RUDN university

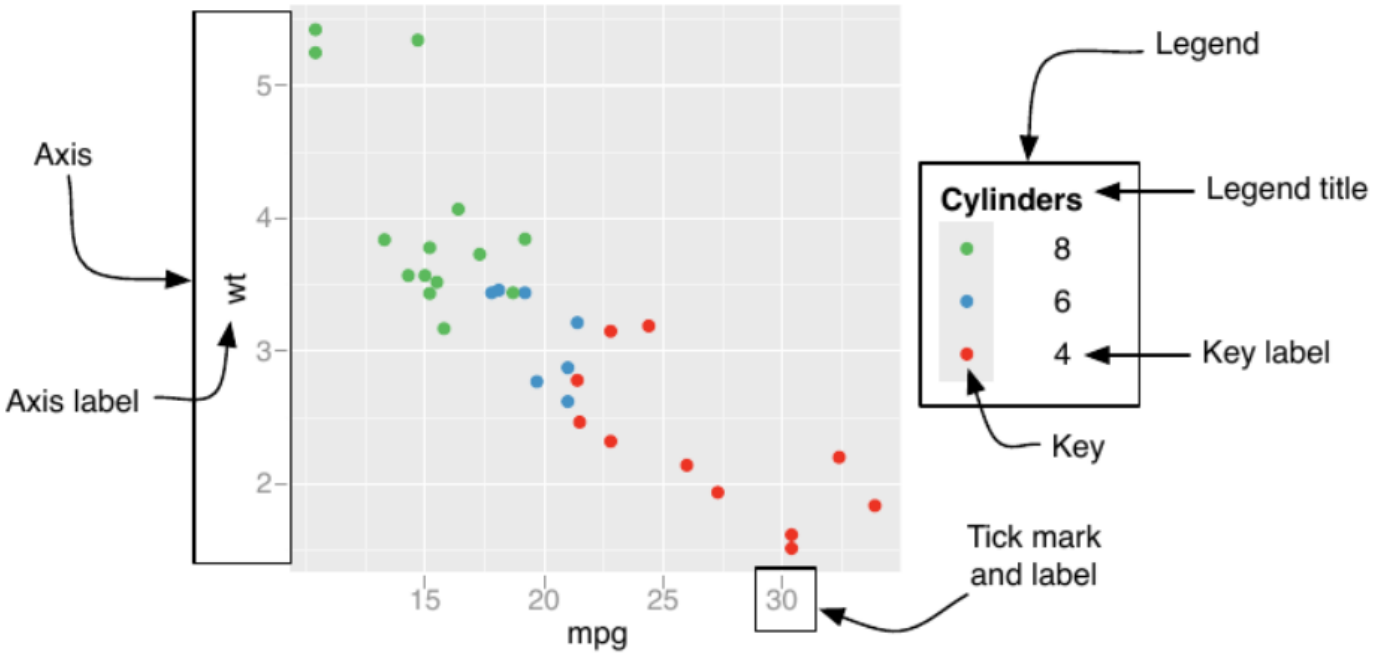
5100

Программирование в среде R

Шевцов Василий Викторович,
директор ДИТ РУДН, shevtsov_vv@rudn.university

Оси, шкалы, координаты, легенды

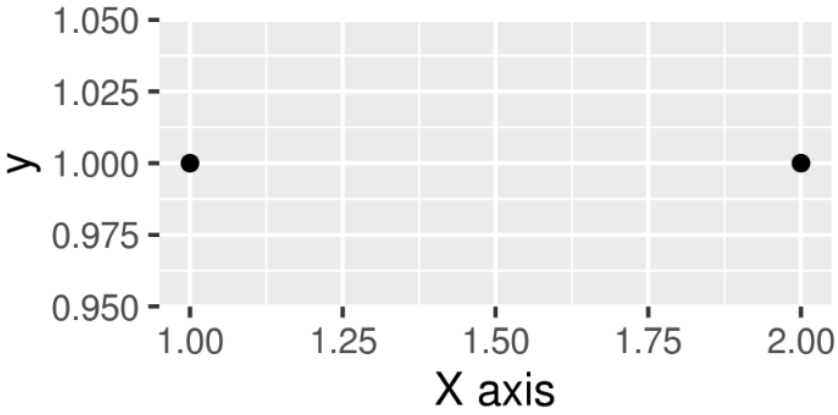
Общие положения

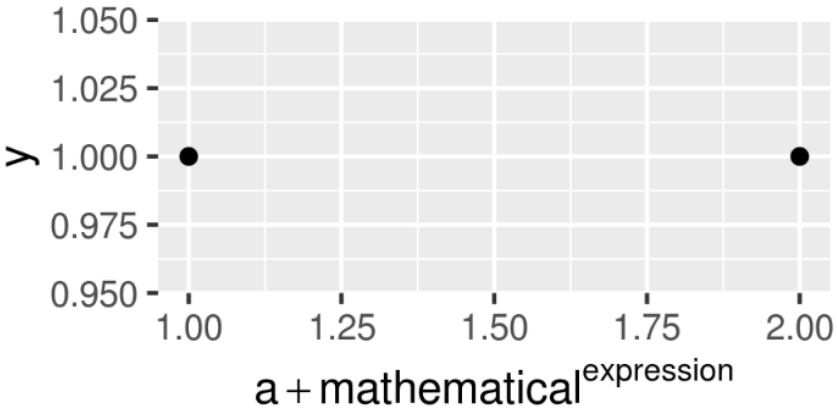


Axis	Legend	Argument name
Label	Title	name
Ticks & grid line	Key	breaks
Tick label	Key label	labels

Подписи осей

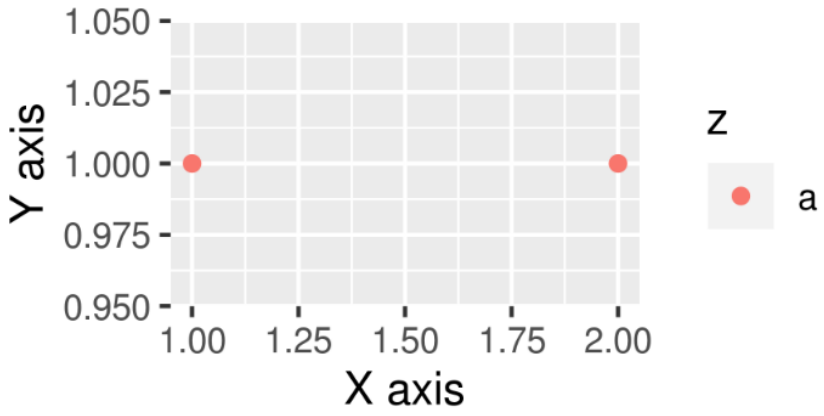
```
df <- data.frame(x = 1:2, y = 1, z = "a")
p <- ggplot(df, aes(x, y)) + geom_point()
p + scale_x_continuous("X axis")
p + scale_x_continuous(quote(a + mathematical ^ expression))
```

