Correspondence Analysis of Plant Remains from Madjedbebe

Anna Florin

Ben Marwick

07 June, 2020

### Read in the data and prepare it

cordata <- readxl::read\_excel(here::here("analysis/data/raw\_data/Correspondence Analysis.xlsx"))  
  
library(FactoMineR)  
library(factoextra)  
library(tidyverse)  
  
# select numeric columns  
cordata\_num <-   
cordata %>%   
 unite(col = "ellipse", c(`Phase-num`, `context-type`)) %>%   
 dplyr::select(where(is.numeric), ellipse)

### Compute the correspondence analysis

library(ca)  
  
ca.fit <- ca(cordata\_num[-ncol(cordata\_num)])  
ca.plot <- plot(ca.fit)  
  
ca.dims1 <- tibble(dim1 = ca.plot$rows[,1],   
 dim2 = ca.plot$rows[,2],  
 Phase = as.factor(cordata$`Phase-num`),  
 Context = cordata$`context-type`)  
  
ca.dims2 <- tibble(dim1 = ca.plot$cols[,1],   
 dim2 = ca.plot$cols[,2],  
 labels = colnames(cordata\_num)[-ncol(cordata\_num)])

### Visualise the CA with plots

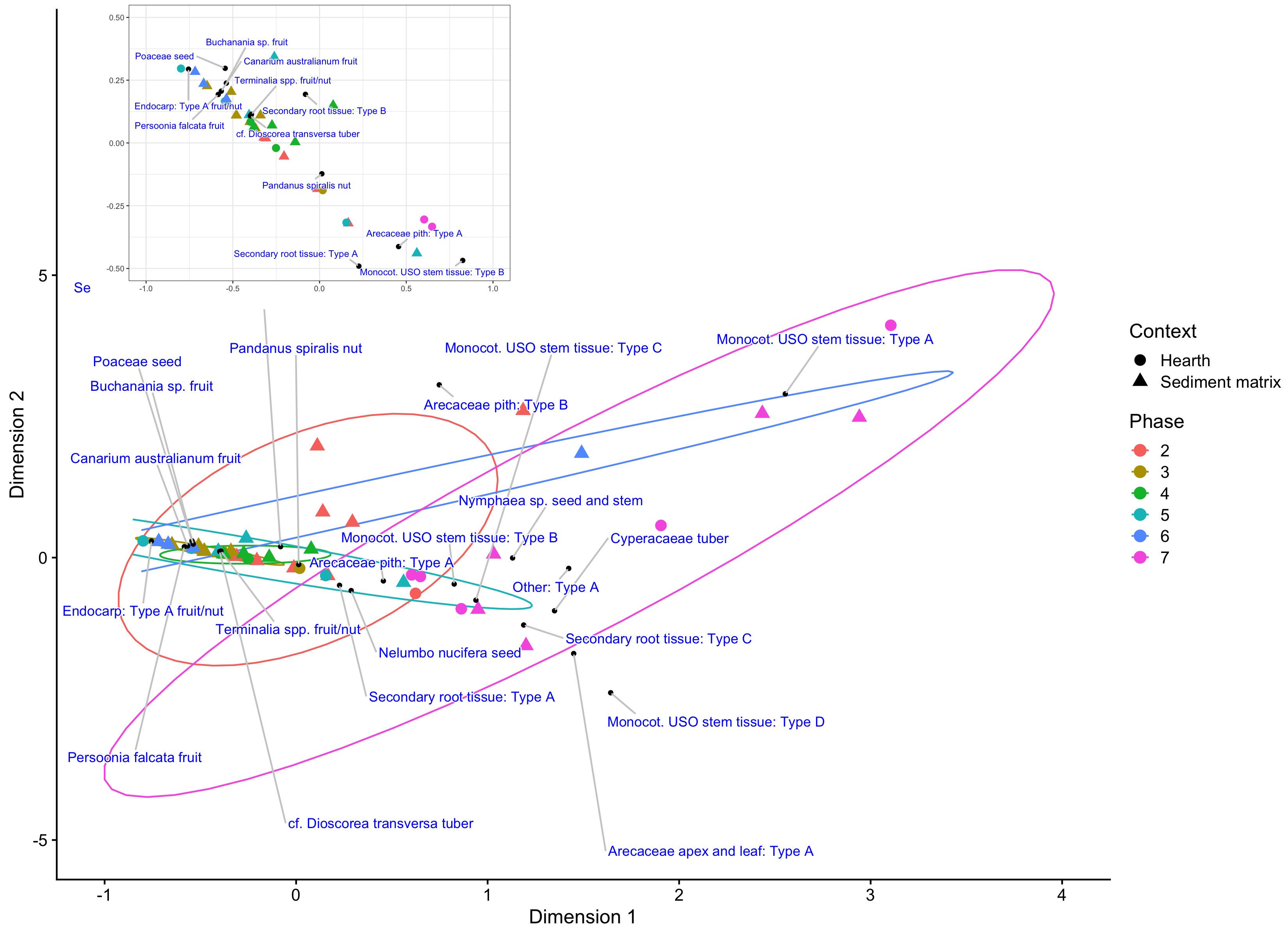
library(ggrepel)  
ca\_plot\_with\_ellipse <-   
ggplot() +  
 stat\_ellipse(data = ca.dims1,  
 aes(dim1,  
 dim2,  
 colour = Phase)) +  
 geom\_point(data = ca.dims1,  
 aes(dim1,  
 dim2,  
 shape = Context,  
 colour = Phase),  
 size = 3) +  
 geom\_point(data = ca.dims2,  
 aes(dim1,  
 dim2),  
 colour = "black",  
 shape = 20) +  
 geom\_text\_repel(data = ca.dims2,  
 aes(dim1,  
 dim2,  
 label = labels),  
 size = 3,  
 bg.colour = "white",  
 bg.r = 0.1,  
 force = 100,  
 segment.color = "grey80",  
 max.overlaps = 20,  
 min.segment.length = 0.1,  
 colour = "blue") +  
 labs(x = "Dimension 1",  
 y = "Dimension 2") +  
 scale\_colour\_discrete() +  
 scale\_x\_continuous(limits = c(-1, 4)) +  
 scale\_y\_continuous(limits = c(-5, 9)) +  
 theme\_minimal()  
  
