

main

July 12, 2021

1 Run a Monte Carlo simulation to determine significant overlap of DEGs between multiple brain regions

```
[1]: library(tidyverse)
```

```
Attaching packages          tidyverse
1.3.1
```

```
ggplot2 3.3.5    purrr  0.3.4
tibble  3.1.2    dplyr  1.0.7
tidyr   1.1.3    stringr 1.4.0
readr   1.4.0    forcats 0.5.1
```

Conflicts

```
tidyverse_conflicts()
dplyr::filter() masks stats::filter()
dplyr::lag()    masks stats::lag()
```

1.1 Load overlap annotation

```
[2]: annot = data.table::fread("../_m/brainseq_deg_across_tissues_comparison.csv")
annot %>% head(2)
```

	gene_id <chr>	ensemblID <chr>	gene_name <chr>	seqname <chr>	gene_type <chr>
A data.table: 2 × 9	ENSG00000227232.5	ENSG00000227232	WASH7P	chr1	unprocessed_pseudoge
	ENSG00000279457.3	ENSG00000279457	FO538757.1	chr1	protein_coding

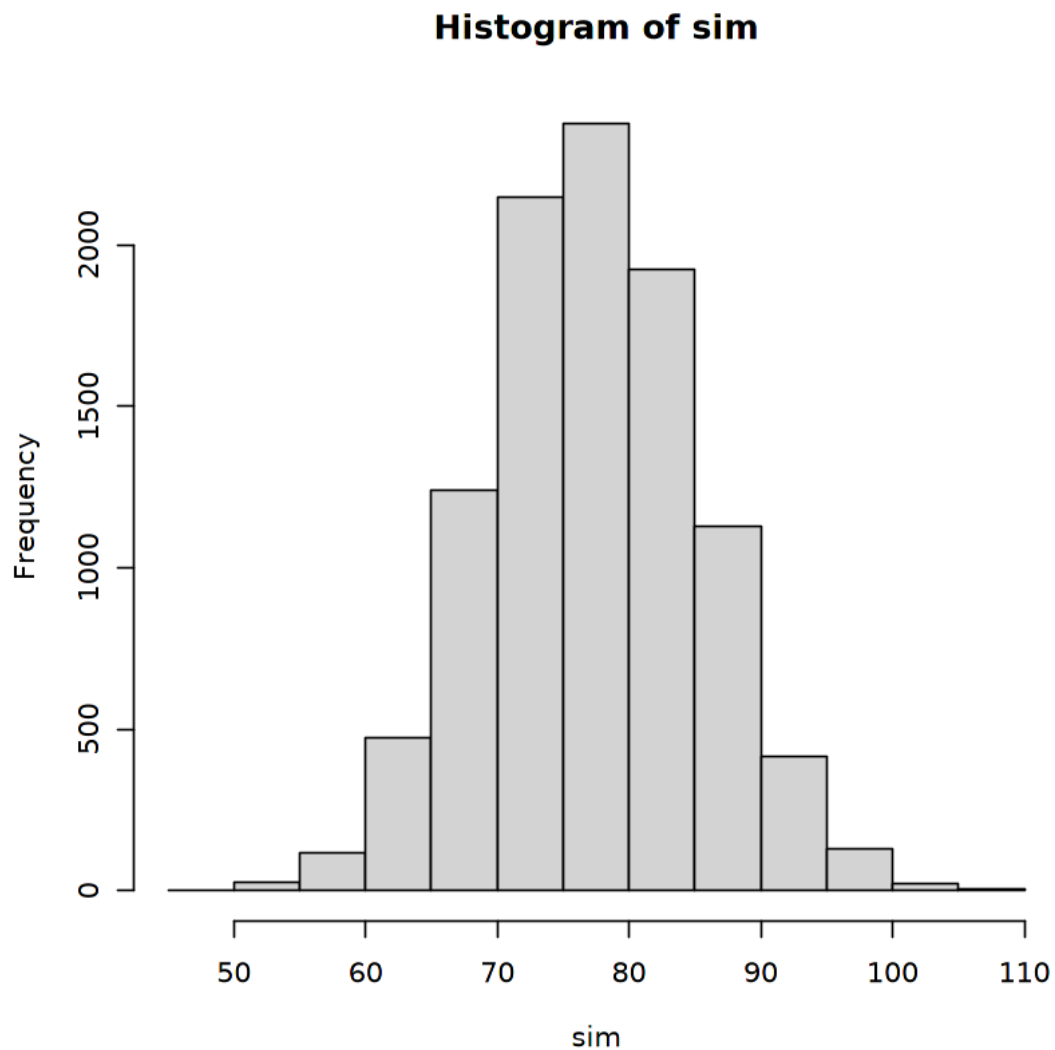
```
[3]: total = dim(annot)[1] ## Unique elements in the list
caudate = dim(filter(annot, Caudate == 1))[1]
dlpfc = dim(filter(annot, DLPFC == 1))[1]
hippocampus = dim(filter(annot, Hippocampus == 1))[1]
gyrus = dim(filter(annot, `Dentate Gyrus` == 1))[1]
```

1.2 Monte Carlo simulation

1.2.1 Four brain regions

```
[4]: sim = unlist(lapply(1:10000, function(i){  
  C=sample(1:total, caudate);  
  D=sample(1:total, dlpfc);  
  H=sample(1:total, hippocampus);  
  G=sample(1:total, gyrus);  
  return(length(Reduce(intersect, list(C,D,H,G))))  
}))
```

```
[5]: hist(sim)
```



```
[6]: shared = dim(filter(annot, Caudate == 1, DLPFC == 1,
                        Hippocampus == 1, `Dentate Gyrus` == 1))[1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

