Intro to linear models in R

Rose Hartman

Office of the Vice President for Research and Innovation
Center for Assessment Statistics & Evaluation

May 17th, 2016

Workshop Overview

1. Background and overview
   - What is the general linear model?
   - Tips for presenting model results
2. Run some models!
   - Prepping the data
   - Running the model
   - Summarizing the results
   - Plotting
3. More models

This workshop

- Focus on the R code rather than the stats — if you'd like to learn about the stats behind the general linear model more deeply, I can recommend several excellent classes and texts.

This workshop

- Focus on the R code rather than the stats — if you'd like to learn about the stats behind the general linear model more deeply, I can recommend several excellent classes and texts.
- Lots of practice. Learn R by using R!

This workshop

- Focus on the R code rather than the stats — if you'd like to learn about the stats behind the general linear model more deeply, I can recommend several excellent classes and texts.
- Lots of practice. Learn R by using R!
- We'll be using haven to read in an SPSS data file and ggplot2 for plotting

Keep an eye out for...

Keep an eye out for...

How would you do this?

Keep an eye out for...

How would you do this?

What will this do?

Keep an eye out for...

How would you do this?

What will this do?

Learn more: resources to check out

Keep an eye out for...

How would you do this?

What will this do?

Learn more: resources to check out

Key idea: the big ideas you need to hold on to

Keep an eye out for...

How would you do this?

What will this do?

Learn more: resources to check out

Key idea: the big ideas you need to hold on to

See also: other functions or packages that do a similar thing

If you don't already have ggplot2, and haven installed, do that now:

```
install.packages("ggplot2")
install.packages("haven")

library("ggplot2")
library("haven")
```

What counts as part of the general linear model?

Quite a lot! Most basic stats techniques are based on the general linear model.

- regression (simple and multiple)
- t-tests (all of the flavors)
- ANOVAs, ANCOVAs, etc.
- MANOVAs, multivariate multiple regression
- advanced techniques like HLM, SEM, etc. are extensions

What counts as part of the general linear model?

Quite a lot! Most basic stats techniques are based on the general linear model.

- regression (simple and multiple)
- t-tests (all of the flavors)
- ANOVAs, ANCOVAs, etc.
- MANOVAs, multivariate multiple regression
- advanced techniques like HLM, SEM, etc. are extensions

Key idea: Most basic stats rely on the same underlying math. Sometimes we want the results displayed differently depending on the design, but the calculations are the same.

What counts as part of the general linear model?

Quite a lot! Most basic stats techniques are based on the general linear model.

- regression (simple and multiple)
- t-tests (all of the flavors)
- ANOVAs, ANCOVAs, etc.
- MANOVAs, multivariate multiple regression
- advanced techniques like HLM, SEM, etc. are extensions

Key idea: Most basic stats rely on the same underlying math. Sometimes we want the results displayed differently depending on the design, but the calculations are the same.

What counts as part of the general linear model?

Quite a lot! Most basic stats techniques are based on the general linear model.

- regression (simple and multiple)
- t-tests (all of the flavors)
- ANOVAs, ANCOVAs, etc.
- MANOVAs, multivariate multiple regression
- advanced techniques like HLM, SEM, etc. are extensions

Key idea: Most basic stats rely on the same underlying math. Sometimes we want the results displayed differently depending on the design, but the calculations are the same.

What counts as part of the general linear model?

Quite a lot! Most basic stats techniques are based on the general linear model.

- regression (simple and multiple)
- t-tests (all of the flavors)
- ANOVAs, ANCOVAs, etc.
- MANOVAs, multivariate multiple regression
- advanced techniques like HLM, SEM, etc. are extensions

Key idea: Most basic stats rely on the same underlying math. Sometimes we want the results displayed differently depending on the design, but the calculations are the same.

What is the general linear model?

All the general linear model does is represent your data generating process as a single equation for a line:

$$\mathbf{Y} = \mathbf{X}*\mathbf{B} + \mathbf{error}$$

What is the general linear model?

All the general linear model does is represent your data generating process as a single equation for a line:

$$\mathbf{Y = X*B + error}$$

In most cases (when you have a single outcome variable), this can be written in its more familiar form:

$$Y = b0 + b1 * X1 + b2 * X2 + ... + error$$

Overview

Building dynamic documents

No matter what software you're using, it's a pain to copy-paste over numbers from a table in your output to a table in your manuscript.

More importantly, any human error in that copy-pasting can create a huge mess.

Wouldn't it be great if you could have R automatically send your output to word or pdf? Wish granted!

