ggplot2: axis manipulation and themes

```
## knitr configuration: http://yihui.name/knitr
/options#chunk_options
opts_chunk$set(comment = "", error= TRUE, warning = FALSE,
message = FALSE,
               tidy = FALSE, cache = F, echo = T,
               fig.width = 6, fig.height = 6)

## R configuration
options(width = 116, scipen = 5)
```

References

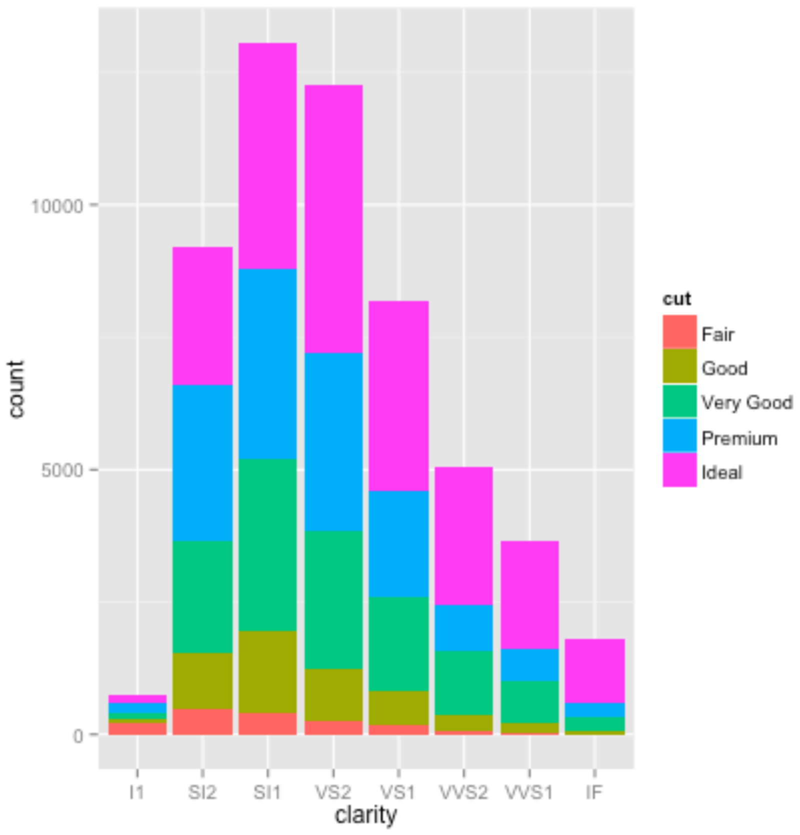
- ggplot2 book: http://ggplot2.org/book/
- Help topics: http://docs.ggplot2.org/current/
- http://wiki.stdout.org/rcookbook/Graphs/Axes%20(ggplot2)/

Load ggplot2

```
library(ggplot2)
```

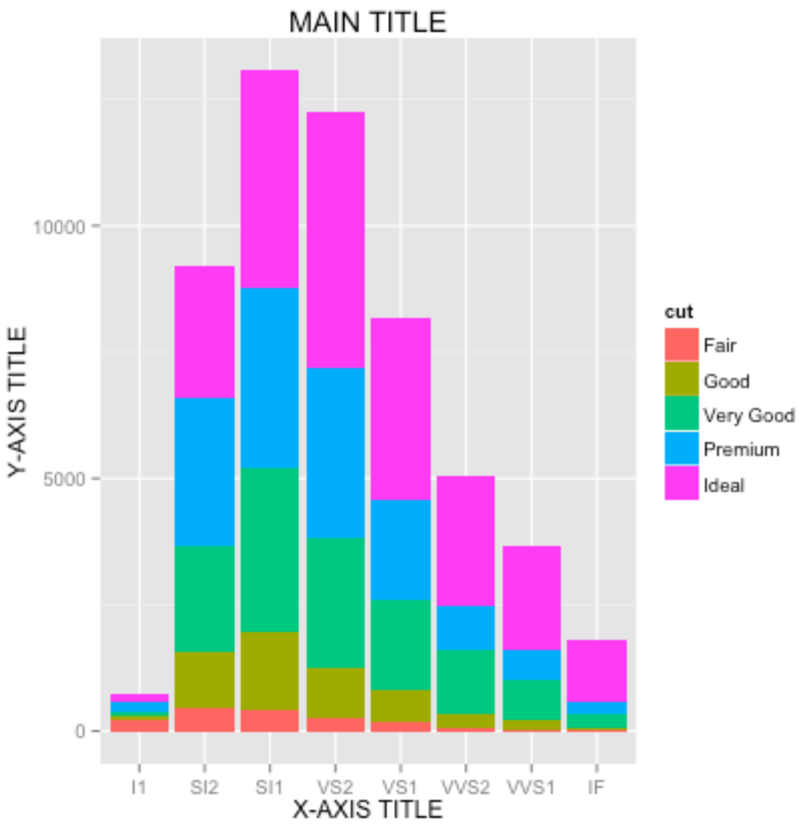
Create plot

```
data(diamonds)
p.dia <- ggplot(data = diamonds, mapping = aes(x = clarity))

p <- p.dia + layer(geom = "bar", mapping = aes(fill = cut))
p
```



