

Problem Set #1

Marwin Carmo

2023-10-05

Overview

Welcome to Data Cleaning and Management using R! This problem set is intended to give you some practice becoming familiar with using R. In this problem set, I'm asking you to: create an R project; render to pdf; load and investigate an R data frame that is stored on the web; and apply some basic functions to atomic vectors.

- Note: Change the values of the YAML header above to your name and the date.

Question 1: Creating an R project

Create an R project

- Create a folder where you want to save files associated with problem set 1. Let's call that folder "problemset1", but you can name it whatever you want.
 - For instance, it could be psc290-fq23 » problem-sets » problemset1.
- In RStudio, click on "File" » "New Project" » "Existing Directory" » "Browse".
- Browse to find and select your problem set 1 folder.
- Click on "Create Project".
 - An R project file has the extension ".Rproj".
 - The name of the file should be "problemset1.Rproj", or whatever you named the folder.

Save this problemset1.Rmd file anywhere in the folder named problemset1.

- Use this naming convention "lastname-firstname-ps#" for your .qmd files (e.g. beck_emorye_ps1.qmd).
 - If you want, you can change the name of this file to include your first and last name.
- Run the `getwd()` function and the `list.files()` function in the code chunk below.

- What is the output? Why?

```
getwd()
list.files()
```

ANSWER:

```
[1] "C:/Users/marwin/Documents/Rprojects/phd/23-Fall/psc290-fq23/problem-sets/problemset1"
[1] "carmo_marwin_ps1.qmd" "problemset1.Rproj"
```

The first line of the output shows the current working directory of the R process. It was given by the function `getwd()`. The second output line says which files are in the current working directory and was produced by the function `list.files()`. There are only two: this .qmd file and a shortcut to the problem set 1 project.

Question 2: Knit to pdf

- At the top of this .qmd file, type in your first and last name in the appropriate place in the YAML header (e.g. "Hadley Wickham").
- In the date field of the YAML header, insert the date within quotations (any date format is fine).
- Now click the "Render" button near the top of your RStudio window (icon with blue yarn ball).
 - Alternatively you can use the shortcut: **Cmd/Ctrl + Shift + k**.
 - *Note:* One goal of this assignment is to make sure you are able to render to a PDF without running into errors.

Question 3: Load .Rdata directly with url and then investigate the data frame

1. This question asks you to load a dataframe by specifying the `url()` function within the `load()` function.
 - Url link for data frame: <https://github.com/emoriebeck/psc290-data-FQ23/raw/main/05-assignments/01-ps1/pwe-ps1-small.RData>
 - Hint: to load .Rdata use the `load()` and `url()` functions because you are using a link. follow this approach: `load(url("url_link"))`.
 - * Note: the `url_link` is put within quotes

Load the dataframe within this code chunk below.

```
load(url("https://github.com/emoriebeck/psc290-data-FQ23/raw/main/05-assignments/01-ps1/pwe-ps1-small.RData"))
```

2. Print the data frame `df_pwe` by typing its name.

```
df_pwe
```

	country	introelapse	testelapse	surveyelapse	TIPI1	TIPI2	TIPI3	TIPI4	TIPI5
1	US	1118	103	111	2	6	1	7	7
2	GB	8	116	128	3	2	6	1	5
3	US	9	142	217	5	6	7	5	5
4	US	2	79	90	2	5	1	2	3
5	VN	545	210	276	3	1	6	5	5
6	IN	4397	625	79	4	5	2	6	7
7	NO	2	181	136	3	5	3	6	7
8	NONE	4	208	198	5	6	2	5	7
9	PL	13	191	266	6	7	4	3	5
10	IN	37	394	380	7	1	7	1	1
11	NO	2	99	123	2	6	5	6	5
12	AU	11	213	199	6	2	6	2	5
13	US	2	95	143	3	2	5	6	7
14	IN	50	133	139	3	2	2	2	7
15	DE	7	112	77	1	5	5	3	6
16	PL	14	134	147	5	7	6	3	7
17	ZA	262	262	148	7	5	5	3	6
18	US	679	151	192	6	2	6	2	6
19	NZ	309	207	5	7	1	7	1	6
20	US	13	124	190	1	3	3	7	6
21	AU	7	123	111	3	6	3	7	7
22	CH	2	116	104	5	5	6	1	7
23	RS	287	206	120	2	3	6	5	6
24	RS	12	378	257	7	7	7	2	6
25	MY	147	77	108	3	4	7	2	4
26	CA	58	138	142	3	2	6	6	7
27	AU	56	109	128	4	5	5	3	6
28	CA	5	100	1082	1	5	4	6	7
29	GB	5	149	178	5	5	1	6	6
30	US	3	257	152	6	6	3	5	7
31	NONE	43	173	216	6	2	6	3	6
32	US	20	79	94	5	5	5	7	6
33	DE	2	100	148	4	5	6	6	7
34	FR	3	87	109	2	1	2	2	6
35	US	2	127	101	2	3	2	4	7
36	DE	1185	142	79	3	6	6	2	4
37	NZ	23	90	215	3	5	5	3	6

