Valiveti Manikanta Bhuvanesh

19BCD7088

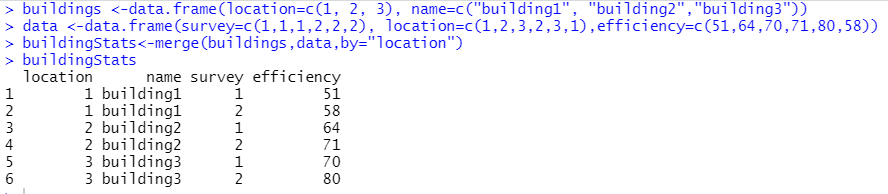
L55+L56

buildings <-data.frame(location=c(1, 2, 3), name=c("building1", "building2","building3"))

data <-data.frame(survey=c(1,1,1,2,2,2), location=c(1,2,3,2,3,1),efficiency=c(51,64,70,71,80,58))

buildingStats<-merge(buildings,data,by="location")

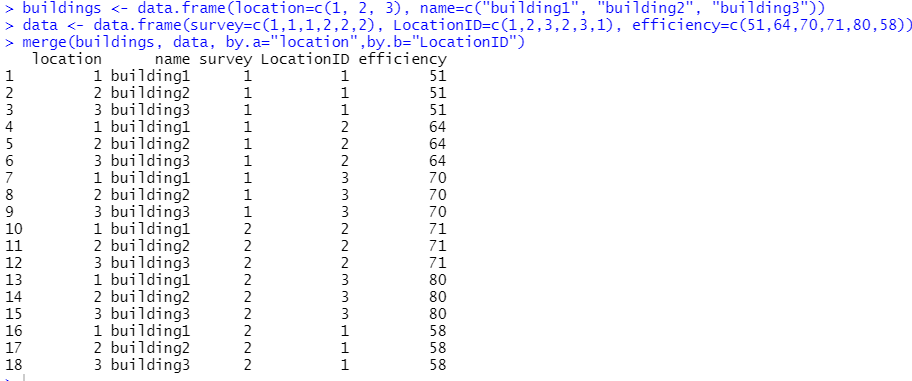
buildingStats



buildings <- data.frame(location=c(1, 2, 3), name=c("building1", "building2", "building3"))

data <- data.frame(survey=c(1,1,1,2,2,2), LocationID=c(1,2,3,2,3,1), efficiency=c(51,64,70,71,80,58))

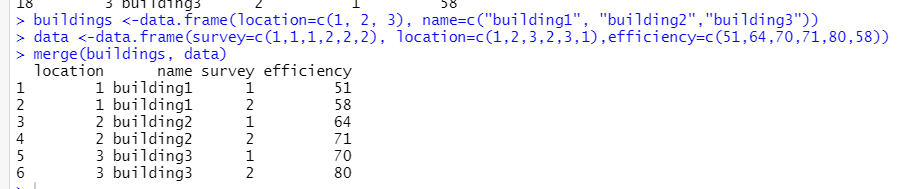
merge(buildings, data, by.x="location",by.y="LocationID")



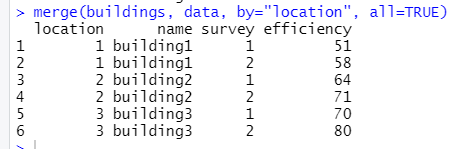
buildings <-data.frame(location=c(1, 2, 3), name=c("building1", "building2","building3"))

data <-data.frame(survey=c(1,1,1,2,2,2), location=c(1,2,3,2,3,1),efficiency=c(51,64,70,71,80,58))

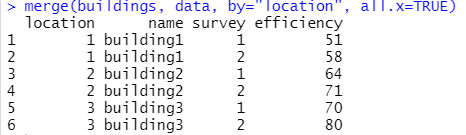
merge(buildings,data)



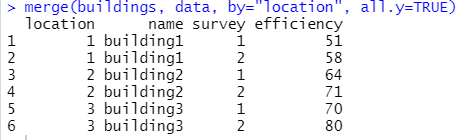
merge(buildings, data, by="location", all=TRUE)



