R Notebook: Workspace Audit for [11373111]

Contents

Workspace Audit **2**
Set your WSU id and location of your github USERNAME/REPO . . . 2
Code Frequency on GitHub . . . 2
Commit Frequency on GitHub . . . 2
General Workspace Environment Data . . . 3
General Workspace Environment Data . . . 3
Local Computer Setup . . . 3
Dropbox . . . 3
Git tool . . . 3
Git History . . . 3
Git Log Data . . . 5
Git Log Graphics . . . 6
Conclusive Summary . . . 17


```
library(devtools);
```

```
## Loading required package: usethis
```

```
library(humanVerseWSU);

path.github = "https://raw.githubusercontent.com/MonteShaffer/humanVerseWSU/master/";

include.me = paste0(path.github, "misc/functions-nlp.R");
source_url( include.me );
```

```
## SHA-1 hash of file is 704afa69d52215d315cb5f59cdc020b0bbfd0b13
```

```
## Warning: package 'tm' was built under R version 4.0.3
```

```
## Loading required package: NLP
```

```
## Warning: package 'NLP' was built under R version 4.0.3
```

```
## Warning: package 'SentimentAnalysis' was built under R version 4.0.3
```

```
##
## Attaching package: 'SentimentAnalysis'
```

```
## The following object is masked from 'package:base':
##
##     write
```

```
include.me = paste0(path.github, "misc/functions-nlp-str.R");
source_url( include.me );
```

```
## SHA-1 hash of file is 6bdb234fa84eea995969dc29d6ff2a78f3982131
```

```
include.me = paste0(path.github, "misc/functions-nlp-stack.R");
source_url( include.me );
```

```
## SHA-1 hash of file is 034efbce0405954198545f8798e119b77a4809c9
```

```
include.me = paste0(path.github, "misc/functions-nlp-pos.R");
source_url( include.me );
```

```
## SHA-1 hash of file is d8c8cf01c8ead1b6d4228891aa52bac77084a6e7
```

```
## Warning: package 'openNLP' was built under R version 4.0.3
```

```
include.me = paste0(path.github, "humanVerseWSU/R/functions-encryption.R");
source_url( include.me );
```

```
## SHA-1 hash of file is da71dde620bed33db055778b752eefb476f7bf6b
```

```
###### UPDATES TO dataframe subset function ######
# inflation adjustments for NA ... and improvements on subsetting
include.me = paste0(path.github, "humanVerseWSU/R/functions-str.R");
source_url( include.me );
```

```
## SHA-1 hash of file is 5f57d466d946285783f15bb3b4d97ca48a951d0b
```

```
include.me = paste0(path.github, "humanVerseWSU/R/functions-file.R");
source_url( include.me );
```

```
## SHA-1 hash of file is 091aefe67c59f7dc1644c268fdd0b0a183c525ef
```

```
include.me = paste0(path.github, "humanVerseWSU/R/functions-dataframe.R");
source_url( include.me );
```

```
## SHA-1 hash of file is 1149cbf3e865f692b50d4d1983e6364dc56ce62d
```

```
include.me = paste0(path.github, "humanVerseWSU/R/functions-inflation.R");
source_url( include.me );
```

```
## SHA-1 hash of file is b6d29327e3fe030ca132b135f4a89b6fc6a61a66
```

Workspace Audit

You should place this file in your github local path for this course, in a subfolder called "final" ... for me that is `C:/_git_/WSU_STATS419_FALL2020/final/`;

Set your WSU id and location of your github USERNAME/REPO

```
wsu.id = "Joshua_Bennett"; # change this to yours.
github.repository = "Joshua-Bennett/WSU_STATS419_FALL2020";
# be sure to change the notebook title at the top of this document to be "you" based on your wsu.id
```

Code Frequency on GitHub

Browse to https://github.com/MonteShaffer/WSU_STATS419_FALL2020/graphs/code-frequency where you replace my repository name with yours. Screenshot and include the graphic in this folder, named "github-code-frequency.png".

Source: GitHub

Commit Frequency on GitHub

Browse to https://github.com/MonteShaffer/WSU_STATS419_FALL2020/graphs/commit-activity where you replace my repository name with yours. Screenshot and include the graphic in this folder, named

"github-commits.png".

Source: GitHub

General Workspace Environment Data

```
path.to.git.local = "C:/Users/Galac/Desktop/git419/Stats419_FALL2020/";
setwd(path.to.git.local);

path.to.git.final = paste0(path.to.git.local,"final/");
```

General Workspace Environment Data

```
my.object = list();

sg = Sys.getenv();
  snames = names(sg);
  svals = as.character(sg);
sgdf = as.data.frame(cbind(snames,svals));

my.object$environment = sgdf;

sw = Sys.which(c("ftp", "ping", "texi2dvi", "this-does-not-exist", "make", "latex", "pdflatex", "luatex
  snames = names(sw);
  svals = as.character(sw);
swdf = as.data.frame(cbind(snames,svals));

my.object$bin = swdf;
my.object$packages = as.data.frame(installed.packages());

my.object$local.dirs = list.dirs(path.to.git.local,recursive = FALSE);
my.object$local.files = list.files(path.to.git.local,recursive = TRUE);
```

Local Computer Setup

Dropbox

Using your file-explorer, browse to your "Dropbox" folder on your computer, enter the "**student_access**" folder for this course, and take a screenshot. Place in this folder "final/dropbox.png" ...

