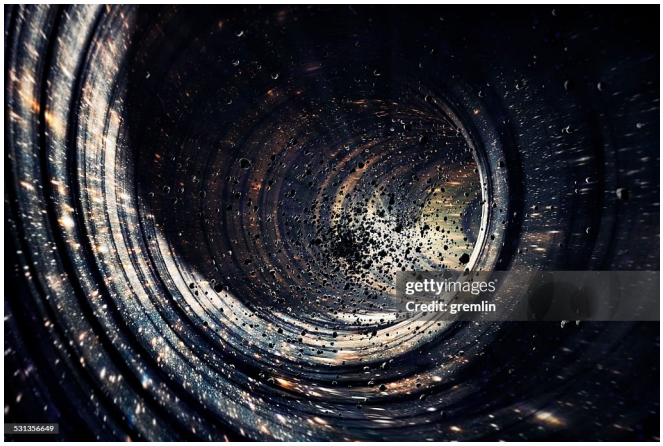
Principal Component Analysis (PCA) in R

Table of contents

REDUNDANCY

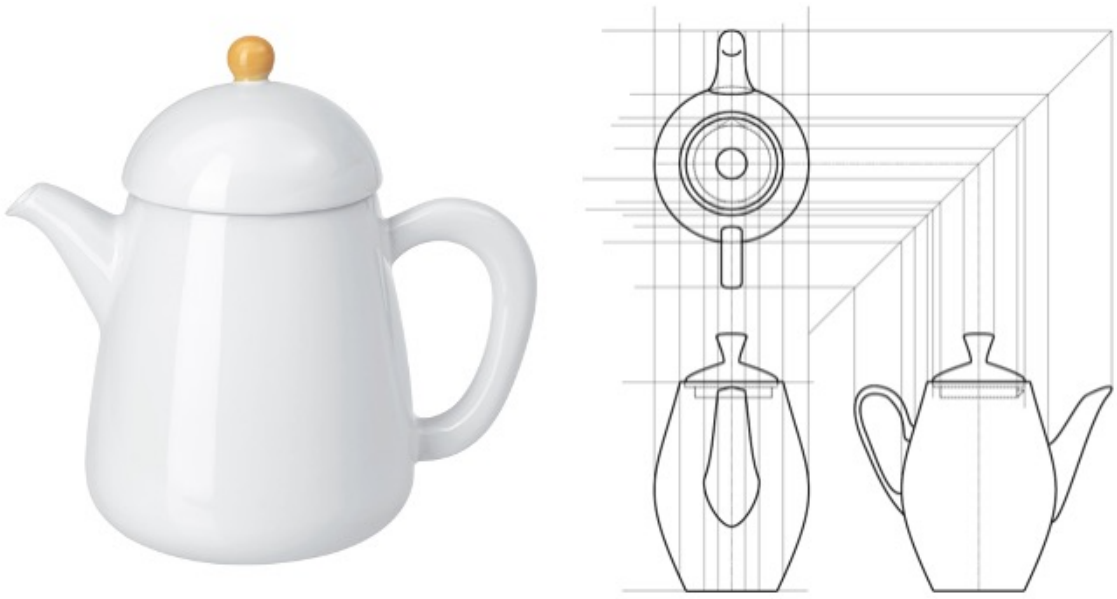
1 Introduction to PCA

- Principal Component Analysis (PCA) is a statistical method used to reduce dimensionality while retaining most of the original variance.
- In simple terms, PCA helps to simplify complex data sets by focusing on the most important parts that capture the majority of the variation in the data.



1.1 Dimensionality Reduction Simplified

- Imagine you have a 3D teapot, which represents data in three dimensions: height, width, and depth.
- Viewing the teapot from above, you see a 2D outline, reducing the dimensions from three to two, while still capturing the essential shape of the teapot.



1.2 How PCA Achieves Dimensionality Reduction

- **Step 1: Find New View (PCA)**:
  - PCA finds the best angle to view the teapot from above to see the most distinctive outline.
- **Step 2: Keep Important Views**:
  - Keeps the views (principal components) that show the most distinctive features, and ignores the rest.
- **Step 3: Re-draw Data**:
  - Re-draws the teapot using these new views, which are fewer in number (reduced dimensions) but still show most of the distinctive features.

