

Data Input and Output

Statistics 427: R Programming

Module 5

2020

Input

Using `c()` and reading in data works, but it is not efficient. Think of a dataset that has over 100 observations and 5 variables. The thought of inputting all of that data manually, . . . just no.

Vectors can be combined together to create datasets (called **data frames**)

Data Frames in R

A dataset entered into R that is ready for analysis is called a **data frame**. A data frame is basically a list (collection) of vectors. You can think of a data frame as a rectangular table of data, with the columns being vectors of numerical or categorical data. The rows of the data frame correspond to the individuals or subjects from which the data variables were recorded.

Basic approach to data input and output

- (1) Set up data in form of text (.txt), comma separated value (.csv), etc. in the folder of choice on your computer.
- (2) Designate a working directory for R.
- (3) Use `read.table()` or `read.csv()` to input data into R
- (4) Can use `attach()` or `with()` to eliminate the need for a two-level name
- (5) Analyze the data
- (6) Print results to the console or printer and/or store results in a file on your computer
- (7) Close R session when finished: use `detach()` if `attach()` was used and use `rm()` to remove the objects from the environment

Data formats

Your uidaho OneDrive works great as the folder you have for your data.

I use .csv and .txt primarily for data input rather than others for ease of use (they cause me less frustration). Additionally, there are other file types that require specific packages ¹ to be installed in order to input them.

You can also read in datasets from websites with just a web address, which may just be my preferred way (if my data is on a website)

Working Directory

R needs to know where to read data files to and from, and always inserting a long computer address is not always fun nor easy. Changing the working directory to the location of choice makes the process of reading in the dataset more manageable.

¹R packages will be a topic for future lectures

R: The easiest way to change the directory is with the Misc task bar (where File is) and Change Working Directory

RStudio: Session, Set Working Directory, Choose Directory

Navigate to the folder where you are keeping your datasets.

Setting working directory in console

The function `getwd()` will state where the working directory (address) is currently. You can change the directory in the console if you want rather than with mouse gestures.

`setwd('address')` with `address` being the folder location. An example would be: `setwd('c:/Documents/My R Folder')`

`read.table()`

Reading in *most* datasets:

General form `read.table('filename.ext', sep=' ', header=F, ...)`

`filename.ext`: the file name and its extension type (.txt,.csv, etc.), can also be a website site address of a dataset

`sep=' '`: separator; .txt is ' ' (white space), .csv is ',' (a comma)

`header=F`: F is false, T is true; treats the first row of the dataset as the variable names

...: other options available

`read.csv()`

This one is specifically for .csv files so that you do not have to use the `sep=' '` options since it is specifically for .csv files.

General form `read.csv('filename.ext', header=F, ...)`

`filename.ext`: the file name and its extension type (.txt,.csv, etc.), can also be a website site address of a dataset

`header=F`: F is false, T is true; treats the first row of the dataset as the variable names

...: other options available

Negating the need for a two-level name using `attach()`

When reading in a data frame, by default, to access the variables in the dataset, you must use a two-level name, as in: `datasetname$variablename`. You can then avoid all that extra typing and avoid unsightly labels on your graphs; the results are that you can now access variables without having to use the two-level name.

`attach(x)`: `x` is the data frame name.

Negating the need for a two-level name using `with()`

`with(x,expr,...)`: `x` is the data frame name and `expr` is an expression (function) to evaluate

Analyses

Plot, calculate, subset, etc. your dataset

Printing and storing results

You can write a newly created dataset with `write.table()` or `write.csv()`.

```
write.table(x,file='filename.ext',sep=' ',row.names=F,...)
write.csv(x,file='filename.csv',row.names=F,...)
```

`x`: the R data frame to write out

`filename.ext` or `filename.csv`: new file name with extension

`sep=' '`: separator (same as in `read.table()`); not needed in `write.csv()`

`row.names=F`: does not print row names (if no row names in original data, `=T` (default) will print row numbers)

`...`: other options

Closing your R session

If you used `attach()` you should always use `detach()` when you are finishing using that dataset. It is considered good practice to clean up your workspace before exiting the program. Additionally, if you have other datasets using `attach()` that have the same variable names, there could be confusion as to which ones R will be using.

