More in Visualization

Data Visualization and Data Transformation

Carley Dziewicki

January 26, 2019

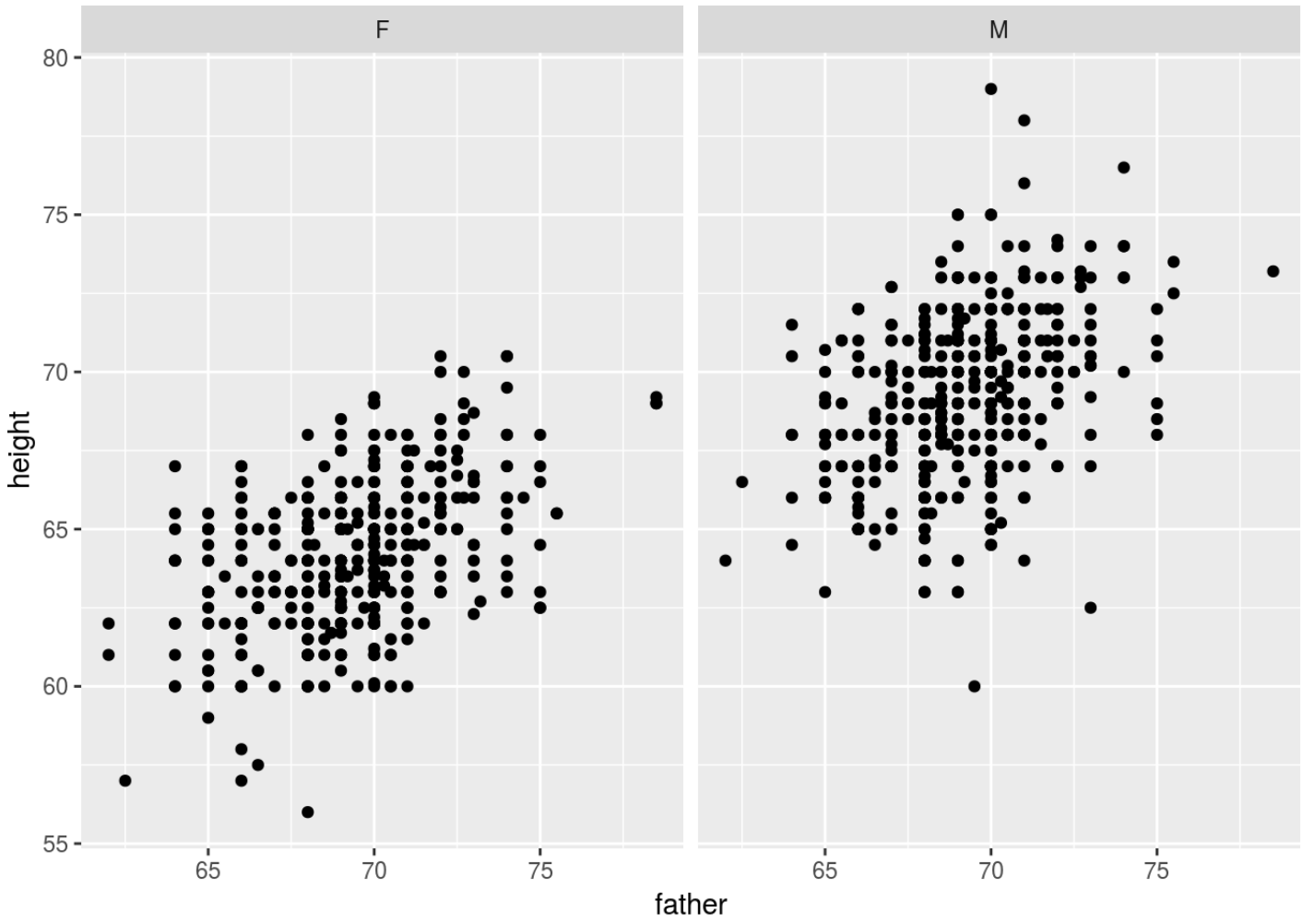
1. **Regression to the Mean**

The phenomenon known as “regression to the mean” was first identified by Sir Francis Galton in the late 19th century. The `Galton` data set from the `mosaicData` package contains Galton’s famous data. Use the `?Galton` command to get help about the data set after you have loaded the package `mosaicData`.

Create a scatterplot of each person’s `height` (y) against their father’s height (x) with the following characteristics:

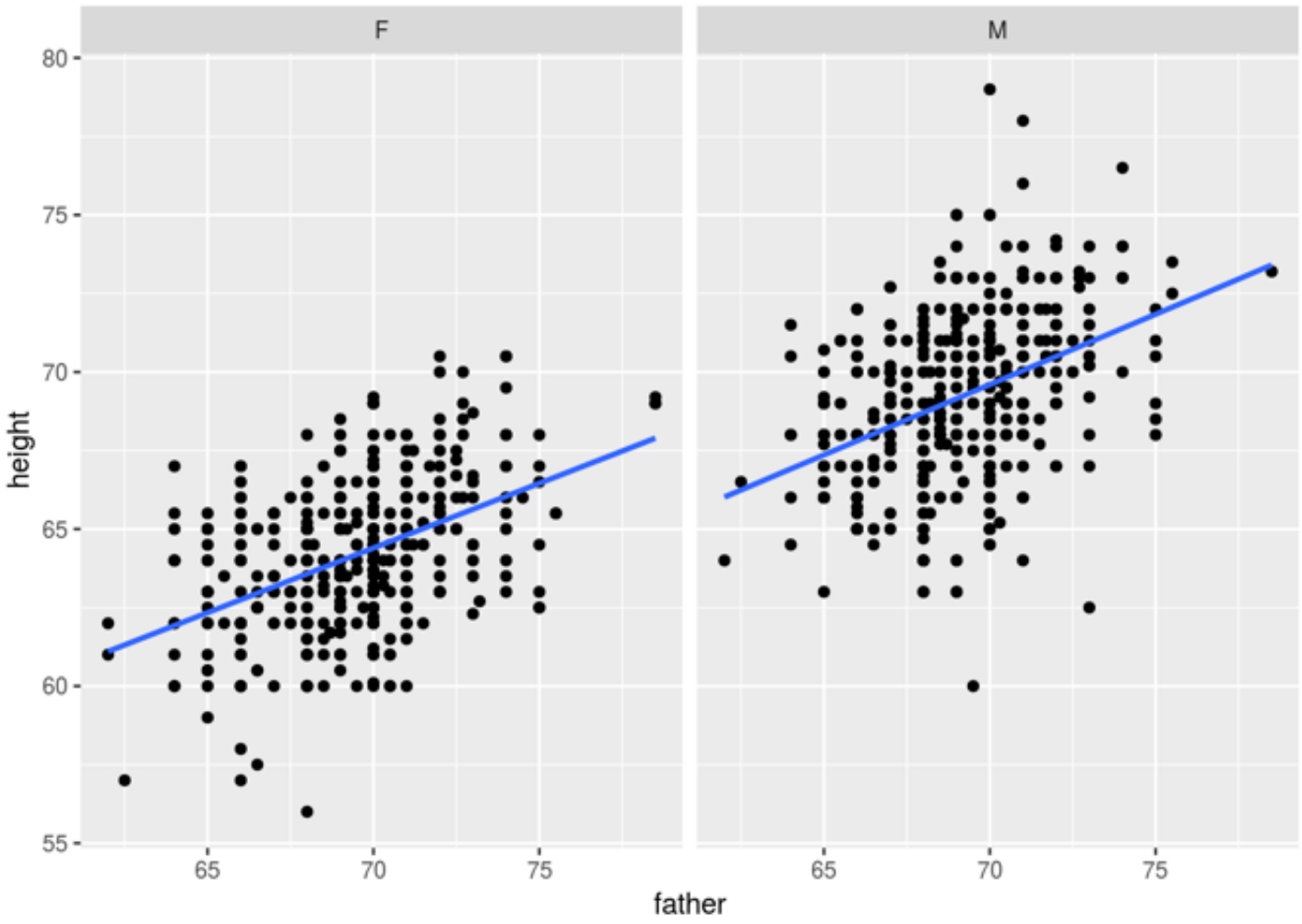
- Separate your plot into facets by `sex`.

```
ggplot(data=Galton) + geom_point(mapping=aes(x=father, y=height))+ facet_wrap(~sex)
```



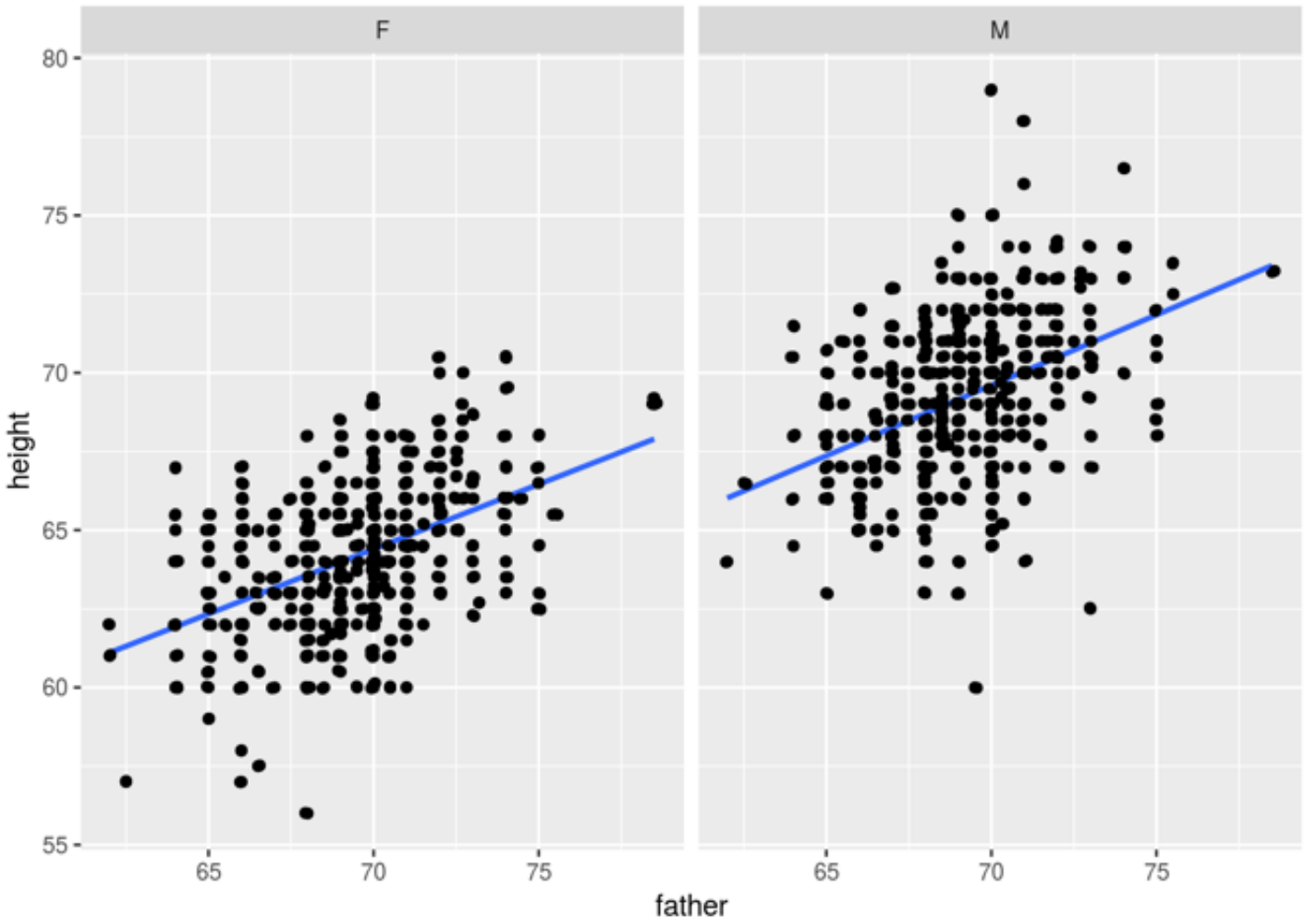
- Add a regression line to all of your facets, turning off the standard error shading.

```
ggplot(data=Galton) + geom_point(mapping=aes(x=father, y=height))+ facet_wrap(~sex) +
geom_smooth(method=lm, mapping=aes(x=father, y=height), se=FALSE)
```



