

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",
              "IL17A_CD4")
```

```
Facs_lab_imp <- paste(Facs_lab, "imp", sep = "_")

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

Facs_wild_imp <- paste(Facs_wild, "imp", sep = "_")
```

FACS

Lab

PCA on the lab genes *-imputed*

```
#select the genes and lab mice
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_imp))) %>%
  rename_with(~str_remove(., '_imp'))

# I will remove the treg prop, had way too few values for the imputation
# anyway
facs_mouse <- facs_mouse %>%
  dplyr::select(-Treg_prop)

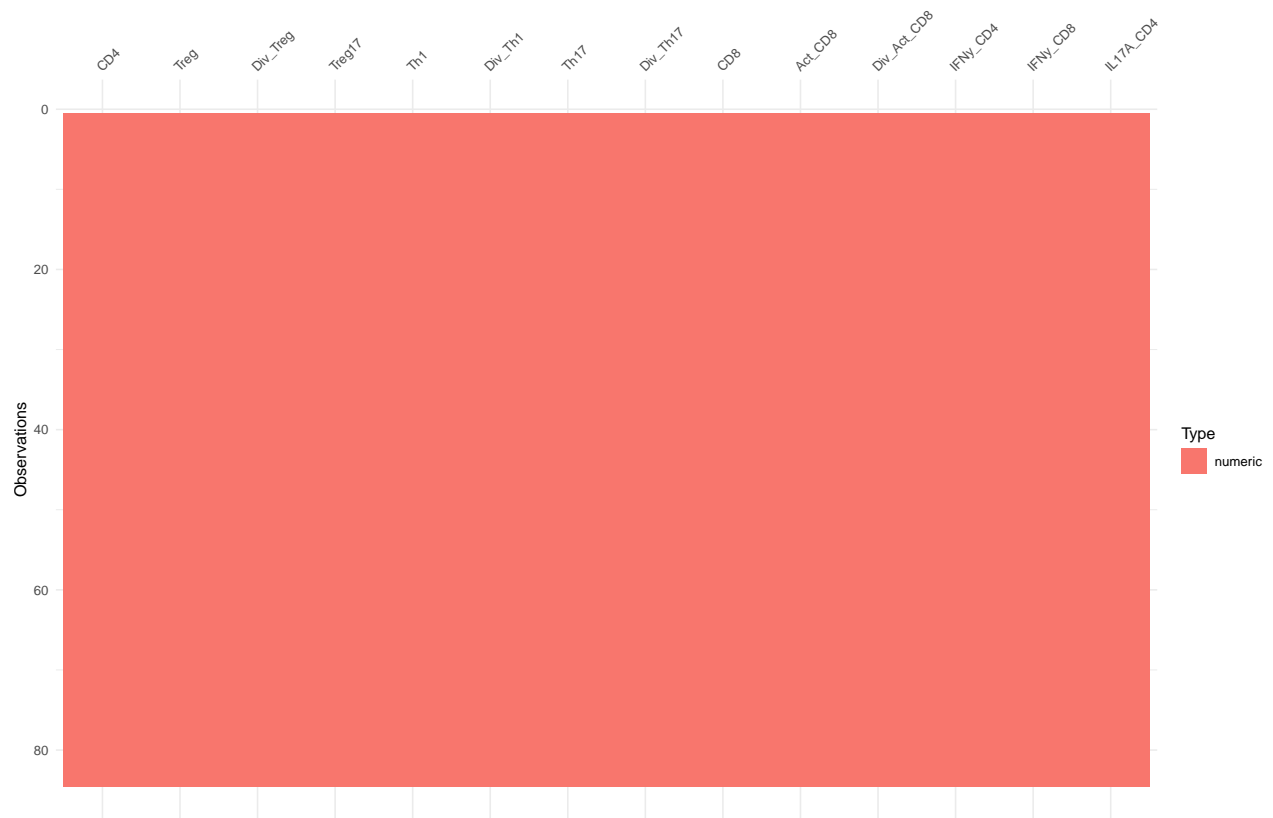
facs <- facs_mouse[, -1]

