denmark
#17: Luxembourg
#18: Italy
#19: Korea
#20: Greece
#21: Portugal
#22: France
#23: Netherlands
#24: Spain
#25: Iceland
#26: Israel
#27: Belgium
#28: United Kingdom
#29: Mexico
#30: Turkey
#31: Ireland
#32: Switzerland
#33: New Zealand
#34: Australia
#35: United States
Percent of income paid (average over entitlement period)
Length of paid leave entitlement (months)

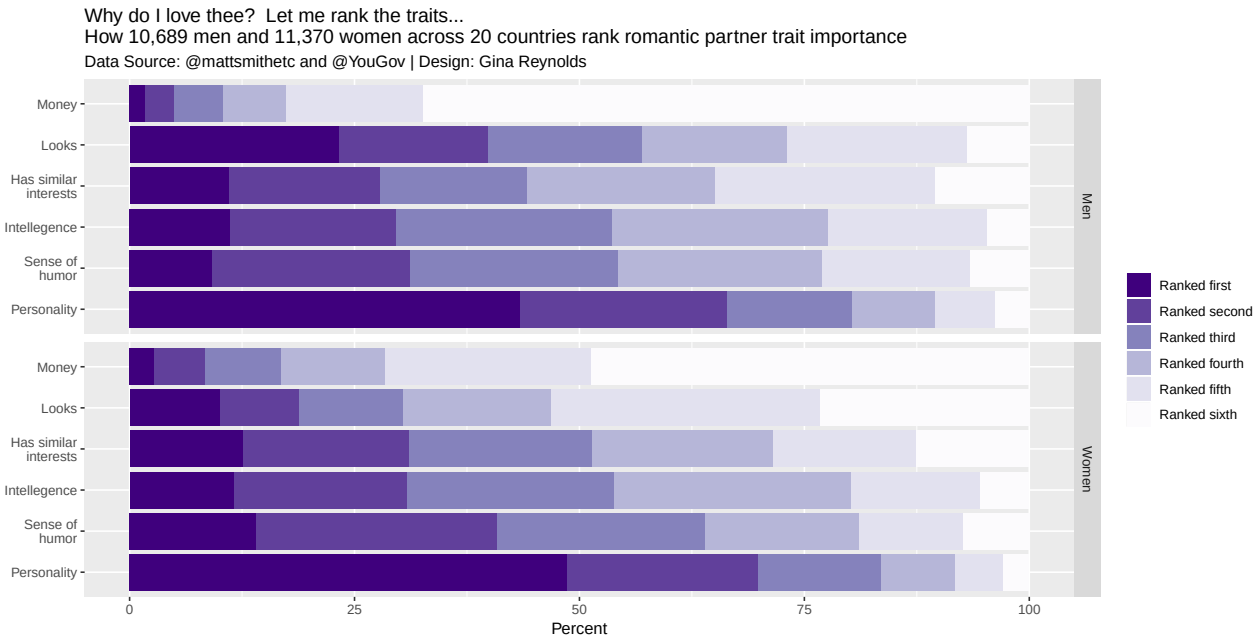
Chapter 7

Traits

A random sample from the data set:

Gender	Question_short	Rank (text)	Rank (number)	n	Percent
Women	Sense of humor	Ranked fifth	5	1314.15	11.554714
Women	Has similar interests	Ranked fourth	4	2292.38	20.111190
Men	Looks	Ranked third	3	1828.66	17.107868
Women	Intellegence	Ranked fifth	5	1636.64	14.406548
Men	Looks	Ranked sixth	6	742.71	6.948358

```
ggplot(data = world) +
  aes(x = Question_short_wrap) +
  aes(y = Percent) +
  aes(fill = `Rank (text)`) +
  facet_grid(Gender ~ .) +
  geom_col() +
  coord_flip() +
  scale_fill_manual(
    values = colorRampPalette(RColorBrewer::brewer.pal(9, "Purples"))(6)[1:6],
    guide = guide_legend(reverse = TRUE)
  ) +
  labs(fill = "") +
  xlab("") +
  labs(title = "Why do I love thee?  Let me rank the traits... \nHow 10,689 men and 11,370 women across
  labs(subtitle = "Data Source: @mattsmithetc and @YouGov | Design: Gina Reynolds")
```


Why do I love thee? Let me rank the traits...
How 10,689 men and 11,370 women across 20 countries rank romantic partner trait importance
Data Source: @mattsmithetc and @YouGov | Design: Gina Reynolds
Money
Looks
Has similar interests
Intellegence
Sense of humor
Personality
Men
Women
0
25
50
75
100
Percent
Ranked first
Ranked second
Ranked third
Ranked fourth
Ranked fifth
Ranked sixth

Chapter 8

Salarys of Trump and Obama White House Employees

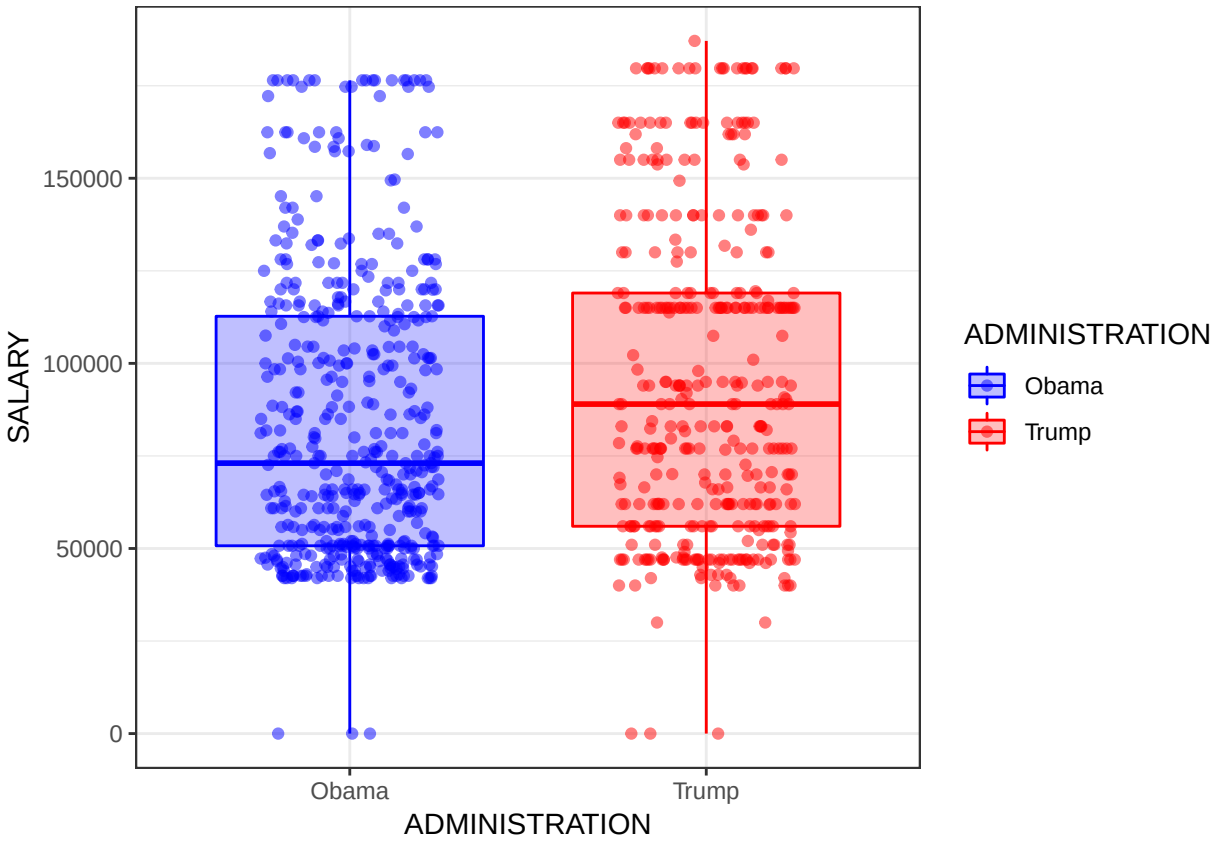
The data set, originally reported on in an NPR article, shows the difference in the distribution of salaries for the Obama and early Trump White House.

