CodePorfolio

Ajay Kumar Upadhyay

7 June 2019

#Week 1  
  
# get working directory  
getwd()

## [1] "E:/Harrisburg University/ANLY 506"

# get all elements in working directory  
ls()

## character(0)

# get all files in the directory  
  
list.files()

## [1] "artofdatascience.pdf" "Code Portfolio.Rmd"   
## [3] "Code\_Portfolio.log" "Code\_Portfolio.Rmd"   
## [5] "CodePortfolio.docx" "CodePortfolio.html"   
## [7] "CodePortfolio.Rmd" "exdata.pdf"   
## [9] "Notes - Onsite Lecture 1.txt" "R studio Book.pdf"

# Week 2  
  
library(readr)  
# Read CSV  
ozone <- read.csv('c:\\users\\ajayu\\downloads\\USEPA.csv')  
  
# Get field names  
names(ozone) <- make.names(names(ozone))  
  
# Check number of rows and columns  
nrow(ozone)

## [1] 66869

ncol(ozone)

## [1] 55

# Week 3  
  
# Create Data Frame  
  
study1.df <- data.frame(id = 1:5,   
 sex = c("m", "m", "f", "f", "m"),   
 score = c(51, 20, 67, 52, 42))  
  
  
# Use aggregate function using mean function  
  
score.by.sex <- aggregate(score ~ sex,   
 FUN = mean,   
 data = study1.df)  
  
# Perform T-test  
study1.htest <- t.test(score ~ sex,   
 data = study1.df)  
  
  
# Week 4  
  
# import tidyverse  
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.3

## -- Attaching packages ------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 2.1.1 v dplyr 0.7.8  
## v tidyr 0.8.2 v stringr 1.4.0  
## v ggplot2 3.1.0 v forcats 0.3.0

## Warning: package 'ggplot2' was built under R version 3.5.3

## Warning: package 'tibble' was built under R version 3.5.3

## Warning: package 'dplyr' was built under R version 3.5.2

## Warning: package 'stringr' was built under R version 3.5.3

## -- Conflicts ---------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

# Week 5  
# Using Matrices and Determinants  
  
x <- 1:5  
y <- 6:10  
z <- 11:15  
  
  
#Using R bind and Cbind, Creating maxtrix  
cbind(x, y, z)

## x y z  
## [1,] 1 6 11  
## [2,] 2 7 12  
## [3,] 3 8 13  
## [4,] 4 9 14  
## [5,] 5 10 15

rbind(x, y, z)

## [,1] [,2] [,3] [,4] [,5]  
## x 1 2 3 4 5  
## y 6 7 8 9 10  
## z 11 12 13 14 15

cbind(c(1, 2, 3, 4, 5), c("a", "b", "c", "d", "e"))

## [,1] [,2]  
## [1,] "1" "a"   
## [2,] "2" "b"   
## [3,] "3" "c"   
## [4,] "4" "d"   
## [5,] "5" "e"

matrix(data = 1:10, nrow = 5, ncol = 2)

## [,1] [,2]  
## [1,] 1 6  
## [2,] 2 7  
## [3,] 3 8  
## [4,] 4 9  
## [5,] 5 10

survey <- data.frame("index" = c(1, 2, 3, 4, 5), "sex" = c("m", "m", "m", "f","f"), "age" = c(99, 46, 23, 54, 23))  
str(survey)

## 'data.frame': 5 obs. of 3 variables:  
## $ index: num 1 2 3 4 5  
## $ sex : Factor w/ 2 levels "f","m": 2 2 2 1 1  
## $ age : num 99 46 23 54 23

# head() shows the first few rows  
head(ChickWeight)

## Grouped Data: weight ~ Time | Chick  
## weight Time Chick Diet  
## 1 42 0 1 1  
## 2 51 2 1 1  
## 3 59 4 1 1  
## 4 64 6 1 1  
## 5 76 8 1 1  
## 6 93 10 1 1

# tail() shows he last few rows  
tail(ChickWeight)

## Grouped Data: weight ~ Time | Chick  
## weight Time Chick Diet  
## 573 155 12 50 4  
## 574 175 14 50 4  
## 575 205 16 50 4  
## 576 234 18 50 4  
## 577 264 20 50 4  
## 578 264 21 50 4

# using summary function  
summary(ToothGrowth)

## len supp dose   
## Min. : 4.20 OJ:30 Min. :0.500   
## 1st Qu.:13.07 VC:30 1st Qu.:0.500   
## Median :19.25 Median :1.000   
## Mean :18.81 Mean :1.167   
## 3rd Qu.:25.27 3rd Qu.:2.000   
## Max. :33.90 Max. :2.000

names(ToothGrowth)