2 The Purple Rock Crab Dataset

2.1 Dataset Description

The dataset has 200 rows and 8 columns, describing **5** morphological measurements on 50 crab each of two color forms and both sexes:

Figure 1: *Leptograpsus variegatus*

Column	Description
`sp`	species – `B` or `O` for blue or orange
`sex`	as it says
`index`	index 1:50 within each of the four groups
`FL`	frontal lobe size (mm)
`RW`	rear width (mm)
`CL`	carapace length (mm)
`CW`	carapace width (mm)
`BD`	body depth (mm)

2.2 Exploring the Dataset

```
Attaching package: 'MASS'
```

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

    select
```

```
1 head(crabs)
```

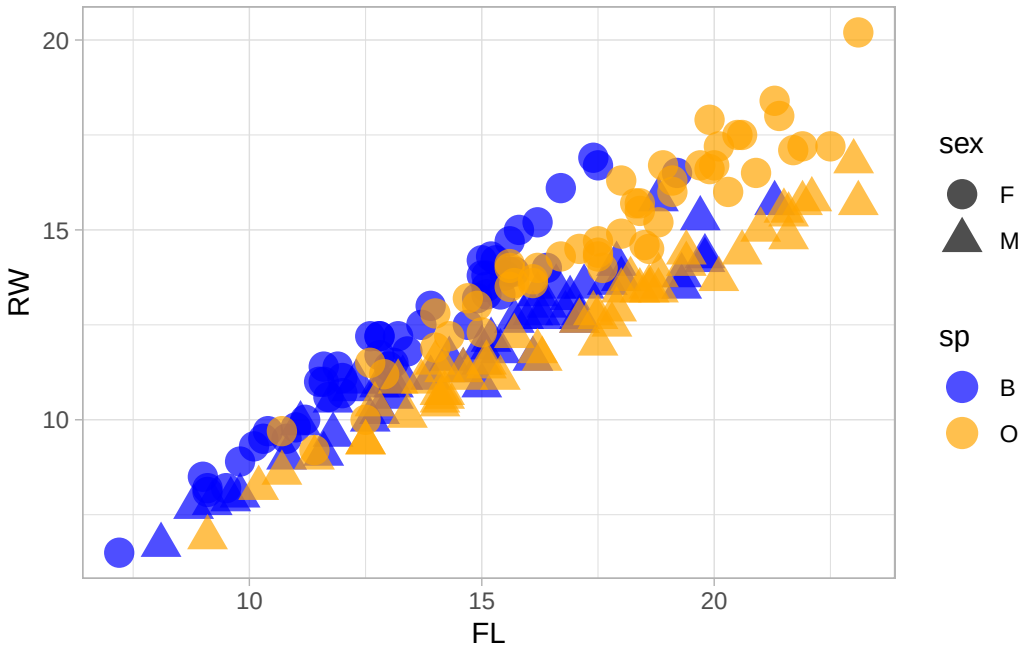
sp	sex	index	FL	RW	CL	CW	BD
B	M	1	8.1	6.7	16.1	19.0	7.0
B	M	2	8.8	7.7	18.1	20.8	7.4
B	M	3	9.2	7.8	19.0	22.4	7.7
B	M	4	9.6	7.9	20.1	23.1	8.2
B	M	5	9.8	8.0	20.3	23.0	8.2
B	M	6	10.8	9.0	23.0	26.5	9.8

2.3 Which morphological measurements can classify the species and the sex?

- `FL` (frontal lobe size),
- `RW` (rear width),
- `CL` (carapace length),
- `CW` (carapace width), or
- `BD` (body depth)?

2.4 FL (frontal lobe) & RW (rear width)

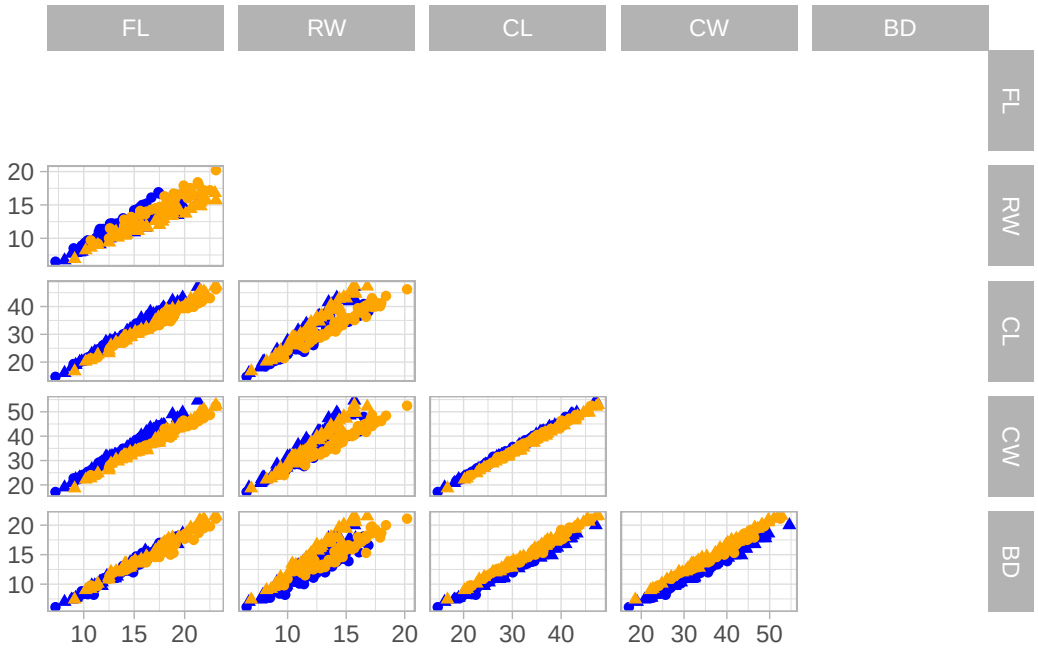
```
ggplot(crabs) +
  geom_point (aes(x = FL, y = RW, color = sp, shape = sex), size = 5,
   ↪  alpha = 0.7) +
  scale_color_manual(values = c("blue", "orange"))
```



2.5 All Pairs

