Psych 100A Spring 2019: Week 6 Slides

Amanda Montoya

May 7, 2019

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

Learning Outcomes Today

- Developing expectations for an F-ratio
- Estimating a model with a continuous predictor
- Comparing models of different complexity

Life Satisfaction

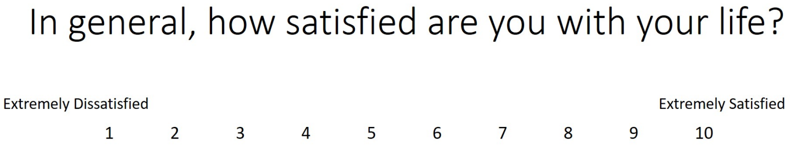
What's the question?

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

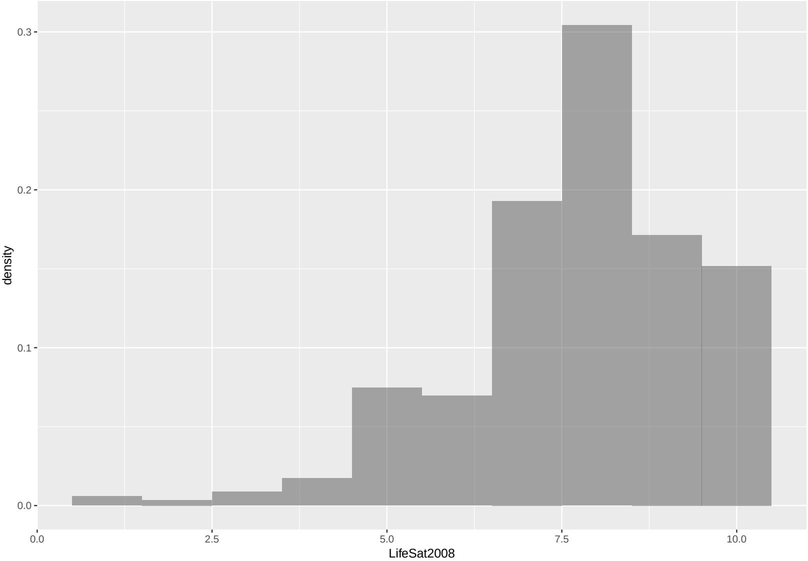
Extremely Satisfied

1 2 3 4 5 6 7 8 9 10

Figure 1: Life Satisfaction

Visualizing Life Satisfaction

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



Summarizing Life Satisfaction

```r
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
```

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?

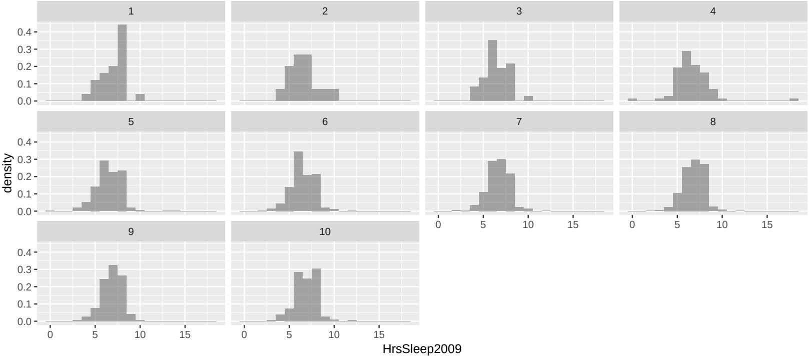
We can do this in `R` by using `factor(LifeSat2008)`, which tells R we want to treat Life Satisfaction as a categorical variable.

```
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 groups

We can calculate the means of HrsSleep2009 for each group.

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

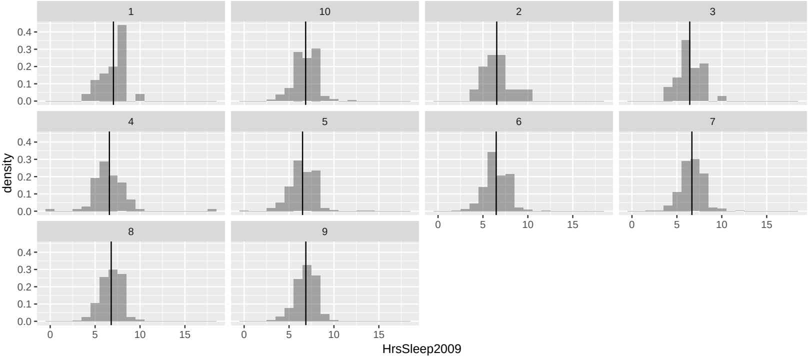
```
##    fLifeSat min  Q1 median Q3 max     mean       sd    n missing
## 1         1   4 6.0      7  8  10 7.040000 1.368698   25       0
## 2         2   4 5.5      6  7  10 6.533333 1.597617   15       0
## 3         3   4 6.0      6  7  10 6.432432 1.344547   37       0
## 4         4   0 6.0      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

