Psych 100A Spring 2019: Week 5 Slides

Amanda Montoya

April 30, 2019

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

Learning Outcomes Today

- Quantify Error in the Two Group Model (Sums of Squares)
- Compare the simple model to the two group model
- Evaluating Model using Proportional Reduced Error and Cohen's D

Prediction?

If having information about whether someone lives with their partner helps us know better how long someone sleeps, we would say that cohabitation predicts sleep!

But to know whether cohabitation predicts sleep we need to fit the model with all people together.

Explanatory Variable: Two groups

When there are two groups, we can use a general linear model to fit the data, which will predict the group mean for individuals in each group (i.e., $\bar{Y}_{cohab}$ or $\bar{Y}_{nocohab}$ depending on which group someone is in)

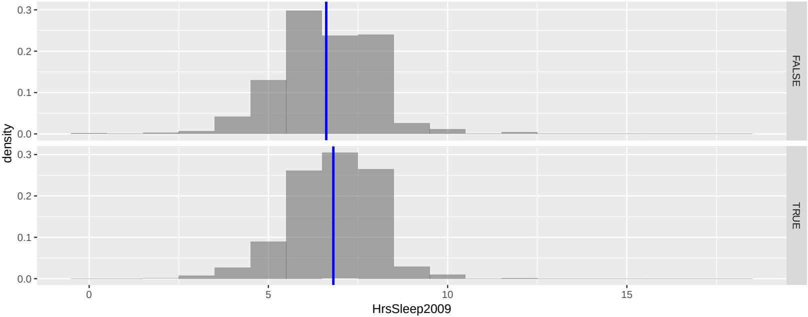
```
SleepCohabStats <- favstats(HrsSleep2009~Cohab2009, data = NLSdata)
SleepCohabStats
```

```
##   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

```
##   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
```



Fitting a linear model

We can fit a linear model using Cohab2009 as a predictor using R

```r
Cohab.model <- lm(HrsSleep2009~Cohab2009, data = NLSdata)
Cohab.model
```

```
##
## Call:
## lm(formula = HrsSleep2009 ~ Cohab2009, data = NLSdata)
##
## Coefficients:
##   (Intercept)  Cohab2009TRUE
##        6.6137         0.1982
```

These coefficients correspond to our general linear model notation

$$\hat{Y} = b_0 + b_1 X_i$$

b_0: Intercept b_1: Coefficient for X_i

$$\hat{Y} = 6.614 + 0.198 X_i$$

Predicting Y

X_i is an indicator which says whether or not someone lives with their partner.

Let's think about someone who does not live with their partner: $X_i = 0$

$$\hat{Y} = b_0 + b_1 X_i$$

$$\hat{Y} = 6.614 + 0.198 \times 0 = 6.614$$

The predicted Y for someone who does not live with their partner is 6.614.

The intercept will always be the predicted Y_i for individuals with a score of 0 on X_i.

Predicting Y

Let's think about someone who does live with their partner: $X_i = 1$

$$\hat{Y} = b_0 + b_1 X_i$$

$$\hat{Y} = 6.614 + 0.198 \times 1 = 6.614 + 0.198 = 6.812$$

The predicted Y for someone who does not live with their partner is 6.6812.

For the two group model, the coefficient for X_i (b_1) will always be the difference between the Group coded as 0 and the Group coded as 1.

In general, b_1 will always be the change in predicted Y_i with a one unit increase in X_i.

Calculating Predictions

```
NLSdata$CohabPred <- predict(Cohab.model)

head(select(NLSdata, HrsSleep2009, Cohab2009, CohabPred))
```

```
##   HrsSleep2009 Cohab2009 CohabPred
## 1            7     FALSE  6.613682
## 2            8     FALSE  6.613682
## 3            6      TRUE  6.811928
## 4            5      TRUE  6.811928
## 5            6     FALSE  6.613682
## 6            5      TRUE  6.811928
```

Calculating Residuals

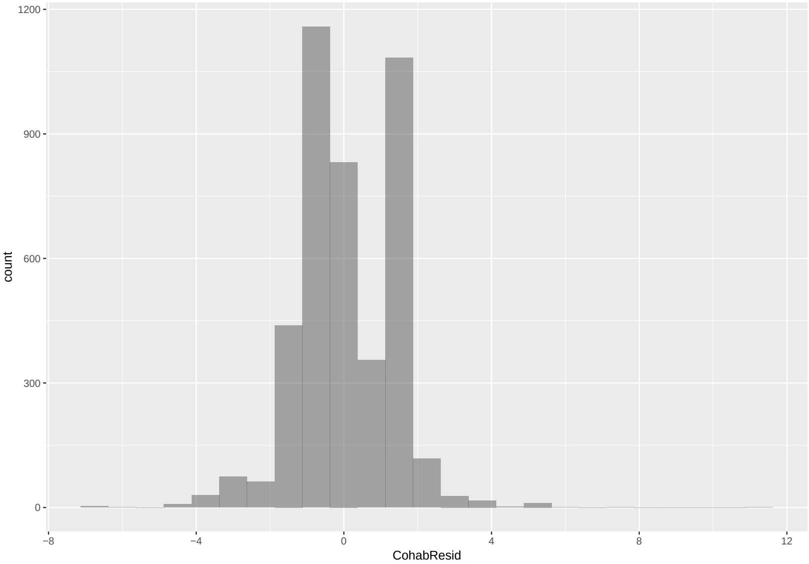
```r
NLSdata$CohabResid <- resid(Cohab.model)

head(select(NLSdata, HrsSleep2009, Cohab2009, CohabPred, CohabResid))
```

```
##   HrsSleep2009 Cohab2009 CohabPred CohabResid
## 1            7     FALSE  6.613682  0.3863179
## 2            8     FALSE  6.613682  1.3863179
## 3            6      TRUE  6.811928 -0.8119283
## 4            5      TRUE  6.811928 -1.8119283
## 5            6     FALSE  6.613682 -0.6136821
## 6            5      TRUE  6.811928 -1.8119283
```

Visualizing Residuals

```
gf_histogram(~CohabResid, data = NLSdata)
```



Explaining Error

The simple model does nothing to “explain” variance, but rather acts as a worst case scenario bench mark.

Do you think that a model which does a better job of predicting Hours of Sleep should have smaller or larger residuals than the simple model?

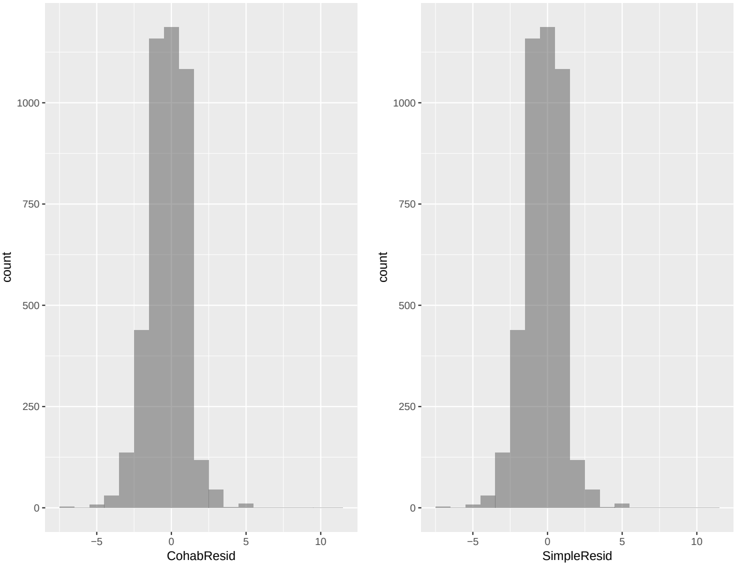
Explaining Error

Any variable which explains error should result in residuals which look much smaller on average than the simple model.

Let's consider a variety of ways to compare the simple model and the two-group model.

A visual comparison of residuals

```r
simplemodel <- lm(HrsSleep2009~NULL, data = NLSdata)
NLSdata$SimpleResid <- resid(simplemodel)
```



Are residuals smaller for Cohab Model?

If the Cohab model does a better job of predicting Hours of Sleep than the simple model, then for each individual we might expect their residual to be smaller for the Cohab model than the simple model.

```
NLSdata$Cohabsmall <- abs(NLSdata$CohabResid) < abs(NLSdata$SimpleResid
head(select(NLSdata, CohabResid, SimpleResid, Cohabsmall))
```

```
##   CohabResid SimpleResid Cohabsmall
## 1  0.3863179   0.2580492      FALSE
## 2  1.3863179   1.2580492      FALSE
## 3 -0.8119283  -0.7419508      FALSE
## 4 -1.8119283  -1.7419508      FALSE
## 5 -0.6136821  -0.7419508       TRUE
## 6 -1.8119283  -1.7419508      FALSE
```

```
tally(~Cohabsmall, data = NLSdata)
```

```
## Cohabsmall
##  TRUE FALSE
##  2390  1834
```

Are residuals smaller for Cohab Model?

```r
tally(~Cohabsmall, data = NLSdata, format = "proportion")
```

```
## Cohabsmall
##      TRUE     FALSE 
## 0.5658144 0.4341856
```

For a little over 50% of the people in the dataset, the residual from the Cohab model is **smaller** than the residual from the simple model.

Smaller residuals mean the predictions are more accurate

Quantifying Total Error

For the cohab model we can use the concept of **sum of squared error** to quantify total error in the model.

Sum of Squared Error can be broadly defined as:

$$SS_{residual} = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

In the simple model $\hat{Y}_i = \bar{Y}$.

What does $\hat{Y}_i$ equal in the two group model?

- The overall mean $\bar{Y}$
- The group mean from whichever group individual i is from
- Everyone will have a different $\hat{Y}_i$ because everyone is a little different

Quantifying Total Error

We can take the errors which are the deviations from the group means, square them and sum them to get a sum of squared error for the Cohab model.

Let's also compare it to the sum of squared residuals from the simple model

```
sum(NLSdata$CohabResid^2)
```

```
## [1] 7154.812
```

```
sum(NLSdata$SimpleResid^2)
```

```
## [1] 7192.726
```

It's smaller! What does that mean?

Quantifying Total Error

Another way to calculate sums of squares is with the anova() function

```
anova(Cohab.model)
```

```
## Analysis of Variance Table
##
## Response: HrsSleep2009
##             Df Sum Sq Mean Sq F value    Pr(>F)
## Cohab2009    1   37.9  37.914  22.373 2.318e-06 ***
## Residuals 4222 7154.8   1.695
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There's another sums of squares on here, for Cohab2009 what do you think that is?

Notice that $SS_{residual} + SS_{cohab} = SS_{total}$

Explained Variance

SS_{cohab} is a measure of how much variance has been explained by including Cohabitation in the model for Hours of Sleep.

We can quatify this by looking at the difference in predicted scores between the simple model and the cohab model.

