Exploratory Data Analysis of Leptograpsus variegatus data

Kim Reijntjens

14-9-2021

Relevance of the project

Electrophoretic study established the distinctness of the rock crabs from the genus Leptograpsus. in the blue and orange colours given the species name L. variegatus. now that we know that there are two colour forms in the species we can use the morphological data to examine the two species and develop objective criterea for identification of sex and colour.

goal

Is it possible to predict the sex and species colour of a Leptograpsus variegatus, found at the waters of fremantle, W. Australia, based on morphological measurements.

The data was from an article of the Australian Journal of Zoology in January 1974. The study "A multivariate study of variation in two species of rock crab of genus Leptograpsus" by N. A. Campbell and R. J. Mahon. The dataset has 200 rows and 8 columns, describing 5 morphological measurements on 50 crabs each of two colour forms and both sexes, of the species Leptograpsus variegatus collected at Fremantle, W. Australia. Measurements taken were: (1) Frontal lobe size (mm) (FL); (2) Rear width (mm) (RW); (3) Carapace length (mm) along the midline (CL); (4) the maximum width of the carapace (mm) (CW); (5) Body depth (mm) (BD). https://www.researchgate.net/publication/243766527_A_multivariate_study_of_variation_in_two_species_of_rock_crab_of_genus_Leptograpsus

exploration of the data

```r
library(knitr)
crab_data <- read.csv(file = "C:/Users/kimre/Documents/thema-09/archive/data.csv")
head(crab_data)
```

```
##   sp sex index   FL  RW   CL   CW  BD
## 1  B   M     1  8.1 6.7 16.1 19.0 7.0
## 2  B   M     2  8.8 7.7 18.1 20.8 7.4
## 3  B   M     3  9.2 7.8 19.0 22.4 7.7
## 4  B   M     4  9.6 7.9 20.1 23.1 8.2
## 5  B   M     5  9.8 8.0 20.3 23.0 8.2
## 6  B   M     6 10.8 9.0 23.0 26.5 9.8
```

```r
knitr::kable(summary(crab_data))
```

sp	sex	index	FL	RW	CL	CW	BD
B:100	F:100	Min. : 1.0	Min. : 7.20	Min. : 6.50	Min. :14.70	Min. :17.10	Min. : 6.10
O:100	M:100	1st Qu.:13.0	1st Qu.:12.90	1st Qu.:11.00	1st Qu.:27.27	1st Qu.:31.50	1st Qu.:11.40
NA	NA	Median :25.5	Median :15.55	Median :12.80	Median :32.10	Median :36.80	Median :13.90
NA	NA	Mean :25.5	Mean :15.58	Mean :12.74	Mean :32.11	Mean :36.41	Mean :14.03
NA	NA	3rd Qu.:38.0	3rd Qu.:18.05	3rd Qu.:14.30	3rd Qu.:37.23	3rd Qu.:42.00	3rd Qu.:16.60
NA	NA	Max. :50.0	Max. :23.10	Max. :20.20	Max. :47.60	Max. :54.60	Max. :21.60

```r
str(crab_data)
```

```
## 'data.frame':    200 obs. of  8 variables:
##  $ sp   : Factor w/ 2 levels "B","O": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sex  : Factor w/ 2 levels "F","M": 2 2 2 2 2 2 2 2 2 2 ...
##  $ index: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ FL   : num  8.1 8.8 9.2 9.6 9.8 10.8 11.1 11.6 11.8 11.8 ...
##  $ RW   : num  6.7 7.7 7.8 7.9 8 9 9.9 9.1 9.6 10.5 ...
##  $ CL   : num  16.1 18.1 19 20.1 20.3 23 23.8 24.5 24.2 25.2 ...
##  $ CW   : num  19 20.8 22.4 23.1 23 26.5 27.1 28.4 27.8 29.3 ...
##  $ BD   : num  7 7.4 7.7 8.2 8.2 9.8 9.8 10.4 9.7 10.3 ...
```

We created a own codebook with a description per column. The details for the description where present on the kaggle but not in a codebook format.

```r
code_book <- read.table(file = "C:/Users/kimre/Documents/thema-09/archive/codebook.txt", sep = ";", head
kable(code_book, caption = "A codebook for the data ")
```

Table 2: A codebook for the data

column	description	type
sp	species B= BLUE O=ORANGE	factor
sex	M= MALE F=FEMALE	factor
FL	Frontal lobe size (mm)	numeric
RW	Rear width (mm)	numeric
CL	Carapace length (mm)	numeric
CW	Carapace width (mm)	numeric
BD	Body depth (mm)	numeric

We see in the summary of crab_data that the column index has a maximum of 50 as a unique row identifier. this is because it counts from 1-50 for a blue male then 1-50 for a blue female and the same for the orange species.

The species and sex column abbreviations for orange and blue, and male and female. we changed this for the full name for a better readability.

```r
#check for NA falues
kable(apply(crab_data, 2, function(x) any(is.na(x))), caption = "Table to show per column whether there
```

Table 3: Table to show per column whether there are missing values
FALSE= no missing values found / TRUE = missing values found

	x
sp	FALSE
sex	FALSE
index	FALSE
FL	FALSE
RW	FALSE
CL	FALSE
CW	FALSE
BD	FALSE

```
library(dplyr)
```

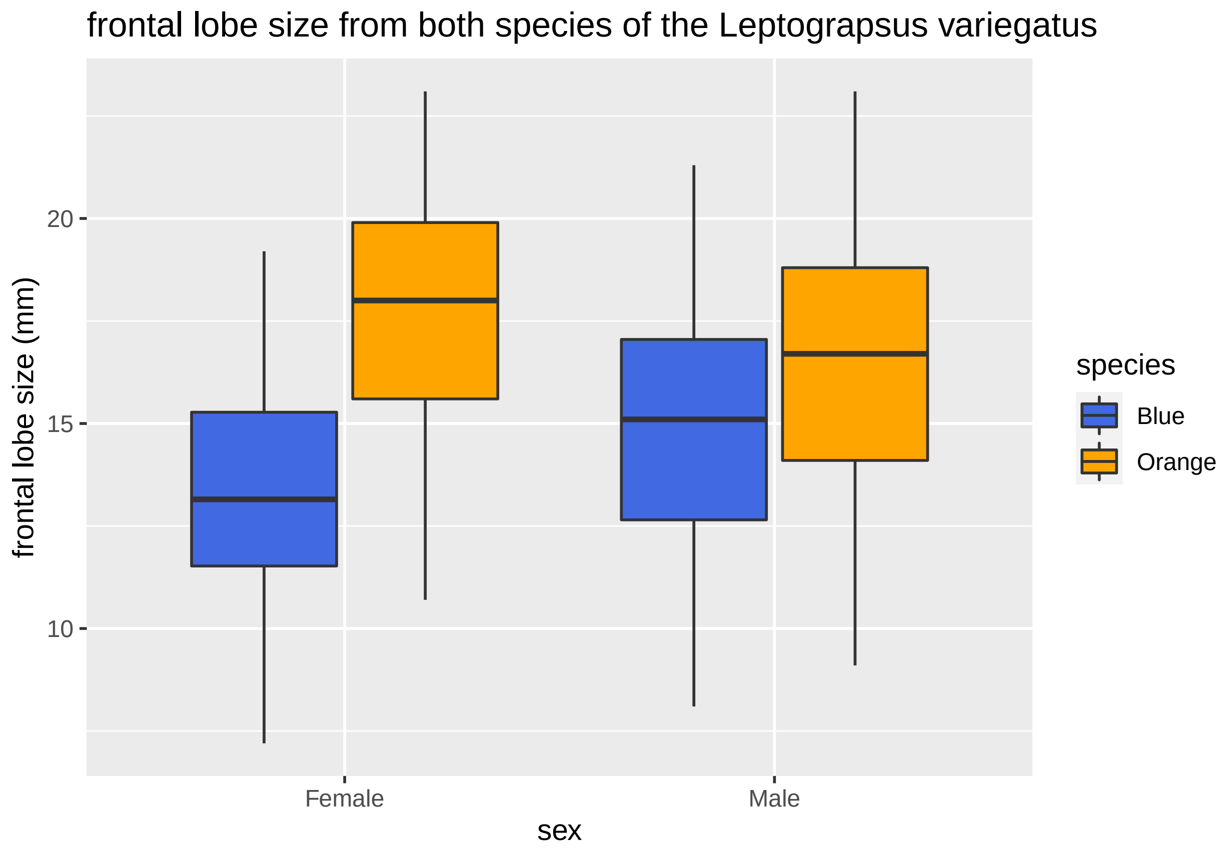
```
##
## Attaching package: 'dplyr'

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

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

