# Psych 100A Spring 2019: Week 3 Slides

Amanda Montoya

April 16, 2019

# While we're setting up

```
NLSdata <- read.csv("http://bit.ly/NLSdata", header=TRUE)
```

# Learning Outcomes Today

- Compare statistics to parameters, be able to provide examples of each
- Explain why the mean is used as a model in statistics
- Estimate a Null/Simple model of a continuous outcome
- Connect the concepts of a model, prediction, residual, and error.

Setting up R

```
NLSdata <- read.csv("http://bit.ly/NLSdata", header=TRUE)
```

# Statistics vs. Parameters

Statistics courses are focused around using information from a sample to make educated guesses about what's going on in the population. This process is called **inference.**

If we want to know a value in the population (e.g., the population mean $\mu$), then we calculate the equivalent value from a sample (e.g., sample mean, $\bar{Y}$).

Anything that we calculate from our sample is a **statistic.** Whereas values from the population are called **parameters.**

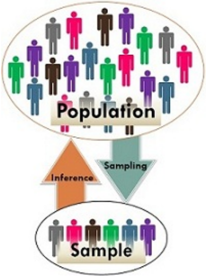


Figure 1: Populations to Samples

## Statistics vs. Parameters

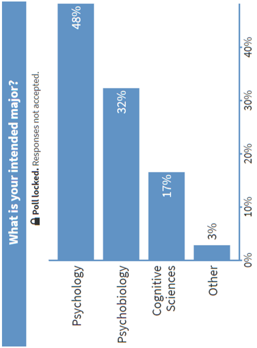
We've seen examples of statistics and parameters already!

**Sample**

Psychology: 40%. Psychobiology: 25%, Cognitive Science: 30%, Other 5%

40%, 25%, 30%, and 5% are all **statistics** because they're calculated from a sample (subset) of the population, and they're only an **estimate** about what's going on in the population

48%, 32%, 17%, and 3% are **parameters** since they come from the entire population.



# Differentiating parameters and statistics

- A. The average score of 1,211 students at a particular elementary school was 3.74 on a standardized test. We know this because we have each and every students' test score.
- B. A sample of US residents suggests that 60% agree with the latest health care proposal.
- C. Based on the Intuit payroll, the median pay for a data analyst at Intuit is $134,709

Which of the above are parameters, which are statistics?

# Understanding the mean

Lots of the time we're interested in knowing the average of a population ($\mu$). So when we collect a sample, we estimate $\mu$ using the sample mean $\bar{Y}$.

$$Sample.Average = \frac{Sum.of.values.over.all.cases}{Number.of.Cases} = \frac{\sum_{i=1}^{n} y_i}{n} = \bar{Y}$$

$n =$ Number of cases.

Averages are easier to predict than individual cases. The more things that we average over, the closer we get to the population mean.

# Building Models with averages

Averages are easier to predict than individual cases. The more things that we average over, the closer we get to the population mean.

Instead of trying to build a model that creates a unique prediction for each person, we can create a model using an average. This should work well for many people, and have good properties.

Regardless of how many people you collect, individuals will always vary, but when we get more people averages will become less variable.

# Averages ($\bar{Y}$)

Averages are easier to predict than individual cases. The more things that we average over, the closer we get to the population mean.

The average height in the US is 5' 6.5" $=$ 66.5 inches.

Click in with how close your height is to the population average.

# Averages of Height

Find 2 - 3 people near you, calculate what your average height is as a group (I recommend using inches).

For example in a group of 3 with peoplewho are 57.5 inches, 69 inches, and 67 inches, there average height would be:

$\frac{57.5+69+67}{3} = 64.5$ inches

Calculate your average in your group. Is the average of your group closer to the average population height (66.5”) than your original height?

Notice that not all students in the same group will answer the same. Everyone click in!

# Average of Height

What if you had a larger group? Try combining your average with the average of a group around you!

