

Анализ главных компонент

Математические методы в зоологии с использованием R

Марина Варфоломеева

Знакомимся с ординацией на примере метода главных компонент

Вы сможете

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

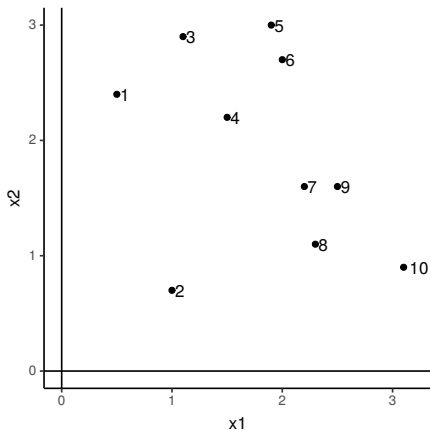
Снижение размерности многомерных данных

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

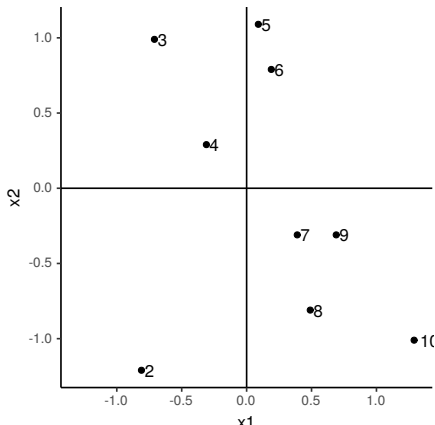
Центрирование

Из каждого значения переменной нужно вычесть среднее значение этой переменной

Если центрировать данные, 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	0.6165556	-0.2490000
# x2	-0.2490000	0.7165556

Матрица ковариаций

- 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  0.6165556 -0.2490000
# x2 -0.2490000  0.7165556
```

Собственные числа

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

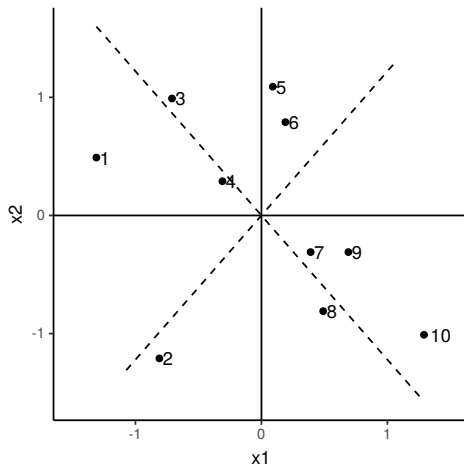
```
# [1] 0.9205260 0.4125851
```

Собственные векторы

- их столько же, сколько исходных переменных
- перпендикулярны друг другу
- задают направление осей главных компонент
- вдоль первого — максимальная дисперсия данных, вдоль следующего — максимальная дисперсия из оставшейся и т.д.

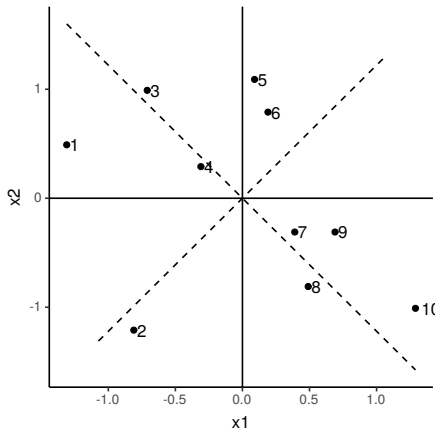
```
#           [,1]           [,2]
# [1,] -0.6336903 -0.7735869
# [2,]  0.7735869 -0.6336903
```

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

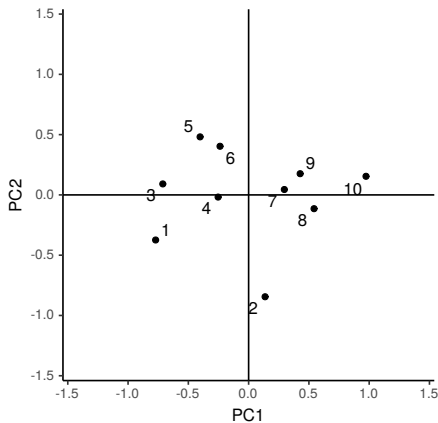


Можно найти новые координаты точек в получившемся новом пространстве

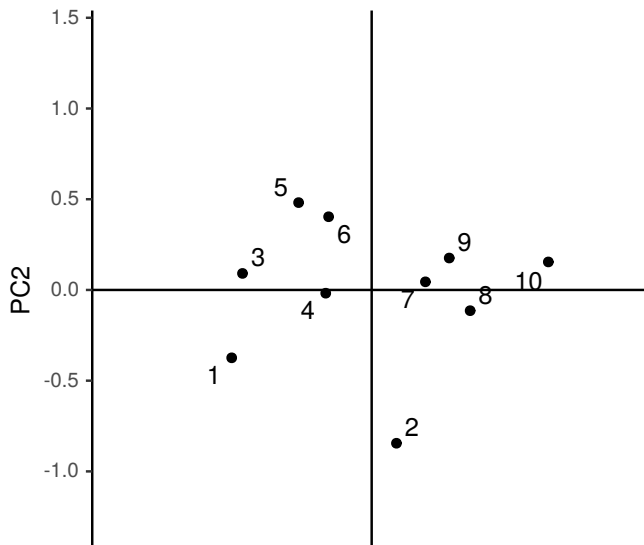
Объекты и оси в пространстве исходных признаков:



Объекты в пространстве новых осей (главных компонент):



На графике ординации изображено новое пространство



Анализ главных компонент в R

Пример: Морфометрия поссумов



© Hasitha Tudugalle | Photography

Знакомимся с данными

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

```
# [1] "case"      "site"      "Pop"       "sex"       "age"       "hdlength"
# [7] "skullw"    "totlength" "tail"      "footlength" "earconch"  "eye"
# [13] "chest"     "belly"
```

```
colSums(is.na(possum))
```

```
#      case      site      Pop      sex      age      hdlength      skullw
#         0         0         0         0         2         0         0
# totlength      tail footlength earconch      eye      chest      belly
#         0         0         1         0         0         0         0
```

```
# оставим только строки с полными наблюдениями
pos <- possum[complete.cases(possum), ]
```

```
# поссумы из разных сайтов из 2 популяций  
table(pos$site, pos$Pop)
```

```
#  
#      Vic other  
#  1  33      0  
#  2  10      0  
#  3   0      7  
#  4   0      7  
#  5   0     13  
#  6   0     13  
#  7   0     18
```

```
# половой состав выборки из разных сайтов
table(pos$sex, pos$site, pos$Pop)
```

```
# , , = Vic
```

```
#
```

```
#
```

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

```
# f 19  4  0  0  0  0  0
```

```
# m 14  6  0  0  0  0  0
```

```
#
```

```
# , , = other
```

```
#
```

```
#
```

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

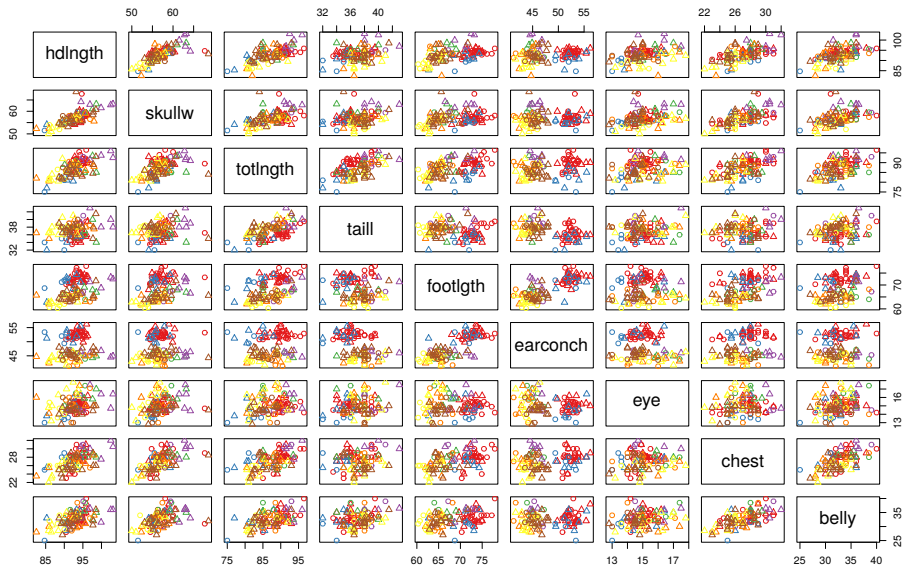
```
# f  0  0  3  2  6  4  4
```

```
# m  0  0  4  5  7  9 14
```

Как связаны признаки между собой?

Можно построить серию графиков с признаками во всех возможных комбинациях.

```
# сколько всего сайтов
n_sites <- length(unique(pos$site))
# цвета из Брюеровской палитры 'Set1'
library(RColorBrewer)
cols <- brewer.pal(n = n_sites, name = 'Set1')
# график морфометрических переменных
pairs(pos[, 6:14], col = cols[pos$site],
      pch = as.numeric(pos$sex))
```

Анализ главных компонент

```
library(vegan)  
# ординация, используем морфометрические переменные (с hdlngth по belly)  
ord <- rda(pos[, 6:14], scale = TRUE)  
  
summary(ord)
```

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

```
#
# Call:
# rda(X = pos[, 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
# Eigenvalue      3.9314 1.9486 0.9084 0.75157 0.57685 0.30986
# Proportion Explained 0.4368 0.2165 0.1009 0.08351 0.06409 0.03443
# Cumulative Proportion 0.4368 0.6533 0.7543 0.83777 0.90186 0.93629
#               PC7    PC8    PC9
# Eigenvalue      0.26713 0.16252 0.14373
# Proportion Explained 0.02968 0.01806 0.01597
# Cumulative Proportion 0.96597 0.98403 1.00000
#
# Scaling 2 for species and site scores
# * Species are scaled proportional to eigenvalues
```

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 = "species", correlation = TRUE`** — отношения между переменными (нагрузки переменных пересчитаны с учетом соб. чисел, интерпретируются как корреляции)
- **`scaling = "sites"`** — отношения между объектами (факт. координаты пересчитаны с учетом соб. чисел)

Что нужно знать, чтобы интерпретировать результаты?

Мы хотим снизить размерность данных и вместо множества 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.

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

Чтобы все это получилось, нужно ответить на несколько вопросов:

- 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?

1А. Сколько компонент нужно оставить?

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

```
eigenvals(ord) # собственные числа
```

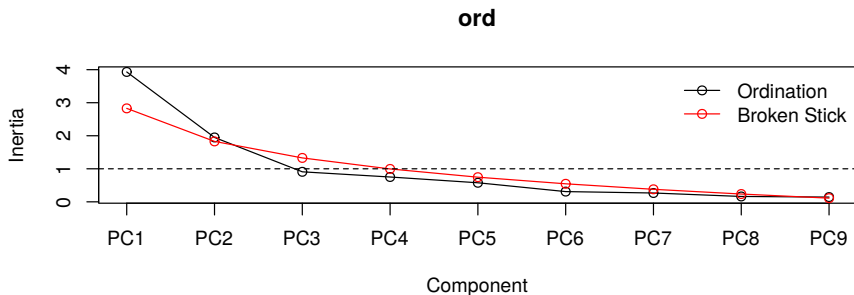
```
#      PC1      PC2      PC3      PC4      PC5      PC6      PC7
# 3.931419 1.948555 0.908362 0.751571 0.576850 0.309860 0.267133
#      PC8      PC9
# 0.162520 0.143728
```

1Б. Сколько компонент нужно оставить?

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

Строим график собственных чисел

