

Фонд
интеллект



Теплокарты и другие рисунки

*Анастасия Жарикова
Лекция 11 - 2022*

Теплокарты

Теплокарты в R

ggplot() + [geom_tile\(\)](#)

[pheatmap](#)

[ComplexHeatmap](#)

ComplexHeatmap и pheatmap

Работают с матрицами

Вспомним данные про тыквы

```
pump <- read_delim('pumpkins.tab')  
glimpse(pump)
```

```
Rows: 54  
Columns: 5  
$ year    <dbl> 2013, 2013, 2013, 2013, 2013, 2013, 2014, 2014, 2014, 2014, 20...  
$ type    <chr> "F", "L", "P", "S", "T", "W", "F", "L", "P", "S", "T", "W", "F...  
$ weight  <dbl> 154.50, 139.25, 2032.00, 1264.00, 6.83, 350.50, 211.00, 138.50...  
$ country <chr> "United States", "Canada", "United States", "United States", "...  
$ name    <chr> "pumpkin", "not_pumpkin", "pumpkin", "pumpkin", "not_pumpkin",...
```

```
head(pump)
```

```
# A tibble: 6 × 5  
  year type  weight country      name  
  <dbl> <chr>  <dbl> <chr>      <chr>  
1  2013 F      154.  United States pumpkin  
2  2013 L      139.  Canada      not_pumpkin  
3  2013 P     2032  United States pumpkin  
4  2013 S     1264  United States pumpkin  
5  2013 T        6.83 France      not_pumpkin  
6  2013 W     350.  United States not_pumpkin
```

Сделаем матрицу

```
pump %>%  
  dplyr::select(year, type, weight) %>%  
  mutate(year = as.character(year)) %>%  
  pivot_wider(names_from = type, values_from = weight) -> pump.w
```

```
pump.w
```

```
# A tibble: 9 × 7  
  year      F      L      P      S      T      W  
  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 2013   154.  139. 2032  1264   6.83  350.  
2 2014   211   138. 2324.  1578   8.41  298.  
3 2015   173   150. 2230.  1534.   7.27  302  
4 2016   189   142. 2625.  1844.   8.61  318.  
5 2017   182.  144. 2363  2118   8.22  332  
6 2018   174   149. 2528  1534   9.44  328.  
7 2019   205   156. 2517  1936.   9.65  342.  
8 2020   186   146. 2594.  1838  10.8   300  
9 2021   204.  155. 2703.  2164   9.09  329
```

Это еще не матрица

Вот теперь сделаем матрицу

```
pump.w %>%  
  select(where(is.numeric)) %>% as.matrix() -> pump.m  
pump.m[1:3,]
```

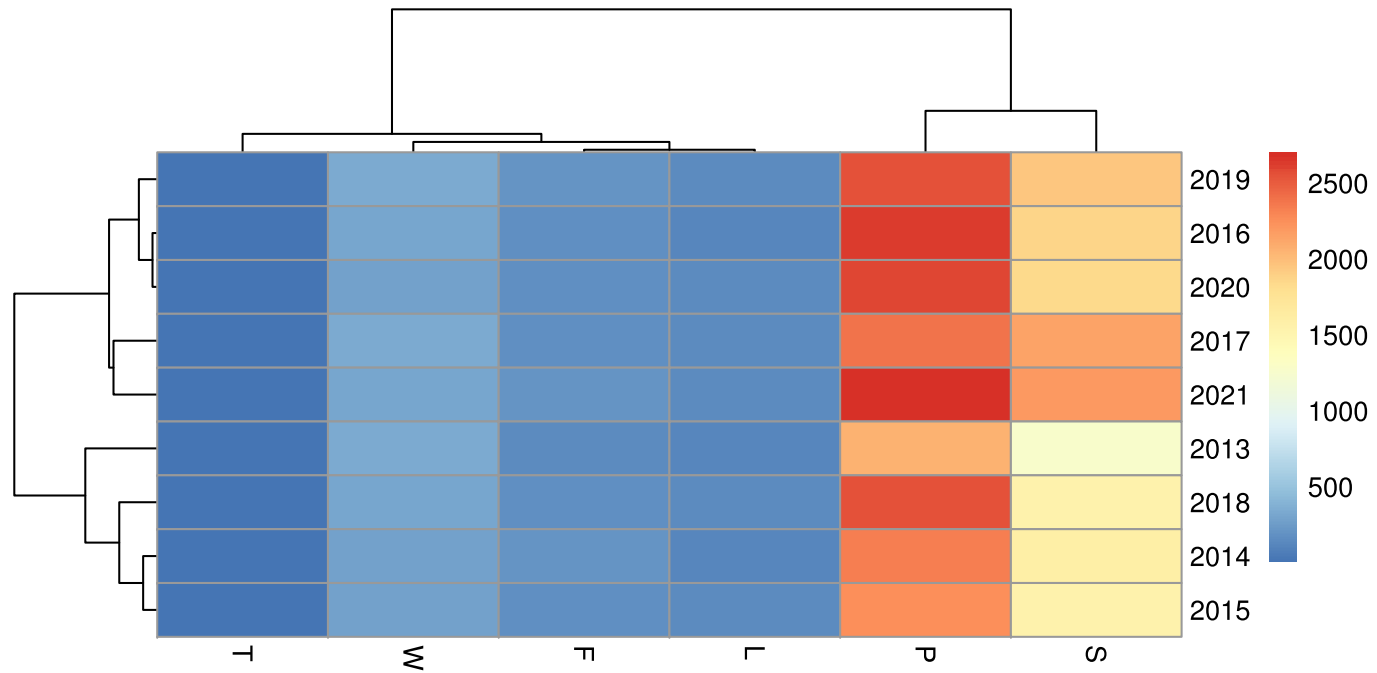
```
      F      L      P      S      T      W  
[1,] 154.5 139.25 2032.0 1264.0 6.83 350.5  
[2,] 211.0 138.50 2323.7 1578.0 8.41 297.6  
[3,] 173.0 149.50 2230.5 1533.5 7.27 302.0
```

```
rownames(pump.m) <- pump.w$year  
pump.m[1:3,]
```

```
      F      L      P      S      T      W  
2013 154.5 139.25 2032.0 1264.0 6.83 350.5  
2014 211.0 138.50 2323.7 1578.0 8.41 297.6  
2015 173.0 149.50 2230.5 1533.5 7.27 302.0
```