```
crab_data <- crab_data %>% mutate(sp=recode(sp,
                         `B`="Blue",
                         `O`="Orange"),
                     sex=recode(sex,
                         `M`="Male",
                         `F`="Female"))


colnames(crab_data)<- c("species", "sex", "index","Frontal lobe", "Rear width", "Carapace length", "Cara

head(crab_data)
```

```
##   species  sex index Frontal lobe Rear width Carapace length Carapace width
## 1    Blue Male     1          8.1        6.7            16.1           19.0
## 2    Blue Male     2          8.8        7.7            18.1           20.8
## 3    Blue Male     3          9.2        7.8            19.0           22.4
## 4    Blue Male     4          9.6        7.9            20.1           23.1
## 5    Blue Male     5          9.8        8.0            20.3           23.0
## 6    Blue Male     6         10.8        9.0            23.0           26.5
##   Body depth
## 1        7.0
## 2        7.4
## 3        7.7
## 4        8.2
## 5        8.2
## 6        9.8
```

Lets start to explore the data The first measurement of the crab, the frontal lobe size.

```
library(ggplot2)


ggplot(crab_data, aes(x=sex, y=`Frontal lobe`, fill=species)) +
  geom_boxplot( )  + scale_fill_manual(values=c("royalblue", "orange")) +
  ggtitle("frontal lobe size from both species of the Leptograpsus variegatus")+
  ylab("frontal lobe size (mm)")
```



If we compare the frontal lobe size of both species as we did in this figure, we see that for both genders (male and female) that the oragne specis has a bigger frontal lobe size.

To get a quick overview of the most useful graphs with this data we use a ggpairs plot.

```
library(GGally)
```

```
##
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':
##
##     nasa
```

```r
ggpairs(crab_data, aes(fill = species) ,lower = list(continuous = wrap("points", alpha = 0.3,    size=0
              )) + scale_fill_manual(values=c("royalblue", "orange")) + scale_color_manual(values=c("Ro
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

/ /

```r
ggpairs(crab_data, mapping = aes(color = species),lower = list(continuous = wrap("points", alpha = 0.6,
```

As you can see the boxplot form the last plot is also shown in the ggpairs figure. Now you can see clearly that the orange species had bigger size in all the measurements compared to the blue species. Also is the female of the blue species in all measurements a slightly bit bigger than the male. This is not the same for the orange species.

In the histograms you can already see a little bit of the distribution from the data. We'll take a closer look by using a density plot. this uses the same concept as a histogram, but in a smoothed version. A bonus on using density plots is that it is not affected by the number of bins.

```r
library(tidyverse)
```

```
## -- Attaching packages ---- tidyverse 1.3.0 --

## v tibble  3.0.0     v purrr   0.3.3
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
long_data <- pivot_longer(data = crab_data, cols = 4:8, names_to = "body_part", values_to = "size")

long_data %>%  ggplot(aes(x = size,  colour = species)) +
    geom_density(show.legend = TRUE) +
   facet_wrap(~body_part + sex, ncol = 5) +    scale_color_manual(values=c("Royalblue", "orange"))
```

The peaks of a Density Plot help display where values are concentrated over the interval. We can already see that the gender female will be of good use for machine learning because is does not overlap as much as the other plots do. If we look for patterns in the data. We want to use a Principal component analysis (PCA).

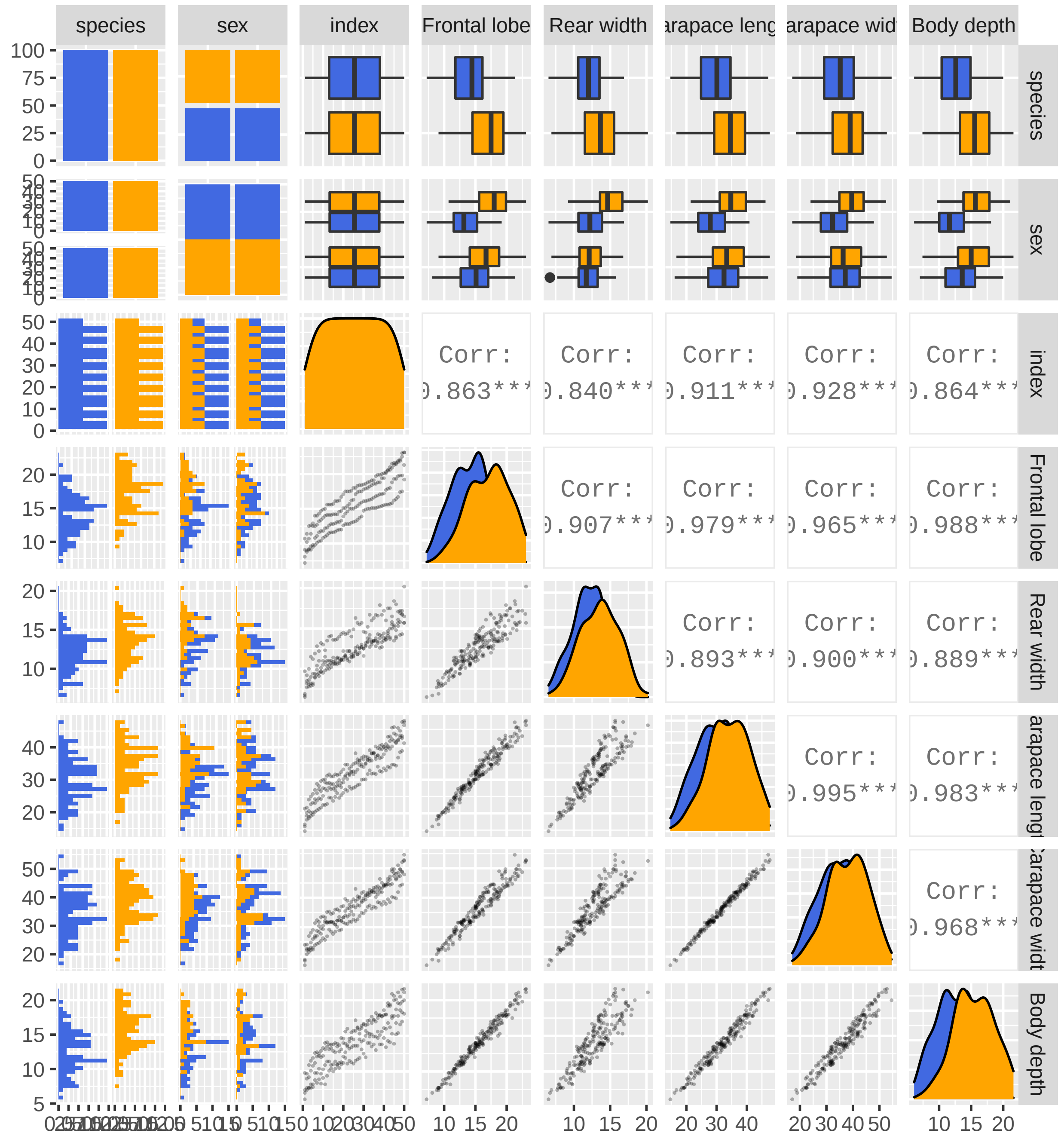


Figure 1: Title: pairs plot of the data: on the X and Y axis the colums of the Leptograpsus variegatus data. Orange colour is the orange species/blue colour is the blue species