Take the average from Group 1 add it to the average from Group 2, and divide by two.

$$\frac{\bar{Y}_1 + \bar{Y}_2}{2}$$

Is the average of your group closer to the average population height (66.5") than your original height?

Notice that not all students in the same group will answer the same. Everyone click in!

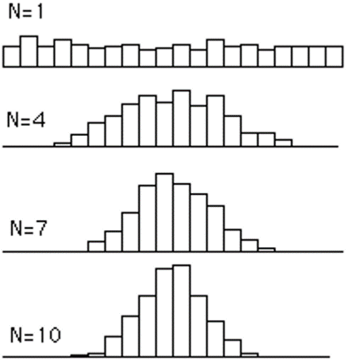
# Variability and sample size

Notice that the individual scores we were dealing with didn't change. The individual values didn't get "less variable."

But when we compute averages of more and more things, the averages get closer and closer to the "true value."

Regardless of how many people you collect, individuals will always vary, but when we get more people averages will become less variable.

We can take advantage of this nice properties of means when we use statistical models.



# The Mean as a Model

Statistical models are used to describe a distribution.

We use statistical models to create a prediction for each observation in the data.

The mean is a simple model where we predict the same value for each individual.

```
smallNLS <- sample(select(NLSdata, HrsSleep2009), 20)
Sleepstats <- favstats(smallNLS$HrsSleep2009)
print(Sleepstats)
```

```
##  min Q1 median Q3 max mean       sd  n missing
##    5  6      7  8   8 6.65 1.136708 20       0
```

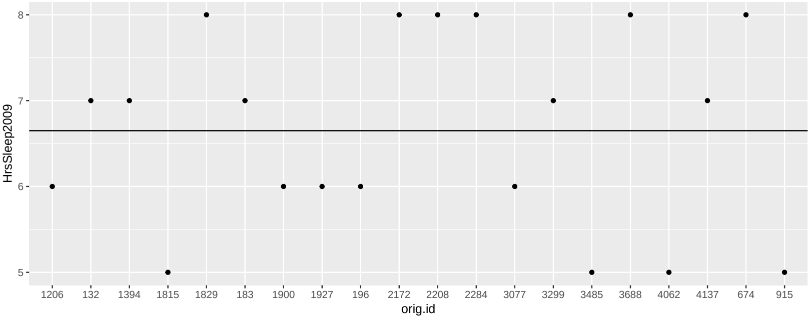
# The Mean as a Model

Statistical models are used to describe a distribution.

We use statistical models to create a prediction for each observation in the data.

The mean is a simple model where we predict the same value for each individual.

```r
gf_point(HrsSleep2009~orig.id, data = smallNLS) %>%
  gf_hline(yintercept = Sleepstats$mean)
```



# Why use the mean?

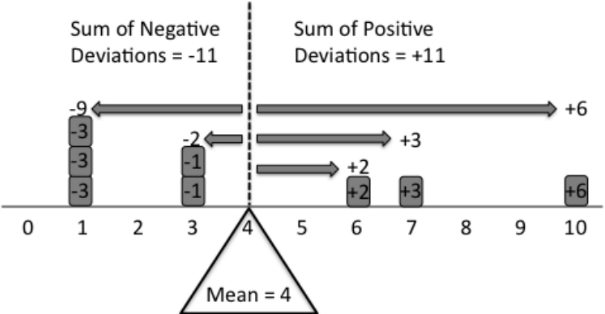
The mean is a good model for the data because it minimizes **sums of squared error.**

$$\text{Outcome} = \text{Model} + \text{Error}$$

We want a model which has minimal error.

The distance between the prediction (model) and the observed value (outcome) is the error.

The mean ensures that the positive errors and negative errors are balanced with regard to their magnitude.


