Анализ главных компонент

Анализ и визуализация многомерных данных с использованием R

Вадим Хайтов, Марина Варфоломеева

Вы сможете

- Найти собственые значения матрицы и собственные вектора матрицы и объяснить их смысл.
- Проводить анализ главных компонент при помощи базовых функций R и 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

Еще немного матричной алгебры

Ортогональные вектора

Скалярное произведение двух векторов

$$\mathbf{a} \bullet \mathbf{b} = \begin{pmatrix} a_1 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{pmatrix} \times \begin{pmatrix} b_1 & b_3 & b_4 & b_5 & b_6 & b_7 \end{pmatrix} = x$$

Скалярное произведение может быть представлено в виде

$$\mathbf{a} \bullet \mathbf{b} = ||a|| \cdot ||b|| \cdot \cos\Theta$$

Если $\Theta = 90^\circ$, то скалярное произведение равно нулю.

Важно! Те векторы, скалярное произведение которых равно нулю, называются **ортогональными**

Ортогональная матрица

Это матрица, все колонки которой являются ортогональными векторамиПростейший пример ортогональной матрицы

```
A <- matrix(c( 0,  0, 10,
               0,  5, 0,
               40, 0, 0  ), ncol = 3)
```

Свойство ортогональной матрицы: По главной диагонали произведения $\mathbf{AA}'$ будут располагаться длины векторов (столбцов)

```
A %*% t(A)
```

```
##      [,1] [,2] [,3]
## [1,] 1600    0    0
## [2,]    0   25    0
## [3,]    0    0  100
```

Номированные (Нормализованные) вектора

$$\mathbf{c} = \mathbf{b} \cdot 1/||\mathbf{b}||$$

Нормирование векторов, делает их длины равными единице.

Ортонормальные матрицы

Это матрицы все вектора (колонки) которой нормированы и при этом ортогональны.

```
normA <-  matrix(c( 0,  0, 1,
                    0,  1, 0,
                    1,  0, 0  ), ncol = 3)
```

Свойство ортонормальных матриц: $\mathbf{A}' = \mathbf{A}^{-1}$

```
solve(normA)
```

```
##      [,1] [,2] [,3]
## [1,]    0    0    1
## [2,]    0    1    0
## [3,]    1    0    0
```

```
t(normA)
```

```
##      [,1] [,2] [,3]
## [1,]    0    0    1
## [2,]    0    1    0
## [3,]    1    0    0
```

Каноническая форма матрицы ассоциации

Обычно в биологических исслдеованиях признаки объектов взаимозависимы. То есть между признаками есть ненулевая корреляция.

Однако часто нам необходимо описать наши объекты в терминах *независимых* переменных. То есть вместо p взаимозависимых признаков получить p взамонезависимых признаков, корреляция между которыми равна нулю.

То же самое на языке матриц

$$\text{Вместо}\mathbf{A}_{pp} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1p} \\ a_{21} & a_{22} & \cdots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{p1} & a_{p2} & \cdots & a_{pp} \end{pmatrix} \text{хотим}\mathbf{\Lambda} = \begin{pmatrix} \lambda_{11} & 0 \cdots & 0 & \\ 0 & \lambda_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_{pp} \end{pmatrix}$$

Каноническая форма матрицы ассоциации

Матрица $\mathbf{\Lambda}$ называется **канонической формой** матрицы $\mathbf{A}$.Значения, располагающиеся по главной диагонали матрицы $\mathbf{\Lambda}$ называются **собственными числами** (eigen values) матрицы $\mathbf{A}$. Собственных чисел столько же, сколько признаков, т.е. столбцов в матрице $\mathbf{A}$.

Собственные значения и собственные вектора матрицы

Связь между матрицей $\mathbf{A}$ и матрицей $\mathbf{\Lambda}$ осуществляется с помощью собственных векторов. Собственных векторов столько же, сколько и собственных чисел.

$$\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}$$

Тогда

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$$

То есть, матрицу ассоциации можно представить в виде произведения двух "вспомогательных" матриц. При этом матрица $\mathbf{V}$ ортонормальная матрица, то есть $\mathbf{V}' = \mathbf{V}^{-1}$.

Симулированная матрица

```
##         Sp1 Sp2 Sp3 Sp4 Sp5
## Sample1 153 149 146 157 153
## Sample2 154 149 154 147 149
## Sample3 149 145 146 142 154
## Sample4 148 149 148 142 161
## Sample5 153 159 156 159 160
## Sample6 141 152 151 148 158
## Sample7 153 153 154 153 151
```

Найдем матрицу ассоциации признаков

```
A <- round(cor(Dat), 2)
A
```

```
##       Sp1  Sp2   Sp3   Sp4   Sp5
## Sp1  1.00 0.14  0.28  0.46 -0.51
## Sp2  0.14 1.00  0.79  0.73  0.37
## Sp3  0.28 0.79  1.00  0.44 -0.07
## Sp4  0.46 0.73  0.44  1.00 -0.05
## Sp5 -0.51 0.37 -0.07 -0.05  1.00
```

Вычислим собственные числа и собственные вектора

```
EIG <- eigen(A)
L <- EIG$values #Вектор собственных чисел
V <- EIG$vectors #Матрица собственных векторов
```

Матрица $\mathbf{V}$ ортонормальная матрица, то есть должно выполняться $\mathbf{V}' = \mathbf{V}^{-1}$. Проверим

```
round(t(V) - solve(V))
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

Разложение матрицы ассоциации

Матрица ассоциации

```
A
```

```
##        Sp1  Sp2   Sp3   Sp4   Sp5
## Sp1  1.00 0.14  0.28  0.46 -0.51
## Sp2  0.14 1.00  0.79  0.73  0.37
## Sp3  0.28 0.79  1.00  0.44 -0.07
## Sp4  0.46 0.73  0.44  1.00 -0.05
## Sp5 -0.51 0.37 -0.07 -0.05  1.00
```

Восстановленная через собственные вектора и собственные числа матрица