`rm(x)`: with `x` being an object name, removes those objects from the R environment and will have to be read in again for future use.

GPA data: Addresses

The folder where I will have my files stored is: `"/Users/renaes/OneDrive - University of Idaho/renaes/Documents/Classes/R Class"`. I also have it at `http://webpages.uidaho.edu/~renaes/Data/gpdata.csv` for .csv and `http://webpages.uidaho.edu/~renaes/Data/gpdata.txt` for text file. Because address formats differ between OS, using a web address is easiest, *if* it is available.

GPA data: Working directory

Since I do not want to change the wd here for real, I won't but here is the use of `getwd()`. And I just prefer to use mouse gestures and do it with the menu items. :-)

Mac: `setwd("/Users/renaes/OneDrive - University of Idaho/renaes/Documents/Classes/R Class")`

PC: `setwd("C:/renaes/OneDrive - University of Idaho/renaes/Documents/Classes/R Class")`

One way in Windows to find the address is in the Windows Explorer but you will have to change `\` to `/` in the address.

GPA data: `read.table()`, `read.csv()`

I will have different object names for the different methods reading in the dataset. The file name is `GPAdata`.

```
# gpa.text=read.table('GPAdata.txt',header=T,sep='')
# gpa.csv=read.csv('GPAdata.csv',header=T)
gpa.webtxt=read.table('http://webpages.uidaho.edu/~renaes/Data/gpdata.txt',
                      header=T,sep='')
gpa.webcsv=read.csv('http://webpages.uidaho.edu/~renaes/Data/gpdata.csv',
                    header=T)

# we will look at just one since they are all the same dataset
gpa.webcsv
```

```
      UIGPA ACT SEX HOUSING COLLEGE
1      4.00 32  F         R        COS
```

2	4.00	29	F	R	CNR
3	3.00	26	M	T	COS
4	3.80	32	M	G	CLASS
5	3.80	27	F	G	CLASS
6	3.90	23	M	G	CLASS
7	3.90	32	F	T	COS
8	3.05	21	M	T	CALS
9	2.40	NA	M	T	CNR
10	2.75	NA	M	G	CBE
11	2.66	27	M	T	CBE
12	3.00	NA	M	T	CBE
13	2.79	NA	F	G	COS
14	2.90	NA	F	T	CLASS
15	3.00	25	M	T	CALS
16	3.20	29	M	T	COS
17	2.80	18	F	G	CALS
18	3.18	NA	F	G	COS
19	3.80	NA	F	T	CALS
20	3.00	NA	M	T	CBE
21	2.80	17	F	T	CALS
22	2.63	NA	F	G	CBE
23	3.70	NA	F	G	CED
24	2.90	NA	F	T	CNR
25	2.90	NA	F	T	CALS
26	3.30	NA	M	T	COS
27	4.00	32	F	T	CBE
28	3.60	NA	F	T	CBE
29	3.54	NA	F	T	CALS
30	2.80	20	F	T	CBE
31	2.98	29	M	R	CED
32	4.00	24	F	R	CLASS
33	3.51	28	F	T	CLASS
34	3.70	29	M	T	CBE
35	4.00	29	F	T	CLASS
36	3.53	21	M	G	CLASS
37	2.50	29	F	T	CBE
38	3.00	NA	M	R	CNR
39	3.20	NA	F	G	CLASS
40	NA	NA	M	T	CNR
41	3.00	23	M	T	CNR
42	2.30	19	M	T	CED
43	2.90	18	M	T	CNR
44	3.00	22	M	T	CBE
45	3.50	NA	F	T	COS
46	3.60	NA	M	T	CBE
47	3.30	NA	M	T	CLASS
48	4.00	28	M	G	CLASS
49	3.60	21	M	G	CLASS
50	3.40	25	M	G	CBE
51	3.57	25	F	G	CED
52	3.90	NA	F	G	CBE
53	3.67	NA	M	G	COS
54	3.20	NA	M	G	COS
55	3.49	NA	M	G	CBE