First I plot a histogram of each administration. Then I also contrast boxplots for each administration; the data points are overlayed, jittered to the widths of the boxplots. Plotly is used to make the graph interactive; mousing over will allow you to see who the point represents, their job description and exactly how much they are paid.

A random sample from the data set:

ADMINISTRATION	NAME	STATUS	SALARY	PAY BASIS	POSITION TITLE
Trump	Clemens, Nicholas J.	Employee	40000	Per Annum	WRITER FOR CORRESPOND
Trump	Larimer, Becky S.	Employee	70100	Per Annum	CALLIGRAPHER
Obama	Dyer, Deesha A.	Employee	119723	Per Annum	SPECIAL ASSISTANT TO TH
Obama	Gianotti, Claire L.	Employee	51700	Per Annum	ASSOCIATE RESEARCH DIR
Obama	Samuels, Jr., Wendell A.	Employee	73270	Per Annum	RECORDS MANAGEMENT IN

```
ggplot(both_data) +
  aes(x = ADMINISTRATION) +
  aes(y = SALARY) +
  geom_jitter(alpha = .5, height = 0, width = .25) +
  aes(col = ADMINISTRATION) +
  geom_boxplot(alpha = .25) +
  aes(fill = ADMINISTRATION) +
  scale_colour_manual(values = c("blue", "red")) +
  scale_fill_manual(values = c("blue", "red")) +
  theme_bw()
```


SALARY
150000
100000
50000
0
Obama
Trump
ADMINISTRATION
ADMINISTRATION
Obama
Trump

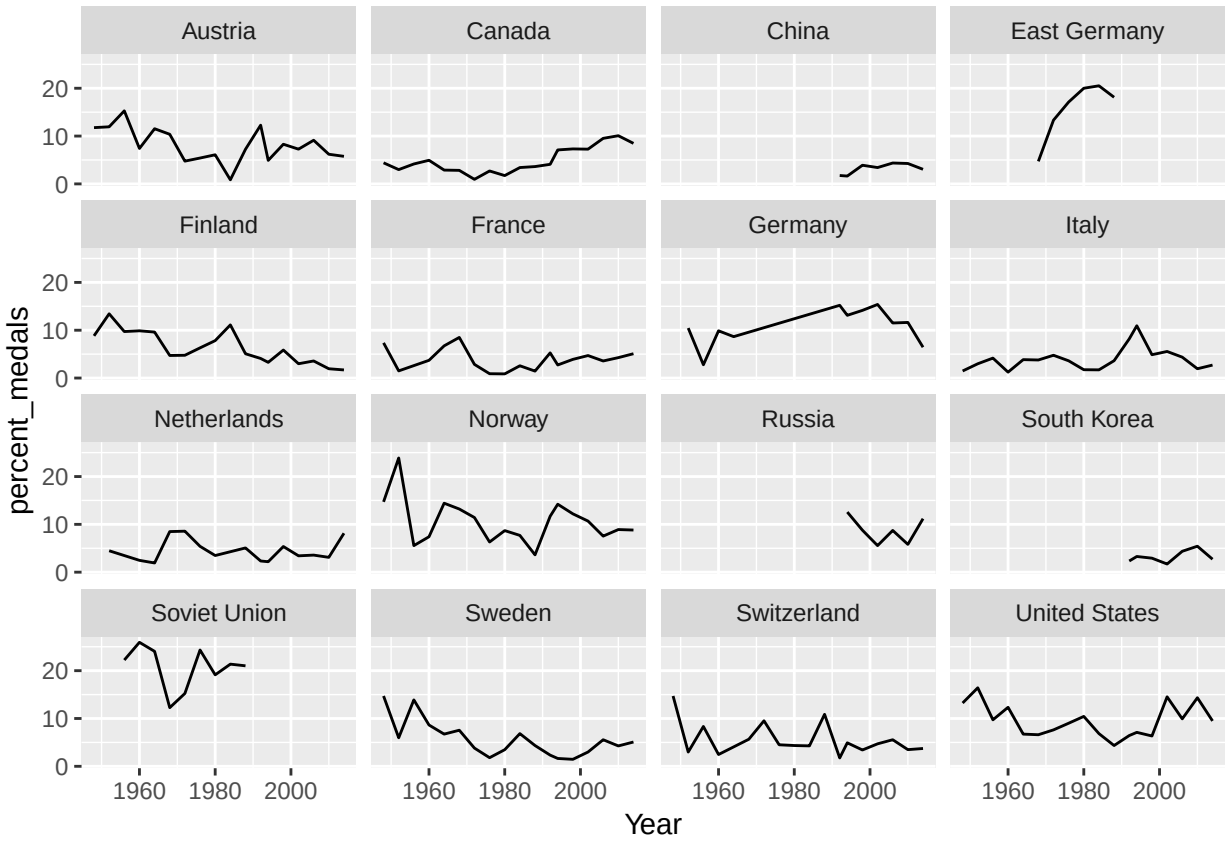
Chapter 9

Winter Games

A random sample from the data set:

Year	Sport	Event	Country	Gender	Medal Rank	Medal	Na
1960	Speedskating	Men's 5,000 Meters	Norway	Men	2	silver	K
2014	Cross-Country Skiing	Men's 15-Kilometer Classic	Sweden	Men	2	silver	Jo
1968	Ski Jumping	Men's Normal Hill, Individual	Czechoslovakia	Men	1	gold	Ji
1964	Cross-Country Skiing	Women's 3 Ã— 5-Kilometer Relay	Sweden	Women	2	silver	Sv
2014	Alpine Skiing	Women's Downhill	Slovenia	Women	1	gold	Ti

```
ggplot(data = dta) +
  aes(x = Year) +
  aes(y = percent_medals) +
  geom_line() +
  facet_wrap(~ Country)
```



