L1 Exploring Data

kmg

1/13/2019

Reading in Data

Data can be stored in various formats. Here we demonstrate how you can read in and manipulate the three data sets that have common data formats. We use data that have been made available with this course.

Long Form Data

Long form data represents the most typical form of data storage in ecological momentary assessments. Here, each individual's data are pasted below each other individual's data to arrive at one long concatenated data set.

First, we read in the data:

```
WrightData_LF <-read.csv("~/Dropbox/Classes/IAV/Data/Wright/Daily Diary Long Form.csv", header = TRUE)
```

We'll put the data in list form so that it is easier to manipulate and select each individual.

In list form, each person's matrix is saved as a slice or element of the list.

```
# View first few rows:
head(WrightData_LF[,1:6])
```

```
##   ID  Datessmt True_date Time Weekday Weekend
## 1  1 2/11/2013 2/11/2013    1       1       0
## 2  1 2/11/2013 2/12/2013    2       2       0
## 3  1 2/11/2013 2/13/2013    3       3       0
## 4  1 2/11/2013 2/14/2013    4       4       0
## 5  1 2/11/2013 2/15/2013    5       5       0
## 6  1 2/11/2013 2/16/2013    6       6       1
```

```
# We can see that the columns are variables and the rows are observations across time.

# "ID" is how each participant is identified here.
WrightData<-split(WrightData_LF,WrightData_LF$ID)
length(WrightData)
```

```
## [1] 112
```

WrightData is a list, and has the same number of slices as there are individuals in the data set ($N = 112$). We'll explore how to manipulate lists below.

Separate Data Files

Functional MRI studies often have a separate data file for each individual. For instance, when pulling the time courses for brain region activity the MarsBar tool in SPM creates a different .csv file for each individual.

The Borkenau-Orsendorf data also are in the form of separate .csv files for each individual. Let's load each one and put them all in a list.

```
BorkLoc <- "~/Dropbox/Classes/IAV/Data/Borkenau/Data/"
file.names <- list.files(BorkLoc)
BorkData <- list()
```

```
for (p in 1:length(file.names))
  BorkData[[p]] <- read.csv(paste0(BorkLoc, file.names[p]), header = FALSE)
```

Rdata

Fisher's data was originally obtained from Aaron Fisher's OSF. The original format of the data was separate .Rdata files for each individual that also contained output related to the paper.

The file provided to this class contains two lists: one with the raw data for each individual, the other with the raw interpolated data. To read in data that are in .Rdata format, you simply load the file.

```
load("~/Dropbox/Classes/IAV/Data/Fisher/FisherData.Rdata")
```

Two lists appear in your environment. One is called "FisherData", the other is "FisherDataInterp". Let's just work with the raw data today.

Manipulating lists.

Now all the data are in list form. To look at one indivudal, we can either use numeric indexing to select a given slice (or individual data matrix) of the list or, since the slices here are named, we can use the slice names by using $.

Let's use FisherData as an example.

Selecting one individual using numeric indexing (i.e., the number the matrix is in the list):

```
person1 <- FisherData[[1]]
# View first rows of data using "head"; select only columns 1:5
head(person1[,1:5])
```

```
##               start            finish energetic enthusiastic content
## 1 10/21/2014 17:55 10/21/2014 17:58        19           23      33
## 2 10/21/2014 21:20             <NA>        NA           NA      NA
## 3 10/22/2014 13:38             <NA>        NA           NA      NA
## 4 10/22/2014 17:44 10/22/2014 18:50        37           31      25
## 5 10/22/2014 21:40 10/22/2014 21:53        20           39      40
## 6  10/23/2014 9:20  10/23/2014 9:36        18           17      36
```

Selecing one individual using $:

```
# names of slices
names(FisherData)
```

```
##  [1] "P001_final" "p003_final" "p004_final" "p006_final" "p007_final"
##  [6] "p008_final" "P009_final" "P010_final" "P012_final" "P013_final"
## [11] "P014_final" "P019_final" "P021_final" "P023_final" "P025_final"
## [16] "P033_final" "P037_final" "P040_final" "P048_final" "P068_final"
## [21] "P072_final" "P074_final" "P075_final" "P100_final" "P111_final"
## [26] "P113_final" "P115_final" "P117_final" "P127_final" "P137_final"
## [31] "P139_final" "P145_final" "P160_final" "P163_final" "P169_final"
## [36] "P202_final" "P203_final" "P204_final" "P215_final" "P217_final"
```

```
person1 <- FisherData$P001_final
# View first rows of data using "head"; select only columns 1:5
head(person1[,1:5])
```

```
##               start            finish energetic enthusiastic content
## 1 10/21/2014 17:55 10/21/2014 17:58        19           23      33
## 2 10/21/2014 21:20             <NA>        NA          NA      NA
## 3 10/22/2014 13:38             <NA>        NA          NA      NA
## 4 10/22/2014 17:44 10/22/2014 18:50        37          31      25
## 5 10/22/2014 21:40 10/22/2014 21:53        20          39      40
## 6  10/23/2014 9:20  10/23/2014 9:36        18          17      36
```

