

Unique R Objects

William F. Lamberti ¹

George Mason University

April 18, 2018

¹MS Statistical Science and
PhD Student Computational Sciences and Informatics



Outline

Introduction

Course

Session Themes

Object Type

Factors

Lists

melt()

General Tips

Introduction - Course

- ▶ 1 hour talk (6:00 PM - 7:00 PM) with 30 minutes after for discussion
- ▶ 3 Sessions Goal: Intro to R concepts in Data Cleaning
- ▶ Assuming little to no experience in coding languages like C or Java
- ▶ Will reference RGalleon.com pages for additional information
- ▶ Will *not* go over the "why" of statistics (simply do not have the time)
- ▶ If you have any questions, please feel free to ask at any point



Introduction - Session Themes

- ▶ Day 1 - Partitioning and Loading Data
- ▶ Day 2 - Unique R Objects
- ▶ Day 3 - Getting Clean Data for ggplot2

Object Types - Factors

- ▶ Factors are special means of establishing groups
- ▶ Factors used heavily in ggplot2
- ▶ Examples of potential factors
 - ▶ Binary Variables
 - ▶ Character data (i.e. - "Dog", "Cat", "Fox")



Object Types - Example Code Factors

```
1 #create factors
2 animals<-c("Dog", "Dog", "Fox", "Cat", "Dog",
3           "Fox")
4
5 #comparing objects
6 animals
7 fcts
```

Object Types - Lists

- ▶ Lists can combine multiple objects into one object
- ▶ Very flexible



Object Types - Example Code Lists

```
1 #create numeric object
2 x = c(1, 2, 3)
3
4 #create character object
5 y = c("one", "100", "fox")
6
7 #combine into list
8 z = list(x, y)
9 z
```


Object Types - Example Code Lists

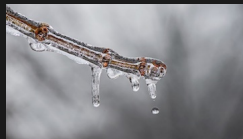
```
1 #look at first "item" on list
2 z[[1]]
3
4 #look at first part of first "item" on list
5 z[[1]][1]
6
7 #How long is the first "item"?
8 length(z[[1]])
```

Object Types - Example Code Lists

```
1 #create a list directly
2 maths = list(c(5, 10, 15), c(1, 2, 5))
3
4 #to add up everything
5 sum(maths[[1]], maths[[2]])
6 sum(as.data.frame(maths))
7
8 #convert to data frame
9 df <- as.data.frame(maths)
10 df
```

melt() - General Tips

- ▶ melt() from the reshape2 package
 - ▶ Particularly helpful for ggplot2
- ▶ Data partitioning helps in many situations



melt() - Example Code for Partitioning

```
1 #load famous iris data
2 data(iris)
3
4 #get idea of what data looks like
5 head(iris)
6
7 #learn dimension of data
8 dim(iris)
9
10 #select first 50 observations
11 mydat<-iris[1:50,]
12
13 #select 2nd and 3rd column
14 mydat2<-iris[,2:3]
```

melt() - Example Code for melt()

```
1 #load reshape2 library
2 library(reshape2)
3
4 #look at melt documentation
5 ?melt
6
7 #using melt() example
8 iris.new<-melt(data=iris, id.vars='Species')
```

Exercise - Question

- ▶ Install the 'dplyr' package
- ▶ Use the following

```
1 #load dplyr library
2 library(dplyr)
3
4 #load nasa data
5 data(nasa)
6 mydat <- as.data.frame(nasa)
```

- ▶ Use melt() on the surftemp and temperature variables grouped by the year
- ▶ Save the data as a CSV
- ▶ Remove everything from your workspace
- ▶ Load the CSV back into R
- ▶ You have 10 minutes

Exercise - Solution

```
1 #install dplyr package if needed
2 install.packages('dplyr')
3
4 #load dplyr library
5 library(dplyr)
6
7 #load nasa data
8 data(nasa)
9 mydat<-as.data.frame(nasa)
10
11 #obtaining subset of data
12 subs<-cbind(mydat[,4],mydat[,10:11])
13 colnames(subs)[1]<-c("year")
```

Exercise - Solution

```
1 #using the melt command to create new data
2 new.dat<-melt(data=subs, id.vars='year')
3
4 #writing as CSV
5 write.csv(new.dat, "nasa_temp.csv")
6
7 #clear workspace
8 rm(list=ls())
9 ls()
10
11 #checking directory
12 dir()
13
14 #loading nasa temp data csv
15 nasa_temp<-read.table("nasa_temp.csv", header
    =TRUE, sep=",")
16
17 head(nasa_temp)
```


Any Questions?



Introducing ggplot2 - General Tips

- ▶ Powerful visualization package available on R
- ▶ Sacrifices complete control to gain accessibility and speed
- ▶ Heavily relies on ggplot()
- ▶ <http://ggplot2.tidyverse.org/reference/> is a good resource

Introducing ggplot2 - ggplot()

```
1 #load ggplot2
2 library(ggplot2)
3
4 #load data
5 data(iris)
6 df<-as.data.frame(iris)
7
8 #create scatterplot
9 s<-ggplot(data=df, aes(x=Sepal.Length, y=
    Sepal.Width))
10 s
11
12 s<-s + geom_point()
13 s
```

Introducing ggplot2 - Exercise

- ▶ Use the iris data set
- ▶ Create scatterplot of petal length vs. petal width
- ▶ Bonus: Use factors to create colors for points by species
- ▶ You have 15 minutes

Introducing ggplot2 - ggplot()

```
1 #load ggplot2
2 library(ggplot2)
3
4 #load data
5 data(iris)
6 df<-as.data.frame(iris)
7
8 #create scatterplot
9 s<-ggplot(data=df, aes(x=Petal.Length, y=
    Petal.Width))
10 s1<-s + geom_point()
11 s1
12
13 #include factors
14 s2<-s + geom_point(aes(colour = factor(
    Species)))
15 s2
```