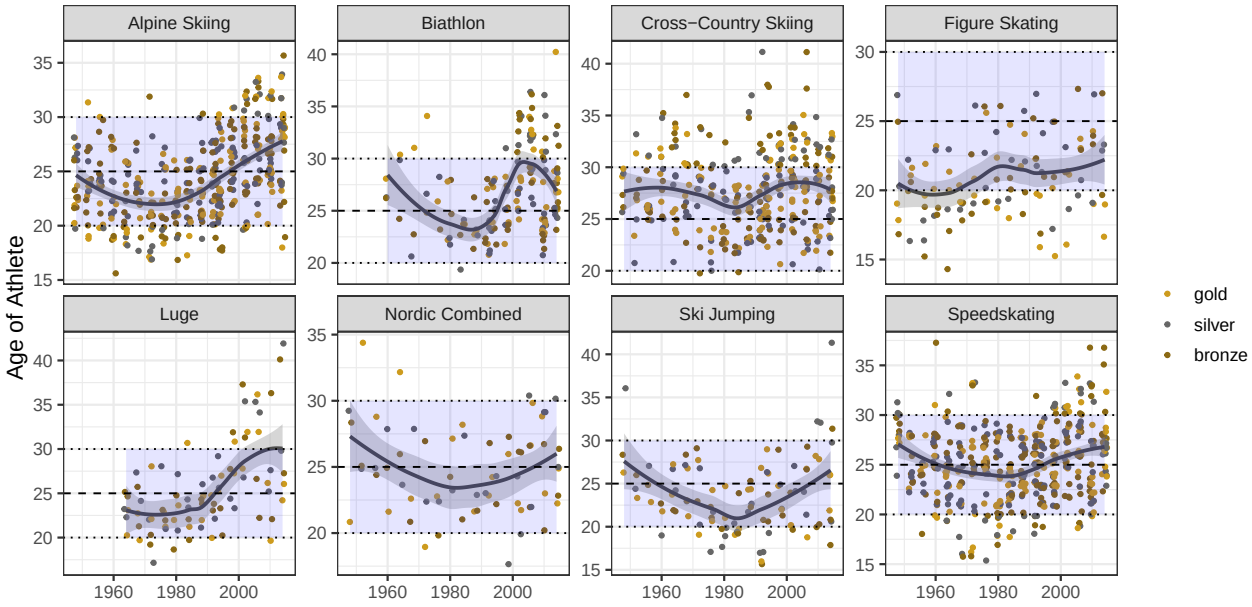
A random sample from the data set:

Year	Sport	Event	Country	Gender	Medal Rank	Medal	Name of Athle
1952	Speedskating	Men's 10,000 Meters	Norway	Men	1	gold	Hjalmar Ande
1976	Luge	Men's Doubles	Austria	Men	3	bronze	Austria-1
1992	Figure Skating	Mixed Pairs	Unified Team	Mixed	1	gold	Unified Team-
2010	Ski Jumping	Men's Normal Hill, Individual	Poland	Men	2	silver	Adam MaÅ,ys
1948	Speedskating	Men's 10,000 Meters	Finland	Men	2	silver	Lassi Parkkine

```
ggplot(dta) +
  aes(x = Year) +
  aes(y = `Age of Athlete`) +
  facet_wrap(~ Sport, scales = "free_y", nrow = 2) +
  geom_jitter(size = 1, mapping = aes(col = fct_inorder(Medal))) +
  geom_smooth(col = "grey30") +
  geom_ribbon(ymin = 20, ymax = 30, alpha = .1, fill = "blue") +
  geom_hline(yintercept = c(20, 30), lty = "dotted") +
  geom_hline(yintercept = c(25), lty = "dashed") +
  scale_color_manual(values = c("goldenrod3", "grey40", "goldenrod4"), name = "") +
  labs(x = "") +
  labs(title = "Young and old at the Winter Olympics: medalists' declared ages have risen in recent yea
  labs(subtitle = "Includes individual sports that have been in Olympic since 1965") +
  labs(caption = "Source: Sports-Reference.com | Vis: Gina Reynolds @EvaMaeRey \nValues 'jittered' to r
  theme_bw(base_size = 13)
```

Young and old at the Winter Olympics: medalists' declared ages have risen in recent years

Includes individual sports that have been in Olympic since 1965



Source: Sports−Reference.com | Vis: Gina Reynolds @EvaMaeRey
Values 'jittered' to reduce overplotting

Chapter 10

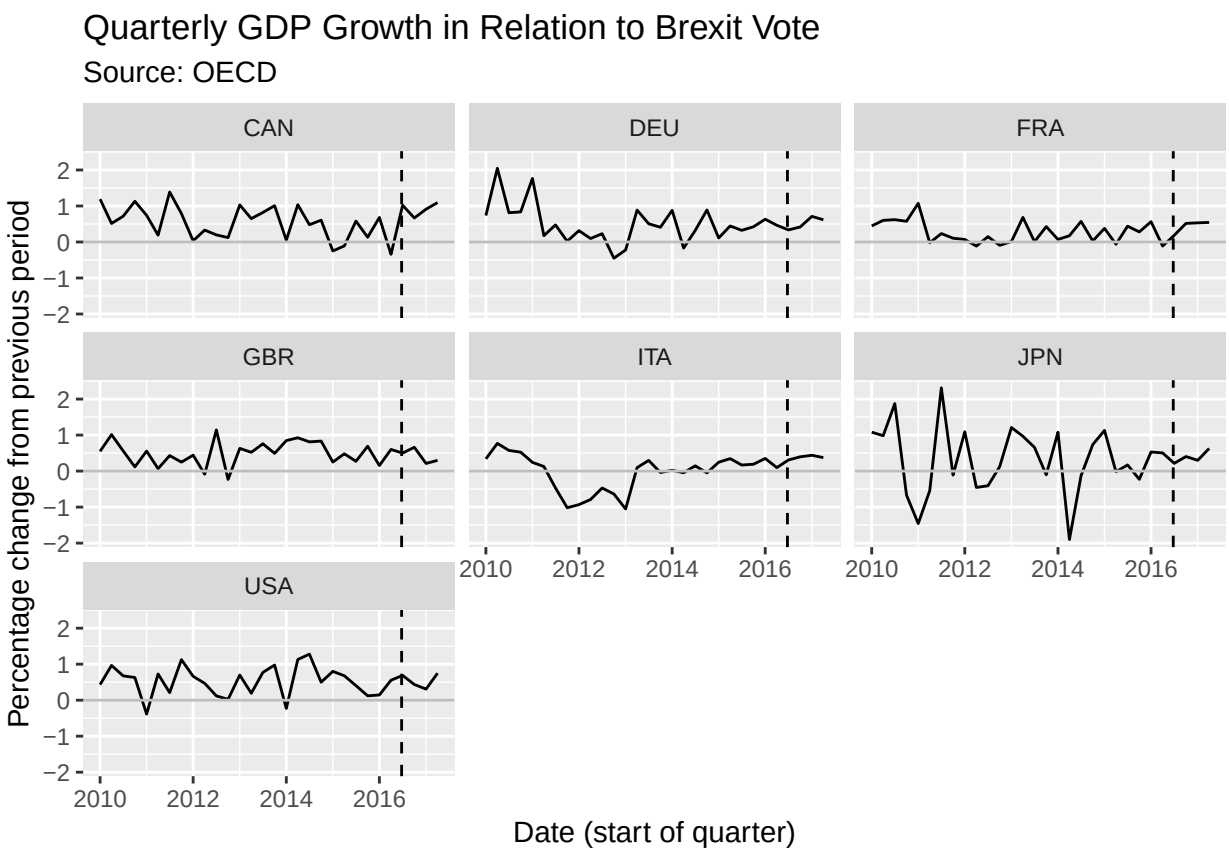
Brexit

This visualization challenge was a proposed makeover for a Financial Times visualization focusing on relative economic growth in G7 countries, with an emphasis on growth in the UK, focusing especially since Brexit. The visualization I present here is not what I created at the time of the challenge; instead it is inspired by Alan Smith a data journalist at the Financial Times, who created a really compelling visualization a couple of months after MakeoverMonday's treatment. I try to recreate his plot - which uses a ribbon to contain all G7 countries, and plot the UK's stats thereover. This declutters the graph, and makes you focus on where the UK falls among other countries, without being needlessly specific about those countries; the data story isn't about them anyway, might be Smith's thinking. My graph actually lightly traces economic growth in other countries, but deemphasizes their importance, like Smith.