We can also look at the names of the data matrices that comprise each slice of the list.

```
names(FisherData$P001_final)
```

```
##  [1] "start"        "finish"       "energetic"    "enthusiastic"
##  [5] "content"      "irritable"    "restless"     "worried"
##  [9] "guilty"       "afraid"       "anhedonia"    "angry"
## [13] "hopeless"     "down"         "positive"     "fatigue"
## [17] "tension"      "concentrate"  "accepted"     "threatened"
## [21] "ruminate"     "avoid_act"    "reassure"     "procrast"
## [25] "hours"        "difficult"    "unsatisfy"    "avoid_people"
```

If there are only a few variables we want to work with we can create a new data set with fewer variables:

```
library(dplyr)
```

```
## 
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
## 
##     filter, lag

## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

```
FisherData_fewer <- FisherData
for(i in 1:length(FisherData))
  FisherData_fewer[[i]] <- FisherData_fewer[[i]] %>%
  select(energetic, irritable, angry, hopeless, positive)

# Make sure you selected the variables you wanted for a random person:
names(FisherData_fewer[[2]])
```

```
## [1] "energetic" "irritable" "angry"     "hopeless"  "positive"
```

Maybe we only want to retain individuals who provided at least $T = 120$ time points.

```
FisherData120 <-FisherData[lapply(1:length(FisherData), function(x)
  length(complete.cases(FisherData[[x]])))>120]

# We can see if any individuals were removed by looking at the length of the lists.
# All individuals:
length(FisherData)
```

