TROST Select: LASSO Model

Heike Sprenger

07/06/2020

Contents

Setup

Load functions

```
source("func_prep_pca.R")
source("func_remove_factors.R")
```

Load data

Tolerance information

```
tolerance <- read.table("tolerance.txt", sep = "\t", header = T)
```

Transcript data

TROST: 202 samples, 42 transcripts

all data: 1159 samples

```
# ALL
transcript_data_all <- read.table("log_norm_ct_trost_valdis.txt", header = T, sep = "\t")
transcript_samples_all <-
  read.table("transcript_samplelist_trost_valdis.txt", header=TRUE, sep="\t") %>%
  left_join(tolerance[,-3], by = "subspecies_id")

transcript_samples_all$row_id <- rownames(transcript_data_all)

# VALDIS lines
transcript_samples_valdis <-
  read.table("transcript_samplelist_valdis.txt", header = T, sep = "\t") %>%
  dplyr::select(line_id, crossing, name) %>% distinct
```

Metabolite data

911 samples, XXX metabolites

```
metabolite_data_all <- read.table("metabolite_data_trost_valdis.txt",
                                  header = T, sep = "\t", check.names = F)
dim(metabolite_data_all)
```

```
## [1] 2336   81
```

```
# 2336 samples, 81 metabolites

metabolite_samples_all <-
  read.table("metabolite_samplelist_trost_valdis.txt", header=TRUE, sep="\t") %>%
  left_join(tolerance, by = "cultivar")

# Import analytes overlap table (overlap regarding 17 measured TROST experiments, not the QC experiment
analytes <-
  read.table("analytes_trost_valdis.txt", sep = "\t", header = T) %>%
  filter(select_part == "yes") %>%
  arrange(analyte) %>% droplevels

length(analytes$name) # 81
```

```
## [1] 81
```

```
colnames(metabolite_data_all) <- analytes$MPIMP_ID
```

Remove batch effect in metabolite data

```
metabolite_data_norm <- apply(metabolite_data_all, 2,
                              RemoveFactors,
                              sam = metabolite_samples_all,
                              facs=c("genotype", "treatment", "SequenceID", "BatchID", "log10_AvgAnnota
                              keep=c("genotype", "treatment"))

sum(is.na(metabolite_data_norm)) / (ncol(metabolite_data_norm)*nrow(metabolite_data_norm))*100
```

```
## [1] 4.538728
```

```
# 4.5%
```

PCA

Transcripts

```
pca_transcript_all <- func_prep_pca(transcript_data_all,
                                     scale_method = "none",
                                     center_option = FALSE,
                                     pc_number = 5,
                                     pca_method = "rnipals")
```

```
## [1] "explained variances by PCs"
## [1] 0.22642814 0.14352148 0.08569786 0.06833892 0.05485333
```

```
# complete observations
compObs_transcript_all <-
  pca_transcript_all@completeObs %>%
  as.data.frame
```

Metabolites

```
pca_metabolite_all <- func_prep_pca(metabolite_data_norm,
                                    scale_method = "none",
                                    center_option = FALSE,
                                    pc_number = 10,
                                    pca_method = "rnipals")
```

```
## [1] "explained variances by PCs"
##  [1] 0.15627116 0.12251035 0.09522709 0.07235949 0.05679950 0.04030909
##  [7] 0.03366405 0.03064685 0.02537000 0.02533060
```

```
# complete observations
compObs_metabolite_all <-
  pca_metabolite_all@completeObs %>%
  as.data.frame
```

Define training and test data

Subsets: TROST and VALDIS

Transcripts

TROST data for training model

```
transcript_samples_training <-
  transcript_samples_all %>%
  filter(cultivation == "field") %>%
  filter(trost_valdis == "trost") %>%
  filter(!is.na(model_set)) %>%
  droplevels()

compObs_transcript_training <-
  compObs_transcript_all %>%
  rownames_to_column("row_id") %>%
  filter(row_id %in% transcript_samples_training$row_id) %>%
  column_to_rownames("row_id")

dim(compObs_transcript_training) # 202 samples
```

```
## [1] 202  42
```