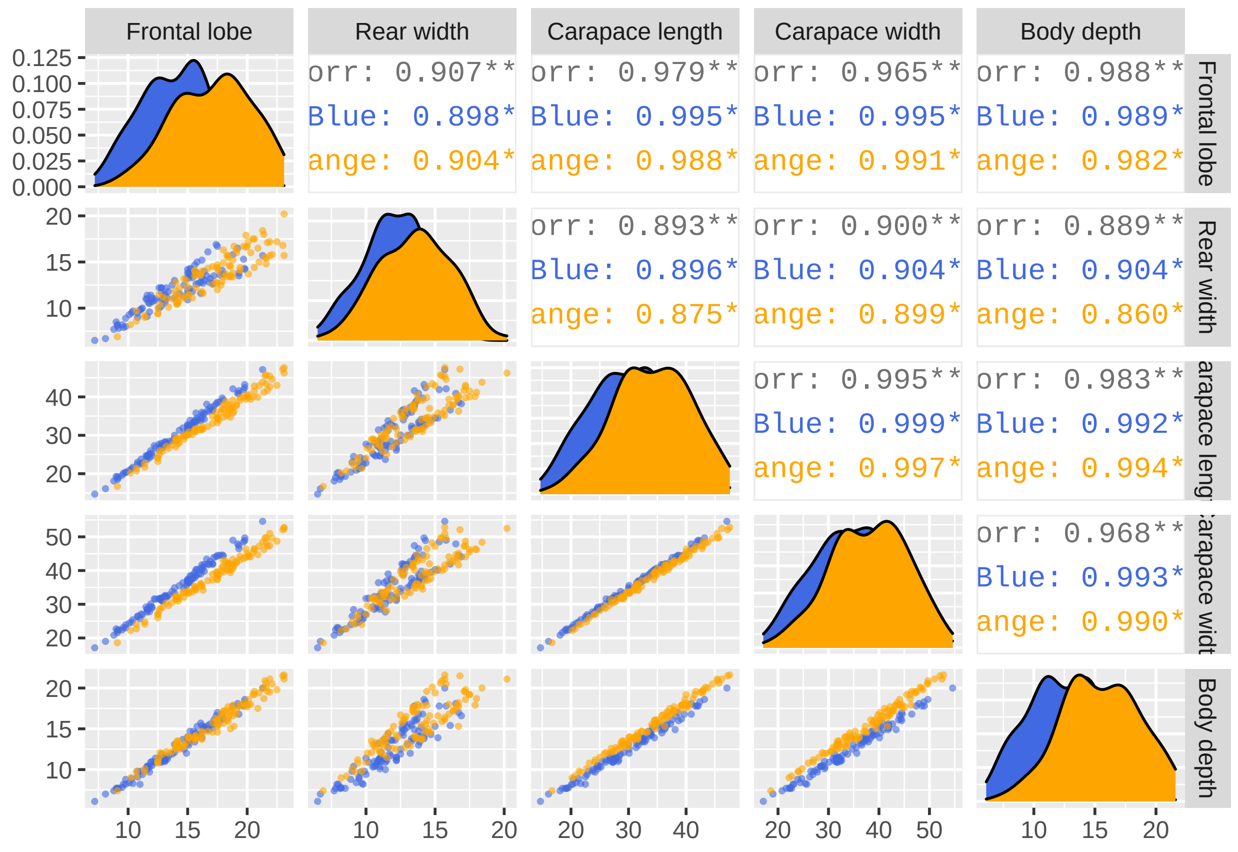


Figure 2: Title: Pairs plot of the lower half.: selection of the pairs plot in figure 1. Orange colour is the orange species/blue colour is the blue species

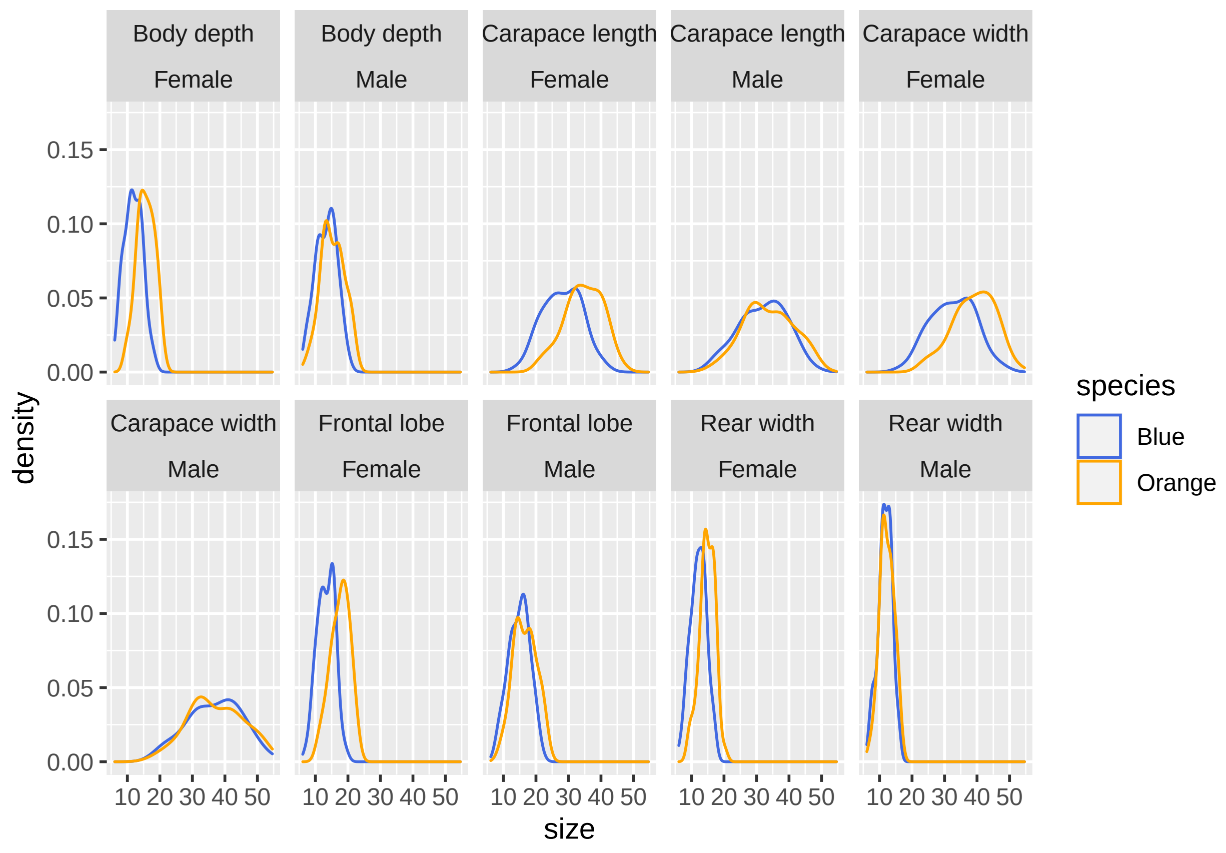


Figure 3: Title: density plot of the leptograpsus variegatus body measurements.: density plot of the body measurements with the size in milimeters in the x-axis and density on the Y-axis

```r
w_density <- crab_data[c(1,4:8)]  # columns you want densities for
w_density$species <- w_density$species[101:200]  # maybe you have a variable to group by
```

```r
w_density %>%
    pivot_longer(!species, names_to = "variable", values_to = "value") %>%
    ggplot(aes(x = value, colour = species)) +
    geom_density(show.legend = TRUE) +
    facet_wrap(~variable, ncol = 5)
```