```
round(V %*% diag(L) %*% t(V), 2)
```

```
##       [,1] [,2]  [,3]  [,4]  [,5]
## [1,]  1.00 0.14  0.28  0.46 -0.51
## [2,]  0.14 1.00  0.79  0.73  0.37
## [3,]  0.28 0.79  1.00  0.44 -0.07
## [4,]  0.46 0.73  0.44  1.00 -0.05
## [5,] -0.51 0.37 -0.07 -0.05  1.00
```

Свойства собственных векторов:

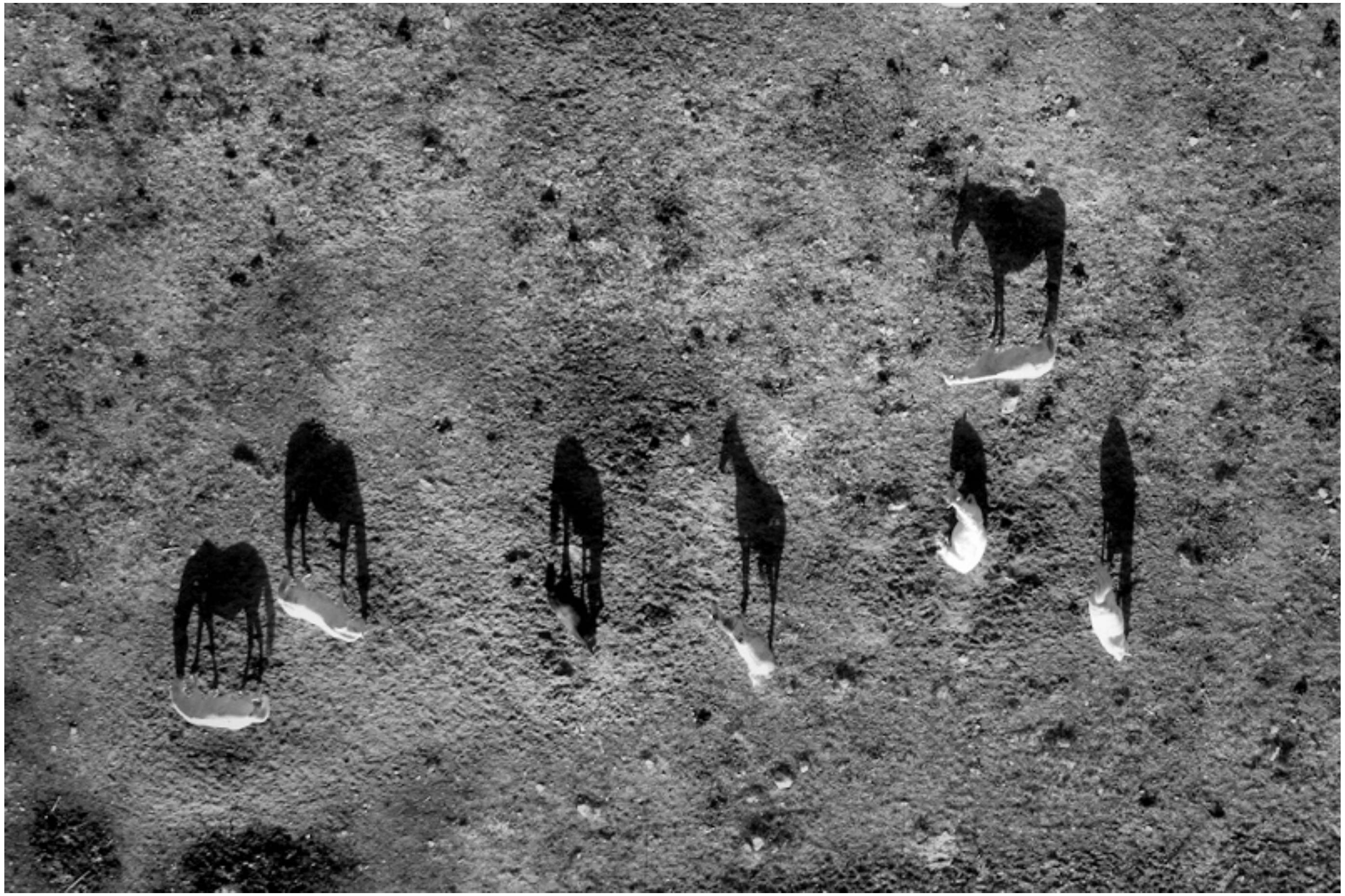
- Собственные вектора и собственные числа есть только у некоторых квадратных матриц
- У матрицы $n \times n$ будет n собственных векторов
- Собственные вектора перпендикулярны друг другу (ортогональны)
- Собственные вектора задают направления в пространстве, вдоль которых дисперсия максимальна
- Первый собственны вектор, связан с первым собственным числом (максимальное значение λ_i), соответствует максимальной дисперсии, второй - чуть меньшей и т.д.

Анализ главных компонент:
Principal Component Analysis

Суть PCA - проекция точек в многомерном пространстве на пространство с меньшей размерностью

Migration by Don McCulloughon Flickr

Не все проекции несут важную информацию



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

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

Cat's shadow by Marina del Castell on Flickr

Применение PCA

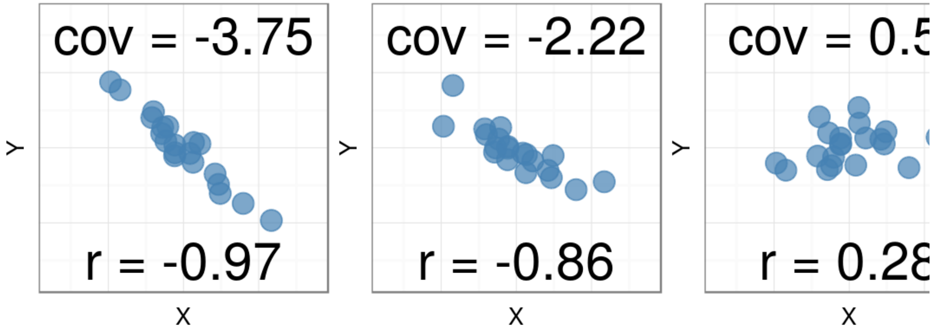
- Снижение размерности в многомерных данных
- сжатие изображений
- распознавание лиц
- анализ формы (геометрическая морфометрия)
- создание комплексных признаков для других анализов

Описание связи нескольких г матрицы ассоциации

- Ковариация - мера совместного варьирования двух признаков

$$Cov(X, Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n-1},$$

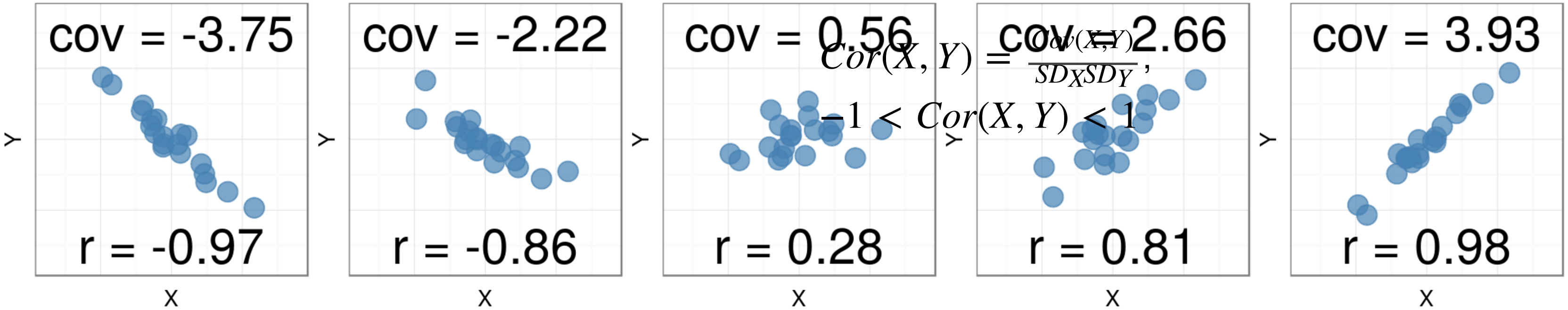
$$-\infty < Cov(X, Y) < \infty$$



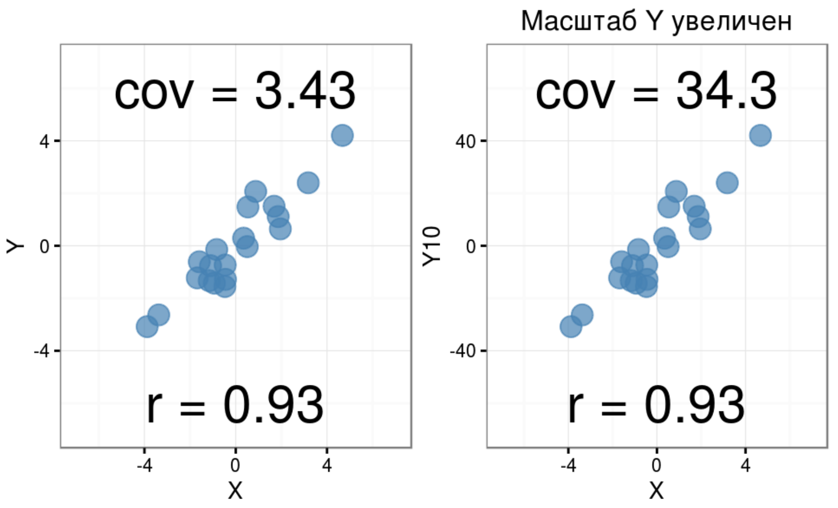
- Корреляция Пирсона - стандартизованная ковариация

$$Cor(X, Y) = \frac{Cov(X, Y)}{SD_X SD_Y},$$

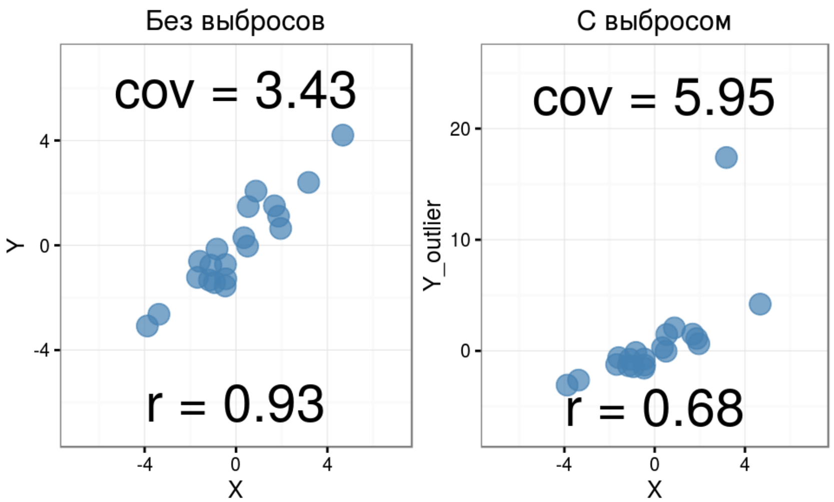
$$-1 < Cor(X, Y) < 1$$



Ковариация зависит от масштаба шкалы



Ковариация чувствительна к выбросам



Матрица ковариаций - описывает совместное варьирование множества переменных

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(y,z) \end{pmatrix}$$

Поскольку дисперсия - это ковариация признака с самим собой

$$Cov(X,X) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})}{n-1} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1} = SD^2$$

большая диагональ матрицы ковариаций содержит дисперсии признаков

$$\mathrm{C} = \begin{pmatrix} SD_x^2 & cov(x,y) & cov(x,z) \\ cov(y,x) & SD_y^2 & cov(y,z) \\ cov(z,x) & cov(z,y) & SD_z^2 \end{pmatrix}$$

- **Раз матрица ковариаций описывает изменчивость всех признаков, хорошо бы научиться находить с ее помощью максимально варьирующие направления...**

Пример работы PCA: морфометрия медуз из реки Хоксбери

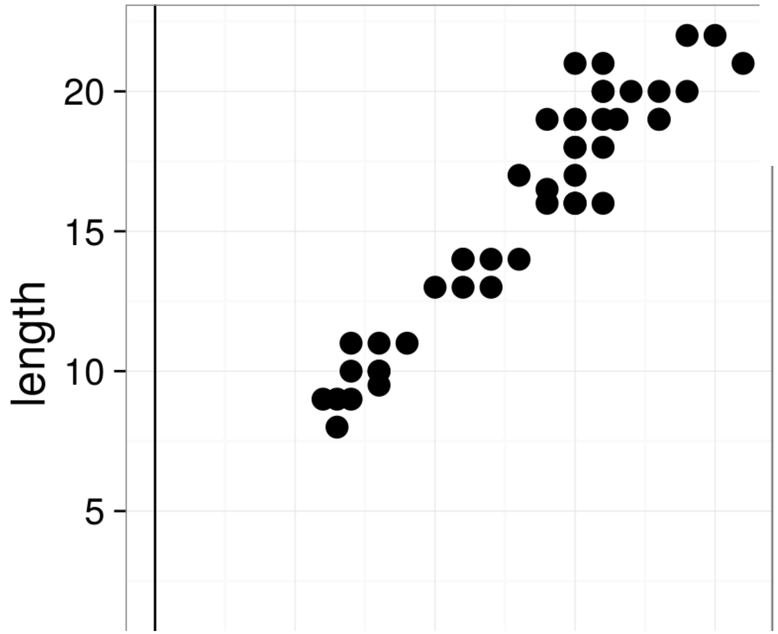
Данные о размерах медуз *Catostylus mosaicus* из реки Хоксбери (Новый Южный Уэльс, Австралия). Часть медуз собрана на острове Дангар, другая - в заливе Саламандер.

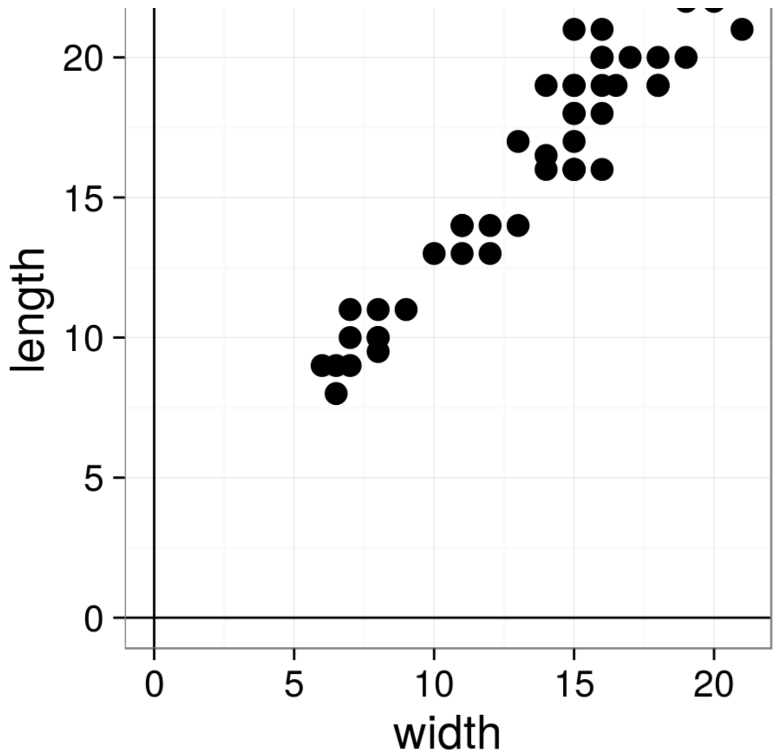
Blubber jellies! by Kirsti Scotton Flickr

Данные из Lunn & McNeil 1991

Двумерные исходные данны

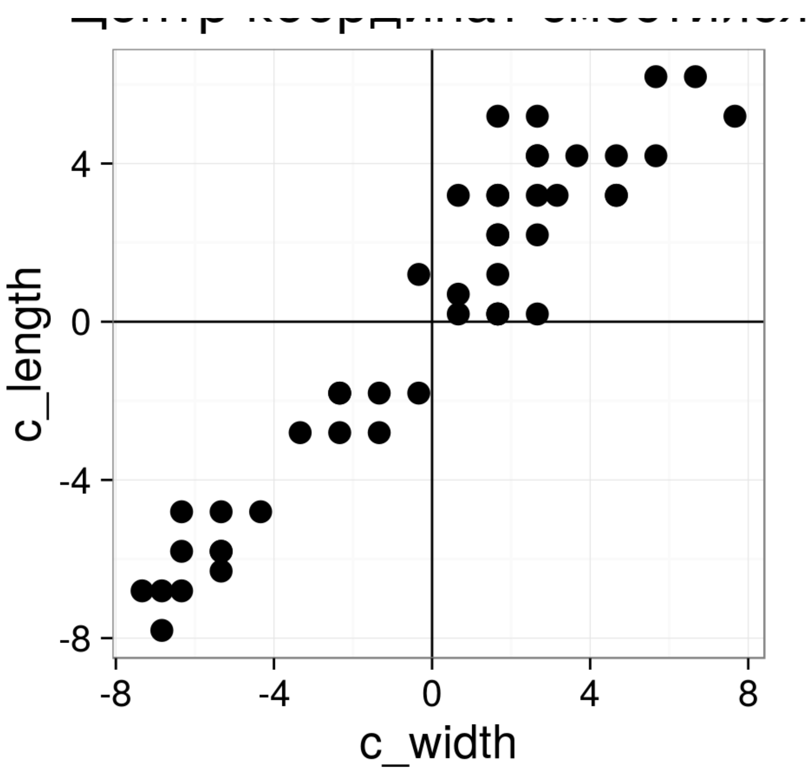
```
##     location width length
## 1          D   6.0    9.0
## 2          D   6.5    8.0
## 3          D   6.5    9.0
## 4          D   7.0    9.0
## 5          D   7.0   10.0
## 6          D   7.0   11.0
## 7          D   8.0    9.5
## 8          D   8.0   10.0
## 9          D   8.0   10.0
## 10         D   8.0   11.0
```





Центрируем исходные даннь

