4 Statistical Frameworks Study Guide (and other notes?)

EA30 2023 Students and Marc

2023-03-27

# Using R to Create R Study Guide

## Generic Template to Create Word Documents

I created this document to give you a “head-start” in creating the study guide with a format that everyone can use. By using Rstudio, you will have the capacity to create an integrated approach that merges R commands and outputs and text.

I have defined the output as a word document to start with – but we can create a pdf or html output if we decide that is better. But by putting into Word, we can use Word to spot typos easier!

There are ways to share these documents via github.com – but frankly, it works great if the class is smaller, e.g. 10-12. So, we’ll use an assortment of sakai upload tools – not ideal, but I’ll make it work!

## Developing Content

Each week, I intend to dedicate 40-60 minutes to develop content.

We’ll start with creating an example. But as we do, we want to be answering these following questions:

* What questions can be answered with the method?
* Are their limitations to the method?
* What are the assumptions?
* How can the assumptions be tested?

The example should include data to be analyzed – either imported or created in R, then the R codes to analyze the results and then a discussion about how to interpret the data.

### Defining Variable Types

#### Dependent vs. Independent Data (add text here…)

#### Continuous vs. Count vs. Categorical

NOTE: Spatial Data and Time Series data have special concerns… “auto-correlation”

## Class Assignments

The study guide development will go for five weeks, so everyone can have a change to look at each of the for frameworks and then one session to integrate everything into one document. See sakai to see your group assignment.

# Getting Started with R

## R and Rstudio

Step-by-step Guide

Getting started in R

If you havenâ€™t used R before, start by downloading R and R Studio.

## Editing the Guide

Once you have both of these programs downloaded, open R Studio and click on File > New File > R Script.

##Now you can copy and paste the code from the rest of this example into your script. To run the code, highlight the lines you want to run and click on the Run button on the top right of the text editor (or press ctrl + enter on the keyboard).

##Install and load the packages# ##First, install the packages you will need for the analysis (this only needs to be done once): ##Then load these packages into your R environment (do this every time you restart the R program): ##Step 1: Load the data into R

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars) # extract summary statistics from built-in cars dataset.

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

I added the comment line to help the reader remember what’s happening. These comments are useful to help everyone see the intentions of the author.

### Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot. But as a guide, we might want to show the code – which can be done with echo = TRUE, which is the default.

# Four Frameworks

## Linear Regression

### Linnear Regression Examples

Example #1

Example #2

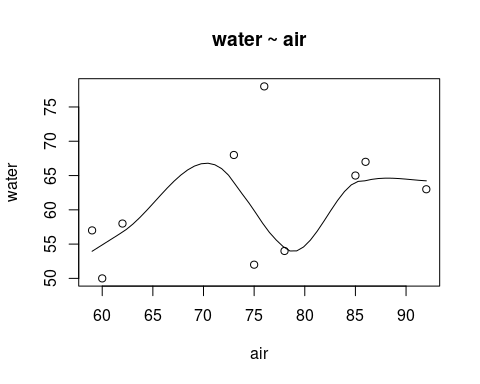
Is water and air temperature correlated?

water = c(50, 52, 65, 67, 68, 78, 54, 63, 57, 58)  
air = c(60, 75, 85, 86, 73, 76, 78, 92, 59,62)  
lm(formula = water ~ air)

##   
## Call:  
## lm(formula = water ~ air)  
##   
## Coefficients:  
## (Intercept) air   
## 36.3333 0.3333

making a graph…

scatter.smooth(x=air, y=water, main="water ~ air") # scatterplot



#linearMod <- lm(water ~ air, data=temperature) # build linear regression model on full data  
#print(linearMod)

Example #3: SRP and HABs

Following temperatures and SRP a nd dwith

Define Data:

SRP <- c(.02, .03, 0.4, 0.04, .05, 0.033, 1.4)  
cyano<- c(22, 34, 23, 22, 23, 2, 122)

Create Linear Model using lm() function

The linear model coerces the data into a new object using a least squares method to calculate the slope and intercept (and other statitical date.)