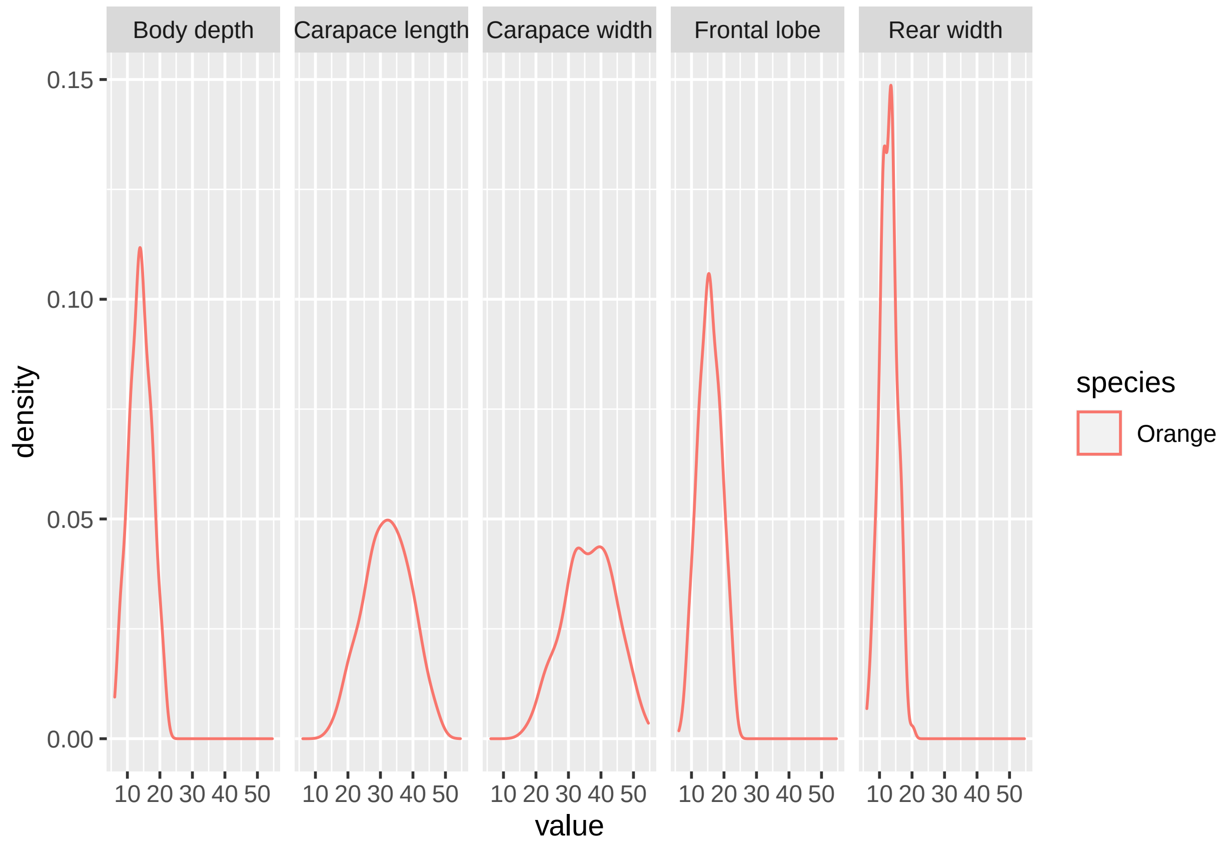
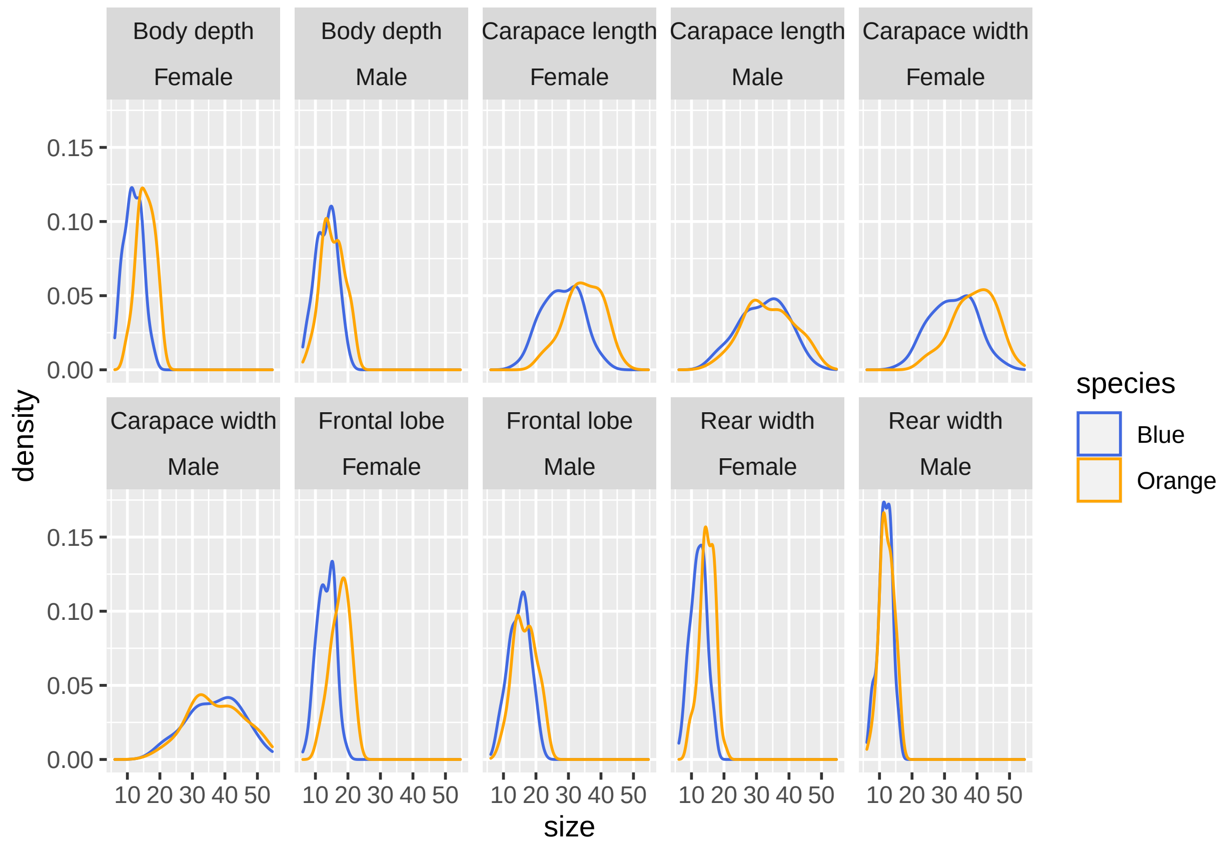


Figure 4: Title: density plot of the orange species measurements.

```r
#head(crab_data)

long_data <- pivot_longer(data = crab_data, cols = 4:8, names_to = "body_part", values_to = "size")

#head(long_data)
```