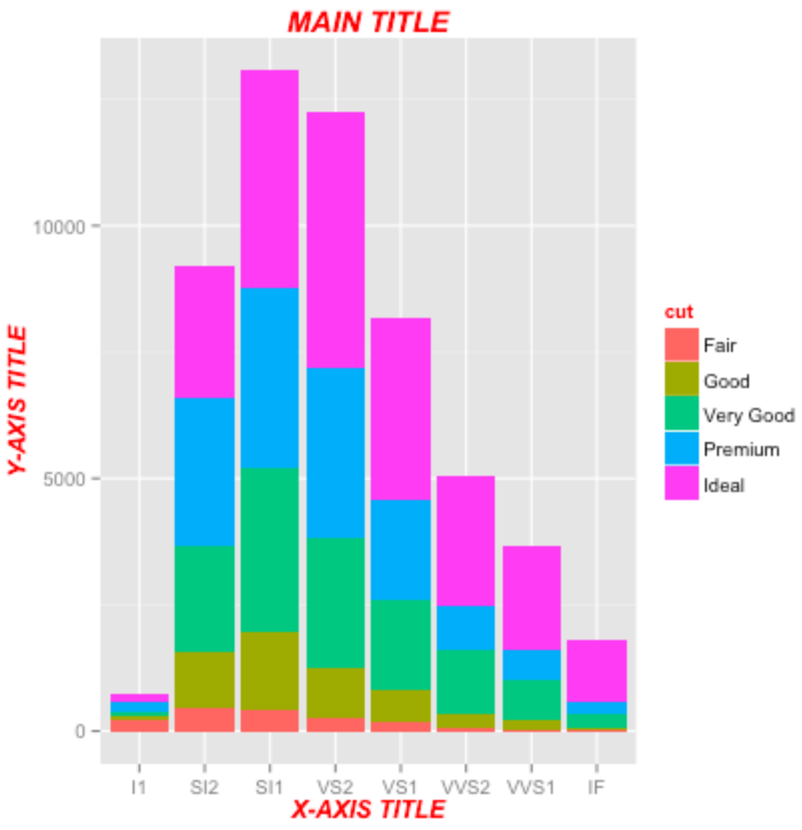
Change title, X axis label, and Y axis label

```
p.labs <- p + labs(title = "MAIN TITLE", x = "X-AXIS TITLE", y
= "Y-AXIS TITLE")
p.labs
```



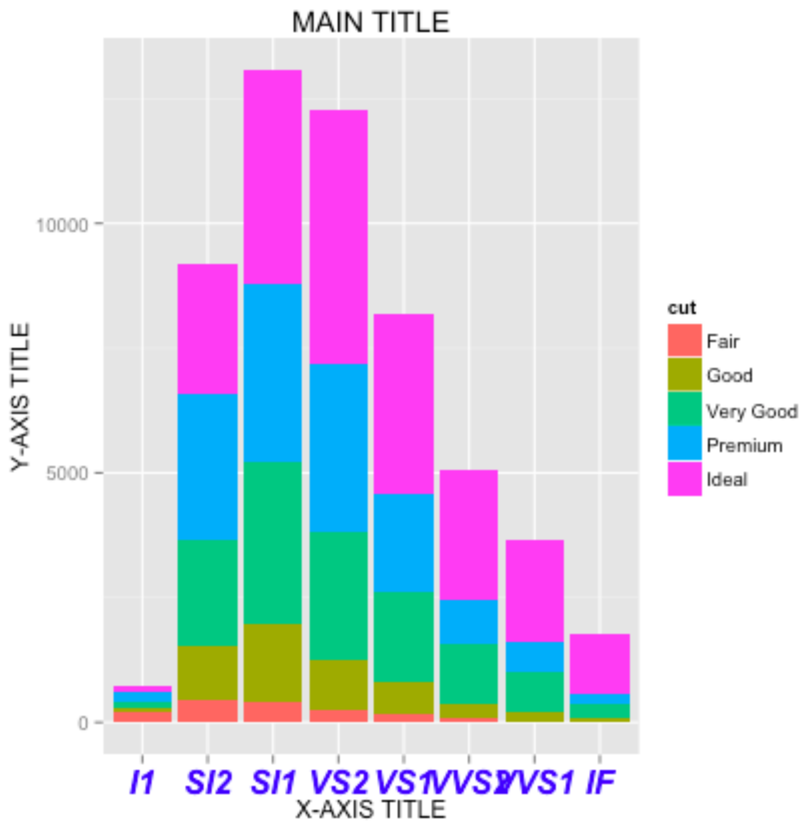
Change text style in title and X/Y axis labels

```
red.bold.italic.text <- element_text(face = "bold.italic",
color = "red")

p.labs + theme(title = red.bold.italic.text, axis.title =
red.bold.italic.text)
```

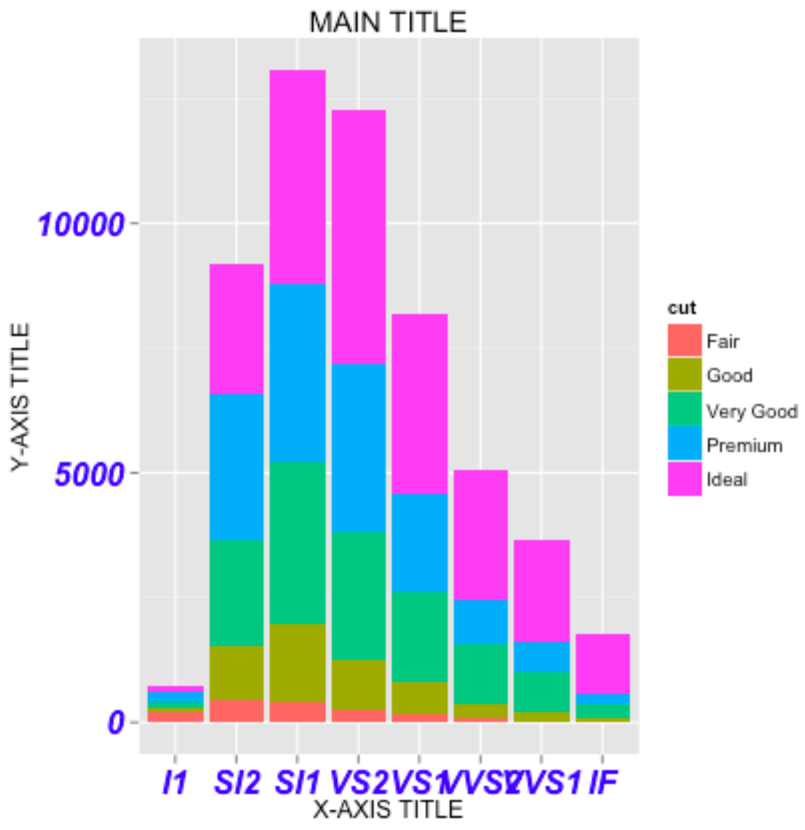


Change axis text style

```
blue.bold.italic.16.text <- element_text(face = "bold.italic",
color = "blue", size = 16)

## axis.text.x for x axis only
p.labs + theme(axis.text.x = blue.bold.italic.16.text)
```



```
## axis.text for both axes
p.labs + theme(axis.text = blue.bold.italic.16.text)
```

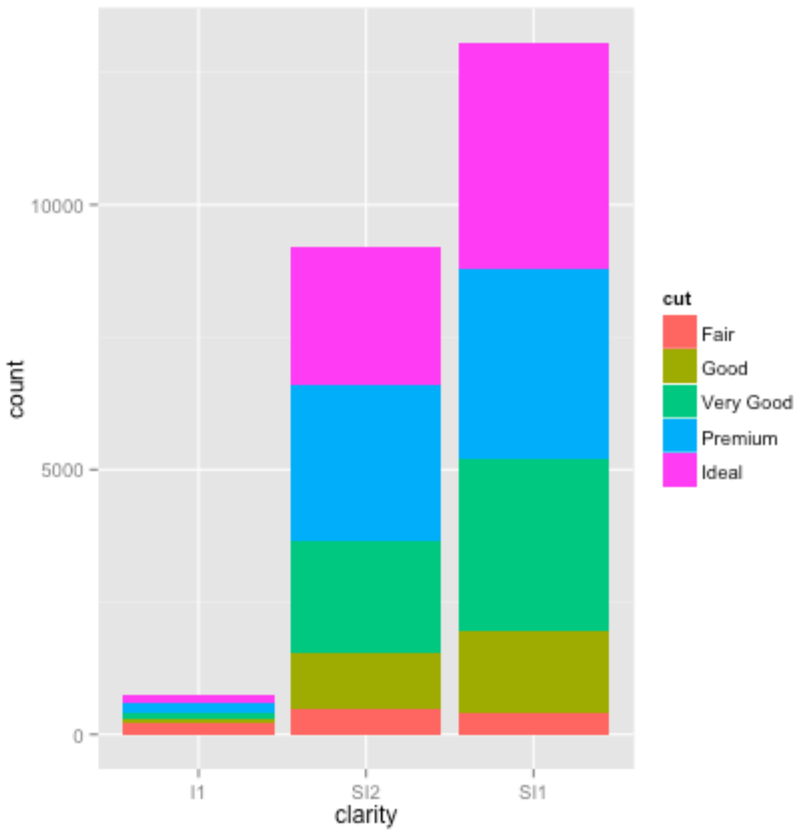


element_text() options

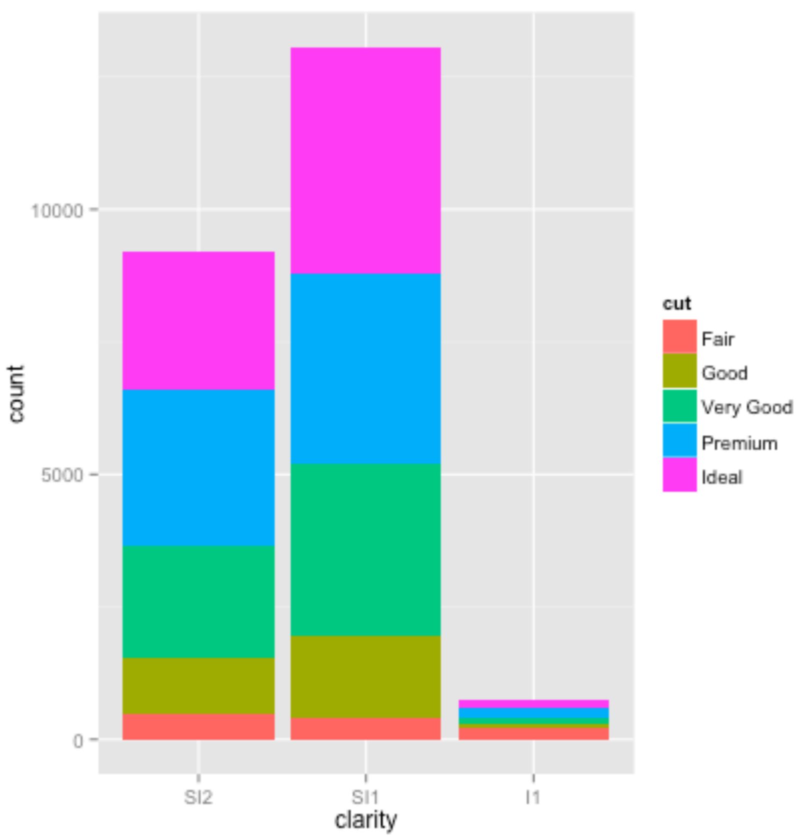
```
  family: font family
    face: font face ("plain", "italic", "bold", "bold.italic")
  colour: text colour
    size: text size (in pts)
   hjust: horizontal justification (in [0, 1])
   vjust: vertical justification (in [0, 1])
   angle: angle (in [0, 360])
lineheight: line height
   color: an alias for ‘colour’
```