```
[1] "The max overlap from simulation: 108"
[1] "The overlap is: 198"
[1] "Enrichment p-value < 1e-04"
[1] "Depletion p-value < 1"
```

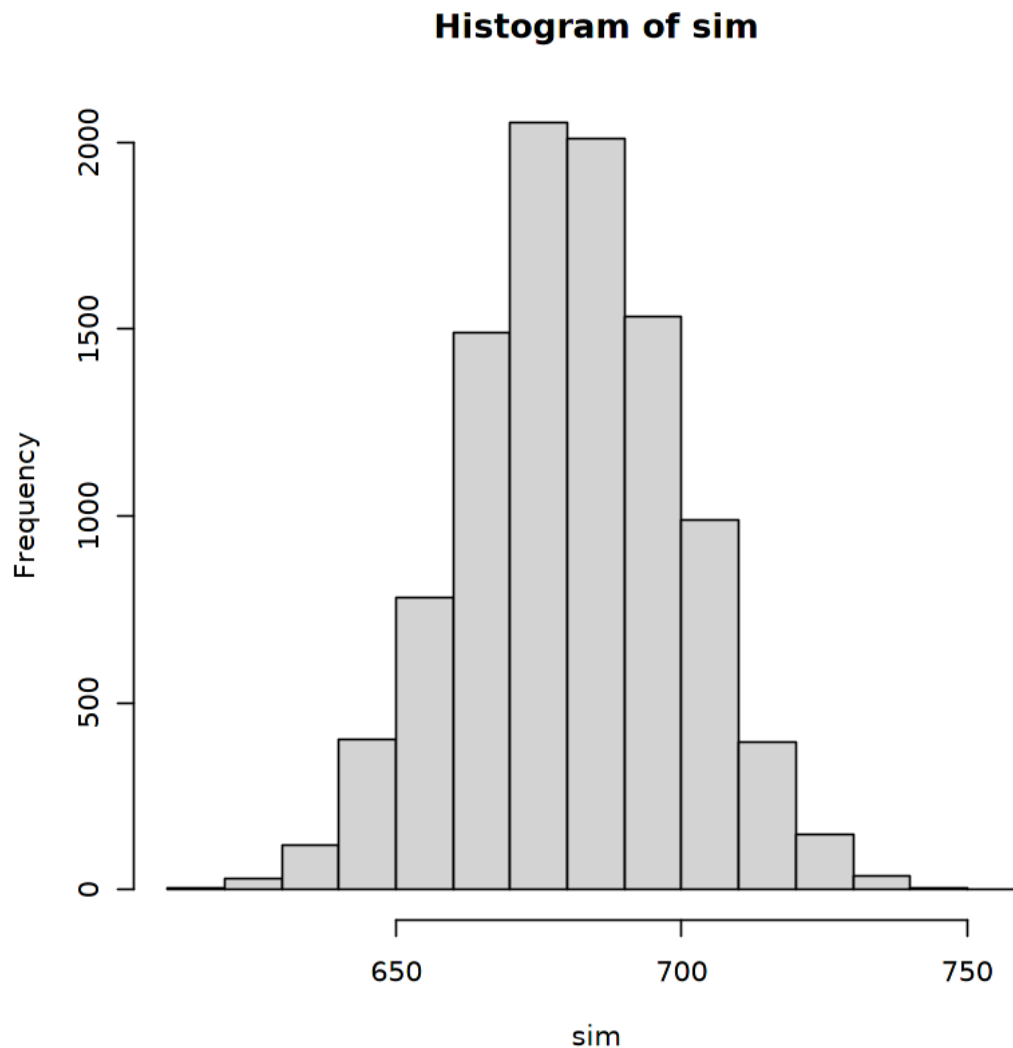
1.2.2 Three brain regions

Caudate vs DLPFC vs Hippocampus

```
[7]: total = Reduce(union, list(filter(annot, Caudate == 1)$gene_id,
                                     filter(annot, DLPFC == 1)$gene_id,
                                     filter(annot, Hippocampus == 1)$gene_id)) %>% length

sim = unlist(lapply(1:10000, function(i){
  C=sample(1:total, caudate);
  D=sample(1:total, dlpfc);
  H=sample(1:total, hippocampus);
  return(length(Reduce(intersect, list(C,D,H))))
})))

hist(sim)
```



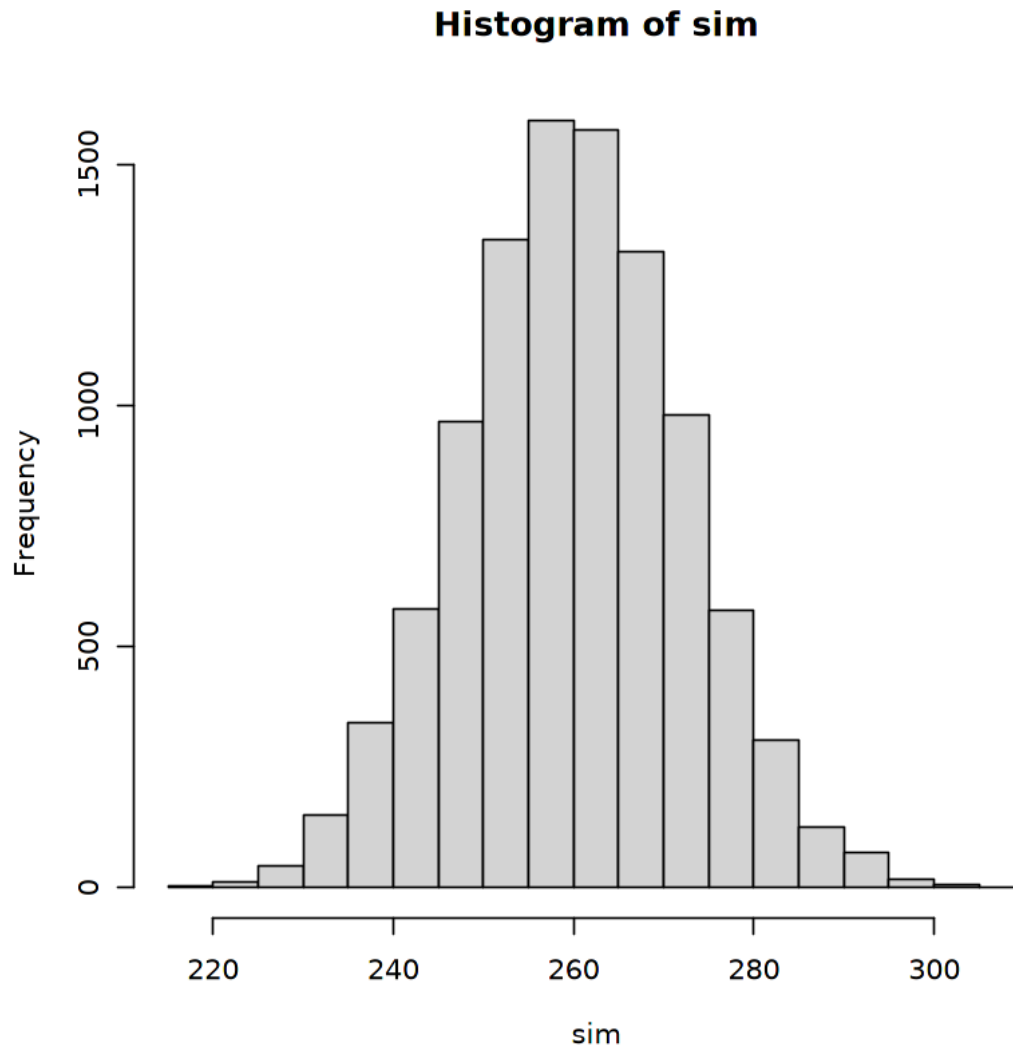
```
[8]: shared = dim(filter(annot, Caudate == 1, DLPFC == 1, Hippocampus == 1))[1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

```
[1] "The max overlap from simulation: 757"
[1] "The overlap is: 786"
[1] "Enrichment p-value < 1e-04"
```

```
[1] "Depletion p-value < 1"
```