Sum of Negative Deviations = -11
Sum of Positive Deviations = +11
-9
+6
-2
+3
+2
-3
-3
-3
-1
-1
+2
+3
+6
0
1
2
3
4
5
6
7
8
9
10
Mean = 4

# Fitting the model

We can use the `lm` function to fit the model with no predictors (Null Model / Empty model)

```
empty.model <- lm(HrsSleep2009~NULL, data = smallNLS)
empty.model
```

```
##
## Call:
## lm(formula = HrsSleep2009 ~ NULL, data = smallNLS)
##
## Coefficients:
## (Intercept)
##        6.65
```

```
favstats(~HrsSleep2009, data = smallNLS)
```

```
##  min Q1 median Q3 max mean       sd  n missing
##    5  6      7  8   8 6.65 1.136708 20       0
```

# Intercept vs. Mean

In the empty model the “intercept” is the same as the mean of the outcome!

The mean is our model. It is out prediction for each person.

```r
smallNLS$pred <- predict(empty.model)
smallNLS
```

```
##      HrsSleep2009 orig.id pred
## 1394            7    1394 6.65
## 2284            8    2284 6.65
## 915             5     915 6.65
## 1829            8    1829 6.65
## 3077            6    3077 6.65
## 1927            6    1927 6.65
## 1206            6    1206 6.65
## 196             6     196 6.65
## 3299            7    3299 6.65
## 4062            5    4062 6.65
## 1815            5    1815 6.65
## 674             8     674 6.65
## 3688            8    3688 6.65
## 1900            6    1900 6.65
## 3485            5    3485 6.65
## 4137            7    4137 6.65
```

# Residuals/Errors

The residuals (errors) are the differences between the prediction (model) and the outcome.

$$Y_i - \bar{Y} = e_i$$

```
smallNLS$resid <- resid(empty.model)
smallNLS
```

```
##      HrsSleep2009 orig.id pred resid
## 1394            7    1394 6.65  0.35
## 2284            8    2284 6.65  1.35
## 915             5     915 6.65 -1.65
## 1829            8    1829 6.65  1.35
## 3077            6    3077 6.65 -0.65
## 1927            6    1927 6.65 -0.65
## 1206            6    1206 6.65 -0.65
## 196             6     196 6.65 -0.65
## 3299            7    3299 6.65  0.35
## 4062            5    4062 6.65 -1.65
## 1815            5    1815 6.65 -1.65
## 674             8     674 6.65  1.35
## 3688            8    3688 6.65  1.35
## 1900            6    1900 6.65 -0.65
```

# Sums of Squares

If we take the residuals and add them all together, we'll just get zero! Try it!

Remember the mean balances the positive and negative residuals based on magnitude.

```r
sum(smallNLS$resid)
```

```
## [1] 1.054712e-15
```

So to know how big the residuals are we calculate the **sum of the squared residuals**

$$e_i^2 = SS_{residual}$$

$SS_{residual}$ is a measure of how close the observed data are to the prediction. When $SS_{residual}$ is big that means there is a lot of error. But when it's small, there is only a little bit of error.

# General linear model notation

Throughout this class we will be using the *general linear model.*

This is a broad type of models where the outcome is always a linear function of the predictors (if there are any).

We use a general notation where the intercept is always: $b_0$.

The empty model can be represented as

$$Y_i = b_0 + e_i$$

To differentiate between sample estimates and pouplation values, we use greek letters to indicate population parameters:

$$Y_i = \beta_0 + \epsilon_i$$

# Parameters vs. Statistics

$$Y_i = \beta_0 + \epsilon_i$$

**Think, pair, share**: Why is $Y_i$ not a greek letter, but $\epsilon_i$ is a greek letter?

# Explaining Variability

Does knowing someone's value on an explanatory variable, give us information about their value on the outcome variable?

# Why do we need a simple model?

Ultimately, we want to know if an explanatory variable predicts an outcome variable. We can create a model, that allows this to happen, but then we need to know if it's a good model!

To know if the explanatory model is good, we need something to compare it to.