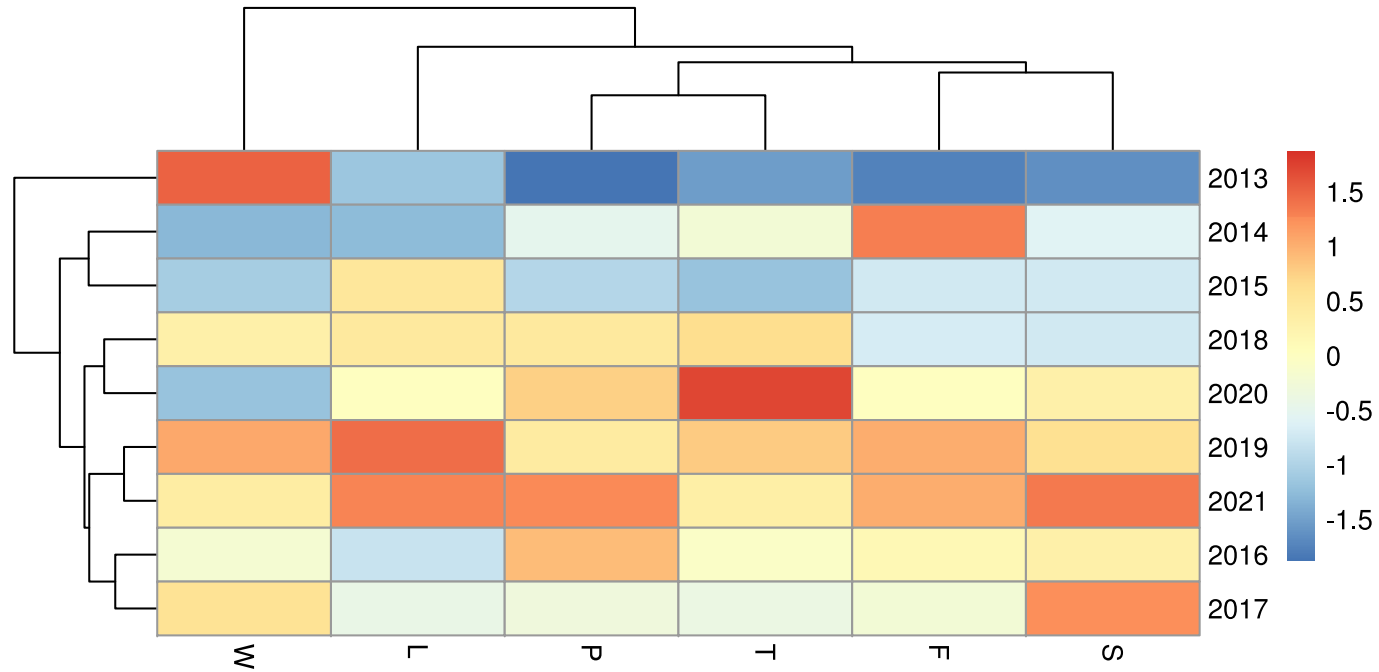

Теплокарта

```
library(pheatmap)  
pheatmap(pump.m)
```



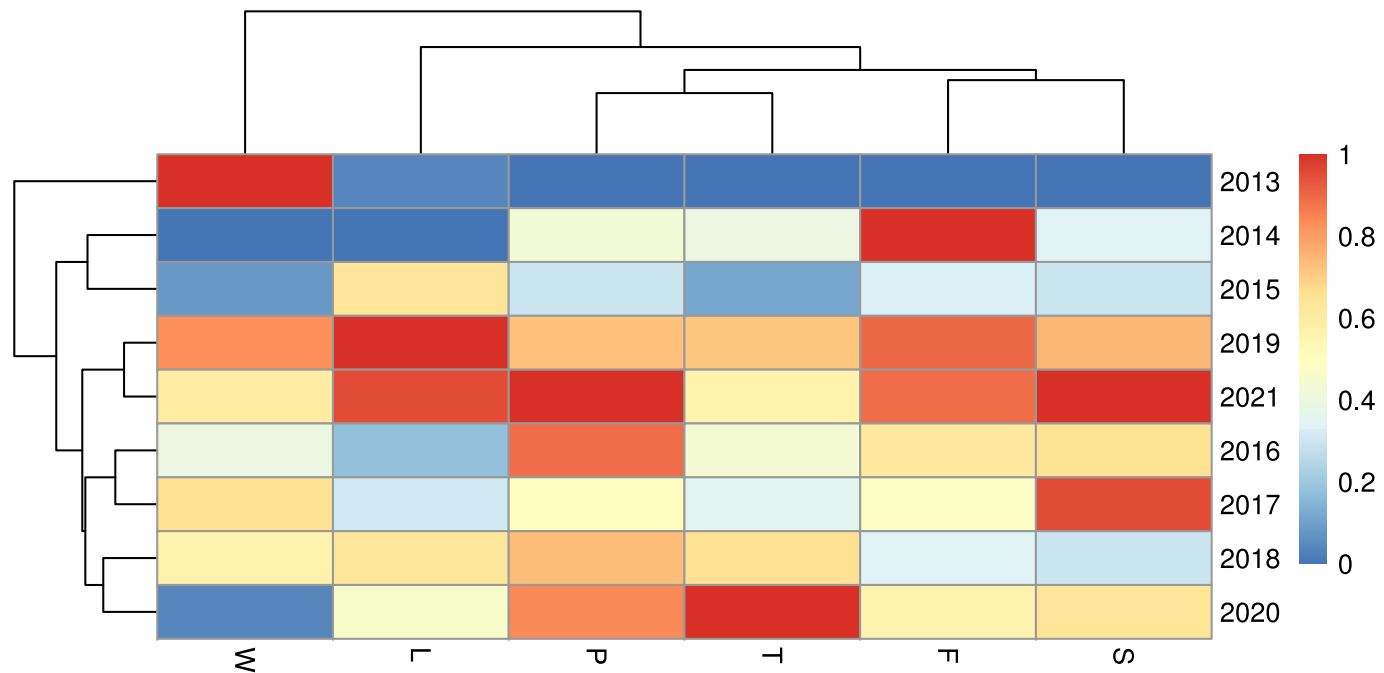
Стандартизация данных

```
pheatmap(pump.m, scale = "column")
```



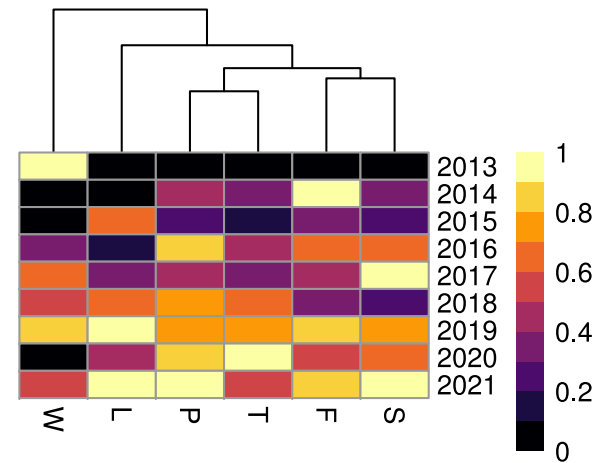
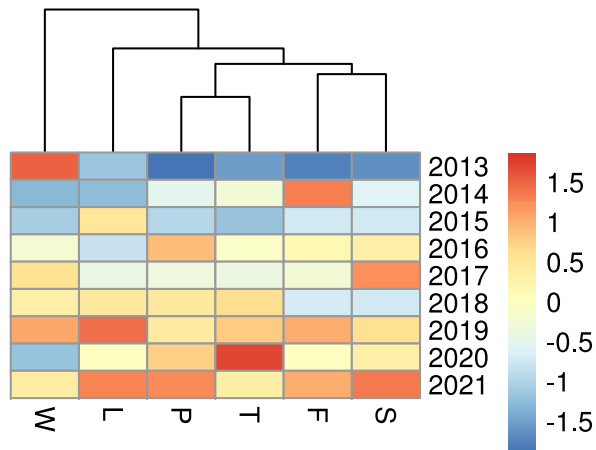
Нормализация данных

```
library(BBmisc)  
pheatmap(normalize(pump.m, method = "range", range = c(0, 1), margin = 2))
```