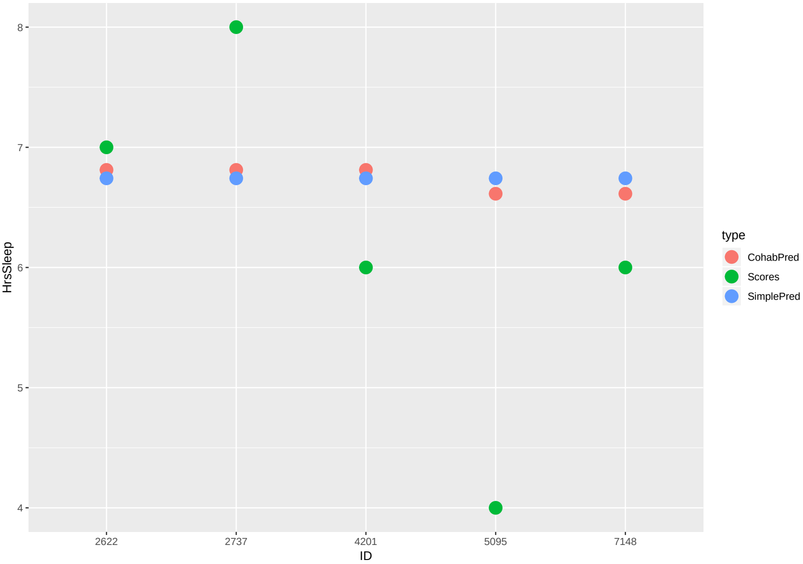
```r
NLSdata$SimplePred <- predict(simplemodel)

NLSdata$CohabvSimple <- NLSdata$CohabPred - NLSdata$SimplePred

sum(NLSdata$CohabvSimple^2)
```

```
## [1] 37.9143
```

Visualizing SS_{cohab}



Visualizing SS_{cohab}

A Sum of Squares is always the sum of squared things. Each of those things comes from a participant. Let's try to match those distances to sums of squares

Which element do you think corresponds to each of the following sums of squares?

- Sum of Squares Total / Sums of Squares for the simple model
- Sum of Squares Residuals (residuals from the Cohab Model)
- Sum of Squares Model (from the Cohab Model)

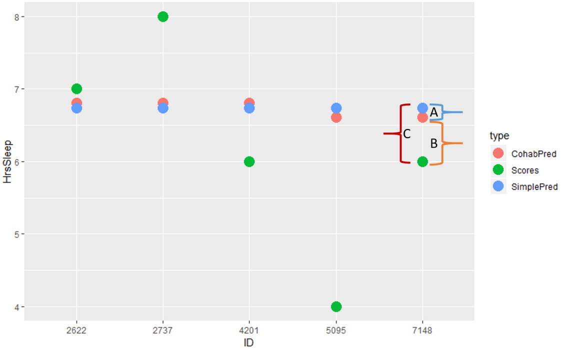


Figure 1: Elements of Sums of Squares

Proportional Reduction in Error

SS_{cohab} is the *amount* of error which was *explained* by Cohabitation.

As we've seen before, sums of squares is not an easy thing to interpret. What is a big sums of squares and what makes a small sums of squares?

One way to solve this problem is to consider the proportion of the total sums of squares that the model explains.

$$PRE = SS_{model}/SS_{total}$$

PRE is a measure of the proportional reduction in error.

Interpretting PRE

What would it mean if $PRE = SS_{model}/SS_{total} = 0$?

What would it mean if $PRE = SS_{model}/SS_{total} = 1$?

Calculating PRE

```
sum(NLSdata$CohabvSimple^2)/sum(NLSdata$SimpleResid^2)
```

```
## [1] 0.0052712
```

```
supernova(Cohab.model)
```

```
##  Analysis of Variance Table (Type III SS)
##  Model: HrsSleep2009 ~ Cohab2009
##
##                                SS   df     MS      F    PRE     p
##  ----- ----------------- -------- ---- ------ ------ ------ -----
##  Model (error reduced) |   37.914    1 37.914 22.373 0.0053 .0000
##  Error (from model)    | 7154.812 4222  1.695
##  ----- ----------------- -------- ---- ------ ------ ------ -----
##  Total (empty model)   | 7192.726 4223  1.703
```

Cohabitating only explains 0.5% of the variance in Hours of Sleep! That's not very much!

Cohen's d: Measuring Effect Size

Sometimes it's helpful to think about an difference in means, relative to the spread of the data.

Cohen's d is a measure of group differences relative to standard deviation (kind of like a Z-score)

$$d = \frac{\bar{Y}_1 - \bar{Y}_2}{s}$$

```r
cohensD( HrsSleep2009 ~ Cohab2009, data = NLSdata )
```

```
## [1] 0.1522877
```

The average effect size in psychology is $d = .2 - .3$ so this is a little smaller than effects we typically see.

Trying another Variable

There is one other two-group variable in the NLSdata: Sex. This data is pretty reductive in that everyone is classified as male or female.

Estimate the group means for Hours of Sleep for the two sexes in the data.

Plot HrsSleep2009 by Sex, does it looks like there's a difference by Sex?

Try estimating a two-group model using sex as the explanatory variable.

What does each part of this equation mean for this new model:
$Y_i = b_0 + b_1 X_i + e_i$

Compare the $SS_{residual}$ to SS_{total}, has the model reduced the error very much?

Compare SS_{model} with SS_{total} using PRE, what proportion of the variance in hours of sleep is explained by sex?

Calculate the Cohen's d for sex, is it bigger or smaller than the average effect in psychology?

Compare the PRE and d for the Cohab model and the Sex model, which one does a better job of explaining variance in Hours of Sleep?

When you're done write a one minute paper about the sex model, and how it compares to the cohab model

Next Time

- More than Two Groups
- Quantitative Predictors
- F-ratios

Thursday May 2nd, 2019

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

Learning Outcomes

- Estimate linear models with categorical variables (including more than two groups)
- Compare and contrast a model with the same predictor treated two groups vs. more groups
- Articulate the role of degrees of freedom in evaluating model complexity

A new predictor: Life Satisfaction

Today we'll consider the question: Does life satisfaction in 2008 predict hours of sleep in 2009?

Click in with your prediction: Individuals with higher life satisfaction in 2008 will have ________ hours of sleep 2009.

- Higher
- Lower
- No different

Life Satisfaction

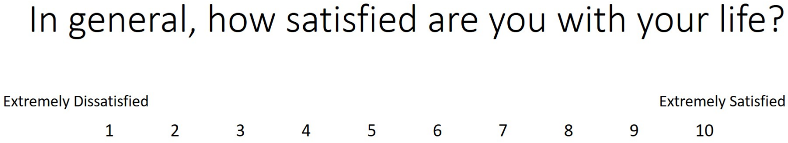
What's the question?

