Assignment 2

Elliot Gavrin

2023-01-16

https://github.com/19eag3/BIOL432_Assignment2

Part I: Data Exploration 1.

```
MyData <-read.csv("C:/Users/egavr/OneDrive/Documents/BIOL432/Csv files/BirdBehaviour.csv")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(ggfortify)
```

2.

```
dim(MyData)
```

```
## [1] 1032    6
```

```
head(MyData)
```

```
##               Species   ID Groom Forage Mate Sleep
## 1  Erithacus rubecula 8837   245    127  237   184
## 2    Sturnus vulgaris 9282   234    177  245   158
## 3  Erithacus rubecula 8772   247    111  245   186
## 4 Carduelis carduelis 2179   232    108  302   152
## 5    Sturnus vulgaris 9116   254    147  266   139
## 6    Sturnus vulgaris   39   256    185  274   170
```

```
tail(MyData)
```

```
##                 Species   ID Groom Forage Mate Sleep
## 1027   Passer domestius 6862   234     95  254   185
## 1028 Erithacus rubecula  441   290    131  290   195
## 1029       Turdus merula 2902   322    110  289   146
## 1030       Turdus merula  344   297    117  271   161
## 1031   Sturnus vulgaris 7589   242    187  265   175
## 1032 Erithacus rubecula 9372   274    106  262   168
```

```
str(MyData)
```

```
## 'data.frame':    1032 obs. of  6 variables:
##  $ Species: chr  "Erithacus rubecula" "Sturnus vulgaris" "Erithacus rubecula" "Carduelis carduelis" ...
##  $ ID     : int  8837 9282 8772 2179 9116 39 5314 4718 5952 424 ...
##  $ Groom  : int  245 234 247 232 254 256 284 258 233 241 ...
##  $ Forage : int  127 177 111 108 147 185 114 164 160 128 ...
##  $ Mate   : int  237 245 245 302 266 274 260 280 252 268 ...
##  $ Sleep  : int  184 158 186 152 139 170 154 146 150 221 ...
```

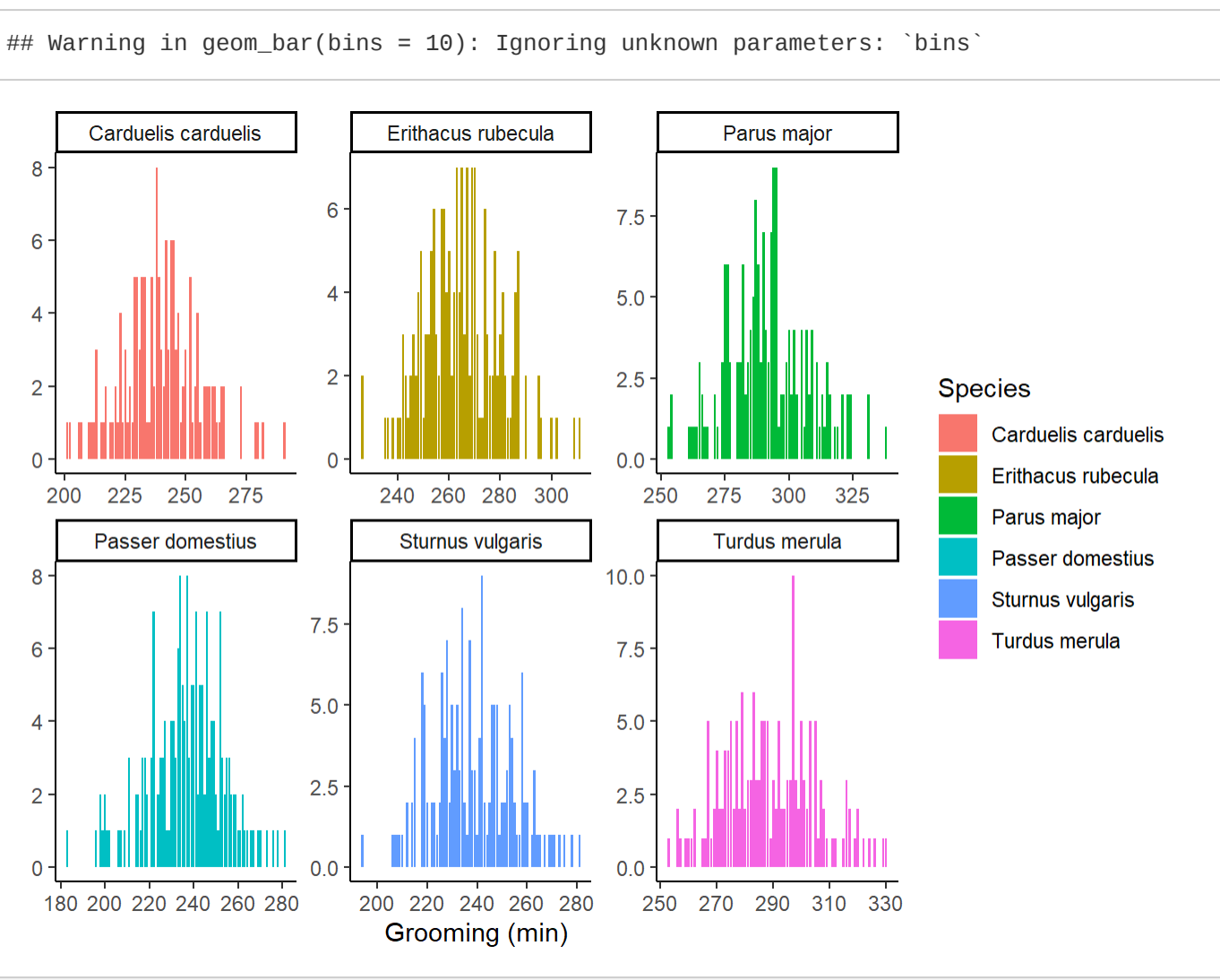
```
summary(MyData)
```

```
##    Species                ID           Groom           Forage
##  Length:1032        Min.   :   6   Min.   :183.0   Min.   : 89.0
##  Class :character   1st Qu.:2603   1st Qu.:238.0   1st Qu.:114.0
##  Mode  :character   Median :5207   Median :258.0   Median :123.0
##                     Mean   :5098   Mean   :260.3   Mean   :128.6
##                     3rd Qu.:7547   3rd Qu.:282.0   3rd Qu.:135.0
##                     Max.   :9973   Max.   :338.0   Max.   :199.0
##       Mate           Sleep
##  Min.   :205.0   Min.   :122.0
##  1st Qu.:252.0   1st Qu.:156.0
##  Median :269.0   Median :168.0
##  Mean   :275.1   Mean   :173.7
##  3rd Qu.:298.0   3rd Qu.:189.0
##  Max.   :363.0   Max.   :244.0
```