Source: LOCAL COMPUTER

Git tool

Take a screenshot of your gitbash tool (or if you use Github Desktop, that will do). Be certain the screenshot is referencing your repository for this course. Place in this folder "final/git.png" ...

Source: LOCAL COMPUTER

Git History

Open your git tool from the command line and browse to the correct folder ... Type `git status` to verify you are in the right place.

Now type: `git log --shortstat --pretty=format:'%h|%t|%p|%ai|%ae|%an|%ci|%ce|%cn|%N|%f|%s|%b' | paste - - - > final/git-history.log` which will create a file with your git history as a text file. We

will import and perform a basic graph.

```
#  git log --shortstat --pretty=format:'"%h","%an","%ae","%aD","%s",' | paste - - - > final/git-history
# git log --oneline --decorate > final/git-history.log
# git log --shortstat --pretty=format:'%h|%t|%p|%ai|%ae|%an|%ci|%ce|%cn|%N|%f|%s|%b' > final/git-histor
#####################################################
############ BELOW IS THE FINAL ANSWER ############
#####################################################
# git log --shortstat --pretty=format:'%h|%t|%p|%ai|%ae|%an|%ci|%ce|%cn|%N|%f|%s|%b' | paste - - - > fi

# https://mirrors.edge.kernel.org/pub/software/scm/git/docs/git-log.html#_pretty_formats
```

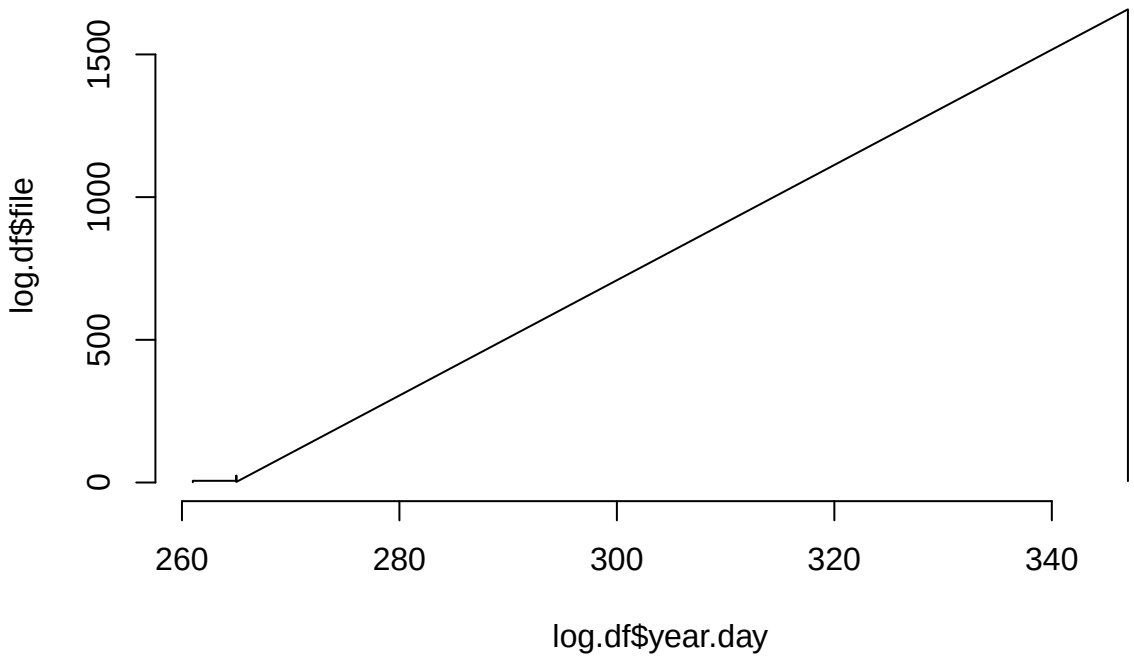
```
my.git = list();

raw.df = read.csv(paste0(path.to.git.final,"git-history.log"), header=FALSE, sep="|", quote="");
colnames(raw.df) = c("hash","tree","parent","author.date","author.email","author.name","commit.date","c
#raw.df;

total.commits = nrow(raw.df);



date.strings = raw.df$commit.date;
date.types = c("%m","%d","%W","%j","%u","%H");
date.names = c("month", "day", "year.week", "year.day", "day.week","day.hour");

time.df = convertDateStringToFormat(date.strings, date.types, date.names);
#time.df;

changes.df = NULL;
for(i in 1:total.commits)
  {
  my.commit = removeWhiteSpace(strsplit(raw.df$content.b[i],",",fixed=TRUE)[[1]]);

  my.detail = strsplit(my.commit," ",fixed=TRUE);

  row = c(0,0,0);
  for(j in 1:length(my.detail))
    {
    val = as.numeric( my.detail[[j]][1] );
    row[j] = val;
    }
  changes.df = rbind(changes.df,row);
  }

changes.df = as.data.frame(changes.df);
colnames(changes.df) = c("file","insert","delete");
#changes.df;

log.df = cbind(raw.df[,c(1:3,7:9,12)], time.df, changes.df);

my.git$raw.df = raw.df;
my.git$total.commits = total.commits;
```

```
my.git$time.df = time.df;
my.git$changes.df = changes.df;
my.git$log.df = log.df;

log.df;
```