We'll compare the explanatory model to a **simple model** (AKA **Null model**)

The Null Model is a very basic model: The best guess for the outcome for any observation is the average (mean) of the outcome.

# Hours of Sleep 2009

Let's look at what this model would be for Hours of Sleep in 2009

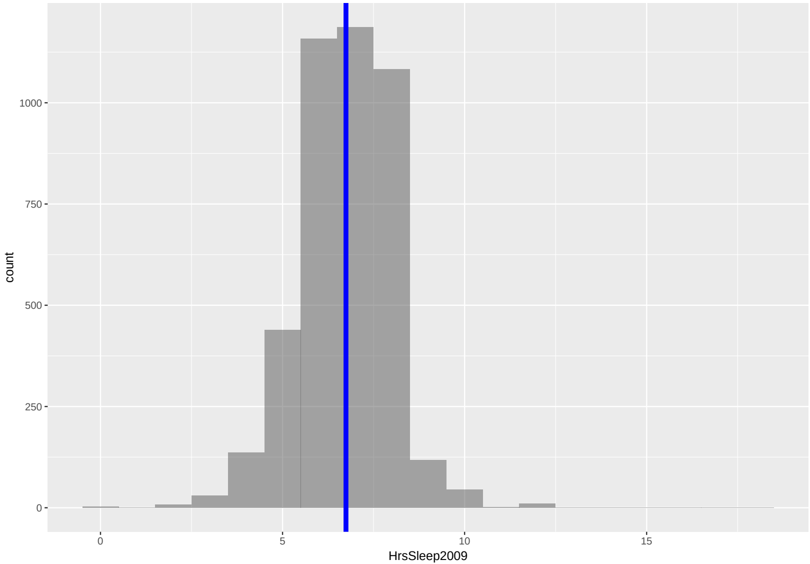
```
SleepStats <- favstats(NLSdata$HrsSleep2009)
print(SleepStats)
```

```
##  min Q1 median Q3 max     mean       sd    n missing
##    0  6      7  8  18 6.741951 1.305077 4224       0
```

Regardless of any explanatory variables, the average hours of sleep in 2009 is 6.7419!

# Adding prediction to the plot

```
gf_histogram(~HrsSleep2009, data = NLSdata, binwidth = 1) %>%
  gf_vline(xintercept=~mean,data =SleepStats,color="blue",size=2)
```



# Explanatory Variable: Two groups

When there are two groups, we can use the mean from each group as the prediction, and we can compare this to the simple model

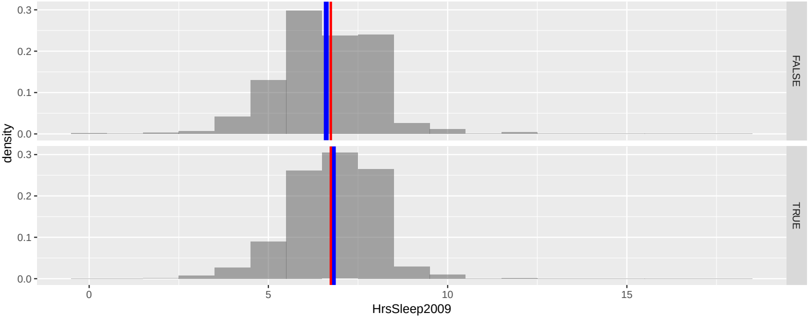
```
SleepCohabStats <- favstats(HrsSleep2009~Cohab2009, data = NLSdata)
SleepCohabStats
```

```
##   Cohab2009 min Q1 median Q3 max     mean       sd    n missing
## 1     FALSE   0  6      7  8  12 6.613682 1.361868 1491       0
## 2      TRUE   0  6      7  8  18 6.811928 1.267820 2733       0
```

