Discussion Forums

Week 4

SUBFORUMS

All

Assignment: swrl Lesson 2: Simulation

Assignment: swirl Lesson 3: Base Graphics

Assignment: swirl Lesson 1: Looking at Data

← Week 4

[Tips] A few pointers for assignment 3

Al Warren Week 4 · 3 years ago · Edited

The following are some suggestions based on some common questions and issues that always seem to come up with the hospital assignment. Keep in mind there is no one right method of solving the assignment. The main purpose of the post is to get you thinking about some different ways to approach things.

Reading the Data

If you use read.csv with na.strings="Not Available" and stringsAsFactors=FALSE you shouldn't need to convert any columns to numeric or character. Plus, your sorts will work as expected.

Selecting Columns

You only need at most three columns - hospital name, state, and one of three outcome columns. To make things easier to work with use the names function to rename the three columns. Something like *names(my_data) <- c("hospital", "state", "outcome")*.

When selecting an outcome column, think about how you might go about doing that. Everyone has their preferences but personally I like to do the column subsets with column indexes. One way might be to use a column index variable that's based on the outcome function argument. For this particular data set, heart attack is column 11, heart failure is column 17, and pneumonia is column 23. All you have to do to select columns is use brackets and pass a numeric vector to the right side of the comma. For example, if you named your outcome index column_index, you could select the columns with something like *df[, c(2,7,column_index)]*.

Hint: if you setup a named vector with something like *outcomes <- c("heart attack"=11, "heart failure"=17, "pneumonia"=23)* then you can use that to both test the function argument and select the column. Something like *df[, c(2,7,outcomes[outcome])]*. Also, when you validate the outcome argument instead of using %in% outcomes you'd use %in% names(outcomes).

Removing NA Values

You can use na.omit or complete.cases to remove NA values but don't do that until you've reduced the data to only one output column plus name and state. If you remove NAs immediately after reading the data you can lose data that you need.

Ranking

Think logically about the data and ranking. For example, if you were to sort by state then outcome then hospital name then your data will be in proper rank order including the tie breaker. At that point, all you have to do is take a state subset and use an index to get best, worst, or a particular rank. For best you'd use 1 for the index, for worst you'd use nrow(your_subset) and for just num you'd use that num. Don't worry about testing for num greater than number of rows because if you use an index larger then the number of rows you'll get NA which is what you want anyway.

Processing the Data

If you need to process data in groups consider using one of the apply functions. One of the easiest to work with is lapply. For a walk-through on how lapply works, see this post - https://www.coursera.org/learn/r-programming/module/6BaZ2/discussions/qcv_orZHEeWF2Q53QdZUbw.

Suppose you have data in three columns ordered by state then outcome then hospital. Then suppose you split on state using something like *split(your_data, your_data$state)*. You get a list of data frames ordered by state. Because you already ordered the data everything else is in rank order. Then suppose you only want a list of hospitals by rank. You run the list of data frames through lapply and use a function to store the hospital names in a list.

What happens is lapply runs each item in the list through the function. The data passed to the function is one of the data frames from the list created by split(). The value returned by that function gets stored in the results of lapply() which is a list.

So, if you split the data frame on state then *lapply(split.data, hospitalNameFunction)* would return a list where the name of each item is state and the value is the hospital name. You can get the values as a vector using *unlist(results_of_lapply)*. You can get the states using *names(results_of_lapply)*. You could also use sapply and then you wouldn't need to use unlist.

Then all you have to do is assemble the data frame using the unlisted values and list names using something like this - *data.frame(hospital=unlisted_values, state=list_names, row.names=list_names).*

Finally

As with everything R there are many ways to do things. These are just a few suggestions. Most of all, have fun and experiment.

Note: This post is now pinned and available to all sessions but has been closed for comments. If you have questions on any of these methods please post a new thread. Thanks.

⇧ 128 Upvotes Reply Follow this discussion

🔒This thread is closed. You cannot add any more responses.

