PCA

2023-05-04

1. Load packages

```
library(tidyverse)
library(DESeq2)
library(ggrepel)
```

2. Data transformation

2.a. Import data sets

The following data sets are provided by Dr. Rebecca L. Young.

```
frog_raw_counts_greaterthan5 <- read.table(file = "/stor/work/Bio321G_RY_Spring2023/MiniProjects/PoisonF
                                           row.names = 1,
                                           header = TRUE,
                                           sep = ",")

frog_metadata <- read.delim(file = "/stor/work/Bio321G_RY_Spring2023/MiniProjects/PoisonFrogLivers/Metad
```

2.b. Normalization

Since the gene length is not provided, we will use DeSeq2 to normalize our raw counts

```
dds <- DESeqDataSetFromMatrix(countData=frog_raw_counts_greaterthan5, design = ~condition,
                              colData=frog_metadata)

dds <- estimateSizeFactors(dds)

normalized_counts_greater_5 <- counts(dds, normalized = TRUE)
```

3. Principal Component Analysis (PCA)

PCA is a dimensionality reduction method which reduces the number of dimensions of multi-dimensional data sets, which helps us to visualize and interpret the data much better. While reducing the dimension, PCA still preserves the amount of information, allowing a comprehensive overview of the data set.

```r
x <- normalized_counts_greater_5 %>%
  t()   # Transpose the matrix

PC_x <- prcomp(x)   # Calculate the PCs

PCs_x <- data.frame(PC_x$x) %>%
  rownames_to_column(var = "sample_id")

head(PCs_x, 10)
```

3.a. Calculate the PCs

```
##      sample_id       PC1         PC2         PC3        PC4         PC5
## 1   A1.6854_S1 -39562.41   -342.5773  -9404.7575 -2073.1319   3305.1680
## 2  A3.6830_S17  21909.17  27432.2706  -8311.4987 -1777.0711 -11904.6700
## 3   B1.6855_S2  71864.29  10319.9164   1552.8794  1712.2488   2081.8945
## 4  B2.6813_S10  18565.00 -13279.3162   6328.9341 -1895.6037  -2819.1610
## 5  B3.6832_S18   7140.44 -14883.7208  -5635.1653  3048.1746    274.9454
## 6   C1.6872_S3 -18569.83   -296.5142  -5994.8365 -1842.0134   3851.0146
## 7  C2.6821_S11 -23194.78 -22919.1079    221.8982  7814.0353  -4806.9118
## 8  C3.6835_S19 -13121.78  -1836.0159 -10054.3150  -231.2435   1617.8117
## 9   D1.6863_S4  81802.01   5185.5556   4677.7633  -193.3651   2021.8303
## 10 D2.6822_S12  69158.21  11080.2149   4600.0748  3547.2010    998.4706
##           PC6        PC7         PC8        PC9       PC10        PC11
## 1   3536.1971 -3685.3603 -1874.67620  262.12854  -771.0563  1593.94782
## 2  -2022.9889 -1031.0081  -111.07274  116.52500  -157.5630   -18.53249
## 3   1823.5021  -426.0612   133.64454  718.78058 -1328.5323   292.11726
## 4    803.8003  2457.5121 -3407.07094 -871.51686 -1417.1613   551.76973
## 5  -1972.6157  3658.5318 -1114.77746 1127.91413 -1335.1018  -569.91620
## 6  -1246.1407 -2199.0722    68.05519 -806.05795  -968.7915  -302.13275
## 7   4920.5982 -1945.0609  1929.57188  -27.43375  -107.8694 -1288.72222
## 8   1156.5282  4388.7478  2178.49084  281.50823  2453.7572   629.57654
## 9  -1267.8385 -1294.2309 -1517.26271 3528.18086  1643.8254 -1434.76485
## 10  1501.0252  -704.6848   502.03412 -704.49876   746.7140  1150.37772
##             PC12        PC13       PC14        PC15        PC16        PC17
## 1    -861.311380   541.70561 -913.06251    85.11898  397.166369  -639.24666
## 2       7.380523   -44.96605   43.91261    14.88528   -6.435194   -21.42287
## 3    -926.788212  -832.44594   29.58555  1378.59841 -460.290638  1190.53482
## 4     936.206540  1553.29916  849.74595   121.73696  564.854821   212.44162
## 5   -2191.008381  -620.60140  695.20608  -445.95992 -383.587327  -479.47799
## 6    1430.572363  -572.47558 1346.25492   558.15052   -9.569817   375.21058
## 7      21.884458   860.42629 -141.12892  -176.08788  -33.115249   294.13914
## 8     420.231837   546.56404  392.03775   921.44236  650.218390   -80.26483
## 9     422.733715   673.50165 -426.62006   163.34380  444.987201  -133.59878
## 10   -273.432763 -1416.43456  802.03862 -1424.12220 1240.828080    29.92035
##           PC18          PC19
## 1     74.14985  2.332797e-11
```

