



# Ординация. Многомерное шкалирование.

Анализ и визуализация многомерных данных с  
использованием R

Марина Варфоломеева, Вадим Хайтов

- Общие принципы ординации объектов
- Неметрическое многомерное шкалирование
- Как работает неметрическое многомерное шкалирование
- Оценка качества подгонки ординации
- Сравнение ординаций
- Трактовка результатов ординации

### Вы сможете

- Построить диаграмму простейшей непрямой ординации.
- Построить диаграмму nMDS.
- Охарактеризовать качество ординации с помощью величины стресса.
- Сравнить результаты нескольких ординаций

# Общие принципы ординации объектов

# Классификация методов ординации

**Ординация** (лат. *ordinatio* — расположение в порядке) — это упорядочивание объектов вдоль некоторых осей, определяющих варьирование свойств этих объектов.

Элемент	(син)экология	морфология	молекулярная биология
Объекты	площадки / пробы	особи / клетки и т.п.	особи / клетки и т.п.
Собственные свойства объектов	численность / биомасса особей разных видов	измерения, свойства	интенсивность экспрессии гена/пептида
Внешние факторы	свойства среды	свойства среды / особей / клеток	свойства среды / особей / клеток

---

Методы ординации делят на две большие группы (ter Braak, Prentice, 1988):

- Прямая ординация (*direct ordination = direct gradient analysis*) — когда мы проверяем, влияют ли непосредственно на свойства объектов измеренные внешние факторы.
- Непрямая ординация (*indirect ordination = indirect gradient analysis*) — когда мы пытаемся найти какие-то закономерности изменения свойств самих объектов, и затем пытаемся связать их с внешними факторами.

# Прямая ординация

**Прямая ординация** — упорядочивание объектов с разными свойствами вдоль осей, которые отражают значения переменных. (Например, упорядочение площадок с разным обилием видов в зависимости от параметров среды).

В самом простом случае — это ординация в осях самих внешних переменных. Но, поскольку обычно переменных больше чем 2–3, обычно пользуются осями, искусственно созданными как функции от нескольких переменных.

Синоним — ограниченная ординация (*constrained ordination*).

Методы прямой ординации позволяют тестировать гипотезы о влиянии различных внешних переменных на свойства объектов:

- анализ избыточности (*Redundancy Analysis*, RDA)
- канонический корреспондентный анализ (*Canonical Correspondence Analysis*, CCA).

# Непрямая ординация

**Непрямая ординация** — упорядочивание происходит вдоль направления изменения свойств объектов.

Используются только признаки объектов, и только затем для интерпретации результатов используется информация о значениях других переменных.

Синоним — неограниченная ординация (*unconstrained ordination*).

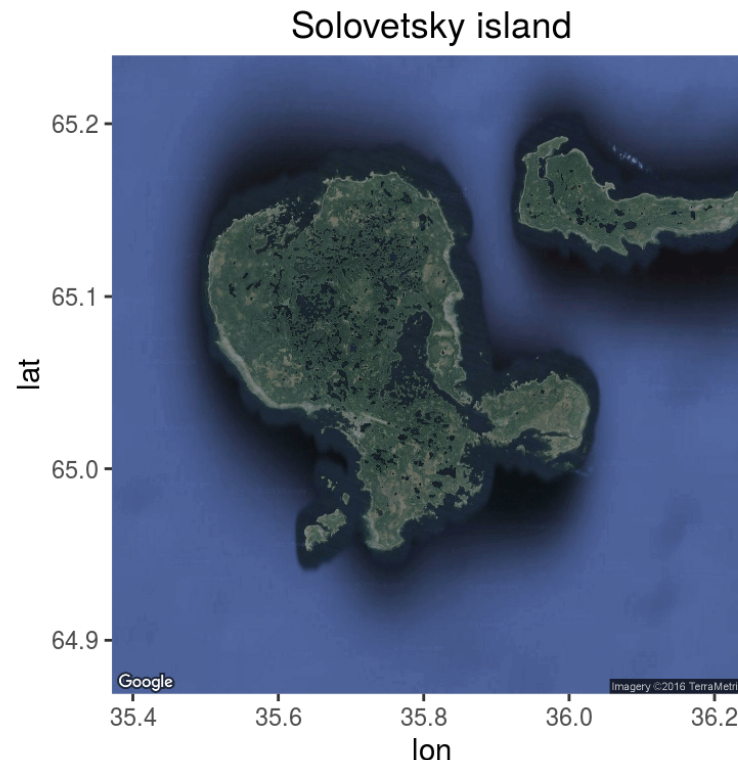
Методы не прямой ординации:

- неметрическое многомерное шкалирование (*non-Metric Multidimensional Scaling*, nMDS)
- анализ главных компонент (*Principal Component Analysis*, PCA)
- корреспондентный анализ (*Correspondence Analysis*, CA).

## Пример: Сообщества бентоса акватории Долгой губы (о. Б. Соловецкий, Белое море)

В этом примере используется сокращенный набор данных о бентосе Долгой губы (Нинбург, 1990; Хайтов и др., 2013) — только численность наиболее обильных видов на 68 станциях.

- `dolg_abundance.txt` — данные об обилии 25 видов
- `dolg_hydrology.txt` — данные о 4 гидрологических характеристиках: глубина, Температура придонной воды, Соленость, Степень гидратации грунта



## Читаем данные

```
abund <- read.table("data/dolg_abundance.txt", skip = 1, header = TRUE,  
  sep = ";")  
hydrol <- read.table("data/dolg_hydrology.txt", skip = 1, header = TRUE,  
  sep = ";")
```



## Задание: Прямая ординация станций в осях Температуры и Солености

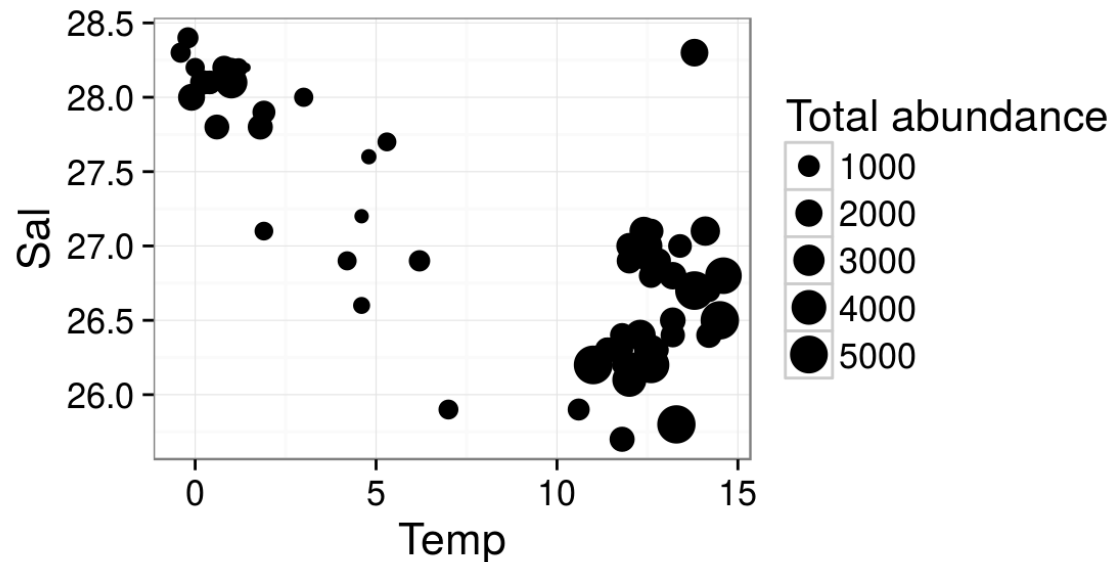
Постройте диаграмму, отражающую ординацию станций в осях Температуры и Солености.

Модифицируйте график так, чтобы была еще видна связь с суммарным обилием видов в пробах.

# Прямая ординация станций в осях Температуры и Солености

## Решение

```
library(ggplot2)
tot_abund <- apply(abund[, -1], 1, FUN = sum)
theme_set(theme_bw(base_size = 16))
ggplot(hydrol, aes(x = Temp, y = Sal, size = tot_abund)) + geom_point() +
  labs(size = "Total abundance")
```



# Висконсинская полярная ординация

Это простейший прием не прямой ординации

Шаг 1. Подготовим данные для анализа.

```
log_abund <- log(abund[, -1] + 1)  
row.names(log_abund) <- abund$Station
```

# Висконсинская полярная ординация

Шаг 2. Вычисление матрицы сходства/различия между объектами.  
Из дидактических соображений возьмем матрицу Евклидовых расстояний.

```
library(vegan)

## Loading required package: permute

## Loading required package: lattice

## This is vegan 2.3-3

E_dist <- vegdist(log_abund, method = "euclidean")
# E_dist
```

# Висконсинская полярная ординация

Шаг 3. Найдем наиболее различающиеся объекты (максимальное Евклидово расстояние между ними).

```
dist <- unfolding(log_abund) # Пользовательская функция
```

```
dist[dist$Distance == max(dist$Distance), ]
```

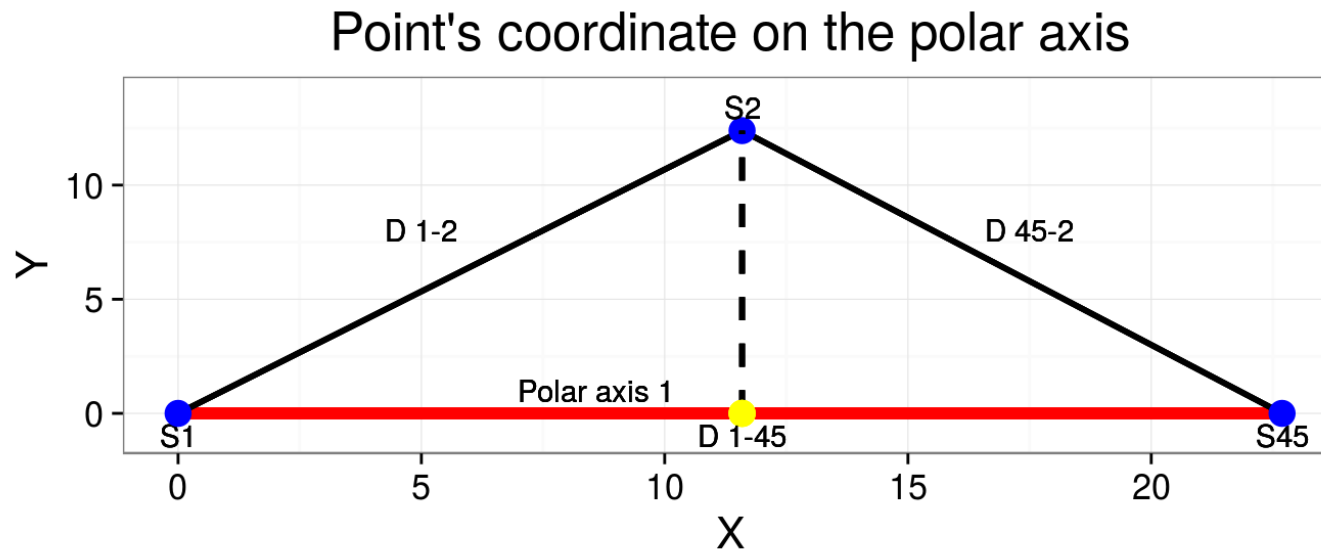
```
##      i Object_j Object_k Distance  
## 44 44         1        45      22.7
```

Эти два объекта и задают ось, вдоль которой будет производиться ординация.

# Висконсинская полярная ординация

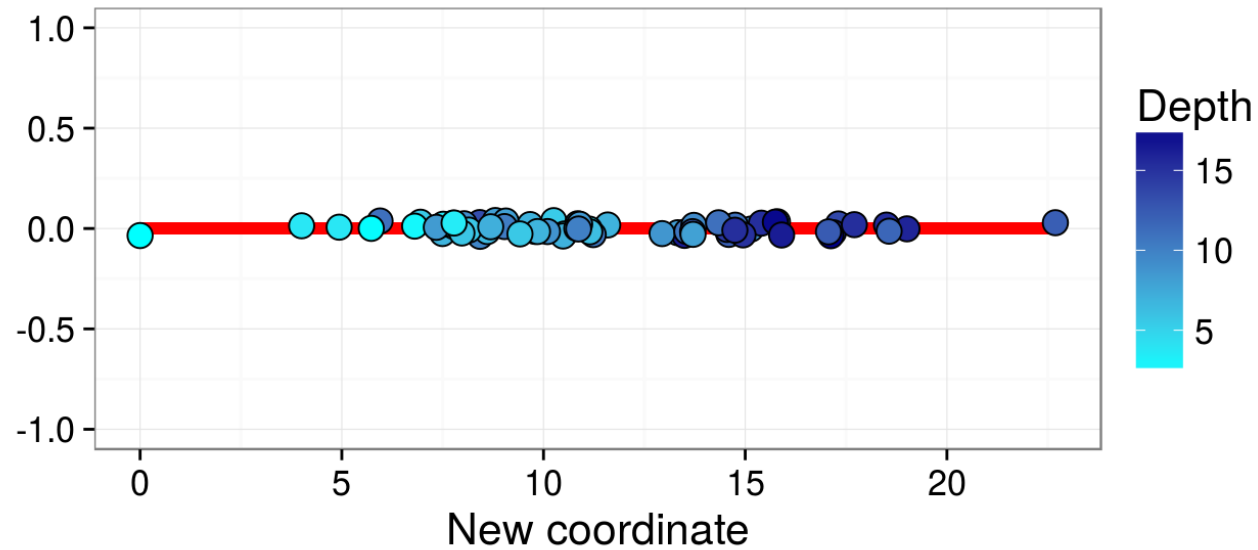
Шаг 4. Координаты каждого из объектов на этой оси могут быть найдены, согласно правилам треугольника

Возьмем любую другую точку, например "S2"



# Висконсинская полярная ординация

Шаг 5. Вычисляем координаты на полярной оси для каждого объекта.



# Висконсинская полярная ординация

Шаг 6. При необходимости, находим следующую пару наиболее несходных объектов и вычисляем координаты для каждой точки на новой оси.

*At!* Одновременное изображение информации об обеих полярных осях на одной диаграмме невозможно!

Мы не знаем как взаимосвязаны эти оси. Они могут быть неортогональны.



Non-metric multidimensional scaling (nMDS)

**Неметрическое многомерное  
шкалирование**

## Результаты функции metaMDS ( )

```
library(vegan)
ord <- metaMDS(log_abund, distance = "bray", k = 2) # результаты
сохраняются в объекте ord
```

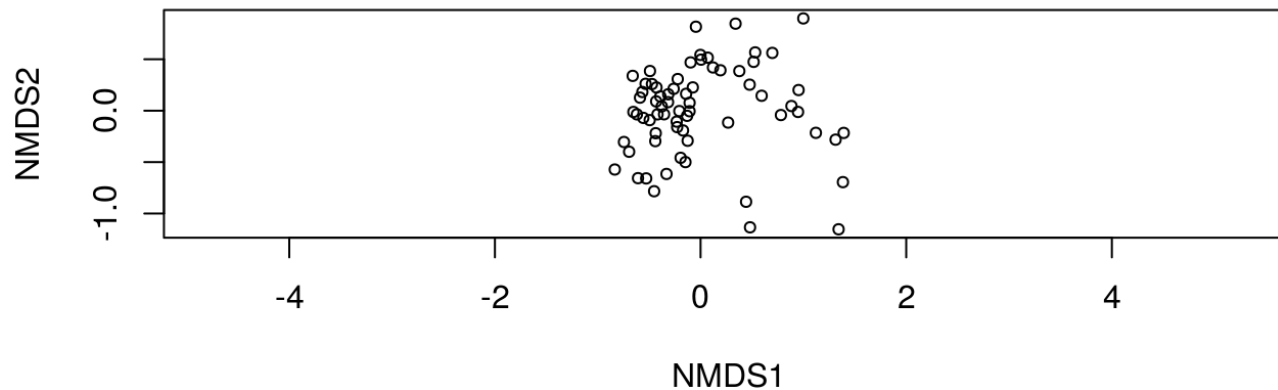
```
## Run 0 stress 0.16
## Run 1 stress 0.159
## ... New best solution
## ... procrustes: rmse 0.0204  max resid 0.154
## Run 2 stress 0.159
## ... New best solution
## ... procrustes: rmse 0.000733  max resid 0.00454
## *** Solution reached
```

```
ord
```

```
##
## Call:
## metaMDS(comm = log_abund, distance = "bray", k = 2)
##
## global Multidimensional Scaling using monoMDS
##
## Data:      log_abund
## Distance: bray
##
## Dimensions: 2
## Stress:     0.159
```

# Графическое представление результатов средствами пакета **vegan**

```
ordiplot(ord, display = "sites")
```



*# text(ord) #Можно добавить обозначения сайтов (объектов)*

## Задание: Графическое представление результатов средствами **ggplot2**

- Исследуйте объект `ord` и извлеките из него данные с новыми координатами
- Постройте график ординации при помощи пакета `ggplot2`
- Раскрасьте точки на ординации согласно глубине (данные в `hydro1`)

## Решение

Рассчитанные при помощи nMDS координаты объектов хранятся в элементе `points`

`ord$points`

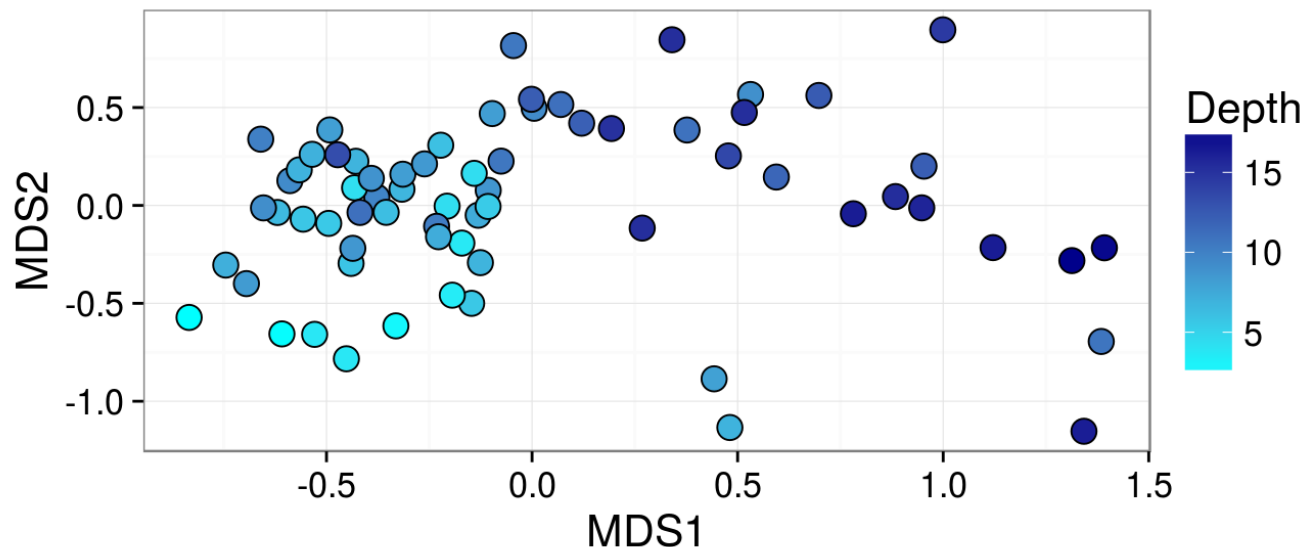
##		MDS1	MDS2
##	S1	-0.83420	-0.57207
##	S2	0.48112	-1.13413
##	S3	0.94786	-0.01230
##	S4	-0.09679	0.46923
##	S5	-0.58905	0.12674
##	S6	-0.37761	0.04179
##	S7	0.88395	0.04522
##	S8	0.12070	0.42002
##	S9	-0.42854	0.22542
##	S10	-0.43173	0.09027
##	S12	1.38447	-0.69470
##	S13	0.53139	0.56658
##	S15	-0.13098	-0.05002
##	S16	1.34239	-1.15332
##	S17	-0.10662	0.07705
##	S18	-0.10700	-0.00532
##	S19	-0.31646	0.08287
##	S20	-0.22212	0.30756
##	S21	0.34038	0.84611
##	S22	0.47755	0.25292
##	S23	-0.47277	0.25892

# Решение

Координаты легко использовать для графика после преобразования в датафрейм.

```
ord_MDS <- data.frame(ord$points)
```

```
ggplot(ord_MDS, aes(x = MDS1, y = MDS2)) + geom_point(aes(fill =  
  hydrol$Depth),  
  size = 4, shape = 21) + scale_fill_gradient(low = "cyan", high =  
  "darkblue") +  
  labs(fill = "Depth")
```



## Важные свойства MDS

1. Ординация **сохраняет ранг расстояний между объектами** (похожие располагаются близко, непохожие — далеко; если объект А похож на В, больше чем на С, то и на ординации он окажется ближе к В, чем к С).
2. Значения координат точек в ординации лишены смысла (их вообще можно не приводить на итоговой ординации), имеет смысл лишь взаиморасположение объектов.
3. Облако точек в осях MDS можно вращать, перемещать, зеркально отражать. Суть ординации от этого не изменится.

# Как работает неметрическое многомерное шкалирование



## Задание:

- Создайте датафрейм, содержащий исходные данные (без логарифмирования) только по сайтам S17, S33, S37, S38, S44, S59.
- Постройте ординацию этих объектов с использованием в качестве меры различия коэффициент Брея-Куртиса.
- Измерьте линейкой расстояния между точками на ординации
- Сравните матрицу коэффициентов Брея-Куртиса и матрицу расстояний между точками на ординации.

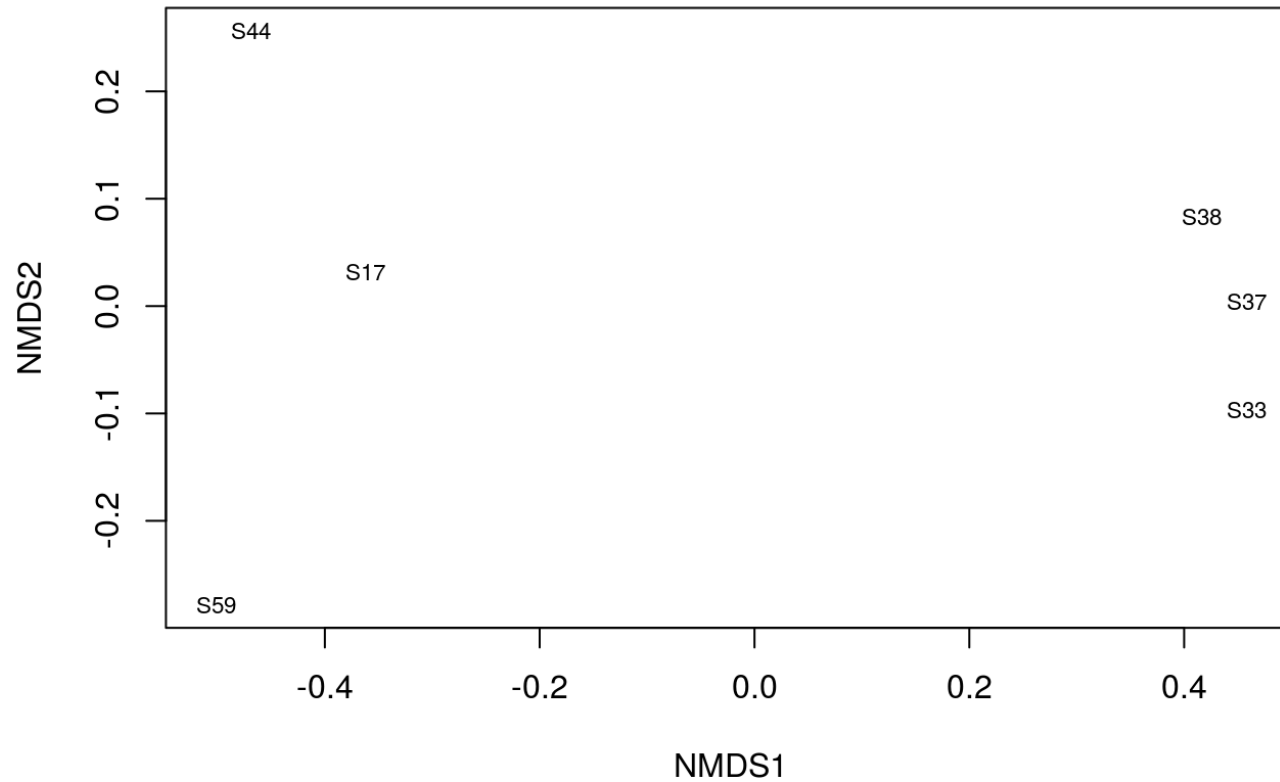
## Решение

```
obj <- c("S17", "S33", "S37", "S38", "S44", "S59")
red_abund <- log_abund[abund$Station %in% obj, ]
row.names(red_abund) <- obj
ord1 <- metaMDS(red_abund, distance = "bray")
```

```
## Run 0 stress 0
## Run 1 stress 0.0000878
## ... procrustes: rmse 0.188  max resid 0.33
## Run 2 stress 0
## ... procrustes: rmse 0.0915  max resid 0.128
## Run 3 stress 0
## ... procrustes: rmse 0.158  max resid 0.245
## Run 4 stress 0
## ... procrustes: rmse 0.121  max resid 0.16
## Run 5 stress 0
## ... procrustes: rmse 0.112  max resid 0.162
## Run 6 stress 0
## ... procrustes: rmse 0.126  max resid 0.2
## Run 7 stress 0
## ... procrustes: rmse 0.0628  max resid 0.105
## Run 8 stress 0
## ... procrustes: rmse 0.196  max resid 0.268
## Run 9 stress 0
## ... procrustes: rmse 0.115  max resid 0.204
## Run 10 stress 0
## ... procrustes: rmse 0.163  max resid 0.276
```

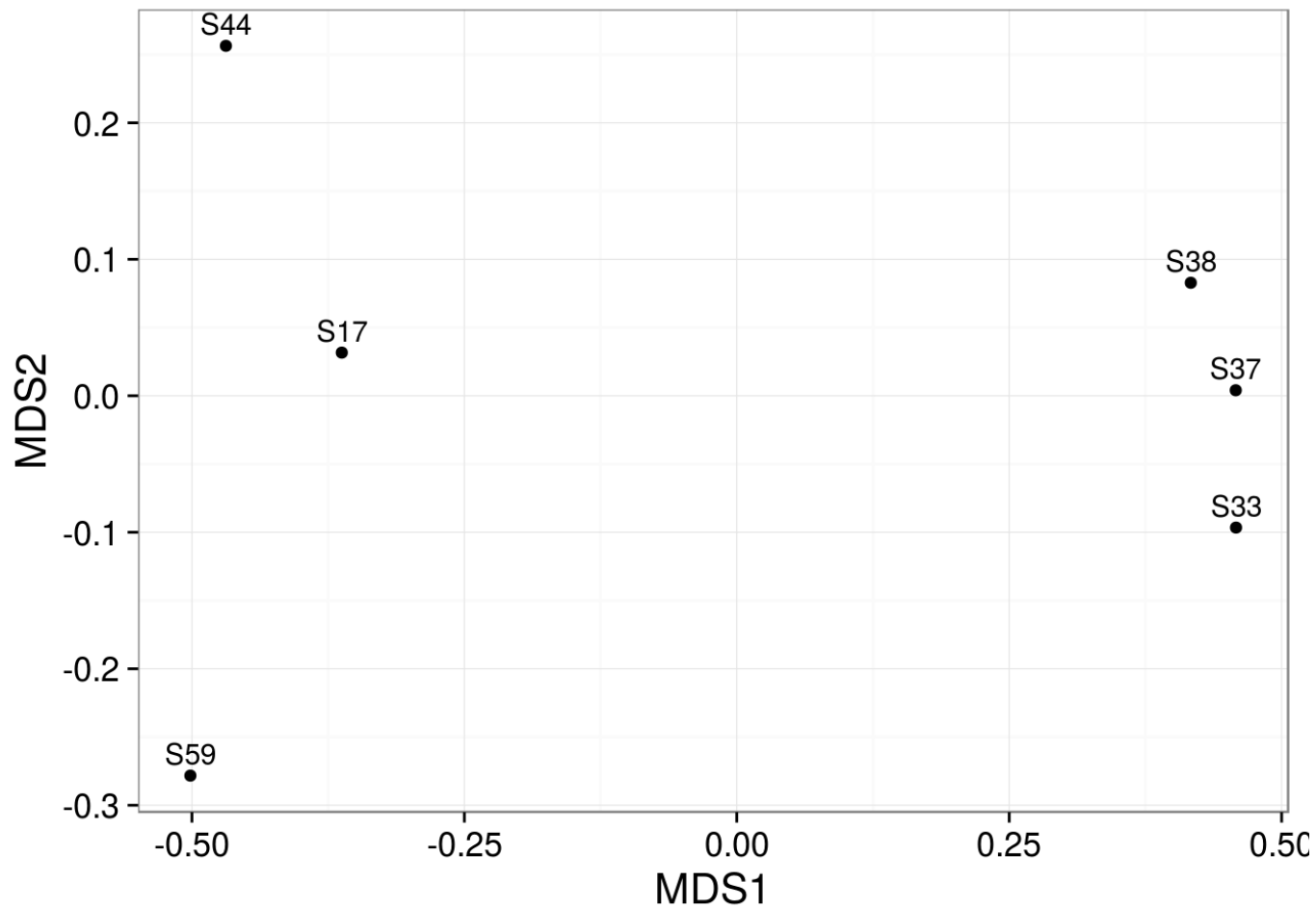
# Решение

```
ordiplot(ord1, display = "sites", type = "text")
```



## Решение

```
ggplot(data.frame(ord1$points), aes(x = MDS1, y = MDS2)) + geom_point() +  
  geom_text(aes(label = row.names(ord1$points)), vjust = -0.5)
```

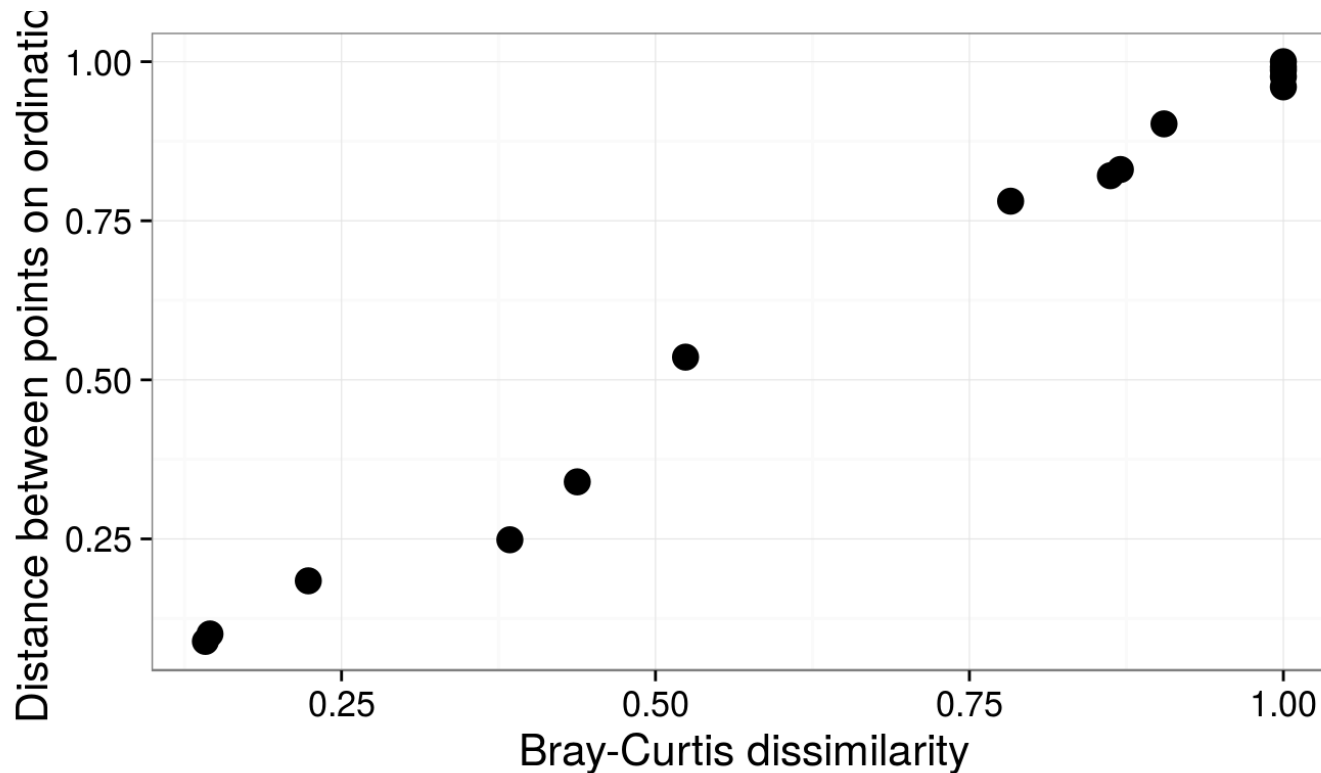


```
vegdist(red_abund[])
```

```
##          S17    S33    S37    S38    S44
```

# Взаиморасположение точек на плоскости подобно взаиморасположению точек в многомерном пространстве признаков

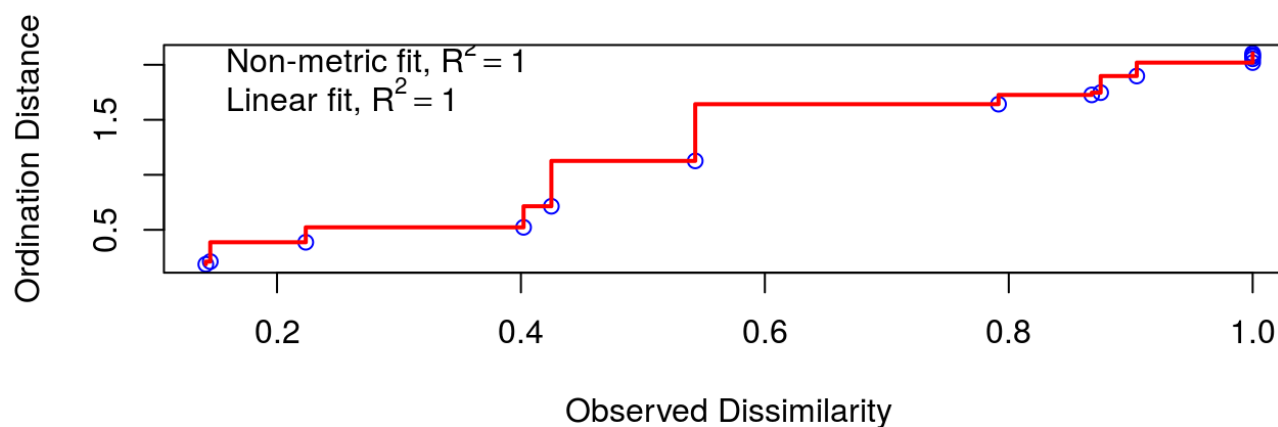
- В этом легко убедиться, построив график. Но мы видим, что расстояния не полностью совпадают



# Диаграмма Шепарда

Диаграмма Шепарда показывает соответствие расстояний на ординации и расстояний в исходном многомерном пространстве.

`stressplot(ord1)`

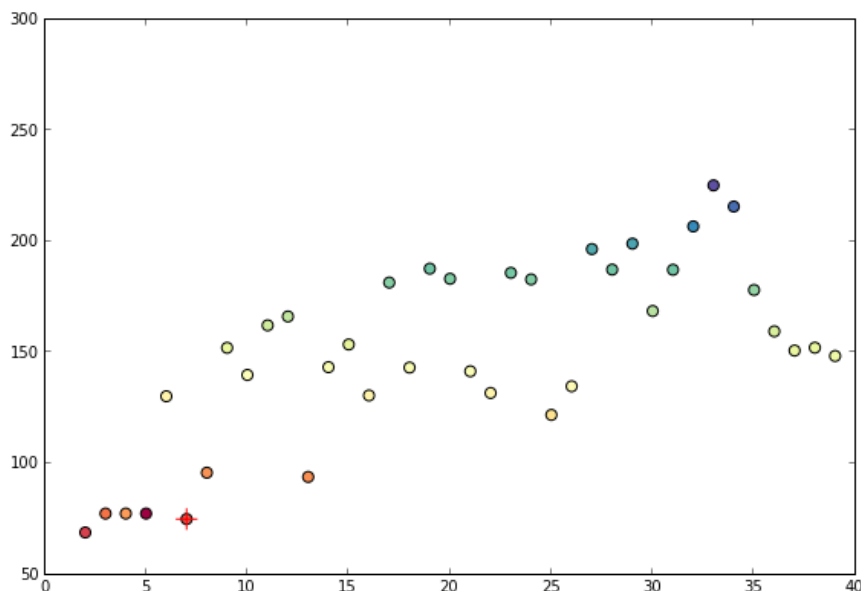


- По оси X - откладываются ранжированные (расположенные в порядке возрастания) значения D (коэффициент Брея-Куртиса)
- По оси Y- соответствующие им, значения расстояний на плоскости – Z
- Ломаная линия на диаграмме - это монотонная регрессия.

# Монотонная регрессия

**Монотонная регрессия** минимизирует сумму квадратов отклонений значений координат на ординации от значений в исходном многомерном пространстве.

При одном условии: так, чтобы при этом сохранялись монотонные отношения между ними (если возрастает расстояние в исходном пространстве, то и на ординации оно должно возрастать). Именно это свойство монотонной регрессии позволяет сохранить ранги расстояний при построении ординации.



# Stress

**Stress** — мера соответствия ординации на плоскости взаиморасположению точек в многомерном пространстве признаков

$$Stress = \sqrt{\frac{\sum (Z_i - Zh_i)^2}{\sum Z_i^2}}$$

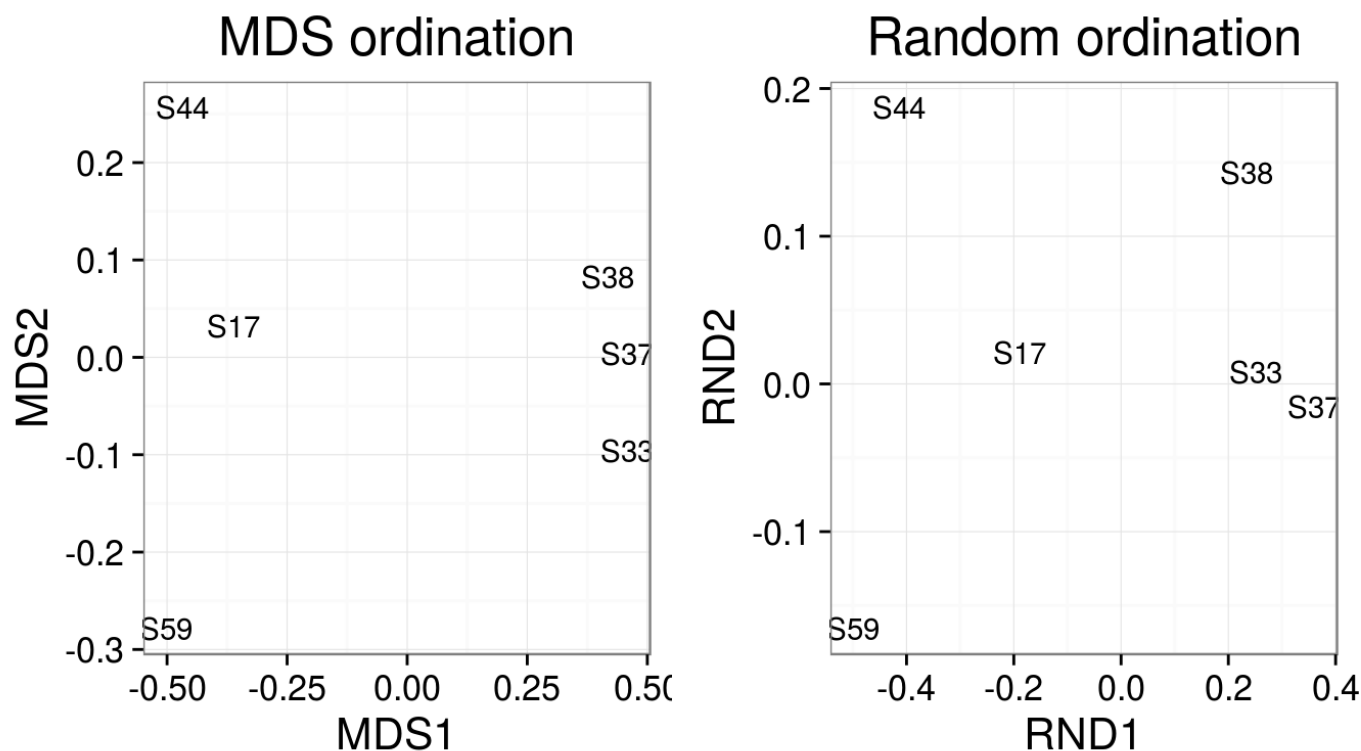
Для вычисления величины стресса необходимо построить диаграмму Шепарда и подобрать монотонную регрессию



## Посмотрим, что означает Stress.

Чтобы лучше представить себе, откуда берется стресс, оценим его величину для случайно выбранного расположения объектов по сравнению с нашей лучшей ординацией.

Создадим случайную ординацию для этого примера: передвинем все шесть точек на случайное расстояние.



# Построение монотонной регрессии

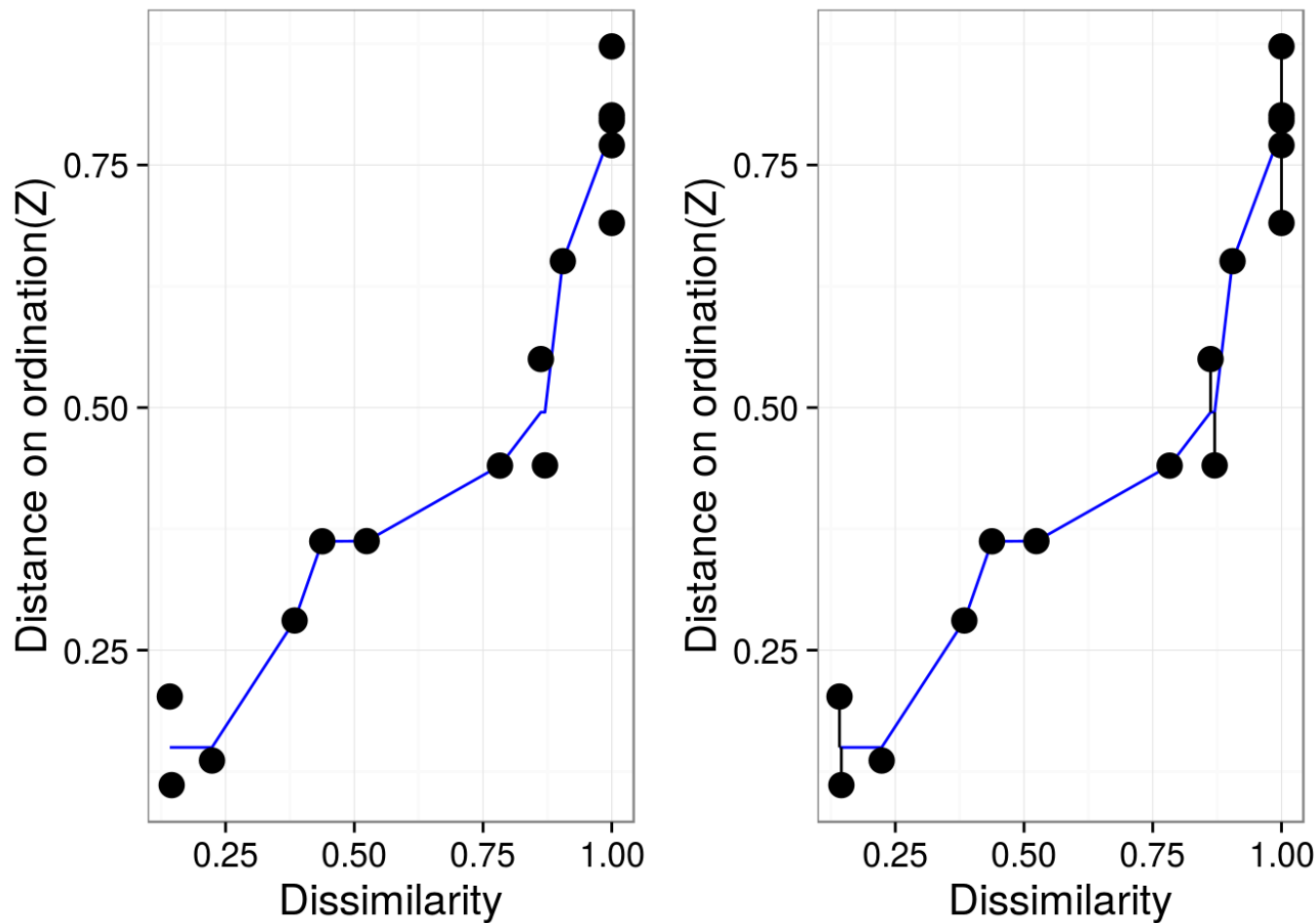
Выпишем в развернутом виде матрицу коэффициентов Брея-Куртиса между точками на ординации, а так же между случайно расположенными точками. Отсортируем их в порядке возрастания расстояния на ординации.

##	i	Object_j	Object_k	Distance	RND_Distance	
##	10	10	3	4	0.142	0.202
##	6	6	2	3	0.145	0.111
##	7	7	2	4	0.224	0.136
##	4	4	1	5	0.384	0.281
##	5	5	1	6	0.438	0.362
##	15	15	5	6	0.524	0.362
##	3	3	1	4	0.783	0.440
##	2	2	1	3	0.862	0.550
##	1	1	1	2	0.870	0.440
##	13	13	4	5	0.905	0.651
##	8	8	2	5	1.000	0.690
##	9	9	2	6	1.000	0.770
##	11	11	3	5	1.000	0.801
##	12	12	3	6	1.000	0.872
##	14	14	4	6	1.000	0.796

# Построение монотонной регрессии

Вычисляем последовательно средние значения так, чтобы каждое новое было не меньше предыдущего.

В R это делается при помощи функции `isoreg()`



## Вычисление величины стресса

$$Stress = \sqrt{\frac{\sum (Z_i - Zh_i)^2}{\sum Z_i^2}}$$

Для данной, случайной, ординации точек Stress = 0.077

# Алгоритм MDS

1. Вычисляем матрицу коэффициентов различия между объектами.
2. Распределяем объекты в случайном порядке на плоскости [NB! от первичного распределения точек может зависеть и результат, поэтому на практике используются процедуры оптимального начального размещения точек (PCA)].
3. Вычисляем значение стресса.
4. Двигаем точки по плоскости так, чтобы найти минимальное значение стресса.
5. Повторяем процедуру 2-4 несколько раз для того чтобы избежать локальных минимумов стресса.
6. Обычно финальную ординацию поворачивают вдоль оси X так, чтобы ось отражала максимальное варьирование.



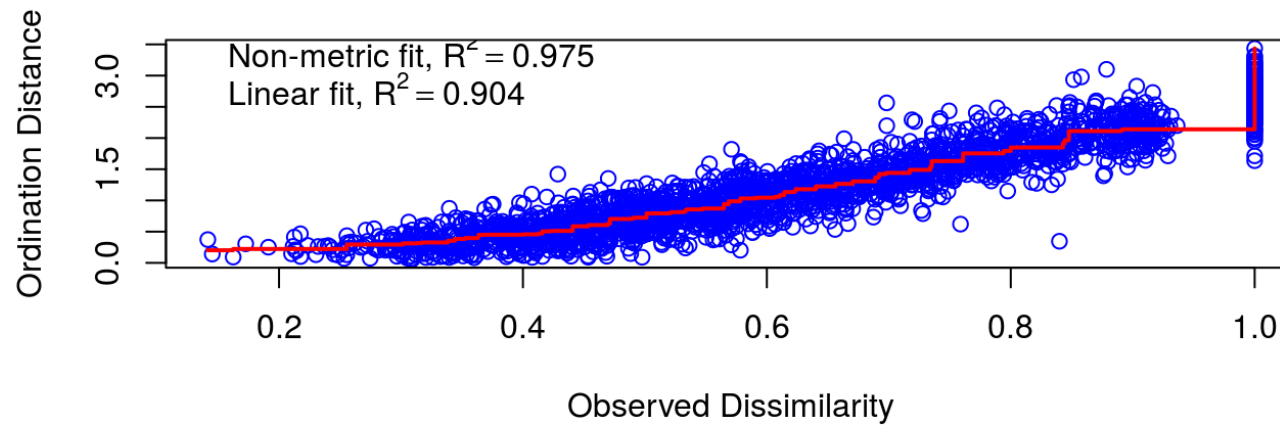
## Задание:

Постройте диаграмму Шепарда вместе с монотонной регрессией на полном материале по Долгой губе. Найдите величину стресса.

Hint: Используйте функцию `stressplot()`

# Решение

stressplot(ord)



Stress = 0.159

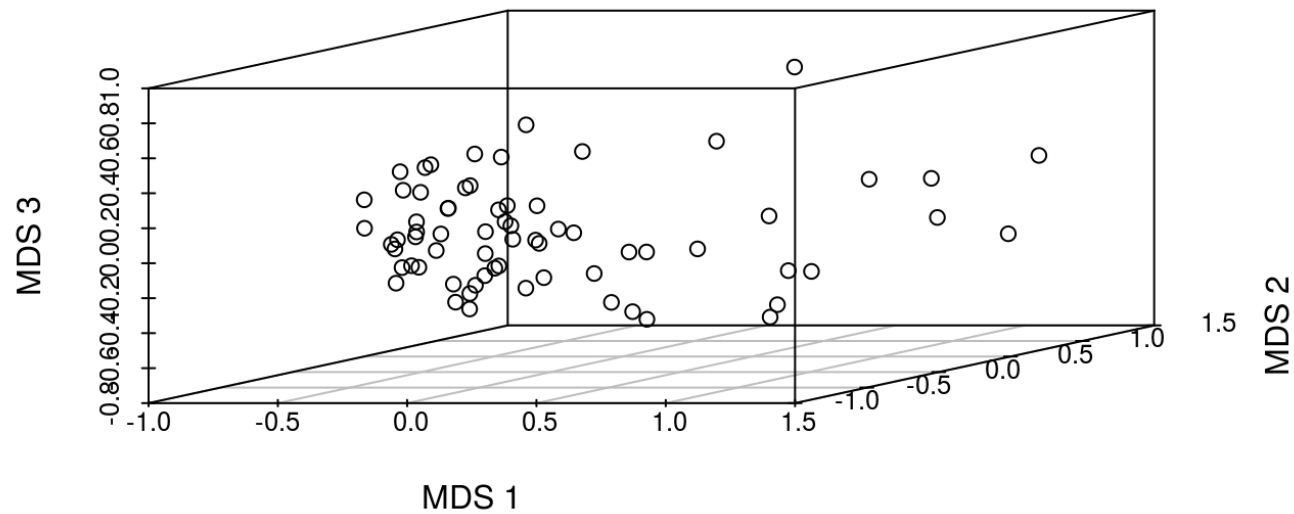
Надежна ли такая ординация?