```
GGally::ggpairs(crabs[4:8],
                mapping = aes(color = crabs$sp, shape = crabs$sex),
                upper = "blank",
                diag = "blank") +
  scale_color_manual(values = c("blue", "orange"))
```

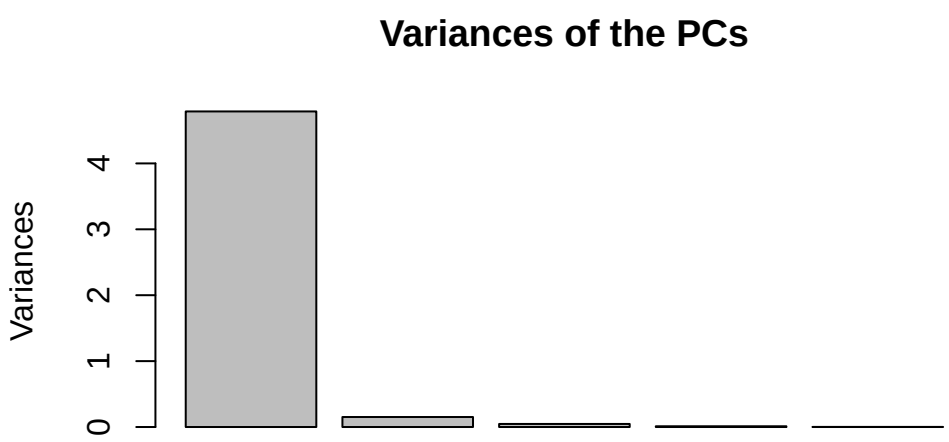
```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```



3 Compute PCA in R

3.1 Compute PCA of the Purple Rock Crab Dataset

```
result = prcomp(crabs[4:8], scale. = TRUE)
plot(result, main = "Variances of the PCs")
```




```
summary(result)
```

```
Importance of components:
                          PC1     PC2     PC3     PC4     PC5
Standard deviation     2.1883 0.38947 0.21595 0.10552 0.04137
Proportion of Variance 0.9578 0.03034 0.00933 0.00223 0.00034
Cumulative Proportion  0.9578 0.98810 0.99743 0.99966 1.00000
```

3.2 Elements of the `prcomp()` result

Element	Description
`sdev`	Standard deviations of the principal components.
`rotation`	Loadings of original variables on principal components.
`center`	Logical indicating if data were centered.
`scale`	Logical indicating if data were scaled.
`x`	Principal component scores (transformed data).
`rank`	Rank of the original data matrix.
`call`	Call that generated the "prcomp" object.
`centering`	Centering values (mean values of original variables).