```
##    location c_width c_length
## 1         D   -7.34     -6.8
## 2         D   -6.84     -7.8
## 3         D   -6.84     -6.8
## 4         D   -6.34     -6.8
## 5         D   -6.34     -5.8
## 6         D   -6.34     -4.8
## 7         D   -5.34     -6.3
## 8         D   -5.34     -5.8
## 9         D   -5.34     -5.8
## 10        D   -5.34     -4.8
```



Находим новые оси с помощью собственных векторов и собственных чисел

Матрица ковариаций:

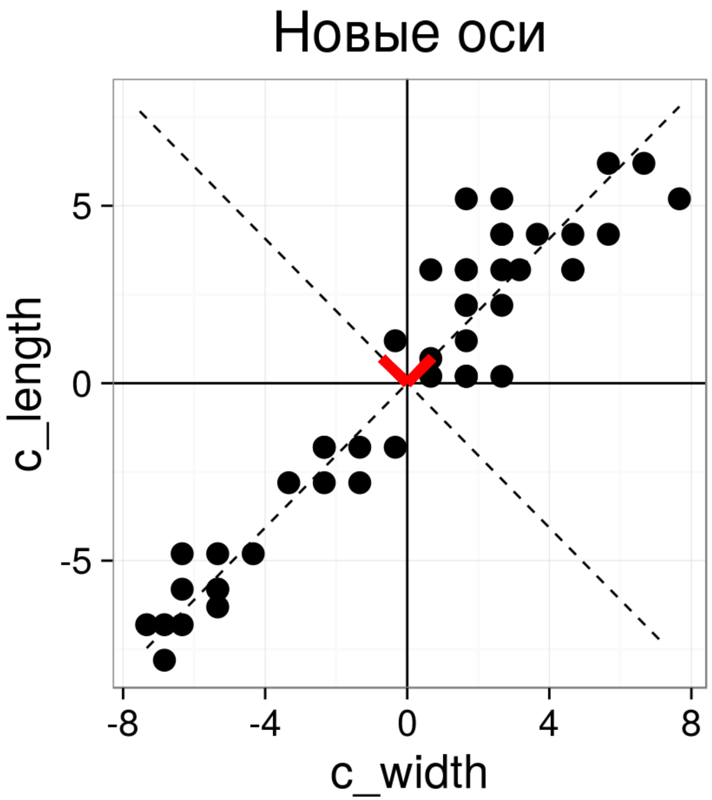
```
##           c_width c_length
## c_width     16.8     16.2
## c_length    16.2     17.4
```

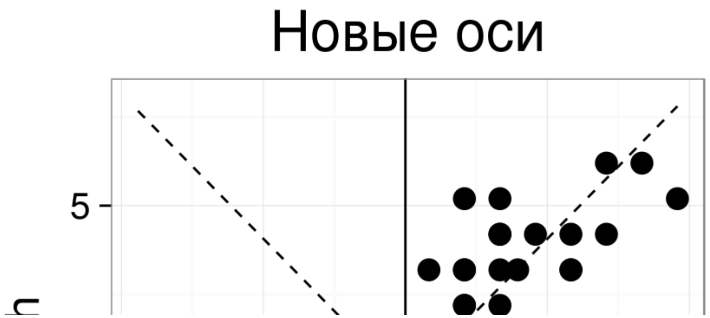
Собственные числа:

```
## [1] 33.339  0.844
```

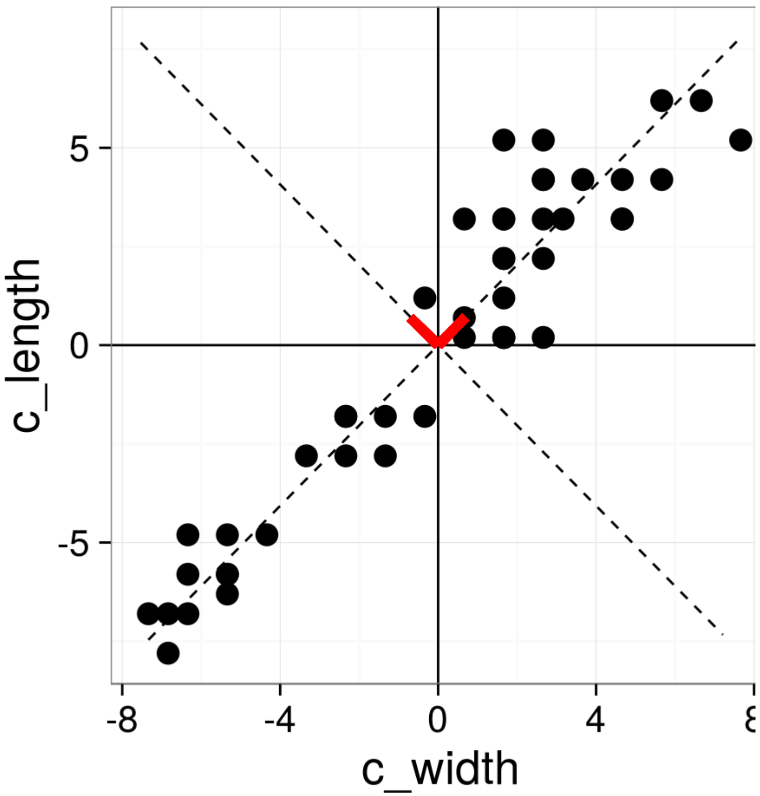
Собственные вектора:

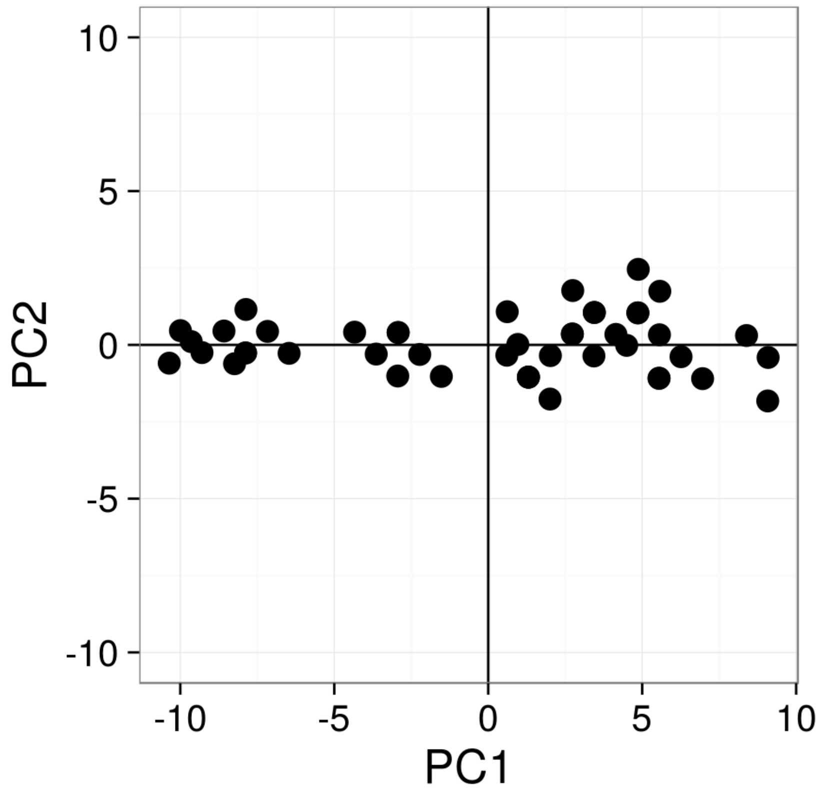
```
##        [,1]   [,2]
## [1,] 0.701 -0.713
## [2,] 0.713  0.701
```





Пересчитываем координаты точек в новом пространстве





До поворота осей

После поворота осей

Все то же самое с техникой SVD

```
SVD <-svd(jelly[,4:5])
U <- SVD$u
V <- SVD$v
D <- SVD$d
```

Сравним результаты

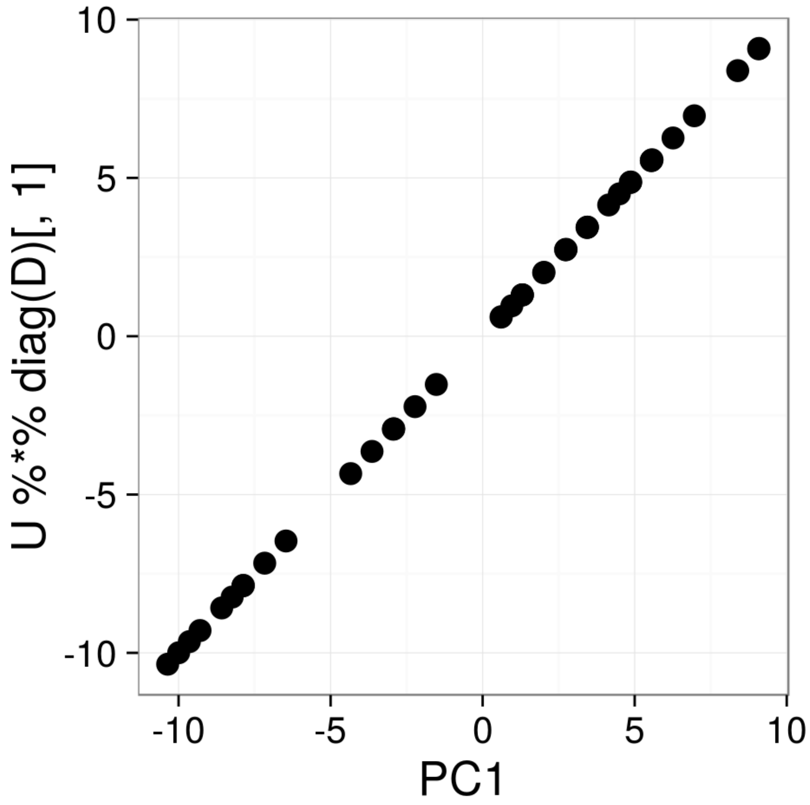
Собственные вектора, вычисленные по матрице ковариации

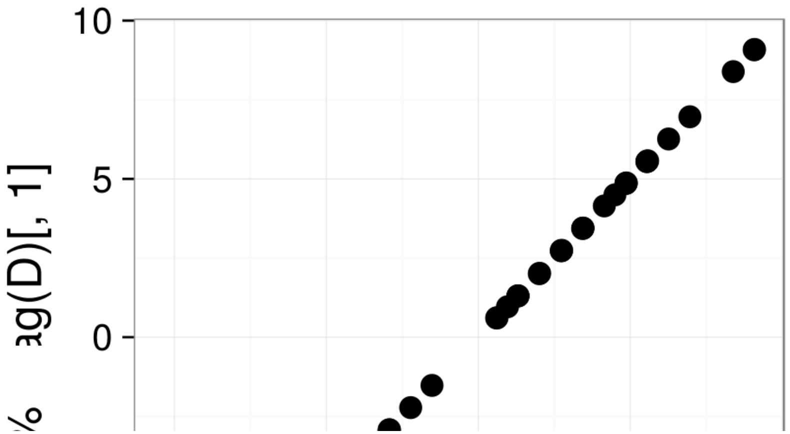
```
##        [,1]   [,2]
## [1,] 0.701 -0.713
## [2,] 0.713  0.701
```

Правая матрица сингулярных векторов