Manipulate discrete scale

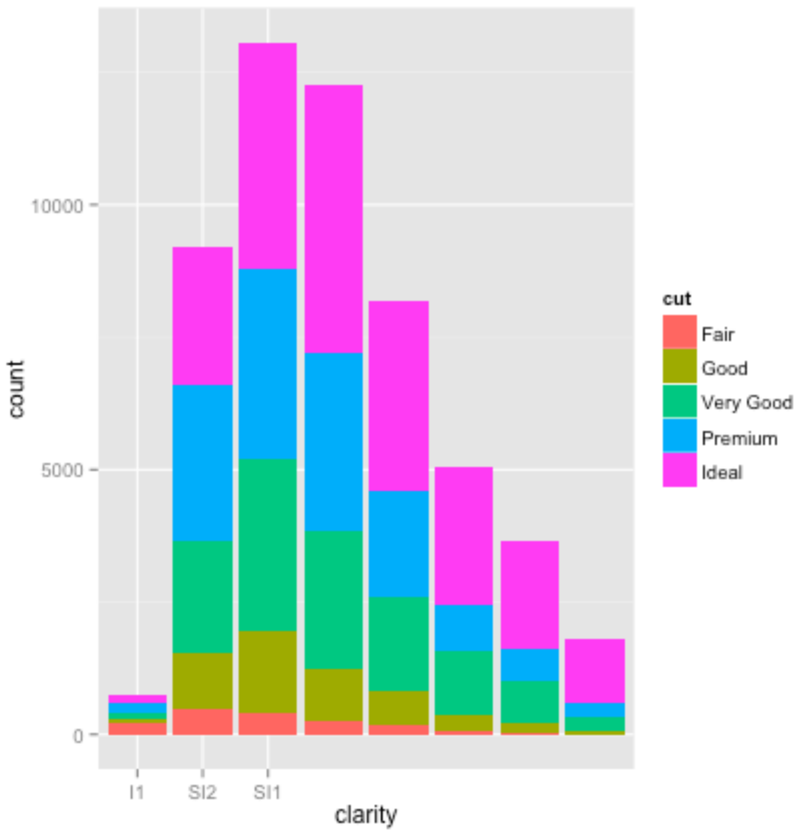
```
## Only show I1, SI2, SI1
p + scale_x_discrete(limit = c("I1", "SI2", "SI1"))
```



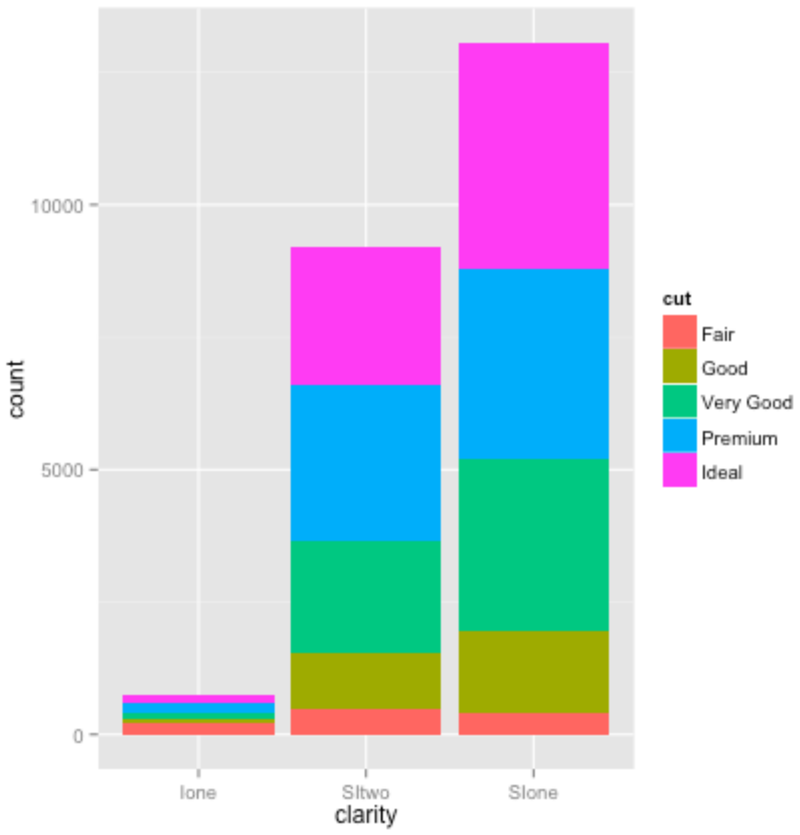
```
## Reorder: SI2, SI1, I1
p + scale_x_discrete(limit = c("SI2", "SI1", "I1"))
```



```
## Same thing with breaks will erase breaks at other points
p + scale_x_discrete(breaks = c("I1", "SI2", "SI1"))
```



```
##
p + scale_x_discrete(limit = c("I1", "SI2", "SI1"),
                     labels = c("Ione","SItwo","SIone"))
```