/ /

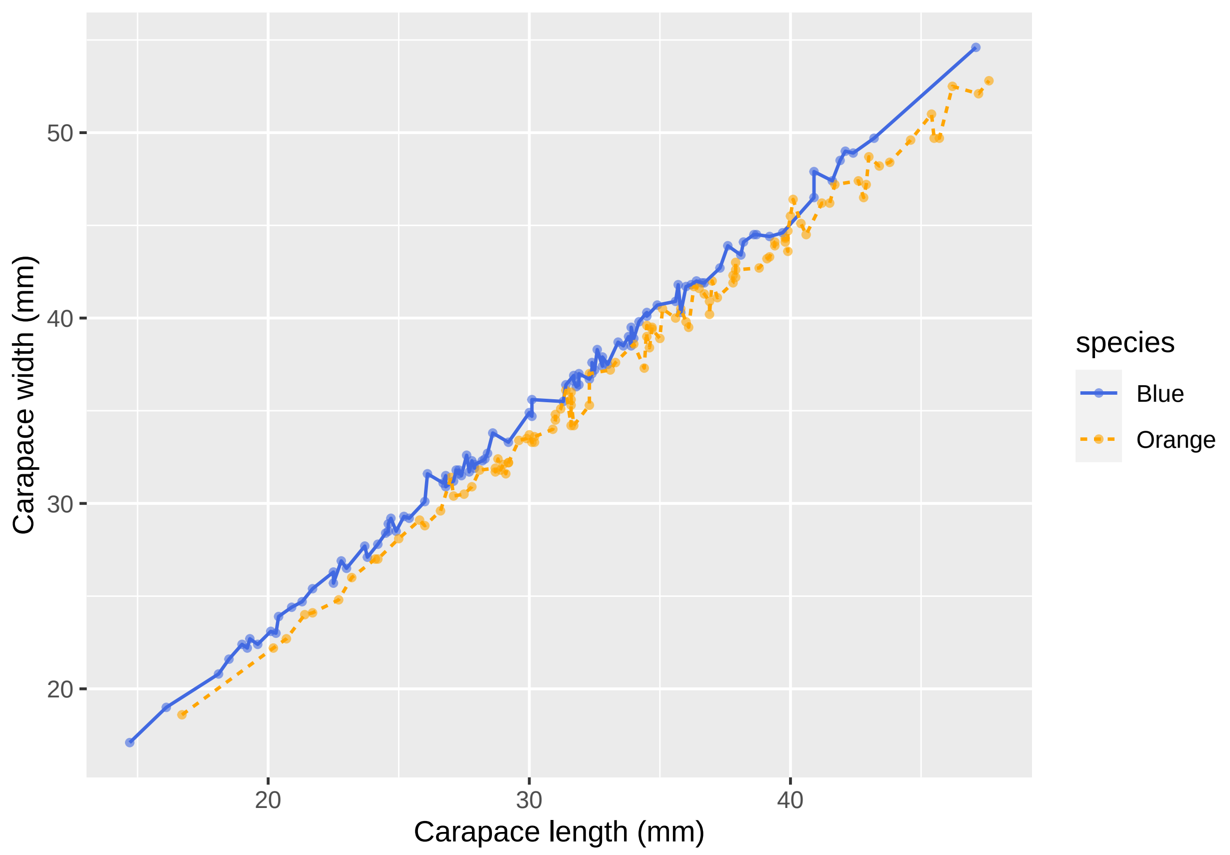
```r
long_data %>%  ggplot(aes(x = size,  colour = species)) +
    geom_density(show.legend = TRUE) +
    facet_wrap(~body_part + sex, ncol = 5) +scale_color_manual(values=c("Royalblue", "orange"))
```



/ /

First we see the 5 columns of each species, and then both species colors. But this time also devided by gender. The peaks of a Density Plot helps display where values are concentrated over the interval.

```
ggplot(crab_data, aes(x=‘Carapace length‘, y=‘Carapace width‘, colour = species)) +
  geom_point(size=1.0,alpha=0.6) +
  geom_line(aes(linetype=species), size = 0.6) + scale_color_manual(values=c("Royalblue", "orange")) +
  xlab("Carapace length (mm)") + ylab("Carapace width (mm)")
```

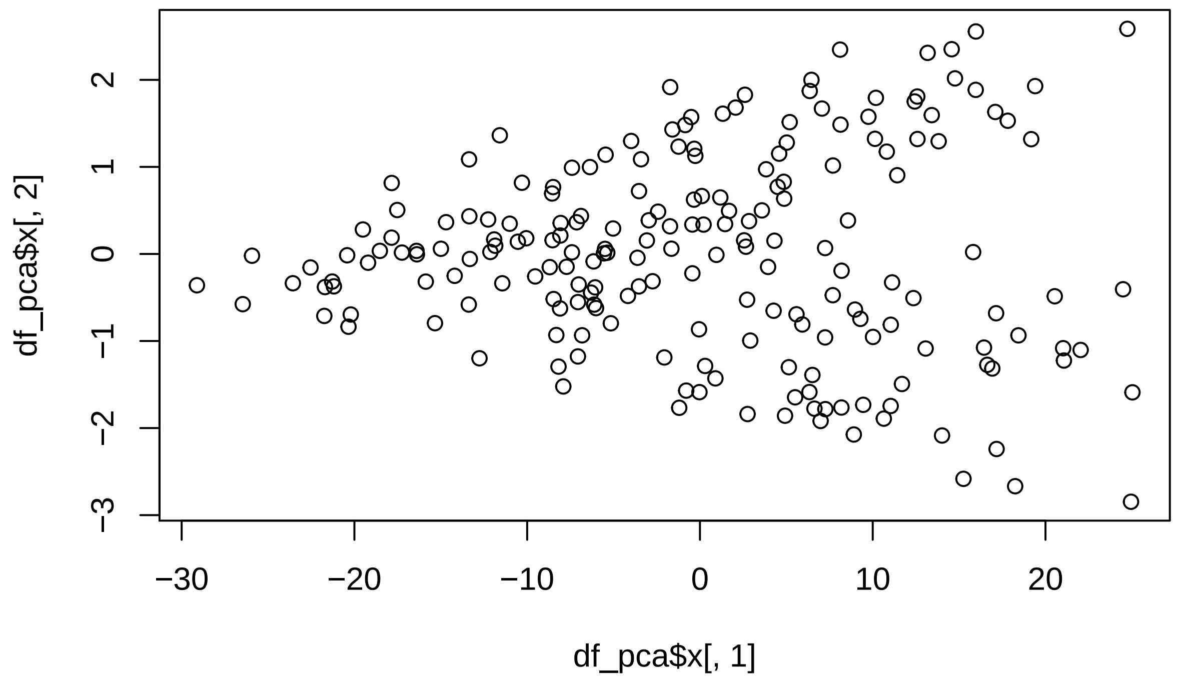


lets now try and find patterns in the data. for this we want to use a Principal component analysis (PCA).

```
#Dimension conversion for futher analysis:
#PCA analysis using prcomp

df <- subset(crab_data, select = -c(2,3) )
row.names(df) <- paste(df$species, row.names(df), sep="_")
df$species <- NULL

head(df)
```

```
##         Frontal lobe Rear width Carapace length Carapace width Body depth
## Blue_1           8.1        6.7            16.1           19.0        7.0
## Blue_2           8.8        7.7            18.1           20.8        7.4
## Blue_3           9.2        7.8            19.0           22.4        7.7
## Blue_4           9.6        7.9            20.1           23.1        8.2
## Blue_5           9.8        8.0            20.3           23.0        8.2
## Blue_6          10.8        9.0            23.0           26.5        9.8
```

```
df_pca <- prcomp(df)

plot(df_pca$x[,1], df_pca$x[,2])
```