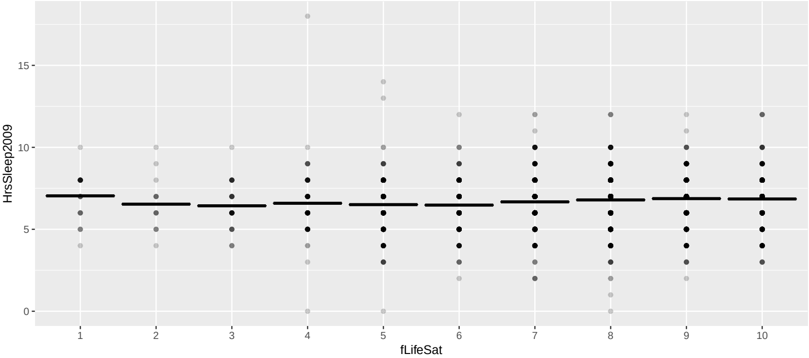
```
gf_dhistogram(~HrsSleep2009, data = NLSdata, binwidth = 1) %>%
  gf_facet_wrap(fLifeSat~.)%>%
  gf_vline(xintercept = ~mean, data = LifeStats)
```



Another visualization

We can add the means to the visualization

```
gf_point(HrsSleep2009~fLifeSat, data = NLSdata, alpha = 0.2) %>%
  gf_crossbar(mean+mean+mean~fLifeSat, 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:

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

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

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

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.

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
```

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

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.

On the surface it looks like Life Satisfaction is a better predictor than cohabitation!

Model Complexity

It doesn't necessarily seem *fair* to compare the Cohab model (two groups) 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?

Often times we want to know if adding complexity will make a model "much" better!

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

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

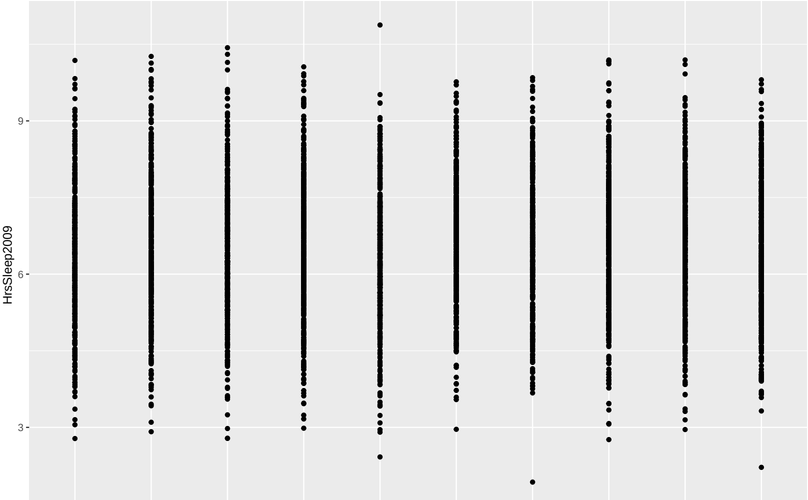
```r
supernova(Lifemodel)
```

```
## 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

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

```
##
## Call:
## lm(formula = HrsSleep2009 ~ LifeSat, data = Fakedata)
##
## Coefficients:
## (Intercept)     LifeSat2     LifeSat3     LifeSat4     LifeSat5
##     6.39415      0.16994      0.10260      0.17107      0.01964
##    LifeSat6     LifeSat7     LifeSat8     LifeSat9    LifeSat10
##     0.22602      0.13399      0.21365      0.13174      0.03527
```

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) |   24.299    9 2.700 1.564 0.0033 .1202
##  Error (from model)    | 7275.934 4214 1.727
##  ----- ----------------- -------- ---- ----- ----- ------ -----
##  Total (empty model)   | 7300.233 4223 1.729
```

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

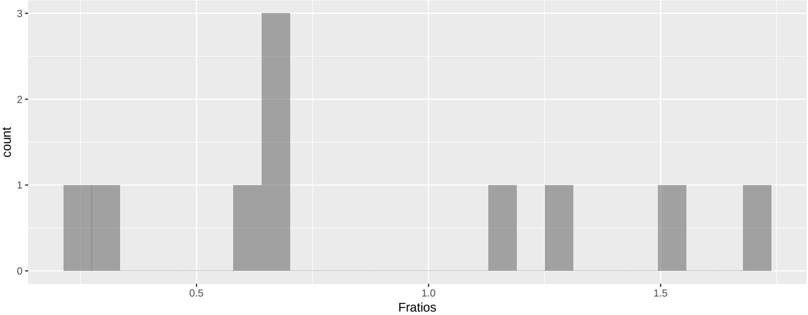
```r
Nsims <- 10
Fratios <- vector(length = Nsims)

for (i in 1:Nsims){
    Fakedata <- data.frame(HrsSleep2009=rnorm(4224,mean=6.5,sd=1.3),
               LifeSat=factor(sample(1:10,size=4224,replace=TRUE)))
  FakeModel <- lm(HrsSleep2009~LifeSat, data = Fakedata)
  FakeModel
  Fratios[i] <- supernova(FakeModel)$tbl$F[1]
}
```