Building dynamic documents

No matter what software you're using, it's a pain to copy-paste over numbers from a table in your output to a table in your manuscript.

More importantly, any human error in that copy-pasting can create a huge mess.

Wouldn't it be great if you could have R automatically send your output to word or pdf? Wish granted!

Building dynamic documents

No matter what software you're using, it's a pain to copy-paste over numbers from a table in your output to a table in your manuscript.

More importantly, any human error in that copy-pasting can create a huge mess.

Wouldn't it be great if you could have R automatically send your output to word or pdf? Wish granted!

Building dynamic documents

```
install.packages("knitr")
library("knitr")
```

You can use knitr tools built right into R Studio to write r-markdown documents that include both code and text, and then "knit" them up into .docx, .pdf, or .html.
This makes it easy to get your R output into (close to) the table formatting you'll need in your document.

Building dynamic documents

```
install.packages("knitr")
library("knitr")
```

You can use knitr tools built right into R Studio to write r-markdown documents that include both code and text, and then "knit" them up into .docx, .pdf, or .html.
This makes it easy to get your R output into (close to) the table formatting you'll need in your document.

Building dynamic documents

```
install.packages("knitr")
library("knitr")
```

You can use knitr tools built right into R Studio to write r-markdown documents that include both code and text, and then "knit" them up into .docx, .pdf, or .html.
This makes it easy to get your R output into (close to) the table formatting you'll need in your document.

See also: If you want to use latex instead of markdown for the text part, use Sweave files (.rnw) instead of R-markdown (.rmd)

Building dynamic documents

Markdown is a super simple language for formatting text. It can't do much, but it handles most of what you'll need for a basic document, and it's really quick to learn.

Building dynamic documents

Markdown is a super simple language for formatting text. It can't do much, but it handles most of what you'll need for a basic document, and it's really quick to learn.

Learn more: `https://daringfireball.net/projects/markdown/basics`

Building dynamic documents

Markdown is a super simple language for formatting text. It can't do much, but it handles most of what you'll need for a basic document, and it's really quick to learn.

Learn more: `https://daringfireball.net/projects/markdown/basics`

Try opening a new .rmd file in R Studio now!

Building dynamic documents

We'll use pander to automatically format model summaries into lovely tables.

```
install.packages("pander")
```

Overview

Getting your data into R

Check your working directory:

```
getwd()
```

If you want R to find something on your computer, you have three options:

1. Put the file in R's working directory
2. Move R's working directory to where ever the file is saved using setwd()
3. Specify the file path when you tell R to look for the file

Getting your data into R

Check your working directory:

```
getwd()
```

If you want R to find something on your computer, you have three options:

1. Put the file in R's working directory
2. Move R's working directory to where ever the file is saved using setwd()
3. Specify the file path when you tell R to look for the file

Getting your data into R

Check your working directory:

```
getwd()
```

If you want R to find something on your computer, you have three options:

1. Put the file in R's working directory
2. Move R's working directory to where ever the file is saved using setwd()
3. Specify the file path when you tell R to look for the file

See also: You can also use the menu options in R Studio to change R's working directory.

Getting your data into R

I'll use option 3, specifying the file path for the file when I tell R to read it in. Find the file on your computer, get its location, and add that file path to your read_sav command:

```
atlas <- read_sav("data/ATLAS.sav")
```

Getting your data into R

I'll use option 3, specifying the file path for the file when I tell R to read it in. Find the file on your computer, get its location, and add

that file path to your read_sav command:

```
atlas <- read_sav("data/ATLAS.sav")
```

Getting your data into R

I'll use option 3, specifying the file path for the file when I tell R to read it in. Find the file on your computer, get its location, and add

that file path to your read_sav command:

```
atlas <- read_sav("data/ATLAS.sav")
```

Key idea: Remember, if your data file isn't in R's working directory you need to tell R where to look for it.

Check your data

These are data from an intervention intended to reduce steroid use in student athletes by targeting their strength training self-efficacy.

```
str(atlas)
head(atlas)
summary(atlas)
```

Check your data

These are data from an intervention intended to reduce steroid use in student athletes by targeting their strength training self-efficacy.

```
str(atlas)
head(atlas)
summary(atlas)
```

Key idea: Always check your data frame using str() and View() or head() before you run any tests.