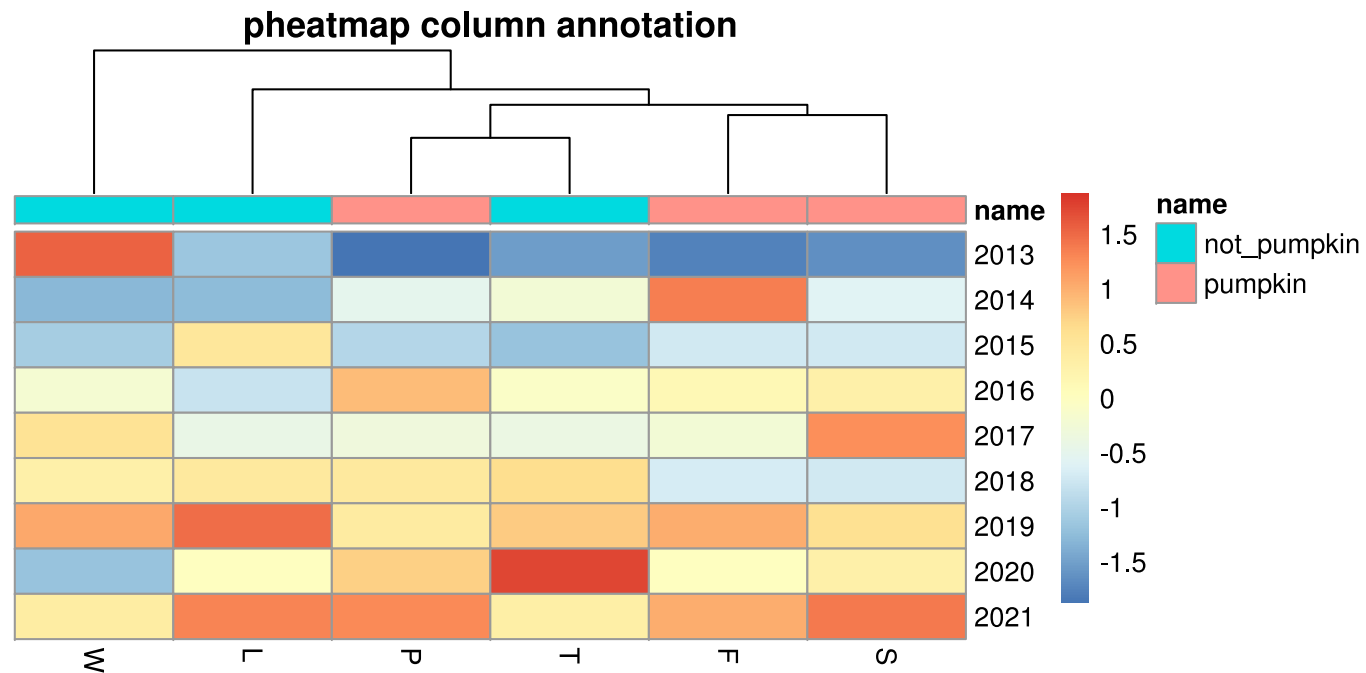
Цвета

```
library(viridis)
library(gridExtra)
pheatmap(pump.m, scale = "column", silent = T, cellwidth=25, cellheight = 10, cluster_rows = F) -> HM1
pheatmap(normalize(pump.m, method = "range", range = c(0, 1), margin = 2), color = inferno(10), silent =
  T, cellwidth=25, cellheight = 10, cluster_rows = F) -> HM2
grid.arrange(grobs = list(HM1[[4]], HM2[[4]]), ncol=2)
```



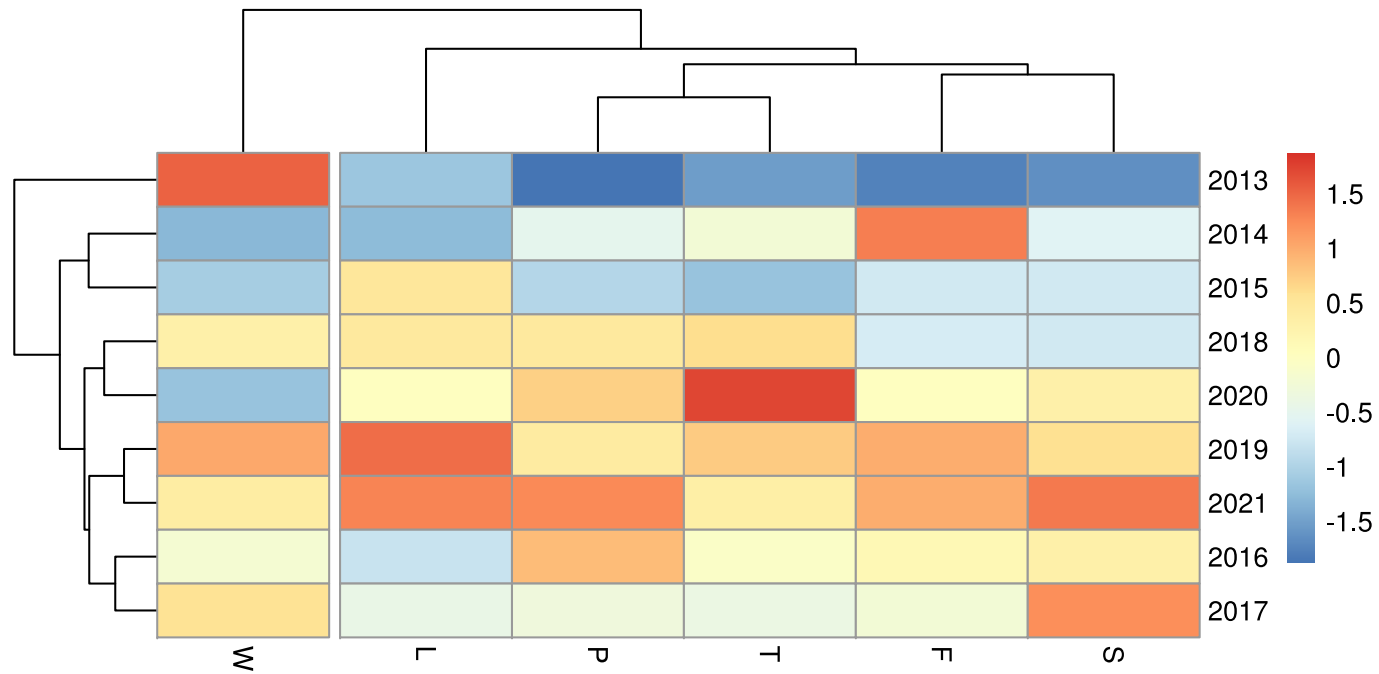
Аннотация

```
annot = pump %>% select(type, name) %>% distinct() %>% as.data.frame()
rownames(annot) <- annot$type
annot$type = NULL
pheatmap(pump.m, scale = "column", cluster_rows = F, annotation_col = annot, main = "pheatmap column
annotation")
```



Разделить на кластеры

```
pheatmap(pump.m, scale = "column", cutree_cols = 2)
```



Матрица из слов

```
pump %>%  
  dplyr::select(year, type, country) %>%  
  pivot_wider(names_from = type, values_from = country, ) -> pump.w  
  
pump.w
```

```
# A tibble: 9 × 7  
  year F          L          P          S          T          W  
  <dbl> <chr>      <chr>      <chr>      <chr>      <chr> <chr>  
1  2013 United States Canada      United States United States France Unite...  
2  2014 Canada      United States Switzerland United States Unite... Italy  
3  2015 United States Canada      United States Canada      Switz... Unite...  
4  2016 Canada      Canada      Belgium    United States Unite... Unite...  
5  2017 Germany    United States United States United States Unite... Unite...  
6  2018 Canada      United States United States United States Unite... Unite...  
7  2019 The Netherlands Slovenia    United States United States Unite... Unite...  
8  2020 United States United States United Kingdom United States Unite... Unite...  
9  2021 United States Canada      Italy      United States Unite... Unite...
```