Caudate vs DLPFC vs Dentate Gyrus

```
[9]: total = Reduce(union, list(filter(annot, Caudate == 1)$gene_id,  
                                filter(annot, DLPFC == 1)$gene_id,  
                                filter(annot, `Dentate Gyrus` == 1)$gene_id)) %>%  
  ↪length  
  
sim = unlist(lapply(1:10000, function(i){  
  C=sample(1:total, caudate);  
  D=sample(1:total, dlpfc);  
  G=sample(1:total, gyrus);  
  return(length(Reduce(intersect, list(C,D,G))))  
}))  
  
hist(sim)
```



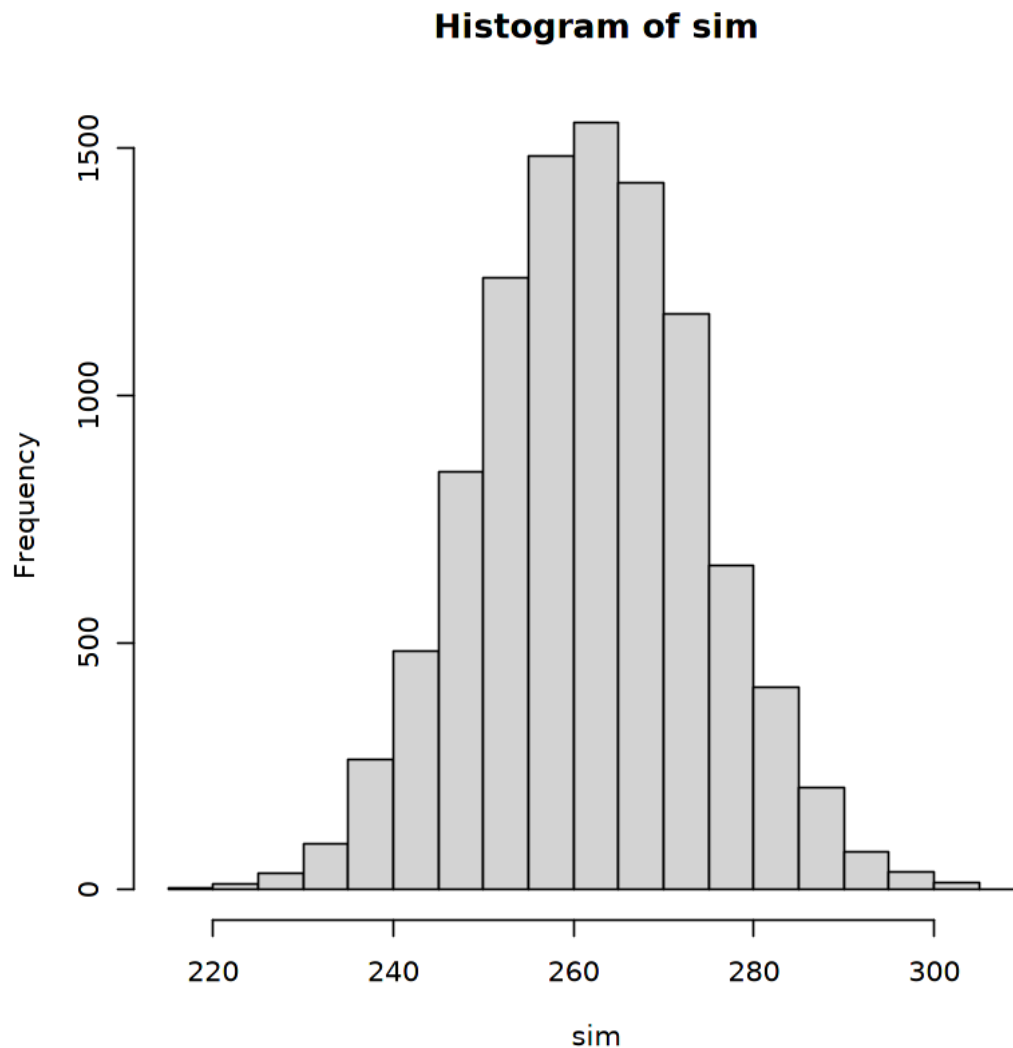
```
[10]: shared = dim(filter(annot, Caudate == 1, DLPFC == 1, `Dentate Gyrus` == 1))[1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

```
[1] "The max overlap from simulation: 306"
[1] "The overlap is: 225"
[1] "Enrichment p-value < 1"
```

```
[1] "Depletion p-value < 0.0015"
```

Caudate vs Hippocampus vs Dentate Gyrus

```
[11]: total = Reduce(union, list(filter(annot, Caudate == 1)$gene_id,  
                                filter(annot, Hippocampus == 1)$gene_id,  
                                filter(annot, `Dentate Gyrus` == 1)$gene_id)) %>%  
  ↪length  
  
sim = unlist(lapply(1:10000, function(i){  
  C=sample(1:total, caudate);  
  H=sample(1:total, hippocampus);  
  G=sample(1:total, gyrus);  
  return(length(Reduce(intersect, list(C,H,G))))  
}))  
  
hist(sim)
```



```
[12]: shared = dim(filter(annot, Caudate == 1, `Dentate Gyrus` == 1, Hippocampus == 1)) [1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

```
[1] "The max overlap from simulation: 308"
```

```
[1] "The overlap is: 232"
```