3. NEED FIGURE CAPTIONS

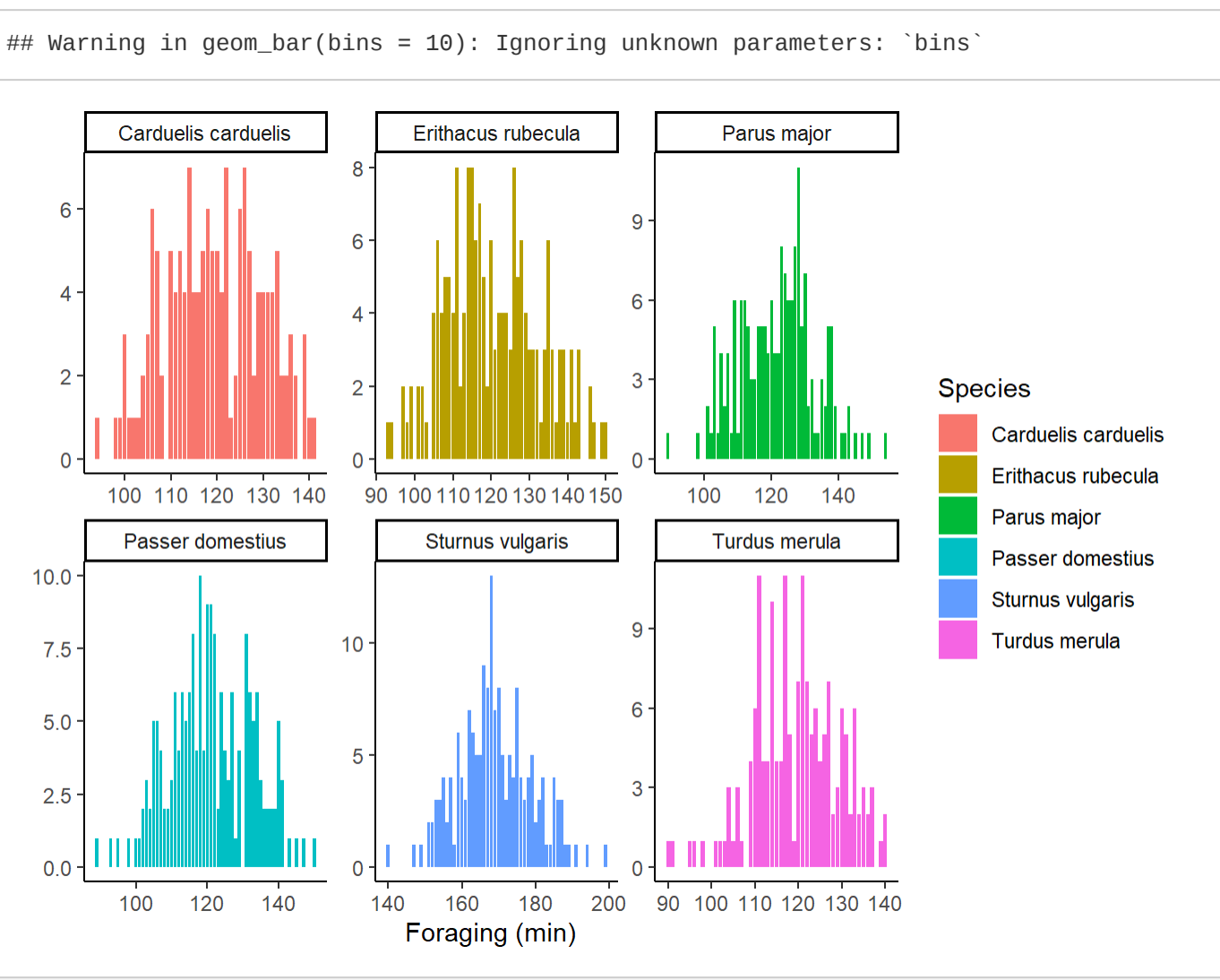
```
ggplot(aes(x=Groom,fill=(Species)), data=MyData)+
  geom_bar(bins=10)+
  labs(x = "Grooming (min)", y="", fill="Species")+
  facet_wrap(vars(Species), scales='free')+
  theme_classic()
```

```
## Warning in geom_bar(bins = 10): Ignoring unknown parameters: `bins`
```



