

Homework 2

For this homework you will create a github repo, set up github pages, clone the repo to your computer as an R project, create a `.qmd` file, and push those changes back to github to create a webpage! You'll submit the link to your github pages site (the one that looks like a nice website).

If you were unable to get RStudio and github connected, try to set up a meeting with Dr. Post or our TA to get that figured out! For now, it is ok to use the web interface but you want to move past that method quickly!

Step 1

- Head to github and create a new repo.
 - Be sure to make the repo public and **do not** choose a `.gitignore`

Step 2

- Create a new R project from version control (as we did in the notes/videos) that clones this repository locally.
 - Recall you can click on the green button on the github.com repo website to copy the repo link.
 - A `.gitignore` file may be created in this process. That isn't a worry!

Step 3

- Create a new `.qmd` document that outputs to HTML. You can give this a title about programming in Base R. Save the file in the main repo folder.
- In this document, answer the questions below. **Use BaseR manipulations for all problems below to obtain full credit.**

Task 1: Basic Vector practice

Suppose we have data from a medical experiment on blood pressure. We have the following pre-treatment values for subjects 1 through 20:

- 120, 151, 125, 126, 115, 132, 132, 129, 134, 139, 127, 122, 127, 135, 133, 128, 147, 138, 140, 132

after treatment, the subjects were measured again (subjects 1 through 20 match)

- 127, 145, 135, 122, 115, 122, 123, 126, 126, 129, 132, 146, 120, 114, 121, 120, 128, 120, 133, 115

1. Create two vectors. One vector corresponding to the pre measurements and one to the post measurements.

```
pre_trt_bp <- c(120, 151, 125, 126, 115, 132, 132,
               129, 134, 139, 127, 122, 127, 135,
               133, 128, 147, 138, 140, 132)
post_trt_bp <- c(127, 145, 135, 122, 115, 122, 123,
                126, 126, 129, 132, 146, 120, 114,
                121, 120, 128, 120, 133, 115)
```

2. Assign **names** to the vector elements using the **paste()** function. Note that **names()** can be overwritten by a character vector. To quickly create the names, try running the code

```
paste("Subject", 1:20, sep = "_")
```

```
## [1] "Subject_1" "Subject_2" "Subject_3" "Subject_4" "Subject_5"
## [6] "Subject_6" "Subject_7" "Subject_8" "Subject_9" "Subject_10"
## [11] "Subject_11" "Subject_12" "Subject_13" "Subject_14" "Subject_15"
## [16] "Subject_16" "Subject_17" "Subject_18" "Subject_19" "Subject_20"
```

Create the same names for each vector's elements.

```
subjects <- paste("Subject", 1:20, sep = "_")
names(pre_trt_bp) <- names(post_trt_bp) <- subjects
```

3. Calculate the change in blood pressure for each patient by subtracting post-treatment measurements from pre-treatment measurements. Recall that R does math element-wise! Save this calculation as a new object in R (also a vector).

```
diff_bp <- pre_trt_bp - post_trt_bp
diff_bp
```

```
## Subject_1 Subject_2 Subject_3 Subject_4 Subject_5 Subject_6 Subject_7
##      -7         6      -10         4         0         10         9
## Subject_8 Subject_9 Subject_10 Subject_11 Subject_12 Subject_13 Subject_14
##       3         8         10        -5       -24         7        21
## Subject_15 Subject_16 Subject_17 Subject_18 Subject_19 Subject_20
##      12         8         19         18         7         17
```

4. Calculate the average decrease in blood pressure across all patients. Use the **mean()** function here!

```
mean(diff_bp)
```

```
## [1] 5.65
```

5. Determine which patients experienced a decrease in blood pressure after treatment (a positive change). Use the **which()** function to just return the indices (and names) associated with this type of change.

```
which(diff_bp > 0)
```

```
## Subject_2 Subject_4 Subject_6 Subject_7 Subject_8 Subject_9 Subject_10
##          2          4          6          7          8          9          10
## Subject_13 Subject_14 Subject_15 Subject_16 Subject_17 Subject_18 Subject_19
##          13          14          15          16          17          18          19
## Subject_20
##          20
```

6. Subset the vector of differences to only return those that have a positive change. (Note that you don't need the `which()` function here. A Boolean used as an indexing vector returns only indices where a TRUE occurred.)

```
diff_bp[diff_bp > 0]
```

```
## Subject_2 Subject_4 Subject_6 Subject_7 Subject_8 Subject_9 Subject_10
##          6          4          10          9          3          8          10
## Subject_13 Subject_14 Subject_15 Subject_16 Subject_17 Subject_18 Subject_19
##          7          21          12          8          19          18          7
## Subject_20
##          17
```

7. Calculate the average decrease in blood pressure for those where the blood pressure decreased (positive change).

```
mean(diff_bp[diff_bp > 0])
```

```
## [1] 10.6
```

Try to render your document locally just to make sure your `.qmd` document is working!