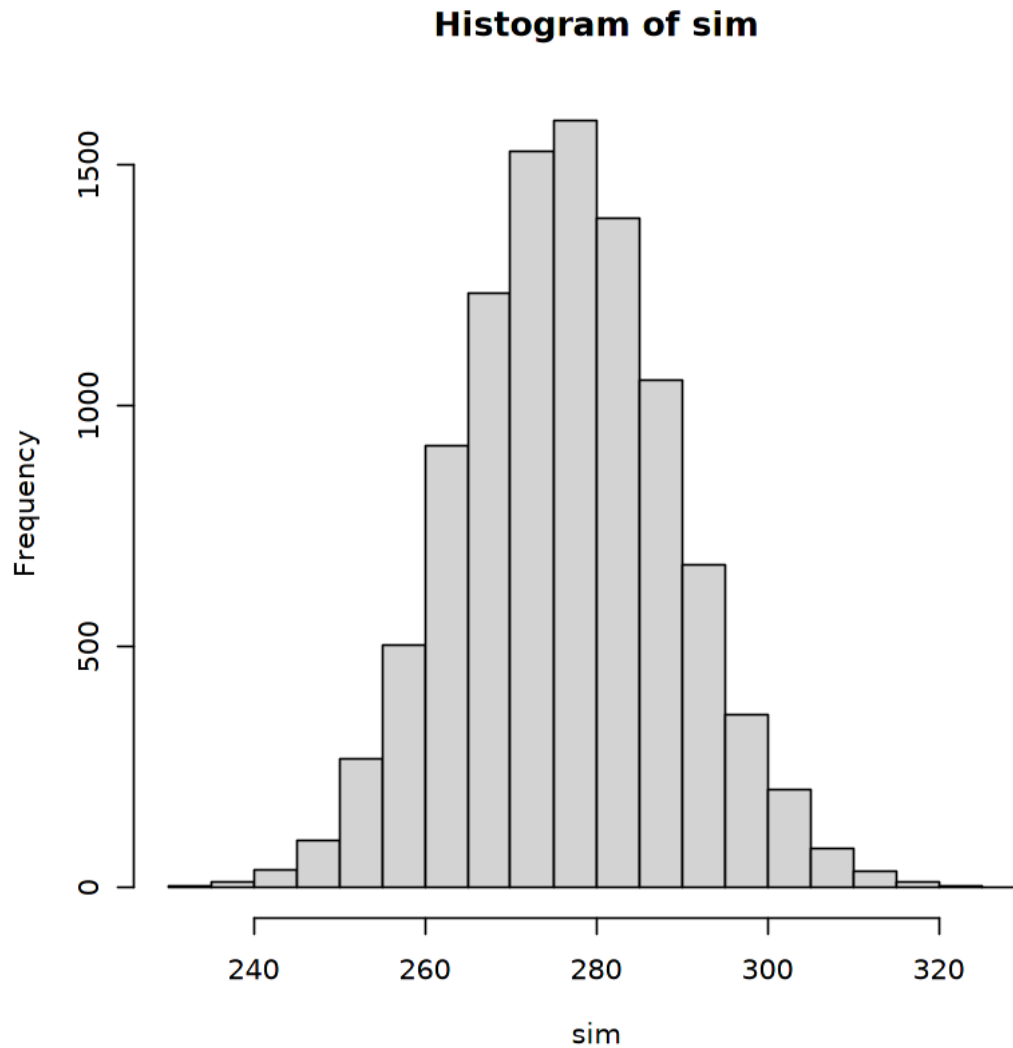
```
[1] "Enrichment p-value < 0.99"
[1] "Depletion p-value < 0.0074"
```

DLPFC vs Hippocampus vs Dentate Gyrus

```
[13]: total = Reduce(union, list(filter(annot, Hippocampus == 1)$gene_id,
                                   filter(annot, DLPFC == 1)$gene_id,
                                   filter(annot, `Dentate Gyrus` == 1)$gene_id)) %>%
  ↪length

sim = unlist(lapply(1:10000, function(i){
  D=sample(1:total, dlpfc);
  H=sample(1:total, hippocampus);
  G=sample(1:total, gyrus);
  return(length(Reduce(intersect, list(D,H,G))))
}))

hist(sim)
```



```
[14]: shared = dim(filter(annot, `Dentate Gyrus` == 1, DLPFC == 1, Hippocampus == 1)) [1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

```
[1] "The max overlap from simulation: 329"
```

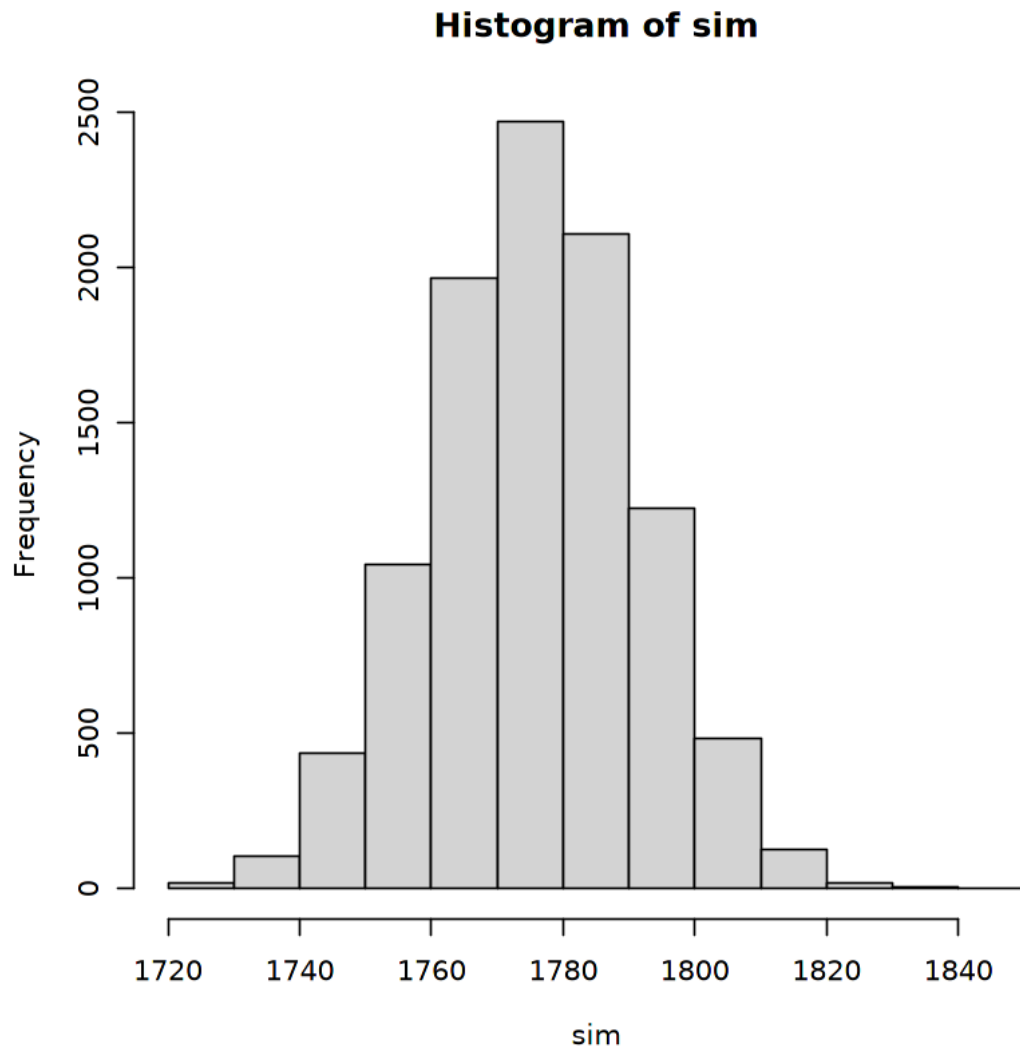
```
[1] "The overlap is: 264"
```