# Explanatory Variable: Two groups

When there are two groups, we can use the mean from each group as the prediction, and we can compare this to the simple model

```
##   Cohab2009 min Q1 median Q3 max     mean       sd    n missing
## 1     FALSE   0  6      7  8  12 6.613682 1.361868 1491       0
## 2      TRUE   0  6      7  8  18 6.811928 1.267820 2733       0
```



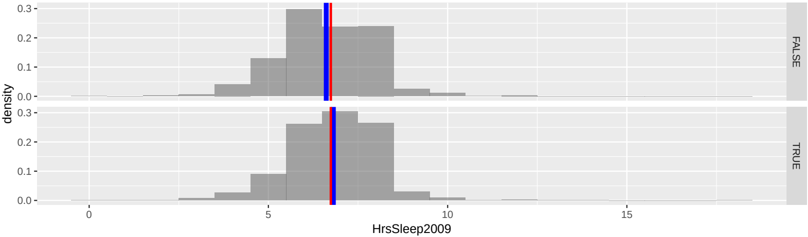
# Explanatory Variable: Two groups

When there are two groups, we can use the mean from each group as the prediction, and we can compare this to the simple model

$$Sleep = Cohab + Error$$

$$Y_i = b_0 + b_1 X_i + e_i$$

```
gf_dhistogram(~HrsSleep2009, data = NLSdata, binwidth = 1)%>%
gf_facet_grid(Cohab2009 ~ .) %>%
gf_vline(xintercept=~mean,data=SleepCohabStats,color="blue",size=2)%>%
gf_vline(xintercept=~mean,data=SleepStats,color = "red",size=1)
```



# Explanatory Variable: More groups

When there are more groups, we can use the mean from each group as the prediction, and we can compare this to the simple model

```r
SleepEthnStats <- favstats(HrsSleep2009~Ethnicity, data = NLSdata)
select(SleepEthnStats, Ethnicity, mean)
```

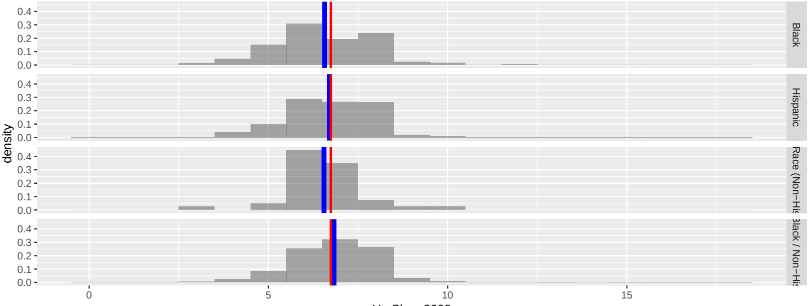
```
##                     Ethnicity     mean
## 1                       Black 6.566290
## 2                    Hispanic 6.702206
## 3 Mixed Race (Non-Hispanic) 6.550000
## 4  Non-Black / Non-Hispanic 6.830063
```

# Explanatory Variable: More groups

$$Sleep = Ethnicity + Error$$
$$Y_i = b_0 + b_1X_{1i} + b_2X_{2i} + b_3X_{3i} + e_i$$

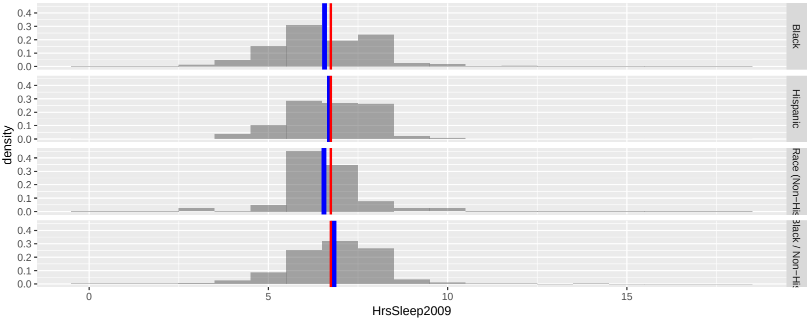
When there are more groups, we can use the mean from each group as the prediction, and we can compare this to the simple model

```
##                        Ethnicity     mean
## 1                          Black 6.566290
## 2                       Hispanic 6.702206
## 3 Mixed Race (Non-Hispanic) 6.550000
## 4  Non-Black / Non-Hispanic 6.830063
```