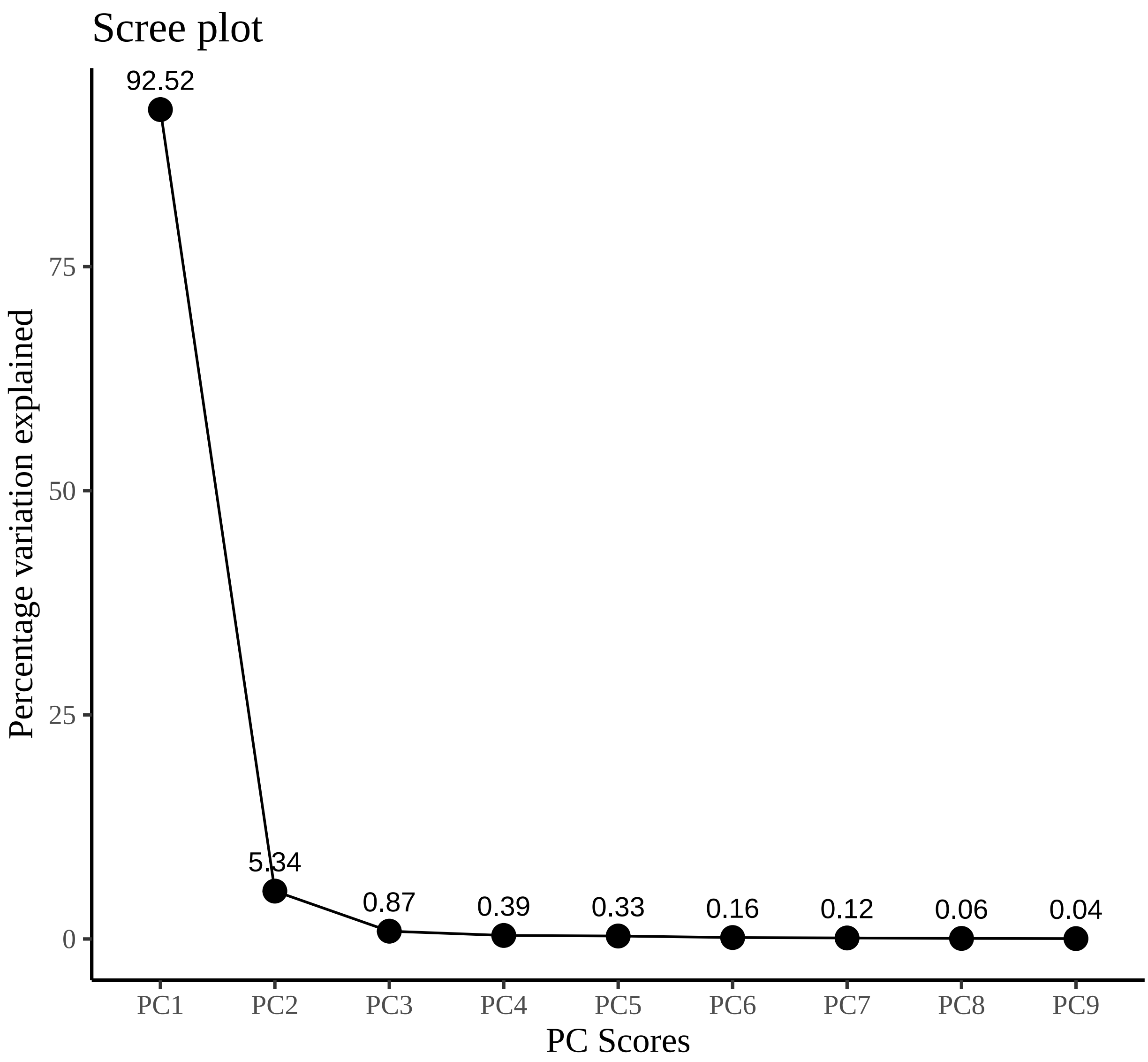
```
## 2     11.53935  3.227832e-12
## 3    680.90310 -2.043285e-12
## 4    335.57861  1.041319e-11
## 5   -398.64053 -1.209406e-11
## 6  -1220.18891  3.211840e-12
## 7   -310.60342  6.223478e-12
## 8    285.34503 -1.921234e-11
## 9   -387.67161  7.029710e-13
## 10    27.69222 -1.132524e-10
```