38	US	10	91	201	5	6	6	2	6
39	FI	273	93	55	0	0	0	0	0
40	US	7	77	91	1	5	3	5	6
41	US	1	140	96	6	1	6	1	7
42	PL	2	144	81	2	3	5	2	7
43	US	2	67	89	6	5	6	6	6
44	US	3	129	152	6	6	6	2	5
45	CA	2309	142	103	1	6	3	7	5
46	US	5	204	155	6	4	6	2	7
47	US	8	78	369	3	3	6	5	5
48	US	15	521	342	5	2	7	2	6
49	PL	7	175	180	2	6	6	4	6
50	US	2	56	92	1	5	3	5	2
51	US	963503	53	100	6	5	7	5	5
52	US	2	82	91	2	2	6	4	7
53	GB	6	227	320	5	5	2	1	7
54	NONE	82	224	211	5	2	6	1	7
55	US	13	141	125	1	4	7	6	5
56	US	96	141	115	7	5	4	5	5
57	US	33	126	237	2	7	5	7	5
58	US	20	102	72	5	6	6	5	6
59	BD	2389	25	133	2	2	2	3	4
60	GB	18	169	204	2	1	7	4	7
61	US	4	178	130	6	2	5	6	7
62	US	16	177	124	2	3	4	3	7
63	US	4	130	149	2	1	6	1	7
64	DE	4	170	192	3	6	5	3	6
65	GB	3	101	211	2	2	5	6	7
66	US	7	99	112	1	6	7	6	5
67	US	1	76	83	1	6	7	7	7
68	AU	16	128	141	2	1	4	6	4
69	RS	494	140	162	1	5	7	1	6
70	US	3	134	88	3	5	6	5	5
71	DE	2639	201	157	3	2	3	6	6
72	US	41789	130	88	3	5	5	3	3
73	US	4	129	161	3	2	5	4	7
74	PL	8	66	109	7	6	7	5	7
75	US	8852	105	212186	5	7	2	6	5
76	US	22	663	212	5	6	4	3	6
77	US	2	85	99	3	1	3	6	7
78	US	342	171	217	2	5	6	5	5
79	US	6	126	129	7	5	3	6	7
80	US	36	193	175	2	1	5	7	4

81	TH	8	95	113	6	2	5	5	6
82	RS	25	167	154	5	6	4	2	7
83	IT	2	150	212	7	5	7	4	7
84	NONE	25	210	208	6	1	7	2	7
85	US	30	86	131	3	2	6	1	7
86	US	3	103	136	5	2	1	6	6
87	US	284	272	137	7	3	6	2	6
88	US	4	81	116	6	3	7	7	6
89	GB	3	103	172	2	5	1	5	6
90	GB	4	134	189	1	3	6	5	6
91	US	2	245	227	6	3	7	7	2
92	US	3	330	173	5	4	7	5	7
93	US	18	115	111	3	6	5	3	6
94	US	11	270	280	2	4	7	1	7
95	US	141	282	162	6	2	7	1	7
96	GB	3	84	96	3	2	7	6	6
97	US	3	125	164	3	4	6	2	7
98	NONE	7	112	117	2	1	5	5	4
99	ZA	7	180	535	7	2	7	2	7
100	US	7	117	120	6	5	5	4	6

	TIIPI6	TIIPI7	TIIPI8	TIIPI9	TIIPI10	VCL1	VCL2	VCL3	VCL4	VCL5	VCL6	VCL7	VCL8	VCL9
1	7	7	7	1	1	1	1	0	1	1	0	0	0	0
2	6	5	2	6	5	1	1	1	1	1	0	1	1	0
3	5	1	1	6	7	1	1	1	1	1	0	0	0	0
4	5	1	7	5	3	1	1	0	0	1	0	0	0	0
5	6	7	1	6	3	1	1	0	1	1	0	1	0	0
6	4	5	6	3	2	1	0	0	1	1	0	0	1	0
7	5	5	5	5	1	1	1	0	1	1	0	0	1	0
8	3	7	1	6	2	0	1	0	1	0	1	0	1	0
9	1	5	6	6	4	1	0	0	1	1	0	0	1	0
10	6	7	1	7	4	0	0	0	0	1	0	0	0	0
11	7	7	6	4	5	1	0	0	1	1	0	0	0	0
12	1	5	3	6	0	1	1	1	1	1	0	1	0	0
13	5	6	5	2	2	1	1	1	1	1	0	0	1	0
14	5	6	6	2	1	0	0	0	1	1	0	0	0	0
15	7	5	3	6	1	1	0	0	0	1	0	0	0	0
16	5	7	5	6	1	1	1	0	1	0	0	0	0	0
17	2	5	5	3	1	1	1	0	1	1	0	0	0	0
18	5	6	2	6	2	1	0	0	1	1	0	0	0	0
19	3	7	1	7	1	1	1	1	1	1	1	0	1	1
20	7	3	3	1	2	1	1	1	1	1	0	0	0	0
21	5	7	3	2	6	1	1	1	1	1	0	1	1	0
22	4	4	2	7	2	1	1	1	1	1	1	1	1	0