Distribution of F-statistics

```r
favstats(Fratios)
```

```
##       min       Q1    median       Q3      max      mean        sd
##  0.239198 0.617558 0.6957783 1.265786 1.703214 0.8870359 0.5098927 1
##  missing
##        0
```

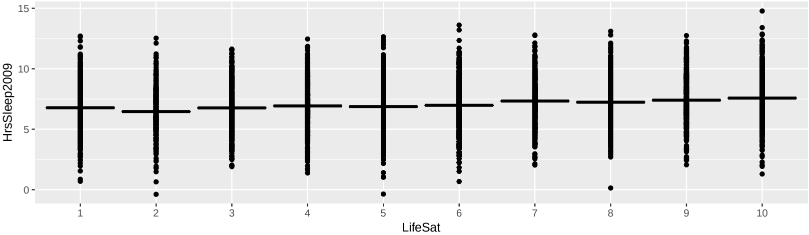
```r
gf_histogram(~Fratios)
```



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,replace=TRUE)))
Fakedata$HrsSleep2009<-6.5+0.1*as.numeric(Fakedata$LifeSat)+
                        rnorm(4224,0,2)
stats <- favstats(HrsSleep2009~LifeSat, data = Fakedata)
gf_point(HrsSleep2009~LifeSat, data = Fakedata)%>%
  gf_crossbar(mean+mean+mean~LifeSat, data = stats)
```



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
##     6.77562     -0.31647     -0.01215      0.15116      0.09581
##    LifeSat6     LifeSat7     LifeSat8     LifeSat9    LifeSat10
##     0.20057      0.55705      0.45628      0.62584      0.79764
```

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) |   444.930    9 49.437 12.426 0.0259 .0000
##  Error (from model)    | 16765.752 4214  3.979
##  ----- ----------------- --------- ---- ------ ------ ------ -----
##  Total (empty model)   | 17210.682 4223  4.075
```

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

```r
Fratios <- vector(length = Nsims)

for (i in 1:Nsims){
  Fakedata <- data.frame(
    LifeSat2008 = factor(sample(1:10, size = 4224, replace = TRUE)))
  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]
}
```

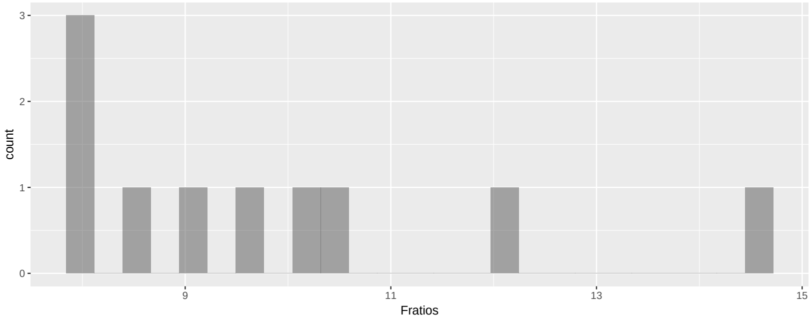
Distribution of F-statistics

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

```
favstats(Fratios)
```

```
##       min       Q1  median       Q3      max     mean       sd  n mi
##  8.023719 8.151754 9.30561 10.37089 14.62682 9.852588 2.136913 10
```