VALDIS data for model prediction

```
transcript_samples_pred <-
  transcript_samples_all %>%
  filter(trost_valdis == "valdis") %>%
  droplevels()

compObs_transcript_pred <-
```

```
  compObs_transcript_all %>%
  rownames_to_column("row_id") %>%
  filter(row_id %in% transcript_samples_pred$row_id) %>%
  column_to_rownames("row_id")

dim(compObs_transcript_pred) # 803 samples
```

```
## [1] 803  42
```

Metabolites

TROST data for training model

```
metabolite_samples_training <-
  metabolite_samples_all %>%
  filter(cultivation == "field") %>%
  filter(trost_valdis == "trost") %>%
  filter(!is.na(model_set)) %>%
  droplevels()

compObs_metabolite_training <-
  compObs_metabolite_all %>%
  rownames_to_column("chromatogram") %>%
  filter(chromatogram %in% metabolite_samples_training$chromatogram) %>%
  column_to_rownames("chromatogram")

dim(compObs_metabolite_training) # 911 samples
```

```
## [1] 911  81
```

VALDIS transcript data for model prediction

```
metabolite_samples_pred <-
  metabolite_samples_all %>%
  filter(trost_valdis == "valdis") %>%
  droplevels()

compObs_metabolite_pred <-
  compObs_metabolite_all %>%
  rownames_to_column("chromatogram") %>%
  filter(chromatogram %in% metabolite_samples_pred$chromatogram) %>%
  column_to_rownames("chromatogram")

dim(compObs_metabolite_pred) # 806 samples
```

```
## [1] 806  81
```

LASSO Model

Transcripts

Moodel training

`mdrym_fve`: Median DRYM

```
set.seed(1)
lasso_fit_transcript <- glmnet(x = as.matrix(compObs_transcript_training),
                                y = transcript_samples_training$mdrym_fve)
# plot(lasso_fit_transcript)

# Cross-validation (10-fold)
set.seed(1111)
lasso_cv_transcript <- glmnet::cv.glmnet(x = as.matrix(compObs_transcript_training),
                                          y = transcript_samples_training$mdrym_fve)
lasso_cv_transcript$lambda.1se # 0.001209806
```

```
## [1] 0.001209806
```

```
# plot(lasso_cv_transcript, ylim=c(0, 0.005))

lasso_cv_transcript_coef_1se <- predict(lasso_cv_transcript,
                                         type = "coefficients",
                                         s = lasso_cv_transcript$lambda.1se)
table(as.matrix(lasso_cv_transcript_coef_1se) == 0) # 23 transcripts left
```

```
##
## FALSE  TRUE
##    24    19
```

Predict DRYM for VALDIS data

```
# use largest value of lambda such that CV-error is within 1 standard error of the minimum (ca. 50 meta
lasso_transcript_predicted_drym_valdis_1se <-
  predict(lasso_cv_transcript,
          new = as.matrix(compObs_transcript_pred),
          s = "lambda.1se")

lasso_transcript_predicted_drym_valdis <-
  predict(lasso_cv_transcript,
          new = as.matrix(compObs_transcript_pred),
          s = "lambda.min")
```

Median DRYM values

```
# join predicted DRYM values with line IDs and calculate median
lasso_transcript_predicted_drym_valdis_1se_median <-
  data.frame(drym = lasso_transcript_predicted_drym_valdis_1se[,1],
             line_id = transcript_samples_pred$subspecies_id) %>%
  group_by(line_id) %>%
```

```r
  summarize(median_drym = median(drym)) %>%
  rename(drym = median_drym) %>%
  left_join(transcript_samples_valdis, by = "line_id")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
lasso_transcript_predicted_drym_valdis_median <-
  data.frame(drym = lasso_transcript_predicted_drym_valdis[,1],
             line_id = transcript_samples_pred$subspecies_id) %>%
  group_by(line_id) %>%
  summarize(median_drym = median(drym)) %>%
  rename(drym = median_drym) %>%
  left_join(transcript_samples_valdis, by = "line_id")
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
# save median of predicted drym
write.table(lasso_transcript_predicted_drym_valdis_1se_median,
            "lasso_transcript_predicted_drym_valdis_1se_median.txt",
            sep = "\t", row.names = F)
```