56	3.30	25	M	G	CBE
57	3.20	26	M	T	CNR
58	3.80	NA	F	R	CLASS
59	3.10	NA	M	G	CNR
60	3.13	22	M	G	CALS
61	2.40	27	F	T	COS
62	3.20	29	F	G	CBE
63	3.10	29	F	G	CBE
64	3.00	NA	M	T	CLASS
65	3.29	NA	F	T	CLASS
66	2.70	NA	F	R	CLASS
67	3.30	NA	M	T	CNR
68	3.27	NA	M	T	CNR
69	3.60	NA	M	T	CBE
70	3.03	19	F	T	CALS
71	3.44	25	F	T	CLASS
72	1.80	19	M	R	COS
73	3.00	23	M	T	COS
74	3.10	22	F	T	COS
75	2.97	17	F	T	CALS
76	3.50	25	M	T	COS
77	3.25	24	F	T	CED
78	2.77	NA	F	G	CLASS
79	4.00	23	F	G	CBE
80	2.93	NA	M	T	CLASS
81	3.40	27	M	G	CBE
82	3.76	29	M	G	CBE
83	4.00	27	M	G	CED
84	3.60	NA	F	R	CALS
85	3.80	19	M	G	CNR
86	NA	27	F	R	CLASS
87	3.85	NA	M	G	CALS
88	3.81	20	M	G	CALS
89	3.60	NA	F	R	CLASS
90	4.00	27	F	T	CBE
91	4.00	NA	F	T	CBE
92	3.95	NA	F	G	COS
93	4.00	NA	F	R	CLASS
94	3.98	32	F	R	CBE
95	2.96	NA	F	T	CBE
96	3.64	NA	F	G	CLASS
97	3.20	18	F	G	<NA>
98	3.43	19	F	G	CBE
99	2.35	22	M	T	CBE
100	3.50	28	F	T	CLASS
101	3.50	NA	F	T	CLASS
102	3.70	29	F	T	CBE
103	2.40	NA	F	G	CNR
104	3.27	NA	F	G	COS
105	3.43	26	F	G	CED
106	3.50	NA	F	G	CALS
107	3.58	28	F	G	CLASS
108	3.44	24	F	G	CBE
109	2.20	32	M	T	CBE

110	3.30	28	F	G	CALS
111	3.30	24	F	G	CLASS
112	2.57	NA	M	T	CLASS
113	4.00	NA	M	R	CBE
114	4.00	NA	F	T	CALS
115	3.50	NA	M	T	CBE
116	3.35	NA	F	R	CLASS
117	4.00	NA	F	R	CLASS
118	4.00	34	M	R	CBE
119	2.91	18	M	R	CLASS
120	3.00	21	M	T	CNR
121	4.00	28	M	T	CBE
122	2.65	22	M	R	CLASS
123	3.50	30	M	T	COS
124	3.80	26	M	R	COS
125	3.50	32	M	T	CBE
126	3.20	23	M	T	CBE
127	3.30	24	M	G	CLASS
128	3.40	24	M	G	CBE
129	3.30	30	M	T	CNR
130	3.00	27	M	T	COS
131	3.30	22	F	G	CBE
132	1.98	24	M	R	CBE
133	NA	36	F	T	COS
134	2.75	25	M	T	CBE
135	3.40	28	M	T	CBE
136	NA	21	F	G	CLASS
137	3.20	26	M	R	CLASS
138	3.92	NA	F	T	COS
139	3.50	24	F	G	CBE
140	3.80	29	M	R	COS
141	4.00	34	F	R	CLASS
142	2.50	33	M	T	CBE
143	3.00	NA	M	T	COS
144	2.90	27	M	T	CBE
145	2.00	28	M	T	COS
146	NA	29	M	G	CLASS
147	3.50	19	F	T	CLASS
148	3.34	25	M	T	CNR
149	3.35	24	M	R	CALS
150	2.90	19	F	T	CLASS
151	3.50	32	F	T	CALS
152	2.20	26	M	T	CED
153	3.00	23	M	G	CBE
154	2.90	28	M	T	CBE
155	2.50	NA	M	T	CNR
156	3.75	29	M	R	CLASS
157	3.65	NA	F	T	COS
158	3.51	24	M	G	CNR
159	3.00	23	M	T	CBE
160	3.00	21	M	G	CLASS
161	3.75	21	M	T	CBE
162	3.90	20	F	T	CNR
163	3.40	18	F	T	CLASS