Task 2: Basic Data Frame practice

Continue the previous example.

1. Create a data frame object with four columns corresponding to your data above: `patient`, `pre_bp`, `post_bp`, and `diff_bp`

```
bp_df <- data.frame(patient = subjects,
                    pre_bp = pre_trt_bp,
                    post_bp = post_trt_bp,
                    diff_bp = diff_bp)
```

2. Return only rows of the data frame where the `diff_bp` column is negative. (Use `[]` or learn about the `subset()` function if you'd like. If you use `[]`, don't reference the original vector from the first part, access the column of the data frame to make your comparison with 0.)

```
bp_df[bp_df$diff_bp < 0, ]
```

```
##           patient pre_bp post_bp diff_bp
## Subject_1  Subject_1  120    127     -7
## Subject_3  Subject_3  125    135    -10
## Subject_11 Subject_11  127    132     -5
## Subject_12 Subject_12  122    146    -24
```

3. Add a new column to the data frame corresponding to **TRUE** if the **post_bp** is less than 120 and **FALSE** if it is not. Similar to the previous question, don't reference the original vector from task 1, access the column of the data frame to make your comparison.

Recall you can use **\$** to access a column. If you reference a column that doesn't exist and save a vector (of appropriate length in it), that vector becomes a column of your data frame! Use this way of creating the new column.

```
bp_df$normal <- bp_df$post_bp < 120
```

4. Finally, print the data frame out nicely in your final document by modifying the code below appropriately.

```
knitr::kable(bp_df)
```

```
knitr::kable(bp_df)
```

	patient	pre_bp	post_bp	diff_bp	normal
Subject_1	Subject_1	120	127	-7	FALSE
Subject_2	Subject_2	151	145	6	FALSE
Subject_3	Subject_3	125	135	-10	FALSE
Subject_4	Subject_4	126	122	4	FALSE
Subject_5	Subject_5	115	115	0	TRUE
Subject_6	Subject_6	132	122	10	FALSE
Subject_7	Subject_7	132	123	9	FALSE
Subject_8	Subject_8	129	126	3	FALSE
Subject_9	Subject_9	134	126	8	FALSE
Subject_10	Subject_10	139	129	10	FALSE
Subject_11	Subject_11	127	132	-5	FALSE
Subject_12	Subject_12	122	146	-24	FALSE
Subject_13	Subject_13	127	120	7	FALSE
Subject_14	Subject_14	135	114	21	TRUE
Subject_15	Subject_15	133	121	12	FALSE
Subject_16	Subject_16	128	120	8	FALSE
Subject_17	Subject_17	147	128	19	FALSE
Subject_18	Subject_18	138	120	18	FALSE
Subject_19	Subject_19	140	133	7	FALSE
Subject_20	Subject_20	132	115	17	TRUE

I'd again render your document to make sure everything is looking good!

Task 3: List practice

Continue the previous example. Suppose we now also have data from another experiment where the 'treatment' was actually a placebo.

We have the following pre-treatment values for subjects 1 through 10 (different set of subjects):

- 138, 135, 147, 117, 152, 134, 114, 121, 131, 130

after treatment, the subjects were measured again (subjects 1 through 10 match)

- 105, 136, 123, 130, 134, 143, 135, 139, 120, 124

1. Create a new data frame with this data in it that is similar to the data frame from task 2 (including the new column of TRUE/FALSE values).

```
bp_df_placebo <- data.frame(patient = subjects[1:10],
  pre_bp = c(138, 135, 147, 117, 152,
             134, 114, 121, 131, 130),
  post_bp = c(105, 136, 123, 130, 134,
              143, 135, 139, 120, 124))
bp_df_placebo$diff_bp <- bp_df_placebo$pre_bp - bp_df_placebo$post_bp
bp_df_placebo$normal <- bp_df_placebo$post_bp < 120
```

2. Now create and store a list with two elements:

- 1st element named `treatment` and contains the first data frame you created.
- 2nd element named `placebo` and contains the second data frame you created.

```
bp_list <- list(treatment = bp_df,
  placebo = bp_df_placebo)
```

