1. R or Python?
   - Large volume of statistics and charting libraries
   - Similar data structure operation
   - R—a popular language and environment to statistically explore datasets
   - Download R 3.3.1 : https://cran.cnr.berkeley.edu
   - After installation, type in "search()" to see the basic packages you have installed for R. R also has a popular IDE(integrated development environment) called RStudio: https://www.rstudio.com/products/rstudio/download/

2. **R as a Calculator:**

```
> 1+2
[1] 3
> 3^2
[1] 9
#Try built-in functions
> exp(2)-log(100)  # Try  "log(10,100)"
[1] 2.783886
# Define a compound function
> sqrt(abs(-2))
[1] 1.414214
```

```
> a<-1
> b=2      # ("=" is the same as "<-")
> (a+b)^2
[1] 9
#Define a function z=f(x,y)
> f<-function(x, y) z<-(y^2-x^2)*pi
>print(f(1,2))
#See what variables you have
>ls()
#Remove a and b from working space
>rm(a,b) # Remove all with "rm(list=ls())"
```

3. **Create Vectors in R:**
   Besides the Vector, R has other data types like **matrix** and **data frame**. Some other useful built-in functions: runif() and rnorm() generating random numbers conforming to uniform and normal distributions; max(), min() and range() get biggest, smallest and the range of vectors.

```
> A<-c(2,3,5,7,11)
> B<-seq(100,108, by=2) # How about "by=3"
> B
[1] 100 102 104 106 108
> c(A,B)
 [1]   2   3   5   7  11 100 102 104 106 108
> A+B
[1] 102 105 109 113 119
```

```
> airports<-c("JFK","LGA","EWR","SFO")
> length(airports)
[1] 4
> airports[4]   #How about airports[-4] ?
[1] "SFO"
> airports[1:3]
[1] "JFK" "LGA" "EWR"
> airports[c(2,4)]
[1] "LGA" "SFO"
```

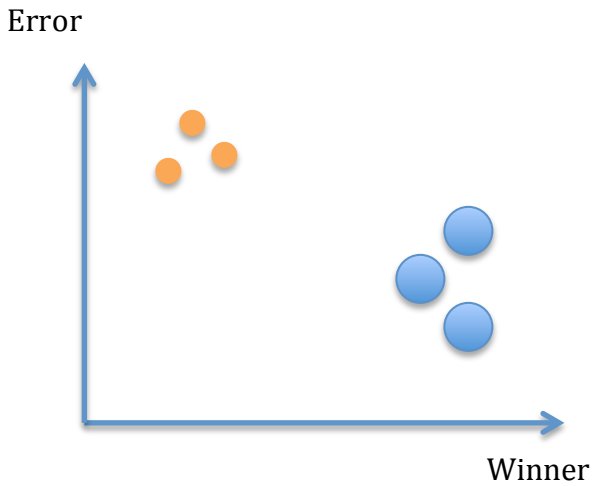
Q1: What are the differences among vector, matrix, data frame, and factor? Your answer should provide a concrete example in codes and annotations.

4. Exploratory Analysis:

Data: similar as assignment1: **sample.csv**

Assumption: high winner, low error, more total point won =: win the match

Plot x-axis (Winner) & y-axis(Error) statistics of each Player in last 10 years Australian open Championship match, Color the winners in Blue and losers in Orange, Size of the dots represent total points won.



year	player	victory	winner	error	total
2009	Rafael Nadal	1	50	41	173
2009	Roger Federer	0	71	64	174
2010	Roger Federer	1	46	42	116
2010	Andy Murray	0	29	36	100
… …	… …	… …			

Why do we need to clean messy data?

- Column headers are values, not variable names.
- Multiple variables are stored in one column.
- Variables are stored in both rows and columns.
- Multiple types of observational units are stored in the same table.
- A single observational unit is stored in multiple tables.

Wrangle data packages: reshape2, dplyr

http://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html

Popular charting packages: ggplot2 & rCharts

http://docs.ggplot2.org/current/

http://rcharts.io/gallery/

```
#plot all winner and error dots
p <- ggplot(sample, aes(winner, error))
p + geom_point()
```