Git Log Data

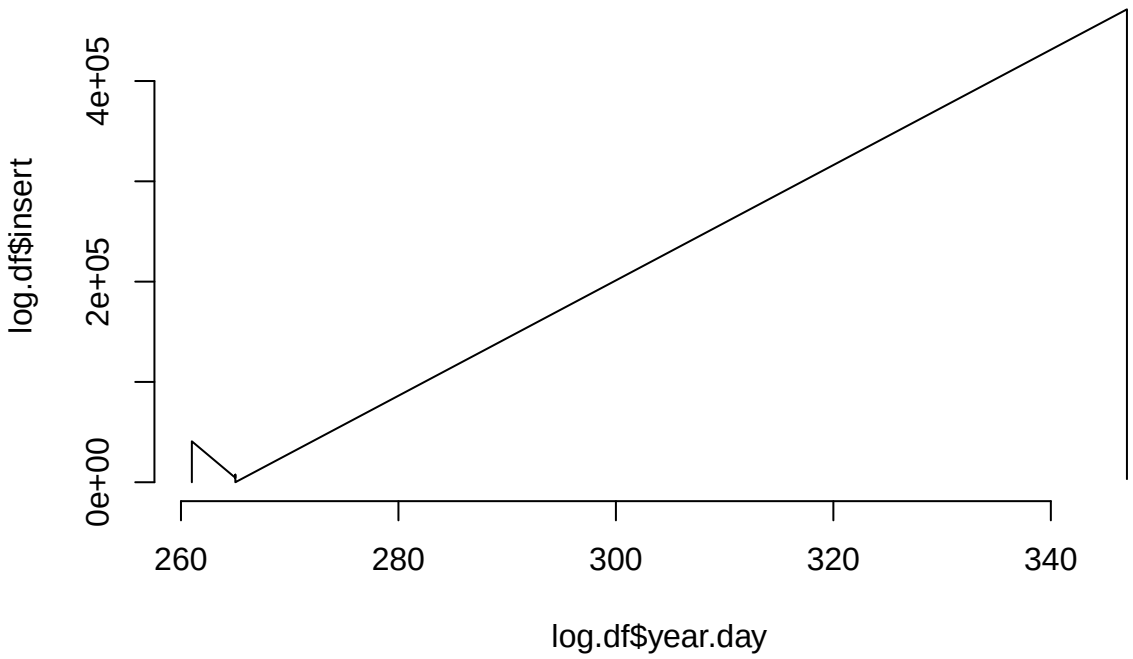
```
##          hash    tree  parent               commit.date
## row   20a4aa3 f2ab436 d09a6d0 2020-12-12 18:17:42 -0800
## row.1 d09a6d0 fe24d12 634448f 2020-12-12 17:57:11 -0800
## row.2 634448f b27589f f5987c2 2020-09-21 20:03:23 -0700
## row.3 f5987c2 2ee4866 84e5680 2020-09-21 19:47:09 -0700
## row.4 84e5680 129db18 0901acc 2020-09-21 17:44:45 -0700
## row.5 0901acc dcc6787 72b3588 2020-09-17 16:01:27 -0700
## row.6 72b3588 f7ffd2c         2020-09-17 15:55:11 -0700
##                    commit.email    commit.name
## row   joshua.r.bennett@wsu.edu Joshua Bennett
## row.1 joshua.r.bennett@wsu.edu Joshua Bennett
## row.2 joshua.r.bennett@wsu.edu Joshua Bennett
## row.3 joshua.r.bennett@wsu.edu Joshua Bennett
## row.4 joshua.r.bennett@wsu.edu Joshua Bennett
## row.5 joshua.r.bennett@wsu.edu Joshua Bennett
## row.6       noreply@github.com         GitHub
##                                                          content.s month day
## row   Added this from my other computer to make sure it was on here.    12  12
## row.1               hey, I guess I should have been doing this more.    12  12
## row.2                                                         fixed     9  21
## row.3                                                           hw3     9  21
## row.4                                             personality Dataset     9  21
## row.5                                                adding old files     9  17
## row.6                                                  Initial commit     9  17
##       year.week year.day day.week day.hour file insert delete
## row          49      347        6       18    5   3137      0
## row.1        49      347        6       17 1658 471445    322
## row.2        38      265        1       20    2      4     47
## row.3        38      265        1       19   24   7556      9
## row.4        38      265        1       17    6   4440      0
## row.5        37      261        4       16    6  40798      0
## row.6        37      261        4       15    1      1      0
```

```
plot(log.df$year.day, log.df$file, bty="n", type="l");
```