```
screepplot(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
# 43.68244 21.65061 10.09291  8.35079  6.40945  3.44289  2.96814
#      PC8      PC9
#  1.80578  1.59698
```

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 = 'species', correlation = TRUE)
```

```
#           PC1          PC2          PC3
# hdlngth -0.4713851 -0.04837773  0.078655520
# skullw  -0.4194429 -0.08480655  0.131206176
# totlngth -0.4542416 -0.05969730 -0.177801904
# taill    -0.2098116 -0.36809068 -0.279173018
# footlght -0.3333944  0.38003868 -0.041289909
# earconch -0.1504873  0.48821273 -0.011420156
# eye      -0.2017138 -0.21130983  0.370315121
# chest    -0.4446740  0.06787162 -0.005893116
# belly    -0.3983862 -0.06276943 -0.023506174
# attr(,"const")
# [1] 5.477226
```

3. Что означают получившиеся компоненты?

	PC1	PC2	PC3
hdlngth	-0.4713851	-0.04837773	0.078655520
skullw	-0.4194429	-0.08480655	0.131206176
totlngth	-0.4542416	-0.05969730	-0.177801904
taill	-0.2098116	-0.36809068	-0.279173018
footlngth	-0.3333944	0.38003868	-0.041289909
earconch	-0.1504873	0.48821273	-0.011420156
eye	-0.2017138	-0.21130983	0.370315121
chest	-0.4446740	0.06787162	-0.005893116
belly	-0.3983862	-0.06276943	-0.023506174

- PC1 — 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.
- PC2 — длина ушей, ног и хвоста. Высокие значения по этой компоненте у поссумов с большими ушами, длинными ногами и коротким хвостом.
- PC3 — размеры глаз. Высокие значения по этой компоненте будут у поссумов с большими глазами.

