Intro to RMD and some Functions

Segbehoe, Lawrence Sethor

January 5, 2019

Contents

List of Tables

List of Figures

```
install.packages("caret")
install.packages("knitr")
install.packages("abind")
install.packages("party")
install.packages("mvtnorm")
install.packages("mlbench")
install.packages("modeltools")
install.packages("vcd")
install.packages("kableExtra")
install.packages("grid")
install.packages("lattice")
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(mlbench)
library(vcd)
```

```
## Loading required package: grid
```

```
library(lattice)
```

```
library(party)
```

```
## Loading required package: mvtnorm
## Loading required package: modeltools
## Loading required package: stats4
## Loading required package: strucchange
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package: sandwich
```

```
library(mvtnorm)
library(modeltools)
library(grid)
```

Including Plots

You can also embed plots, for example:

```
plot(pressure)
```

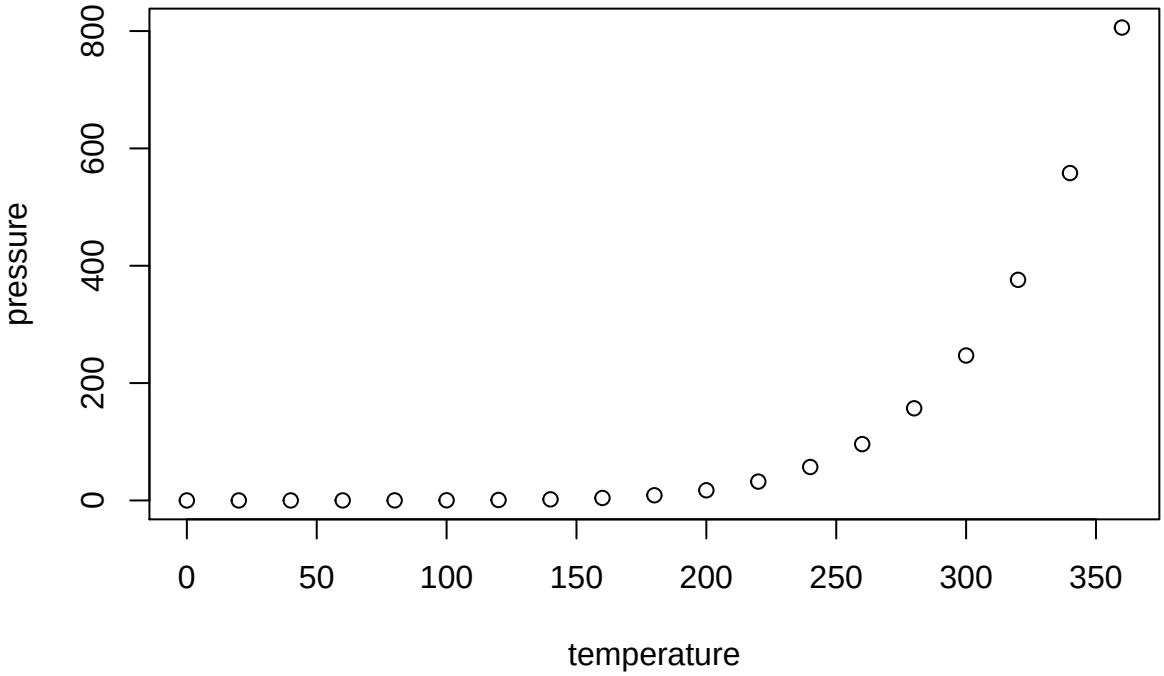


Figure 1: Plot of Pressure

Table 1: Relationship between Temperature and Presure

temperature	pressure
0	0.0002
20	0.0012
40	0.0060
60	0.0300
80	0.0900
100	0.2700
120	0.7500
140	1.8500
160	4.2000
180	8.8000
200	17.3000
220	32.1000
240	57.0000
260	96.0000
280	157.0000