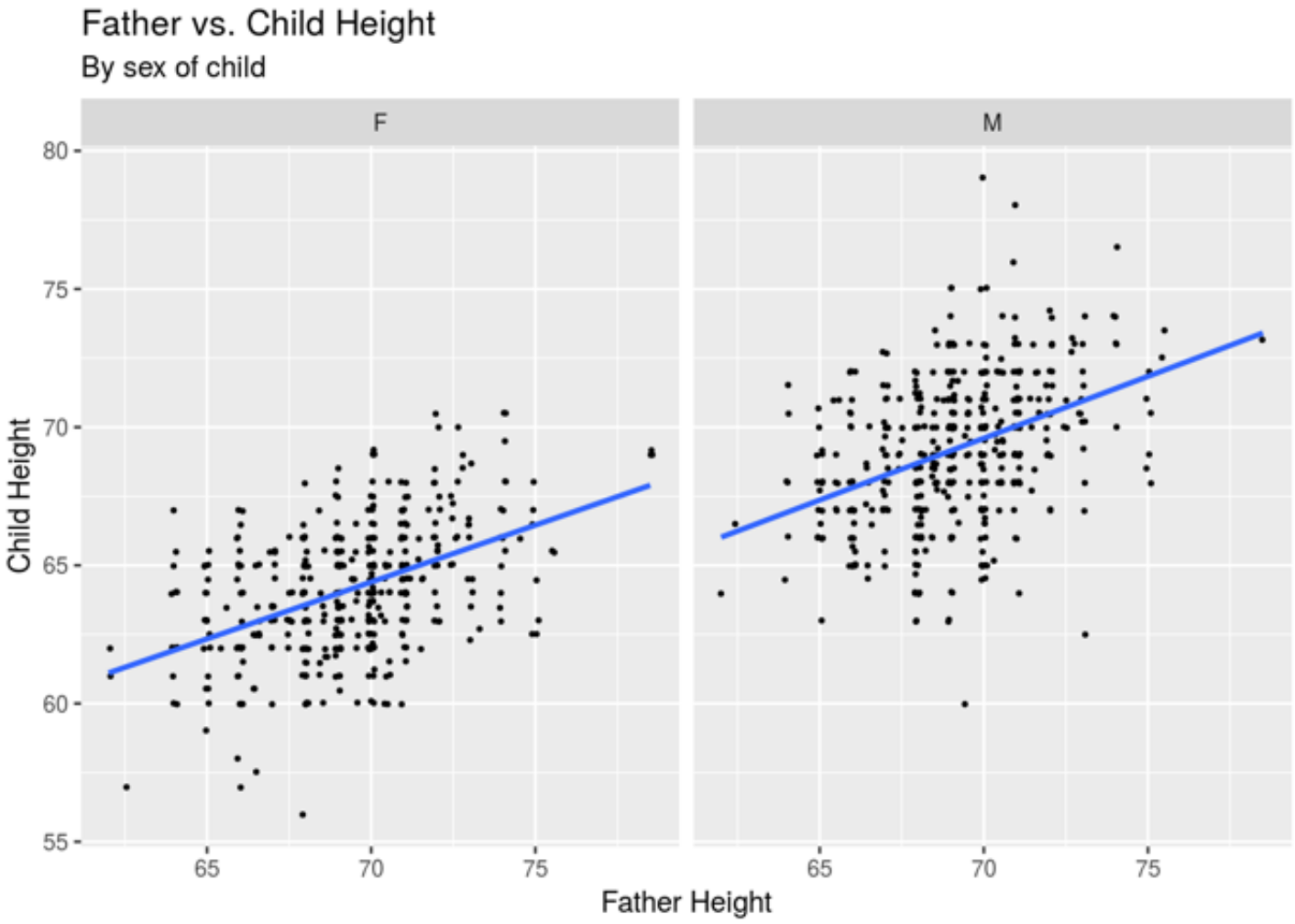
- If you notice, overplotting, add jittering.

```
ggplot(data=Galton) + geom_point(mapping=aes(x=father, y=height))+ facet_wrap(~sex) + geom_smooth(method=lm, mapping=aes(x=father, y=height), se=FALSE) +geom_jitter(mapping=aes(x=father, y=height))
```



- Make the points only half their normal size.

```
ggplot(data=Galton, aes(x=father, y=height)) + geom_point(position = position_jitter(
width=.1), size=.5)+ facet_wrap(~sex) + geom_smooth(method=lm, se=FALSE) +labs(title
= "Father vs. Child Height",subtitle = "By sex of child", x= "Father Height", y= "Chi
ld Height")
```



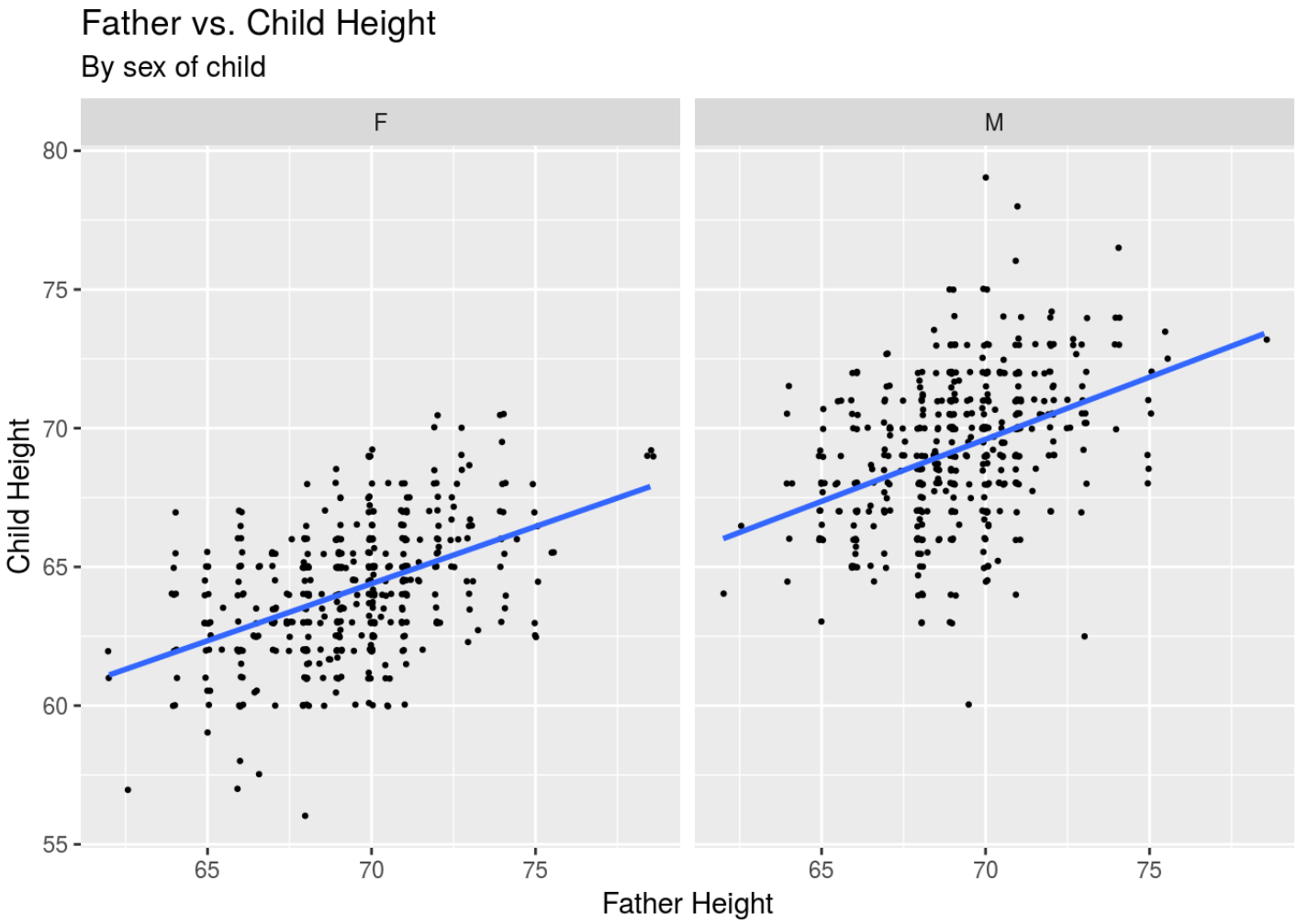
- Add a descriptive title to your plot and labels for the axes.

SOLUTION:

```
library(mosaicData)
glimpse(Galton)
```

```
## Observations: 898
## Variables: 6
## $ family <fct> 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, ...
## $ father <dbl> 78.5, 78.5, 78.5, 78.5, 75.5, 75.5, 75.5, 75.5, 75.0, 75.0, ...
## $ mother <dbl> 67.0, 67.0, 67.0, 67.0, 66.5, 66.5, 66.5, 66.5, 64.0, 64.0, ...
## $ sex    <fct> M, F, F, F, M, M, F, F, M, F, M, M, F, F, F, M, M, M, F, F, ...
## $ height <dbl> 73.2, 69.2, 69.0, 69.0, 73.5, 72.5, 65.5, 65.5, 71.0, 68.0, ...
## $ nkids  <int> 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, ...
```

```
ggplot(data=Galton, aes(x=father, y=height)) + geom_point(position = position_jitter(
width=.1), size=.5)+ facet_wrap(~sex) + geom_smooth(method=lm, se=FALSE) +labs(title
= "Father vs. Child Height",subtitle = "By sex of child", x= "Father Height", y= "Chi
ld Height")
```



2. **Baby Name Trends**

The R library `babynames` provides information on the historical incidence of baby names in the U.S. since 1880 as provided by the Social Security Administration. Load the `babynames` library and study the help information (hint: use `?babynames`). You may also wish to use the `View()` command to get familiar with the data. This question will require both data transformation and plotting skills. (Note: `babynames` has almost 2 million observations so don't print it!)

Complete the following:
* Identify the top 10 names given to males in 2017 and the top 10 names given to females in 2017.

```
library(babynames)
glimpse(babynames)
```

```
## Observations: 1,924,665
## Variables: 5
## $ year <dbl> 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 1880, 18...
## $ sex  <chr> "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F...
## $ name <chr> "Mary", "Anna", "Emma", "Elizabeth", "Minnie", "Margaret", "Id...
## $ n    <int> 7065, 2604, 2003, 1939, 1746, 1578, 1472, 1414, 1320, 1288, 12...
## $ prop <dbl> 0.07238359, 0.02667896, 0.02052149, 0.01986579, 0.01788843, 0....
```

```
babynames_f<-filter(babynames, sex== "F", year == "2017")
babynames_m<-filter(babynames, sex== "M", year== "2017")
arrange(babynames_f, desc(prop) )%>%
  select(name, n, prop) %>%
  top_n(10)
```

```
## Selecting by prop
```

```
## # A tibble: 10 x 3
##    name          n    prop
##    <chr>     <int>   <dbl>
##  1 Emma      19738 0.0105
##  2 Olivia    18632 0.00994
##  3 Ava       15902 0.00848
##  4 Isabella  15100 0.00805
##  5 Sophia    14831 0.00791
##  6 Mia       13437 0.00717
##  7 Charlotte 12893 0.00688
##  8 Amelia    11800 0.00629
##  9 Evelyn    10675 0.00569
## 10 Abigail   10551 0.00563
```

```
arrange(babynames_m, desc(prop)) %>%
  select (name, n, prop) %>%
  top_n(10)
```

```
## Selecting by prop
```

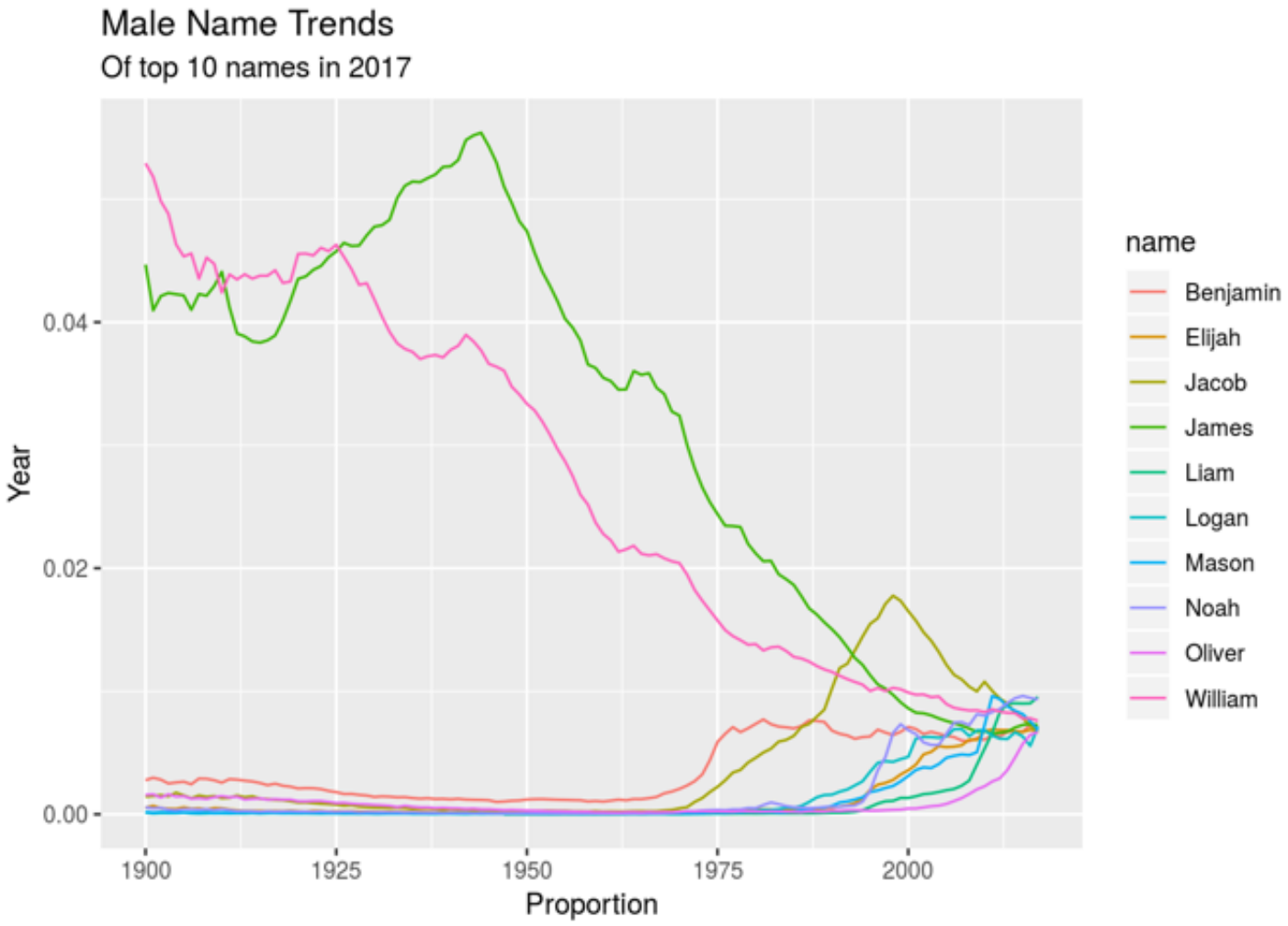
```
## # A tibble: 10 x 3
##    name         n    prop
##    <chr>    <int>   <dbl>
##  1 Liam     18728 0.00954
##  2 Noah     18326 0.00933
##  3 William  14904 0.00759
##  4 James    14232 0.00725
##  5 Logan    13974 0.00712
##  6 Benjamin 13733 0.00699
##  7 Mason    13502 0.00688
##  8 Elijah   13268 0.00676
##  9 Oliver   13141 0.00669
## 10 Jacob    13106 0.00668
```

- Generate two ggplots with the following. As always, put a title on each plot:
    - Plot 1: A line plot of the reported proportion of babies born with each of the top 10 female names. Your plot should start with the year 1900 and should have different colors for each name.

```
topmale<- filter(babynames, name %in% c("Liam","Noah", "William", "James", "Logan", "
Benjamin", "Mason","Elijah","Oliver","Jacob"), year >=1900, sex == "M")
topmale
```

```
## # A tibble: 1,128 x 5
##     year sex   name          n     prop
##    <dbl> <chr> <chr>     <int>    <dbl>
##  1  1900 M     William    8579 0.0529
##  2  1900 M     James      7245 0.0447
##  3  1900 M     Benjamin    450 0.00278
##  4  1900 M     Oliver      256 0.00158
##  5  1900 M     Jacob       233 0.00144
##  6  1900 M     Noah         92 0.000567
##  7  1900 M     Elijah       86 0.000530
##  8  1900 M     Mason        32 0.000197
##  9  1900 M     Logan        22 0.000136
## 10  1901 M     William    5990 0.0518
## # ... with 1,118 more rows
```

```
ggplot(topmale) + geom_line(mapping=aes(x=year, y=prop, color=name)) + labs(title = "
Male Name Trends",subtitle = "Of top 10 names in 2017", x= "Proportion", y= "Year")
```