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 = 'species', correlation = TRUE,
       main = 'PCA - species scaling', display = 'species')
```

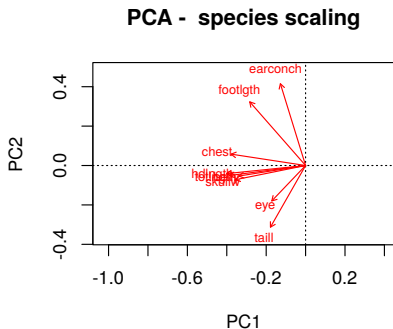
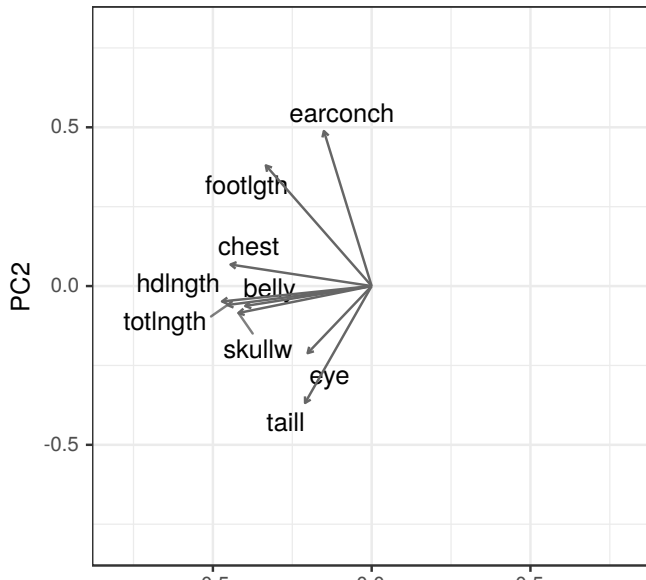


График факторных нагрузок в ggplot2

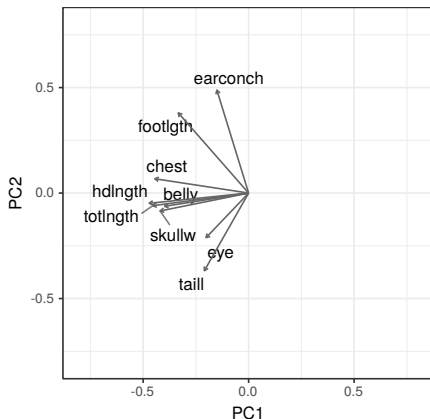
```
library(ggplot2)
theme_set(theme_bw())
library(ggrepel) # для подписей (geom_text_repel)
library(grid) # для стрелочек
# параметры стрелочек
ar <- arrow(length = unit(0.1, 'cm'))
# датафрейм с факторными нагрузками
df_load <- data.frame(scores(ord, display = 'species',
                             choices = c(1, 2), scaling = 'species', correlation = TRUE))
# график
ggloadings <- ggplot(df_load) +
  geom_text_repel(aes(x = PC1, y = PC2,
                     label = rownames(df_load)), segment.alpha = 0.5) +
  geom_segment(aes(x = 0, y = 0, xend = PC1, yend = PC2),
              colour = 'grey40', arrow = ar) +
  coord_equal(xlim = c(-0.8, 0.8), ylim = c(-0.8, 0.8))
ggloadings
```

График факторных нагрузок в ggplot2



Интерпретируем компоненты по графику факторных нагрузок

- PC1 — 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.
- PC2 — 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 сейчас нам нужен только график)
`scores(ord, display = 'sites', choices = c(1, 2, 3), scaling = 'sites')`