```
gf_histogram(~Fratios)
```



F-ratio

An F-ratio is used to examine if the explanatory variable predicts more variance **per degree of freedom** than we would expect due to chance.

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.

Treating Life Satisfaction as a continuous variable

We can use the general linear model to treat life satisfaction as a continuous variable

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

Y_i is person i's hours of sleep

b_0 is expected hours of sleep for someone with a score of 0 on life satisfaction (y-intercept)

b_1 is expected increase in hours of sleep with 1 unit increase in life satisfaction (slope)

X_i is life satisfaction

e_i error in estimating hours of sleep using life satisfaction model

Interpretting the Slope

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

Let's consider someone who scores 1 on life satisfaction.

Their predicted hours of sleep will be: $b_0 + b_1 * 1$

Someone who scores 2 on life satisfaction will have a predicted hours of sleep of $b_0 + b_1 * 2$

The difference between these two is $(b_0 + 2b_1) - (b_0 + b_1) = 2b_1 = b_1 = b_1$.

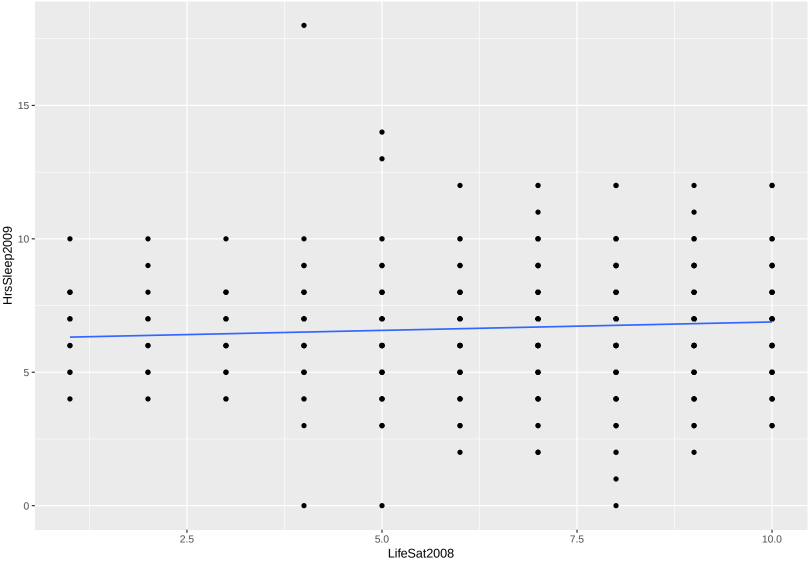
Try this with two people who are either 4 and 5, 7 and 8, or 9 and 10.

What do you get?

What would you expect the difference is between someone who scores 5 and someone who scores 7?

Visualizing the slope

```
gf_point(HrsSleep2009~LifeSat2008, data = NLSdata)%>%
  gf_lm()
```



Estimating the linear model

```
continuousmodel <- lm(HrsSleep2009~LifeSat2008, data = NLSdata)
continuousmodel
```

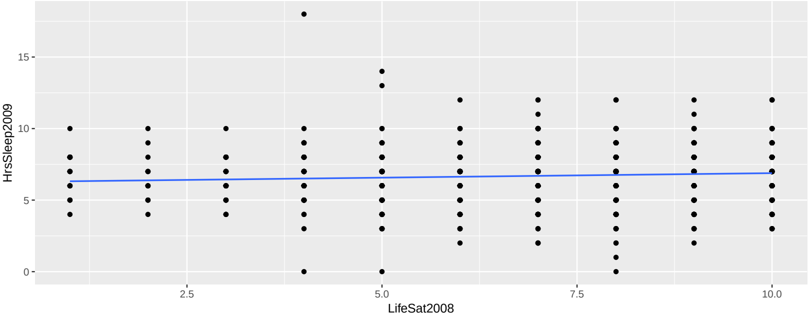
```
##
## Call:
## lm(formula = HrsSleep2009 ~ LifeSat2008, data = NLSdata)
##
## Coefficients:
## (Intercept)  LifeSat2008
##     6.25374      0.06305
```

Write an interpretation for the intercept and the slope.

Quantifying Error

The error e_i for this model is quantified by the vertical distance away from the observed value and the predicted value.

```
gf_point(HrsSleep2009~LifeSat2008, data = NLSdata)%>%
  gf_lm()
```



Saving Residuals & Predictions

```
NLSdata$contresid <- resid(continuousmodel)
NLSdata$contpred <- predict(continuousmodel)

head(select(NLSdata, HrsSleep2009, LifeSat2008, contpred, contresid))
```

```
##   HrsSleep2009 LifeSat2008 contpred  contresid
## 1            7           8 6.758131  0.2418692
## 2            8           9 6.821179  1.1788210
## 3            6          10 6.884227 -0.8842273
## 4            5           7 6.695082 -1.6950825
## 5            6           5 6.568986 -0.5689860
## 6            5           8 6.758131 -1.7581308
```

Residuals have same properties as before