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

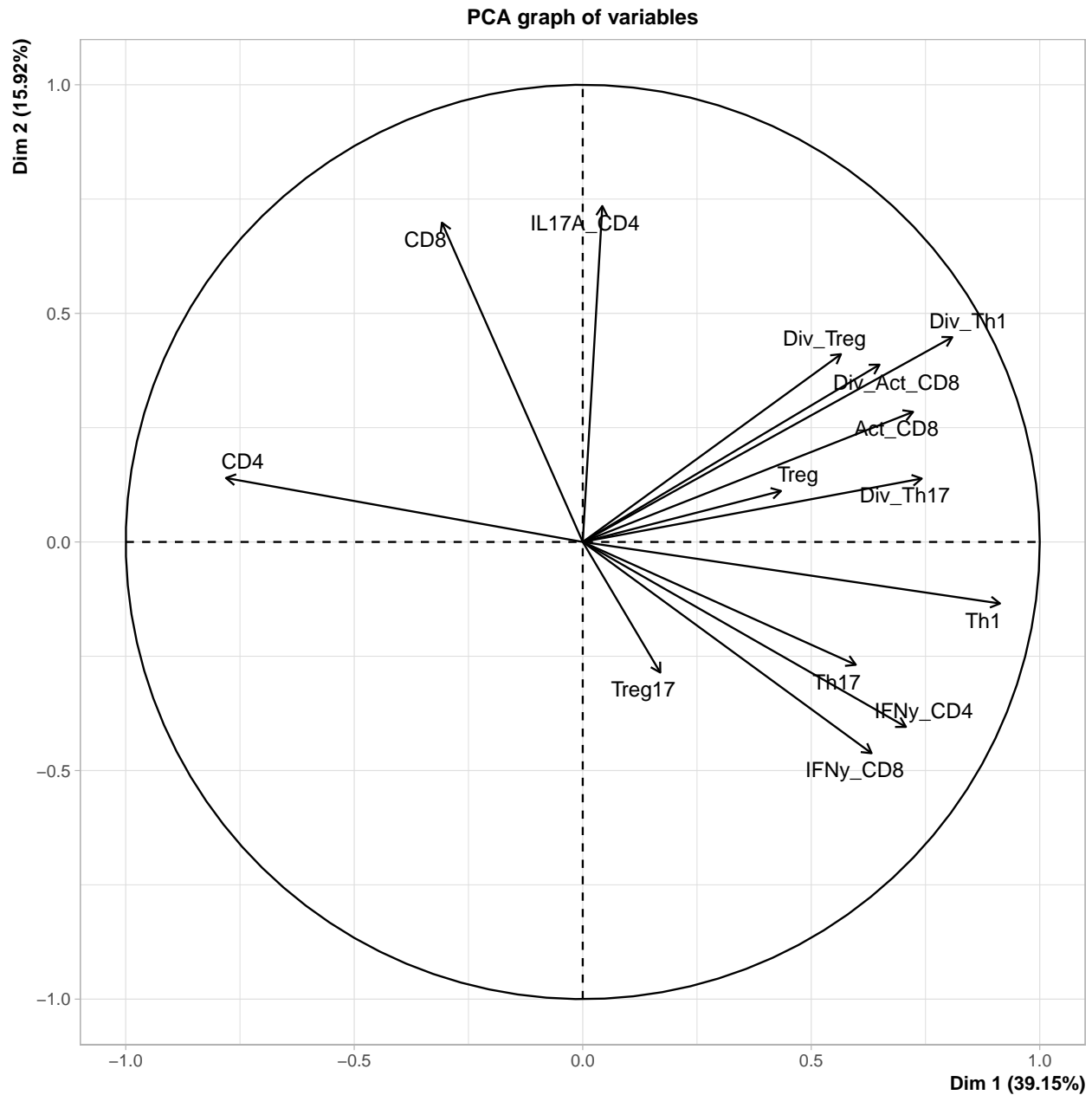
#remove columns with only nas
facs <- facs[rowSums(is.na(facs)) != ncol(facs), ]

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 missing data:

→ multiple 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

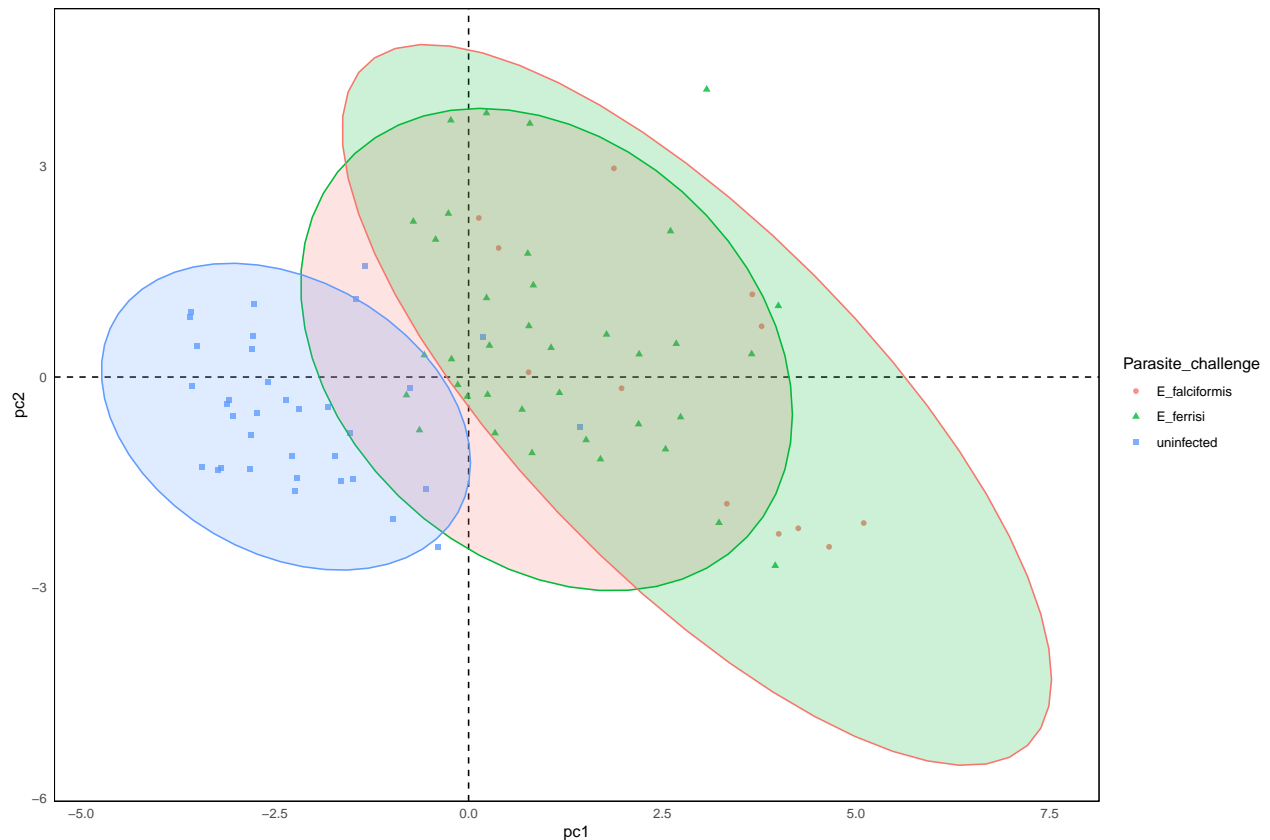
#Now we can make our initial plot of the PCA.