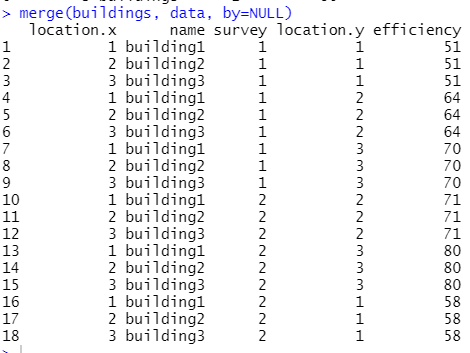
merge(buildings, data, by="location", all.x=TRUE)



merge(buildings, data, by="location", all.y=TRUE)



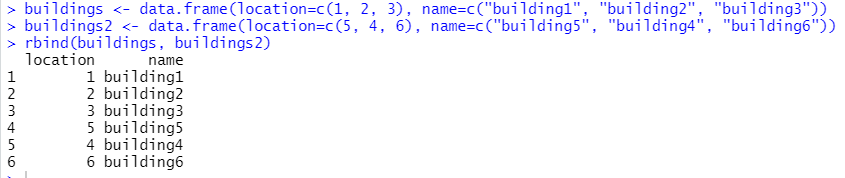
merge(buildings, data, by=NULL)



buildings <- data.frame(location=c(1, 2, 3), name=c("building1", "building2", "building3"))

buildings2 <- data.frame(location=c(5, 4, 6), name=c("building5", "building4", "building6"))

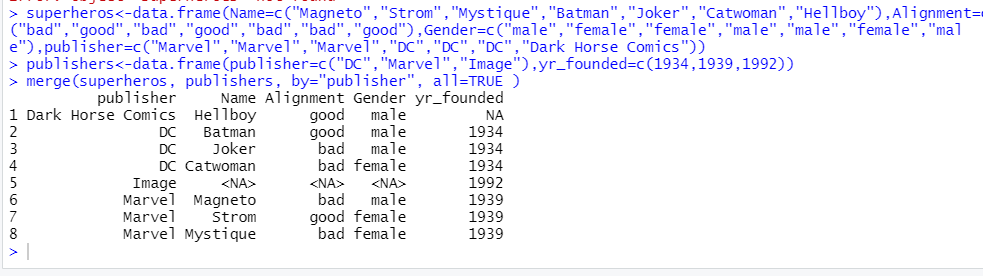
rbind(buildings, buildings2)



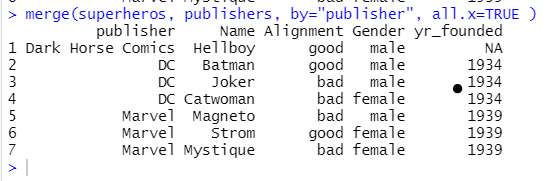
superheros<-data.frame(Name=c("Magneto","Strom","Mystique","Batman","Joker","Catwoman","Hellboy"),Alignment=c("bad","good","bad","good","bad","bad","good"),Gender=c("male","female","female","male","male","female","male"),publisher=c("Marvel","Marvel","Marvel","DC","DC","DC","Dark Horse Comics"))

publishers<-data.frame(publisher=c("DC","Marvel","Image"),yr\_founded=c(1934,1939,1992))

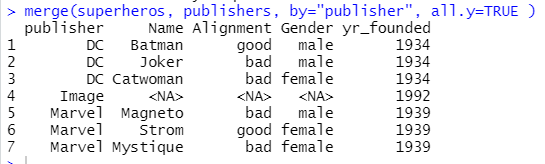
merge(superheros, publishers, by="publisher", all=TRUE )