```
sum(NLSdata$contresid)
```

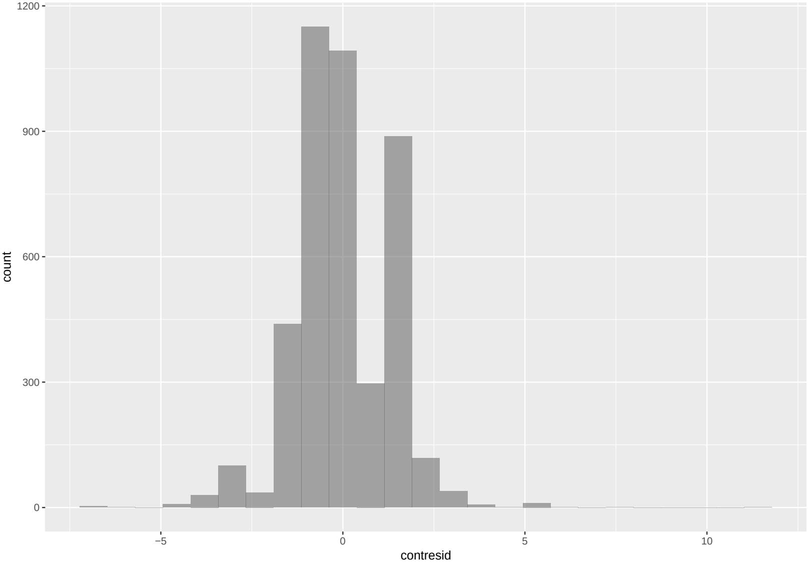
```
## [1] -7.322198e-13
```

Selection of b_0 and b_1 minimizes squared residuals

No other combination of b_0 and b_1 could give us smaller sums of squares

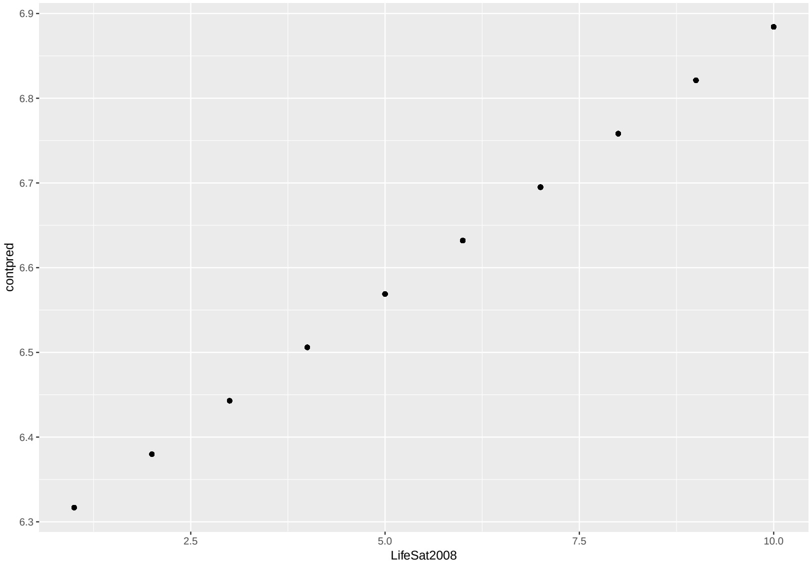
Plotting residuals

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



Plotting prediction

```
gf_point(contpred~LifeSat2008, data = NLSdata)
```



Analysis of Variance

```r
supernova(continuousmodel)
```

```
## Analysis of Variance Table (Type III SS)
## Model: HrsSleep2009 ~ LifeSat2008
##
##                               SS   df     MS      F    PRE     p
## ----- ----------------- -------- ---- ------ ------ ------ -----
## Model (error reduced) |   46.484    1 46.484 27.463 0.0065 .0000
## Error (from model)    | 7146.242 4222  1.693
## ----- ----------------- -------- ---- ------ ------ ------ -----
## Total (empty model)   | 7192.726 4223  1.703
```

$$SS_{total} = SS_{model} + SS_{error}$$

How can we interpret PRE?

Comparing continuous vs. 10 group model

```
supernova(continuousmodel)
```

```
##  Analysis of Variance Table (Type III SS)
##  Model: HrsSleep2009 ~ LifeSat2008
##
##                                SS   df     MS      F    PRE     p
##  ----- ----------------- -------- ---- ------ ------ ------ -----
##  Model (error reduced) |   46.484    1 46.484 27.463 0.0065 .0000
##  Error (from model)    | 7146.242 4222  1.693
##  ----- ----------------- -------- ---- ------ ------ ------ -----
##  Total (empty model)   | 7192.726 4223  1.703
```

```
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
```

Comparing continuous vs. 10 group model

10 Group model has higher PRE, but lower F statistic

This means that the continuous model explains more variance **per degree of freedom** than the 10 group model. *PRE* does not penalize based on how many degrees of freedom a model uses. So the more complex model explains more variance, but when we compare based on variance/df the continuous model does better.

Extra Credit Opportunity

Equivalent to 10 clicker questions. Attend a poster session or a talk. Include a picture of a description of a statistical model. Use GLM notation to describe the model they are using, what the outcome is, what the explanatory variable is. Describe the results of the study (1pg Double Spaced)

2019 UCLA PURC

MAY 10, 2019

9:15-10:15AM	Poster Presentations	Ackerman Union, 2nd Floor Bruin Reception Room
10:45AM-12:00PM	Paper Talks	Ackerman Union Room 2408 Room 3517
1:30-2:30PM	Poster Presentations	Ackerman Union, 2nd Floor Bruin Reception Room
2:45-3:45PM	Poster Presentations	Ackerman Union, 2nd Floor Bruin Reception Room
4:00-5:00PM	Closing Reception	Ackerman Union, 2nd Floor Bruin Reception Room

Next Time

-Bring questions for midterm

-We'll do some practice questions

-Review concepts from Chapters 1 - 8

Thursday 5/9/19

- Practice Questions for Midterm
- Introduction to Distributions of Estimates

Midterm on Friday

- Bring your student ID, a **charged laptop**, 1 page of notes **hand-written** single-sided 8.5''x 11.5" (with your name on it)
- We will give you the Rcheatsheet
- Tonight you will get an email with a survey link, this link is yours and unique to you
- There is a password to open the midterm, we will give you the password in section on Friday
- Some questions are just **statistical thinking**, some are **statistical doing** (`R`), and some are a combination.

Practice Questions

All the practice questions will be based on the following dataset. You can use `R` to calculate anything you need to answer the questions.

Load the data using

