#Basic practice

> print(cat)

function (..., file = "", sep = " ", fill = FALSE, labels = NULL,

append = FALSE)

{

if (is.character(file))

if (file == "")

file <- stdout()

else if (substring(file, 1L, 1L) == "|") {

file <- pipe(substring(file, 2L), "w")

on.exit(close(file))

}

else {

file <- file(file, ifelse(append, "a", "w"))

on.exit(close(file))

}

.Internal(cat(list(...), file, sep, fill, labels, append))

}

<environment: namespace:base>

#Square

Square <- function(x) {

return(x^2)

}

print(Square(4))

print(Square(x=4)) # same thing

[1] 16

[1] 16

#Lists

sum(0:9)

append(LETTERS[1:13],letters[14:26])

c(1,6,4,9)\*2

something <- c(1,4,letters[2]) # indices start at one, you get (1,4,"b")

length(something)

#seq function

seq(from=1,to=4,by=.6)

#Reading user input

readinteger <- function()

{

n <- readline(prompt="Enter an integer: ")

return(as.integer(n))

}

print(readinteger())

#Rounding

> round(22.5,0) # rounds to even number

[1] 22

> round(3.14,1)

[1] 3.1

#Reading data

tbl <- read.table(file.choose(),header=TRUE,sep=",")

population <- tbl["POPESTIMATE2009"]

print(summary(population[-1:-5,]))

#Indexing data frames

tbl["POPESTIMATE2009"]:

#Fetching specific rows and columns

population[1:5,] # first the rows, then the columns

#Summary function

summary(1:10)

#Filtering Data

tbl <- read.table(file.choose(),header=TRUE,sep=',')

population <- tbl[c("NAME","POPESTIMATE2009","NPOPCHG\_2009")]

smallest.state.pop <- min(population$POPESTIMATE2009)

print(population[population$POPESTIMATE2009==smallest.state.pop,])

#Finding data sources

mtcars

print(head(mtcars))

#Matrix

print(matrix(runif(6\*3), nrow=6, ncol=3))

#working directory

getwd()

[1] "/Users/yuktibhatia"

#Read file

read.table(file = "mydata.txt", sep, header)

#changing working directory

setwd(dir = "/Users/yuktibhatia/Dropbox/DataVis"

#Remeove all objects from workspace

rm(list = ls())

#data frame

data.frame("day" = c("mon", "fri", "sun"),

"time" = c(8, 12, 16),

"passengers" = c("16", "24", "56"))

#mean

mean ("Sample Duration")

#for US EPA AirData - Sample Duration coloumn

#check the number of rows and columns

nrow(ozone)

ncol(ozone)

#Running str() - gives reasonable output for any R object

str(ozone)

#Find Top and Bottom data

head(ozone[, c(8:9, 10)])

tail(ozone[, c(8:9, 10)])

#Practiced "filter" function from notes

filter(ozone, State.Code == "36"

& County.Code == "033"

& Date.Local == "2014-09-30") %>%

select(Date.Local, Time.Local,

Sample.Measurement) %>%

as.data.frame

Date.Local Time.Local Sample.Measurement

#Ranking for ozone data

head(ranking, 10)

#sort (sorting data by height)

pirates <- pirates[order(pirates$height),]

#looking at few rows and coloumns

ozone[1:5, 1:4]

# Sort the pirates dataframe by sex and then height

pirates <- pirates[order(pirates$sex, pirates$height),]

#Combing data (practice merge function)

combined.survey <- merge(x = survey.a,

y = survey.b,

by = "coloumn1",

all = TRUE)

# Print results

combined.survey

# Create a vector.

x <- c(12,7,3,4.2,18,2,54,-21,8,-5)

# Find Mean.

result.mean <- mean(x)

print(result.mean)

[1] 8.22

#long\_name function

long\_name <- function(a = "a\_long\_argument" ,

b = "another\_argument",

c = "another\_long\_argument)