```
##        [,1]   [,2]
## [1,] 0.701 -0.713
## [2,] 0.713  0.701
```

Новые координаты в PCA и "реконструированные" значения в SVD

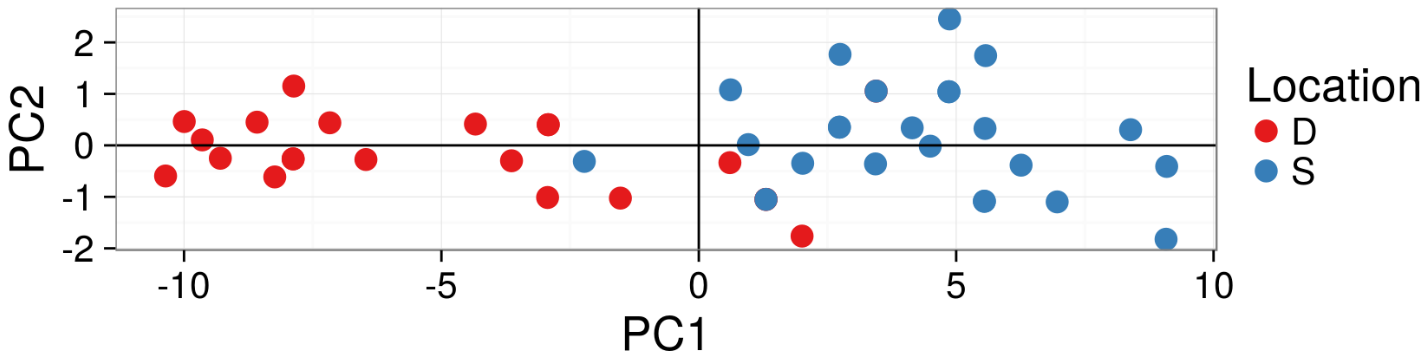




PCA и SVD

PCA - это частный случай SVD, в котором в качестве исходной матрицы взяты центрированные величины.

Медузы упорядочились по размеру



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

PCA для действительно многомерных данных

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

Paleo Diet by zsoolt on Flickr

Данные из Weber, 1973

Открываем данные

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

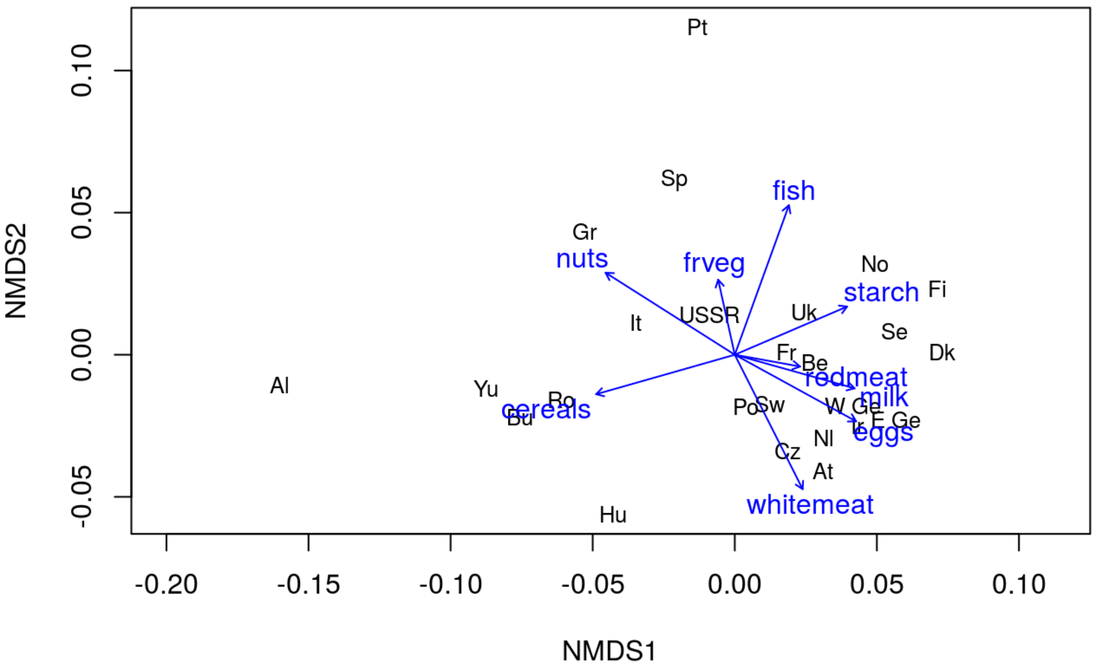
Данные из Weber, 1973

Задание

- Постройте ординацию стран при помощи nMDS с использованием евклидова расстояния
- Постройте график ординации.
- Нанесите при помощи envfit вектора изменения исходных переменных.

Решение

```
library(vegan)
ord_nmds <- metaMDS(protein[, 3:ncol(protein)], distance = "euclidean", trace = 0)
ef <- envfit(ord_nmds, protein[, 3:ncol(protein)])
plot(ord_nmds, display = "site", type = "t", cex = 0.8)
plot(ef)
```



Те же данные после применения PCA

```
library(vegan)
prot_pca <- rda(protein[, -c(1, 2)], scale = TRUE)
summary(prot_pca)
```

Собственные вектора и собственные числа

```
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 ...
```

Свойства главных компонент (Principal Components)

- Не скоррелированы друг с другом (ортогональны)
- Вдоль компонент максимальный разброс
- Чем больше собственное число (**Eigenvalue**), тем больше дисперсия вдоль оси (**Proportion Explained [variance]**)

Сколько компонент нужно оставить?

- Эмпирические правила
  - Объясняют больше чем по Broken Stick Model.
  - Компоненты у которых соб. число > 1 (правило Кайзера-Гатмана)
  - В сумме объясняют заданный % от общей изменчивости (60-80%) - слишком субъективно
- Это условности!

Постройте график собственных чисел в ggplot

Собственные числа можно добыть так:

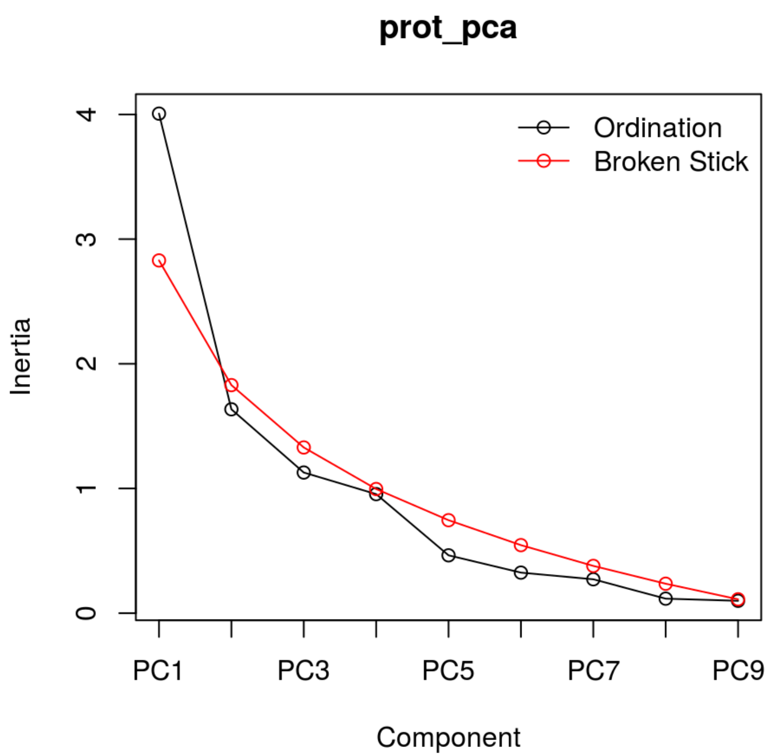
```
eigenvals(prot_pca) # собственные числа
```

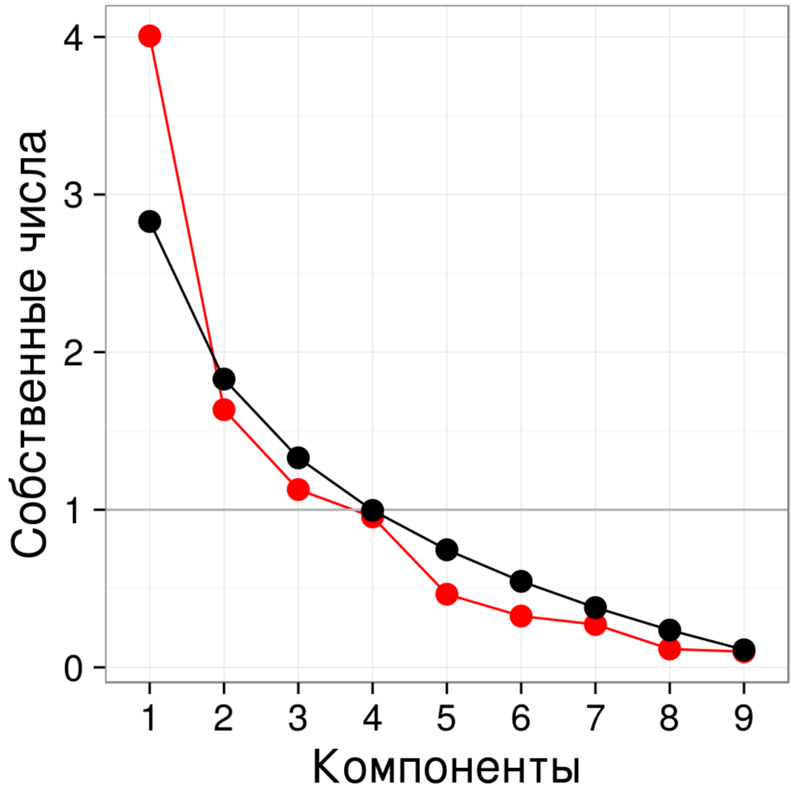
```
##  PC1  PC2  PC3  PC4  PC5  PC6  PC7  PC8  PC9
## 4.01 1.63 1.13 0.95 0.46 0.33 0.27 0.12 0.10
```

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

```
##   PC1   PC2   PC3   PC4   PC5   PC6   PC7   PC8   PC9
## 2.829 1.829 1.329 0.996 0.746 0.546 0.379 0.236 0.111
```

Графики собственных чисел





Интерпретация компонент

Факторные нагрузки оценивают вклады переменных в изменчивость по главной компоненте

- Модуль значения нагрузки - величина вклада
- Знак значения нагрузки - направление вклада

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