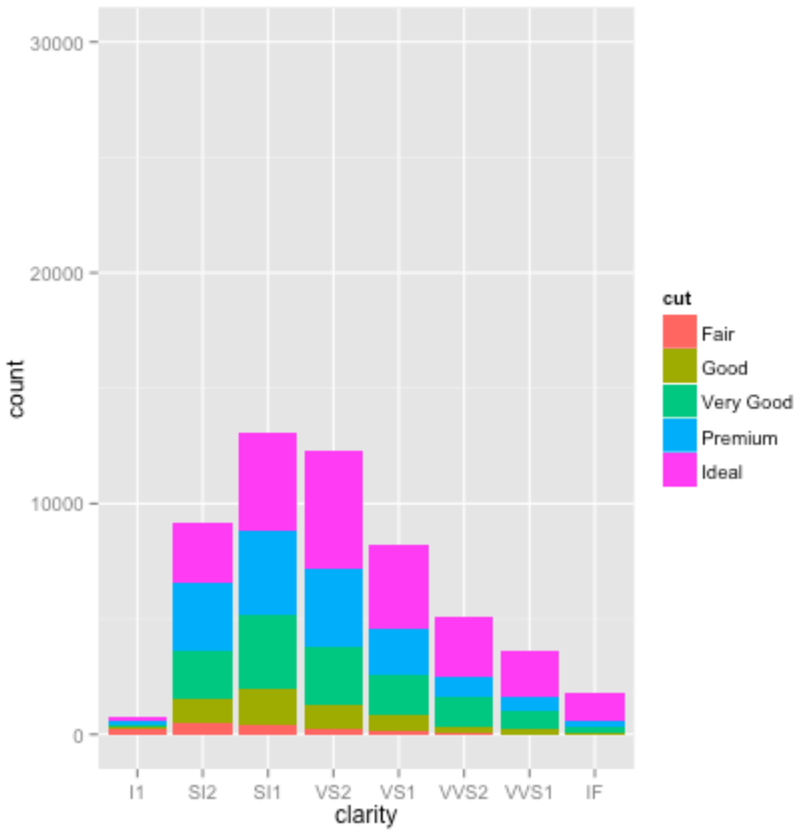
Manipulate continuous scale

Options for continuous scales

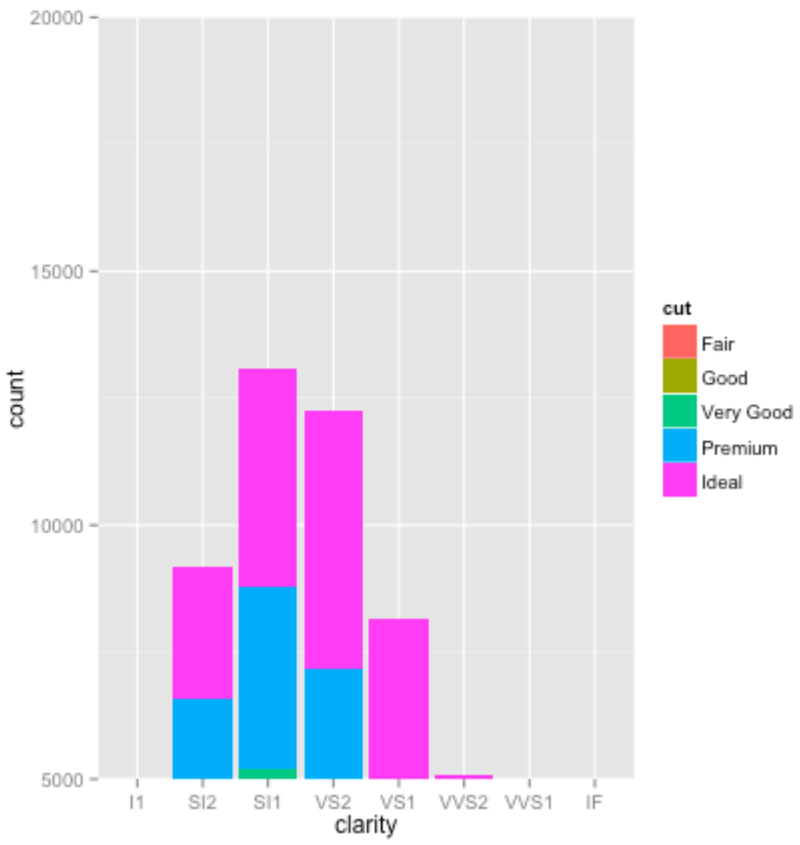
```
     ...: common continuous scale parameters: ‘name’, ‘breaks’,
          ‘labels’, ‘na.value’, ‘limits’ and ‘trans’.  See
          ‘continuous_scale’ for more details

  expand: a numeric vector of length two giving multiplicative
and
          additive expansion constants. These constants ensure
that the
          data is placed some distance away from the axes.
```

Change range

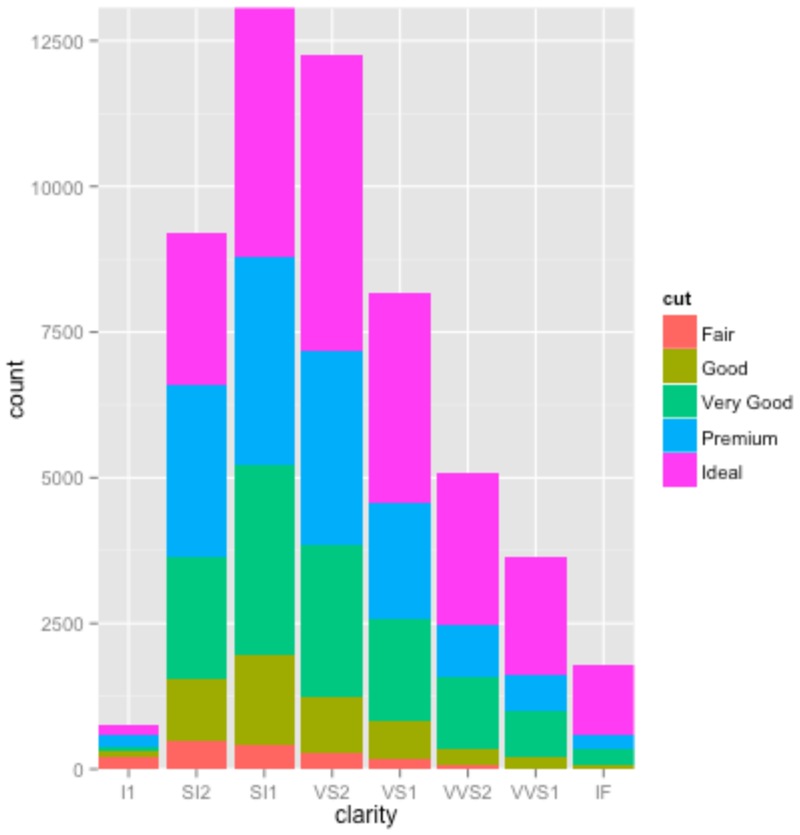
```
## Change range of Y axis
p + scale_y_continuous(limit = c(0, 30000))
```



```
## Use coord_cartesian(ylim) to zoom in
p + coord_cartesian(ylim = c(5000, 20000))
```



```
## No extra space around plot
p + scale_y_continuous(expand = c(0,0)) +
scale_x_discrete(expand = c(0,0))
```