HAB.lm = lm(cyano ~ SRP)  
HAB.lm

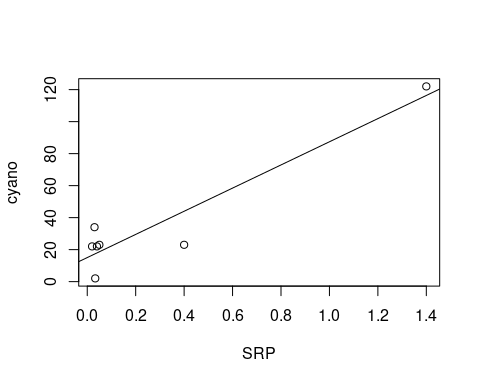
##   
## Call:  
## lm(formula = cyano ~ SRP)  
##   
## Coefficients:  
## (Intercept) SRP   
## 15.03 72.37

Let’s look at the model results with summary()

summary(HAB.lm)

##   
## Call:  
## lm(formula = cyano ~ SRP)  
##   
## Residuals:  
## 1 2 3 4 5 6 7   
## 5.523 16.799 -20.979 4.076 4.352 -15.418 5.647   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 15.029 6.398 2.349 0.06563 .   
## SRP 72.374 11.608 6.235 0.00155 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 14.55 on 5 degrees of freedom  
## Multiple R-squared: 0.886, Adjusted R-squared: 0.8632   
## F-statistic: 38.87 on 1 and 5 DF, p-value: 0.001554

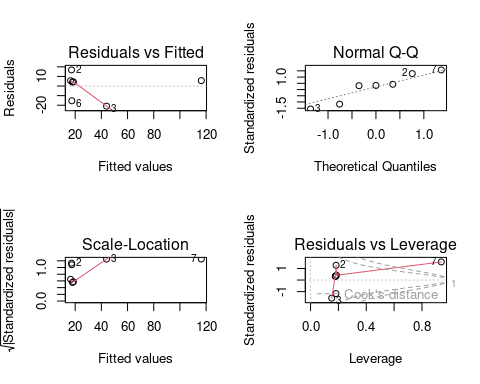
plot(cyano ~ SRP)  
abline(HAB.lm)



### Linear Regression ’Assumptions

Let’s look at a default plot of model…

par(mfrow=c(2,2))  
plot(HAB.lm)



## Logistic Regression

This data predicts graduate school admission (discreet/binary) based on GRE score (continuous), GPA (continuous), and school prestige (discreet/binary, 1-4). This is a great resource to perform this: <https://stats.oarc.ucla.edu/r/dae/logit-regression/>

#### Name the data  
mydata <- read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")  
  
#### View the first few rows of the data  
head(mydata)

## admit gre gpa rank  
## 1 0 380 3.61 3  
## 2 1 660 3.67 3  
## 3 1 800 4.00 1  
## 4 1 640 3.19 4  
## 5 0 520 2.93 4  
## 6 1 760 3.00 2

#### Create a 5 number summary  
summary(mydata)

## admit gre gpa rank   
## Min. :0.0000 Min. :220.0 Min. :2.260 Min. :1.000   
## 1st Qu.:0.0000 1st Qu.:520.0 1st Qu.:3.130 1st Qu.:2.000   
## Median :0.0000 Median :580.0 Median :3.395 Median :2.000   
## Mean :0.3175 Mean :587.7 Mean :3.390 Mean :2.485   
## 3rd Qu.:1.0000 3rd Qu.:660.0 3rd Qu.:3.670 3rd Qu.:3.000   
## Max. :1.0000 Max. :800.0 Max. :4.000 Max. :4.000

I have simplified it to one variable… trying to stay focused on simple analyses!

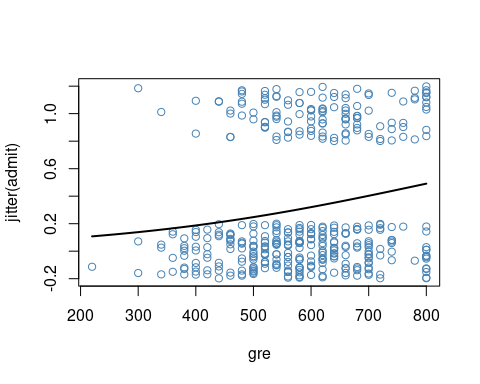
mydata$rank <- factor(mydata$rank)  
mylogit <- glm(admit ~ gre, data = mydata, family = "binomial")

summary(mylogit)

##   
## Call:  
## glm(formula = admit ~ gre, family = "binomial", data = mydata)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.1623 -0.9052 -0.7547 1.3486 1.9879   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.901344 0.606038 -4.787 1.69e-06 \*\*\*  
## gre 0.003582 0.000986 3.633 0.00028 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 499.98 on 399 degrees of freedom  
## Residual deviance: 486.06 on 398 degrees of freedom  
## AIC: 490.06  
##   
## Number of Fisher Scoring iterations: 4