164	2.70	20	F	T	CALS
165	3.10	NA	F	R	CLASS
166	3.05	NA	F	R	CBE
167	4.00	36	M	T	CBE
168	3.40	30	M	G	CBE
169	4.00	27	F	G	CED
170	4.00	23	F	G	CLASS
171	3.67	33	F	T	COS
172	3.62	NA	F	T	CLASS
173	4.00	21	F	T	CLASS
174	2.80	28	M	G	COS
175	3.91	33	F	T	CBE
176	4.00	25	F	R	COS
177	4.00	19	M	G	CBE
178	3.66	27	M	G	COS
179	3.26	25	M	T	COS
180	NA	26	M	G	COS
181	3.30	29	M	T	CBE
182	3.47	NA	F	R	COS
183	3.51	25	M	R	CBE
184	3.64	26	F	G	CED
185	3.00	24	F	G	CLASS
186	3.50	25	M	T	CBE
187	3.72	30	M	R	CBE
188	2.72	27	M	T	CLASS
189	3.41	27	F	T	COS
190	3.50	28	F	T	CALS
191	3.35	28	M	T	CNR
192	3.62	22	F	G	CLASS
193	3.66	24	F	T	CBE
194	3.66	21	F	T	CALS
195	3.20	23	F	T	CALS
196	2.92	23	F	T	CALS
197	2.50	NA	F	T	CLASS
198	1.80	28	M	T	CNR
199	3.47	32	M	T	CNR
200	3.00	20	M	T	CBE
201	3.40	21	F	T	CLASS
202	3.30	20	F	T	CLASS
203	3.00	25	M	R	CALS
204	2.90	21	F	T	CALS
205	NA	31	F	R	CLASS
206	3.60	NA	F	G	CBE
207	NA	23	M	T	COS
208	NA	26	F	T	CBE
209	NA	19	F	R	CBE
210	3.70	28	F	G	COS
211	3.30	32	F	R	COS
212	3.50	21	M	T	CBE
213	3.10	20	F	T	CLASS
214	3.00	29	F	T	CBE
215	3.00	35	F	T	CLASS
216	2.35	26	F	T	CLASS
217	2.68	29	M	G	CBE

```

218  2.90  25  F      T      COS
219  2.25  33  M      T      CLASS
220  3.40  26  M      T      CBE
221  3.60  22  F      G      CBE
222  3.00  23  M      G      CBE
223  3.10  24  M      G      CLASS
224  3.50  24  M      G      CALS
225  2.60  36  M      T      CBE
226  3.43  28  M      T      CALS
227  3.64  18  M      G      CNR

```

GPA data: attach()

I will use `attach()` (and `with()`) in examples; `attach(gpa.webtxt)`

```

attach(gpa.webtxt)
# show names of objects in dataframe
names(gpa.webtxt)

```

```
[1] "UIGPA" "ACT" "SEX" "HOUSING" "COLLEGE"
```

```

# show one variable with just one-level name
UIGPA