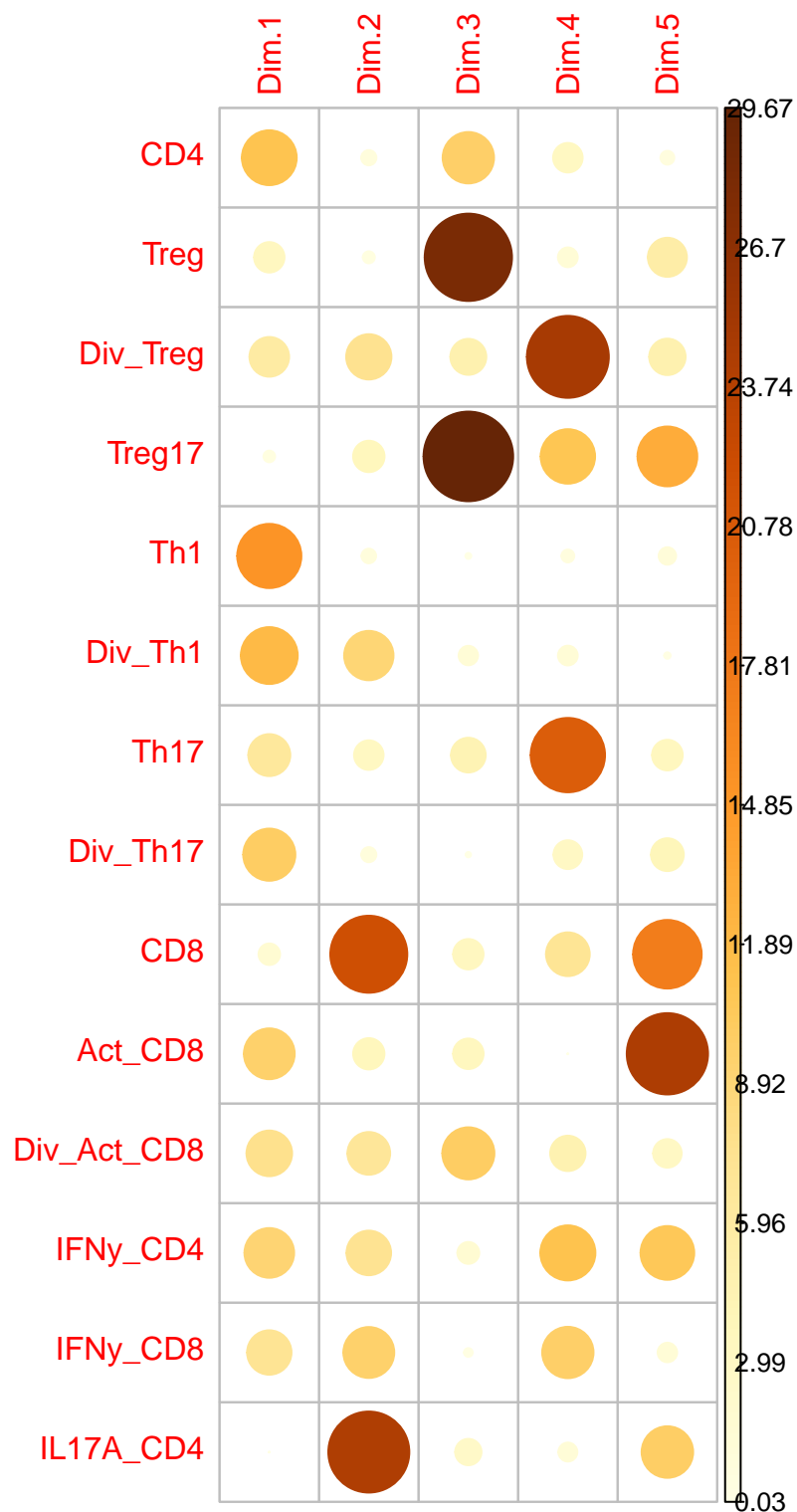
lab %>%

```
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"))
```

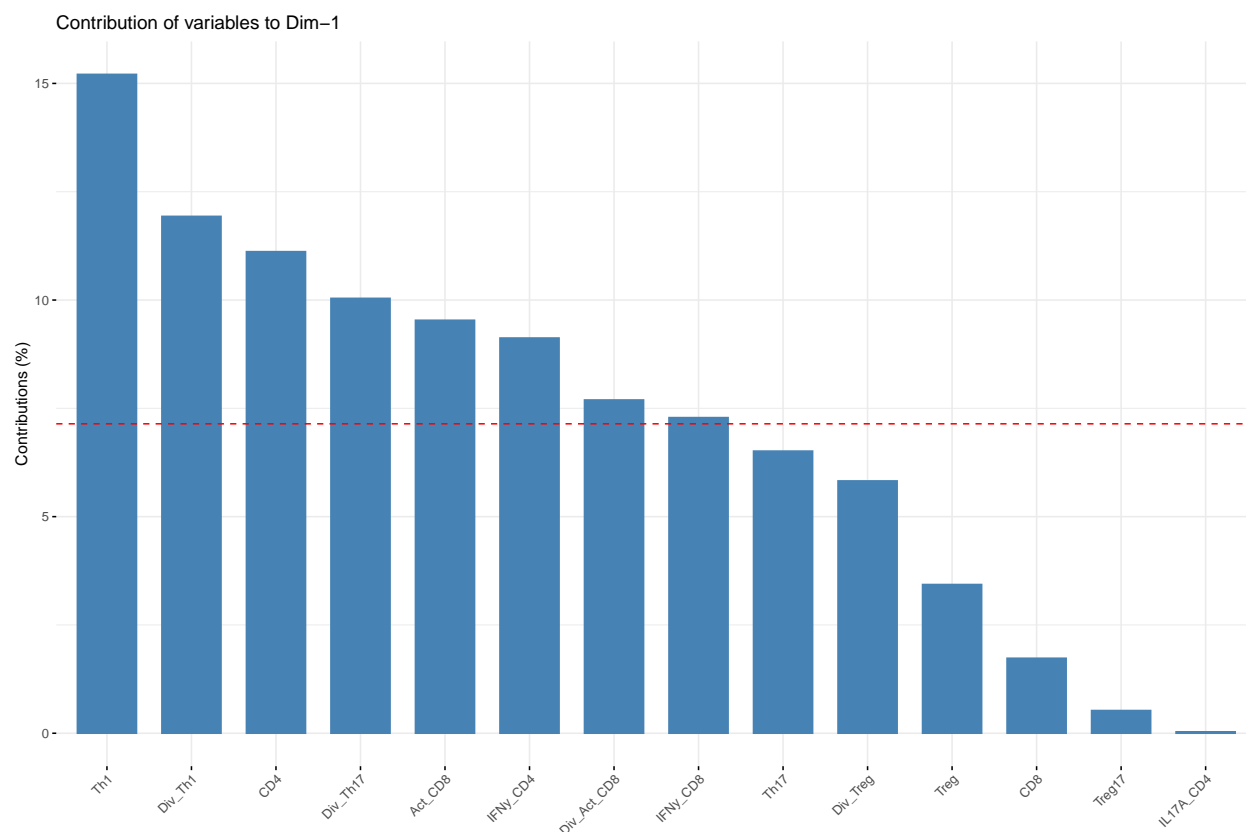
Warning: Removed 42 rows containing non-finite values (stat_ellipse).