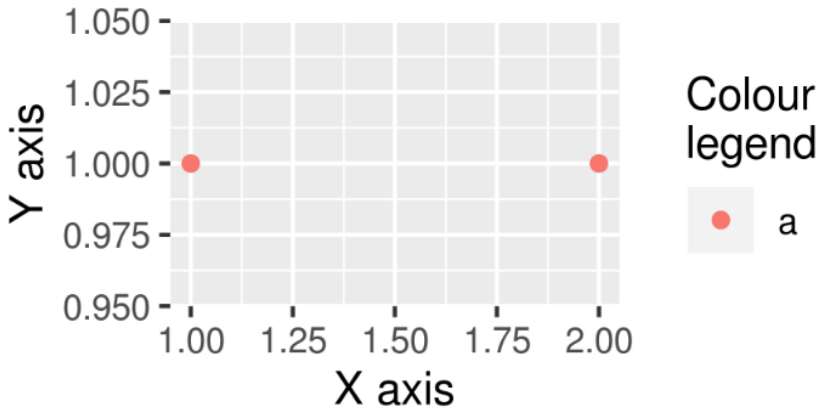




Подписи осей и легенды

```
p <- ggplot(df, aes(x, y)) + geom_point(aes(colour = z))
p +
  xlab("X axis") +
  ylab("Y axis")
p + labs(x = "X axis", y = "Y axis", colour = "Colour\nlegend")
```

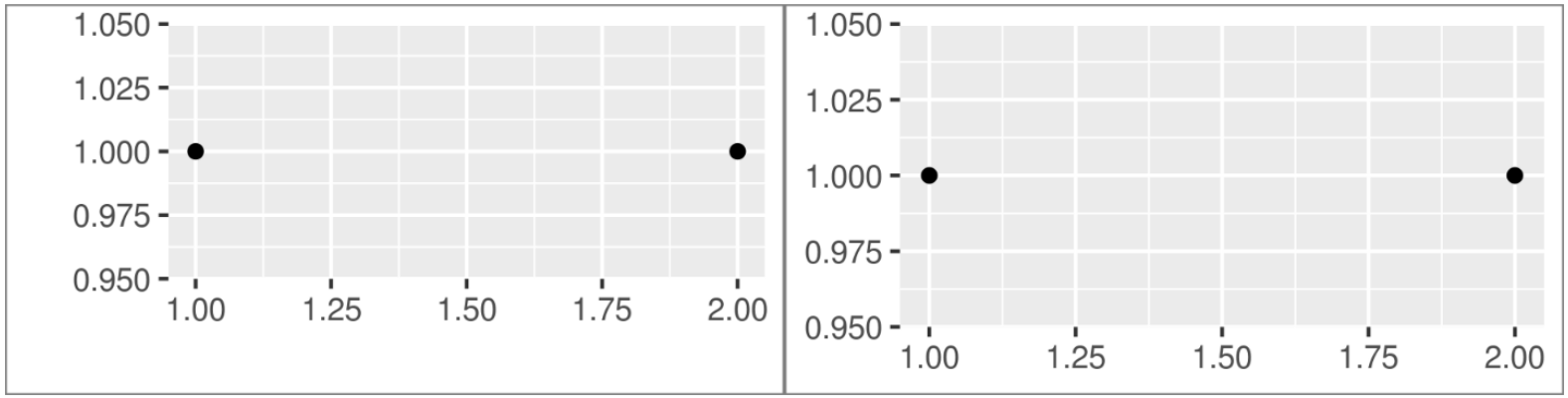




Подписи осей

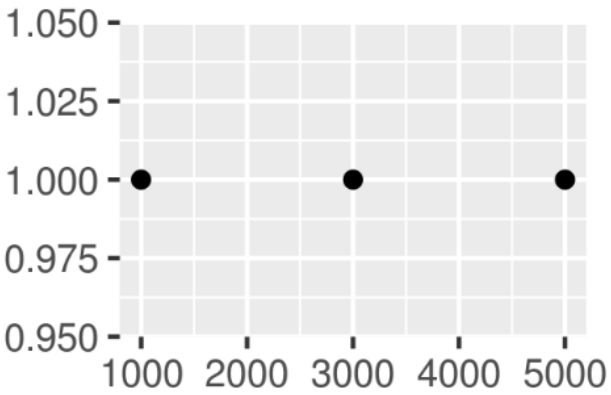
```
p <- ggplot(df, aes(x, y)) +
  geom_point() +
  theme(plot.background = element_rect(colour = "grey50"))
p + labs(x = "",  y = "")
p + labs(x = NULL, y = NULL)
```

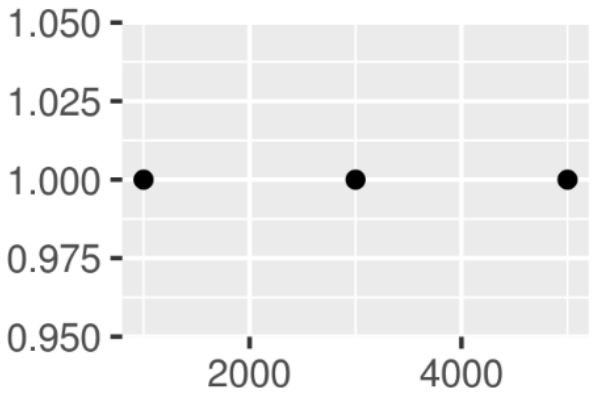
Удаление подписей осей:
"" – удаляет подписи, но сохраняет для них место
NULL – удаляет подписи и место

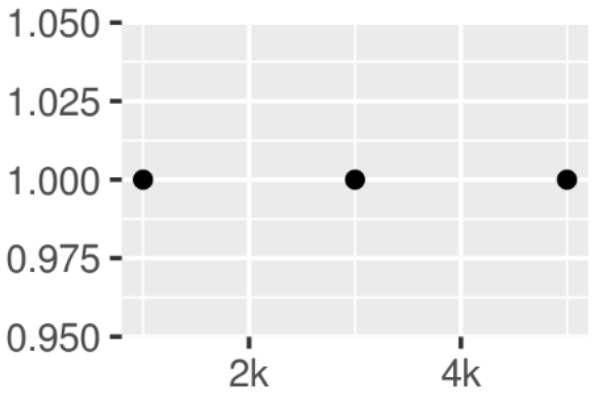


Шкалы и метки

```
df <- data.frame(x = c(1, 3, 5) * 1000, y = 1)
axs <- ggplot(df, aes(x, y)) +
  geom_point() +
  labs(x = NULL, y = NULL)
axs
axs + scale_x_continuous(breaks = c(2000, 4000))
axs + scale_x_continuous(breaks = c(2000, 4000), labels = c("2k", "4k"))
```