A random sample from the data set:

Country	Year	Quarter	Date (start of quarter)	Percentage change from previous period	Date (start o quarter)
FRA	2016	4	2016-10-01	0.516292	2016-10-01
DEU	2016	1	2016-01-01	0.632392	2016-01-01
USA	2016	3	2016-07-01	0.687731	2016-07-01
FRA	2012	3	2012-07-01	0.148280	2012-07-01
USA	2012	4	2012-10-01	0.022756	2012-10-01

```
ggplot(data = data) +
  aes(x = `Date (start of quarter)`) +
  aes(y = `Percentage change from previous period`) +
  facet_wrap(~ Country) +
  geom_line() +
  geom_hline(yintercept = 0, col = "grey") +
  geom_vline(xintercept = as.numeric(as.POSIXct("2016-06-23")), lty = "dashed") +
  labs(title = "Quarterly GDP Growth in Relation to Brexit Vote") +
  labs(subtitle = "Source: OECD")
```

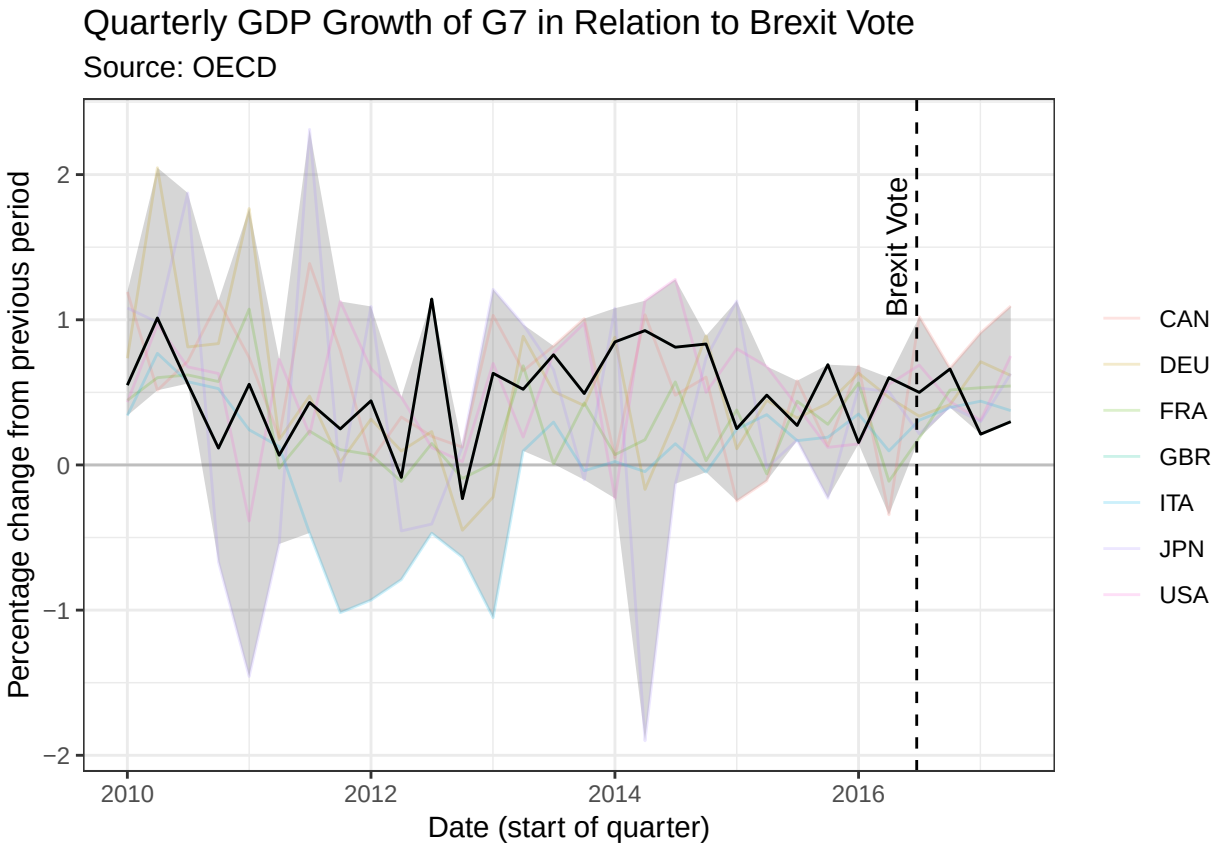


A random sample from the data set:

Country	Year	Quarter	Date (start of quarter)	Percentage change from previous period	Date (start o quarter)	
FRA	2011	4	2011-10-01	0.104859	2011-10-01	-
JPN	2013	4	2013-10-01	-0.102503	2013-10-01	-
FRA	2014	1	2014-01-01	0.070655	2014-01-01	-
FRA	2014	4	2014-10-01	0.029185	2014-10-01	-
JPN	2013	1	2013-01-01	1.210002	2013-01-01	-

```
ggplot(data = data) +
  aes(x = `Date (start of quarter)`) +
  aes(y = `Percentage change from previous period`) +
  aes(ymin = min_) +
  aes(ymax = max_) +
  geom_hline(yintercept = 0, col = "grey") +
  geom_ribbon(alpha = .2) +
  geom_line(aes(col = Country), alpha = .2) +
  geom_line(data = data %>% filter(Country == "GBR"), col = "black") +
  geom_vline(xintercept = as.numeric(as.POSIXct("2016-06-23")), lty = 2) +
  annotate(
    geom = "text", x = as.POSIXct("2016-04-23"), y = 1.5,
    label = "Brexit Vote", angle = 90
  ) +
  labs(
    title = "Quarterly GDP Growth of G7 in Relation to Brexit Vote",
    subtitle = "Source: OECD",
    col = ""
  ) +
```

```
theme_bw()
```


Quarterly GDP Growth of G7 in Relation to Brexit Vote
Source: OECD
Percentage change from previous period
2
1
0
−1
−2
2010
2012
2014
2016
Date (start of quarter)
Brexit Vote
CAN
DEU
FRA
GBR
ITA
JPN
USA

Chapter 11

Curry in London

This visualization task seemed to get at the question: Does where you eat matter. The data was the cost of identical menu items at different locations of the same restaurant, the Wetherspoon, around the UK.

First, I mapped the cost of a single menu item, the Empire Burger, across the UK. Then, I calculated the distance from Wetherspoon restaurants from the Big Ben, and plotted prices as a function of this distance – plotting only the restaurants in a 10 kilometer radius.

A random sample from the data set:

Name	Location	Latitude	Longitude	Empire State Burger	Chicken Tikka	Gammon af
The Green Parrot	Perranporth	50.34451	-5.1559044	10.85	8.40	
The Bowling Green	Otley	53.90427	-1.6928946	8.75	7.40	
The Moon And Spoon	Slough	51.50947	-0.5959543	8.75	7.19	
The George Hotel	Peterborough	52.64967	-0.4783880	8.75	7.19	
The Ice Wharf	Camden, London	51.54079	-0.1454212	11.25	8.39	