#	PC1	PC2	PC3
# C3	-0.360819375	0.304131424	0.0688821891
# C5	-0.240088432	0.143367378	0.0693525874
# C10	-0.551140377	0.107837459	-0.1423865805
# C15	-0.312799927	0.207126158	-0.1273986533
# C23	-0.034485559	0.267285182	0.0488052857
# C24	-0.151004588	0.394274329	-0.1264112959
# C26	-0.298396559	0.251333661	-0.0692674630
# C27	-0.306099050	0.266871815	-0.1203631232
# C28	-0.214566258	0.193184218	-0.0296039420
# C31	-0.095501125	0.209273041	-0.1387304801
# C32	-0.399460404	0.205195000	-0.1800913882
# C34	-0.187568929	0.198586258	0.0183336106
# C36	-0.191356099	0.080859321	0.1644518057
# C37	-0.250735386	0.312058901	0.0033603952
# C39	-0.084377646	0.232837138	0.2515360165
# C40	0.039106450	0.358378412	0.0251344458
# C45	-0.431885643	0.225796864	0.1077821334
# C47	-0.159125483	0.359986504	0.0058833057

График факторных координат (= график ординации)

```
biplot(ord, scaling = 'sites', display = 'sites',
       type = 't', main = 'PCA - sites scaling')
```

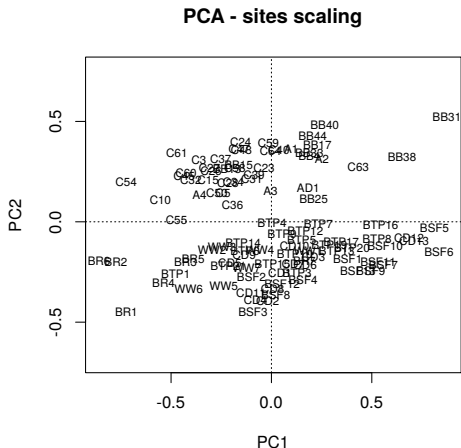
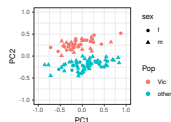


График факторных координат в ggplot2

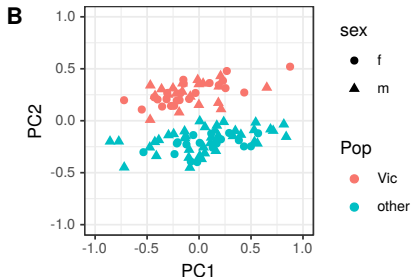
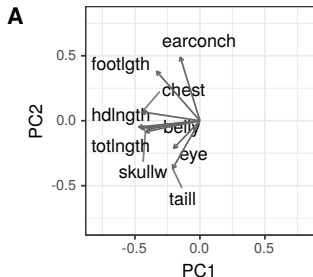
```
# данные для графика: факторные координаты и исходные переменные
df_scores <- data.frame(pos,
                        scores(ord, display = 'sites', scaling = 'sites',
                              choices = c(1, 2, 3)))