3.b. Scree plot A common method for determining the number of PCs to be retained is a graphical representation known as a scree plot.

```
# Calculate the variation explained by each PCs
var_explained <- data.frame(PC = paste0("PC", 1:ncol(PC_x$x)),
                            var_explained=(PC_x$sdev)^2/sum((PC_x$sdev)^2))

PC1to9_Var <- var_explained[1:9,]

ggplot(PC1to9_Var, aes(x= PC,y = var_explained * 100, group = 1)) +
  geom_point(size=4) +
  geom_line() +
  geom_text(aes(label = round(var_explained, 4)*100, vjust = -1)) +
  labs(title = "Scree plot", y = "Percentage variation explained", x = "PC Scores") +
  theme_classic(base_family = "Times",
                base_size = 14)
```

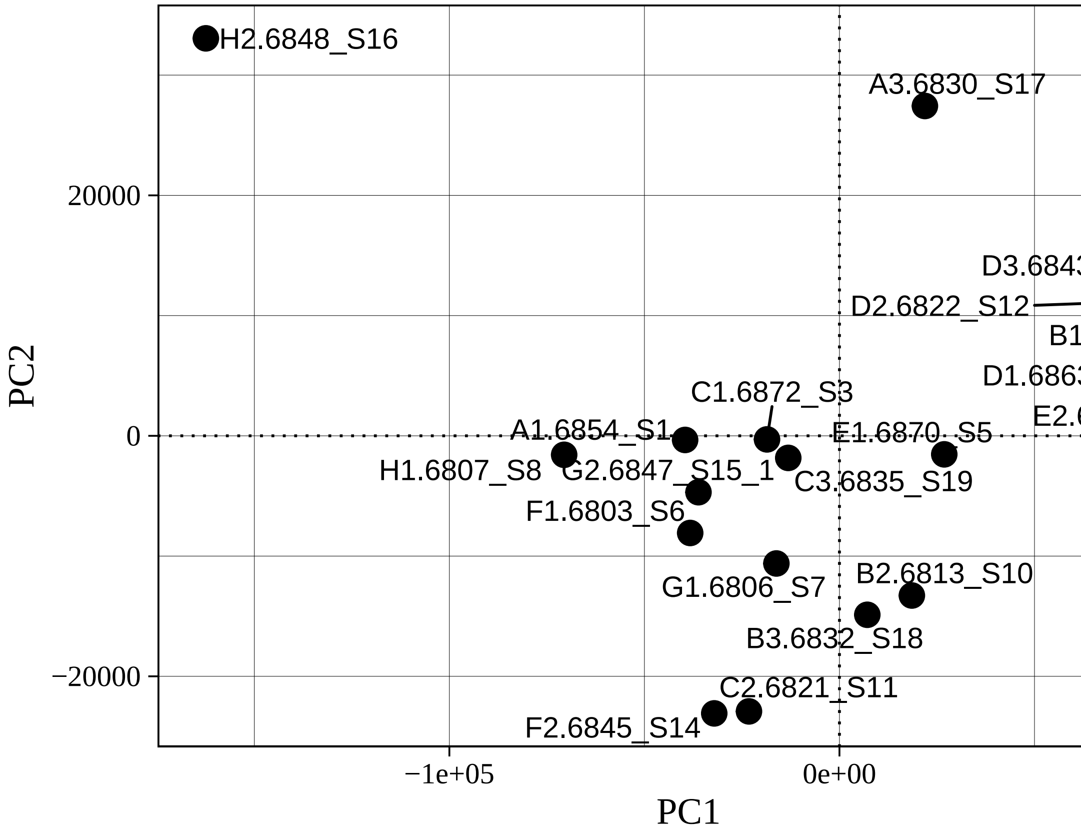


The scree plot criterion looks for the "elbow" in the curve and selects all components just before the line flattens out. In this scree plot, the elbow point is at PC2, indicating that we only need to focus on the first two PCs. Furthermore, the first two PCs explain for almost 93% of the variation, indicating that PCA is appropriate to use in this case