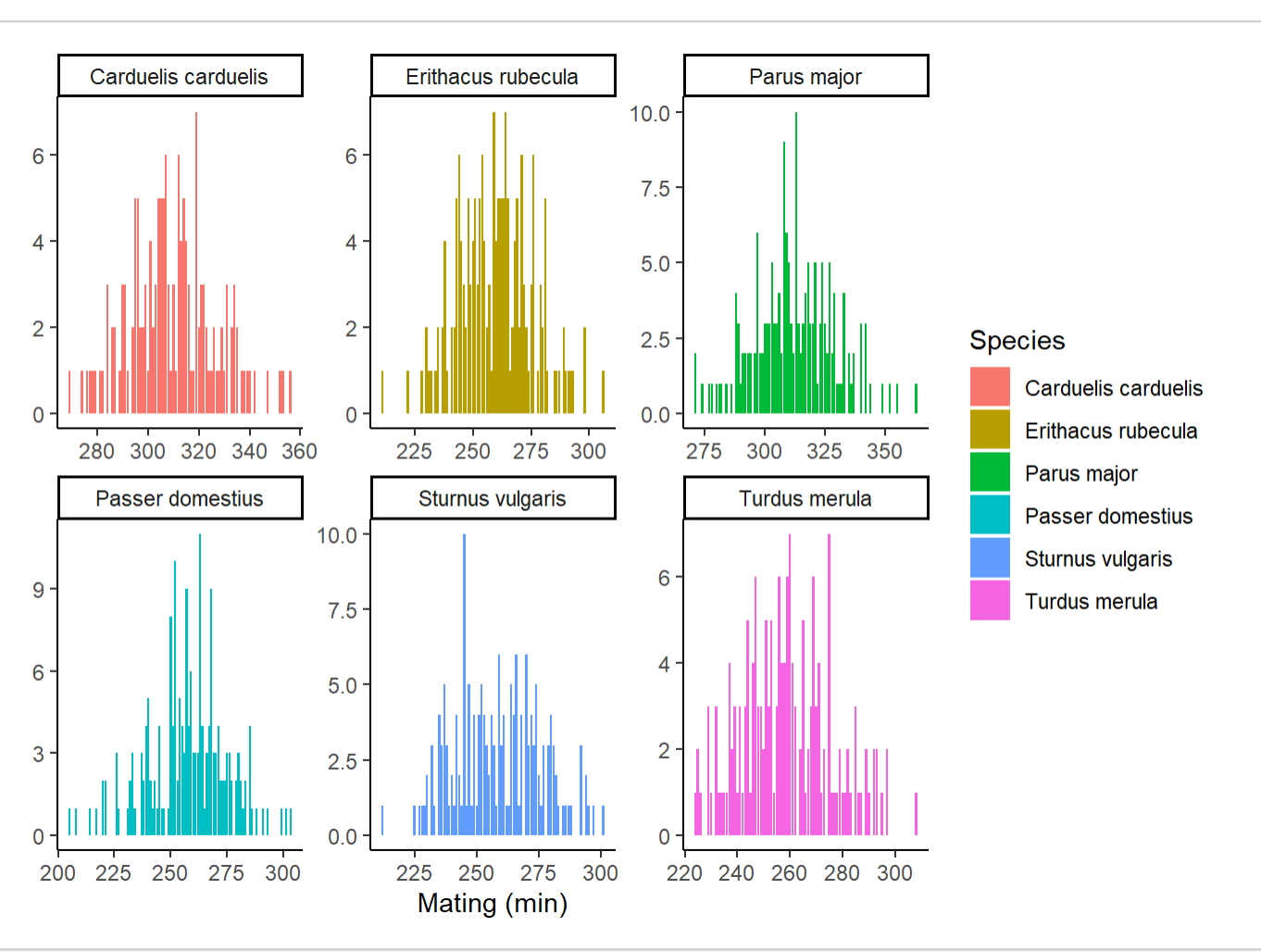

```
ggplot(aes(x=Forage,fill=(Species)), data=MyData)+
  geom_bar(bins=10)+
  labs(x = "Foraging (min)", y="", fill="Species")+
  facet_wrap(vars(Species), scales='free')+
  theme_classic()
```

```
## Warning in geom_bar(bins = 10): Ignoring unknown parameters: `bins`
```



