R and Data Manipulation Using dplyr

Main objective <- Review some commonly used data manipulation features using the dplyr package

Berenica Vejvoda, Data Librarian
bvejvoda@uwindsor.ca
February 13, 2020
Collaboratory, Leddy Library, University of Windsor

Academic Data Centre:
Drop-In Hours: 12:30-4:30PM
By appointment: libdata@uwindsor.ca

Contents

Set Working Directory……3

Install dplyr……3

Load dplyr……3

Load Data into R Using Command Line……4

Attach a Data frame……4

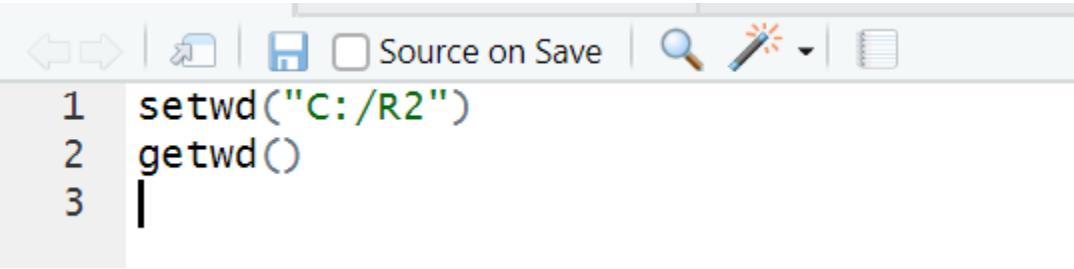
Selecting Columns……5

Subsetting Data……5

Combine R Objects by Rows and Columns……7

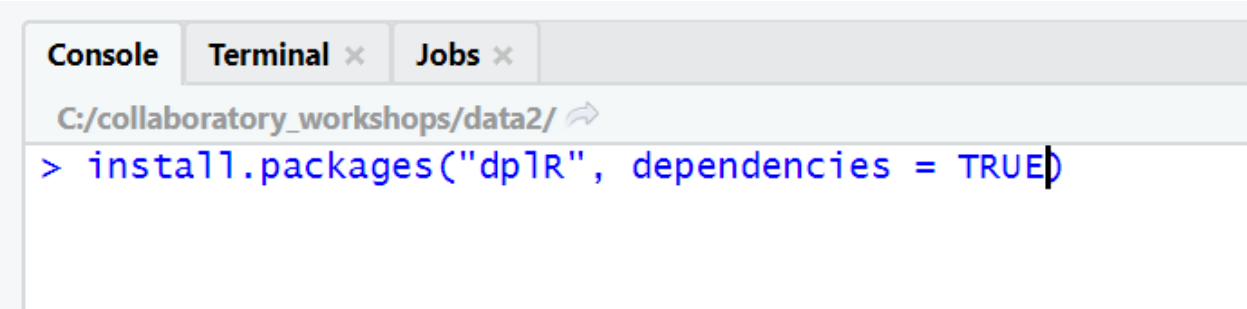
Transpose……8

Set Working Directory



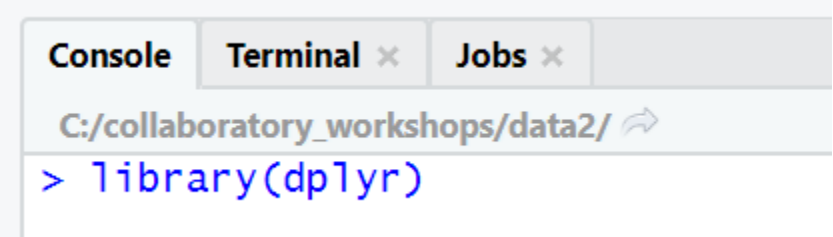
Install dplyr

Install dplyr using the command line and include dependencies.



Load dplyr

Now we need to load dplyr package into R from the /library location.



Loading Data in R

Read a file in table format and create a data frame from it.

Note that there are different functions for different data formats.

Note also the argument “header = TRUE” which holds if the file contains the names of the variables as its first line.

The data we will be using is a public use microdata file from Statistics Canada. Specifically, we will use the latest 2016 Census data.