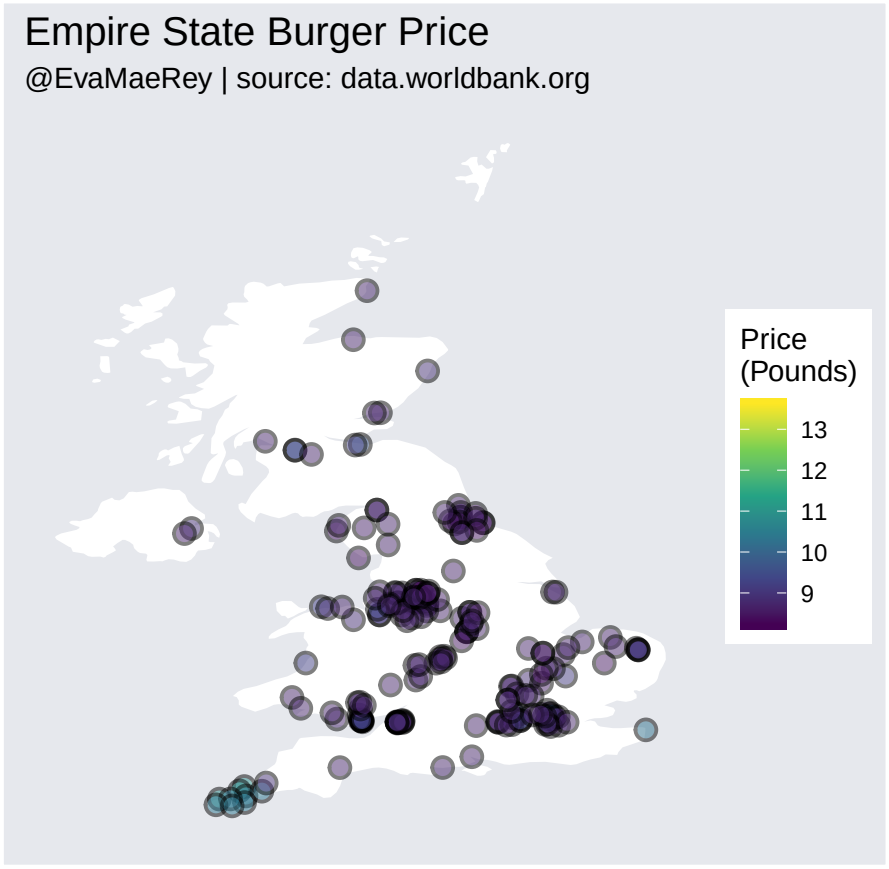
```
# Mapping data
world_map_df <- map_data("world")
```

A random sample from the data set:

	long	lat	group	order	region	subregion
88312	100.6295	6.447998	1413	88312	Thailand	NA
18728	-133.1968	68.739845	345	18728	Canada	NA
8316	137.4836	-34.252148	177	8316	Australia	NA
81664	174.7150	61.947903	1309	81664	Russia	NA
93724	-115.1252	32.683304	1501	93724	USA	NA

```
# create a blank ggplot theme
theme_opts <- theme(
  panel.grid.minor = element_blank(),
  panel.grid.major = element_blank(),
  panel.background = element_blank(),
  plot.background = element_rect(fill = "#e6e8ed"),
  panel.border = element_blank(),
  axis.line = element_blank(),
  axis.text.x = element_blank(),
  axis.text.y = element_blank(),
  axis.ticks = element_blank(),
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
```

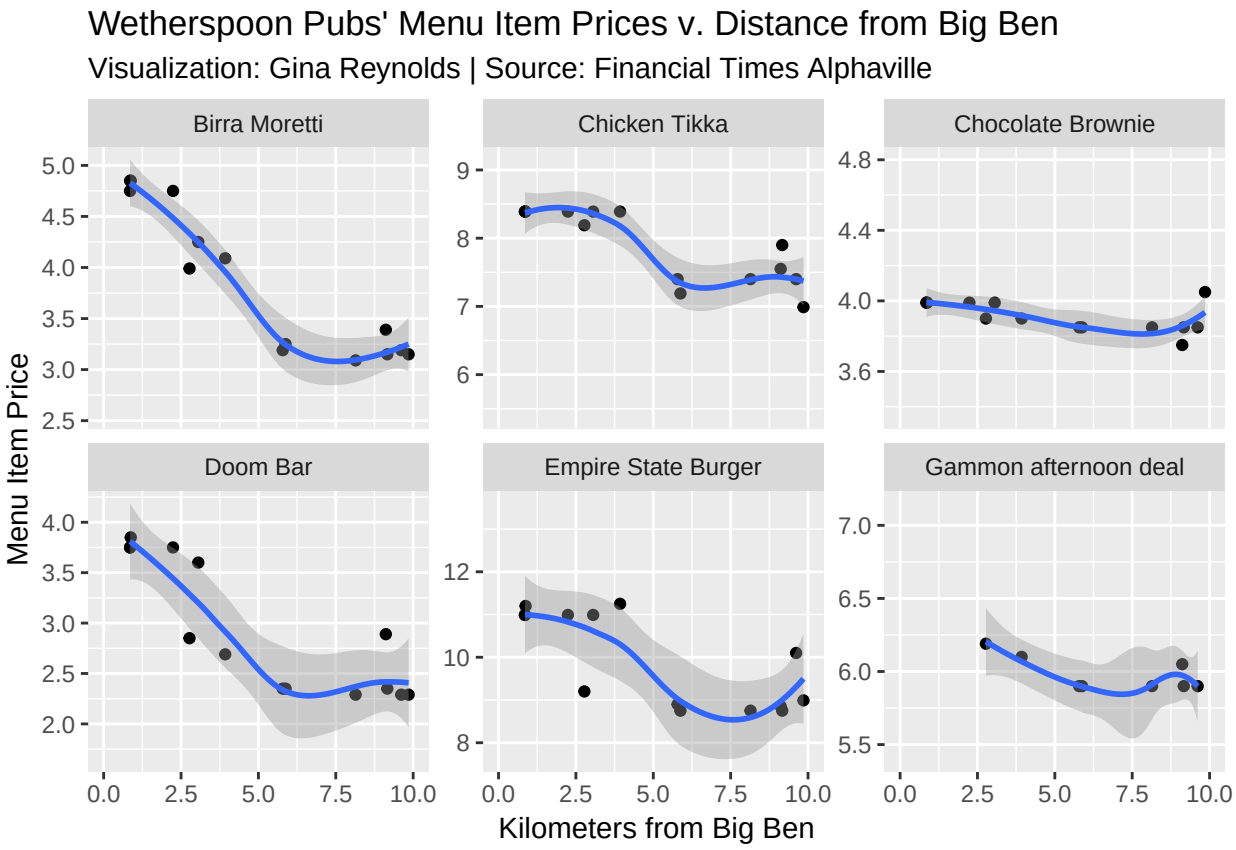
```
  plot.title = element_text(size = 15)
)


ggplot(data = world_map_df %>% filter(region == "UK")) +
  aes(x = long) +
  aes(y = lat) +
  aes(group = group) +
  geom_polygon(fill = "white") +
  coord_equal() +
  scale_fill_viridis_c(option = "viridis") +
  geom_point(data = data0,
             mapping = aes(x = Longitude, y = Latitude,
                           group = NULL, fill = `Empire State Burger`),
    colour = "black", shape = 21, stroke = 1, alpha = .5, size = 3
  ) +
  labs(fill = "Price\n(Pounds)") +
  labs(title = "Empire State Burger Price") +
  labs(subtitle = "@EvaMaeRey | source: data.worldbank.org") +
  theme_opts
```



A random sample from the data set:

Name	Location	Latitude	Longitude	Notes	Moretti as a % of a tikka	Moretti as % of bu
The Whiffler	Norwich	52.65430	1.268524	NA	0.4113924	0.359
The Kingfisher	Poynton	53.34644	-2.124271	NA	0.3920386	0.371
Bull and Stirrup Hotel	Chester	53.19436	-2.893331	NA	0.4172015	0.357
The Great Central	Wilmslow Road	53.44093	-2.219457	NA	0.4256757	0.360
The Hornet	Birmingham	52.49268	-1.819488	NA	0.4741379	0.314

```
ggplot(data = dataLong) +
  aes(x = `Kilometers from Big Ben`) +
  aes(y = `Menu Item Price`) +
  facet_wrap(~ Item, scales = "free_y") +
  geom_point() +
  geom_smooth() +
  xlim(c(0, 10)) +
  labs(title = "Wetherspoon Pubs' Menu Item Prices v. Distance from Big Ben") +
  labs(subtitle = "Visualization: Gina Reynolds | Source: Financial Times Alphaville")
```