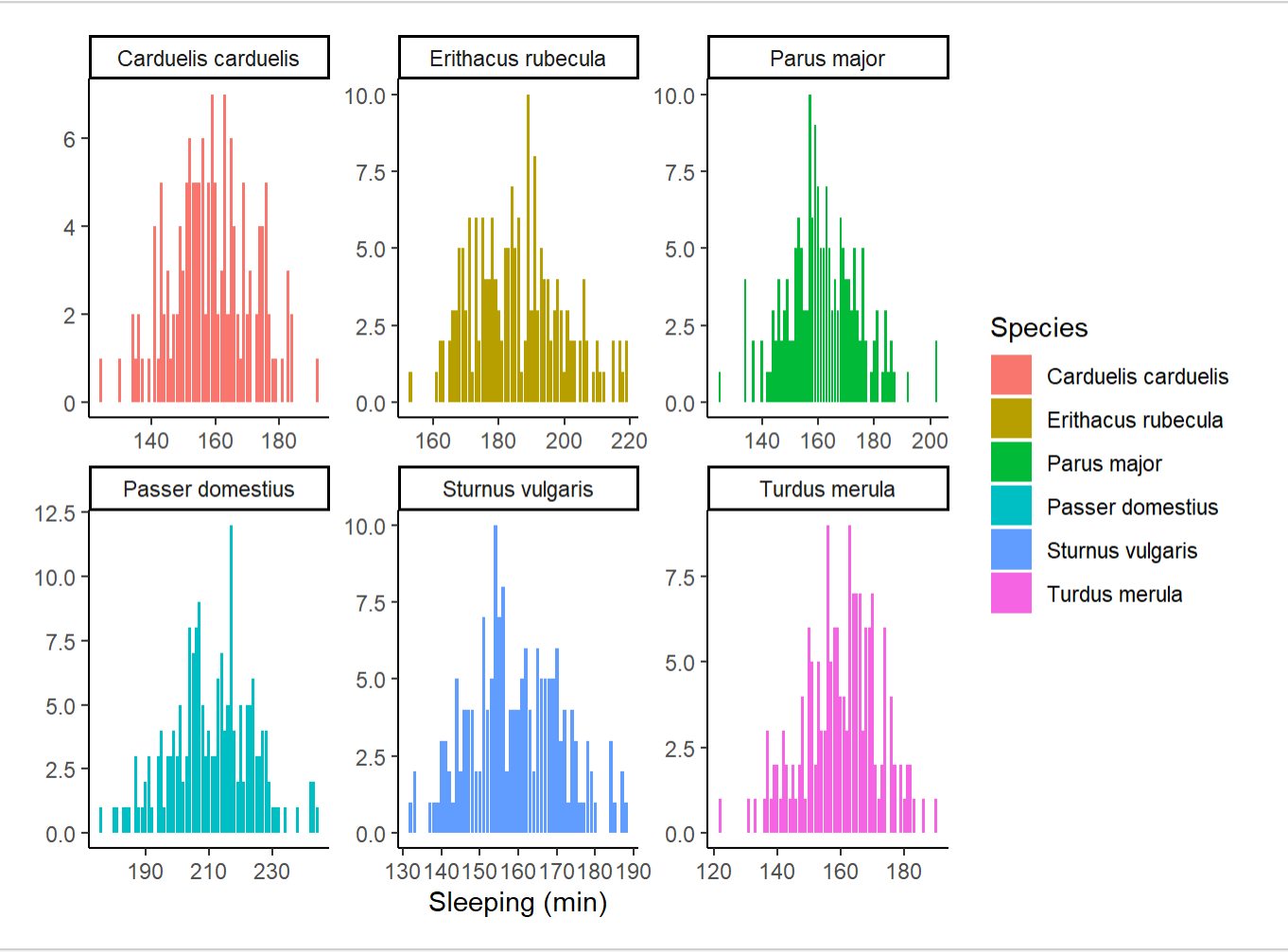

```
ggplot(aes(x=Mate,fill=(Species)), data=MyData)+
  geom_bar(bins=10)+
  labs(x = "Mating (min)", y="", fill="Species")+
  facet_wrap(vars(Species), scales='free')+
  theme_classic()
```

```
## Warning in geom_bar(bins = 10): Ignoring unknown parameters: `bins`
```




```
ggplot(aes(x=Sleep,fill=(Species)), data=MyData)+
  geom_bar(bins=10)+
  labs(x = "Sleeping (min)", y="", fill="Species")+
  facet_wrap(vars(Species), scales='free')+
  theme_classic()
```

```
## Warning in geom_bar(bins = 10): Ignoring unknown parameters: `bins`
```




```
avgData <- MyData %>% group_by(Species) %>%
  summarise(mean_groom=mean(Groom),
            mean_forage= mean(Forage),
            mean_mate=mean(Mate),
            mean_sleep=mean(Sleep))%>%
  as.data.frame()
avgData
```

```
##               Species mean_groom mean_forage mean_mate mean_sleep
## 1 Carduelis carduelis   239.9533    119.5067  309.6000   158.7667
## 2  Erithacus rubecula   264.9106    120.4860  259.6648   185.6257
## 3         Parus major   291.7829    121.7771  312.0686   161.6514
## 4    Passer domestius   237.0481    120.7968  257.5455   210.8770
## 5    Sturnus vulgaris   238.4244    168.9767  258.5814   159.3895
## 6       Turdus merula   288.5917    119.6036  258.6450   159.9941
```