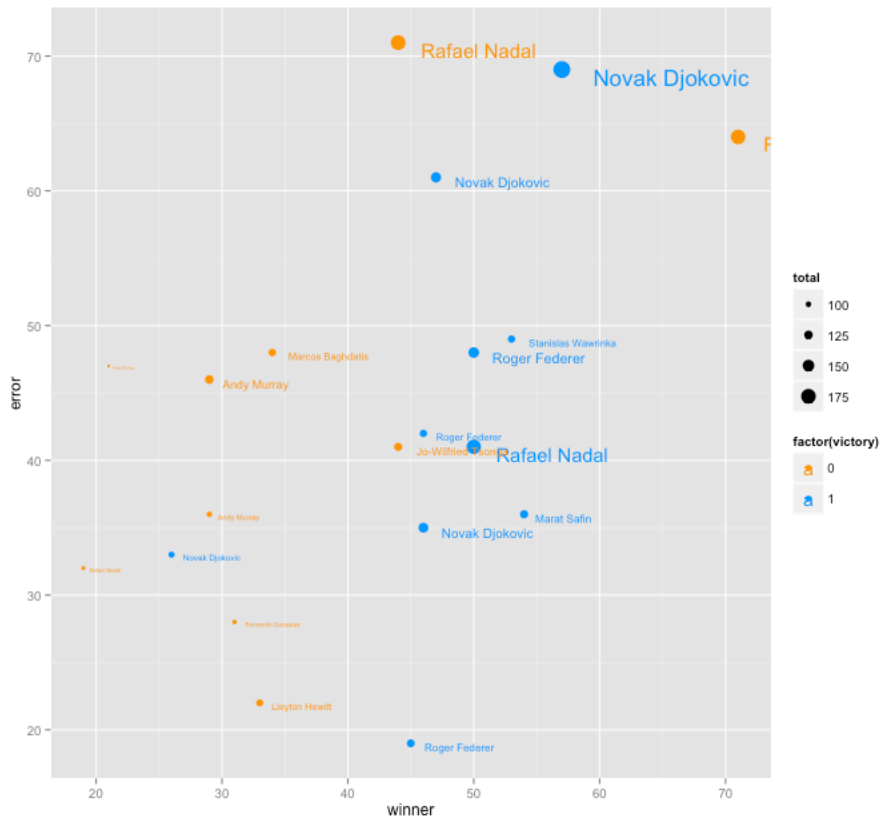
```
# color data points and scale the dot size
```

```
p <- ggplot(sample, aes(winner, error, colour=factor(victory), size=total))
p + geom_point()
```

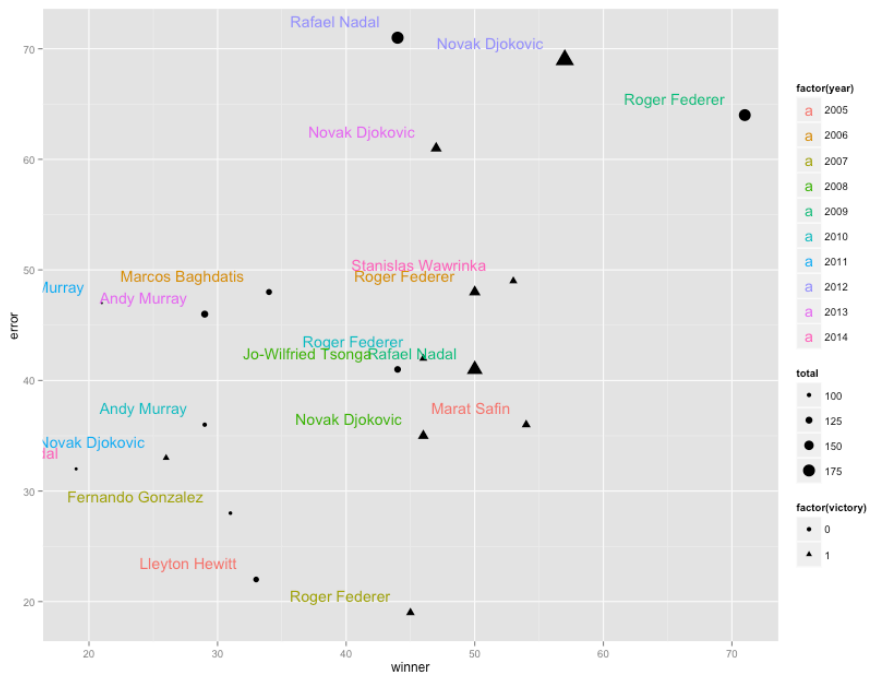
```
# custome color palette
myPalette <- c("#FF9900", "#0099FF")
p + geom_point() + scale_colour_manual(values=myPalette)
```

```
# add label
p <- ggplot(sample, aes(winner, error, colour=factor(victory), size=total, label=player))
p + geom_point() +scale_colour_manual(values=myPalette) + geom_text()
```

```
# jitter the label a bit
p + geom_point() +scale_colour_manual(values=myPalette) + geom_text(hjust=-0.2,
vjust=1)
```