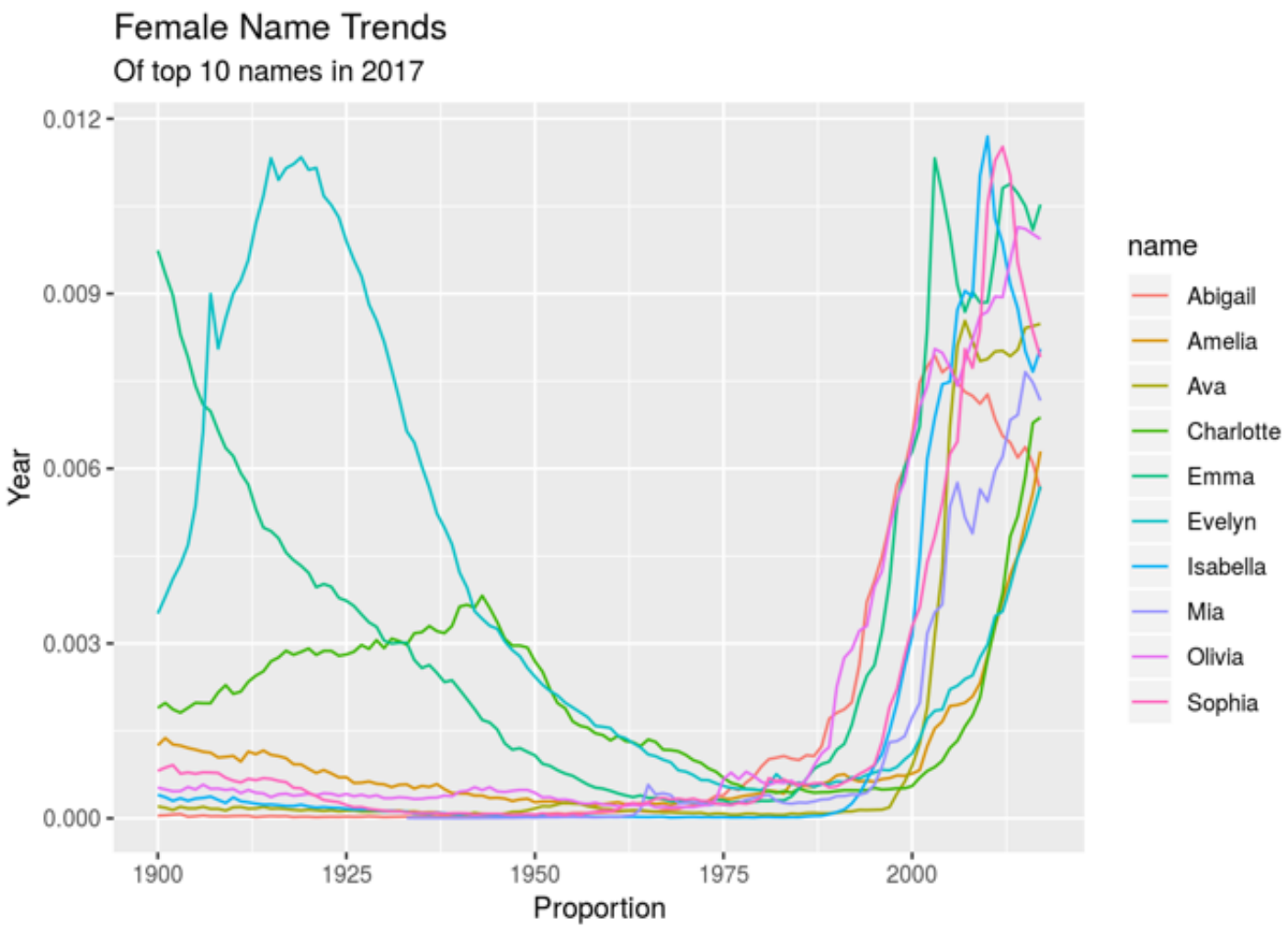


+ Plot 2: A line plot of the reported proportion of babies born with each of the top 10 male names. Your plot should start with the year 1900 and should have different colors for each name.

```
topfemale<- filter(babynames, name %in% c("Emma", "Olivia", "Ava", "Isabella", "Sophia", "Mia", "Charlotte", "Amelia", "Evelyn", "Abigail"), year >=1900, sex == "F")
topfemale
```

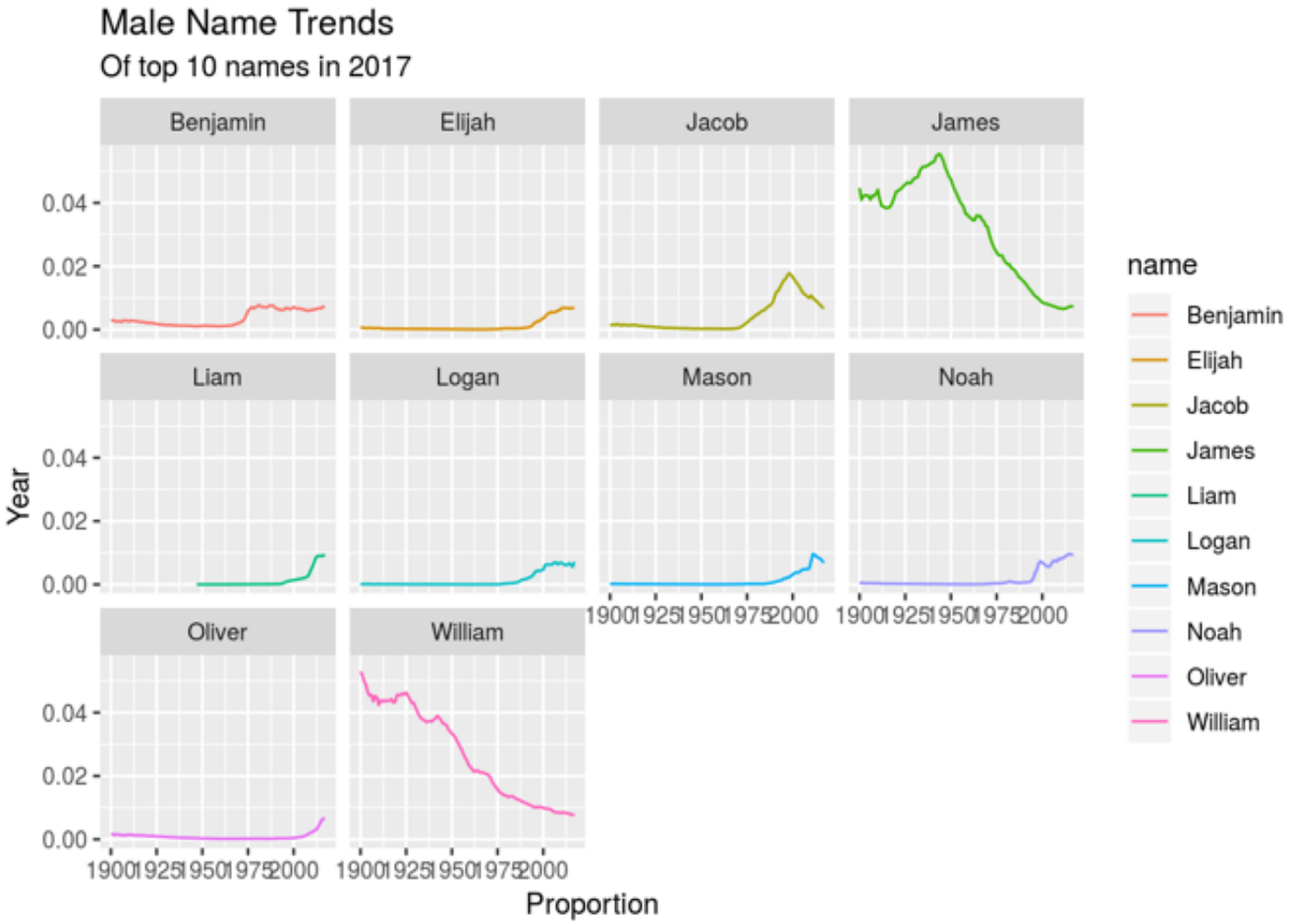
```
## # A tibble: 1,140 x 5
##     year sex   name           n      prop
##    <dbl> <chr> <chr>      <int>     <dbl>
##  1  1900 F     Emma        3095 0.00974
##  2  1900 F     Evelyn      1116 0.00351
##  3  1900 F     Charlotte    602 0.00189
##  4  1900 F     Amelia       398 0.00125
##  5  1900 F     Sophia       259 0.000815
##  6  1900 F     Olivia       167 0.000526
##  7  1900 F     Isabella     128 0.000403
##  8  1900 F     Ava           65 0.000205
##  9  1900 F     Abigail       14 0.0000441
## 10  1901 F     Emma        2374 0.00934
## # ... with 1,130 more rows
```

```
ggplot(topfemale) + geom_line(mapping=aes(x=year, y=prop, color=name)) + labs(title = "Female Name Trends",subtitle = "Of top 10 names in 2017", x= "Proportion", y= "Year" )
```

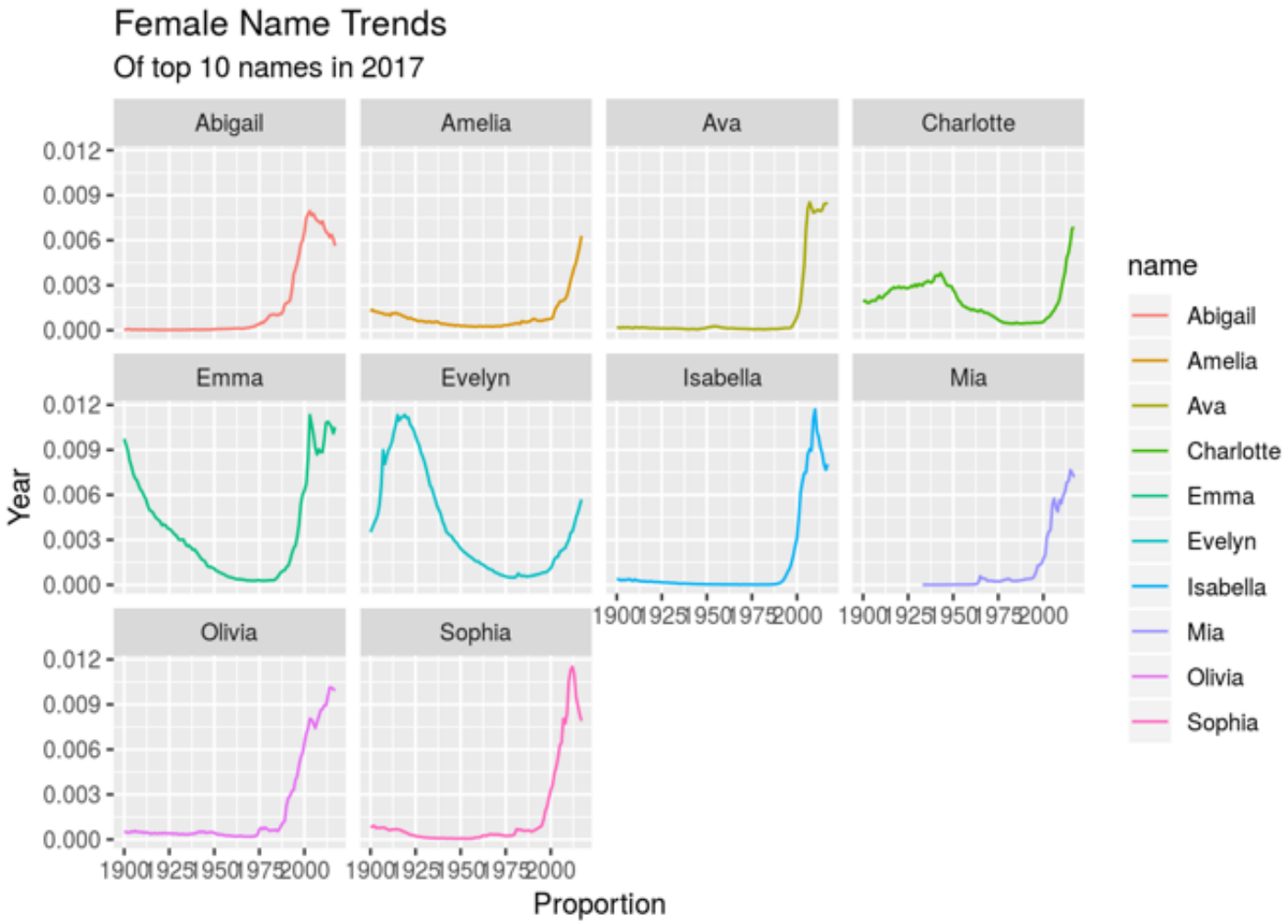


- Now introduce faceting by name so that you have 2 sets of 10 plots, one for each name.

```
ggplot(topmale) + geom_line(mapping=aes(x=year, y=prop, color=name)) + facet_wrap(~name)+ labs(title = "Male Name Trends",subtitle = "Of top 10 names in 2017", x= "Proportion", y= "Year")
```




```
ggplot(topfemale) + geom_line(mapping=aes(x=year, y=prop, color=name)) + facet_wrap(~name) + labs(title = "Female Name Trends",subtitle = "Of top 10 names in 2017", x= "Proportion", y= "Year")
```

