

# DS 116 - Data Visualization

## Visualizing Uncertainty

Habet Madoyan

American University of Armenia

# Visualizing uncertainty

There are many sources of uncertainty

- Uncertainty of point estimates
- Uncertainty of distributions
- Uncertainty of predictions (curve fit) etc..

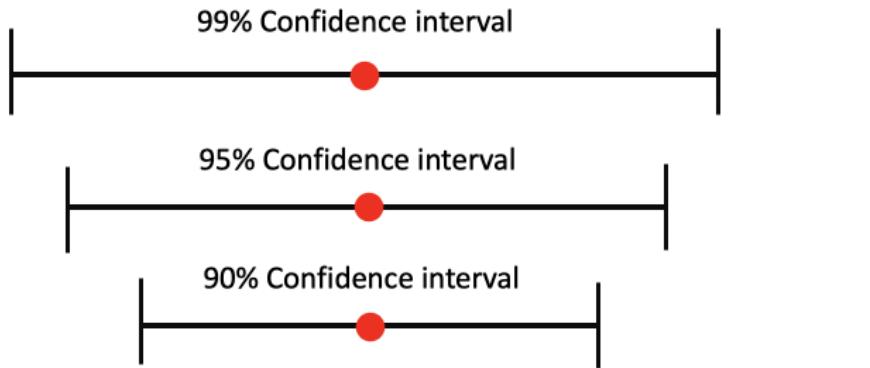
## Section 1

# Uncertainty of point estimates

# Error bars

Uncertainty of point estimates

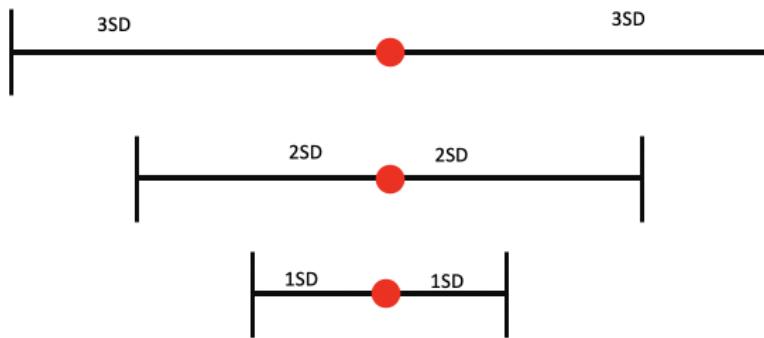
Error bars with **standard errors** (**Confidence intervals**)



# Error bars

Uncertainty with standard deviations

Error bars with **standard deviations**

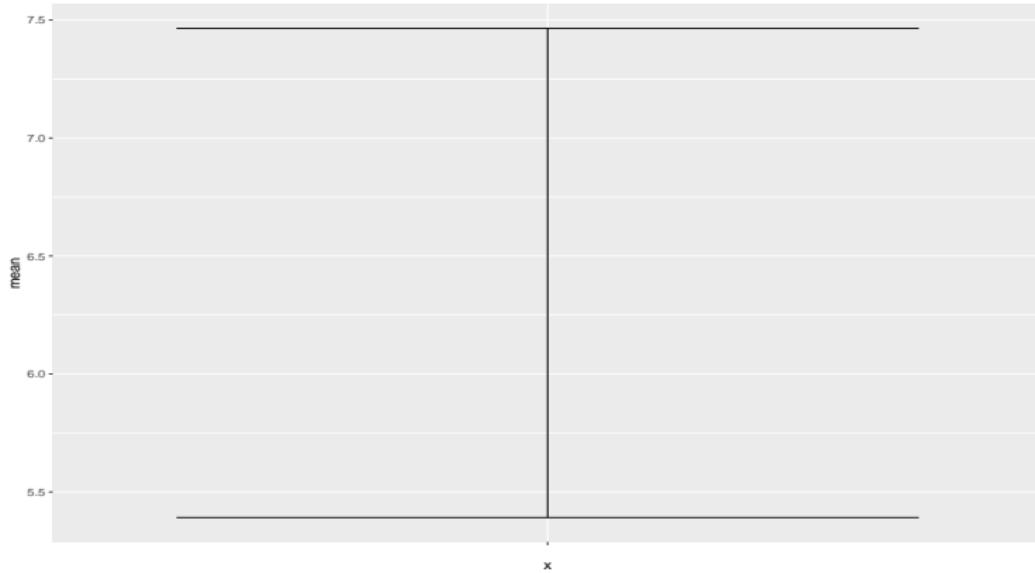


# Error bars

```
movies_small <- read.csv('Data/movies_small.csv')
summarised <- movies_small %>% summarise(mean = mean(imdbRating),
                                             sd = sd(imdbRating))
summarised
##      mean      sd
## 1 6.428149 1.036453
```