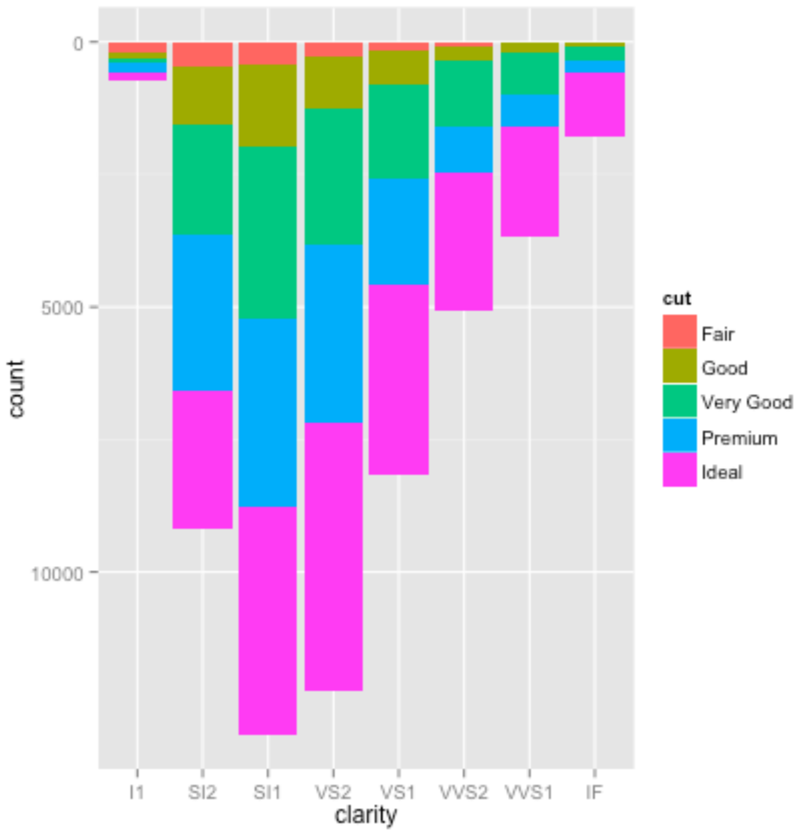
Setting limits on a scale vs coordinate system

```
     The Cartesian coordinate system is the most familiar, and
common,
     type of coordinate system. Setting limits on the
coordinate system
     will zoom the plot (like you're looking at it with a
magnifying
     glass), and will not change the underlying data like
setting
     limits on a scale will.

       coord_cartesian(xlim = NULL, ylim = NULL, wise = NULL)
```

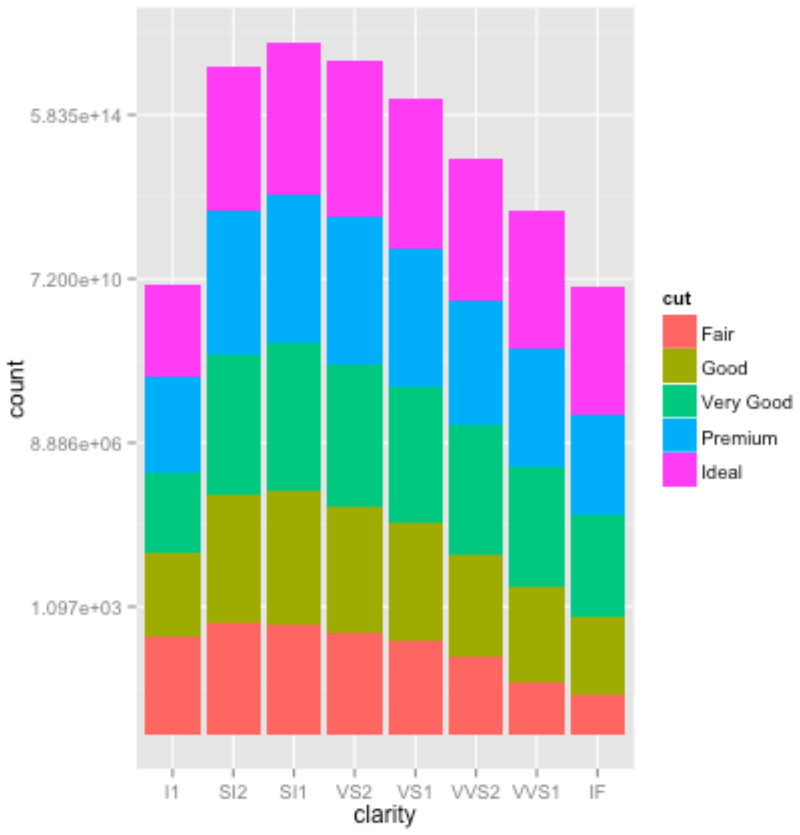
Transformation

Available functions: asn, exp, identitiy, log, log10, log2, logit, pow10, probit, recip, reverse, sqrt

```
## Reversal
p + scale_y_continuous(trans = "reverse")
```

