Available themes in the INBOtheme package

ir. Thierry Onkelinx

November 10, 2014

Contents

1 Introduction

This vignette gives a short introduction on the INBOtheme package. The INBOtheme package provides a few themes for ggplot2. This vignette demonstrates the look and feel of the themes by showing several plots. Many of the figures in this vignettes are taken from the examples from the ggplot2 package.

The ggplot2 package must be loaded prior to INBOtheme.

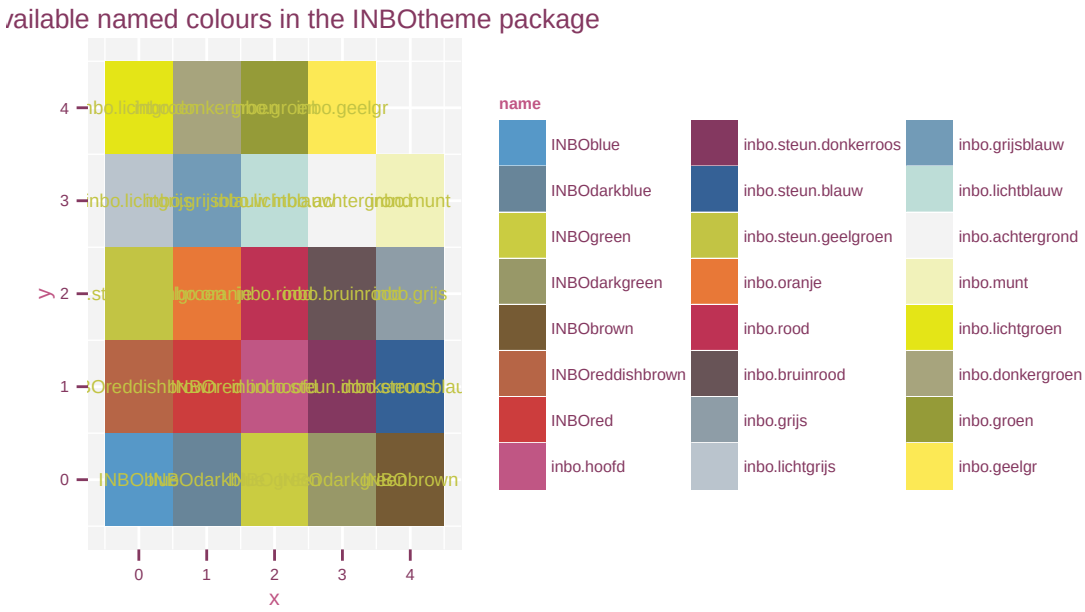
```
options(stringsAsFactors = FALSE)
library(ggplot2)
library(INBOtheme)

##
## Attaching package:  'INBOtheme'
##
## The following objects are masked from 'package:ggplot2':
##
##    scale_colour_discrete, scale_colour_gradient, scale_fill_discrete,
##    scale_fill_gradient

# Create a simple example dataset
pp <- function (n, r = 4){
  x <- seq(-r * pi, r * pi, len = n)
  df <- expand.grid(x = x, y = x)
  df$r <- sqrt(df$x ^ 2 + df$y ^ 2)
  df$z <- cos(df$r ^ 2) * exp(-df$r / 6)
  df
}

# Create a simple example dataset
df <- data.frame(
```

```
  trt = factor(c(1, 1, 2, 2)),
  resp = c(1, 5, 3, 4),
  group = factor(c(1, 2, 1, 2)),
  se = c(0.1, 0.3, 0.3, 0.2)
)
```

2 Available colours

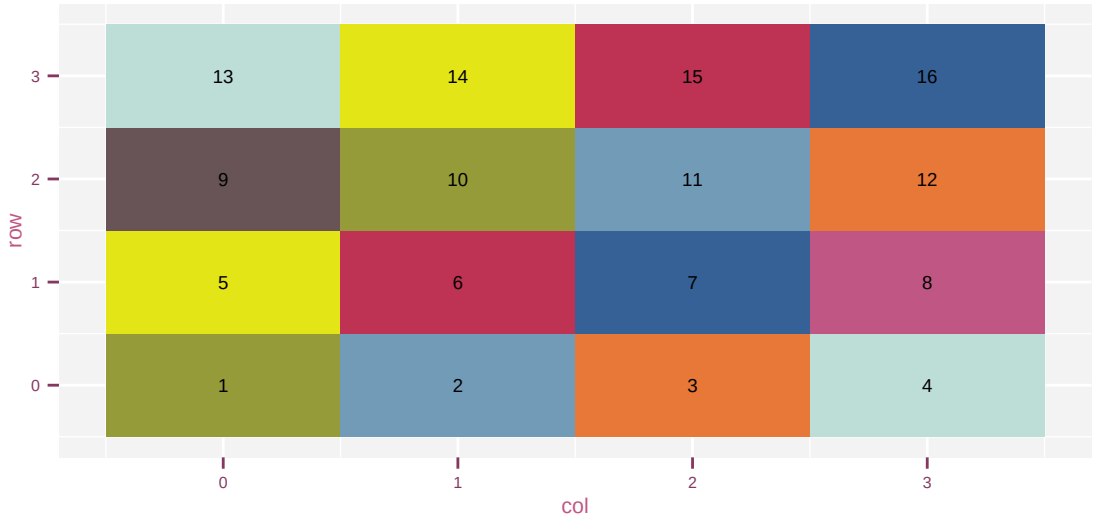
2.1 Named colours

```
palette <- data.frame(
  name = c(
    #version <= 2014
      "INBOblue", "INBOdarkblue", "INBOgreen", "INBOdarkgreen", "INBObrown",
      "INBOreddishbrown", "INBOred",
    #version >= 2015
      "inbo.hoofd", "inbo.steun.donkerroos", "inbo.steun.blauw",
      "inbo.steun.geelgroen", "inbo.oranje", "inbo.rood", "inbo.bruinrood",
      "inbo.grijs", "inbo.lichtgrijs", "inbo.grijsblauw", "inbo.lichtblauw",
      "inbo.achtergrond", "inbo.munt", "inbo.lichtgroen", "inbo.donkergroen",
      "inbo.groen", "inbo.geelgr"
  )
)
palette$x <- (seq_along(palette$name) - 1) %% ceiling(sqrt(nrow(palette)))
palette$y <- (seq_along(palette$name) - 1) %/% ceiling(sqrt(nrow(palette)))
palette$colour <- sapply(
  palette$name,
  function(i){
    eval(parse(text = i))
  }
)
rownames(palette) <- palette$name
palette$name <- factor(palette$name, levels = palette$name)
ggplot(palette, aes(x = x, y = y, label = name, fill = name)) +
  geom_tile() + geom_text() +
  scale_fill_manual(
    values = palette$colour,
    guide = guide_legend(ncol = 3)
  ) +
  ggtitle("All available named colours in the INBOtheme package")
```

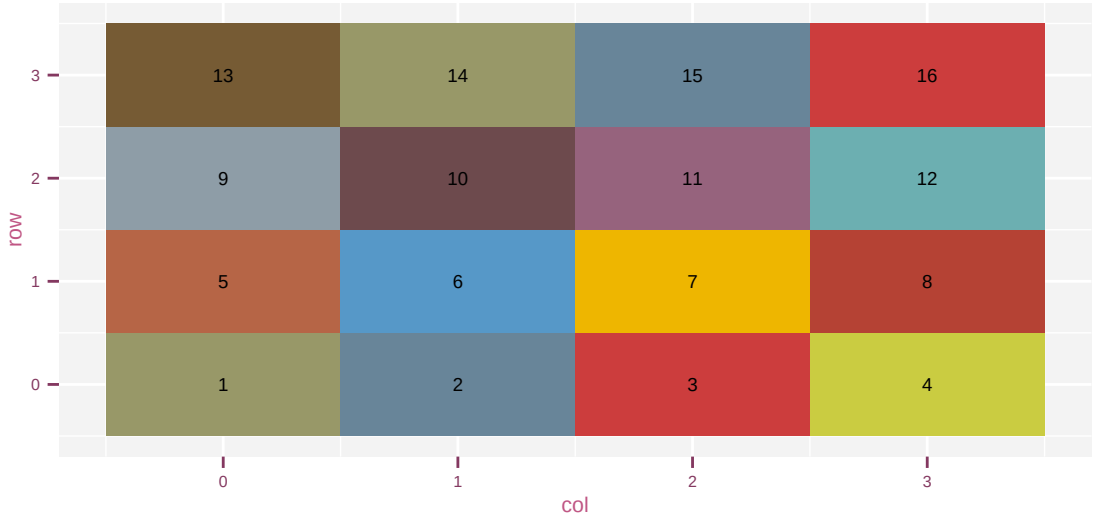


2.2 Standard INBO palettes for discrete factors

```
# version >= 2015
n <- 16
inbo.2015.colours(n = n)

## Warning in inbo.2015.colours(n = n):  generated palette has duplicated colours.  The
palette has only 9 unique colours.

##  [1] "#959B38" "#729BB7" "#E87837" "#BDDDD7" "#E4E517" "#BE3254" "#356196"
##  [8] "#C05684" "#685457" "#959B38" "#729BB7" "#E87837" "#BDDDD7" "#E4E517"
## [15] "#BE3254" "#356196"

palette <- data.frame(n = seq_len(n))
palette$row <- (palette$n - 1) %/% ceiling(sqrt(n))
palette$col <- (palette$n - 1) %% ceiling(sqrt(n))
ggplot(palette, aes(x = col, y = row, fill = factor(n))) +
  geom_tile() +
  geom_text(aes(label = n), colour = "black") +
  scale_fill_manual(values = inbo.2015.colours(n = n), guide = "none")

## Warning in inbo.2015.colours(n = n):  generated palette has duplicated colours.  The
palette has only 9 unique colours.
```