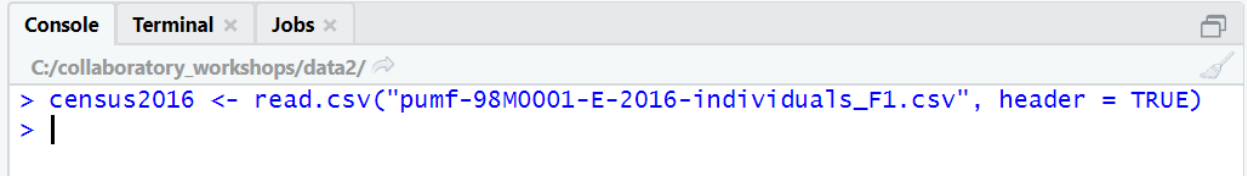
Go to https://search1.odesi.ca/#/. Click on Browse. Drill through Census of Population -> Canada -> 2016 -> Public Use Microdata Files (PUMF). Click on the diskette icon. Choose Comma Separated Value File and download the “Individuals File”.

Unzip the files and extract the data and documentation to your set working directory.



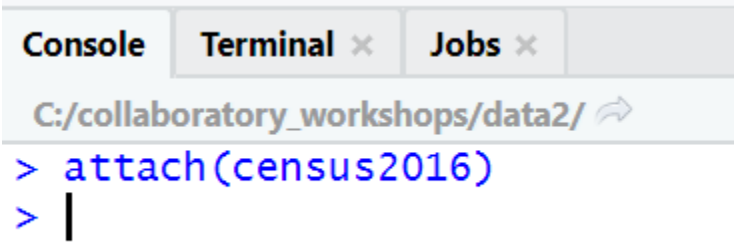
Load Data into R Using Command Line

Go ahead and load the data into R by creating a new data frame (the object where the data will be stored) and using the read.csv command. Remember to add the header = true parameter (indicates that the first row are variable names).



Attach a Data Frame

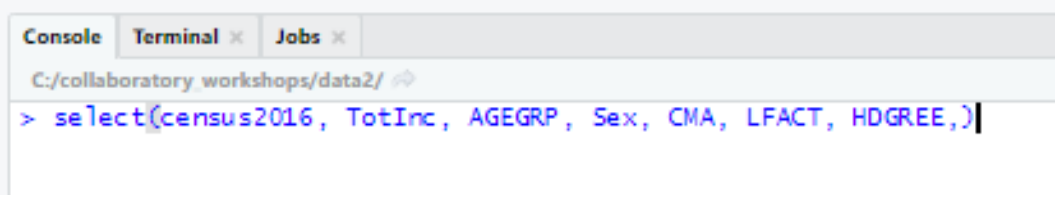
To access specific variables, we need to use e.g. census2016$TotInc
However, to access the columns directly in R, you can use the function attach, after which you can refer to the columns directly as shown below:



If you at some point no longer need to access variables directly, you can also use the detach function.

Selecting Columns

To select columns of a data frame, use select(). The first argument to this function is the data frame (census2016), and the subsequent arguments are the columns to keep.



Selecting Select Columns

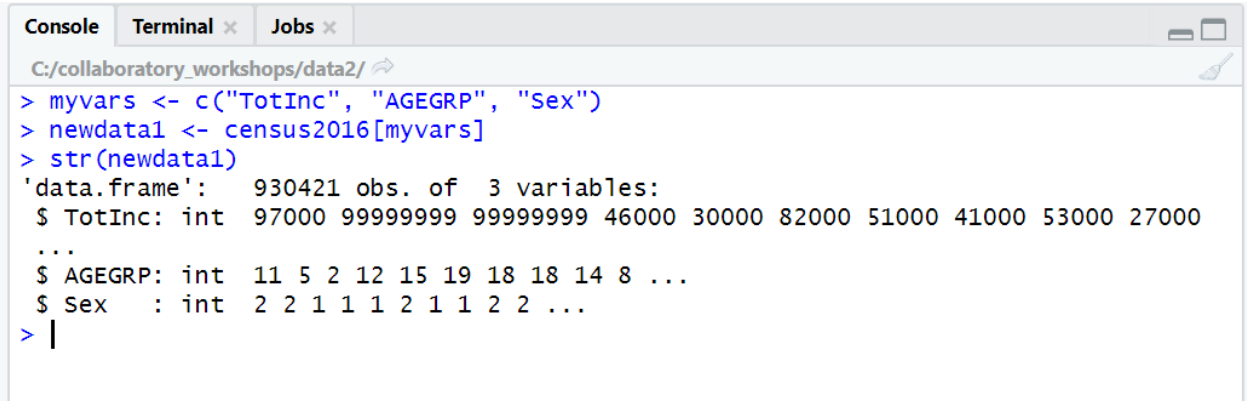


Subsetting Data

This time we won't just select variables but KEEP them in a new data frame as sometimes you want to split files either by columns or rows.