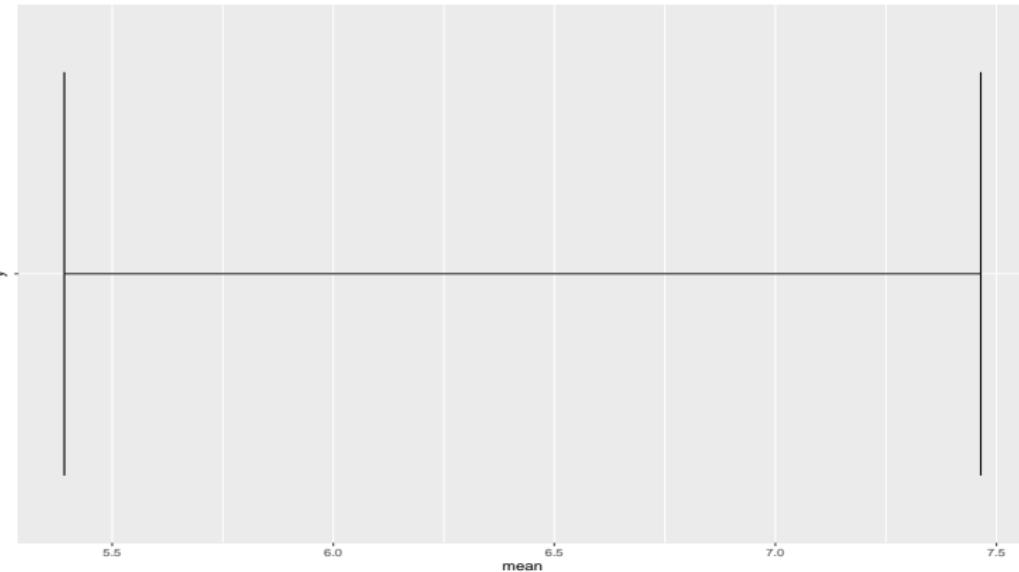
# Error bars

```
ggplot(summarised, mapping = aes(x = "", y = mean, ymax = mean+sd,  
                                 ymin = mean-sd)) + geom_errorbar()
```



# Error bars

```
ggplot(summarised, mapping = aes(x = mean, y = "",  
                                  xmax = mean+sd,  
                                  xmin = mean-sd)) + geom_errorbar()
```



# Error bars

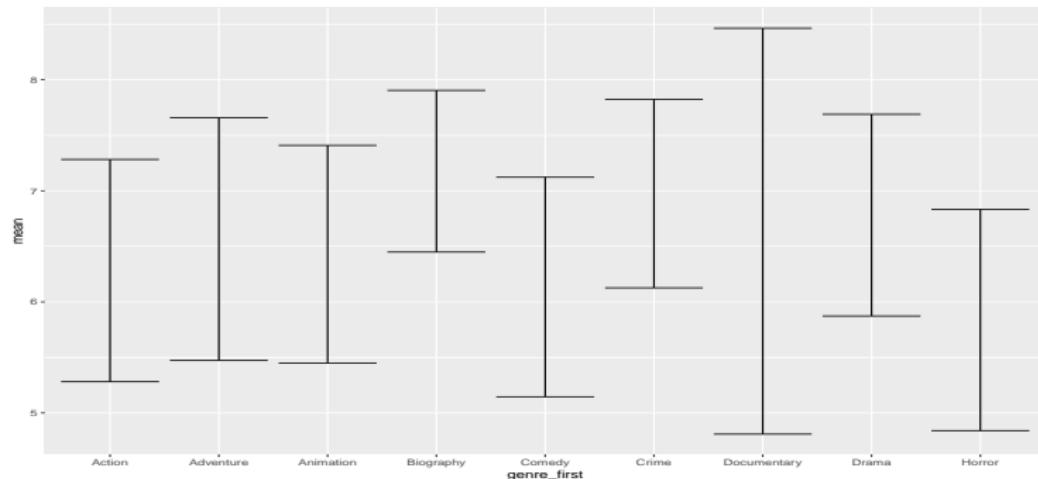
If you want to look at the error bars by groups - summarise by groups using dplyr

```
summarised <- movies_small %>% group_by(genre_first) %>%  
  summarise(mean = mean(imdbRating), sd = sd(imdbRating))  
summarised  
## # A tibble: 9 x 3  
##   genre_first  mean     sd  
##   <chr>        <dbl>  <dbl>  
## 1 Action        6.28  1.00  
## 2 Adventure    6.56  1.09  
## 3 Animation    6.43  0.980  
## 4 Biography    7.18  0.728  
## 5 Comedy        6.13  0.988  
## 6 Crime         6.97  0.848  
## 7 Documentary   6.64  1.83  
## 8 Drama         6.78  0.908  
## 9 Horror        5.84  0.995
```

