Data Wrangling in R

Dr Josh A Hodge

require(tidyr)

## Loading required package: tidyr

require(dplyr)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Brief

These exercises have been designed for you to practice the data wrangling functions introduced in the lecture. The exercises may require you to integrate other functions, such as min(), max() and abs(), so do have a google with these or use the help function. Otherwise, ask a demonstrator or myself for help. xenomorphs.csv

The practical is split into two sections. These sections increase in difficulty.

## Section One

You have been given a dataset called “new\_xenomorphs.csv” from the Weyland that has recorded the weights of 188 xenomorphs captured on the moon Acheron (39 light years away from Earth). The xenomorphs exhibit instar stages from egg, facehugger to fully grown xenomorph. The original dataset has been recorded in wide format with every column representing a different instar and each row a different individual. Use this data to tackle the following exercises.

1. Set your working directory

setwd("~/Library/Mobile Documents/com~apple~CloudDocs/Imperial/Teaching/PG/Biological Computing in R/21:22/3. Thursday/Practical/")

1. Load in the dataframe “new\_xenomorphs.csv”.

data <- read.csv("new\_xenomorphs.csv", row.names = 1)

1. Print out the structure of the data frame.

str(data)

## 'data.frame': 302 obs. of 8 variables:  
## $ ID : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ BodySize: num 2.1 9.6 6 5.9 6.1 ...  
## $ CO2ppm : num 2.875 2.201 0.965 3.82 6.106 ...  
## $ X1 : num 0.1865 0.204 0.0895 0.3511 0.3629 ...  
## $ X2 : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X3 : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X4 : num NA NA NA NA NA NA NA NA NA NA ...  
## $ X5 : num NA NA NA NA NA NA NA NA NA NA ...

1. Gather the Instars into a new column of “Instar” and their metabolic rates to a new column of “Mrate”. HINT: you’ll need to figure out what to do with the NA’s (Use: ?pivot\_longer). BONUS: can you code it so the Instars are recorded as 1 and not X1.

gathered\_data<-pivot\_longer(data, names\_to = "Instar", values\_to = "Mrate", cols=X1:X5, values\_drop\_na = TRUE)  
##Bonus Point   
gathered\_data<-pivot\_longer(data, names\_to = "Instar", values\_to = "Mrate", cols=X1:X5, values\_drop\_na = TRUE, names\_prefix = "X")

1. Reverse what you did in Exercise 1 by giving to the dataset its initial form.

spread\_data <- pivot\_wider(gathered\_data,names\_from=Instar, values\_from=Mrate)

1. Using the gathered data from exercise 4, create a new column called Weight, which is calculated as 0.2+8.4\*Mrate.

data2<-mutate(gathered\_data, Weight = 0.2+(8.4\*Mrate))

1. Split the xenomorph data used in exercise 6 by Instar, then calculate the mean and standard deviation of the weight and metabolic rate.

groups<-group\_by(data2,Instar)   
summary\_stats<- summarise(groups, mean\_weight = mean(Weight),sd\_weight = sd(Weight),mean\_rate = mean(Mrate),sd\_rate=sd(Mrate))

1. Save the summary statistics of the xenomorph data as csv file to your working directory.

write.csv(summary\_stats, "summary\_stats\_xenomorphs.csv")

## Section Two

You’ll have some similar questions to section one, but these questions will require you to use the pipe operator (%>%) to help combine functions into a single line of code.

1. Load the iris dataset. (R has some in-built datasets that can be really useful to practice your R skills. Research how to load in these datasets).

data(iris)

1. Select all the columns in the iris dataset that measure petal traits and filter the rows for lengths that are less than or equal to 2cm and widths less than 1cm.

petals<-iris%>%select(starts\_with("Petal"))%>%filter(Petal.Length<=2, Petal.Width<1)

1. Select three columns from iris, arrange the rows by Sepal.Length, then arrange the rows by Sepal.Width.

arranged<- iris%>%select(Species, Sepal.Length, Sepal.Width)%>%arrange(Sepal.Length, Sepal.Width)

1. Calculate the minimium and maximium of sepal and petal length for each species.

min\_max<- iris%>%group\_by(Species)%>%summarise(min\_sepal=min(Sepal.Length), max\_sepal=max(Sepal.Length), min\_petal=min(Petal.Length), max\_petal=max(Petal.Length))

1. Filter the iris dataset for virginica and setosa species and arrange them from the largest Petal length to smallest.

set\_vir<-iris%>%filter(Species%in%c("virginica", "setosa"))%>%arrange(desc(Petal.Length))

1. Using the original iris data, select only the sepal width and species column, z-standardise (using the scale function) the data using the mutate function and filter the measurements for those that are two standard deviations away. Finally, using group\_by and summarise, count how many measurements there are per species.

counts<- iris%>%select(Sepal.Width, Species)%>%mutate(Standardise= scale(Sepal.Width))%>%filter(Standardise<=2, Standardise>=-2)%>%group\_by(Species)%>%summarise(Count=n())