Матрица из слов

```
pump.w %>%  
  select(where(is.character)) %>% as.matrix() -> pump.m  
pump.m[1:3,1:3]
```

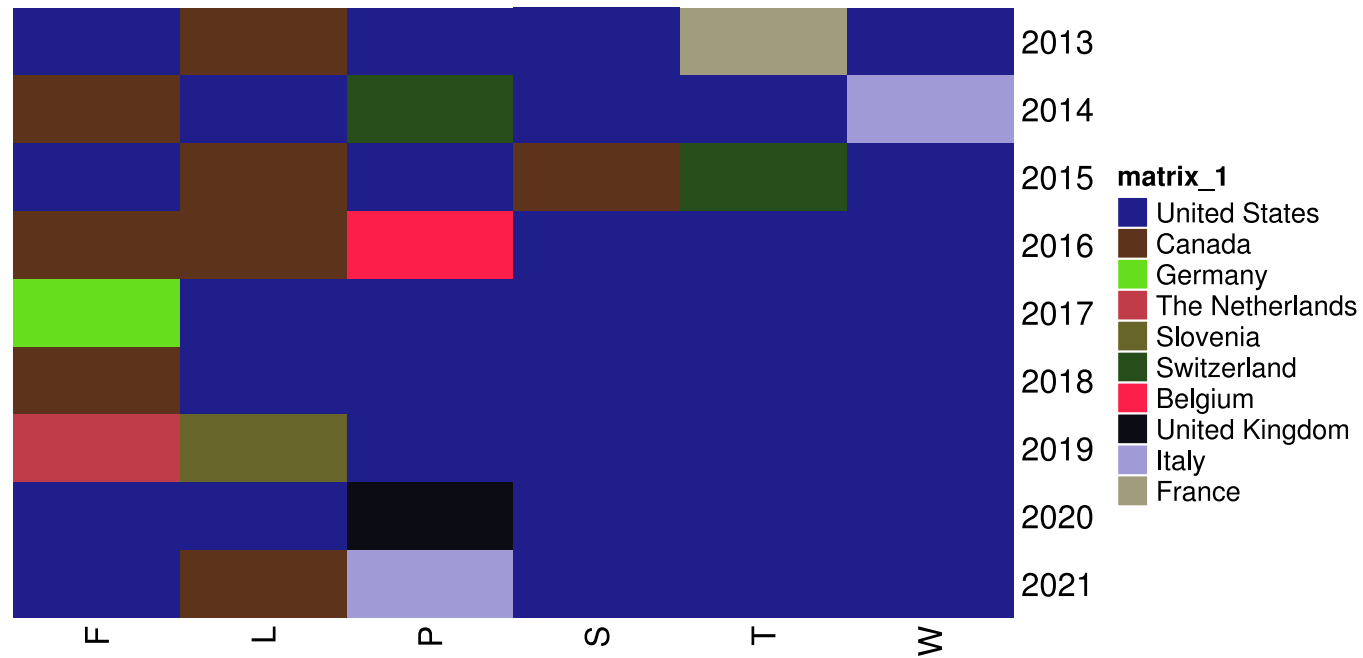
```
      F          L          P  
[1,] "United States" "Canada" "United States"  
[2,] "Canada"        "United States" "Switzerland"  
[3,] "United States" "Canada"   "United States"
```

```
rownames(pump.m) <- pump.w$year  
pump.m[1:3,1:3]
```

```
      F          L          P  
2013 "United States" "Canada" "United States"  
2014 "Canada"        "United States" "Switzerland"  
2015 "United States" "Canada"   "United States"
```


ComplexHeatmap

```
library(ComplexHeatmap)  
Heatmap(pump.m)
```



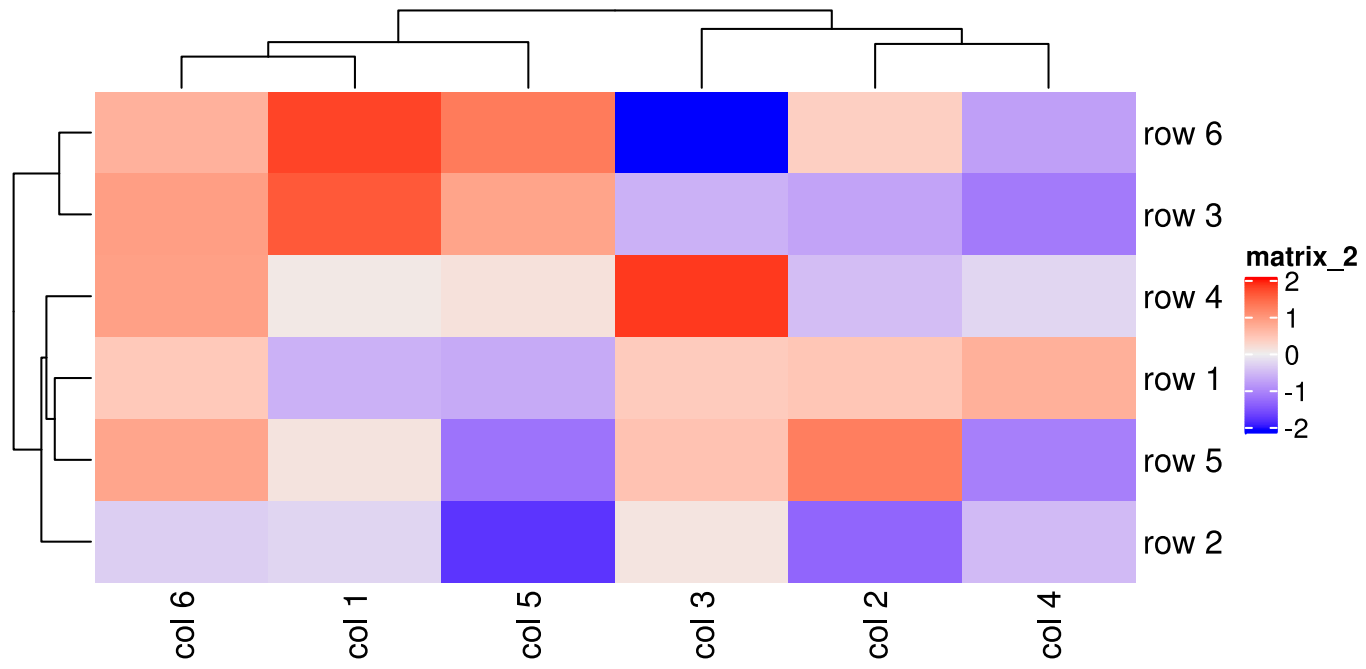
Случайная матрица

```
set.seed(123)

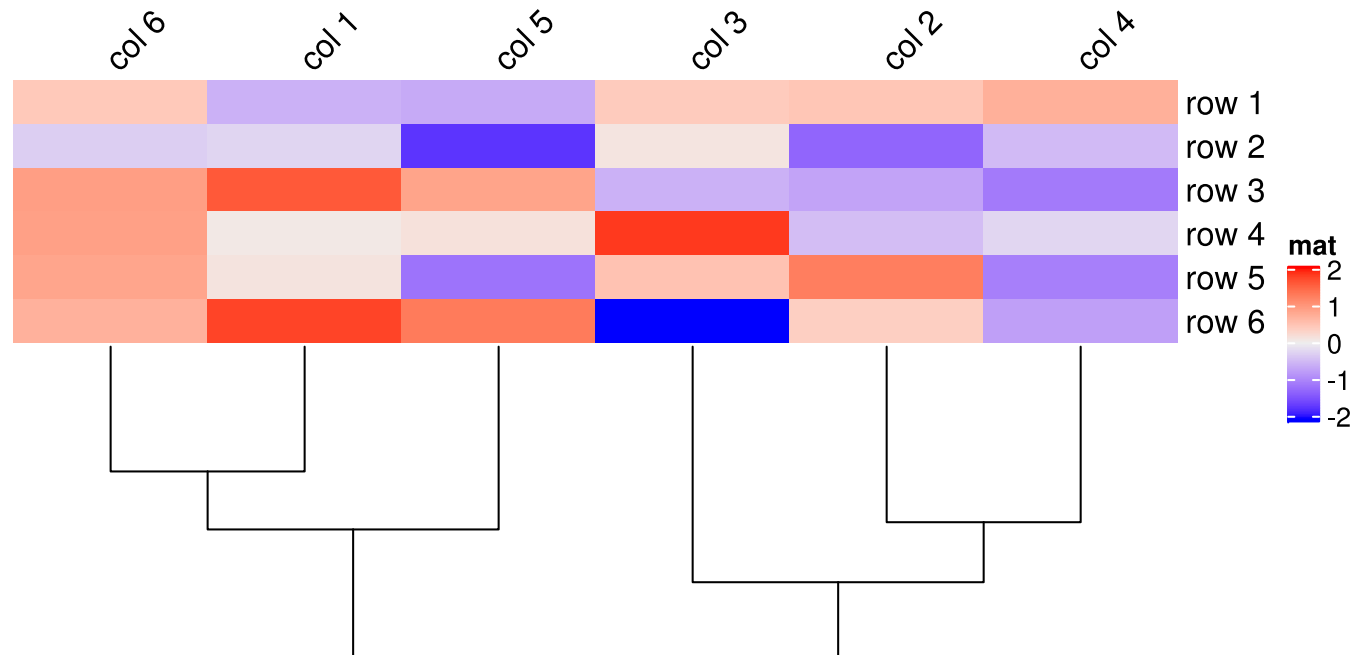
M1<-matrix(rnorm(36),nrow=6)
colnames(M1) <- paste('col', 1:ncol(M1))
rownames(M1) <- paste('row', 1:nrow(M1))
M1
```