```
[1] "Enrichment p-value < 0.86"  
[1] "Depletion p-value < 0.16"
```

1.2.3 Two brain regions

Caudate vs DLPFC

```
[15]: total = Reduce(union, list(filter(annot, Caudate == 1)$gene_id,  
                                filter(annot, DLPFC == 1)$gene_id)) %>% length  
  
sim = unlist(lapply(1:10000, function(i){  
  C=sample(1:total, caudate);  
  D=sample(1:total, dlpfc);  
  return(length(Reduce(intersect, list(C,D))))  
}))  
  
hist(sim)
```



```
[16]: shared = dim(filter(annot, Caudate == 1, DLPFC == 1))[1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

```
[1] "The max overlap from simulation: 1848"
[1] "The overlap is: 1115"
[1] "Enrichment p-value < 1"
[1] "Depletion p-value < 1e-04"
```

Caudate vs Hippocampus

```
[17]: total = Reduce(union, list(filter(annot, Caudate == 1)$gene_id,
                                     filter(annot, Hippocampus == 1)$gene_id)) %>% length

sim = unlist(lapply(1:10000, function(i){
  C=sample(1:total, caudate);
  H=sample(1:total, hippocampus);
  return(length(Reduce(intersect, list(C,H))))
})))

hist(sim)
```



```
[18]: shared = dim(filter(annot, Caudate == 1, Hippocampus == 1))[1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

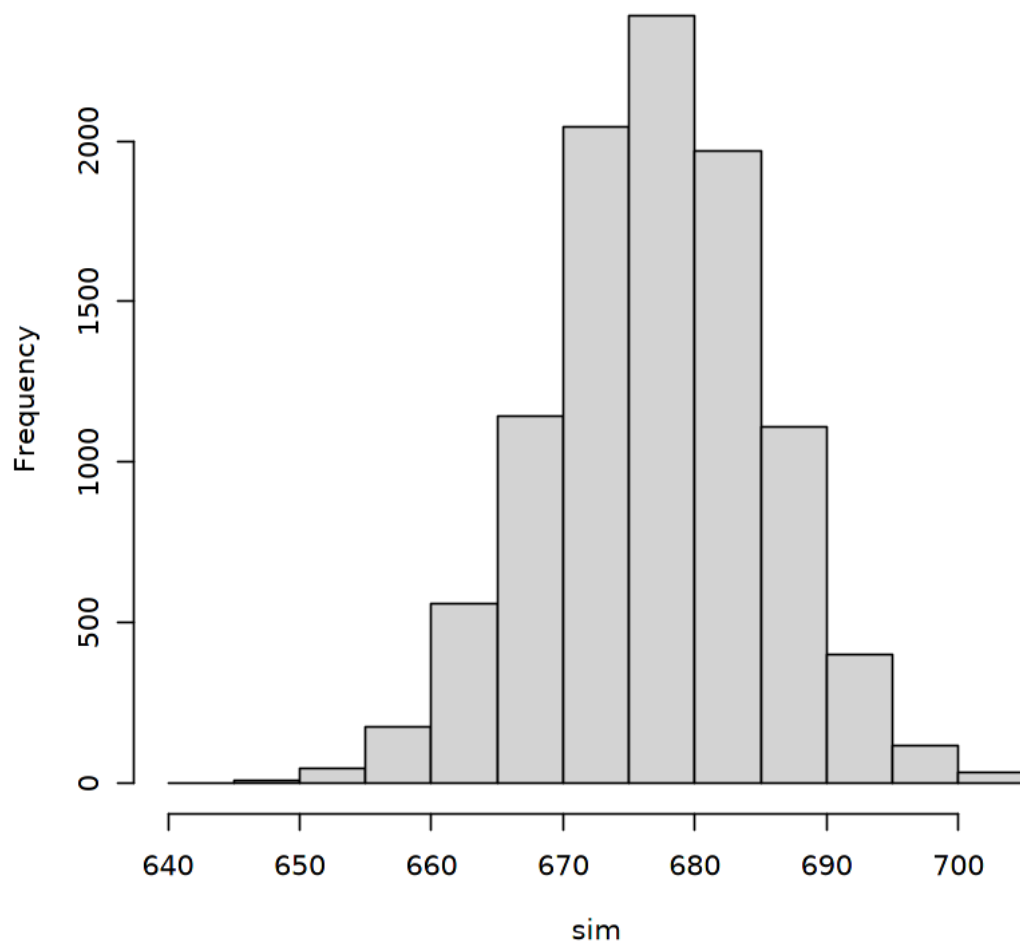
```
[1] "The max overlap from simulation: 1898"
[1] "The overlap is: 1142"
[1] "Enrichment p-value < 1"
```

```
[1] "Depletion p-value < 1e-04"
```

Caudate vs Dentate Gyrus

```
[19]: total = Reduce(union, list(filter(annot, Caudate == 1)$gene_id,  
                                filter(annot, `Dentate Gyrus` == 1)$gene_id)) %>%  
  ↪ length  
  
sim = unlist(lapply(1:10000, function(i){  
  C=sample(1:total, caudate);  
  G=sample(1:total, gyrus);  
  return(length(Reduce(intersect, list(C,G))))  
}))  
  
hist(sim)
```

Histogram of sim



```
[20]: shared = dim(filter(annot, Caudate == 1, `Dentate Gyrus` == 1))[1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

