Анализ главных компонент

Анализ и визуализация многомерных данных с использованием R

М. Варфоломеева В. Хайтов А. Лянгузова

Вы сможете

- Проводить анализ главных компонент при помощи функций из пакета vegan
- Оценивать долю дисперсии, объясненной компонентами
- Снизить размерность данных, оставив небольшое число компонент
- Интерпретировать смысл компонент по их факторным нагрузкам
- Строить ординацию объектов в пространстве главных компонент
- Создавать комплексные переменные и использовать их в других видах анализов

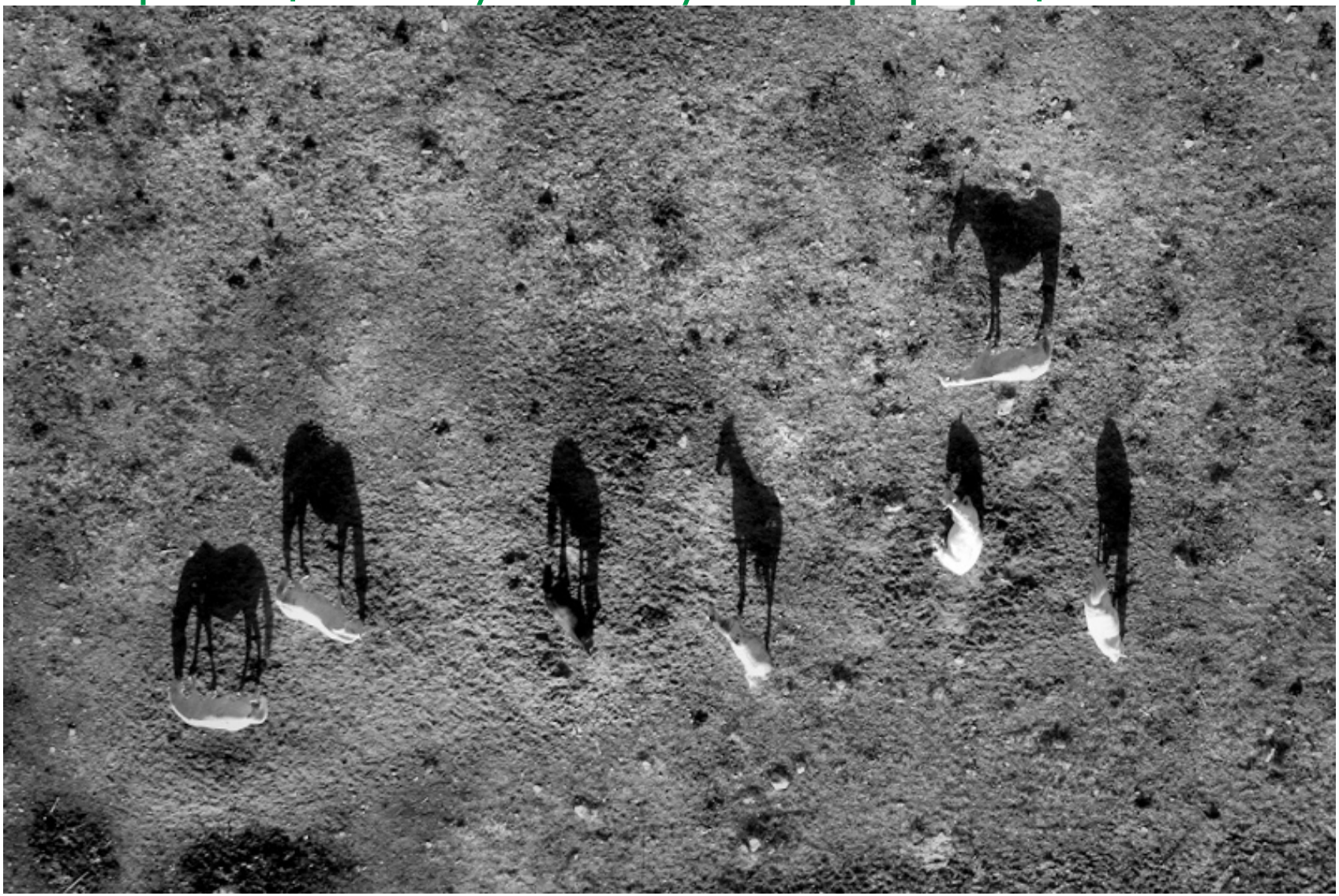
Постановка задачи для анализа главных компонент

Зачем нужен анализ главных компонент?

Когда признаков много, можно представить все объекты как облако точек в многомерном пространстве. Обычно в биологических исследованиях признаки объектов взаимозависимы (между ними есть ненулевая ковариация или корреляция).

Migration by Don McCullough on Flickr

Не все проекции несут важную информацию



black shadows for a white horses / les negres ombres dels cavalls blancs by Ferran Jordà on Flickr

Можно найти оптимальную проекцию, чтобы сохранить максимум информации в минимуме измерений

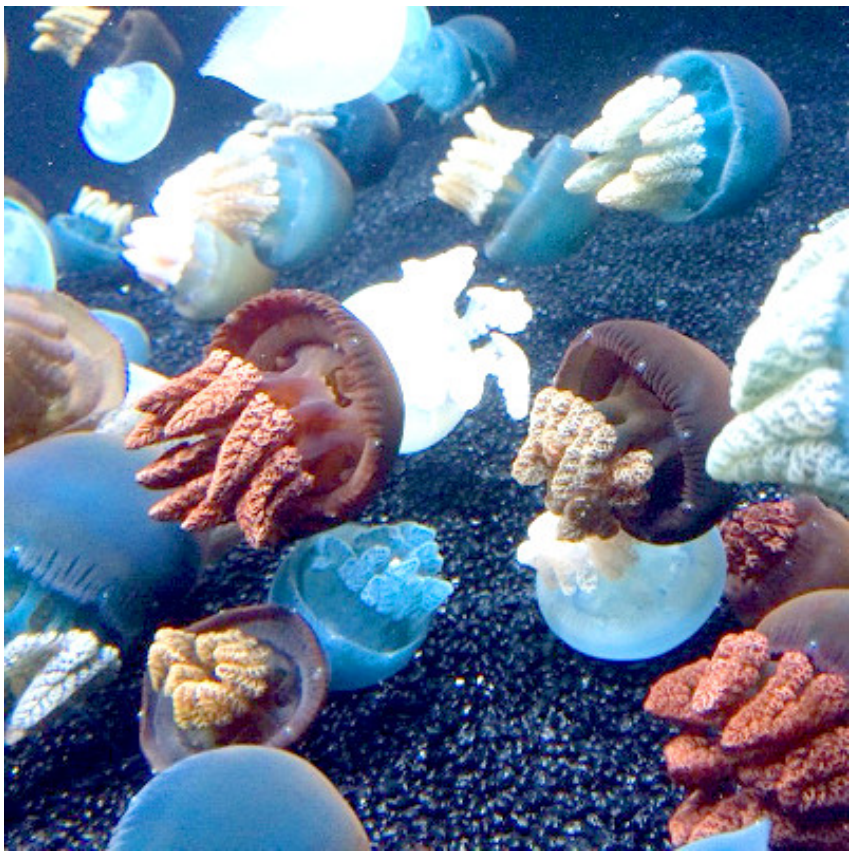
Cat's shadow by Marina del Castell on Flickr

Анализ главных компонент (Principal Component Analysis, PCA)

- Ординация объектов по многим признакам.
- Описание системы взаимосвязей между множеством исходных признаков и ранжирование признаков по важности.
- Снижение размерности многомерных данных (dimension reduction) и создание синтетических взаимонезависимых признаков для других анализов (например, для регрессии, дискриминантного анализа)

Пример: Размеры медуз

Данные о размерах медуз *Catostylus mosaicus* (Lunn & McNeil 1991). Медузы собраны в реке Хоксбери (Новый Южный Уэльс, Австралия): часть — на острове Дангар, другая — в заливе Саламандер.



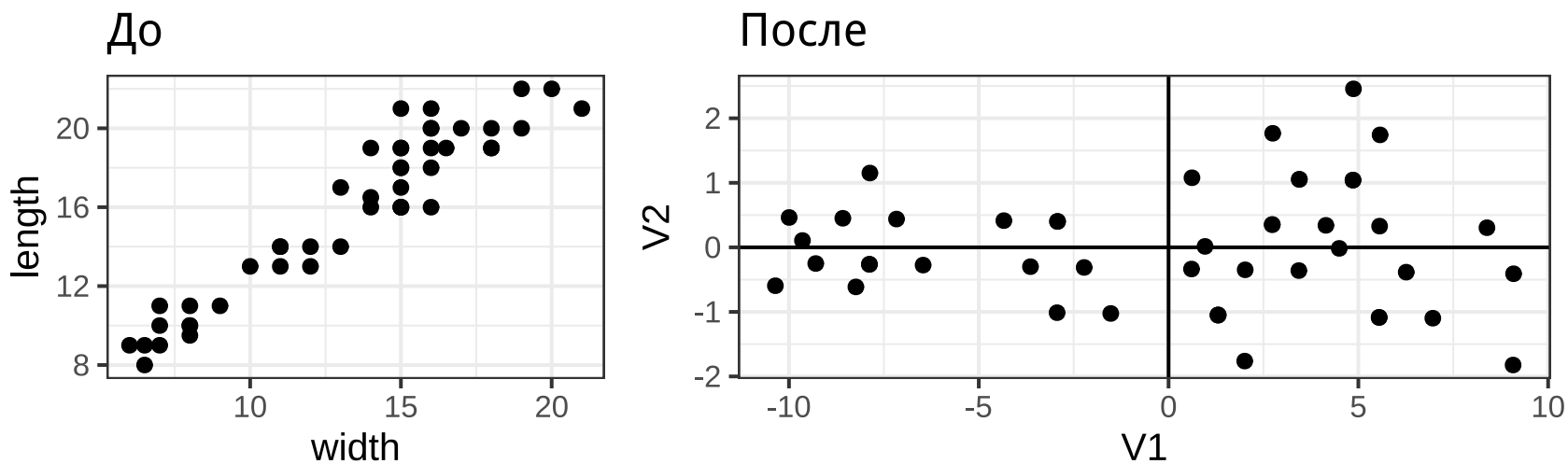
location	width	length
D	6.0	9.0
D	6.5	8.0
D	6.5	9.0
D	7.0	9.0
D	7.0	10.0
D	7.0	11.0
D	8.0	9.5
D	8.0	10.0
D	8.0	10.0
D	8.0	11.0

Задача анализа главных компонент

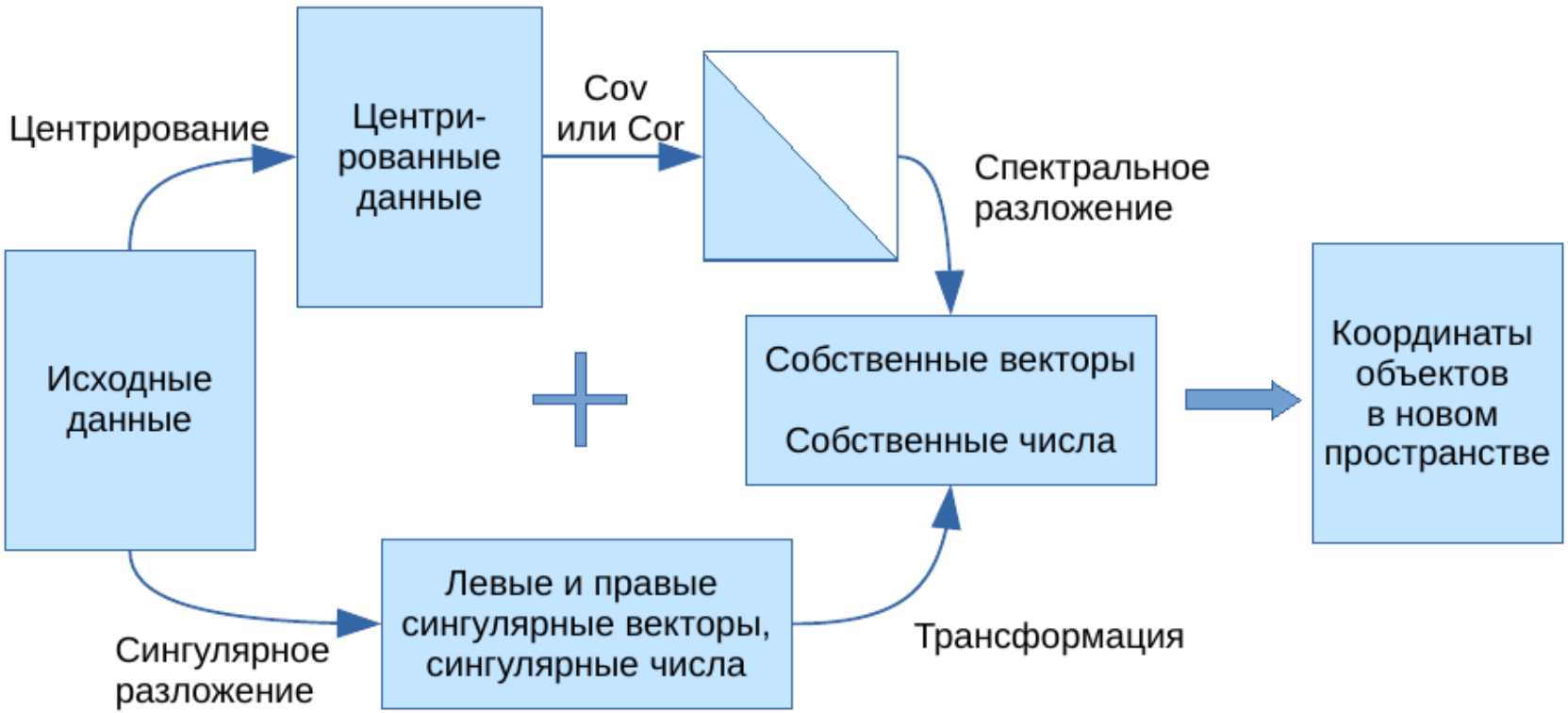
Нужно найти такую трансформацию исходных данных, чтобы “новые” переменные:

- содержали всю исходную информацию;
- были независимы друг от друга;
- были ранжированы в порядке убывания важности (например, в порядке убывания их дисперсии).

Интуитивно, мы можем добиться этого, если проведем одну ось вдоль направления, в котором максимально вытянуто облако исходных данных. Вторую ось проведем перпендикулярно первой (и они будут независимы).



Алгоритм PCA



PCA в R своими руками

Открываем данные по медузам и извлекаем из таблицы числовые измерения.

```
jelly <- read.delim("data/jellyfish.csv")
X_raw <- jelly[, 2:3]
```

Задание 1

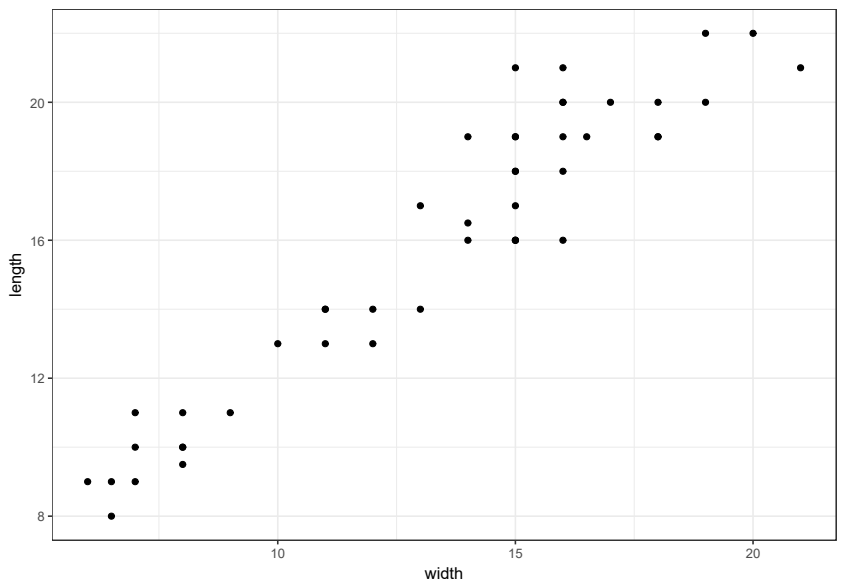
Получите новые координаты для датасета с медузами в осях главных компонент.

Решение

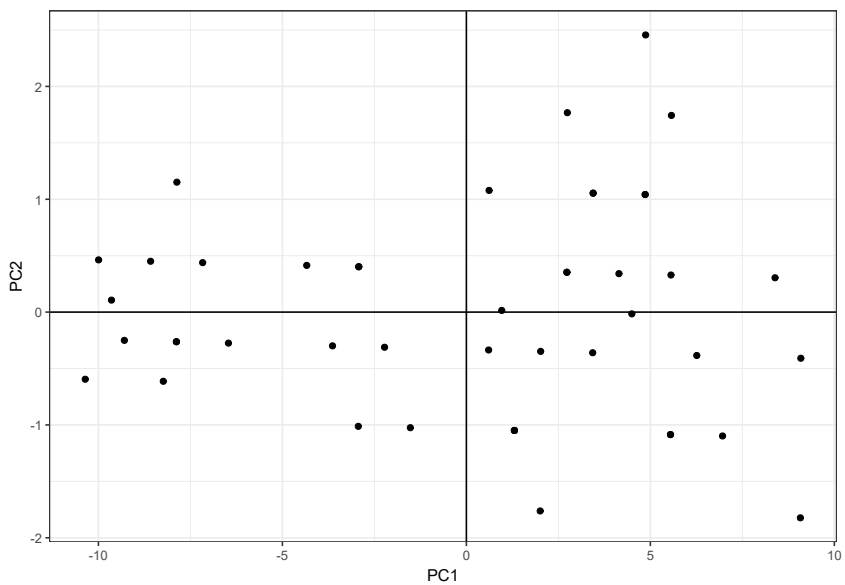
```r
# Исходные данные
X <- scale(X_raw, center = TRUE, scale = FALSE) # Центрируем
A <- t(X) %*% X / (nrow(X) - 1)   # Матрица ковариаций
E <- eigen(A)             # Спектральное разложение
U <- E$vectors            # Собственные векторы
Lambda <- E$values

# Координаты точек в новом пространстве
Y <- X %*% U
```

Визуализация результатов PCA

```
# график исходных данных
gg_jelly_raw <-
  ggplot(as.data.frame(X_raw),
         aes(x = width, y = length))
         ↪  +
  geom_point(size = 2)
gg_jelly_raw
```

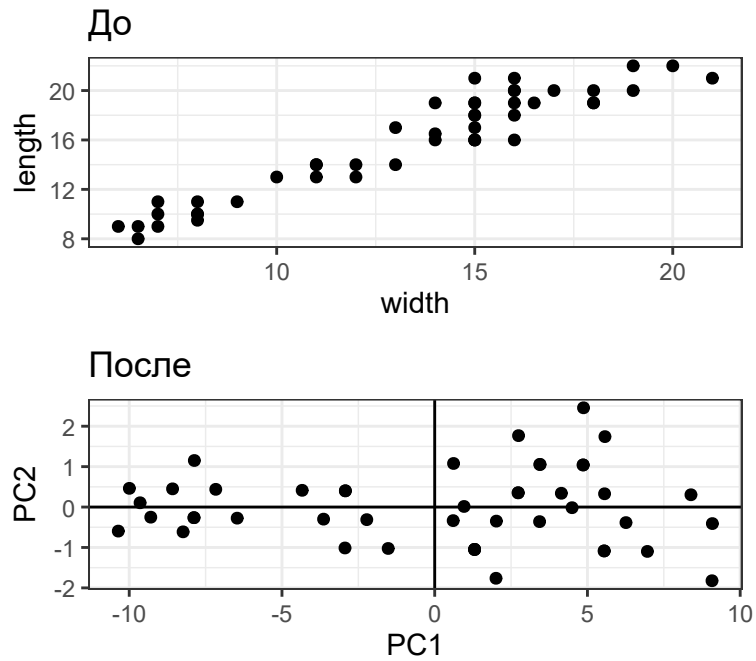



```
# график в главных осях
gg_jelly_rotated <-
  ggplot(as.data.frame(Y),
  ↪    aes(x = V1, y = V2)) +
    geom_point(size = 2) +
    geom_hline(yintercept = 0) +
    geom_vline(xintercept = 0) +
    labs(x = "PC1", y = "PC2")
gg_jelly_rotated
```



Результаты работы PCA

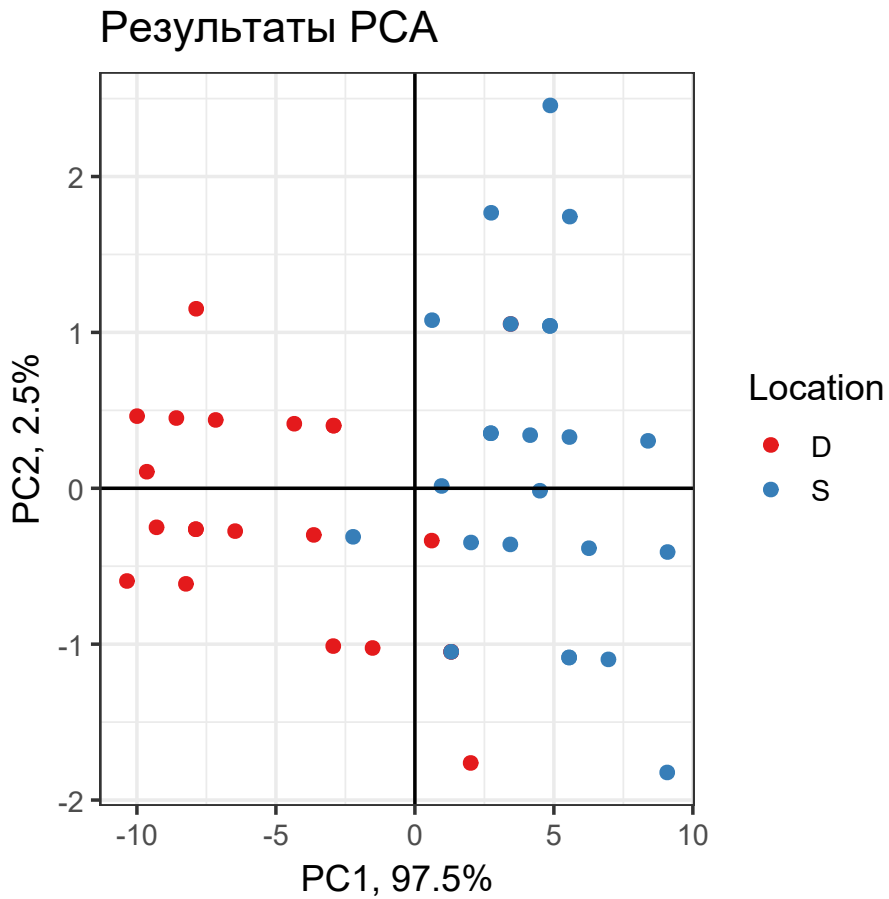
- **Собственные векторы** (**факторные нагрузки**)
  - перпендикулярны друг другу (ортогональны, независимы)
  - задают **главные компоненты** — направления новых осей
  - линейные комбинации исходных признаков
  - упорядочены в порядке убывания дисперсии
- **Собственные числа**
  - показывают дисперсию вдоль главных компонент
  - упорядочены в порядке убывания дисперсии
  - используются для вычисления доли общей изменчивости, связанной с каждой из главных компонент



- **Факторные координаты**
  - координаты объектов в пространстве главных компонент

Результаты работы PCA

- Главные компоненты
  - новые “синтетические” признаки объектов, которые сочетают несколько исходных признаков
  - упорядочены по убыванию доли объясненной изменчивости
  - используя разное число главных компонент можно снизить размерность исходных данных



- PC1 — “размер медузы” — больше всего изменчивости;
- PC2 — остаточная изменчивость.

Действительно многомерные данные

Пример: Потребление белков в странах Европы с разными видами продуктов питания

Paleo Diet by zsoolt on Flickr

Данные из Weber, 1973

Открываем данные

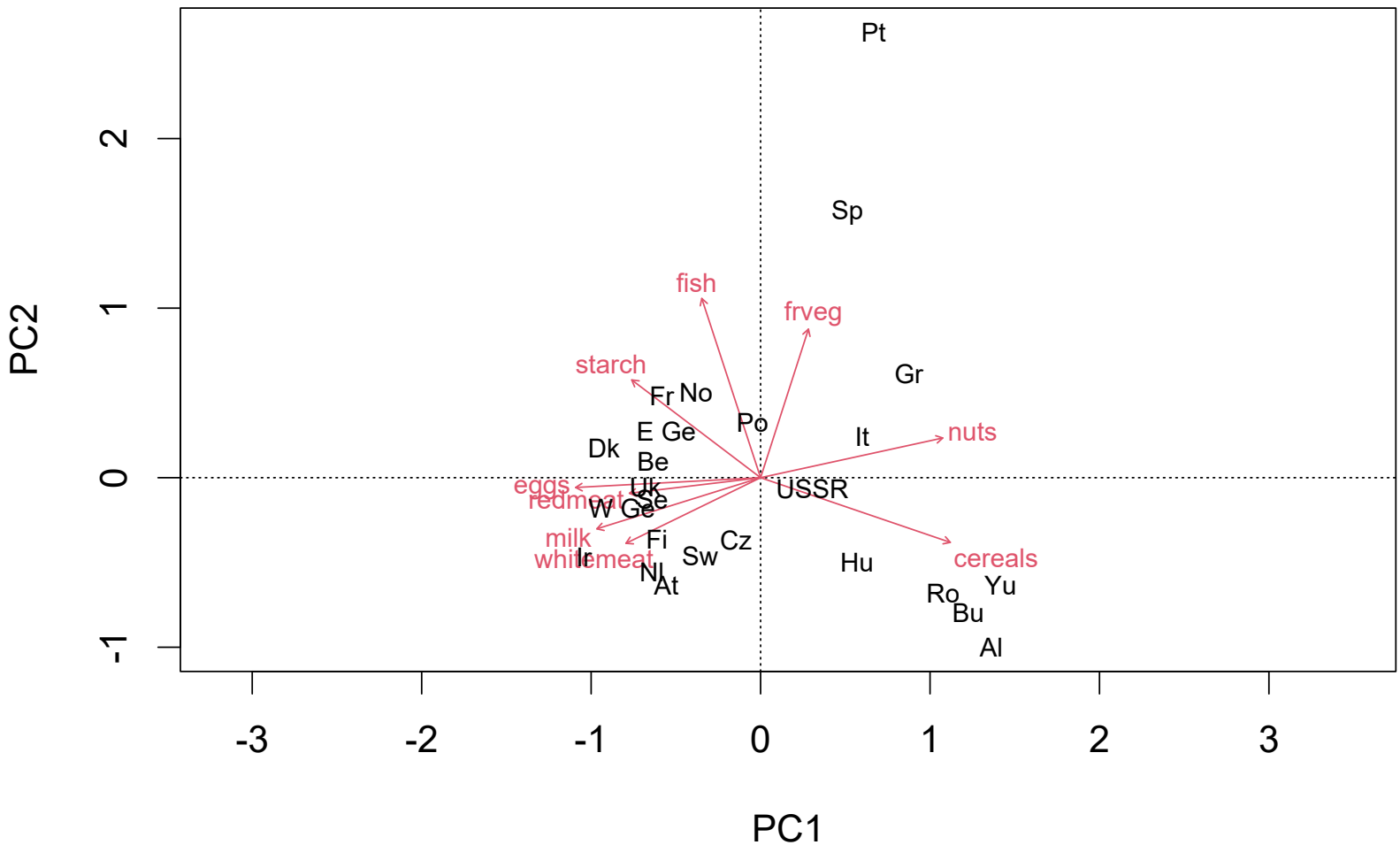
```
protein <- read.table(file="data/protein.csv", sep="\t", dec=".",
↪  header=TRUE)
protein$region <- factor(protein$region)
rownames(protein) <- protein$country
head(protein)
```

	country	region	redmeat	whitemeat	eggs	milk	fish	cereals	starch	nuts	frveg
Al	Al	Balkans	10.1	1.4	0.5	8.9	0.2	42.3	0.6	5.5	1.7
At	At	W Europe	8.9	14.0	4.3	19.9	2.1	28.0	3.6	1.3	4.3
Be	Be	W Europe	13.5	9.3	4.1	17.5	4.5	26.6	5.7	2.1	4.0
Bu	Bu	Balkans	7.8	6.0	1.6	8.3	1.2	56.7	1.1	3.7	4.2
Cz	Cz	E Europe	9.7	11.4	2.8	12.5	2.0	34.3	5.0	1.1	4.0
Dk	Dk	Scandinavia	10.6	10.8	3.7	25.0	9.9	21.9	4.8	0.7	2.4

Данные из Weber, 1973

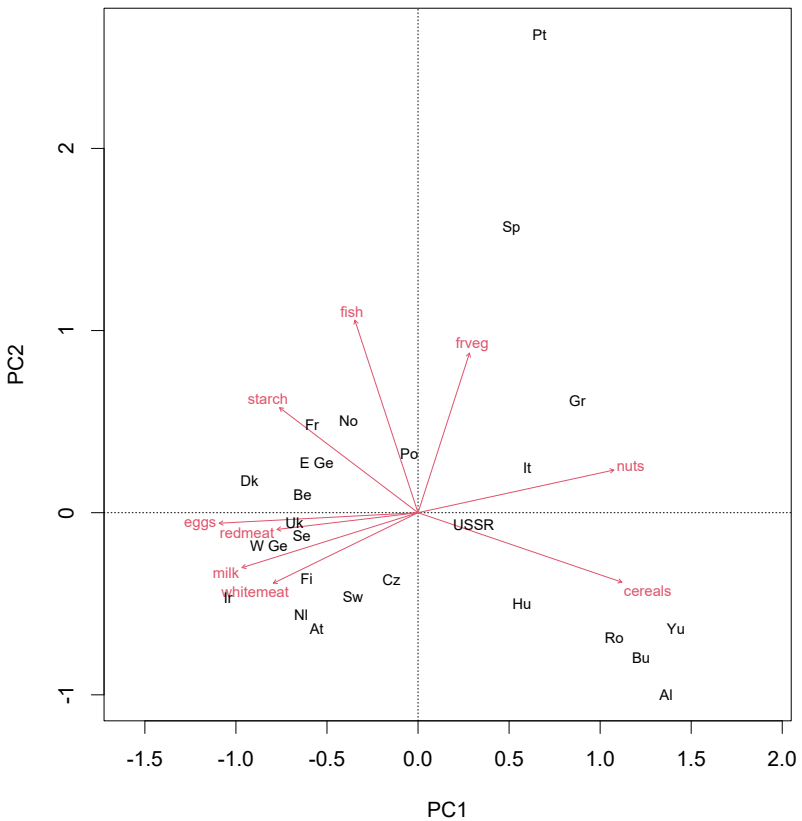
Делаем PCA

```
library(vegan)
prot_pca <- rda(protein[, -c(1, 2)], scale = TRUE)
op <- par(mar = c(5, 4, 0, 2) + 0.1, cex = 1.5)
biplot(prot_pca)
```



Делаем PCA

```
library(vegan)
prot_pca <- rda(protein[, -c(1, 2)], scale = TRUE)
op <- par(mar = c(5, 4, 0, 2) + 0.1, cex = 1.5)
biplot(prot_pca)
par(op)
```



Разбираемся с результатами PCA

```
summary(prot_pca)
```

```
Call:
rda(X = protein[, -c(1, 2)], scale = TRUE)

Partitioning of correlations:
              Inertia Proportion
Total               9          1
Unconstrained       9          1

Eigenvalues, and their contribution to the correlations

Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7
Eigenvalue            4.0064 1.6350 1.1279 0.9547 0.46384 0.32513 0.27161
Proportion Explained  0.4452 0.1817 0.1253 0.1061 0.05154 0.03613 0.03018
Cumulative Proportion 0.4452 0.6268 0.7522 0.8582 0.90976 0.94589 0.97607
                          PC8     PC9
Eigenvalue            0.11629 0.09911
Proportion Explained  0.01292 0.01101
Cumulative Proportion 0.98899 1.00000
```

1. Сколько компонент нужно оставить?

Собственные числа показывают вклады главных компонент в общую изменчивость

```
Eigenvalues, and their contribution to the correlations

Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6  ...
Eigenvalue             4.0064 1.6350 1.1279 0.9547 0.46384 0.32513 ...
Proportion Explained   0.4452 0.1817 0.1253 0.1061 0.05154 0.03613 ...
Cumulative Proportion 0.4452 0.6268 0.7521 0.8582 0.90976 0.94589 ...
```

```
eigenvals(prot_pca) # собственные числа
```

```
     PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
4.006438 1.634999 1.127920 0.954664 0.463838 0.325131 0.271606 0.116292
     PC9
0.099112
```

Сколько компонент нужно оставить, если мы хотим редуцировать данные?

- Эмпирические правила (выберите любое, но одно)
  - Компоненты у которых соб. число > 1 (правило Кайзера-Гатмана)
  - В сумме объясняют заданный % от общей изменчивости (60-80%) — слишком субъективно
  - Объясняют больше, чем по Broken Stick Model

```
eigenvals(prot_pca) # собственные числа
```

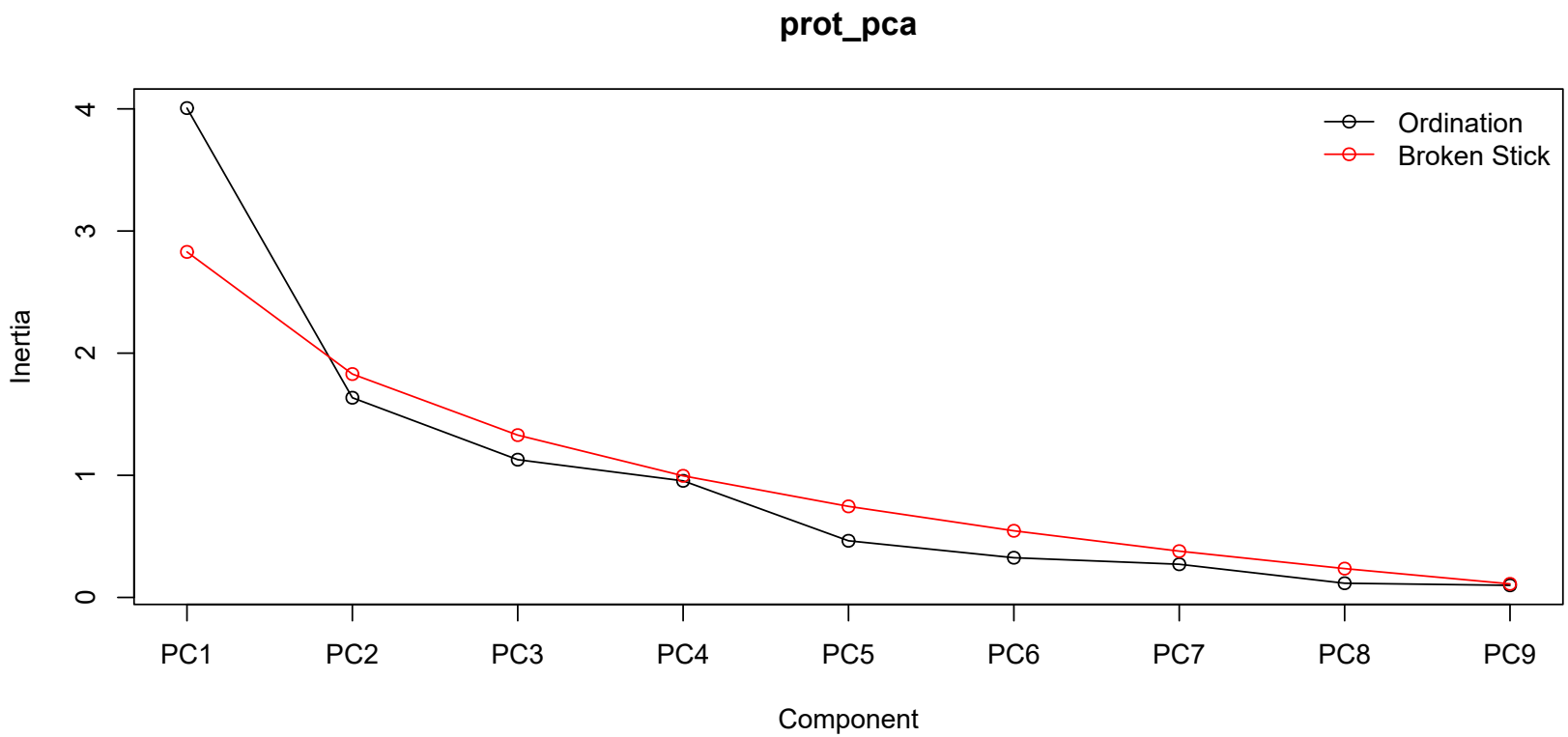
```
     PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8
4.006438 1.634999 1.127920 0.954664 0.463838 0.325131 0.271606 0.116292
     PC9
0.099112
```

```
bstick(prot_pca) # ожидаемое по Broken Stick Model
```

```
      PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
2.8289683 1.8289683 1.3289683 0.9956349 0.7456349 0.5456349 0.3789683 0.2361111
      PC9
0.1111111
```

График собственных чисел

```
screeplot(prot_pca, type = "lines", bstick = TRUE) # график собственных
↪  чисел
```



2. Графики факторных нагрузок и ординации

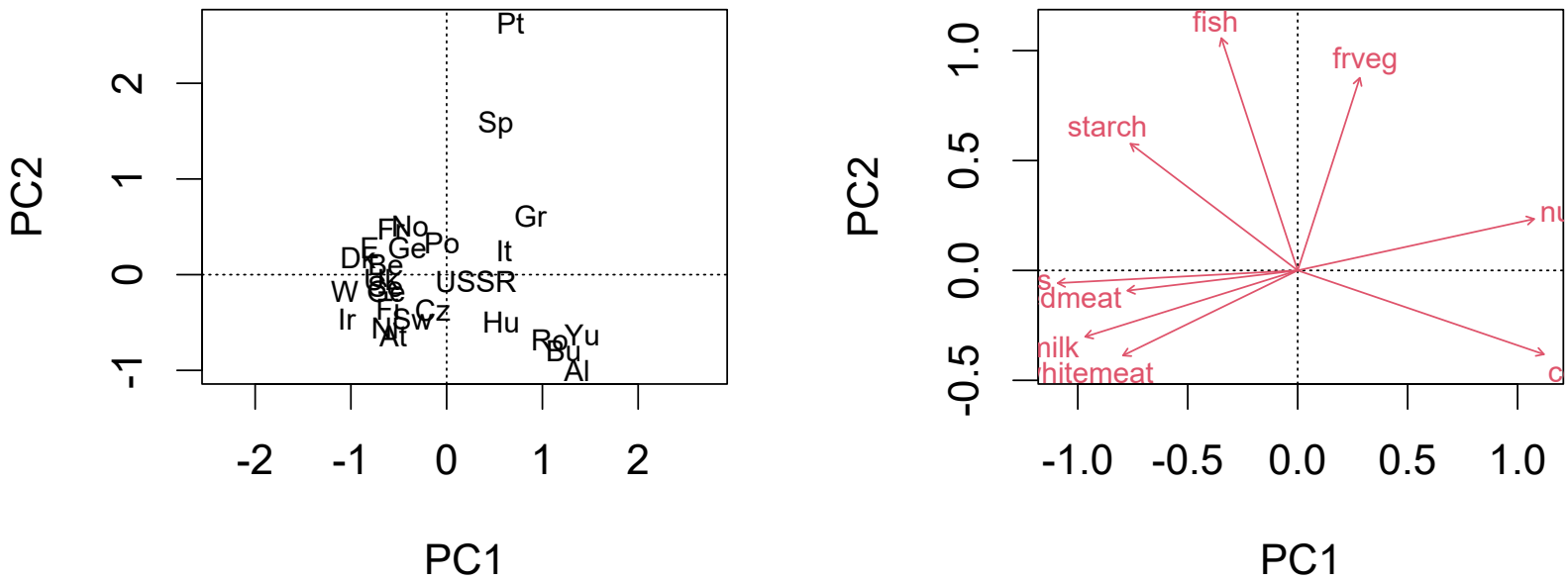
Параметр `scaling`

Внимание! Координаты объектов или переменных можно получить в нескольких вариантах, отличающихся масштабом. От этого масштаба будет зависеть интерпретация.