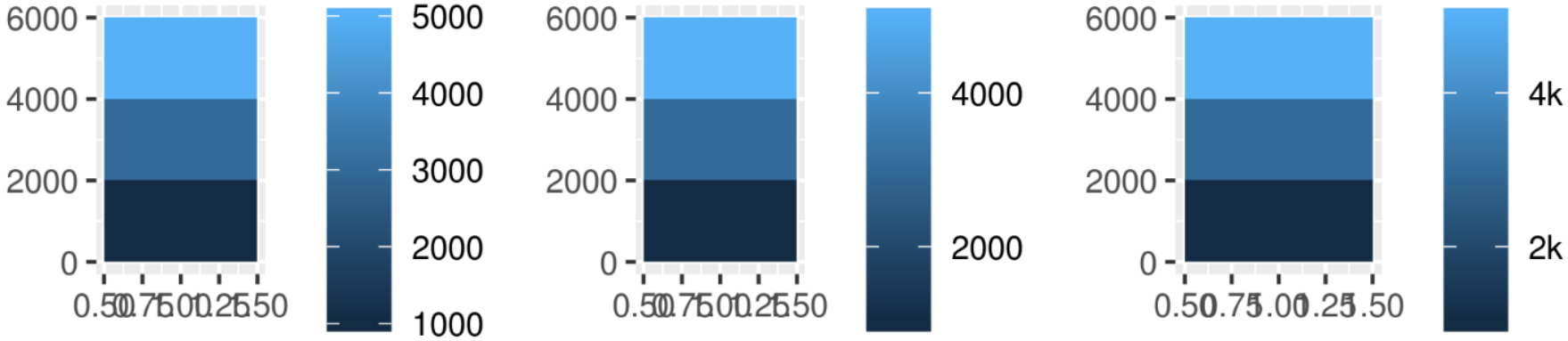






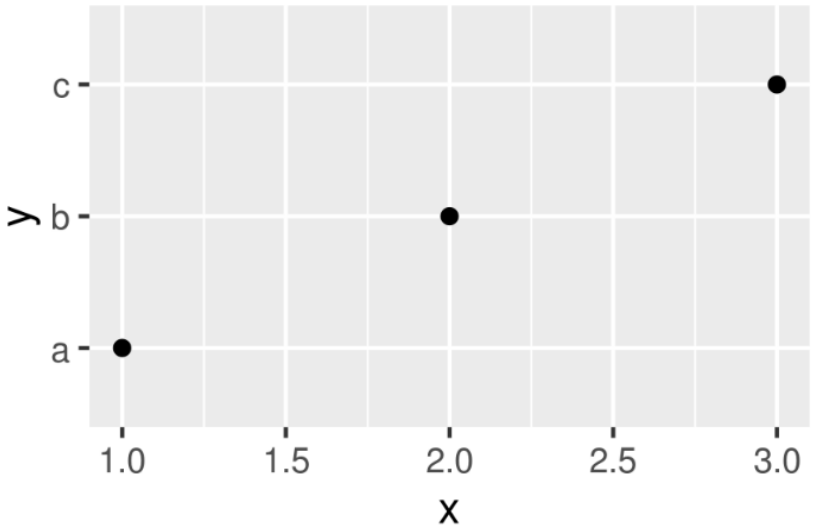
Легенды

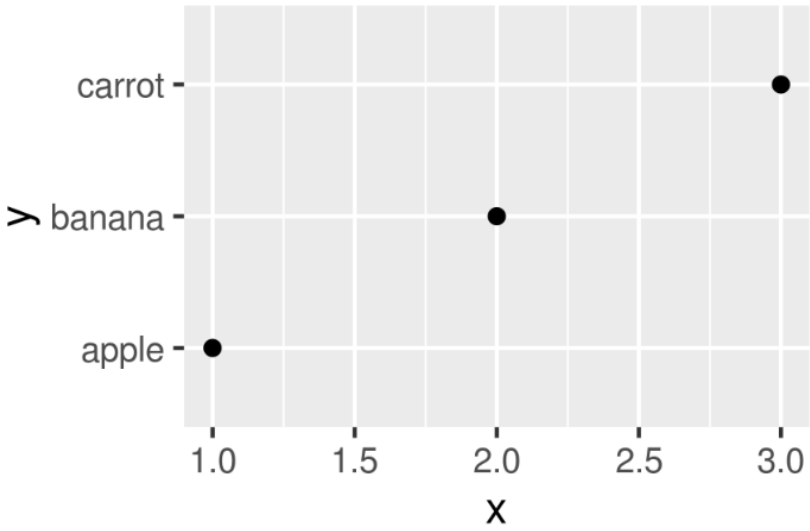
```
leg <- ggplot(df, aes(y, x, fill = x)) +
  geom_tile() +
  labs(x = NULL, y = NULL)
leg
leg + scale_fill_continuous(breaks = c(2000, 4000))
leg + scale_fill_continuous(breaks = c(2000, 4000), labels = c("2k", "4k"))
```



Подписи осей

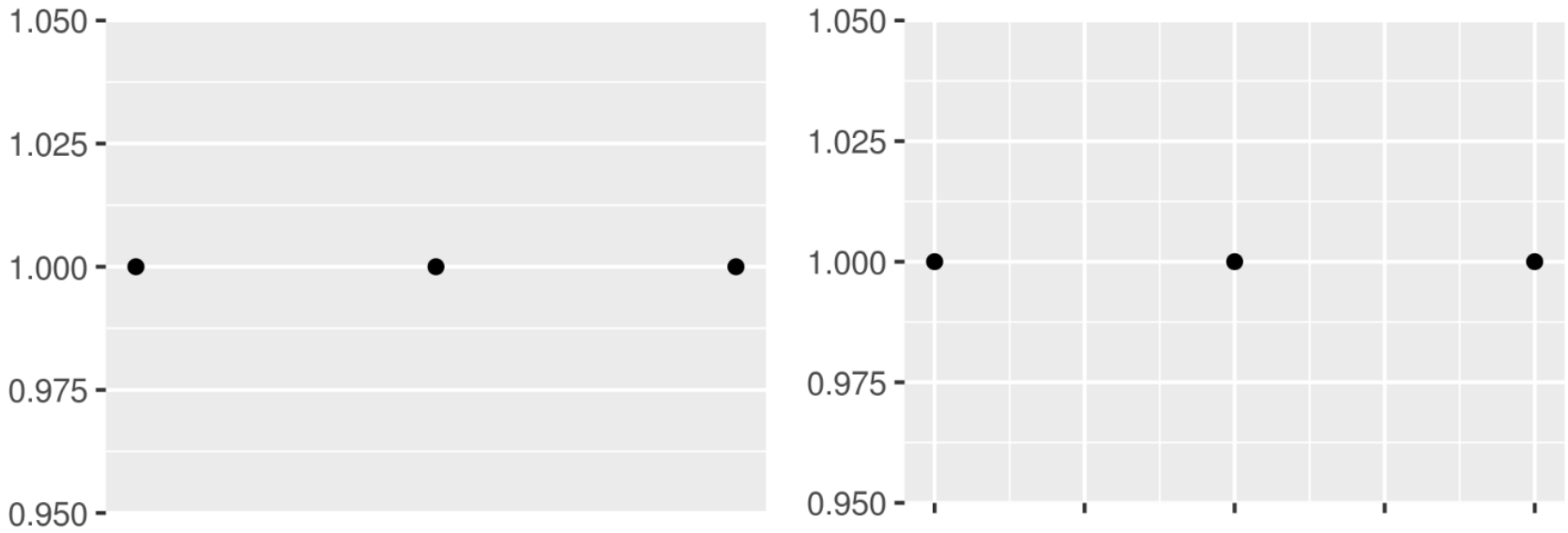
```
df2 <- data.frame(x = 1:3, y = c("a", "b", "c"))
ggplot(df2, aes(x, y)) +
  geom_point()
ggplot(df2, aes(x, y)) +
  geom_point() +
  scale_y_discrete(labels = c(a = "apple", b = "banana", c = "carrot"))
```





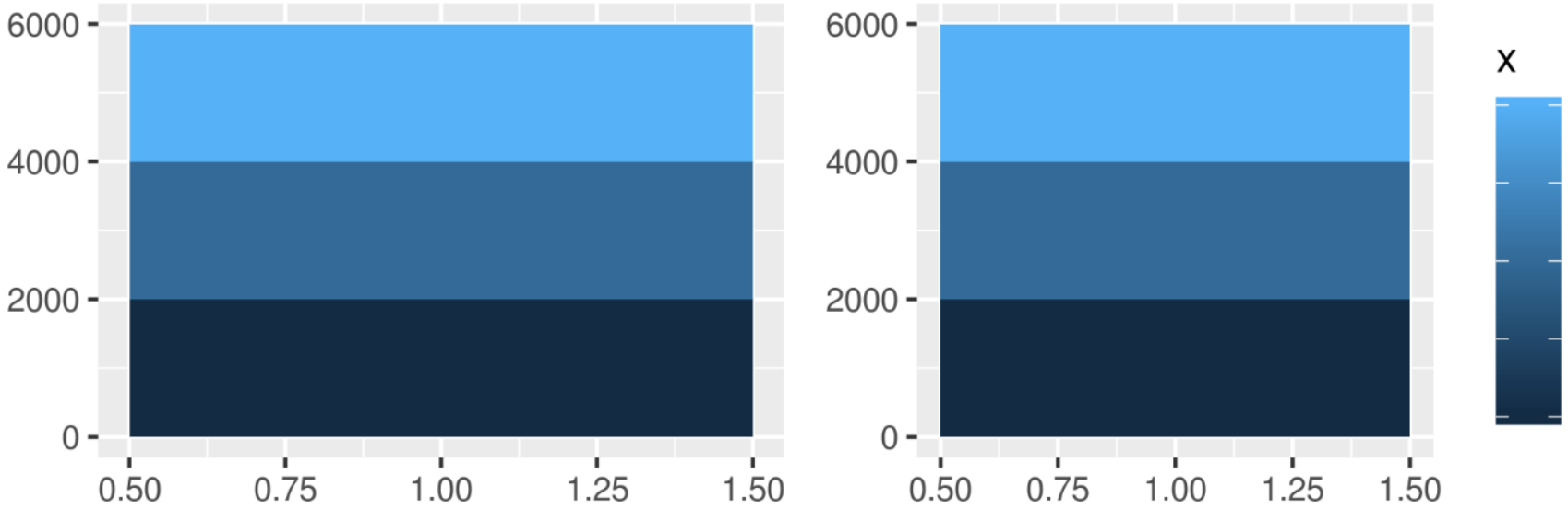
Сетка

axs + scale_x_continuous(breaks = NULL)
axs + scale_x_continuous(labels = NULL)



Легенда

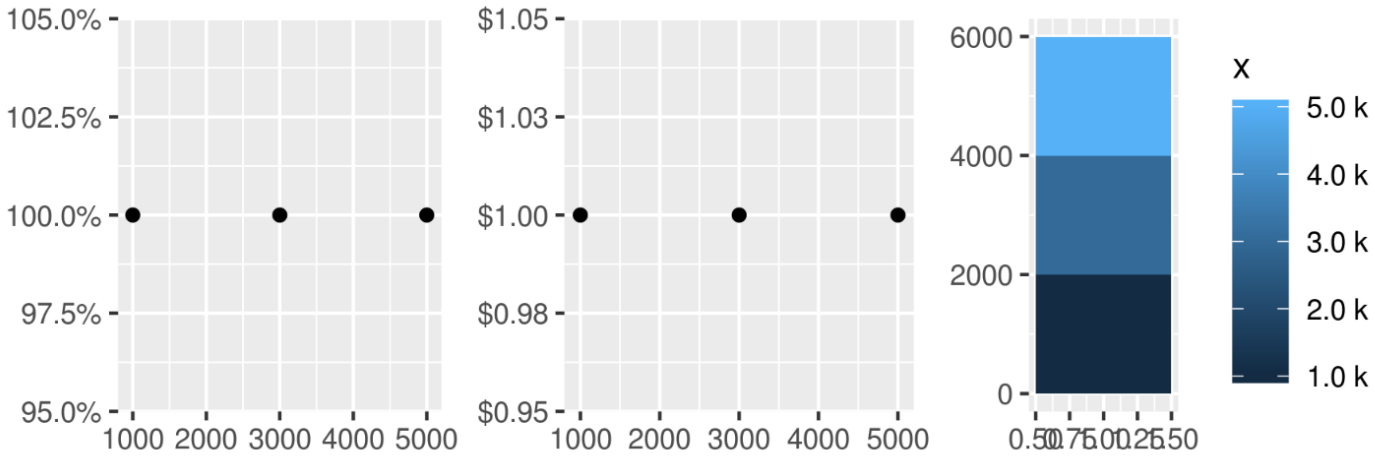
leg + scale_fill_continuous(breaks = NULL)
leg + scale_fill_continuous(labels = NULL)



Формат подисей

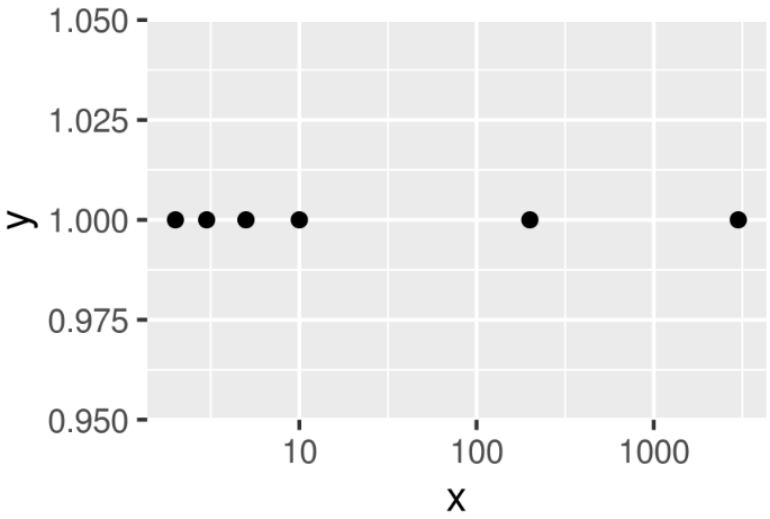
scales::comma_format()
добавляет запятые, чтобы облегчить чтение больших чисел.
scales::unit_format(unit, scale)
добавляет суффикс единицы измерения, опционально масштабируя его.
scales::dollar_format(prefix, suffix)
отображает значения валют, округляя их до двух знаков после запятой и добавляя префикс или суффикс.
scales::wrap_format()
длинные надписи в несколько строк.