Earliest **Top** **Most Recent**

MK

Michael Krafick · 3 years ago

Al, I would not have done this assignment without your guide on a general approach with various scenarios. Matter of fact I was so frustrated with week 3 and 4 that I was tempted to cheat or quit. But I didn't.

Although P4 of the last assignment wasn't completely finished (see post by me on where I am stuck) - I had enough of the core to take the test and pass with an 80%, giving me close to a 90% for the month.

You made a difference for me. Thank you so much.

⇧ 9 Upvotes Reply

AJ

Ayo Johnson · 3 years ago

Hi Al,

For some reason, my code is not quite working for the "worst" scenario. I have done everything in order, used the anonymous function etc. Tried nrow and length interchangeably to define the worst, but still getting different results as below

```
tail(rankall("pneumonia", "worst"),3)

hospital state

WI AURORA WEST ALLIS MEDICAL CENTER WI

WV BECKLEY VA MEDICAL CENTER WV

WY MEMORIAL HOSPITAL OF CARBON COUNTY WY
```

I get the feeling that there's something wrong with the lapply anonymous function which reads as follows, but can't figure it out.

```
fini <- lapply(splitdata, function (x, num){

x <- x[order(x[3], x[1]),]

if(class(num)=="character"){

if (num == "best"){

return(x$hospital[1])

}

else if(num =="worst"){

return (x$hospital[length(x)])

}

}

else{

return(x$hospital[num])

}

}, num)
```

like i said i tried nrow instead of length but still getting the wrong result. Please help!

JM

José Morgado · 3 years ago

Thank you for the tips.

Truth is, I complained allot about how there isn't enough information or connection between the assignments and the classes. However, in the end, it really builds your knowledge to be clueless for a while.

12 Upvotes Reply

Shankar Raju · 3 years ago

Hi,

I struggled a lot for this assignment and learned a lot. I ended up completing by the following steps:

1) Read the data

2) Sorted and assigned the rank on the data for the required columns (asc or desc) based on the input argument. I used plyr package's functions for this

3) Then used merge function to populate "NA" for the missing states

4) Then displaying the "num" hospitals for each state by simple subsetting on the above data frame.

I spent lot of time on figuring out why my results are not as expected, then found out the outcome field which I am sorting is character instead of numeric is the root cause of the issue !

All the best!

Thanks.

3 Upvotes Reply

Vamsee Addepalli · 3 years ago

Thanks a lot Al. Your tips helped me to organise my thoughts on approaching this assignment much better than if I didn't follow your tips. Nevertheless, for the fourth part, I did try to deviate from the way you explained (without using apply function) but met with no success. Completed my assignment and quiz though thanks to you.

0 Upvotes Reply

PF

Peter Fleer · 3 years ago

Hi Al,

Thanks a lot for these hints. It helped me a lot, above all for the rankall function. With the help of your suggestions I was able to build - I think - a rather elegant function (for my current capabilities) using split and lapply.

0 Upvotes Reply

DL

David Larlick · 3 years ago

Thank you for the tips here. I had to read this very carefully... I don't think I would have been successful without this excellent guidance.

0 Upvotes Reply

Jagdeep Bhaura · 3 years ago

Hi, just a quick question about the sample out put from RankAll

The output is showing

STATE, HOSPITAL NAME, STATE

Is that required, or is

STATE, HOSPITAL NAME ?

0 Upvotes Hide 2 Replies

Al Warren · 3 years ago

The states you see on the left side of the examples are row names.

0 Upvotes

SD

Santhoshi Dharmireddi · 3 years ago

Do I need to submit this assignment on github?

4 Upvotes

steve zhang · 3 years ago

Hi Warren: I found your tips are critical for assignment 3.

I have test and completed the first 3 part but struggle with part 4.

mine solution could work with simply loop through all 54 states but fails to combine with apply function.