First, we will create a subset of three variables: TotInc, AGEGRP, Sex.

Then, we will create a new data frame for the new vector "myvars"



Now let's create a second subset using CMA, LFACT, HDGREE. Then create a second new data frame for

```
Console  Terminal  Jobs
C:/collaboratory_workshops/data2/
> myvars <- c("CMA", "LFACT", "HDGREE")
> newdata2 <- census2016[myvars]
> str(newdata2)
'data.frame':   930421 obs. of  3 variables:
 $ CMA   : int  505 505 505 999 999 999 999 825 421 532 ...
 $ LFACT : int  1 99 99 1 1 13 13 13 1 1 ...
 $ HDGREE: int  4 99 99 9 6 2 6 6 3 2 ...
> str(newdata2)
'data.frame':   930421 obs. of  3 variables:
 $ CMA   : int  505 505 505 999 999 999 999 825 421 532 ...
 $ LFACT : int  1 99 99 1 1 13 13 13 1 1 ...
 $ HDGREE: int  4 99 99 9 6 2 6 6 3 2 ...
>
```

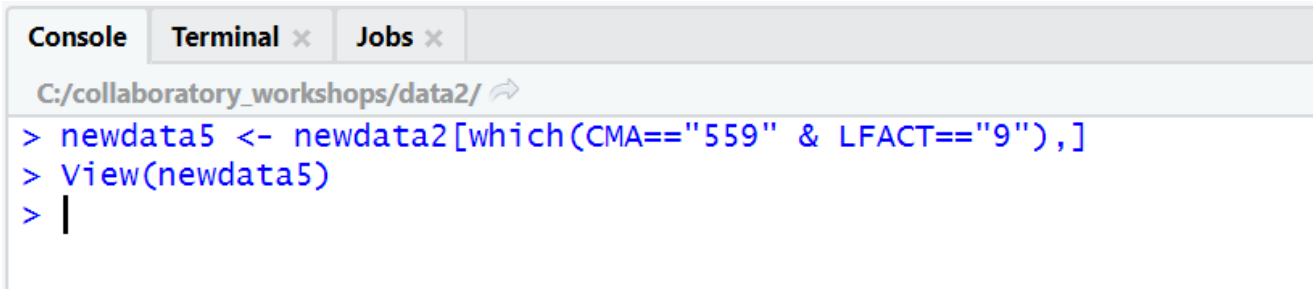
Alternatively, you can subset another way.

```
Console  Terminal  Jobs
C:/collaboratory_workshops/data2/
> newdata3 <- census2016[c(1,5:10)]
> str(newdata3)
'data.frame':   930421 obs. of  7 variables:
 $ PPSORT: int  453141 923226 385097 732612 143665 459269 632419 284347 52611 36927
 ...
 $ WT3   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ WT4   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ WT5   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ WT6   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ WT7   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ WT8   : num  0 0 0 593 0 ...
>
```

Let' try another one and select the first five observations and first three variables.

```
Console  Terminal  Jobs
C:/collaboratory_workshops/data2/
> newdata4 <- census2016[1:5, 1:3]
> str(newdata4)
'data.frame':   5 obs. of  3 variables:
 $ PPSORT: int  453141 923226 385097 732612 143665
 $ WEIGHT: num  37 37 37 37 37.1
 $ WT1   : num  0 0 0 0 0
>
```