```r
data(Dimes)
```

1. There are 2 ___________s and 30 ________s in the data set.

- A. Factors; Cases
- B. Variables; Values
- C. Variables; Cases
- D. Cases; Variables

2. I am going to create a model that predicts the `mass` (measured in grams) of the dime based on the `year` it was made. But first I want to make a visualization of this relationship. Which types of plots could I use?

- A. Histogram
- B. Box Plot
- C. Bar Graph
- D. None of the Above

Practice Questions

3. Imagine I have a dime, but I don't know what year it was made. What type of model might I use to estimate a mass for this dime?

- A. Simple/Empty/Null Model
- B. Two group model
- C. Quantitative predictor model
- D. All of the above

4. What would the predicted mass of my dime be based on the above model?

- A. 2.258
- B. 0.014
- C. 29
- D. 2.031

Practice Questions

5. I created a variable called New using the r command

```
Dimes$New <- (Dimes$year > 1990)
```

What would the GLM equation be for me to estimate a two-group model for `mass` using this New variable

- A. $X_i = Y_i + b_0 + e_i$
- B. $Y_i = b_1 X_i + e_i$
- C. $Y_i = b_0 + b_1 X_i + e_i$
- D. $Y_i = b_0 + e_i$

Practice Questions

6. I estimated the two-group model and called it `twogroup`. Based on the output below does it seem like the mass of the dime depends on whether the dime is New (year $>$ 1990) or Old (year $=<$ 1990). Provide a reasoning.

```
supernova(twogroup)
```

```
##  Analysis of Variance Table (Type III SS)
##  Model: mass ~ New
##
##                             SS df    MS     F    PRE     p
##  ----- ----------------- ----- -- ----- ----- ------ -----
##  Model (error reduced) | 0.000  1 0.000 0.015 0.0005 .9035
##  Error (from model)    | 0.014 28 0.001
##  ----- ----------------- ----- -- ----- ----- ------ -----
##  Total (empty model)   | 0.014 29 0.000
```

Practice Questions

7. What command would I use to use `year` as a continuous variable to predict `mass`

- A. predict(year~mass, data = Dimes)
- B. lm(year~mass, data = Dimes)
- C. lm(mass~year, data = Dimes)
- D. predict(mass~year, data = Dimes)

```
##
## Call:
## lm(formula = mass ~ year, data = Dimes)
##
## Coefficients:
## (Intercept)         year
##   2.0309742    0.0001139
```

8. Based on the model, Samwell says that the predicted weight for a dime minted in 1970 is 2.03 grams, since that is the lowest year observed in the data. Is Samwell interpretting the intercept of the model correctly or incorrectly? Explain your answer. If Samwell is explaining incorrectly, provide a correct interpretation of the intercept.

Questions about the Midterm

Part III: Evaluating Models

Up to this point we have focused on (I) Exploring Variability and (II) Modeling Variation.

Our next task is to consider information we have about a model and *evaluate* it's appropriateness at the population level (Bottom Up thinking)

Topics in this Unit

Distributions of Estimates: Quantifying variability in estimates (e.g., sample mean) across random samples.

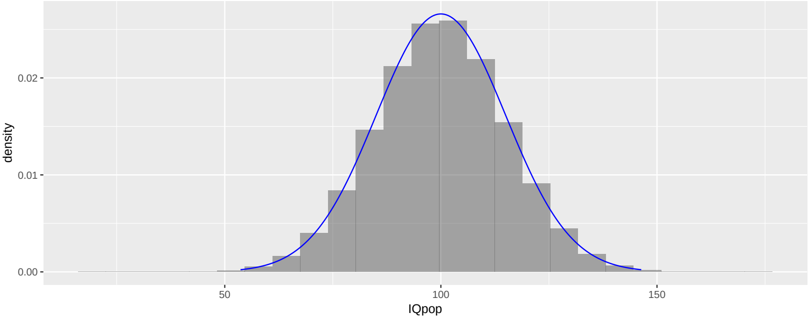
Confidence Intervals: Identify a range of population parameters which could have generated our data.

Model Comparison: Use hypothesis testing to evaluate whether our data seems likely for certain populations. Use inference to make conclusions about population.

A Brief Example

IQ (Intelligence Quotient) is a test designed to have a normal distribution with a population mean of 100 and standard deviation of 15.