Metabolites

Moodel training

`mdrym_fve`: Median DRYM

```r
set.seed(1)
lasso_fit_metabolite <- glmnet(x = as.matrix(compObs_metabolite_training),
                               y = metabolite_samples_training$mdrym_fve)
# plot(lasso_fit_metabolite)

# Cross-validation (10-fold)
set.seed(1111)
lasso_cv_metabolite <- glmnet::cv.glmnet(x = as.matrix(compObs_metabolite_training),
                                         y = metabolite_samples_training$mdrym_fve)
lasso_cv_metabolite$lambda.1se # 0.0006964521
```

```
## [1] 0.0006964521
```

```r
# plot(lasso_cv_metabolite, ylim=c(0, 0.005))

# define lambda for sparse model with 29 variables
lambda.sparse <- 0.002

lasso_cv_metabolite_coef_1se <- predict(lasso_cv_metabolite,
                                        type = "coefficients",
                                        s = lambda.sparse)
table(as.matrix(lasso_cv_metabolite_coef_1se) == 0)
```

```
##
## FALSE  TRUE
##    30    52
```

Predict DRYM for VALDIS data

```r
# use largest value of lambda such that CV-error is within 1 standard error of the minimum (ca. 50 meta
lasso_metabolite_predicted_drym_valdis_sparse <-
  predict(lasso_cv_metabolite,
          new = as.matrix(compObs_metabolite_pred),
          s = lambda.sparse)

lasso_metabolite_predicted_drym_valdis <-
  predict(lasso_cv_metabolite,
          new = as.matrix(compObs_metabolite_pred),
          s = "lambda.min")
```

Median DRYM values

```r
# join predicted DRYM values with line IDs and calculate median
lasso_metabolite_predicted_drym_valdis_sparse_median <-
  data.frame(drym = lasso_metabolite_predicted_drym_valdis_sparse[,1],
             line_id = metabolite_samples_pred$cultivar) %>%
  group_by(line_id) %>%
  summarize(median_drym = median(drym)) %>%
  rename(drym = median_drym) %>%
  mutate(name = str_replace(line_id, "_", "")) %>%
  mutate(name = str_replace(name, "AxR", "AR")) %>%
  mutate(name = str_replace(name, "ExA", "EA")) %>%
  mutate(name = str_replace(name, "ALBATROS", "Albatros")) %>%
  mutate(name = str_replace(name, "DESIREE", "Desiree")) %>%
  mutate(name = str_replace(name, "EURORESA", "Euroresa")) %>%
  mutate(name = str_replace(name, "RAMSES", "Ramses"))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
lasso_metabolite_predicted_drym_valdis_median <-
  data.frame(drym = lasso_metabolite_predicted_drym_valdis[,1],
             line_id = metabolite_samples_pred$cultivar) %>%
  group_by(line_id) %>%
  summarize(median_drym = median(drym)) %>%
  rename(drym = median_drym) %>%
  mutate(name = str_replace(line_id, "_", "")) %>%
  mutate(name = str_replace(name, "AxR", "AR")) %>%
  mutate(name = str_replace(name, "ExA", "EA")) %>%
  mutate(name = str_replace(name, "ALBATROS", "Albatros")) %>%
  mutate(name = str_replace(name, "DESIREE", "Desiree")) %>%
  mutate(name = str_replace(name, "EURORESA", "Euroresa")) %>%
  mutate(name = str_replace(name, "RAMSES", "Ramses"))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
# save median of predicted drym
write.table(lasso_metabolite_predicted_drym_valdis_sparse_median,
            "lasso_metabolite_predicted_drym_valdis_sparse_median.txt",
            sep = "\t", row.names = F)
```

Combine predicted DRYM values

```
lasso_predicted_drym_sparse <-
  lasso_metabolite_predicted_drym_valdis_sparse_median %>%
  dplyr::select(name, drym) %>%
  rename(drym_metabolite = drym) %>%
  left_join(lasso_transcript_predicted_drym_valdis_1se_median, by = "name") %>%
  rename(drym_transcript = drym) %>%
  mutate(drym_avg = (drym_metabolite + drym_transcript)/2) %>%
  arrange(drym_avg)

lasso_predicted_drym <-
  lasso_metabolite_predicted_drym_valdis_median %>%
  dplyr::select(name, drym) %>%
  rename(drym_metabolite = drym) %>%
  left_join(lasso_transcript_predicted_drym_valdis_median, by = "name") %>%
  rename(drym_transcript = drym) %>%
  mutate(drym_avg = (drym_metabolite + drym_transcript)/2) %>%
  arrange(drym_avg)

write.table(lasso_predicted_drym,
            "lasso_predicted_drym_valdis_sparse_median.txt",
            sep = "\t", row.names = F)
```