300	247.0000
320	376.0000
340	558.0000
360	806.0000

```
library(knitr)
kable(pressure, caption = "Relationship between Temperature and Presure")
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

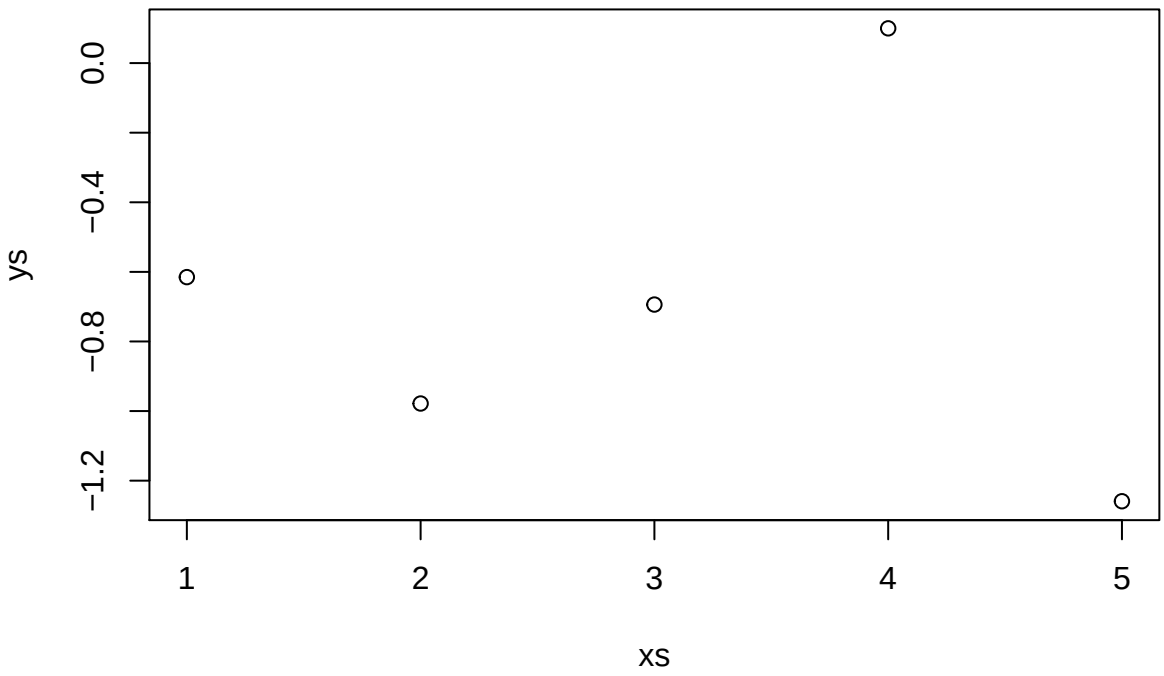


Figure 2: Plot of Random Data.

```
xs = seq(1,5, by = 1)
ys = rnorm(5)
plot(xs, ys)
```

We can see in Figure 2

Table 2: Random Standard Normals

xs	ys
1	-0.6150775
2	-0.9782840
3	-0.6938962
4	0.0998384
5	-1.2591291

Table 3: Boston Housing: A look at the Data

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7

```
kable(cbind(xs, ys), caption = "Random Standard Normals")
```

```
library(mlbench)
data("BostonHousing")
kable(head(BostonHousing), caption = "Boston Housing: A look at the Data")
```

Include Code!

```
# inline code!
x1 = rnorm(10)
x2 = rnorm(10, mean = -5)
xs12 = c(x1, x2)
```

The mean of the X's is -2.8854062.

```
library(HSAUR3)
```

```
## Warning: package 'HSAUR3' was built under R version 3.5.2
```

```
## Loading required package: tools
```

```
data(clouds)
mod1 <- lm(rainfall ~ ., data = clouds)
mod1sum <- summary(mod1)
```

The only significant variable for this model seems to be seedingyes, with the p-value being 0.0128, which is significant at the $\alpha = 0.05$ level.

Play with Piping and Source!

