Lab 02

Jason Gerber

2/1/2021

Contents

Shell commands

In this repo is a zip file named `organize_me.zip`. It can by unzipped by running R code `unzip("organize_me.zip")`. Do this in your console so you only unzip the folder once and not each time you knit the Rmd file.

The files in `organize_me/` contain a sample of various files you may have if you run a simulation on a high performance computer. Organize the contents in `organize_me/` using the terminal and shell according to the following steps.

1. delete all `.input2` files;
2. remove the `flags/` directory and all of its contents;
3. place the test data text files in a directory named `test_data/`;
4. place the `.sh` files in a directory named `scripts_shell/`;
5. place the `.R` files in a directory named `scripts_R/`;
6. place the `.qsub` files in a directory named `hpc_start/`;
7. copy directory `organize_me/` and name the copy `simulations/`;
8. delete directory `organize_me/` and all of its contents.

Attributes

Create a date-time object based on the atomic double vector below. You'll need to set the class attribute to `"POSIXct"`

```r
z <- seq(0, 3600 * 10, by = 3600)

as.Date.POSIXct(z)
```

```
#>  [1] "1970-01-01" "1970-01-01" "1970-01-01" "1970-01-01" "1970-01-01"
#>  [6] "1970-01-01" "1970-01-01" "1970-01-01" "1970-01-01" "1970-01-01"
#> [11] "1970-01-01"
```

What do you notice about the date-time? If you are unsure, check out Unix Epoch. A time converter is available here.

Subsetting

Consider the atomic vector `x`.

```
set.seed(0826) # set seed to ensure reproducibility
x <- sample(x = c(letters, -15:15), size = 50, replace = TRUE)
x
```

```
#>  [1] "2"   "13"  "5"   "k"   "15"  "i"   "r"   "-11" "m"   "f"   "-8"  "10" 
#> [13] "-1"  "0"   "o"   "-12" "-1"  "-6"  "-12" "-8"  "9"   "-4"  "-13" "9"  
#> [25] "3"   "m"   "-9"  "2"   "m"   "n"   "-3"  "d"   "8"   "t"   "j"   "o"  
#> [37] "f"   "-4"  "-14" "t"   "-4"  "-4"  "-6"  "11"  "k"   "10"  "-14" "11" 
#> [49] "y"   "5"
```

Complete the following without using any loops.

1. Subset only the letters from `x`.

```
x[is.na(as.numeric(x))]
```

```
#>  [1] "k" "i" "r" "m" "f" "o" "m" "m" "n" "d" "t" "j" "o" "f" "t" "k" "y"
```

2. Subset only the negative numbers from `x`.

```
x <- x[!is.na(as.numeric(x))]
x[x<0]
```

```
#>  [1] "-11" "-8"  "-1"  "-12" "-1"  "-6"  "-12" "-8"  "-4"  "-13" "-9"  "-3" 
#> [13] "-4"  "-14" "-4"  "-4"  "-6"  "-14"
```

3. Select the numbers divisible by 3 from `x`.

```
x <- x[!is.na(as.numeric(x))]
x <- as.numeric(x)
x[(x %% 3) == 0]
```

```
#>  [1]  15   0 -12  -6 -12   9   9   3  -9  -3  -6
```

4. Remove all values with an even index from `x`.

```
x[-seq(2, length(x), 2)]
```

```
#>  [1]   2   5 -11  10   0  -1 -12   9 -13   3   2   8 -14  -4  11 -14   5
```

Git from the command line

Git cheat sheet

1. Assuming you completed the shell commands tasks above, use git from the command line to stage and commit your work.

2. After committing, delete directory `simulations/`.

3. Use git from the command line to stage and commit your work.

4. Use git from the command line to revert your last commit in order to recover `simulations/`. Refer to the above cheat sheet for assistance.