In general, how satisfied are you with your life?

How did they answer?

Response Scale from 1 - 10 (only whole numbers allowed)

1 = Extremely dissatisfied, 10 = Extremely satisfied



In general, how satisfied are you with your life?

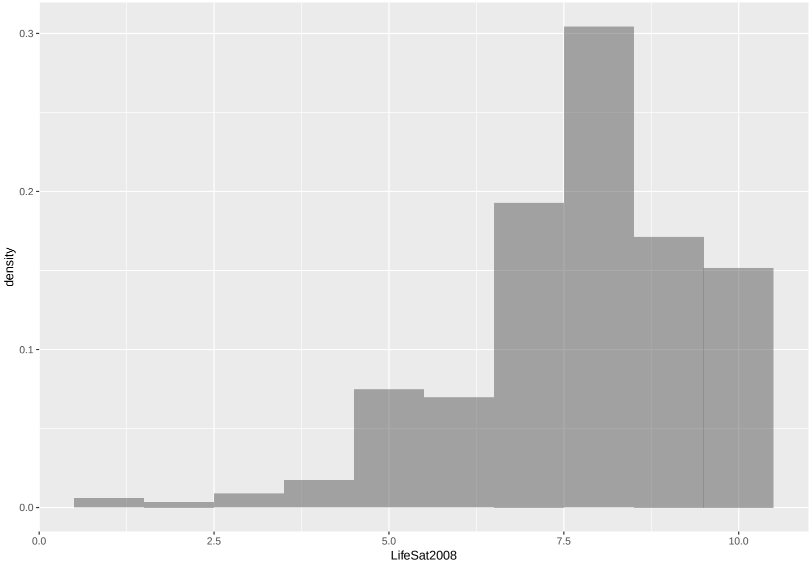
Extremely Dissatisfied Extremely Satisfied

1 2 3 4 5 6 7 8 9 10

Figure 2: Life Satisfaction

Visualizing Life Satisfaction

```
gf_dhistogram(~LifeSat2008, data = NLSdata, binwidth = 1)
```



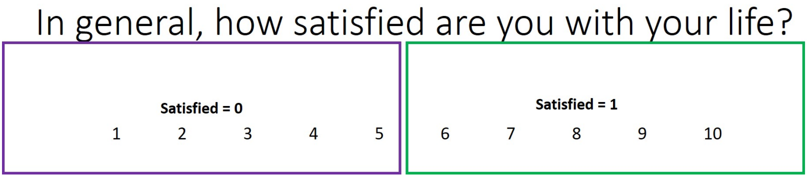
Summarizing Life Satisfaction

```
tally(~LifeSat2008, data = NLSdata, format = "proportion")
```

```
## LifeSat2008
##           1           2           3           4           5
## 0.005918561 0.003551136 0.008759470 0.017282197 0.074573864 0.069602
##           7           8           9          10
## 0.192945076 0.304214015 0.171401515 0.151751894
```

Are they satisfied?

Let's try dividing up the data into two groups, people who are and people who are not satisfied with their life (ignoring the degree of their satisfaction)



In general, how satisfied are you with your life?

Satisfied = 0

1 2 3 4 5

Satisfied = 1

6 7 8 9 10

Figure 3: Dichtomoizing Satisfaction

```
tally(~(LifeSat2008>5), data = NLSdata, format = "proportion")
```

```
## (LifeSat2008 > 5)
##      TRUE     FALSE
## 0.8899148 0.1100852
```

```
NLSdata$satisfied <- (NLSdata$LifeSat2008 > 5)
```

Using a two group test

We're familiar with using a two group model for HrsSleep2009, but this is a new predictor, "satisfied".

Let's say I code X_i as 0 if someone is dissatisfied and 1 if someone is satisfied.

We represent the two group model with the GLM equation: $Y_i = b_0 + b_1 X_i + e_i$

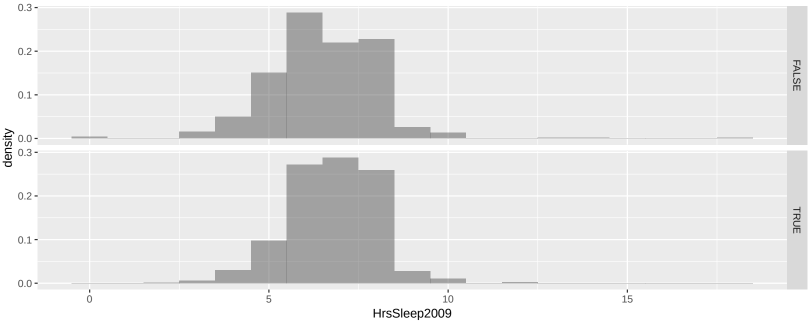
Answer the following questions on polleverywhere

- What is Y_i?
- What is b_0?
- What is b_1?
- What is X_i?
- What is e_i?

Visualizing Two Group Model

We can create separate plots of HrsSleep2009 based on which group individuals are in: Satisfied or Dissatisfied

```
gf_dhistogram(~HrsSleep2009, data = NLSdata, binwidth = 1) %>%
  gf_facet_grid(satisfied~.)
```



Summarizing two groups

We can calculate the means of HrsSleep2009 for each group. Notice how the code for `favstats` aligns with the code for `gf_dhistogram`

```
SatStats <- favstats(HrsSleep2009~satisfied, data = NLSdata)
SatStats
```

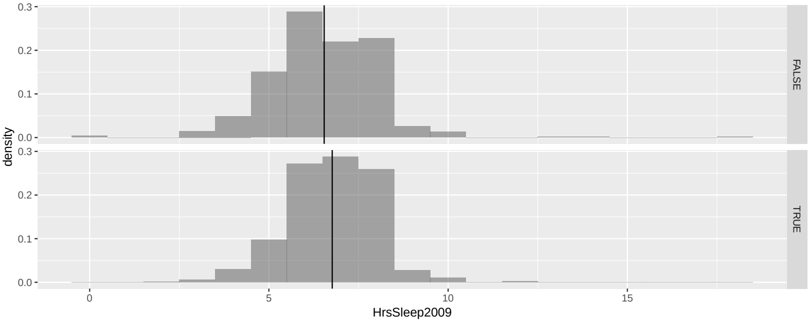
```
##   satisfied min Q1 median Q3 max     mean      sd    n missing
## 1     FALSE   0  6      6  8  18 6.541935 1.57255  465       0
## 2      TRUE   0  6      7  8  12 6.766693 1.26613 3759       0
```

In our sample, individuals who are satisfied with their life sleep about 6.76-6.54 = .22 hours more than those who are dissatisfied.