	col 1	col 2	col 3	col 4	col 5	col 6
row 1	-0.56047565	0.4609162	0.4007715	0.7013559	-0.6250393	0.4264642
row 2	-0.23017749	-1.2650612	0.1106827	-0.4727914	-1.6866933	-0.2950715
row 3	1.55870831	-0.6868529	-0.5558411	-1.0678237	0.8377870	0.8951257
row 4	0.07050839	-0.4456620	1.7869131	-0.2179749	0.1533731	0.8781335
row 5	0.12928774	1.2240818	0.4978505	-1.0260044	-1.1381369	0.8215811
row 6	1.71506499	0.3598138	-1.9666172	-0.7288912	1.2538149	0.6886403

Heatmap(M1)

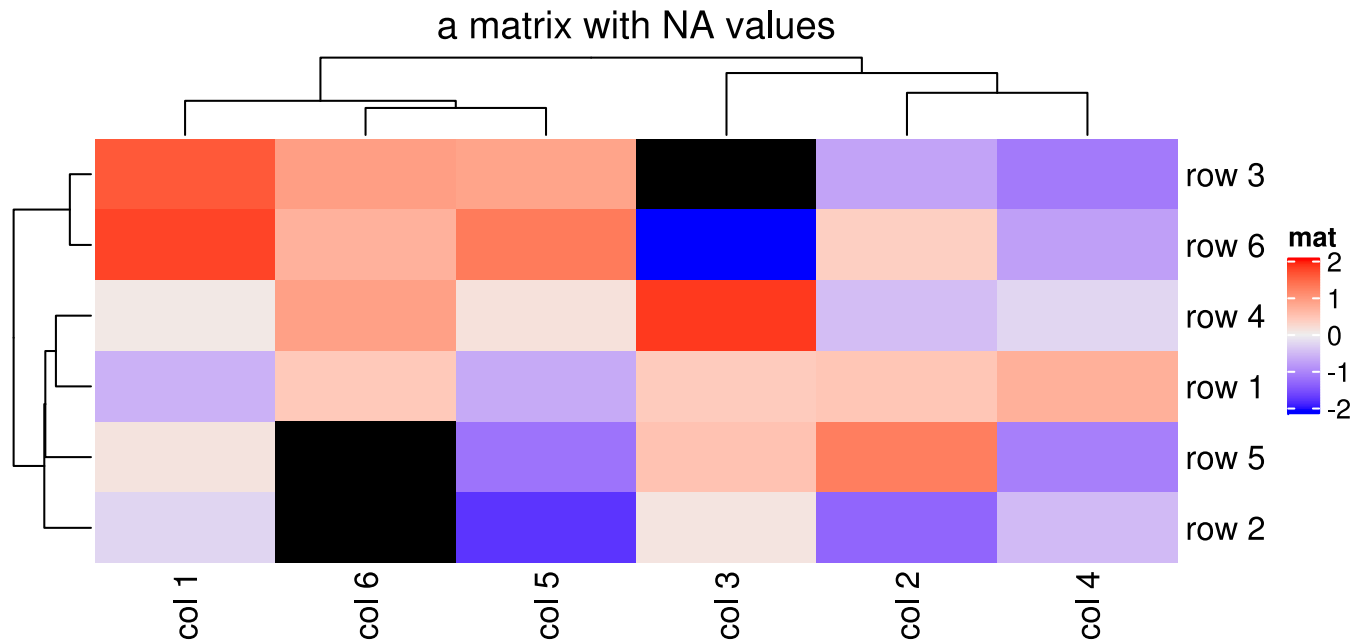


```
Heatmap(M1, name = "mat", cluster_rows = FALSE,  
column_dend_side = "bottom", column_dend_height = unit(4, "cm"),  
column_names_rot = 45, column_names_side = "top")
```



Пропущенные значения

```
na_index = sample(c(TRUE, FALSE), nrow(M1)*ncol(M1), replace = TRUE, prob = c(1, 9))  
M1[na_index] = NA  
Heatmap(M1, name = "mat", na_col = "black", column_title = "a matrix with NA values")
```