Chapter 12

Life Expectancy Increases

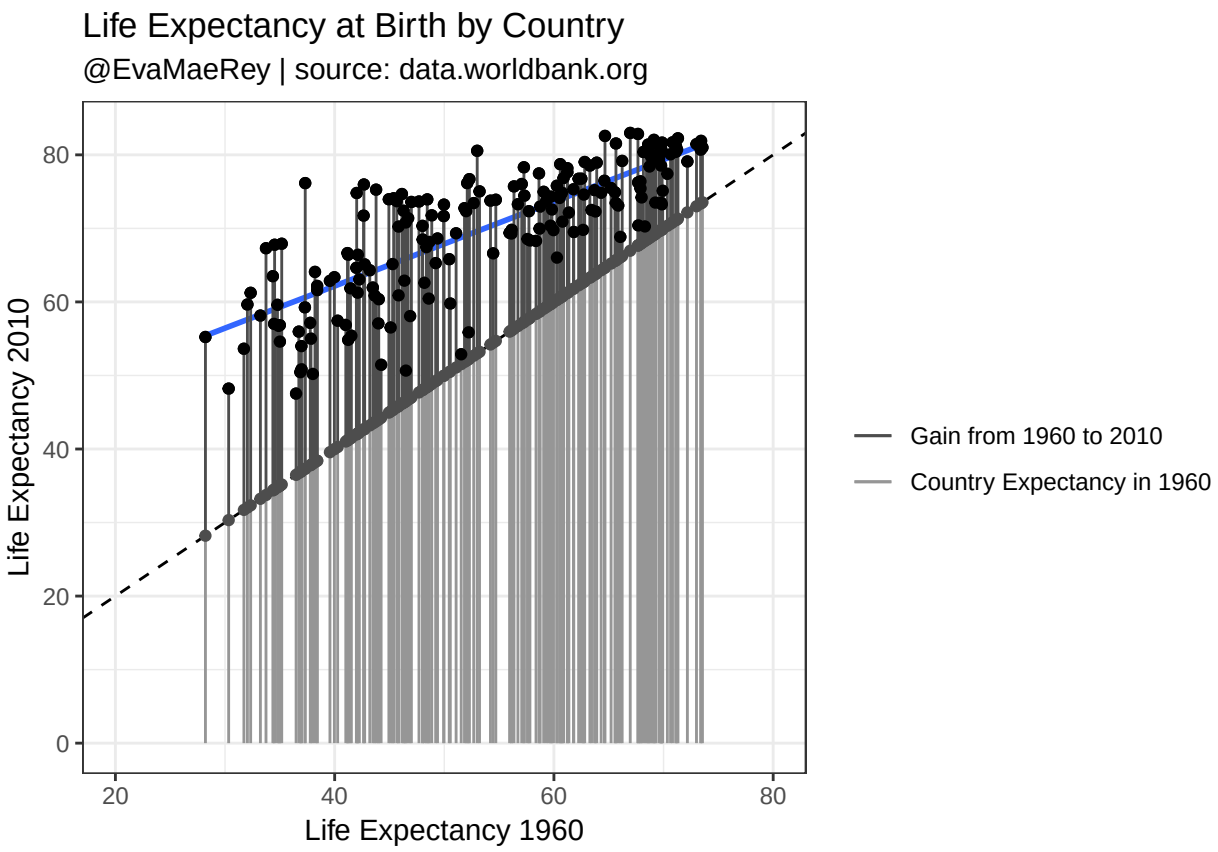
To dramatically show the increases in life expectancy by country from 1960 to 2010, I plot the variable in 1960 versus itself in 2010. The line of equivilance (a 45% angle) is used as a reference and shows the result that you would see if there where no growth. The vertical distance from this line is the increase in life expectancy. I also superimpose a linear model on top of the scatter plot. You can see that the gains are greater for countries that started off with lower life expectancies.

A random sample from the data set:

Life Expectancy 1960	Country Code	Country Name	Region	Income Group	Year
69.10927	ESP	Spain	Europe & Central Asia	High income	2010
65.56937	ABW	Aruba	Latin America & Caribbean	High income	2010
41.98105	PNG	Papua New Guinea	East Asia & Pacific	Lower middle income	2010
60.78085	AZE	Azerbaijan	Europe & Central Asia	Upper middle income	2010
68.29954	UKR	Ukraine	Europe & Central Asia	Lower middle income	2010

```
ggplot(compare) +
  aes(x = `Life Expectancy 1960`) +
  aes(y = `Life Expectancy 2010`) +
  geom_point() +
  geom_smooth(se = F, method = "lm") +
  geom_abline(slope = 1, intercept = 0, lty = 2) +
  # coord_fixed() +
  aes(xend = `Life Expectancy 1960`) +
  aes(yend = `Life Expectancy 1960`) +
  geom_segment(mapping = aes(col = "Gain from 1960 to 2010")) +
  geom_segment(mapping = aes(y = 0, col = "Country Expectancy in 1960")) +
  scale_color_manual(
    breaks = c(
      "Gain from 1960 to 2010",
      "Country Expectancy in 1960"
    ),
    values = c("grey59", "grey30", "grey30")
  ) +
  geom_point(aes(y = `Life Expectancy 1960`), col = "grey30") +
  geom_point() +
  labs(subtitle = "@EvaMaeRey | source: data.worldbank.org", size = .7) +
  labs(title = "Life Expectancy at Birth by Country") +
  labs(col = "") +
  theme(legend.title = element_blank()) +
```

```
  theme_bw() +
  xlim(c(20, 80))
```