```

```

[1] 4.00 4.00 3.00 3.80 3.80 3.90 3.90 3.05 2.40 2.75 2.66 3.00 2.79 2.90 3.00
[16] 3.20 2.80 3.18 3.80 3.00 2.80 2.63 3.70 2.90 2.90 3.30 4.00 3.60 3.54 2.80
[31] 2.98 4.00 3.51 3.70 4.00 3.53 2.50 3.00 3.20 NA 3.00 2.30 2.90 3.00 3.50
[46] 3.60 3.30 4.00 3.60 3.40 3.57 3.90 3.67 3.20 3.49 3.30 3.20 3.80 3.10 3.13
[61] 2.40 3.20 3.10 3.00 3.29 2.70 3.30 3.27 3.60 3.03 3.44 1.80 3.00 3.10 2.97
[76] 3.50 3.25 2.77 4.00 2.93 3.40 3.76 4.00 3.60 3.80 NA 3.85 3.81 3.60 4.00
[91] 4.00 3.95 4.00 3.98 2.96 3.64 3.20 3.43 2.35 3.50 3.50 3.70 2.40 3.27 3.43
[106] 3.50 3.58 3.44 2.20 3.30 3.30 2.57 4.00 4.00 3.50 3.35 4.00 4.00 2.91 3.00
[121] 4.00 2.65 3.50 3.80 3.50 3.20 3.30 3.40 3.30 3.00 3.30 1.98 NA 2.75 3.40
[136] NA 3.20 3.92 3.50 3.80 4.00 2.50 3.00 2.90 2.00 NA 3.50 3.34 3.35 2.90
[151] 3.50 2.20 3.00 2.90 2.50 3.75 3.65 3.51 3.00 3.00 3.75 3.90 3.40 2.70 3.10
[166] 3.05 4.00 3.40 4.00 4.00 3.67 3.62 4.00 2.80 3.91 4.00 4.00 3.66 3.26 NA
[181] 3.30 3.47 3.51 3.64 3.00 3.50 3.72 2.72 3.41 3.50 3.35 3.62 3.66 3.66 3.20
[196] 2.92 2.50 1.80 3.47 3.00 3.40 3.30 3.00 2.90 NA 3.60 NA NA NA 3.70
[211] 3.30 3.50 3.10 3.00 3.00 2.35 2.68 2.90 2.25 3.40 3.60 3.00 3.10 3.50 2.60
[226] 3.43 3.64

```

GPA data: with()

Using `with(gpa.webcsv)`. There will not be prep work when using `with()` like there was using `attach()`.

```

# show one variable with just one-level name
with(gpa.webcsv, UIGPA)

```

```

[1] 4.00 4.00 3.00 3.80 3.80 3.90 3.90 3.05 2.40 2.75 2.66 3.00 2.79 2.90 3.00
[16] 3.20 2.80 3.18 3.80 3.00 2.80 2.63 3.70 2.90 2.90 3.30 4.00 3.60 3.54 2.80
[31] 2.98 4.00 3.51 3.70 4.00 3.53 2.50 3.00 3.20 NA 3.00 2.30 2.90 3.00 3.50
[46] 3.60 3.30 4.00 3.60 3.40 3.57 3.90 3.67 3.20 3.49 3.30 3.20 3.80 3.10 3.13
[61] 2.40 3.20 3.10 3.00 3.29 2.70 3.30 3.27 3.60 3.03 3.44 1.80 3.00 3.10 2.97
[76] 3.50 3.25 2.77 4.00 2.93 3.40 3.76 4.00 3.60 3.80 NA 3.85 3.81 3.60 4.00

```



```

[91] 4.00 3.95 4.00 3.98 2.96 3.64 3.20 3.43 2.35 3.50 3.50 3.70 2.40 3.27 3.43
[106] 3.50 3.58 3.44 2.20 3.30 3.30 2.57 4.00 4.00 3.50 3.35 4.00 4.00 2.91 3.00
[121] 4.00 2.65 3.50 3.80 3.50 3.20 3.30 3.40 3.30 3.00 3.30 1.98 NA 2.75 3.40
[136] NA 3.20 3.92 3.50 3.80 4.00 2.50 3.00 2.90 2.00 NA 3.50 3.34 3.35 2.90
[151] 3.50 2.20 3.00 2.90 2.50 3.75 3.65 3.51 3.00 3.00 3.75 3.90 3.40 2.70 3.10
[166] 3.05 4.00 3.40 4.00 4.00 3.67 3.62 4.00 2.80 3.91 4.00 4.00 3.66 3.26 NA
[181] 3.30 3.47 3.51 3.64 3.00 3.50 3.72 2.72 3.41 3.50 3.35 3.62 3.66 3.66 3.20
[196] 2.92 2.50 1.80 3.47 3.00 3.40 3.30 3.00 2.90 NA 3.60 NA NA NA 3.70
[211] 3.30 3.50 3.10 3.00 3.00 2.35 2.68 2.90 2.25 3.40 3.60 3.00 3.10 3.50 2.60
[226] 3.43 3.64