23	7	7	5	5	3	1	1	0	1	1	0	0	1	0
24	2	5	1	6	1	1	0	0	1	0	0	0	1	0
25	6	2	2	5	1	1	1	1	1	1	0	1	1	0
26	3	7	2	3	1	1	1	0	1	1	0	0	1	0
27	4	5	2	6	1	1	1	1	1	1	0	1	0	0
28	4	5	6	2	1	1	1	1	1	1	0	1	1	0
29	3	4	7	2	3	1	1	1	1	1	0	0	1	0
30	2	3	5	6	2	1	1	1	1	1	0	0	1	0
31	4	7	2	6	1	1	1	0	1	1	0	0	0	0
32	7	5	3	3	3	1	1	1	1	1	0	1	0	0
33	2	7	2	1	1	1	1	0	1	1	0	0	1	0
34	6	5	7	5	2	1	1	1	1	1	0	0	1	0
35	7	1	6	7	2	1	1	1	1	1	0	0	1	0
36	4	2	6	6	6	1	1	1	1	1	0	1	0	0
37	6	4	3	6	3	1	1	1	1	1	0	1	1	0
38	5	2	5	6	3	1	1	1	1	1	0	1	0	0
39	0	0	0	0	0	1	1	0	1	1	0	0	1	0
40	7	3	5	6	3	1	1	1	1	1	0	1	1	0
41	5	7	3	6	1	1	1	1	1	1	1	1	1	1
42	6	5	4	5	1	1	1	0	1	1	0	0	1	0
43	4	6	3	2	1	1	1	1	1	1	0	1	0	0
44	3	7	3	6	1	1	1	1	1	1	0	1	1	0
45	7	5	5	3	2	1	1	0	1	1	0	0	1	0
46	4	6	4	6	4	1	1	0	1	1	0	0	1	1
47	5	6	6	3	3	1	1	0	1	1	0	0	0	0
48	6	6	1	7	4	1	1	0	1	1	0	0	0	0
49	2	2	2	4	1	1	1	0	1	1	1	0	1	0
50	7	1	5	2	4	1	1	1	1	1	1	1	1	0
51	1	1	3	5	1	1	1	1	1	0	0	0	1	0
52	7	7	4	4	7	1	1	1	1	1	0	1	1	0
53	1	6	6	6	2	1	1	1	1	1	0	1	1	0
54	6	6	1	6	2	1	0	0	1	0	0	0	0	0
55	6	5	2	3	5	1	1	1	1	1	0	0	1	0
56	1	1	4	4	1	1	1	0	1	1	0	0	1	0
57	7	6	5	7	2	1	1	1	1	1	0	0	1	0
58	4	6	3	6	5	1	0	0	1	1	0	0	1	1
59	3	4	4	7	5	1	0	0	0	0	0	0	0	0
60	7	7	6	5	1	1	1	1	1	1	0	1	1	0
61	5	6	5	6	1	1	1	1	1	1	0	1	1	0
62	6	5	5	5	1	1	1	1	1	1	0	0	0	0
63	7	7	5	6	5	1	1	0	1	1	0	0	0	0
64	3	6	5	5	6	1	1	0	1	1	0	1	1	0
65	3	7	6	2	1	1	1	1	1	1	0	1	1	0

66	5	6	1	5	2	1	1	1	1	1	0	1	1	0
67	2	6	7	2	1	1	1	1	1	1	0	1	1	0
68	7	6	5	6	4	1	1	1	1	1	0	1	1	0
69	7	4	1	6	2	1	1	0	1	1	0	0	0	0
70	5	4	5	5	5	1	1	1	1	0	0	1	0	0
71	6	7	2	5	1	1	1	1	1	1	0	1	1	0
72	6	6	1	6	5	1	1	1	1	1	0	0	1	0
73	5	6	5	4	1	1	1	1	1	1	0	1	1	0
74	1	5	1	6	5	1	1	1	1	1	0	1	1	0
75	5	3	6	4	2	1	1	1	1	1	0	1	1	0
76	3	6	4	5	1	1	1	0	1	1	0	0	0	0
77	7	6	3	2	1	1	1	1	1	1	0	0	0	0
78	5	6	3	5	3	1	1	1	1	1	0	0	1	0
79	1	6	7	5	1	1	1	0	1	1	0	0	0	0
80	7	2	6	3	5	1	1	1	1	1	0	1	1	0
81	2	7	5	5	3	1	1	0	1	1	0	0	0	0
82	2	5	6	4	1	1	1	0	1	1	0	0	1	0
83	7	7	4	5	1	1	1	1	1	1	0	1	0	0
84	3	6	1	7	2	1	0	1	1	0	0	0	1	1
85	4	7	1	7	2	1	1	1	1	1	0	0	1	0
86	7	6	5	2	3	1	0	0	1	1	0	0	0	0
87	2	5	2	6	4	1	1	0	1	1	0	0	0	0
88	4	5	3	6	2	1	1	1	1	1	0	0	1	0
89	6	6	6	7	3	1	1	0	1	1	0	0	0	0
90	7	2	1	7	4	1	1	1	1	1	0	1	1	0
91	4	7	4	5	1	1	1	0	1	1	0	0	0	0
92	7	7	2	7	5	0	0	0	0	0	0	0	0	0
93	4	4	5	6	5	1	1	1	1	1	0	1	1	0
94	1	5	1	6	2	1	0	0	1	0	0	0	0	1
95	5	3	2	7	3	1	1	1	1	1	1	1	1	0
96	5	6	1	3	2	1	1	1	1	1	0	1	0	0
97	5	3	2	6	1	1	1	1	1	1	0	0	1	0
98	6	5	5	4	6	1	1	1	1	1	0	1	1	0
99	1	7	1	6	1	1	1	1	1	1	1	1	1	0
100	2	5	5	5	3	1	1	1	1	1	0	1	1	0
VCL10 VCL11 VCL12 VCL13 VCL14 VCL15 VCL16 education urban gender engnat age														
1	1	0	0	1	1	1	1		2	1	2	1	19	
2	1	1	0	1	1	1	1		3	1	2	1	64	
3	1	0	0	1	1	1	1		2	2	2	1	18	
4	0	1	0	1	0	1	1		2	2	1	1	25	
5	1	1	0	0	1	1	1		2	3	2	2	19	
6	1	0	0	1	1	1	1		3	3	1	2	22	
7	1	1	0	1	1	1	1		1	2	2	2	32	