Element	Description
`scaling`	Scaling values (standard deviations of variables).

```r
str(result)
```

```
List of 5
 $ sdev    : num [1:5] 2.1883 0.3895 0.2159 0.1055 0.0414
 $ rotation: num [1:5, 1:5] 0.452 0.428 0.453 0.451 0.451 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:5] "FL" "RW" "CL" "CW" ...
  .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
 $ center  : Named num [1:5] 15.6 12.7 32.1 36.4 14
  ..- attr(*, "names")= chr [1:5] "FL" "RW" "CL" "CW" ...
 $ scale   : Named num [1:5] 3.5 2.57 7.12 7.87 3.42
  ..- attr(*, "names")= chr [1:5] "FL" "RW" "CL" "CW" ...
 $ x       : num [1:200, 1:5] -4.92 -4.38 -4.12 -3.87 -3.82 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:200] "1" "2" "3" "4" ...
  .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
 - attr(*, "class")= chr "prcomp"
```

4 Projections

In PCA, projections are the positions of your original data points on the new principal component axes. They represent how the data looks when viewed from the perspective of the principal components, simplifying complex, multi-dimensional data into a more manageable form.

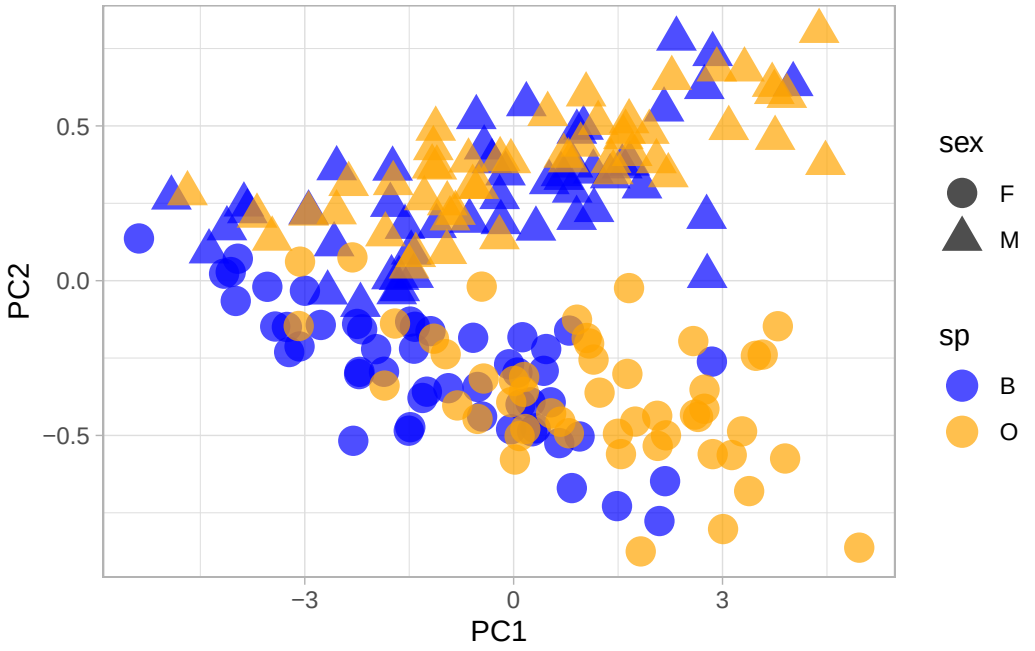
4.1 Tranformed Data (Projections)

```
pca_df = data.frame(sp = crabs$sp, sex = crabs$sex, result$x)
head(pca_df)
```

sp	sex	PC1	PC2	PC3	PC4	PC5
B	M	-4.915239	0.2677733	0.1219517	-0.0390459	0.0692952
B	M	-4.375197	0.0938381	0.0391337	0.0054535	-0.0030446
B	M	-4.118329	0.1684532	-0.0335594	0.0380015	0.0379655
B	M	-3.873960	0.2453925	-0.0144647	0.0190459	0.0013117
B	M	-3.824458	0.2236052	0.0150296	0.0544971	-0.0248217
B	M	-2.945564	0.2194700	-0.0383320	-0.0696657	0.0189264