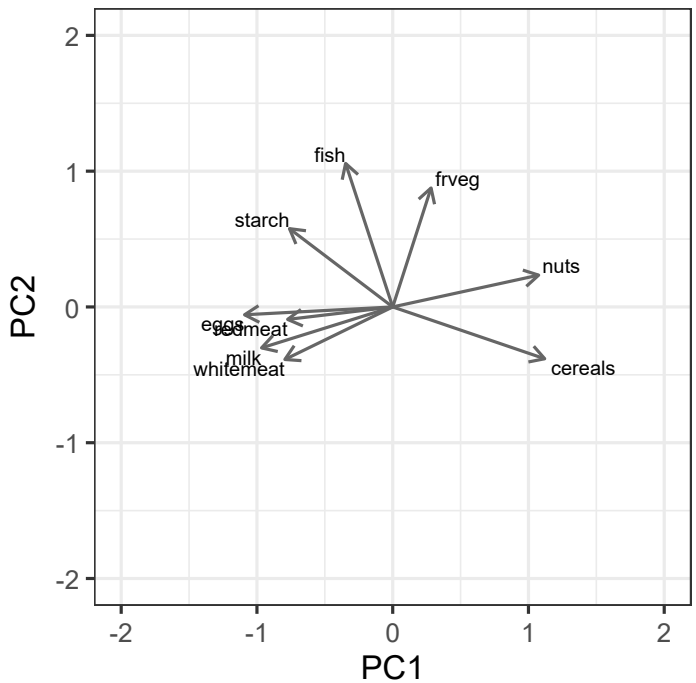
scaling	Название графика	Масштаб	Расстояния между объектами	Углы между векторами
1, sites	биплот расстояний	координаты объектов масштабированы (x корень из соб. чисел)	аппроксимируют евклидовы	нет смысла
2, species	биплот корреляций	координаты признаков масштабированы (x корень из соб. чисел)	НЕ аппроксимируют евклидовы	отражают корреляции
3, symmetric		масштабированы координаты объектов и признаков (x корень 4-й степени из соб. чисел)		
0, none		нет масштабирования		

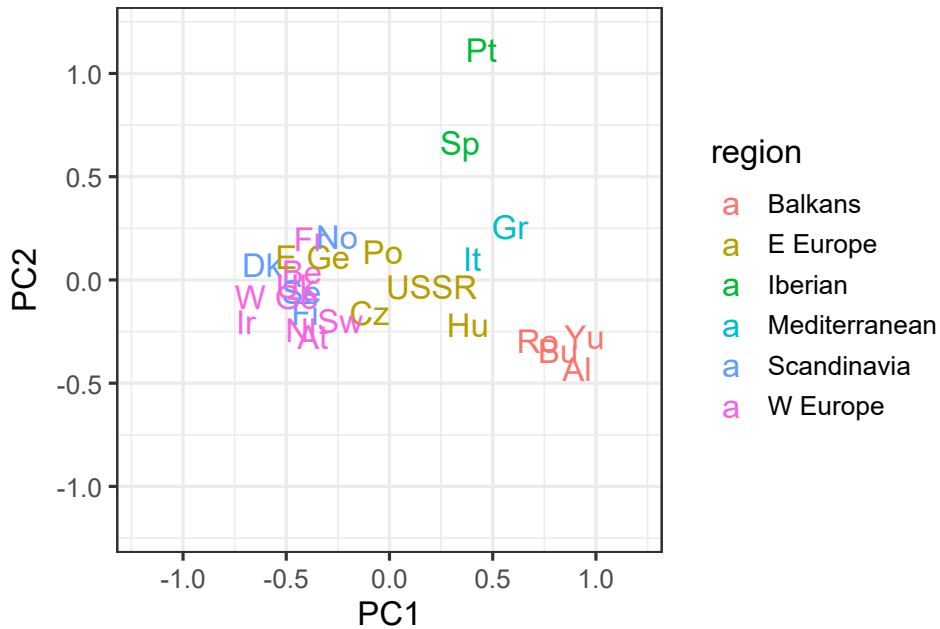
Графики

```
op <- par(mfrow = c(1, 2), cex = 1.5)
# График факторных координат
biplot(prot_pca, display = "sites")
# График факторных нагрузок
biplot(prot_pca, display = "species", scaling = "species")
par(op)
```



Те же самые графики можно построить в ggplot2





region
- a Balkans
- a E Europe
- a Iberian
- a Mediterranean
- a Scandinavia
- a W Europe

Исходный код графика

```
# Данные для графиков
df_scores <- data.frame(
  protein[, 1:2],
  scores(prot_pca,
         display = "sites",
         choices = c(1, 2, 3),
         scaling = "sites"))
## График ординации в ggplot
p_scores <- ggplot(df_scores,
                   aes(x = PC1, y = PC2,
                       colour = region))
                       ↪  +
  geom_text(aes(label = country)) +
  coord_equal(xlim = c(-1.2, 1.2), ylim
  ↪  = c(-1.2, 1.2))
```

```
## Данные для графика нагрузок
df_load <- as.data.frame(
  scores(prot_pca,
         display = "species",
         choices = c(1, 2, 3),
         scaling = "species"))
# поправки для размещения подписей
df_load$hjust <- ifelse(df_load$PC1 >= 0, -0.1, 1)
df_load$vjust <- ifelse(df_load$PC2 >= 0, -0.1, 1)
library(grid) # для стрелочек
ar <- arrow(length = unit(0.25, "cm"))
## График нагрузок в ggplot
p_load <- ggplot(df_load) +
  geom_text(aes(x = PC1, y = PC2,
                label = rownames(df_load)),
            size = 3,
            vjust = df_load$vjust,
            hjust = df_load$hjust) +
  geom_segment(aes(x = 0, y = 0,
    xend = PC1, yend = PC2),
    colour = "grey40", arrow = ar) +
  coord_equal(xlim = c(-2, 2), ylim = c(-2, 2))
```

3. Интерпретация компонент

Интерпретация компонент

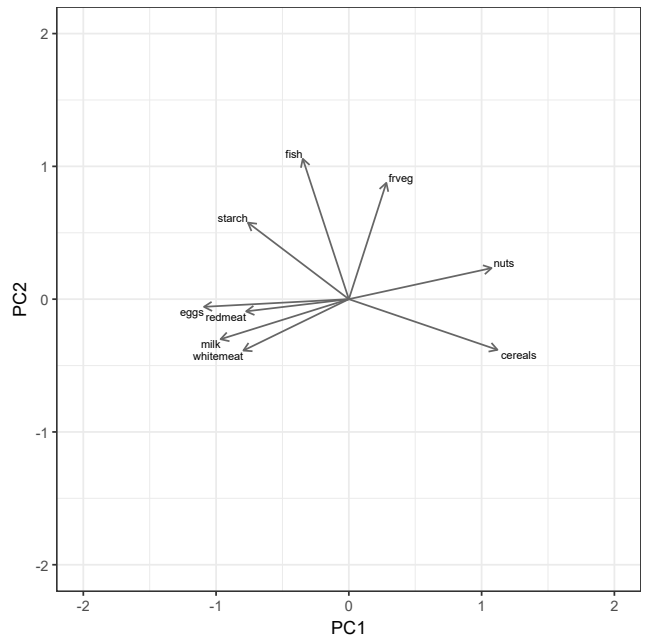
Факторные нагрузки оценивают вклады переменных в изменчивость по главной компоненте

- Модуль нагрузки — величина вклада
- Знак нагрузки — направление вклада

```
scores(prot_pca,
       display = "species",
       choices = c(1, 2, 3),
       scaling = 0)
```

```
                PC1         PC2         PC3
redmeat   -0.3026094 -0.05625165 -0.29757957
whitemeat -0.3105562 -0.23685334  0.62389724
eggs      -0.4266785 -0.03533576  0.18152828
milk      -0.3777273 -0.18458877 -0.38565773
fish      -0.1356499  0.64681970 -0.32127431
cereals    0.4377434 -0.23348508  0.09591750
starch    -0.2972477  0.35282564  0.24297503
nuts       0.4203344  0.14331056 -0.05438778
frveg      0.1104199  0.53619004  0.40755612
attr(,"const")
[1] 3.833659
```

Первая главная компонента:
Высокие **положительные нагрузки по первой главной компоненте** у переменных `cereals` и `nuts`. Значит, чем больше значение PC1, тем больше потребление этих продуктов.
Высокие **отрицательные нагрузки** у переменных `eggs`, `milk`, `whitemeat`, `redmeat`. Т.е., чем меньше значение PC1, тем больше их потребление.



Интерпретация компонент

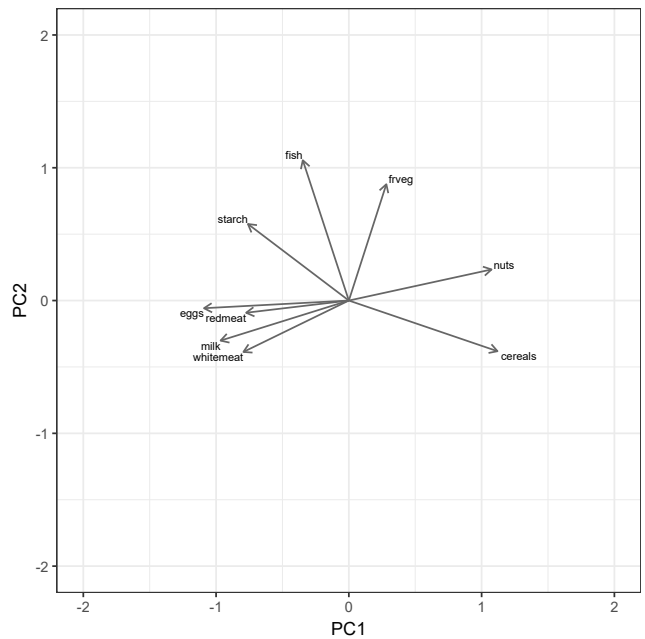
Факторные нагрузки оценивают вклады переменных в изменчивость по главной компоненте

- Модуль нагрузки — величина вклада
- Знак нагрузки — направление вклада

```
scores(prot_pca,
       display = "species",
       choices = c(1, 2, 3),
       ↪  scaling = 0)
```

```
                PC1         PC2         PC3
redmeat   -0.3026094 -0.05625165 -0.29757957
whitemeat -0.3105562 -0.23685334  0.62389724
eggs      -0.4266785 -0.03533576  0.18152828
milk      -0.3777273 -0.18458877 -0.38565773
fish      -0.1356499  0.64681970 -0.32127431
cereals    0.4377434 -0.23348508  0.09591750
starch    -0.2972477  0.35282564  0.24297503
nuts       0.4203344  0.14331056 -0.05438778
frveg      0.1104199  0.53619004  0.40755612
attr(,"const")
[1] 3.833659
```

Первая главная компонента:
Высокие **положительные нагрузки по первой главной компоненте** у переменных `cereals` и `nuts`. Значит, чем больше значение PC1, тем больше потребление этих продуктов.
Высокие **отрицательные нагрузки** у переменных `eggs`, `milk`, `whitemeat`, `redmeat`. Т.е., чем меньше значение PC1, тем больше их потребление.



- Т.е. первую компоненту можно назвать “Мясо – злаки и орехи”

Интерпретация компонент

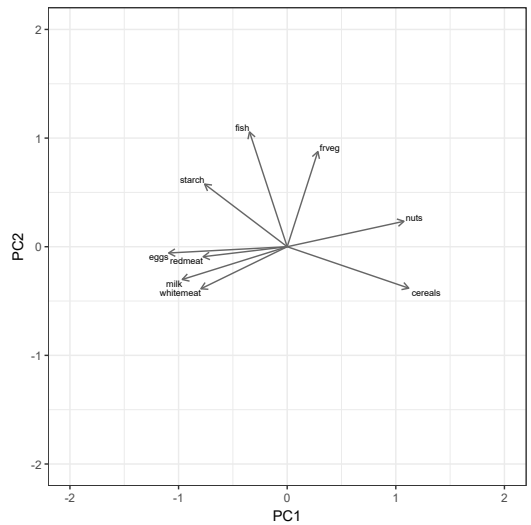
Факторные нагрузки оценивают вклады переменных в изменчивость по главной компоненте

- Модуль нагрузки — величина вклада
- Знак нагрузки — направление вклада

```
scores(prot_pca, display = "species",
       choices = c(1, 2, 3), scaling = 0)
```

```
                PC1         PC2         PC3
redmeat   -0.3026094 -0.05625165 -0.29757957
whitemeat -0.3105562 -0.23685334  0.62389724
eggs      -0.4266785 -0.03533576  0.18152828
milk      -0.3777273 -0.18458877 -0.38565773
fish      -0.1356499  0.64681970 -0.32127431
cereals    0.4377434 -0.23348508  0.09591750
starch    -0.2972477  0.35282564  0.24297503
nuts       0.4203344  0.14331056 -0.05438778
frveg      0.1104199  0.53619004  0.40755612
attr(,"const")
[1] 3.833659
```

Вторая главная компонента: Высокие **положительные нагрузки по второй главной компоненте** у переменных `fish`, `frveg`. Значит, чем больше значение PC2, тем больше потребление рыбы, овощей. Высоких **отрицательных нагрузок по второй главной компоненте** нет ни у одной из переменных.



Интерпретация компонент

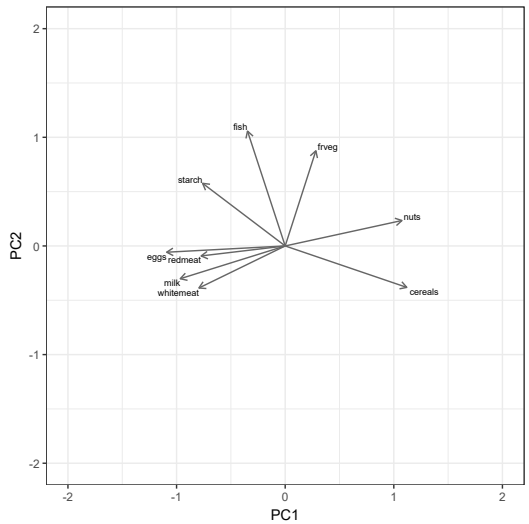
Факторные нагрузки оценивают вклады переменных в изменчивость по главной компоненте

- Модуль нагрузки — величина вклада
- Знак нагрузки — направление вклада

```
scores(prot_pca, display = "species",
       choices = c(1, 2, 3), scaling = 0)
```

```
                PC1         PC2         PC3
redmeat   -0.3026094 -0.05625165 -0.29757957
whitemeat -0.3105562 -0.23685334  0.62389724
eggs      -0.4266785 -0.03533576  0.18152828
milk      -0.3777273 -0.18458877 -0.38565773
fish      -0.1356499  0.64681970 -0.32127431
cereals    0.4377434 -0.23348508  0.09591750
starch    -0.2972477  0.35282564  0.24297503
nuts       0.4203344  0.14331056 -0.05438778
frveg      0.1104199  0.53619004  0.40755612
attr(,"const")
[1] 3.833659
```

Вторая главная компонента: Высокие **положительные нагрузки по второй главной компоненте** у переменных `fish`, `frveg`. Значит, чем больше значение PC2, тем больше потребление рыбы, овощей. Высоких **отрицательных нагрузок по второй главной компоненте** нет ни у одной из переменных.



- Т.е. вторую компоненту можно назвать “Рыба и овощи”

PCA и другие методы

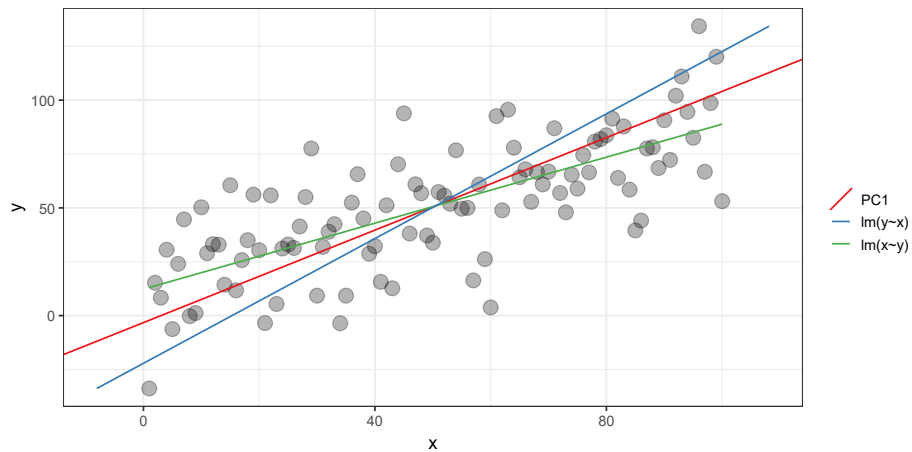
PCA и другие методы

PCA

- Метод обучения “без учителя” (unsupervised learning)
- Все переменные-признаки равноправны
- Задачи:
  - описать сходство объектов
  - снизить размерность данных
  - интерпретировать связи между переменными
- Главные компоненты — линейные комбинации переменных, задающие направления максимального варьирования исходных данных.

Линейная регрессия

- Метод обучения “с учителем” (supervised learning)
- Переменные делятся на зависимые (отклики) и независимые (предикторы)
- Задачи:
  - описать зависимость значений отклика от предикторов
  - предсказать значения отклика при известных значениях предикторов
- Линия регрессии — направление минимального разброса значений зависимой переменной (сумма квадратов остатков).



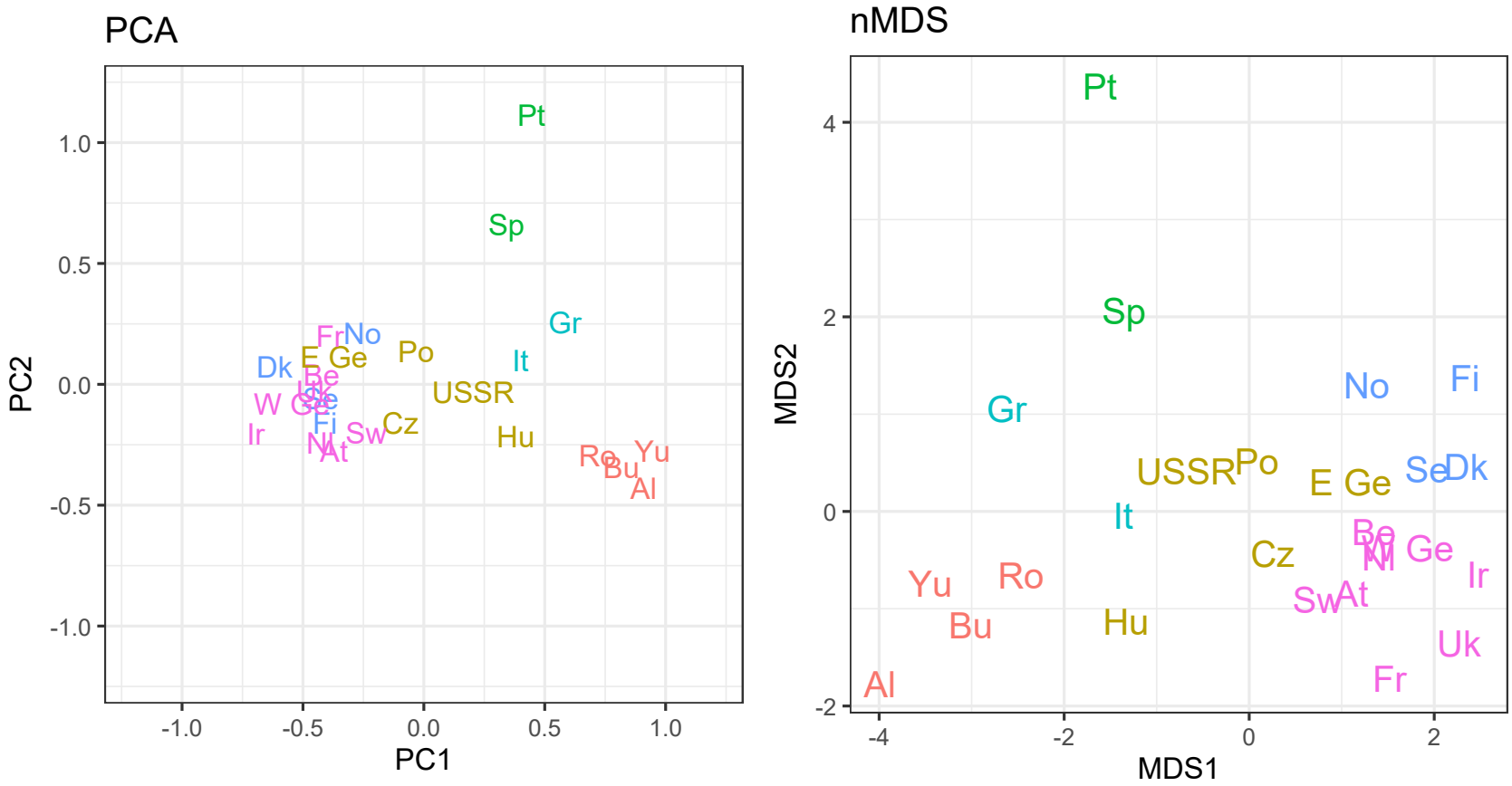
PCA

- PCA представляет многомерные данные в пространстве независимых осей, ранжированных по важности, поэтому **есть возможность оставить только самые важные оси изменчивости**.
- Расстояния между объектами на любой ординации PCA соответствуют их евклидовым расстояниям в пространстве главных компонент.
- Исходные признаки — количественные переменные, связанные друг с другом линейно. Для описания различий между такими объектами подходит евклидово расстояние.

nMDS

- nMDS пытается найти отображение многомерного пространства в заданном числе измерений (например, на плоскости) **с максимальным сохранением информации из всех измерений**.
- Ранги расстояний между объектами на nMDS будут соответствовать их рангам в исходной матрице различий.
- Исходные признаки могут быть любыми, т.к. может быть использована любая мера различий между объектами.

Результаты PCA и nMDS будут похожи, если для nMDS-ординации использовано евклидово расстояние



Создание составных переменных при помощи PCA

Создание составных переменных

Факторные координаты — это новые составные признаки, которых можно использовать вместо исходных переменных.

Свойства факторных координат:

- Среднее = 0, Дисперсия = 1;
- Не коррелируют друг с другом.

Применение:

- Уменьшение числа зависимых переменных — для дисперсионного анализа;
- Уменьшение числа предикторов — во множественной регрессии.

```
          PC1         PC2         PC3
Al  0.90914911 -0.42530648 -0.45941317
At -0.37109966 -0.27160251  0.34896271
Be -0.42310297  0.04160401  0.05648245
Bu  0.81751706 -0.33937963  0.03946349
Cz -0.09663475 -0.15720451  0.31195835
Dk -0.61697429  0.07445781 -0.19622597
```

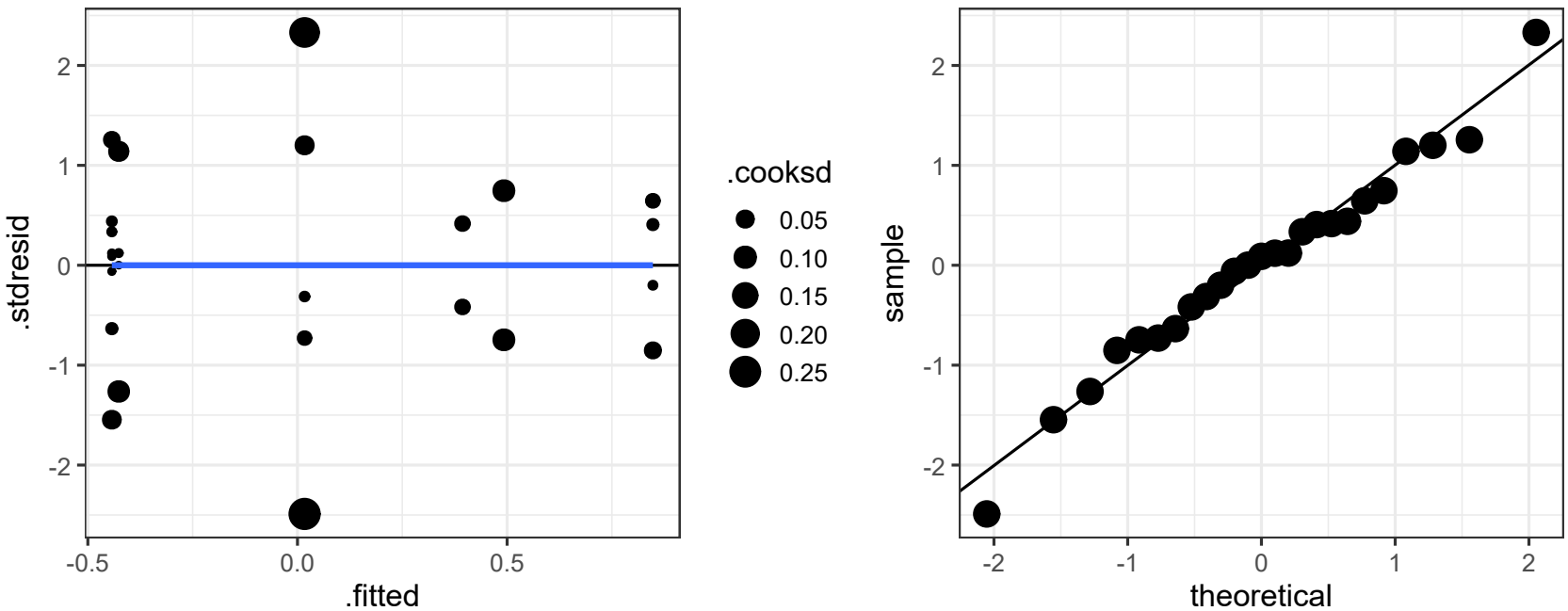
При помощи дисперсионного анализа можно проверить, различается ли значение первой главной компоненты (“Мясо – злаки и орехи”) между разными регионами Европы

```
# Значения факторов (= факторные координаты)
df <- data.frame(region = protein$region,
  scores(prot_pca, display = "sites", choices = c(1, 2, 3), scaling =
  ↪  "sites"))
mod <- lm(PC1 ~ region, data = df)
anova(mod)
```

```
Analysis of Variance Table

Response: PC1
          Df Sum Sq Mean Sq F value    Pr(>F)
region     5 5.9656 1.19312  39.296 2.238e-09 ***
Residuals 19 0.5769 0.03036
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

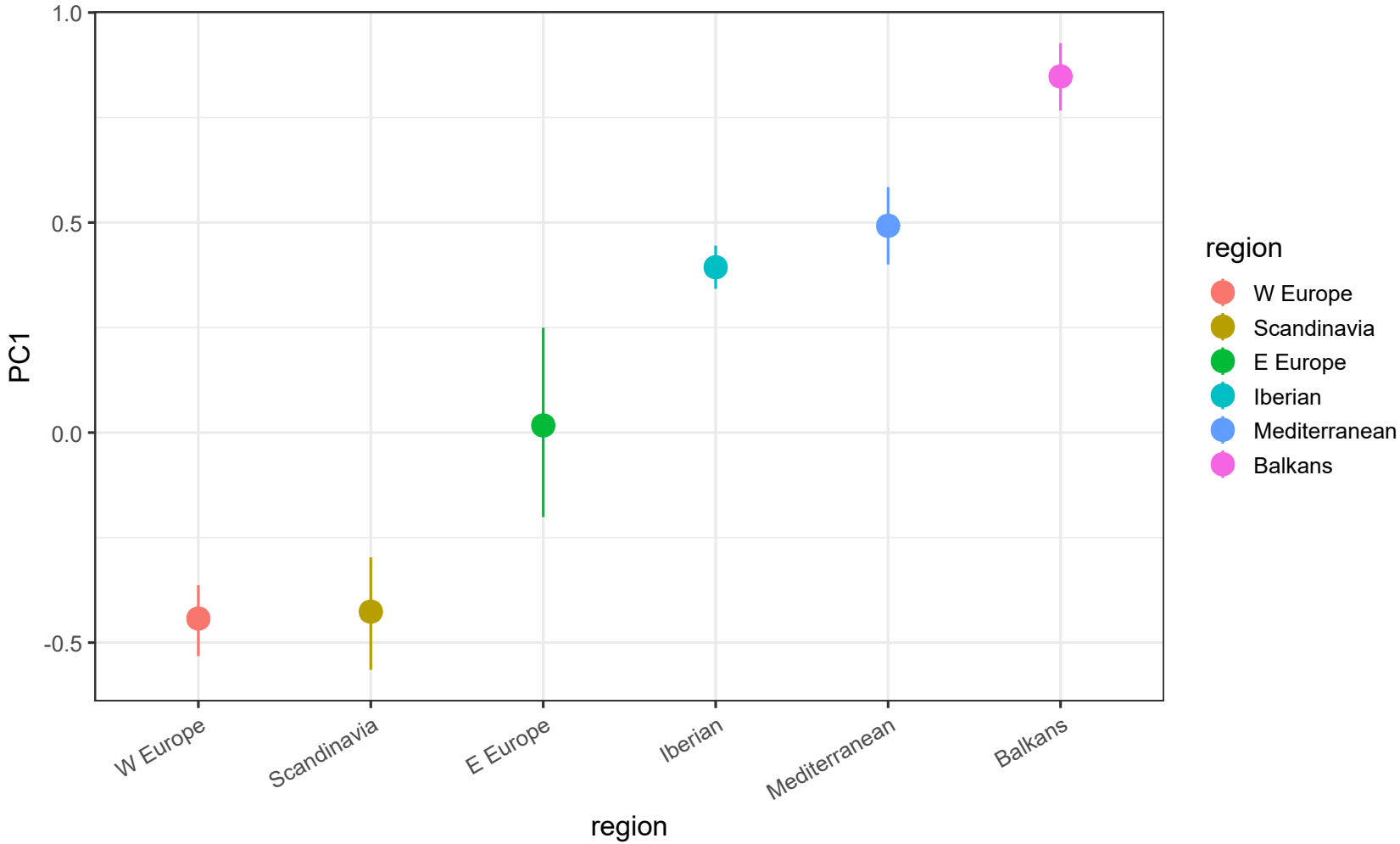
- Регионы Европы различаются по потреблению мяса, злаков и орехов

Проверка условий применимости дисперсионного анализа



- Условия применимости дисперсионного анализа выполняются

График значений первой компоненты по регионам



Пост-хок тест

```
TukeyHSD(aov(mod))
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = mod)

$region
                                  diff         lwr         upr     p adj
E Europe-Balkans           -0.83072574 -1.20005586 -0.46139563 0.0000122
Iberian-Balkans            -0.45413185 -0.93093498  0.02267129 0.0674003
Mediterranean-Balkans      -0.35545275 -0.83225588  0.12135038 0.2210025
Scandinavia-Balkans        -1.27403276 -1.66334089 -0.88472463 0.0000000
W Europe-Balkans           -1.29046656 -1.62761729 -0.95331583 0.0000000
Iberian-E Europe            0.37659390 -0.08404169  0.83722948 0.1498216
Mediterranean-E Europe      0.47527299  0.01463740  0.93590858 0.0407955
Scandinavia-E Europe       -0.44330702 -0.81263713 -0.07397690 0.0133492
W Europe-E Europe          -0.45974082 -0.77361106 -0.14587057 0.0021733
Mediterranean-Iberian       0.09867909 -0.45188574  0.64924392 0.9921236
Scandinavia-Iberian        -0.81990092 -1.29670405 -0.34309779 0.0003797
W Europe-Iberian           -0.83633472 -1.27159443 -0.40107500 0.0000987
Scandinavia-Mediterranean  -0.91858001 -1.39538314 -0.44177688 0.0000955
W Europe-Mediterranean     -0.93501381 -1.37027352 -0.49975409 0.0000229
W Europe-Scandinavia       -0.01643380 -0.35358453  0.32071693 0.9999856
```

Take-home messages

- Применение метода главных компонент (PCA):
  - снижение размерности данных
  - исследование связей между переменными
  - построение ординации объектов
  - создание комплексных переменных
- Терминология:
  - Собственные числа — вклад компонент в общую изменчивость
  - Факторные нагрузки — корреляции исходных переменных с компонентами — используются для интерпретации
  - Значения факторов — новые координаты объектов в пространстве уменьшенной размерности

Анализ главных компонент в геометрической морфометрии

Анализ морфометрических данных при помощи анализа главных компонент

- Классический подход к морфометрии
- Геометрическая морфометрия
- Эволюция формы

Вы сможете

- Проанализировать морфометрические данные корректно удалив влияние абсолютного размера
- Рассказать, что происходит во время обобщенного прокрустова анализа
- Проанализировать данные о координатах меток используя методы геометрической морфометрии
- Понимать, каким образом происходит отображение филогенетического древа в пространство форм

Классический подход к морфометрии

Классический подход к морфометрии

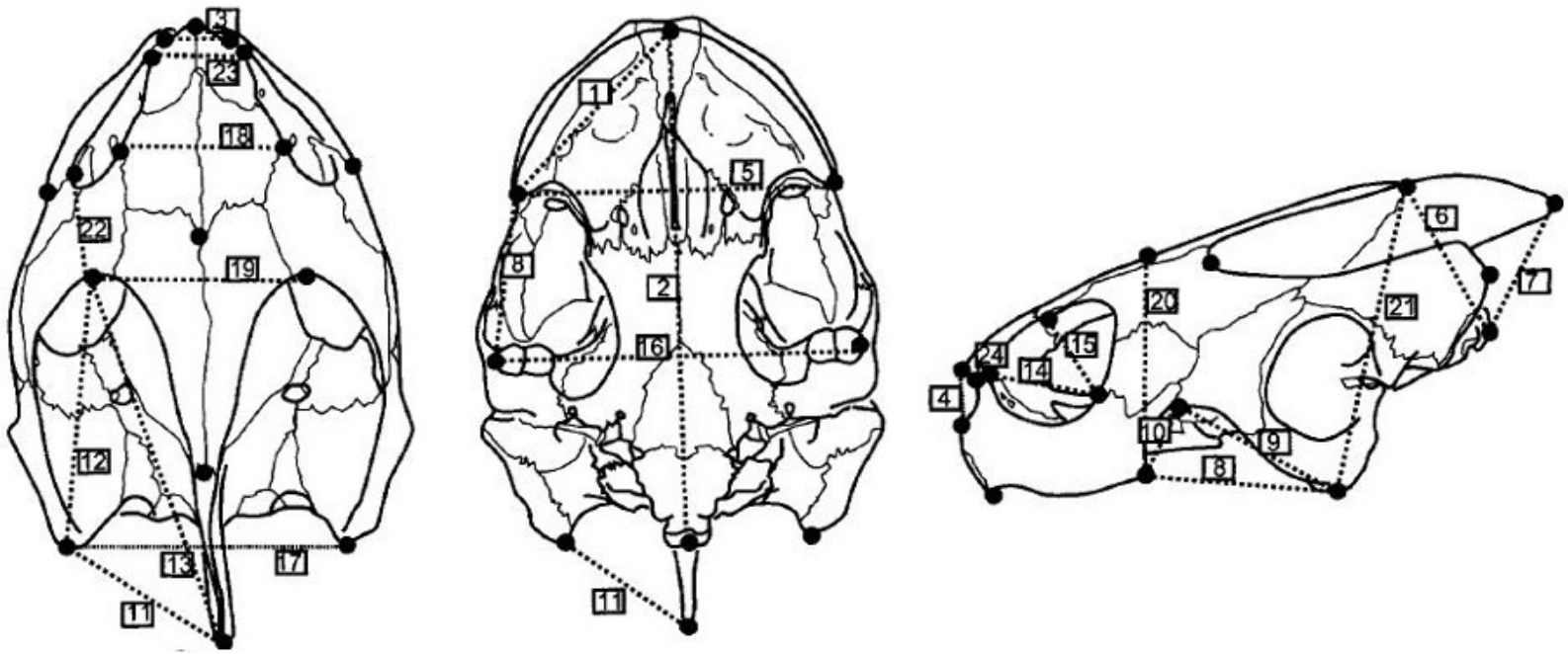
Для анализа формы различных структур анализируются расстояния между метками, а не их координаты.

Признаки сильно интегрированных структур, например частей скелета, лучше анализировать совместно друг с другом. Один из вариантов анализа — анализ главных компонент.

Пример: морфометрия черепах

Черепахи — единственные живые представители анапсид (череп не имеет височных окон). Морфология черепа важна для их систематики (Claude et al., 2004).

Данные — 24 разных измерения черепов черепах 122 ныне живущих пресноводных, морских и наземных видов и одного ископаемого.



Морфометрия черепах

Рис. 30.1 из Zuur et al. 2007

Читаем данные

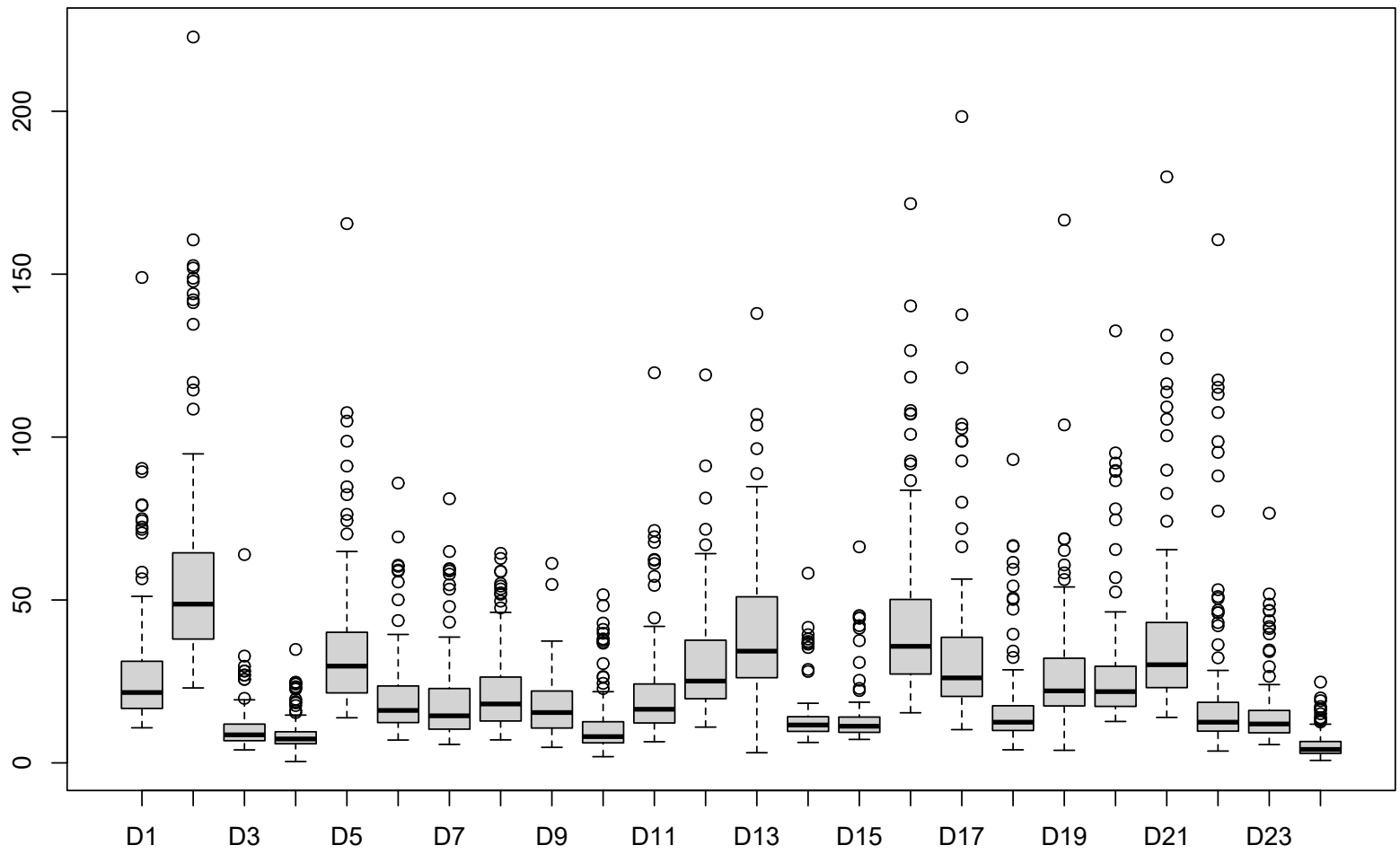
```
turt <- read.table("data/turtles.txt", header = TRUE)
turt$Environment3 <- factor(turt$Environment3, levels = c(0, 1, 2, 9),
          labels = c("Freshwater", "Terrestrial", "Marine", "Fossil"))
colnames(turt)
```

```
 [1] "nspecies"     "species_name" "Family"       "SuperFamily"  "Order"
 [6] "Environment"  "Environment3" "D1"           "D2"           "D3"
[11] "D4"           "D5"           "D6"           "D7"           "D8"
[16] "D9"           "D10"          "D11"          "D12"          "D13"
[21] "D14"          "D15"          "D16"          "D17"          "D18"
[26] "D19"          "D20"          "D21"          "D22"          "D23"
[31] "D24"
```

Данные из Zuur et al. 2007

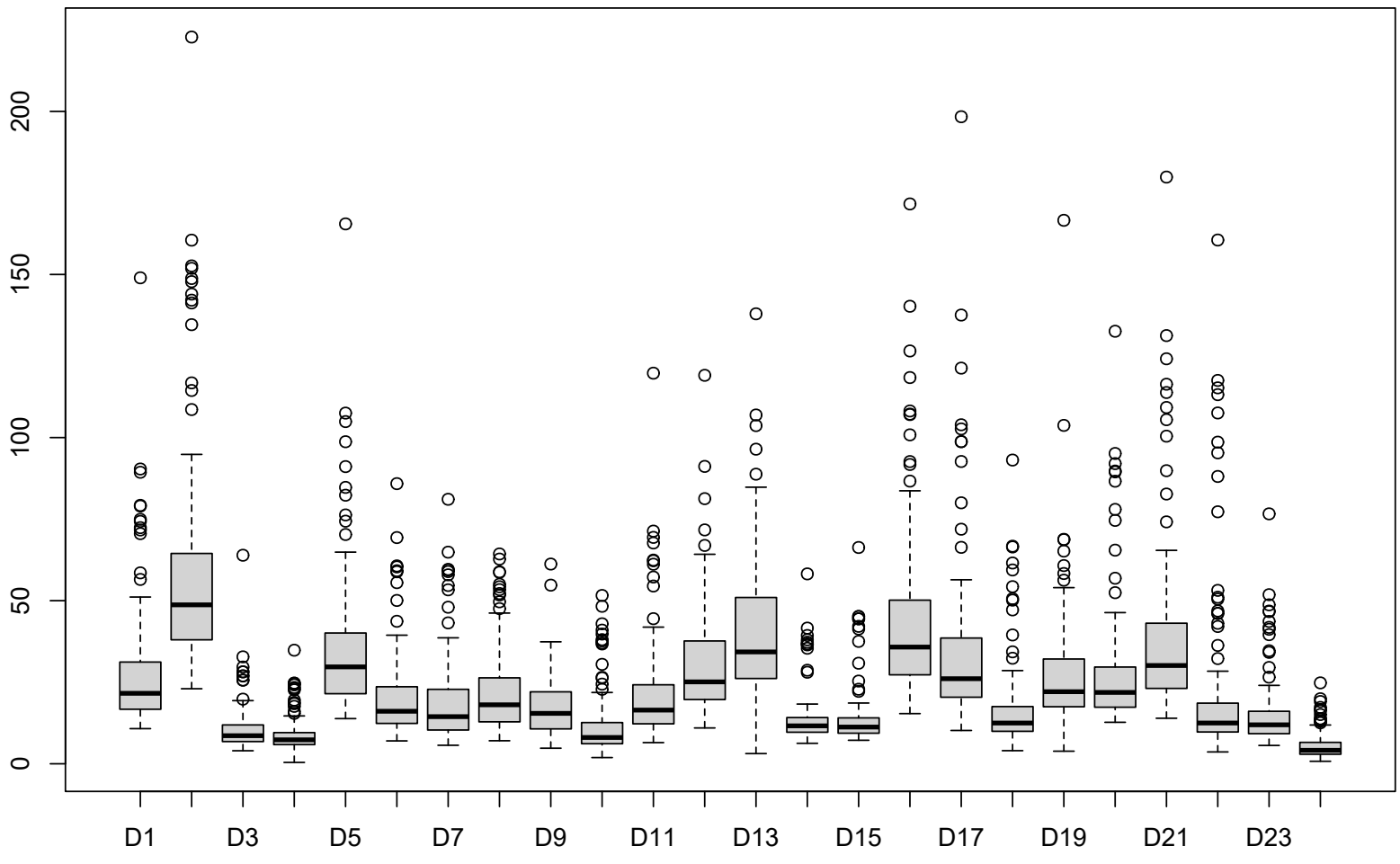
Чтобы понять, нужно ли стандартизовать исходные данные, построим боксплот

```
boxplot(x = turt[8:31])
```



Чтобы понять, нужно ли стандартизовать исходные данные, построим боксплот

```
boxplot(x = turt[8:31])
```



Делаем анализ главных компонент по стандартизованным данным

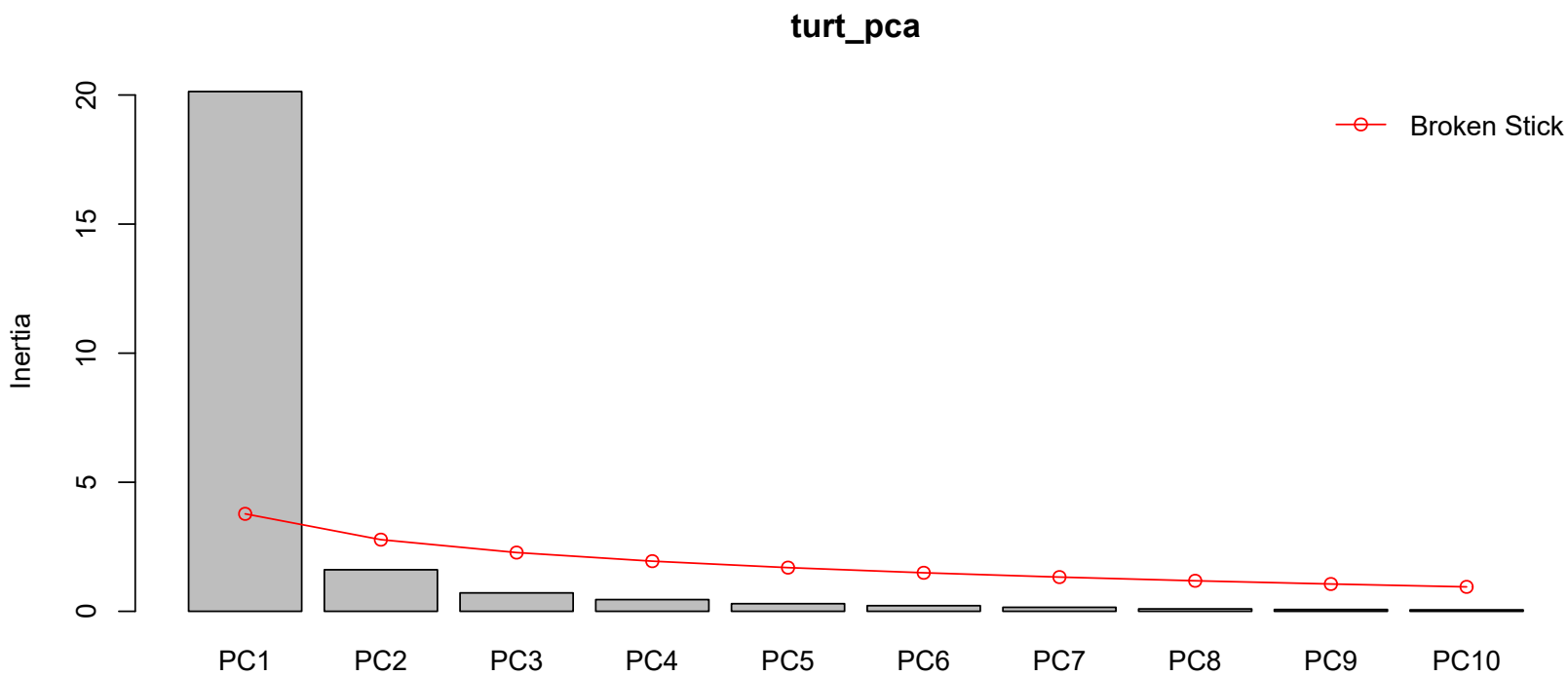
```
library(vegan)
turt_pca <- rda(turt[, 8:31], scale = TRUE)
```

Сколько компонент достаточно для описания данных?

```
eig <- eigenvals(turt_pca)[1:5]
eig*100/sum(eig) # доля объясненной изменчивости
```

```
      PC1       PC2       PC3       PC4       PC5
86.760221  6.936034  3.074255  1.956025  1.273465
```

```
screeplot(turt_pca, bstick = TRUE)
```

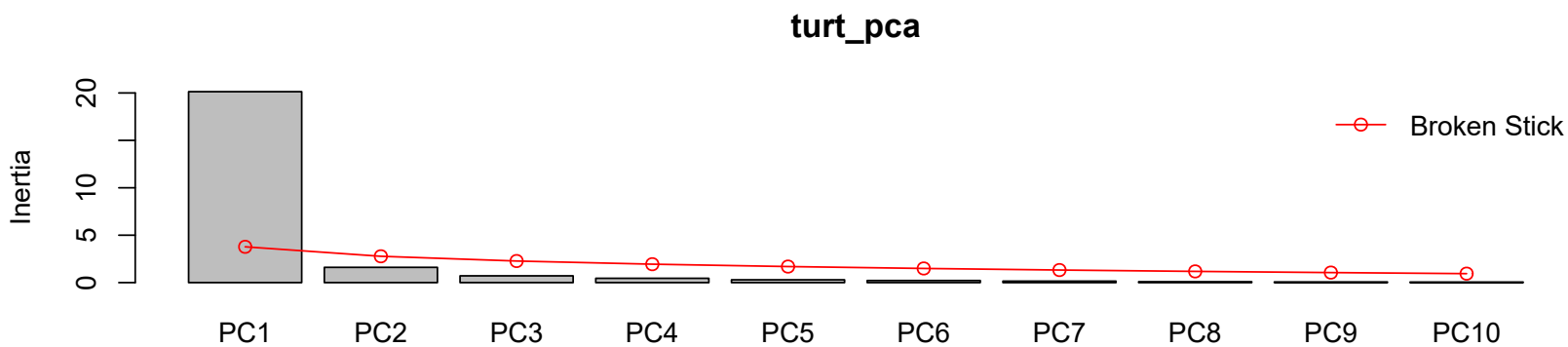


Сколько компонент достаточно для описания данных?

```
eig <- eigenvals(turt_pca)[1:5]
eig*100/sum(eig) # доля объясненной изменчивости
```

```
      PC1       PC2       PC3       PC4       PC5
86.760221  6.936034  3.074255  1.956025  1.273465
```

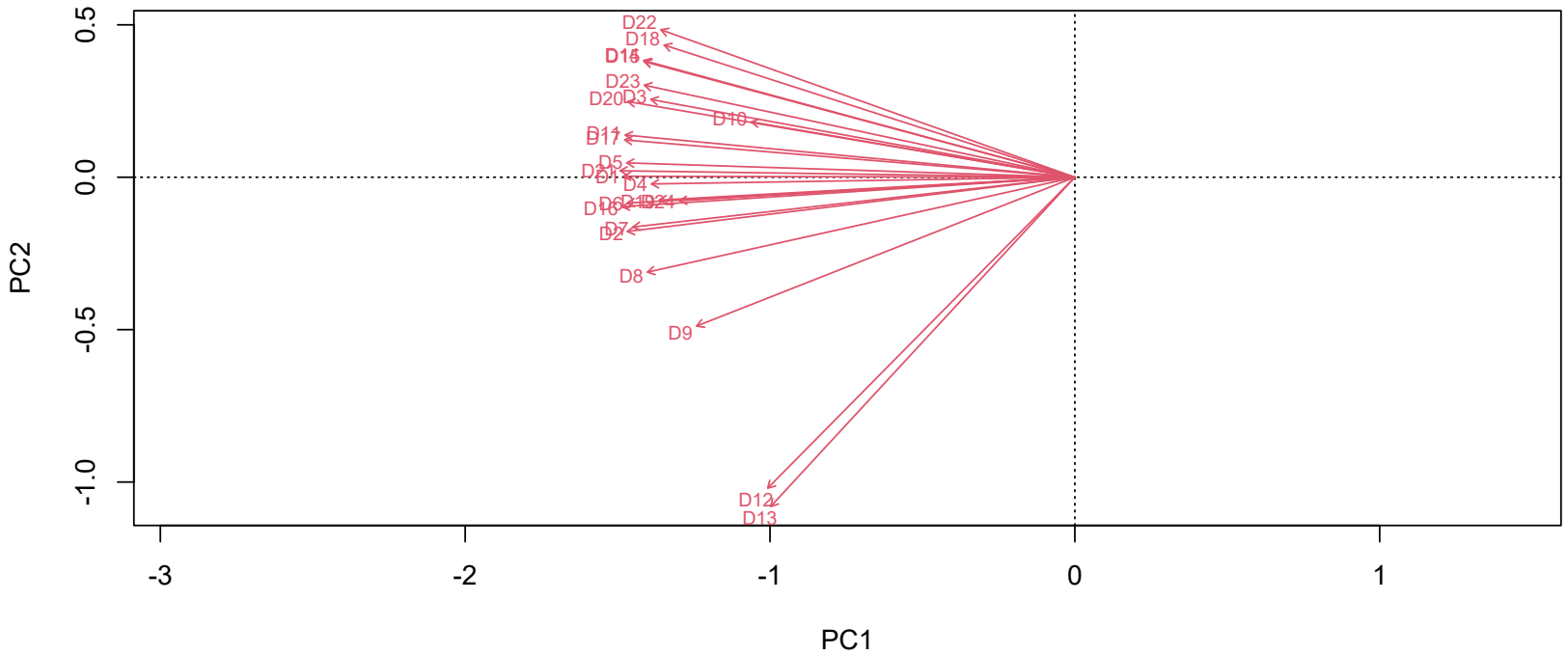
```
screeplot(turt_pca, bstick = TRUE)
```



- Первая компонента объясняет очень много, остальные - почти ничего. Одной компоненты достаточно?
- Нет! Не все так просто.

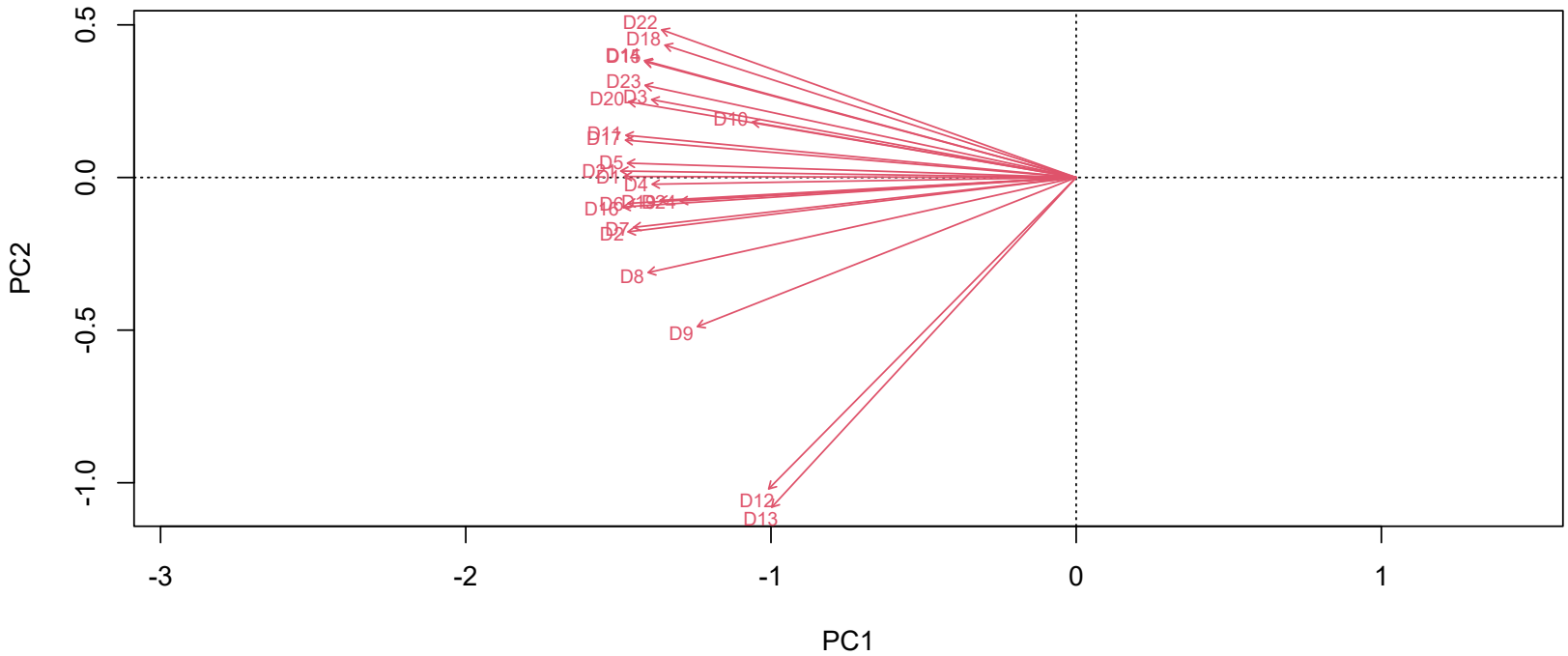
Что странного в этой картинке?

```
biplot(turt_pca, display = "species", scaling = 2)
```



- Как вы думаете, почему у всех переменных большие нагрузки по первой компоненте?
- Что отражает первая компонента?

При анализе сырых морфометрических данных первая компонента отражает размер объектов и, возможно, немножко — их форму



Классические способы избавиться от влияния размера

- использовать одну из исходных переменных как оценку “размера”: использовать в PCA остатки от регрессий исходных признаков от “размера”
- стандартизация исходных данных при помощи деления на величину “размера” для каждого образца (корень из суммы квадратов измерений)
- сделать двойное центрирование (логарифмированных) исходных данных
- и т.д. и т.п.

Двойное центрирование

Нам достаточно центрировать строки, т.к. столбцы будут центрированы автоматически в процессе анализа главных компонент.

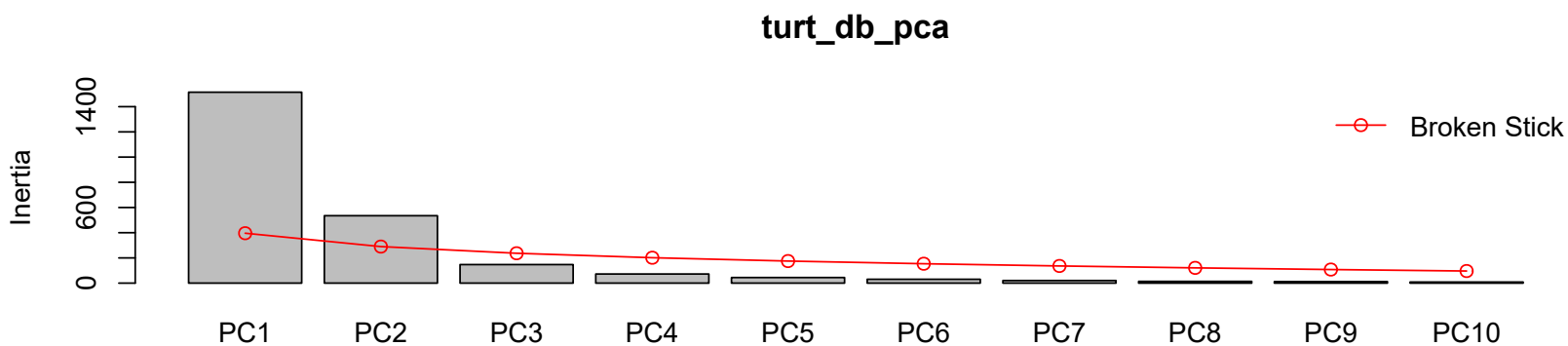
```
# Функция, которая может центрировать вектор
center <- function(x){
  x - mean(x, na.rm = TRUE)
}
# применяем эту функцию к каждой строке
dbcent <- t(apply(turt[, 8:31], 1, center))
# получившийся датафрейм пришлось транспонировать,
# поскольку apply() результаты от каждой строки
# возвращает в виде столбцов
```

После двойного центрирования большие собственные числа у нескольких компонент

```
turt_db_pca <- rda(dbcent)
eig_db <- eigenvals(turt_db_pca)[1:5]
eig_db*100/sum(eig_db)
```

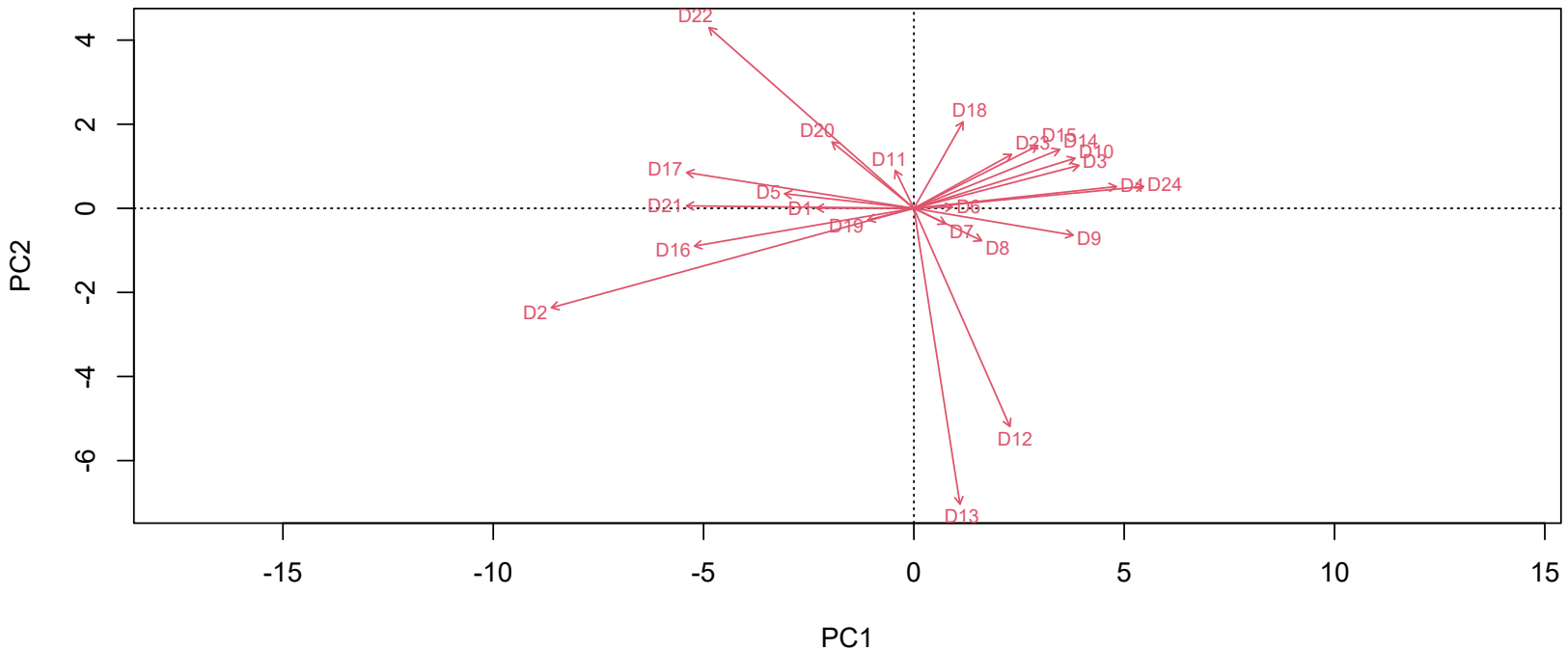
```
      PC1       PC2       PC3       PC4       PC5
65.477109 23.121299  6.361581  3.125069  1.914941
```

```
screeplot(turt_db_pca, bstick = TRUE)
```



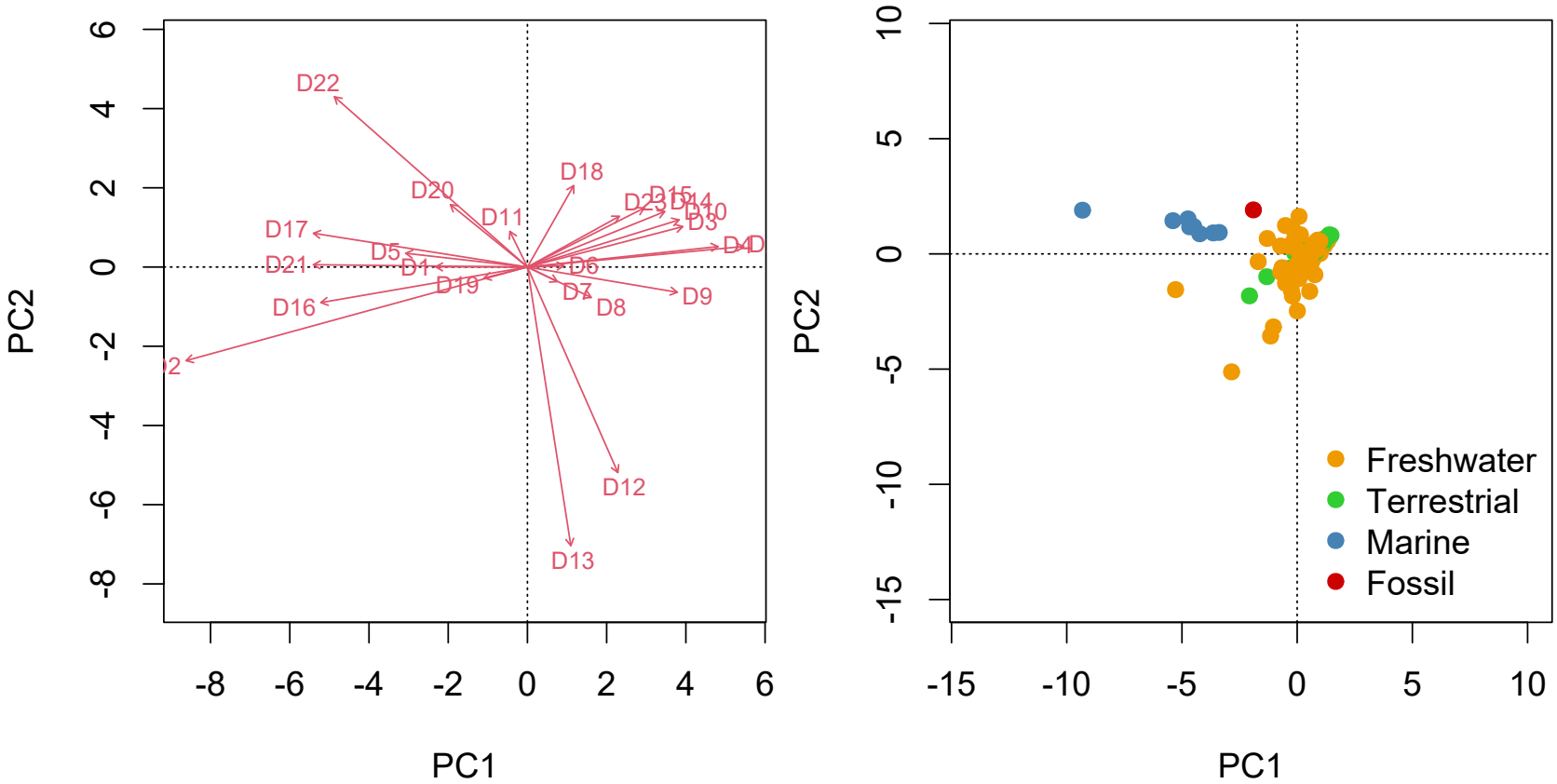
После двойного центрирования у переменных высокие нагрузки на несколько компонент, влияние размера удалено

```
biplot(turt_db_pca, display = "species", scaling = 2)
```



Интерпретируем как обычно: компонента отражает несколько признаков

Ординация черепах по морфометрии черепов (двойное центрирование данных)



- У пресноводных большие D12 и D13, и маленькая D2. У морских наоборот
- Ископаемая черепаха похожа на нынешних морских

Код для графика ординации черепах по морфометрии черепов

```
op <- par(mfrow = c(1, 2), mar = c(4, 4, 0.5, 0.5), cex = 1.3)
biplot(turt_db_pca, display = "species", scaling = 2)
# цвета для графика факторных координат
colvec <- c("orange2", "limegreen", "steelblue", "red3")
# пустой график
plot(turt_db_pca, type = "n", scaling = 1)
# точки, раскрашенные по уровням фактора turt$Environment3
points(turt_db_pca, display = "sites", scaling = 1, pch = 21,
       col = colvec[turt$Environment3], bg = colvec[turt$Environment3])
# легенда
legend("bottomright", legend = levels(turt$Environment3), bty = "n", pch =
↪  21,
       col = colvec, pt.bg = colvec)
par(op)
```

Но настоящие джедаи теперь анализируют координаты меток, а не расстояния между ними!

Геометрическая морфометрия

Пример: Форма головы Апалачских саламандр рода *Plethodon*

Plethodon jordani и *P. teyahalee* встречаются вместе и раздельно. В совместно обитающих популяциях меняется форма головы обоих видов. В разных группах популяций этот процесс параллельно приводит к одинаковым результатам. По-видимому, одной из причин параллельной эволюции может быть межвидовая конкуренция (Adams, 2004, 2010).

Plethodon jordani

Plethodon cf. *teyahalee*

Plethodon jordani — Jordan's Salamander by John P Clare on Flickr

Plethodon cf. *teyahalee* by squamatologist on Flickr

Морфометрия головы саламандр

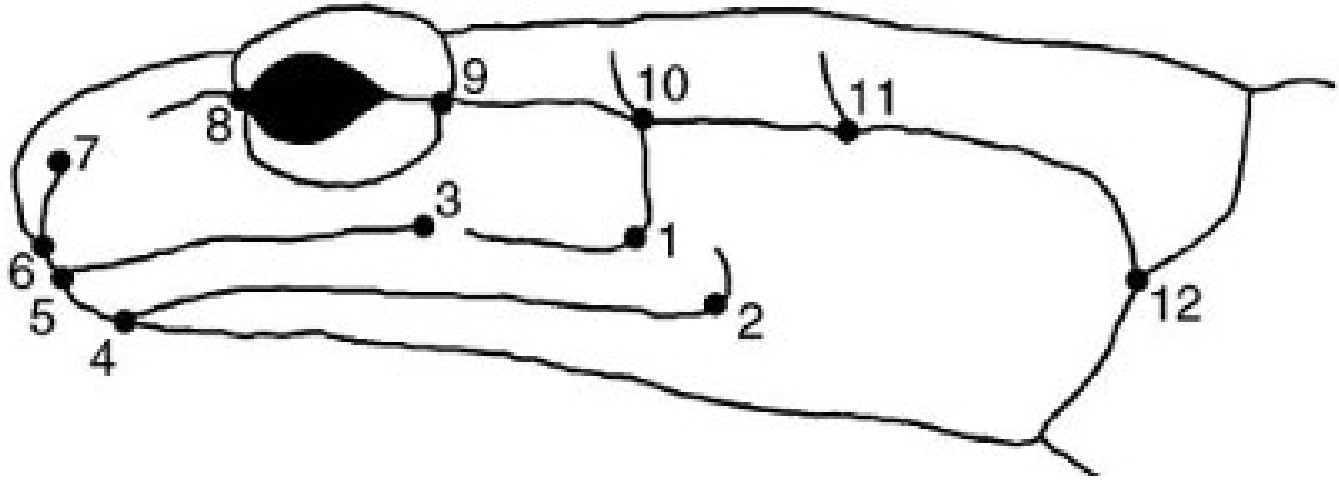


Схема измерений

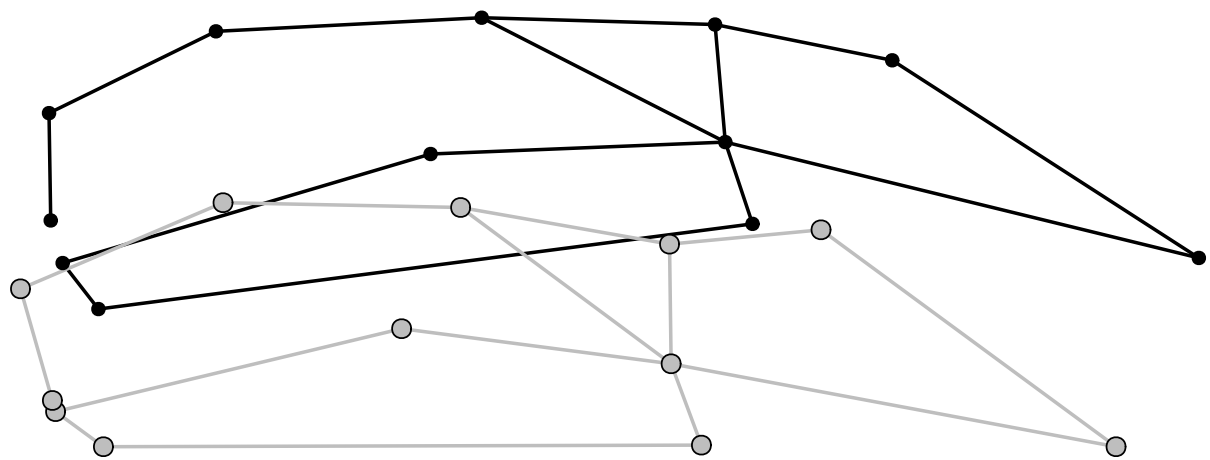
```
# install.packages("geomorph", dependencies = TRUE)
library(geomorph)
data(plethodon)
str(plethodon, vec.len = 2, give.attr = F)
```

```
List of 5
 $ land   : num [1:12, 1:2, 1:40] 8.89 9.27 ...
 $ links  : num [1:14, 1:2] 4 3 2 1 1 ...
 $ species: Factor w/ 2 levels "Jord","Teyah": 1 1 1 1 1 ...
 $ site   : Factor w/ 2 levels "Allo","Symp": 2 2 2 2 2 ...
 $ outline: num [1:3631, 1:2] 0.399 0.4 ...
```

Сырые морфометрические данные еще не выравнены

Все образцы разного размера и разной ориентации в пространстве. На этом графике — два образца для примера.

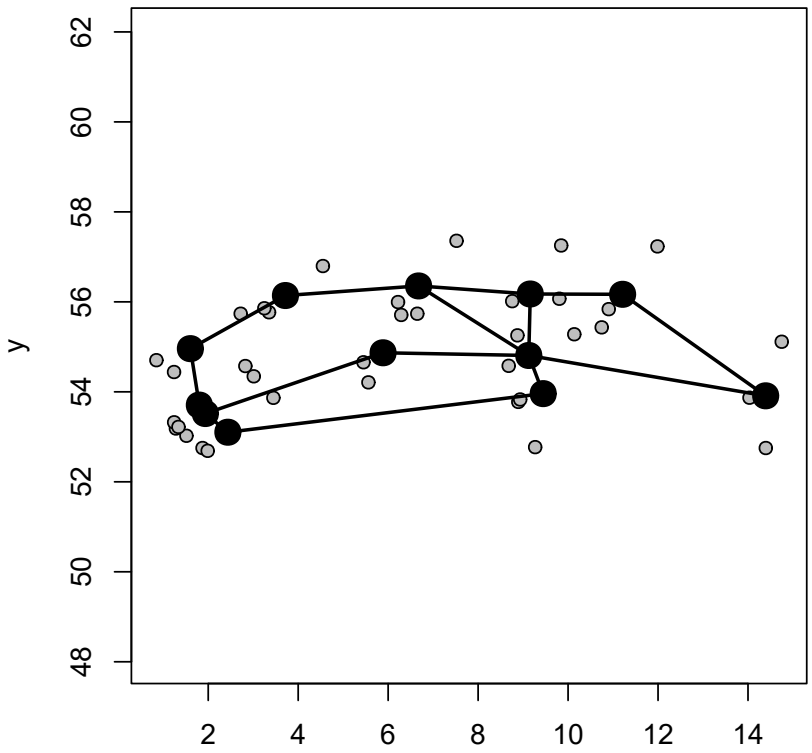
```
plotRefToTarget(plethodon$land[, , 1], plethodon$land[, ,10],
                method = "points", mag = 1,
                links = plethodon$links)
```

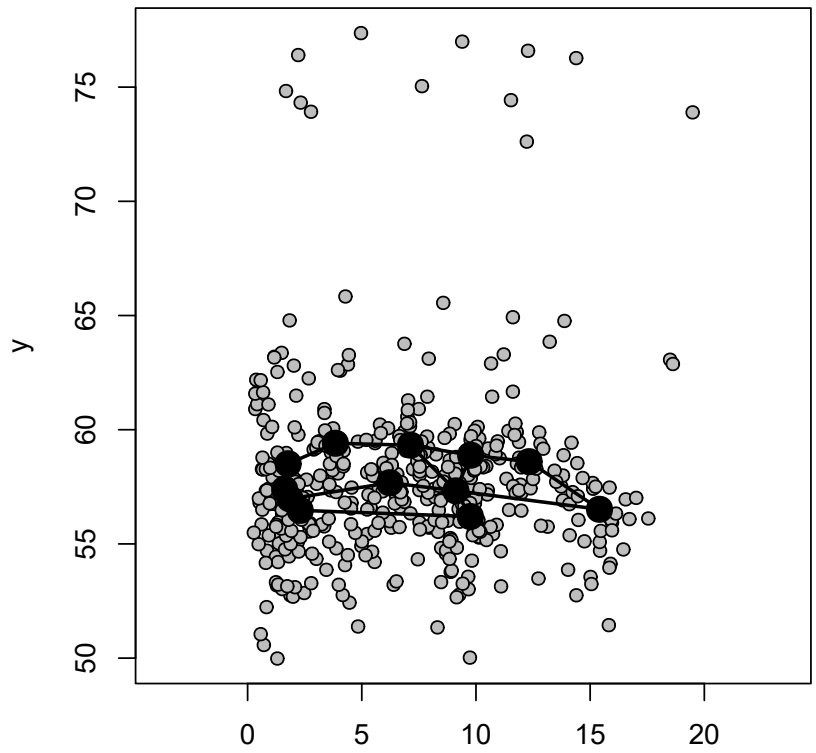


Если нарисовать не выравненные образцы, получится полная каша. Что делать?

Слева — три образца, справа — все. Жирные точки — центроиды соответствующих меток.

```
op <- par(mfrow = c(1, 2), mar = c(4, 4, 1, 1))
plotAllSpecimens(plethodon$land[, , 1:3], links=plethodon$links)
plotAllSpecimens(plethodon$land,links=plethodon$links)
par(op)
```





Геометрическая морфометрия

1. Влияние размера удаляется при помощи обобщенного прокрустова анализа (масштабирование, поворот и сдвиг координат)
2. Преобразованные координаты меток используются как признаки объектов (конкретных особей) в анализе главных компонент. Получается морфопространство. Главные компоненты отражают изменения формы.

- можно получить усредненную форму для любой группы выравненных координат
- можно сравнить форму любой особи со средней формой
- можно проследить изменение формы вдоль осей главных компонент

Прокрустов анализ

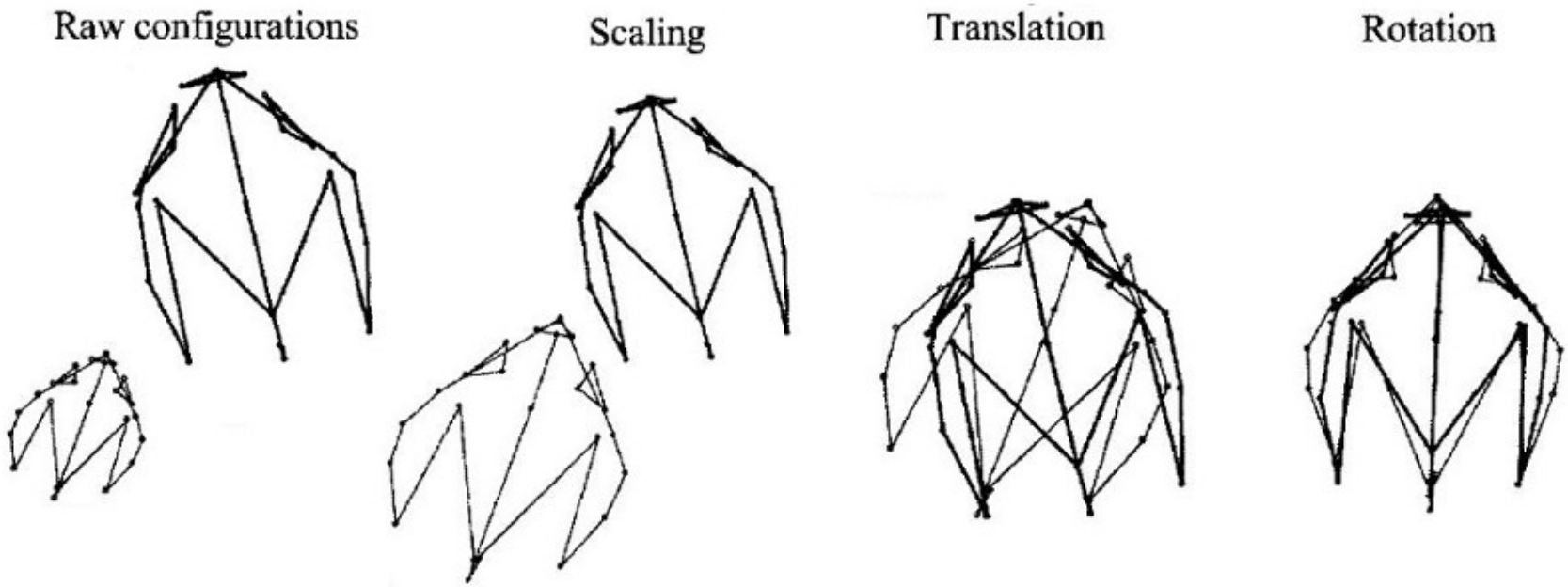
Прокрустово ложе

Тезей убивает разбойника Прокруста (источник https://mrpsmythopedia.wikispaces.com/Procrustes)

Шаг 1. Выравниваем данные при помощи обобщенного прокрустова анализа

Generalized Procrustes Analysis (GPA)

Минимизируем сумму квадратов расстояний между одноименными метками, меняя масштаб, поворачивая и сдвигая координаты. Вот как это выглядит на данных про черепах:

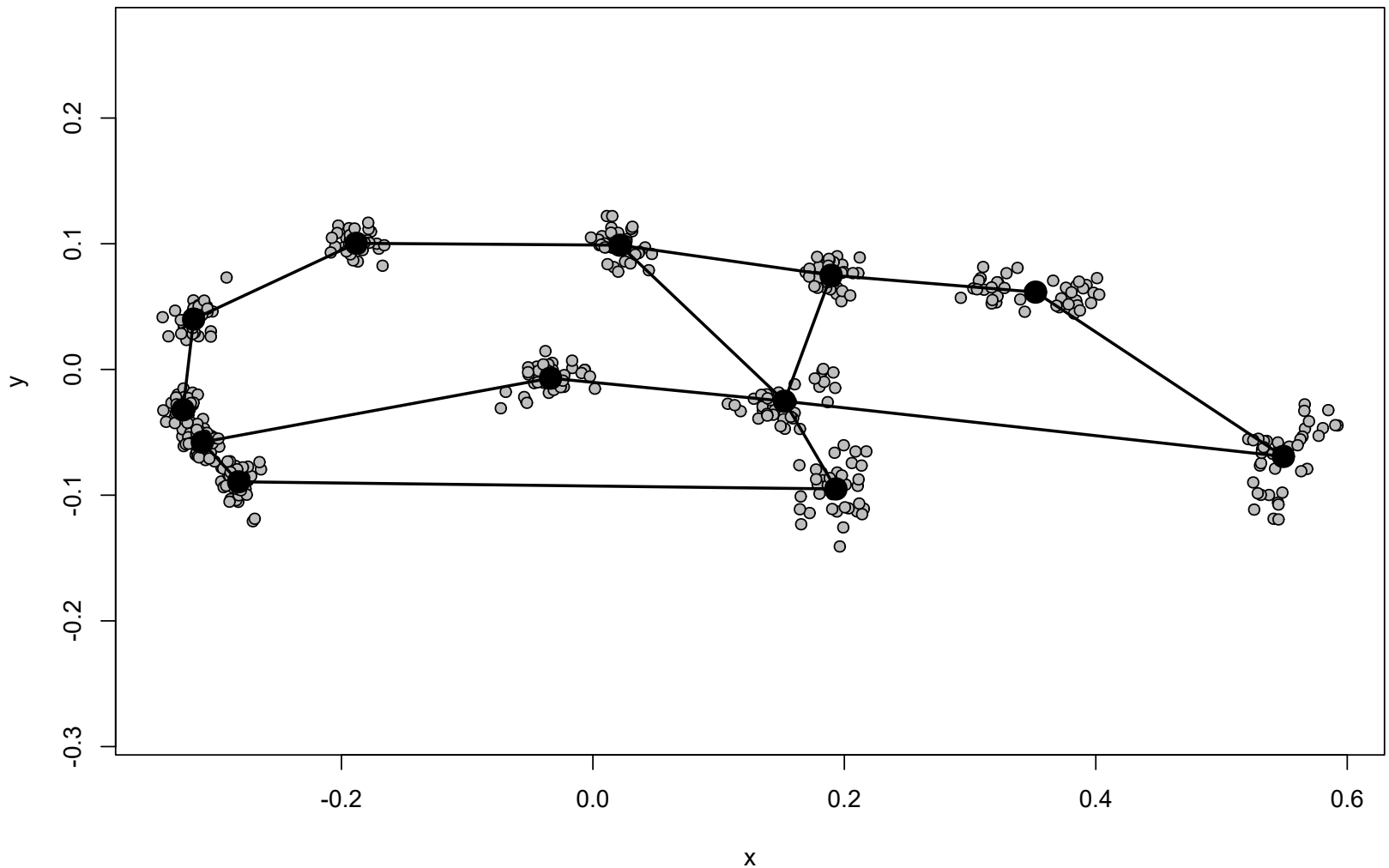


Прокрустово преобразование

Рис. 30.8 из Zuur et al. 2007 с изменениями

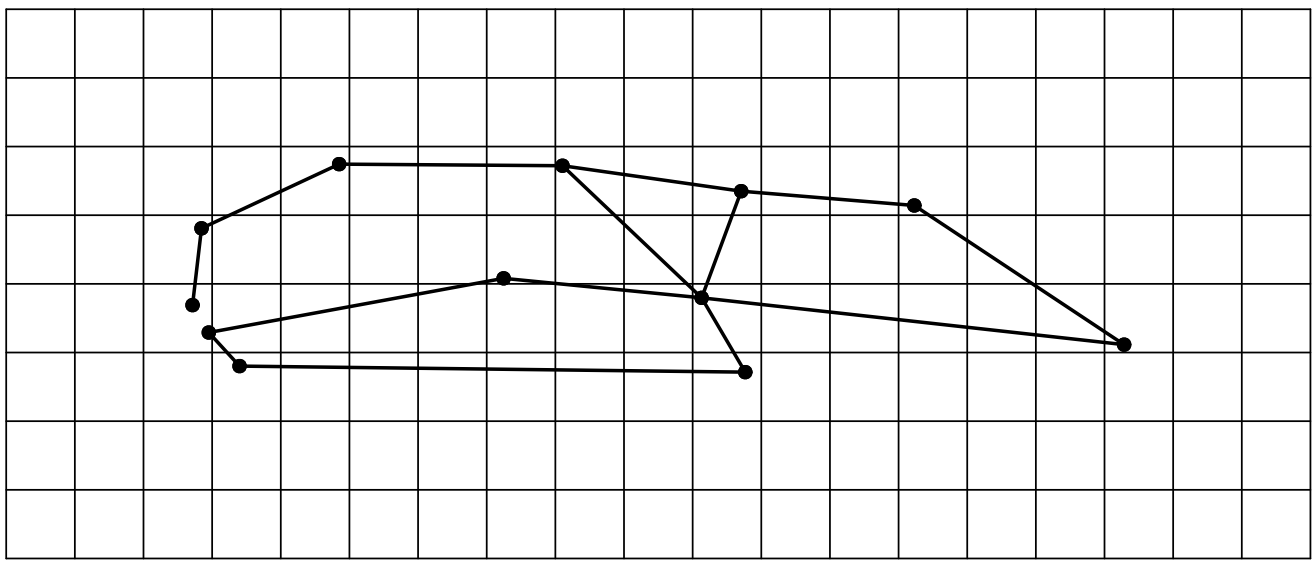
Выравниваем головы саламандр

```
gpa <- gpagen(plethodon$land, print.progress = FALSE)
plotAllSpecimens(gpa$coords,links=plethodon$links)
```



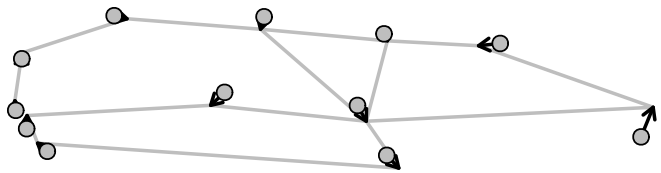
Усредненная форма

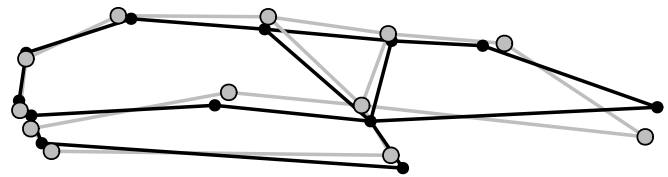
```
ref <- mshape(gpa$coords)
plotRefToTarget(ref, ref, method = "TPS", links = plethodon$links)
```

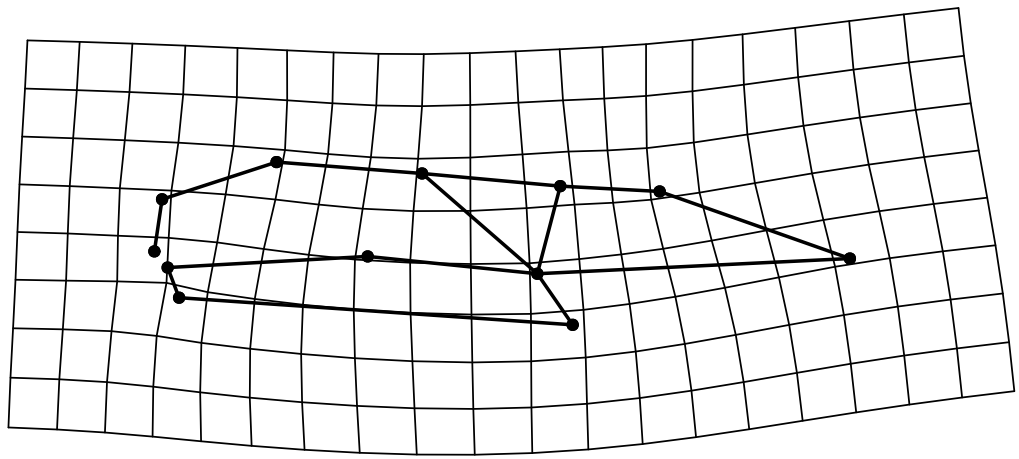


Можем посмотреть, как отличается любой из образцов от усредненной формы

Изменение формы можно представить графически несколькими способами







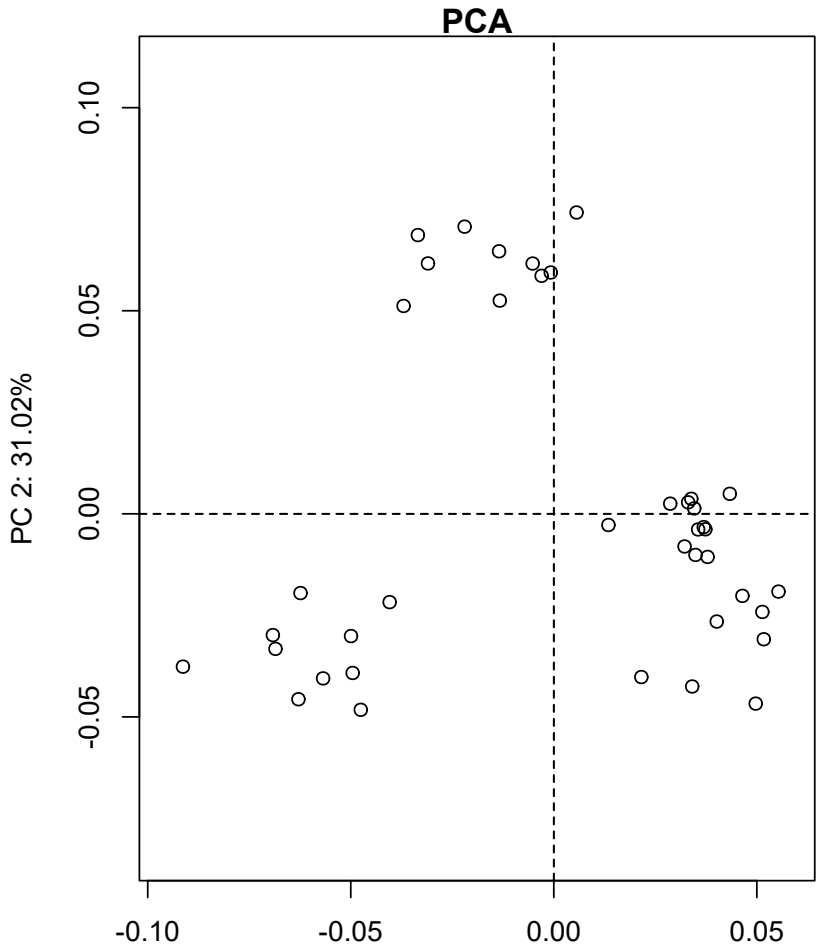
Код для графиков сравнения образцов с усредненной формой

```
# матрица, в которой хранится разметка общего графика
m <- matrix(data = c(1, 2,
                     3, 3),
            nrow = 2, ncol = 2, byrow = TRUE)
l <- layout(m, heights = c(1, 1), widths = c(1, 1))
# layout.show(l) # можно просмотреть разметку

# Графики
op <- par( mar = c(0, 0, 0, 0))
# 1) изменение конфигурации обозначено векторами
plotRefToTarget(ref, gpa$coords[, , 11],
                method = "vector", mag = 1,
                links = plethodon$links)
# 2) формы обозначены точками
plotRefToTarget(ref, gpa$coords[, , 11],
                method = "points", mag = 1,
                links = plethodon$links)
# 3) сплайн
plotRefToTarget(ref, gpa$coords[, , 11],
                method = "TPS", mag = 1,
                links = plethodon$links)
par(op)
```

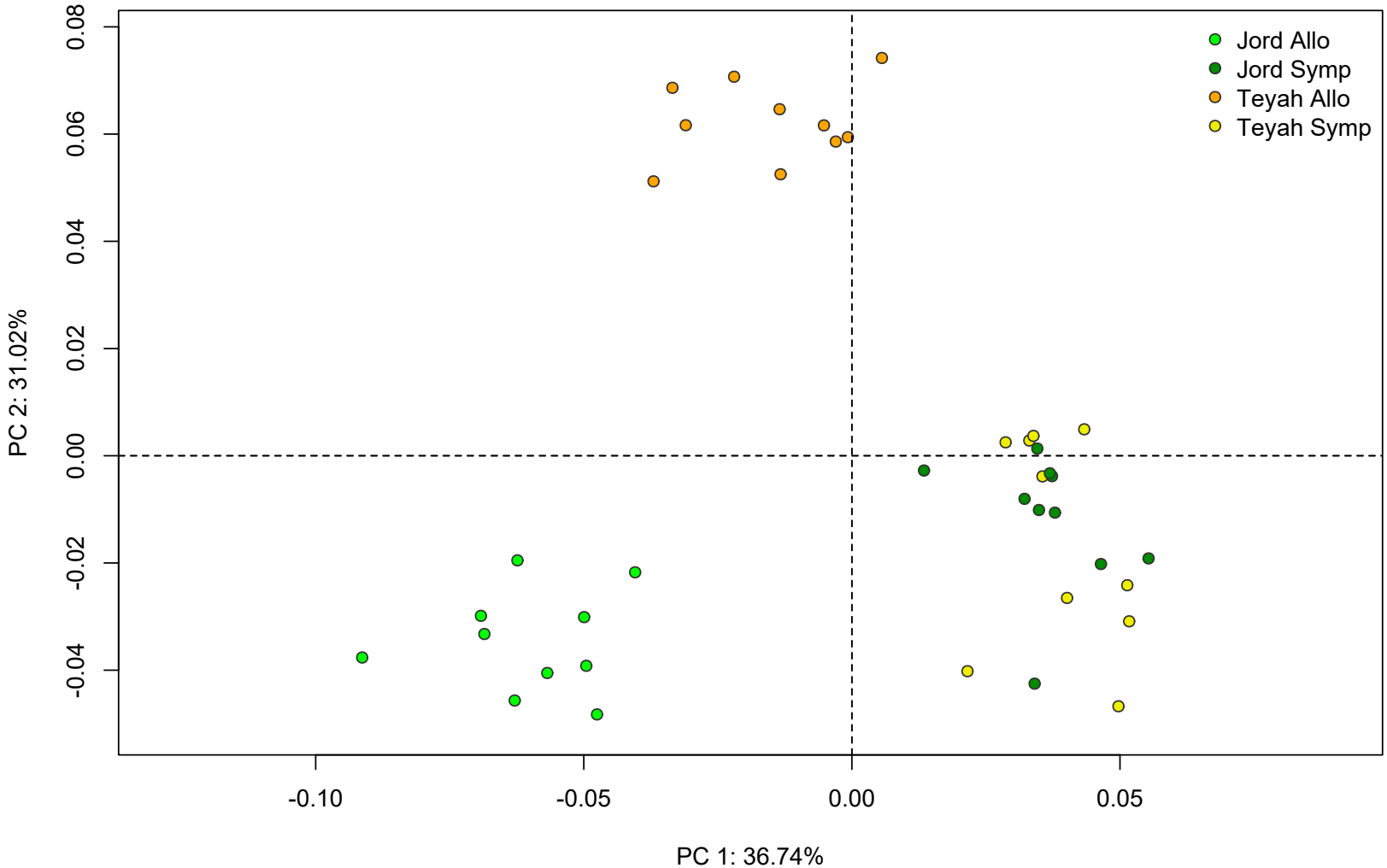
Шаг 2. Создаем морфопространство

Анализ главных компонент по координатам меток для выравненных образцов. Главные компоненты отражают изменения формы.