```

GPA data analyses: plot

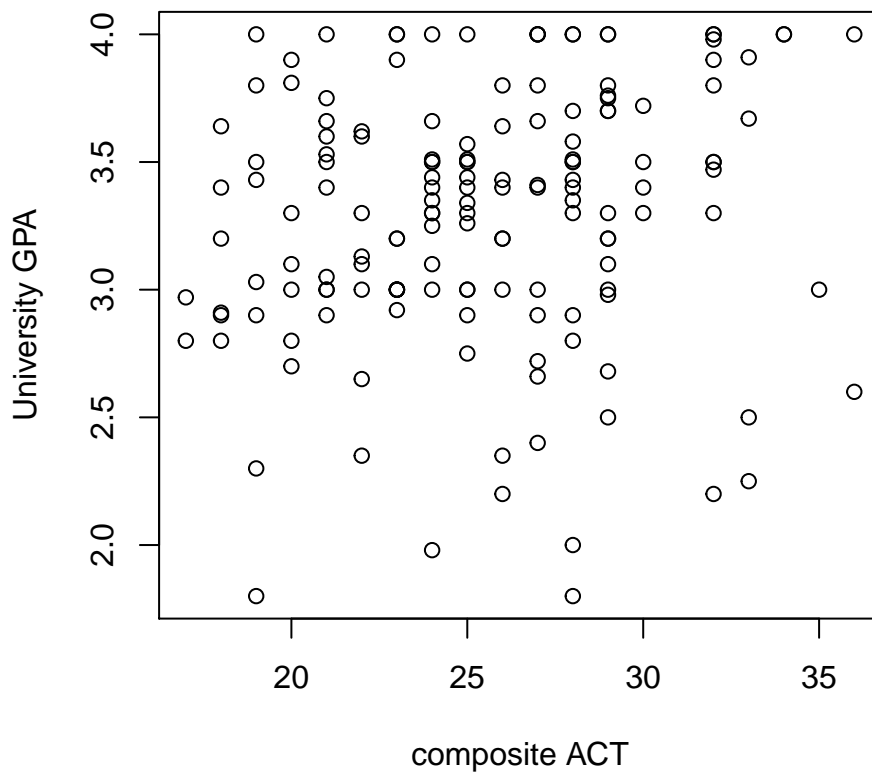
Looking at the relationship between college GPAs and ACT scores.

```

plot(UIGPA~ACT,main='Scatterplot of College GPA and ACT score',
     xlab='composite ACT',ylab='University GPA')

```

Scatterplot of College GPA and ACT score



```

# with(gpa.webcsv,plot(UIGPA~ACT,main='Scatterplot of College GPA and ACT score',xlab='composite ACT',
# ylab='University GPA'))

```

GPA data analyses: prepping data I

Here we will look at separating the dataset into males and females, and then later store into two separate files.

```
uGPA.male=UIGPA[SEX=='M']
ACT.male=ACT[SEX=='M']
HOUSING.male=HOUSING[SEX=='M']
COLLEGE.male=COLLEGE[SEX=='M']
# create dataframe
GPA.male=data.frame(uGPA.male,ACT.male,HOUSING.male,COLLEGE.male)
GPA.male
```

	uGPA.male	ACT.male	HOUSING.male	COLLEGE.male
1	3.00	26	T	COS
2	3.80	32	G	CLASS
3	3.90	23	G	CLASS
4	3.05	21	T	CALS
5	2.40	NA	T	CNR
6	2.75	NA	G	CBE
7	2.66	27	T	CBE
8	3.00	NA	T	CBE
9	3.00	25	T	CALS
10	3.20	29	T	COS
11	3.00	NA	T	CBE
12	3.30	NA	T	COS
13	2.98	29	R	CED
14	3.70	29	T	CBE
15	3.53	21	G	CLASS
16	3.00	NA	R	CNR
17	NA	NA	T	CNR
18	3.00	23	T	CNR
19	2.30	19	T	CED
20	2.90	18	T	CNR
21	3.00	22	T	CBE
22	3.60	NA	T	CBE
23	3.30	NA	T	CLASS
24	4.00	28	G	CLASS
25	3.60	21	G	CLASS
26	3.40	25	G	CBE
27	3.67	NA	G	COS
28	3.20	NA	G	COS
29	3.49	NA	G	CBE
30	3.30	25	G	CBE
31	3.20	26	T	CNR
32	3.10	NA	G	CNR
33	3.13	22	G	CALS
34	3.00	NA	T	CLASS
35	3.30	NA	T	CNR
36	3.27	NA	T	CNR
37	3.60	NA	T	CBE
38	1.80	19	R	COS
39	3.00	23	T	COS
40	3.50	25	T	COS
41	2.93	NA	T	CLASS