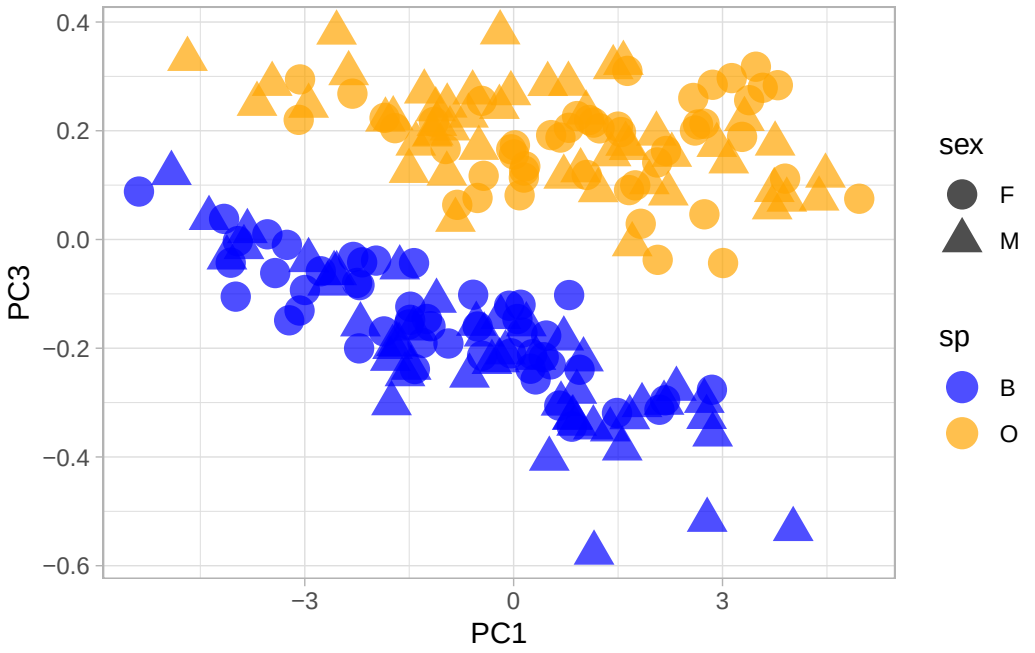
4.2 PC1 & PC2

```
ggplot(pca_df) +
  geom_point (aes(x = PC1, y = PC2, color = sp, shape = sex), size = 5,
   ↪  alpha = 0.7) +
  scale_color_manual(values = c("blue", "orange"))
```



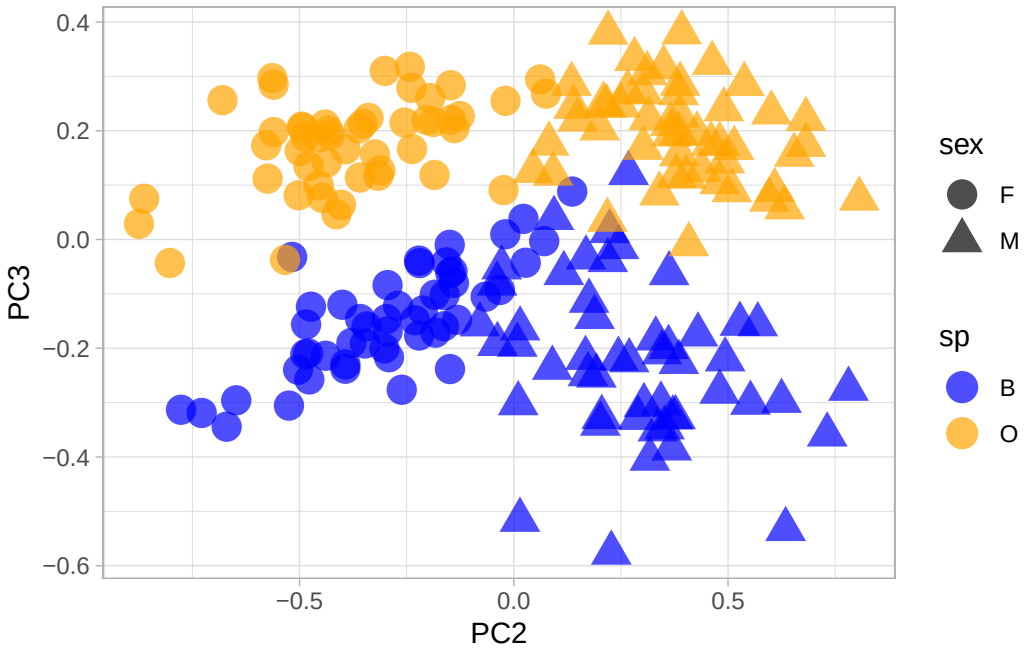
4.3 PC1 & PC3

```
ggplot(pca_df) +
  geom_point (aes(x = PC1, y = PC3, color = sp, shape = sex), size = 5,
   ↪  alpha = 0.7) +
  scale_color_manual(values = c("blue", "orange"))
```