Figures 1-4: Display the distribution of minutes of each behavior between all 6 of the different observed species. The figures in order are the times grooming, foraging, mating, and sleeping respectively. 4.

```
corData <-MyData %>%
  select(3:6)
corData <-round(cor(corData),3)
```

5. The correlation efficient matrix shows the correlation between grooming and mating is 0.481, which indicates that these are somewhat positively correlated. Mating and sleeping are somewhat negatively correlated. The pairs, grooming and foraging, grooming and sleeping, and mating and foraging are weakly negatively correlated. Sleeping and foraging have very little association. The figures above show a large range in each behavior and between species

Part II: PCA 1.

```
scaleData <-scale(corData, center = FALSE, scale = TRUE)
```

Data should be scaled to avoid quantitatively large variables from dominating the analysis. To scale the data by dividing by the standard deviation, the parameter scale needs to equal true. Using this default scaling puts all the variables on the same scale and makes their standard deviations equal to 1.

2.

```
DataPCA <-princomp(scaleData)
```

I only used columns 3 to 6 because those are the four columns that contain values for the different behaviors (grooming, mating, foraging, and sleeping)

3.

```
DataPCA <-princomp(corData, cor=F)
```

Using cor=F indicates that to the function to use the covariation matrix. This does not scale the data. We do not want to scale the data, because we already scaled the variables to the same scale. If the variables are on a different scale, cor should be true to use the correlation matrix.

4.

```
str(DataPCA)
```

```
## List of 7
##  $ sdev    : Named num [1:4] 8.94e-01 4.95e-01 2.55e-01 9.98e-09
##   ..- attr(*, "names")= chr [1:4] "Comp.1" "Comp.2" "Comp.3" "Comp.4"
##  $ loadings: 'loadings' num [1:4, 1:4] 0.562 -0.394 0.559 -0.466 0.114 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "Groom" "Forage" "Mate" "Sleep"
##   .. ..$ : chr [1:4] "Comp.1" "Comp.2" "Comp.3" "Comp.4"
##  $ center  : Named num [1:4] 0.238 0.138 0.242 0.11
##   ..- attr(*, "names")= chr [1:4] "Groom" "Forage" "Mate" "Sleep"
##  $ scale   : Named num [1:4] 1 1 1 1
##   ..- attr(*, "names")= chr [1:4] "Groom" "Forage" "Mate" "Sleep"
##  $ n.obs   : int 4
##  $ scores  : num [1:4, 1:4] 0.895 -0.827 0.891 -0.959 0.125 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "Groom" "Forage" "Mate" "Sleep"
##   .. ..$ : chr [1:4] "Comp.1" "Comp.2" "Comp.3" "Comp.4"
##  $ call    : language princomp(x = corData, cor = F)
##  - attr(*, "class")= chr "princomp"
```

```
head(DataPCA$scores)
```

```
##            Comp.1      Comp.2      Comp.3        Comp.4
## Groom   0.8947940  0.12527209  0.35415142 -2.498002e-16
## Forage -0.8267556 -0.71761033  0.05348038 -3.261280e-16
## Mate    0.8909784 -0.07422657 -0.35870531  4.440892e-16
## Sleep  -0.9590168  0.66656481 -0.04892649  1.110223e-16
```

```
DataPCA$loadings
```

```
##
## Loadings:
##        Comp.1 Comp.2 Comp.3 Comp.4
## Groom   0.562  0.114  0.693  0.437
## Forage -0.394 -0.740  0.102  0.536
## Mate    0.559        -0.707  0.425
## Sleep  -0.466  0.658         0.584
##
##                Comp.1 Comp.2 Comp.3 Comp.4
## SS loadings      1.00   1.00   1.00   1.00
## Proportion Var   0.25   0.25   0.25   0.25
## Cumulative Var   0.25   0.50   0.75   1.00
```

```
DataPCA$sdev
```

```
##       Comp.1       Comp.2       Comp.3       Comp.4
## 8.941109e-01 4.950949e-01 2.546305e-01 9.976989e-09
```

The object that was created displays information on the correlation between different variables. These are emphasised in the scores, loadings, and principle components. Negative values indicate a negative correlation, and positive will indicate a positive correlation.

