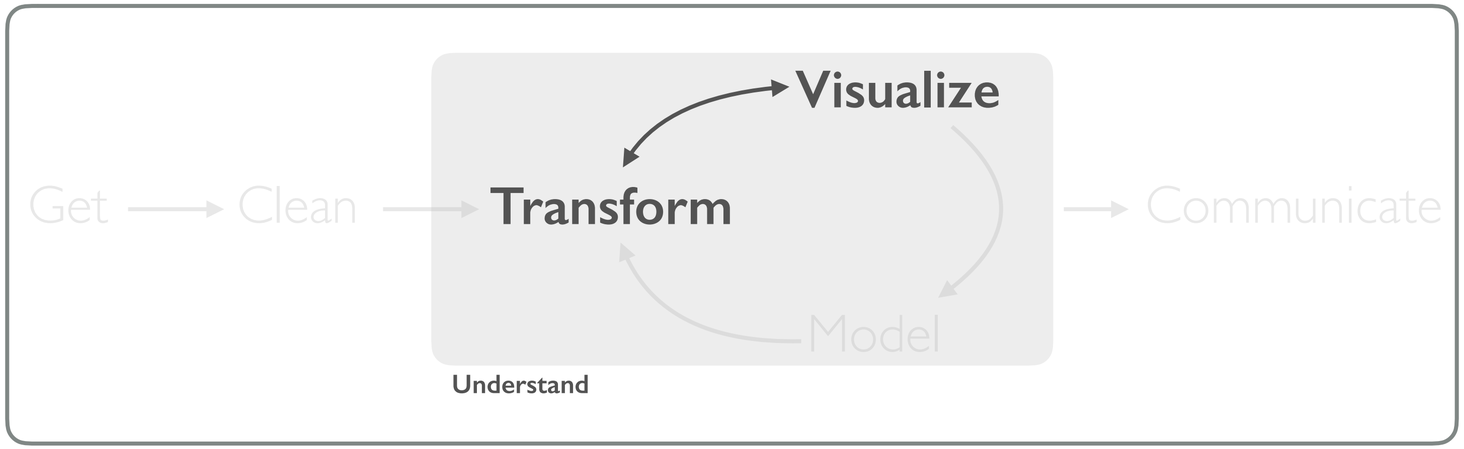
EXPLORATORY DATA ANALYSIS



†A modified version of Hadley Wickham's analytic process

“Exploratory data analysis is detective work – numerical detective work – or counting detective work – or graphical detective work”

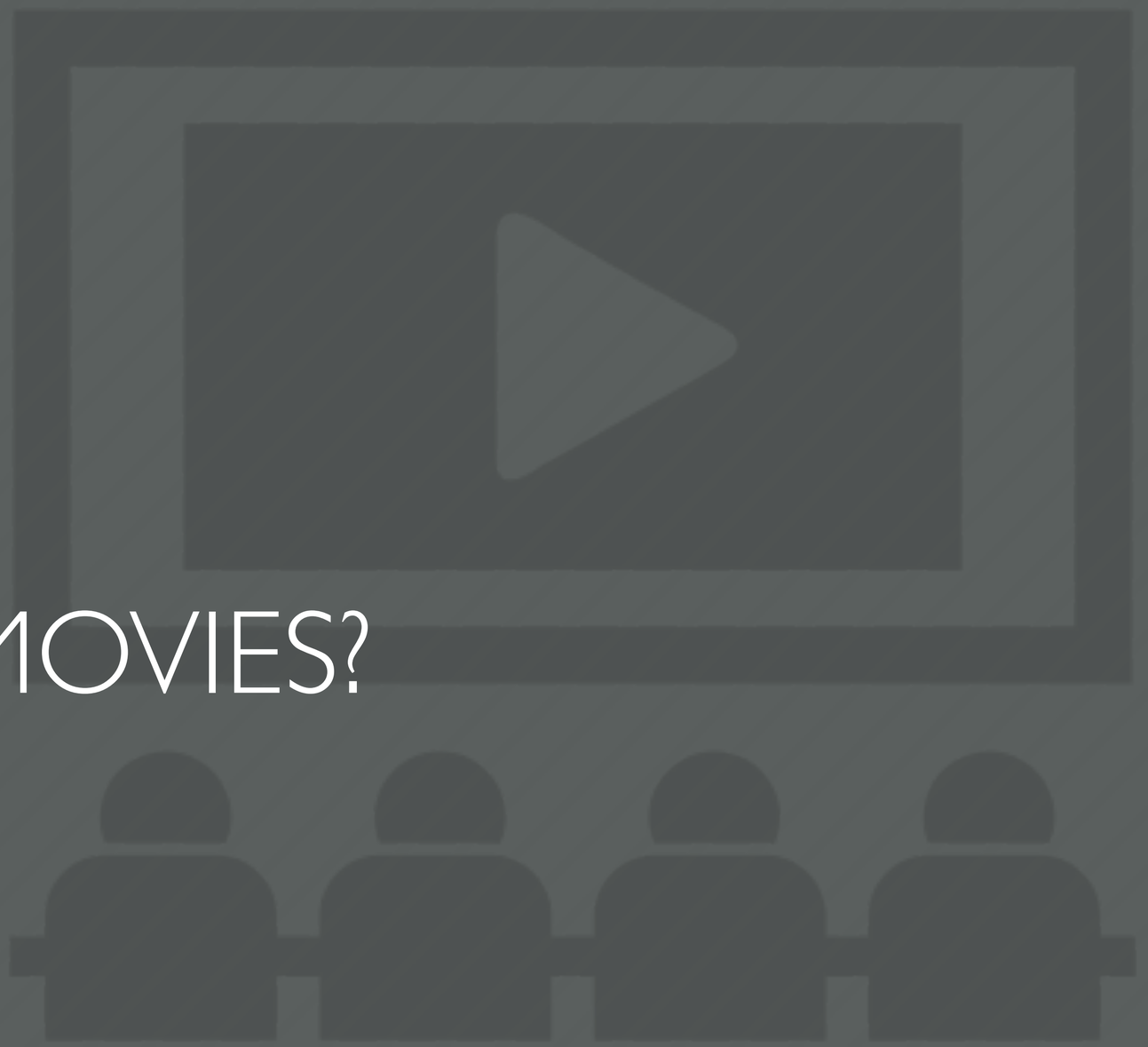
- John Tukey

“Exploratory data analysis can never be the whole story, but nothing else can serve as the foundation stone.”

EDA

- Now that you have the basics of data transformation & visualization down, lets use this knowledge to systematically explore data.
- This section contains
  - Lots of data
  - Lots of questions

HOW LONG ARE MOVIES?



PREREQUISITE

```
library(ggplot2movies)
library(tidyverse)

movies
# A tibble: 58,788 × 24
                      title  year length budget rating votes    r1    r2    r3    r4    r5    r6    r7    r8
                      <chr> <int>  <int>  <int>  <dbl> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1                         $  1971    121     NA    6.4   348   4.5   4.5   4.5   4.5  14.5  24.5  24.5  14.5
2           $1000 a Touchdown  1939     71     NA    6.0    20   0.0  14.5   4.5  24.5  14.5  14.5  14.5   4.5
3      $21 a Day Once a Month  1941      7     NA    8.2     5   0.0   0.0   0.0   0.0   0.0  24.5   0.0  44.5
4                    $40,000  1996     70     NA    8.2     6  14.5   0.0   0.0   0.0   0.0   0.0   0.0   0.0
5  $50,000 Climax Show, The  1975     71     NA    3.4    17  24.5   4.5   0.0  14.5  14.5   4.5   0.0   0.0
6                      $pent  2000     91     NA    4.3    45   4.5   4.5   4.5  14.5  14.5  14.5   4.5   4.5
7                    $windle  2002     93     NA    5.3   200   4.5   0.0   4.5   4.5  24.5  24.5  14.5   4.5
8                       '15'  2002     25     NA    6.7    24   4.5   4.5   4.5   4.5   4.5  14.5  14.5  14.5
9                       '38  1987     97     NA    6.6    18   4.5   4.5   4.5   0.0   0.0   0.0  34.5  14.5
```

LONG MOVIES

1. *Assess the distribution of movie lengths*
2. *How would you define "long"?*
3. *How many long movies are there?*
4. *What are the top 5 longest movies?*
5. *Create a new variable that signals these as "long" movies*

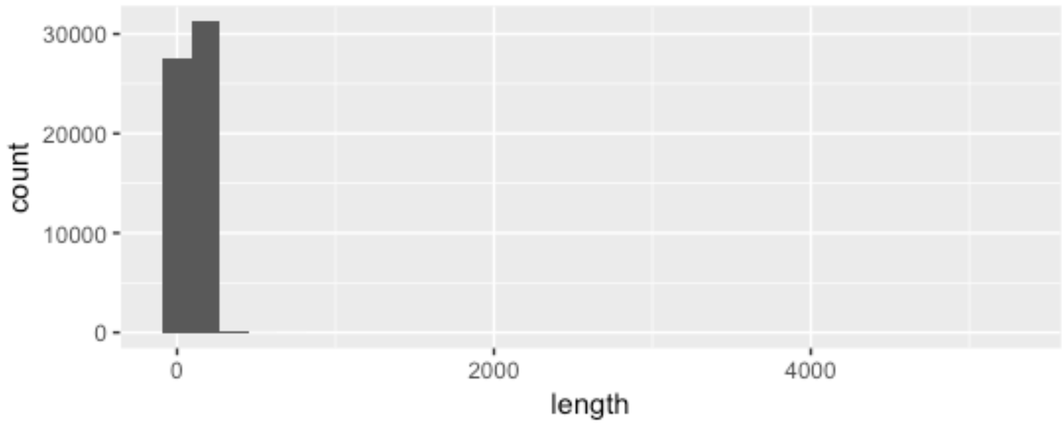
LONG MOVIES

1. *Assess the distribution of movie lengths*
2. *How would you define "long"?*
3. *How many long movies are there?*
4. *What are the top 5 longest movies?*
5. *Create a new variable that signals these as "long" movies*

ASSESSING THE DISTRIBUTION

```
ggplot(movies, aes(length)) +
  geom_histogram()
```

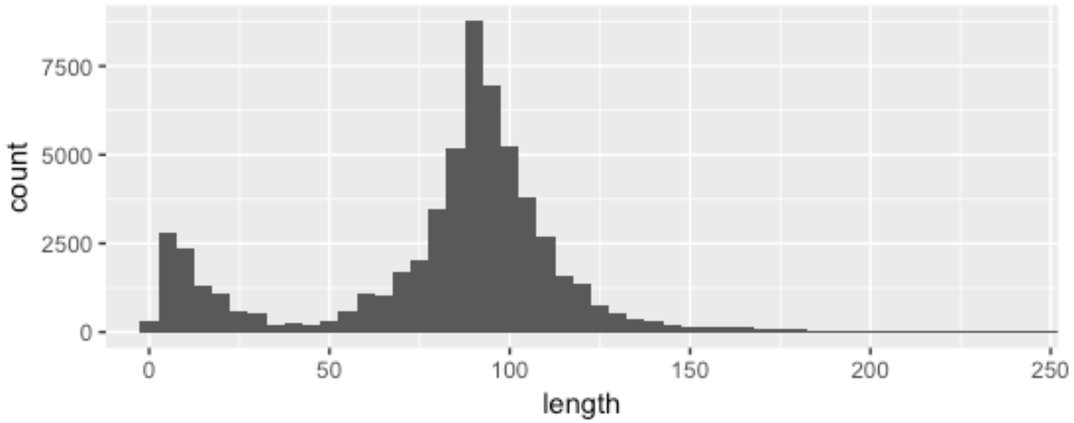
- Our basic distribution is not very informative
- However it does signal that some unusually long movies exist, we just can't tell where