Could you provide some guidance? thanks

assume **df** contains 3 col named "hospital","state","outcome" and already sorted as you suggested (order by state then outcome then hospital name)

I just cannot properly manipulate the list object (**statewise**) after the split statement

```
df <- data[,c(2,7,colindex)]
names(df)<-c("hospital","state","outcome")
## For each state, find the hospital of the given rank
interest <- df[!is.na(df$outcome),]
sorted <- interest[order(interest$outcome,interest$hospital),]
statewise <- split(sorted,sorted$state)

lapplied <- lapply(statewise,"[",1)
sta <- names(lapplied)
hos <- lapply(lapplied,"[[",num)


```

lapplied contain the following format as list (sorted hospital name in respective state)

```
> head(lapplied,2)
$AK
[1] "PROVIDENCE ALASKA MEDICAL CENTER" "ALASKA REGIONAL HOSPITAL"
[3] "FAIRBANKS MEMORIAL HOSPITAL" "ALASKA NATIVE MEDICAL CENTER"
[5] "MAT-SU REGIONAL MEDICAL CENTER"
$AL
[1] "CRESTWOOD MEDICAL CENTER"
[2] "BAPTIST MEDICAL CENTER EAST"
[3] "SOUTHEAST ALABAMA MEDICAL CENTER"
```

If I want to see the best hospital in every state, I could use lapply again and store the result into **hos** as a list

```
> head(hos)
$AK
[1] "PROVIDENCE ALASKA MEDICAL CENTER"
$AL
[1] "CRESTWOOD MEDICAL CENTER"
$AR
[1] "ARKANSAS HEART HOSPITAL"
$AZ
[1] "MAYO CLINIC HOSPITAL"
$CA
[1] "GLENDALE ADVENTIST MEDICAL CENTER"
$CO
[1] "ST MARYS HOSPITAL AND MEDICAL CENTER"
```

I am also aware of that I could get the best hospital in state='AK' using

```
> hos[[1]]
[1] "PROVIDENCE ALASKA MEDICAL CENTER"
> hos$AK
[1] "PROVIDENCE ALASKA MEDICAL CENTER"
```

But i dont know how to write a good *hospitalNameFunction* could be used as input for new data frame.

if I use following directly, it would give unwanted behaves

```
ndf <- data.frame(hospital=hos,state=sta,row.names = sta)
```

ndf would have 54 rows(expected) but also **55 cols (wrong)**

I think I am hot on rails of the answer, really appreciate your helps. Thanks in advance

0 Upvotes Hide 6 Replies

Al Warren · 3 years ago

Use a function for lapply that handles num. See my response to Michael.

4 Upvotes

YL

yuchen li · 3 years ago

Hi, Warren,

I have one question related to the comments:

If you order properly the worst hospital will always be the last one in the state subset. Just be sure you've done things in the correct order prior to running split and lapply -

## Read the data

## Subset to three columns

## Remove NA Values

## Order by state then outcome then hospital name

## Split by state

## Run lapply

Your function for lapply should take one parameter as the data frame for a state and output a hospital name. The results of lapply will be a named list where the list names are state and the list values are hospital name (one for each state).

## The following function receives a state data frame from the split data

## function_for_lapply(data) { do something with data }

I understand the process. but where should I DEFINE the function properly? I defined it independently(in one file, but two independent functions, rankall, and function_for_lapply), but it did not work. Thanks.

0 Upvotes

Al Warren · 3 years ago

The function should be defined within your assignment function.

```
assignment_function <- function() {
    lapply_function <- function() {
        ## do something
    }
}
```

0 Upvotes

YL

yuchen li · 3 years ago

Thanks, Al, that is exactly I am thinking. But I think it will make the code less compact.

coursera

1 Upvote

Al Warren · 3 years ago

In R, compact code is not necessarily efficient code.

1 Upvote

YL

yuchen li · 3 years ago

Thank, Al. I successfully finished the course and will keep advancing in the specialization track. You are very helpful. Maybe it is a lot to ask, but can I befriend you and consult with you in the future? and how?

0 Upvotes

M

Michael · 3 years ago

Thanks, this was really helpful.