Creating factors

```
atlas$use0 <- factor(atlas$use0,
                     levels=c(0, 1),
                     labels=c("no", "yes"))

atlas$intervention <- factor(atlas$intervention,
                             levels=c(0, 1),
                             labels=c("no", "yes"))

summary(atlas)
```

Overview

lm()

```
?lm
```

Writing a formula

Example question: Do students who got the intervention show improved strength-training self-efficacy, taking into account their pretest strength-training self-efficacy scores?

```
model1 <- lm( STSE1 ~ STSE0 + intervention, data=atlas,
              na.action=na.exclude)
```

Writing a formula

Example question: Do students who got the intervention show improved strength-training self-efficacy, taking into account their pretest strength-training self-efficacy scores?

```
model1 <- lm( STSE1 ~ STSE0 + intervention, data=atlas,
              na.action=na.exclude)
```

Test your knowledge

How would you do this? Specify a model formula to test whether students' strength-training self-efficacy at post test (STSE1) are related to their post-test self-esteem (SE1)

Test your knowledge

How would you do this? Specify a model formula to test whether students' strength-training self-efficacy at post test (STSE1) are related to their post-test self-esteem (SE1)

```
lm( STSE1 ~ SE1, data=atlas,
    na.action=na.exclude)
```

Test your knowledge

How would you do this? Specify a model formula to test whether students' strength-training self-efficacy at post test (STSE1) are different for the intervention group vs. control

Test your knowledge

How would you do this? Specify a model formula to test whether students' strength-training self-efficacy at post test (STSE1) are different for the intervention group vs. control

```
lm( STSE1 ~ intervention, data=atlas,
    na.action=na.exclude)
```

Test your knowledge

What will this do? What will this model test?

```
lm( STSE1 ~ intervention + SE1 + STSE0, data=atlas,
    na.action=na.exclude)
```

Test your knowledge

What will this do? What will this model test?

```
lm( STSE1 ~ intervention + SE1 + STSE0, data=atlas,
    na.action=na.exclude)
```

It will test whether students' strength-training self-efficacy at post test (STSE1) can be predicted from whether they got the intervention, their pretest self-esteem (SE1), controlling for their pretest strength-training self-efficacy (STSE0).

Test your knowledge

How would you do this? Expand the previous model to also test whether the effect of the intervention depends on the students' pretest self-esteem (SE1). In other words, add an interaction between intervention and pretest self-esteem.

Test your knowledge

How would you do this? Expand the previous model to also test whether the effect of the intervention depends on the students' pretest self-esteem (SE1). In other words, add an interaction between intervention and pretest self-esteem.

```
lm( STSE1 ~ intervention + SE1 + STSE0 + SE1:STSE0, data=atlas,
    na.action=na.exclude)
```

Test your knowledge

How would you do this? Expand the previous model to also test whether the effect of the intervention depends on the students' pretest self-esteem (SE1). In other words, add an interaction between intervention and pretest self-esteem.

```
lm( STSE1 ~ intervention + SE1 + STSE0 + SE1:STSE0, data=atlas,
    na.action=na.exclude)
```

```
lm( STSE1 ~ intervention + SE1*STSE0, data=atlas,
    na.action=na.exclude)
```

Test your knowledge

How would you do this? Specify a model formula to test whether students' strength-training self-efficacy at post test (STSE1) can be predicted from whether they got the intervention, their pretest strength-training self-efficacy (STSE0). Allow for the possibility that the effect of the intervention depends on pretest score.

Test your knowledge

How would you do this? Specify a model formula to test whether students' strength-training self-efficacy at post test (STSE1) can be predicted from whether they got the intervention, their pretest strength-training self-efficacy (STSE0). Allow for the possibility that the effect of the intervention depends on pretest score.

```
lm( STSE1 ~ intervention + STSE0 + intervention:STSE0, data=atlas,
    na.action=na.exclude)
```

Test your knowledge

How would you do this? Specify a model formula to test whether students' strength-training self-efficacy at post test (STSE1) can be predicted from whether they got the intervention, their pretest strength-training self-efficacy (STSE0). Allow for the possibility that the effect of the intervention depends on pretest score.

```
lm( STSE1 ~ intervention + STSE0 + intervention:STSE0, data=atlas,
    na.action=na.exclude)
```

```
lm( STSE1 ~ intervention*STSE0, data=atlas,
    na.action=na.exclude)
```

The model

Let's run that last one.

```
model1 <- lm( STSE1 ~ intervention*STSE0, data=atlas,
              na.action=na.exclude)
```

The model

Let's look at that object we just created:

```
model1

str(model1)

plot(model1)
```

Overview

Summarize the model

```
summary(model1)
```

Summarize the model

add this code to your .rmd file, and make sure you've set your knitr options to results='asis'

```
model.sum <- summary(model1)
pander(model.sum)
```

