Анализ главных компонент

Математические методы в зоологии - на R, осень 2014

Марина Варфоломеева

Знакомимся с ординацией на примере метода главных компонент

- Снижение размерности многомерных данных
- Анализ главных компонент в R

Вы сможете

- Проводить анализ главных компонент
- Снижать размерность данных, отбирая меньшее число главных компонент
- Оценивать долю объясненной изменчивости
- Интерпретировать компоненты по факторным нагрузкам
- Строить ординацию объектов в пространстве главных компонент
- Извлекать значения факторов объектов для дальнейшего использования с другими видами анализов

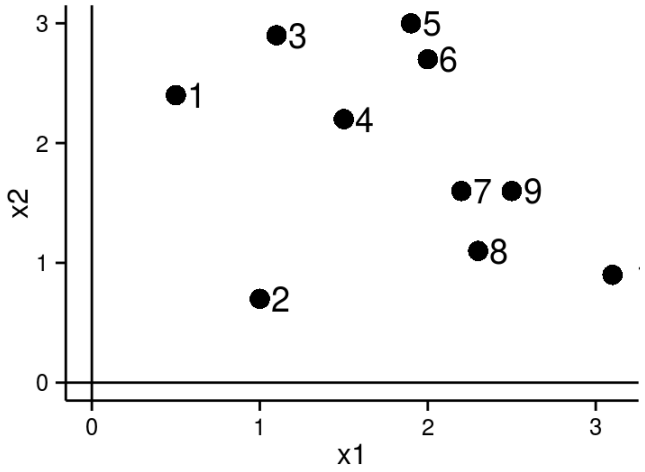
Снижение размерности многомерных данных

Анализ главных компонент - способ снижения размерности

Многомерные исходные данные

В этом примере для простоты - двумерные

```
##     x1  x2
## 1  0.5 2.4
## 2  1.0 0.7
## 3  1.1 2.9
## 4  1.5 2.2
## 5  1.9 3.0
## 6  2.0 2.7
## 7  2.2 1.6
## 8  2.3 1.1
## 9  2.5 1.6
## 10 3.1 0.9
```

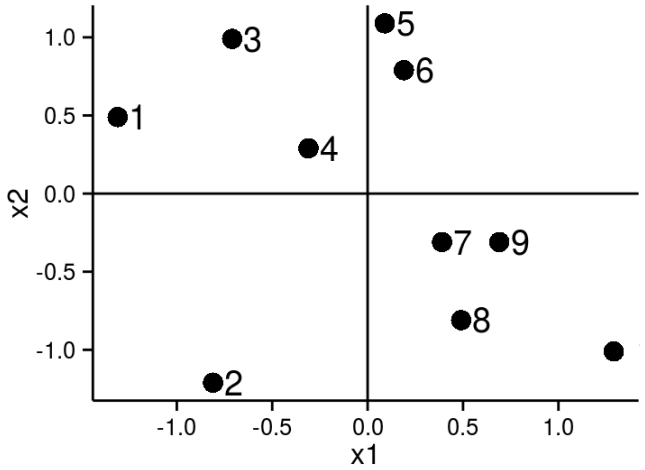


Центрирование

Если из каждого значения переменной вычесть среднее значение этой переменной, то центр координат переместится в точку $(\bar{x}_1, \bar{x}_2)$

Центрированные данные:

```
##       x1    x2
## 1  -1.31  0.49
## 2  -0.81 -1.21
## 3  -0.71  0.99
## 4  -0.31  0.29
## 5   0.09  1.09
## 6   0.19  0.79
## 7   0.39 -0.31
## 8   0.49 -0.81
## 9   0.69 -0.31
## 10  1.29 -1.01
```



Матрица ковариаций между признаками

Исходные данные:

```
##       x1    x2
## 1  -1.31  0.49
## 2  -0.81 -1.21
## 3  -0.71  0.99
## 4  -0.31  0.29
## 5   0.09  1.09
## 6   0.19  0.79
## 7   0.39 -0.31
## 8   0.49 -0.81
## 9   0.69 -0.31
## 10  1.29 -1.01
```

Матрица ковариаций:

```
##        x1     x2
## x1  0.617 -0.249
## x2 -0.249  0.717
```

- описывает совместное варьирование нескольких переменных
- по диагонали - дисперсии признаков
- выше и ниже диагонали - ковариации признаков друг с другом

Матрицу ковариаций можно представить в виде собственных векторов и собственных чисел

Матрица ковариаций:

```
##        x1     x2
## x1  0.617 -0.249
## x2 -0.249  0.717
```

Собственные числа:

- используются для оценки вклада главных компонент в общую изменчивость
- дисперсия вдоль собственных векторов пропорциональна их собственным числам

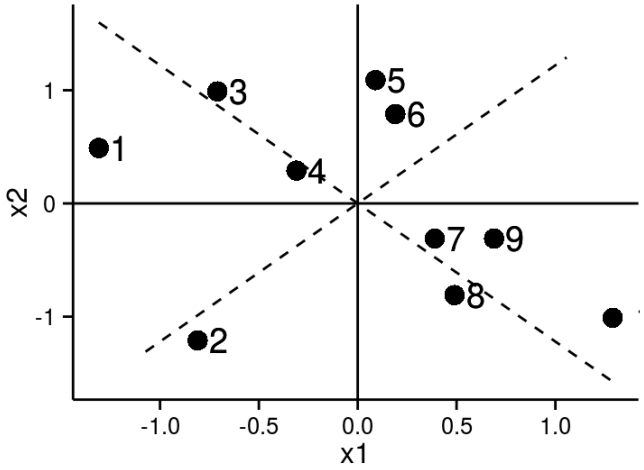
```
## [1] 0.921 0.413
```

Собственные вектора:

- задают направление осей главных компонент
- перпендикулярны друг другу
- вдоль первого - максимальная дисперсия данных, вдоль следующего - максимальная дисперсия из оставшейся

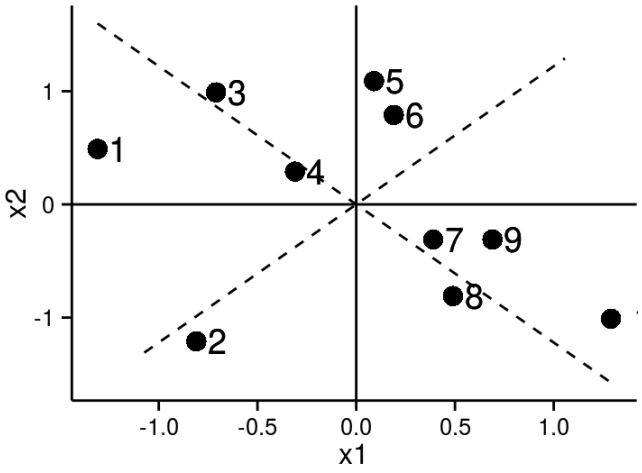
```
##        [,1]   [,2]
## [1,] -0.634 -0.774
## [2,]  0.774 -0.634
```

С помощью собственных векторов и собственных чисел можно найти в пространстве признаков новые оси, вдоль которых будет максимальный разброс точек.