Subpopulations

```
lines_MPt <- c("AR1", "AR23", "AR56", "AR67", "AR106", "AR121", "AR157", "AR163", "AR183",
               "AR185", "AR196", "AR197", "AR200", "AR241", "AR245", "AR254", "AR269", "AR282",
               "AR285", "AR293", "EA28", "EA74", "EA87")
lines_MPs <- c("AR55", "EA2", "EA8", "EA19", "EA22", "EA54", "EA55", "EA71", "EA92",
               "EA111", "EA112", "EA131", "EA154", "EA165", "EA172", "EA173", "EA174",
               "EA252", "EA269", "EA273", "EA279", "EA280")

lasso_predicted_drym_sparse$select <- "not"
lasso_predicted_drym_sparse$select[which(lasso_predicted_drym_sparse$name %in% lines_MPt)] <- "MPt"
lasso_predicted_drym_sparse$select[which(lasso_predicted_drym_sparse$name %in% lines_MPs)] <- "MPs"
table(lasso_predicted_drym_sparse$select)
```

```
##
## MPs MPt not
##  22  23 154
```

```
lasso_predicted_drym$select <- "not"
lasso_predicted_drym$select[which(lasso_predicted_drym$name %in% lines_MPt)] <- "MPt"
lasso_predicted_drym$select[which(lasso_predicted_drym$name %in% lines_MPs)] <- "MPs"
table(lasso_predicted_drym$select)
```

```
##
## MPs MPt not
##  22  23 154
```

```
# ggplot(lasso_predicted_drym_sparse, aes(x = select, y = drym_avg)) + geom_boxplot()
```

Plots

Transcripts

Boxplot of predicted DRYM vs line

```r
lasso_transcript_predicted_drym_valdis_df <-
  data.frame(drym = lasso_transcript_predicted_drym_valdis[,1],
             line_id = transcript_samples_pred$subspecies_id) %>%
  left_join(transcript_samples_valdis, by = "line_id")

# sort predicted DRYM values for plot
lasso_transcript_predicted_drym_valdis_sorted <-
  with(lasso_transcript_predicted_drym_valdis_df, reorder(name, drym, median, na.rm=T))

lasso_color <- rep("#00756D", 199) # for ExA
lasso_color [which( grepl("^AR", levels(lasso_transcript_predicted_drym_valdis_sorted) ))] <- "#BF5300"
lasso_color [which( levels (lasso_transcript_predicted_drym_valdis_sorted) == "Albatros" )] <- "grey"
lasso_color [which( levels (lasso_transcript_predicted_drym_valdis_sorted) == "Euroresa" )] <- "#5778B9
lasso_color [which( levels (lasso_transcript_predicted_drym_valdis_sorted) == "Ramses" )] <- "#F7B944"
lasso_color [which( levels (lasso_transcript_predicted_drym_valdis_sorted) == "Desiree" )] <- "white"
# lasso_color <- subset(lasso_color, levels (lasso_transcript_predicted_drym_valdis_sorted) != "DESIREE
lasso_color_fac <- factor(lasso_color)

png("boxplot_lasso_transcript_predicted_drym_valdis_full.png", width=3000, height=1500, res=300)
par(mar=c(4.5,5,2,0.5))
boxplot(drym ~ lasso_transcript_predicted_drym_valdis_sorted, data = lasso_transcript_predicted_drym_val
        ylab="DRYM", cex.lab=1.5, cex.axis=1.2, las=2, col=lasso_color, main = "transcript model", names
legend("bottomright", fill=levels(lasso_color_fac), legend=c("ExA", "E", "AxR", "R", "A"), horiz=T)
dev.off()
```

```
## pdf
##   2
```