42	3.40	27	G	CBE
43	3.76	29	G	CBE
44	4.00	27	G	CED
45	3.80	19	G	CNR
46	3.85	NA	G	CALS
47	3.81	20	G	CALS
48	2.35	22	T	CBE
49	2.20	32	T	CBE
50	2.57	NA	T	CLASS
51	4.00	NA	R	CBE
52	3.50	NA	T	CBE
53	4.00	34	R	CBE
54	2.91	18	R	CLASS
55	3.00	21	T	CNR
56	4.00	28	T	CBE
57	2.65	22	R	CLASS
58	3.50	30	T	COS
59	3.80	26	R	COS
60	3.50	32	T	CBE
61	3.20	23	T	CBE
62	3.30	24	G	CLASS
63	3.40	24	G	CBE
64	3.30	30	T	CNR
65	3.00	27	T	COS
66	1.98	24	R	CBE
67	2.75	25	T	CBE
68	3.40	28	T	CBE
69	3.20	26	R	CLASS
70	3.80	29	R	COS
71	2.50	33	T	CBE
72	3.00	NA	T	COS
73	2.90	27	T	CBE
74	2.00	28	T	COS
75	NA	29	G	CLASS
76	3.34	25	T	CNR
77	3.35	24	R	CALS
78	2.20	26	T	CED
79	3.00	23	G	CBE
80	2.90	28	T	CBE
81	2.50	NA	T	CNR
82	3.75	29	R	CLASS
83	3.51	24	G	CNR
84	3.00	23	T	CBE
85	3.00	21	G	CLASS
86	3.75	21	T	CBE
87	4.00	36	T	CBE
88	3.40	30	G	CBE
89	2.80	28	G	COS
90	4.00	19	G	CBE
91	3.66	27	G	COS
92	3.26	25	T	COS
93	NA	26	G	COS
94	3.30	29	T	CBE
95	3.51	25	R	CBE

96	3.50	25	T	CBE
97	3.72	30	R	CBE
98	2.72	27	T	CLASS
99	3.35	28	T	CNR
100	1.80	28	T	CNR
101	3.47	32	T	CNR
102	3.00	20	T	CBE
103	3.00	25	R	CALS
104	NA	23	T	COS
105	3.50	21	T	CBE
106	2.68	29	G	CBE
107	2.25	33	T	CLASS
108	3.40	26	T	CBE
109	3.00	23	G	CBE
110	3.10	24	G	CLASS
111	3.50	24	G	CALS
112	2.60	36	T	CBE
113	3.43	28	T	CALS
114	3.64	18	G	CNR

GPA data analyses: prepping data II

Here we will look at separating the dataset into males and females, and then later store into two separate files.

```
uGPA.f=UIGPA[SEX=='F']
ACT.f=ACT[SEX=='F']
HOUSING.f=HOUSING[SEX=='F']
COLLEGE.f=COLLEGE[SEX=='F']
# create dataframe
GPA.female=data.frame(uGPA.f,ACT.f,HOUSING.f,COLLEGE.f)
GPA.female
```

	uGPA.f	ACT.f	HOUSING.f	COLLEGE.f
1	4.00	32	R	COS
2	4.00	29	R	CNR
3	3.80	27	G	CLASS
4	3.90	32	T	COS
5	2.79	NA	G	COS
6	2.90	NA	T	CLASS
7	2.80	18	G	CALS
8	3.18	NA	G	COS
9	3.80	NA	T	CALS
10	2.80	17	T	CALS
11	2.63	NA	G	CBE
12	3.70	NA	G	CED
13	2.90	NA	T	CNR
14	2.90	NA	T	CALS
15	4.00	32	T	CBE
16	3.60	NA	T	CBE
17	3.54	NA	T	CALS
18	2.80	20	T	CBE
19	4.00	24	R	CLASS

