

R Handout - R Terminology

Northland College, Midwest (Wichita)

Dr. Derek Ogle

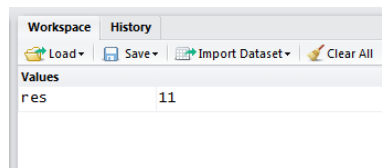
Dec 2013

```
> library(FSA)           # Subset, view, lencat, mrClosed
```

0.1 Very Basics

Expressions & Assignments

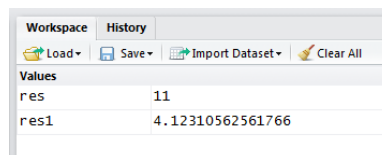
```
> 3+4*2
[1] 11
> res <- 3+4*2
> res
[1] 11
```



Workspace	
History	
Load	Save
Import Dataset	Clear All
Values	
res	11

Functions & Arguments

```
> sqrt(17)
[1] 4.123
> ( res1 <- sqrt(17) )
[1] 4.123
```



Workspace	
History	
Load	Save
Import Dataset	Clear All
Values	
res	11
res1	4.12310562561766

```
> dat <- c(3,6,8,3,5,6,2,7,6,8,2,10)
> mean(dat)
[1] 5.5
> mean(dat,trim=0.1)
[1] 5.4
```

Types of Functions

```
> mr1 <- mrClosed(346,184,49)
> summary(mr1)
Used the 'naive' Petersen method with M=346, n=184, and m=49.
```

```
      N
[1,] 1299
> confint(mr1)
The binomial method was used.
```

```
      95% LCI 95% UCI
[1,]      1034      1666
```

Vectors, Data Types, & Dataframes

```
> ( numSpec <- c(4,8,9,3) )
[1] 4 8 9 3
> ( lake <- c("Star","Twin","Long","Deep") )
[1] "Star" "Twin" "Long" "Deep"
> ( springFed <- c(TRUE,FALSE,FALSE,TRUE) )
[1] TRUE FALSE FALSE TRUE
> ( maxDepth <- c(6.5,7.8,3.8,25.6) )
[1] 6.5 7.8 3.8 25.6
```

```
> ( df <- data.frame(lake,numSpec,maxDepth,springFed) )
  lake numSpec maxDepth springFed
1 Star      4      6.5      TRUE
2 Twin      8      7.8     FALSE
3 Long      9      3.8     FALSE
4 Deep      3     25.6      TRUE
> str(df)
'data.frame': 4 obs. of 4 variables:
 $ lake      : Factor w/ 4 levels "Deep","Long",...: 3 4 2 1
 $ numSpec   : num  4 8 9 3
 $ maxDepth  : num  6.5 7.8 3.8 25.6
 $ springFed : logi  TRUE FALSE FALSE TRUE
```

Workspace History	
Load Save Import Dataset Clear All	
Data	
df	4 obs. of 4 variables
Values	
dat	numeric[12]
lake	character[4]
maxDepth	numeric[4]
mr1	MRC[11]
numSpec	numeric[4]
res	11
res1	4.12310562561766
springFed	logical[4]

0.2 Dataframes from External Files

Working Directory

You must change (i.e., “set”) the working directory in R to where your external data file is located. The easiest way to do this is to save your R script file in the same directory that holds your external data file. If you do this, then you can select the “Session” menu, “Set Working Directory” submenu, and “To Source File Location” item to set the working directory. This will send a `setwd()` command to the *Console* pane. This command should then be copied from the Console and pasted into your R script in the Script Editor so that, in the future, you can set the working directory by submitting the `setwd()` command rather than using the menu options (which requires user interaction with RStudio). I set the working directory for **MY COMPUTER** below.

```
> setwd("C:/aaaWork/Web/fishR/courses/Midwest2012/CourseMaterial/")
```