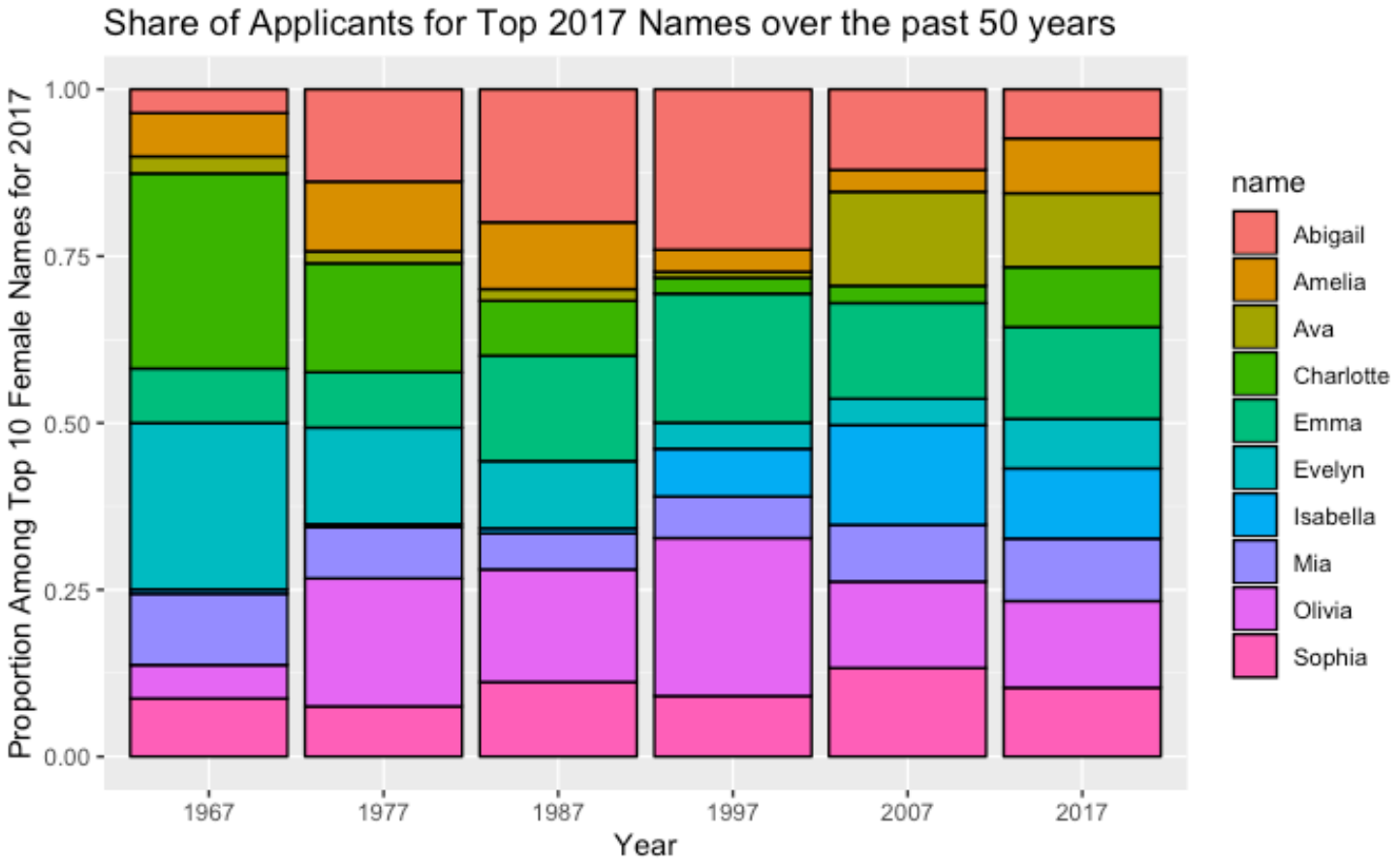


- Write a few observations including identifying the names that were popular in the past as well as those that are newly popular.

Observations: Names that were popular in the past for males were William and James, but are both no longer as popular as the new names of Jacob, Liam and Noah. As for females names popular in the past were Evelyn and Emma, funny enouguh these names became unpopular for quite a while and are now newly popular again along with other names such as Isabella and Ava.

3. **Baby Name Trends (cont'd)**

Use the **babynames** data to create two versions of the chart shown in the figure below: one as shown for Females and another for Males. (Hint: This is an important chance to learn about the difference between `year` and `factor(year)`). Be sure to also write a couple sentences commenting on what can be seen about name trends but these two plots.