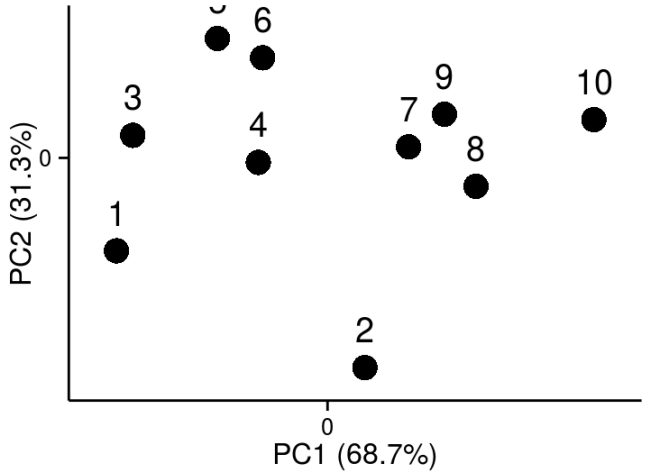


Можно найти новые координаты точек в получившемся новом пространстве

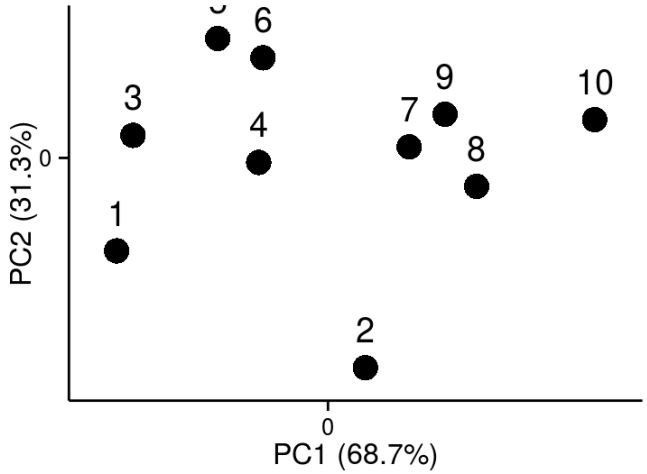
Объекты и оси в пространстве исходных признаков:



Объекты в пространстве новых осей (главных компонент):



На графике ординации изображено новое пространство



По собственным числам судим о доле изменчивости, объясненной новыми направлениями осей (компонентами)

- PC1 - больше всего изменчивости
- PC2 - то, что осталось

По новым координатам судим о близости объектов

По факторным нагрузкам исходных переменных на компоненты интерпретируем новые направления

Анализ главных компонент в R

Пример: Морфометрия поссумов

Possum by Hasitha Tudugalle on Flickr https://www.flickr.com/photos/hasitha_tudugalle/6037880962

Данные Lindenmayer et al. (1995)

Знакомимся с данными

```
library(DAAG)
data(possum)
colnames(possum)
```

```
##  [1] "case"     "site"     "Pop"      "sex"      "age"      "hdlngth"
##  [7] "skullw"   "totlngth" "taill"    "footlgth" "earconch" "eye"
## [13] "chest"    "belly"
```

```
sum(is.na(possum))
```

```
## [1] 3
```

```
possum[!complete.cases(possum), ]
```

```
##      case site Pop sex age hdlngth skullw totlngth taill footlgth
## BB36   41    2 Vic   f   5    88.4   57.0       83  36.5       NA
## BB41   44    2 Vic   m  NA    85.1   51.5       76  35.5     70.3
## BB45   46    2 Vic   m  NA    91.4   54.4       84  35.0     72.8
##      earconch  eye chest belly
## BB36     40.3 15.9  27.0  30.5
## BB41     52.6 14.4  23.0  27.0
## BB45     51.2 14.4  24.5  35.0
```

```r
# поссумы из разных сайтов
table(possum$site)
```

```
##
##  1  2  3  4  5  6  7
## 33 13  7  7 13 13 18
```

```r
# поссумы из 2 популяций
table(possum$Pop)
```

```
##
##   Vic other
##    46    58
```

```
# половой состав выборок из разных сайтов
with(possum, table(sex, site, Pop))

## , , Pop = Vic
##
##    site
## sex  1  2  3  4  5  6  7
##   f 19  5  0  0  0  0  0
##   m 14  8  0  0  0  0  0
##
## , , Pop = other
##
##    site
## sex  1  2  3  4  5  6  7
##   f  0  0  3  2  6  4  4
##   m  0  0  4  5  7  9 14
```

```r
# В исходных данных сайты закодированы цифрами
unique(possum$site)
```

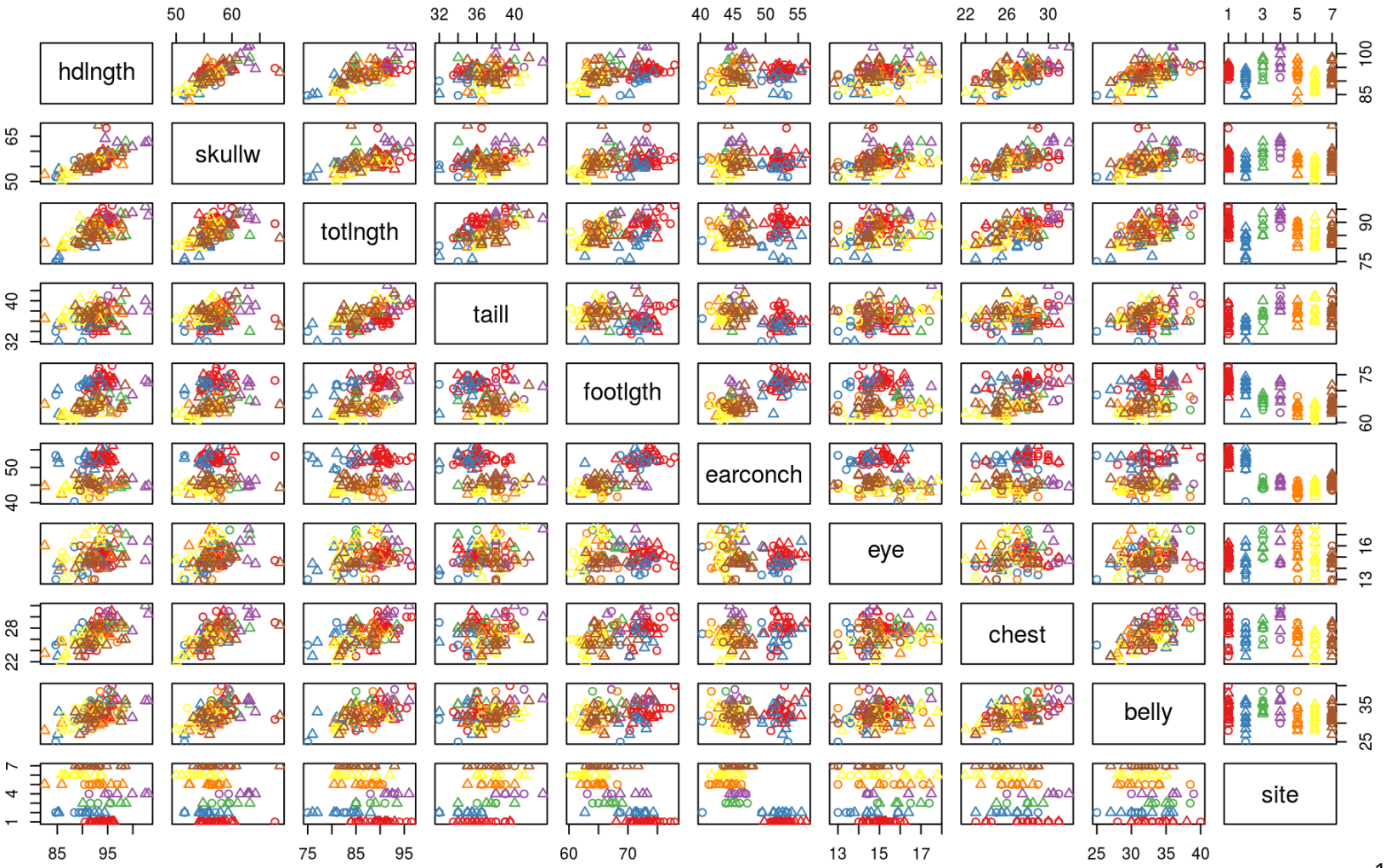
```
## [1] 1 2 3 4 5 6 7
```

```r
# Добавим названия сайтов
possum$site <- factor(possum$site, levels = 1:7, labels = c("Cambarville",
    "Bellbird", "Whian Whian", "Byrangery", "Conondale ", "Allyn River",
    "Bulburin"))
```