4. Visualization - PCAs

Now, let"s plot PCA for PC1 and PC2

```r
ggplot(data = PCs_x, aes(x = PC1, y = PC2)) +
  geom_point(size = 4) +
  geom_text_repel(aes(x = PC1, y = PC2, label = sample_id)) +
  geom_hline(yintercept = 0, linetype = "dotted") +
  geom_vline(xintercept = 0, linetype = "dotted") +
  theme_linedraw(base_family = "Times",
                 base_size = 14)
```



4.a. Check for anomalies

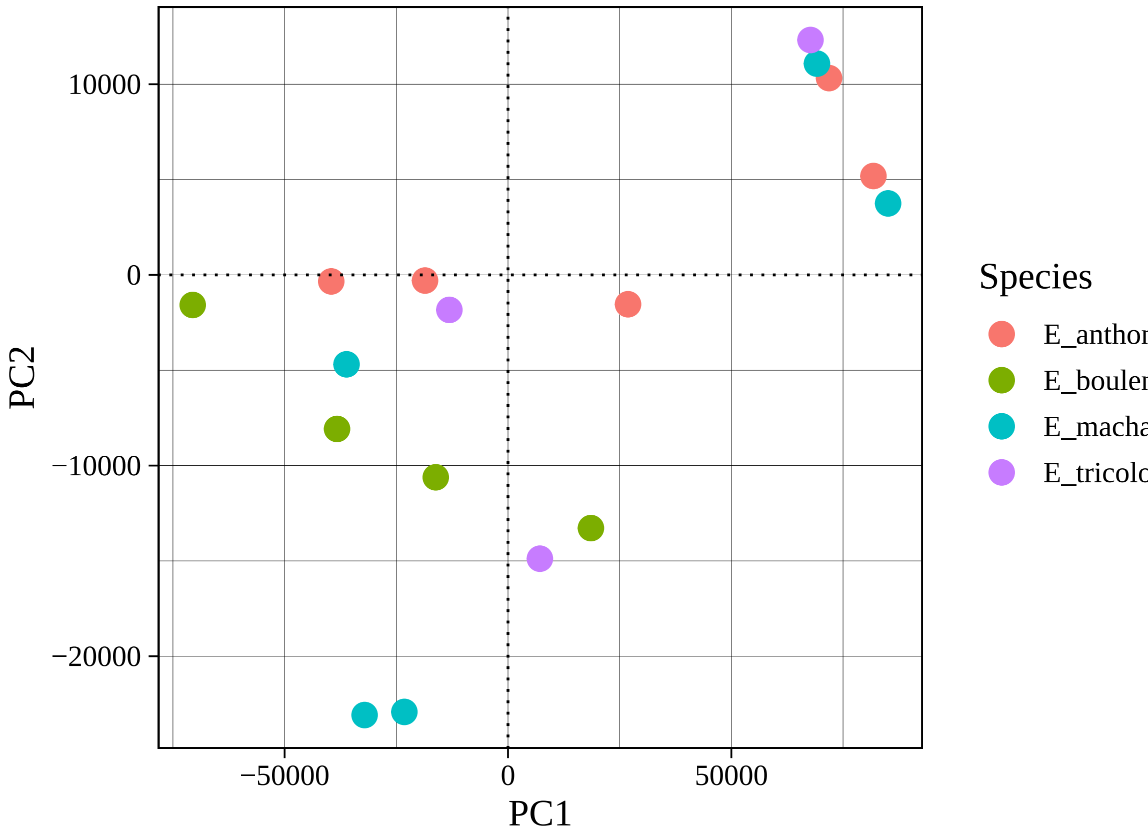
This might look daunting at first, but the purpose is to identify the anomalies, which are the two samples `H2.6848_S16` and `A3.6830_S17`.

```r
# Remove rows containing the two anomalies
PCs_x <- PCs_x %>%
  subset(sample_id != c("H2.6848_S16", "A3.6830_S17"))
```

```r
# Add another column which provides the species related to each gene
PCs_x <- PCs_x %>%
  mutate(Species = case_when(
```

```
    sample_id == "A1.6854_S1" ~ "E_anthonyi",