Adding means to visualization

We can add the means to the visualization

```
gf_dhistogram(~HrsSleep2009, data = NLSdata, binwidth = 1) %>%
  gf_facet_grid(satisfied~.)%>%
  gf_vline(xintercept = ~mean, data = SatStats)
```



Fitting the linear model

We can use the lm function to fit the linear model predicting Hours of Sleep 2009 with the two group life satisfation measure. Notice how the `lm` code is similar to the `favstats` code and the `gf_dhistogram` code.

```r
Satmodel <- lm(HrsSleep2009~satisfied, data = NLSdata)
Satmodel
```

```
##
## Call:
## lm(formula = HrsSleep2009 ~ satisfied, data = NLSdata)
##
## Coefficients:
##   (Intercept)  satisfiedTRUE
##        6.5419         0.2248
```

Going back to our general linear model: $Y_i = b_0 + b_1 X_i + e_i$

- What does the output "(Intercept) $=$ 6.5419" correspond to in the GLM equation?
- What does the output "satisfiedTRUE $=$ 0.2248" correspond to in the GLM equation?

Evaluating the model

```r
supernova(Satmodel)
```

```
## Analysis of Variance Table (Type III SS)
## Model: HrsSleep2009 ~ satisfied
##
##                                 SS   df     MS      F    PRE     p
## ----- ----------------- -------- ---- ------ ------ ------ -----
## Model (error reduced) |   20.904    1 20.904 12.306 0.0029 .0005
## Error (from model)    | 7171.822 4222  1.699
## ----- ----------------- -------- ---- ------ ------ ------ -----
## Total (empty model)   | 7192.726 4223  1.703
```

The satisfied variable explains about 21 sums of squares. This corresponds to only 0.3% of the variance in Hours of Sleep.

Remember that Cohabitation explained about 38 sums of squares and 0.5% of variance in Hours of Sleep.

Based on this, which do you think is a better predictor of hours of sleep: Life Satisfaction or Cohabitation? (raise of hands)

SS_{cohab} was greater than SS_{sat} and PRE_{cohab} was greater than PRE_{sat}, was this a concidence? Could one model have a higher PRE but lower SS?

More Groups

When we broke up life satisfaction we only made two groups.

This is troubling because we might imagine that someone who scores 1 and someone who scores 5 are pretty different from each other, but we treated them the same.

Similarly we might expect someone who scores 5 to be pretty similar to someone who scores 6, but they were in different groups. Whereas someone who scores 6 might be pretty different than a 10, even though we put them in the same group.

There are 10 possible scores on Life Satisfaction 2008, so why not just make 10 groups?

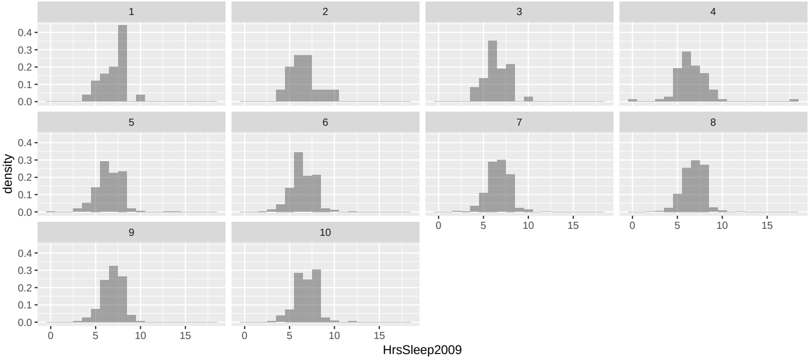
We can do this in `R` by using `factor(LifeSat2008)`, which tells R we want to treat Life Satisfaction as a categorical variable.

```r
NLSdata$fLifeSat <- factor(NLSdata$LifeSat2008)
```

Visualizing 10 Group Model

We can create separate plots of HrsSleep2009 based on which group individuals are in.

```
gf_dhistogram(~HrsSleep2009, data = NLSdata, binwidth = 1) %>%
  gf_facet_wrap(fLifeSat~.)
```



Summarizing two groups

We can calculate the means of HrsSleep2009 for each group. Notice how the code for `favstats` aligns with the code for `gf_dhistogram`

```
LifeStats <- favstats(HrsSleep2009~fLifeSat, data = NLSdata)
LifeStats
```