#define new data frame that contains predictor variable  
newdata <- data.frame(gre=seq(min(mydata$gre), max(mydata$gre),len=100))  
  
#use fitted model to predict values of vs  
newdata$admit = predict(mylogit, newdata, type="response")

#plot logistic regression curve  
plot(jitter(admit) ~ gre, data=mydata, col="steelblue")  
lines(admit ~ gre, newdata, lwd=2)



## ANOVA

### Example #1

##Research question: Which fertilizer produces the most crop yield? ##Independent variable/predictor: Fertilizer A, Fertilizer B, Fertilizer C ##Dependent variable/response: amount of crop yield ##Question: Which fertilizer provides the best crop yield?## ##LookedatScribbr, inputed data in R according to directions “ANOVA in R | A Complete Step-by-step Guide”## ##Getting started in R## ##If you havenâ€™t used R before, start by downloading R and R Studio. Once you have both of these programs downloaded, open R Studio and click on File > New File > R Script.

##Now you can copy and paste the code from the rest of this example into your script. To run the code, highlight the lines you want to run and click on the Run button on the top right of the text editor (or press ctrl + enter on the keyboard).

##Install and load the packages# ##First, install the packages you will need for the analysis (this only needs to be done once): ##Then load these packages into your R environment (do this every time you restart the R program): ##Step 1: Load the data into R ##Note that this data was generated for this example, itâ€™s not from a real experiment.

##We will use the same dataset for all of our examples in this walkthrough. The only difference between the different analyses is how many independent variables we include and in what combination we include them.

##It is common for factors to be read as quantitative variables when importing a dataset into R. To avoid this, you can use the read.csv() command to read in the data, specifying within the command whether each of the variables should be quantitative (â€œnumericâ€) or categorical (â€œfactorâ€).

##Use the following code, replacing the path/to/your/file text with the actual path to your file:

# install.packages(c(“ggplot2”, “ggpubr”, “tidyverse”, “broom”, “AICcmodavg”))   
  
# Read Data  
# crop.data.csv = file.choose()  
  
crop.data.csv = "/home/mwl04747/beginnersluck/Stats/EA30\_Study\_Guide/ANOVA\_cropdata.csv"  
crop.data <- read.csv(crop.data.csv)  
  
names(crop.data)

## [1] "Density" "Block" "Fertilizer" "Yield"

crop.data.aov <- aov(Yield ~ Fertilizer, data=crop.data)  
  
summary(crop.data.aov)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Fertilizer 1 5.74 5.743 14.91 0.000207 \*\*\*  
## Residuals 94 36.21 0.385   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

## Example 2

Five mesocosms have been randomly selected to test the impact of zebra mussels on cyanobacteria concentrations, while five are controls. Water is circulate in each, and the experiment sampled the water column for bacteria counts every day for 10 days.

Here are the (fake) data:

mesocosm = rep(1:10, 10); mesocosm

## [1] 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5  
## [26] 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10  
## [51] 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5  
## [76] 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 10

treatment = c(rep("Control", 10), rep("Control", 10), rep("Control", 10),  
 rep("Control", 10), rep("Control", 10), rep("Mussel", 10),  
 rep("Mussel", 10), rep("Mussel", 10), rep("Mussel", 10),   
 rep("Mussel", 10)); treatment

## [1] "Control" "Control" "Control" "Control" "Control" "Control" "Control"  
## [8] "Control" "Control" "Control" "Control" "Control" "Control" "Control"  
## [15] "Control" "Control" "Control" "Control" "Control" "Control" "Control"  
## [22] "Control" "Control" "Control" "Control" "Control" "Control" "Control"  
## [29] "Control" "Control" "Control" "Control" "Control" "Control" "Control"  
## [36] "Control" "Control" "Control" "Control" "Control" "Control" "Control"  
## [43] "Control" "Control" "Control" "Control" "Control" "Control" "Control"  
## [50] "Control" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel"   
## [57] "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel"   
## [64] "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel"   
## [71] "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel"   
## [78] "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel"   
## [85] "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel"   
## [92] "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel" "Mussel"   
## [99] "Mussel" "Mussel"