8	1	1	0	0	1	1	1	2	3	2	2	21
9	1	1	0	0	0	1	1	2	3	2	2	19
10	1	0	0	0	0	0	1	4	1	1	2	26
11	1	0	0	1	1	1	1	2	1	1	2	23
12	1	1	0	1	1	1	1	2	2	1	1	60
13	1	1	1	1	1	1	1	3	2	2	1	22
14	1	0	0	0	1	0	1	2	3	1	1	17
15	1	0	0	0	0	1	0	4	2	1	2	23
16	1	0	0	0	1	1	1	2	1	1	2	21
17	1	0	0	1	1	1	1	4	3	1	2	37
18	1	0	0	0	0	1	1	3	3	2	2	33
19	1	0	1	1	1	1	1	3	3	1	1	30
20	1	0	0	0	1	1	1	2	2	2	1	19
21	1	1	0	1	1	1	1	3	2	2	1	29
22	1	0	0	1	1	1	1	3	2	1	1	30
23	1	1	0	1	1	1	1	4	3	1	2	29
24	1	0	0	0	1	1	1	4	3	2	2	41
25	1	1	0	1	1	1	1	3	2	1	1	33
26	1	1	0	1	1	1	1	4	3	2	2	53
27	1	0	0	1	1	1	1	2	2	1	1	34
28	1	1	0	1	1	1	1	3	2	1	1	48
29	1	1	0	1	1	1	1	4	2	1	1	32
30	1	0	0	1	1	1	1	2	2	1	1	21
31	1	0	0	1	1	1	1	2	3	2	2	26
32	1	1	0	1	1	1	1	4	1	1	1	36
33	1	1	0	1	1	1	1	3	2	2	2	35
34	1	1	0	1	1	1	1	3	3	2	2	22
35	1	1	0	1	1	1	1	2	3	1	1	21
36	1	0	0	1	1	1	1	3	0	1	2	43
37	1	1	0	1	1	1	1	3	3	1	1	40
38	1	0	0	1	1	1	1	3	1	1	1	30
39	1	0	0	1	1	1	1	2	2	2	2	20
40	1	1	0	1	1	1	1	3	2	2	1	27
41	1	0	0	1	1	1	1	3	2	2	1	35
42	1	1	0	1	1	1	1	1	3	3	2	18
43	1	0	0	1	1	1	1	3	1	2	1	21
44	1	1	0	1	1	1	1	3	3	1	1	61
45	1	0	1	1	1	1	1	2	2	1	1	18
46	1	0	1	0	1	1	1	2	1	2	1	18
47	1	0	0	1	1	1	1	2	1	2	1	19
48	1	0	0	0	1	1	1	2	2	2	1	22
49	1	1	1	1	1	1	1	3	3	2	2	33
50	1	1	0	1	1	1	1	2	1	1	1	18

51	1	0	0	0	1	1	1	2	3	1	1	18
52	1	1	1	1	1	1	1	3	2	2	1	58
53	1	1	0	1	1	1	1	3	2	2	1	55
54	1	0	0	0	0	1	1	1	3	2	2	35
55	1	0	0	0	0	1	1	3	1	2	1	23
56	1	0	0	1	1	1	1	1	2	1	1	36
57	1	0	0	1	1	1	1	2	3	2	1	22
58	1	0	0	1	1	1	1	1	2	2	1	17
59	0	0	0	0	0	0	0	3	3	1	1	22
60	1	1	1	1	1	1	1	1	3	1	1	45
61	1	1	1	1	1	1	1	3	3	1	1	27
62	1	0	0	1	1	1	1	3	2	1	1	47
63	1	0	0	1	1	1	1	2	2	1	1	45
64	1	0	0	1	1	1	1	3	3	1	2	32
65	1	1	0	1	1	1	1	3	3	1	1	53
66	1	1	0	1	1	1	1	3	2	2	1	49
67	1	0	1	1	1	1	1	3	2	1	1	17
68	1	1	0	1	1	1	1	3	3	1	1	46
69	1	0	0	0	1	1	1	3	2	1	2	20
70	1	1	0	1	1	1	1	4	2	2	1	33
71	1	1	0	1	1	1	1	2	1	3	2	33
72	1	0	1	1	1	1	1	4	3	1	1	32
73	1	1	0	1	1	1	1	3	2	1	1	39
74	1	1	0	1	1	1	1	4	2	2	1	36
75	1	1	0	1	1	1	1	2	2	1	1	24
76	1	0	0	1	0	1	1	4	3	2	1	33
77	1	0	0	1	1	1	1	1	3	2	1	16
78	1	0	0	1	1	1	1	3	2	2	1	23
79	1	0	0	0	1	1	1	2	1	2	1	16
80	1	1	0	1	1	1	1	4	2	1	1	51
81	1	0	0	1	1	1	1	3	2	1	1	32
82	1	1	0	1	1	1	1	2	3	1	2	18
83	1	0	0	1	1	1	1	2	0	1	2	28
84	1	0	0	0	1	1	1	2	3	2	2	41
85	1	0	1	1	1	1	1	3	3	1	1	47
86	1	0	0	0	0	1	1	1	3	2	1	16
87	1	0	0	1	1	1	0	3	2	1	1	36
88	1	0	1	1	1	1	1	3	1	2	1	20
89	1	0	1	1	1	1	1	2	2	1	1	17
90	1	1	0	1	1	1	1	3	1	1	1	46
91	0	0	1	0	1	1	1	1	2	1	1	17
92	0	0	0	0	0	0	0	2	3	2	1	17
93	1	1	1	1	1	1	1	2	2	1	1	31