Как связаны признаки между собой?

Можно построить серию графиков с признаками во всех возможных комбинациях.

```
library(RColorBrewer)
# цвета
cols <- brewer.pal(n = length(levels(possum$site)), name = "Set1")
# график
pairs(possum[, c(6:14, 2)], col = cols[possum$site], pch = as.numeric(possum$se
```


hdlngth
skullw
totlngth
taill
footlgth
earconch
eye
chest
belly
site

Анализ главных компонент

```
library(vegan)
# возьмем только строки, где нет пропущенных значений
possum <- possum[complete.cases(possum), ]
# ординация, используем переменные с hdlngth по belly
ord <- rda(possum[, 6:14], scale = TRUE)

summary(ord)
```

Все результаты можно посмотреть при помощи функции `summary()`

```
##
## Call:
## rda(X = possum[, 6:14], 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               3.931 1.949 0.908 0.7516 0.5769 0.3099 0.2671
## Proportion Explained     0.437 0.217 0.101 0.0835 0.0641 0.0344 0.0297
## Cumulative Proportion    0.437 0.653 0.754 0.8378 0.9019 0.9363 0.9660
##                            PC8   PC9
## Eigenvalue               0.1625 0.144
## Proportion Explained     0.0181 0.016
## Cumulative Proportion    0.9840 1.000
##
```

Части результатов в `summary()`

- Importance of components - собственные числа (eigenvalues) и доля объясненной изменчивости
- Species scores - факторные нагрузки исходных переменных на каждую из компонент
- Site scores - факторные координаты объектов

Масштабирование - scaling

- **scaling 2** - отношения между переменными (нагрузки переменных пересчитаны с учетом соб. чисел)
- **scaling 1** - отношения между объектами (факт. координаты пересчитаны с учетом соб. чисел)
- **scaling 0** - нет масштабирования
- **отрицательные значения scaling** - стандартизованные координаты и нагрузки

Что нужно знать, чтобы интерпретировать результаты?

Мы хотим снизить размерность данных и вместо n-дцати исходных признаков получить несколько главных компонент (лучше 2 или 3 для удобства интерпретации).

Эти главные компоненты будут описывать данные почти так же хорошо, как исходные признаки, но при этом будут независимы друг от друга.

Эти компоненты мы сможем трактовать как сложные признаки и описывать отношения между объектами в терминах этих признаков.

1. Сколько компонент нужно оставить?
2. Сколько общей изменчивости объясняют оставленные компоненты?
3. Что означают получившиеся компоненты?
4. Как располагаются объекты в пространстве главных компонент?

1A. Сколько компонент нужно оставить?

Вариант А. Оставляем компоненты с соб. числами > 1 (правило Кайзера)

```
eigenvals(ord)
```

```
##  PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9
## 3.93 1.95 0.91 0.75 0.58 0.31 0.27 0.16 0.14
```

```
eigenvals(ord) > mean(eigenvals(ord))
```

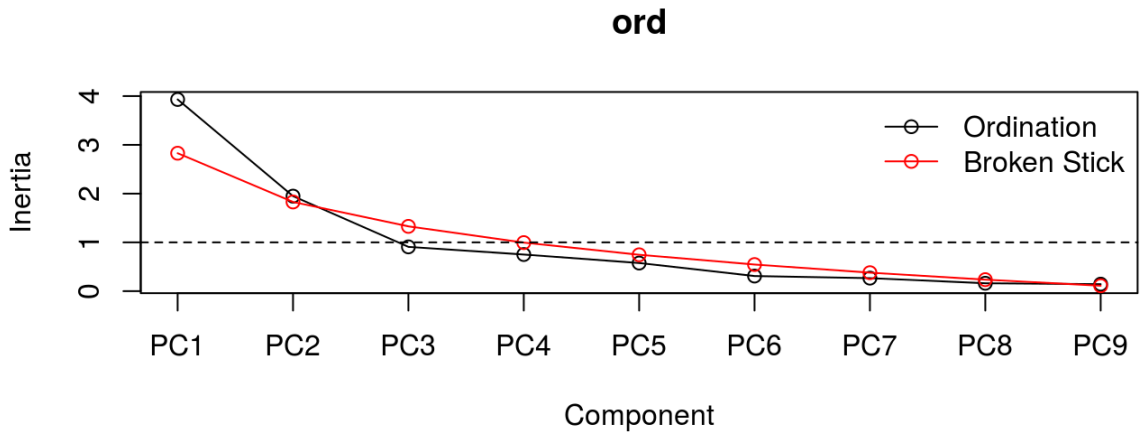
```
##   PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9
##  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

1Б. Сколько компонент нужно оставить?

Вариант Б. Оставляем компоненты, кот объясняют больше изменчивости, чем возможно случайно (по модели сломанной палки).

Строим график собственных чисел

```
screeplot(ord, bstick = TRUE, type = "lines")
abline(h = 1, lty = 2)
```



2. Сколько изменчивости объясняют компоненты?

Допустим, мы решили оставить первые две компоненты.

Изменчивость, объясненная каждой из компонент, в процентах

```
eigenvals(ord)/sum(eigenvals(ord)) * 100
```

```
##  PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9
## 43.7 21.7 10.1  8.4  6.4  3.4  3.0  1.8  1.6
```

Первые две компоненты объясняют 65 % общей изменчивости

3. Что означают получившиеся компоненты?

Факторные нагрузки описывают связь переменных с компонентами

- Вклад переменных в изменчивость вдоль компоненты тем сильнее, чем больше модуль их факторной нагрузки.
- Знак факторной нагрузки означает направление изменения исходной переменной вдоль главной компоненты.

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