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

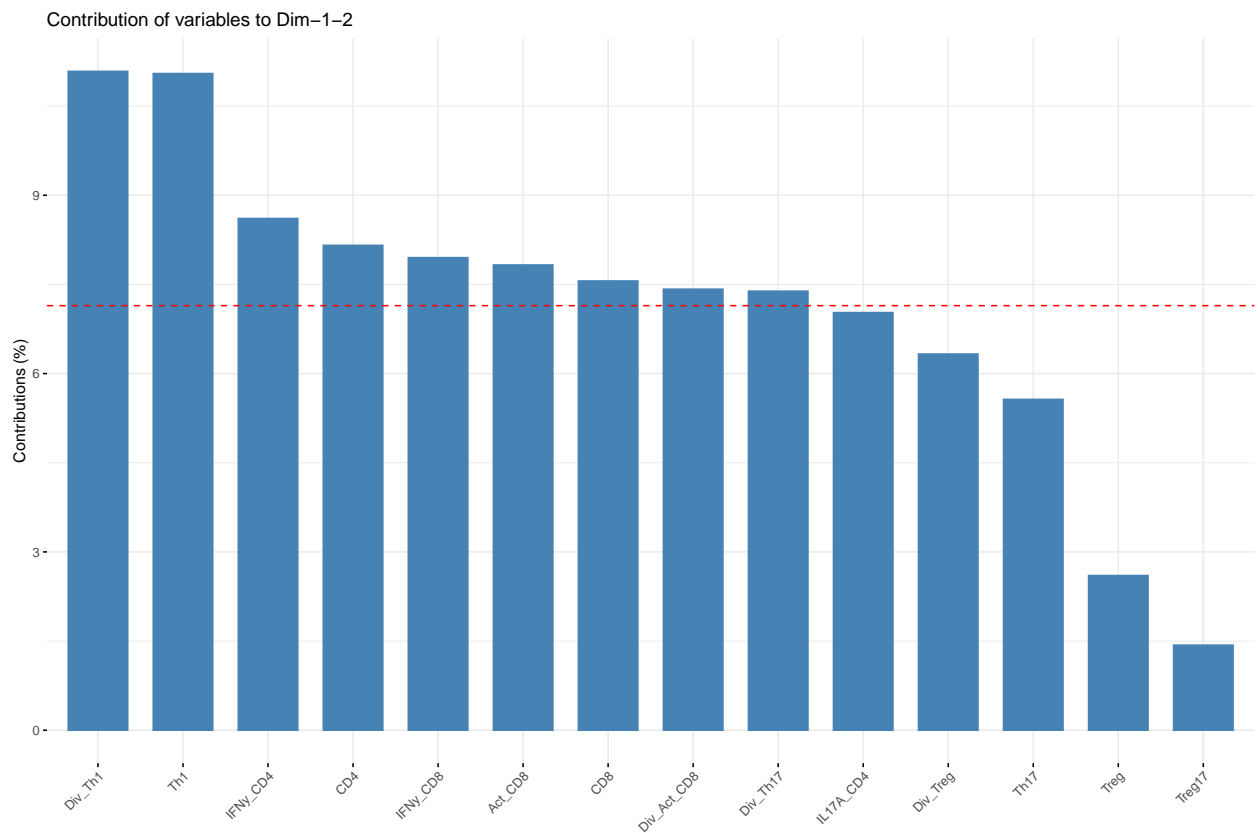
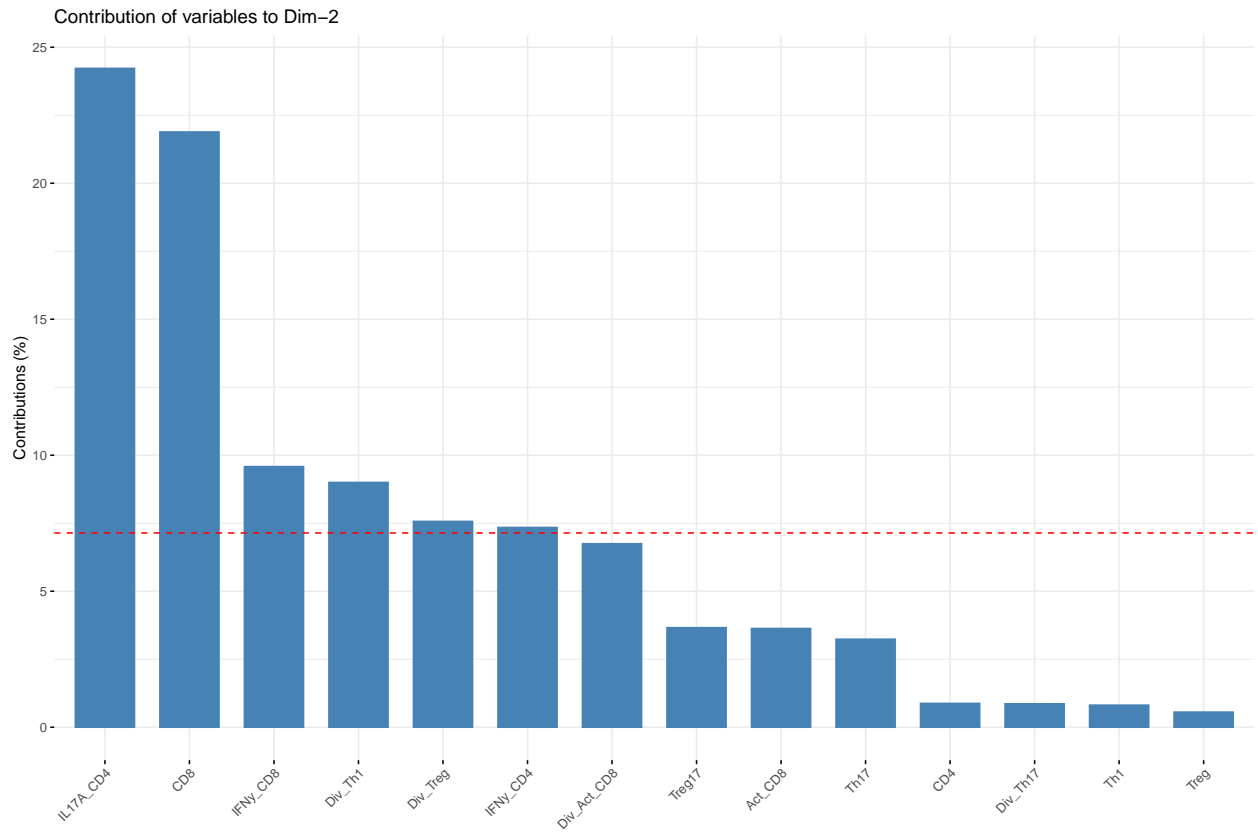




The function `fviz_contrib()` [factoextra package] can be used to draw a bar plot of variable contributions. If your data contains many variables, you can decide to show only the top contributing variables. The R code below shows the top 10 variables contributing to the principal components:



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

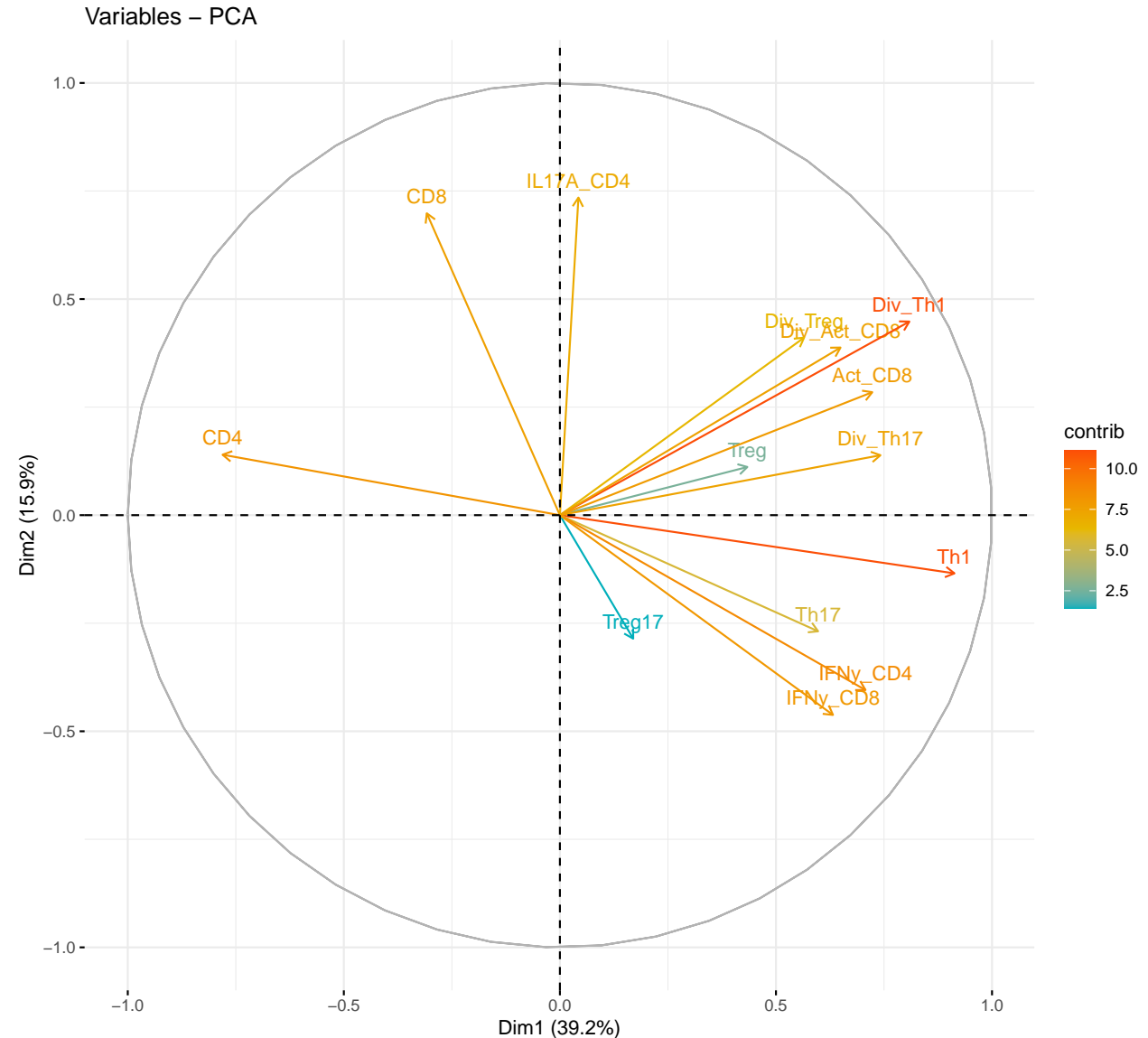


The red dashed line on the graph above indicates the expected average contribution. If the contribution of the variables were uniform, the expected value would be $1/\text{length}(\text{variables}) = 1/10 = 10\%$. For a given

component, a variable with a contribution larger than this cutoff could be considered as important in contributing to the component.

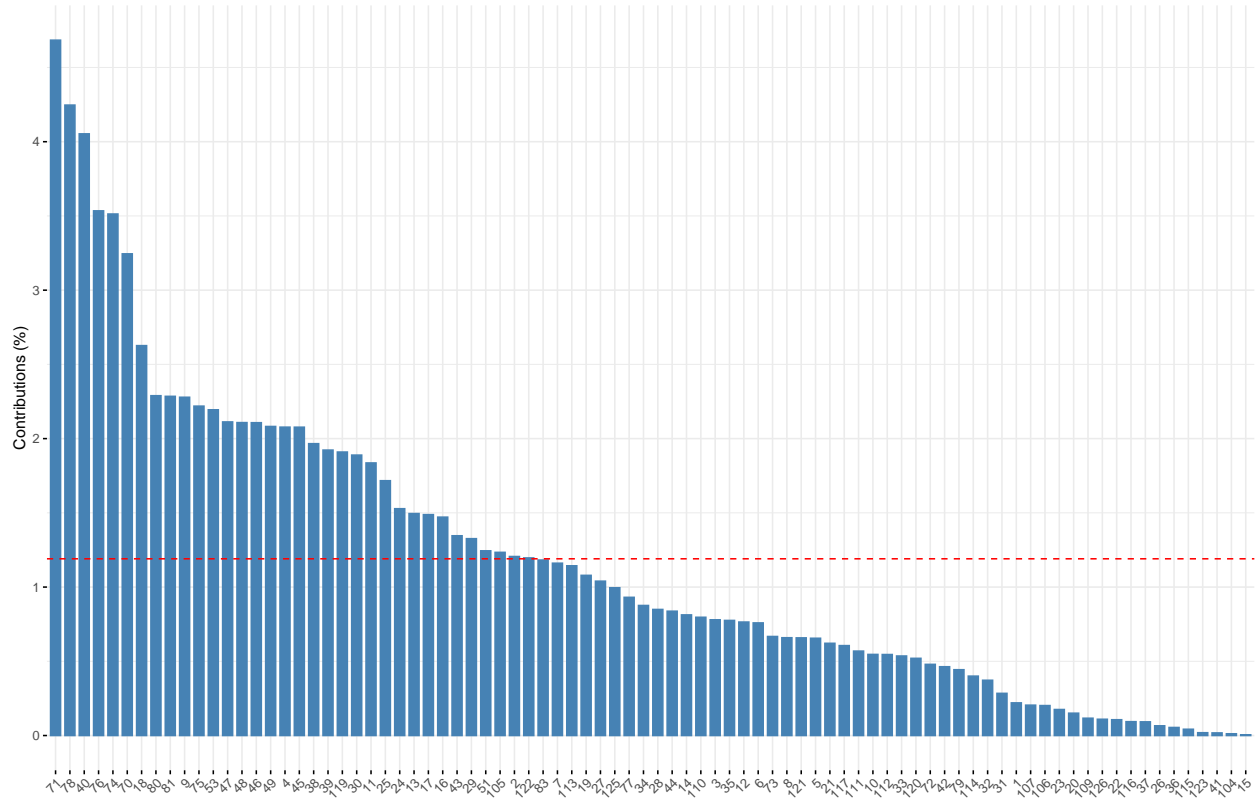
Note that, the total contribution of a given variable, on explaining the variations retained by two principal components, say PC1 and PC2, is calculated as $\text{contrib} = [(C1 * \text{Eig1}) + (C2 * \text{Eig2})] / (\text{Eig1} + \text{Eig2})$, where