# Explanatory Variable: More groups

```
gf_dhistogram(~HrsSleep2009, data = NLSdata, binwidth = 1)%>%
gf_facet_grid(Ethnicity ~ .) %>%
gf_vline(xintercept=~mean,data=SleepEthnStats,color="blue",size=2)%>%
gf_vline(xintercept=~mean,data=SleepStats,color="red",size = 1)
```



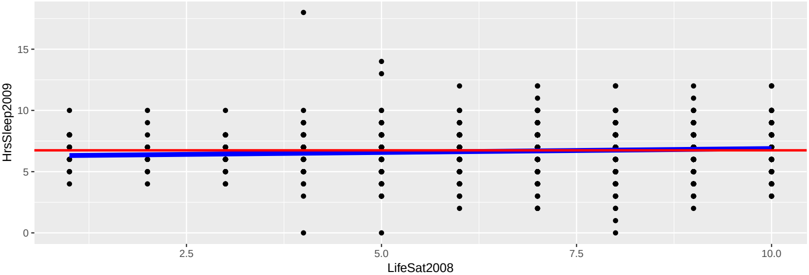
# Explanatory Variable: Continuous Explanatory Variable

$$Sleep = LifeSatisfaction + Error$$

$$Y_i = b_0 + b_1 X_i + e_i$$

When we have a continuous variable we create a line which relates the two variables to each other, and we can compare this to the simple model.

```
gf_point(HrsSleep2009~LifeSat2008, data = NLSdata)%>%
gf_lm(color = "blue", size = 2 ) %>%
  gf_hline(yintercept=~mean,data=SleepStats,color="red",size=1)
```



# Comparing Models

- To compare models we're going to need to quantify how "good" the model is as predicting the data. We'll use **sums of squares** to measure the distance between the observed points and the predicted points from each model! May the best model win!
- Adding more explanatory variables will always reduce the error, but we want a model that isn't too complicated. We'll use a metric called **degrees of freedom** to measure how complicated a model is.
- All of these models can be represented in **general linear model** notation

# Before Next Time

- Start Chapter 6, Due Monday 4/22 11:59pm

# Thursday April 18, 2019

While we're setting up

```r
NLSdata <- read.csv("http://bit.ly/NLSdata")
set.seed(123)
smallNLS <- sample(select(NLSdata, HrsSleep2009), 20)
```

## Learning Objectives

- Describe some characteristics of Sums of Squares
- Define variance and standard deviation
- Compare sums of squares with variance

## Residuals/Errors/Deviations

The residuals (errors) are the differences between the prediction (model) and the outcome.

$$Y_i - \bar{Y} = e_i$$

```
smallNLS$resid <- resid(empty.model)
smallNLS
```

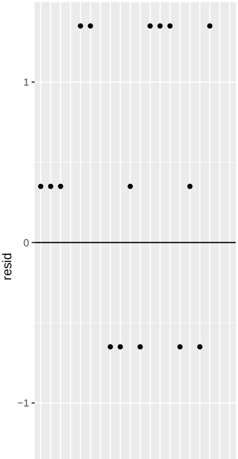
```
##      HrsSleep2009 orig.id resid
## 1215            7    1215  0.35
## 3330           10    3330  1.35
## 1727            7    1727 -1.65
## 3728            6    3728  1.35
## 3969            6    3969 -0.65
## 193             7     193 -0.65
## 2228            6    2228 -0.65
## 3764            8    3764 -0.65
## 2325            5    2325  0.35
## 1925            6    1925 -1.65
## 4033            9    4033 -1.65
## 1910            8    1910  1.35
## 2854            6    2854  1.35
## 2412            8    2412 -0.65
```

## Distribution of residuals

The distribution of residuals is identical to the distribution of the original data, but centered around zero!