```
##              PC1     PC2     PC3
## redmeat   -0.303 -0.0563 -0.2976
## whitemeat -0.311 -0.2369  0.6239
## eggs      -0.427 -0.0353  0.1815
## milk      -0.378 -0.1846 -0.3857
## fish      -0.136  0.6468 -0.3213
## cereals    0.438 -0.2335  0.0959
## starch    -0.297  0.3528  0.2430
## nuts       0.420  0.1433 -0.0544
## frveg      0.110  0.5362  0.4076
## attr(,"const")
## [1] 3.83
```

Первая главная компонента:

Высокие **положительные нагрузки по первой главной компоненте** у переменных `cereals` и `nuts`. Значит, чем больше значение первой компоненты, тем больше потребление злаков и орехов.

Высокие **отрицательные нагрузки** у переменных `eggs`, `milk`, `whitemeat`, `redmeat`. Значит, чем меньше значение первой компоненты, тем больше потребление яиц, молока, белого и красного мяса.

- Т.е. первую компоненту можно назвать "Мясо - злаки и орехи"

Интерпретация компонент

Факторные нагрузки оценивают вклады переменных в изменчивость по главной компоненте

- Модуль значения нагрузки - величина вклада
- Знак значения нагрузки - направление вклада

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

```
##               PC1     PC2     PC3
## redmeat   -0.303 -0.0563 -0.2976
## whitemeat -0.311 -0.2369  0.6239
## eggs      -0.427 -0.0353  0.1815
## milk      -0.378 -0.1846 -0.3857
## fish      -0.136  0.6468 -0.3213
## cereals    0.438 -0.2335  0.0959
## starch    -0.297  0.3528  0.2430
## nuts       0.420  0.1433 -0.0544
## frveg      0.110  0.5362  0.4076
## attr(,"const")
## [1] 3.83
```

Вторая главная компонента:

Высокие **положительные нагрузки по второй главной компоненте** у переменных `fish`, `frveg`. Значит, чем больше значение второй компоненты, тем больше потребление рыбы, овощей.

Высоких **отрицательных нагрузкок по второй главной компоненте** нет ни у одной из переменных.

- Т.е. вторую компоненту можно назвать "Потребление рыбы и овощей"

Параметр `scaling`

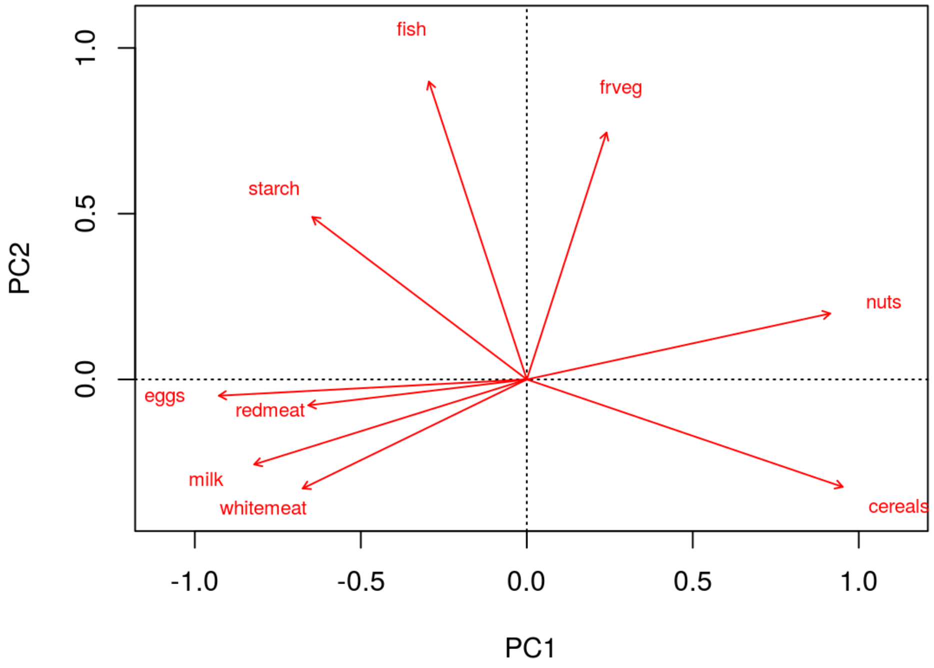
Внимание! Координаты объектов или переменных можно получить в нескольких вариантах, отличающихся масштабом. От этого масштаба будет зависеть интерпретация.

scaling	**Название графика**	**Масштаб**	**Расстояния между объектами**	**Углы между векторами**
1	биплот расстояний	координаты объектов масштабированы (х корень из соб. чисел)	апроксимируют евклидовы	нет смысла
2	биплот корреляций	координаты признаков масштабированы (х корень из соб. чисел)	НЕ апроксимируют евклидовы	отражают корреляции
3		масштабированы координаты объектов и признаков (х корень 4-й степени из соб. чисел)		
0		нет масштабирования		

Можно нарисовать график факторных нагрузок

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

```
##                PC1     PC2     PC3
## redmeat   -0.774 -0.0919 -0.4039
## whitemeat -0.794 -0.3870  0.8467
## eggs      -1.091 -0.0577  0.2464
## milk      -0.966 -0.3016 -0.5234
## fish      -0.347  1.0569 -0.4360
## cereals    1.120 -0.3815  0.1302
## starch    -0.760  0.5765  0.3298
## nuts       1.075  0.2342 -0.0738
## frveg      0.282  0.8761  0.5531
## attr(,"const")
## [1] 3.83
```




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

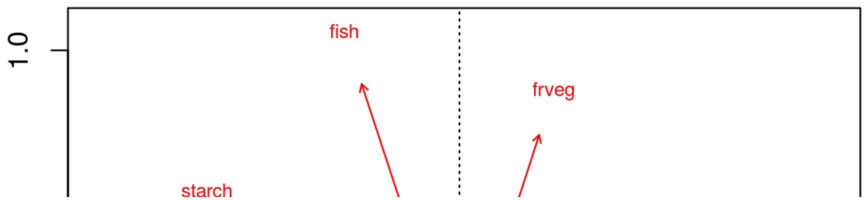
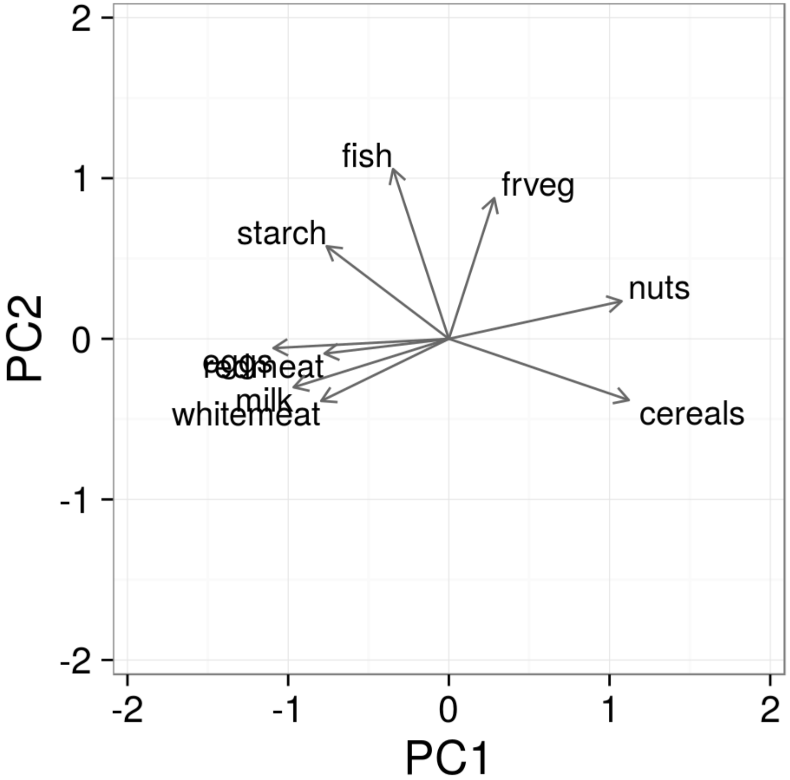


График факторных нагрузок в ggplot



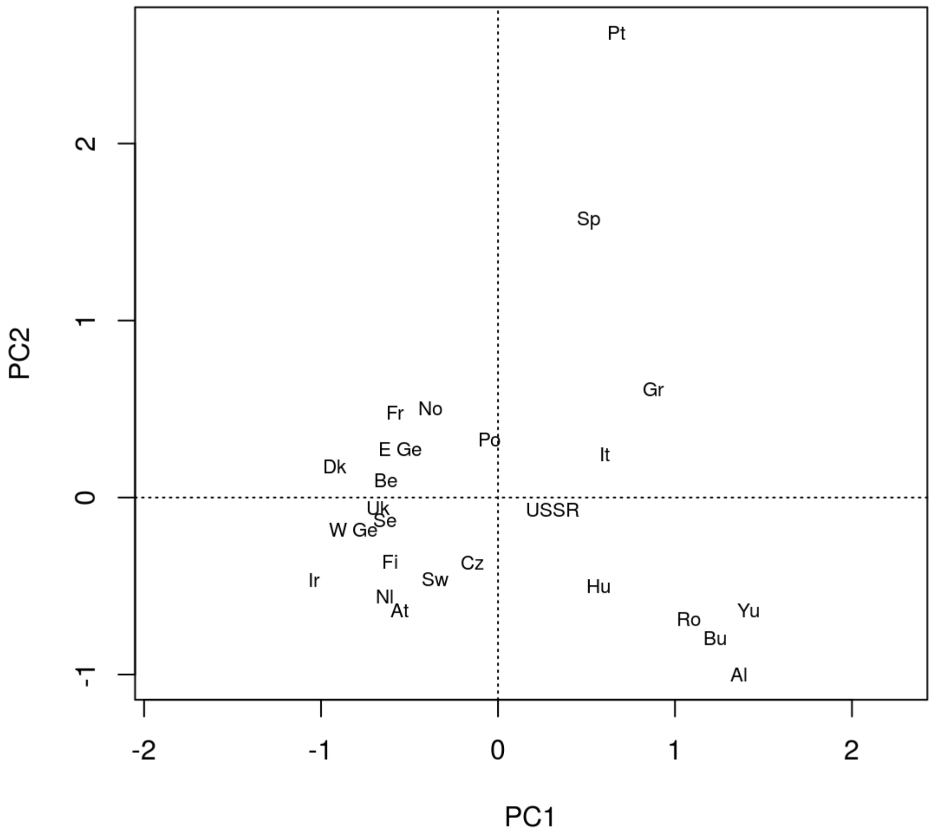
Ординация объектов

Факторные координаты отражают положение объектов в пространстве главных компонент