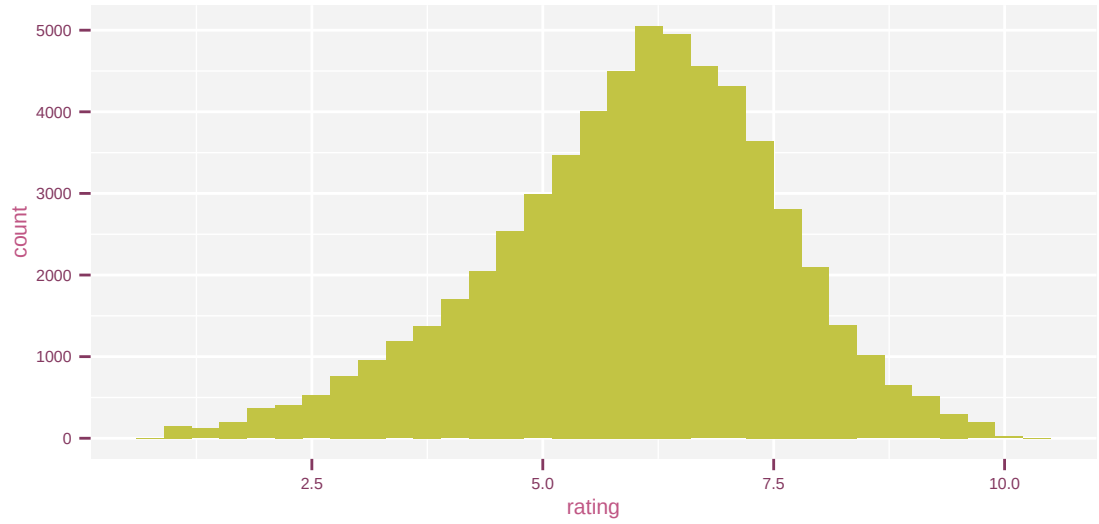

```
# version <= 2014
n <- 16
INBO.colours(n = n)

## Warning in INBO.colours(n = n):  generated palette has duplicated colours.  The palette
has only 13 unique colours.

##  [1] "#989868" "#688599" "#CC3D3D" "#CACC41" "#B66546" "#5698C8" "#EEB600"
##  [8] "#B54234" "#8E9DA7" "#6D4A4D" "#96637D" "#6CAFB1" "#765B34" "#989868"
## [15] "#688599" "#CC3D3D"

palette <- data.frame(n = seq_len(n))
palette$row <- (palette$n - 1) %/% ceiling(sqrt(n))
palette$col <- (palette$n - 1) %% ceiling(sqrt(n))
ggplot(palette, aes(x = col, y = row, fill = factor(n))) +
  geom_tile() +
  geom_text(aes(label = n), colour = "black") +
  scale_fill_manual(values = INBO.colours(n = n), guide = "none")

## Warning in INBO.colours(n = n):  generated palette has duplicated colours.  The palette
has only 13 unique colours.
```



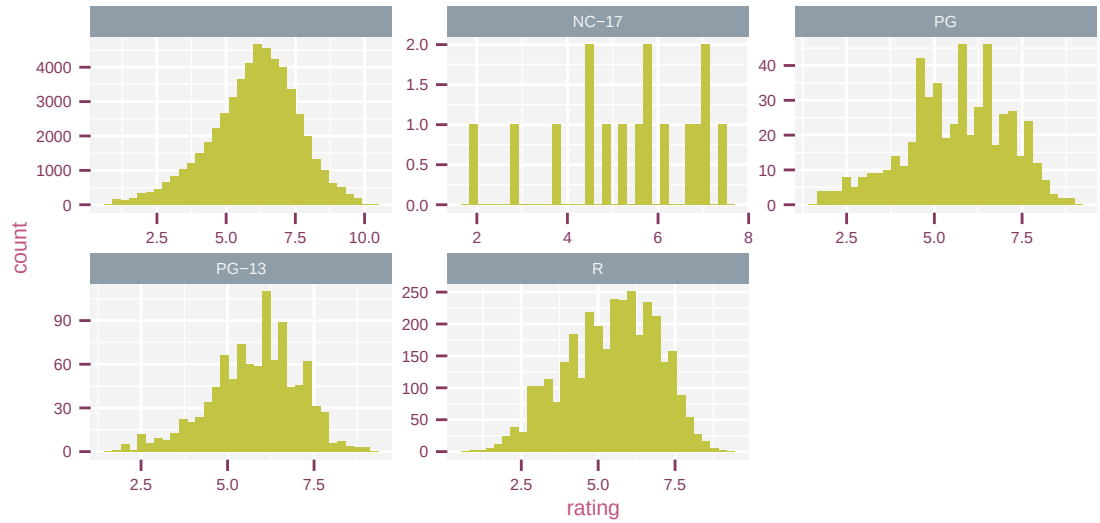
3 theme_inbo2015

```
theme_set(theme_inbo2015(8))
switchColour(inbo.steun.geelgroen)
```

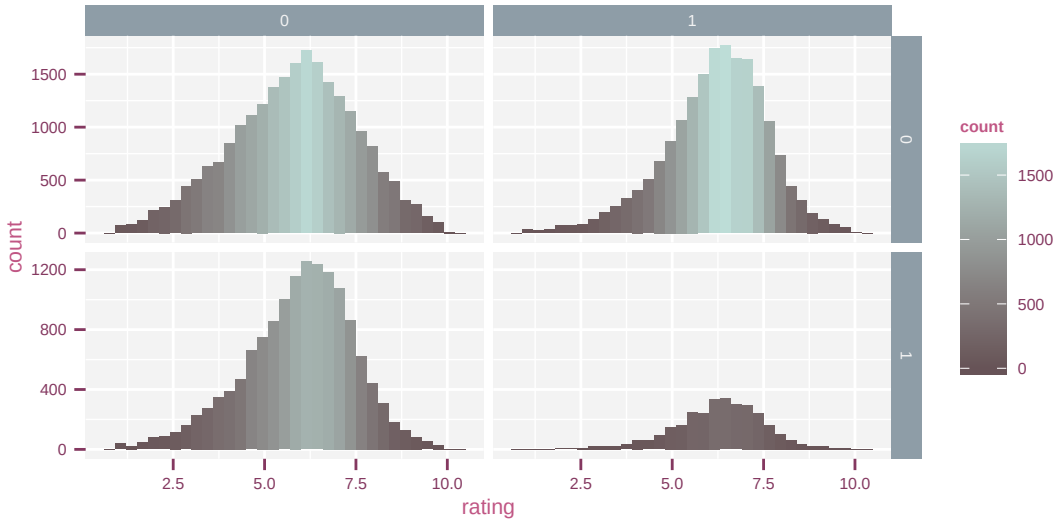
```
ggplot(movies, aes(x = rating)) +
  geom_histogram()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



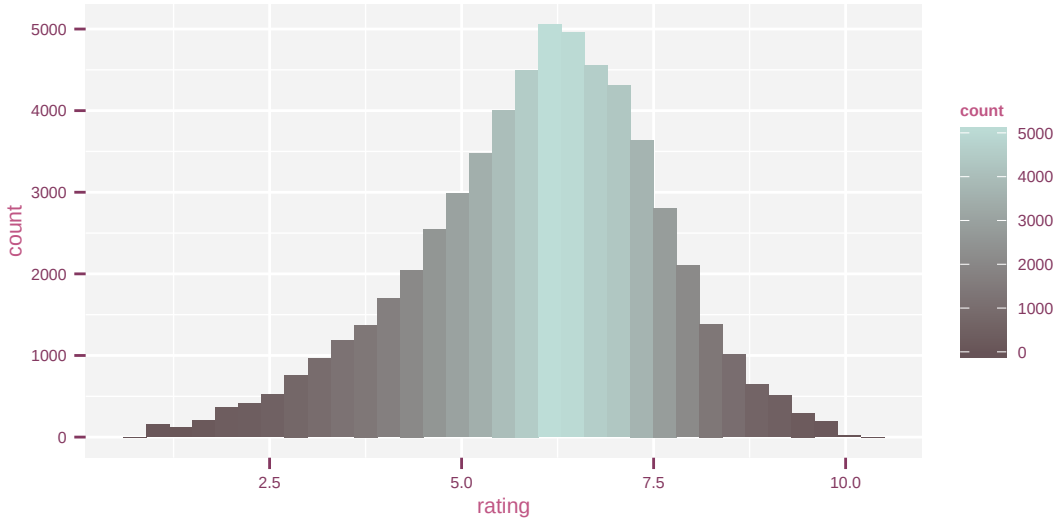

```
ggplot(movies, aes(x = rating)) +
  geom_histogram() +
  facet_wrap(~mpaa, scales = "free")

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## Warning:  position_stack requires constant width:  output may be incorrect
```



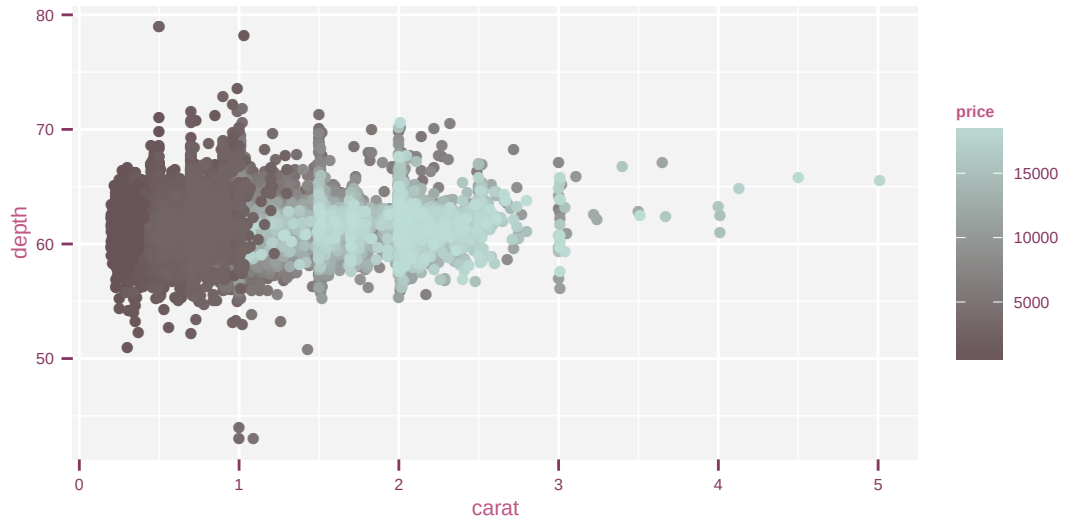
```
ggplot(movies, aes(x = rating)) +
  geom_histogram(aes(fill = ..count..)) +
  facet_grid(Comedy ~ Drama, scales = "free") +
  scale_fill_gradient()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



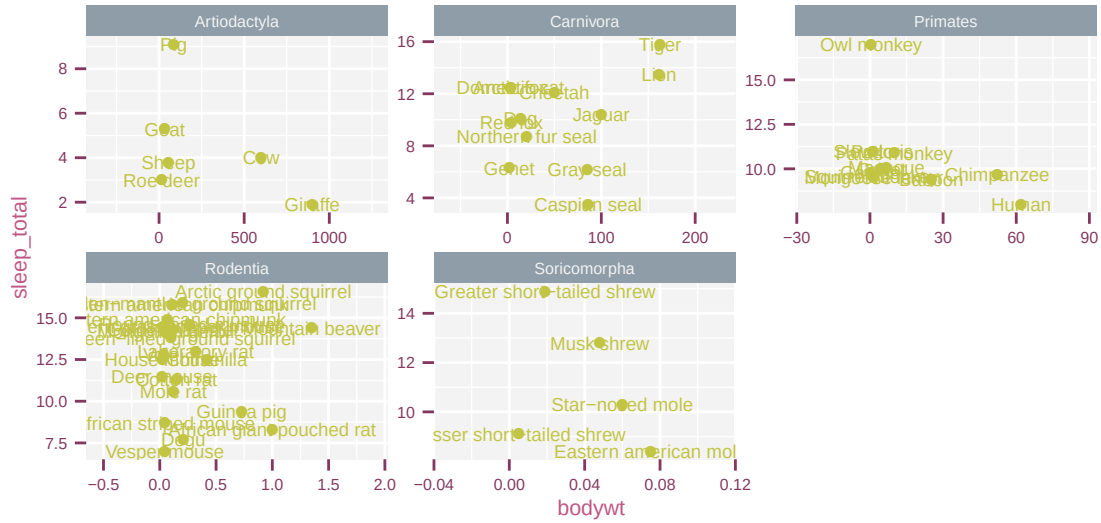

```
ggplot(movies, aes(x = rating)) +
  geom_histogram(aes(fill = ..count..)) +
  scale_fill_gradient()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



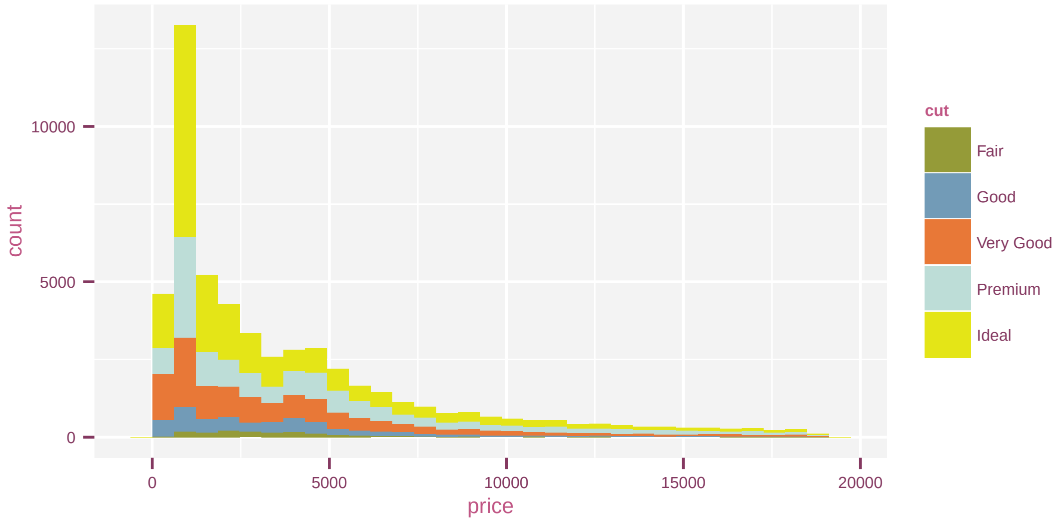

```
ggplot(diamonds, aes(x = carat, y = depth, colour = price)) +
  geom_point() +
  scale_colour_gradient()
```




```
selection <- table(msleep$order)
selection <- names(selection)[selection > 3]
ggplot(
  subset(msleep, order %in% selection),
  aes(x = bodywt, y = sleep_total, label = name)
) +
  geom_point() +
  geom_text() +
  facet_wrap(~order, scales = "free") +
  scale_x_continuous(expand = c(0.5, 0.01))
```

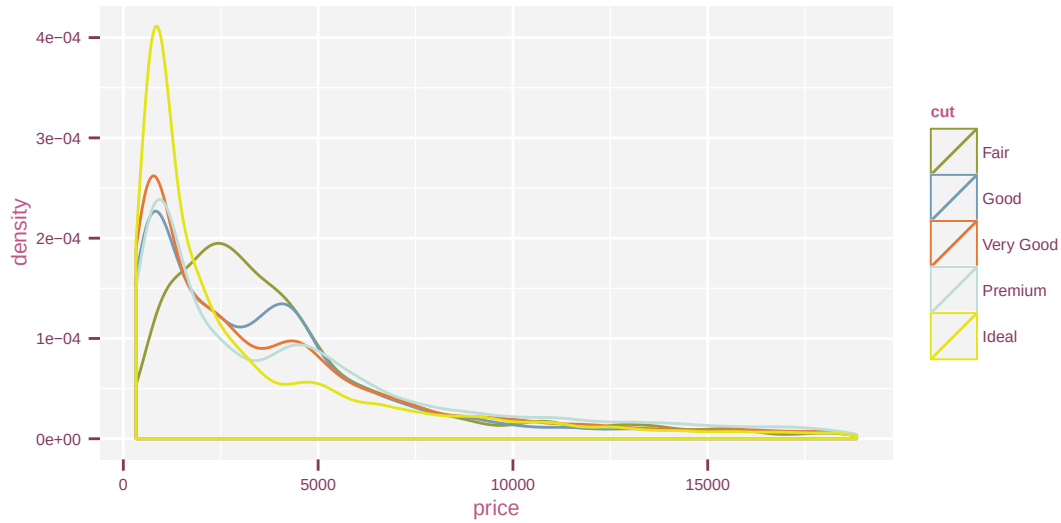



```
ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_histogram()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```

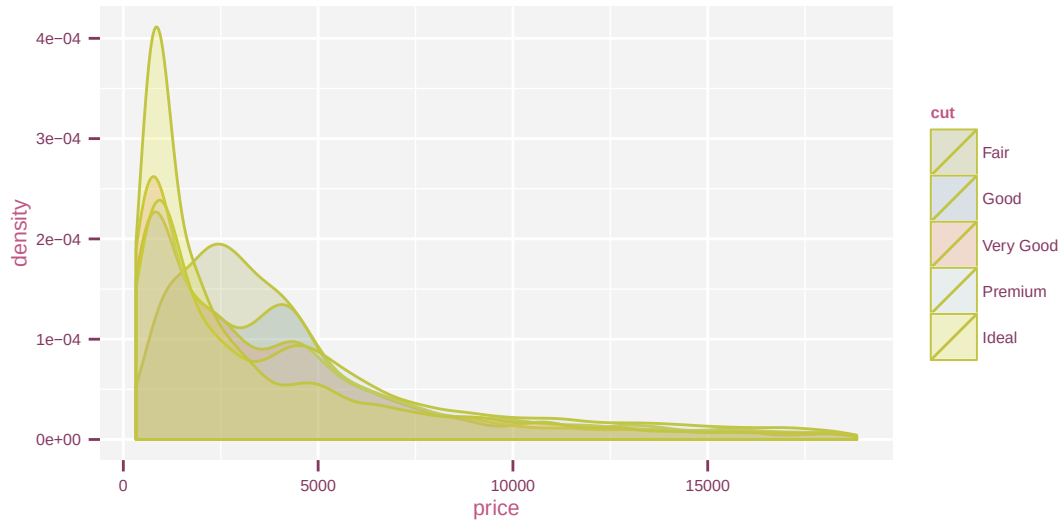



```
ggplot(diamonds, aes(x = log10(price), fill = cut)) +
  geom_histogram() +
  facet_wrap(~ clarity)

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```

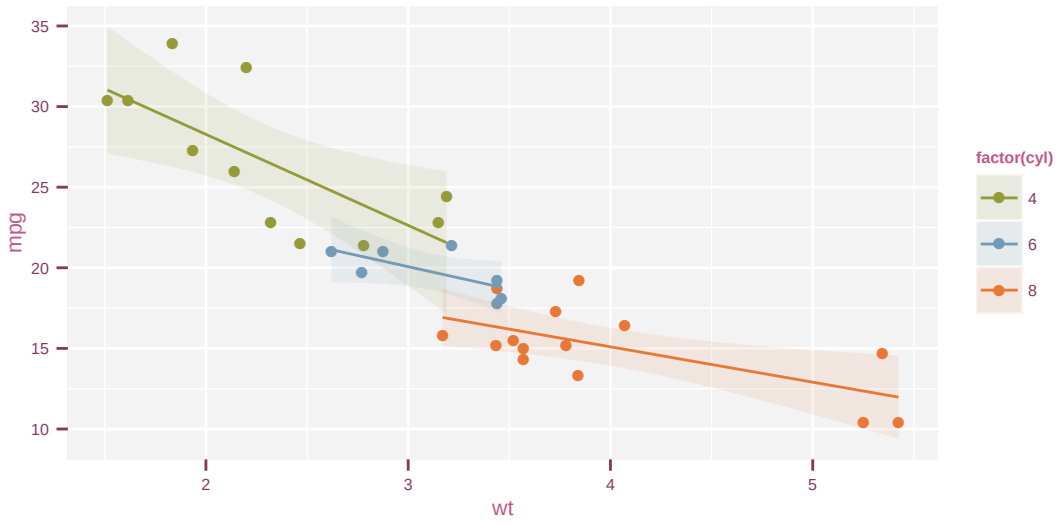

```
ggplot(diamonds, aes(x = price, colour = cut)) +
  geom_density()
```



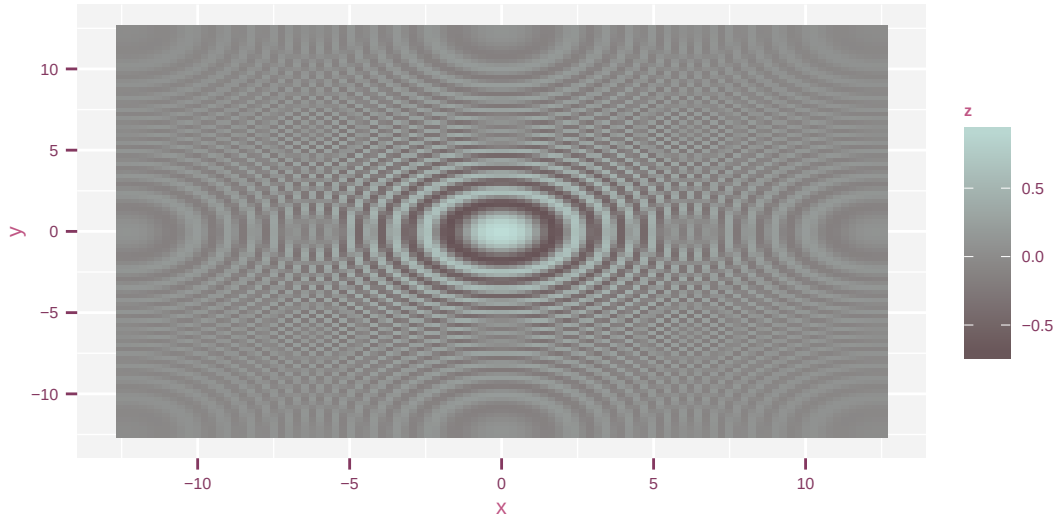

```
ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_density(alpha = 0.2)
```



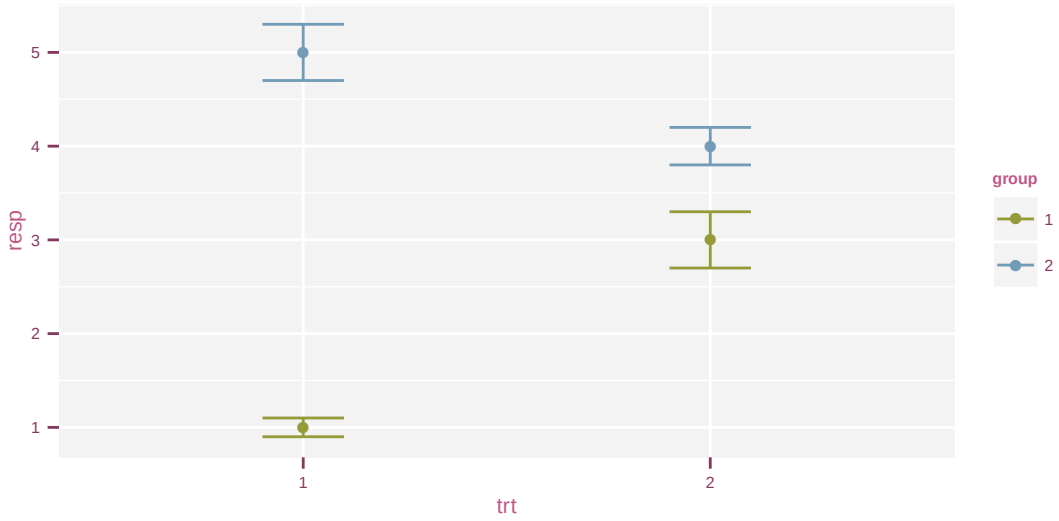