#reading a csv file (practice from book)

read\_csv("a,b,c

1,2,3

4,5,6")

#parsing a vector (wil take a character vector and return a more specialised vector like a logical, integer, or date)

str(parse\_integer(c("1", "2", "3")))

parse\_integer(c("1", "231", ".", "456"), na = ".")

parse\_double("2.54")

#computation

table1 %>%

mutate(rate = cases / population \* 100)

# A tibble: 6 x 5

country year cases population rate

<chr> <int> <int> <int> <dbl>

1 Afghanistan 1999 745 19987071 0.00373

2 Afghanistan 2000 2666 20595360 0.0129

3 Brazil 1999 37737 172006362 0.0219

4 Brazil 2000 80488 174504898 0.0461

5 China 1999 212258 1272915272 0.0167

6 China 2000 213766 1280428583 0.0167

#plotting mpg via ggplot

ggplot(data = mpg) +

geom\_point(mapping = aes(x = displ, y = hwy))

ggplot(data = mpg) +

geom\_point(mapping = aes(x = displ, y = hwy, color = class))

#Quiz

library (yarrr)

#matrix

matrix(1:9,nrow=3, ncol = 3)

#vectorization

a <- c(2,3,4)

b <- c(20,30,40)

(a+b) / 10

[1] 2.2 3.3 4.4

# tidy data - Converts data to tbl class. tbl’s are easier to examine than data frames.

dplyr::glimpse(iris)

# Information dense summary of tbl data.

View(iris)

%>%

iris %>%

group\_by(Species) %>%

summarise(avg = mean(Sepal.Width)) %>%

arrange(avg)

#Passes object on lef hand side as first argument of function on righthand side. (with group by, summarise and average)

str(iris)

#Get a summary of an object’s structure.

class(iris)

#Find the class an object belongs to.

install.packages(‘dplyr’)

#Download and install a package from CRAN.

library(dplyr)

#Load the package into the session, making all its functions available to use.

data(iris)

#Load a build-in dataset into the environment.

#Reading & writing data

df <- read.table(‘file.txt’)

write.table(df, ‘file.txt’)

#Remove x from the dataset

rm(x)

#Remove all variables from the environment

rm(list = ls())

#functions

> x <- c(1,2,3,4,5,6)

> var(x)

[1] 3.5

> sd(x)

[1] 1.870829

> log(x)

[1] 0.0000000 0.6931472 1.0986123 1.3862944 1.6094379 1.7917595

> max(x)

[1] 6

> round(x, 2)

[1] 1 2 3 4 5 6

> sum(x)

[1] 21

> median(x)

[1] 3.5

> quantile(x)

0% 25% 50% 75% 100%

1.00 2.25 3.50 4.75 6.00

> rank(x)

[1] 1 2 3 4 5 6

#boxplot

boxplot(pollution$pm25, col = "blue")

abline(h = 12)

#histogram

hist(pollution$pm25, col = "green")

#barplot

table(pollution$region) %>% barplot(col = "green")

#multiple boxplot

boxplot(pm25 ~ region, data = pollution, col = "green")

#multiple histogram

par(mfrow = c(2, 1), mar = c(4, 4, 2, 1))

> hist(subset(pollution, region == "east")$pm25, col = "green")

> hist(subset(pollution, region == "west")$pm25, col = "green")

#scatterplot

with(pollution, plot(latitude, pm25))

> abline(h = 12, lwd = 2, lty = 2)

#Multiple Scatterplots

> par(mfrow = c(1, 2), mar = c(5, 4, 2, 1))

> with(subset(pollution, region == "west"), plot(latitude, pm25, main = "West"))

> with(subset(pollution, region == "east"), plot(latitude, pm25, main = "East")

#base plotting system

data(airquality)

> with(airquality, {

+ plot(Temp, Ozone)

+ lines(loess.smooth(Temp, Ozone))

+ })

## Add annotation

> title("Speed vs. Stopping distance")