# график ординации
ggscores <- ggplot(df_scores, aes(x = PC1, y = PC2,
                                colour = Pop, shape = sex)) +
  geom_point(size = 2) +
  coord_equal(xlim = c(-1, 1), ylim = c(-1, 1))
ggscores
```



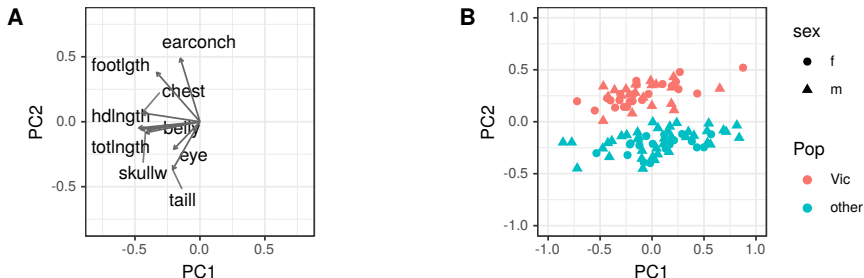
Для удобства интерпретации ординации, располагаем ее рядом с графиком факторных нагрузок

```
library(cowplot)
plot_grid(ggloadings, ggscorers, labels = c('A', 'B'), align = 'hv', axis = 'r
```



Для удобства интерпретации ординации, располагаем ее рядом с графиком факторных нагрузок

```
library(cowplot)
plot_grid(ggloadings, ggcores, labels = c('A', 'B'), align = 'hv', axis = 'r'
```



Первые две компоненты объясняют 65% общей изменчивости. 44% общей изменчивости объясняет первая главная компонента, связанная с размером особей. Вторая компонента — которую мы интерпретировали как пропорции ног, ушей и хвоста — объясняет 21% общей изменчивости. Внутри популяций поссумы не сильно отличаются по этим параметрам (об этом говорит небольшой разброс точек вдоль второй компоненты). Зато поссумы из провинции Виктория не похожи на других: у них относительно более крупные уши, длинные ноги и короткие хвосты.

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.

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

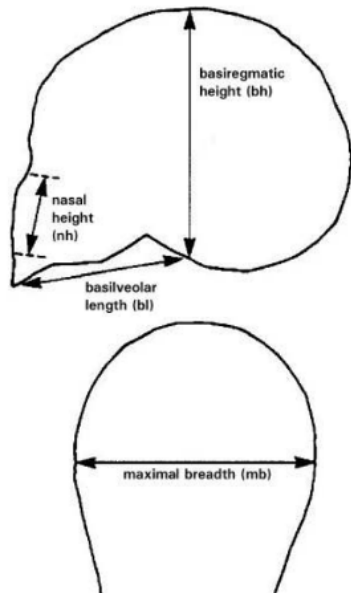
Пример: Морфометрия египетских черепов

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.)



Знакомимся с данными