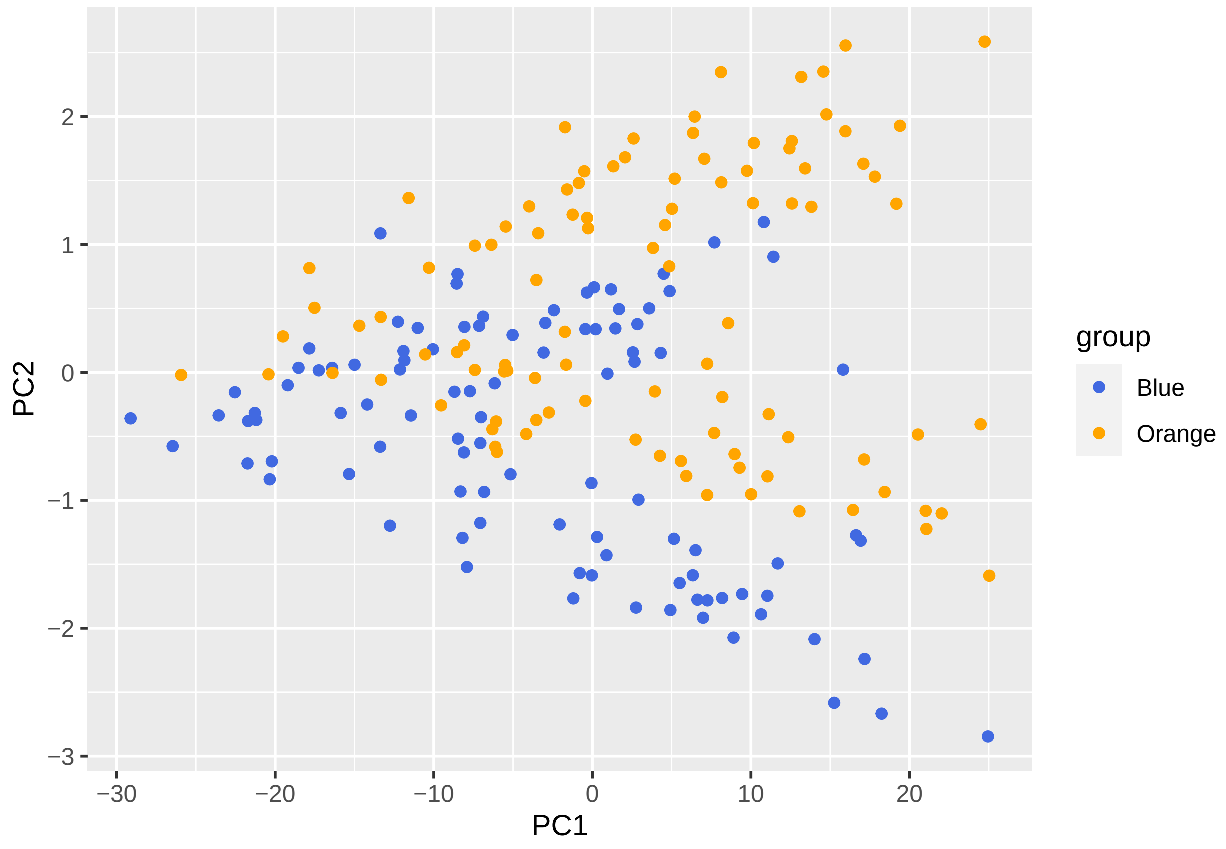
```
df_out <- as.data.frame(df_pca$x)
df_out$group <- sapply( strsplit(as.character(row.names(df)), "_"), "[[", 1 )
head(df_out)
```

```
##               PC1        PC2         PC3         PC4          PC5 group
## Blue_1 -26.46457 -0.5765335 -0.61156772 -0.02868117  0.496584518  Blue
## Blue_2 -23.56174 -0.3364196 -0.23738771  0.02220942 -0.016520723  Blue
## Blue_3 -21.74319 -0.7118646  0.06549720  0.18255551  0.237405035  Blue
## Blue_4 -20.34351 -0.8358049 -0.21829826  0.07424426  0.006636867  Blue
## Blue_5 -20.21227 -0.6955307 -0.36252218  0.16498985 -0.177928180  Blue
## Blue_6 -15.33787 -0.7949508 -0.08414418 -0.21201913  0.154763677  Blue
```

```
library(ggplot2)
library(grid)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

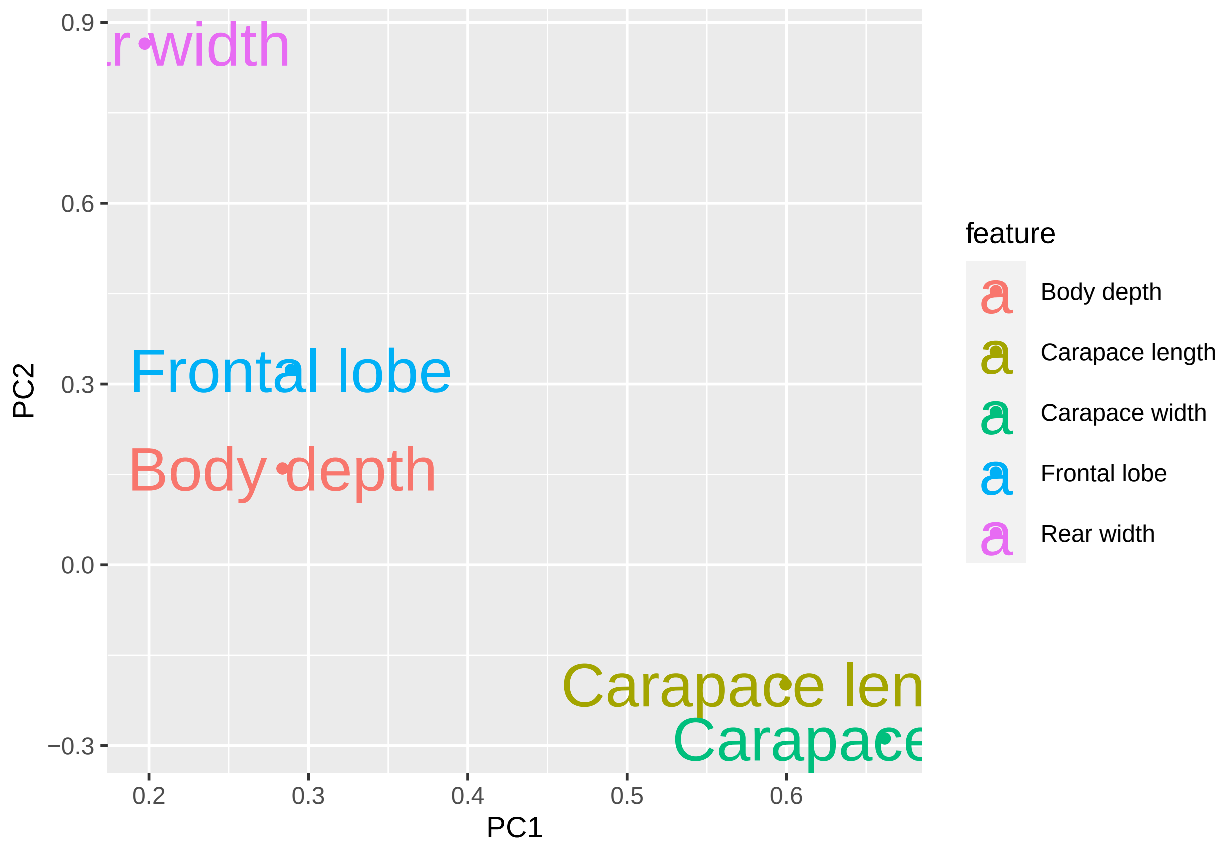
```r
p<-ggplot(df_out,aes(x=PC1,y=PC2,color=group ))
p<-p+geom_point() +    scale_color_manual(values=c("Royalblue", "orange"))
p
```



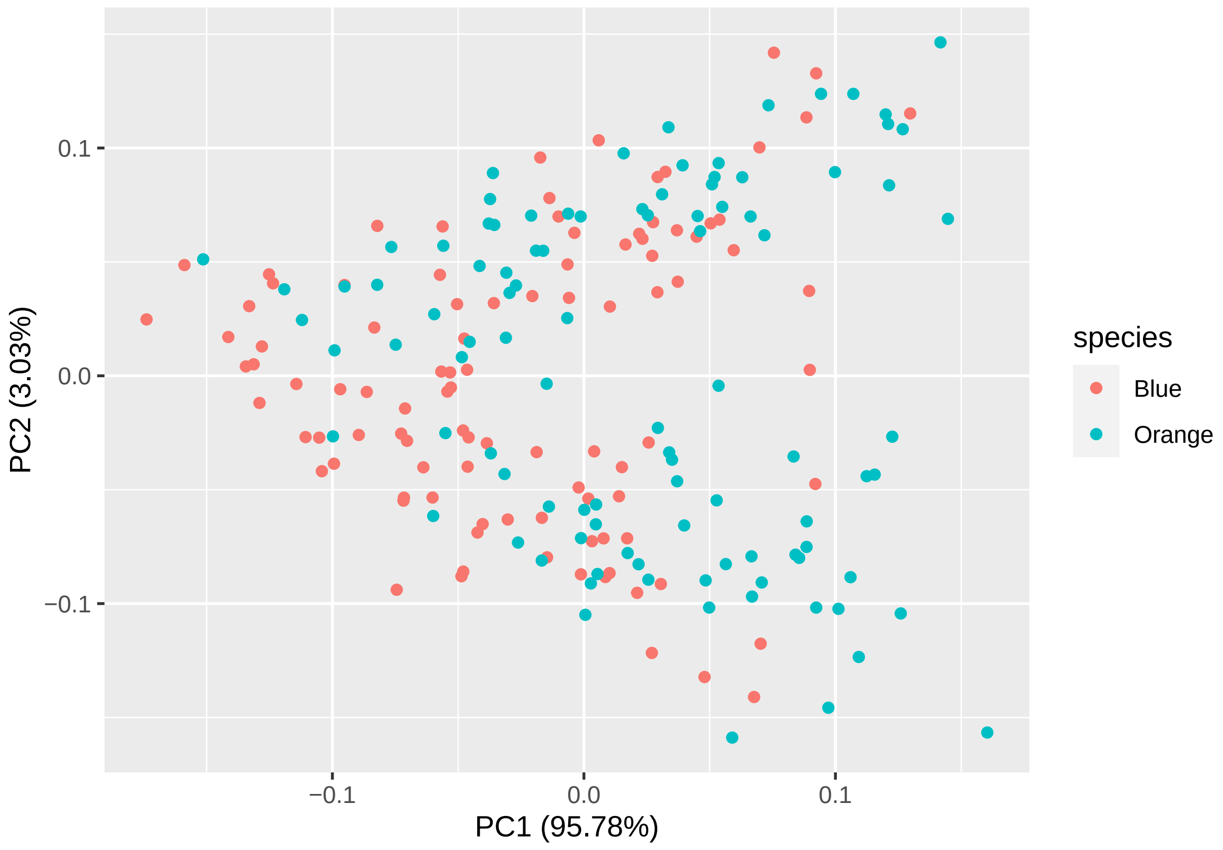
```r
#Plot features that contribute to the classification