Same spread/variance but a different center.

```
gf_point(resid~orig.id, data = smallNLS) %>%
  gf_hline(yintercept = 0)
```



# Sums of Squares

If we take the residuals and add them all together, we'll just get zero! Try it!

Remember the mean balances the positive and negative residuals based on magnitude.

```r
sum(smallNLS$resid)
```

```
## [1] 1.054712e-15
```

So to know how big the residuals are we calculate the **sum of the squared residuals**

$$\sum_{i=1}^{n} e_i^2 = SS_{residual}$$

$SS_{residual}$ is a measure of how close the observed data are to the prediction. When $SS_{residual}$ is big that means there is a lot of error. But when it's small, there is only a little bit of error.

# Sums of Squares

We can take each of the residuals and square them so the each person has a variable which is a **squared residual**

```
smallNLS$resid2 <- smallNLS$resid^2
smallNLS
```

```
##      HrsSleep2009 orig.id resid resid2
## 1215            7    1215  0.35 0.1225
## 3330           10    3330  1.35 1.8225
## 1727            7    1727 -1.65 2.7225
## 3728            6    3728  1.35 1.8225
## 3969            6    3969 -0.65 0.4225
## 193             7     193 -0.65 0.4225
## 2228            6    2228 -0.65 0.4225
## 3764            8    3764 -0.65 0.4225
## 2325            5    2325  0.35 0.1225
## 1925            6    1925 -1.65 2.7225
## 4033            9    4033 -1.65 2.7225
## 1910            8    1910  1.35 1.8225
## 2854            6    2854  1.35 1.8225
## 2412            8    2412 -0.65 0.4225
## 434             6     434 -1.65 2.7225
## 3788            6    3788  0.35 0.1225
## 1036            7    1036  0.35 0.1225
```

# Sums of Squares

Now we can add up all the squared residuals and this can give us a **total measure of error**.

Also all those calculations were a lot of work, so there is a function that will do this all for us `anova()`.

```
sum(smallNLS$resid2)
```

```
## [1] 24.55
```

```
anova(empty.model)
```

```
## Analysis of Variance Table
##
## Response: HrsSleep2009
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 19  24.55  1.2921
```

## The mean as a unique solution

The sample mean will always give the minimum possible sums of squares! Below is some code that calculates the sums of squares given a specific model. Try a new value other than the mean, and see what sums of squares you get.

```r
#this guarantees we all get the same dataset
set.seed(123)
#this creates the small NLS dataset
smallNLS <- sample(select(NLSdata, HrsSleep2009), 20)

#choose a value you want to try (not 6.95)
Value <- 7

#this calculates the errors/residuals
smallNLS$resid.v <- smallNLS$HrsSleep2009 - Value

#this calculates the squares errors/residuals
smallNLS$resid.v2 <- smallNLS$resid^2

#calculates the sum of squares
sum(smallNLS$resid.v2)
```

```
## [1] 29
```

Is there a value different from the mean that gets you sums of squares which are

# Other measures of total error

You may wonder why did we take the square of all the values and not the absolute value?

The **sum of absolute deviations** (SAD) is

$$\sum_{i=1}^{n} \mid e_i \mid = \sum_{i=1}^{n} \mid Y_i - \hat{Y} \mid = SAD$$

Our goal is to find the value of $\hat{Y}$ which minimizes SAD.

# Sum of Absolute Deviations

Well, let's see what happens when we do this. Can you find a value which minimizes the sum of absolute deviations?

```r
#choose a value you want to try
Value <- 6.9504

#this calculates the errors/residuals
smallNLS$resid.a <- smallNLS$HrsSleep2009 - Value

#this calculates the squares errors/residuals
smallNLS$resid.a2 <- smallNLS$resid.a^2

#calculates the sum of squares
sum(smallNLS$resid.a2)
```

```
## [1] 28.95
```

# One Minute Paper

Take one minute to write about what you learned from doing this exercise. What questions do you still have?