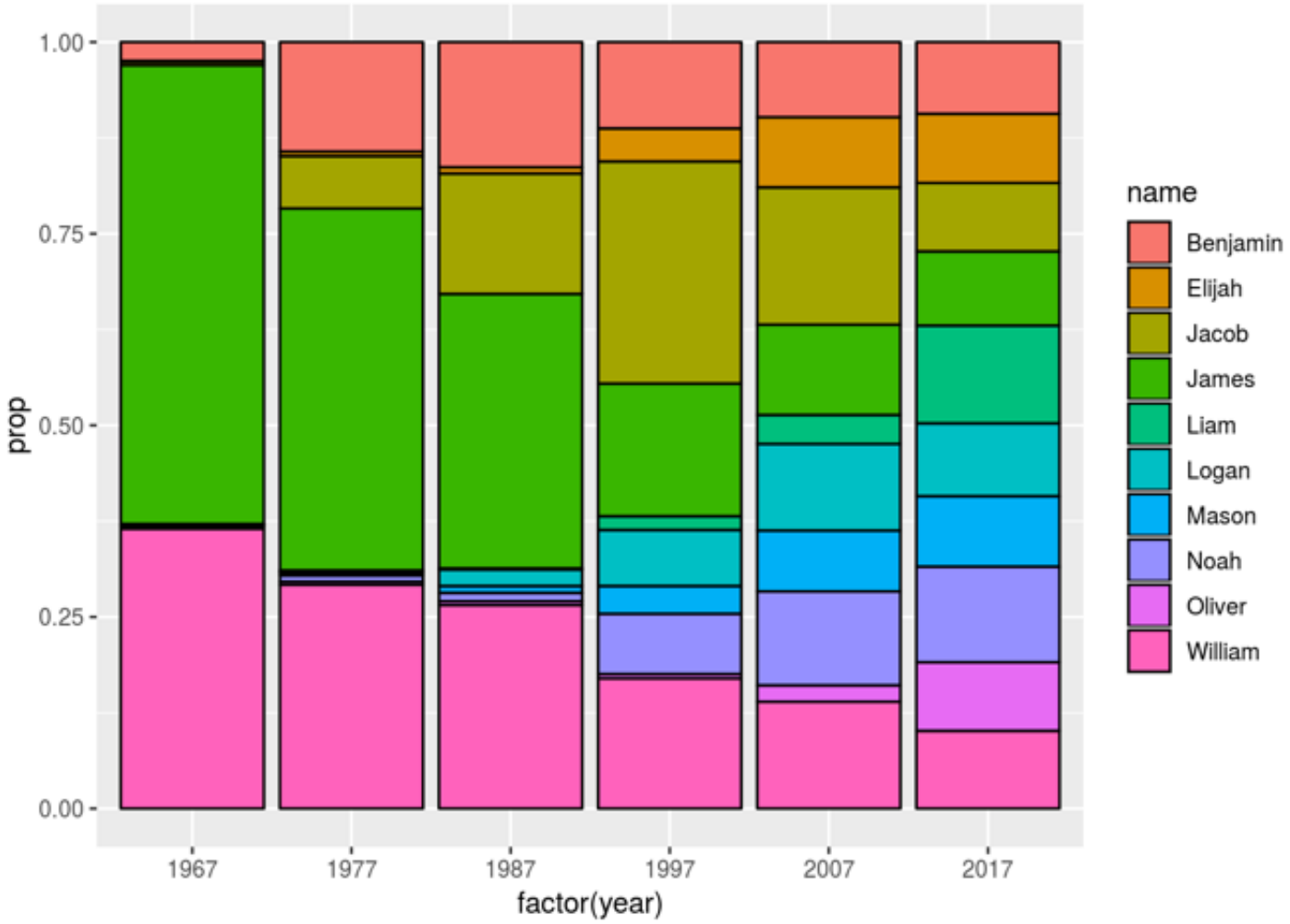


Bar chart

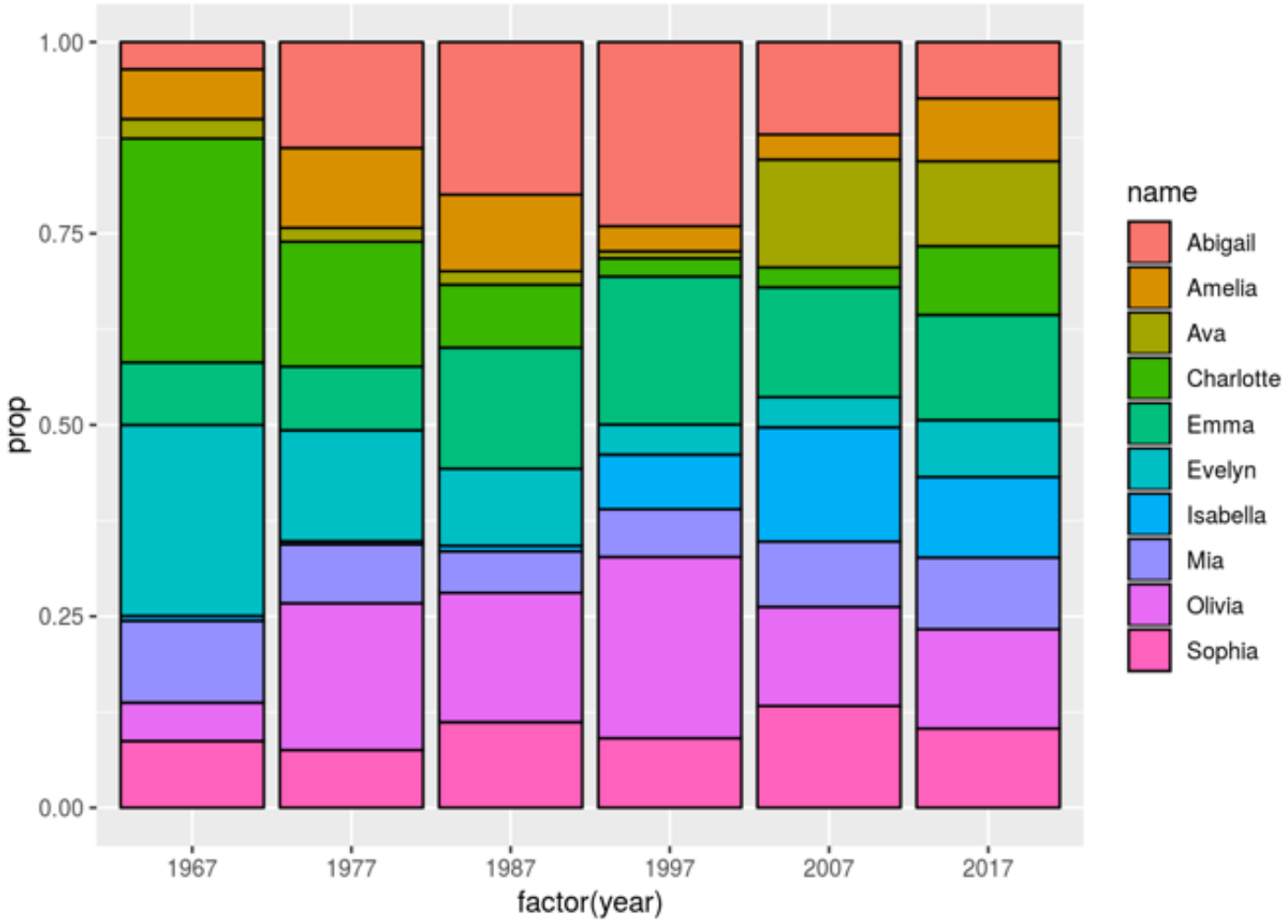
```
topmale
```

```
## # A tibble: 1,128 x 5
##     year sex   name          n     prop
##    <dbl> <chr> <chr>     <int>    <dbl>
##  1  1900 M     William    8579 0.0529
##  2  1900 M     James      7245 0.0447
##  3  1900 M     Benjamin    450 0.00278
##  4  1900 M     Oliver      256 0.00158
##  5  1900 M     Jacob       233 0.00144
##  6  1900 M     Noah         92 0.000567
##  7  1900 M     Elijah       86 0.000530
##  8  1900 M     Mason        32 0.000197
##  9  1900 M     Logan        22 0.000136
## 10  1901 M     William    5990 0.0518
## # ... with 1,118 more rows
```

```
topMaleinc<-filter(topmale, year %in% seq(1967,2017,10))
#topMaleinc<- filter(topmale, year == select(topmale, year, ends_with("7")))
ggplot(topMaleinc, aes(fill=name, y=prop, x=factor(year))) +
    geom_bar(position="fill", stat="identity", color="black")
```




```
topFemaleinc<-filter(topfemale, year %in% seq(1967,2017,10))
#topMaleinc<- filter(topmale, year == select(topmale, year, ends_with("7")))
ggplot(topFemaleinc, aes(fill=name, y=prop, x=factor(year))) +
    geom_bar(position="fill", stat="identity", color="black")
```



4. **Fuel Economy Data**

According to the help pages, the **fueleconomy** package contains "fuel economy data from the EPA, 1985-2015. This dataset contains selected variables, and removes vehicles with incomplete data (e.g. no drive train data)." The `cardata` created in the chunk below restricts this dataset to more common vehicles that were made in at least 10 years. To find out more about the variables.

Use separate `filter()` commands to find all vehicles in `cardata` that:

a. Get between 40 and 50 mpg (inclusive) on the highway.

```
library(fueleconomy)
cardata <- left_join(common, vehicles)
```

```
## Joining, by = c("make", "model")
```

```
glimpse(cardata)
```

```
## Observations: 14,531
## Variables: 14
## $ make  <chr> "Acura", "Acura", "Acura", "Acura", "Acura", "Acura", "Acura"...
## $ model <chr> "Integra", "Integra", "Integra", "Integra", "Integra", "Integ...
## $ n     <int> 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 42, 4...
## $ years <int> 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16, 1...
## $ id    <int> 1833, 1834, 3037, 3038, 4183, 4184, 5303, 5304, 6442, 6443, 7...
## $ year  <int> 1986, 1986, 1987, 1987, 1988, 1988, 1989, 1989, 1990, 1990, 1...
## $ class <chr> "Subcompact Cars", "Subcompact Cars", "Subcompact Cars", "Sub...
## $ trans <chr> "Automatic 4-spd", "Manual 5-spd", "Automatic 4-spd", "Manual...
## $ drive <chr> "Front-Wheel Drive", "Front-Wheel Drive", "Front-Wheel Drive"...
## $ cyl   <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4...
## $ displ <dbl> 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.6, 1.8, 1.8, 1.8, 1.8, 1...
## $ fuel  <chr> "Regular", "Regular", "Regular", "Regular", "Regular", "Regul...
## $ hwy   <int> 28, 28, 28, 28, 27, 28, 27, 28, 24, 26, 26, 26, 26, 26, 28, 2...
## $ cty   <int> 22, 23, 22, 23, 22, 23, 22, 23, 20, 21, 21, 21, 21, 21, 22, 2...
```

```
fourtytofifty<-filter(cardata, hwy<=50, hwy>=40)
fourtytofifty
```

```
## # A tibble: 175 x 14
## # Groups:   make [10]
##    make  model     n years    id  year class trans drive   cyl displ fuel    hwy
##    <chr> <chr> <int> <int> <int> <int> <chr> <chr> <chr> <int> <dbl> <chr> <int>
##  1 Chev… Spri…    20    10    37  1985 Mini… Manu… Fron…     3     1 Regu…    47
##  2 Chev… Spri…    20    10  1729  1986 Mini… Manu… Fron…     3     1 Regu…    45
##  3 Chev… Spri…    20    10  2992  1987 Subc… Manu… Fron…     3     1 Regu…    43
##  4 Chev… Spri…    20    10  4219  1988 Subc… Manu… Fron…     3     1 Regu…    43
##  5 Chev… Spri…    20    10  5336  1989 Subc… Manu… Fron…     3     1 Regu…    45
##  6 Chev… Spri…    20    10  6466  1990 Subc… Manu… Fron…     3     1 Regu…    45
##  7 Chev… Spri…    20    10  7554  1991 Subc… Manu… Fron…     3     1 Regu…    45
##  8 Chev… Spri…    20    10  8644  1992 Subc… Manu… Fron…     3     1 Regu…    44
##  9 Chev… Spri…    20    10 28977  1993 Subc… Manu… Fron…     3     1 Regu…    44
## 10 Chev… Spri…    20    10 10697  1994 Subc… Manu… Fron…     3     1 Regu…    45
## # ... with 165 more rows, and 1 more variable: cty <int>
```

b. Are made by either Chevrolet or Ford and were made after the year 2000?

```
ChevFord<-filter(cardata, make== "Chevrolet"| make== "Ford", year>=2000)
ChevFord
```

```
## # A tibble: 1,013 x 14
## # Groups:   make [2]
##    make  model     n years    id  year class trans drive   cyl displ fuel    hwy
##    <chr> <chr> <int> <int> <int> <int> <chr> <chr> <chr> <int> <dbl> <chr> <int>
##  1 Chev… Astr…    50    21 16176  2000 Vans… Auto… Rear…     6   4.3 Regu…    20
##  2 Chev… Astr…    50    21 17047  2001 Vans… Auto… Rear…     6   4.3 Regu…    20
##  3 Chev… Astr…    50    21 17983  2002 Vans… Auto… Rear…     6   4.3 Regu…    20
##  4 Chev… Astr…    50    21 18953  2003 Vans… Auto… Rear…     6   4.3 Regu…    21
##  5 Chev… Astr…    50    21 20086  2004 Vans… Auto… Rear…     6   4.3 Regu…    19
##  6 Chev… Astr…    50    21 21188  2005 Vans… Auto… Rear…     6   4.3 Regu…    20
##  7 Chev… Astr…    41    21 16196  2000 Vans… Auto… Rear…     6   4.3 Regu…    19
##  8 Chev… Astr…    41    21 17068  2001 Vans… Auto… Rear…     6   4.3 Regu…    20
##  9 Chev… Astr…    41    21 18002  2002 Vans… Auto… Rear…     6   4.3 Regu…    18
## 10 Chev… Astr…    41    21 18985  2003 Vans… Auto… Rear…     6   4.3 Regu…    19
## # ... with 1,003 more rows, and 1 more variable: cty <int>
```

c. Were made in a year that is a multiple of 5 (don't list all such years) and have Manual transmission. (Hint: This is a great opportunity to learn about the `str_detect()` function that is part of the Tidyverse. If you Google this, don't use a version of the `grep` command.)

```
fiveyrs<- filter(cardata, year %in% seq(2005,2015,5))
fiveyrs
```

```
## # A tibble: 930 x 14
## # Groups:   make [38]
##    make  model     n years    id  year class trans drive   cyl displ fuel    hwy
##    <chr> <chr> <int> <int> <int> <int> <chr> <chr> <chr> <int> <dbl> <chr> <int>
##  1 Acura MDX …    12    12 21351  2005 Spor… Auto… 4-Wh…     6   3.5 Prem…    21
##  2 Acura MDX …    12    12 29797  2010 Spor… Auto… All-…     6   3.7 Prem…    21
##  3 Acura NSX      28    14 20451  2005 Two … Auto… Rear…     6   3   Prem…    22
##  4 Acura NSX      28    14 20452  2005 Two … Manu… Rear…     6   3.2 Prem…    22
##  5 Acura TSX      27    11 20657  2005 Comp… Manu… Fron…     4   2.4 Prem…    27
##  6 Acura TSX      27    11 20658  2005 Comp… Auto… Fron…     4   2.4 Prem…    28
##  7 Acura TSX      27    11 28571  2010 Comp… Manu… Fron…     4   2.4 Prem…    28
##  8 Acura TSX      27    11 28572  2010 Comp… Auto… Fron…     4   2.4 Prem…    30
##  9 Acura TSX      27    11 28573  2010 Comp… Auto… Fron…     6   3.5 Prem…    27
## 10 Audi  A4       49    19 20659  2005 Comp… Auto… Fron…     4   1.8 Prem…    27
## # ... with 920 more rows, and 1 more variable: cty <int>
```