Chapter 13

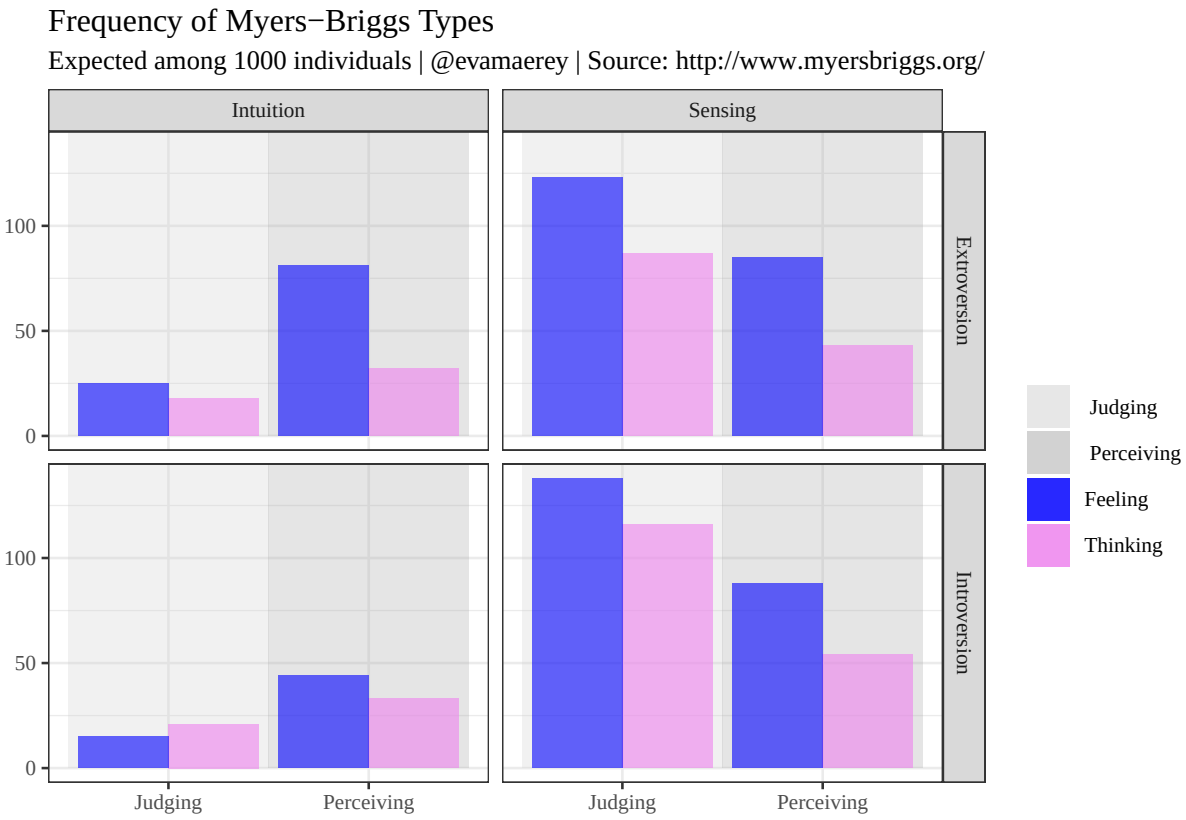
Myers Briggs

This data looks at the relationship betweeen four binary variables. The challenge is how to display that in one visualization. My first idea was to use a mosaic plot. However, I came across advice from "The Perceptual Edge", that generally advised against the use of the mosaic plot, instead favoring a kind of nested bar plot. I tried to implement that. While I do think that it is pretty, I think that it still requires a lot of the reader to interpret the graph. Perhaps more annotation could alleviate this burden.

A random sample from the data set:

(S)ensing/I(N)tuition	(T)hinking/(F)eeling	(J)udging/(P)erceiving	(E)xtroversion/(I)ntroversion	count
Intuition	Thinking	Perceiving	Introversion	1
Intuition	Feeling	Judging	Introversion	1
Intuition	Thinking	Judging	Extroversion	1
Intuition	Feeling	Perceiving	Introversion	1
Sensing	Thinking	Perceiving	Extroversion	1

```
ggplot(d) +
  aes(x = `(J)udging/(P)erceiving`) +
  aes(fill = `(T)hinking/(F)eeling`) +
  facet_grid(`(E)xtroversion/(I)ntroversion` ~
               `(S)ensing/I(N)tuition`) +
  geom_rect(aes(x = NULL, y = NULL,
                xmin = mins, xmax = max,
                fill = `judging perceiving`),
                ymin = 0, ymax = 700, data = background
  ) +
  geom_bar(position = "dodge") +
  scale_fill_manual(values = alpha(c("lightgrey", "darkgrey", "blue", "violet"), c(.3, .3, .6, .6))) +
  labs(x = "") +
  labs(y = "") +
  labs(fill = "") +
  labs(title = "Frequency of Myers-Briggs Types") +
  labs(subtitle = "Expected among 1000 individuals | @evamaerey | Source: http://www.myersbriggs.org/") +
  theme_bw(base_size = 10, base_family = "Times")
```


Frequency of Myers−Briggs Types
Expected among 1000 individuals | @evamaerey | Source: http://www.myersbriggs.org/
Intuition
Sensing
Extroversion
Introversion
100
50
0
Judging
Perceiving
Feeling
Thinking

Chapter 14

Wine

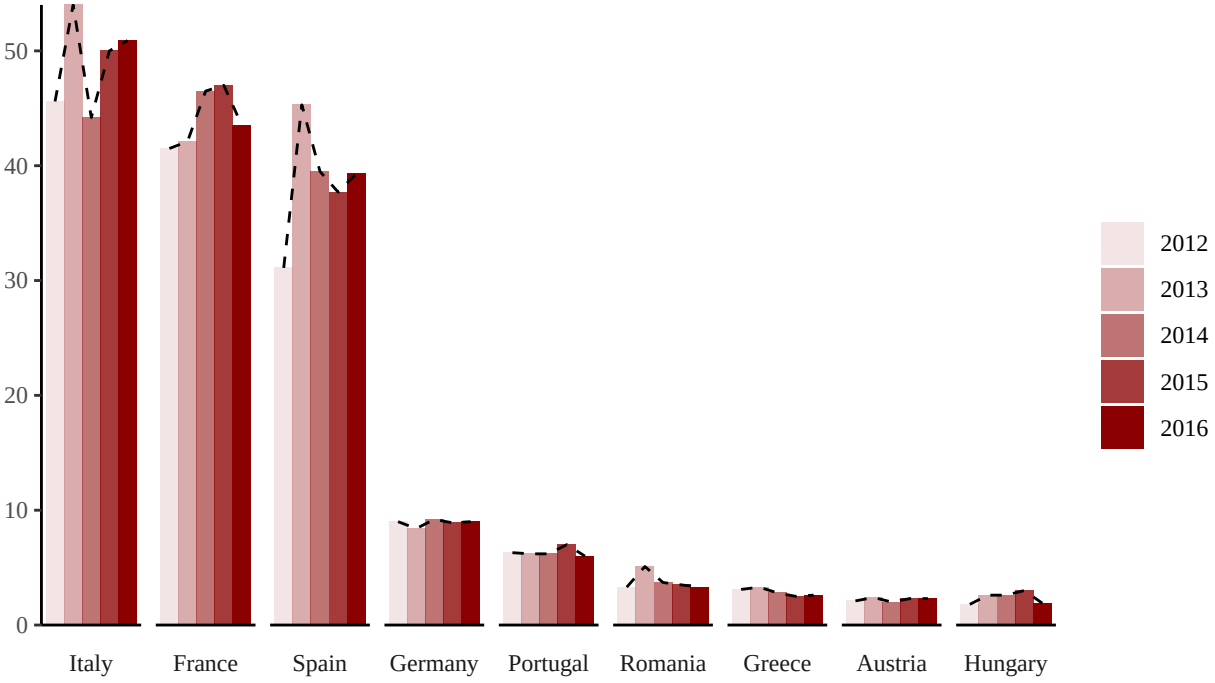
Wine production in Europe may have been volitile during the years plotted because of production control policies implemented by the EU. The big three, Italy, France and Spain, particularly saw a lot of volitility early in this period.

```
df <- readxl::read_xlsx("raw_data/Wine_Production_by_country.xlsx") %>%
  filter(Country != "World total")

Europe <- c(
  "Italy", "France", "Spain",
  "Germany", "Portugal", "Romania",
  "Austria", "Greece", "Hungary"
)
```

```
  ggplot(df %>% filter(Country %in% Europe)) +
  aes(x = Year) +
  aes(y = `Wine production in mhl`) +
  facet_wrap(~ fct_inorder(Country), strip.position = "bottom", nrow = 1) +
  geom_col(aes(alpha = Year), position = "dodge", fill = "darkred", width = 1) +
  geom_line(col = "black", lty = 2) +
  scale_y_continuous(expand = c(0, 0)) +
  labs(fill = "") +
  labs(alpha = "")+
  labs(title = "Wine production (mhl) in principle European markets, 2012-2016") +
  labs(subtitle = "The EU program to regulate viticultural production ended upon the 2011/2012 harvest.
    labs(caption = "Design: Gina Reynolds @EvaMaeRey  \nData Source: International Organisation of Vine
  theme_classic(base_family = "Times") +
  theme(
    axis.title = element_blank(),
    strip.placement = "outside",
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    strip.background = element_blank(),
    plot.caption = element_text(size = 10)
  )
```

Wine production (mhl) in principle European markets, 2012−2016

The EU program to regulate viticultural production ended upon the 2011/2012 harvest.



