Code-Documentation

Vivienne Maxwell

12/14/2021

Contents

I. R Set-up

In order to run the necessary code, you will need the following R packages:

- tidyverse
- pls
- scales
- readr
- Metrics
- ggplot2
- moderndive

Use the 'install.packages(" ")' function to install the packages. Use the 'library()' function to load the packages.

II. Script Summaries

The code is separated into three different scripts.

01_createDataFrameScript.R

The code loads the OPUS files into R, creates two separate dataframes (one consists of wavenumbers and the other consists of absorbance values), and adds the actual BSi percentages to the absorbance dataframe.

02_createModelScript.R

This code loads the absorbance data that contains the actual BSi percentages. That data is run through the partial least squares regression model. After the model is run, you create a root mean squared error plot (RMSEP) to determine the number of components. Then you load in the wavenumber data and combine it with the loadings from the first three components of the pls model. Once the dataframe is created, you can generate the loading plot to determine the parts of the spectrum that are most heavily weighted in the model.

03_modelAccuracy.R

The final script assess the model's prediction accuracy in two ways. You will calculate the regression error, which is the predicted BSi percentages minus the actual BSi percentages. Then there is code for two visualizations. The first is a comparison of the regression error; it is a side-by-side of the actual BSi percentage versus the predicted BSi percentage for each sample. The second visualization shows you where the model is overpredicting (green) and underpredicting (red).

III. Code Review: line by line

01_createDataFrameScript.R

```
#Load relevant libraries----
library(tidyverse)

## read in txt files automatically----
fname <- list.files("Samples/greenlandSamples", full.names = T)
###Dimensions = 1:28 (Greenland) ###1:100 (Alaska)
```

Load the 'tidyverse' package and read in the list of OPUS files from your local device. Make sure the path correctly reflects where the files are stored on your local device.

Values	
fname	chr [1:28] "Samples/greenlandSamples/FISK-10.0.txt…

This is what the fname vector will look like. For the Greenland samples, it should be a vector with 28 samples.

```
filelist <- lapply(fname, read.table, sep="")
### Dimensions = 28 [1:3697] (Greenland) 100 [1:1882] (Alaska)
```

This line of code creates the filelist object where 'read.table' function is mapped to each sample contained in fname using the 'lapply' function. Beware of the 'sep=' argument, depending on the file it could either be an empty space '(sep= " ")' or a comma '(sep= ",")'.