## Оценка качества подгонки

- $\text{Stress} < 0.05$  Ординация отличная. Можно в деталях проанализировать взаиморасположение точек
- $0.05 < \text{Stress} < 0.1$  Хорошая конфигурация.
- $0.1 < \text{Stress} < 0.2$  Приемлемая конфигурация. Надо быть осторожными в интерпретации.
- $\text{Stress} > 0.2$  Возможно, надо проводить ординацию в большем количестве осей. Двумерное изображение неадекватно отражает взаиморасположение точек.



# MDS в трехмерном пространстве



Stress = 0.127

At! Ни в коем случае не используете координаты 3D MDS для двумерной ординации.

# Сравнение ординаций

## Задание:

- Постройте ординацию всех станций с использованием Евклидова расстояния

# Решение

```
ord4 <- metaMDS(log_abund, distance = "euclidean")
```

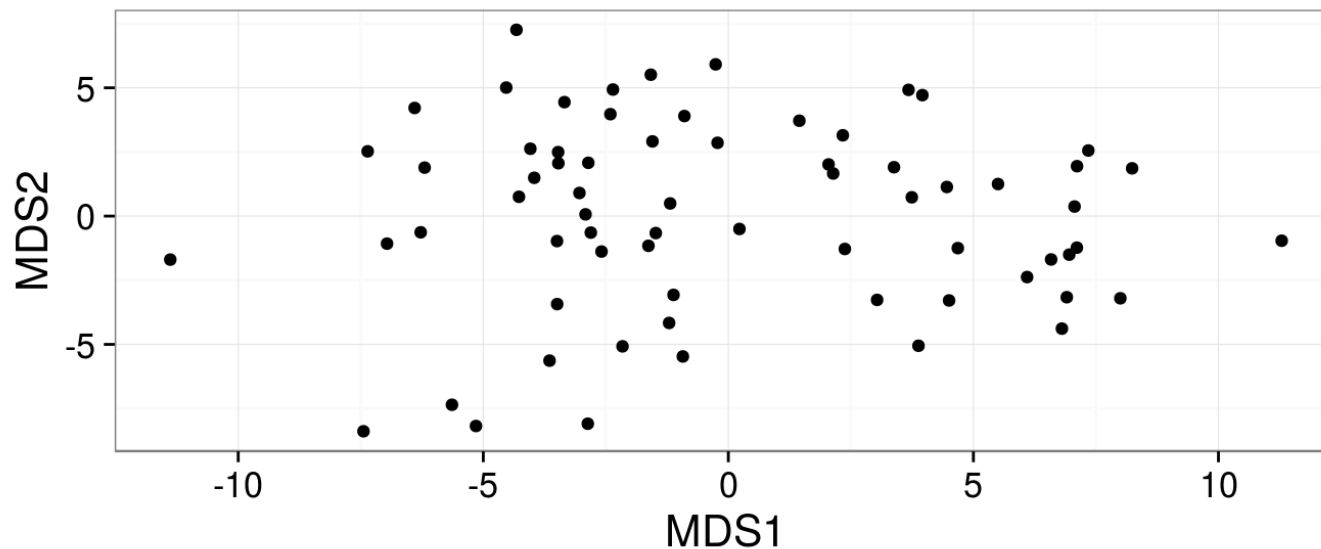
```
## Run 0 stress 0.179
```

```
## Run 1 stress 0.179
```

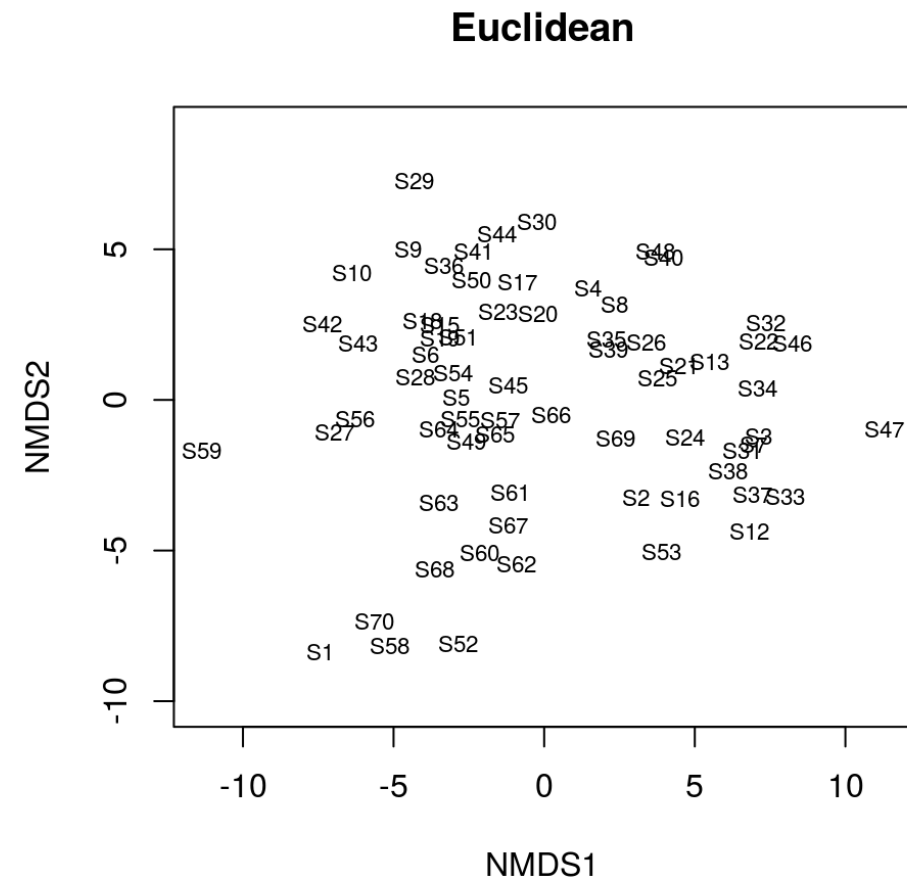
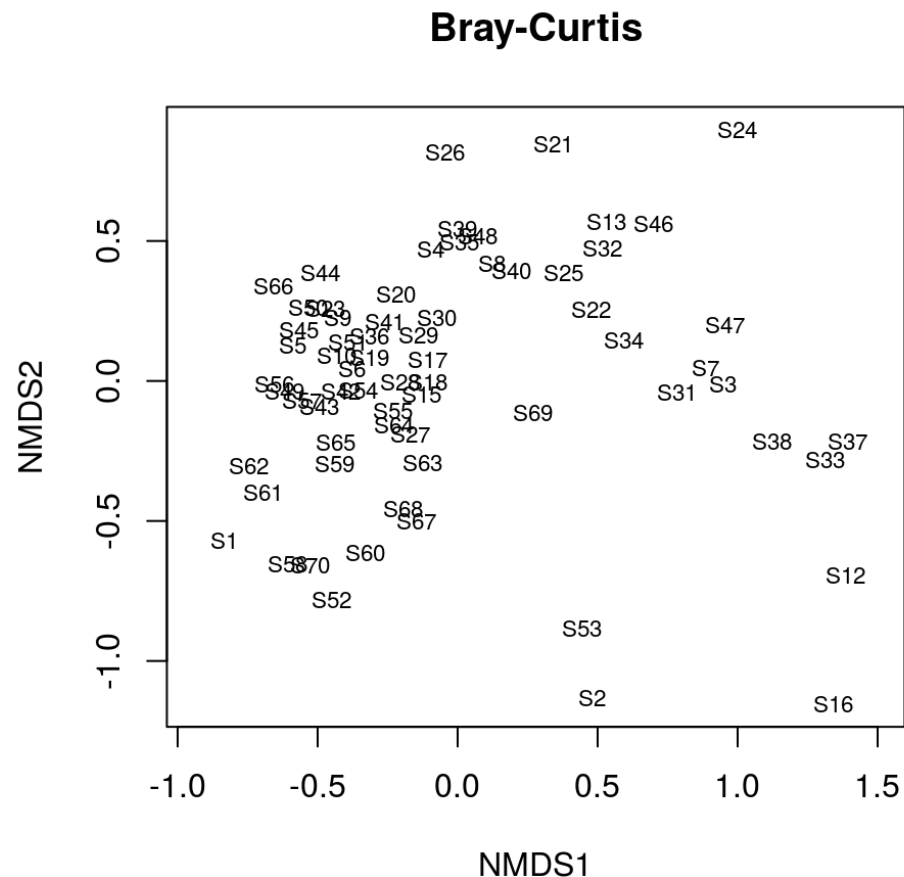
```
## ... procrustes: rmse 0.00143  max resid 0.00676
```

```
## *** Solution reached
```

```
ggplot(data.frame(ord4$points), aes(x = MDS1, y = MDS2)) + geom_point()
```