ASSESSING THE DISTRIBUTION

```
ggplot(movies, aes(length)) +
  geom_histogram(binwidth = 5) +
  coord_cartesian(xlim = c(0, 60*4))
```

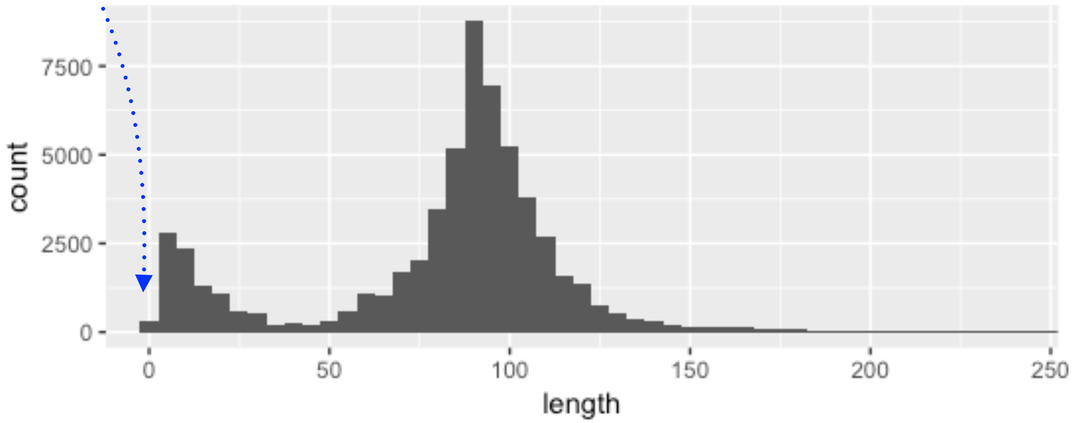
- Lets set our x-axis to a max of 4 hours and adjust our binwidth to get more details of the distribution
- Looks like "normal" length movies don't get much longer than 150 mins



ASSESSING THE DISTRIBUTION

```
movies %>%
  count(cut_width(length, 5))
# A tibble: 85 × 2
   `cut_width(length, 5)`     n
                   <fctr> <int>
1                [-2.5,2.5]   285
2                 (2.5,7.5]  2812
3                (7.5,12.5]  2366
4               (12.5,17.5]  1300
5               (17.5,22.5]  1092
6               (22.5,27.5]   595
7               (27.5,32.5]   532
8               (32.5,37.5]   194
9               (37.5,42.5]   236
10              (42.5,47.5]   182
# ... with 75 more rows
```

- We can use cut_width to see the actual counts within our histogram bins



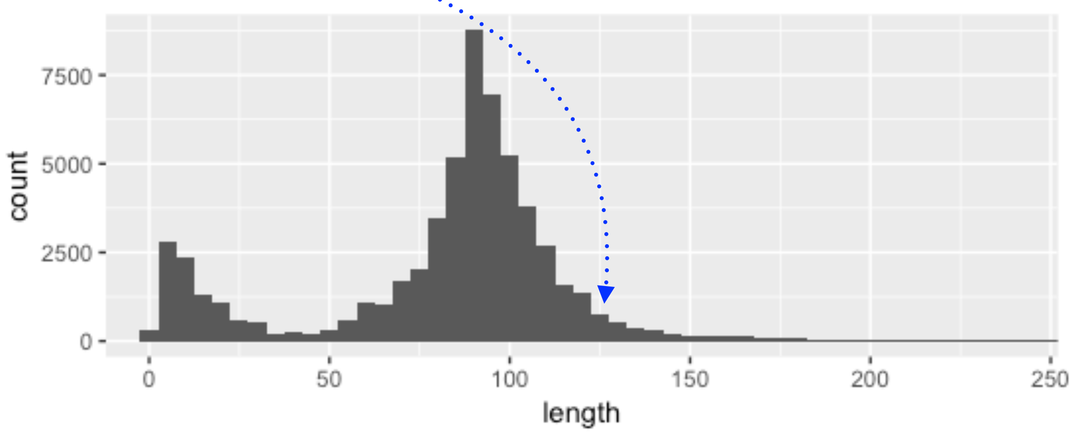
LONG MOVIES

1. *Assess the distribution of movie lengths*
2. *How would you define "long"?*
3. *How many long movies are there?*
4. *What are the top 5 longest movies?*
5. *Create a new variable that signals these as "long" movies*

HOW I DEFINE LONG

```
movies %>%
  count(cut_width(length, 5)) %>%
  mutate(cum_pct = cumsum(n)/sum(n)) %>%
  filter(cum_pct > .95)
# A tibble: 60 × 3
   `cut_width(length, 5)`     n   cum_pct
                   <fctr> <int>     <dbl>
1            (122.5,127.5]   766 0.9549398
2            (127.5,132.5]   514 0.9636831
3            (132.5,137.5]   378 0.9701129
4            (137.5,142.5]   322 0.9755903
5            (142.5,147.5]   216 0.9792645
6            (147.5,152.5]   165 0.9820712
7            (152.5,157.5]   139 0.9844356
8            (157.5,162.5]   135 0.9867320
```

- We can use cut_width to see the actual counts within our histogram bins
- We can use this to identify where the 95th percentile for length is



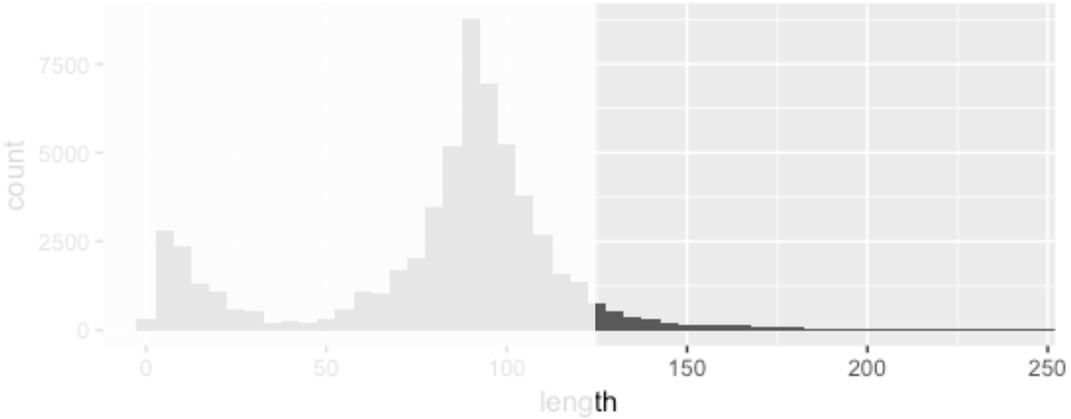
LONG MOVIES

1. *Assess the distribution of movie lengths*
2. *How would you define "long"?*
3. *How many long movies are there?*
4. *What are the top 5 longest movies?*
5. *Create a new variable that signals these as "long" movies*

HOW MANY LONG MOVIES ARE THERE?

```
movies %>%
  count(cut_width(length, 5)) %>%
  mutate(cum_pct = cumsum(n)/sum(n)) %>%
  filter(cum_pct > .95) %>%
  summarise(sum(n))
# A tibble: 1 × 1
  `sum(n)`
     <int>
1     3415
```

- We can use cut_width to see the actual counts within our histogram bins
- We can use this to identify where the 95th percentile for length is
- We can easily add a **`filter`** and **`summarize`** to identify how many "long" movies there are



LONG MOVIES

1. *Assess the distribution of movie lengths*
2. *How would you define "long"?*
3. *How many long movies are there?*
4. *What are the top 5 longest movies?*
5. *Create a new variable that signals these as "long" movies*

TOP 5 LONGEST MOVIES

```
movies %>%
  arrange(desc(length)) %>%
  top_n(5, wt = length)
# A tibble: 5 × 24
                                                title  year length budget rating votes
                                                <chr> <int>  <int>  <int>  <dbl> <int>
1                                Cure for Insomnia, The  1987   5220     NA    3.8    59
2 Longest Most Meaningless Movie in the World, The  1970   2880     NA    6.4    15
3                                        Four Stars  1967   1100     NA    3.0    12
4                                             Resan  1987    873     NA    5.5    12
5                                             Out 1  1971    773     NA    6.7    20
# ... with 18 more variables: r1 <dbl>, r2 <dbl>, r3 <dbl>, r4 <dbl>, r5 <dbl>,
#   r6 <dbl>, r7 <dbl>, r8 <dbl>, r9 <dbl>, r10 <dbl>, mpaa <chr>, Action <int>,
#   Animation <int>, Comedy <int>, Drama <int>, Documentary <int>, Romance <int>,
#   Short <int>
```

LONG MOVIES

1. *Assess the distribution of movie lengths*
2. *How would you define "long"?*
3. *How many long movies are there?*
4. *What are the top 5 longest movies?*
5. *Create a new variable that signals these as "long" movies*

CREATE NEW VARIABLE

```
movies %>%
  select(1:3) %>%
  mutate(Long = length >= 122.5)
# A tibble: 58,788 × 4
                      title  year length  Long
                      <chr> <int>  <int> <lgl>
1                         $  1971    121 FALSE
2           $1000 a Touchdown  1939     71 FALSE
3      $21 a Day Once a Month  1941      7 FALSE
4                    $40,000  1996     70 FALSE
5   $50,000 Climax Show, The  1975     71 FALSE
6                      $pent  2000     91 FALSE
7                    $windle  2002     93 FALSE
8                       '15'  2002     25 FALSE
9                        '38  1987     97 FALSE
```

- Use `logical comparison` to identify the movies that have lengths equal to or greater than the 95th percentile

We'll do more with this later

SHORT FILMS

1. *How did you determine where short films start and stop?*
2. *How many short films are there?*
3. *What is the average length of short films?*
4. *Create a new variable the signals these as "short" movies*

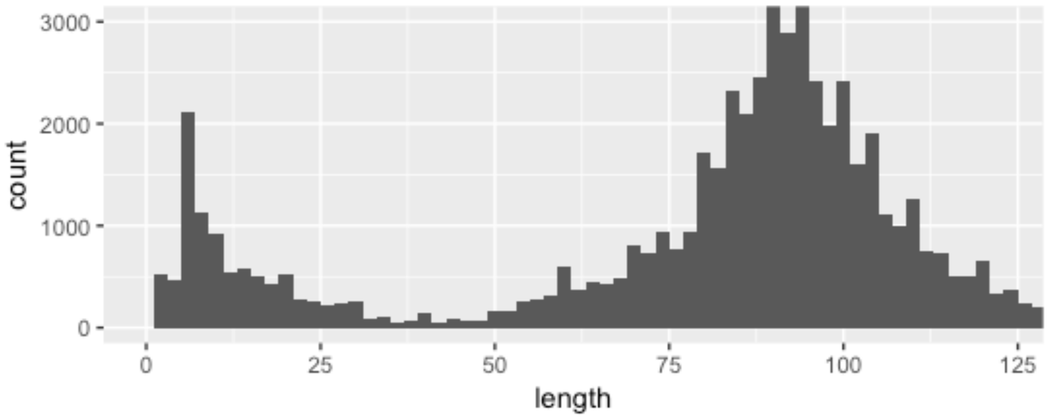
SHORT FILMS

1. *How did you determine where short films start and stop?*
2. *How many short films are there?*
3. *What is the average length of short films?*
4. *Create a new variable the signals these as "short" movies*

DEFINING SHORT FILMS

```
ggplot(movies, aes(length)) +
  geom_histogram(binwidth = 2) +
  coord_cartesian(
    xlim = c(0, 122.5),
    ylim = c(0, 3000)
    )
```

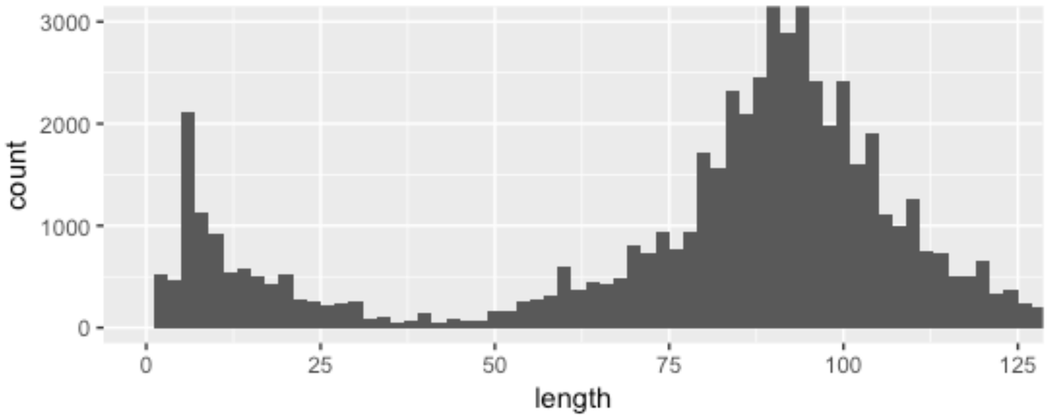
- If we zoom into our distribution again, we see that a group of short films exist



DEFINING SHORT FILMS

```
ggplot(movies, aes(length)) +
  geom_histogram(binwidth = 2) +
  coord_cartesian(
    xlim = c(0, 122.5),
    ylim = c(0, 3000)
    )
```

- If we zoom into our distribution again, we see that a group of short films exist
- Defining short films from the data is not as easy
- Luckily wikipedia defines it as 40 mins or less



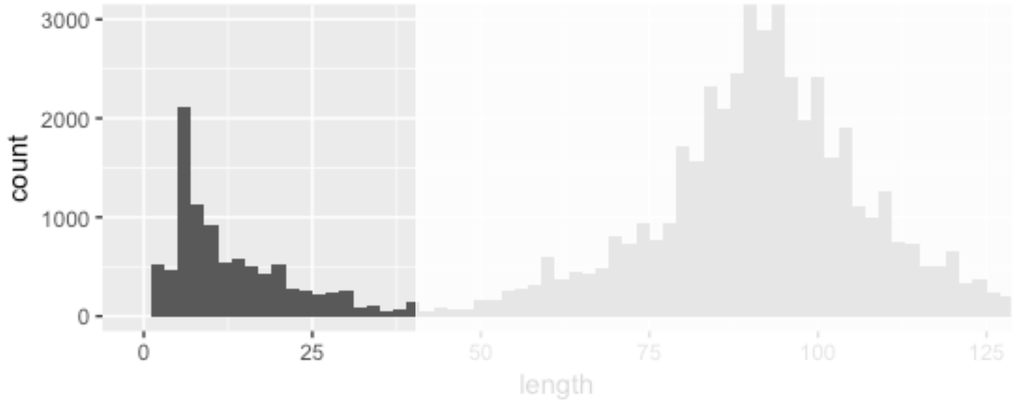
SHORT FILMS

1. *How did you determine where short films start and stop?*
2. *How many short films are there?*
3. *What is the average length of short films?*
4. *Create a new variable the signals these as "short" movies*

HOW MANY SHORT FILMS ARE THERE?

```
movies %>%
  filter(length <= 40) %>%
  summarise(n())
# A tibble: 1 × 1
  `n()`
  <int>
1  9353
```

- We can easily add a `filter` and `summarize` to identify how many "short" movies there are



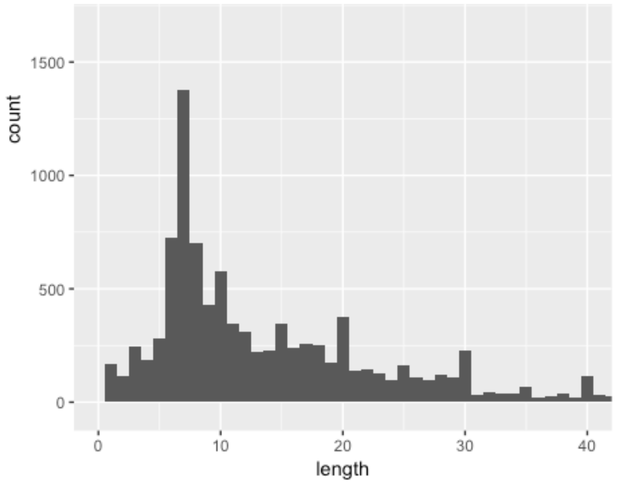
SHORT FILMS

1. *How did you determine where short films start and stop?*
2. *How many short films are there?*
3. *What is the average length of short films?*
4. *Create a new variable the signals these as "short" movies*

AVERAGE LENGTH OF A SHORT FILM

```
movies %>%
  count(cut_width(length, 1))
# A tibble: 305 × 2
   `cut_width(length, 1)`     n
                   <fctr> <int>
1                [0.5,1.5]   169
2                (1.5,2.5]   116
3                (2.5,3.5]   243
4                (3.5,4.5]   185
5                (4.5,5.5]   279
6                (5.5,6.5]   726
7                (6.5,7.5]  1379
8                (7.5,8.5]   700
9                (8.5,9.5]   432
10              (9.5,10.5]   576
```

- By tweaking our `cut_width` parameter we see that the most common length of short films are 7 minutes




```
movies %>%
  count(cut_width(length, .05))
# A tibble: 305 × 2
   `cut_width(length, 0.05)`     n
                       <fctr> <int>
1                [0.975,1.025]   169
2                (1.975,2.025]   116
3                (2.975,3.025]   243
4                (3.975,4.025]   185
5                (4.975,5.025]   279
6                (5.975,6.025]   726
7                (6.975,7.025]  1379
8                (7.975,8.025]   700
9                (8.975,9.025]   432
10              (9.975,10.025]   576
```

AVERAGE LENGTH OF A SHORT FILM

```
movies %>%
  filter(length <= 40) %>%
  summarize(mean = mean(length, na.rm = T),
            median = median(length, na.rm = T))
# A tibble: 1 × 2
     mean median
    <dbl>  <int>
1 13.4235     10
```

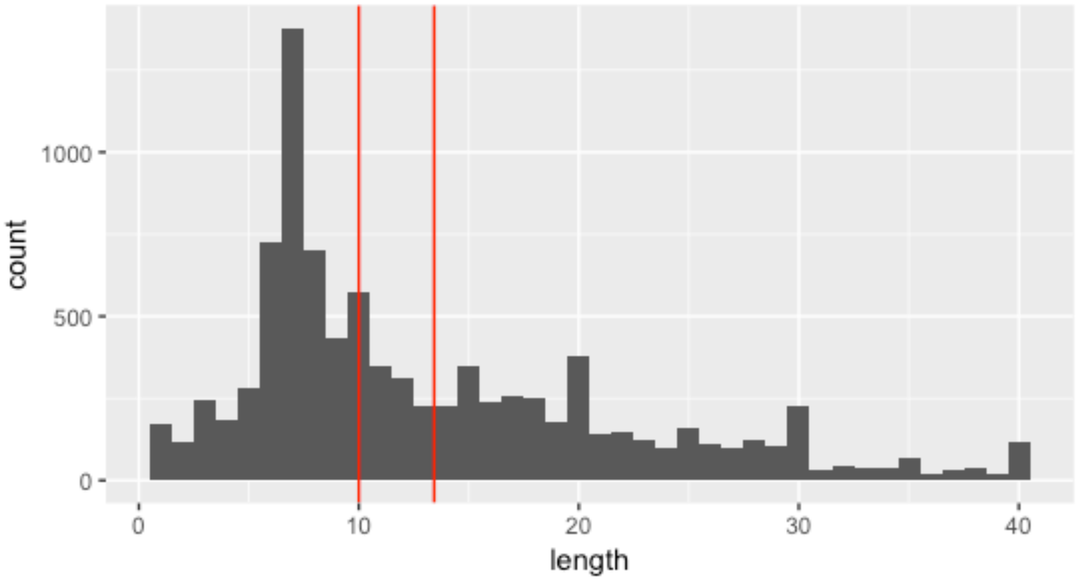
- We can also get our central measures numerically

AVERAGE LENGTH OF A SHORT FILM

```
avg <- movies %>%
  filter(length <= 40) %>%
  summarize(mean = mean(length, na.rm = T),
            median = median(length, na.rm = T))

movies %>%
  filter(length <= 40) %>%
  ggplot(aes(length)) +
  geom_histogram(binwidth = 1) +
  geom_vline(xintercept = c(avg$mean, avg$median),
             color = "red")
```

- We can also get our central measures numerically
- We can also do this graphically



SHORT FILMS

1. *How did you determine where short films start and stop?*
2. *How many short films are there?*
3. *What is the average length of short films?*
4. *Create a new variable that signals these as "short" movies*

CREATE NEW VARIABLE

```
movies %>%
  select(1:3) %>%
  mutate(Description = ifelse(length >= 122.5, "Long",
                       ifelse(length <= 40, "Short",
                              "Regular")))
# A tibble: 58,788 × 4
                      title  year length Description
                      <chr> <int>  <int>       <chr>
1                         $  1971    121     Regular
2           $1000 a Touchdown  1939     71     Regular
3      $21 a Day Once a Month  1941      7       Short
4                    $40,000  1996     70     Regular
5  $50,000 Climax Show, The   1975     71     Regular
6                      $pent  2000     91     Regular
7                    $windle  2002     93     Regular
```

- Let's use `ifelse` statements to create a variable identifying movies that are:
  - "Short"
  - "Regular"
  - "Long"

We'll do more with this later

REGULAR FILMS

1. *What is the average length of "regular" films?*
2. *Are there certain length cut-offs that are favored over others?*
3. *How do ratings differ between short, regular, and long length films?*

REGULAR FILMS

1. *What is the average length of "regular" films?*
2. *Are there certain length cut-offs that are favored over others?*
3. *How do ratings differ between short, regular, and long length films?*

AVG LENGTH OF "REGULAR" FILMS

```
movies %>%
  count(cut_width(length, 2), sort = TRUE)
# A tibble: 174 × 2
   `cut_width(length, 2)`     n
                   <fctr> <int>
1                  (89,91]  4810
2                  (93,95]  3171
3                  (91,93]  2897
4                  (87,89]  2455
5                  (95,97]  2411
6                 (99,101]  2411
7                  (83,85]  2323
8                    (5,7]  2105
9                  (85,87]  2093
10                 (97,99]  1991
# ... with 164 more rows
```

- Most common length is about 90 minutes and…

AVG LENGTH OF "REGULAR" FILMS

```
movies %>%
  filter(length > 40 & length < 122.5) %>%
  summarize(mean = mean(length, na.rm = T),
            median = median(length, na.rm = T))
# A tibble: 1 × 2
     mean median
    <dbl>  <dbl>
1 91.0598     92
```

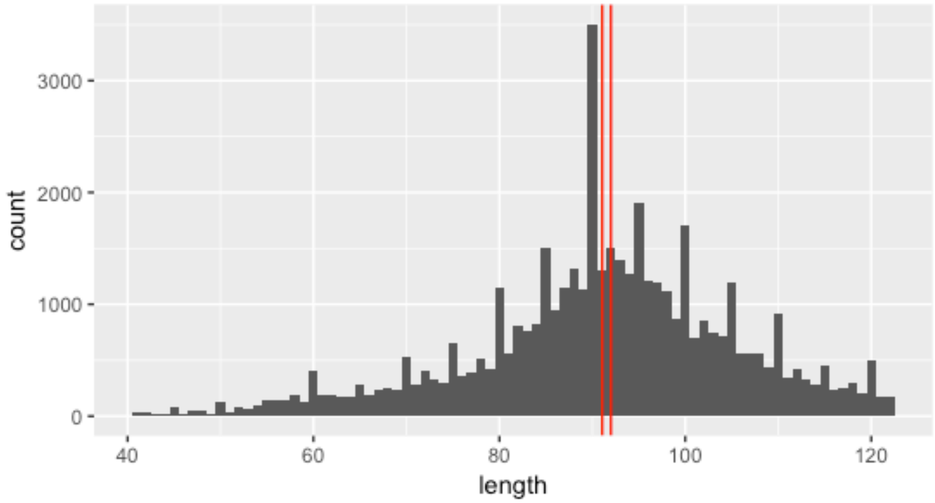
- Most common length is about 90 minutes and…
- Mean and median are slightly higher