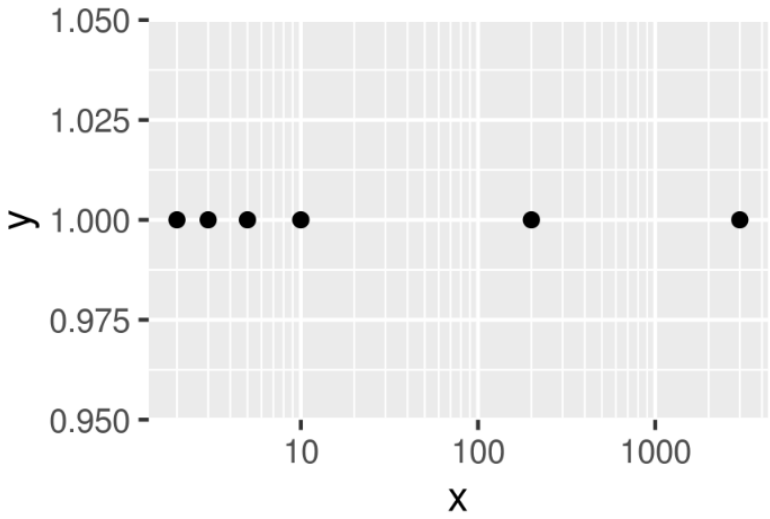
```
axs + scale_y_continuous(labels = scales::percent_format())
axs + scale_y_continuous(labels = scales::dollar_format(prefix = "$"))
leg + scale_fill_continuous(labels = scales::unit_format(unit = "k", scale = 1e-3))
```



Дополнительные линии сетки

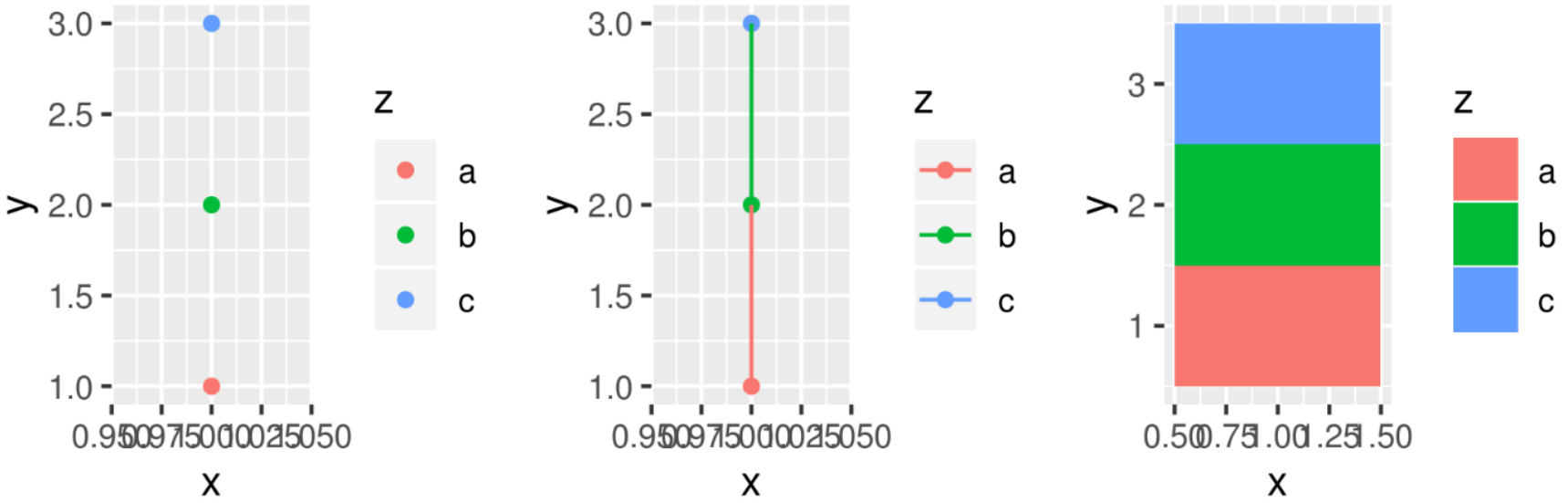
```
df <- data.frame(x = c(2, 3, 5, 10, 200, 3000), y = 1)
ggplot(df, aes(x, y)) +
  geom_point() +
  scale_x_log10()

mb <- as.numeric(1:10 %o% 10 ^ (0:4))
ggplot(df, aes(x, y)) +
  geom_point() +
  scale_x_log10(minor_breaks = mb)
```





Слои и легенды

В легенде может потребоваться нарисовать символы из нескольких слоев. Например, если вы нанесли цвет на обе точки и линии, то клавиши будут показывать как точки, так и линии. Если вы нанесли на карту цвет заливки, то получите прямоугольник. Обратите внимание на то, как меняется легенда в приведенных ниже сюжетах:



Слои и легенды

```
df <- data.frame(x = 1:3, y = 1:3, z = c("a", "b", "c"))
ggplot(df, aes(y, y)) +
  geom_point(size = 4, colour = "grey20") +
  geom_point(aes(colour = z), size = 2)
ggplot(df, aes(y, y)) +
  geom_point(size = 4, colour = "grey20", show.legend = TRUE) +
  geom_point(aes(colour = z), size = 2)
```

