**Useful functions in R**

**Installing packages**

To install a package, use function install.packages(“*packageName*”), where *packageName is the name of the package*

**Creating variable names for directories**

To view the current directory, use function: getwd()

Example output: "C:/Users/Documents"

To create a variable called *directoryName* for the current directory which stores "C:/Users/Documents",

*directoryName*  <- getwd()

The variable *directoryName* can be used when importing and exporting csv files

**Opening csv files**

Uses the function: read\_csv(“*filename*”) where *filename* is the name of the file to be read. The quotation marks are necessary.

This function is part of the tidyverse package. For new R users, install tidyverse with

install.packages(‘tidyverse’)

Open tidyverse package with:

library(tidyverse)

To read files from a specific directory (folder where data is stored), use the following:

read\_csv(paste0(*directoryName*,*”filename”*)) , where *directoryName* is the variable you created for the directory

You will often want to read the contents of the CSV file into a variable, which you can do as follows:

*varName* <- read\_csv(“*filename*”), which creates a variable named *varName* and reads into it the contents of the file.

**Examining variables**

print(*varName*) is a useful way to inspect the contents of variables you have created to see if the file was read correctly.

**Convert data.frame (Ex: df) to matrix**

When you read a CSV file into R Studio, the variable first created has a generic type called a “data frame”. You will generally want to convert this variable to a matrix, which supports doing needed mathematical manipulations. You can convert any variable to a matrix type as follows:

*varName* <- as.matrix(*varName*)

You can check what type any variable is as follows: class(*varName)*.

**Getting column vectors**

df$column\_name

**Matrix multiplication function**

%\*%

e.g. you can perform a matrix multiplication and assign the result to a variable as follows:

*resultVar* <- *matrixVar* %\*% *vectorVar*

**Writing output to a CSV file**

write.csv(*varName*, paste0(*directoryName*, *“exportName.csv”*), where *varName* is the name of the variable with data you want to save, *directoryName* is the variable with the location of where you want to store the data, and *exportName* is the name of the csv file.

**Opening shapefiles**

Read shapefile called *shapeFile* with function

st\_read(paste0(*directoryName, “shapeFile.shp”*)), where *directoryName* is the directory where the shapefile is stored

Note: there are four extensions (.shp, .prj, .dbf, .shx) for the provided shapefile. All 4 files must be in the directory where you read *shapeFile.shp*

**Plotting output**

Run *plot\_concentrations.R* to load plot function *plot\_PM25*. You can run the script by selecting the contents and clicking ‘Run’.

The arguments in the function *plot\_PM25* are

*shapefile*: variable name of shapefile

*df\_name*: variable name of data frame that contains concentrations

*var\_name*: name of PM2.5 concentration variable **in quotations**

*export\_dir*: variable name of directory to save data

*export\_name*: name of file to export **in quotations**

For other inquiries, please ask us and/or see the R manual:

<https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf>