6. PCA FACS -lab

Fay

2022-11-03

Always change the knitting directory to the working directory! # Load libraries

```
library(tidyverse)
library(dplyr)
library(stringr)
library(FactoMineR)
library(reshape2)
library(corrplot)
library(factoextra)
library(lmtest)
library(ggpubr)
library(janitor)
library(pheatmap)
library(visdat)
```

Load data

```
hm <- read.csv("output_data/imputed_mice.csv")
```

vectors for selecting

```
Gene_lab   <- c("IFNy", "CXCR3", "IL.6", "IL.13", "IL.10",
                "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                "TICAM1", "TNF") # "IL.12", "IRG6")

#add a suffix to represent changes in data file
Gene_lab_imp <- paste(Gene_lab, "imp", sep = "_")

Genes_wild   <- c("IFNy", "CXCR3", "IL.6", "IL.13", "IL.10",
                  "IL1RN","CASP1", "CXCL9", "IDO1", "IRGM1", "MPO",
                  "MUC2", "MUC5AC", "MYD88", "NCR1", "PRF1", "RETNLB", "SOCS1",
                  "TICAM1", "TNF", "IL.12", "IRG6")

Genes_wild_imp <- paste(Genes_wild, "imp", sep = "_")

Facs_lab <- c("CD4", "Treg", "Div_Treg", "Treg17", "Th1",
                    "Div_Th1", "Th17", "Div_Th17", "CD8", "Act_CD8",
                    "Div_Act_CD8", "IFNy_CD4", "IFNy_CD8") #"Treg_prop", removed due to many missing va
                    #"IL17A_CD4"
```

```
Facs_wild <- c( "Treg", "CD4", "Treg17", "Th1", "Th17", "CD8",
                     "Act_CD8", "IFNy_CD4", "IL17A_CD4", "IFNy_CD8")
```

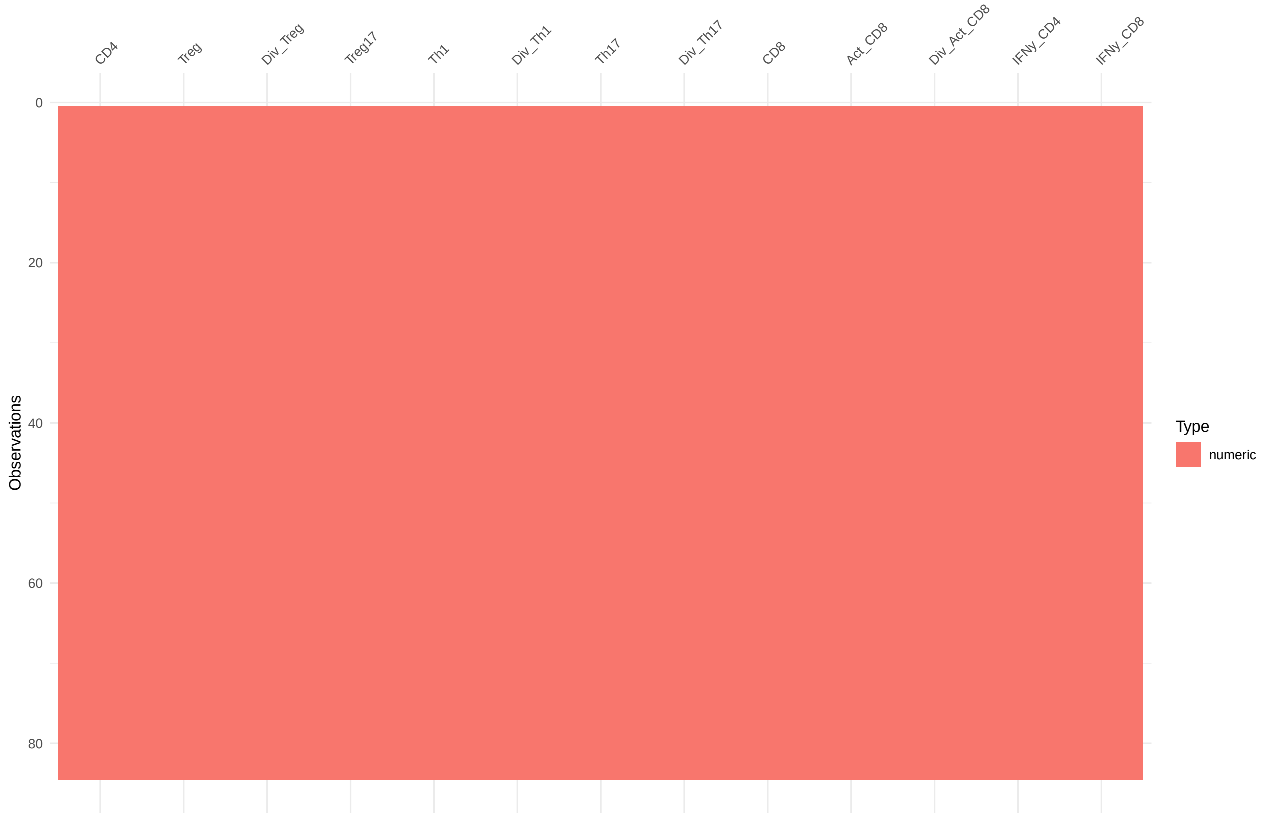
FACS

Lab

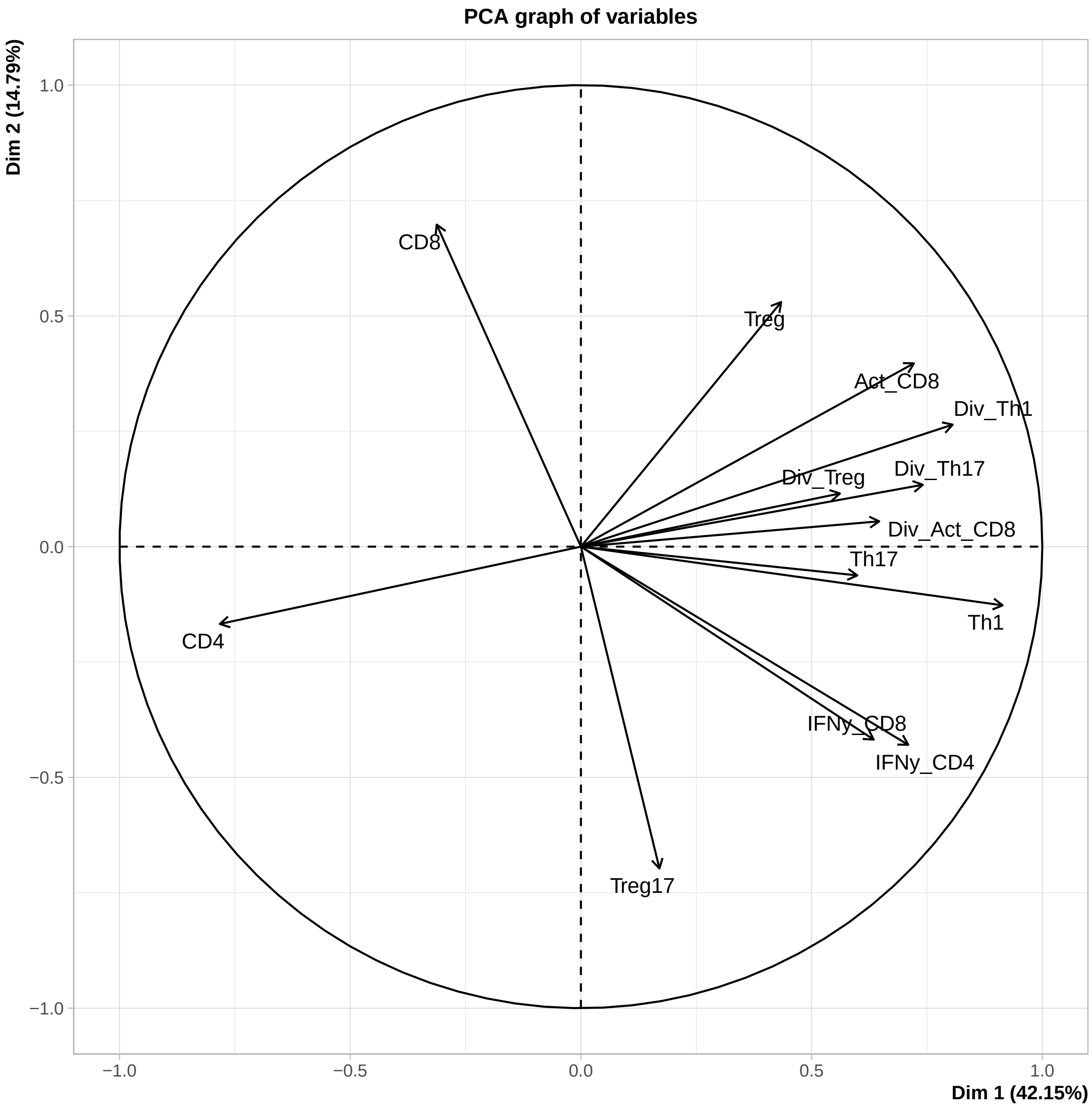
PCA on the lab genes -*imputed*

```
#select the genes and lab muce
lab <- hm %>%
  dplyr::filter(origin == "Lab", Position == "mLN") #selecting for mln to avoid

# duplicates

lab <- unique(lab)

facs_mouse <- lab %>%
  dplyr::select(c(Mouse_ID, all_of(Facs_lab)))


facs <- facs_mouse[, -1]

#remove rows with only nas
facs <- facs[,colSums(is.na(facs))<nrow(facs)]

#remove colums with only nas
facs <- facs[rowSums(is.na(facs)) != ncol(facs), ]

vis_dat(facs)
```

```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## Please use `gather()` instead.
```



```
#select same rows in the first table
facs_mouse <- facs_mouse[row.names(facs), ]


# we can now run a normal pca on the complete data set
res.pca <- PCA(facs)
```



Dimensions of the pca

Caution: When imputing data, the percentages of inertia associated with the first dimensions will be overestimated.

Another problem: the imputed data are, when the pca is performed considered like real observations. But they are estimations!!

Visualizing uncertainty due to issing data:

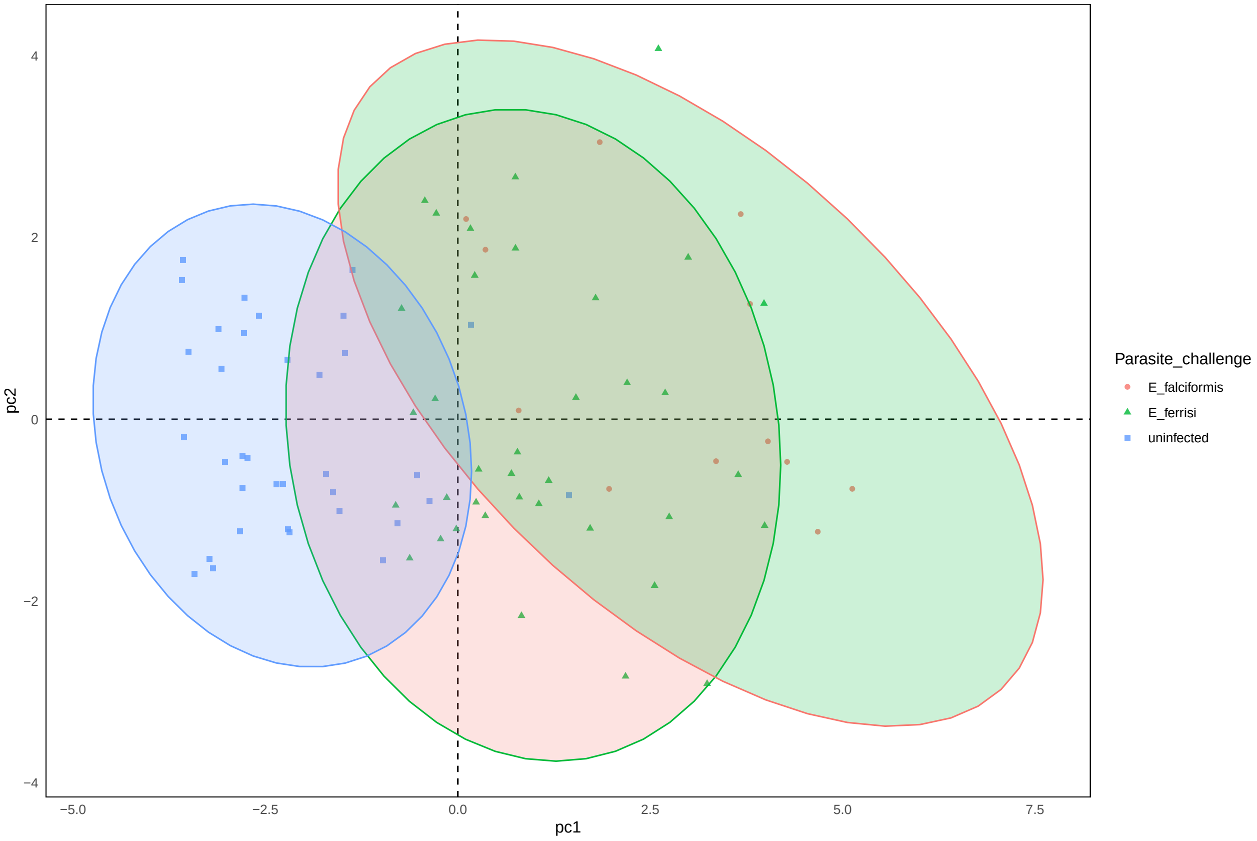
–> mulrimple imputation: generate several plausible values for each missing data point

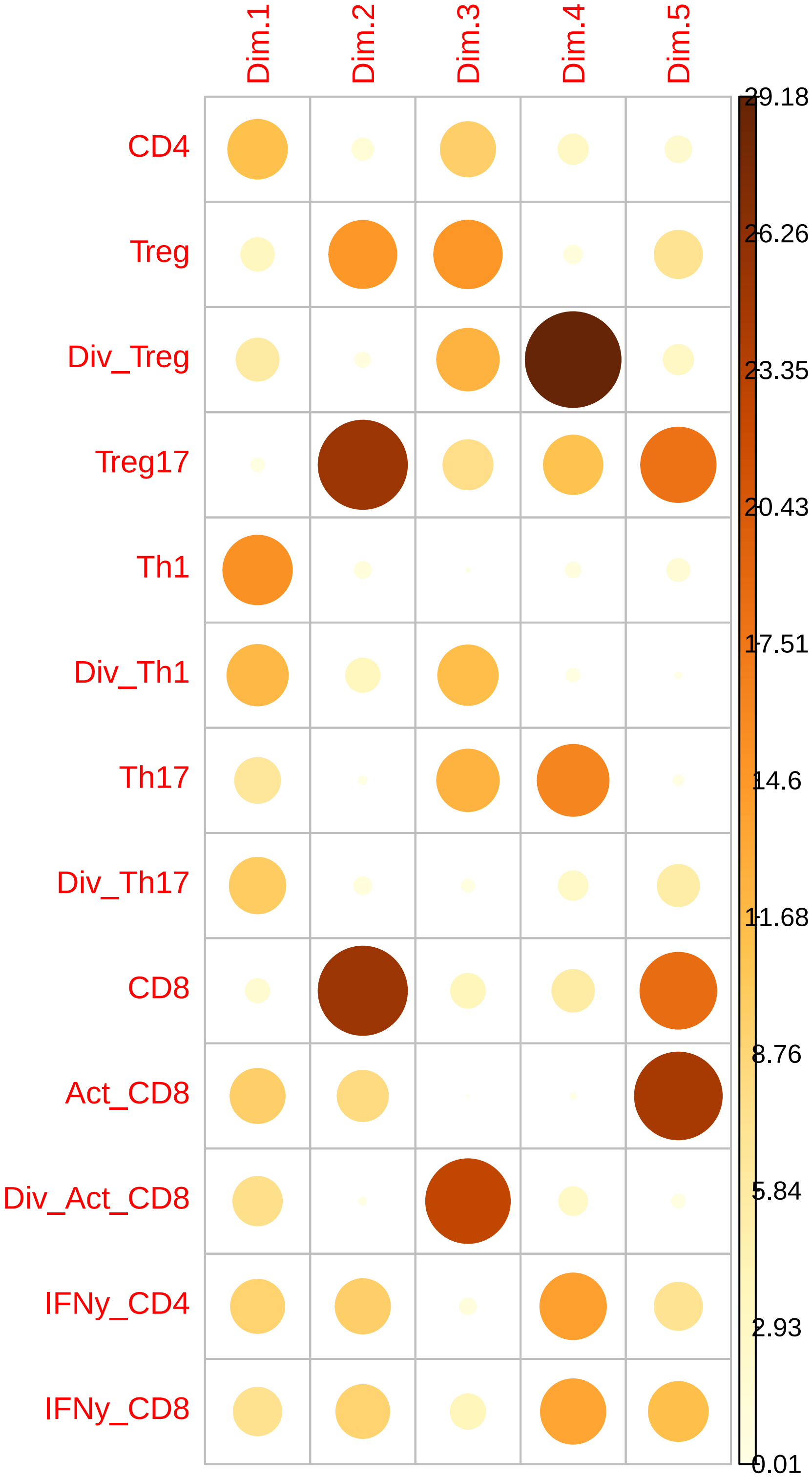
We here visualize the variability, that is uncertainty on the plane defined by two pca axes.

Biplot of the imputed facs pca

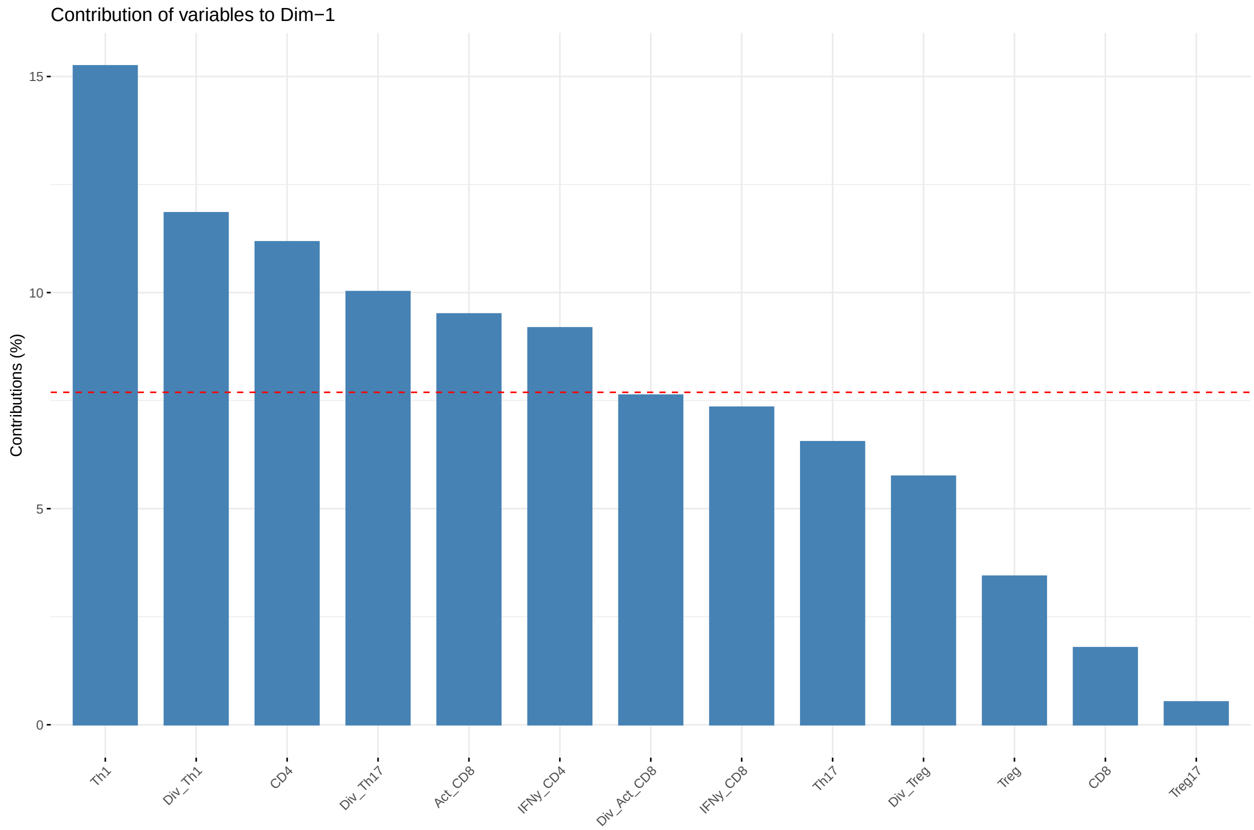
```
#Now we can make our initial plot of the PCA.
lab %>%
  ggplot(aes(x = pc1, y = pc2,
             color = Parasite_challenge,
             shape = Parasite_challenge)) +
  geom_hline(yintercept = 0, lty = 2) +
  geom_vline(xintercept = 0, lty = 2) +
  geom_point(alpha = 0.8) +
  stat_ellipse(geom="polygon",
               aes(fill = challenge_infection),
               alpha = 0.2, show.legend = FALSE,
               level = 0.95) +
  theme_minimal() +
  theme(panel.grid = element_blank(),
        panel.border = element_rect(fill= "transparent"))
```

```
## Warning: Removed 42 rows containing non-finite values (stat_ellipse).

## Warning: Removed 42 rows containing missing values (geom_point).
```

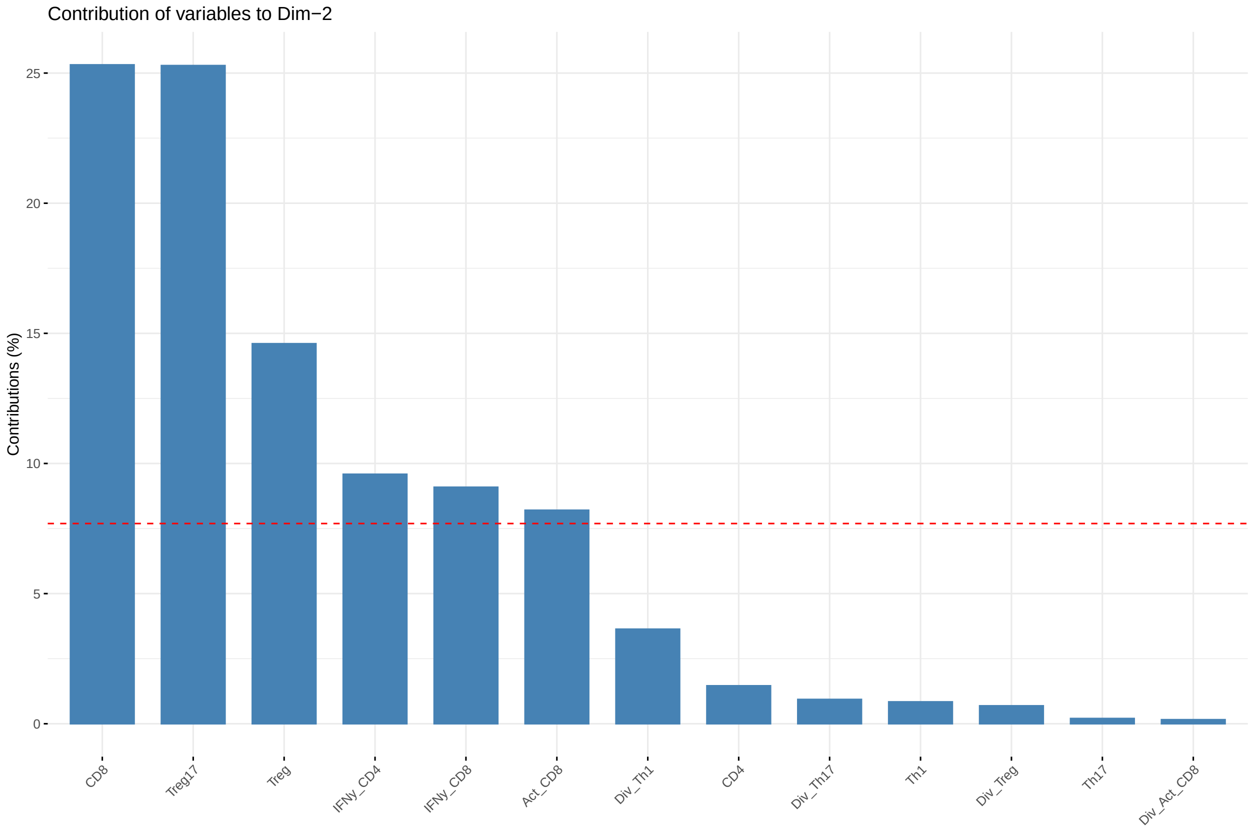


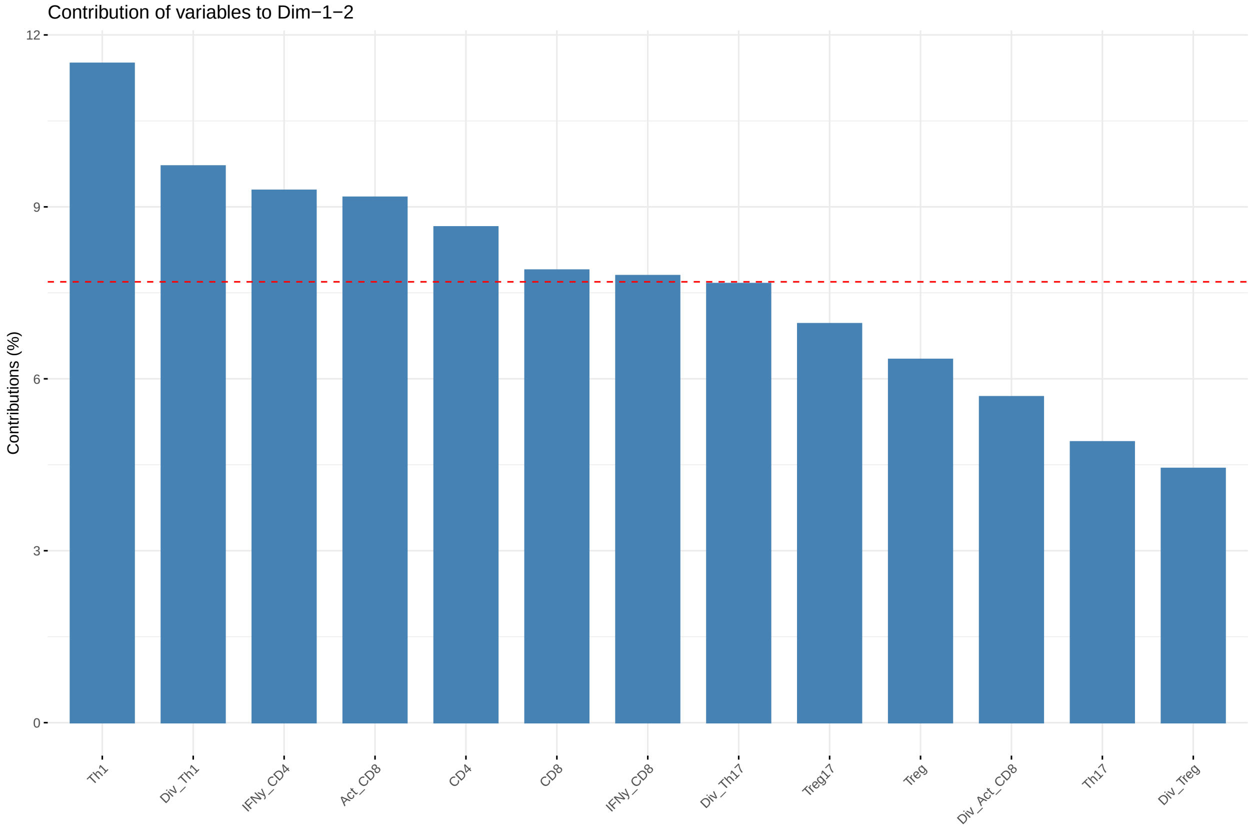


The function fviz_contrib() [factoextra package] can be used to draw a bar plot of variable contributions. If your data contains many variables, you can decide to show only the top contributing variables. The R code below shows the top 10 variables contributing to the principal components:



```
# Contributions of variables to PC2
fviz_contrib(res.pca, choice = "var", axes = 2, top = 18)
```

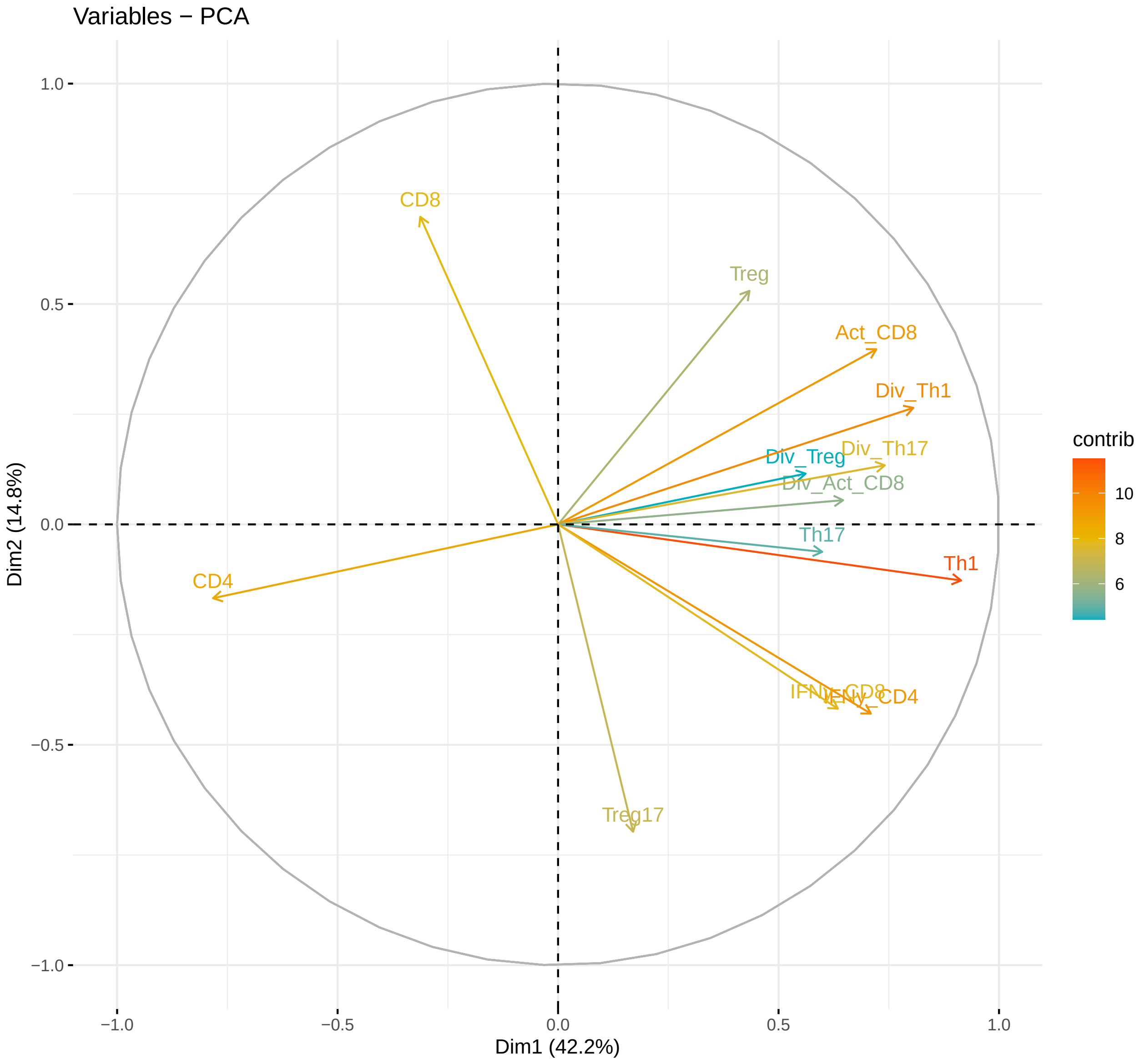




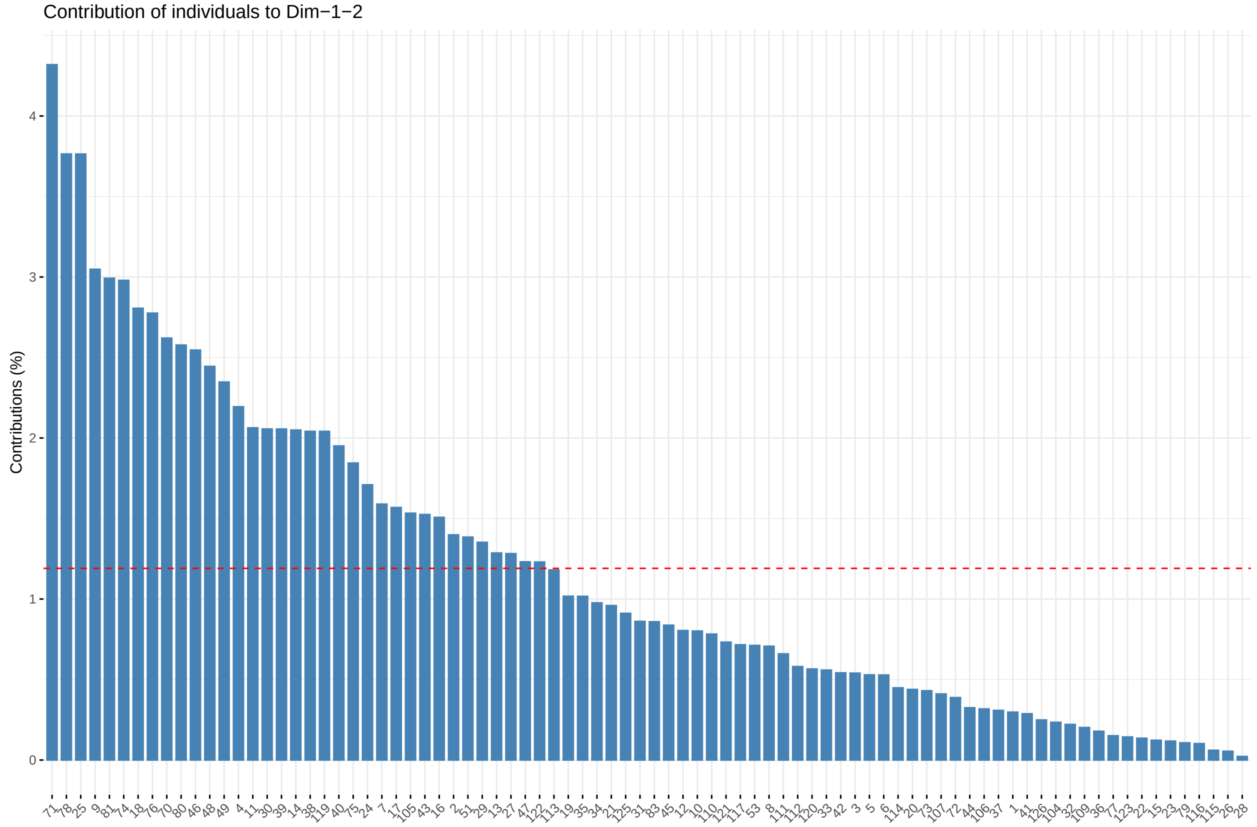
The red dashed line on the graph above indicates the expected average contribution. If the contribution of the variables were uniform, the expected value would be 1/length(variables) = 1/10 = 10%. For a given component, a variable with a contribution larger than this cutoff could be considered as important in contributing to the component.

Note that, the total contribution of a given variable, on explaining the variations retained by two principal components, say PC1 and PC2, is calculated as contrib = [(C1 * Eig1) + (C2 * Eig2)]/(Eig1 + Eig2), where

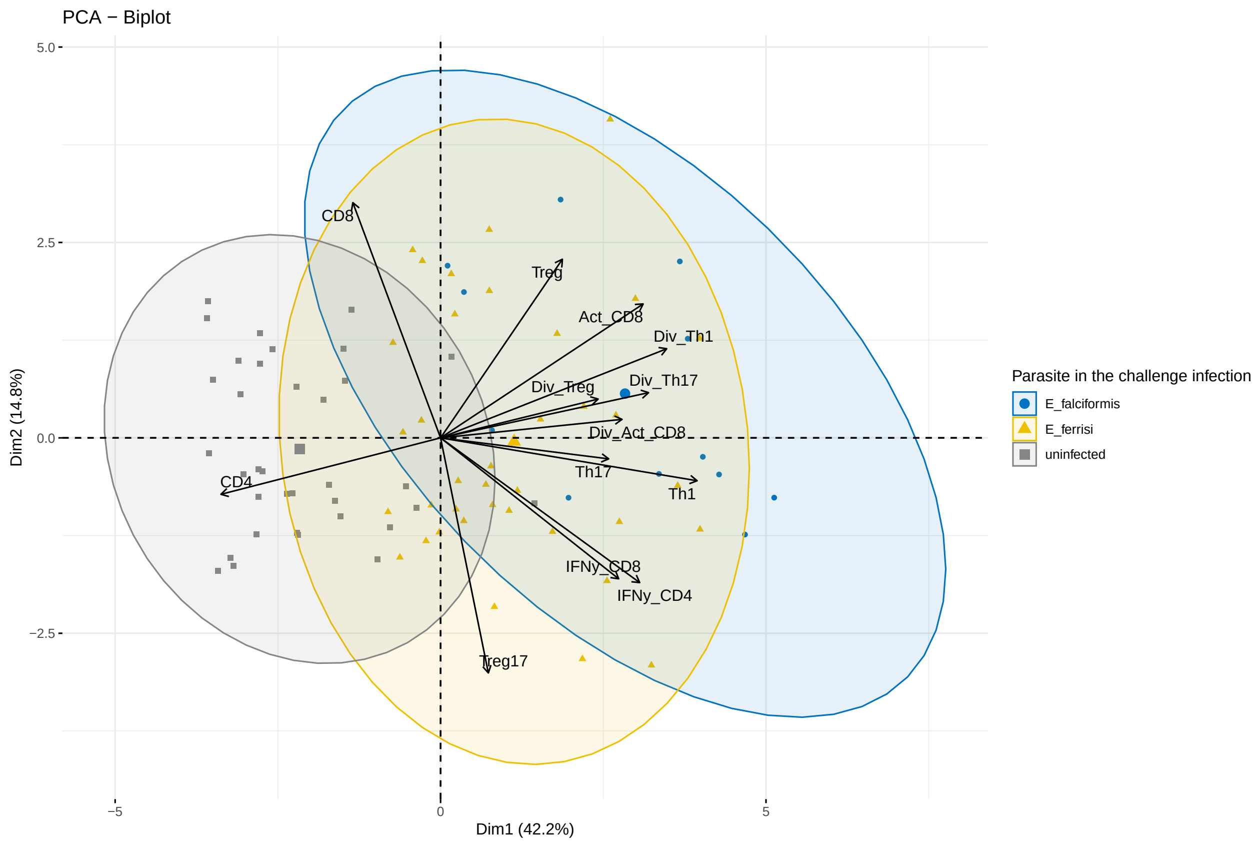
C1 and C2 are the contributions of the variable on PC1 and PC2, respectively Eig1 and Eig2 are the eigenvalues of PC1 and PC2, respectively. Recall that eigenvalues measure the amount of variation retained by each PC. In this case, the expected average contribution (cutoff) is calculated as follow: As mentioned above, if the contributions of the 10 variables were uniform, the expected average contribution on a given PC would be 1/10 = 10%. The expected average contribution of a variable for PC1 and PC2 is : [(10* Eig1) + (10 * Eig2)]/(Eig1 + Eig2)



To visualize the contribution of individuals to the first two principal components:



PCA + Biplot combination



In the following example, we want to color both individuals and variables by groups. The trick is to use pointshape = 21 for individual points. This particular point shape can be filled by a color using the argument fill.ind. The border line color of individual points is set to "black" using col.ind. To color variable by groups, the argument col.var will be used.

Linear models:

```
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + Parasite_challenge, data = lab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.4741  -3.0434   0.0069   3.6193  10.0630
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   80.9688     1.9035  42.537  < 2e-16 ***
## pc1                            0.9581     0.4134   2.318    0.023 *
## pc2                            0.2961     0.4075   0.727    0.470
## Parasite_challengeE_ferrisi   10.9489     1.8448   5.935 7.43e-08 ***
## Parasite_challengeuninfected  17.0908     2.7188   6.286 1.67e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.019 on 79 degrees of freedom
## Multiple R-squared:  0.4205, Adjusted R-squared:  0.3912
## F-statistic: 14.33 on 4 and 79 DF,  p-value: 7.686e-09
```

```
## [1] 516.2597
```

```
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + Parasite_challenge + hybrid_status,
##     data = lab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0276  -3.3387   0.5902   3.7328   9.1033
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       79.8025     2.1074  37.867  < 2e-16 ***
## pc1                                0.8080     0.4673   1.729   0.0879 .
## pc2                               -0.6942     0.7558  -0.918   0.3614
## Parasite_challengeE_ferrisi       10.3598     1.9809   5.230 1.52e-06 ***
## Parasite_challengeuninfected      15.7622     2.7361   5.761 1.80e-07 ***
## hybrid_statusF0 M. m. musculus     3.0943     2.7907   1.109   0.2711
## hybrid_statusF1 hybrid             4.0526     1.7731   2.286   0.0251 *
## hybrid_statusF1 M. m. domesticus  -1.0159     2.0166  -0.504   0.6159
## hybrid_statusF1 M. m. musculus     5.2409     3.0924   1.695   0.0943 .
## hybrid_statusother                 2.3646     2.2221   1.064   0.2907
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.946 on 74 degrees of freedom
## Multiple R-squared:  0.473,  Adjusted R-squared:  0.4089 
## F-statistic: 7.378 on 9 and 74 DF,  p-value: 1.343e-07
```

```
## [1] 518.2932
```

Try instead: LLR test (likelihood ration) (LM4 package)?

https://www.rdocumentation.org/packages/lmtest/versions/0.9-38/topics/lrtest

In this way you compare each model, with the different variables usesd to predict.

Another way is to compare the AIC. (function : step)

```
weight_lm3 <- lm(max_WL ~ pc1 + pc2 + hybrid_status, data = lab)
weight_no_pc1 <- lm(max_WL ~ pc2 + hybrid_status, data = lab)
weight_no_pc2 <- lm(max_WL ~ pc1  + hybrid_status, data = lab)
weight_no_hybrid <- lm(max_WL ~ pc1 + pc2, data = lab)
lrtest(weight_lm3, weight_no_pc1)
```

```
## Likelihood ratio test
## 
## Model 1: max_WL ~ pc1 + pc2 + hybrid_status
## Model 2: max_WL ~ pc2 + hybrid_status
##   #Df  LogLik Df  Chisq Pr(>Chisq)  
## 1   9 -264.86                       
## 2   8 -267.95 -1 6.1858    0.01288 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lrtest(weight_lm3, weight_no_pc2)
```

```
## Likelihood ratio test
## 
## Model 1: max_WL ~ pc1 + pc2 + hybrid_status
## Model 2: max_WL ~ pc1 + hybrid_status
##   #Df  LogLik Df Chisq Pr(>Chisq)  
## 1   9 -264.86                      
## 2   8 -267.43 -1 5.138    0.02341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
lrtest(weight_lm3, weight_no_hybrid)
```

```
## Likelihood ratio test
## 
## Model 1: max_WL ~ pc1 + pc2 + hybrid_status
## Model 2: max_WL ~ pc1 + pc2
##   #Df  LogLik Df  Chisq Pr(>Chisq)  
## 1   9 -264.86                       
## 2   4 -270.36 -5 10.995    0.05147 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## 
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + hybrid_status, data = lab)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -14.7941  -3.9487   0.5969   4.0292  13.2617
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       89.8897     1.5060  59.687  < 2e-16 ***
## pc1                               -0.7531     0.3125  -2.410  0.01837 *
## pc2                               -1.9165     0.8753  -2.189  0.03163 *
## hybrid_statusF0 M. m. musculus     5.5814     3.3176   1.682  0.09660 .
## hybrid_statusF1 hybrid             6.0398     2.0974   2.880  0.00517 **
## hybrid_statusF1 M. m. domesticus  -0.7613     2.4155  -0.315  0.75349
## hybrid_statusF1 M. m. musculus     8.3437     3.6664   2.276  0.02568 *
## hybrid_statusother                 2.2420     2.3853   0.940  0.35023
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.955 on 76 degrees of freedom
## Multiple R-squared:  0.2153, Adjusted R-squared:  0.1431
## F-statistic: 2.979 on 7 and 76 DF,  p-value: 0.008133
```

```
## [1] 547.7233
```

```
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + infection_history, data = lab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.6200  -2.7393  -0.1766   3.3119   8.7910
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              84.3265     2.6943  31.298  < 2e-16
## pc1                                       1.0756     0.4051   2.655 0.009731
## pc2                                       0.3095     0.4141   0.747 0.457224
## infection_historyfalciformis_ferrisi      7.3079     2.6336   2.775 0.007009
## infection_historyfalciformis_uninfected  15.1902     3.4750   4.371 4.03e-05
## infection_historyferrisi_falciformis     -4.4071     3.0643  -1.438 0.154652
## infection_historyferrisi_ferrisi          9.3843     2.8404   3.304 0.001479
## infection_historyferrisi_uninfected      13.3682     3.4287   3.899 0.000213
## infection_historyuninfected              13.7948     3.9599   3.484 0.000840
## infection_historyuninfected_falciformis  -8.9692     4.0447  -2.218 0.029701
## infection_historyuninfected_ferrisi      -1.1569     3.3462  -0.346 0.730544
##
## (Intercept)                             ***
## pc1                                     **
## pc2
## infection_historyfalciformis_ferrisi    **
## infection_historyfalciformis_uninfected ***
## infection_historyferrisi_falciformis
## infection_historyferrisi_ferrisi        **
## infection_historyferrisi_uninfected     ***
## infection_historyuninfected             ***
## infection_historyuninfected_falciformis *
## infection_historyuninfected_ferrisi
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.514 on 73 degrees of freedom
## Multiple R-squared:  0.5669, Adjusted R-squared:  0.5075
## F-statistic: 9.554 on 10 and 73 DF,  p-value: 5.886e-10
```

```
## [1] 503.8078
```

```
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2, data = lab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.865  -2.911   1.123   4.404   9.741
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  92.9127     0.6719 138.277  < 2e-16 ***
## pc1          -0.8623     0.2870  -3.004  0.00354 **
## pc2          -0.3550     0.4845  -0.733  0.46583
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.158 on 81 degrees of freedom
## Multiple R-squared:  0.1056, Adjusted R-squared:  0.08351
## F-statistic: 4.781 on 2 and 81 DF,  p-value: 0.01089
```

```
##                    df      AIC
## weight_lm           6 516.2597
## weight_lm_exp_only  4 548.7185
```

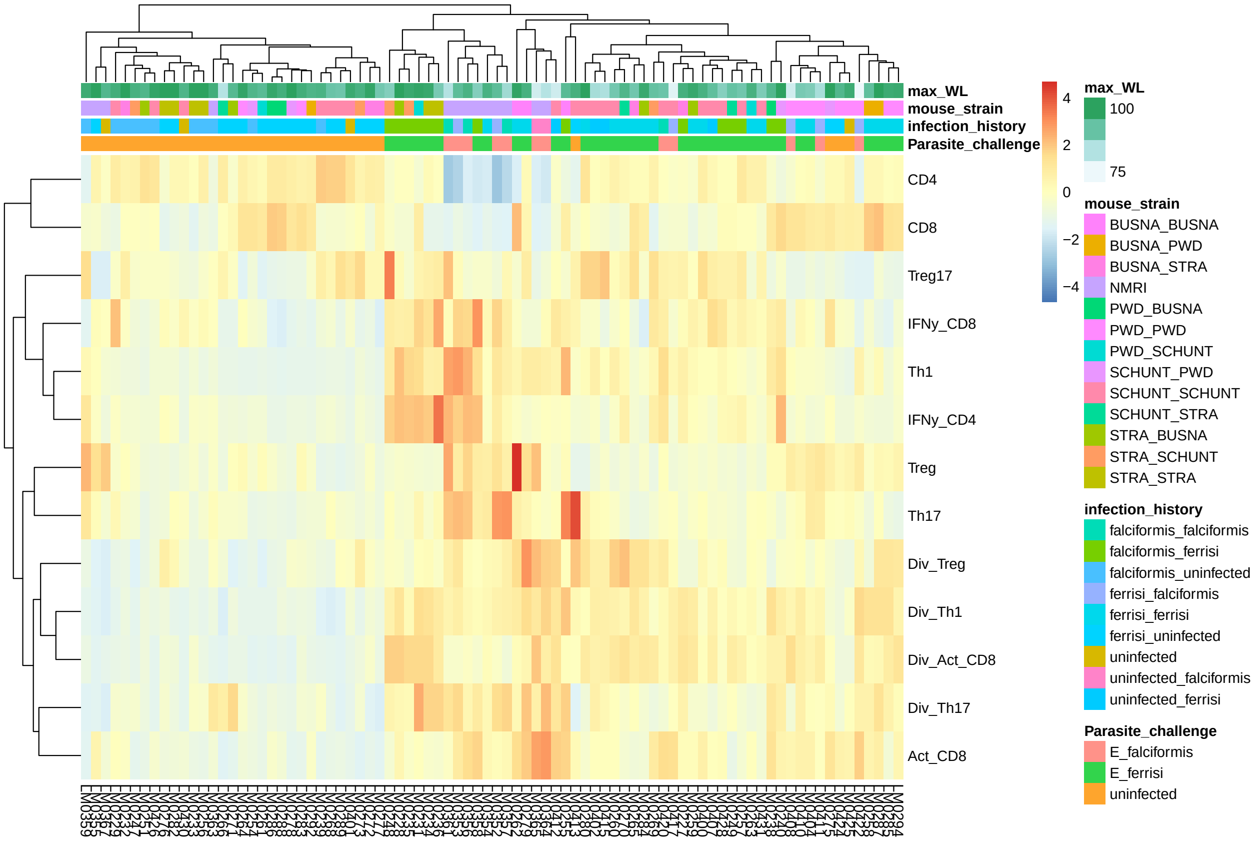
repeating the heatmap on the now imputed data

```
# turn the data frame into a matrix and transpose it. We want to have each cell
# type as a row name
facs_mouse <- t(as.matrix(facs_mouse))

# turn the first row into column names
facs_mouse %>%
    row_to_names(row_number = 1) -> heatmap_data

heatmap_data <- as.data.frame(heatmap_data)

table(rowSums(is.na(heatmap_data)) == nrow(heatmap_data))
```

```
##
## FALSE
##    13
```

```
# turn the columns to numeric other wise the heatmap function will not work
 heatmap_data[] <- lapply(heatmap_data, function(x) as.numeric(as.character(x)))

 # remove columns with only NAs
 heatmap_data <- Filter(function(x)!all(is.na(x)), heatmap_data)
 #remove rows with only Nas
 heatmap_data <-  heatmap_data[, colSums(is.na(heatmap_data)) !=
                                   nrow(heatmap_data)]


#Prepare the annotation data frame
annotation_df <- as_tibble(lab) %>%
    dplyr::select(c("Mouse_ID", "Parasite_challenge", "infection_history",
                    "mouse_strain", "max_WL"))

annotation_df <- unique(annotation_df)

annotation_df <- as.data.frame(annotation_df)




### Prepare the annotation columns for the heatmap
rownames(annotation_df) <- annotation_df$Mouse_ID


# Match the row names to the heatmap data frame
rownames(annotation_df) <- colnames(heatmap_data)

#remove the unecessary column
annotation_df <- annotation_df %>% dplyr::select(-Mouse_ID, )
```

Heatmap on facs expression data:


max_WL
mouse_strain
infection_history
Parasite_challenge
CD4
CD8
Treg17
IFNy_CD8
Th1
IFNy_CD4
Treg
Th17
Div_Treg
Div_Th1
Div_Act_CD8
Div_Th17
Act_CD8
4
2
0
−2
−4
max_WL
100
75
mouse_strain
BUSNA_BUSNA
BUSNA_PWD
BUSNA_STRA
NMRI
PWD_BUSNA
PWD_PWD
PWD_SCHUNT
SCHUNT_PWD
SCHUNT_SCHUNT
SCHUNT_STRA
STRA_BUSNA
STRA_SCHUNT
STRA_STRA
infection_history
falciformis_falciformis
falciformis_ferrisi
falciformis_uninfected
ferrisi_falciformis
ferrisi_ferrisi
ferrisi_uninfected
uninfected
uninfected_falciformis
uninfected_ferrisi
Parasite_challenge
E_falciformis
E_ferrisi
uninfected