```
ggplot(mtcars, aes(x = wt, y = mpg, colour = factor(cyl), fill = factor(cyl))) +
  geom_point() +
  geom_smooth(method = "lm")
```




```
ggplot(pp(100), aes(x = x, y = y, fill = z)) +
  geom_tile() +
  scale_fill_gradient()
```



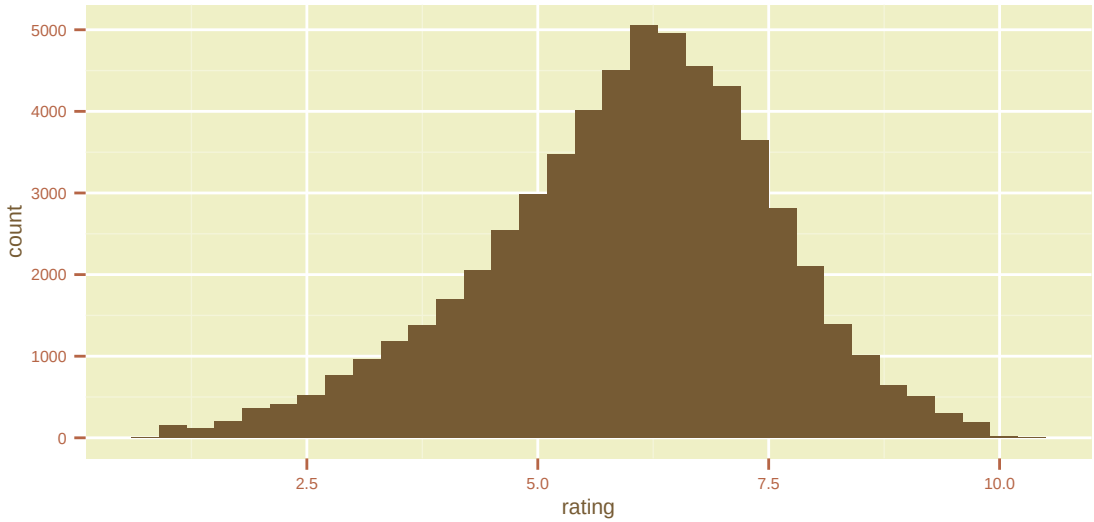

```
ggplot(
  df,
  aes(colour = group, y = resp, x = trt, ymax = resp + se, ymin = resp - se)
) +
  geom_point() +
  geom_errorbar(width = 0.2)
```



4 theme_INBO

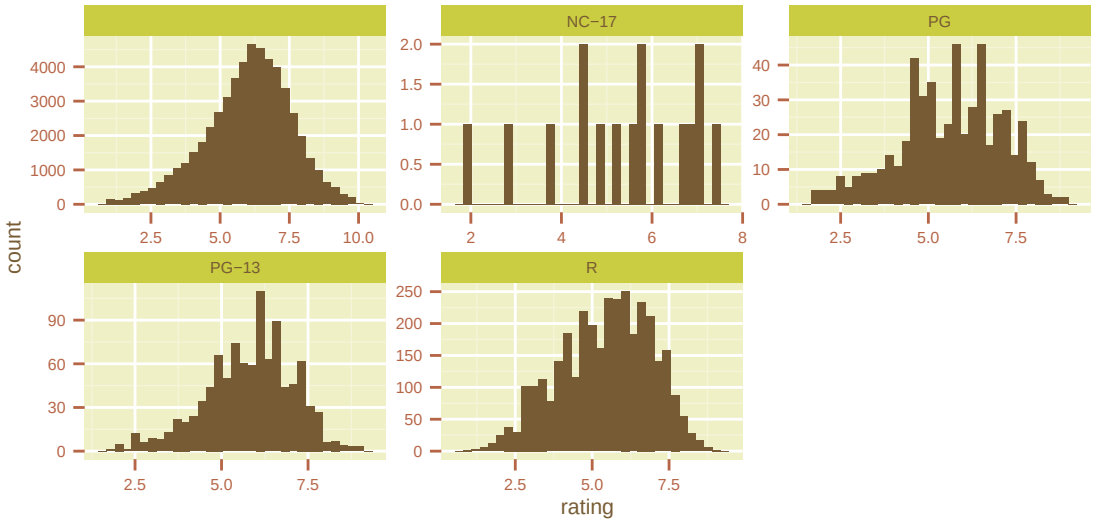
```
theme_set(theme_INBO(8))
switchColour(INBObrown)
```

```
ggplot(movies, aes(x = rating)) +
  geom_histogram()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



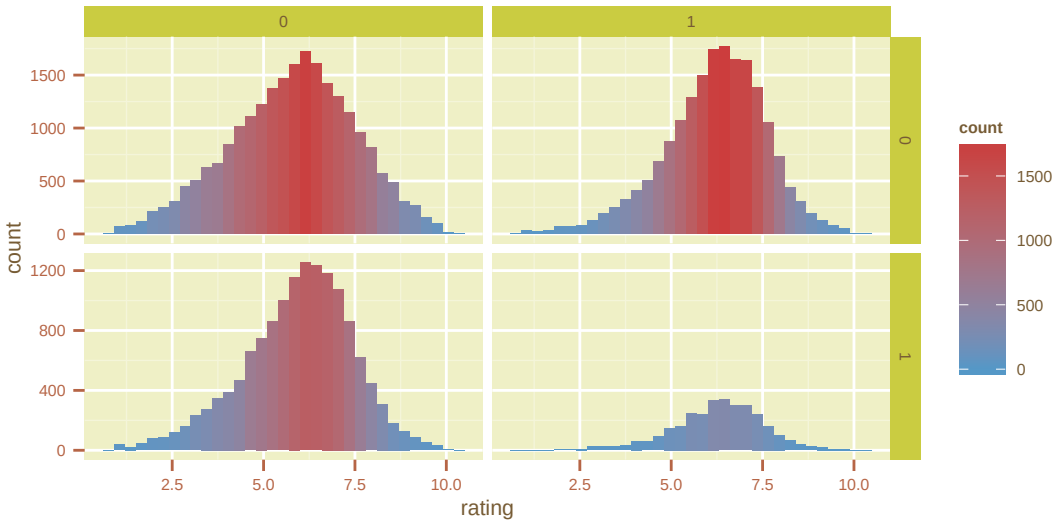

```
ggplot(movies, aes(x = rating)) +
  geom_histogram() +
  facet_wrap(~mpaa, scales = "free")

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```

```
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## Warning:  position_stack requires constant width:  output may be incorrect
```



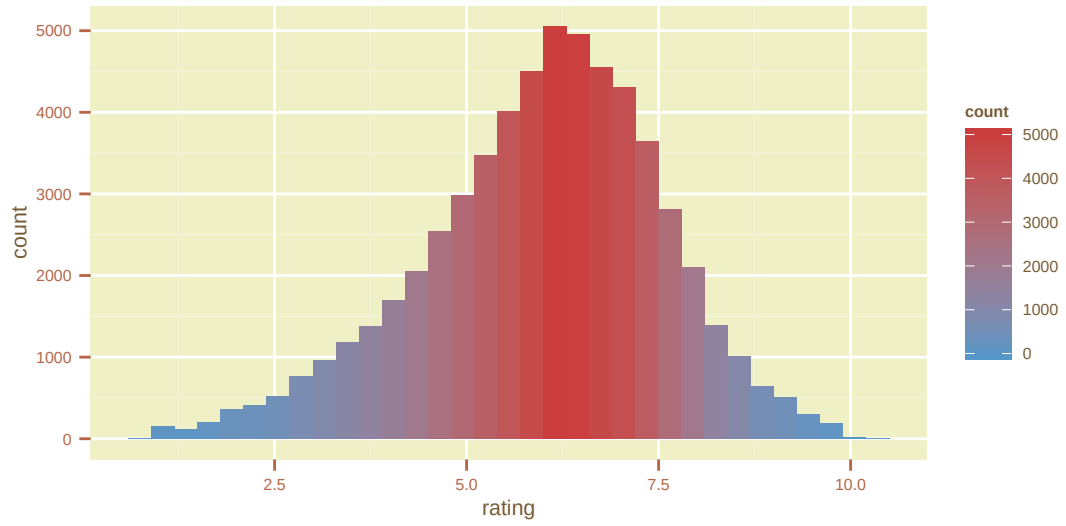

```
ggplot(movies, aes(x = rating)) +
  geom_histogram(aes(fill = ..count..)) +
  facet_grid(Comedy ~ Drama, scales = "free") +
  scale_fill_gradient()
```

```
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```

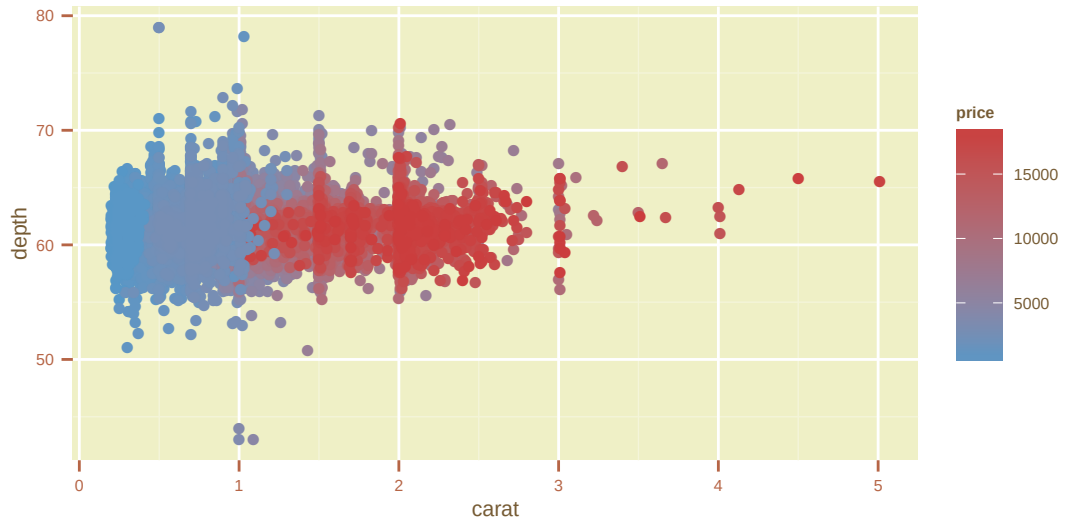



```
ggplot(movies, aes(x = rating)) +
  geom_histogram(aes(fill = ..count..)) +
  scale_fill_gradient()
```