5.

```
PCLoad<-data.frame(Eigenvector=c(1:4),
                   Eigenvalue=DataPCA$sdev^2)
ggplot(aes(x=Eigenvector,y=Eigenvalue),data=PCLoad)+
  geom_point() + geom_line()
```

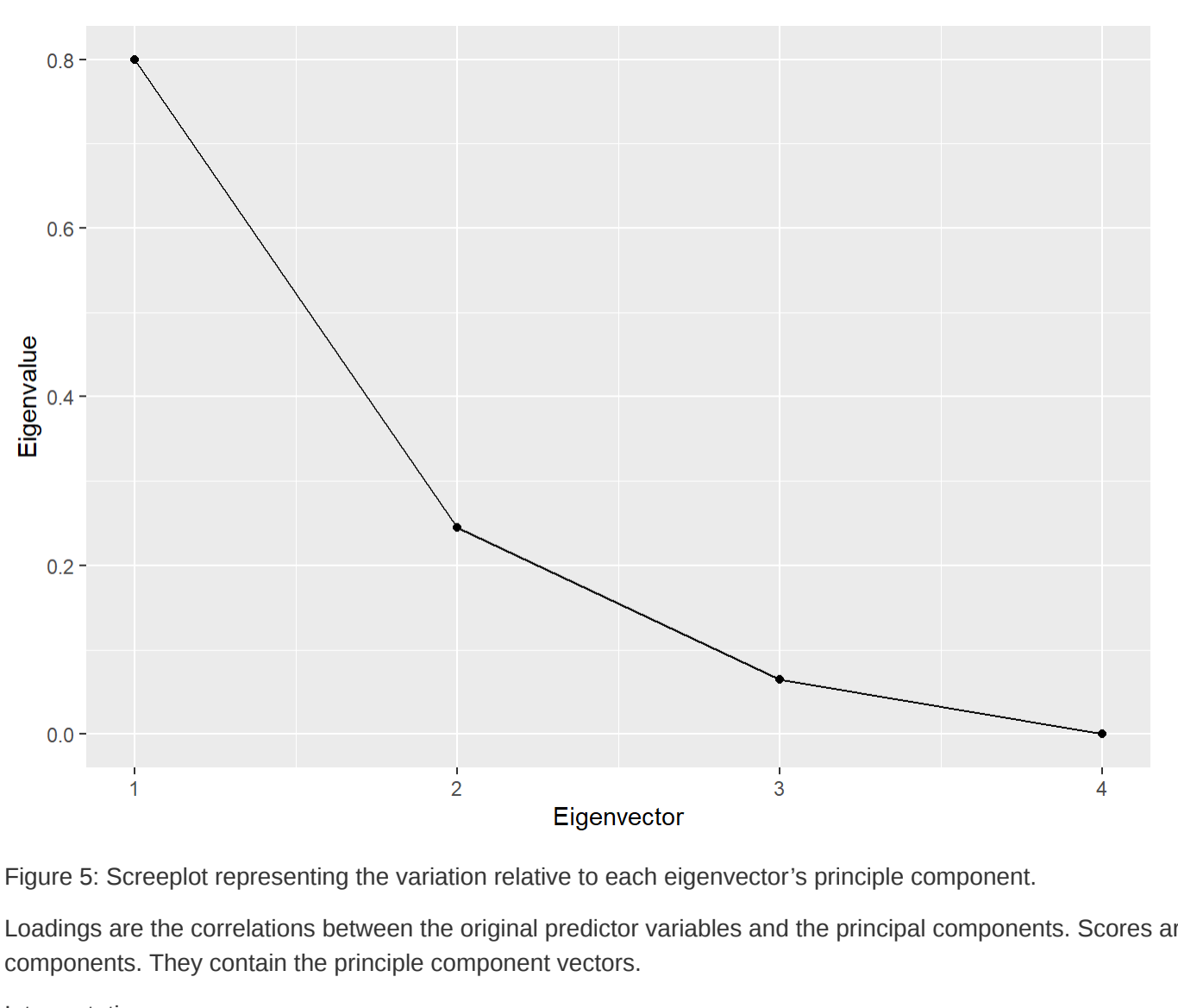


Figure 5: Screeplot representing the variation relative to each eigenvector's principle component.

Loadings are the correlations between the original predictor variables and the principal components. Scores are the calcaulted princple components. They contain the principle component vectors.