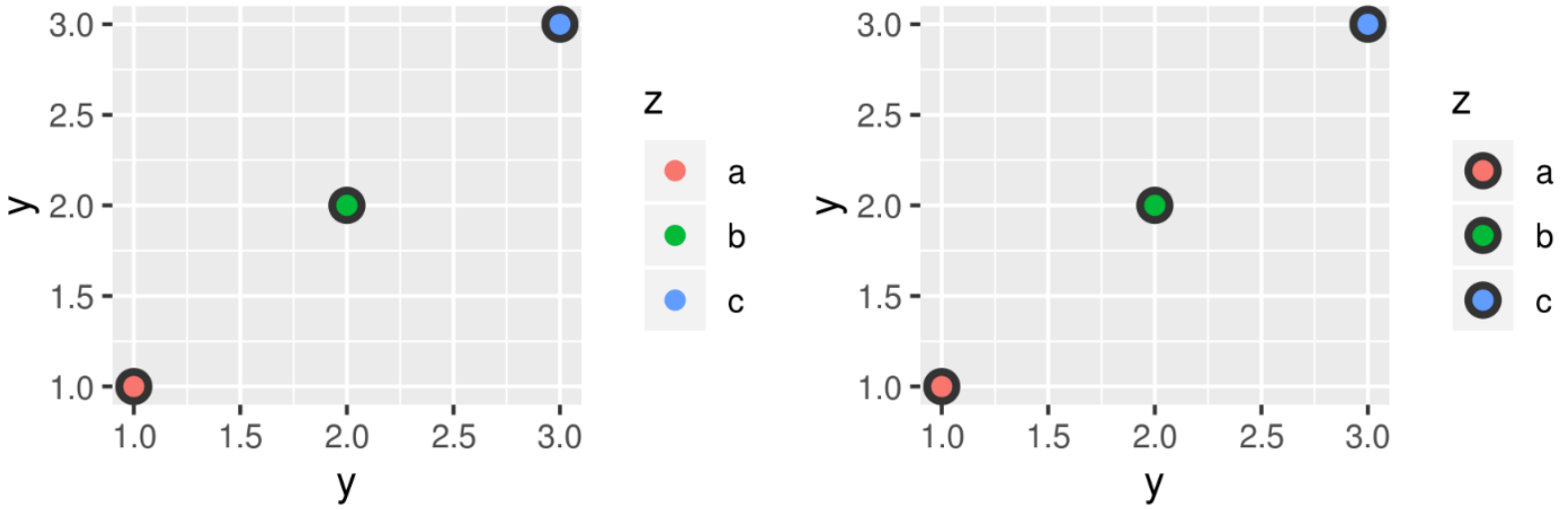
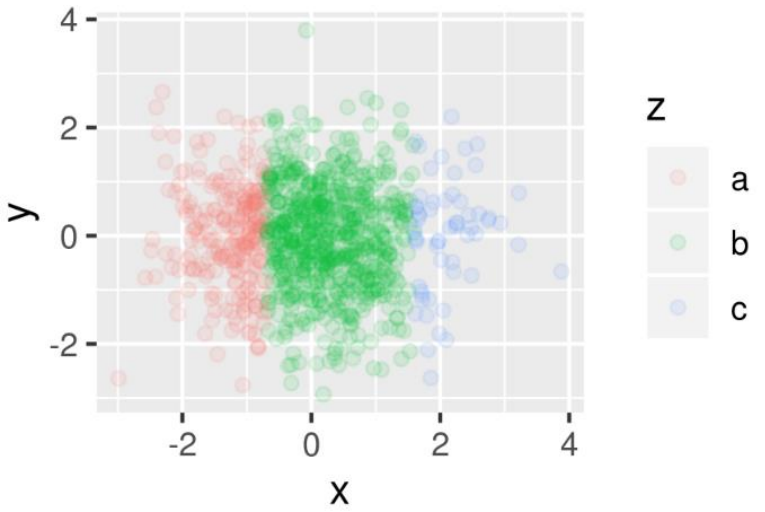


График и легенда

geom() в легенде отображаются иначе, чем geom() в графике. Это особенно полезно, когда вы используется прозрачность или размер.

```
norm <- data.frame(x = rnorm(1000), y = rnorm(1000))
norm$z <- cut(norm$x, 3, labels = c("a", "b", "c"))
ggplot(norm, aes(x, y)) +
  geom_point(aes(colour = z), alpha = 0.1)
ggplot(norm, aes(x, y)) +
  geom_point(aes(colour = z), alpha = 0.1) +
  guides(colour = guide_legend(override.aes = list(alpha = 1)))
```



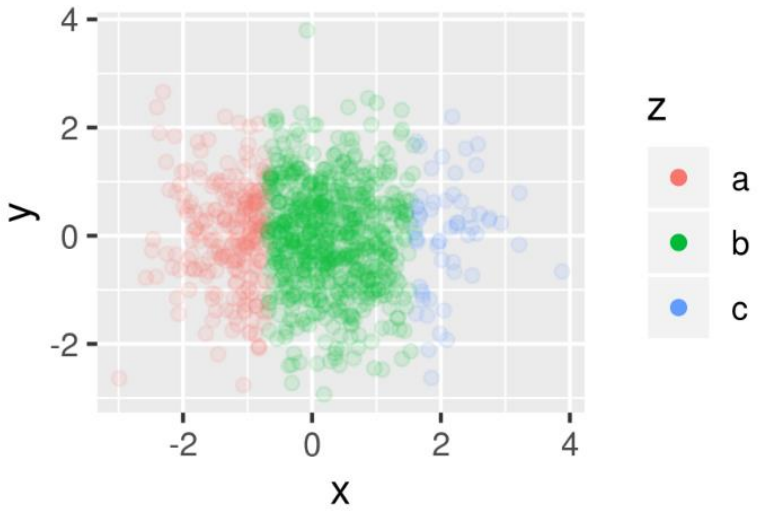
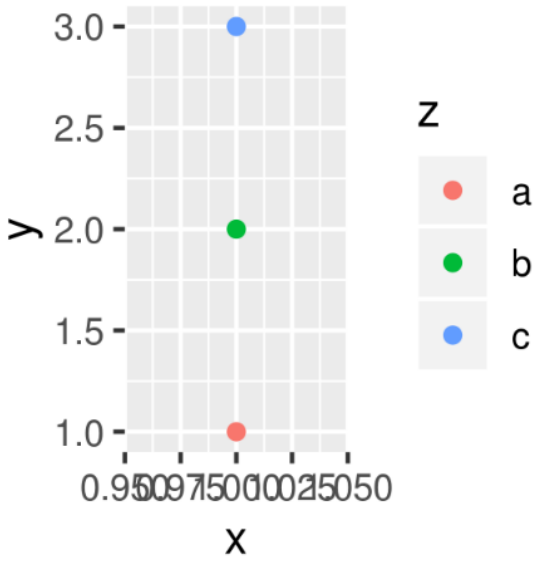
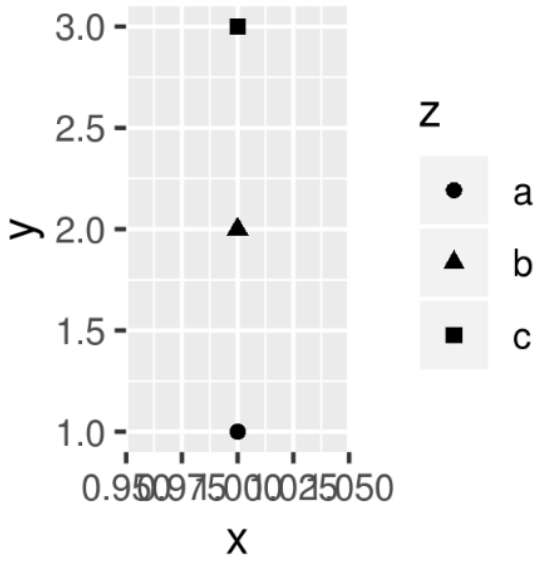


График и легенда

```
df <- data.frame(x = 1:3, y = 1:3, z = c("a", "b", "c"))
ggplot(df, aes(x, y)) + geom_point(aes(colour = z))
ggplot(df, aes(x, y)) + geom_point(aes(shape = z))
ggplot(df, aes(x, y)) + geom_point(aes(shape = z, colour = z))
```