AVG LENGTH OF "REGULAR" FILMS

```
avg <- movies %>%
  filter(length > 40 & length < 122.5) %>%
  summarize(mean = mean(length, na.rm = T),
            median = median(length, na.rm = T))

movies %>%
  filter(length > 40 & length < 122.5) %>%
  ggplot(aes(length)) +
  geom_histogram(binwidth = 1) +
  geom_vline(xintercept = c(avg$mean, avg$median),
             color = "red")
```

- Most common length is about 90 minutes and…
- Mean and median are slightly higher
- Visualizing this, we can see our distribution of "regular" length films appears normally distributed



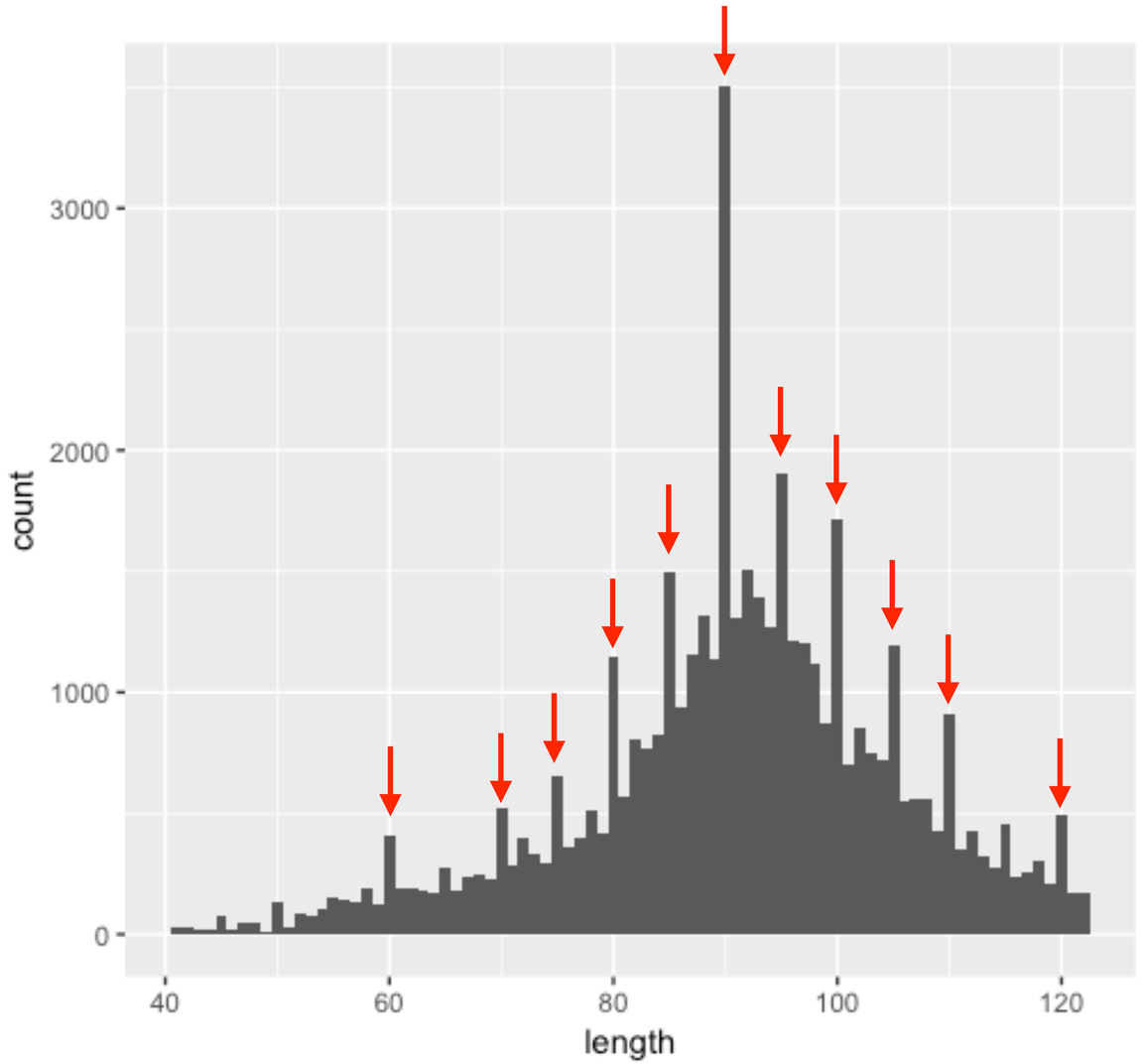
REGULAR FILMS

1. *What is the average length of "regular" films?*

2. *Are there certain length cut-offs that are favored over others?*

3. *How do ratings differ between short, regular, and long length films?*

ARE CERTAIN CUT-OFFS FAVORED?



- It appears that certain cut-off values are preferred?

ARE CERTAIN CUT-OFFS FAVORED?

```
movies %>%
  filter(length > 40 & length < 122.5) %>%
  count(cut_width(length, 1 )) %>%
  mutate(change = (n - lag(n)) / lag(n))
# A tibble: 82 × 3
   `cut_width(length, 1)`     n      change
                   <fctr> <int>       <dbl>
1             [40.5,41.5]    30          NA
2             (41.5,42.5]    29 -0.03333333
3             (42.5,43.5]    22 -0.24137931
4             (43.5,44.5]    22  0.00000000
5             (44.5,45.5]    74  2.36363636
6             (45.5,46.5]    21 -0.71621622
7             (46.5,47.5]    43  1.04761905
8             (47.5,48.5]    50  0.16279070
```

- It appears that certain cut-off values are preferred?
- We can find out where this is happening by computing the change in value from one bin to the next.

ARE CERTAIN CUT-OFFS FAVORED?

```
movies %>%
  filter(length > 40 & length < 122.5) %>%
  count(cut_width(length, 1 )) %>%
  mutate(change = (n - lag(n)) / lag(n)) %>%
  arrange(desc(change))
# A tibble: 82 × 3
   `cut_width(length, 1)`     n   change
                   <fctr> <int>    <dbl>
1             (49.5,50.5]   129 9.750000
2             (44.5,45.5]    74 2.363636
3             (59.5,60.5]   409 2.325203
4             (51.5,52.5]    88 2.259259
5             (89.5,90.5]  3506 2.091711
6             (79.5,80.5]  1149 1.729216
7           (119.5,120.5]   496 1.350711
8             (69.5,70.5]   533 1.254310
```

- It appears that certain cut-off values are preferred?
- We can find out where this is happening by computing the change in value from one bin to the next.
- And then looking at the bins that experience the largest change

ARE CERTAIN CUT-OFFS FAVORED?

```
movies %>%
  filter(length > 40 & length < 122.5) %>%
  count(cut_width(length, 1 )) %>%
  mutate(change = (n - lag(n)) / lag(n)) %>%
  arrange(desc(change))
# A tibble: 82 × 3
   `cut_width(length, 1)`     n   change
                   <fctr> <int>    <dbl>
1             (49.5,50.5]   129 9.750000
2             (44.5,45.5]    74 2.363636
3             (59.5,60.5]   409 2.325203
4             (51.5,52.5]    88 2.259259
5             (89.5,90.5]  3506 2.091711
6             (79.5,80.5]  1149 1.729216
7           (119.5,120.5]   496 1.350711
```

- It appears that certain cut-off values are preferred?
- We can find out where this is happening by computing the change in value from one bin to the next.
- And then looking at the bins that experience the largest change
- We find that the largest changes occur at the 5 and 10 min marks

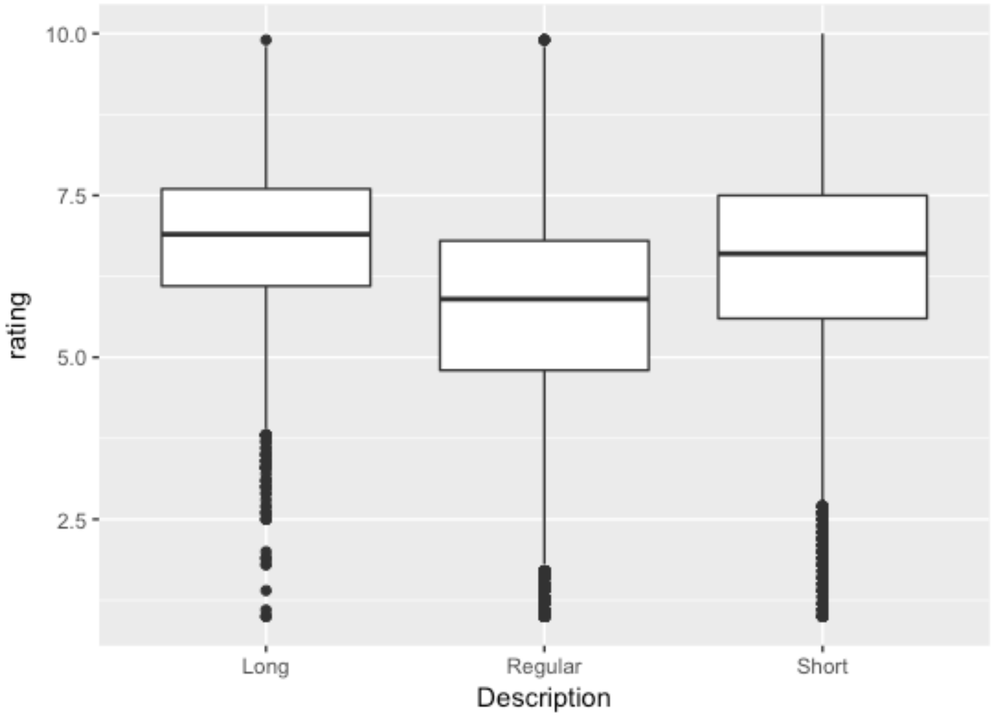
REGULAR FILMS

1. *What is the average length of "regular" films?*
2. *Are there certain length cut-offs that are favored over others?*
3. *How do ratings differ between short, regular, and long length films?*

HOW DO RATINGS DIFFER?

```
movies %>%
  mutate(
    Description = ifelse(length >= 122.5, "Long",
                         ifelse(length <= 40, "Short",
                                "Regular"))
    ) %>%
  ggplot(aes(Description, rating)) +
  geom_boxplot()
```