Boxplot of predicted DRYM vs line (sparse)

```r
lasso_transcript_predicted_drym_valdis_1se_df <-
  data.frame(drym = lasso_transcript_predicted_drym_valdis_1se[,1],
             line_id = transcript_samples_pred$subspecies_id) %>%
  left_join(transcript_samples_valdis, by = "line_id")

# sort predicted DRYM values for plot
lasso_transcript_predicted_drym_valdis_1se_sorted <-
  with(lasso_transcript_predicted_drym_valdis_1se_df, reorder(name, drym, median, na.rm=T))

lasso_color <- rep("#00756D", 199) # for ExA
lasso_color [which( grepl("^AR", levels(lasso_transcript_predicted_drym_valdis_1se_sorted) ))] <- "#BF5
lasso_color [which( levels (lasso_transcript_predicted_drym_valdis_1se_sorted) == "Albatros" )] <- "gre
lasso_color [which( levels (lasso_transcript_predicted_drym_valdis_1se_sorted) == "Euroresa" )] <- "#57
lasso_color [which( levels (lasso_transcript_predicted_drym_valdis_1se_sorted) == "Ramses" )] <- "#F7B9
lasso_color [which( levels (lasso_transcript_predicted_drym_valdis_1se_sorted) == "Desiree" )] <- "whit
# lasso_color <- subset(lasso_color, levels (lasso_transcript_predicted_drym_valdis_1se_sorted) != "DES
lasso_color_fac <- factor(lasso_color)

png("boxplot_lasso_transcript_predicted_drym_valdis_1se.png", width=3000, height=1500, res=300)
par(mar=c(4.5,5,2,0.5))
boxplot(drym ~ lasso_transcript_predicted_drym_valdis_1se_sorted, data = lasso_transcript_predicted_drym
        ylab="DRYM", cex.lab=1.5, cex.axis=1.2, las=2, col=lasso_color, main = "transcript model", names
legend("bottomright", fill=levels(lasso_color_fac), legend=c("ExA", "E", "AxR", "R", "A"), horiz=T)
dev.off()
```

```
## pdf
##   2
```

Metabolites

Boxplot of predicted DRYM vs line

```
lasso_metabolite_predicted_drym_valdis_df <-
  data.frame(drym = lasso_metabolite_predicted_drym_valdis[,1],
             line_id = metabolite_samples_pred$cultivar)

# sort predicted DRYM values for plot
lasso_metabolite_predicted_drym_valdis_sorted <-
  with(lasso_metabolite_predicted_drym_valdis_df, reorder(line_id, drym, median, na.rm=T))

lasso_color <- rep("#00756D", 199) # for ExA
lasso_color [which( grepl("^AxR", levels(lasso_metabolite_predicted_drym_valdis_sorted) ))] <- "#BF5300
lasso_color [which( levels (lasso_metabolite_predicted_drym_valdis_sorted) == "ALBATROS" )] <- "grey"
lasso_color [which( levels (lasso_metabolite_predicted_drym_valdis_sorted) == "EURORESA" )] <- "#5778B9
lasso_color [which( levels (lasso_metabolite_predicted_drym_valdis_sorted) == "RAMSES" )] <- "#F7B944"
lasso_color [which( levels (lasso_metabolite_predicted_drym_valdis_sorted) == "DESIREE" )] <- "white"
# lasso_color <- subset(lasso_color, levels (lasso_metabolite_predicted_drym_valdis_sorted) != "DESIREE
lasso_color_fac <- factor(lasso_color)

png("boxplot_lasso_metabolite_predicted_drym_valdis_full.png", width=3000, height=1500, res=300)
par(mar=c(4.5,5,2,0.5))
boxplot(drym ~ lasso_metabolite_predicted_drym_valdis_sorted, data = lasso_metabolite_predicted_drym_val
        ylab="DRYM", cex.lab=1.5, cex.axis=1.2, las=2, col=lasso_color, main = "metabolite model", names
legend("bottomright", fill=levels(lasso_color_fac), legend=c("ExA", "E", "AxR", "R", "A"), horiz=T)
dev.off()
```

```
## pdf
##   2
```