94	1	0	0	0	1	1	1	2	2	1	2	18
95	1	0	0	1	1	1	1	0	3	1	1	59
96	1	1	0	1	1	1	1	3	2	1	1	42
97	1	1	0	1	1	1	1	1	2	1	1	13
98	1	1	1	1	1	1	1	3	1	2	1	35
99	1	1	0	1	1	1	1	2	2	2	2	37
100	1	1	0	1	1	1	1	3	3	2	1	45

	screenw	screenh	hand	religion	orientation	race	voted	married	familysize
1	1093	615	1	1	1	16	2	1	2
2	768	1024	1	2	1	16	2	3	2
3	1366	768	1	2	1	16	1	1	3
4	1536	864	1	1	1	13	1	1	3
5	1152	720	1	3	1	11	2	1	2
6	1440	900	1	8	1	11	1	1	2
7	768	1024	1	1	1	16	2	1	2
8	412	732	1	2	1	16	1	1	2
9	375	667	1	2	5	16	1	1	1
10	1920	1080	1	8	2	11	1	2	3
11	360	640	1	1	1	16	2	1	4
12	2560	1440	1	2	1	16	2	2	4
13	412	846	1	1	2	16	2	1	2
14	1280	720	1	8	3	17	2	1	2
15	1920	1080	1	12	4	16	2	1	2
16	360	692	2	4	2	16	1	1	3
17	1920	1080	1	7	1	16	1	2	1
18	1440	900	1	4	2	11	2	2	5
19	1536	864	1	12	5	17	1	2	6
20	1366	768	1	7	4	17	1	1	2
21	360	640	1	2	2	16	2	1	2
22	1920	1080	1	12	1	16	2	1	2
23	360	640	1	2	1	16	1	2	2
24	1536	864	1	1	1	16	1	1	2
25	1440	900	1	0	0	10	0	0	0
26	1920	1080	1	2	1	16	2	2	2
27	360	740	2	1	1	16	1	2	3
28	1920	1080	2	1	1	16	1	1	3
29	360	640	1	2	1	16	1	2	2
30	412	846	1	1	2	16	1	1	4
31	360	720	1	7	1	16	1	1	3
32	1920	1080	1	6	1	16	1	2	3
33	768	1024	1	2	3	16	1	2	2
34	360	720	1	1	2	16	1	1	1
35	1440	900	1	1	1	16	2	1	2

36	1920	1080	1	1	1	16	2	1	2
37	360	640	1	4	1	16	2	2	5
38	414	736	1	6	1	16	2	2	2
39	1440	900	1	2	1	16	1	1	1
40	1600	900	2	2	5	16	1	1	2
41	1024	768	2	12	1	16	1	1	3
42	1366	768	1	2	4	16	2	1	1
43	412	732	1	2	5	16	1	1	1
44	1280	800	1	1	3	16	1	1	4
45	1920	1080	2	9	3	16	1	1	2
46	1280	720	1	12	1	16	2	1	3
47	1366	768	1	2	2	16	2	1	3
48	1280	800	1	10	1	16	1	1	2
49	347	347	1	4	0	16	2	1	2
50	1440	900	2	1	1	16	2	1	2
51	360	740	1	9	1	16	2	1	1
52	1600	900	1	1	1	16	1	2	5
53	1280	800	1	6	1	11	2	1	2
54	1920	1080	1	7	5	16	2	3	2
55	1280	720	1	6	1	16	2	1	2
56	1280	800	1	8	1	17	2	1	2
57	1366	768	1	2	1	17	1	1	3
58	1280	720	1	1	2	16	2	1	1
59	1366	768	1	10	1	15	1	1	2
60	360	640	1	1	1	16	2	1	2
61	768	1024	1	12	1	16	2	1	1
62	1600	900	1	5	1	16	2	2	14
63	1920	1080	1	7	1	16	1	2	2
64	1366	768	1	12	2	16	1	1	0
65	414	736	1	4	3	16	1	1	7
66	1280	752	1	2	1	16	1	2	2
67	1920	1080	1	1	2	16	2	1	10
68	1920	1080	1	1	1	16	1	1	0
69	1440	900	2	7	2	16	2	1	1
70	1280	800	1	2	2	16	1	1	3
71	1280	1024	1	1	2	16	1	1	3
72	1680	1050	1	2	1	16	1	2	3
73	1440	960	2	4	1	16	2	2	5
74	1440	900	2	2	1	16	2	2	1
75	414	896	2	12	1	16	1	1	2
76	375	667	1	12	2	17	2	1	2
77	1366	768	1	12	4	16	2	1	2
78	414	896	1	1	1	16	1	2	2