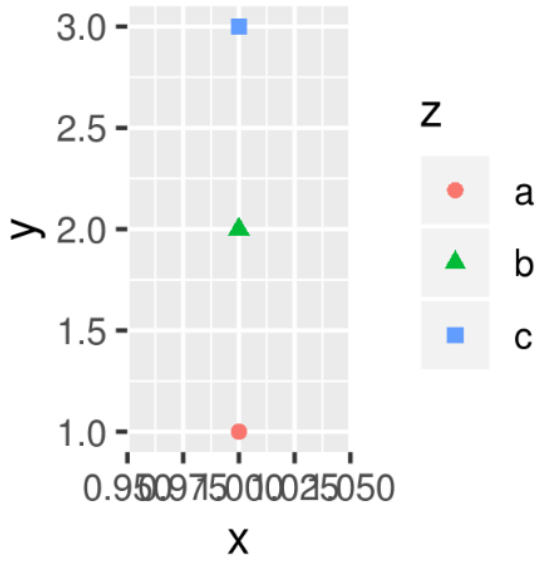
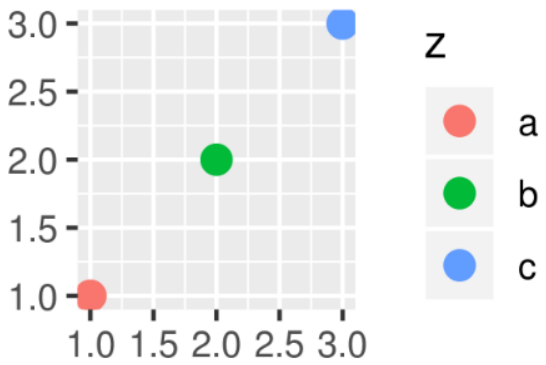
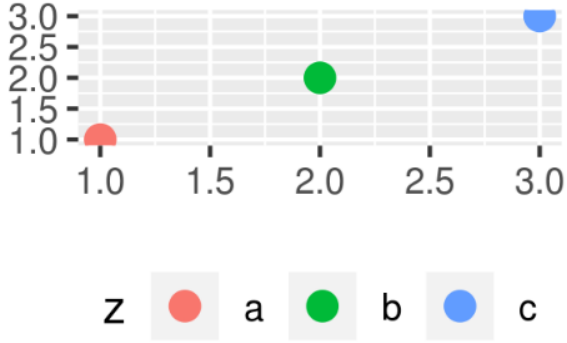
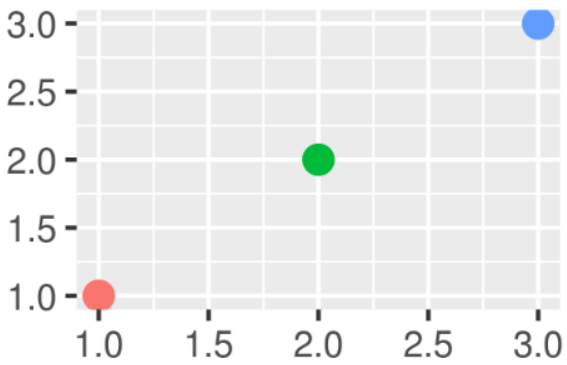


График и легенда

```
df <- data.frame(x = 1:3, y = 1:3, z = c("a", "b", "c"))
base <- ggplot(df, aes(x, y)) +
  geom_point(aes(colour = z), size = 3) +
  xlab(NULL) +
  ylab(NULL)

base + theme(legend.position = "right") # the default
base + theme(legend.position = "bottom")
base + theme(legend.position = "none")
```





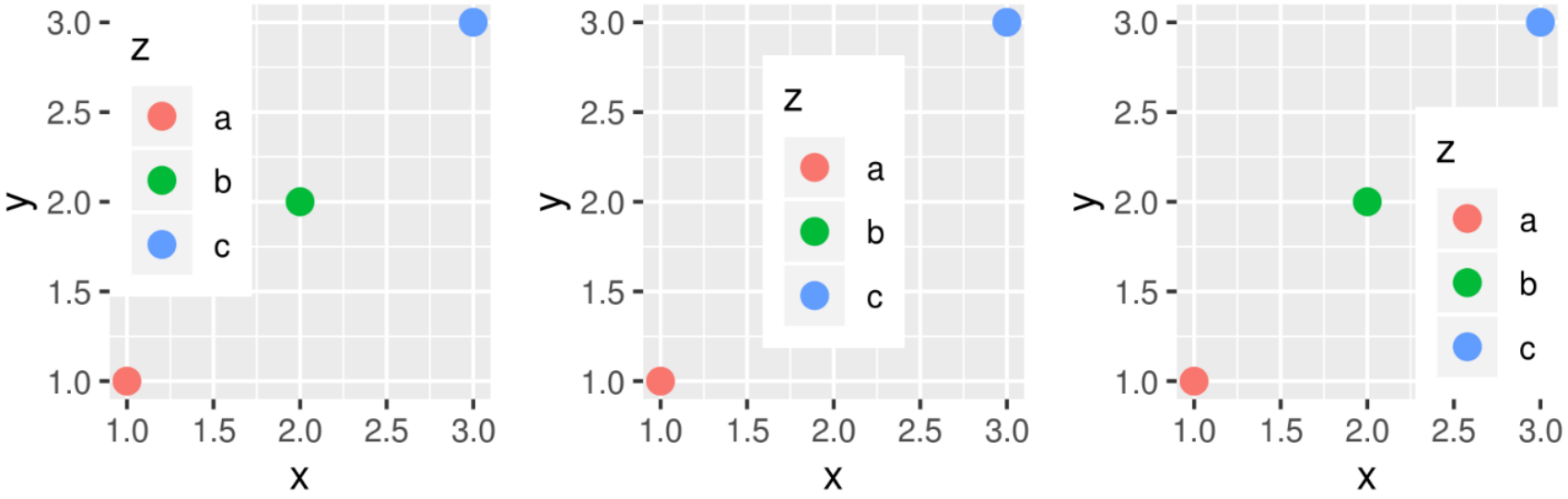


Опции легенды

- **legend.direction:** layout of items in legends (“horizontal” or “vertical”).
- **legend.box:** arrangement of multiple legends (“horizontal” or “vertical”).
- **legend.box.just:** justification of each legend within the overall bounding box, when there are multiple legends (“top”, “bottom”, “left”, or “right”).

