Lab 1

Kyle N. Payne
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 : Wed 10:00am-11:00am
- TA Email :

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

- SAS Software has over 200 components
  - Can perform many statistical analyses with limited effort
  - Presents a robust suite of industry standard tools. -

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 :
- On many workstation on campus
- Can be downloaded on most computers
- Very lightweight software

Reading Data SAS --- Inline

```
/*
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

```
proc print data= survey;
run;
proc univariate data=survey;
run;
```

Reading Data SAS --- Inline

```
/*
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

```
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

```
## 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
## 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

```
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

```
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

```
proc univariate <options> ;
  <options>
run;

proc univariate data=dcn;
  var ITSN1_N;
run;
```

proc means ---- SAS

```
proc meansa <options> ;
  <options>
run;

proc means data=dcn;
  var ITSN1_N;
run;
```

Basic Descriptive Statistics ---- R

```
summary(data_cortex[2], na.rm=TRUE)
```

```
   DYRK1A_N
 Min.   :0.1453
 1st Qu.:0.2881
 Median :0.3664
 Mean   :0.4258
 3rd Qu.:0.4877
 Max.   :2.5164
 NA's   :3
```

Basic Descriptive Statistics ---- R

```
var(data_cortex[2], na.rm=TRUE)
```

```
           DYRK1A_N
DYRK1A_N 0.06218139
```

Basic Descriptive Statistics ---- R

```r
colMeans(data_cortex[c(2,5)], na.rm=TRUE)
```

```
 DYRK1A_N     NR1_N
0.4258102 2.2972691
```

Basic Descriptive Statistics ---- R

```
sapply(data_cortex[2:5], mean, na.rm=TRUE)
```

```
 DYRK1A_N   ITSN1_N    BDNF_N     NR1_N 
0.4258102 0.6171020 0.3190884 2.2972691 
```

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:

1d	Atomic vector	List
2d	Matrix	Data frame
nd	Array	

- (“Advanced R”, Wickham 2014)

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

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

```r
data <- c(1,2,3,4,10)
data
```

```
[1]  1  2  3  4 10
```

```r
data <- c("Help", "Me", "I'm", "Not", "Good",
"At", "Examples")
data
```

```
[1] "Help"      "Me"        "I'm"       "Not"
"Good"      "At"
[7] "Examples"
```

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

What happens with mixed types?

```r
data<-c("r", "u", "da", 1)
data
```

```
[1] "r"  "u"  "da" "1"
```

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

- 1 dimensional data structures
- Can be mixed types i.e. “heterogeneous”
- Formed by the “list()” function

```
data <- list(1,2,3)
data
```

```
[[1]]
[1] 1

[[2]]
[1] 2

[[3]]
[1] 3
```

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

- Mixed Types?

```r
data <- list("r", "u", "da", 1)
data
```

```
[[1]]
[1] "r"

[[2]]
[1] "u"

[[3]]
[1] "da"

[[4]]
[1] 1
```

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

```
data dcn_sub;
  set dcn;
  keep MouseID DYRK1A_N ITSN1_N;
run;
```

Subsetting ---- SAS ---- Observations

- if

```
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

```
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))
```

	18899_1 : 1	Min. :0.1453	Min. :0.2454
	18899_10: 1	1st Qu.:0.2881	1st Qu.:0.4734
	18899_11: 1	Median :0.3664	Median :0.5658
	18899_12: 1	Mean :0.4258	Mean :0.6171
	18899_13: 1	3rd Qu.:0.4877	3rd Qu.:0.6980
	18899_14: 1	Max. :2.5164	Max. :2.6027
	(Other) :1074	NA's :3	NA's :3

Subsetting ---- R ---- Observations

```r
data_cor_no_rm <- data_cor_sub[c(2,5),]
data_cor_no_rm
```

```
  MouseID  DYRK1A_N   ITSN1_N
2   309_2 0.5146171 0.6890635
5   309_5 0.4349402 0.6174298
```

Subsetting ---- R ---- Observations

```
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))
```

1	309_1	0.5036439	0.7471932
2	309_2	0.5146171	0.6890635
3	309_3	0.5091831	0.7302468
16	311_1	0.7431179	0.8626527
17	311_2	0.7114799	0.8070539
18	311_3	0.7046332	0.8025372

Subsetting ---- R

- In general, one can select rows of a dataframe by subsetting on l.h.s. of [,]'s

```
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

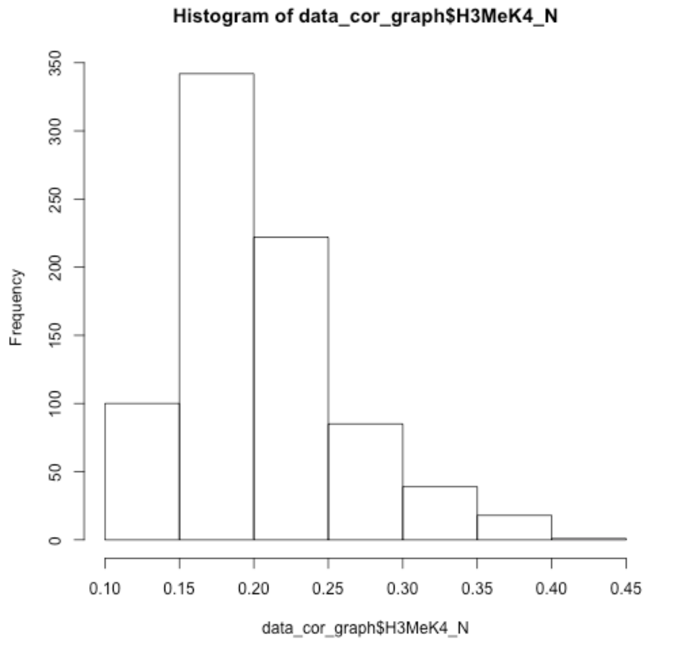
Basic Statistical Graphics

- Visualization is hugely important
- Histogram, Boxplot

Basic Statistical Graphics ---- Histogram

```
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)
```

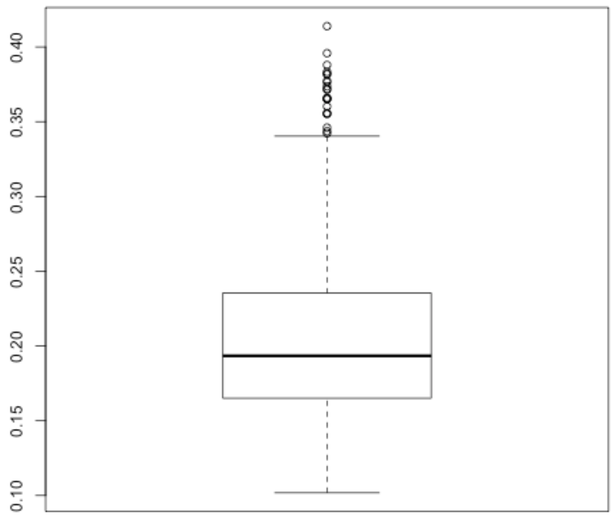


```
proc univariate data=dcn_sub_act;
  var H3MeK4_N;
  histogram;
```

```
run;
```

Basic Statistical Graphics ---- Boxplot

```
boxplot(data_cor_graph$H3MeK4_N)
```



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