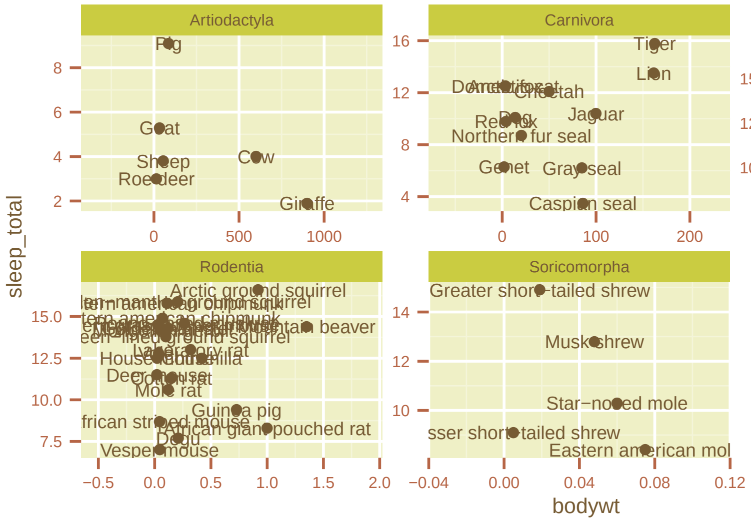
```
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



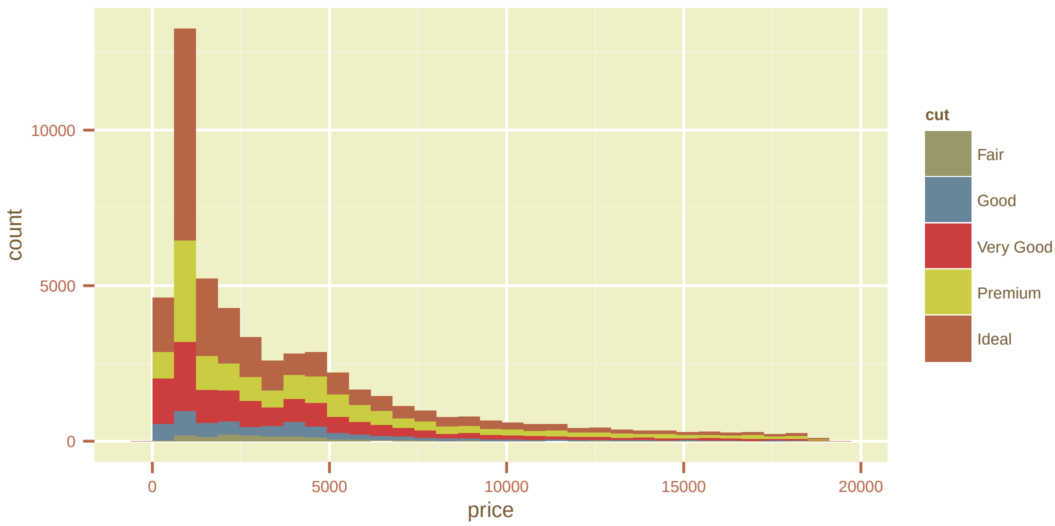

```
ggplot(diamonds, aes(x = carat, y = depth, colour = price)) +
  geom_point() +
  scale_colour_gradient()
```



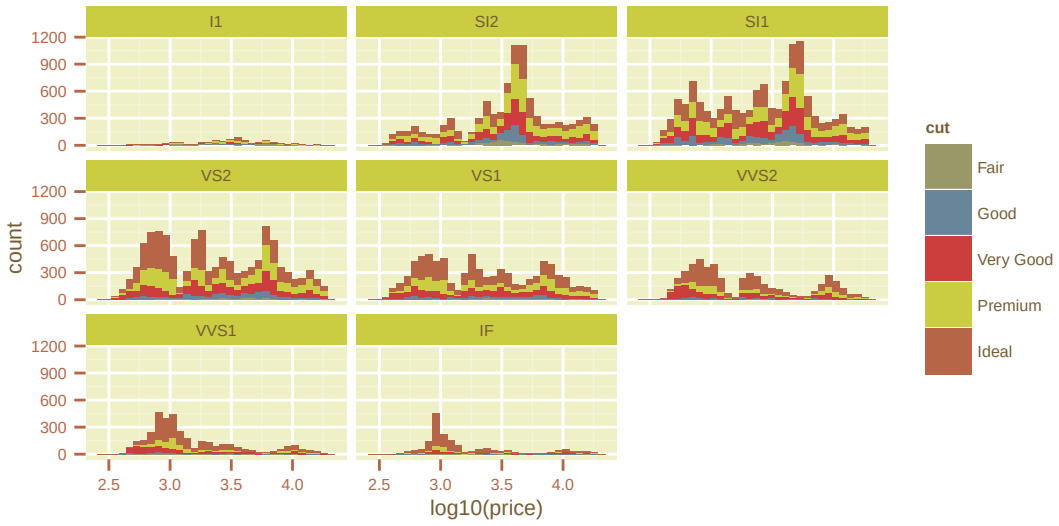

```
selection <- table(msleep$order)
selection <- names(selection)[selection > 3]
ggplot(
  subset(msleep, order %in% selection),
  aes(x = bodywt, y = sleep_total, label = name)
) +
  geom_point() +
  geom_text() +
  facet_wrap(~order, scales = "free") +
  scale_x_continuous(expand = c(0.5, 0.01))
```



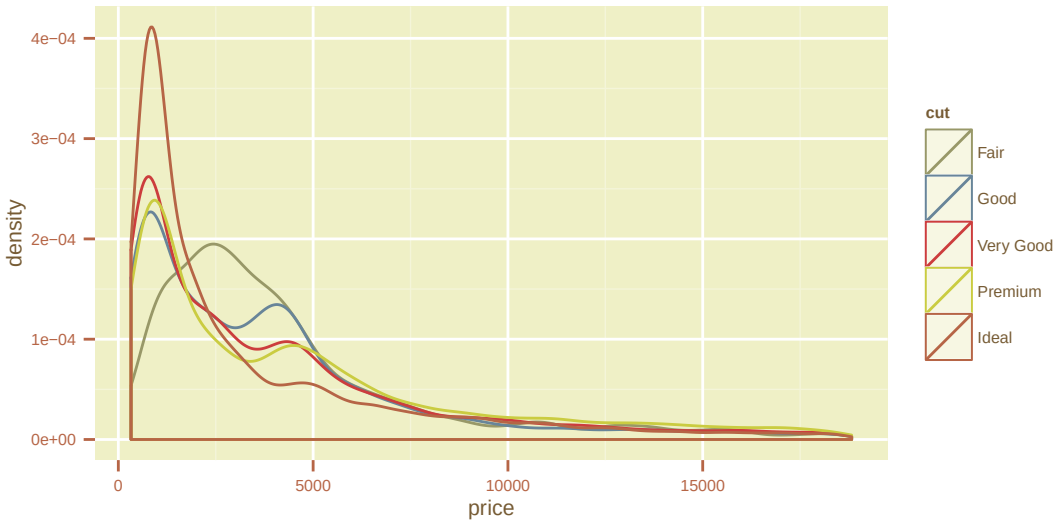

```r
ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_histogram()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



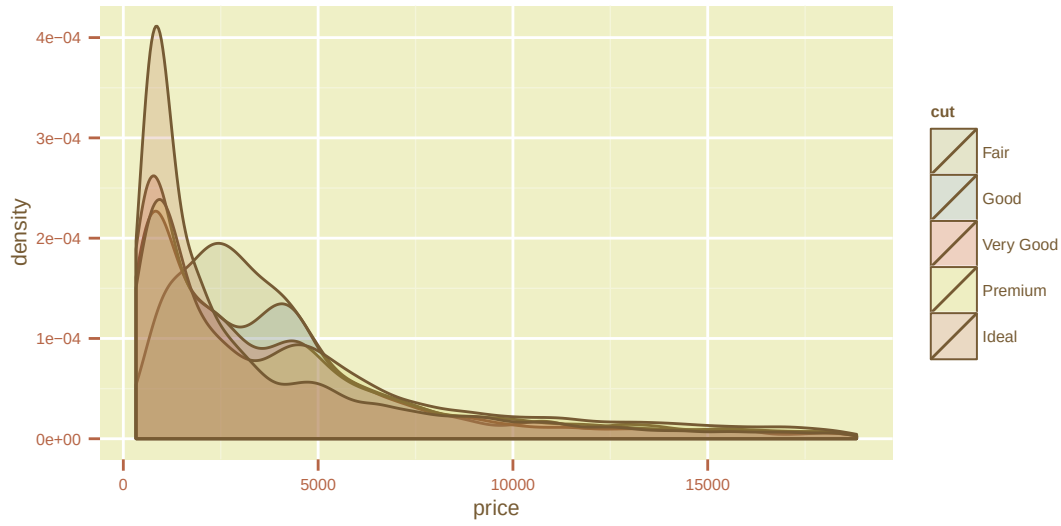

```r
ggplot(diamonds, aes(x = log10(price), fill = cut)) +
  geom_histogram() +
  facet_wrap(~ clarity)

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



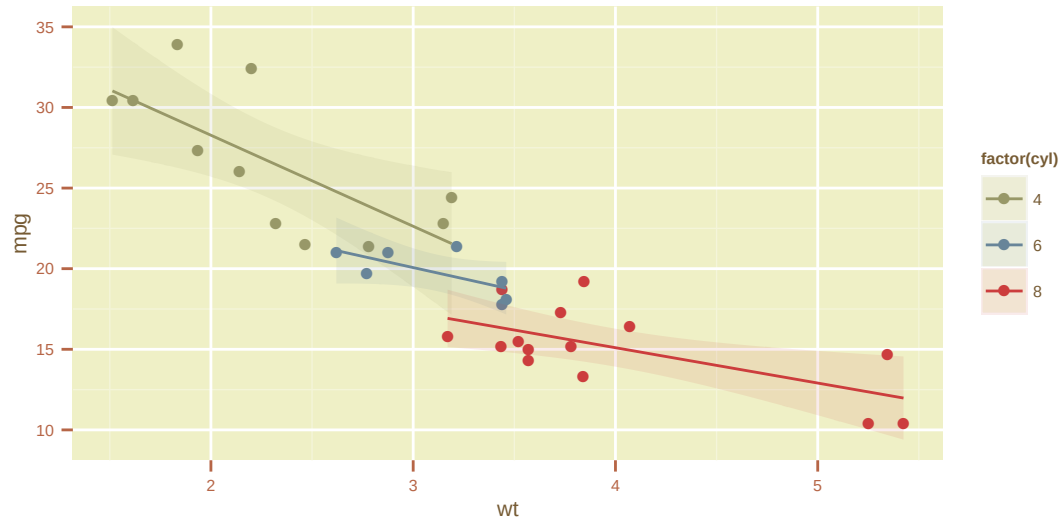