# Error bars

- use the grouping variable as x aesthetics

```
ggplot(summarised, aes(x = genre_first, y = mean, ymin=mean-sd,  
                         ymax = mean+sd)) + geom_errorbar()
```



# Error bars

There are few functions in R that can be used for generating the data for error bars

- $\pm 1 \text{ SD}$

```
smean.sdl(movies_small$imdbRating, mult = 1)
##      Mean     Lower     Upper
## 6.428149 5.391696 7.464602
```

- $\pm 2\text{sd}$

```
smean.sdl(movies_small$imdbRating, mult = 2)
##      Mean     Lower     Upper
## 6.428149 4.355243 8.501055
```

# Error bars

Hmisc also has functions to construct Confidence Intervals

- 95%

```
smean.cl.normal(movies_small$imdbRating, conf.int = 0.95)
##      Mean     Lower     Upper
## 6.428149 6.384806 6.471493
```

- 99% confidence interval

```
smean.cl.normal(movies_small$imdbRating, conf.int = 0.99)
##      Mean     Lower     Upper
## 6.428149 6.371168 6.485130
```

# Error bars

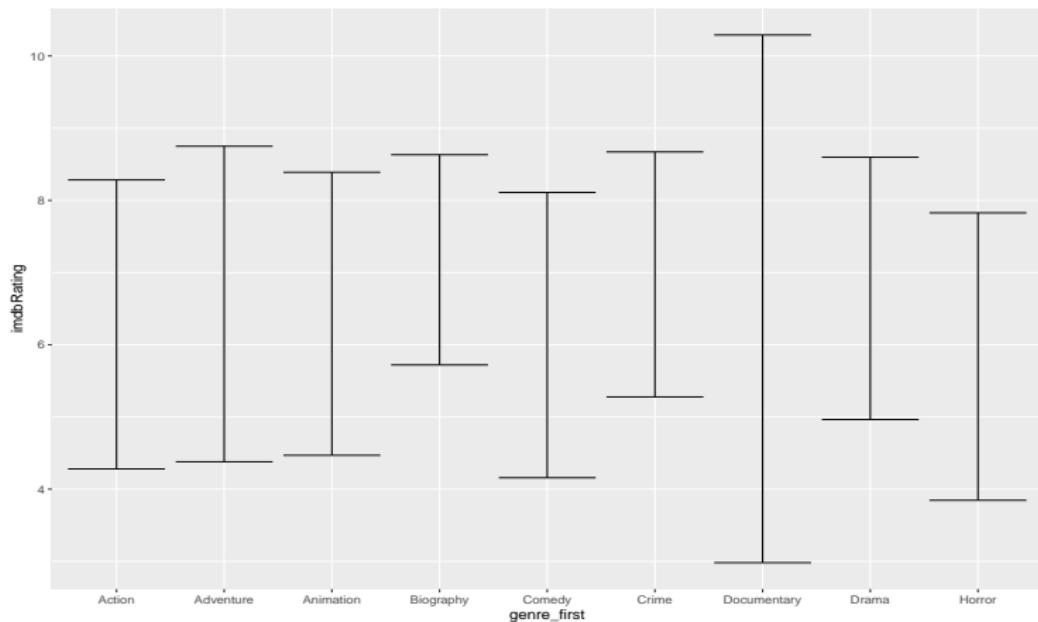
- ggplot2 has a wrapper for Hmisc functions: `mean_cl_normal`, `mean_sdl`, `mean_se`
- To construct the error bars with the functions we will use `stat_summary()` layer