    sample_id == "B1.6855_S2"~ "E_anthonyi",
    sample_id == "C1.6872_S3" ~ "E_anthonyi",
    sample_id ==  "D1.6863_S4"~ "E_anthonyi",
    sample_id == "E1.6870_S5"~ "E_anthonyi",
    sample_id == "F1.6803_S6" ~ "E_boulengeri",
    sample_id == "G1.6806_S7" ~ "E_boulengeri",
    sample_id == "H1.6807_S8"~ "E_boulengeri",
    sample_id == "B2.6813_S10" ~ "E_boulengeri",
    sample_id ==  "C2.6821_S11"~ "E_machalilla",
    sample_id == "D2.6822_S12"~ "E_machalilla",
    sample_id == "E2.6826_S13" ~ "E_machalilla",
    sample_id == "F2.6845_S14" ~ "E_machalilla",
    sample_id == "G2.6847_S15_1" ~ "E_machalilla",
    sample_id == "H2.6848_S16" ~ "E_machalilla",
    TRUE ~ "E_tricolor"))   # The remaining ones are E_tricolor


# Plot
ggplot(data = PCs_x, aes(x = PC1, y = PC2, color = Species)) +
  geom_point(size = 4) +
  geom_hline(yintercept = 0, linetype = "dotted") +
  geom_vline(xintercept = 0, linetype = "dotted") +
  theme_linedraw(base_family = "Times",
                 base_size = 14)
```