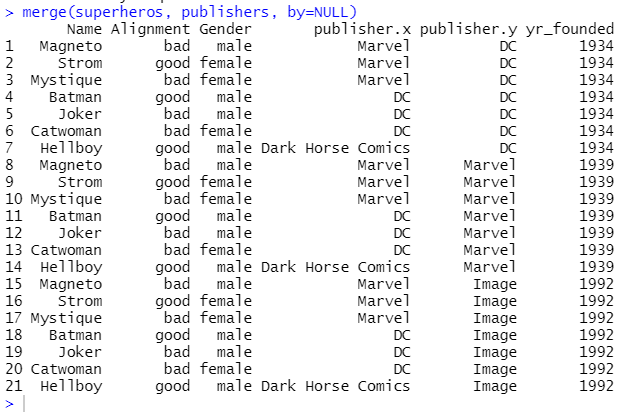
merge(superheros, publishers, by="publisher", all.x=TRUE )



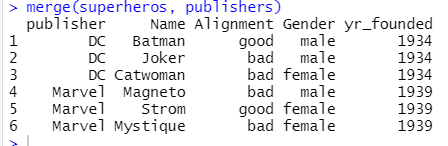
merge(superheros, publishers, by="publisher", all.y=TRUE )

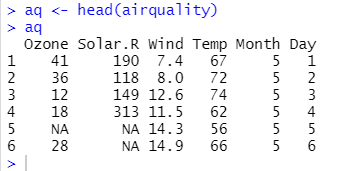


merge(superheros, publishers, by=NULL)



merge(superheros, publishers)





* Write a command to export (store/save) data into the file cat\_test1.txt (Use only two arguments). After creating file check the output.

write.table(aq, file = "cat\_test1.txt")

write.table(aq, file = " cat\_test1.csv" , row.names = TRUE)

Write a command to export data into the file cat\_test2.txt. Use separator as comma

write.table(aq, file = "cat\_test2.txt", sep = ",")

write.csv(aq, file = " cat\_test2.csv" , row.names = TRUE,sep=”,”)

* Write a command to export data into the file cat\_test3.txt. Use separator as semi- colon

write.table(aq, file = "cat\_test3.txt", sep = ";")

write.csv2(aq, file = " cat\_test3.csv" , row.names = TRUE, sep = ";")

* Write a command to export data into the file cat\_test3.txt. Use separator as tab (use \t to insert tab)

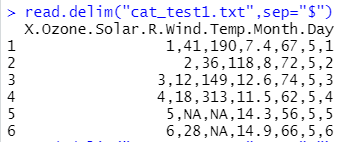
write.table(aq, file = "cat\_test3.txt", sep = "\t")

write.csv(aq, file = "cat\_test3.csv", sep = "\t")

* Can a separator be any string that you want to insert as delim?? Experiment it.

Any separator can be any string. Here I kept the separator as “$”

Any single valued character can be used



* What are Delimiters? What is the need of delimiters?

A delimiter is a sequence of one or more characters for specifying the boundary between separate, independent regions in plain text, mathematical expressions or other data streams.

This is used to read a file in table format and creates a dataframe from it, where cases corresponding to lines and variables to fields in the file.

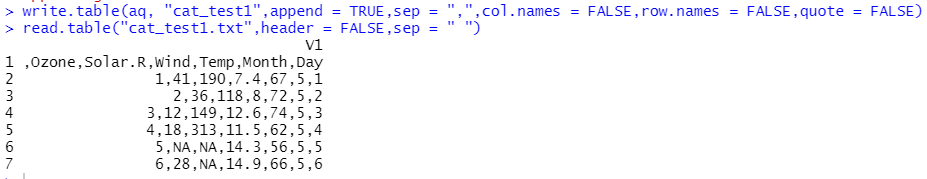
* What is the difference between over writing and appending?

The existing file can be replaced and the records can be appended to the existing content.If Append is set to true, records are appended to the file.If Append is set to false ,the file is overwritten.