`stat_summary()` applies defined function on y by the given values of x. No need to summarise with dplyr and create new dataframe

# Error bars

Error bars for imdbRating by genre (+- 2 SD)

```
ggplot(movies_small, aes(x = genre_first, y = imdbRating)) +  
  stat_summary(fun.data = mean_sdl, geom = "errorbar")
```

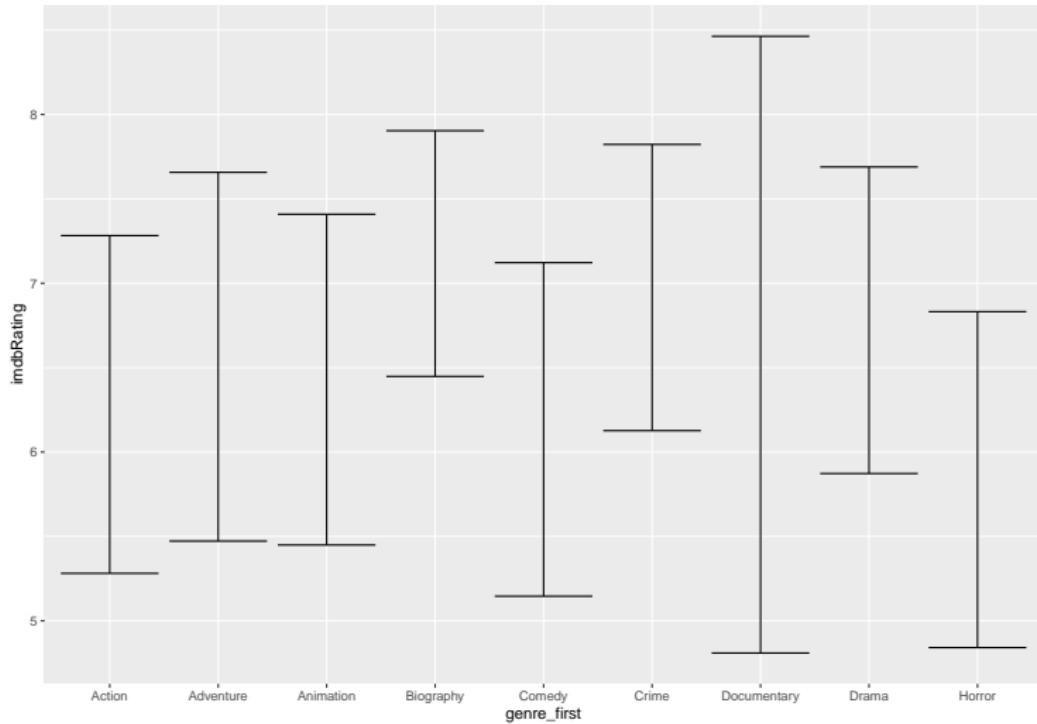


# Error bars

- By default, `mult = 2`.
- If you want to change this, use `fun.args` in `stat_summary()`

```
ggplot(movies_small, aes(x = genre_first, y = imdbRating)) +  
  stat_summary(fun.data = mean_sdl, geom = "errorbar",  
              fun.args = list(mult = 1))
```

# Error bars

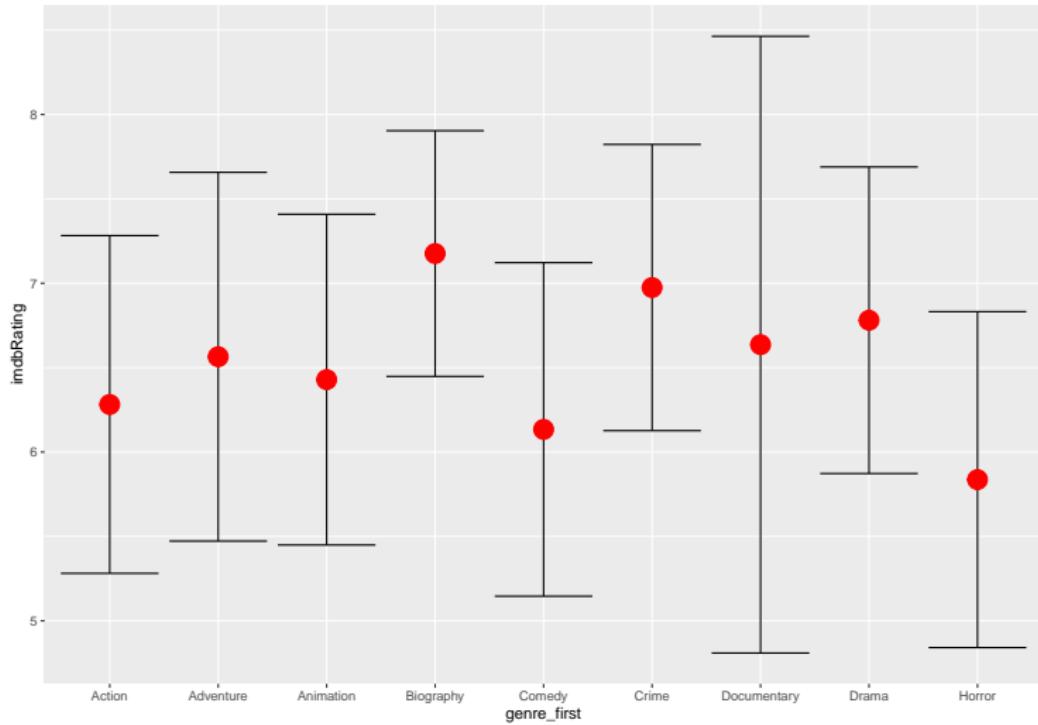


# Error bars

use `stat_summary()` again to add the mean as a red point to the error bar

```
ggplot(movies_small, aes(x = genre_first, y = imdbRating)) +
  stat_summary(fun.data = mean_sdl, geom = "errorbar",
              fun.args = list(mult = 1)) +
  stat_summary(fun = mean, geom = 'point', color = 'red', size = 6)
```

# Error bars



## Section 2

# Confidence intervals

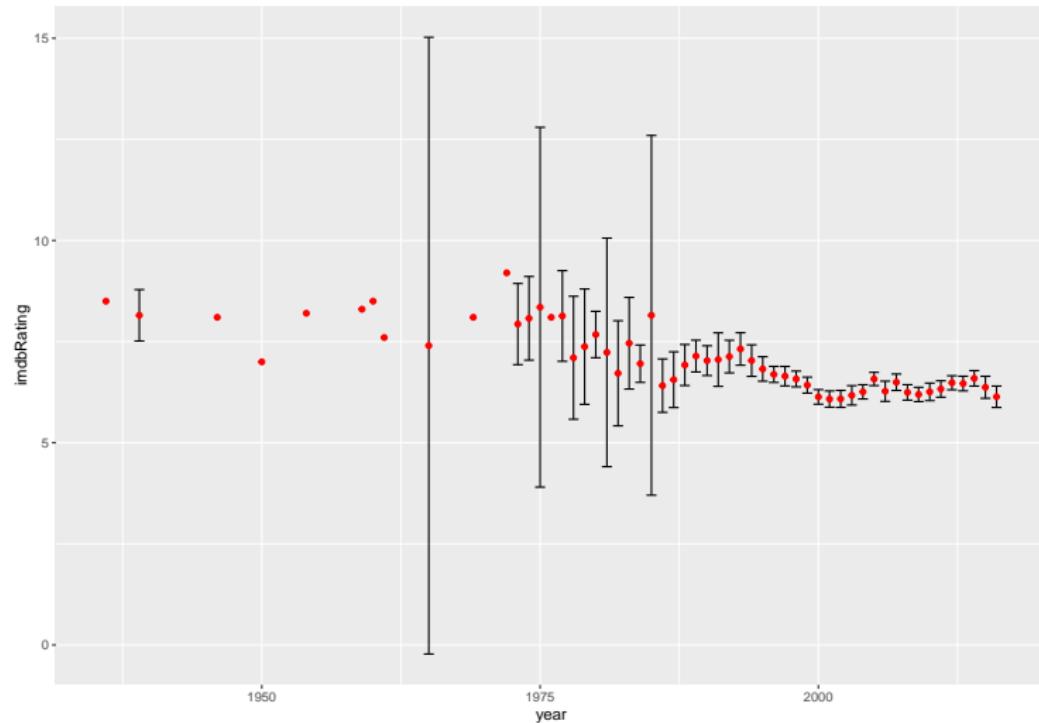
# Confidence intervals

We can visualize confidence intervals with error bars as well

- `mean_cl_normal` will create 95% CI by default

```
ggplot(movies_small, aes(x = year, y = imdbRating)) +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar") +
  stat_summary(fun = mean, geom='point', color = 'red')
```