```
# Значения факторов (= факторные координаты)
head(scores(prot_pca, display = "sites",
choices = c(1, 2, 3), scaling = 1))
```

```
##         PC1     PC2     PC3
## Al  0.9091 -0.4253 -0.4594
## At -0.3711 -0.2716  0.3490
## Be -0.4231  0.0416  0.0565
## Bu  0.8175 -0.3394  0.0395
## Cz -0.0966 -0.1572  0.3120
## Dk -0.6170  0.0745 -0.1962
```

```
biplot(prot_pca, display = "sites")
```



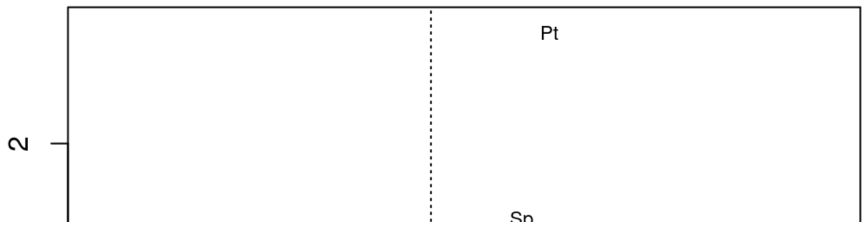
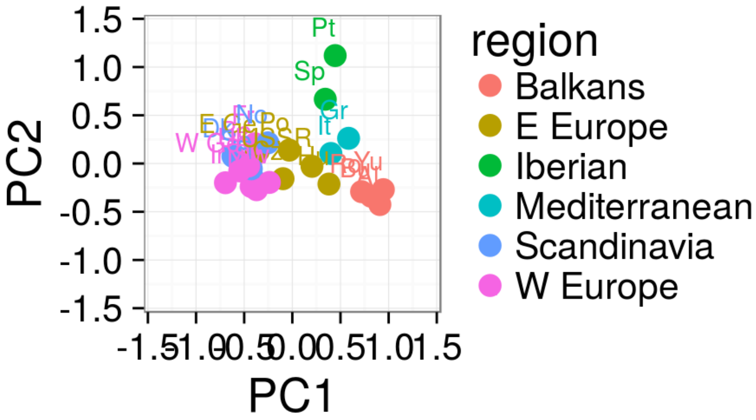
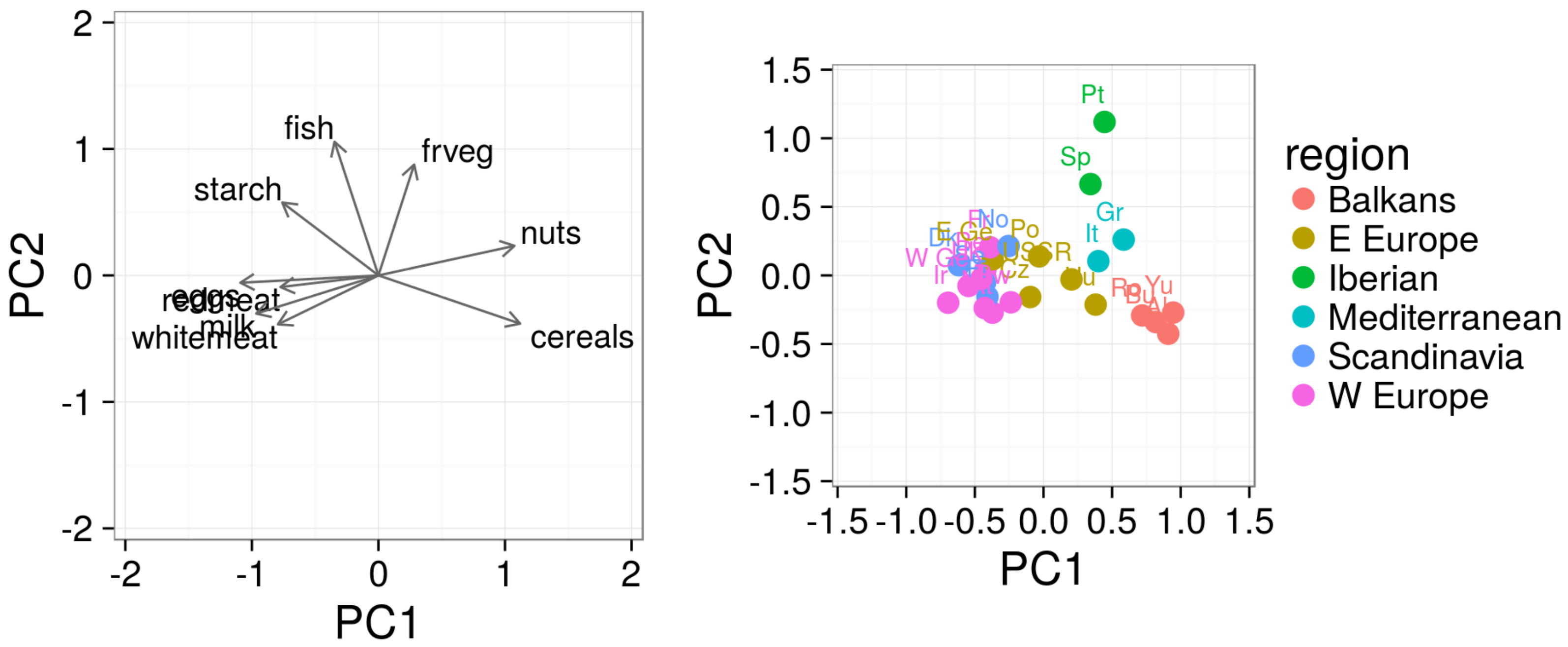


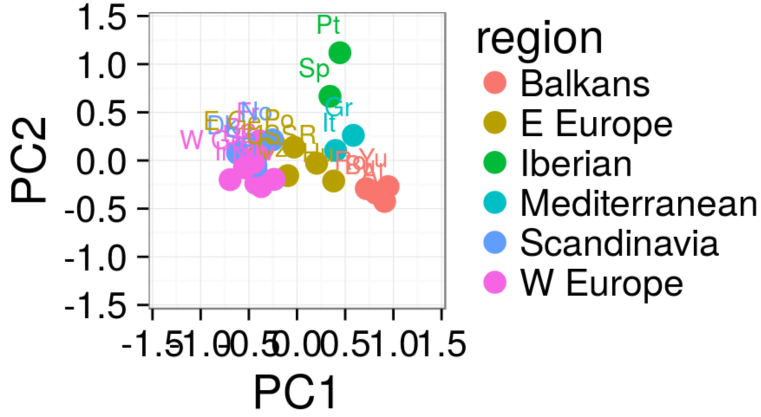
График ординации в ggplot

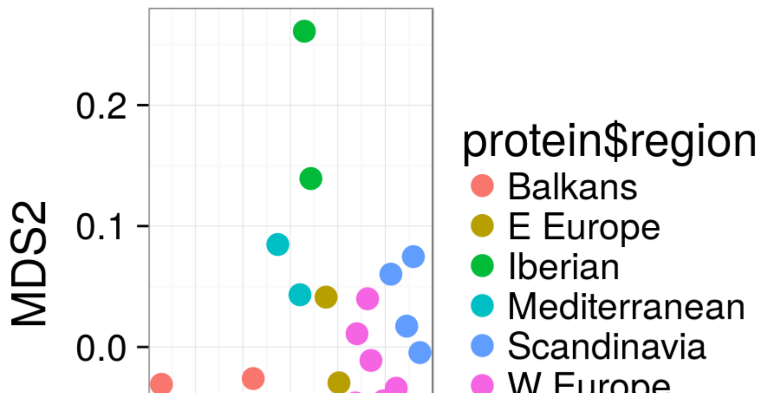


Несколько графиков рядом



В данном случае, результаты PCA похожи на nMDS





В этом примере результаты PCA похожи на nMDS потому что:

- исходные признаки - количественные, нормально распределенные переменные, связанные друг с другом линейно. Для ординации объектов с такими признаками подходит PCA. Для описания различий между объектами подходит евклидово расстояние
- для данной nMDS-ординации использовано евклидово расстояние
- расстояния между объектами на любой ординации PCA соответствуют их евклидовым расстояниям в пространстве главных компонент

Создание комплексных 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  0.9091 -0.4253 -0.4594
## At -0.3711 -0.2716  0.3490
## Be -0.4231  0.0416  0.0565
## Bu  0.8175 -0.3394  0.0395
## Cz -0.0966 -0.1572  0.3120
## Dk -0.6170  0.0745 -0.1962
```