I've been able to create working functions, but I'm having a little trouble understanding how to get the "worst" for Part 4. I used the lapply function as you suggested, using an anonymous function to extract the hospital name. In the rankhospital function, obtaining the worst hospital was pretty simple since you could just take the length of one of your columns in the data frame for the specified state (or probably more elegantly nrows as you suggest), but in this case, the index of the worst hospital changes for each state. I'm thinking my anonymous function might not be the right way to go. Any other leads I should follow?

2 Upvotes Hide 5 Replies

Al Warren · 3 years ago · Edited

If you order properly the worst hospital will always be the last one in the state subset. Just be sure you've done things in the correct order prior to running split and lapply -

```
## Read the data
## Subset to three columns
## Remove NA Values
## Order by state then outcome then hospital name
## Split by state
## Run lapply
```

Your function for lapply should take one parameter as the data frame for a state and output a hospital name. The results of lapply will be a named list where the list names are state and the list values are hospital name (one for each state).

```
## The following function receives a state data frame
    from the split data
## function_for_lapply(data) { do something with data }
```

1 Upvote

M

Michael · 3 years ago

Thanks for the reply. I ended up putting my call to lapply in my "if" conditional to deal with inputs of "worst" and "best" and used nrow in the anonymous function (as you mentioned above). Previously I just used the conditional to redefine "num" when "worst" or "best" was entered, but couldn't figure out how to make it work with the structure of lapply outside the conditional. I'd guess there is a way to do it, but it's working now so I'm satisfied. Thanks again for your help!

0 Upvotes

Al Warren · 3 years ago

Ideally, the "if" conditional would go in the function called by lapply.

```
function_for_lapply(data) {
    if(condition1) do something
    if(condition2) do something else
    return a value
}
```

1 Upvote

Pamela Monaghan · 3 years ago

This was precisely the same problem I had at the end of a long session. I was trying the same thing as Michael but it was not working. Your assurance that that method could indeed work convinced me that my logic was OK, perhaps I just had a syntax error. After a brief investigation I got my code to work and I am done!! Thanks to you both!! Al, I had your tips open in a separate window and you guided me right along. These posts have been invaluable!! Cheers!

0 Upvotes

SD

Steven Desmarais · 3 years ago

I have a split data set that is sorted correctly by state, hospital and outcome. My problem is that I don't know how to access the split data frames in the function, i.e., to get the number of rows and select the correct ranked record in each state. The example in the book uses a function(x) that somehow could use the subset data frames. My split data is called "subStates"

lapply (subStates, function(x) { colMeans(x[,"Outcome"])})

This gets message "Error in colMeans(x[,"Outcome"]): x must be an array of at least two dimensions. How do I access the data frames by state in the split data? Thanks.

⇧ 6 Upvotes

Al Warren · 3 years ago

Make sure you're not using colclasses in read.csv.

⇧ 7 Upvotes Reply

SR

Schneider, René · 3 years ago

Dear Al,

thanks for these helpful hints.

I use na.strings="Not Available" and stringsAsFactors=FALSEwhile reading the file.

Nevertheless, my sorting algorithm behaves strangely.

It only sorts values starting from 10.0, although I declared the value of the new dataframe as numeric before subsetting and selecting this row for the new dataframe.

⇧ 1 Upvote Hide 2 Replies

JO

Janaina Lima de Oliveira · 3 years ago

Dear René,

did you manage to solve this problem? It's happening the same here, and i have no idea on what to do.

Thanks!

⇧ 3 Upvotes

AS

Andreas Schuderer · 3 years ago

René, you stated that you declared the value of the new dataframe as numeric. You cannot make a whole data frame the same class, only vectors. So

```
df$outcome <- as.numeric(df$outcome)
```

should work, whereas

```
df <- as.numeric(df)
```

shouldn't work (remember you got the state and hospital names in there, too).

Did you order before or after cleaning your data of NA's?

order has an optional argument na.last = TRUE, for what it's worth.

⇧ 0 Upvotes

‹ 1 ›