```
source("C:/Users/Lawrence Sethor/Desktop/Spring 2019/Academic/My_Course_Work/STAT602_2019_Modern_Applie
```

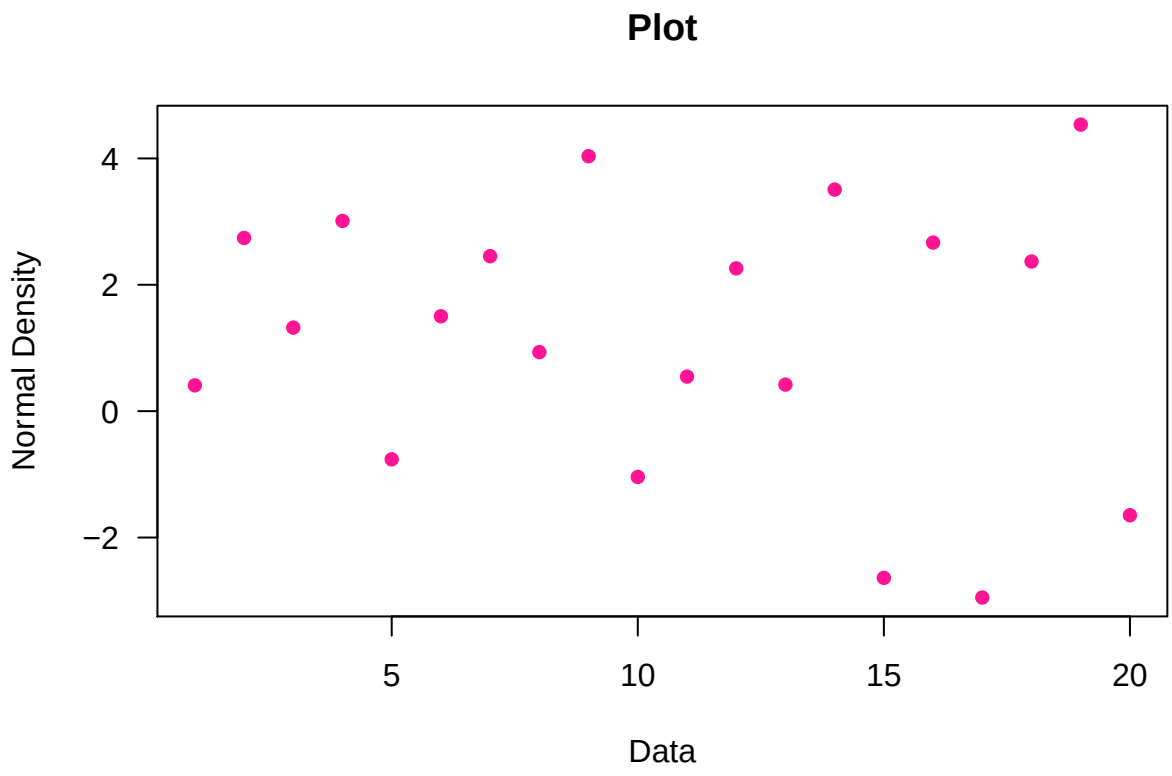
```
##
```

```
## 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
```

```
datx %>% plot(xlab = "Data",
              ylab = "Normal Density",
              main = "Plot",
              col = "Deep Pink",
              pch = 16,
              las = 1)
```




```
sin(cos(datx))
```

```
##  [1]  0.79429275 -0.79612099  0.24377629 -0.83684260  0.66169780
##  [6]  0.06856173 -0.69751970  0.56010132 -0.58666551  0.48400248
## [11]  0.75392323 -0.59396287  0.79124557 -0.80402992 -0.76822664
## [16] -0.77695516 -0.83138801 -0.65624735 -0.17571205 -0.07605172
```

```
datx %>%
  cos %>%
  sin
```

```
##  [1]  0.79429275 -0.79612099  0.24377629 -0.83684260  0.66169780
```

```
##  [6]  0.06856173 -0.69751970  0.56010132 -0.58666551  0.48400248
## [11]  0.75392323 -0.59396287  0.79124557 -0.80402992 -0.76822664
## [16] -0.77695516 -0.83138801 -0.65624735 -0.17571205 -0.07605172
```

```
datx.frame = data.frame("Var1" = rnorm(20,1,3), "Var2" = rnorm(20, 6.5, 171))
datx.frame %>%
  head
```