d. Had lower highway mileage than city mileage? Do all these vehicles have anything else in common?

```
betterhwy<- filter(cardata, hwy<=cty)
betterhwy
```

```
## # A tibble: 46 x 14
## # Groups:   make [8]
##    make  model     n years    id  year class trans drive   cyl displ fuel    hwy
##    <chr> <chr> <int> <int> <int> <int> <chr> <chr> <chr> <int> <dbl> <chr> <int>
##  1 Chev… G10/…    96    12  6158  1989 Vans  Auto… Rear…     8   5   Regu…    12
##  2 Dodge B350…    36    11 27482  1984 Vans… Auto… 2-Wh…     8   5.2 Regu…    11
##  3 Dodge Cara…    88    18 30973  1999 Mini… Auto… 2-Wh…    NA  NA   Elec…    33
##  4 Dodge D250…    73    10 27341  1984 Stan… Auto… 2-Wh…     8   5.2 Regu…    11
##  5 Ford  Bron…    86    13 28476  1984 Spec… Auto… 4-Wh…     8   5.8 Regu…    10
##  6 Ford  Bron…    86    13  1099  1985 Spec… Auto… 4-Wh…     6   4.9 Regu…    13
##  7 Ford  Bron…    86    13  2814  1986 Spec… Auto… 4-Wh…     6   4.9 Regu…    14
##  8 Ford  Bron…    86    13  4047  1987 Spec… Auto… 4-Wh…     6   4.9 Regu…    13
##  9 Ford  E150…    77    22  1608  1985 Vans  Auto… Rear…     8   5.8 Regu…    10
## 10 Ford  E150…   105    23 27491  1984 Vans… Auto… 2-Wh…     8   5.8 Regu…    11
## # ... with 36 more rows, and 1 more variable: cty <int>
```

e. Were missing data for drivetrain.

```
filter(cardata, is.na(drive))
```

```
## # A tibble: 0 x 14
## # Groups:   make [0]
## # ... with 14 variables: make <chr>, model <chr>, n <int>, years <int>,
## #   id <int>, year <int>, class <chr>, trans <chr>, drive <chr>, cyl <int>,
## #   displ <dbl>, fuel <chr>, hwy <int>, cty <int>
```

5. **National Health and Nutrition Examination Data (NHANES)**

The US National Center for Health Statistics (NCHS) has conducted a series of health and nutrition surveys since the early 1960's. Since 1999 approximately 5,000 individuals of all ages are interviewed in their homes every year and complete the health examination component of the survey. Data from two survey years is contained in the `NHANES` data set provided with the **NHANES** package. Note: The **NHANES* library provides two data sets: `NHANES` and `NHANESraw`. Be sure to use just the `NHANES` data set for this problem.

Use the `NHANES` data set to carry out the following separately:

a. Use `select()` to reorganize the columns as follows: the first 5 variables, followed by all the blood pressure measurements, and then all the other variables.

```
library(NHANES)
glimpse(NHANES)
```

```
## Observations: 10,000
## Variables: 76
## $ ID               <int> 51624, 51624, 51624, 51625, 51630, 51638, 51646, 5...
```

```
## $ SurveyYr         <fct> 2009_10, 2009_10, 2009_10, 2009_10, 2009_10, 2009_...
## $ Gender           <fct> male, male, male, male, female, male, male, female...
## $ Age              <int> 34, 34, 34, 4, 49, 9, 8, 45, 45, 45, 66, 58, 54, 1...
## $ AgeDecade        <fct>  30-39,  30-39,  30-39,  0-9,  40-49,  0-9,  0-9, ...
## $ AgeMonths        <int> 409, 409, 409, 49, 596, 115, 101, 541, 541, 541, 7...
## $ Race1            <fct> White, White, White, Other, White, White, White, W...
## $ Race3            <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ Education        <fct> High School, High School, High School, NA, Some Co...
## $ MaritalStatus    <fct> Married, Married, Married, NA, LivePartner, NA, NA...
## $ HHIncome         <fct> 25000-34999, 25000-34999, 25000-34999, 20000-24999...
## $ HHIncomeMid      <int> 30000, 30000, 30000, 22500, 40000, 87500, 60000, 8...
## $ Poverty          <dbl> 1.36, 1.36, 1.36, 1.07, 1.91, 1.84, 2.33, 5.00, 5....
## $ HomeRooms        <int> 6, 6, 6, 9, 5, 6, 7, 6, 6, 6, 5, 10, 6, 10, 10, 4,...
## $ HomeOwn          <fct> Own, Own, Own, Own, Rent, Rent, Own, Own, Own, Own...
## $ Work             <fct> NotWorking, NotWorking, NotWorking, NA, NotWorking...
## $ Weight           <dbl> 87.4, 87.4, 87.4, 17.0, 86.7, 29.8, 35.2, 75.7, 75...
## $ Length           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ HeadCirc         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ Height           <dbl> 164.7, 164.7, 164.7, 105.4, 168.4, 133.1, 130.6, 1...
## $ BMI              <dbl> 32.22, 32.22, 32.22, 15.30, 30.57, 16.82, 20.64, 2...
## $ BMICatUnder20yrs <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ BMI_WHO          <fct> 30.0_plus, 30.0_plus, 30.0_plus, 12.0_18.5, 30.0_p...
## $ Pulse            <int> 70, 70, 70, NA, 86, 82, 72, 62, 62, 62, 60, 62, 76...
## $ BPSysAve         <int> 113, 113, 113, NA, 112, 86, 107, 118, 118, 118, 11...
## $ BPDiaAve         <int> 85, 85, 85, NA, 75, 47, 37, 64, 64, 64, 63, 74, 85...
## $ BPSys1           <int> 114, 114, 114, NA, 118, 84, 114, 106, 106, 106, 12...
## $ BPDia1           <int> 88, 88, 88, NA, 82, 50, 46, 62, 62, 62, 64, 76, 86...
## $ BPSys2           <int> 114, 114, 114, NA, 108, 84, 108, 118, 118, 118, 10...
## $ BPDia2           <int> 88, 88, 88, NA, 74, 50, 36, 68, 68, 68, 62, 72, 88...
## $ BPSys3           <int> 112, 112, 112, NA, 116, 88, 106, 118, 118, 118, 11...
## $ BPDia3           <int> 82, 82, 82, NA, 76, 44, 38, 60, 60, 60, 64, 76, 82...
## $ Testosterone     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ DirectChol       <dbl> 1.29, 1.29, 1.29, NA, 1.16, 1.34, 1.55, 2.12, 2.12...
## $ TotChol          <dbl> 3.49, 3.49, 3.49, NA, 6.70, 4.86, 4.09, 5.82, 5.82...
## $ UrineVol1        <int> 352, 352, 352, NA, 77, 123, 238, 106, 106, 106, 11...
## $ UrineFlow1       <dbl> NA, NA, NA, NA, 0.094, 1.538, 1.322, 1.116, 1.116,...
## $ UrineVol2        <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ UrineFlow2       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ Diabetes         <fct> No, No, No, No, No, No, No, No, No, No, No, No, No...
## $ DiabetesAge      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ HealthGen        <fct> Good, Good, Good, NA, Good, NA, NA, Vgood, Vgood, ...
## $ DaysPhysHlthBad  <int> 0, 0, 0, NA, 0, NA, NA, 0, 0, 0, 10, 0, 4, NA, NA,...
## $ DaysMentHlthBad  <int> 15, 15, 15, NA, 10, NA, NA, 3, 3, 3, 0, 0, 0, NA, ...
## $ LittleInterest   <fct> Most, Most, Most, NA, Several, NA, NA, None, None,...
## $ Depressed        <fct> Several, Several, Several, NA, Several, NA, NA, No...
## $ nPregnancies     <int> NA, NA, NA, NA, 2, NA, NA, 1, 1, 1, NA, NA, NA, NA...
## $ nBabies          <int> NA, NA, NA, NA, 2, NA, NA, NA, NA, NA, NA, NA, NA,...
```

```
## $ Age1stBaby       <int> NA, NA, NA, NA, 27, NA, NA, NA, NA, NA, NA, NA, NA...
## $ SleepHrsNight    <int> 4, 4, 4, NA, 8, NA, NA, 8, 8, 8, 7, 5, 4, NA, 5, 7...
## $ SleepTrouble     <fct> Yes, Yes, Yes, NA, Yes, NA, NA, No, No, No, No, No...
## $ PhysActive       <fct> No, No, No, NA, No, NA, NA, Yes, Yes, Yes, Yes, Ye...
## $ PhysActiveDays   <int> NA, NA, NA, NA, NA, NA, NA, 5, 5, 5, 7, 5, 1, NA, ...
## $ TVHrsDay         <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ CompHrsDay       <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
## $ TVHrsDayChild    <int> NA, NA, NA, 4, NA, 5, 1, NA, NA, NA, NA, NA, NA, 4...
## $ CompHrsDayChild  <int> NA, NA, NA, 1, NA, 0, 6, NA, NA, NA, NA, NA, NA, 3...
## $ Alcohol12PlusYr  <fct> Yes, Yes, Yes, NA, Yes, NA, NA, Yes, Yes, Yes, Yes...
## $ AlcoholDay       <int> NA, NA, NA, NA, 2, NA, NA, 3, 3, 3, 1, 2, 6, NA, N...
## $ AlcoholYear      <int> 0, 0, 0, NA, 20, NA, NA, 52, 52, 52, 100, 104, 364...
## $ SmokeNow         <fct> No, No, No, NA, Yes, NA, NA, NA, NA, NA, No, NA, N...
## $ Smoke100         <fct> Yes, Yes, Yes, NA, Yes, NA, NA, No, No, No, Yes, N...
## $ Smoke100n        <fct> Smoker, Smoker, Smoker, NA, Smoker, NA, NA, Non-Sm...
## $ SmokeAge         <int> 18, 18, 18, NA, 38, NA, NA, NA, NA, NA, 13, NA, NA...
## $ Marijuana        <fct> Yes, Yes, Yes, NA, Yes, NA, NA, Yes, Yes, Yes, NA,...
## $ AgeFirstMarij    <int> 17, 17, 17, NA, 18, NA, NA, 13, 13, 13, NA, 19, 15...
## $ RegularMarij     <fct> No, No, No, NA, No, NA, NA, No, No, No, NA, Yes, Y...
## $ AgeRegMarij      <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 20, 15...
## $ HardDrugs        <fct> Yes, Yes, Yes, NA, Yes, NA, NA, No, No, No, No, Ye...
## $ SexEver          <fct> Yes, Yes, Yes, NA, Yes, NA, NA, Yes, Yes, Yes, Yes...
## $ SexAge           <int> 16, 16, 16, NA, 12, NA, NA, 13, 13, 13, 17, 22, 12...
## $ SexNumPartnLife  <int> 8, 8, 8, NA, 10, NA, NA, 20, 20, 20, 15, 7, 100, N...
## $ SexNumPartYear   <int> 1, 1, 1, NA, 1, NA, NA, 0, 0, 0, NA, 1, 1, NA, NA,...
## $ SameSex          <fct> No, No, No, NA, Yes, NA, NA, Yes, Yes, Yes, No, No...
## $ SexOrientation   <fct> Heterosexual, Heterosexual, Heterosexual, NA, Hete...
## $ PregnantNow      <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA...
```

```
# solution goes here
select(NHANES, 1:5, starts_with("BP"), everything())
```

```
## # A tibble: 10,000 x 76
##       ID SurveyYr Gender   Age AgeDecade BPSysAve BPDiaAve BPSys1 BPDia1 BPSys2
##    <int> <fct>    <fct>  <int> <fct>        <int>    <int>  <int>  <int>  <int>
##  1 51624 2009_10  male      34 " 30-39"       113       85    114     88    114
##  2 51624 2009_10  male      34 " 30-39"       113       85    114     88    114
##  3 51624 2009_10  male      34 " 30-39"       113       85    114     88    114
##  4 51625 2009_10  male       4 " 0-9"          NA       NA     NA     NA     NA
##  5 51630 2009_10  female    49 " 40-49"       112       75    118     82    108
##  6 51638 2009_10  male       9 " 0-9"          86       47     84     50     84
##  7 51646 2009_10  male       8 " 0-9"         107       37    114     46    108
##  8 51647 2009_10  female    45 " 40-49"       118       64    106     62    118
##  9 51647 2009_10  female    45 " 40-49"       118       64    106     62    118
## 10 51647 2009_10  female    45 " 40-49"       118       64    106     62    118
## # ... with 9,990 more rows, and 66 more variables: BPDia2 <int>, BPSys3 <int>,
## #   BPDia3 <int>, AgeMonths <int>, Race1 <fct>, Race3 <fct>, Education <fct>,
## #   MaritalStatus <fct>, HHIncome <fct>, HHIncomeMid <int>, Poverty <dbl>,
## #   HomeRooms <int>, HomeOwn <fct>, Work <fct>, Weight <dbl>, Length <dbl>,
## #   HeadCirc <dbl>, Height <dbl>, BMI <dbl>, BMICatUnder20yrs <fct>,
## #   BMI_WHO <fct>, Pulse <int>, Testosterone <dbl>, DirectChol <dbl>,
## #   TotChol <dbl>, UrineVol1 <int>, UrineFlow1 <dbl>, UrineVol2 <int>,
## #   UrineFlow2 <dbl>, Diabetes <fct>, DiabetesAge <int>, HealthGen <fct>,
## #   DaysPhysHlthBad <int>, DaysMentHlthBad <int>, LittleInterest <fct>,
## #   Depressed <fct>, nPregnancies <int>, nBabies <int>, Age1stBaby <int>,
## #   SleepHrsNight <int>, SleepTrouble <fct>, PhysActive <fct>,
## #   PhysActiveDays <int>, TVHrsDay <fct>, CompHrsDay <fct>,
## #   TVHrsDayChild <int>, CompHrsDayChild <int>, Alcohol12PlusYr <fct>,
## #   AlcoholDay <int>, AlcoholYear <int>, SmokeNow <fct>, Smoke100 <fct>,
## #   Smoke100n <fct>, SmokeAge <int>, Marijuana <fct>, AgeFirstMarij <int>,
## #   RegularMarij <fct>, AgeRegMarij <int>, HardDrugs <fct>, SexEver <fct>,
## #   SexAge <int>, SexNumPartnLife <int>, SexNumPartYear <int>, SameSex <fct>,
## #   SexOrientation <fct>, PregnantNow <fct>
```