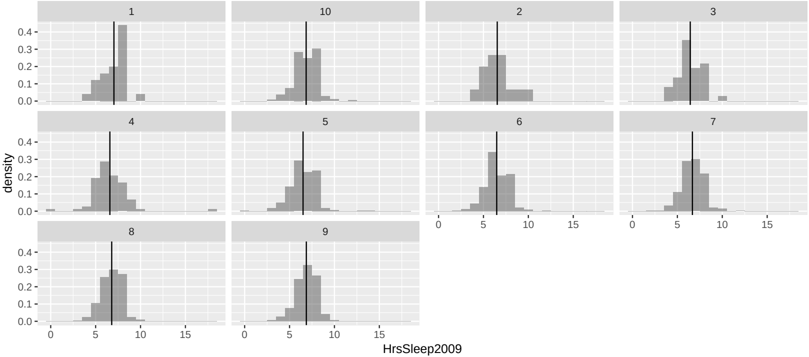
```
##    fLifeSat min  Q1 median Q3 max     mean       sd    n missing
## 1         1   4 6.0      7  8  10 7.040000 1.368698   25       0
## 2         2   4 5.5      6  7  10 6.533333 1.597617   15       0
## 3         3   4 6.0      6  7  10 6.432432 1.344547   37       0
## 4         4   0 6.0      6  8  18 6.589041 2.073883   73       0
## 5         5   0 6.0      6  8  14 6.504762 1.476723  315       0
## 6         6   2 6.0      6  7  12 6.476190 1.344098  294       0
## 7         7   2 6.0      7  8  12 6.673620 1.274963  815       0
## 8         8   0 6.0      7  8  12 6.791440 1.241201 1285       0
## 9         9   2 6.0      7  8  12 6.871547 1.214854  724       0
## 10       10   3 6.0      7  8  12 6.850234 1.300014  641       0
```

The means vary quite a lot, in fact the highest mean is 7.04 for group 1! Notice though that we have many more people in some groups compared to others. Think back to aggregating. Some of these means are going to be more accurate and some will be less accurate.

Adding means to visualization

We can add the means to the visualization

```
gf_dhistogram(~HrsSleep2009, data = NLSdata, binwidth = 1) %>%
  gf_facet_wrap(fLifeSat~.)%>%
  gf_vline(xintercept = ~mean, data = LifeStats)
```



Using a 10 group test!

We represent the two group model with the GLM equation: $Y_i = b_0 + b_1 X_i + e_i$

But for a 10 group test, we need 9 X variables, to represent the comparison between the *reference* group and each other group.

Group	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9
1	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0
7	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	0	1

GLM Equation

$$Y_i = b_0 + b_1 X_{1i} + b_2 X_{2i} + b_3 X_{3i} + b_4 X_{4i} + b_5 X_{5i} + b_6 X_{6i} + b_7 X_{7i} + b_8 X_{8i} + b_9 X_{9i} + e_i$$

b_0 average for Group 1

b_1 difference between Group 1 and Group 2

b_2 difference between Group 1 and Group 3

b_3 difference between Group 1 and Group 4

etc.

Predicted Scores

Using the GLM equation and the table for the X values we can derive predicted scores for each group!

Predicted score for someone in Group 3 would be:

$$\hat{Y}_i = b_0 + b_1X_{1i} + b_2X_{2i} + b_3X_{3i} + b_4X_{4i} + b_5X_{5i} + b_6X_{6i} + b_7X_{7i} + b_8X_{8i} + b_9X_{9i}$$

$$\hat{Y}_i = b_0 + b_1 * 0 + b_2 * 1 + b_3 * X_{3i} * 0 + b_4 * X_{4i} * 0 + b_5 * 0 + b_6 * 0 + b_7 * 0 + b_8 * 0 + b_9 * 0$$

$$\hat{Y}_i = b_0 + b_2$$

Here we can see that the predicted score for someone in Group 3 is the average from Group 1 plus the difference between Group 1 and Group 3.

Give it a try for Group 9!

Fitting the linear model

We can use the `lm` function to fit the linear model predicting Hours of Sleep 2009 with the two group life satisfation measure. Notice how the `lm` code is similar to the `favstats` code and the `gf_dhistogram` code.

```
Lifemodel <- lm(HrsSleep2009~fLifeSat, data = NLSdata)
Lifemodel
```

```
##
## Call:
## lm(formula = HrsSleep2009 ~ fLifeSat, data = NLSdata)
##
## Coefficients:
## (Intercept)    fLifeSat2    fLifeSat3    fLifeSat4    fLifeSat5
##      7.0400      -0.5067      -0.6076      -0.4510      -0.5352
##   fLifeSat6    fLifeSat7    fLifeSat8    fLifeSat9   fLifeSat10
##     -0.5638      -0.3664      -0.2486      -0.1685      -0.1898
```

$$\hat{Y}_i = b_0 + b_1X_{1i} + b_2X_{2i} + b_3X_{3i} + b_4X_{4i} + b_5X_{5i} + b_6X_{6i} + b_7X_{7i} + b_8X_{8i} + b_9X_{9i}$$

Use the answer from the previous slide to calculate the predicted score for someone in Group 9.

Evaluating the model

```
supernova(Lifemodel)
```

```
## Analysis of Variance Table (Type III SS)
## Model: HrsSleep2009 ~ fLifeSat
##
##                                SS   df    MS     F    PRE     p
## ----- ----------------- -------- ---- ----- ----- ------ -----
## Model (error reduced) |   73.239    9 8.138 4.817 0.0102 .0000
## Error (from model)    | 7119.487 4214 1.689
## ----- ----------------- -------- ---- ----- ----- ------ -----
## Total (empty model)   | 7192.726 4223 1.703
```

The 10 group version of Life Satisfaction explains 73 sums of squares! That's more than 3 times as much as the two group model (SS = 21).

Similarly the PRE is much higher, now we've explained 1% of the variance in Hours of Sleep (compared to 0.3%).

Cohabitation only explained 38 sums of squares and 0.5% of variance in Hours of Sleep.

Based on this, which do you think is a better predictor of hours of sleep: Life Satisfaction or Cohabitation? (raise of hands)

Model Complexity

It doesn't necessarily seem *fair* to compare the Cohab model to the 10 group Life Satisfaction model, since the 10 group model is much more complex.

Model Complexity: The number of parameter estimates in the model. The more parameters we estimate, the more flexible the model is, which gives the model the opportunity to explain more variance. It's much more "impressive" to explain lots of variance with fewer estimates.

Degrees of Freedom is the measure we use to indicate either how complex a model is (df_{model}) or how much flexibility is left over ($df_{residual}$).