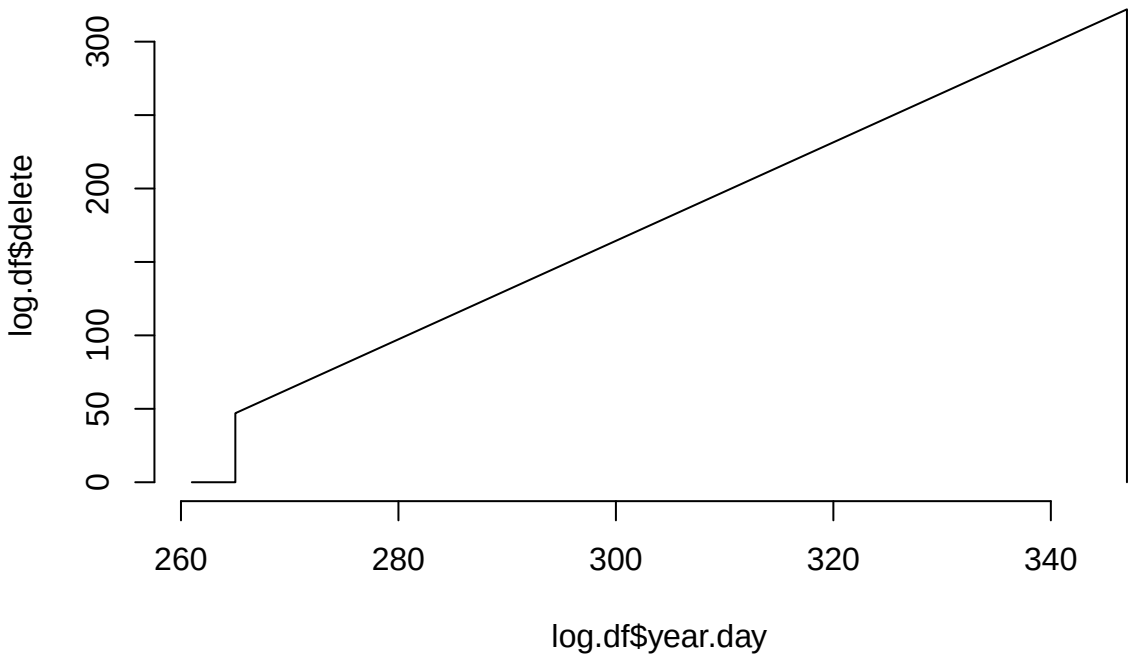


Git Log Graphics

```
plot(log.df$year.day, log.df$insert, bty="n", type="l");
```

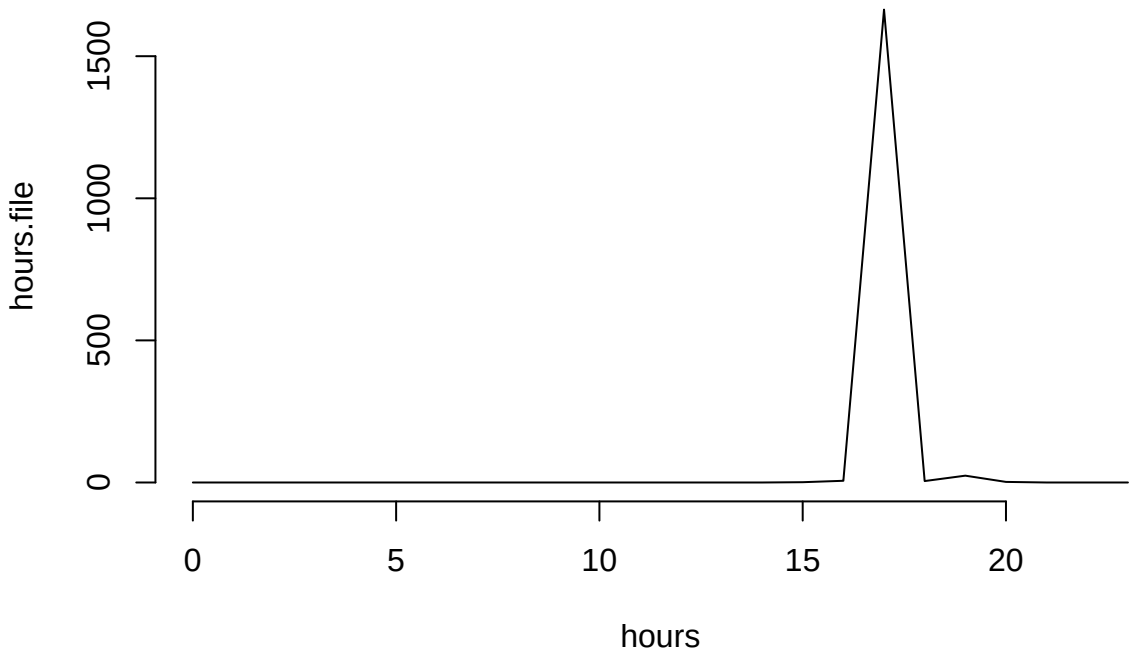



```
plot(log.df$year.day, log.df$delete, bty="n", type="l");
```