Много распределений

```
diamonds
```

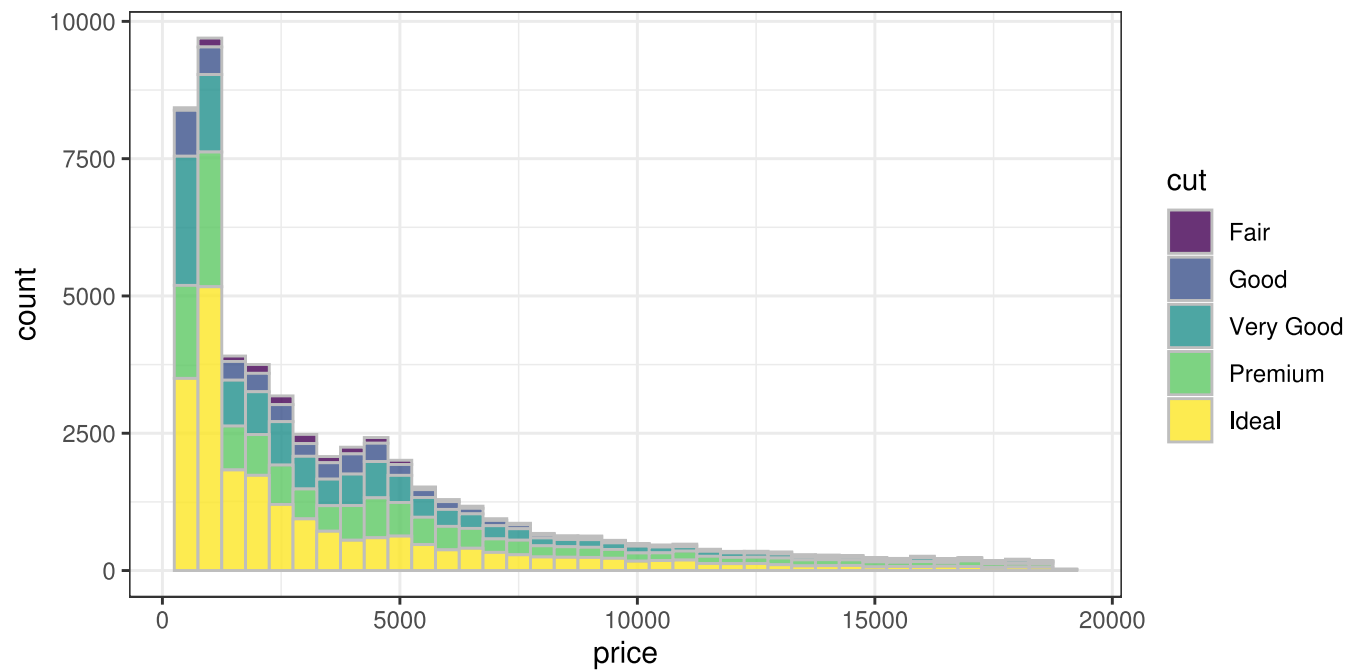
```
# A tibble: 53,940 × 10
  carat cut      color clarity depth table price      x      y      z
  <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
1  0.23 Ideal      E      SI2     61.5    55   326  3.95  3.98  2.43
2  0.21 Premium   E      SI1     59.8    61   326  3.89  3.84  2.31
3  0.23 Good      E      VS1     56.9    65   327  4.05  4.07  2.31
4  0.29 Premium   I      VS2     62.4    58   334  4.2   4.23  2.63
5  0.31 Good      J      SI2     63.3    58   335  4.34  4.35  2.75
6  0.24 Very Good J      VVS2     62.8    57   336  3.94  3.96  2.48
7  0.24 Very Good I      VVS1     62.3    57   336  3.95  3.98  2.47
8  0.26 Very Good H      SI1     61.9    55   337  4.07  4.11  2.53
9  0.22 Fair      E      VS2     65.1    61   337  3.87  3.78  2.49
10 0.23 Very Good H      VS1     59.4    61   338  4     4.05  2.39
# ... with 53,930 more rows
```

Хотим увидеть распределения цен по группам качества бриллиантов

Много распределений

Гистограмма

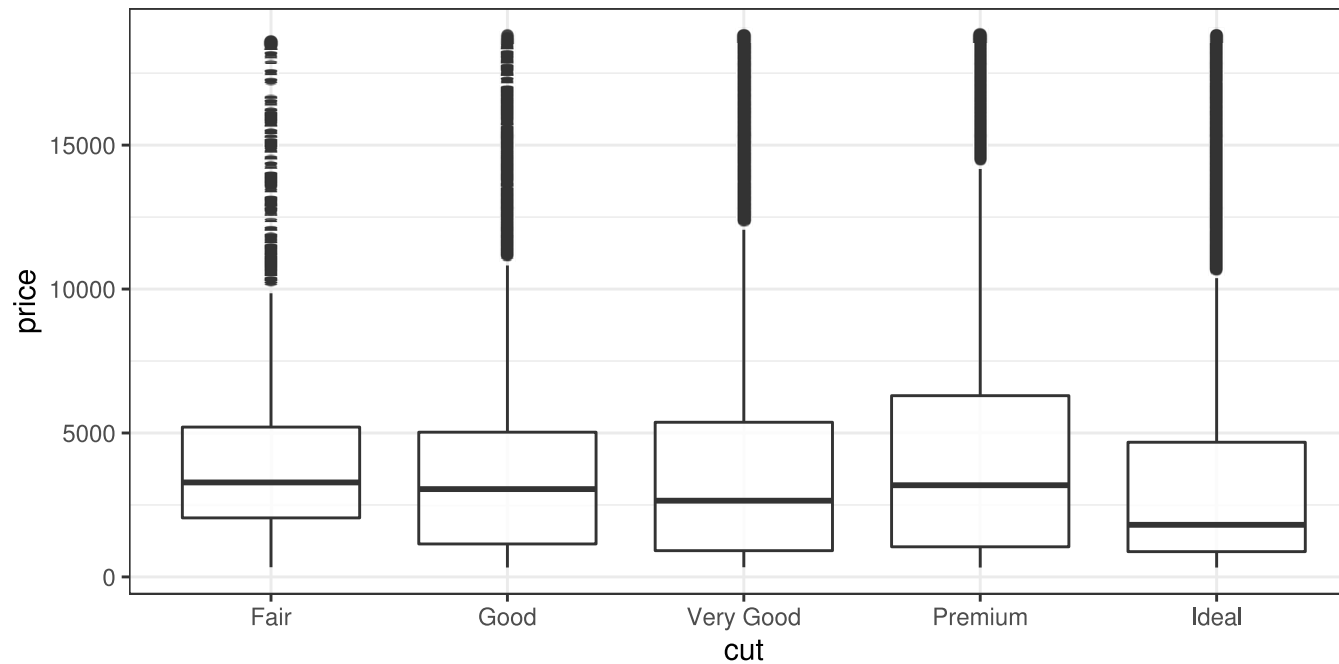
```
diamonds %>%  
  ggplot() + geom_histogram(aes(x = price, fill = cut), alpha = 0.8, color = 'grey', binwidth = 500) +  
  theme_bw()
```