* Write a command to append the same data into the file cat\_test1.txt. After creating file check the output.

write.table(aq, "cat\_test1",append = TRUE,sep = ",",col.names = FALSE,row.names = FALSE,quote = FALSE)

read.table("cat\_test1.txt",header = FALSE,sep = " ")



* Write a command to append the same data into the file cat\_test2.txt. Use separator as comma

write.table(aq, "cat\_test2",append = TRUE,sep = ",",col.names = FALSE,row.names = FALSE,quote = FALSE)

* Write a command to append the same data into the file cat\_test3.txt. Use separator as semi-colon

write.table(aq, "cat\_test3",append = TRUE,sep = ";",col.names = FALSE,row.names = FALSE,quote = FALSE)

* Write a command to append the same data into the file cat\_test3.txt. Use separator as tab (use \t to insert tab)

write.table(aq, "cat\_test3",append = TRUE,sep = "\t",col.names = FALSE,row.names = FALSE,quote = FALSE)

* Can you store the data into .csv files using cat command?

**Yes**

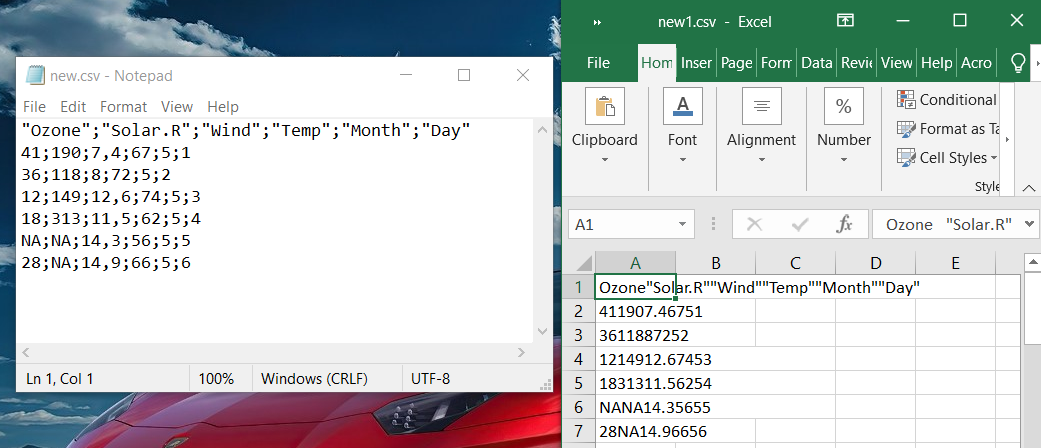
* Write a command to export the data to cat\_test1.csv.

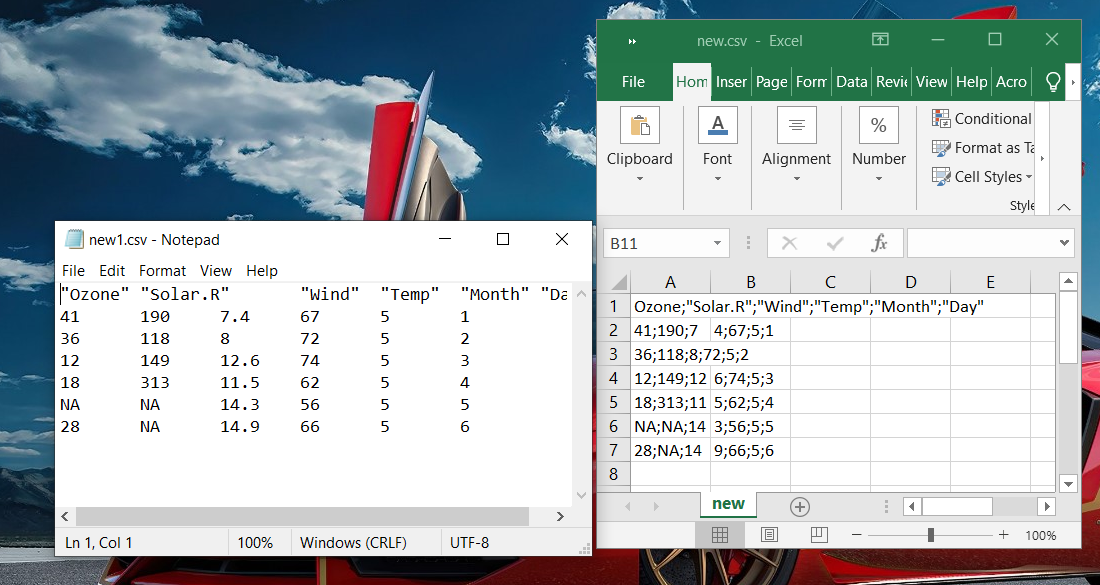
write.csv(aq, file = "cat\_test1.csv")