```
##              PC1     PC2     PC3
## hdlngth  -0.434 -0.0633  0.1507
## skullw   -0.386 -0.1109  0.2513
## totlngth -0.418 -0.0781 -0.3406
## taill    -0.193 -0.4814 -0.5348
## footlgth -0.307  0.4971 -0.0791
## earconch -0.139  0.6385 -0.0219
## eye      -0.186 -0.2764  0.7094
## chest    -0.409  0.0888 -0.0113
## belly    -0.367 -0.0821 -0.0450
## attr(,"const")
## [1] 5.48
```

Можно нарисовать факторные нагрузки на графике

- Чем ближе стрелки исходных признаков к оси компоненты, тем выше их нагрузка.
- Стрелки направлены в сторону увеличения значения исходного признака

```
biplot(ord, scaling = 2, main = "PCA - scaling 2", display = "species")
```

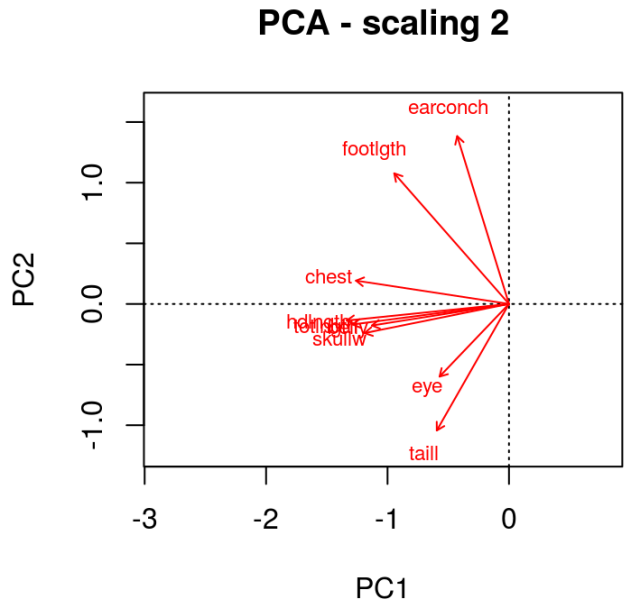
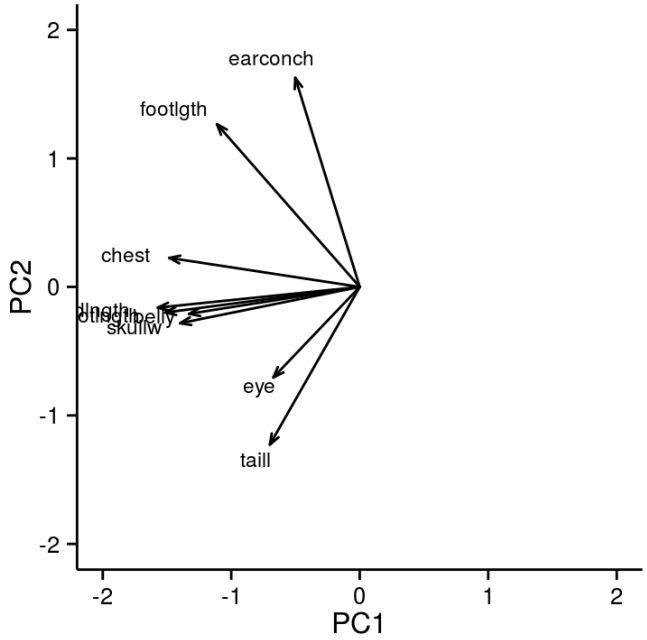


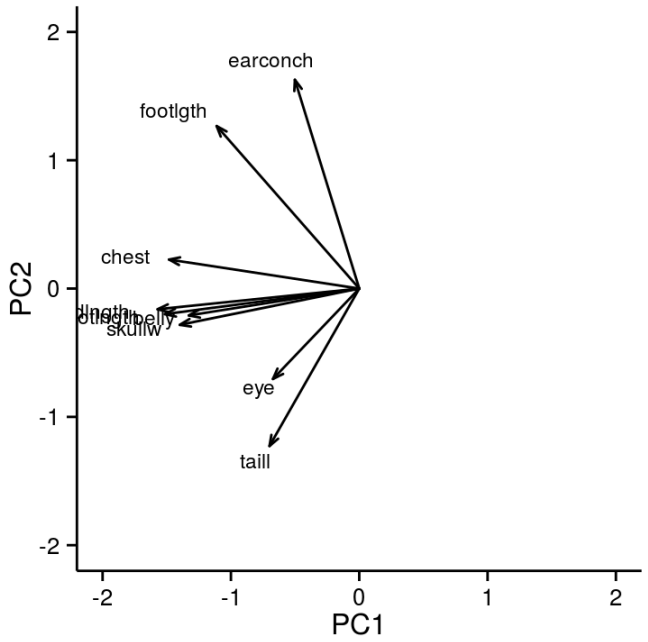
График факторных нагрузок в ggplot

```
# install.packages('devtools') library('devtools')
# install_github('jiho/autoplot')
library(autoplot)
# fortify(ord) # исходные данные, если нужно
ggloadings <- autoplot(ord, data = possum, type = "var", PC = c(1, 2)) +
    labs(x = "PC1", y = "PC2") + xlim(c(-2, 2)) + ylim(c(-2, 2))
ggloadings
```



Интерпретируем компоненты по графику факторных нагрузок

- Первая главная компонента - это физические размеры поссумов (высокие нагрузки у переменных длина головы, общая длина, измерения черепа, груди и живота). У нагрузок отрицательный знак, значит у крупных поссумов будут маленькие значения координат по первой компоненте.
- Вторая главная компонента - длина ушей, ног и хвоста. Высокие значения по этой компоненте у поссумов с большими ушами, длинными ногами и коротким хвостом.



4. Значения факторов (= факторные координаты) - координаты объектов в пространстве главных компонент

```
# Координаты можно добыть так (но на самом деле сейчас нам нужен только
# график)
scores(ord, display = "sites", choices = c(1, 2, 3), scaling = 1)
```

```
##            PC1      PC2       PC3
## C3    -0.36082  0.30413  0.068882
## C5    -0.24009  0.14337  0.069353
## C10   -0.55114  0.10784 -0.142387
## C15   -0.31280  0.20713 -0.127399
## C23   -0.03449  0.26729  0.048805
## C24   -0.15100  0.39427 -0.126411
## C26   -0.29840  0.25133 -0.069267
## C27   -0.30610  0.26687 -0.120363
## C28   -0.21457  0.19318 -0.029604
## C31   -0.09550  0.20927 -0.138730
## C32   -0.39946  0.20520 -0.180091
## C34   -0.18757  0.19859  0.018334
## C36   -0.19136  0.08086  0.164452
## C37   -0.25074  0.31206  0.003360
```

График факторных координат (= график ординации)