```
ggplot(diamonds, aes(x = price, colour = cut)) +
  geom_density()
```



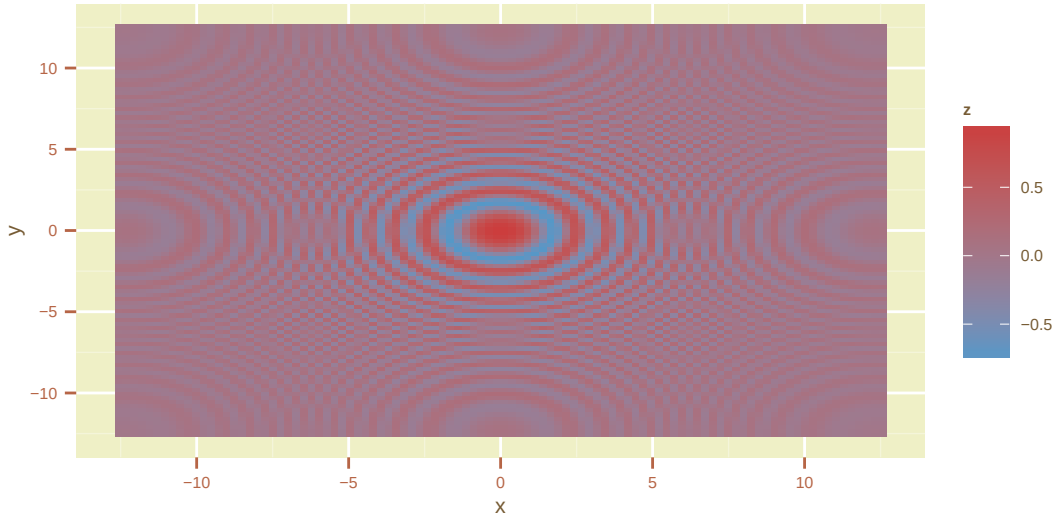

```
ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_density(alpha = 0.2)
```



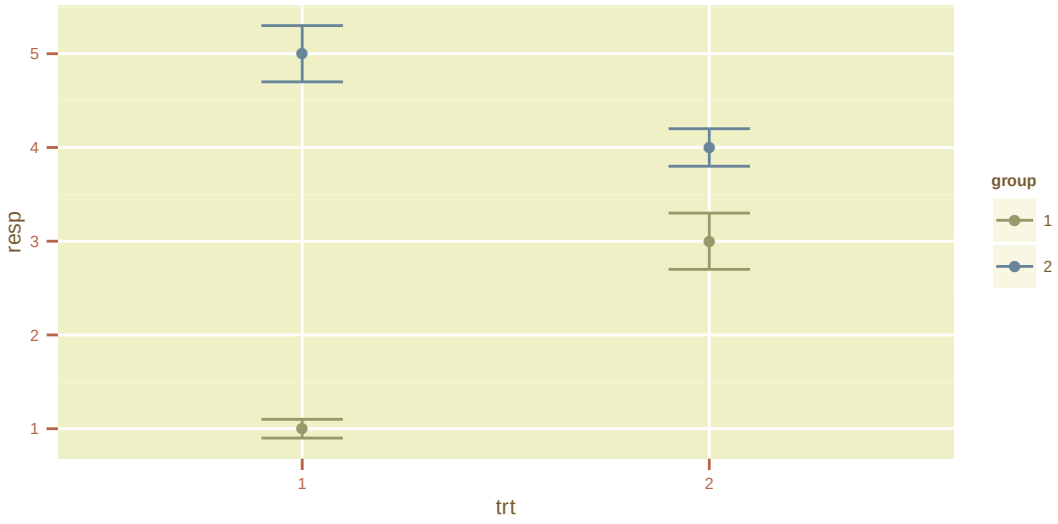

```
ggplot(mtcars, aes(x = wt, y = mpg, colour = factor(cyl), fill = factor(cyl))) +
  geom_point() +
  geom_smooth(method = "lm")
```




```
ggplot(pp(100), aes(x = x, y = y, fill = z)) +
  geom_tile() +
  scale_fill_gradient()
```



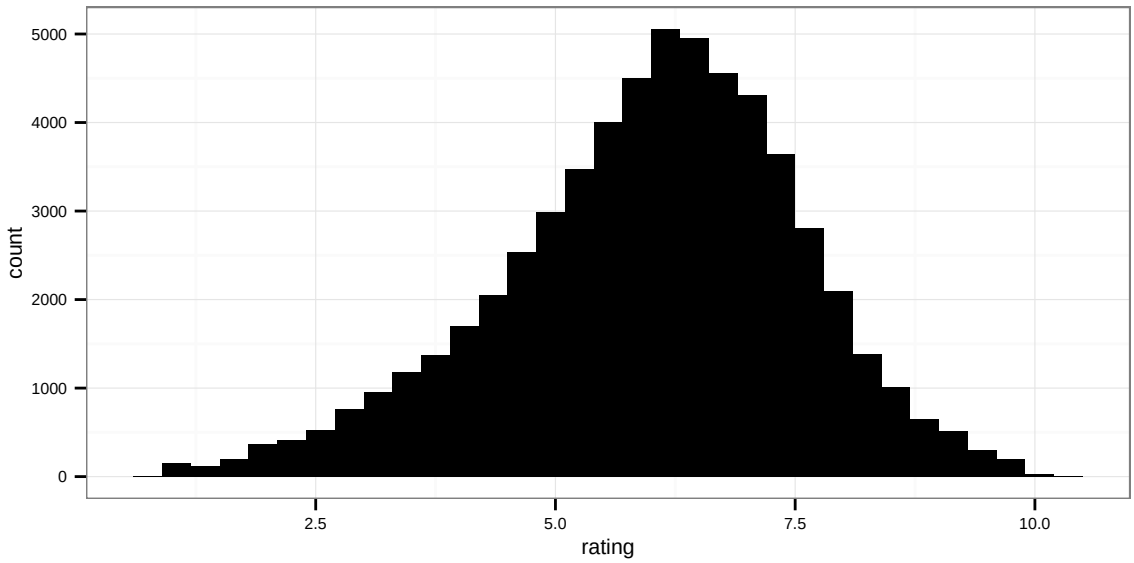

```
ggplot(
  df,
  aes(colour = group, y = resp, x = trt, ymax = resp + se, ymin = resp - se)
) +
  geom_point() +
  geom_errorbar(width = 0.2)
```



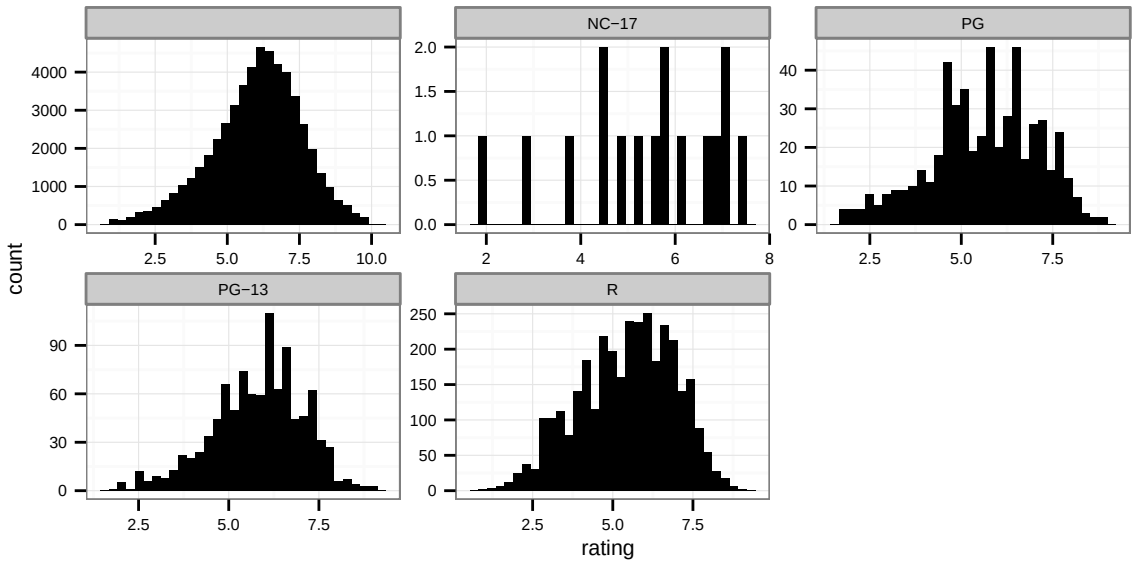
5 theme_elsevier

```
theme_set(theme_elsevier(8))
switchColour("black")
```

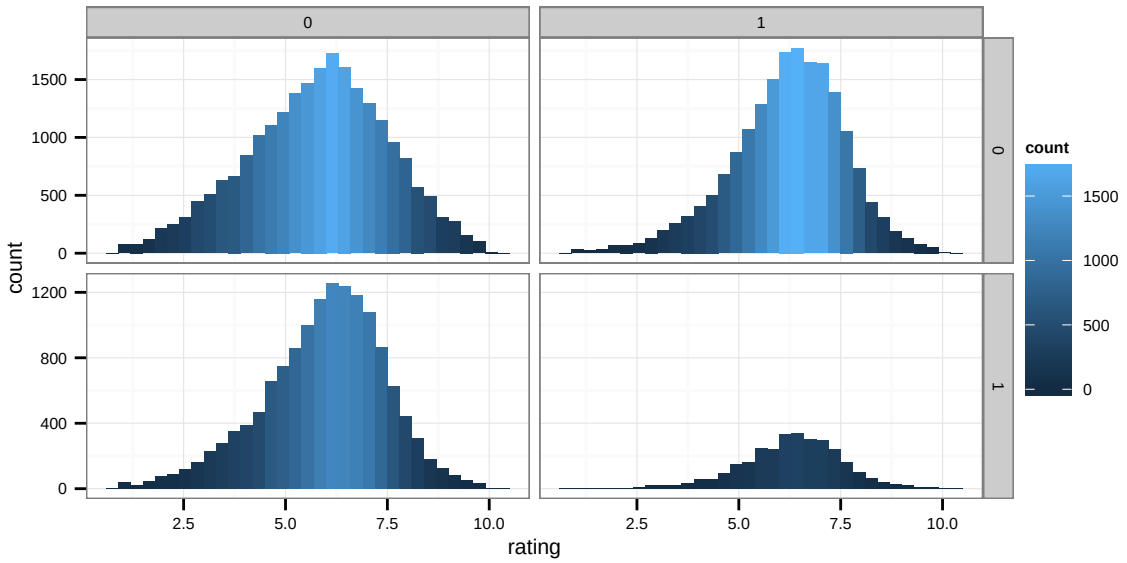
```
ggplot(movies, aes(x = rating)) +
  geom_histogram()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