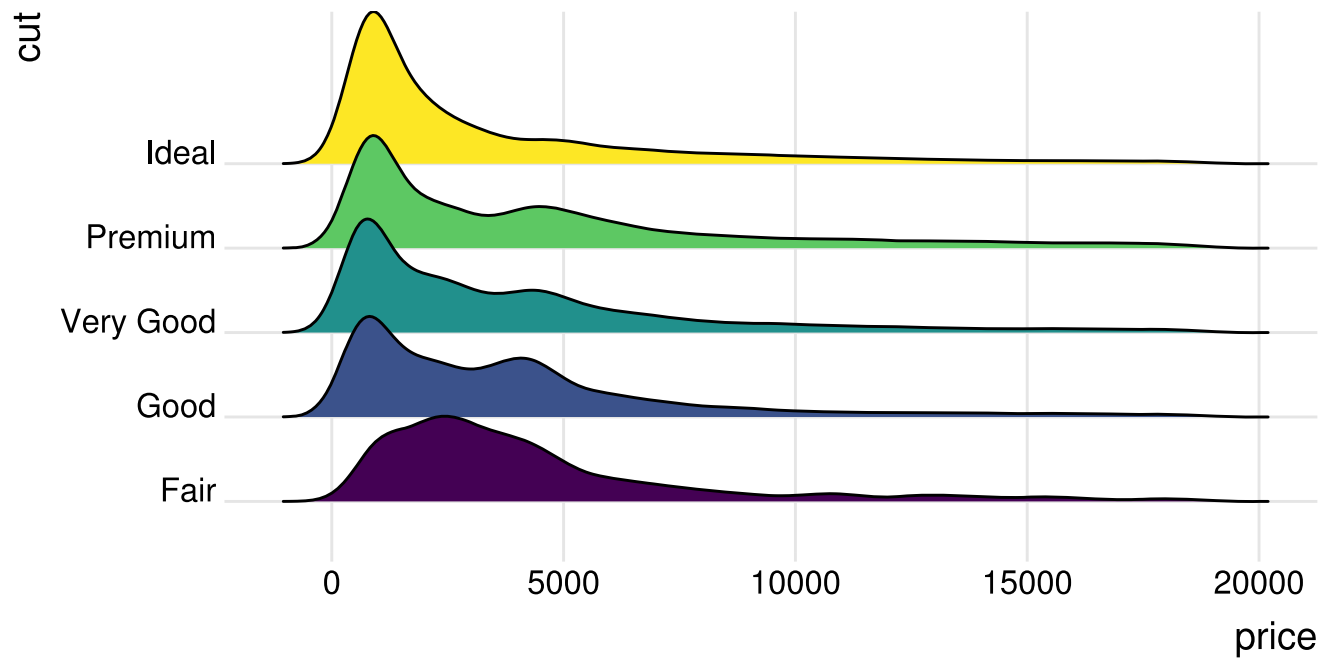
Много распределений

Боксплот

```
diamonds %>%  
  ggplot() + geom_boxplot(aes(x = cut, y = price), alpha = 0.8) +  
  theme_bw()
```




```
library(ggribes)
ggplot(diamonds, aes(x = price, y = cut, fill = cut)) +
  geom_density_ridges() +
  theme_ridges() +
  theme(legend.position = "none")
```



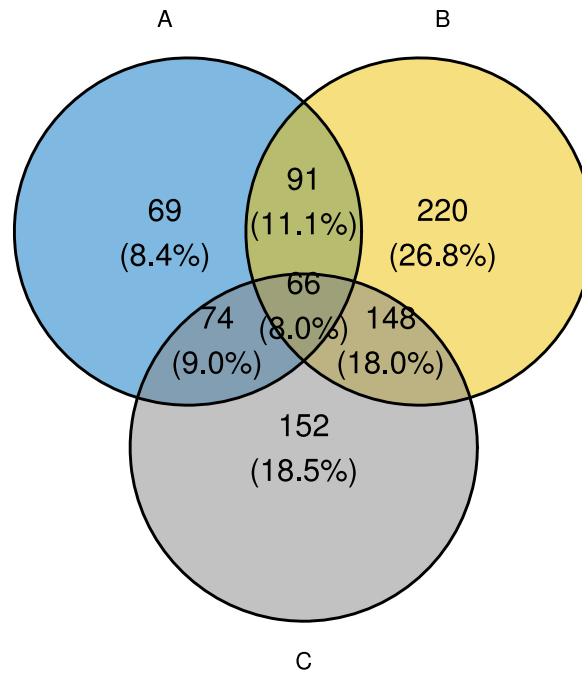
Если распределений очень много, то стоит вовремя остановиться

```
set.seed(123)
ex <- paste("sample",1:1000,sep="")
x <- list(
  A = sample(ex,300),
  B = sample(ex,525),
  C = sample(ex,440))
str(x)
```

List of 3

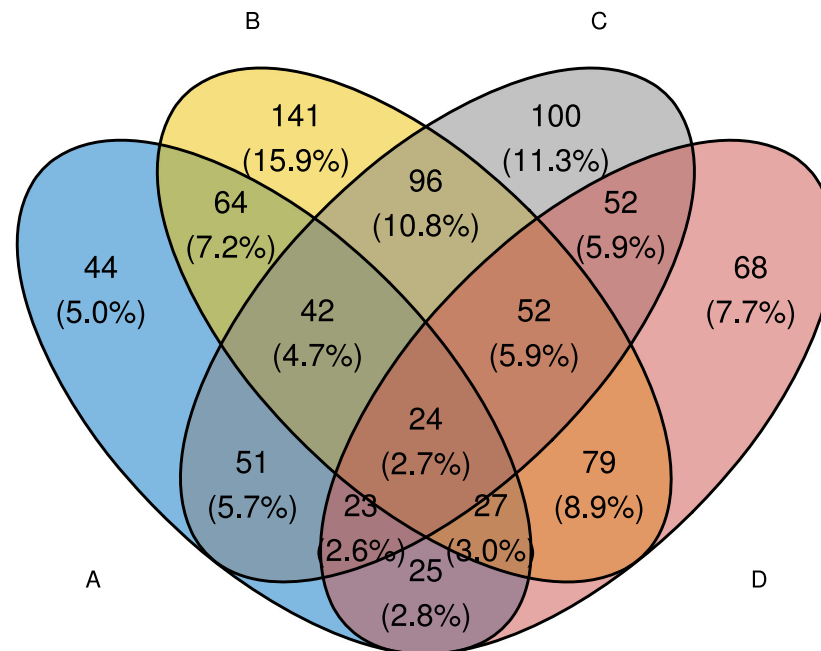
```
$ A: chr [1:300] "sample415" "sample463" "sample179" "sample526" ...
$ B: chr [1:525] "sample165" "sample703" "sample588" "sample377" ...
$ C: chr [1:440] "sample655" "sample746" "sample615" "sample58" ...
```

```
library(ggvenn)
ggvenn(
  x,
  fill_color = c("#0073C2FF", "#EFC000FF", "#868686FF", "#CD534CFF"),
  stroke_size = 0.5, set_name_size = 3)
```

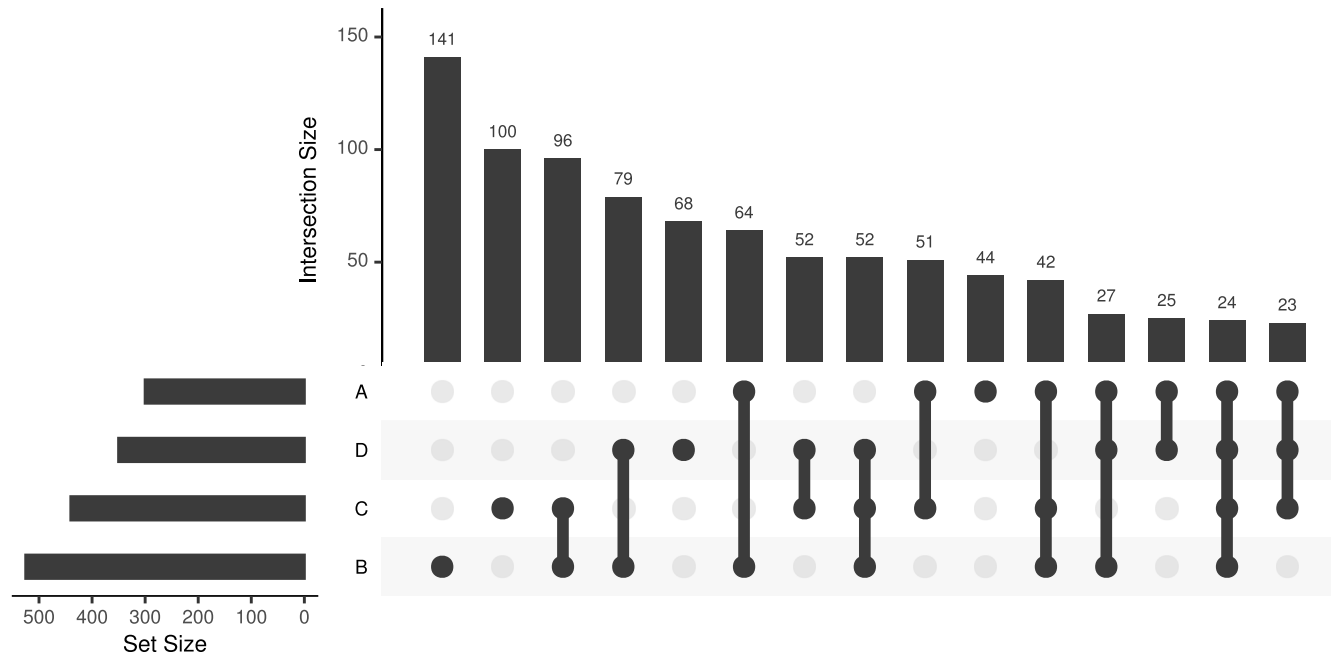


```
set.seed(123)
ex <- paste("sample",1:1000,sep="")
x <- list(
  A = sample(ex,300),
  B = sample(ex,525),
  C = sample(ex,440),
  D = sample(ex,350))
```

```
library(ggvenn)
ggvenn(
  x,
  fill_color = c("#0073C2FF", "#EFC000FF", "#868686FF", "#CD534CFF"),
  stroke_size = 0.5, set_name_size = 3)
```



```
library(UpSetR)
upset(fromList(x), order.by = "freq", point.size = 3.5, line.size = 2, mb.ratio = c(0.55, 0.45))
```