ggsave(here::here("analysis/figures/ca-ellipse-plot.png"),  
 dpi = 300)  
  
ggsave(here::here("analysis/figures/ca-ellipse-plot.svg"))

# checking the two plot methods  
library(cowplot)  
  
plot\_grid(ca\_plot\_with\_ellipse,   
 ~{plot(ca.fit)})

ca\_plot\_zoom <-   
 ggplot() +  
 geom\_point(data = ca.dims1,  
 aes(dim1,  
 dim2,  
 shape = Context,  
 colour = Phase),  
 size = 2) +  
 geom\_point(data = ca.dims2,  
 aes(dim1,  
 dim2),  
 colour = "black",  
 shape = 20) +  
 geom\_text\_repel(data = ca.dims2,  
 aes(dim1,  
 dim2,  
 label = labels),  
 size = 2,  
 bg.colour = "white",  
 bg.r = 0.1,  
 force = 100,  
 segment.color = "grey80",  
 max.overlaps = 20,  
 min.segment.length = 0.1,  
 colour = "blue") +  
 scale\_colour\_discrete() +  
 theme\_minimal() +  
 xlim(-1, 1.0) +   
 ylim(-0.5, 0.5) +  
 labs(x = "",  
 y = "") +  
 guides(colour = FALSE,  
 shape = FALSE) +  
 theme\_bw(base\_size = 6)  
  
ggsave(here::here("analysis/figures/ca-zoom-plot.png"),  
 dpi = 300)  
  
ggsave(here::here("analysis/figures/ca-zoom-plot.svg"))

# combine plots  
library(cowplot)  
  
ggdraw(ca\_plot\_with\_ellipse + theme\_half\_open(12)) +  
 draw\_plot(ca\_plot\_zoom,   
 x = 0.07,   
 y = 0.67,   
 width = 0.33,   
 height = 0.33)   
  
ggsave(here::here("analysis/figures/ca-inset-plot.png"),  
 height = 8,  
 width = 11,  
 dpi = 300)  
  
ggsave(here::here("analysis/figures/ca-inset-plot.svg"),  
 height = 8,  
 width = 11)

knitr::include\_graphics(here::here("analysis/figures/ca-inset-plot.png"))