Reading External Text (Tab-Delimited) Files

```
> ex1a <- read.table("Example1.txt",header=TRUE,sep="\t")
> str(ex1a)
'data.frame': 25 obs. of  4 variables:
 $ species: Factor w/ 2 levels "BKT","SLS": 1 1 1 1 1 1 1 1 1 1 ...
 $ t1      : int  40 47 55 75 82 97 105 106 114 171 ...
 $ w       : num  0.5 0.9 1.6 6 5.4 14.3 18 14.5 14.7 52.2 ...
 $ segment: Factor w/ 2 levels "Down","Up": 1 1 1 1 1 1 1 1 1 1 ...
> ex1a
  species t1    w segment
1     BKT 40  0.5   Down
2     BKT 47  0.9   Down
3     BKT 55  1.6   Down
4     BKT 75  6.0   Down
5     BKT 82  5.4   Down
6     BKT 97 14.3   Down
7     BKT 105 18.0   Down
8     BKT 106 14.5   Down
9     BKT 114 14.7   Down
10    BKT 171 52.2   Down
11    SLS  89  8.0   Down
12    BKT  90  7.4    Up
13    BKT 102  8.6    Up
14    BKT 107 11.5    Up
15    BKT 126 20.7    Up
16    BKT 140 28.8    Up
17    SLS  47  1.1    Up
18    SLS  49  1.6    Up
19    SLS  50  1.3    Up
20    SLS  55  1.7    Up
21    SLS  60  2.3    Up
22    SLS  62  2.8    Up
23    SLS  66  2.4    Up
24    SLS  67  3.1    Up
25    SLS  74  5.2    Up
```

Reading External CSV (comma-separated-values) Files

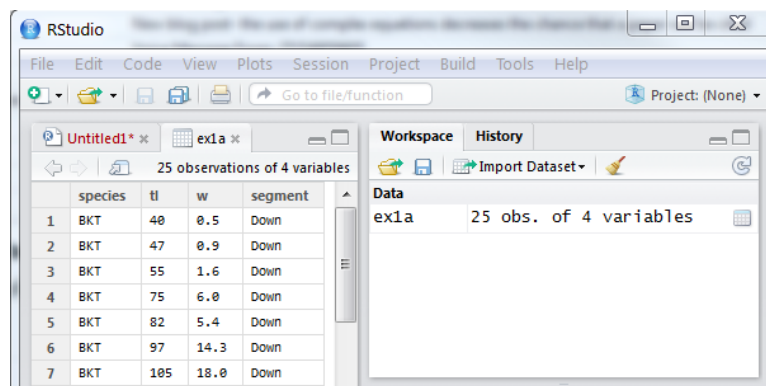
```
> ex1b <- read.csv("Example1.csv")
> str(ex1b)

'data.frame': 25 obs. of 4 variables:
 $ species: Factor w/ 2 levels "BKT","SLS": 1 1 1 1 1 1 1 1 1 1 ...
 $ tl      : int  40 47 55 75 82 97 105 106 114 171 ...
 $ w       : num  0.5 0.9 1.6 6 5.4 14.3 18 14.5 14.7 52.2 ...
 $ segment: Factor w/ 2 levels "Down","Up": 1 1 1 1 1 1 1 1 1 1 ...
```

0.3 Working With Data Frames

Seeing Entire Data Frame

One can examine the entire data frame by either typing the name of the data frame (as illustrated above for `ex1a`) or opening it in RStudio. To open the dataframe, locate and then double-click on the name of the dataframe in the *Workspace* pane (upper-right pane). This will open the dataframe in a tab in the *Script Editor* window.



Accessing Portions of the Entire Data Frame

```
> ex1a[2,]
  species tl    w segment
2    BKT 47 0.9    Down

> ex1a[,2]
 [1] 40 47 55 75 82 97 105 106 114 171 89 90 102 107 126 140 47 49 50 55
[21] 60 62 66 67 74

> ex1a$tl
 [1] 40 47 55 75 82 97 105 106 114 171 89 90 102 107 126 140 47 49 50 55
[21] 60 62 66 67 74

> ex1a$w
 [1] 0.5 0.9 1.6 6.0 5.4 14.3 18.0 14.5 14.7 52.2 8.0 7.4 8.6 11.5 20.7 28.8
[17] 1.1 1.6 1.3 1.7 2.3 2.8 2.4 3.1 5.2

> ex1a$w[2]
[1] 0.9

> ex1a$w[-2]
 [1] 0.5 1.6 6.0 5.4 14.3 18.0 14.5 14.7 52.2 8.0 7.4 8.6 11.5 20.7 28.8 1.1
[17] 1.6 1.3 1.7 2.3 2.8 2.4 3.1 5.2
```

Subsetting Data Frames

```
> ( ex1a.1 <- Subset(ex1a,segment=="Down") )
```

	species	tl	w	segment
1	BKT	40	0.5	Down
2	BKT	47	0.9	Down
3	BKT	55	1.6	Down
4	BKT	75	6.0	Down
5	BKT	82	5.4	Down
6	BKT	97	14.3	Down
7	BKT	105	18.0	Down
8	BKT	106	14.5	Down
9	BKT	114	14.7	Down
10	BKT	171	52.2	Down
11	SLS	89	8.0	Down

```
> ( ex1a.2 <- Subset(ex1a,species!="BKT") )
```

	species	tl	w	segment
11	SLS	89	8.0	Down
17	SLS	47	1.1	Up
18	SLS	49	1.6	Up
19	SLS	50	1.3	Up
20	SLS	55	1.7	Up
21	SLS	60	2.3	Up
22	SLS	62	2.8	Up
23	SLS	66	2.4	Up
24	SLS	67	3.1	Up
25	SLS	74	5.2	Up

```
> ( ex1a.3 <- Subset(ex1a,tl<50) )
```

	species	tl	w	segment
1	BKT	40	0.5	Down
2	BKT	47	0.9	Down
17	SLS	47	1.1	Up
18	SLS	49	1.6	Up

```
> ( ex1a.4 <- Subset(ex1a,tl<50 & species=="SLS") )
```

	species	tl	w	segment
17	SLS	47	1.1	Up
18	SLS	49	1.6	Up

```
> ( ex1a.5 <- Subset(ex1a,tl<50 | w>20) )
```

	species	tl	w	segment
1	BKT	40	0.5	Down
2	BKT	47	0.9	Down
10	BKT	171	52.2	Down
15	BKT	126	20.7	Up
16	BKT	140	28.8	Up
17	SLS	47	1.1	Up
18	SLS	49	1.6	Up

Creating New Variables in a Data Frame

```
> str(ex1a)
'data.frame': 25 obs. of  4 variables:
 $ species: Factor w/ 2 levels "BKT","SLS": 1 1 1 1 1 1 1 1 1 ...
 $ tl      : int   40 47 55 75 82 97 105 106 114 171 ...
 $ w       : num   0.5 0.9 1.6 6 5.4 14.3 18 14.5 14.7 52.2 ...
 $ segment: Factor w/ 2 levels "Down","Up": 1 1 1 1 1 1 1 1 1 ...

> ex1a$tl.in <- ex1a$tl/25.4
> ex1a$w.lbs <- ex1a$w/454
> str(ex1a)
'data.frame': 25 obs. of  6 variables:
 $ species: Factor w/ 2 levels "BKT","SLS": 1 1 1 1 1 1 1 1 1 ...
 $ tl      : int   40 47 55 75 82 97 105 106 114 171 ...
 $ w       : num   0.5 0.9 1.6 6 5.4 14.3 18 14.5 14.7 52.2 ...
 $ segment: Factor w/ 2 levels "Down","Up": 1 1 1 1 1 1 1 1 1 ...
 $ tl.in   : num   1.57 1.85 2.17 2.95 3.23 ...
 $ w.lbs   : num   0.0011 0.00198 0.00352 0.01322 0.01189 ...

> ex1a <- lencat(~tl,data=ex1a,breaks=c(40,75,125,175))
> str(ex1a)
'data.frame': 25 obs. of  7 variables:
 $ species: Factor w/ 2 levels "BKT","SLS": 1 1 1 1 1 1 1 1 1 ...
 $ tl      : int   40 47 55 75 82 97 105 106 114 171 ...
 $ w       : num   0.5 0.9 1.6 6 5.4 14.3 18 14.5 14.7 52.2 ...
 $ segment: Factor w/ 2 levels "Down","Up": 1 1 1 1 1 1 1 1 1 ...
 $ tl.in   : num   1.57 1.85 2.17 2.95 3.23 ...
 $ w.lbs   : num   0.0011 0.00198 0.00352 0.01322 0.01189 ...
 $ LCat    : Factor w/ 3 levels "40","75","125": 1 1 1 2 2 2 2 2 2 3 ...

> view(ex1a)
  species  tl    w segment tl.in   w.lbs LCat
9      BKT 114 14.7   Down 4.488 0.032379   75
10     BKT 171 52.2   Down 6.732 0.114978  125
12     BKT  90  7.4    Up  3.543 0.016300   75
16     BKT 140 28.8    Up  5.512 0.063436  125
17     SLS  47  1.1    Up  1.850 0.002423   40
25     SLS  74  5.2    Up  2.913 0.011454   40
```