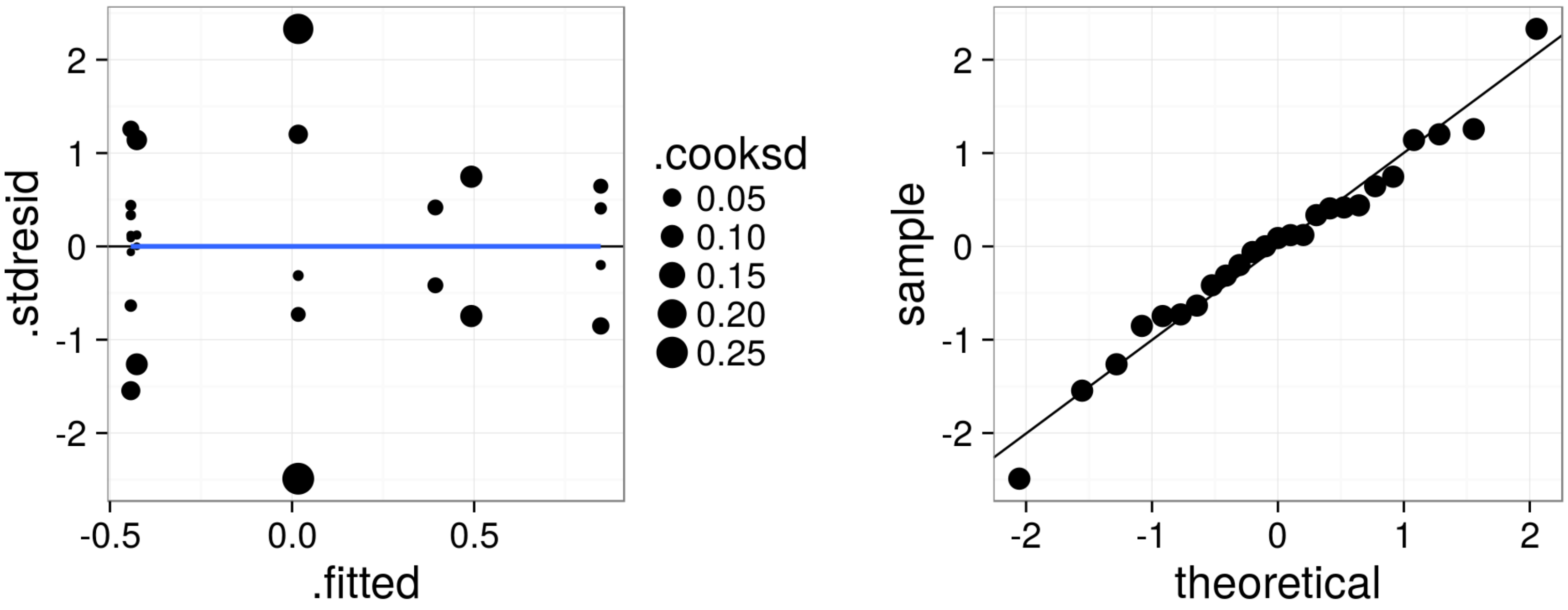
При помощи дисперсионного анализа проверьте, различается ли значение первой главной компоненты ("Мясо - злаки и орехи") между разными регионами Европы

```
# Значения факторов (= факторные координаты)
df <- data.frame(region = protein$region,
  scores(prot_pca, display = "sites", choices = c(1, 2, 3), scaling = 1))
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   5.97    1.19    39.3 0.0000000022 ***
## Residuals 19   0.58    0.03
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

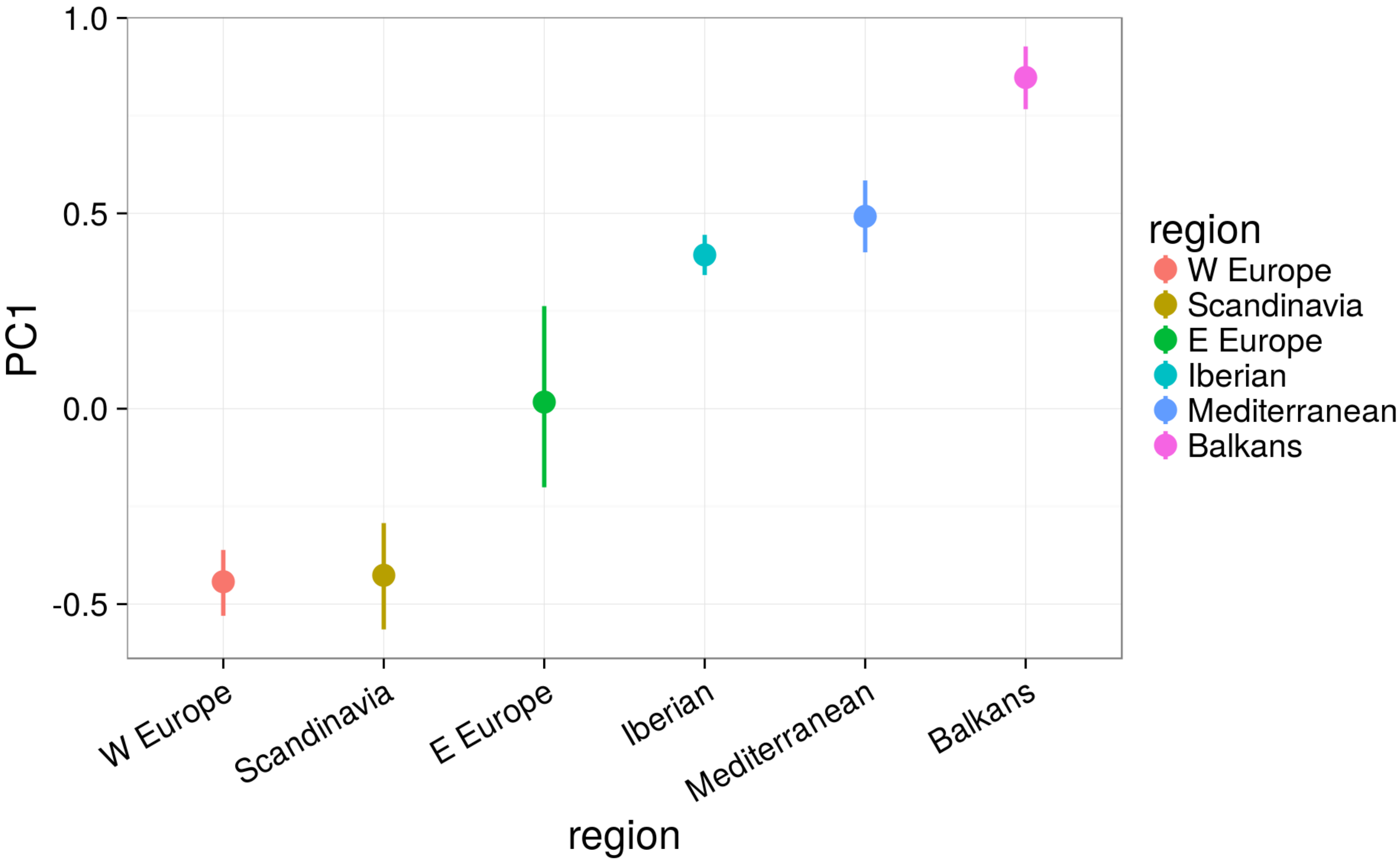
- Регионы Европы различаются по потреблению мяса, злаков и орехов

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           -0.8307 -1.2001 -0.4614 0.000
## Iberian-Balkans            -0.4541 -0.9309  0.0227 0.067
## Mediterranean-Balkans      -0.3555 -0.8323  0.1214 0.221
## Scandinavia-Balkans        -1.2740 -1.6633 -0.8847 0.000
## W Europe-Balkans           -1.2905 -1.6276 -0.9533 0.000
## Iberian-E Europe            0.3766 -0.0840  0.8372 0.150
## Mediterranean-E Europe      0.4753  0.0146  0.9359 0.041
## Scandinavia-E Europe       -0.4433 -0.8126 -0.0740 0.013
## W Europe-E Europe          -0.4597 -0.7736 -0.1459 0.002
## Mediterranean-Iberian       0.0987 -0.4519  0.6492 0.992
## Scandinavia-Iberian        -0.8199 -1.2967 -0.3431 0.000
## W Europe-Iberian           -0.8363 -1.2716 -0.4011 0.000
## Scandinavia-Mediterranean -0.9186 -1.3954 -0.4418 0.000
## W Europe-Mediterranean     -0.9350 -1.3703 -0.4998 0.000
## W Europe-Scandinavia       -0.0164 -0.3536  0.3207 1.000
```

Summary

- 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

Что почитать

- 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 [WWW Document], n.d. 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.