Boxplot of predicted DRYM vs line (sparse)

```
lasso_metabolite_predicted_drym_valdis_sparse_df <-
  data.frame(drym = lasso_metabolite_predicted_drym_valdis_sparse[,1],
             line_id = metabolite_samples_pred$cultivar)

# sort predicted DRYM values for plot
lasso_metabolite_predicted_drym_valdis_sparse_sorted <-
  with(lasso_metabolite_predicted_drym_valdis_sparse_df, reorder(line_id, drym, median, na.rm=T))
```

```r
lasso_color <- rep("#00756D", 199) # for ExA
lasso_color [which( grepl("^AxR", levels(lasso_metabolite_predicted_drym_valdis_sparse_sorted) ))] <- "#
lasso_color [which( levels (lasso_metabolite_predicted_drym_valdis_sparse_sorted) == "ALBATROS" )] <- "g
lasso_color [which( levels (lasso_metabolite_predicted_drym_valdis_sparse_sorted) == "EURORESA" )] <- "#
lasso_color [which( levels (lasso_metabolite_predicted_drym_valdis_sparse_sorted) == "RAMSES" )] <- "#F
lasso_color [which( levels (lasso_metabolite_predicted_drym_valdis_sparse_sorted) == "DESIREE" )] <- "wh
# lasso_color <- subset(lasso_color, levels (lasso_metabolite_predicted_drym_valdis_sparse_sorted) != "
lasso_color_fac <- factor(lasso_color)

png("boxplot_lasso_metabolite_predicted_drym_valdis_sparse.png", width=3000, height=1500, res=300)
par(mar=c(4.5,5,2,0.5))
boxplot(drym ~ lasso_metabolite_predicted_drym_valdis_sparse_sorted, data = lasso_metabolite_predicted_d
        ylab="DRYM", cex.lab=1.5, cex.axis=1.2, las=2, col=lasso_color, main = "metabolite model", names
legend("bottomright", fill=levels(lasso_color_fac), legend=c("ExA", "E", "AxR", "R", "A"), horiz=T)
dev.off()
```

```
## pdf
##   2
```

Save workspace

```r
save.image("lasso_model.RData")
```

Session Info

```r
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Linux Mint 19.3
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
## LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8              LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8               LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=de_DE.iso885915@euro  LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=de_DE.iso885915@euro     LC_NAME=C
##  [9] LC_ADDRESS=C                      LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_DE.iso885915@euro LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] pcaMethods_1.78.0   Biobase_2.46.0      BiocGenerics_0.32.0
##  [4] forcats_0.5.0       stringr_1.4.0       dplyr_1.0.0
##  [7] purrr_0.3.4         readr_1.3.1         tidyr_1.1.0
## [10] tibble_3.0.1        tidyverse_1.3.0     glmnet_4.0
## [13] Matrix_1.2-18       caret_6.0-86        ggplot2_3.3.1
## [16] lattice_0.20-40     knitr_1.28
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.1           jsonlite_1.6.1       splines_3.6.3
##  [4] foreach_1.5.0        prodlim_2019.11.13   modelr_0.1.8
##  [7] assertthat_0.2.1     stats4_3.6.3         blob_1.2.1
## [10] cellranger_1.1.0     yaml_2.2.1           ipred_0.9-9
## [13] pillar_1.4.4         backports_1.1.7      glue_1.4.1
## [16] pROC_1.16.2          digest_0.6.25        rvest_0.3.5
## [19] colorspace_1.4-1     recipes_0.1.12       htmltools_0.4.0
## [22] plyr_1.8.6           timeDate_3043.102    pkgconfig_2.0.3
## [25] broom_0.5.6          haven_2.3.1          scales_1.1.1
## [28] gower_0.2.1          lava_1.6.7           generics_0.0.2
## [31] ellipsis_0.3.1       withr_2.2.0          nnet_7.3-13
## [34] cli_2.0.2            survival_3.1-11      magrittr_1.5
## [37] crayon_1.3.4         readxl_1.3.1         evaluate_0.14
## [40] fansi_0.4.1          fs_1.4.1             nlme_3.1-144
## [43] MASS_7.3-51.5        xml2_1.3.2           class_7.3-15
## [46] tools_3.6.3          data.table_1.12.8    hms_0.5.3
## [49] lifecycle_0.2.0      munsell_0.5.0        reprex_0.3.0
## [52] compiler_3.6.3       rlang_0.4.6          grid_3.6.3
## [55] iterators_1.0.12     rstudioapi_0.11      rmarkdown_2.2
## [58] gtable_0.3.0         ModelMetrics_1.2.2.2 codetools_0.2-16
## [61] DBI_1.1.0            reshape2_1.4.4       R6_2.4.1
## [64] lubridate_1.7.8      shape_1.4.4          stringi_1.4.6
## [67] Rcpp_1.0.4.6         vctrs_0.3.1          rpart_4.1-15
## [70] dbplyr_1.4.4         tidyselect_1.1.0     xfun_0.14
```