# Other methods of CA that we explored

library(anacor)  
  
ndim <- 1  
res <- anacor(cordata\_num[-ncol(cordata\_num)],   
 #ndim = ndim,   
 scaling = rep("Benzecri", ndim))  
res  
plot(res)  
  
  
anacor.dims1 <- tibble(dim1 = res$row.scores[,1],   
 dim2 = res$row.scores[,2],  
 Phase = as.factor(cordata$`Phase-num`),  
 Context = cordata$`context-type`)  
  
anacor.dims2 <- tibble(dim1 = res$col.scores [,1],   
 dim2 = res$col.scores [,2],  
 labels = colnames(cordata\_num)[-ncol(cordata\_num)])  
  
library(ggrepel)  
anacor\_plot\_with\_ellipse <-   
ggplot() +  
 stat\_ellipse(data = anacor.dims1,  
 aes(dim1,  
 dim2,  
 colour = Phase)) +  
 geom\_point(data = anacor.dims1,  
 aes(dim1,  
 dim2,  
 shape = Context,  
 colour = Phase),  
 size = 3) +  
 geom\_point(data = anacor.dims2,  
 aes(dim1,  
 dim2),  
 colour = "black",  
 shape = 20) +  
 geom\_text\_repel(data = anacor.dims2,  
 aes(dim1,  
 dim2,  
 label = labels),  
 size = 3,  
 bg.colour = "white",  
 bg.r = 0.1,  
 force = 100,  
 segment.color = "grey80",  
 max.overlaps = 20,  
 min.segment.length = 0.1,  
 colour = "blue") +  
 labs(x = "Dimension 1",  
 y = "Dimension 2") +  
 scale\_colour\_discrete() +  
 coord\_fixed() +  
 theme\_minimal()

library(CAinterprTools)  
  
caPlot\_data <-   
caPlot(cordata\_num[-ncol(cordata\_num)],1,2,adv.labls=FALSE)  
  
caPlot.dims1 <- tibble(dim1 = caPlot\_data[caPlot\_data$Categories == 'rows',]$coord.1Dim,   
 dim2 = caPlot\_data[caPlot\_data$Categories == 'rows',]$coord.2Dim,  
 Phase = as.factor(cordata$`Phase-num`),  
 Context = cordata$`context-type`)  
  
caPlot.dims2 <- tibble(dim1 = caPlot\_data[caPlot\_data$Categories == 'columns',]$coord.1Dim,   
 dim2 = caPlot\_data[caPlot\_data$Categories == 'columns',]$coord.2Dim,  
 labels = colnames(cordata\_num)[-ncol(cordata\_num)])  
  
library(ggrepel)  
caPlot\_plot\_with\_ellipse <-   
ggplot() +  
 stat\_ellipse(data = caPlot.dims1,  
 aes(dim1,  
 dim2,  
 colour = Phase)) +  
 geom\_point(data = caPlot.dims1,  
 aes(dim1,  
 dim2,  
 shape = Context,  
 colour = Phase),  
 size = 3) +  
 geom\_point(data = caPlot.dims2,  
 aes(dim1,  
 dim2),  
 colour = "black",  
 shape = 20) +  
 geom\_text\_repel(data = caPlot.dims2,  
 aes(dim1,  
 dim2,  
 label = labels),  
 size = 3,  
 bg.colour = "white",  
 bg.r = 0.1,  
 force = 100,  
 segment.color = "grey80",  
 max.overlaps = 20,  
 min.segment.length = 0.1,  
 colour = "blue") +  
 labs(x = "Dimension 1",  
 y = "Dimension 2") +  
 scale\_colour\_discrete() +  
 coord\_fixed() +  
 theme\_minimal()

### Colophon

This report was generated on 2020-06-07 15:46:52 using the following computational environment and dependencies:

# which R packages and versions?  
if ("devtools" %in% installed.packages()) devtools::session\_info()  
#> ─ Session info ───────────────────────────────────────────────────────────────  
#> setting value   
#> version R version 4.0.0 (2020-04-24)  
#> os macOS Catalina 10.15.2   
#> system x86\_64, darwin17.0   
#> ui X11   
#> language (EN)   
#> collate en\_US.UTF-8   
#> ctype en\_US.UTF-8   
#> tz America/Los\_Angeles   
#> date 2020-06-07   
#>   
#> ─ Packages ───────────────────────────────────────────────────────────────────  
#> package \* version date lib source   
#> assertthat 0.2.1 2019-03-21 [1] CRAN (R 4.0.0)   
#> backports 1.1.7 2020-05-13 [1] CRAN (R 4.0.0)   
#> blob 1.2.1 2020-01-20 [1] CRAN (R 4.0.0)   
#> bookdown 0.19 2020-05-15 [1] CRAN (R 4.0.0)   
#> broom 0.5.6 2020-04-20 [1] CRAN (R 4.0.0)   
#> ca \* 0.71.1 2020-01-24 [1] CRAN (R 4.0.0)   
#> callr 3.4.3 2020-03-28 [1] CRAN (R 4.0.0)   
#> cellranger 1.1.0 2016-07-27 [1] CRAN (R 4.0.0)   
#> cli 2.0.2 2020-02-28 [1] CRAN (R 4.0.0)   
#> cluster 2.1.0 2019-06-19 [1] CRAN (R 4.0.0)   
#> colorspace 1.4-1 2019-03-18 [1] CRAN (R 4.0.0)   
#> cowplot \* 1.0.0 2019-07-11 [1] CRAN (R 4.0.0)   
#> crayon 1.3.4 2017-09-16 [1] CRAN (R 4.0.0)   
#> DBI 1.1.0 2019-12-15 [1] CRAN (R 4.0.0)   
#> dbplyr 1.4.4 2020-05-27 [1] CRAN (R 4.0.0)   
#> desc 1.2.0 2018-05-01 [1] CRAN (R 4.0.0)   
#> devtools 2.3.0 2020-04-10 [1] CRAN (R 4.0.0)   
#> digest 0.6.25 2020-02-23 [1] CRAN (R 4.0.0)   
#> dplyr \* 1.0.0 2020-05-29 [1] CRAN (R 4.0.0)   
#> ellipsis 0.3.1 2020-05-15 [1] CRAN (R 4.0.0)   
#> evaluate 0.14 2019-05-28 [1] CRAN (R 4.0.0)   
#> factoextra \* 1.0.7 2020-04-01 [1] CRAN (R 4.0.0)   
#> FactoMineR \* 2.3 2020-02-29 [1] CRAN (R 4.0.0)   
#> fansi 0.4.1 2020-01-08 [1] CRAN (R 4.0.0)   
#> farver 2.0.3 2020-01-16 [1] CRAN (R 4.0.0)   
#> flashClust 1.01-2 2012-08-21 [1] CRAN (R 4.0.0)   
#> forcats \* 0.5.0 2020-03-01 [1] CRAN (R 4.0.0)   
#> fs 1.4.1 2020-04-04 [1] CRAN (R 4.0.0)   
#> gdtools \* 0.2.2 2020-04-03 [1] CRAN (R 4.0.0)   
#> generics 0.0.2 2018-11-29 [1] CRAN (R 4.0.0)   
#> ggplot2 \* 3.3.1 2020-05-28 [1] CRAN (R 4.0.0)   
#> ggrepel \* 0.9.0 2020-05-11 [1] Github (slowkow/ggrepel@3941cf1)  
#> glue 1.4.1 2020-05-13 [1] CRAN (R 4.0.0)   
#> gridGraphics 0.5-0 2020-02-25 [1] CRAN (R 4.0.0)   
#> gtable 0.3.0 2019-03-25 [1] CRAN (R 4.0.0)   
#> haven 2.3.0 2020-05-24 [1] CRAN (R 4.0.0)   
#> here 0.1 2017-05-28 [1] CRAN (R 4.0.0)   
#> hms 0.5.3 2020-01-08 [1] CRAN (R 4.0.0)   
#> htmltools 0.4.0 2019-10-04 [1] CRAN (R 4.0.0)   
#> httr 1.4.1 2019-08-05 [1] CRAN (R 4.0.0)   
#> jsonlite 1.6.1 2020-02-02 [1] CRAN (R 4.0.0)   
#> knitr 1.28 2020-02-06 [1] CRAN (R 4.0.0)   