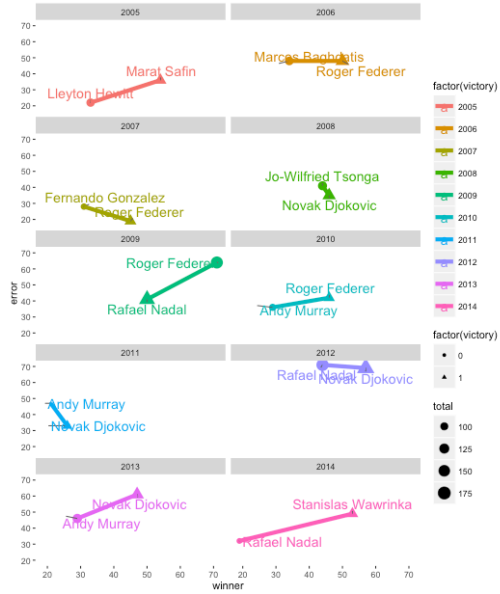

```
# color by match (year), change shape for victory variable
p <- ggplot(sample, aes(winner, error))
p + geom_point(aes(shape=factor(victory), size=total))+
  geom_text(aes(colour=factor(year),label=player), hjust=1.2, vjust=-1)
```




```
# split by match, jitter text
  p + geom_point(aes(shape=factor(victory), size=total))+
  geom_text(aes(colour=factor(year),label=player), position = position_jitter(width=5, height=1.5) ) + facet_grid(~year)
```

Now, your turn:

1. set figure background to white
2. connect each pair by year with a line
3. if jitter the text doesn't work, try *ggrepel* library
4. generate charts not by variable, by columns (hint: *facet_wrap* function)



Q2: Come up with another assumption and vision the outcome may be in a similar comparative small multiples chart?

5. Deeper Analysis (Modeling)
   Data: extract from assignment too: **big3.csv**

```
#only look at the big 3 players
p <- ggplot(big3, aes(factor(year), winner1))
p + geom_boxplot() + facet_grid(~player1)+ geom_jitter(height = 0)
```

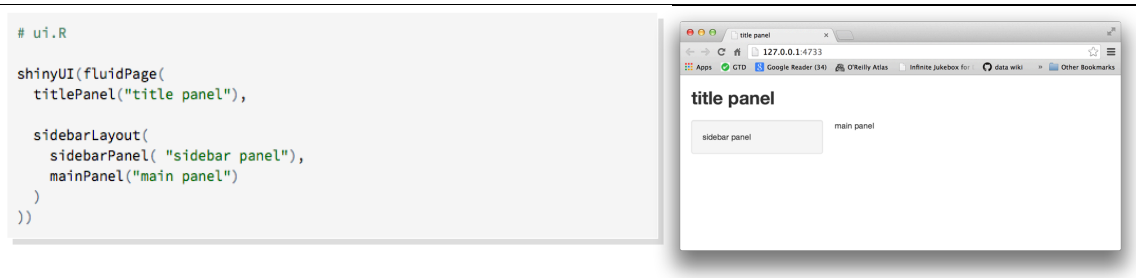
```
# distribution and density
p + geom_violin() +facet_grid(~player1)+ geom_jitter(height = 0)
```

```
# regression line
ggplot(big3, aes(x=total1, y=winner1, size=total1, color=player1)) +
  geom_point()+
  geom_smooth(method=lm, se=F)
```

```
# regression + prediction
ggplot(big3, aes(x=total1, y=winner1, size=total1, color=player1)) +
  geom_point()+
  geom_smooth(method=lm)
```

6. Interactive Visualization with R: Shiny
   - Zero knowledge of HTML/JS/CSS is required, but fully extensible.
   - Extend Interactivity to Reactivity: **Reactive Programming!**
   - To build a simple Shiny application, you need:
     - **a user interface file (ui.R)**—define the client part with inputs and outputs;
     - **a server file (server.R)**—define the task for the server part given inputs and outputs;

   http://shiny.rstudio.com/tutorial/




```
#ui.R
library(shiny)
shinyUI (
  pageWithSidebar
      (
       #Specify Application title
       headerPanel ("Differences Between Champions and Runnerups"),

       #Sidebar with controls to select the variable to plot against match result
       sidebarPanel
           ( selectInput ("variable", "Variable:",
```

```
                    list("Winner" = "winner",
                         "Error" = "error",
                         "Total" = "total")
                ),

        # Add an optional input: to specify whether outliers should be displayed
        checkboxInput ("outliers", "Show outliers", FALSE)
        ),

      #Show the caption and plot of the requested variable against match result as outputs
        mainPanel
         (h3(textOutput("caption")),
              plotOutput("tennisPlot")
        ) #
       ) # pageWithSidebar  end
) #UI end
```

```
#server.R
library(shiny)

shinyServer(function(input, output)
{
  # Construct the formula for the title of the plot
  formulaText <- reactive(
    { paste(input$variable, "against match results") }
    )

  # Return the formula text for printing as a caption
    output$caption <- renderText (
              { formulaText() }
            )

#Generate a boxplot of requested variable against result and include outliers if requested
    output$tennisPlot <- renderPlot(
            {                #Construct a formula for the plot
              boxplot(as.formula(paste(input$variable, "~victory" )),
                    data = sample,
                    outline = input$outliers,
                    col="orange")
            }
    )
}
)
```

```
In R console:
> runApp()
```

* refer to rCharts http://rcharts.io for more interactive visualizations