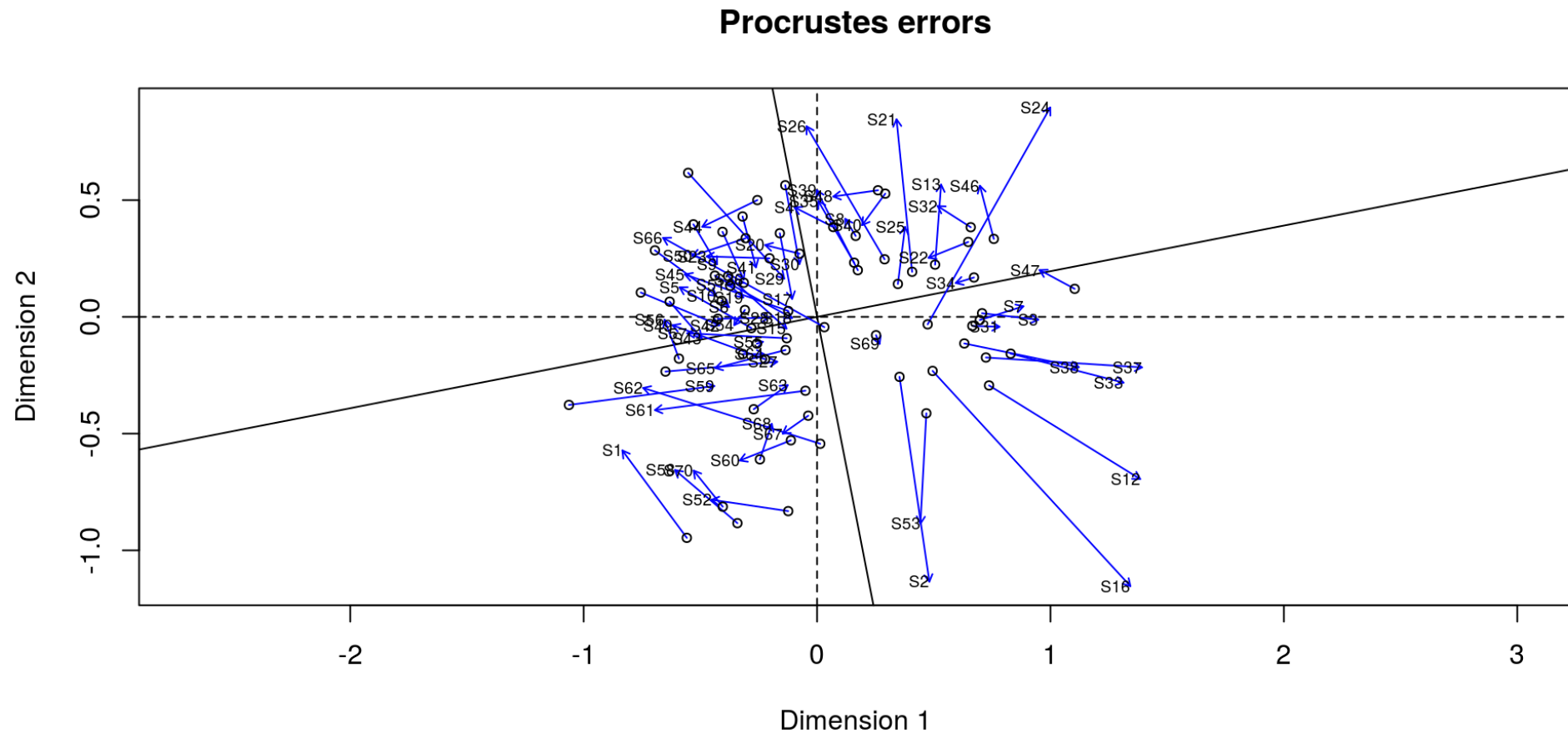


# Насколько похожи ординации, полученные при использовании коэффициента Брея-Куртиса и Евклидова расстояния?



# Прокурстово преобразование

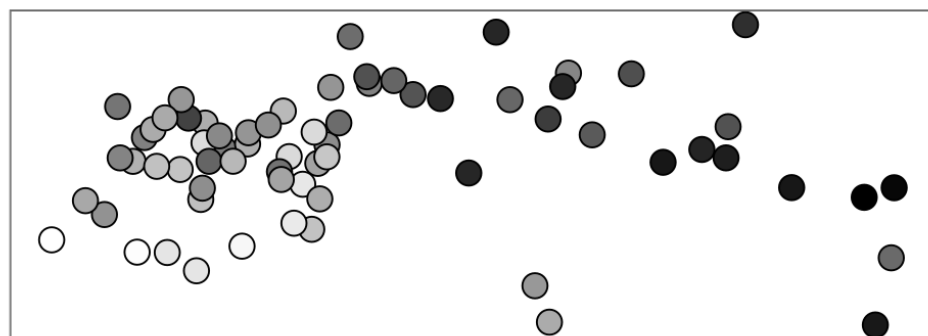
```
procrust <- procrustes(ord, ord4)  
plot(procrust)  
text(procrust, adj = 1, cex = 0.6)
```



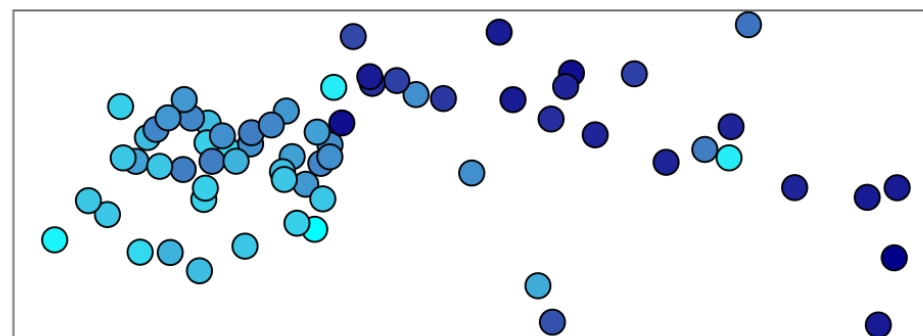
**Трактовка результатов ординации**

# Трактовка результатов ординации

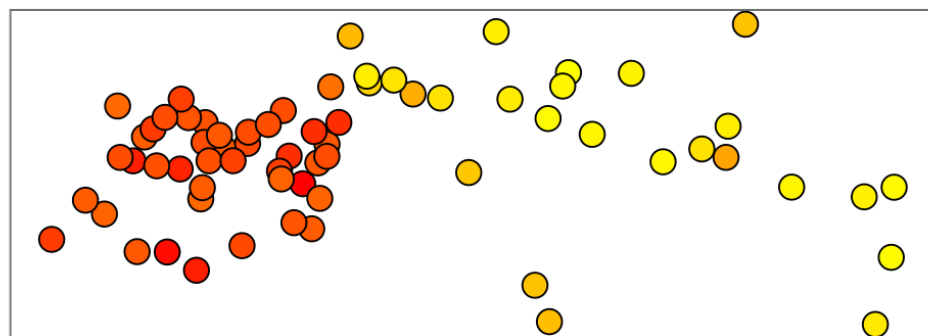
О более изощренных способах — на следующей лекции



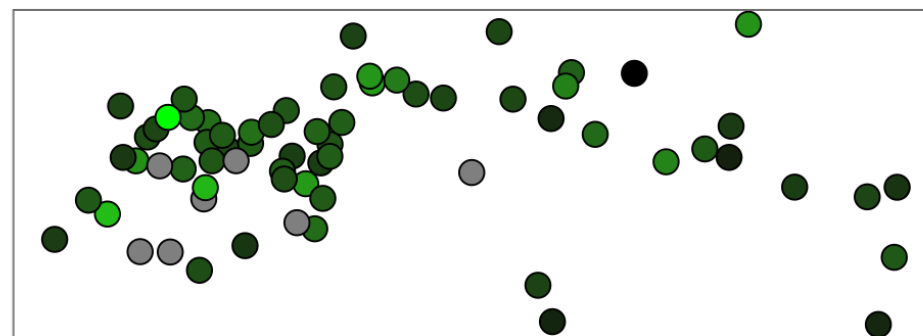
Depth  
5 10 15



Salinity  
26.0 26.5 27.0 28.0



Temperature  
0 5 10



Water content  
25 50 75



## Take-home messages

- Существует много методов не прямой ординации.
- nMDS - один из методов ординации в пространстве сниженной размерности. При ординации этим методом сохраняются ранги расстояний между объектами.
- Оценка качества подгонки ординации производится с помощью величины стресса.
- Значения координат не имеют особенного смысла. Имеет значение только взаиморасположение точек.
- Результат nMDS зависит от выбора меры различия

# Литература

- 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.
- Zuur, A. F., Ieno, E. N., Smith, G. M. Analysing Ecological Data. Springer 2007
- Миркин Б.М., Розенберг Г.С., Наумова Л.Г. Словарь понятий и терминов современной фитоцентологии. М. Наука, 1989.