df_out_r <- as.data.frame(df_pca$rotation)
df_out_r$feature <- row.names(df_out_r)

df_out_r
```

```
##                       PC1        PC2        PC3        PC4        PC5
## Frontal lobe    0.2889810  0.3232500 -0.5071698  0.7342907  0.1248816
## Rear width      0.1972824  0.8647159  0.4141356 -0.1483092 -0.1408623
## Carapace length 0.5993986 -0.1982263 -0.1753299 -0.1435941 -0.7416656
## Carapace width  0.6616550 -0.2879790  0.4913755  0.1256282  0.4712202
## Body depth      0.2837317  0.1598447 -0.5468821 -0.6343657  0.4386868
##                         feature
## Frontal lobe       Frontal lobe
## Rear width           Rear width
## Carapace length Carapace length
## Carapace width   Carapace width
## Body depth           Body depth
```

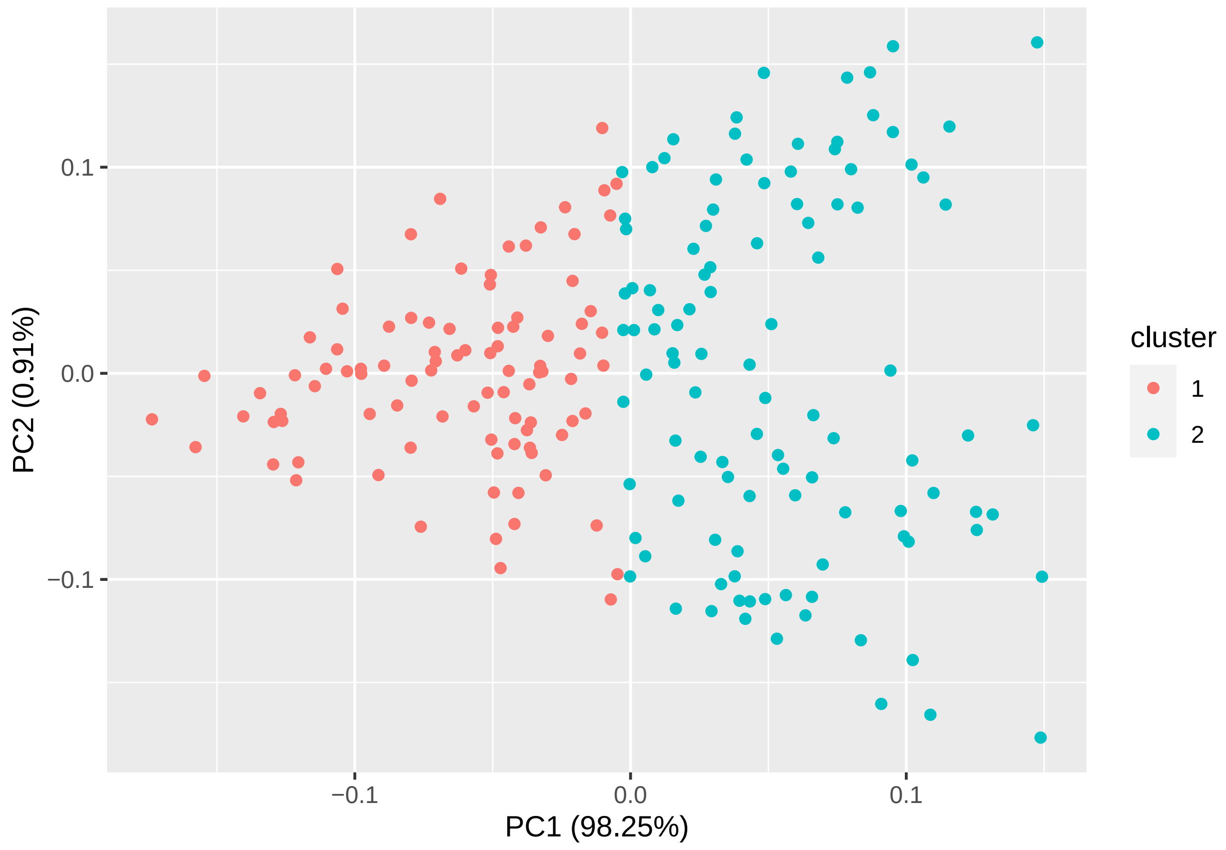
```r
p<-ggplot(df_out_r,aes(x=PC1,y=PC2,label=feature,color=feature ))
p<-p+geom_point() + geom_text(size=8)
p
```