```
biplot(ord, scaling = 1, main = "PCA - scaling 1", display = "sites",
    type = "t")
```

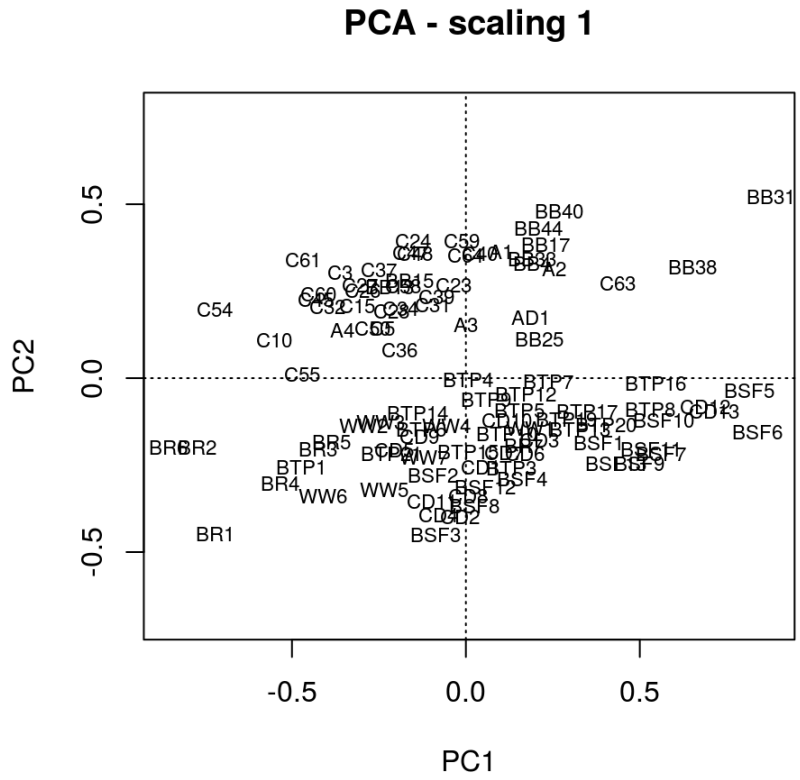
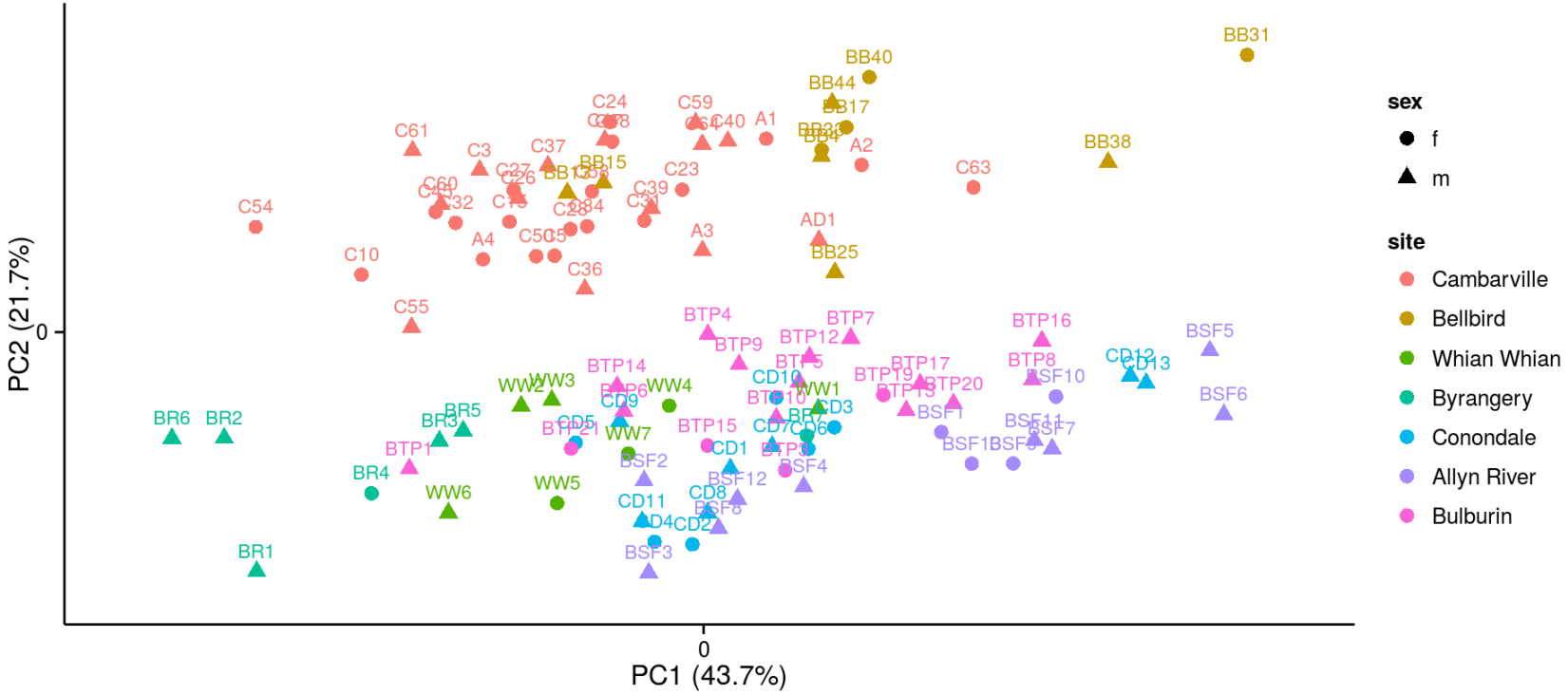


График факторных координат в ggplot

```
ggscores <- autoplot(ord, data = possum, type = "obs", PC = c(1, 2), aes(colour
    shape = sex), size = 3)
ggscores
```



Делаем красивый график ординации.

```
# Подписи можно удалить из объекта `autoplot` и вставить свои.  Смотрим
# на слои графика, второй из них содержит geom_text и он нам не нужен
ggscores$layers
```