Design: Gina Reynolds @EvaMaeRey
Data Source: International Organisation of Vine and Wine

Chapter 15

Arctic Ice

This visualization shows the trend in Arctic Ice Sea Extent, data from the National Snow and Ice Data Center. If I recall correctly, the definition for coverage is the case where at least 15 percent of the sea is ice.

The visualization shows melting and freezing cycles, in accordance with the seasons — and the disconcerting trend of a general decrease in ice extent over the years.

One problem that arises is due to inconsistant number of days in each year. There is a measurement for every day, but leap years contain a extra day. Which means that plotting years over years leads to imperfect alignment. My solution was just to pretend that all the data come from a single year, 2000, and plot each of the years on that scale. The earliest year cycle and last year cycle are highlighted in white.

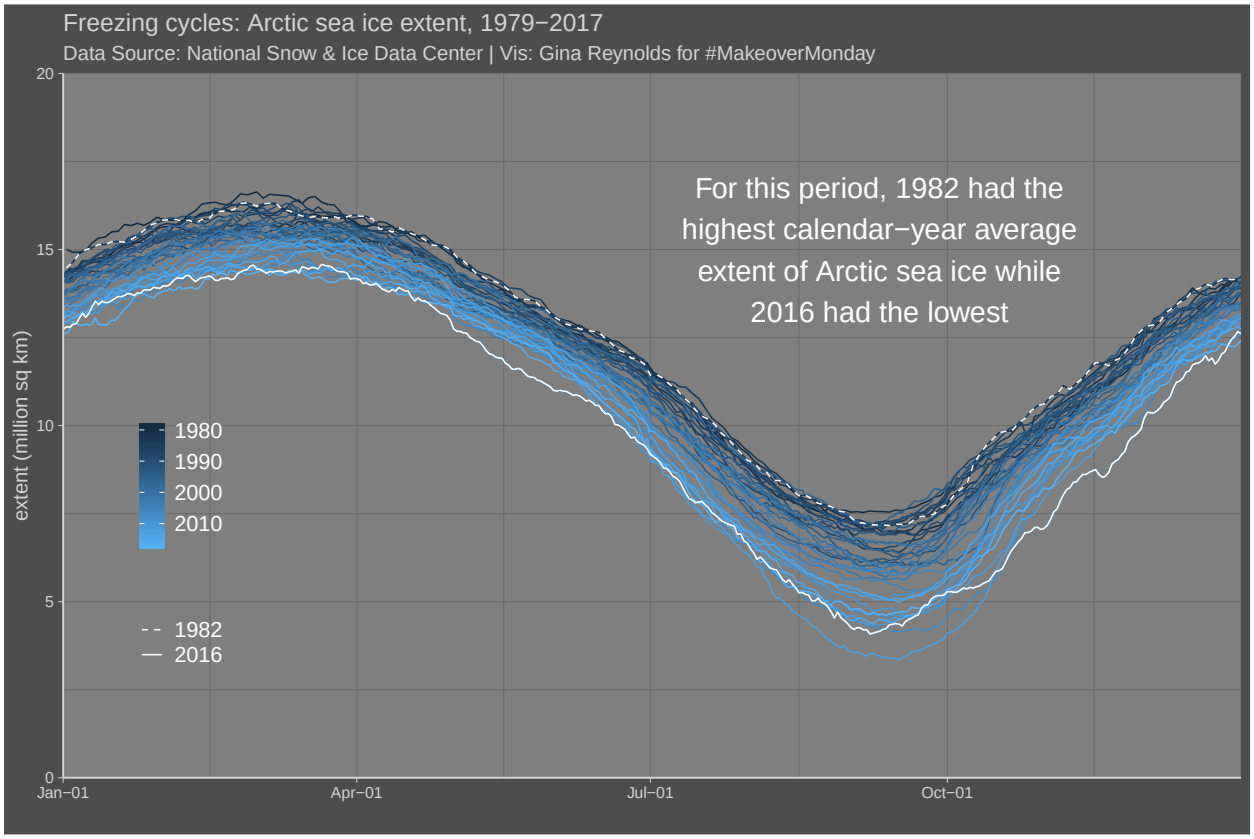
A random sample from the data set:

Date	Extent (million sq km)	year	month_day	month_day_plus	proportion_ocean_covered_in_ice	mea
2008-02-27	15.354	2008	02-27	2000-02-27	0.0426500	
1992-11-15	11.138	1992	11-15	2000-11-15	0.0309389	
1994-05-01	14.126	1994	05-01	2000-05-01	0.0392389	
2014-04-07	14.479	2014	04-07	2000-04-07	0.0402194	
2001-06-09	12.073	2001	06-09	2000-06-09	0.0335361	

year	average_coverage	num_days	average_day
1982	12.43945	182	1982-07-02 00:00:00
2016	10.15069	366	2016-07-01 12:00:00

```
# breaks for x axis.
br <- as.numeric(lubridate::ymd(c(
  "2000-01-01", "2000-04-01",
  "2000-07-01", "2000-10-01", "2001-01-01"
)))

ggplot(df) +
  aes(x = as.numeric(month_day_plus)) +
  aes(y = `Extent (million sq km)`) +
  aes(group = year) +
  geom_line() +
  aes(col = year) +
  scale_x_continuous(
    breaks = br,
    labels = c("Jan-01", "Apr-01", "Jul-01", "Oct-01", "Jan-01"),
    expand = c(0, 0)
  ) +
```

```
  scale_y_continuous(expand = c(0, 0), limits = c(0, 20)) +
  scale_color_continuous(
    guide = guide_colourbar(reverse = TRUE),
    breaks = seq(2010, 1980, -10)
  ) +
  geom_line(aes(lty = factor(year)),
    data = df %>% filter(year == 2016 | year == 1982),
    col = "white"
  ) +
  scale_linetype_manual(
    name = "",
    values = c("dashed", "solid")
  ) +
  annotate(
    geom = "text", x = 11210, y = 15,
    label = str_wrap("For this period, 1982 had the highest calendar-year average extent of Arctic sea i
    col = "white",
    size = 7
  ) +
  labs(x = "") +
  labs(y = "extent (million sq km)") +
  labs(col = "") +
  labs(lty = "") +
  labs(title = "Freezing cycles: Arctic sea ice extent, 1979-2017") +
  labs(subtitle = "Data Source: National Snow & Ice Data Center | Vis: Gina Reynolds for #MakeoverMonday
  theme_dark(base_size = 14) +
  theme(
    legend.background = element_blank(),
    legend.position = c(0.1, .35),
    legend.text = element_text(colour = "white", size = 15),
    plot.background = element_rect(fill = "grey30"),
    plot.title = element_text(colour = "lightgrey"),
    plot.subtitle = element_text(colour = "lightgrey"),
    axis.title = element_text(colour = "lightgrey"),
    axis.line = element_line(colour = "lightgrey"),
    axis.text = element_text(colour = "lightgrey"),
    axis.ticks = element_line(colour = "lightgrey")
  )
```


Freezing cycles: Arctic sea ice extent, 1979−2017
Data Source: National Snow & Ice Data Center | Vis: Gina Reynolds for #MakeoverMonday
For this period, 1982 had the highest calendar−year average extent of Arctic sea ice while 2016 had the lowest
extent (million sq km)
20
15
10
5
0
1980
1990
2000
2010
1982
2016
Jan−01
Apr−01
Jul−01
Oct−01