3. Access the first list element using three different types of syntax (the result can be a list or just the data frame).

```
bp_list$treatment
```

```
##           patient pre_bp post_bp diff_bp normal
## Subject_1 Subject_1   120    127     -7  FALSE
## Subject_2 Subject_2   151    145      6  FALSE
## Subject_3 Subject_3   125    135    -10  FALSE
## Subject_4 Subject_4   126    122      4  FALSE
## Subject_5 Subject_5   115    115      0   TRUE
## Subject_6 Subject_6   132    122     10  FALSE
## Subject_7 Subject_7   132    123      9  FALSE
## Subject_8 Subject_8   129    126      3  FALSE
## Subject_9 Subject_9   134    126      8  FALSE
## Subject_10 Subject_10  139    129     10  FALSE
## Subject_11 Subject_11  127    132     -5  FALSE
## Subject_12 Subject_12  122    146    -24  FALSE
## Subject_13 Subject_13  127    120      7  FALSE
## Subject_14 Subject_14  135    114     21   TRUE
## Subject_15 Subject_15  133    121     12  FALSE
## Subject_16 Subject_16  128    120      8  FALSE
## Subject_17 Subject_17  147    128     19  FALSE
## Subject_18 Subject_18  138    120     18  FALSE
## Subject_19 Subject_19  140    133      7  FALSE
## Subject_20 Subject_20  132    115     17   TRUE
```

```
bp_list[[1]]
```

```
##           patient pre_bp post_bp diff_bp normal
## Subject_1 Subject_1   120    127     -7  FALSE
```

```
## Subject_2 Subject_2 151 145 6 FALSE
## Subject_3 Subject_3 125 135 -10 FALSE
## Subject_4 Subject_4 126 122 4 FALSE
## Subject_5 Subject_5 115 115 0 TRUE
## Subject_6 Subject_6 132 122 10 FALSE
## Subject_7 Subject_7 132 123 9 FALSE
## Subject_8 Subject_8 129 126 3 FALSE
## Subject_9 Subject_9 134 126 8 FALSE
## Subject_10 Subject_10 139 129 10 FALSE
## Subject_11 Subject_11 127 132 -5 FALSE
## Subject_12 Subject_12 122 146 -24 FALSE
## Subject_13 Subject_13 127 120 7 FALSE
## Subject_14 Subject_14 135 114 21 TRUE
## Subject_15 Subject_15 133 121 12 FALSE
## Subject_16 Subject_16 128 120 8 FALSE
## Subject_17 Subject_17 147 128 19 FALSE
## Subject_18 Subject_18 138 120 18 FALSE
## Subject_19 Subject_19 140 133 7 FALSE
## Subject_20 Subject_20 132 115 17 TRUE
```

```
bp_list[1] #returns a list with one element in it
```

```
## $treatment
##      patient pre_bp post_bp diff_bp normal
## Subject_1 Subject_1 120 127 -7 FALSE
## Subject_2 Subject_2 151 145 6 FALSE
## Subject_3 Subject_3 125 135 -10 FALSE
## Subject_4 Subject_4 126 122 4 FALSE
## Subject_5 Subject_5 115 115 0 TRUE
## Subject_6 Subject_6 132 122 10 FALSE
## Subject_7 Subject_7 132 123 9 FALSE
## Subject_8 Subject_8 129 126 3 FALSE
## Subject_9 Subject_9 134 126 8 FALSE
## Subject_10 Subject_10 139 129 10 FALSE
## Subject_11 Subject_11 127 132 -5 FALSE
## Subject_12 Subject_12 122 146 -24 FALSE
## Subject_13 Subject_13 127 120 7 FALSE
## Subject_14 Subject_14 135 114 21 TRUE
## Subject_15 Subject_15 133 121 12 FALSE
## Subject_16 Subject_16 128 120 8 FALSE
## Subject_17 Subject_17 147 128 19 FALSE
## Subject_18 Subject_18 138 120 18 FALSE
## Subject_19 Subject_19 140 133 7 FALSE
## Subject_20 Subject_20 132 115 17 TRUE
```

4. Use your list from question 2. In one line of code, access the `pre_bp` column of the placebo data frame.

```
bp_list$placebo$pre_bp
```

```
## [1] 138 135 147 117 152 134 114 121 131 130
```

You should render the document to check things are looking good.

- Make sure that all code chunks show (and are evaluated).
- **Use headings to separate the sections.**
- Write text before each code chunk explaining what you are trying to do.
- Use markdown where appropriate (to create lists, bold things, etc.).

Step 4

(The subsequent steps are the same steps from homework 1 - more detail is given there.)

In your repo folder (locally), create a file called `_quarto.yml`. Open this file (perhaps in RStudio or a text editor) and place the following in the file (spacing is important!):

```
project:
  type: website
  output-dir: docs
```

Step 5

Now create a file called `.nojekyll` in your project repo. This file doesn't need to have anything in it! You just need that file there (it may be a hidden file after you create it. Github should still track it.)

Step 6

Open the terminal in RStudio and run the following code:

```
quarto render
```

Step 7

Push all changes up to your repo! You can do this via menus or the command line (or via the github web interface).

Step 8

Head to your github repo page. Go to settings, choose pages, and under "Branch" choose 'main' and change the folder to /docs. Then hit save!

Step 9

Wait about 2 minutes... Head back to your main github repo page. You'll now see a 'Deployments' section on the bottom right.

Click on that. Hopefully, after a minute you see a green check and something that says your site is ready!

Click on that and you should see a nicely rendered website! **Copy the link to that site and that is what you'll turn in for this assignment!**