```
IQpop <- rnorm(1000000, mean = 100, sd = 15)
gf_dhistogram(~IQpop) %>%
gf_dist( "norm" , color = "blue" , params =
list( 100, 15))
```



Sampling from the Population

If I took a random sample of 20 people from the population of individuals, and administered an IQ test and took the mean, we would expect that the mean would be close to 100, but perhaps not exactly 100.

```r
sample1 <- sample(IQpop, 20)
mean(sample1)
```

```
## [1] 103.4325
```

Similarly, if I took another random sample of 20 people, the mean would be close to 100, not exactly 100, and not exactly the same as from the first sample.

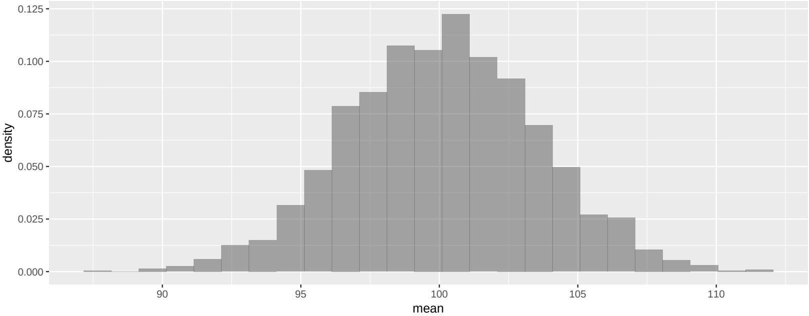
```r
sample2 <- sample(IQpop, 20)
mean(sample2)
```

```
## [1] 98.44692
```

A Distribution of Estimates

I can think about taking many many samples from the population, to consider the distribution of possible means that are possible when sampling from a population with mean 100 and standard deviation 15.

```
SDoM <- do(2000)*mean(sample(IQpop, 20))
gf_dhistogram(~mean, data = SDoM)
```

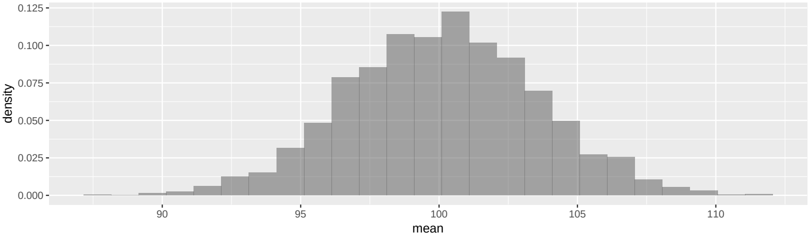


What do we learn from the distribution?

Does it seem likely or unlikely to get a mean of a sample greater than 110?

Does it seem likely or unlikely to get a mean of a sample between 95 and 100?

```
gf_dhistogram(~mean, data = SDoM)
```



Thinking backwards

What if I took a *random sample of 20 UCLA students* instead and I wanted to know if their IQ was higher than the population average?

```
mean(UCLAsample)
```

```
## [1] 116.6433
```

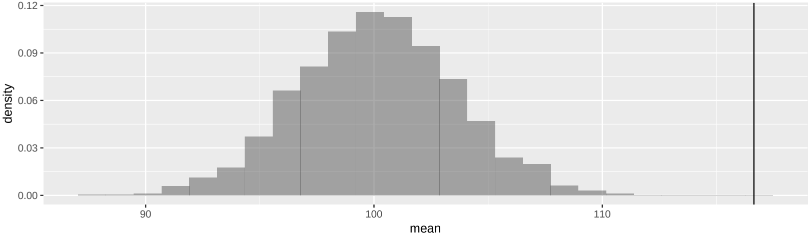
We can use what we know about samples that come from a "typical" IQ distribution (Mean = 100) to evaluate whether this sample seems particularly unusual.

Thinking backwards

Based on what we know about samples from a distribution with a mean of 100 and standard deviation of 15, does this seem like a likely sample from this distribution?

If not, then perhaps it comes from a different distribution, one with a higher mean?

```
gf_dhistogram(~mean, data = SDoM)%>%
  gf_vline(xintercept = mean(UCLAsample))
```



Confidence Intervals

The purpose of confidence intervals are to identify a range of **population means** from which are data are considered likely.