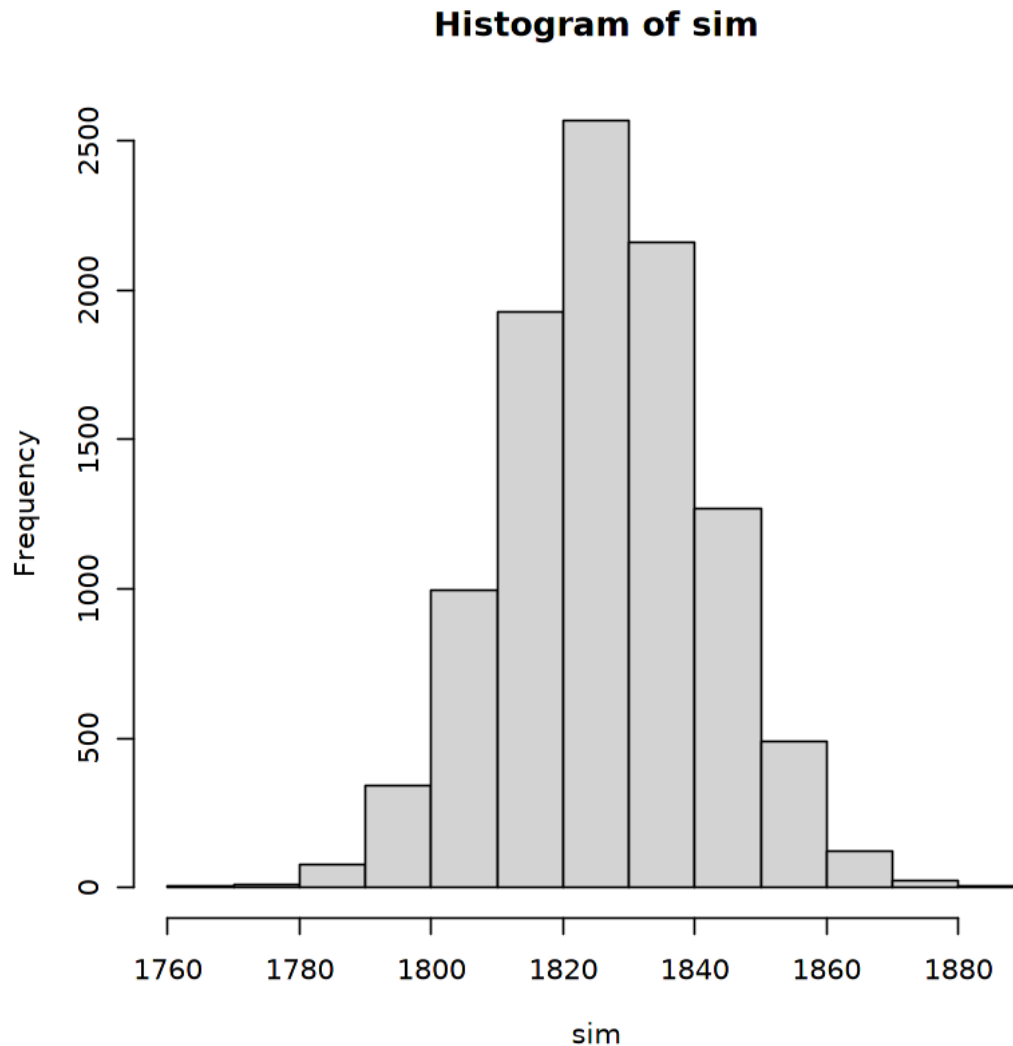
```
[1] "The max overlap from simulation: 705"
[1] "The overlap is: 311"
[1] "Enrichment p-value < 1"
[1] "Depletion p-value < 1e-04"
```

DLPFC vs Hippocampus

```
[21]: total = Reduce(union, list(filter(annot, DLPFC == 1)$gene_id,
                                     filter(annot, Hippocampus == 1)$gene_id)) %>% length

sim = unlist(lapply(1:10000, function(i){
  D=sample(1:total, dlpfc);
  H=sample(1:total, hippocampus);
  return(length(Reduce(intersect, list(D,H))))
}))

hist(sim)
```



```
[22]: shared = dim(filter(annot, DLPFC == 1, Hippocampus == 1))[1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

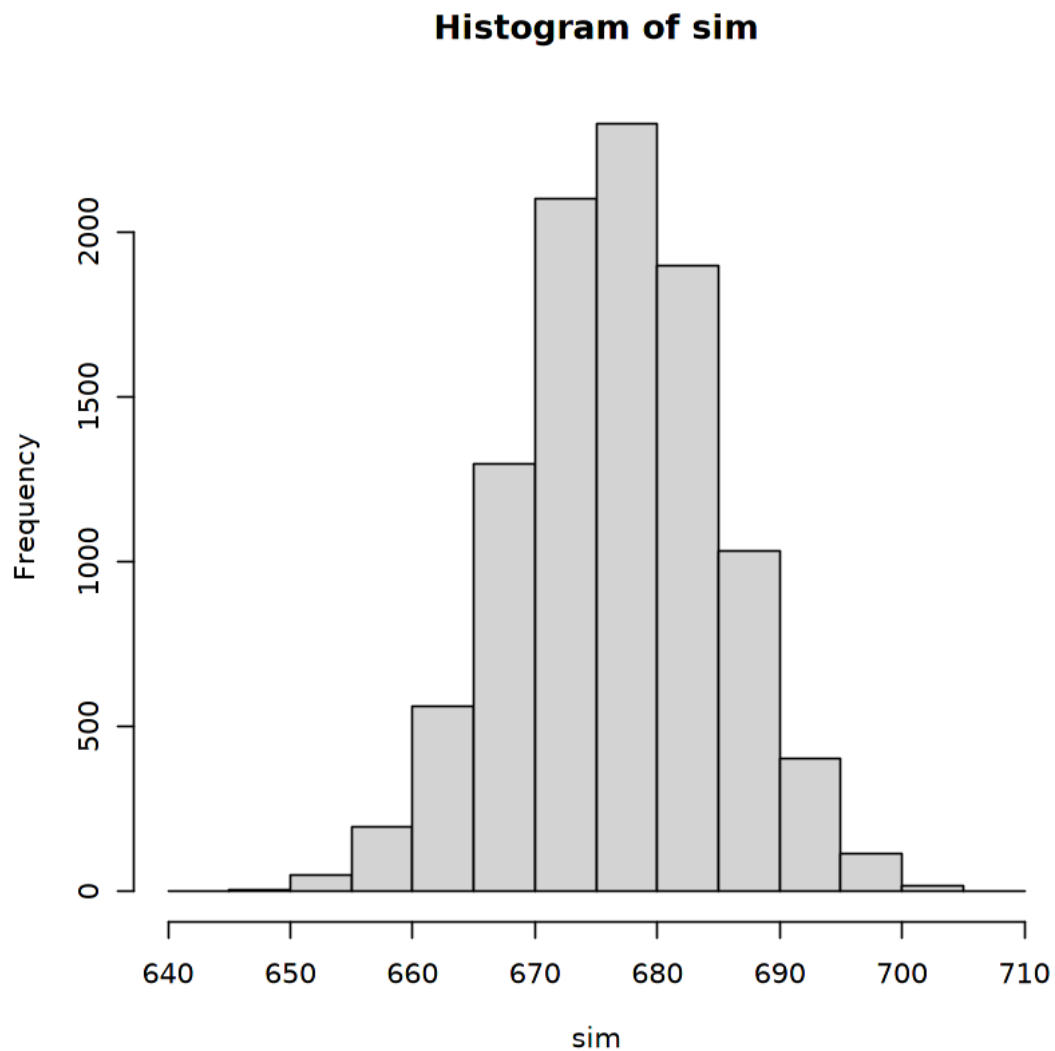
```
[1] "The max overlap from simulation: 1887"
[1] "The overlap is: 1251"
[1] "Enrichment p-value < 1"
```