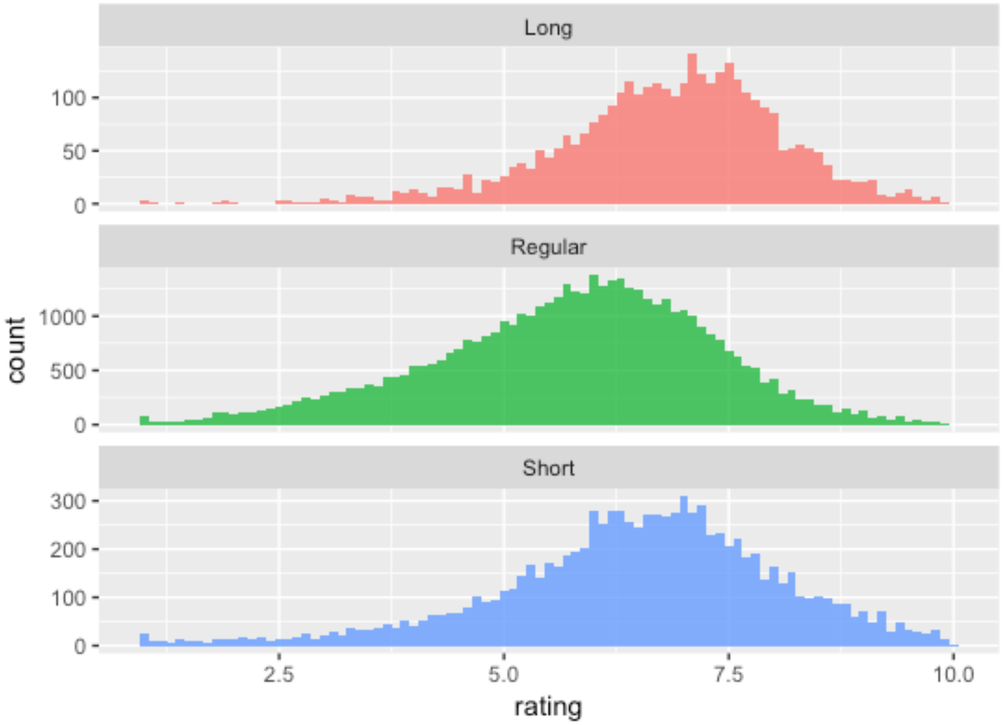
- Boxplot



HOW DO RATINGS DIFFER?

```
movies %>%
  mutate(
    Description = ifelse(length >= 122.5, "Long",
                         ifelse(length <= 40, "Short",
                                "Regular"))
    ) %>%
  ggplot(aes(rating, fill = Description)) +
  geom_histogram(
    binwidth = .1,
    alpha = .8,
    show.legend = FALSE
    ) +
  facet_wrap(
    ~Description,
    ncol = 1,
    scales = "free_y"
    )
```

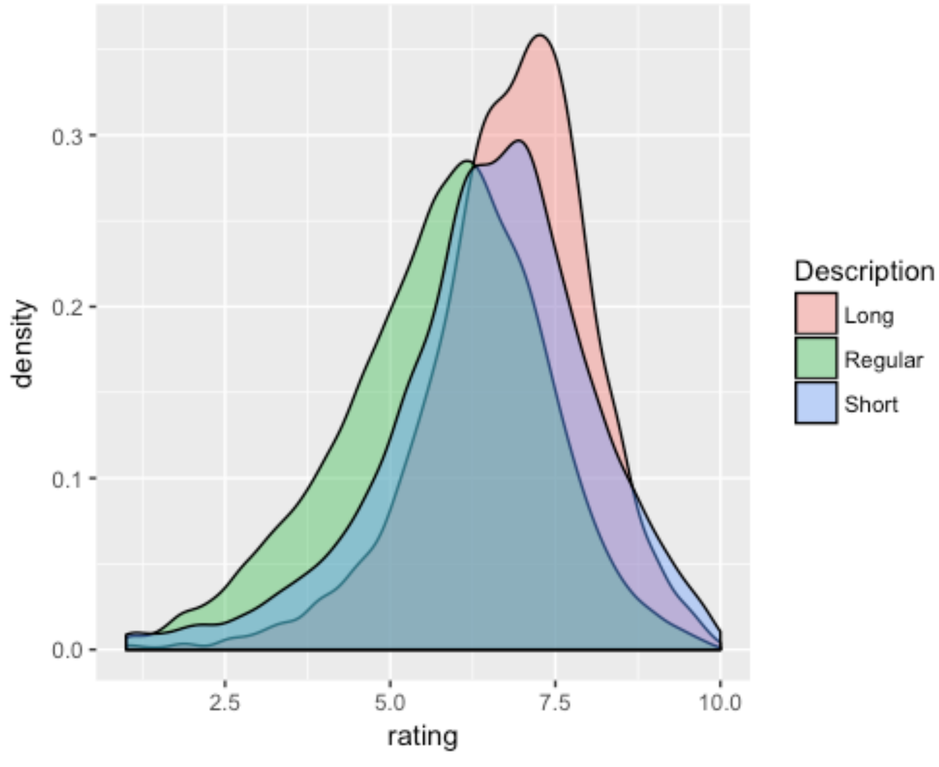
- Boxplot
- Facetted histogram



HOW DO RATINGS DIFFER?

```
movies %>%
  mutate(
    Description = ifelse(length >= 122.5, "Long",
                         ifelse(length <= 40, "Short",
                                "Regular"))
    ) %>%
  ggplot(aes(rating, fill = Description)) +
  geom_density(alpha = .4)
```

- Boxplot
- Facetted histogram
- Density plot



HOW FAST ARE DOWNHILL SKIERS?

PREREQUISITE

```
library(GDAdata)
library(tidyverse)

(SpeedSki <- as_tibble(SpeedSki))
# A tibble: 91 × 10
    Rank   Bib FIS.Code                 Name  Year Nation  Speed    Sex     Event
   <int> <int>    <int>               <fctr> <int> <fctr>  <dbl> <fctr>    <fctr>
1      1    61     7039       ORIGONE Simone  1979    ITA 211.67   Male Speed One
2      2    59     7078         ORIGONE Ivan  1987    ITA 209.70   Male Speed One
3      3    66   190130       MONTES Bastien  1985    FRA 209.69   Male Speed One
4      4    57     7178 SCHROTTSHAMMER Klaus  1979    AUT 209.67   Male Speed One
5      5    69   510089         MAY Philippe  1970    SUI 209.19   Male Speed One
6      6    75     7204          BILLY Louis  1993    FRA 208.33   Male Speed One
7      7    67     7053       PERSSON Daniel  1975    SWE 208.03   Male Speed One
8      8    58     7170          BILLY Simon  1991    FRA 207.59   Male Speed One
```

SPEED SKIERS

1. *How fast are downhill skiers?*
2. *Has the average speed changed over time?*
3. *Is there a difference between events?*
4. *Is there a difference between genders?*
5. *What is driving the difference between genders?*

SPEED SKIERS

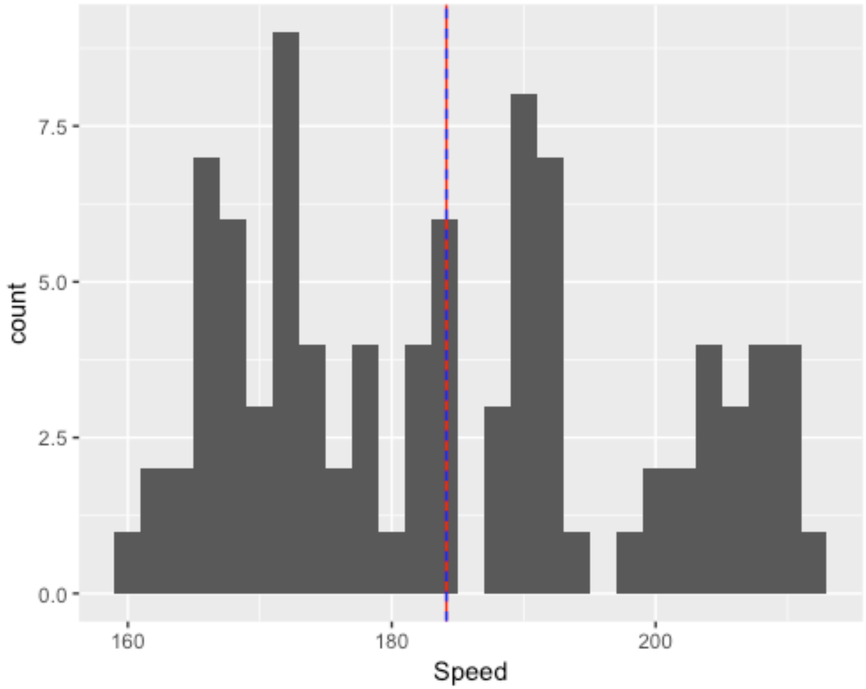
1. *How fast are downhill skiers?*
2. *Has the average speed changed over time?*
3. *Is there a difference between events?*
4. *Is there a difference between genders?*
5. *What is driving the difference between genders?*

HOW FAST ARE DOWNHILL SKIERS?

```
(avg <- SpeedSki %>%
  summarise(Mean = mean(Speed),
            Median = median(Speed)))
# A tibble: 1 × 2
     Mean Median
    <dbl>  <dbl>
1 184.1442 183.13

SpeedSki %>%
  ggplot(aes(Speed)) +
  geom_histogram(binwidth = 2) +
  geom_vline(xintercept = avg$Mean, color = "red") +
  geom_vline(xintercept = avg$Mean, color = "blue",
             linetype = "dashed")
```

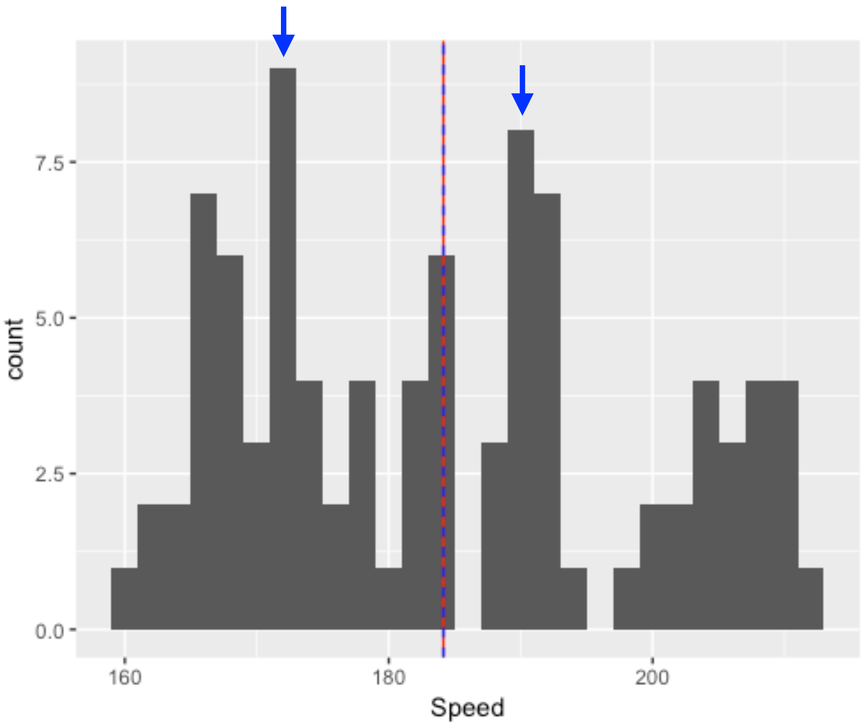
- Not a simple answer
  - Mean & Median ~184



HOW FAST ARE DOWNHILL SKIERS?

```
SpeedSki %>%
  count(cut_width(Speed, 2)) %>%
  arrange(desc(n))
# A tibble: 25 × 2
   `cut_width(Speed, 2)`     n
                  <fctr> <int>
1               (171,173]     9
2               (189,191]     8
3               (165,167]     7
4               (191,193]     7
5               (167,169]     6
6               (183,185]     6
7               (173,175]     4
8               (177,179]     4
9               (181,183]     4
10              (203,205]     4
# ... with 15 more rows
```