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 : $[(10 * \text{Eig1}) + (10 * \text{Eig2})] / (\text{Eig1} + \text{Eig2})$



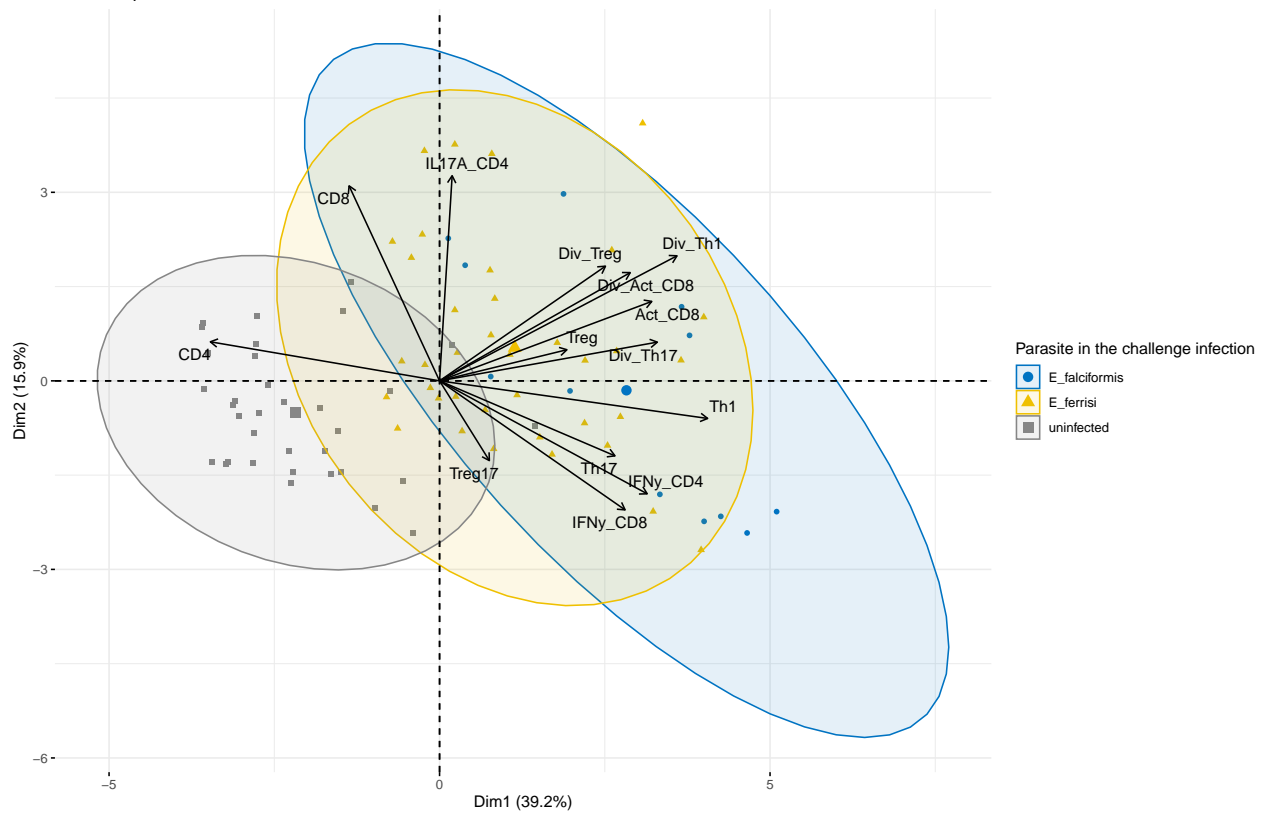
To visualize the contribution of individuals to the first two principal components:

Contribution of individuals to Dim-1-2



PCA + Biplot combination

PCA - Biplot



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
## -16.3641  -2.8972   0.1769   3.5488  10.0776
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      81.1312     1.8913  42.897 < 2e-16 ***
## pc1              0.9707     0.4359   2.227  0.0288 *
## pc2              0.1608     0.4143   0.388  0.6989
## Parasite_challengeE_ferrisi 10.6659     1.8080   5.899 8.63e-08 ***
## Parasite_challengeuninfected 17.0002     2.7972   6.078 4.06e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.029 on 79 degrees of freedom
## Multiple R-squared:  0.4182, Adjusted R-squared:  0.3887
## F-statistic: 14.2 on 4 and 79 DF,  p-value: 8.958e-09
## [1] 516.5965
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + Parasite_challenge + hybrid_status,
##     data = lab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.8963  -2.9751   0.5561   3.5844   9.4405
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      80.1192     2.0945  38.252 < 2e-16 ***
## pc1              0.8634     0.4708   1.834  0.0707 .
## pc2             -0.2589     0.6387  -0.405  0.6864
## Parasite_challengeE_ferrisi 10.6971     1.9478   5.492 5.35e-07 ***
## Parasite_challengeuninfected 16.0761     2.8283   5.684 2.46e-07 ***
## hybrid_statusF0 M. m. musculus 1.6445     2.2908   0.718  0.4751
## hybrid_statusF1 hybrid      3.4533     1.6269   2.123  0.0371 *
## hybrid_statusF1 M. m. domesticus -0.8890     2.0195  -0.440  0.6611
## hybrid_statusF1 M. m. musculus  3.8064     2.5860   1.472  0.1453
## hybrid_statusother          1.4421     2.1372   0.675  0.5019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.966 on 74 degrees of freedom
```

```
## Multiple R-squared:  0.4686, Adjusted R-squared:  0.404
## F-statistic:  7.25 on 9 and 74 DF,  p-value: 1.767e-07
## [1] 518.9854
```

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 used 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 -265.88
## 2    8 -267.62 -1  3.4778    0.0622 .
## ---
## 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 -265.88
## 2    8 -267.45 -1  3.1392    0.07643 .
## ---
## 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 -265.88
## 2    4 -270.36 -5  8.9728    0.1102
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + hybrid_status, data = lab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.0096  -3.7787   0.5772   4.7004   8.8613
##
```

```
## Coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      91.3168      1.2467  73.244 <2e-16 ***
## pc1             -0.5810      0.3242  -1.792  0.0771 .
## pc2             -1.1963      0.7032  -1.701  0.0930 .
## hybrid_statusF0 M. m. musculus    2.7873      2.6771    1.041  0.3011
## hybrid_statusF1 hybrid           4.7950      1.9518    2.457  0.0163 *
## hybrid_statusF1 M. m. domesticus -0.3158      2.4372   -0.130  0.8973
## hybrid_statusF1 M. m. musculus    5.4516      3.1138    1.751  0.0840 .
## hybrid_statusother -1.2815      2.3761   -0.539  0.5912
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.027 on 76 degrees of freedom
## Multiple R-squared:  0.1961, Adjusted R-squared:  0.1221
## F-statistic: 2.649 on 7 and 76 DF,  p-value: 0.01662
## [1] 549.7534
##
## Call:
## lm(formula = max_WL ~ pc1 + pc2 + infection_history, data = lab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.7161  -2.9197  -0.2227   3.1785   8.8796
##
## Coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      84.3321      2.7127  31.088 < 2e-16
## pc1              1.1144      0.4151   2.685 0.008979
## pc2              0.2162      0.4317   0.501 0.618043
## infection_historyfalciformis_ferrisi    7.0609      2.6599   2.655 0.009743
## infection_historyfalciformis_uninfected 15.3306      3.4979   4.383 3.86e-05
## infection_historyferrisi_falciformis   -4.2766      3.0837  -1.387 0.169704
## infection_historyferrisi_ferrisi       9.2546      2.9714   3.115 0.002632
## infection_historyferrisi_uninfected    13.5255      3.4222   3.952 0.000177
## infection_historyuninfected            13.9328      3.9544   3.523 0.000739
## infection_historyuninfected_falciformis -8.7632      4.0777  -2.149 0.034947
## infection_historyuninfected_ferrisi    -1.4029      3.3967  -0.413 0.680795
##
## (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: 4.523 on 73 degrees of freedom
## Multiple R-squared:  0.5652, Adjusted R-squared:  0.5056
## F-statistic: 9.488 on 10 and 73 DF,  p-value: 6.717e-10
## [1] 504.1382

##
## Call:
## lm(formula = max_WL ~ pc1 + pc2, data = lab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.8907  -2.7042   0.9352   4.3391   9.2512
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  92.9127     0.6720 138.271 < 2e-16 ***
## pc1         -0.8610     0.2870  -3.000  0.00359 **
## pc2         -0.3353     0.4501  -0.745  0.45844
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.159 on 81 degrees of freedom
## Multiple R-squared:  0.1055, Adjusted R-squared:  0.08343
## F-statistic: 4.777 on 2 and 81 DF,  p-value: 0.01093

##              df      AIC
## weight_lm      6 516.5965
## weight_lm_exp_only 4 548.7262
```

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
##      14

# 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 unnecessary column
annotation_df <- annotation_df %>% dplyr::select(-Mouse_ID, )

```

Heatmap on gene expression data:

