IT497 Lab #1

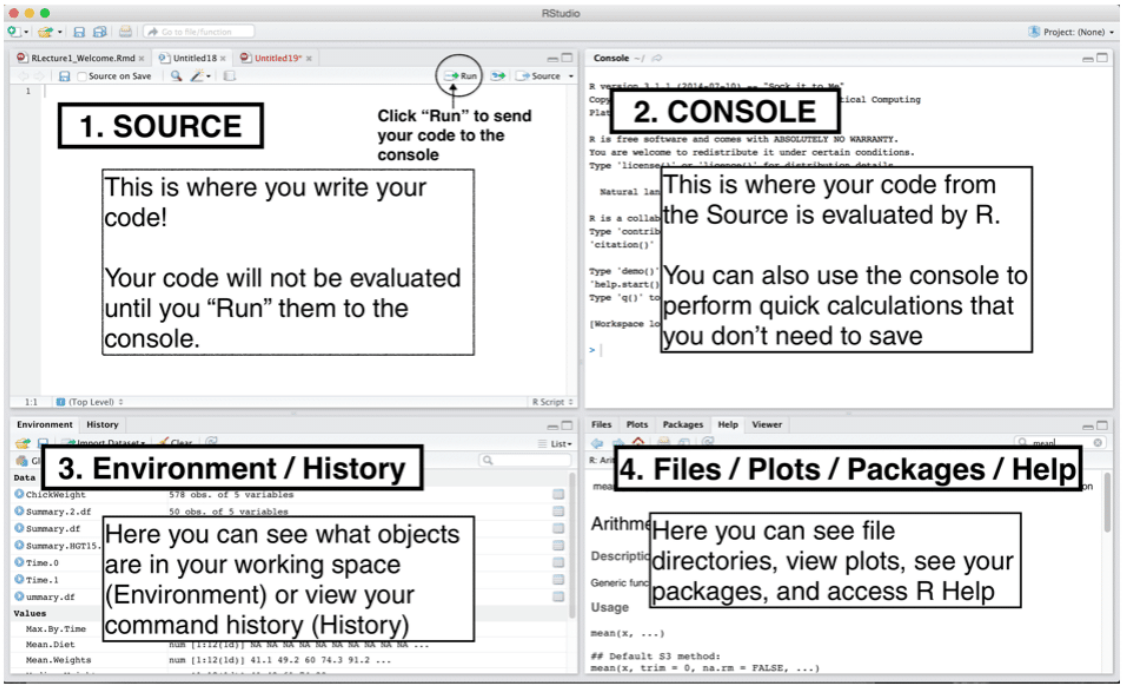
Complete the lab below. Send source code and results via ReggieNet

1. Using your browser, log in to the school's R Studio Server

https://rstudio.it.ilstu.edu/

To login, just use your ISU logon and Password

Here is a quick map of R Studio



2. Under **`File -> New File`** open an R script
3. Anywhere you see R code below, copy it into the R Studio run it.
   `For example, you would copy everything in the box below and run it.`

```
# You can use R like a calculator

x <- 1
y <- 2
z <- x + y
z

str(z)
```

I suggest that you run one line at a time. To do this, **highlight the line of R code and then click on Run.**

Work through each example and then answer the questions at the end. **The lab starts on the next page.**

```
# R is a statistical programming language
# and much more!!!

# We will start with the basics

# You can use R like a calculator

x <- 1
y <- 2
z <- x + y
z

str(z)


# First, let's read in heart attack payment data.

# This data comes from The United States Department of Health and Human
# Services (HHS)

# http://www.healthdata.gov/dataset/heart-attack-payment-national

# I have downloaded a copy of the data to ISU's sever to make it simple

df <- read.table("http://www.itk.ilstu.edu/faculty/jrwolf/hacosts.csv",
header = TRUE, sep = ",")

# In the above df is the name of our data frame. A data frame is used for storing
# data tables.

# It is a list of vectors of equal length.

# We can display the contents of our data frame by typing its name and selection
# run.

df


# The top line of the table is called the header.
# The header contains the column names.

names(df)
head(df, 10)
tail(df)
tail(df, 8)
head(df, 23)
nrow(df)
ncol(df)
str(df)
```

We can access specific data frame columns by name

```
df$State # When accessing columns by name, we use the data frame name
followed by the $ symbol and the column name
df$Cost
mean(df$Cost) # We can perform operations on specific columns
min(df$Cost)
max(df$Cost)
sd(df$Cost)
```

Each horizontal line below the header is called a data
row. Each data element of a row is called a cell.
We can also access columns, rows and even cells by
location

```
df$Cost[2] # This returns the cost cell in the second row
df$Cost[2:5] # This returns the cost cells in rows 2 through 5
df[5,] # This returns all of the rows in column 5
df[1:5,] # This returns all of the rows in columns 1 through 5
df[5,1] # This returns the data in columns 5 row 1
df[5,2] # This returns the data in columns 5 row 2
```

We can find the minimum and maximum values in each column

```
which.min(df$Cost) # this returns the row of the minimum value
which.min(df$Cost) # this returns the row of the maximum value

df[which.min(df$Cost), ] # this returns the minimum value
df[which.max(df$Cost), ] # this returns the maximum value
```

We cn sort the data in our data frame by column

```
df <- df[order(df$Cost), ] # the default is ascending
head(df)

high <- df[1:5,] # This assigns the data from the first 5 rows to a
data frame named high

df <- df[order(-df$Cost), ] # to sort in descending order
head(df)

low <- df[1:5,]  # This assigns the data from the first 5 rows
                 # to a data frame named low
```

```
##################################
# Let's move a bit faster

# read in fitbit data

df <-read.csv("http://www.itk.ilstu.edu/faculty/jrwolf/fitbitstats.csv"
, stringsAsFactors=F)

# Convert the Steps to numeric
df$Steps <- as.numeric(df$Steps)

# Look at the structure of your data frame
str(df)

# Check minimums and maximums

which.min(df$Steps)
which.max(df$Steps)

df[which.min(df$Steps), ]
df[which.max(df$Steps), ]


# Now using the fitbit data answer the following

# 1. Look at the first 6 rows of data
# 2. Look at the last 6 rows of data
# 3. Find the maximum and minimum number of steps taken by FitBit users in our
data,
# 4. Find the total number or rows in the data
# 5. Send me your code and your answers via Reggienet
```