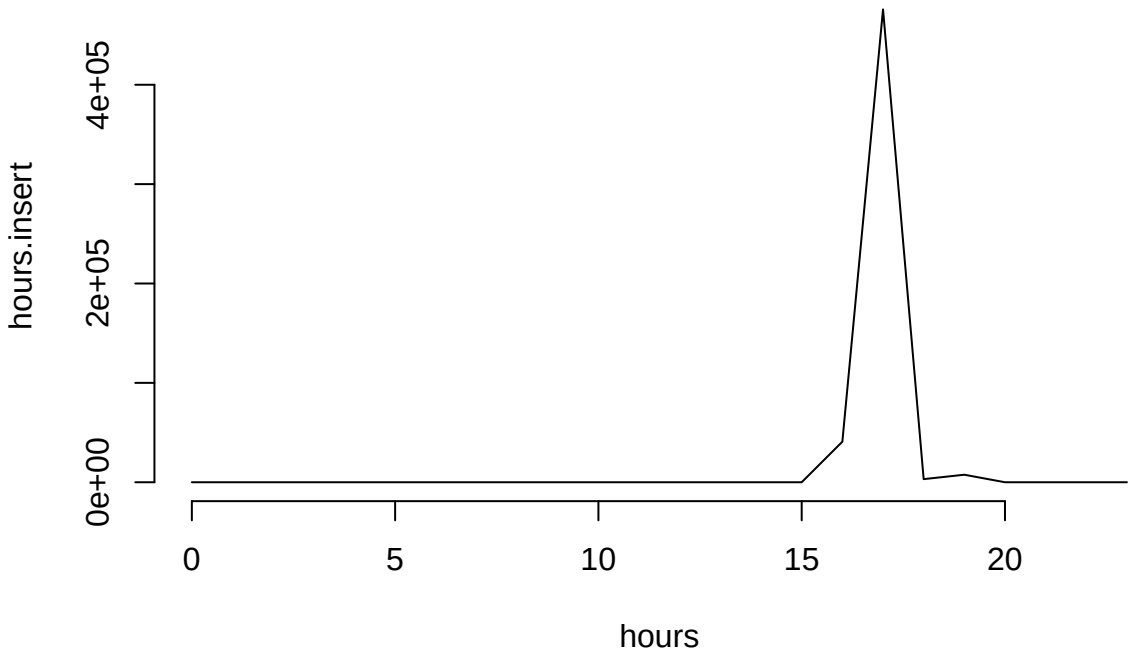



```
weeks = sort(unique(log.df$year.week));
n.weeks = length(weeks);
weeks.file = weeks.insert = weeks.delete = numeric(n.weeks);

days = 1:7;
n.days = length(days);
days.file = days.insert = days.delete = numeric(n.days);

hours = 0:23;
n.hours = length(hours);
hours.file = hours.insert = hours.delete = numeric(n.hours);

for(i in 1:total.commits)
  {
  my.hour = log.df$day.hour[i];
    idx.hour = which(my.hour == hours)[1];
  my.week = log.df$year.week[i];
    idx.week = which(my.week == weeks)[1];
  my.day = log.df$day.week[i];
    idx.day = which(my.day == days)[1];

  n.file = log.df$file[i];
  n.insert = log.df$insert[i];
  n.delete = log.df$delete[i];

  hours.file[idx.hour]   = hours.file[idx.hour]   + n.file;
```

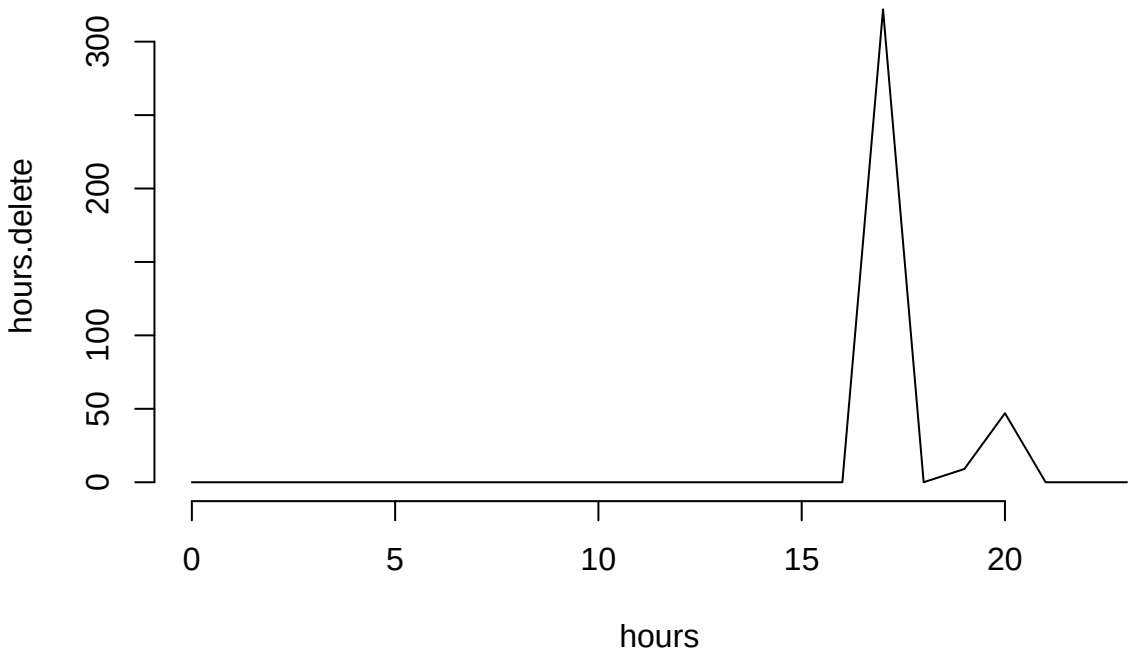
```
  hours.insert[idx.hour] = hours.insert[idx.hour] + n.insert;
  hours.delete[idx.hour] = hours.delete[idx.hour] + n.delete;

  days.file[idx.day]   = days.file[idx.day]   + n.file;
  days.insert[idx.day] = days.insert[idx.day] + n.insert;
  days.delete[idx.day] = days.delete[idx.day] + n.delete;

  weeks.file[idx.week]   = weeks.file[idx.week]   + n.file;
  weeks.insert[idx.week] = weeks.insert[idx.week] + n.insert;
  weeks.delete[idx.week] = weeks.delete[idx.week] + n.delete;
  }


my.git$groups = list();
my.git$groups$hours = list("hours" = hours, "file" = hours.file, "insert" = hours.insert, "delete" = hou
my.git$groups$days = list("days" = days, "file" = days.file, "insert" = days.insert, "delete" = days.del
my.git$groups$weeks = list("weeks" = weeks, "file" = weeks.file, "insert" = weeks.insert, "delete" = wee
my.object$git = my.git;

plot(hours, hours.file,   bty="n", type="l");  # 0 is midnight
```