```
library(HSAUR)
data('skulls')
str(skulls)
```

```
# 'data.frame': 150 obs. of 5 variables:
# $ epoch: Ord.factor w/ 5 levels "c4000BC"<"c3300BC"<...: 1 1 1 1 1 1 1 1 1 ...
# $ mb : num 131 125 131 119 136 138 139 125 131 134 ...
# $ bh : num 138 131 132 132 143 137 130 136 134 134 ...
# $ bl : num 89 92 99 96 100 89 108 93 102 99 ...
# $ nh : num 49 48 50 44 54 56 48 48 51 51 ...
```

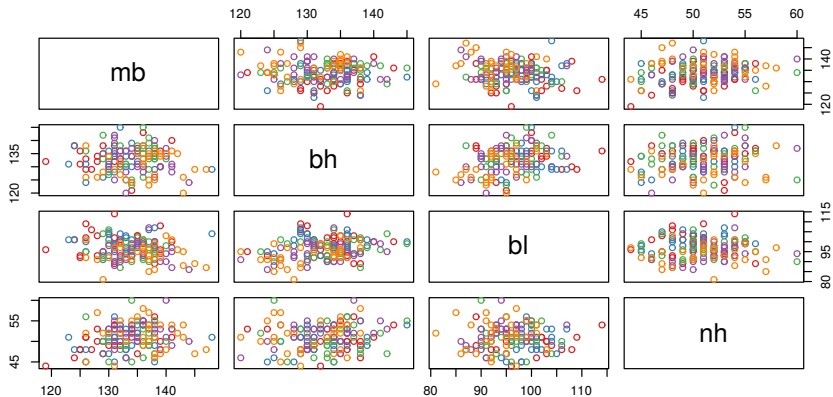
```
sum(is.na(skulls))
```

```
# [1] 0
```

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

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

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



Задание 1

Сделайте анализ главных компонент.

- 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?

Как менялась форма черепов в древнем египте в разные эпохи?

Решение

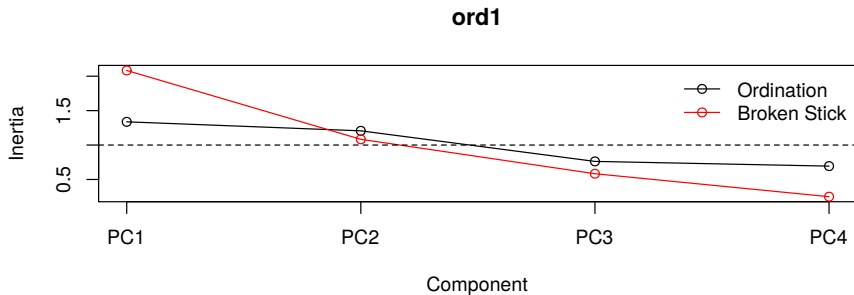
Делаем анализ главных компонент.

Не забудьте оставить в исходных данных только непрерывные переменные

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

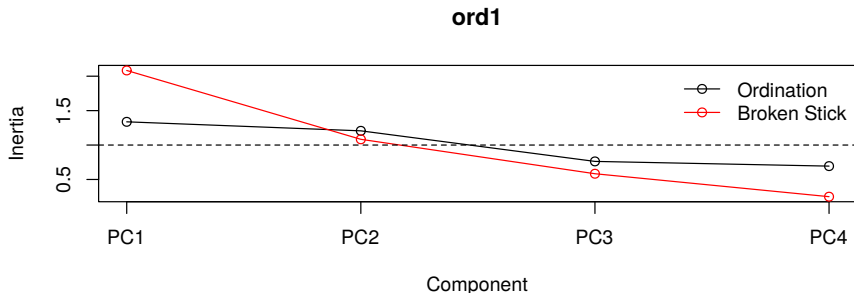
Сколько компонент нужно оставить?