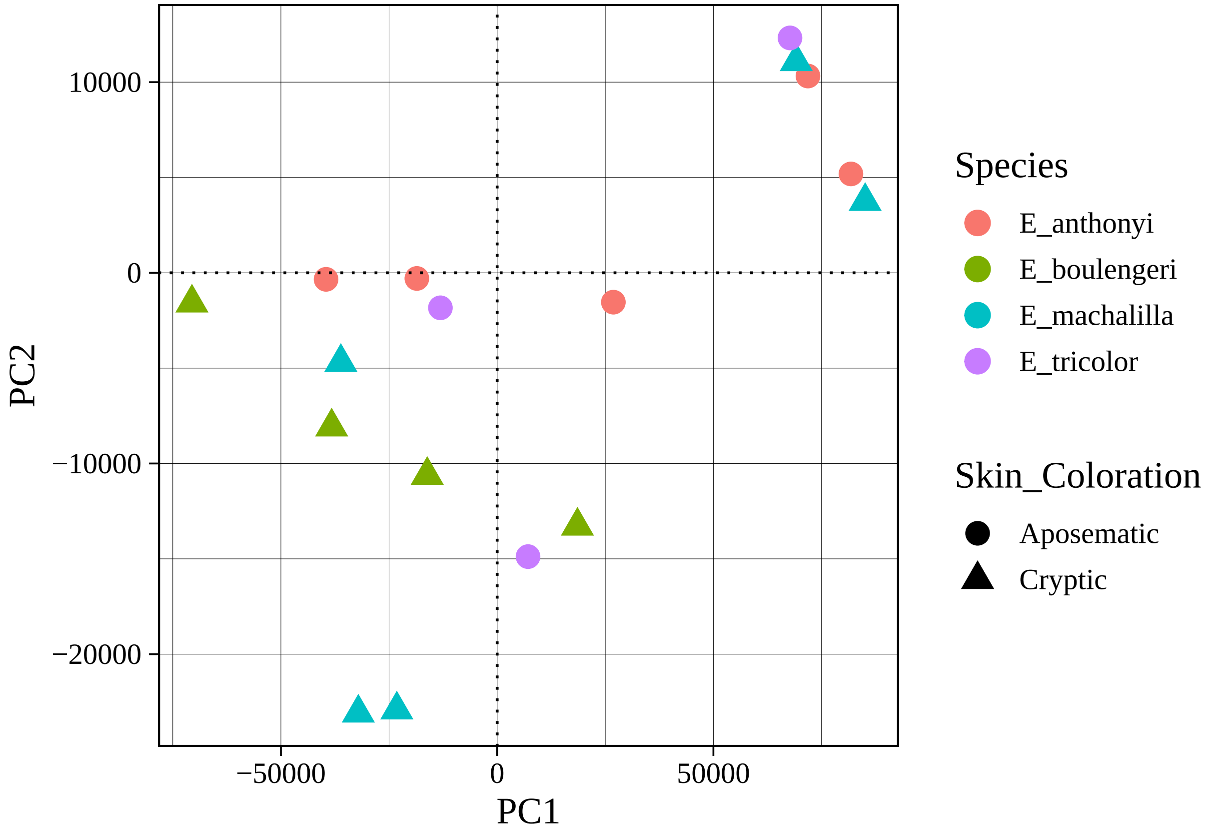
4.b. Plot by species

We can see that there"s no clear pattern/cluster.

4.c. Plot by skin coloration Let"s try a different approach by cateogorizing the points by cryptic/aposematic

```
# Add another column which provides the skin coloration related to each species
PCs_x <- PCs_x %>%
  mutate(Skin_Coloration = case_when(
    Species == "E_anthonyi" ~ "Aposematic",
    Species == "E_tricolor" ~ "Aposematic",
    Species == "E_machalilla" ~ "Cryptic",
    Species == "E_boulengeri" ~ "Cryptic"))

# Plot
ggplot(data = PCs_x, aes(x = PC1, y = PC2, shape = Skin_Coloration, color = Species)) +
  geom_point(size = 4) +
  geom_hline(yintercept = 0, linetype = "dotted") +
  geom_vline(xintercept = 0, linetype = "dotted") +
  theme_linedraw(base_family = "Times",
                 base_size = 14)
```