#> labeling 0.3 2014-08-23 [1] CRAN (R 4.0.0)   
#> lattice 0.20-41 2020-04-02 [1] CRAN (R 4.0.0)   
#> leaps 3.1 2020-01-16 [1] CRAN (R 4.0.0)   
#> lifecycle 0.2.0 2020-03-06 [1] CRAN (R 4.0.0)   
#> lubridate 1.7.8 2020-04-06 [1] CRAN (R 4.0.0)   
#> magrittr 1.5 2014-11-22 [1] CRAN (R 4.0.0)   
#> MASS 7.3-51.6 2020-04-26 [1] CRAN (R 4.0.0)   
#> memoise 1.1.0 2017-04-21 [1] CRAN (R 4.0.0)   
#> modelr 0.1.8 2020-05-19 [1] CRAN (R 4.0.0)   
#> munsell 0.5.0 2018-06-12 [1] CRAN (R 4.0.0)   
#> nlme 3.1-148 2020-05-24 [1] CRAN (R 4.0.0)   
#> pillar 1.4.4 2020-05-05 [1] CRAN (R 4.0.0)   
#> pkgbuild 1.0.8 2020-05-07 [1] CRAN (R 4.0.0)   
#> pkgconfig 2.0.3 2019-09-22 [1] CRAN (R 4.0.0)   
#> pkgload 1.1.0 2020-05-29 [1] CRAN (R 4.0.0)   
#> png 0.1-7 2013-12-03 [1] CRAN (R 4.0.0)   
#> prettyunits 1.1.1 2020-01-24 [1] CRAN (R 4.0.0)   
#> processx 3.4.2 2020-02-09 [1] CRAN (R 4.0.0)   
#> ps 1.3.3 2020-05-08 [1] CRAN (R 4.0.0)   
#> purrr \* 0.3.4 2020-04-17 [1] CRAN (R 4.0.0)   
#> R6 2.4.1 2019-11-12 [1] CRAN (R 4.0.0)   
#> Rcpp 1.0.4.6 2020-04-09 [1] CRAN (R 4.0.0)   
#> readr \* 1.3.1 2018-12-21 [1] CRAN (R 4.0.0)   
#> readxl 1.3.1 2019-03-13 [1] CRAN (R 4.0.0)   
#> remotes 2.1.1 2020-02-15 [1] CRAN (R 4.0.0)   
#> reprex 0.3.0 2019-05-16 [1] CRAN (R 4.0.0)   
#> rlang 0.4.6 2020-05-02 [1] CRAN (R 4.0.0)   
#> rmarkdown 2.2 2020-05-31 [1] CRAN (R 4.0.0)   
#> rprojroot 1.3-2 2018-01-03 [1] CRAN (R 4.0.0)   
#> rstudioapi 0.11 2020-02-07 [1] CRAN (R 4.0.0)   
#> rvest 0.3.5 2019-11-08 [1] CRAN (R 4.0.0)   
#> scales 1.1.1 2020-05-11 [1] CRAN (R 4.0.0)   
#> scatterplot3d 0.3-41 2018-03-14 [1] CRAN (R 4.0.0)   
#> sessioninfo 1.1.1 2018-11-05 [1] CRAN (R 4.0.0)   
#> stringi 1.4.6 2020-02-17 [1] CRAN (R 4.0.0)   
#> stringr \* 1.4.0 2019-02-10 [1] CRAN (R 4.0.0)   
#> svglite 1.2.3 2020-02-07 [1] CRAN (R 4.0.0)   
#> systemfonts 0.2.2 2020-05-14 [1] CRAN (R 4.0.0)   
#> testthat 2.3.2 2020-03-02 [1] CRAN (R 4.0.0)   
#> tibble \* 3.0.1 2020-04-20 [1] CRAN (R 4.0.0)   
#> tidyr \* 1.1.0 2020-05-20 [1] CRAN (R 4.0.0)   
#> tidyselect 1.1.0 2020-05-11 [1] CRAN (R 4.0.0)   
#> tidyverse \* 1.3.0 2019-11-21 [1] CRAN (R 4.0.0)   
#> usethis 1.6.1 2020-04-29 [1] CRAN (R 4.0.0)   
#> vctrs 0.3.0 2020-05-11 [1] CRAN (R 4.0.0)   
#> withr 2.2.0 2020-04-20 [1] CRAN (R 4.0.0)   
#> xfun 0.14 2020-05-20 [1] CRAN (R 4.0.0)   
#> xml2 1.3.2 2020-04-23 [1] CRAN (R 4.0.0)   
#> yaml 2.2.1 2020-02-01 [1] CRAN (R 4.0.0)   
#>   
#> [1] /Library/Frameworks/R.framework/Versions/4.0/Resources/library

The current Git commit details are:

# what commit is this file at?   
if ("git2r" %in% installed.packages() & git2r::in\_repository(path = ".")) git2r::repository(here::here())