- Not a simple answer
  - Mean & Median ~184
  - Common bins ~ 172 &190



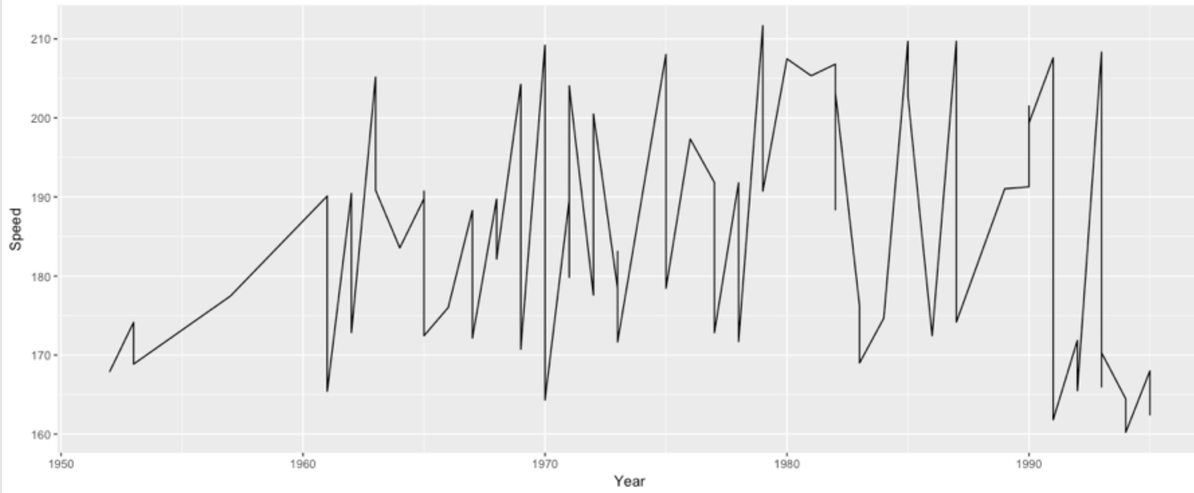
SPEED SKIERS

1. *How fast are downhill skiers?*
2. *Has the average speed changed over time?*
3. *Is there a difference between events?*
4. *Is there a difference between genders?*
5. *What is driving the difference between genders?*

HAS SPEED CHANGED OVER TIME?

```
SpeedSki %>%
  ggplot(aes(Year, Speed)) +
  geom_line()
```

- Year-to-year line chart is a bit ugly

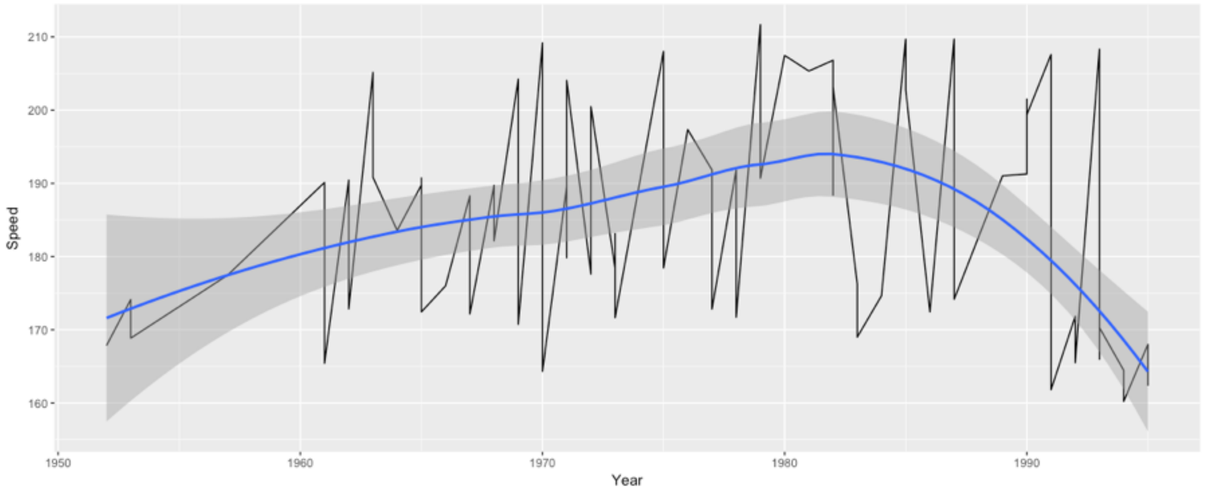


HAS SPEED CHANGED OVER TIME?

```
SpeedSki %>%
  ggplot(aes(Year, Speed)) +
  geom_line() +
  geom_smooth()
```

Try adding `span = .5` and `span = 1`
What does this do?

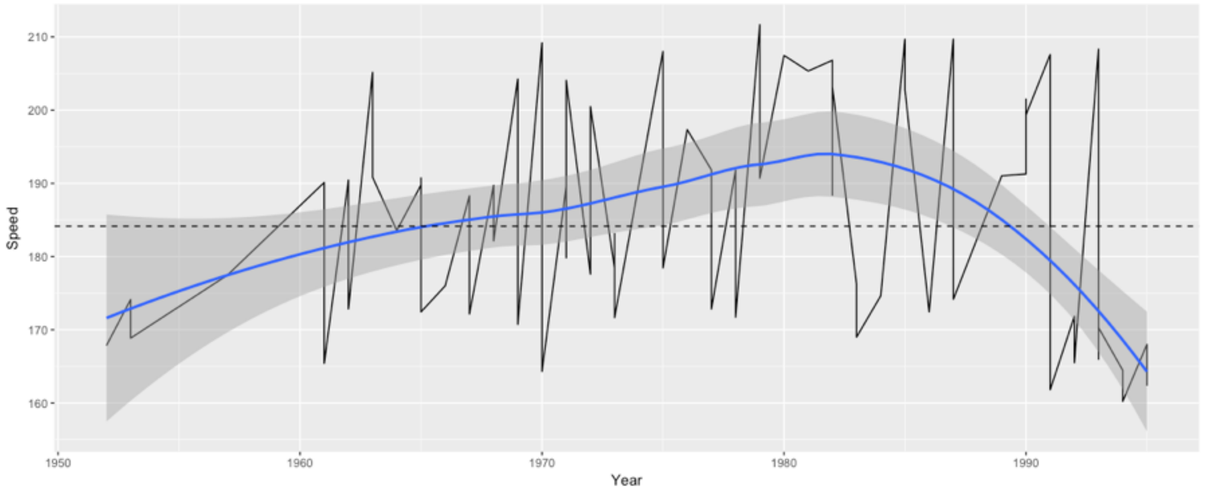
- Year-to-year is a bit ugly
- Smoothing tells us that the average speed peaked in in the early 80's



HAS SPEED CHANGED OVER TIME?

```
SpeedSki %>%
  ggplot(aes(Year, Speed)) +
  geom_line() +
  geom_smooth() +
  geom_hline(yintercept = mean(SpeedSki$Speed),
             linetype = "dashed")
```

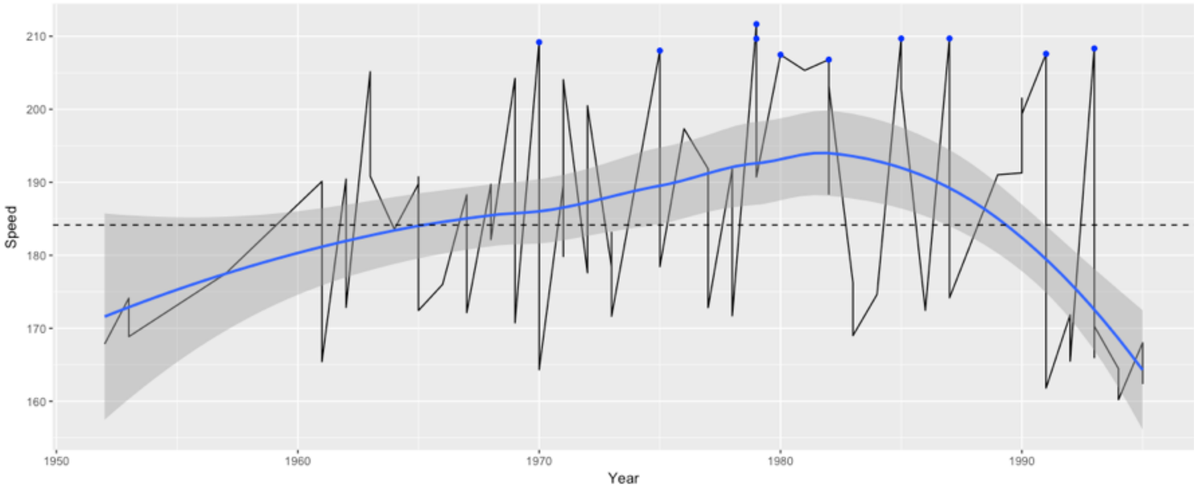
- Year-to-year is a bit ugly
- Smoothing tells us that the average speed peaked in in the early 80's
- Adding our historical avg shows how speed has deviated from it



HAS SPEED CHANGED OVER TIME?

```
SpeedSki %>%
  ggplot(aes(Year, Speed)) +
  geom_line() +
  geom_smooth() +
  geom_hline(yintercept = mean(SpeedSki$Speed),
             linetype = "dashed") +
  geom_point(
    data = top_n(SpeedSki, 10, wt = Speed),
    aes(Year, Speed),
    color = "blue"
    )
```

- Year-to-year is a bit ugly
- Smoothing tells us that the average speed peaked in in the early 80's
- Adding our historical avg shows how speed has deviated from it
- We can even add points to identify the top 10 fastest years



SPEED SKIERS

1. *How fast are downhill skiers?*
2. *Has the average speed changed over time?*
3. *Is there a difference between events?*
4. *Is there a difference between genders?*
5. *What is driving the difference between genders?*

IS THERE A DIFFERENCE BETWEEN EVENTS?

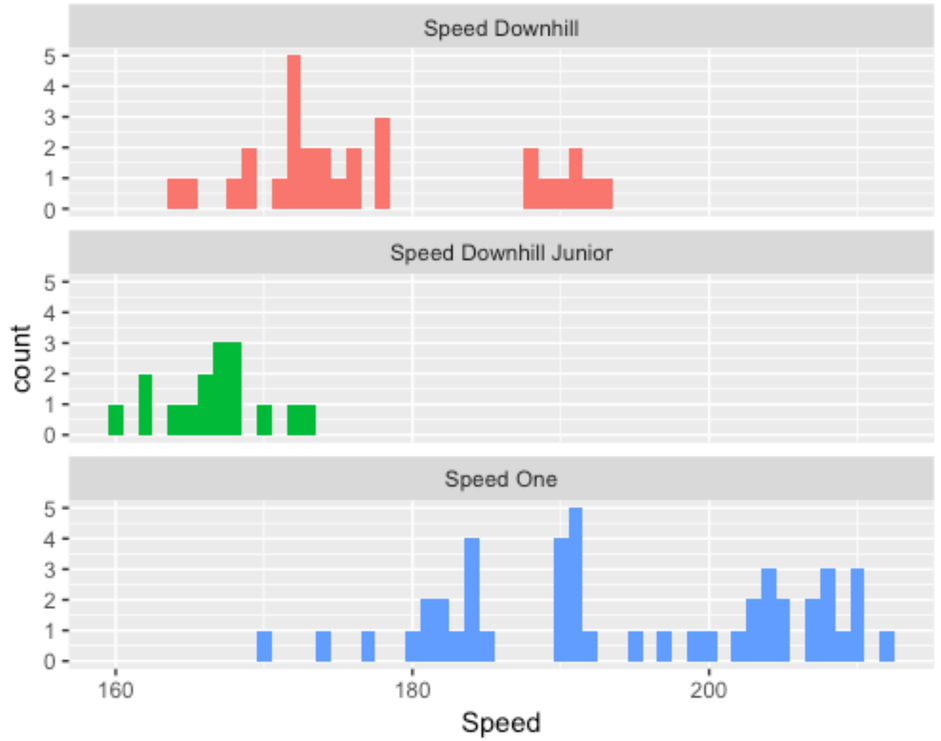
```
(avg <- SpeedSki %>%
  group_by(Event) %>%
  summarise(Mean = mean(Speed),
            Median = median(Speed))) %>%
  arrange(desc(Mean))
# A tibble: 3 × 3
                  Event     Mean  Median
                 <fctr>    <dbl>   <dbl>
1             Speed One 194.5078 191.545
2        Speed Downhill 177.3952 174.240
3 Speed Downhill Junior 166.5813 166.595
```