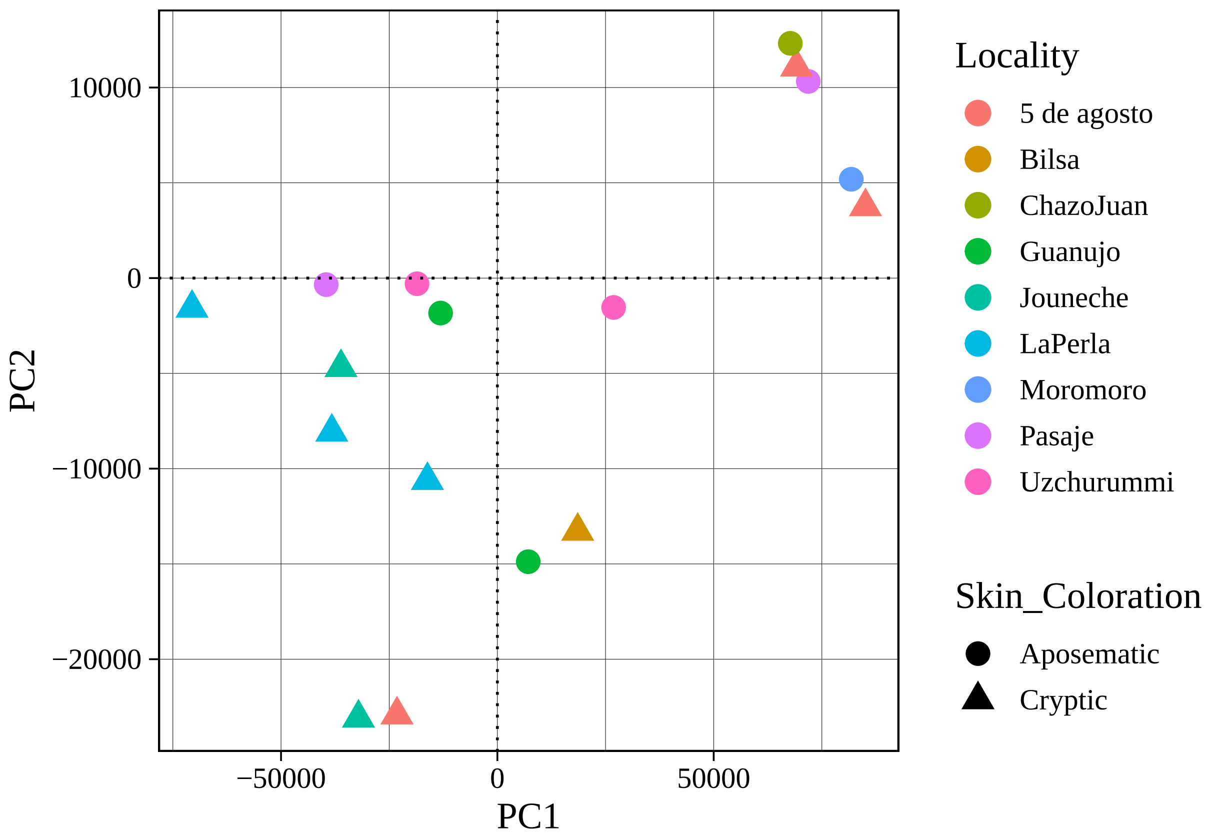


Again, we can see that there"s no clear pattern/cluster.

4.d. Plot by localities Another potential approach could be categorizing the points by where the samples of Epipedobates were collected. The information regarding the localities is provided by Dr. Rebecca L. Young.

```
# Add another column which provides the localities at which the Epipedobates were collected
PCs_x <- PCs_x %>%
  mutate(Locality = case_when(
    sample_id == "A1.6854_S1" ~ "Pasaje", sample_id == "B1.6855_S2" ~ "Pasaje",
    sample_id == "C1.6872_S3" ~ "Uzchurummi", sample_id ==  "D1.6863_S4" ~ "Moromoro",
    sample_id == "E1.6870_S5" ~ "Uzchurummi", sample_id == "F1.6803_S6" ~ "LaPerla",
    sample_id == "G1.6806_S7" ~ "LaPerla", sample_id == "H1.6807_S8" ~ "LaPerla",
    sample_id == "B2.6813_S10" ~ "Bilsa", sample_id ==  "C2.6821_S11"~ "5 de agosto",
    sample_id == "D2.6822_S12" ~ "5 de agosto", sample_id == "E2.6826_S13" ~ "5 de agosto",
    sample_id == "F2.6845_S14" ~ "Jouneche", sample_id == "G2.6847_S15_1" ~ "Jouneche",
    sample_id == "H2.6848_S16" ~ "Jouneche", sample_id == "A3.6830_S17" ~ "Guanujo",
    sample_id == "C3.6835_S19" ~ "Guanujo", sample_id == "B3.6832_S18" ~ "Guanujo",
    sample_id == "D3.6843_S20" ~ "ChazoJuan"))

# Plot
ggplot(data = PCs_x, aes(x = PC1, y = PC2, shape = Skin_Coloration, color = Locality)) +
  geom_point(size = 4) +
  geom_hline(yintercept = 0, linetype = "dotted") +
  geom_vline(xintercept = 0, linetype = "dotted") +
  theme_linedraw(base_family = "Times",
                 base_size = 14)
```