20	3.51	28	T	CLASS
21	4.00	29	T	CLASS
22	2.50	29	T	CBE
23	3.20	NA	G	CLASS
24	3.50	NA	T	COS
25	3.57	25	G	CED
26	3.90	NA	G	CBE
27	3.80	NA	R	CLASS
28	2.40	27	T	COS
29	3.20	29	G	CBE
30	3.10	29	G	CBE
31	3.29	NA	T	CLASS
32	2.70	NA	R	CLASS
33	3.03	19	T	CALS
34	3.44	25	T	CLASS
35	3.10	22	T	COS
36	2.97	17	T	CALS
37	3.25	24	T	CED
38	2.77	NA	G	CLASS
39	4.00	23	G	CBE
40	3.60	NA	R	CALS
41	NA	27	R	CLASS
42	3.60	NA	R	CLASS
43	4.00	27	T	CBE
44	4.00	NA	T	CBE
45	3.95	NA	G	COS
46	4.00	NA	R	CLASS
47	3.98	32	R	CBE
48	2.96	NA	T	CBE
49	3.64	NA	G	CLASS
50	3.20	18	G	<NA>
51	3.43	19	G	CBE
52	3.50	28	T	CLASS
53	3.50	NA	T	CLASS
54	3.70	29	T	CBE
55	2.40	NA	G	CNR
56	3.27	NA	G	COS
57	3.43	26	G	CED
58	3.50	NA	G	CALS
59	3.58	28	G	CLASS
60	3.44	24	G	CBE
61	3.30	28	G	CALS
62	3.30	24	G	CLASS
63	4.00	NA	T	CALS
64	3.35	NA	R	CLASS
65	4.00	NA	R	CLASS
66	3.30	22	G	CBE
67	NA	36	T	COS
68	NA	21	G	CLASS
69	3.92	NA	T	COS
70	3.50	24	G	CBE
71	4.00	34	R	CLASS
72	3.50	19	T	CLASS
73	2.90	19	T	CLASS

74	3.50	32	T	CALS
75	3.65	NA	T	COS
76	3.90	20	T	CNR
77	3.40	18	T	CLASS
78	2.70	20	T	CALS
79	3.10	NA	R	CLASS
80	3.05	NA	R	CBE
81	4.00	27	G	CED
82	4.00	23	G	CLASS
83	3.67	33	T	COS
84	3.62	NA	T	CLASS
85	4.00	21	T	CLASS
86	3.91	33	T	CBE
87	4.00	25	R	COS
88	3.47	NA	R	COS
89	3.64	26	G	CED
90	3.00	24	G	CLASS
91	3.41	27	T	COS
92	3.50	28	T	CALS
93	3.62	22	G	CLASS
94	3.66	24	T	CBE
95	3.66	21	T	CALS
96	3.20	23	T	CALS
97	2.92	23	T	CALS
98	2.50	NA	T	CLASS
99	3.40	21	T	CLASS
100	3.30	20	T	CLASS
101	2.90	21	T	CALS
102	NA	31	R	CLASS
103	3.60	NA	G	CBE
104	NA	26	T	CBE
105	NA	19	R	CBE
106	3.70	28	G	COS
107	3.30	32	R	COS
108	3.10	20	T	CLASS
109	3.00	29	T	CBE
110	3.00	35	T	CLASS
111	2.35	26	T	CLASS
112	2.90	25	T	COS
113	3.60	22	G	CBE

GPA data analyses: writing out to file

```
write.csv(GPA.female, '/Volumes/shared/webpages/~renaes/Data/gpafemale.csv',
          row.names=F)
write.table(GPA.male, '/Volumes/shared/webpages/~renaes/Data/GPAmale.txt',
            sep=' ', row.names=F)
```

Cleanup at the end of session or data use

Use `detach()` if `attach()` was used and then use `rm()` to remove the data from your environment if it will not be used again in the same session.

```
detach(gpa.webtxt)
rm(GPA.female)
rm(GPA.male)
rm(gpa.webtxt)
rm(gpa.webcsv)
```