```
## [[1]]
## geom_point: na.rm = FALSE, size = 3
## stat_identity:
## position_identity: (width = NULL, height = NULL)
##
## [[2]]
## mapping: label = .id
## geom_text: parse = FALSE, size = 3, vjust = -1
## stat_identity:
## position_identity: (width = NULL, height = NULL)
```

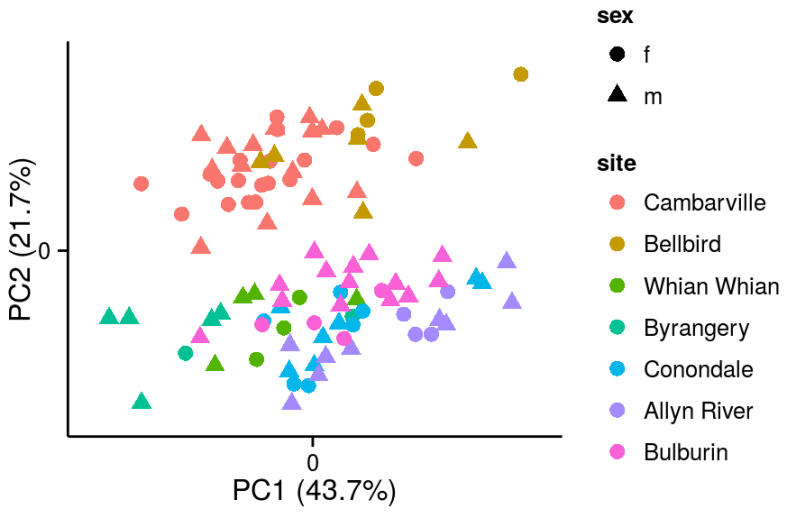
```
# удаляем слой с подписями
ggscores$layers <- ggscores$layers[-2]
```

Интерпретируем сходство объектов по графику ординации

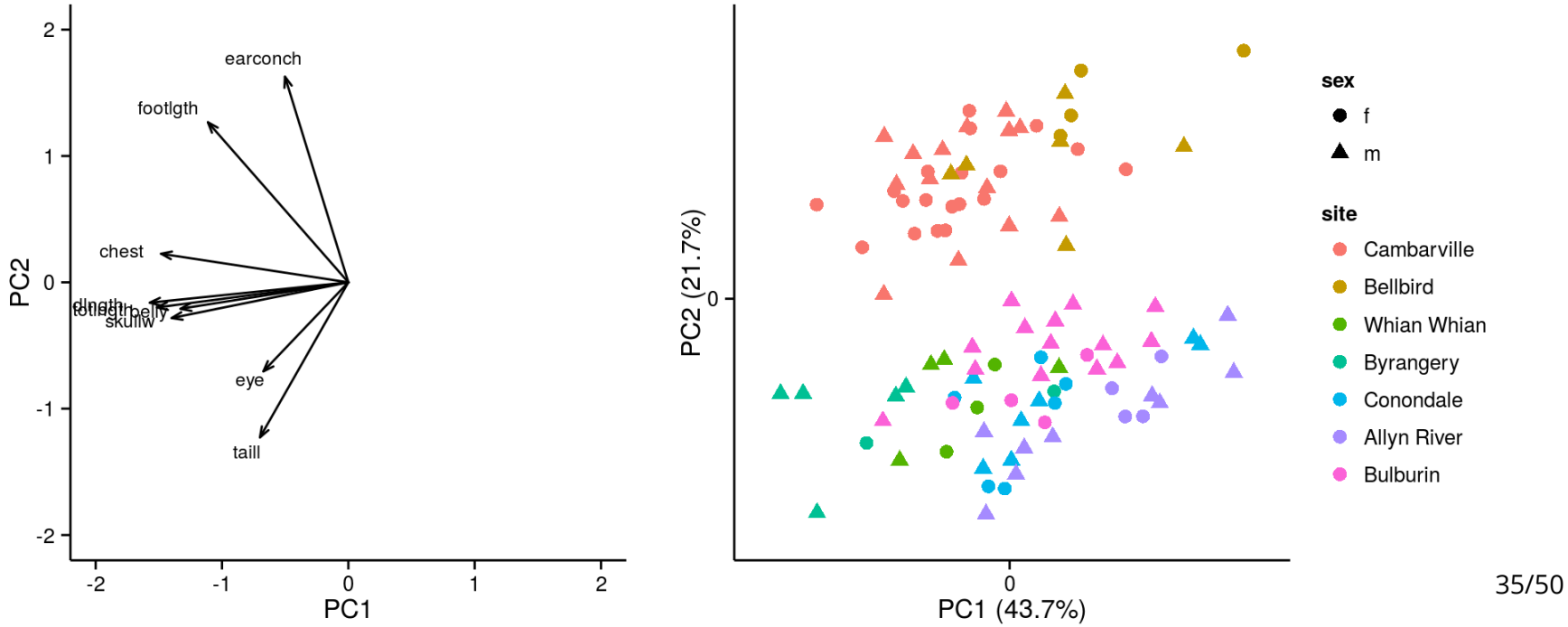
Первые две компоненты объясняют 65% общей изменчивости

44% общей изменчивости объясняет первая главная компонента, связанная с размером особей. Более крупные поссумы встречаются в популяциях из Камбарвиля и Бирангери

Вторая компонента, которую мы интерпретировали как пропорции ног, ушей и хвоста, объясняет 21% общей изменчивости. Внутри отдельных популяций поссумы мало отличаются по этим параметрам (об этом говорит небольшой разброс точек вдоль второй компоненты). Зато поссумы из Камбарвиля и Беллберда не похожи на других: у них относительно более крупные уши, длинные ноги и короткие хвосты, чем у поссумов из других популяций.



```r
# Несколько графиков рядом
library(gtable)
g1 <- ggplotGrob(ggloadings)
g2 <- ggplotGrob(ggscores)
g <- gtable:::cbind_gtable(g1, g2, "first")
grid.newpage()
grid.draw(g)
```



Факторные координаты можно использовать для снижения размерности данных

Было 7 скоррелированных признаков, стало 2 **независимых** (они ведь перпендикулярны) главных компоненты

Значения факторных координат можно использовать в анализах, где нужна независимость переменных:

- Множественная регрессия
- Дискриминантный анализ (например, генетические данные)
- Дисперсионный анализ
- Корреляция с другими признаками, которые не были использованы в анализе главных компонент, и т.д., и т.п.

```
# Так можно экстрагировать компоненты с исходными данными
scrs <- scores(ord, display = "sites", choices = c(1, 2, 3), scaling = 1)
data_with_pc <- data.frame(possum, scrs)
head(data_with_pc)
```

```
##     case        site Pop sex age hdlngth skullw totlngth taill
## C3     1 Cambarville Vic   m   8    94.1   60.4     89.0  36.0
## C5     2 Cambarville Vic   f   6    92.5   57.6     91.5  36.5
## C10    3 Cambarville Vic   f   6    94.0   60.0     95.5  39.0
## C15    4 Cambarville Vic   f   6    93.2   57.1     92.0  38.0
## C23    5 Cambarville Vic   f   2    91.5   56.3     85.5  36.0
```

Условия применимости анализа главных компонент

Похожи на условия применимости множественной линейной регрессии

- Линейные связи между переменными (т.к. матрица корреляций или ковариаций)
- Исключить наблюдения, в которых есть пропущенные значения
- Если много нулей - трансформация данных (например, трансформация Хелингера)
- Если очень много нулей - удалить такие переменные из анализа

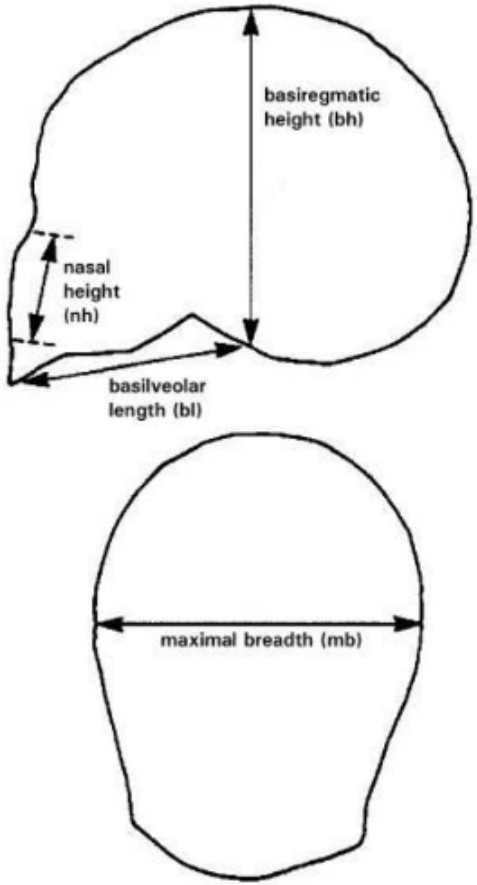
Пример: Морфометрия египетских черепов

Измерения 150 черепов в мм:

- mb - максимальная ширина
- bh - высота от основания до макушки
- bl - расстояние от основания черепа до края в. челюсти
- nh - высота носа

Эпоха (epoch):

- 1 - ранний прединастический период (ок. 4000 до н.э.)
- 2 - поздний прединастический период (ок. 3300 до н.э.)
- 3 - 12 и 13 династии (ок. 1850 до н.э.)
- 4 - Птолемейский период (ок. 200 до н.э.)
- 5 - Римский период (ок. 150 н.э.)



Данные Thompson, Randall-Maciver (1905). Источник Manly (1994).

Знакомимся с данными