Not all models are equally complex

We can't expect to only compare models which are equally complex. So how can we take into account complexity?

SS_{model} is our measure of variance explained.

We could compare sums of squares explained **per degree of freedom**: $SS_{model}/df_{model} = MS_{model}$ where MS stands for "Mean Square"

When we divide by df_{model} the scale gets even more confusing: *squared units/df*

We can also calculate a "Mean Square" for the error: $MS_{error} = SS_{error}/df_{error}$.

F-ratio

MS_{error} is the variance of the residuals left over in the model, taking into account how complex the model is.

We can calculate $F = \frac{SS_{model} / df\ model}{SS_{error} / df_{error}}$

If **in the population** the model doesn't explain any variability in the outcome, then

$$SS_{model} / df_{model} \approx SS_{error} / df\ error$$

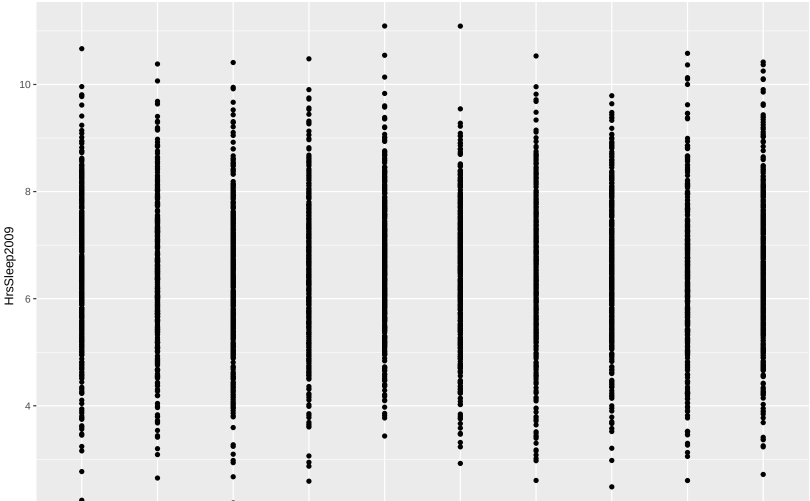
```r
supernova(Lifemodel)
```

```
## Analysis of Variance Table (Type III SS)
## Model: HrsSleep2009 ~ fLifeSat
##
##                                SS   df    MS     F    PRE     p
## ----- ----------------- -------- ---- ----- ----- ------ -----
## Model (error reduced) |   73.239    9 8.138 4.817 0.0102 .0000
## Error (from model)    | 7119.487 4214 1.689
## ----- ----------------- -------- ---- ----- ----- ------ -----
## Total (empty model)   | 7192.726 4223 1.703
```

A brief simulation

I'm going to create a world where Life Satisfaction is unrelated to hours of sleep:

```
Fakedata <- data.frame(HrsSleep2009 = rnorm(4224, mean = 6.5,sd=1.3),
            LifeSat = factor(sample(1:10, size = 4224, replace = TRUE))
gf_point(HrsSleep2009~LifeSat, data = Fakedata)
```



Analyzing the fake data

```
FakeModel <- lm(HrsSleep2009~LifeSat, data = Fakedata)
FakeModel
```

```
##
## Call:
## lm(formula = HrsSleep2009 ~ LifeSat, data = Fakedata)
##
## Coefficients:
## (Intercept)     LifeSat2     LifeSat3     LifeSat4     LifeSat5
##     6.60147     -0.18329     -0.24356     -0.15745      0.05189
##    LifeSat6     LifeSat7     LifeSat8     LifeSat9    LifeSat10
##    -0.11381     -0.14292     -0.03208     -0.21852     -0.07368
```

Analyzing the fake data

```
supernova(FakeModel)
```

```
##  Analysis of Variance Table (Type III SS)
##  Model: HrsSleep2009 ~ LifeSat
##
##                                SS   df    MS     F    PRE     p
##  ----- ----------------- -------- ---- ----- ----- ------ -----
##  Model (error reduced) |   35.696    9 3.966 2.264 0.0048 .0159
##  Error (from model)    | 7382.990 4214 1.752
##  ----- ----------------- -------- ---- ----- ----- ------ -----
##  Total (empty model)   | 7418.687 4223 1.757
```

The mean square model is pretty close to the mean square error ($F \approx 1$), and this is what we expect when there is no effect of the predictor on the outcome.

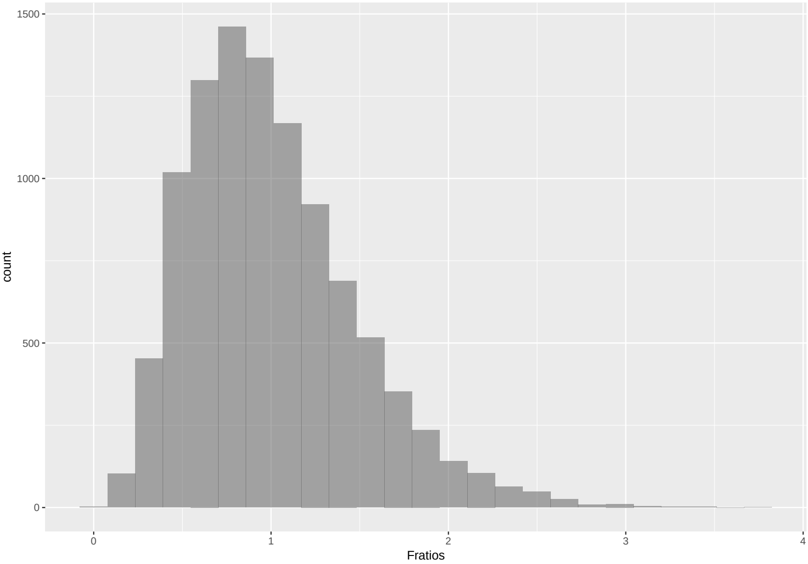
Imagine I did this many times and got a Distribution of F statistics

```
Fratios <- vector(length = 10000)

for (i in 1:10000){
    Fakedata <- data.frame(HrsSleep2009 = rnorm(4224, mean = 6.5, sd =
  FakeModel <- lm(HrsSleep2009~LifeSat, data = Fakedata)
  FakeModel
  Fratios[i] <- supernova(FakeModel)$tbl$F[1]
}
```