b. Create three new variables: ratio of systolic to diastolic blood pressure (using the average BP measures), height in inches and weight in pounds.

```
mutate(NHANES, BPsystodia= BPSysAve/BPDiaAve, height_in=Height*.3937, weight_lb= Weig
ht*2.2046)
```

```
## # A tibble: 10,000 x 79
##       ID SurveyYr Gender   Age AgeDecade AgeMonths Race1 Race3 Education
##    <int> <fct>    <fct>  <int> <fct>         <int> <fct> <fct> <fct>
##  1 51624 2009_10  male      34 " 30-39"        409 White <NA>  High Sch…
##  2 51624 2009_10  male      34 " 30-39"        409 White <NA>  High Sch…
##  3 51624 2009_10  male      34 " 30-39"        409 White <NA>  High Sch…
##  4 51625 2009_10  male       4 " 0-9"           49 Other <NA>  <NA>
##  5 51630 2009_10  female    49 " 40-49"        596 White <NA>  Some Col…
##  6 51638 2009_10  male       9 " 0-9"          115 White <NA>  <NA>
##  7 51646 2009_10  male       8 " 0-9"          101 White <NA>  <NA>
##  8 51647 2009_10  female    45 " 40-49"        541 White <NA>  College …
##  9 51647 2009_10  female    45 " 40-49"        541 White <NA>  College …
## 10 51647 2009_10  female    45 " 40-49"        541 White <NA>  College …
## # ... with 9,990 more rows, and 70 more variables: MaritalStatus <fct>,
## #   HHIncome <fct>, HHIncomeMid <int>, Poverty <dbl>, HomeRooms <int>,
## #   HomeOwn <fct>, Work <fct>, Weight <dbl>, Length <dbl>, HeadCirc <dbl>,
## #   Height <dbl>, BMI <dbl>, BMICatUnder20yrs <fct>, BMI_WHO <fct>,
## #   Pulse <int>, BPSysAve <int>, BPDiaAve <int>, BPSys1 <int>, BPDia1 <int>,
## #   BPSys2 <int>, BPDia2 <int>, BPSys3 <int>, BPDia3 <int>, Testosterone <dbl>,
## #   DirectChol <dbl>, TotChol <dbl>, UrineVol1 <int>, UrineFlow1 <dbl>,
## #   UrineVol2 <int>, UrineFlow2 <dbl>, Diabetes <fct>, DiabetesAge <int>,
## #   HealthGen <fct>, DaysPhysHlthBad <int>, DaysMentHlthBad <int>,
## #   LittleInterest <fct>, Depressed <fct>, nPregnancies <int>, nBabies <int>,
## #   Age1stBaby <int>, SleepHrsNight <int>, SleepTrouble <fct>,
## #   PhysActive <fct>, PhysActiveDays <int>, TVHrsDay <fct>, CompHrsDay <fct>,
## #   TVHrsDayChild <int>, CompHrsDayChild <int>, Alcohol12PlusYr <fct>,
## #   AlcoholDay <int>, AlcoholYear <int>, SmokeNow <fct>, Smoke100 <fct>,
## #   Smoke100n <fct>, SmokeAge <int>, Marijuana <fct>, AgeFirstMarij <int>,
## #   RegularMarij <fct>, AgeRegMarij <int>, HardDrugs <fct>, SexEver <fct>,
## #   SexAge <int>, SexNumPartnLife <int>, SexNumPartYear <int>, SameSex <fct>,
## #   SexOrientation <fct>, PregnantNow <fct>, BPsystodia <dbl>, height_in <dbl>,
## #   weight_lb <dbl>
```

c. Use the `select()` helper functions to select only variables dealing with alcohol or marijuana.

```
select(NHANES, contains("Marij"), contains("Alcohol"))
```

```
## # A tibble: 10,000 x 7
##    Marijuana AgeFirstMarij RegularMarij AgeRegMarij Alcohol12PlusYr AlcoholDay
##    <fct>             <int> <fct>              <int> <fct>                 <int>
##  1 Yes                  17 No                    NA Yes                      NA
##  2 Yes                  17 No                    NA Yes                      NA
##  3 Yes                  17 No                    NA Yes                      NA
##  4 <NA>                 NA <NA>                  NA <NA>                     NA
##  5 Yes                  18 No                    NA Yes                       2
##  6 <NA>                 NA <NA>                  NA <NA>                     NA
##  7 <NA>                 NA <NA>                  NA <NA>                     NA
##  8 Yes                  13 No                    NA Yes                       3
##  9 Yes                  13 No                    NA Yes                       3
## 10 Yes                  13 No                    NA Yes                       3
## # ... with 9,990 more rows, and 1 more variable: AlcoholYear <int>
```

d. Use the `select()` helper functions to select only variables containing `Sex` or `Gender` in the name.

```
select(NHANES, matches("Sex|Gender"))
```

```
## # A tibble: 10,000 x 7
##    Gender SexEver SexAge SexNumPartnLife SexNumPartYear SameSex SexOrientation
##    <fct>  <fct>    <int>           <int>          <int> <fct>   <fct>
##  1 male   Yes         16               8              1 No      Heterosexual
##  2 male   Yes         16               8              1 No      Heterosexual
##  3 male   Yes         16               8              1 No      Heterosexual
##  4 male   <NA>        NA              NA             NA <NA>    <NA>
##  5 female Yes         12              10              1 Yes     Heterosexual
##  6 male   <NA>        NA              NA             NA <NA>    <NA>
##  7 male   <NA>        NA              NA             NA <NA>    <NA>
##  8 female Yes         13              20              0 Yes     Bisexual
##  9 female Yes         13              20              0 Yes     Bisexual
## 10 female Yes         13              20              0 Yes     Bisexual
## # ... with 9,990 more rows
```

e. Include only the cases in the most recent survey year and sort them in descending order by height in inches. How tall were the tallest three individuals?

```
arrange(NHANES, desc(Height)) %>%
  select(ID, SurveyYr, Height) %>%
  top_n(3)
```

```
## Selecting by Height
```

```
## # A tibble: 3 x 3
##      ID SurveyYr Height
##   <int> <fct>     <dbl>
## 1 71315 2011_12    200.
## 2 71315 2011_12    200.
## 3 64151 2011_12    200.
```