```
library(HSAUR)
data("skulls")
str(skulls, vec.len = 2)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ epoch: Ord.factor w/ 5 levels "c4000BC"<"c3300BC"<..: 1 1 1 1 1 ...
##  $ mb   : num  131 125 131 119 136 ...
##  $ bh   : num  138 131 132 132 143 ...
##  $ bl   : num  89 92 99 96 100 ...
##  $ nh   : num  49 48 50 44 54 ...
```

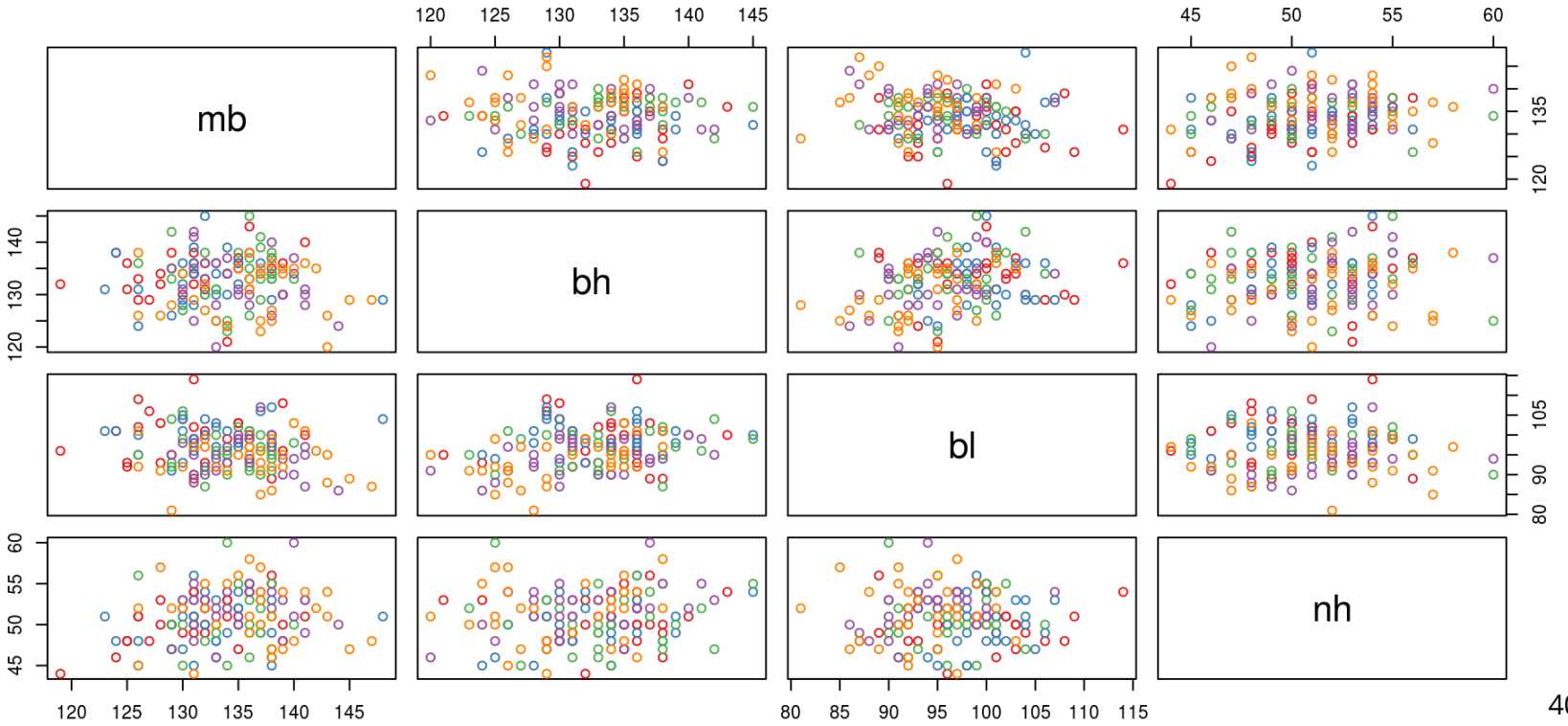
```
sum(is.na(skulls))
```

```
## [1] 0
```

```
table(skulls$epoch)
```

```
##
## c4000BC c3300BC c1850BC  c200BC  cAD150
##      30      30      30      30      30
```

```r
# цвета
library(RColorBrewer)
cols <- brewer.pal(n = length(levels(skulls$epoch)), name = "Set1")
# график
pairs(skulls[, -1], col = cols[skulls$epoch])
```



Задание:

Сделайте анализ главных компонент. Как менялась форма черепов в древнем египте в разные эпохи?

Решение

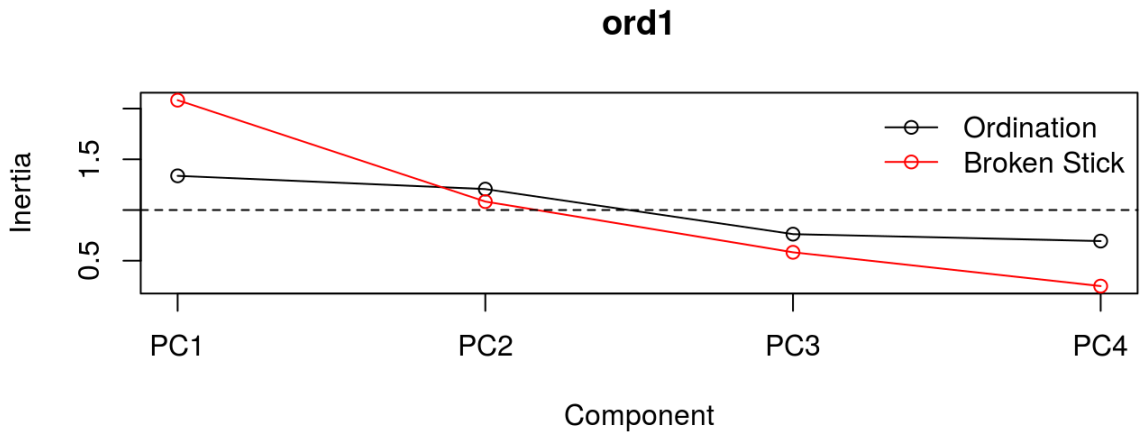
Делаем анализ главных компонент.

Не забудьте оставить в исходных данных только непрерывные переменные

```
ord1 <- rda(skulls[, -1], scale = TRUE)
```

Сколько компонент нужно оставить?

```
screeplot(ord1, bstick = TRUE, type = "lines")
abline(h = 1, lty = 2)
```



- Оставляем две компоненты (можно даже одну)

Сколько изменчивости объясняют компоненты?