# Confidence Intervals

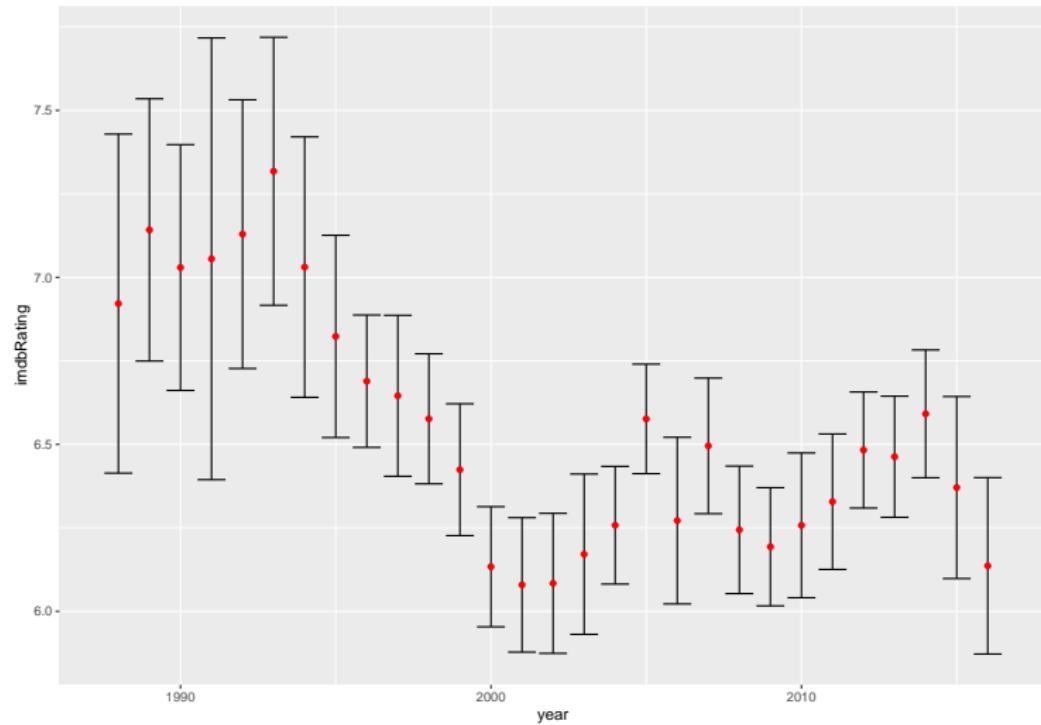


# Confidence Intervals

Subset the data with the years starting from 1987

```
movies_small %>% filter(year > 1987) %>%  
  ggplot(aes(x = year, y = imdbRating)) +  
    stat_summary(fun.data = mean_cl_normal, geom = "errorbar") +  
    stat_summary(fun = mean, geom='point', color = 'red')
```

# Confidence Intervals



# Confidence Intervals

Error bars with confidence intervals can help us to do initial hypothesis testing