```
screplot(ord1, bstick = TRUE, type = 'lines')  
abline(h = 1, lty = 2)
```



Сколько компонент нужно оставить?

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



- Оставляем две компоненты (можно даже одну, но это будет сложно нарисовать)

Сколько изменчивости объясняют компоненты?

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

```
# [1] 63.59221
```

Сколько изменчивости объясняют компоненты?

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

```
# [1] 63.59221
```

- 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 = 'species', correlation = TRUE)
```

```
#           PC1           PC2
# mb  0.19050654 -0.252267512
# bh -0.28889919 -0.153390201
# bl -0.31476765  0.005671088
# nh -0.01663669 -0.332360682
# attr(,"const")
# [1] 4.940963
```

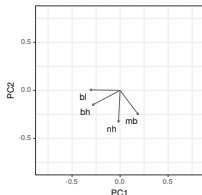
Что означают получившиеся компоненты?

```
# данные для графика факторных нагрузок
df_load <- data.frame(scores(ord1, display = 'species',
                             choices = c(1, 2, 3),
                             scaling = 'species', correlation = TRUE))

# стрелочки
ar <- arrow(length = unit(0.1, 'cm'))

# график факторных нагрузок
ggloadings1 <- ggplot(df_load) +
  geom_text_repel(aes(x = PC1, y = PC2,
                     label = rownames(df_load))) +
  geom_segment(aes(x = 0, y = 0, xend = PC1, yend = PC2),
              colour = 'grey40', arrow = ar) +
  coord_equal(xlim = c(-0.8, 0.8), ylim = c(-0.8, 0.8))
ggloadings1
```


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)

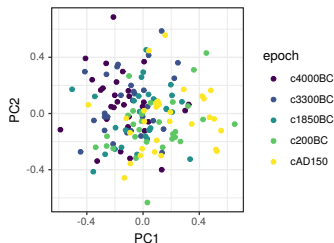
Как располагаются объекты в пространстве главных компонент?

Данные для графика ординации

```
df_scores1 <- data.frame(skulls,
                          scores(ord1, display = 'sites', choices = c(1, 2, 3)
                              scaling = 'sites'))
```

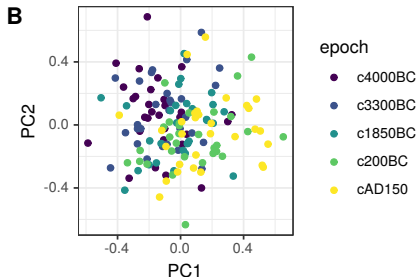
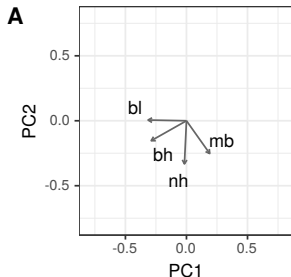
График ординации

```
ggscores1 <- ggplot(df_scores1, aes(x = PC1, y = PC2)) +
  geom_point(aes(colour = epoch)) +
  coord_equal()
ggscores1
```



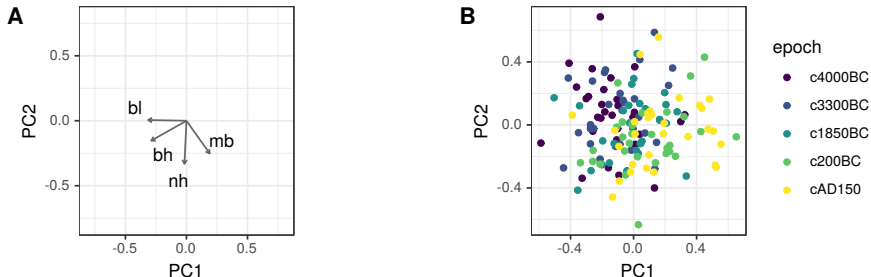
Для удобства интерпретации ординации, располагаем ее рядом с графиком факторных нагрузок

```
# library(cowplot)
plot_grid(ggloadings1, ggcores1, labels = c('A', 'B'), align = 'vh', axis =
```



Для удобства интерпретации ординации, располагаем ее рядом с графиком факторных нагрузок

```
# library(cowplot)
plot_grid(ggloadings1, ggcores1, labels = c('A', 'B'), align = 'vh', axis =
```



- С течением времени форма черепов древних египтян менялась. Размеры черепа постепенно увеличивались, 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.