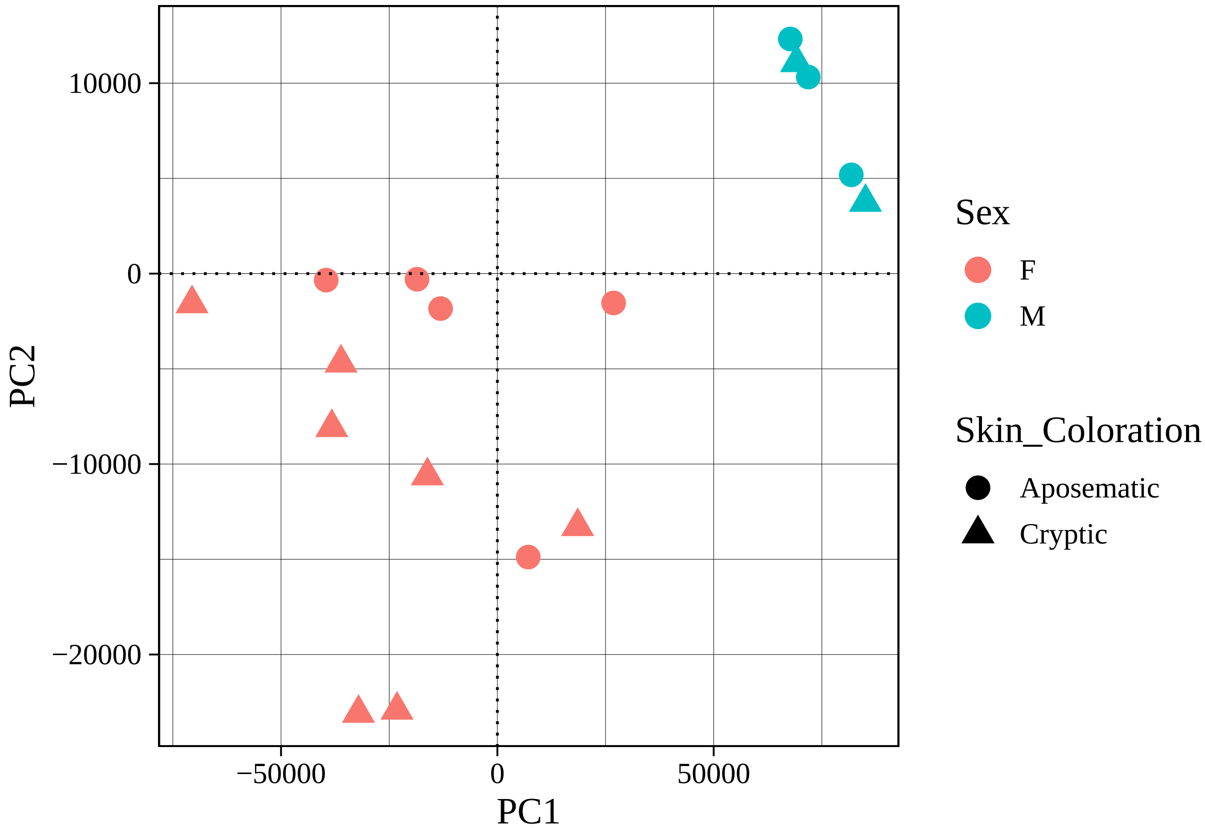
Again, we can see that there"s no clear pattern/cluster.

4.e. Plot by sex We can also try plotting the Epipedobates by sex (i.e., males and females)

```
# Add another column which provides the sex of the Epipedobates
PCs_x <- PCs_x %>%
  mutate(Sex = case_when(
    sample_id == "A1.6854_S1" ~ "F", sample_id == "B1.6855_S2"~ "M",
    sample_id == "C1.6872_S3" ~ "F", sample_id ==  "D1.6863_S4"~ "M",
    sample_id == "E1.6870_S5"~ "F", sample_id == "F1.6803_S6" ~ "F",
    sample_id == "G1.6806_S7" ~ "F", sample_id == "H1.6807_S8"~ "F",
    sample_id == "B2.6813_S10" ~ "F", sample_id ==  "C2.6821_S11"~ "F",
    sample_id == "D2.6822_S12"~ "M", sample_id == "E2.6826_S13" ~ "M",
    sample_id == "F2.6845_S14" ~ "F",sample_id == "G2.6847_S15_1" ~ "F",
    sample_id == "H2.6848_S16" ~ "F",sample_id == "A3.6830_S17" ~ "F",
    sample_id == "C3.6835_S19" ~ "F",sample_id == "B3.6832_S18" ~ "F",
    sample_id == "D3.6843_S20" ~ "M"))

# Plot
ggplot(data = PCs_x, aes(x = PC1, y = PC2, shape = Skin_Coloration, color = Sex)) +
  geom_point(size = 4) +
```

```
geom_hline(yintercept = 0, linetype = "dotted") +
geom_vline(xintercept = 0, linetype = "dotted") +
theme_linedraw(base_family = "Times",
               base_size = 14)
```



Here, we can see a clear cluster formed by the males. However, bare in mind that there's a disproportionate of males (5) compared to females (12) in this plot.

Overall, besides a potential separation due to sex, the PCA displays no separation between Cryptic and Aposematic Epipedobates