79	1366	768	1	12	1	16	2	1	5
80	1920	1080	1	4	1	16	2	1	1
81	1366	768	1	3	1	11	1	2	2
82	360	592	1	7	1	16	2	1	1
83	1366	768	1	4	3	16	1	1	2
84	360	640	1	7	5	16	1	3	2
85	1920	1080	2	1	1	16	2	2	4
86	414	896	1	4	1	16	2	1	4
87	414	736	2	4	1	17	1	2	3
88	414	896	2	9	1	16	2	1	3
89	414	896	1	2	2	16	2	1	1
90	768	1024	1	2	3	16	2	1	2
91	1366	768	1	2	1	15	2	1	4
92	1366	768	1	7	1	13	2	1	3
93	375	667	1	4	1	16	2	2	2
94	1366	768	1	4	4	17	2	1	2
95	1280	800	1	2	1	16	1	3	5
96	1920	1080	1	4	1	16	1	2	1
97	320	568	1	2	1	16	2	1	2
98	393	786	2	4	1	16	1	2	4
99	1536	864	3	7	1	16	1	2	1
100	412	869	1	2	2	16	1	2	1

	major
1	<NA>
2	Psychology
3	Chemistry
4	<NA>
5	<NA>
6	Engineering
7	<NA>
8	Cosmetology
9	History
10	Computer Engineering
11	Grade School Teacher
12	English
13	English
14	<NA>
15	geography
16	Chemical engineering
17	computer engineering
18	Bachelor of Science in Computer Science
19	Accounting
20	<NA>

21	Classics
22	engineering
23	Medicine
24	architecture
25	Management
26	N/A
27	<NA>
28	geography
29	Bioscience
30	Neuroscience
31	<NA>
32	electrical engineering
33	German
34	Economics
35	comp sci
36	Electrotechnics + Computer science
37	Computer science
38	Applied science
39	Social sciences
40	International politics
41	Social Work
42	<NA>
43	English
44	architecture
45	n/a
46	nursing
47	political science
48	Psychology
49	law
50	<NA>
51	<NA>
52	art
53	History/World Religions
54	<NA>
55	Psychology
56	<NA>
57	<NA>
58	<NA>
59	civil engineering
60	<NA>
61	Social Work
62	Engineering Technology
63	<NA>

```

64             maths
65             Literature
66             library science
67             chemistry
68 Business & Information Technology
69             physics
70             religion
71             <NA>
72             Psychology
73             biochemistry
74             chemistry
75             Liberal Arts
76             Social Work
77             <NA>
78             Ecology and Studio Art
79             <NA>
80             Education
81 International Relations
82             <NA>
83             <NA>
84             <NA>
85             I/O Psychology
86             forensics
87             Liberal arts
88             Psychology
89             Social science
90             <NA>
91             History
92             <NA>
93             <NA>
94             <NA>
95             Anesthesia
96             History
97             <NA>
98             Theology
99             <NA>
100            Liberal Studies

```

3. Use the `typeof()` function to investigate the type of data frame `df_pwe`.

```
typeof(df_pwe)
```

```
[1] "list"
```

4. Apply the `length()` function to the data frame `df_pwe`. What does this output mean in your own words?

```
length(df_pwe)
```

```
[1] 45
```

ANSWER:

By applying the `length()` function to the data frame, I got the information of how many columns it has.

5. Use the `str()` function to investigate the structure of the data frame `df_pwe`.

```
str(df_pwe)
```

```
Classes 'tbl_df', 'tbl' and 'data.frame':  100 obs. of  45 variables:
 $ country      : chr  "US" "GB" "US" "US" ...
 $ introelapse  : num  1118 8 9 2 545 ...
 $ testelapse   : num  103 116 142 79 210 625 181 208 191 394 ...
 $ surveyelapse: num  111 128 217 90 276 79 136 198 266 380 ...
 $ TIPI1        : num  2 3 5 2 3 4 3 5 6 7 ...
 $ TIPI2        : num  6 2 6 5 1 5 5 6 7 1 ...
 $ TIPI3        : num  1 6 7 1 6 2 3 2 4 7 ...
 $ TIPI4        : num  7 1 5 2 5 6 6 5 3 1 ...
 $ TIPI5        : num  7 5 5 3 5 7 7 7 5 1 ...
 $ TIPI6        : num  7 6 5 5 6 4 5 3 1 6 ...
 $ TIPI7        : num  7 5 1 1 7 5 5 7 5 7 ...
 $ TIPI8        : num  7 2 1 7 1 6 5 1 6 1 ...
 $ TIPI9        : num  1 6 6 5 6 3 5 6 6 7 ...
 $ TIPI10       : num  1 5 7 3 3 2 1 2 4 4 ...
 $ VCL1         : num  1 1 1 1 1 1 1 0 1 0 ...
 $ VCL2         : num  1 1 1 1 1 0 1 1 0 0 ...
 $ VCL3         : num  0 1 1 0 0 0 0 0 0 0 ...
 $ VCL4         : num  1 1 1 0 1 1 1 1 1 0 ...
 $ VCL5         : num  1 1 1 1 1 1 1 0 1 1 ...
 $ VCL6         : num  0 0 0 0 0 0 0 1 0 0 ...
 $ VCL7         : num  0 1 0 0 1 0 0 0 0 0 ...
 $ VCL8         : num  0 1 0 0 0 1 1 1 1 0 ...
 $ VCL9         : num  0 0 0 0 0 0 0 0 0 0 ...
 $ VCL10        : num  1 1 1 0 1 1 1 1 1 1 ...
```

```

$ VCL11      : num  0 1 0 1 1 0 1 1 1 0 ...
$ VCL12      : num  0 0 0 0 0 0 0 0 0 0 ...
$ VCL13      : num  1 1 1 1 0 1 1 0 0 0 ...
$ VCL14      : num  1 1 1 0 1 1 1 1 0 0 ...
$ VCL15      : num  1 1 1 1 1 1 1 1 1 0 ...
$ VCL16      : num  1 1 1 1 1 1 1 1 1 1 ...
$ education  : num  2 3 2 2 2 3 1 2 2 4 ...
$ urban      : num  1 1 2 2 3 3 2 3 3 1 ...
$ gender     : num  2 2 2 1 2 1 2 2 2 1 ...
$ engnat     : num  1 1 1 1 2 2 2 2 2 2 ...
$ age        : num  19 64 18 25 19 22 32 21 19 26 ...
$ screenw    : num  1093 768 1366 1536 1152 ...
$ screenh    : num  615 1024 768 864 720 ...
$ hand       : num  1 1 1 1 1 1 1 1 1 1 ...
$ religion    : num  1 2 2 1 3 8 1 2 2 8 ...
$ orientation : num  1 1 1 1 1 1 1 1 5 2 ...
$ race       : num  16 16 16 13 11 11 16 16 16 11 ...
$ voted      : num  2 2 1 1 2 1 2 1 1 1 ...
$ married    : num  1 3 1 1 1 1 1 1 1 2 ...
$ familysize : num  2 2 3 3 2 2 2 2 1 3 ...
$ major      : chr   NA "Psychology" "Chemistry" NA ...