```
[1] "Depletion p-value < 1e-04"
```

DLPFC vs Dentate Gyrus

```
[23]: total = Reduce(union, list(filter(annot, DLPFC == 1)$gene_id,  
                                filter(annot, `Dentate Gyrus` == 1)$gene_id)) %>%  
  ↪ length  
  
sim = unlist(lapply(1:10000, function(i){  
  D=sample(1:total, dlpfc);  
  G=sample(1:total, gyrus);  
  return(length(Reduce(intersect, list(D,G))))  
}))  
  
hist(sim)
```



```
[24]: shared = dim(filter(annot, DLPFC == 1, `Dentate Gyrus` == 1))[1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

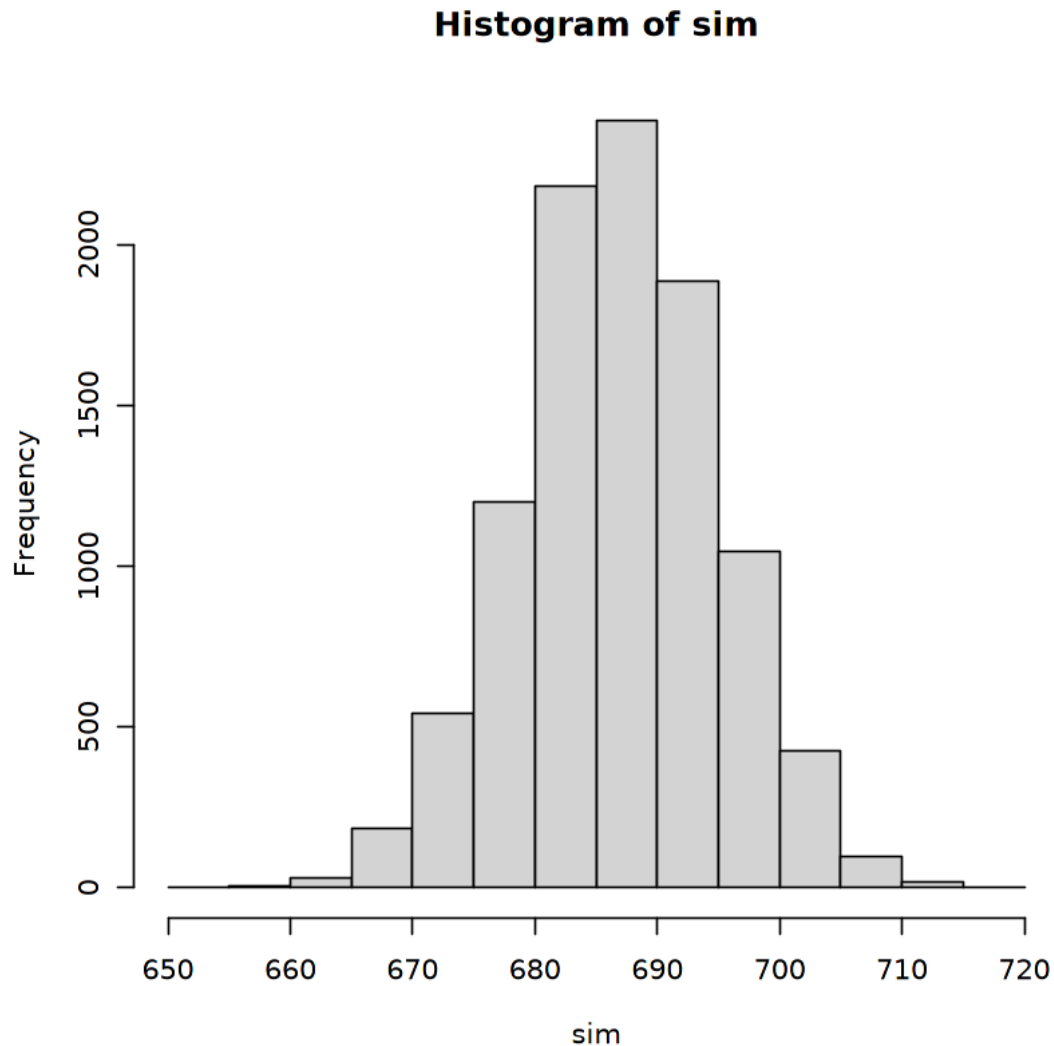
```
[1] "The max overlap from simulation: 710"
[1] "The overlap is: 342"
[1] "Enrichment p-value < 1"
[1] "Depletion p-value < 1e-04"
```

Hippocampus vs Dentate Gyrus

```
[25]: total = Reduce(union, list(filter(annot, Hippocampus == 1)$gene_id,
                                     filter(annot, `Dentate Gyrus` == 1)$gene_id)) %>%
  ↪length

sim = unlist(lapply(1:10000, function(i){
  H=sample(1:total, hippocampus);
  G=sample(1:total, gyrus);
  return(length(Reduce(intersect, list(H,G))))
})))

hist(sim)
```



```
[26]: shared = dim(filter(annot, Hippocampus == 1, `Dentate Gyrus` == 1))[1]
print(paste("The max overlap from simulation:", max(sim)))
print(paste("The overlap is:", shared))
## Zero instances are greater than overlap
## So pvalues is 1 / (10000 + 1), adding my value into this
pval1 = (sum(sim >= shared) + 1) / (10000 + 1)
pval2 = (sum(sim <= shared) + 1) / (10000 + 1)
print(paste("Enrichment p-value <", format(pval1, digits=2)))
print(paste("Depletion p-value <", format(pval2, digits=2)))
```

```
[1] "The max overlap from simulation: 717"
[1] "The overlap is: 361"
[1] "Enrichment p-value < 1"
```

```
[1] "Depletion p-value < 1e-04"
```