4.4 PC2 & PC3

```
ggplot(pca_df) +
  geom_point (aes(x = PC2, y = PC3, color = sp, shape = sex), size = 5,
   ↪  alpha = 0.7) +
  scale_color_manual(values = c("blue", "orange"))
```



5 Loadings

Loadings in PCA are the coefficients that multiply each standard unit of your original variables to get the principal component scores. Loadings are weights indicating the contribution of each variable to each principal component.

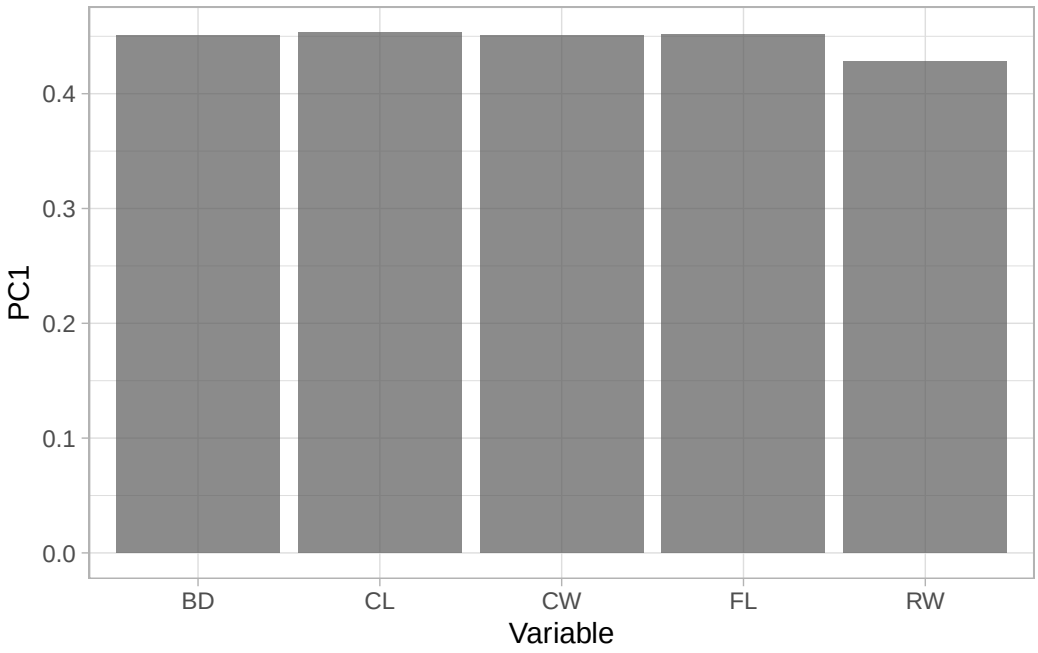
5.1 Measurements) Loadings on the PCs

```
loadings = data.frame(Variable = colnames(crabs[4:8]), result$rotation)
loadings
```