$$H_0 : \mu_0 = \mu_1$$

$$H_1 : \mu_0 \neq \mu_1$$

We can look at the error bars

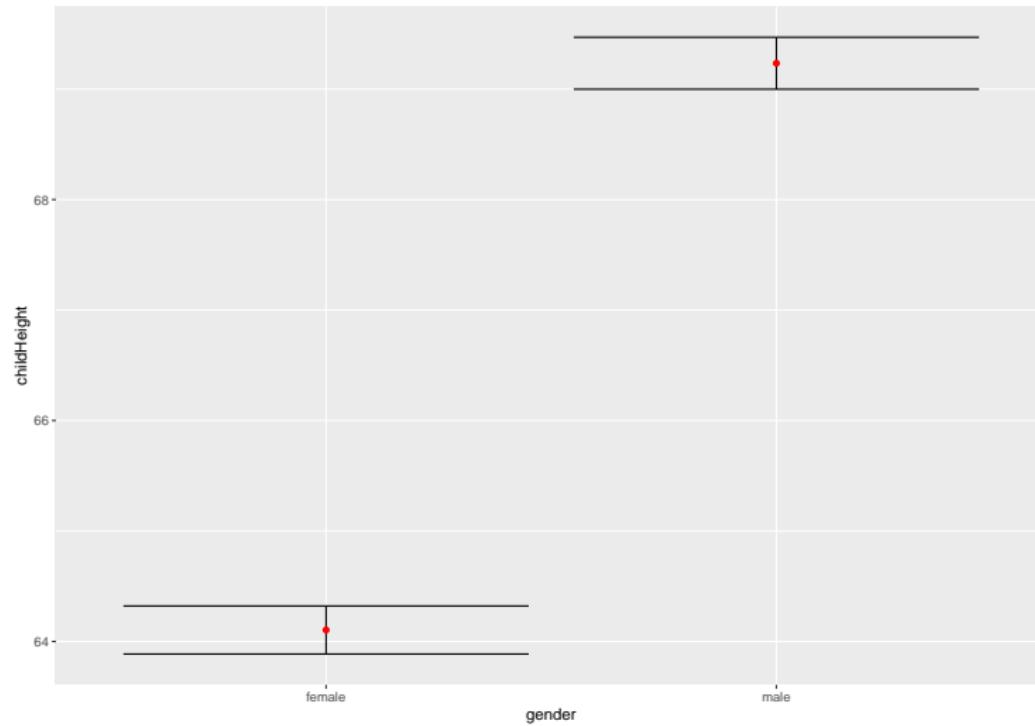
- No intersection of the bars is an indicator of rejecting the null hypothesis
- Larger is the gap between error bars, more likely you will reject  $H_0$

# Confidence intervals

Confidence intervals for Height by gender

```
ggplot(GaltonFamilies, aes(x = gender, y = childHeight)) +  
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar") +  
  stat_summary(fun = mean, geom='point', color = 'red')
```

# Confidence Intervals



# Confidence Intervals

Testing the hypothesis

$$H_0 : \mu_{male} = \mu_{female}$$

$$H_1 : \mu_{male} \neq \mu_{female}$$

```
t.test(GaltonFamilies$childHeight ~ GaltonFamilies$gender)
##
##  Welch Two Sample t-test
##
## data: GaltonFamilies$childHeight by GaltonFamilies$gender
## t = -31.476, df = 929.89, p-value < 0.000000000000022
## alternative hypothesis: true difference in means between group female and
## 95 percent confidence interval:
## -5.449979 -4.810266
## sample estimates:
## mean in group female   mean in group male
##                 64.10397          69.23410
```

# Confidence Intervals

Look at the mtcars data

- hp Horsepower
- am transmission automatics vs manual

First - t.test

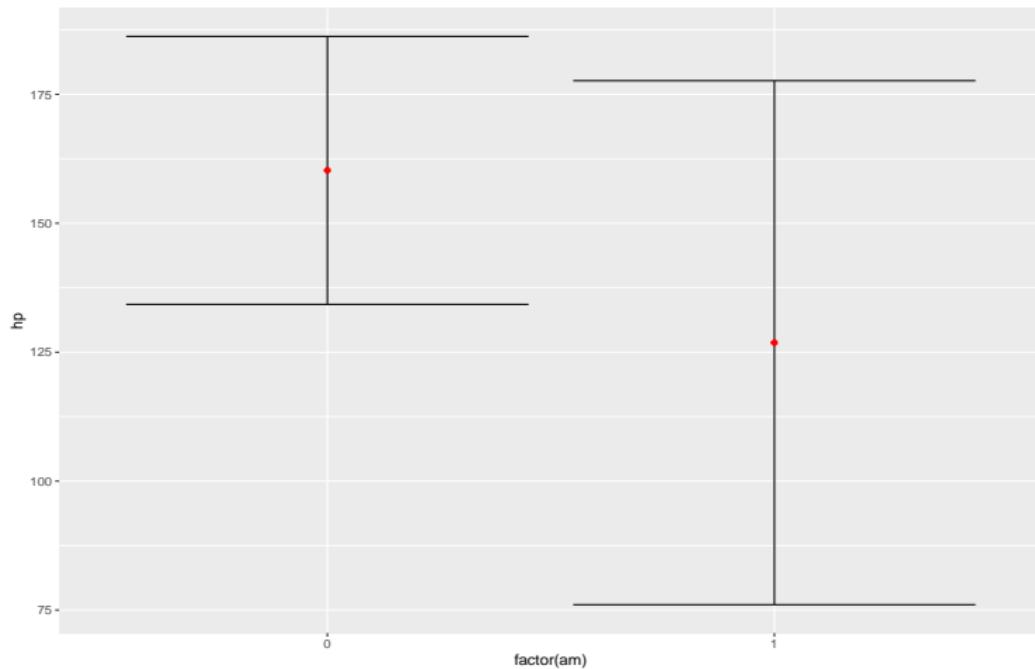
```
t.test(mtcars$hp~mtcars$am)
##
##  Welch Two Sample t-test
##
## data: mtcars$hp by mtcars$am
## t = 1.2662, df = 18.715, p-value = 0.221
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
## -21.87858  88.71259
## sample estimates:
## mean in group 0 mean in group 1
##          160.2632          126.8462
```