Interpretation

1.

```
MyData<-cbind(MyData,DataPCA$scores)
```

```
## Warning in data.frame(..., check.names = FALSE): row names were found from a
## short variable and have been discarded
```

2. Create bivariate plots for PC1 vs PC2 and another plot for PC3 vs PC4. From these four principal components, choose the two 'best' PCs for a bivariate plot and add it to your R markdown file. The criteria for 'best' depends on the question. In this case, let's focus on the question How do species differ in their behavior? Choose the two axes that are 'best' in their ability to identify differences among species.

```
pData<-cbind(corData,DataPCA$scores)
ggplot(aes(x=Comp.1,y=Comp.2),data=pData) +
  geom_point()+
  geom_jitter()
```

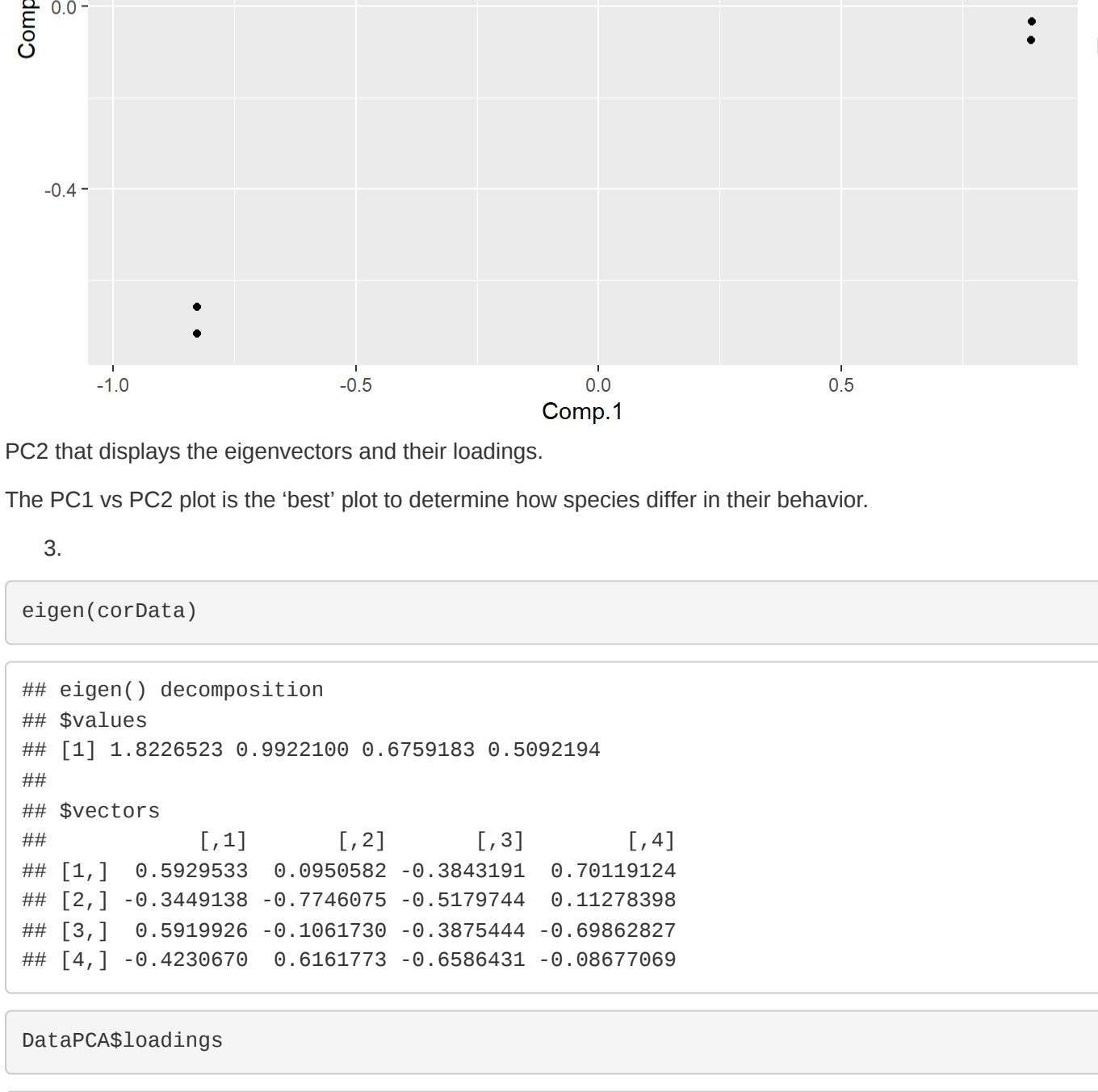


Figure 6: Bivariate plots for PC1 vs PC2 that displays the eigenvectors and their loadings.

The PC1 vs PC2 plot is the 'best' plot to determine how species differ in their behavior.