```

6. Use the `names` function to list the names of the elements (variables) within `df_pwe`.

```
names(df_pwe)
```

```

[1] "country"      "introelapse"  "testelapse"   "surveyelapse" "TIPI1"
[6] "TIPI2"        "TIPI3"        "TIPI4"        "TIPI5"        "TIPI6"
[11] "TIPI7"        "TIPI8"        "TIPI9"        "TIPI10"       "VCL1"
[16] "VCL2"         "VCL3"         "VCL4"         "VCL5"         "VCL6"
[21] "VCL7"         "VCL8"         "VCL9"         "VCL10"        "VCL11"
[26] "VCL12"        "VCL13"        "VCL14"        "VCL15"        "VCL16"
[31] "education"    "urban"        "gender"       "engnat"       "age"
[36] "screenw"      "screenh"      "hand"         "religion"     "orientation"
[41] "race"         "voted"        "married"      "familysize"   "major"

```

7. Wrap your answer above — `names(data_frame_name)` — within the `typeof()` function. Do the same for the `length()` function, and the `str()` function as well. Interpret what the output means in your own words.

```
typeof(names(df_pwe))
```



```
[1] "character"
```

```
typeof(length(df_pwe))
```

```
[1] "integer"
```

```
typeof(str(df_pwe))
```

```
Classes 'tbl_df', 'tbl' and 'data.frame': 100 obs. of 45 variables:
```

```
$ country      : chr  "US" "GB" "US" "US" ...
```

```
$ introelapse : num  1118 8 9 2 545 ...
```

```
$ testelapse  : num  103 116 142 79 210 625 181 208 191 394 ...
```

```
$ surveyelapse: num  111 128 217 90 276 79 136 198 266 380 ...
```

```
$ TIPI1       : num  2 3 5 2 3 4 3 5 6 7 ...
```

```
$ TIPI2       : num  6 2 6 5 1 5 5 6 7 1 ...
```

```
$ TIPI3       : num  1 6 7 1 6 2 3 2 4 7 ...
```

```
$ TIPI4       : num  7 1 5 2 5 6 6 5 3 1 ...
```

```
$ TIPI5       : num  7 5 5 3 5 7 7 7 5 1 ...
```

```
$ TIPI6       : num  7 6 5 5 6 4 5 3 1 6 ...
```

```
$ TIPI7       : num  7 5 1 1 7 5 5 7 5 7 ...
```

```
$ TIPI8       : num  7 2 1 7 1 6 5 1 6 1 ...
```

```
$ TIPI9       : num  1 6 6 5 6 3 5 6 6 7 ...
```

```
$ TIPI10      : num  1 5 7 3 3 2 1 2 4 4 ...
```

```
$ VCL1        : num  1 1 1 1 1 1 1 0 1 0 ...
```

```
$ VCL2        : num  1 1 1 1 1 0 1 1 0 0 ...
```

```
$ VCL3        : num  0 1 1 0 0 0 0 0 0 0 ...
```

```
$ VCL4        : num  1 1 1 0 1 1 1 1 1 0 ...
```

```
$ VCL5        : num  1 1 1 1 1 1 1 0 1 1 ...
```

```
$ VCL6        : num  0 0 0 0 0 0 0 1 0 0 ...
```

```
$ VCL7        : num  0 1 0 0 1 0 0 0 0 0 ...
```

```
$ VCL8        : num  0 1 0 0 0 1 1 1 1 0 ...
```

```
$ VCL9        : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
$ VCL10       : num  1 1 1 0 1 1 1 1 1 1 ...
```

```
$ VCL11       : num  0 1 0 1 1 0 1 1 1 0 ...
```

```
$ VCL12       : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
$ VCL13       : num  1 1 1 1 0 1 1 0 0 0 ...
```

```
$ VCL14       : num  1 1 1 0 1 1 1 1 0 0 ...
```

```
$ VCL15       : num  1 1 1 1 1 1 1 1 1 0 ...
```

```
$ VCL16       : num  1 1 1 1 1 1 1 1 1 1 ...
```

```
$ education   : num  2 3 2 2 2 3 1 2 2 4 ...
```

```

$ urban      : num  1 1 2 2 3 3 2 3 3 1 ...
$ gender     : num  2 2 2 1 2 1 2 2 2 1 ...
$ engnat     : num  1 1 1 1 2 2 2 2 2 2 ...
$ age        : num  19 64 18 25 19 22 32 21 19 26 ...
$ screenw    : num  1093 768 1366 1536 1152 ...
$ screenh    : num  615 1024 768 864 720 ...
$ hand       : num  1 1 1 1 1 1 1 1 1 1 ...
$ religion    : num  1 2 2 1 3 8 1 2 2 8 ...
$ orientation : num  1 1 1 1 1 1 1 1 5 2 ...
$ race       : num  16 16 16 13 11 11 16 16 16 11 ...
$ voted      : num  2 2 1 1 2 1 2 1 1 1 ...
$ married    : num  1 3 1 1 1 1 1 1 1 2 ...
$ familysize : num  2 2 3 3 2 2 2 2 1 3 ...
$ major      : chr  NA "Psychology" "Chemistry" NA ...

