**###################**

**### Challenge 1 ###**

**###################**

**# Build your own project directory structure for this workshop. For our purposes today you will require at least a `data`, `scripts`, and `images` folder**

NO ANSWER

**###################**

**### Challenge 2 ###**

**###################**

**# Add an `R Project` to the existing directory you created in Challenge 1.**

NO ANSWER

**###################**

**### Challenge 3 ###**

**###################**

**# Which of the following are valid variable names?**

**# min\_height** YES

**# max.height** YES

**# \_age** NO

**# .mass** HIDDEN

**# MaxLength** YES

**# min-length** NO

**# 2widths** NO

**# celsius2kelvin** YES

**###################**

**### Challenge 4 ###**

**###################**

**# What will be the value of each variable after each statement in the following program?**

**# mass <- 47.5**

**# age <- 122**

**# mass <- mass \* 2.3** 109.25

**# age <- age – 20** 102

**###################**

**### Challenge 5 ###**

**###################**

**# Run the code from the previous challenge, and write a command to compare mass to age. Is mass larger than age?**

Mass > age

**###################**

**### Challenge 6 ###**

**###################**

**# Clean up your working environment by deleting the mass and age variables.**

Rm(list = c(mass, age))

**###################**

**### Challenge 7 ###**

**###################**

**# Install the following package: `ggplot2`**

Install.packages(“ggplot2”)

**###################**

**### Challenge 8 ###**

**###################**

**# Look at the help for the `paste` function. You'll need to use this later. What is the difference between the `sep` and `collapse` arguments?**

paste(letters, collapse = " ")

paste(letters, sep = " ")

paste(letters, LETTERS, sep = " ")

paste(letters, LETTERS, sep = " ", collapse = "\_")

**###################**

**### Challenge 9 ###**

**###################**

**# Given the following lines of code:**

**# x <- 1:5**

**# names(x) <- letters[1:5]**

**# x**

**# Find at least five different commands to come up with the following subset:**

**# b c d**

**# 2 3 4**

**# Fictional bonus points for anyone who figures out the %in% operator!**

x[2:4]

x[c(2,3,4)]

x[-c(1,5)]

x[c(-1,-5)]

x[c(“b”,”c”,”d”)]

x[x %in% c(“b”, “c”, “d”)

x[names(x) %in% c("b","c","d")]

x[-which(names(x) %in% c("a","e"))]

**####################**

**### Challenge 10 ###**

**####################**

**# Using the following code:**

**# challenge\_list <- list(words = c("alpha", "beta", "gamma"),**

**# numbers = 1:10,**

**# letter = letters)**

**# challenge\_list**

**# Extract the following things:**

**# - The word "gamma"**

**# - The letters "a", "e", "i", "o", and "u"**

**# - The numbers less than or equal to 3**

**# More fictional bonus points if you use a different methods!**

challenge\_list$words[3]

challenge\_list[[1]][3]

challenge\_list[[3]][c(1, 5, 9, 15, 21)]

challenge\_list$letter[c(1, 5, 9, 15, 21)]

challenge\_list$letter[challenge\_list$letter %in% c("a","e","i","o","u")]

challenge\_list$numbers[challenge\_list$numbers <= 3]

challenge\_list$numbers[1:3]

**####################**

**### Challenge 11 ###**

**####################**

**# Look thorugh the help file for the read.csv() command to find an argument to stop character data from being loaded as factors. Hint: Characters are sometimes referred to as strings.**

**# Reload the cats data frame from file without factors**

**# Add the new row of Garfield data to the data frame**

Cats <- read.csv(“data/feline.csv”, stringsAsFactors = FALSE)

Cats <- rbind(cats, Garfield)

**####################**

**### Challenge 12 ###**

**####################**

**# Create a list of length two containing a character vector for each of the sections in this part of the workshop:**

**# - Data types**

**# - Data structures**

**# Populate each character vector with the names of the data types and data structures we've seen so far.**

Data\_types <- c(“Logical”, “Numeric”, “Integer”,”Complex”,”Character”)

Data\_structures <- c(“Vector”,”List”,”Data Frame”)

Challenge\_list <- list(data\_types, data\_structures)

**####################**

**### Challenge 13 ###**

**####################**

**# Create a new data frame called titanic\_35 that includes only the passengers aged 35 or under.**

Titanic\_35 <- titanic[titanic$Age <= 35, ]

**####################**

**### Challenge 14 ###**

**####################**

**# Create a new data frame that includes only the passengers aged < 18 or >= 50.**

Titanic\_18\_50 <- titanic[titanic$Age < 18 | titanic$Age >= 50, ]

**####################**

**### Challenge 15 ###**

**####################**

**# Create a new data frame that includes only female, third class passengers.**

Titanic\_female\_third <- titanic[titanic$Sex == “female” & titanic$Pclass == 3, ]

**####################**

**### Challenge 16 ###**

**####################**

**# Create a new data frame that includes only male passengers aged either 18, 23, 31, or 46.**

**# Hint: Rememberthe %in% operator!**

Titanic\_male <- titanic[titanic$Sex == “male” & titanic$Age %in% c(18, 23, 31, 46), ]

**####################**

**### Challenge 17 ###**

**####################**

**# Switch the order of the point and line layers from the previous example. What happened?**

ggplot(data = titanic\_clean,

aes(x = Age,

y = Fare,

col = as.factor(Pclass))) +

geom\_line() +

geom\_point(col="black")