day = rep(1, times=10 ); day

## [1] 1 1 1 1 1 1 1 1 1 1

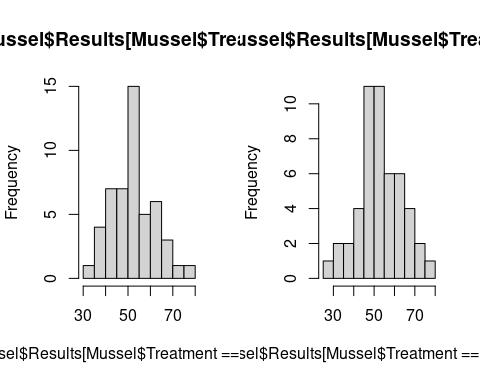
results = round(c(rnorm(50, 50, 10), rnorm(50, 54, 10)), 1); results

## [1] 49.8 47.2 45.0 50.5 49.8 53.6 51.8 34.6 56.0 69.6 55.2 50.2 64.1 43.8 50.6  
## [16] 43.9 61.1 66.1 35.3 54.3 52.7 62.5 35.9 48.9 61.8 56.8 40.3 51.9 44.0 71.0  
## [31] 58.2 54.5 46.3 44.8 39.3 78.2 52.2 39.7 54.8 58.7 54.1 48.9 62.0 67.9 46.6  
## [46] 54.3 52.5 53.4 63.1 43.0 51.9 29.7 54.1 67.0 42.4 43.4 65.9 36.4 46.1 51.4  
## [61] 78.1 51.3 69.6 51.6 49.5 46.1 55.0 65.1 45.1 48.6 57.9 32.4 53.6 71.0 46.1  
## [76] 45.0 63.9 57.5 45.1 57.2 45.5 38.0 55.9 62.9 70.7 54.1 42.7 48.0 31.5 62.0  
## [91] 54.8 63.8 63.5 47.5 58.8 46.2 55.6 52.7 50.5 61.1

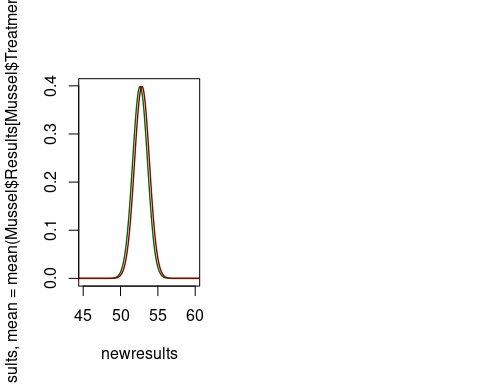
Mussel = data.frame(Mesocosm = mesocosm, Treatment = treatment, Day=day, Results=results)  
  
  
Mussel.aov <- aov(results ~ treatment, data=Mussel)  
  
summary(Mussel.aov)

## Df Sum Sq Mean Sq F value Pr(>F)  
## treatment 1 2 1.69 0.017 0.898  
## Residuals 98 9944 101.47

par(mfrow=c(1,2))  
hist(Mussel$Results[Mussel$Treatment=="Control"])  
hist(Mussel$Results[Mussel$Treatment=="Mussel"])



newresults = seq(min(Mussel$Results), max(Mussel$Results), length=500)  
plot(newresults, dnorm(newresults, mean=mean(Mussel$Results[Mussel$Treatment=="Control"])), ty="l", xlim=c(45,60), col="darkgreen", lwd=1.4)  
lines(newresults, dnorm(newresults, mean=mean(Mussel$Results[Mussel$Treatment=="Mussel"])), ty="l", col="darkred", lwd=1.4)



## Tests for Association

# Example 2

Hospital asthema cases and HABs in Florida

HAB.tab = c(20, 15, 98, 223)  
sum(HAB.tab)

## [1] 356

dim(HAB.tab) = c(2,2)  
dimnames(HAB.tab) = list(c("Non-HAB", "HAB"), c("No Hospitalizations", "Hospitalizations"))  
  
HAB.tab

## No Hospitalizations Hospitalizations  
## Non-HAB 20 98  
## HAB 15 223

#HAB.mat = matrix(HAB.tab, nrow=2)

table(HAB.tab)

## HAB.tab  
## 15 20 98 223   
## 1 1 1 1

chisq.test(HAB.tab)

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: HAB.tab  
## X-squared = 8.9217, df = 1, p-value = 0.002818

# Other Types the Might be Important (Marc might discuss if we have time…)

## Multiple Regression

## Time Series Data

## Count Data

## Mixed Effects Models