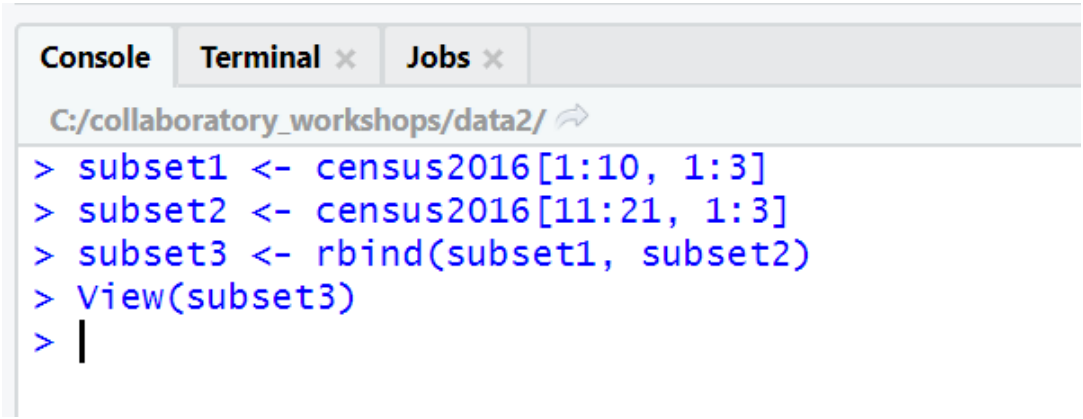
Now let's try subsetting for specific values within variables. We will create a new data frame for those living in Windsor who are unemployed and looking for work.



Combining R Objects Using Rows and Columns

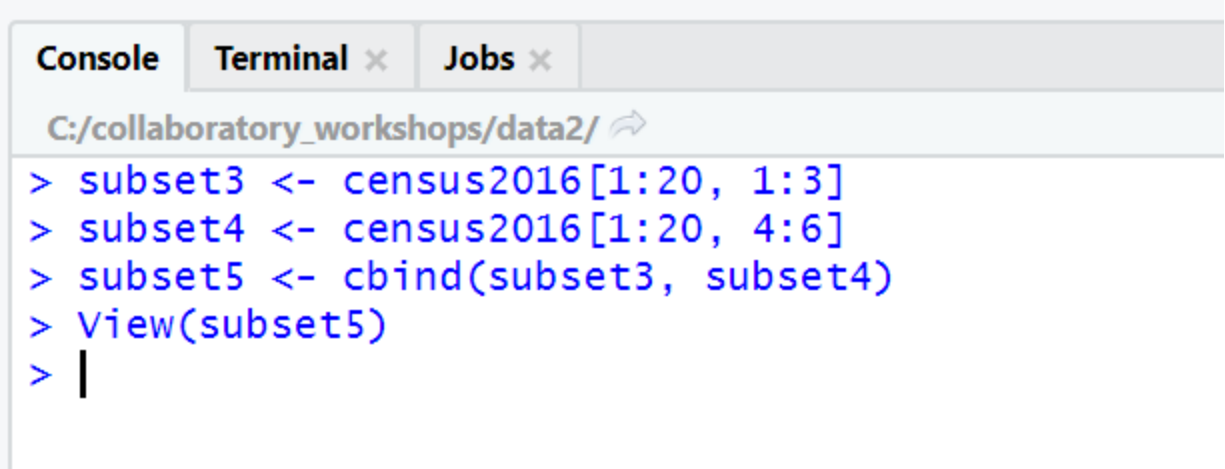
rbind combines data frames (and vectors and matrices) by rows

Start by creating two new subsets from the original census2016 data frame that we started with that selects the first 10 cases and a second subset with the next 10 cases. Then construct the rbind statement.



cbind combines data frames (and vectors and matrices) by columns

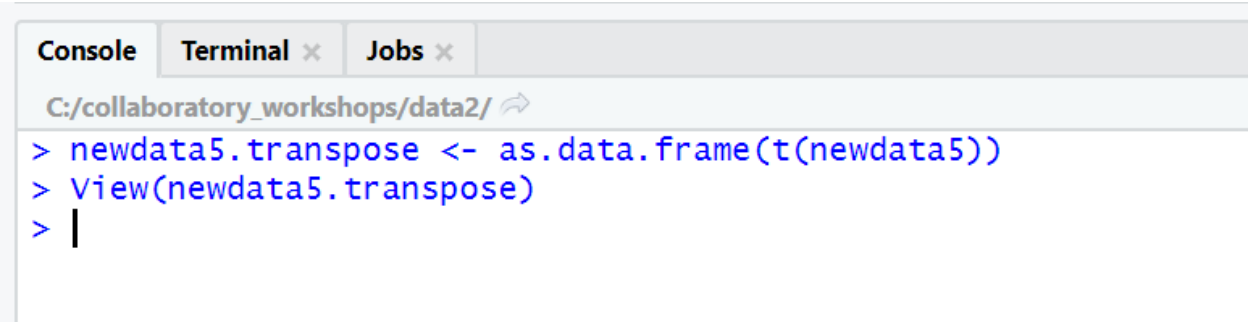
Create two more new subsets from the original census2016 data frame that we started with that selects the first 3 columns and another subset that keeps columns 4 through 6. This time combine the columns using the cbind function.