```
op <- par(mfrow = c(1, 2), mar = c(4, 4, 1, 1))
ord <- gm.prcomp(gpa$coords)
plot(ord, main = "PCA")
```



Можно раскрасить по группам



Код для графика ординации и для легенды

```
# группа должна быть фактором
gp <- as.factor(paste(plethodon$species, plethodon$site))

# задаем соответствие цветов уровням фактора
colvec <- c("Jord Allo" = "yellow2",
            "Jord Symp" = "orange",
            "Teyah Allo" = "green4",
            "Teyah Symp" = "green1")
# вектор цветов в порядке заданном фактором gp
colvec <- colvec[match(gp, names(colvec))]
# график
plot(ord, bg = colvec, pch = 21, col = "grey20")
# легенда
legend("topright", legend = levels(gp),
                   bty = "n", pch = 21,
                   col = "grey20",
                   pt.bg = levels(as.factor(colvec)))
par(op)
```

Доля объясненной изменчивости и факторные координаты

```
# Доля изменчивости объясненной 1-5 компонентами
expl <- round(ord$d[1:5]/sum(ord$d) * 100, 1)

# Факторные координаты по 1-5 компонентам
head(ord$x[, 1:5])
```

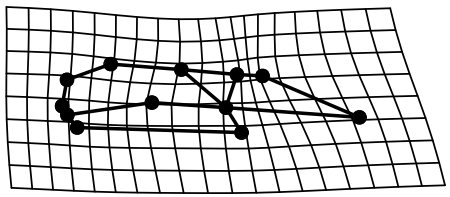
```
          Comp1      Comp2         Comp3        Comp4        Comp5
1 -0.0369930887 0.05118246 -0.0016971586 -0.003128881 -0.010935739
2 -0.0007493689 0.05942083  0.0001371682 -0.002768621 -0.008117767
3  0.0056004751 0.07419599 -0.0052612189 -0.005034502 -0.002747104
4 -0.0134808326 0.06463958 -0.0458436274 -0.007887336  0.009817034
5 -0.0334696064 0.06863518  0.0136292227  0.007359383  0.022347215
6 -0.0052144953 0.06162541 -0.0299332836 -0.005753115 -0.024060256
```

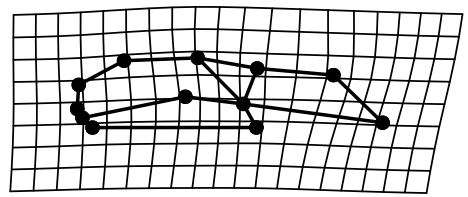
Чтобы легко рисовать изменения формы вдоль главной компоненты, нам понадобится функция

```
plot_shape_change <- function(ord, ref_shape, PC,
                              horiz = TRUE,
                              gridPars = NULL, ...){
  if(horiz){
    op <- par(mfrow = c(1, 2), mar = c(0, 0 , 0, 0))
    plotRefToTarget(M1 = ref_shape, M2 = ord$shapes[[PC]]$min,
                    gridPars = gridPars,  ...)
    plotRefToTarget(M1 = ref_shape, M2 = ord$shapes[[PC]]$max,
                    gridPars = gridPars, ...)
    par(op)
    } else {
     op <- par(mfrow = c(2, 1), mar = c(0, 0 , 0, 0))
     plotRefToTarget(M1 = ref_shape, M2 = ord$shapes[[PC]]$max,
                     gridPars = gridPars,  ...)
     plotRefToTarget(M1 = ref_shape, M2 = ord$shapes[[PC]]$min,
                     gridPars = gridPars, ...)
     par(op)
    }
}
```

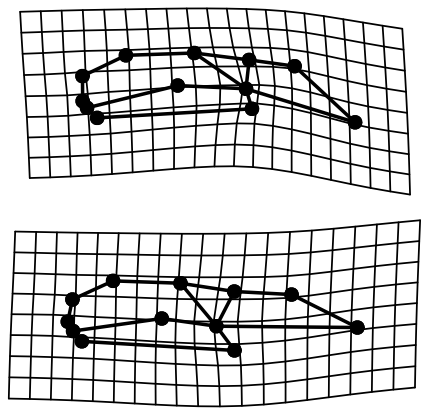
Изменение формы вдоль главных компонент относительно средней формы

```
plot_shape_change(ord, ref_shape = gpa$consensus, PC = 1,
                  links = plethodon$links, method = "TPS")
```

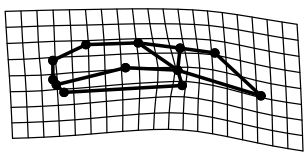


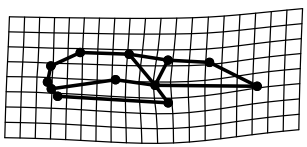


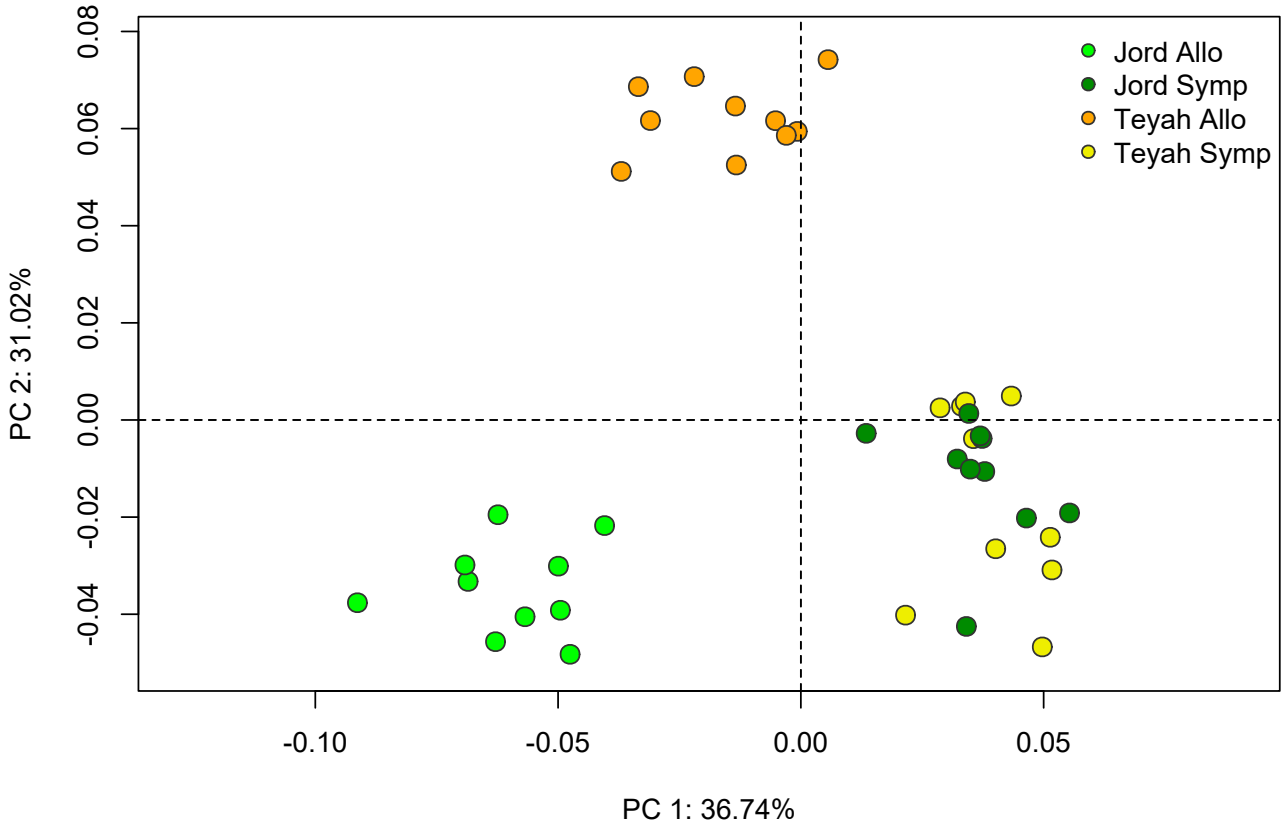
```
plot_shape_change(ord, ref_shape = gpa$consensus, PC = 2,
                  links = plethodon$links, method = "TPS", horiz = FALSE)
```

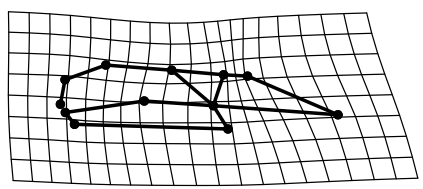


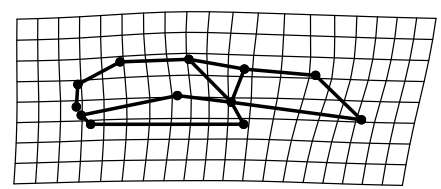
Можно нарисовать одновременно изменение формы вдоль обеих компонент и ординацию











Код для графика

```
my_gridPar <- gridPar(tar.pt.size = 0.6, grid.lwd = 0.7)
library(cowplot)
library(gridGraphics)
gg_pca <- plot_grid(
  # Изменение формы вдоль PC2
  ~ plot_shape_change(ord, ref_shape = gpa$consensus, PC = 2,
                      horiz = FALSE, links = plethodon$links,
                      method = "TPS", gridPars = my_gridPar),
  # Ординация
  ~ {plot(ord, bg = colvec, pch = 21, col = "grey20", cex = 1.5)
    legend("topright", legend = levels(gp),  bty = "n",
           pch = 21, col = "grey20",
           pt.bg = levels(as.factor(colvec)))},
  # пустой график
  NULL,
  # Изменение формы вдоль PC1
  ~ plot_shape_change(ord, ref_shape = gpa$consensus, PC = 1,
                      links = plethodon$links,
                      method = "TPS", gridPars = my_gridPar),
  # Параметры размещения
  ncol = 2, rel_heights = c(5, 1), rel_widths = c(1, 4)
)

gg_pca
```

Эволюционные изменения формы

Фило-морфо пространство

Если у вас есть данные о средних формах для каждого вида и данные о филогении (из любого источника), то можно изобразить эволюционные изменения формы

Этапы:

1. Выравнивание средних форм для таксонов при помощи обобщенного прокрустова анализа
2. Ординация таксонов при помощи анализа главных компонент
3. Поиск анцестральных состояний количественных признаков (форм) методом максимального правдоподобия
4. Наложение филогенетического дерева и анцестральных форм на график ординации

Фило-морфопространство саламандр рода Plethodon

P. serratus, *P. cinereus*, *P. shenandoah*, *P. hoffmani*, *P. virginia*, *P. nettingi*, *P. hubrichti*, *P. electromorphus*, *P. richmondi*

```
data(plethspecies)
str(plethspecies, vec.len = 2, give.attr = F)
```

```
List of 2
 $ land: num [1:11, 1:2, 1:9] 0.217 0.259 ...
 $ phy :List of 4
  ..$ edge       : int [1:16, 1:2] 10 10 11 12 12 ...
  ..$ Nnode      : int 8
  ..$ tip.label  : chr [1:9] "P_serratus" "P_cinereus" ...
  ..$ edge.length: num [1:16] 15.17 3.84 ...
```

Выравниваем средние формы для видов

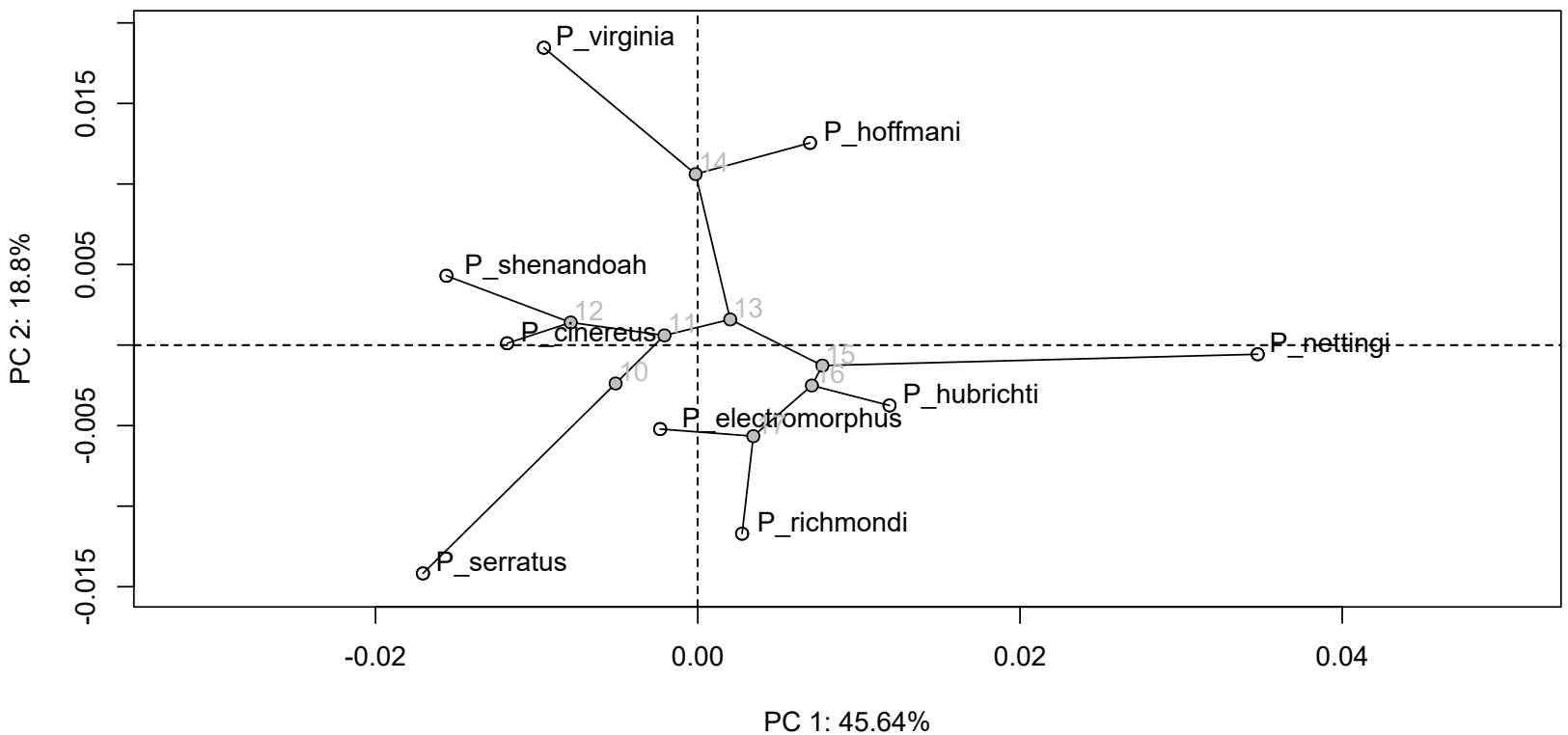
```
species_gpa <- gpagen(plethspecies$land) #GPA-alignment
```

```
Performing GPA

  |
  |                                                                      |   0%
  |
  |==================                                                    |  25%
  |
  |===================================                                   |  50%
  |
  |======================================================================| 100%

Making projections... Finished!
```

Наложение филогенетического дерева и анцестральных форм на график PCA ординации

Филоморфопространство

```
pca_with_phylo <- gm.prcomp(species_gpa$coords, phy = plethspecies$phy)
plot(pca_with_phylo, phylo = TRUE)
```



Take-home messages

- Классический подход к морфометрии
  - анализируют расстояния между метками
  - для корректного анализа необходимо удалить влияние размера и оставить форму, но сделать это корректно почти невозможно
- Геометрическая морфометрия
  - анализируют координаты меток
  - различные конфигурации выравнивают при помощи обобщенного прокрустова анализа
  - преобразованные координаты точек используют в анализе главных компонент
  - чтобы визуализировать эволюцию форм, можно наложить филогенетическое древо на ординацию

Что почитать

- Bookstein, F.L., 2003. Morphometric Tools for Landmark Data Geometry and Biology. Cambridge University Press.
- Borcard, D., Gillet, F., Legendre, P., 2011. Numerical ecology with R. Springer.
- Claude, J., 2008. Morphometrics With R. Springer.
- GEOL G562 - Geometric Morphometrics [WWW Document], n.d. URL http://www.indiana.edu/~g562/PBDB2013/ (accessed 4.1.15).
- Legendre, P., Legendre, L., 2012. Numerical ecology. Elsevier.
- Oksanen, J., 2011. Multivariate analysis of ecological communities in R: vegan tutorial. R package version 2–0.
- The Ordination Web Page http://ordination.okstate.edu/ (accessed 05.04.17).
- Quinn, G.G.P., Keough, M.J., 2002. Experimental design and data analysis for biologists. Cambridge University Press.
- Zelditch, M., Swiderski, D.L., Sheets, D.H., Fink, W.L., 2004. Geometric Morphometrics for Biologists. Academic Press.
- Zuur, A.F., Ieno, E.N., Smith, G.M., 2007. Analysing ecological data. Springer.