Много картинок

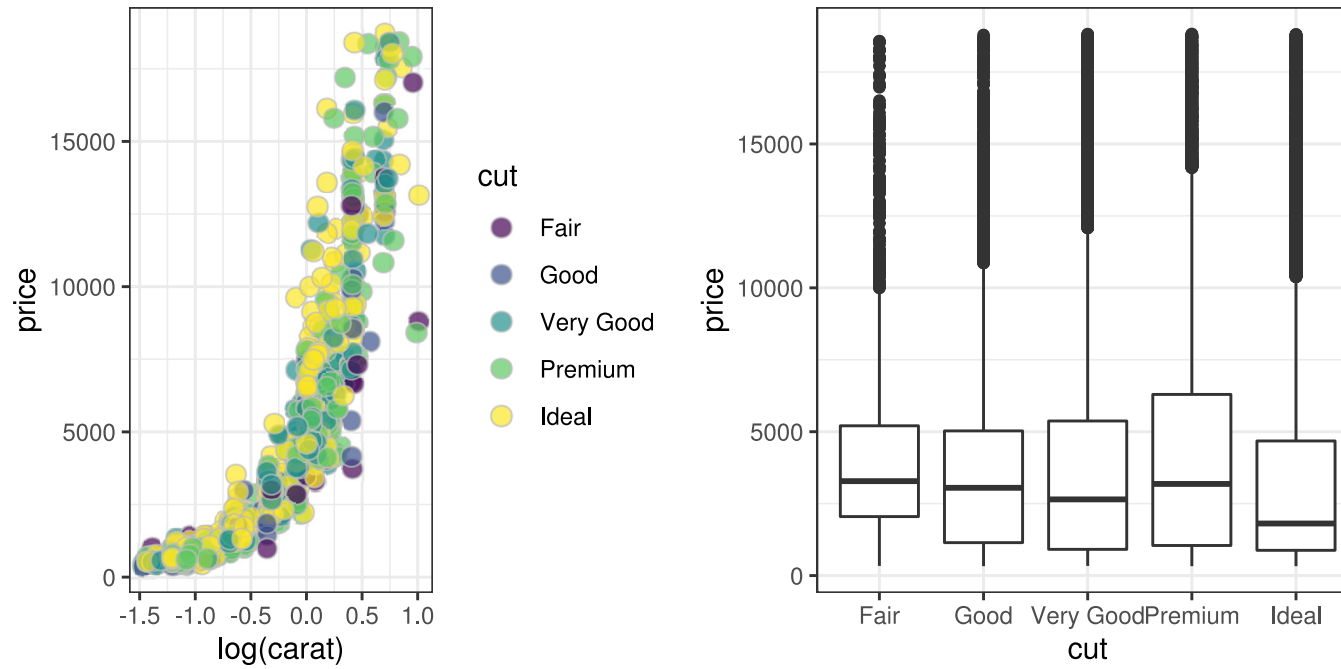
```
diamonds %>%
  sample_n(1000) %>%
  ggplot() + theme_bw() +
  geom_point(aes(x = log(carat), y = price, fill = cut), size = 3, shape = 21, alpha = 0.7, color =
  'grey') -> p1

diamonds %>%
  ggplot() + theme_bw() +
  geom_boxplot(aes(x = cut, y = price)) -> p2

ggplot(diamonds, aes(x = price, y = cut, fill = cut)) +
  geom_density_ridges() +
  theme_ridges() +
  theme(legend.position = "none") -> p3
```

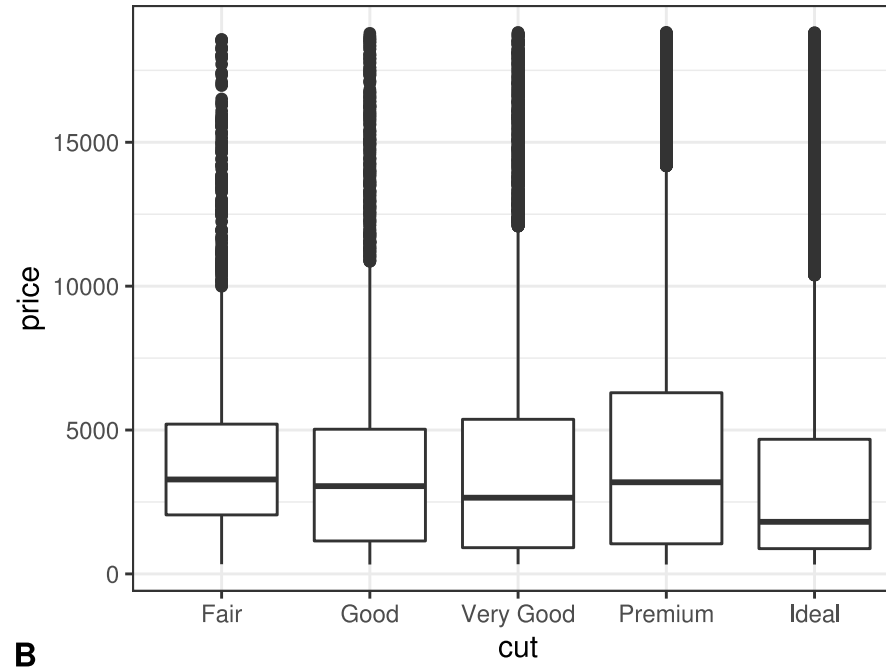
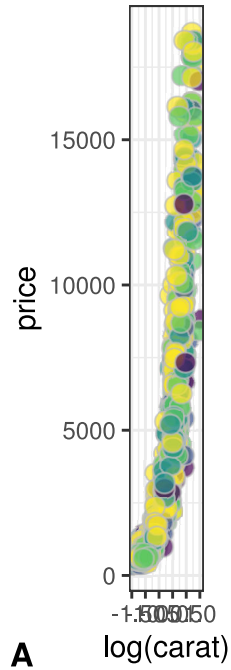
cowplot

```
library(cowplot)
plot_grid(p1, p2)
```




```
plot_grid(p1, p2,  
  rel_widths = c(1,2), labels = c('A', 'B'), label_size = 12,  
  label_x = 0, label_y = 0,  
  hjust = -0.5, vjust = -0.5) -> p12
```

p12



```
plot_grid(p12, p3, ncol=1,  
          rel_heights = c(1,1.5), labels = c('', 'C'), label_size = 12,  
          label_x = 0, label_y = 0, hjust = -0.5, vjust = -0.5)
```

