# Lab 1

author: Kyle N. Payne date: 8/8/2015

# Getting Started

* Lab Time : Wed 12-1:50pm
* Course Website : compass2g.illinois.edu
* TA Name : Kyle N. Payne, (Kyle, K-Pax, "Hey...man")
* TA Office Hours : Mon 8:30am-9:30am
* TA Email : knpayne2@illinois.edu

# Book

* Title : An Introduction to Statistical Methods and Data Analysis
* Author(s) : R Lyman Ott, Michael Longnecker
* Edition : 6th
* Can be found on most retailers like Amazon.com, abebooks.com, etc...
* Also can be found by google-ing Title + 'pdf'

# Software

* Software is a huge component of statistical analysis
* We will support SAS and R usage for this course
* SAS is a suite of tools for statistical analysis
* R is a programming language primarily for statistical analysis

# SAS

SAS inc. is the worlds largest privately held software company.

* Used by most fortune 500 companies http://archive.fortune.com/2010/01/21/technology/sas\_best\_companies.fortune/index.htm
* SAS Software has over 200 components
* Can perform many statistical analyses with limited effort
* Presents a robust suite of industry standard tools. -https://en.wikipedia.org/wiki/SAS\_(software)#cite\_note-encycl-2

# R

* "Much closer to the metal" than SAS
* Can be used to create your own functions, libraries, etc...
* Requires more effort
* But this results in arguably better skill development

# Accessing SAS

* N120
* Many campus computer labs
* Remotely through ACES
* SAS OnDemand

# Accessing R

* Rstudio : https://www.rstudio.com/
* On many workstation on campus
* Can be downloaded on most computers
* Very lightweight software

# Reading Data SAS --- Inline

{r, eval=FALSE} /\* data read inline from a within a SAS document \*/ data survey; INPUT id sex $ age inc r1 r2 r3 ; DATALINES; 1 F 35 17 7 2 2 17 M 50 14 5 5 3 33 F 45 6 7 2 7 49 M 24 14 7 5 7 65 F 52 9 4 7 7 81 M 44 11 7 7 7 2 F 34 17 6 5 3 18 M 40 14 7 5 2 34 F 47 6 6 5 6 50 M 35 17 5 7 5 ; Reading Data SAS --- Inline ======================================================== {r, eval=FALSE} proc print data= survey; run; proc univariate data=survey; run;

# Reading Data SAS --- Inline

{r, eval=FALSE} /\* data read exactly from where in the document it occurs \*/ data thisone; input name $1-20 age city $29-37 state $ ; datalines; Oliver Schabenberger 33 Lansing MI John T. Smith 37 New York NY ; proc print data=thisone; run; Reading Data SAS --- Inline ======================================================== ```{r, eval=FALSE} options nocenter; DATA survey; INPUT id sex $ age inc r1 r2 r3 ; DATALINES; 1 F 35 17 7 2 2 17 M 50 14 5 5 3 33 F 45 6 7 2 7 49 M 24 14 7 5 7 ; PROC PRINT; RUN;

```

# Reading Data R ---- Inline

{r, echo=TRUE, eval=FALSE} ## FROM THE FIRST HW :) pr\_3\_data <- scan( text='26.6 27.8 26.5 26.7 30.6 28.9 30.4 29.2 22.6 27.0 28.2 24.7 27.6 27.6 27.2 ...')

# Reading Data R ---- Inline

{r, echo=TRUE, eval=FALSE} ## FROM THE FIRST HW :) pr\_3\_data <- c(26.6,27.8,26.5,26.7,30.6, 28.9,30.4,29.2,22.6,27.0, 28.2,24.7,27.6,27.6,27.2)

# Reading Data SAS ---- CSV

````{r, echo=TRUE, eval=FALSE} proc import datafile="/home/knpayne2/sas\_examples/ Data\_Cortex\_Nuclear\_2.csv" out=dcn dbms=csv replace; getnames=yes; run;

proc univariate data=dcn; run; ````

# Reading Data SAS ---- CSV

Let's try this example!

* But first note,
* $ in SAS means a character variable
* By default, SAS assumes that a variable is numeric
* If SAS encodes a variable as numeric, but encounters a character it will code is as ., which means missing.

# Reading Data R ---- CSV

{r, eval=TRUE, echo=TRUE} data\_cortex <- read.csv("~/Desktop/data\_cortex.csv") ## kind of ugly output, but just as useful Or just use the "import dataset" button in Rstudio

# Reading Data R ---- CSV

Let's try this example!

# Basic Descriptive Statistics ---- SAS

* proc univariate
* proc means

# proc univariate ---- SAS

````{r, eval=FALSE, echo=TRUE} proc univariate ; run;

proc univariate data=dcn; var ITSN1\_N; run; ````

# proc means ---- SAS

````{r, eval=FALSE, echo=TRUE} proc meansa ; run;

proc means data=dcn; var ITSN1\_N; run; ````

# Basic Descriptive Statistics ---- R

{r, eval=TRUE, echo=TRUE} summary(data\_cortex[2], na.rm=TRUE)

# Basic Descriptive Statistics ---- R

{r, eval=TRUE, echo=TRUE} var(data\_cortex[2], na.rm=TRUE)

# Basic Descriptive Statistics ---- R

{r, eval=TRUE, echo=TRUE} colMeans(data\_cortex[c(2,5)], na.rm=TRUE)

# Basic Descriptive Statistics ---- R

{r, eval=TRUE, echo=TRUE} sapply(data\_cortex[2:5], mean, na.rm=TRUE)

# Basic Descriptive Statistics ---- R

sapply is extremely useful for working with data in R. In order to understand sapply better we first need to understand the some basic data structures.