1.3 Reproducibility Information

```
[27]: print("Reproducibility Information:")
      Sys.time()
      proc.time()
      options(width=120)
      sessioninfo::session_info()
```

```
[1] "Reproducibility Information:"
```

```
[1] "2021-07-12 09:52:06 EDT"
```

```
      user  system elapsed
79.408    0.490   80.529
```

Session info

```
setting  value
version  R version 4.0.3 (2020-10-10)
os       Arch Linux
system   x86_64, linux-gnu
ui       X11
language (EN)
collate  en_US.UTF-8
ctype    en_US.UTF-8
tz       America/New_York
date     2021-07-12
```

Packages

package	* version	date	lib	source
assertthat	0.2.1	2019-03-21	[1]	CRAN (R 4.0.2)
backports	1.2.1	2020-12-09	[1]	CRAN (R 4.0.2)
base64enc	0.1-3	2015-07-28	[1]	CRAN (R 4.0.2)
broom	0.7.8	2021-06-24	[1]	CRAN (R 4.0.3)
Cairo	1.5-12.2	2020-07-07	[1]	CRAN (R 4.0.2)
cellranger	1.1.0	2016-07-27	[1]	CRAN (R 4.0.2)
cli	3.0.0	2021-06-30	[1]	CRAN (R 4.0.3)
colorspace	2.0-2	2021-06-24	[1]	CRAN (R 4.0.3)
crayon	1.4.1	2021-02-08	[1]	CRAN (R 4.0.3)
data.table	1.14.0	2021-02-21	[1]	CRAN (R 4.0.3)
DBI	1.1.1	2021-01-15	[1]	CRAN (R 4.0.2)
dbplyr	2.1.1	2021-04-06	[1]	CRAN (R 4.0.3)
digest	0.6.27	2020-10-24	[1]	CRAN (R 4.0.2)
dplyr	* 1.0.7	2021-06-18	[1]	CRAN (R 4.0.3)
ellipsis	0.3.2	2021-04-29	[1]	CRAN (R 4.0.3)
evaluate	0.14	2019-05-28	[1]	CRAN (R 4.0.2)
fansi	0.5.0	2021-05-25	[1]	CRAN (R 4.0.3)
forcats	* 0.5.1	2021-01-27	[1]	CRAN (R 4.0.2)

fs	1.5.0	2020-07-31	[1]	CRAN	(R 4.0.2)
generics	0.1.0	2020-10-31	[1]	CRAN	(R 4.0.2)
ggplot2	* 3.3.5	2021-06-25	[1]	CRAN	(R 4.0.3)
glue	1.4.2	2020-08-27	[1]	CRAN	(R 4.0.2)
gtable	0.3.0	2019-03-25	[1]	CRAN	(R 4.0.2)
haven	2.4.1	2021-04-23	[1]	CRAN	(R 4.0.3)
hms	1.1.0	2021-05-17	[1]	CRAN	(R 4.0.3)
htmltools	0.5.1.1	2021-01-22	[1]	CRAN	(R 4.0.2)
httr	1.4.2	2020-07-20	[1]	CRAN	(R 4.0.2)
IRdisplay	1.0	2021-01-20	[1]	CRAN	(R 4.0.2)
IRkernel	1.2	2021-05-11	[1]	CRAN	(R 4.0.3)
jsonlite	1.7.2	2020-12-09	[1]	CRAN	(R 4.0.2)
lifecycle	1.0.0	2021-02-15	[1]	CRAN	(R 4.0.3)
lubridate	1.7.10	2021-02-26	[1]	CRAN	(R 4.0.3)
magrittr	2.0.1	2020-11-17	[1]	CRAN	(R 4.0.2)
modelr	0.1.8	2020-05-19	[1]	CRAN	(R 4.0.2)
munsell	0.5.0	2018-06-12	[1]	CRAN	(R 4.0.2)
pbdZMQ	0.3-5	2021-02-10	[1]	CRAN	(R 4.0.3)
pillar	1.6.1	2021-05-16	[1]	CRAN	(R 4.0.3)
pkgconfig	2.0.3	2019-09-22	[1]	CRAN	(R 4.0.2)
purrr	* 0.3.4	2020-04-17	[1]	CRAN	(R 4.0.2)
R6	2.5.0	2020-10-28	[1]	CRAN	(R 4.0.2)
Rcpp	1.0.7	2021-07-07	[1]	CRAN	(R 4.0.3)
readr	* 1.4.0	2020-10-05	[1]	CRAN	(R 4.0.2)
readxl	1.3.1	2019-03-13	[1]	CRAN	(R 4.0.2)
repr	1.1.3	2021-01-21	[1]	CRAN	(R 4.0.2)
reprex	2.0.0	2021-04-02	[1]	CRAN	(R 4.0.3)
rlang	0.4.11	2021-04-30	[1]	CRAN	(R 4.0.3)
rstudioapi	0.13	2020-11-12	[1]	CRAN	(R 4.0.2)
rvest	1.0.0	2021-03-09	[1]	CRAN	(R 4.0.3)
scales	1.1.1	2020-05-11	[1]	CRAN	(R 4.0.2)
sessioninfo	1.1.1	2018-11-05	[1]	CRAN	(R 4.0.2)
stringi	1.6.2	2021-05-17	[1]	CRAN	(R 4.0.3)
stringr	* 1.4.0	2019-02-10	[1]	CRAN	(R 4.0.2)
tibble	* 3.1.2	2021-05-16	[1]	CRAN	(R 4.0.3)
tidyr	* 1.1.3	2021-03-03	[1]	CRAN	(R 4.0.3)
tidyselect	1.1.1	2021-04-30	[1]	CRAN	(R 4.0.3)
tidyverse	* 1.3.1	2021-04-15	[1]	CRAN	(R 4.0.3)
utf8	1.2.1	2021-03-12	[1]	CRAN	(R 4.0.3)
uuid	0.1-4	2020-02-26	[1]	CRAN	(R 4.0.2)
vctrs	0.3.8	2021-04-29	[1]	CRAN	(R 4.0.3)
withr	2.4.2	2021-04-18	[1]	CRAN	(R 4.0.3)
xml2	1.3.2	2020-04-23	[1]	CRAN	(R 4.0.2)

[1] /home/jbenja13/R/x86_64-pc-linux-gnu-library/4.0

[2] /usr/lib/R/library