- Historically Speed One has been the fastest

IS THERE A DIFFERENCE BETWEEN EVENTS?

```
SpeedSki %>%
  ggplot(aes(Speed, fill = Event)) +
  geom_histogram(binwidth = 1,
                 show.legend = FALSE) +
  facet_wrap(~ Event, ncol = 1)
```

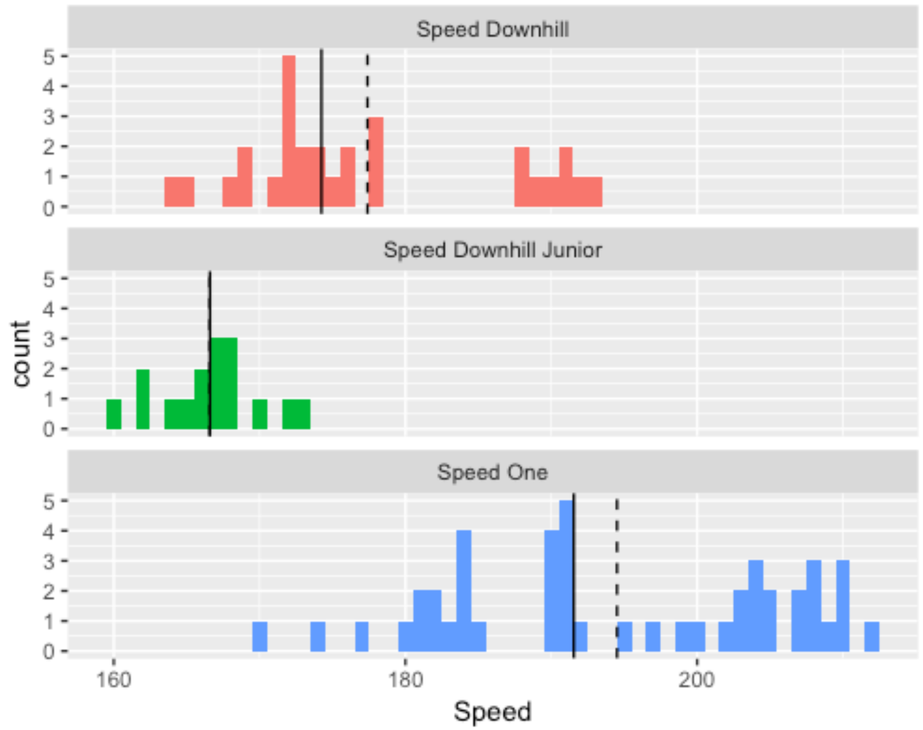
- We can compare distributions between events



IS THERE A DIFFERENCE BETWEEN EVENTS?

```
SpeedSki %>%
  ggplot(aes(Speed, fill = Event)) +
  geom_histogram(binwidth = 1,
                 show.legend = FALSE) +
  geom_vline(data = avg,
             aes(xintercept = Mean),
             linetype = "dashed") +
  geom_vline(data = avg,
             aes(xintercept = Median)) +
  facet_wrap(~ Event, ncol = 1)
```

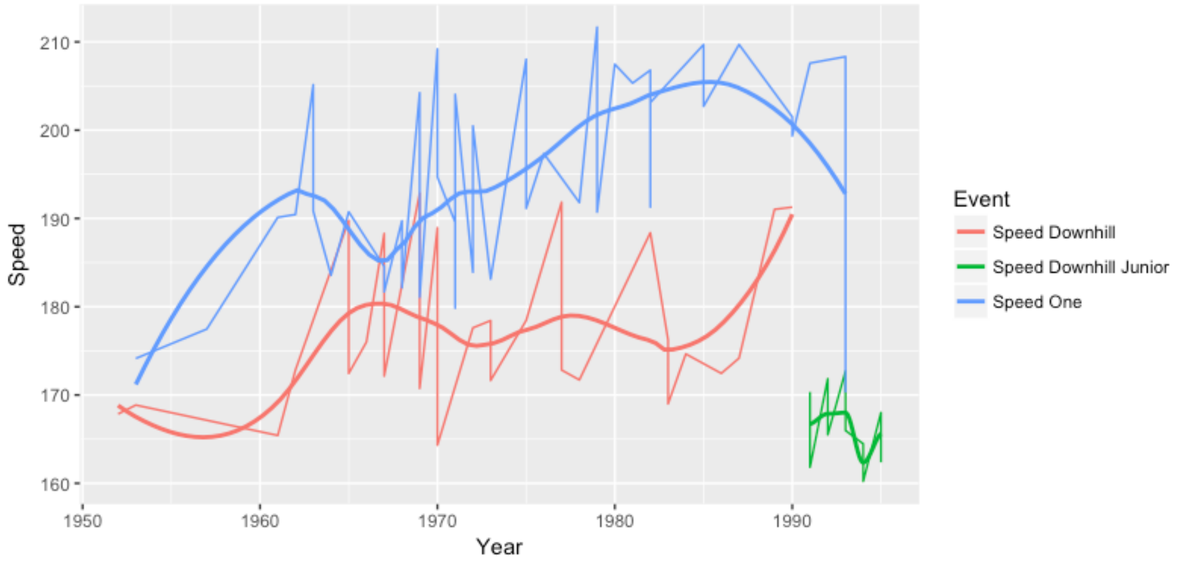
- We can compare distributions between events
- and even add our mean/median lines



IS THERE A DIFFERENCE BETWEEN EVENTS?

```
SpeedSki %>%
  ggplot(aes(Year, Speed, color = Event)) +
  geom_line(show.legend = FALSE) +
  geom_smooth(se = FALSE, span = .5)
```

- Lastly, we can compare how the events differ over time



SPEED SKIERS

1. *How fast are downhill skiers?*
2. *Has the average speed changed over time?*
3. *Is there a difference between events?*
4. *Is there a difference between genders?*
5. *What is driving the difference between genders?*

IS THERE A DIFFERENCE BETWEEN GENDER?

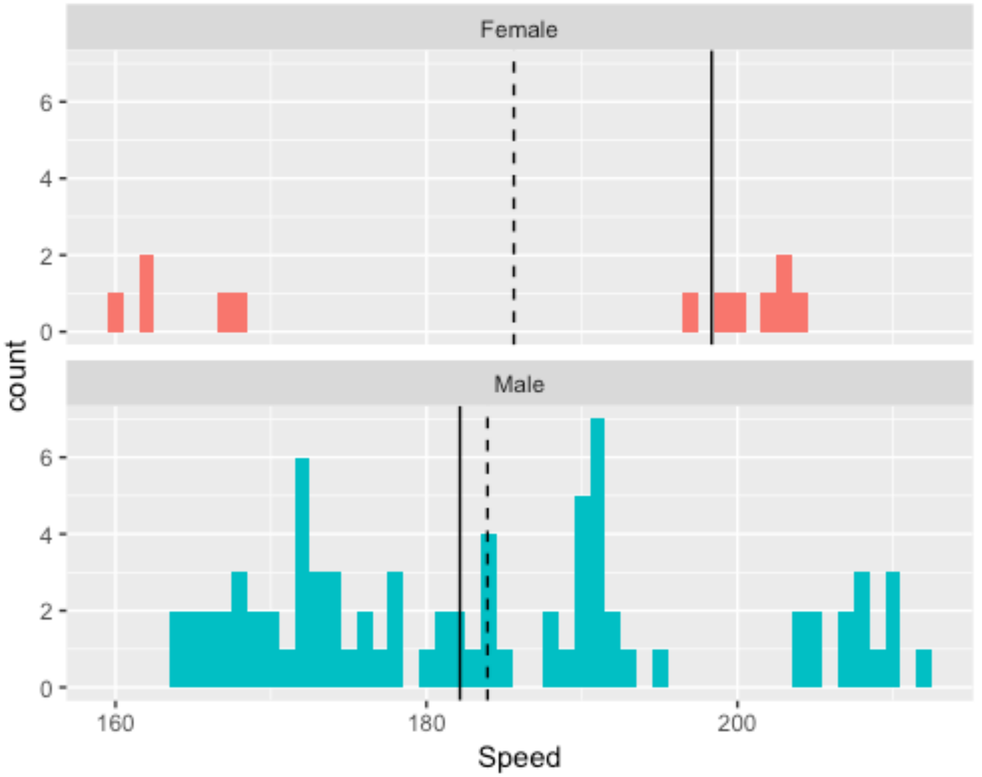
```
(avg <- SpeedSki %>%
   group_by(Sex) %>%
   summarise(Mean = mean(Speed),
             Median = median(Speed))) %>%
  arrange(desc(Mean))
# A tibble: 2 × 3
     Sex     Mean  Median
  <fctr>    <dbl>   <dbl>
1 Female 185.6192 198.345
2   Male 183.9201 182.150
```

- Looks like females have been faster

IS THERE A DIFFERENCE BETWEEN GENDER?

```
SpeedSki %>%
  ggplot(aes(Speed, fill = Sex)) +
  geom_histogram(binwidth = 1,
                 show.legend = FALSE) +
  facet_wrap(~ Sex, ncol = 1)
```

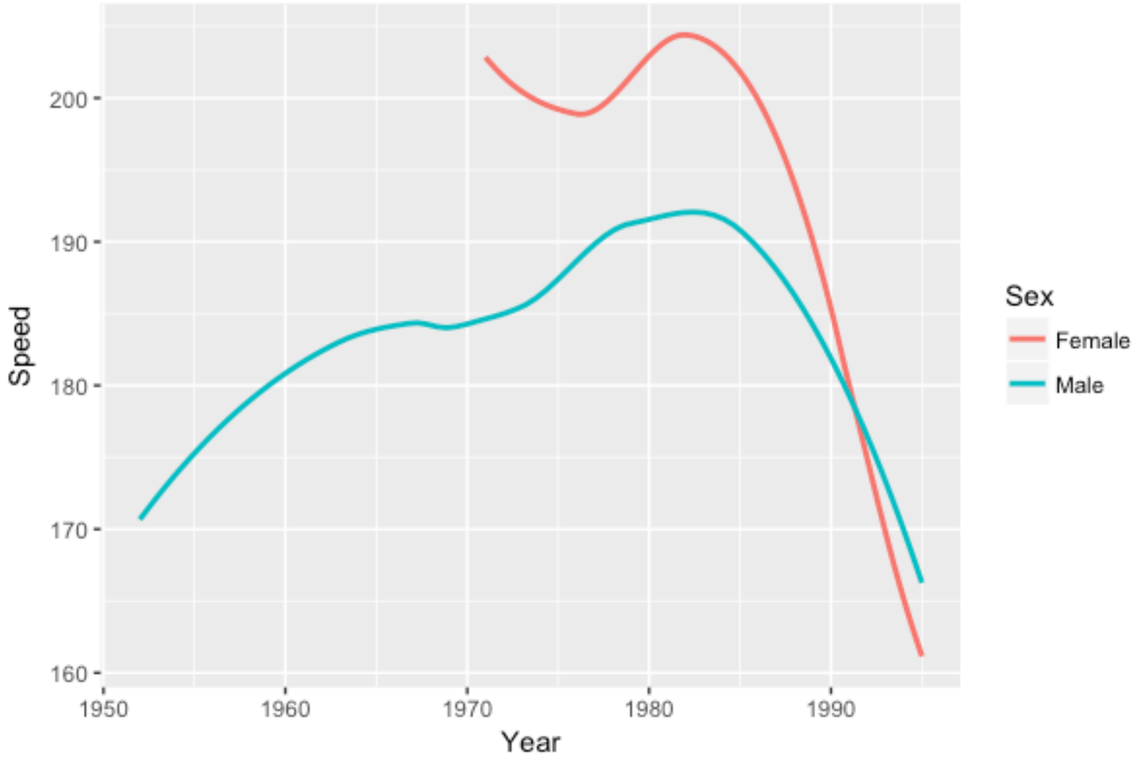
- Looks like females have been faster
- But it looks like there could be other underlying causes for this