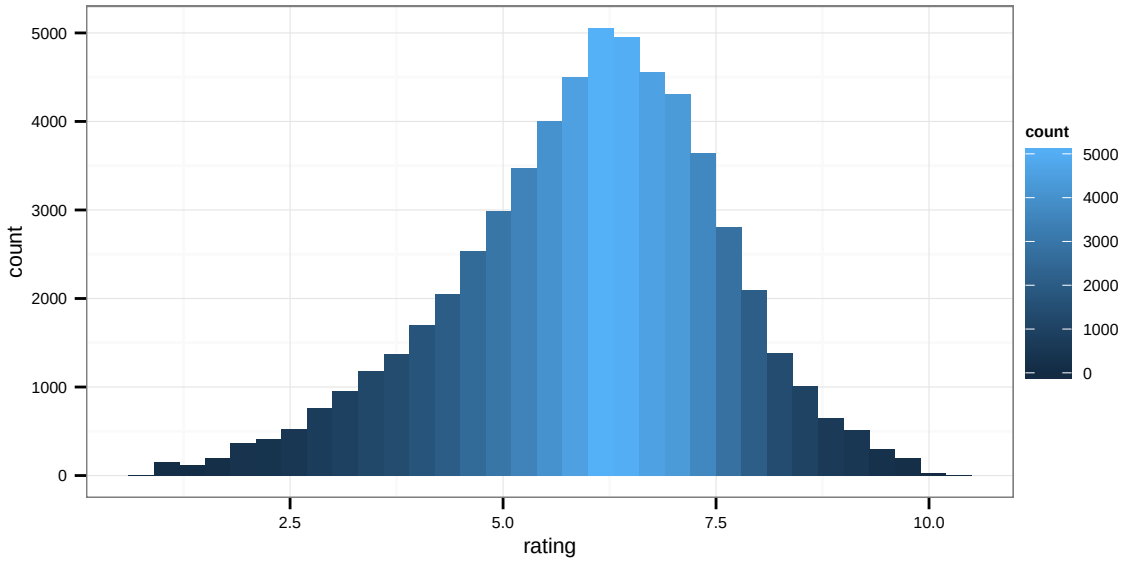

```
ggplot(movies, aes(x = rating)) +
  geom_histogram() +
  facet_wrap(~mpaa, scales = "free")

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## Warning:  position_stack requires constant width:  output may be incorrect
```



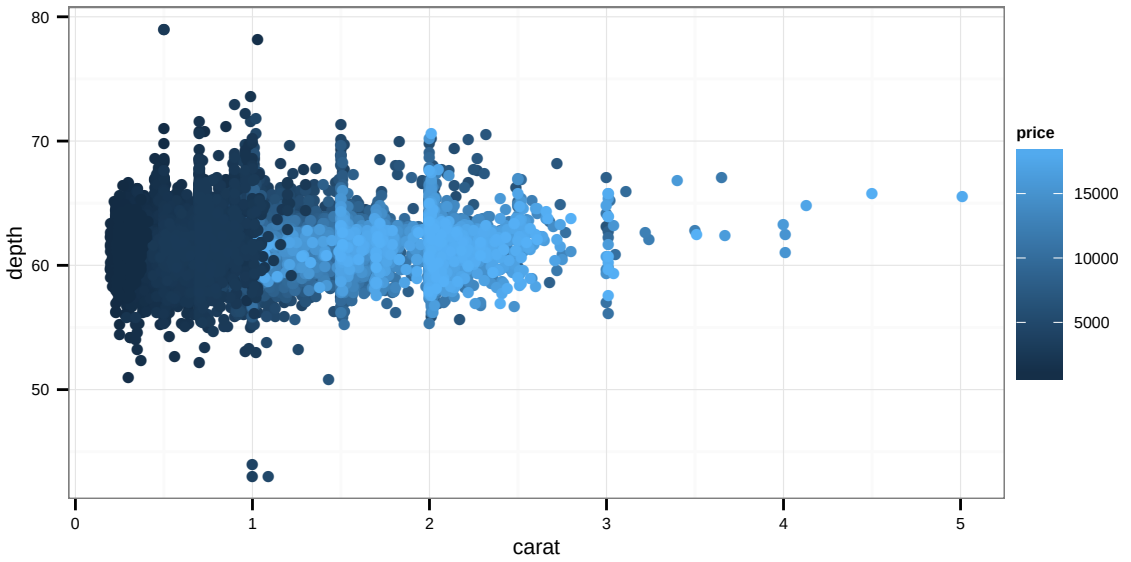

```
ggplot(movies, aes(x = rating)) +
  geom_histogram(aes(fill = ..count..)) +
  facet_grid(Comedy ~ Drama, scales = "free") +
  scale_fill_gradient()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



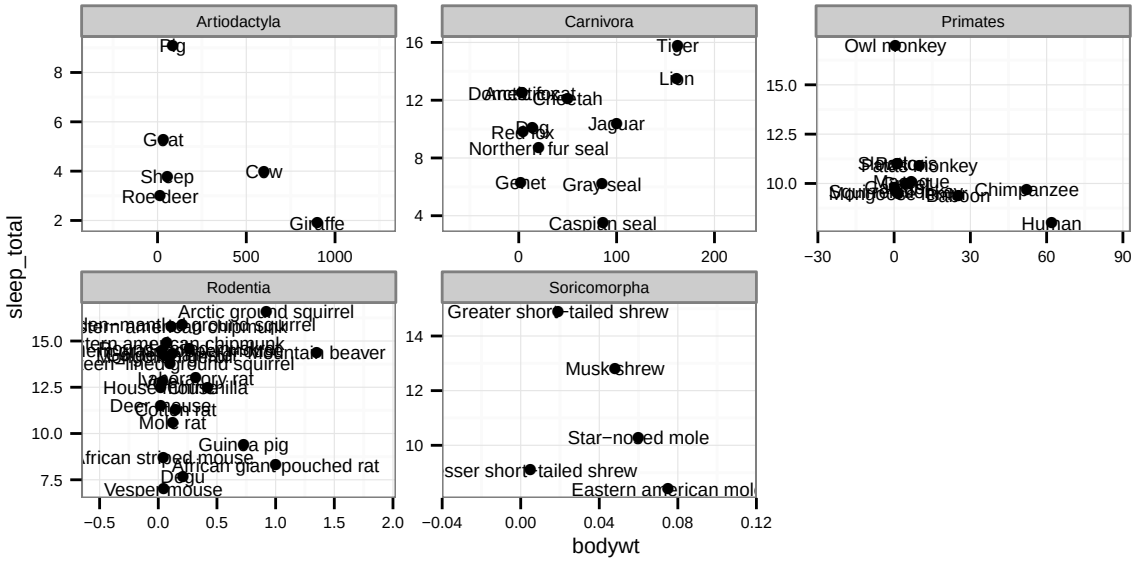

```
ggplot(movies, aes(x = rating)) +
  geom_histogram(aes(fill = ..count..)) +
  scale_fill_gradient()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



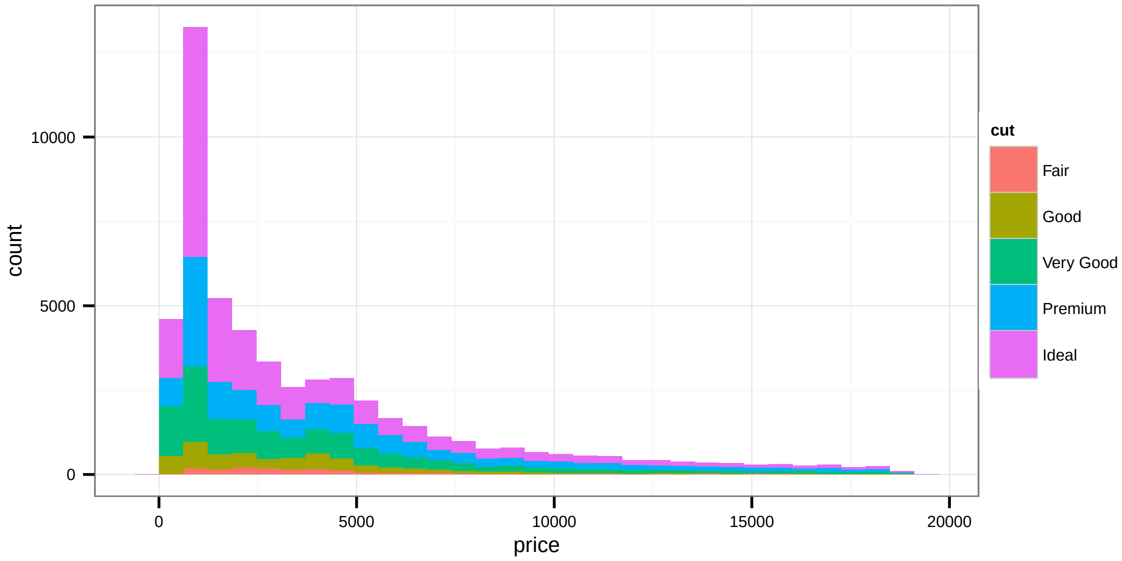

```
ggplot(diamonds, aes(x = carat, y = depth, colour = price)) +
  geom_point() +
  scale_colour_gradient()
```



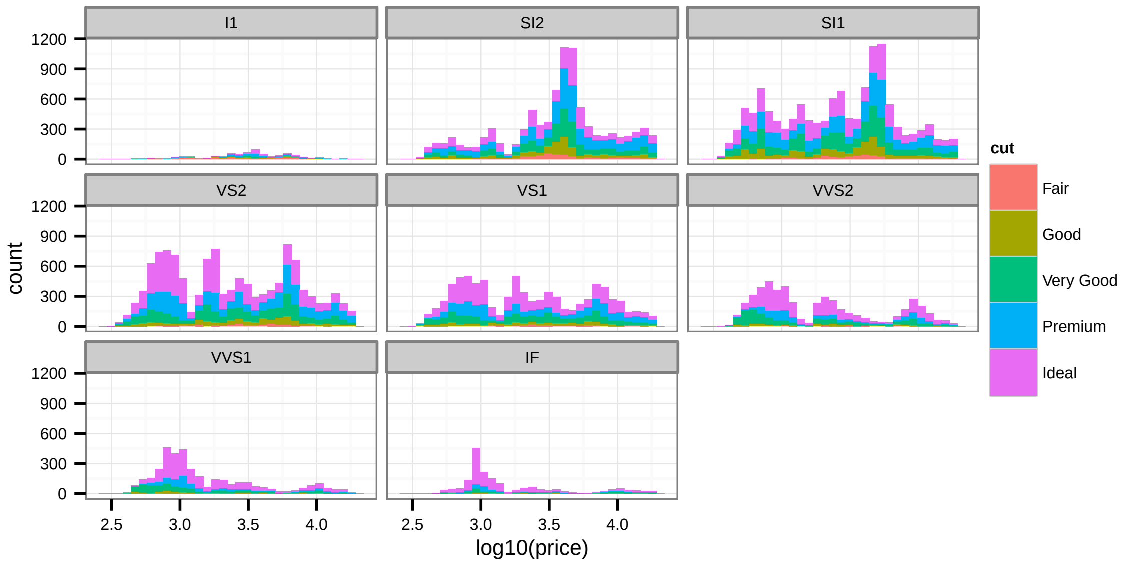

```
selection <- table(msleep$order)
selection <- names(selection)[selection > 3]
ggplot(
  subset(msleep, order %in% selection),
  aes(x = bodywt, y = sleep_total, label = name)
) +
  geom_point() +
  geom_text() +
  facet_wrap(~order, scales = "free") +
  scale_x_continuous(expand = c(0.5, 0.01))
```