# Data Structures ---- R

R's base data structures can be organised by their dimensionality (1d, 2d, or nd) and whether they're homogeneous (all contents must be of the same type) or heterogeneous (the contents can be of different types). This gives rise to the five data types most often used in data analysis:

|  |  |  |
| --- | --- | --- |
|  | Homogeneous | Heterogeneous |
| 1d | Atomic vector | List |
| 2d | Matrix | Data frame |
| nd | Array |  |

* ("Advanced R", Wickham 2014) http://adv-r.had.co.nz/

# Data Structures ---- R ---- Vectors

* 1 dimensional data structures
* Must all be one type i.e. "homogeneous"
* Formed by the "c()" operator e.g.

{r, eval=TRUE, echo=TRUE, cache=TRUE} data <- c(1,2,3,4,10) data data <- c("Help", "Me", "I'm", "Not", "Good", "At", "Examples") data Data Structures ---- R ---- Vectors ======================================================== What happens with mixed types? {r, eval=TRUE, echo=TRUE, cache=TRUE} data<-c("r", "u", "da", 1) data

# Data Structures ---- R ---- Lists

* 1 dimensional data structures
* Can be mixed types i.e. "heterogeneous"
* Formed by the "list()" function {r, eval=TRUE, echo=TRUE} data <- list(1,2,3) data Data Structures ---- R ---- Lists ========================================================
* Mixed Types? {r, eval=TRUE, echo=TRUE} data <- list("r", "u", "da", 1) data Data Structures ---- R ---- data.frames ========================================================
* What are lists that contain vectors? Data Frames.
* R structures data files e.g. .csv, .txt, etc... as data frames
* These come with a bunch of useful functionality
* You will see these alot.

# Subsetting ---- SAS ---- Variables

* keep ```{r, eval=FALSE, echo=TRUE} data dcn\_sub; set dcn; keep MouseID DYRK1A\_N ITSN1\_N; run; ````

# Subsetting ---- SAS ---- Observations

* if

{r, eval=FALSE, echo=TRUE} data dcn\_sub\_no\_miss; set dcn; keep MouseID DYRK1A\_N ITSN1\_N; if MouseID ^= . OR DYRK1A\_N ^=. OR ITSN1\_N ^=.; run; /\* Subset based on value \*/ data dcn\_sub\_gt\_50; set dcn\_sub\_no\_miss; if DYRK1A\_N > .5; run;

# Subsetting ---- R ---- Variables

{r, eval=TRUE, echo=TRUE} library(knitr) ## grabs the three columns MouseID, DYRK1A\_N, ITSN1\_N data\_cor\_sub <- data\_cortex[, c("MouseID", "DYRK1A\_N", "ITSN1\_N")] kable(summary(data\_cor\_sub))

# Subsetting ---- R ---- Observations

{r, eval=TRUE, echo=TRUE} data\_cor\_no\_rm <- data\_cor\_sub[c(2,5),] data\_cor\_no\_rm

# Subsetting ---- R ---- Observations

{r, eval=TRUE, echo=TRUE} library(knitr) # for the kable function ## select all observations > .5 ## on DYRK1A\_N data\_cor\_gt\_.5 <- data\_cor\_sub[which(data\_cor\_sub$DYRK1A\_N > .5),] kable(head(data\_cor\_gt\_.5))

# Subsetting ---- R

* In general, one can select rows of a dataframe by subsetting on l.h.s. of [,]'s {r, eval=FALSE, echo=TRUE} name\_of\_df[rows,columns]
* one can select columns by subsetting on r.h.s. of [,]
* We can also use the {r, eval=FALSE} subset() function

# Activity

* Using either R or SAS:
* Import the Mouse cortex dataset
* Select the variables H3MeK4\_N, TIAM1\_N, Ubiquitin\_N, Genotype, MouseID
* Remove missing values from the dataset
* Calculate the means of all of the numeric variables

# Basic Statistical Graphics

* Visualization is hugely important
* Histogram, Boxplot

# Basic Statistical Graphics ---- Histogram

{r, eval=TRUE, echo=TRUE} data\_cor\_graph <- data\_cortex[,c("H3MeK4\_N", "TIAM1\_N", "Ubiquitin\_N", "Genotype", "MouseID")] data\_cor\_graph <- data\_cor\_graph[complete.cases(data\_cor\_graph),] hist(data\_cor\_graph$H3MeK4\_N)

{r, echo=TRUE, eval=FALSE} proc univariate data=dcn\_sub\_act; var H3MeK4\_N; histogram; run;

# Basic Statistical Graphics ---- Boxplot

{r, eval=TRUE, echo=TRUE} boxplot(data\_cor\_graph$H3MeK4\_N)

{r, eval=FALSE, echo=FALSE} proc univariate data=dcn\_sub\_act plot; var H3MeK4\_N; run; proc sgplot data=dcn\_sub\_act; vbox H3MeK4\_N; run;

# Activity

* Using either R or SAS:
* Take the variable H3MeK4\_N
* Make a histogram and boxplot

# Getting help with software

* Google.com
* Stackoverflow.com
* SAS documentation (at your own risk!)
* R documentation
* TAs