Summarize the model

add this code to your .rmd file, and make sure you've set your knitr options to results='asis'

```
model.sum <- summary(model1)
pander(model.sum)
```

See also: the stargazer package, which makes lovely model tables and works for knitting to pdf but not Word

Summarize the model

add this code to your .rmd file, and make sure you've set your knitr options to results='asis'

```
model.sum <- summary(model1)
pander(model.sum)
```

See also: the stargazer package, which makes lovely model tables and works for knitting to pdf but not Word

See also: knitr has a function for making tables, kable(), but it doesn't work for model summaries

Summarize the model

If you want your results ANOVA style (with the sums of squares and the F tests instead of regression coefficients and t-tests), use the Anova() function in the car package

```
car::Anova(model1, type=3) # type 3 sums of squares
```

Summarize the model

If you want your results ANOVA style (with the sums of squares and the F tests instead of regression coefficients and t-tests), use the Anova() function in the car package

```
car::Anova(model1, type=3) # type 3 sums of squares
```

Key idea: ANOVA style output and regression style output are just two different ways to communicate the same information

Overview

Plotting

The relationship between pretest and posttest scores

```
ggplot(atlas, aes(x=STSE0, y=STSE1)) +
  geom_point()
```

Plotting

The relationship between pretest and posttest scores

```
ggplot(atlas, aes(x=STSE0, y=STSE1)) +
  geom_point()
```

To reduce overplotting:

```
ggplot(atlas, aes(x=STSE0, y=STSE1)) +
  geom_point(alpha=.3)
```

Plotting

Add in whether or not they got the intervention

```
ggplot(atlas, aes(x=STSE0, y=STSE1, color=intervention)) +
  geom_point(alpha=.3)
```

Plotting

Add in whether or not they got the intervention

```
ggplot(atlas, aes(x=STSE0, y=STSE1, color=intervention)) +
  geom_point(alpha=.3)
```

another option

```
ggplot(atlas, aes(x=STSE0, y=STSE1)) +
  geom_point(alpha=.3) +
  facet_wrap(~ intervention)
```

Plotting

Want to change the way levels of a factor display?
You'll need to change the factor itself in the dataframe:

```
atlas$intervention <- factor(atlas$intervention,
                             levels=c("yes", "no"),
                             labels=c("intervention", "control"))
```

Plotting

Show the line of best fit

```
ggplot(atlas, aes(x=STSE0, y=STSE1,
                  color=intervention, fill=intervention)) +
  geom_point(alpha=.3) +
  geom_smooth(method="lm")
```

Plotting

Show the line of best fit

```
ggplot(atlas, aes(x=STSE0, y=STSE1,
                  color=intervention, fill=intervention)) +
  geom_point(alpha=.3) +
  geom_smooth(method="lm")
```

with faceting instead

```
ggplot(atlas, aes(x=STSE0, y=STSE1)) +
  geom_point(alpha=.3) +
  geom_smooth(method="lm") +
  facet_wrap(~ intervention)
```

Plotting

Show the line of best fit

```
ggplot(atlas, aes(x=STSE0, y=STSE1,
                  color=intervention, fill=intervention)) +
  geom_point(alpha=.3) +
  geom_smooth(method="lm")
```

with faceting instead

```
ggplot(atlas, aes(x=STSE0, y=STSE1)) +
  geom_point(alpha=.3) +
  geom_smooth(method="lm") +
  facet_wrap(~ intervention)
```

Key idea: geom_smooth(method="lm") will draw the line of best fit through the data it corresponds to. Adding a factor to color= within aes() splits points and lines into subsets

Plotting

```
ggplot(atlas, aes(x=STSE0, y=STSE1,
                  color=intervention, fill=intervention)) +
  geom_point(alpha=.3) +
  geom_smooth(method="lm") +
  labs(x="pretest strength training self-efficacy",
       y="posttest strength training self-efficacy")
```

Plotting

```
ggplot(atlas, aes(x=STSE0, y=STSE1,
                  color=intervention, fill=intervention)) +
  geom_point(alpha=.3) +
  geom_smooth(method="lm") +
  labs(x="pretest strength training self-efficacy",
       y="posttest strength training self-efficacy")
```

Learn more: For a tutorial on controlling colors in ggplot, see an old R Club post of mine: `http://blogs.uoregon.edu/rclub/2015/02/17/picking-pretty-plot-palates/`