Transpose

The t() function will flip your data – i.e. columns to rows or vice versa.

Let's flip one of our subsets (newdata5).



# CMA: Census metropolitan area or census agglomeration of current residence (2016)	
Information	[Type= discrete] [Format=numeric] [Range= 205-999] [Missing=*]
Statistics [NW/ W]	[Valid=930421 / 34460064] [Invalid=0 / 0]
Universe	Persons in private households
Notes	Refers to the census metropolitan area (CMA), census agglomeration (CA), or non-CMA/CA of current residence (on May 10, 2016). Only the largest census metropolitan areas are included.

Value	Label	Cases	Weighted	Percentage (Weighted)
205	Halifax	10690	395880.4	1.1%
399	Moncton – Saint John	7217	267419.9	0.8%
421	Québec	20961	776427.9	2.3%
462	Montréal	108580	4021971.5	11.7%
499	Sherbrooke – Trois-Rivières	9510	352265.1	1.0%
505	Ottawa – Gatineau	34964	1295009.4	3.8%
532	Oshawa	10096	373928.4	1.1%
535	Toronto	158375	5865778.8	17.0%
537	Hamilton	19826	734301.1	2.1%
539	St. Catharines – Niagara	10631	393743.3	1.1%
541	Kitchener – Cambridge – Waterloo	14024	519410.8	1.5%
555	London	13207	489151.3	1.4%
559	Windsor	8855	327965.1	1.0%
577	Brantford – Guelph – Barrie	12942	479336.4	1.4%
588	Kingston – Peterborough	7457	276187.0	0.8%
599	Greater Sudbury / Grand Sudbury – Thunder Bay	7524	278668.5	0.8%
602	Winnipeg	20457	757166.5	2.2%
799	Regina – Saskatoon	14107	522681.6	1.5%
825	Calgary	37052	1371657.0	4.0%
835	Edmonton	35034	1296951.1	3.8%
933	Vancouver	65582	2429306.8	7.0%
935	Victoria	9817	363644.1	1.1%
988	Kelowna – Abbotsford	9814	363532.9	1.1%
999	Other census metropolitan areas, census agglomerations and o	283699	10507679.1	30.5%

Warning: these figures indicate the number of cases found in the data file. They cannot be interpreted as summary statistics of the population of interest.

LFACT: Labour: Labour force status

Information	[Type= discrete] [Format=numeric] [Range= 1-14] [Missing=*/99]
Statistics [NW/ W]	[Valid=773290 / 28640463.219] [Invalid=157131 / 5819600.781]
Universe	Population aged 15 years and over in private households
Notes	Refers to the labour market activity of the population aged 15 years and over, in private households, in the week of Sunday, May 1 to Saturday, May 7, 2016. Respondents were classified as either 'Employed,' 'Unemployed,' or 'Not in the labour force.' The labour force includes the employed and the unemployed.

Value	Label	Cases	Weighted	Percentage (Weighted)
1	Employed - Worked in reference week	442781	16399240.3	57.3%
2	Employed - Absent in reference week	22833	845673.5	3.0%
3	Unemployed - Temporary layoff - Did not look for work	4491	166335.2	0.6%
4	Unemployed - Temporary layoff - Looked for full-time work	4840	179259.0	0.6%
5	Unemployed - Temporary layoff - Looked for part-time work	444	16444.7	0.1%
6	Unemployed - New job - Did not look for work	2942	108964.6	0.4%
7	Unemployed - New job - Looked for full-time work	2731	101147.3	0.4%
8	Unemployed - New job - Looked for part-time work	1130	41851.8	0.1%
9	Unemployed - Looked for full-time work	15089	558841.4	2.0%
10	Unemployed - Looked for part-time work	7263	269000.6	0.9%
11	Not in the labour force - Last worked in 2016	18572	687846.2	2.4%
12	Not in the labour force - Last worked in 2015	26727	989900.0	3.5%
13	Not in the labour force - Last worked before 2015	161527	5982598.8	20.9%
14	Not in the labour force - Never worked	61920	2293360.0	8.0%
99	Not applicable	157131	5819600.8	

Warning: these figures indicate the number of cases found in the data file. They cannot be interpreted as summary statistics of the population of interest.