# Applet for playing with Sums of Squares

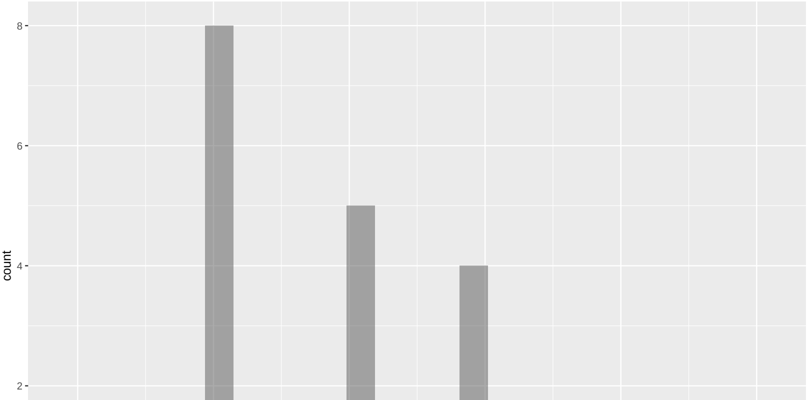
http://www.rossmanchance.com/applets/RegShuffle.htm

## Some issues with Sums of Squares

Sum of Squares is a sum, so the size is going to depend on how many things we're adding up!

This makes it hard to compare distributions, even when they're on the same scale!

```r
#anova(smallNLS$HrsSleep2009)
#anova(NLSdata$HrsSleep2009)
gf_histogram(~HrsSleep2009, data = smallNLS)
```



# Variance

If sums of squares is a sum of squared error, then variance is an average of squared error.

$$s^2 = \frac{SS}{n-1} = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y})^2}{n-1} = sample.variance$$

Variance is the **average squared error**, so it's no longer influenced by sample size!

# Variance

We can use variance to compare distributions with different sample sizes.

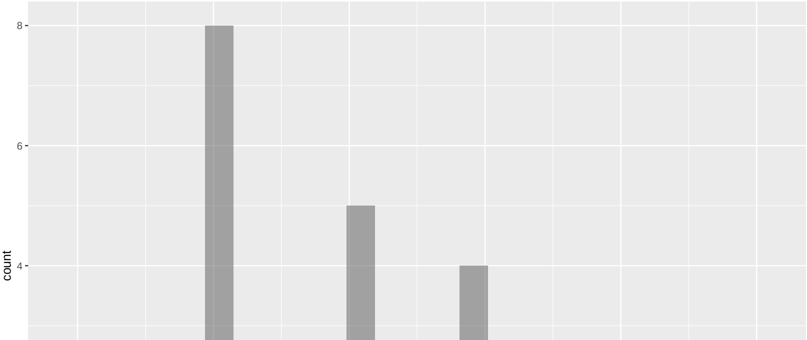
```
var(smallNLS$HrsSleep2009)
```

```
## [1] 1.523684
```

```
var(NLSdata$HrsSleep2009)
```

```
## [1] 1.703227
```

```
gf_histogram(~HrsSleep2009, data = smallNLS)
```



# Why divide by n-1 and not n?

Typically when we average something we have to divide by the number of things we're adding up!

$$s^2 = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y})^2}{n-1}$$

In sample variance it **looks like** we're adding up *n* things, but looks can be deceiving.

**Warning: a bit of scary math ahead! I don't expect that you be able to do this, just take a walk with me**

# Why divide by n-1 and not n?

Let's multiply out the numerator for variance:

$$(Y_i - \hat{Y})^2 = Y_i^2 - Y_i\hat{Y} - Y_i\hat{Y} + \hat{Y}^2$$