```
##         Var1        Var2
## 1 -1.5478382    3.678896
## 2  6.4173707   92.973733
## 3 -1.0908044 -171.923049
## 4 -0.5367784  -29.235930
## 5 -1.4434363  -52.496524
## 6 -1.4723936 -111.134501
```

```
datx.frame %>%
  subset(Var1 < mean(Var1)) %>%
  head
```

```
##         Var1        Var2
## 1 -1.5478382    3.678896
## 3 -1.0908044 -171.923049
## 4 -0.5367784  -29.235930
## 5 -1.4434363  -52.496524
## 6 -1.4723936 -111.134501
## 7 -0.2088585 -229.972749
```

```
data("iris")
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
newiris <- aggregate(. ~ Species, data = iris, FUN = mean)
newiris
```

```
##      Species Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     setosa        5.006       3.428        1.462       0.246
## 2 versicolor        5.936       2.770        4.260       1.326
## 3  virginica        6.588       2.974        5.552       2.026
```

```
# playing with paste
paste("List of", 2)
```

```
## [1] "List of 2"
```

```
# playing with assign
assign(paste("List of", 2), 5)

#playing with get
get(paste("List of", 2))
```

```
## [1] 5
```

```r
# all three
for(i in 1:10){
  x <- rnorm(5, i, sd = 2)
  assign(paste("Random Normals with Mean", 5, sep = " "), x)
}

get(paste("Random Normals with Mean", 5, sep = " "))
```

```
## [1] 12.032412 10.868671  8.233844  8.134212  9.745527
```

```r
# playing with sweep
# see also apply, sapply, lapply

r9 <- rnorm(9)
M <- matrix(r9, nrow = 3, ncol = 3)
M2 <- sweep(M, 2, 5, FUN = "-" )
M
```

```
##             [,1]       [,2]       [,3]
## [1,] -0.26773882 -0.1414983 -1.4099644
## [2,] -0.43283151 -0.3687323 -0.5456731
## [3,]  0.09037501  0.6111819 -1.3648383
```

```r
M2
```

```
##           [,1]      [,2]      [,3]
## [1,] -5.267739 -5.141498 -6.409964
## [2,] -5.432832 -5.368732 -5.545673
## [3,] -4.909625 -4.388818 -6.364838
```

```r
startTime <- Sys.time()
Sys.time()
```

```
## [1] "2019-01-05 21:46:53 CST"
```

```r
# playing with systime.time
system.time({for (i in 1:100) {
  mad(runif(1000)) # Median Absolute Deviation

}})
```

```
##    user  system elapsed
##    0.05    0.00    0.07
```

```r
Sys.time()
```

```
## [1] "2019-01-05 21:46:53 CST"
```

```r
stopTime <- Sys.time()
startTime
```

```
## [1] "2019-01-05 21:46:53 CST"
```

```r
stopTime
```

```
## [1] "2019-01-05 21:46:53 CST"
```

```r
# playing with prod
prod(5,6,7)
```

```
## [1] 210
```

```
cars %>% head
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

```
prod(cars)
```

```
## [1] 3.854737e+134
```

```
# there is also cumprod and cumsum


# %in% -- for comparing two vectors
1:10 %in% c(1,3,5,9)
```

```
##  [1]  TRUE FALSE  TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE FALSE
```

```
subset(1:10, 1:10 %in% c(1,3,5,9))
```

```
## [1] 1 3 5 9
```

In a list, use indent=” in the preable ex

1. First Item
2. Second Item
3. Third Item
    a) Subitem
    b) Subitem

    ```
    library(knitr)
    xs <- rnorm(50)
    yx <- rnorm(50)
    xys <- cbind(xs, ys)
    table(summary(xs))
    ```

    ```
    ##
    ##   -1.9903249795745 -0.595116831425641  0.125602969854782
    ##                  1                   1                  1
    ##  0.325216194004721  0.666459769374808   2.00731038429095
    ##                  1                   1                  1
    ```

    c) Subitem
4. Fourth Item

To update R to the currenct version

```
installr::updateR() # updating R.
```