```r
mean(UCLAsample)
```

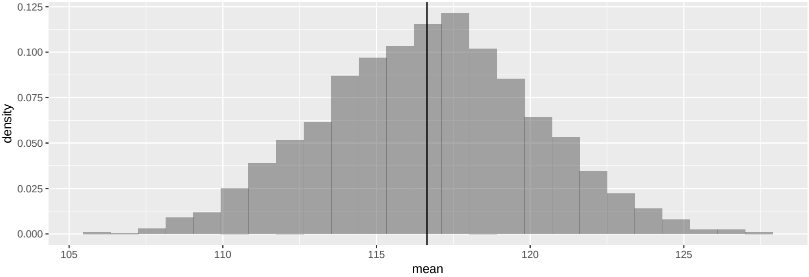
```
## [1] 116.6433
```

What means do you think could have generated this sample?

One guess is the mean we observed, if the population mean is 116.6432502 then our sample would be very probable.

Confidence Intervals

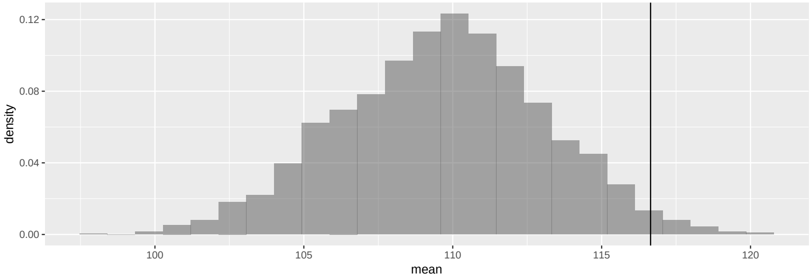
```r
#make a population with the observed mean from the data
IQpop2 <- rnorm(1000000, mean = mean(UCLAsample), sd = 15)
#create a set of samples from that population to consider sampling vari
SDoM2 <- do(2000)*mean(sample(IQpop2, 20))
#Visualize observed sample vs. sampling distribution
gf_dhistogram(~mean, data = SDoM2)%>%
  gf_vline(xintercept = mean(UCLAsample))
```



Defining the edges

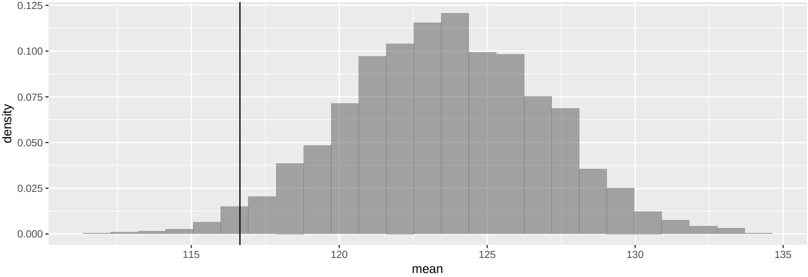
Let's consider some perhaps less perfect options

```
#make a population with mean below the observed mean from the data
IQpop3 <- rnorm(1000000, mean = mean(UCLAsample)-7, sd = 15)
#create a set of samples from that population to consider sampling vari
SDoM3 <- do(2000)*mean(sample(IQpop3, 20))
#Visualize observed sample vs. sampling distribution
gf_dhistogram(~mean, data = SDoM3)%>%
  gf_vline(xintercept = mean(UCLAsample))
```



Defining the edges

```
#make a population with mean above the observed mean from the data
IQpop3 <- rnorm(1000000, mean = mean(UCLAsample)+7, sd = 15)
#create a set of samples from that population to consider sampling vari
SDoM3 <- do(2000)*mean(sample(IQpop3, 20))
#Visualize observed sample vs. sampling distribution
gf_dhistogram(~mean, data = SDoM3)%>%
  gf_vline(xintercept = mean(UCLAsample))
```



Confidence Intervals

The goal of a confidence interval is to define an **upper and lower limit** of the population mean, for which the observed data seems likely (or not too unlikely).

110 - 124 seem like population means for which are data is likely, this means that out data is unlikely for a variety of other possible means.

This 100 is not included in the means for which our data is likely! This would indicate that we're confident the population mean for UCLA students is higher than the population average.

Model Comparison

In model comparison we use the F-ratio, to evaluate whether a proposed model performs better than the simple model.

Consider if the IQ scores from UCLA students are broken up by year in school. We can fit a four group model and compare that to the simple model.

The question we're exploring is: Do IQ scores differ across year in school, or are they about the same across all years?

Which model corresponds to which part of the question:

IQ scores differ across year in school: (Simple Model) or (Four group model)

IQ scores are the same across year in school: (Simple Model) or (Four group model)

Model Comparison

```
## Analysis of Variance Table (Type III SS)
## Model: IQ ~ year
##
##                                SS df      MS     F    PRE     p
## ----- ----------------- -------- -- ------- ----- ------ -----
## Model (error reduced) | 2290.478  3 763.493 2.454 0.3151 .1007
## Error (from model)    | 4977.665 16 311.104
## ----- ----------------- -------- -- ------- ----- ------ -----
## Total (empty model)   | 7268.143 19 382.534
```

The supernova results show that the F-ratio is close to 1. The column *p* tells us the probability of getting an F-ratio of what we observed or higher, if there is actually no difference between the four groups.

Based on this we would conclude there is not sufficient evidence that IQ differs by year, so we would prefer the simple model in describing IQ at UCLA.