Plotting the line from your model, exactly

Because geom_smooth() runs its own calculation to draw the line of best fit, sometimes you may want to specify the line yourself, so you can make sure it represents exactly what's in your model.

```
atlas$pred <- predict(model1, atlas)
```

Plotting the line from your model, exactly

Because geom_smooth() runs its own calculation to draw the line of best fit, sometimes you may want to specify the line yourself, so you can make sure it represents exactly what's in your model.

```
atlas$pred <- predict(model1, atlas)
```

```
ggplot(atlas, aes(x=STSE0, y=STSE1,
                  color=intervention, fill=intervention)) +
  geom_point(alpha=.3) +
  geom_line(aes(y=pred, x=STSE0, color=intervention)) +
  labs(x="pretest strength training self-efficacy",
       y="posttest strength training self-efficacy")
```

Plotting the line from your model, exactly

Because geom_smooth() runs its own calculation to draw the line of best fit, sometimes you may want to specify the line yourself, so you can make sure it represents exactly what's in your model.

```
atlas$pred <- predict(model1, atlas)
```

```
ggplot(atlas, aes(x=STSE0, y=STSE1,
                  color=intervention, fill=intervention)) +
  geom_point(alpha=.3) +
  geom_line(aes(y=pred, x=STSE0, color=intervention)) +
  labs(x="pretest strength training self-efficacy",
       y="posttest strength training self-efficacy")
```

See also: you can also add lines manually with geom_abline(), where you specify the slope and intercept

Additional resources for learning ggplot2

- Read Jenny Bryan's ggplot tutorials — tons of great examples and code! Click on the files that have the file extension .md (those will be the easiest to read)
  `https://github.com/jennybc/ggplot2-tutorial`
- All of the geoms, with pictures
  `http://docs.ggplot2.org/current/`
- For more in-depth material on ggplot2, see the resources at
  `http://ggplot2.org/`

Getting special output for other types of models

```
?t.test
```

Getting special output for other types of models

```
?t.test
```

```
install.packages("car")
library("car")
?Anova
```

Note: I recommend car::Anova() over stats::aov() because it allows you to use type 3 sums of squares, which is probably what you want and what your readers and colleagues are expecting (it's what SPSS and SAS use).

Getting special output for other types of models

Do students' pretest scores differ in the intervention vs. control groups?

```
t.test(STSE0 ~ intervention, data=atlas,
       var.equal=TRUE)
```

Getting special output for other types of models

Do students' pretest scores differ in the intervention vs. control groups?

```
t.test(STSE0 ~ intervention, data=atlas,
       var.equal=TRUE)
```

Welch approximation (does not assume equal variance)

```
t.test(STSE0 ~ intervention, data=atlas,
       var.equal=FALSE)
```

Getting special output for other types of models

Do students' pretest scores differ in the intervention vs. control groups?

```
t.test(STSE0 ~ intervention, data=atlas,
       var.equal=TRUE)
```

Welch approximation (does not assume equal variance)

```
t.test(STSE0 ~ intervention, data=atlas,
       var.equal=FALSE)
```

How many pariticipants got intervention vs. control?

```
summary(atlas$intervention)
```

Test your knowledge

How would you do this? test whether students' pretest self-esteem differ in the intervention vs. control groups

Test your knowledge

How would you do this? test whether students' pretest self-esteem differ in the intervention vs. control groups

```
t.test(SE0 ~ intervention, data=atlas,
       var.equal=TRUE)
```

Test your knowledge

How would you do this? Test whether students' pretest strength-training self-efficacy scores (STSE0) differ from zero

Hint: Take a look at the help documentation for t.test()

Test your knowledge

How would you do this? Test whether students' pretest strength-training self-efficacy scores (STSE0) differ from zero

Hint: Take a look at the help documentation for t.test()
One-sample t-test

```
t.test(x=atlas$STSE0) # can't use formula and data argument for one-sam
```

Test your knowledge

How would you do this? Test whether there was significant change in students strength-training self-efficacy scores

Test your knowledge

How would you do this? Test whether there was significant change in students strength-training self-efficacy scores

Paired-sample t-test

```
t.test(x=atlas$STSE0, y=atlas$STSE1,
       paired=TRUE)
```

Test your knowledge

How would you do this? Use an ANOVA to test whether students pretest scores differed based on whether they were assigned to the intervention

Test your knowledge

How would you do this? Use an ANOVA to test whether students pretest scores differed based on whether they were assigned to the intervention

```
model <- lm(STSE0 ~ intervention, data=atlas)

car::Anova(model, type=3) # type 3 sums of squares
```