```r
library(ggfortify)
df <- crab_data[4:8]
pca_res <- prcomp(df, scale. = TRUE)

autoplot(pca_res, data = crab_data, colour = 'species')
```



```
set.seed(1)
autoplot(kmeans(df, 2), data = df)
```



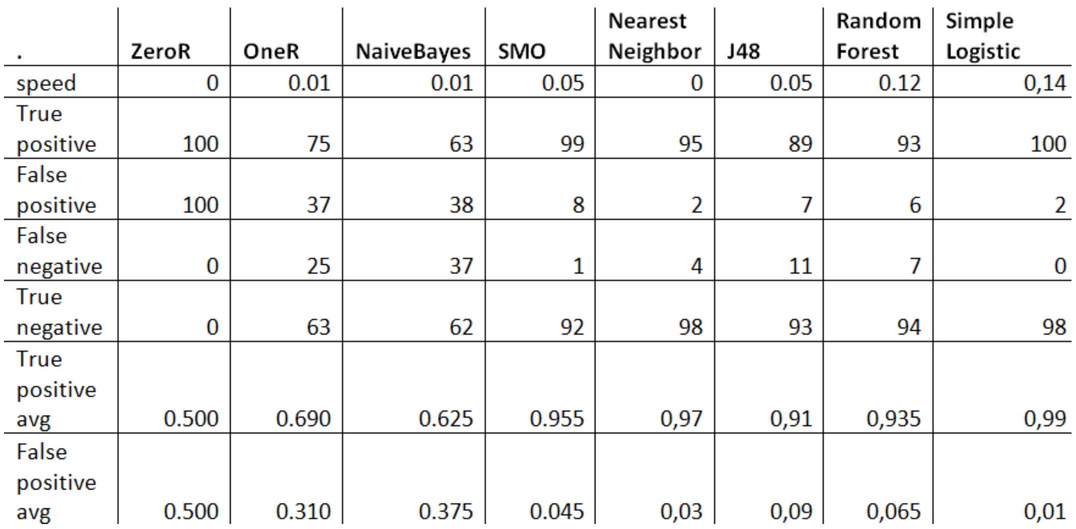
From this point we will start using the data for classifier evalution in Weka. The data is used for training of a model algorithm that will eventually tell us with what species colour of gender we are dealing with, if we will provide it a new instance.

```
crab_data <- crab_data%>%select(-species,everything())

write.csv(crab_data,"C:/Users/kimre/Documents/thema-09/archive/data1.csv", row.names = FALSE)
```

The classifier values is relocated to the last column of the dataset. This because weka expects as default to have the classifier value in the last column.

After exploring the dataset we start investigating the standard algorithms that Weka provides. In the table below you can see the perfomance of these algorithms. You can see that they all have a high true positive avg, this super accurate performance is due to the clean dataset. Only ZeroR scores a 50 percent accurary with 100 out of 200 instances classified true positive and 100 out of 200 classified false positive. This is because the algorithm ZeroR only looks at our species attribute and chooses the most common class value which in our species attribute is 50 percent 'blue species' and 50 percent "orange species".

```
knitr::include_graphics("C:/Users/kimre/Documents/thema-09/algorithmen.png")
```



.	ZeroR	OneR	NaiveBayes	SMO	Nearest Neighbor	J48	Random Forest	Simple Logistic
speed	0	0.01	0.01	0.05	0	0.05	0.12	0,14
True positive	100	75	63	99	95	89	93	100
False positive	100	37	38	8	2	7	6	2
False negative	0	25	37	1	4	11	7	0
True negative	0	63	62	92	98	93	94	98
True positive avg	0.500	0.690	0.625	0.955	0,97	0,91	0,935	0,99
False positive avg	0.500	0.310	0.375	0.045	0,03	0,09	0,065	0,01

Simple Logistic and Nearest Neighbor are both high scoring algortihms. We will try to optimize the Simple Logistic model a bit more, we prefer Simple Logistic over Nearest Neighbor because the score is simply higher. Simple logistic is also preferable because the Nearest neighbor alortihm uses the data as model an does not seperate in trainings and testing data. The use of a training,testing and validation data prevents overfitting on the data.

The Simple logistic algorthim is improved by the use of a Cost sensitive learner: "If the costs are known, they can be incorporated into a financial analysis of the decision-making process. In the two-class case, in which the confusion matrix is like that, the two kinds of error—FPs and FNs—will have different costs; likewise, the two types of correct classification may have different benefits. In the two-class case, costs can be summarized in the form of a 232 matrix in which the diagonal elements represent the two types of correct classification and the off-diagonal elements represent the two types of error. In the multiclass case this generalizes to a square matrix whose size is the number of classes, and again the diagonal elements represent the cost of correct classification." (bron:Data mining practical machine learning tools and techniques. by Ian H. Witten, Eibe Frank and Mark A. Hall)

MinimizeExpectedCost = False with a cost matrix shown below with a 1.5x cost on the false negatives and a 2x cost the false positives. this resulted in a 99.5 percent accuracy.

cost matrix	
0	1.5
2	0

After this we experimented with meta-learners to improve the algotirhm performances of J48. With stacking,bagging and boosting.

Baggin on J48 gives us a improved performance of a true positive rate from 0.91 to 0.93.

Confusion	matix	J48 + bagging
a	b	<– classified as
91	9	a = Blue
5	95	b = Orange

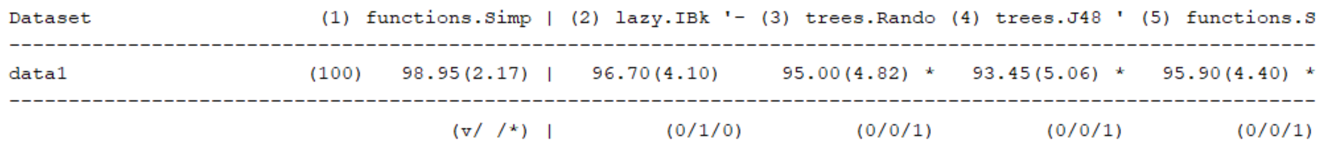
Stacking improved nothing in the algorithms it only made it performe even worse.

Boosting on J48 also made an improvement of 0.02 on the true positive rate but this time in other instances as you can see in the confusion matix on bagging.

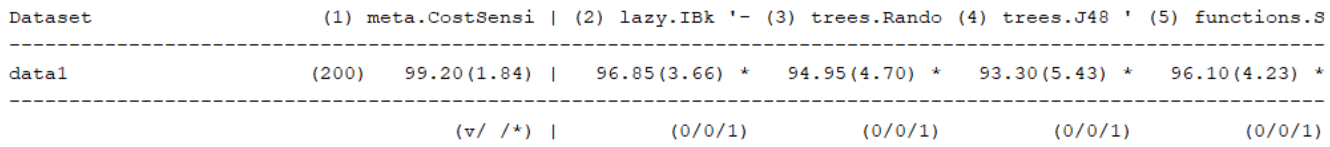
Confusion	matix	J48 + boosting
a	b	<– classified as
92	8	a = Blue
6	94	b = Orange

And finaly the confusion matix from Simple logistic with the use of a costSensativeClassifier. With an improvement from 0.99 to 0.995. with the one instance extra correclty classified

Confusion	matix	Simple Logistic
a	b	<– classified as
100	0	a = Blue
1	99	b = Orange



```
Dataset                    (1) functions.Simp | (2) lazy.IBk '- (3) trees.Rando (4) trees.J48 ' (5) functions.S
--------------------------------------------------------------------------------------------------------------
data1                    (100)   98.95(2.17) |   96.70(4.10)     95.00(4.82) *   93.45(5.06) *   95.90(4.40) *
--------------------------------------------------------------------------------------------------------------
                                   (v/ /*) |         (0/1/0)         (0/0/1)         (0/0/1)         (0/0/1)
```



```
Dataset                    (1) meta.CostSensi | (2) lazy.IBk '- (3) trees.Rando (4) trees.J48 ' (5) functions.S
--------------------------------------------------------------------------------------------------------------
data1                    (200)   99.20(1.84) |   96.85(3.66) *   94.95(4.70) *   93.30(5.43) *   96.10(4.23) *
--------------------------------------------------------------------------------------------------------------
                                   (v/ /*) |         (0/0/1)         (0/0/1)         (0/0/1)         (0/0/1)
```

Here we see an comparison af the highest scoring algorithms in the first run you can see a "*" behind all the algortihms except for the IBK (nearest neighbor) algorithm. The star means that the algorithms are all significantly different. After the second run with 20X repetition and the improvement of the Simple Logisitc algorithm is also IBk significantly different.

https://github.com/kimreijntjens/thema-09 https://github.com/kimreijntjens/wekarunner