# Confidence Intervals

## Confidence interval

```
ggplot(mtcars, aes(x = factor(am), y = hp)) +
  stat_summary(fun.data = mean_cl_normal, geom = "errorbar") +
  stat_summary(fun = mean, geom='point', color = 'red')
```

# Confidence Intervals



## Section 3

# Uncertainty in curve fit

# Uncertainty in curve fit

When we fit a model into the data, we sometimes look for answers to the following questions:

- What is the mean response for a particular value of  $x$ ?
- In which interval will these values lie

# Uncertainty in curve fit

To answer these questions, we can construct confidence interval around the regression line, using the following formula:

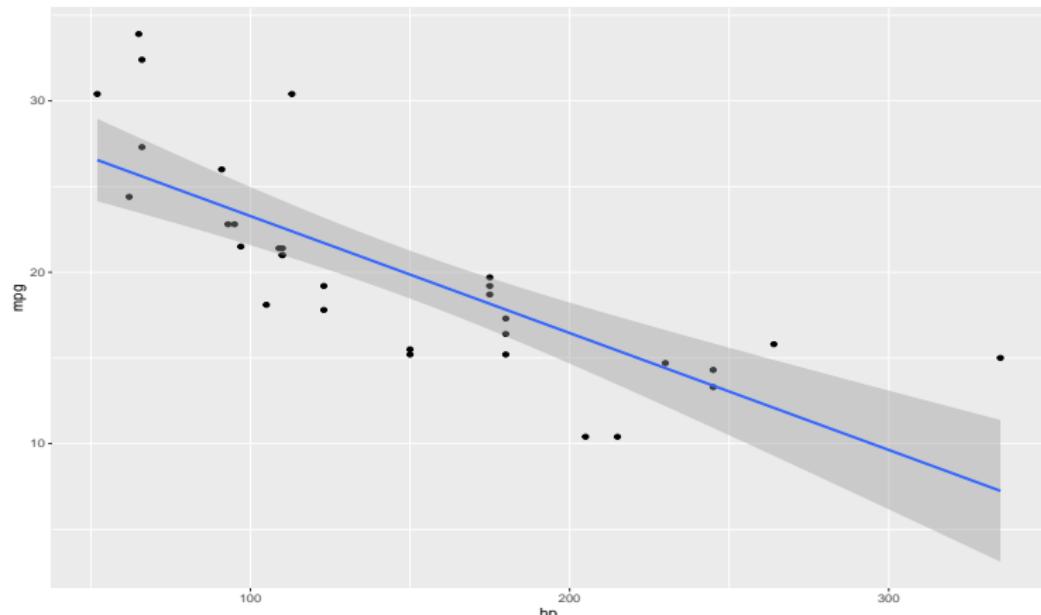
$$\hat{y}_h \pm t_{\alpha/2, n-2} \sqrt{MSE \left( \frac{1}{n} + \frac{(x_k - \bar{x})^2}{\sum(x_i - \bar{x})^2} \right)}$$

Wider is the confidence interval, more uncertainty we would have for that specific  $x$  value.

# Uncertainty in curve fit

To get the confidence interval on the graph, specify `se=TRUE` (the default option) either in `stat_smooth` or `geom_smooth`

```
ggplot(mtcars, aes(x = hp, y = mpg)) + geom_point() +  
  geom_smooth(method = 'lm')
```



# Uncertainty in curve fit

As you can see from the formula, further away you go from the mean, larger becomes the confidence interval

```
ggplot(mtcars, aes(x = hp, y = mpg)) + geom_point() +  
  geom_smooth(method = 'lm') +  
  geom_vline(xintercept = mean(mtcars$hp), color = 'red')
```