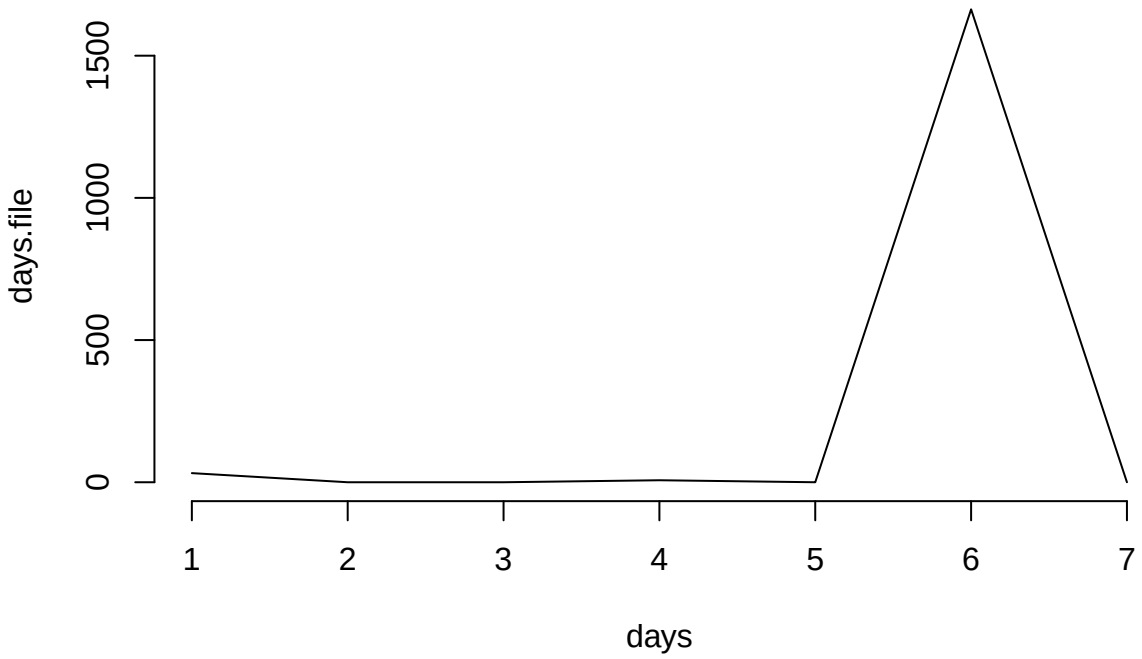

```
plot(hours, hours.insert, bty="n", type="l");
```



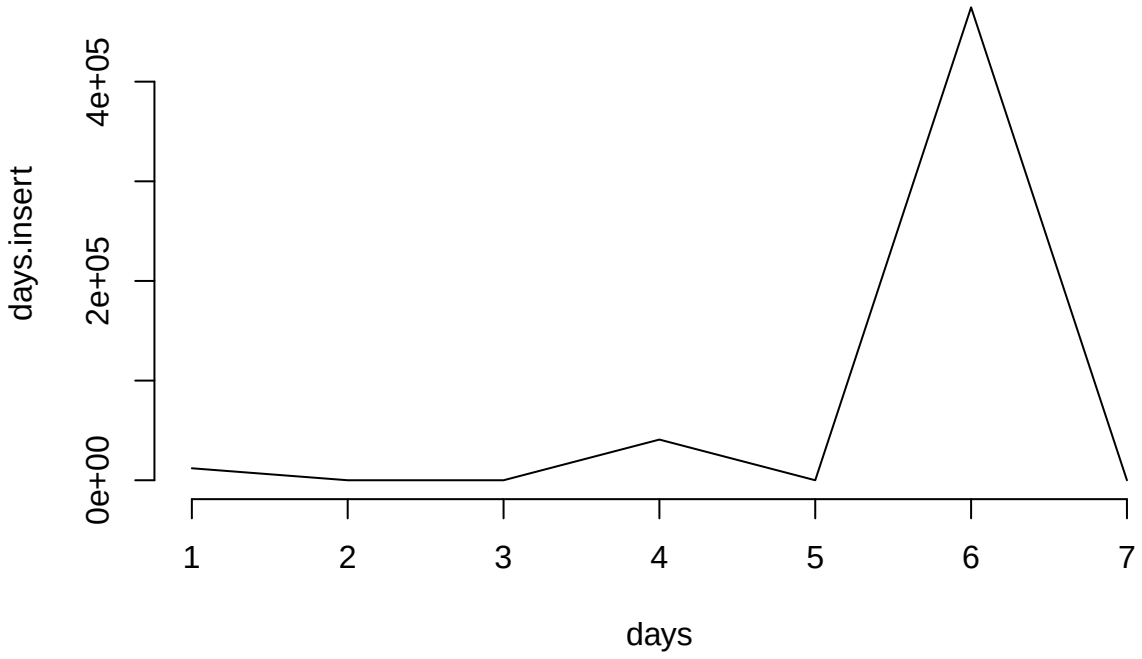

```
plot(hours, hours.delete, bty="n", type="l");
```