```
eig <- eigenvals(ord1)
explained <- sum(eig[1:2])/sum(eig) * 100
explained
```

```
## [1] 63.6
```

- Компоненты вместе объясняют 64 % общей изменчивости

Что означают получившиеся компоненты?

- Вдоль 1й компоненты уменьшается расстояние от основания черепа до края в. челюсти (bl) и высота от основания до макушки (bh)
- Вдоль 2й компоненты уменьшается высота носа (nh) и максимальная ширина (mb)

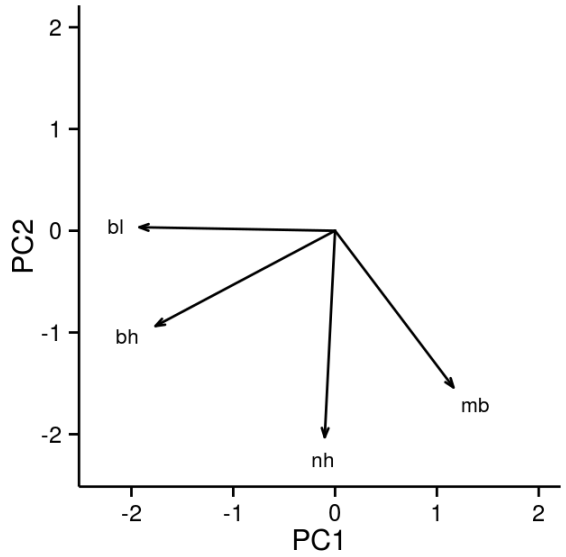
```
scores(ord1, display = "species", choices = c(1, 2), scaling = 0)
```

```
##         PC1     PC2
## mb  0.4070 -0.5674
## bh -0.6172 -0.3450
## bl -0.6724  0.0128
## nh -0.0355 -0.7476
## attr(,"const")
## [1] 4.94
```

Что означают получившиеся компоненты?

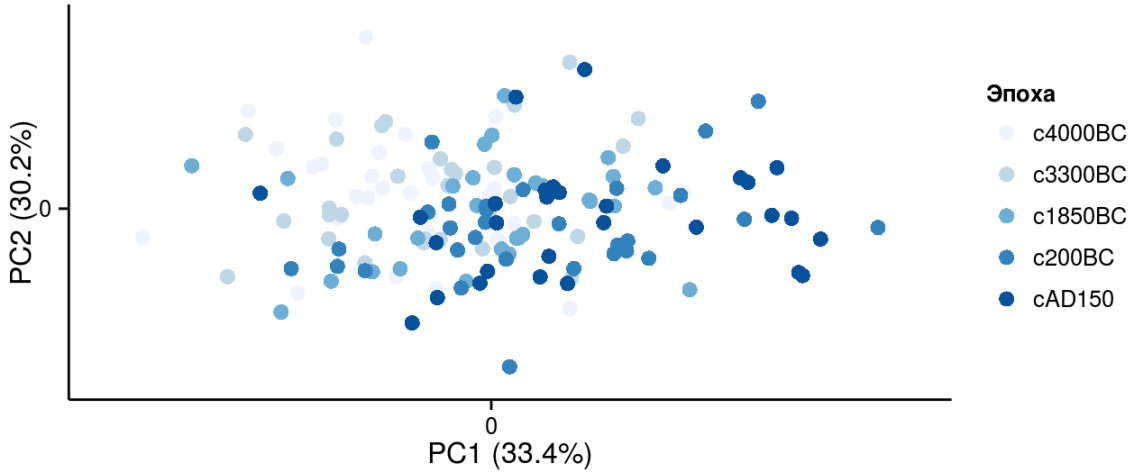
- Вдоль 1й компоненты уменьшается расстояние от основания черепа до края в. челюсти (bl) и высота от основания до макушки (bh)
- Вдоль 2й компоненты уменьшается высота носа (nh) и максимальная ширина (mb)

```
ggloadings1 <- autoplot(ord1, data = skulls, type = "var", PC = c(1, 2)) +
    labs(x = "PC1", y = "PC2") + xlim(c(-2.3, 2)) + ylim(c(-2.3, 2))
ggloadings1
```



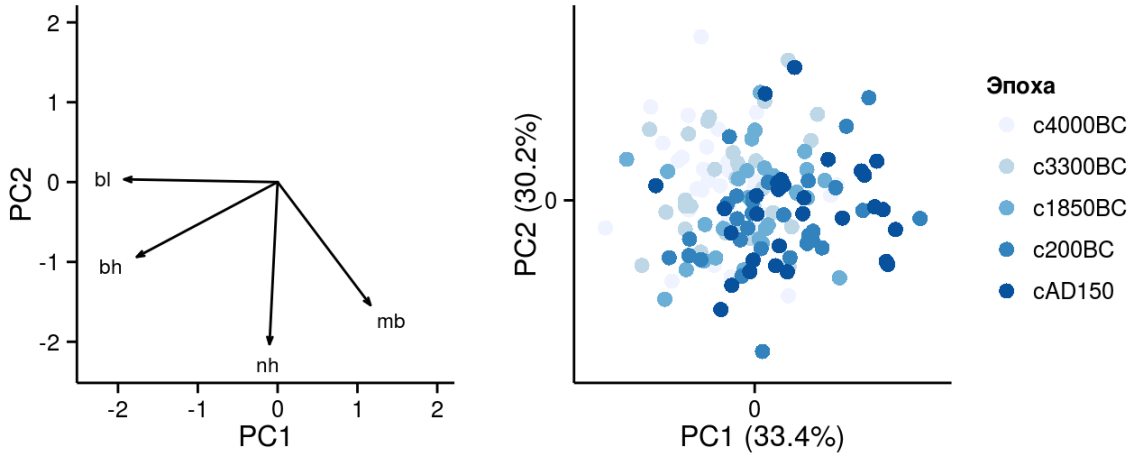
Как располагаются объекты в пространстве главных компонент?

```
ggscores1 <- autoplot(ord1, data = skulls, type = "obs", PC = c(1, 2),
    aes(colour = epoch), size = 3) + scale_color_brewer(name = "Эпоха")
ggscores1$layers <- ggscores1$layers[-2]
ggscores1
```



Для облегчения интерпретации располагаем графики рядом

```
g1 <- ggplotGrob(ggloadings1)
g2 <- ggplotGrob(ggscores1)
g <- gtable:::cbind_gtable(g1, g2, "first")
grid.newpage()
grid.draw(g)
```



- С течением времени форма черепов древних египтян изменялась. Постепенно увеличивались размеры черепа, а длина носа практически не изменялась

Take home messages

- Метод главных компонент:
  - исследование связей между переменными
  - построение ординации объектов
  - снижение размерности данных
- Собственные числа - вклад компонент в общую изменчивость
- Факторные нагрузки - связь исходных переменных с компонентами - используются для интерпретации
- Значения факторов (факторные координаты) - новые координаты объектов в пространстве уменьшенной размерности
- Значения факторов можно использовать как новые комплексные переменные в других видах анализов.

Дополнительные ресурсы

- Borcard, D., Gillet, F., Legendre, P., 2011. Numerical ecology with R. Springer.
- 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 URL http://ordination.okstate.edu/ (accessed 10.21.13).
- Quinn, G.G.P., Keough, M.J., 2002. Experimental design and data analysis for biologists. Cambridge University Press.
- Zuur, A.F., Ieno, E.N., Smith, G.M., 2007. Analysing ecological data. Springer.