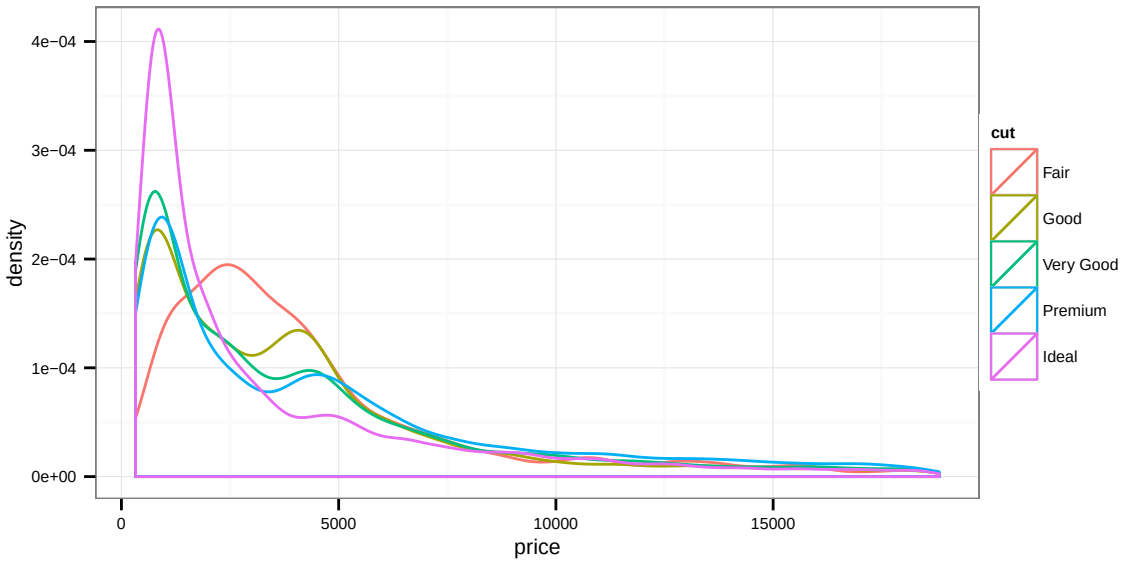

```
ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_histogram()

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



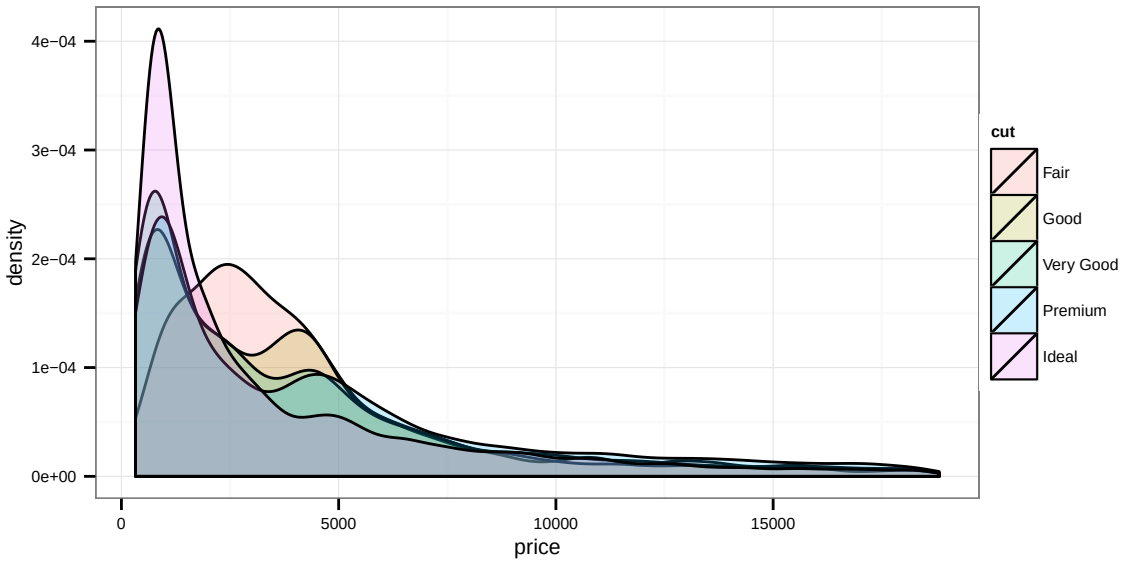

```
ggplot(diamonds, aes(x = log10(price), fill = cut)) +
  geom_histogram() +
  facet_wrap(~ clarity)

## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
## stat_bin:  binwidth defaulted to range/30.  Use 'binwidth = x' to adjust this.
```



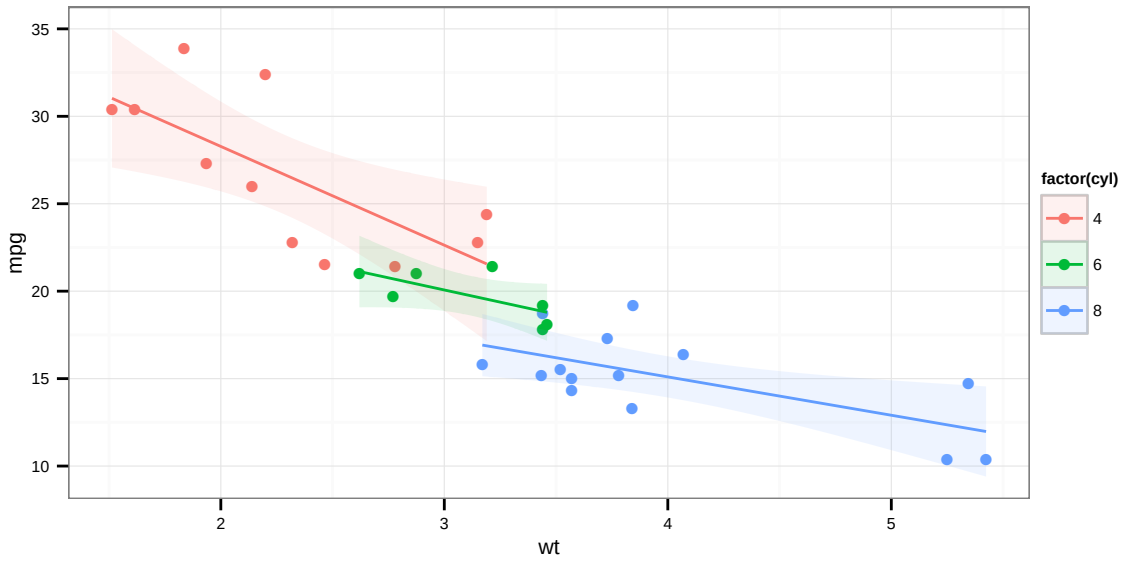

```
ggplot(diamonds, aes(x = price, colour = cut)) +
  geom_density()
```



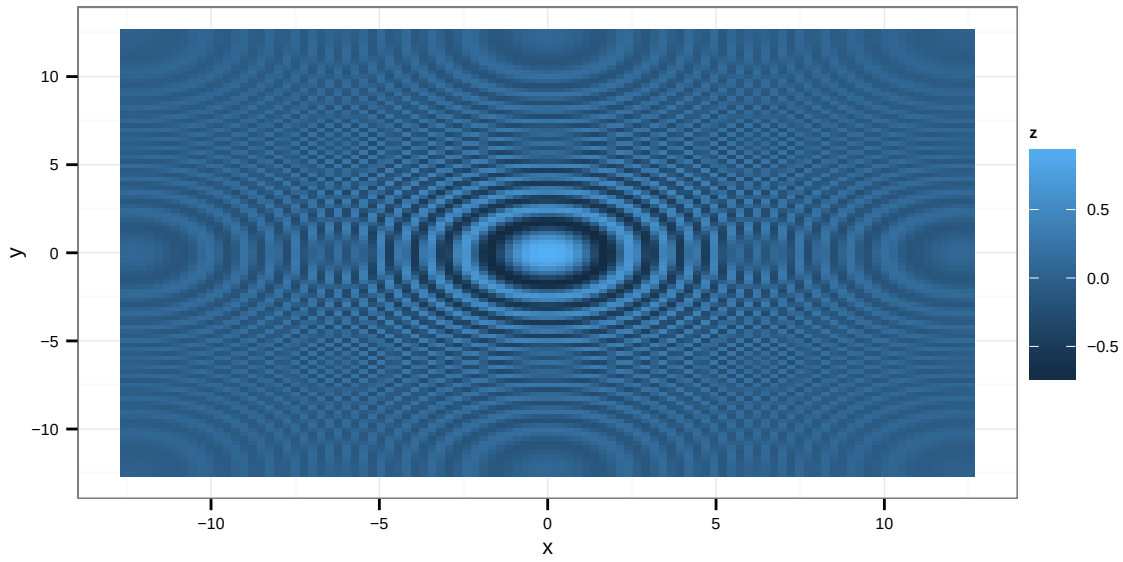

```
ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_density(alpha = 0.2)
```



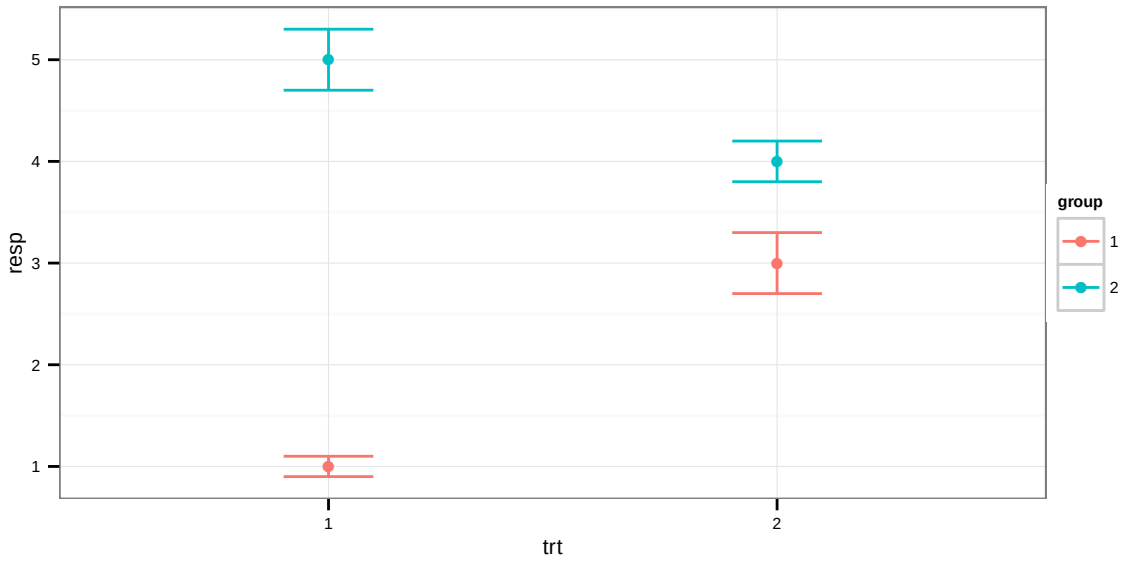

```
ggplot(mtcars, aes(x = wt, y = mpg, colour = factor(cyl), fill = factor(cyl))) +
  geom_point() +
  geom_smooth(method = "lm")
```




```
ggplot(pp(100), aes(x = x, y = y, fill = z)) +
  geom_tile() +
  scale_fill_gradient()
```




```
ggplot(
  df,
  aes(colour = group, y = resp, x = trt, ymax = resp + se, ymin = resp - se)
) +
  geom_point() +
  geom_errorbar(width = 0.2)
```


resp
5
4
3
2
1
1
2
trt
group
1
2