	Variable	PC1	PC2	PC3	PC4	PC5
FL	FL	0.4520437	0.1375813	0.5307684	0.6969234	0.0964916
RW	RW	0.4280774	-0.8981307	-0.0119791	-0.0837032	-0.0544176
CL	CL	0.4531910	0.2682381	-0.3096816	-0.0014446	-0.7916827
CW	CW	0.4511127	0.1805959	-0.6525696	0.0891878	0.5745267
BD	BD	0.4511336	0.2643219	0.4431610	-0.7066364	0.1757433

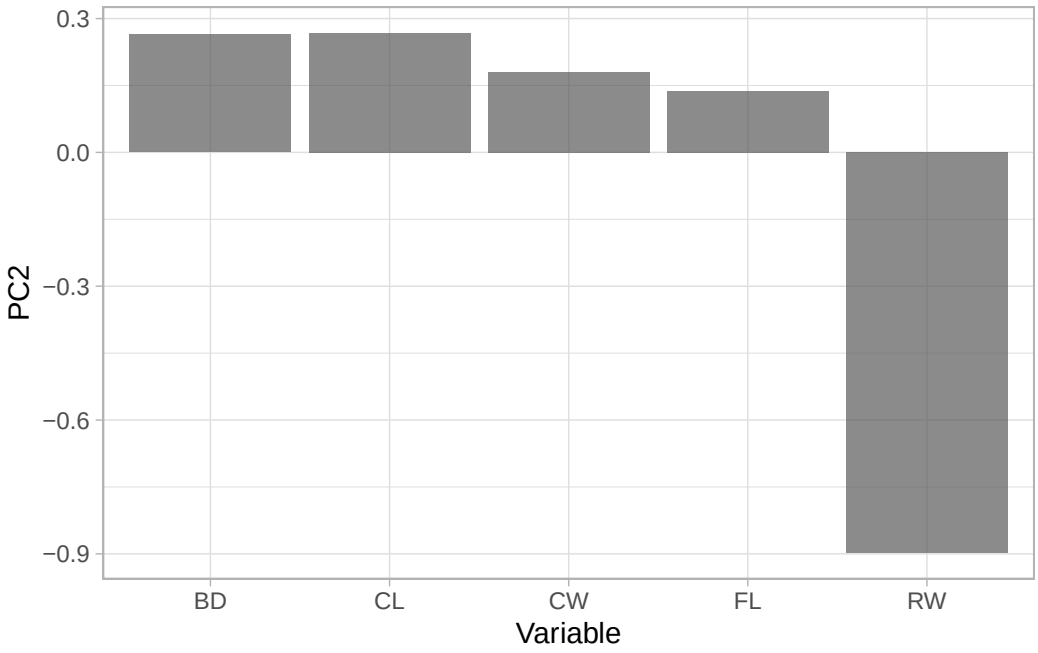
5.2 Loadings on PC1

```
ggplot(loadings) + geom_bar(aes(x = Variable, y = PC1), stat =
↪  "identity", alpha = 0.7)
```



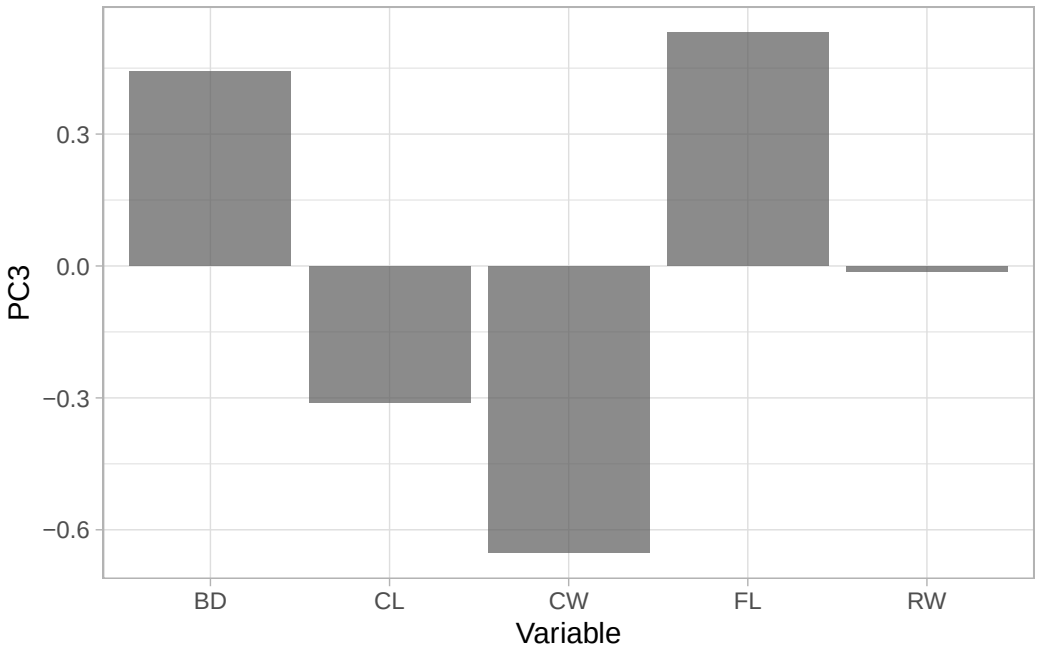
5.3 Loadings on PC2

```
ggplot(loadings) + geom_bar(aes(x = Variable, y = PC2), stat =
↪  "identity", alpha = 0.7)
```