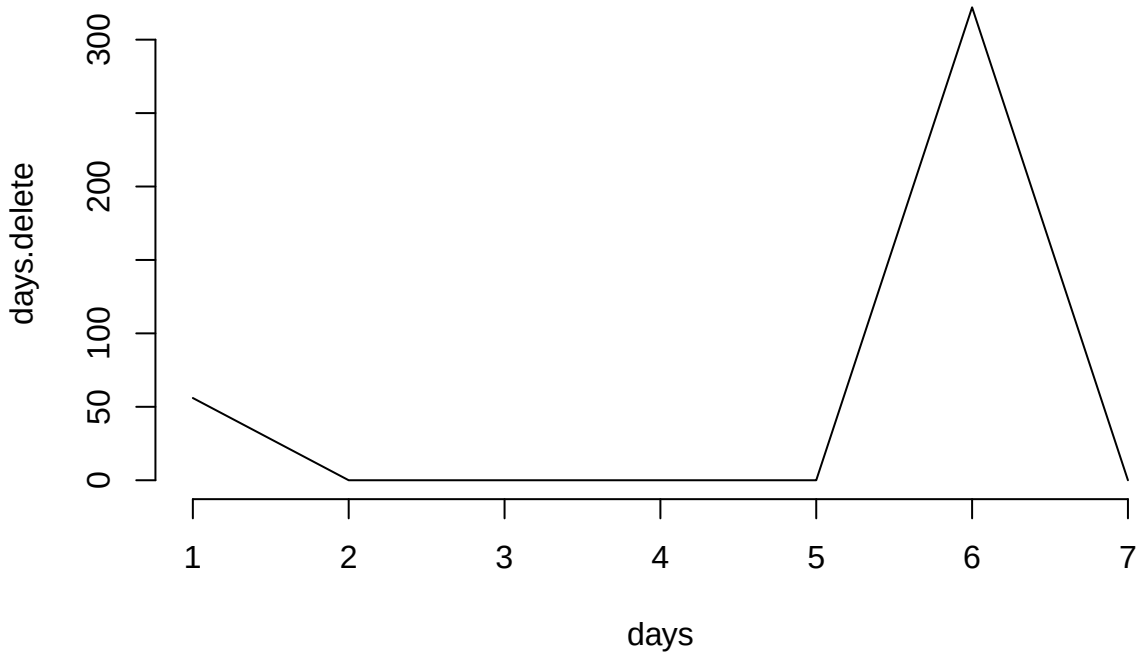

```
plot(days, days.file,   bty="n", type="l");    # 1 is Monday
```



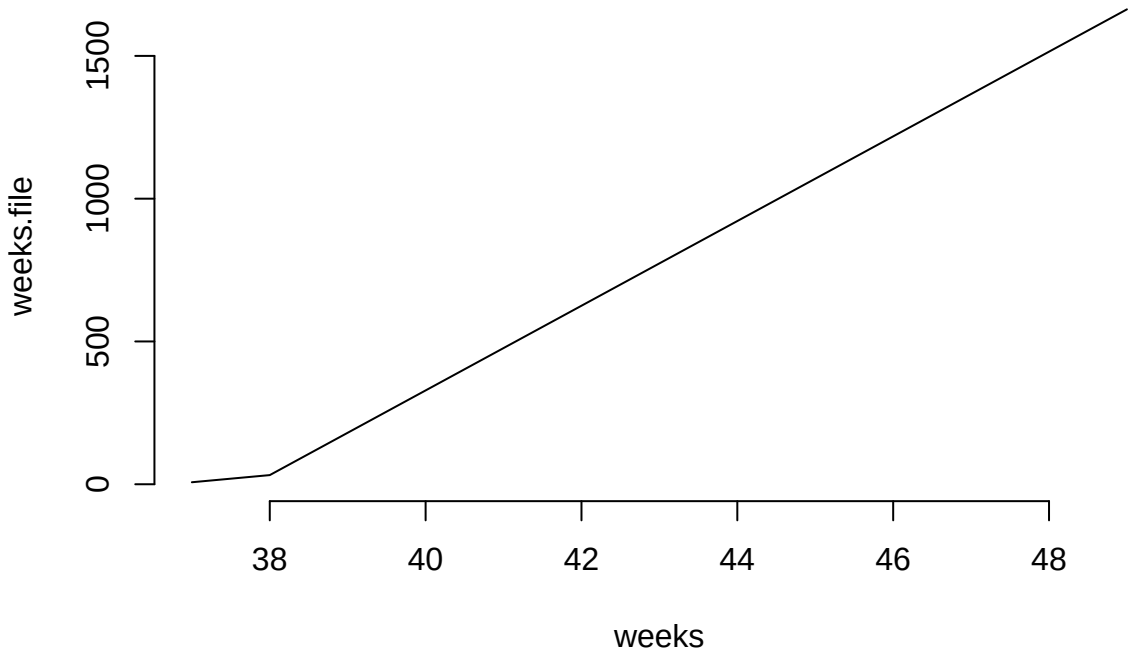

```
plot(days, days.insert, bty="n", type="l");
```