Distribution of F-statistics

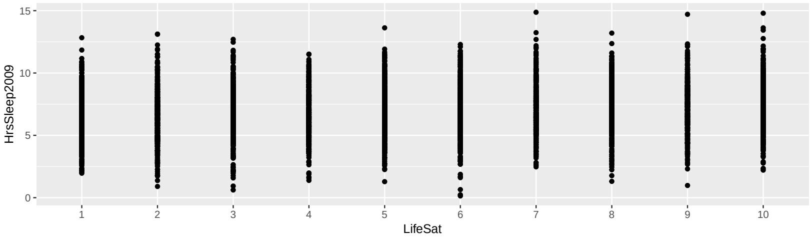
```
gf_histogram(~Fratios)
```



What if there is an effect of Life Satifaction on Sleep?

I'm going to create a world where Life Satisfaction is *related* to hours of sleep:

```
Fakedata <- data.frame(LifeSat = factor(sample(1:10, size = 4224, repla
Fakedata$HrsSleep2009 <- 6.5 + 0.1*as.numeric(Fakedata$LifeSat)+rnorm(4
stats <- favstats(HrsSleep2009~LifeSat, data = Fakedata)
gf_point(HrsSleep2009~LifeSat, data = Fakedata)
```



The average of the distribution depends on Life Satisfaction

Analyzing the fake data

```r
FakeModel <- lm(HrsSleep2009~LifeSat, data = Fakedata)
FakeModel
```

```
##
## Call:
## lm(formula = HrsSleep2009 ~ LifeSat, data = Fakedata)
##
## Coefficients:
## (Intercept)     LifeSat2     LifeSat3     LifeSat4     LifeSat5
##      6.4176       0.1226       0.3190       0.2723       0.6474
##    LifeSat6     LifeSat7     LifeSat8     LifeSat9    LifeSat10
##      0.4758       0.8030       0.9824       0.9432       1.1417
```

Analyzing the fake data

```r
supernova(FakeModel)
```

```
##  Analysis of Variance Table (Type III SS)
##  Model: HrsSleep2009 ~ LifeSat
##
##                                  SS   df     MS      F    PRE     p
##  ----- ----------------- --------- ---- ------ ------ ------ -----
##  Model (error reduced) |   583.930    9 64.881 16.404 0.0338 .0000
##  Error (from model)    | 16666.863 4214  3.955
##  ----- ----------------- --------- ---- ------ ------ ------ -----
##  Total (empty model)   | 17250.793 4223  4.085
```

We expect F to be around 1 when there is no effect, but when there is an effect it tends to be bigger than 1.

A simulation where there is a difference

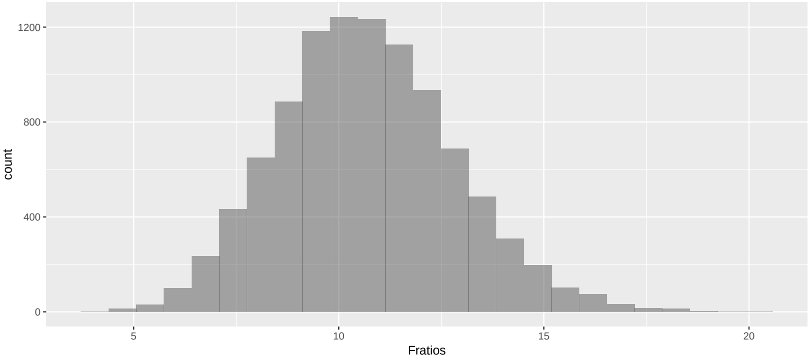
```
Fratios <- vector(length = 10000)

for (i in 1:10000){
  Fakedata <- data.frame(LifeSat2008 = factor(sample(1:10, size = 4224,
  Fakedata$HrsSleep2009 <-6.5 + 0.1*as.numeric(Fakedata$LifeSat)+
    rnorm(4224, 0, 2)
  FakeModel <- lm(HrsSleep2009~LifeSat2008, data = Fakedata)
  Fratios[i] <- supernova(FakeModel)$tbl$F[1]
}
```

Distribution of F-statistics

We expect that if there is an effect of the predictor, the F-ratio will be larger than 1.

```
gf_histogram(~Fratios)
```



F-ratio

An F-ratio is used to examine if the explanatory variable predicts more variance than we would expect due to change.

F-ratios are sensitive to how complex the model is (PRE is not sensitive to this)

$F = \frac{SS_{model} / df\ model}{SS_{error} / df_{error}}$

Life Satisfaction

I said we were going to evaluate the question: Does life satisfaction in 2008 predict hours of sleep in 2009?

```
Lifemodel
```

```
##
## Call:
## lm(formula = HrsSleep2009 ~ fLifeSat, data = NLSdata)
##
## Coefficients:
## (Intercept)    fLifeSat2    fLifeSat3    fLifeSat4    fLifeSat5
##      7.0400      -0.5067      -0.6076      -0.4510      -0.5352
##   fLifeSat6    fLifeSat7    fLifeSat8    fLifeSat9   fLifeSat10
##     -0.5638      -0.3664      -0.2486      -0.1685      -0.1898
```

This model doesn't necessarily answer our question, it's unclear if sleep increases or decreases with Life Satisfaction because we've allowed each group to behave independently.

Next Time

- Estimating model that treating Life Satisfaction as a continuous variable. Forces increasing, decreasing, or flat relationship
- Discussing pros and cons of treating variables as continuous vs. categorical