IS THERE A DIFFERENCE BETWEEN GENDER?

```
SpeedSki %>%
  ggplot(aes(Year, Speed, color = Sex)) +
  geom_smooth(se = FALSE)
```

- Looks like females have been faster
- But it looks like there could be other underlying causes for this



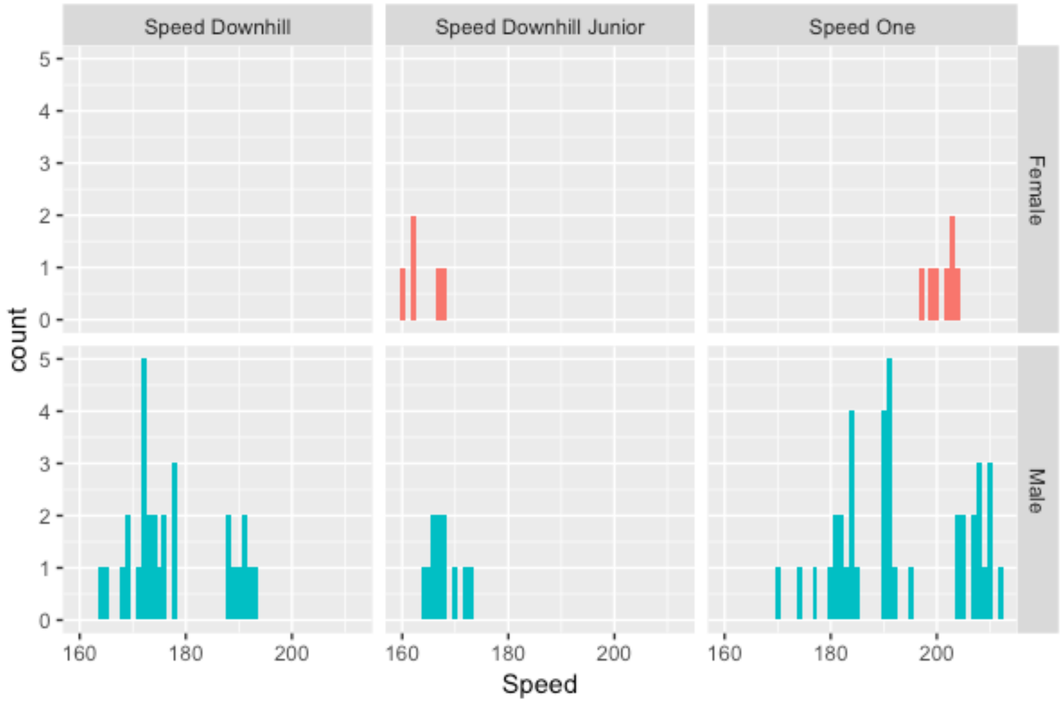
SPEED SKIERS

1. *How fast are downhill skiers?*
2. *Has the average speed changed over time?*
3. *Is there a difference between events?*
4. *Is there a difference between genders?*
5. *What is driving the difference between genders?*

WHY IS THERE A DIFFERENCE BETWEEN GENDER?

```
SpeedSki %>%
  ggplot(aes(Speed, fill = Sex)) +
  geom_histogram(binwidth = 1,
                 show.legend = FALSE) +
  facet_grid(Sex ~ Event)
```

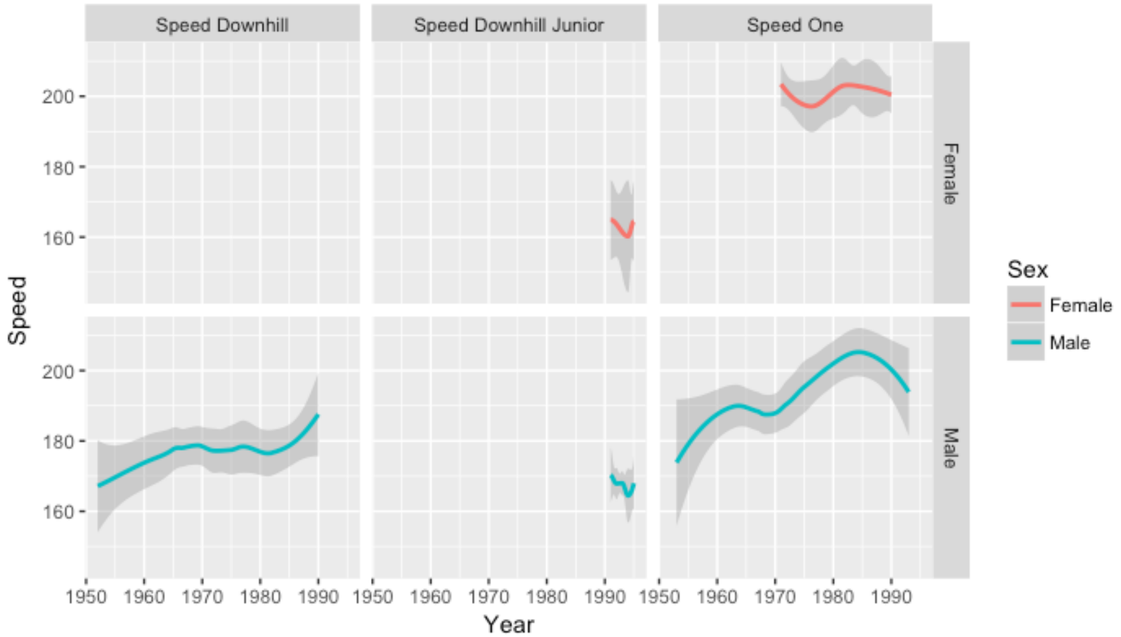
- Females have not raced in the Downhill
- Slightly slower in the Junior
- Females have less observations in the Speed One but have performed well



WHY IS THERE A DIFFERENCE BETWEEN GENDER?

```
SpeedSki %>%
  ggplot(aes(Year, Speed, color = Sex)) +
  geom_smooth() +
  facet_grid(Sex ~ Event)
```

- Females appear to have performed well because their limited observations are more recent

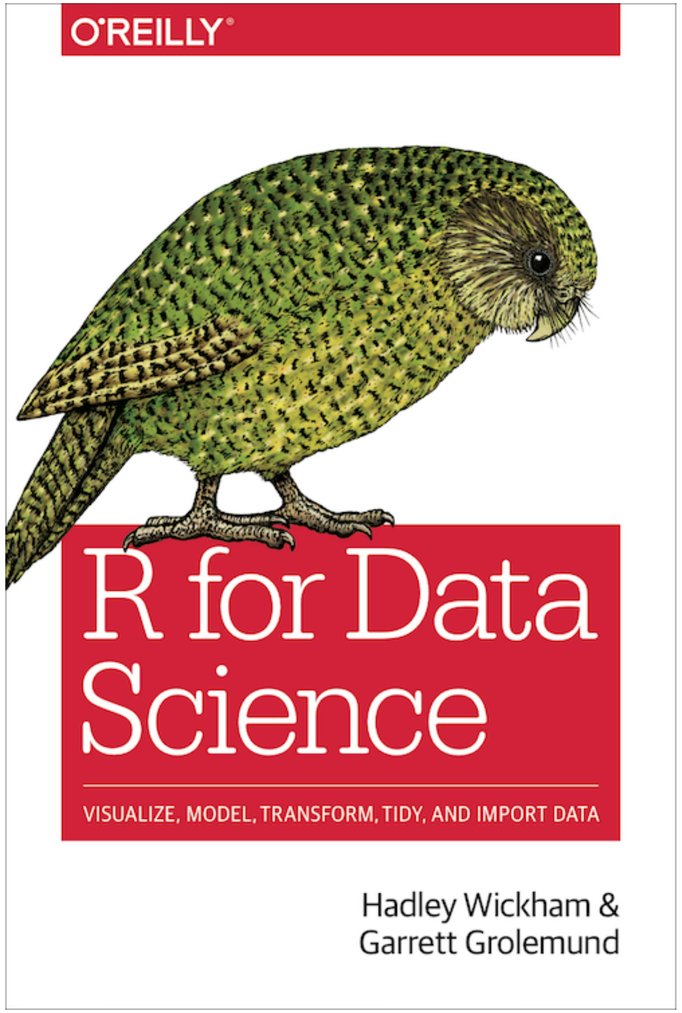


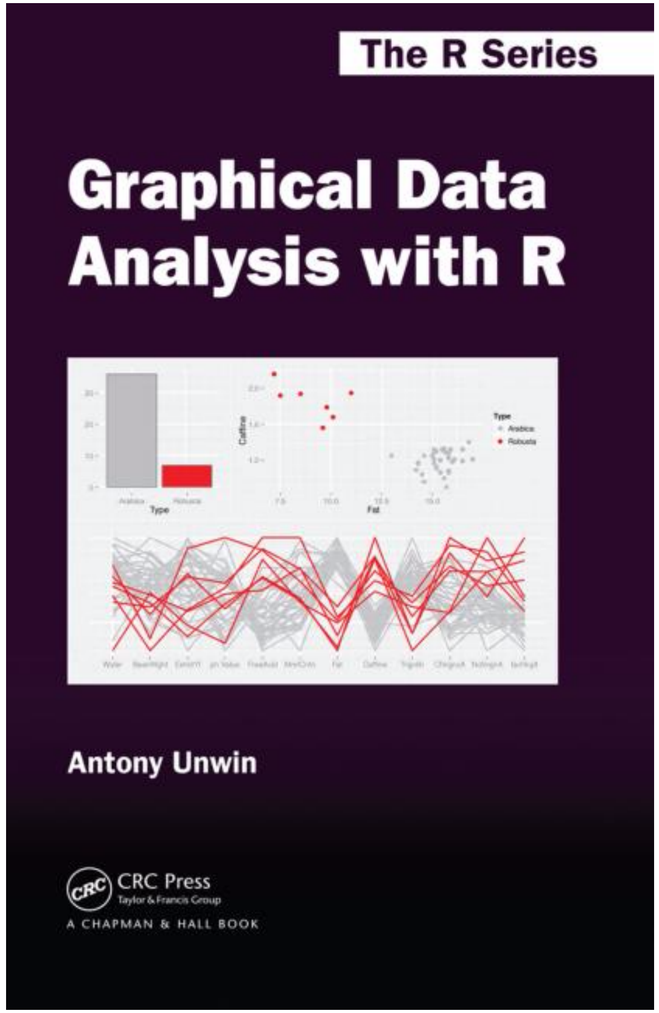
WHY IS THERE A DIFFERENCE BETWEEN GENDER?

- Conclusion:
  - Females have only competed in recent decades
  - Their speeds in Speed One are comparable to the Males
  - Their speeds in the Junior is slightly less than Males
  - The slower speeds that Males had in earlier decades, plus the slower speeds in the Downhill race which the Females do not compete in are pulling the overall averages down for Males

SO LITTLE TIME!

LEARN MORE


O'REILLY
R for Data Science
VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA
Hadley Wickham &
Garrett Grolemund


The R Series
Graphical Data Analysis with R
Antony Unwin
CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

WHAT TO REMEMBER

Combining features of `dplyr` and `ggplot2` is extremely effective and efficient for exploratory data analysis

Learn and internalize them both!