```
## [1] 40
```

```
# Individuals with more than 120 time points:
length(FisherData120)
```

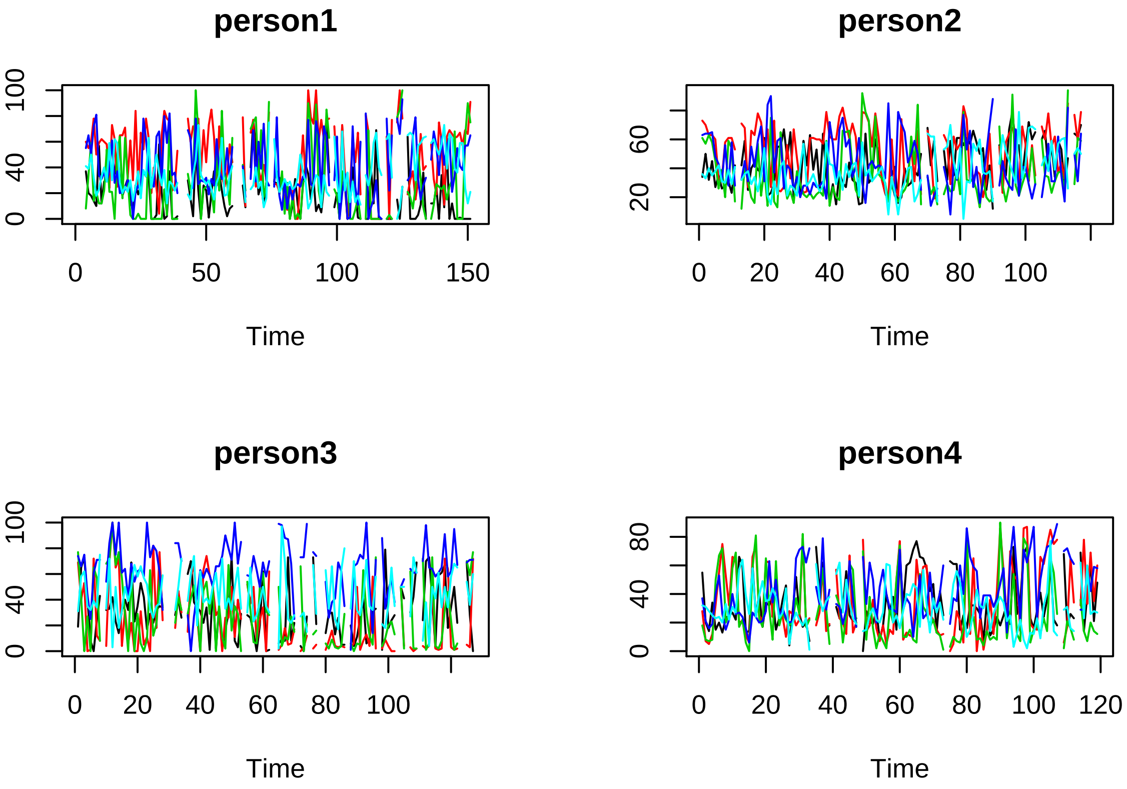
```
## [1] 33
```

Plotting Data

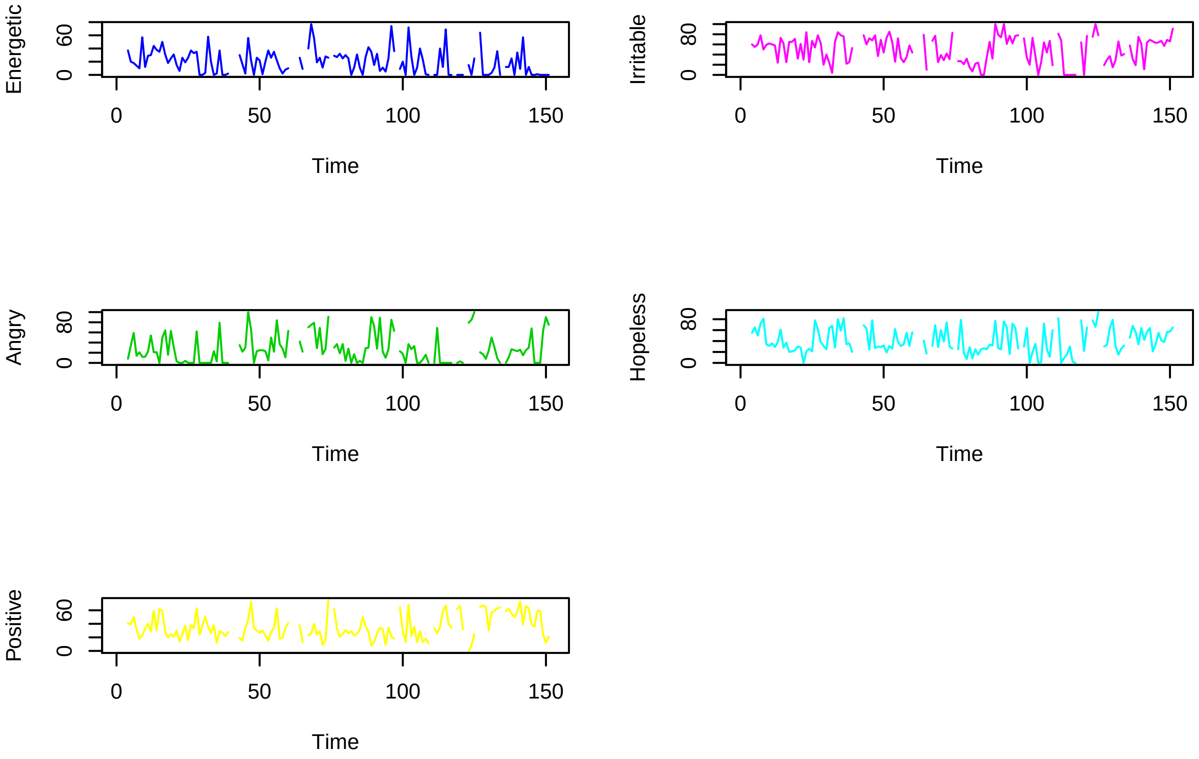
Let's take a look at these variables for a few people.

```
# Set up the figure to contain 4 graphs
par(mfrow = c(2,2))

# plot as a time series by using "ts"
for (p in 1:4)
  ts.plot(FisherData_fewer[[p]], col = 1:5, main = paste0("person", p))
```



```
# look at the varibles for just one person
par(mfrow = c(3,2))
ts.plot(FisherData_fewer[[1]]$energetic, ylab='Energetic', col=4)
ts.plot(FisherData_fewer[[1]]$irritable, ylab="Irritable", col=6)
ts.plot(FisherData_fewer[[1]]$angry, ylab="Angry", col=3)
ts.plot(FisherData_fewer[[1]]$hopeless, ylab="Hopeless", col=5)
ts.plot(FisherData_fewer[[1]]$positive, ylab="Positive", col=7)
```


Energetic
0
60
0
50
100
150
Time
Irritable
0
80
0
50
100
150
Time
Angry
0
80
0
50
100
150
Time
Hopeless
0
80
0
50
100
150
Time
Positive
0
60
0
50
100
150
Time