```

```
[1] "NULL"
```

ANSWER: For `typeof(names(df_pwe))` the output is "character", meaning that the column names of the data frame are of the class character. For `typeof(length(df_pwe))`, the output is an integer, meaning that the length of the data frame is given as an integer number. Lastly, the type of `df_pwe` was given as NULL, meaning that the output of this function does not have a specific class.

Question 4: Applying basic functions to atomic vectors

1. Create an atomic vector object named `age` with the following values: 3, 6, 41, 43.

```
age <- c(3, 6, 41, 43)
```

2. Apply the `typeof()`, `length()`, and `str()` functions to the object `age`.

```
typeof(age)
```

```
[1] "double"
```

```
length(age)
```

```
[1] 4
```

```
str(age)
```

```
num [1:4] 3 6 41 43
```

3. Apply the `sum()` function to `age`.

```
sum(age)
```

```
[1] 93
```

4. Apply the `sum()` function to `age` but this time include the argument `na.rm = FALSE`.

```
sum(age, na.rm = FALSE)
```

```
[1] 93
```

5. In general, what is a function “argument name” and what is an “argument value”? What does the argument `na.rm` do?

ANSWER: A function argument name is the name of a parameter that is passed to a function. A function argument value is the value of that parameter. The argument `na.rm` specifies whether to remove NAs from the calculation.

6. Create a new object `age2` with the following values: 3, 6, 41, 43, NA. Now calculate the sum of `age2` using the argument `na.rm = FALSE` and then calculate the sum using the argument `na.rm = TRUE`. Explain why the outputs of these two `sum()` functions differ.

```
age2 <- c(3, 6, 41, 43, NA)
```

```
sum(age2, na.rm = FALSE)
```

```
[1] NA
```

```
sum(age, na.rm = TRUE)
```

```
[1] 93
```

ANSWER: The outputs of these two `sum()` functions differ because of the `na.rm` argument. When `na.rm` is `FALSE`, all NAs are included in the calculation of the sum, and when there are missing values, the sum is unknown. When `na.rm` is `TRUE`, NAs are removed from the calculation of the sum.

7. Create a vector `tf` using the following code: `tf <- c(TRUE,FALSE,TRUE,FALSE,TRUE)`. Next apply the `typeof()`, `length()`, and `str()` functions to the object `tf`. Based on this output, briefly describe the object `tf` in your own words (one sentence is fine).

```
tf <- c(TRUE,FALSE,TRUE,FALSE,TRUE)
```

```
typeof(tf)
```

```
[1] "logical"
```

```
length(tf)
```

```
[1] 5
```

```
str(tf)
```

```
logi [1:5] TRUE FALSE TRUE FALSE TRUE
```

ANSWER: `tf` is a vector of five elements which are all logical.

8. Apply the `sum()` function to the object, using the option to remove NA values prior to calculation. What numeric value do mathematical calculations in R assign to `TRUE` values and what do they assign to `FALSE` values?

```
sum(tf, na.rm = TRUE)
```

```
[1] 3
```

ANSWER: `TRUE` values are assigned as 1 and `FALSE` values are assigned as 0.

9. This is the syntax of the `mean()` function that includes both argument names and the default values for arguments: `mean(x, trim = 0, na.rm = FALSE)`.

When using a function, R requires you to type the values you assign to each argument, but typing in the argument names is usually optional. Even though it takes a bit more time, I usually like typing in both argument names and argument values, because it forces me to be more conscious about what value I am assigning to which argument, especially when a function is new to me.

Use the `mean()` function to calculate the mean of object `tf` (removing NA values prior to calculation). In your function call, include both the argument name and the argument value for each argument (argument value for the `trim` argument can be 0). Then run the same function, but without typing any argument names.

```
mean(x = tf, trim = 0, na.rm = TRUE)
```

```
[1] 0.6
```

```
mean(tf, 0, TRUE)
```

```
[1] 0.6
```

Render to pdf and submit problem set

Render to pdf by clicking the “Render” button near the top of your RStudio window (icon with blue yarn ball) or drop down and select “Knit to PDF”.

- Go to the Canvas and under the “Assignments”, submit to the Problem Set 1 Assignment.
- Submit both .qmd and .pdf files.
- Use this naming convention “lastname_firstname_ps#” for your .qmd and pdf files (e.g. beck-emorye-ps1.rmd & beck-emorye-ps1.pdf).