3.

```
eigen(corData)
```

```
## eigen() decomposition
## $values
## [1] 1.8226523 0.9922100 0.6759183 0.5092194
##
## $vectors
##            [,1]       [,2]       [,3]        [,4]
## [1,]  0.5929533  0.0950582 -0.3843191  0.70119124
## [2,] -0.3449138 -0.7746075 -0.5179744  0.11278398
## [3,]  0.5919926 -0.1061730 -0.3875444 -0.69862827
## [4,] -0.4230670  0.6161773 -0.6586431 -0.08677069
```

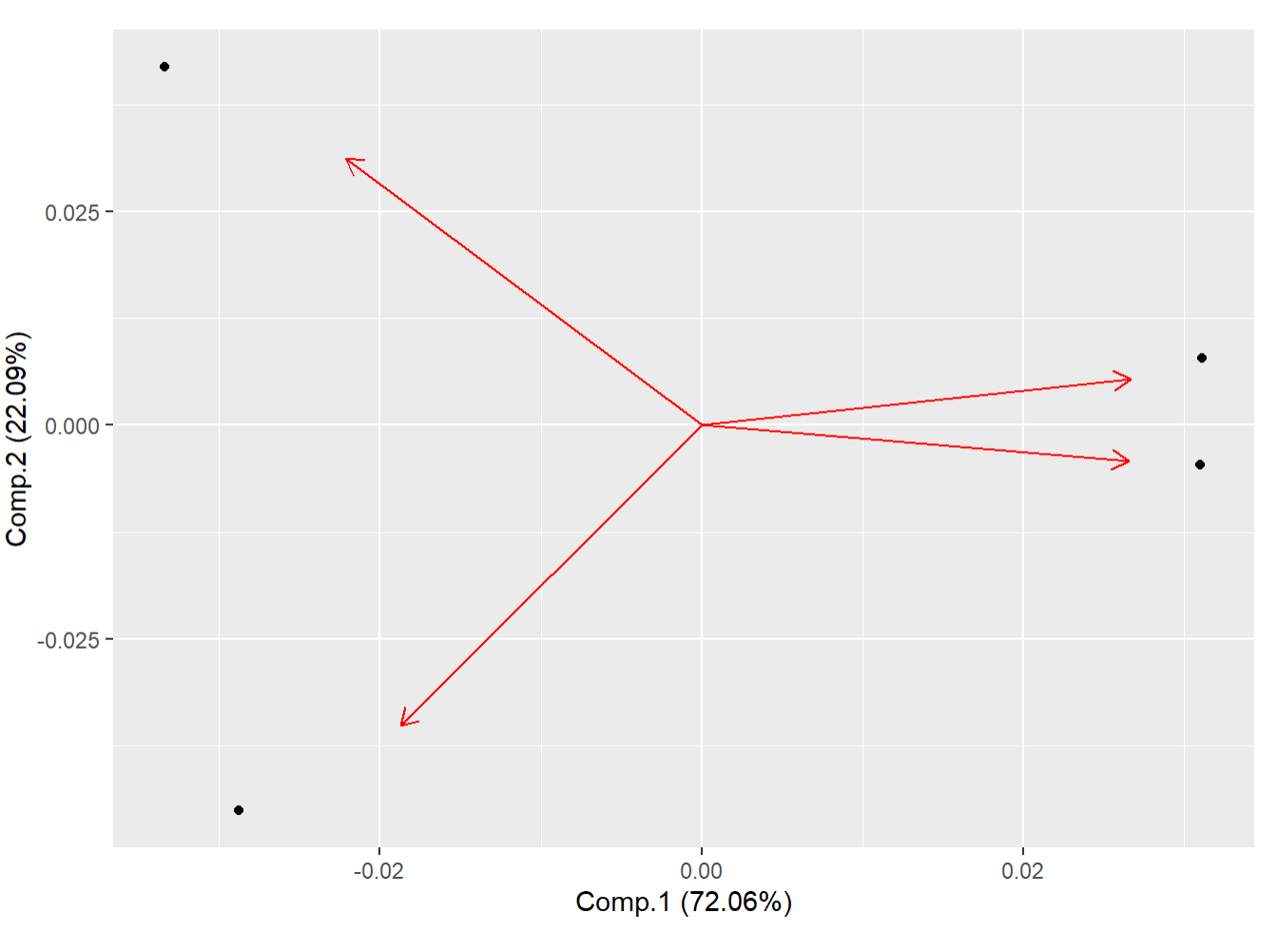
```
DataPCA$loadings
```

```
##
## Loadings:
##        Comp.1 Comp.2 Comp.3 Comp.4
## Groom   0.562  0.114  0.693  0.437
## Forage -0.394 -0.740  0.102  0.536
## Mate    0.559        -0.707  0.425
## Sleep  -0.466  0.658         0.584
##
##                Comp.1 Comp.2 Comp.3 Comp.4
## SS loadings      1.00   1.00   1.00   1.00
## Proportion Var   0.25   0.25   0.25   0.25
## Cumulative Var   0.25   0.50   0.75   1.00
```

```
autoplot(DataPCA,data=MyData,
        loadings=T,loadings.labels=T)
```

```
## Warning in data.frame(..., check.names = FALSE): row names were found from a
## short variable and have been discarded

## Warning in data.frame(..., check.names = FALSE): row names were found from a
## short variable and have been discarded
```



4. Eigenvector loadings indicate the strength and polarity of the correlation. A large number indicates a strong relationship to the principle component. The polarity determines whether the relationship is positive or negativly coorelated. The loadings tell us that Mating and grooming are positively coorelated and foraging and sleeping have little to no correlation.