5.4 Loadings on PC3

```
ggplot(loadings) + geom_bar(aes(x = Variable, y = PC3), stat =
↪  "identity", alpha = 0.7)
```



6 Appendix

6.1 Mathematics of PCA

- **Step 1: Standardization**:
  - Shift and scale your data so that each trait has the same importance.
  - **Formula**: $X_{\text{std}} = \frac{X - \bar{X}}{s}$
- **Step 2: Covariance Matrix Computation**:
  - Find relationships between different traits by calculating the covariance matrix.
  - **Formula**: Covariance Matrix $= \frac{1}{n-1} \sum (X_{\text{std}} - \bar{X})(X_{\text{std}} - \bar{X})^T$
- **Step 3: Eigen Decomposition**:
  - Find the "main" patterns of variation (principal components).
  - Formula: $\Sigma v = \lambda v$
- **Step 4: Sort Eigenvectors**:
  - Rank these patterns by how much variation they show.
- **Step 5: Select Principal Components**:
  - Pick the top patterns you are interested in.

- **Step 6: Project Data**:
  - Re-draw your data using these new patterns as axes.
  - **Formula**: $Y = X_{\text{std}}W$

6.2 Covariance Matrix

1. **Definition**:
   - The covariance matrix is a square matrix that captures the relationships (or covariances) between each pair of variables in a multi-dimensional dataset. Each entry in the matrix represents the covariance between two different variables.
2. **Covariance**:
   - Covariance is a measure that tells you how two variables change together. If they tend to increase together, the covariance is positive; if one decreases while the other increases, the covariance is negative.
3. **Diagonal Elements**:
   - The diagonal elements of the covariance matrix are the variances of each variable, i.e., the covariance of a variable with itself.
4. **Symmetry**:
   - The covariance matrix is symmetric, meaning the value of covariance between variable i and variable j is the same as the covariance between variable j and variable i.

6.3 Eigenvectors in PCA

1. **Direction of Maximum Variance**:
   - The eigenvectors of the covariance matrix represent the directions of maximum variance in the data. These are the directions in which the data spread out the most.
2. **Orthogonality**:
   - The eigenvectors are orthogonal to each other, meaning they are at right angles to each other in the multi-dimensional space. This orthogonality ensures that each principal component (eigenvector) captures a unique and uncorrelated aspect of the data's structure.
3. **Principal Components**:

- The eigenvectors, also known as the principal components in PCA, provide a new set of axes onto which the data is projected. This projection reduces the dimensionality of the data while retaining as much of the original variance as possible.

4. **Ranking and Selection**:
    - The eigenvalues associated with each eigenvector indicate the amount of variance explained by that eigenvector. By ranking the eigenvectors based on their eigenvalues, you can select the top eigenvectors that capture the most significant patterns of variance within the data.
5. **Data Compression and Noise Reduction**:
    - By selecting a subset of eigenvectors (principal components), you essentially compress the data, retaining only the most important features while discarding the noise.

6.4 About the Purple Rock Crab Image

- Image credit: Damon Tighe
- Image source: https://flic.kr/p/9eXoGK