## [1] "len" "supp" "dose"

# calculating mean of a column  
mean(ToothGrowth$len)

## [1] 18.81333

table(ToothGrowth$supp)

##   
## OJ VC   
## 30 30

# adding new column to existing dataframe  
  
survey$sex <- c("m", "m", "f", "f", "m")  
  
# change column names  
  
df <- ToothGrowth  
  
names(df)[1] <- "a"  
  
# Return row 1  
df[1, ]

## a supp dose  
## 1 4.2 VC 0.5

# Rows 1:5 and column 2  
df[1:5, 2]

## [1] VC VC VC VC VC  
## Levels: OJ VC

# using subset functions  
  
subset(x = ToothGrowth,subset = len < 10 & supp == "OJ")

## len supp dose  
## 34 9.7 OJ 0.5  
## 37 8.2 OJ 0.5  
## 38 9.4 OJ 0.5  
## 40 9.7 OJ 0.5

# Week 6  
# load library nycflights  
library(nycflights13)

## Warning: package 'nycflights13' was built under R version 3.5.3

# using filter command to separate dataset  
  
filter(flights, month == 1, day == 1)

## # A tibble: 842 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int>  
## 1 2013 1 1 517 515 2 830  
## 2 2013 1 1 533 529 4 850  
## 3 2013 1 1 542 540 2 923  
## 4 2013 1 1 544 545 -1 1004  
## 5 2013 1 1 554 600 -6 812  
## 6 2013 1 1 554 558 -4 740  
## 7 2013 1 1 555 600 -5 913  
## 8 2013 1 1 557 600 -3 709  
## 9 2013 1 1 557 600 -3 838  
## 10 2013 1 1 558 600 -2 753  
## # ... with 832 more rows, and 12 more variables: sched\_arr\_time <int>,  
## # arr\_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,  
## # origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
## # minute <dbl>, time\_hour <dttm>

# Using logical operators  
  
filter(flights, month == 11 | month == 12)

## # A tibble: 55,403 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int>  
## 1 2013 11 1 5 2359 6 352  
## 2 2013 11 1 35 2250 105 123  
## 3 2013 11 1 455 500 -5 641  
## 4 2013 11 1 539 545 -6 856  
## 5 2013 11 1 542 545 -3 831  
## 6 2013 11 1 549 600 -11 912  
## 7 2013 11 1 550 600 -10 705  
## 8 2013 11 1 554 600 -6 659  
## 9 2013 11 1 554 600 -6 826  
## 10 2013 11 1 554 600 -6 749  
## # ... with 55,393 more rows, and 12 more variables: sched\_arr\_time <int>,  
## # arr\_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,  
## # origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
## # minute <dbl>, time\_hour <dttm>

# Creating tibble and filtering it  
  
df <- tibble(x = c(1, NA, 3))  
filter(df, x > 1)

## # A tibble: 1 x 1  
## x  
## <dbl>  
## 1 3

# using arrange function  
  
arrange(flights, year, month, day)

## # A tibble: 336,776 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int>  
## 1 2013 1 1 517 515 2 830  
## 2 2013 1 1 533 529 4 850  
## 3 2013 1 1 542 540 2 923  
## 4 2013 1 1 544 545 -1 1004  
## 5 2013 1 1 554 600 -6 812  
## 6 2013 1 1 554 558 -4 740  
## 7 2013 1 1 555 600 -5 913  
## 8 2013 1 1 557 600 -3 709  
## 9 2013 1 1 557 600 -3 838  
## 10 2013 1 1 558 600 -2 753  
## # ... with 336,766 more rows, and 12 more variables: sched\_arr\_time <int>,  
## # arr\_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,  
## # origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
## # minute <dbl>, time\_hour <dttm>

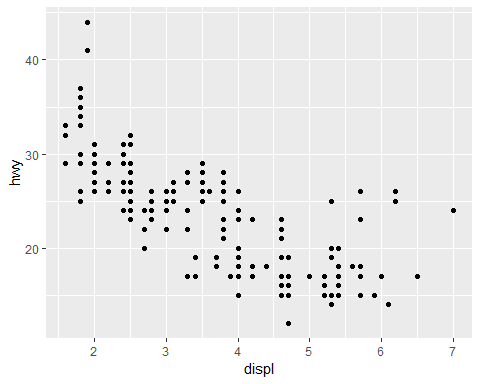
# using mutate function  
  
flights\_sml <- select(flights, year:day, ends\_with("delay"), distance, air\_time)  
mutate(flights\_sml,gain = dep\_delay - arr\_delay,speed = distance / air\_time \* 60)