Опции легенды

```
base <- ggplot(df, aes(x, y)) +
  geom_point(aes(colour = z), size = 3)

base + theme(legend.position = c(0, 1), legend.justification = c(0, 1))
base + theme(legend.position = c(0.5, 0.5), legend.justification = c(0.5, 0.5))
base + theme(legend.position = c(1, 0), legend.justification = c(1, 0))
```



Темы

theme()	Modify components of a theme
theme_grey() theme_gray() theme_bw() theme_linedraw() theme_light() theme_dark() theme_minimal() theme_classic() theme_void() theme_test()	Complete themes
theme_get() theme_set() theme_update() theme_replace() `%+replace%`	Get, set, and modify the active theme
margin() element_blank() element_rect() element_line() element_text() rel()	Theme elements

Система координат

coord_cartesian()	Cartesian coordinates
coord_fixed()	Cartesian coordinates with fixed "aspect ratio"
coord_flip()	Cartesian coordinates with x and y flipped
coord_map() coord_quickmap()	Map projections
coord_polar()	Polar coordinates
coord_trans()	Transformed Cartesian coordinate system

Встроенные данные

diamonds	Prices of over 50,000 round cut diamonds
economics economics_long	US economic time series
faithfuld	2d density estimate of Old Faithful data
midwest	Midwest demographics
mpg	Fuel economy data from 1999 to 2008 for 38 popular models of cars
msleep	An updated and expanded version of the mammals sleep dataset
presidential	Terms of 11 presidents from Eisenhower to Obama
seals	Vector field of seal movements
txhousing	Housing sales in TX
luv_colours	colors() in Luv space

Syntax	Meaning	Syntax	Meaning
x + y	x plus y	x %~% y	x is distributed as y
x - y	x minus y	plain(x)	draw x in normal font
x*y	juxtapose x and y	bold(x)	draw x in bold font
x/y	x forwardslash y	italic(x)	draw x in italic font
x %+-% y	x plus or minus y	bolditalic(x)	draw x in bolditalic font
x %/% y	x divided by y	symbol(x)	draw x in symbol font
x %*% y	x times y	list(x, y, z)	comma-separated list
x %.% y	x cdot y	...	ellipsis (height varies)
x[i]	x subscript i	cdots	ellipsis (vertically centred)
x^2	x superscript 2	ldots	ellipsis (at baseline)
paste(x, y, z)	juxtapose x, y, and z	x %subset% y	x is a proper subset of y
sqrt(x)	square root of x	x %subseteq% y	x is a subset of y
sqrt(x, y)	yth root of x	x %notsubset% y	x is not a subset of y
x == y	x equals y	x %supset% y	x is a proper superset of y
x != y	x is not equal to y	x %supseteq% y	x is a superset of y
x < y	x is less than y	x %in% y	x is an element of y
x <= y	x is less than or equal to y	x %notin% y	x is not an element of y
x > y	x is greater than y	hat(x)	x with a circumflex
x >= y	x is greater than or equal to y	tilde(x)	x with a tilde
!x	not x	dot(x)	x with a dot
x %~~% y	x is approximately equal to y	ring(x)	x with a ring
x %=~% y	x and y are congruent	bar(xy)	xy with bar
x %==% y	x is defined as y	widehat(xy)	xy with a wide circumflex
x %prop% y	x is proportional to y	widetilde(xy)	xy with a wide tilde

Syntax	Meaning	Syntax	Meaning
x %<->% y	x double-arrow y	scriptscriptstyle(x)	draw x in very small size
x %->% y	x right-arrow y	underline(x)	draw x underlined
x %<-% y	x left-arrow y	x ~~ y	put extra space between x and y
x %up% y	x up-arrow y	x + phantom(0) + y	leave gap for "0", but don't draw it
x %down% y	x down-arrow y	x + over(1, phantom(0))	leave vertical gap for "0" (don't draw)
x %<=>% y	x is equivalent to y	frac(x, y)	x over y
x %=>% y	x implies y	over(x, y)	x over y
x %<=% y	y implies x	atop(x, y)	x over y (no horizontal bar)
x %dblup% y	x double-up-arrow y	sum(x[i], i==1, n)	sum x[i] for i equals 1 to n
x %dbldown% y	x double-down-arrow y	prod(plain(P)(X==x), x)	product of P(X=x) for all values of x
alpha -- omega	Greek symbols	integral(f(x)*dx, a, b)	definite integral of f(x) wrt x
Alpha -- Omega	uppercase Greek symbols	union(A[i], i==1, n)	union of A[i] for i equals 1 to n
theta1, phi1, sigma1, omega1	cursive Greek symbols	intersect(A[i], i==1, n)	intersection of A[i]
Upsilon1	capital upsilon with hook	lim(f(x), x %->% 0)	limit of f(x) as x tends to 0
aleph	first letter of Hebrew alphabet	min(g(x), x > 0)	minimum of g(x) for x greater than 0
infinity	infinity symbol	inf(S)	infimum of S
partialdiff	partial differential symbol	sup(S)	supremum of S
nabla	nabla, gradient symbol	x^y + z	normal operator precedence
32*degree	32 degrees	x^(y + z)	visible grouping of operands
60*minute	60 minutes of angle	x^{y + z}	invisible grouping of operands
30*second	30 seconds of angle	group("",list(a, b),"]")	specify left and right delimiters
displaystyle(x)	draw x in normal size (extra spacing)	bgroup("(",atop(x,y),")")	use scalable delimiters
textstyle(x)	draw x in normal size	group(lceil, x, rceil)	special delimiters
scriptstyle(x)	draw x in small size	group(lfloor, x, rfloor)	special delimiters

Подписи осей

labs() xlab() ylab() ggtitle()	Modify axis, legend, and plot labels
lims() xlim() ylim()	Set scale limits
expand_limits()	Expand the plot limits, using data
expansion() expand_scale()	Generate expansion vector for scales

Основные формы для маркеров (точек):

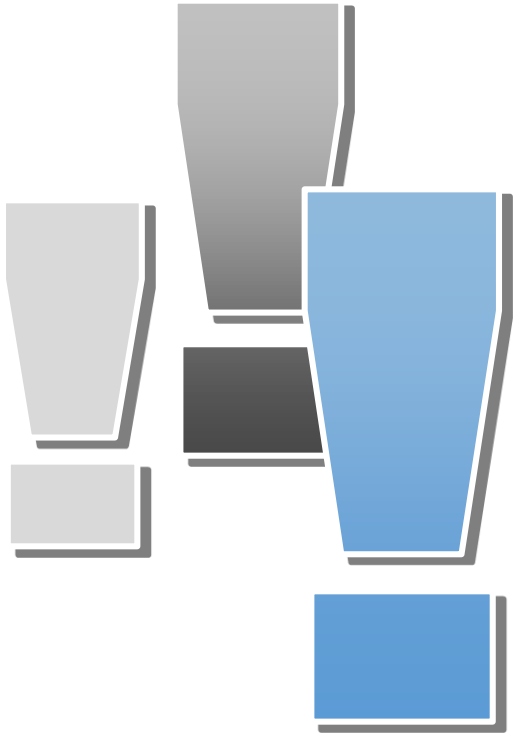
Без заливки

shape=0 - квадрат,
shape=1 - круг,
shape=2 - треугольник,
shape=3 - крест,
shape=4 - крест (диагональный),
shape=5 - ромб.

С заливкой

shape=15 - квадрат,
shape=16 - круг,
shape=17 - треугольник,
shape=18 - ромб.

Спасибо за внимание!



Шевцов Василий Викторович

shevtsov_vv@rudn.university
+7(903)144-53-57