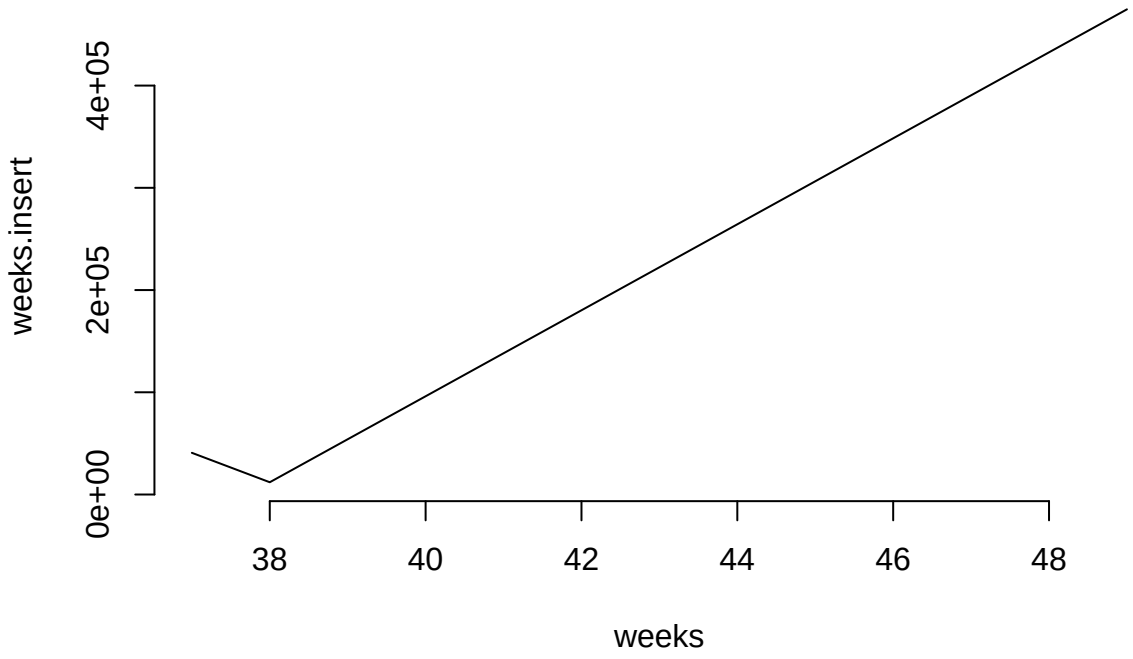

```
plot(days, days.delete, bty="n", type="l");
```



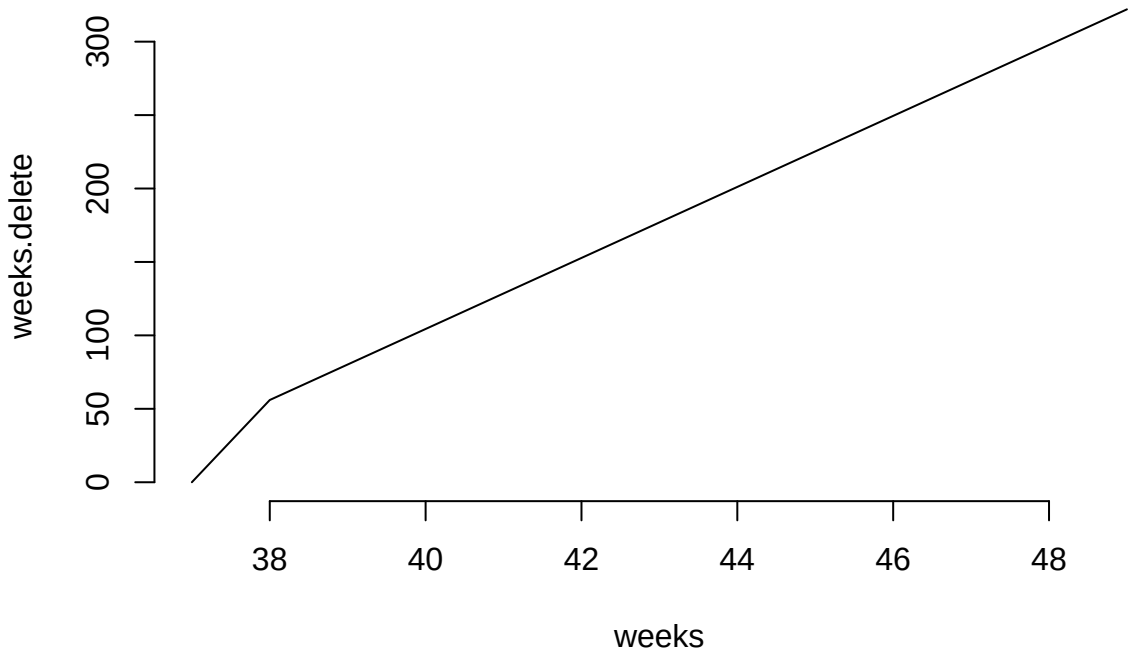

```
plot(weeks, weeks.file,   bty="n", type="l");  # 36 is August 30, 2020
```




```
plot(weeks, weeks.insert, bty="n", type="l");  # https://savvytime.com/week-number
```




```
plot(weeks, weeks.delete, bty="n", type="l");
```




```
################################################
### Feel free to improve on the graphics, if you desire, totally optional ...
```

Conclusive Summary

```
# you need to upload the .rds and .html file after you KNIT ...
saveRDS(my.object, paste0(path.to.git.final,"workspaceAudit.rds") );
```

This audit is worth 50 points. Based on your activity in the workspace, self-assess how many points you merit. Mastery is perfection at 50 points; Developing is about 40 points; Nascent is about 30 points. Justify your decision based on the consistency of your use of the tools and workspace throughout the semester.

Be certain to review the graphs and include self-reflective commentary on your work habits: day of week, hour of day, and so on.