* Write a command to export the data using various separators such as semi-colon and tab and store them in new .csv files. Open newly created files using both excel and notepad. Observe the difference.

write.csv2(aq, file = "new.csv", row.names = FALSE)

write.table(aq, file = "new1.csv", row.names = FALSE,sep = "\t")





* What are other arguments you can use while exporting data to a file using cat function. (**Hint:** Use ?cat command to learn about other arguments and experiment on them)

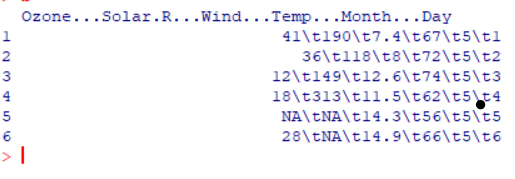
cat(... , file = "", sep = " ", fill = FALSE, labels = NULL, append = FALSE)

Arguments

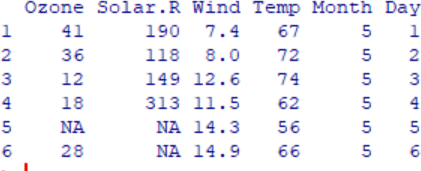
|  |  |
| --- | --- |
| ... | R objects (see ‘Details’ for the types of objects allowed). |
| file | A [connection](http://127.0.0.1:15349/library/base/help/connection), or a character string naming the file to print to. If "" (the default), cat prints to the standard output connection, the console unless redirected by [sink](http://127.0.0.1:15349/library/base/help/sink). |
| sep | a character vector of strings to append after each element. |
| fill | a logical or (positive) numeric controlling how the output is broken into successive lines. If FALSE (default), only newlines created explicitly by "\n" are printed. Otherwise, the output is broken into lines with print width equal to the option width if fill is TRUE, or the value of fill if this is numeric. Non-positive fill values are ignored, with a warning. |
| labels | character vector of labels for the lines printed. Ignored if fill is FALSE. |
| append | logical. Only used if the argument file is the name of file (and not a connection or "|cmd"). If TRUE output will be appended to file; otherwise, it will overwrite the contents of file. |

* Write a command to read the data from cat\_test1.txt. Display the output.

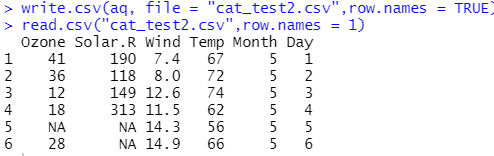
read.csv("cat\_test1.csv",row.names = 1)



* Write a command to read the data from cat\_test2.txt. Display the output. Modify your command to get the output as given below



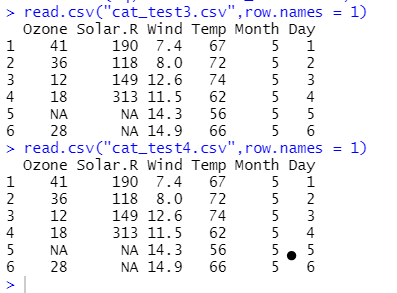
read.csv("cat\_test2.csv",row.names = 1)



* Write a command to read the data from remaining files and display the output as in the above figure.

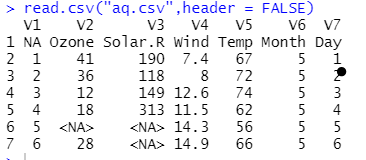
read.csv("cat\_test3.csv",row.names = 1)

read.csv("cat\_test4.csv",row.names = 1)



* Write a command to read the data without column names

read.csv("aq.csv",header = FALSE)



* Write a command to read the data without row names

read.csv("aq.csv",header = FALSE)