## # A tibble: 336,776 x 9  
## year month day dep\_delay arr\_delay distance air\_time gain speed  
## <int> <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2013 1 1 2 11 1400 227 -9 370.  
## 2 2013 1 1 4 20 1416 227 -16 374.  
## 3 2013 1 1 2 33 1089 160 -31 408.  
## 4 2013 1 1 -1 -18 1576 183 17 517.  
## 5 2013 1 1 -6 -25 762 116 19 394.  
## 6 2013 1 1 -4 12 719 150 -16 288.  
## 7 2013 1 1 -5 19 1065 158 -24 404.  
## 8 2013 1 1 -3 -14 229 53 11 259.  
## 9 2013 1 1 -3 -8 944 140 5 405.  
## 10 2013 1 1 -2 8 733 138 -10 319.  
## # ... with 336,766 more rows

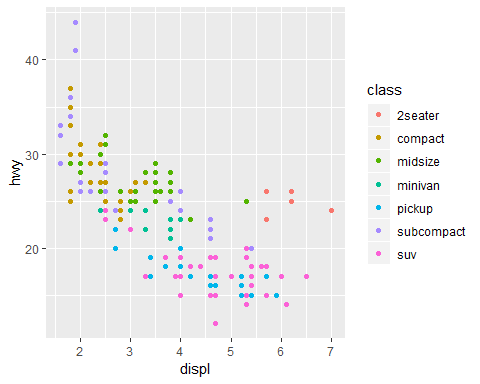
mutate(flights\_sml, gain = dep\_delay - arr\_delay, hours = air\_time / 60, gain\_per\_hour = gain / hours)

## # A tibble: 336,776 x 10  
## year month day dep\_delay arr\_delay distance air\_time gain hours  
## <int> <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2013 1 1 2 11 1400 227 -9 3.78   
## 2 2013 1 1 4 20 1416 227 -16 3.78   
## 3 2013 1 1 2 33 1089 160 -31 2.67   
## 4 2013 1 1 -1 -18 1576 183 17 3.05   
## 5 2013 1 1 -6 -25 762 116 19 1.93   
## 6 2013 1 1 -4 12 719 150 -16 2.5   
## 7 2013 1 1 -5 19 1065 158 -24 2.63   
## 8 2013 1 1 -3 -14 229 53 11 0.883  
## 9 2013 1 1 -3 -8 944 140 5 2.33   
## 10 2013 1 1 -2 8 733 138 -10 2.3   
## # ... with 336,766 more rows, and 1 more variable: gain\_per\_hour <dbl>

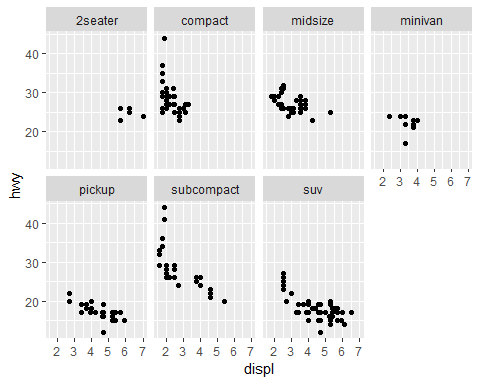
# Week 7  
  
# visualizations  
  
  
# scatter plot  
ggplot(data = mpg) + geom\_point(mapping = aes(x = displ, y = hwy))



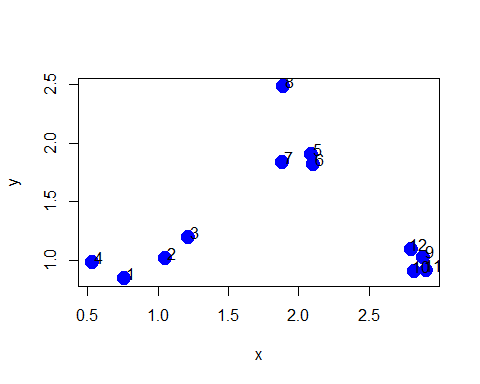
# scatter plot with colors  
ggplot(data = mpg) + geom\_point(mapping = aes(x = displ, y = hwy, color = class))



# using facet  
ggplot(data = mpg) + geom\_point(mapping = aes(x = displ, y = hwy)) + facet\_wrap(~ class, nrow = 2)



# Week 12  
  
# K means clusturing algo  
  
set.seed(1234)  
x <- rnorm(12, mean = rep(1:3, each = 4), sd = 0.2)  
y <- rnorm(12, mean = rep(c(1, 2, 1), each = 4), sd = 0.2)  
plot(x, y, col = "blue", pch = 19, cex = 2)  
text(x + 0.05, y + 0.05, labels = as.character(1:12))



dataFrame <- data.frame(x, y)  
kmeansObj <- kmeans(dataFrame, centers = 3)

## 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:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



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