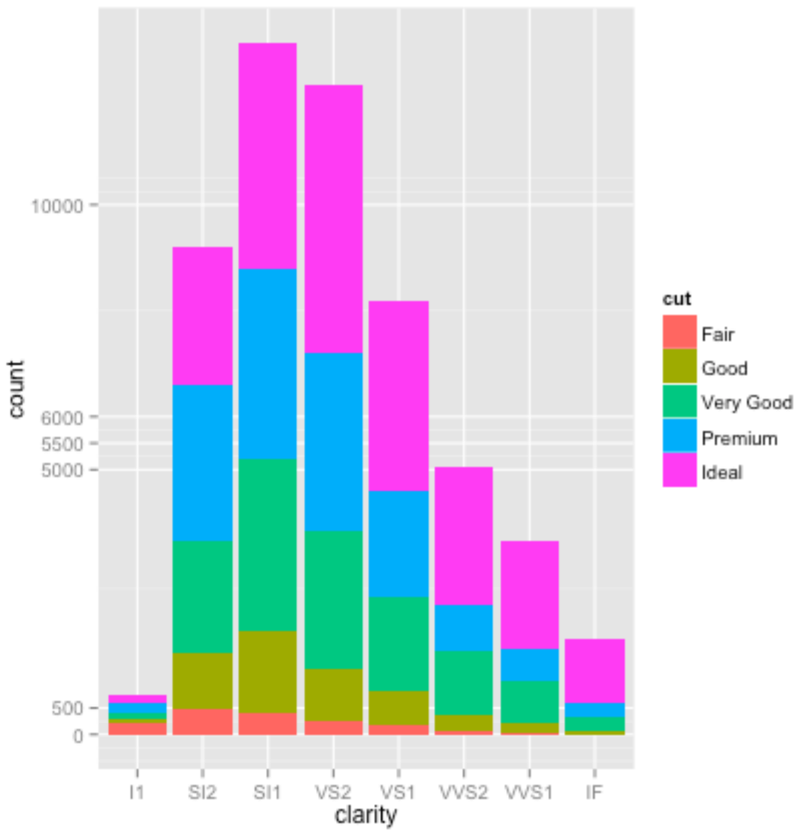


```
## Natural log (log2 and log10 also available)
p + scale_y_continuous(trans = "log")
```

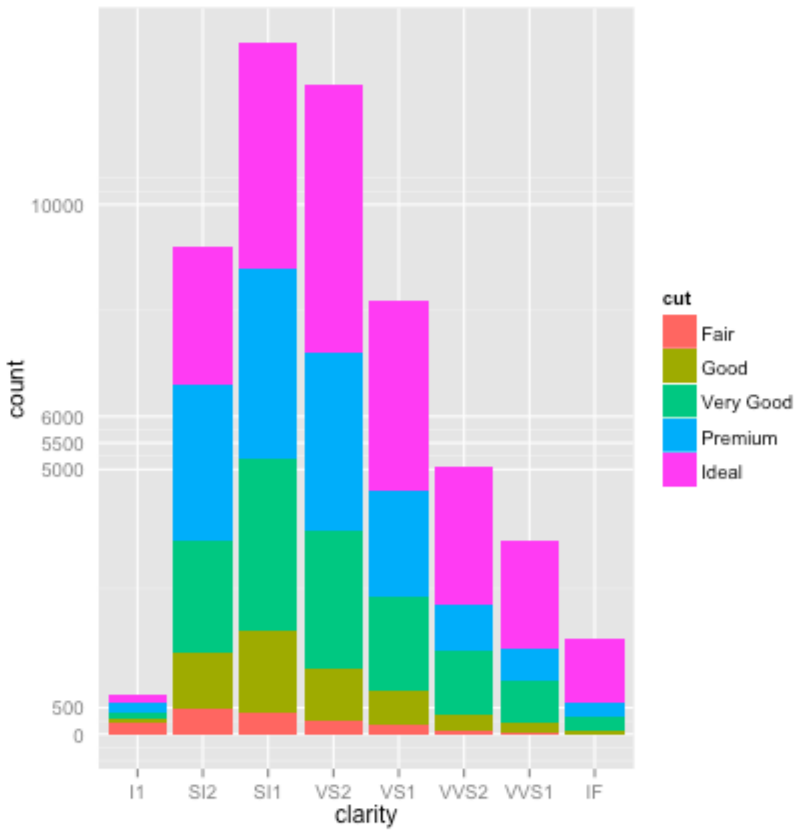


Other manipulations

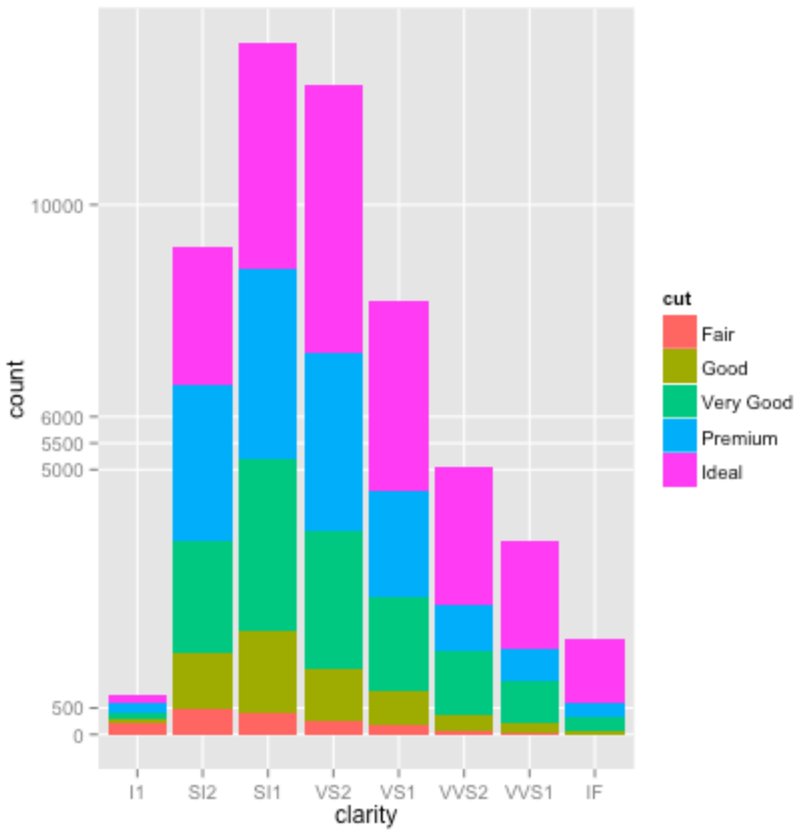
```
## Major breaks at arbitrary points
p.breaks <- p + scale_y_continuous(breaks =
c(0,500,5000,5500,6000,10000))
p.breaks
```



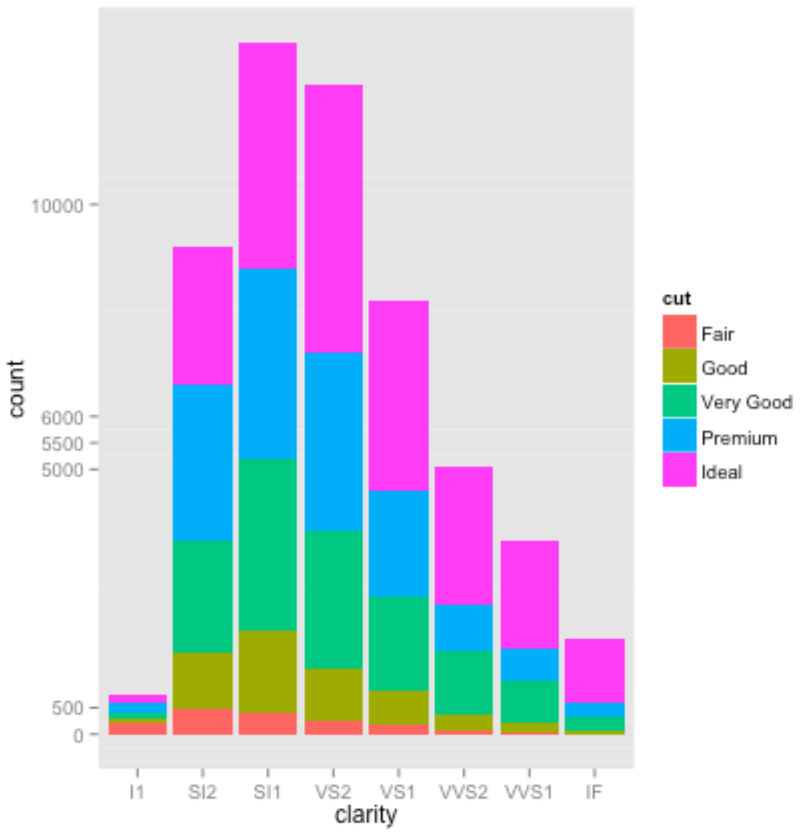
```
## No ticks for Y axis
p.breaks + theme(axis.ticks.y = element_blank())
```



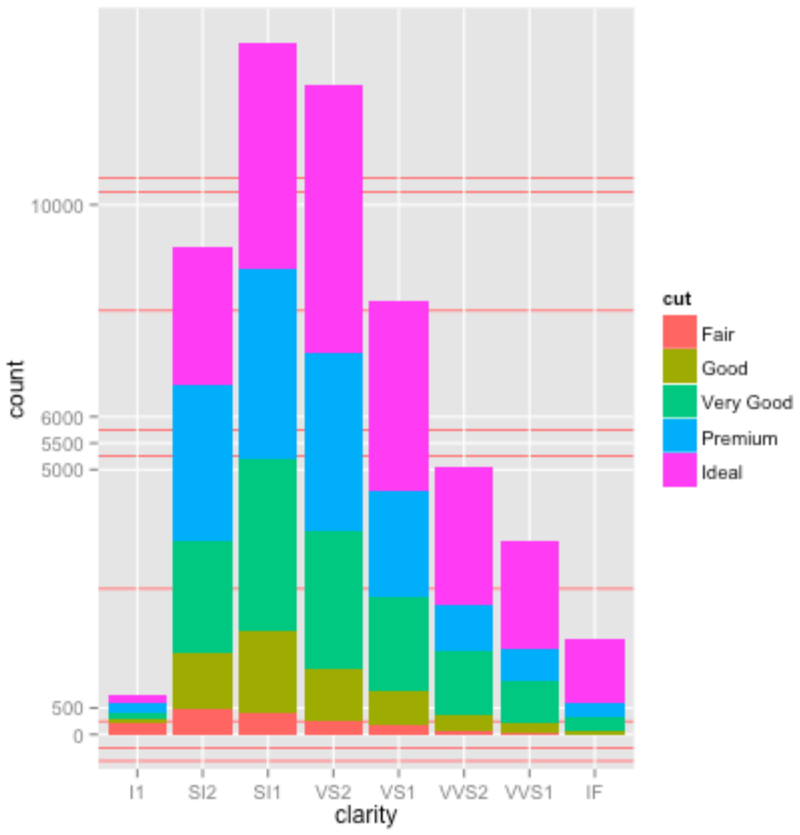
```
## No minor panel grid (major grid will remain)
p.breaks + theme(panel.grid.minor = element_blank())
```



```
## No major panel grid (ticks and labels will remain)
p.breaks + theme(panel.grid.major = element_blank())
```

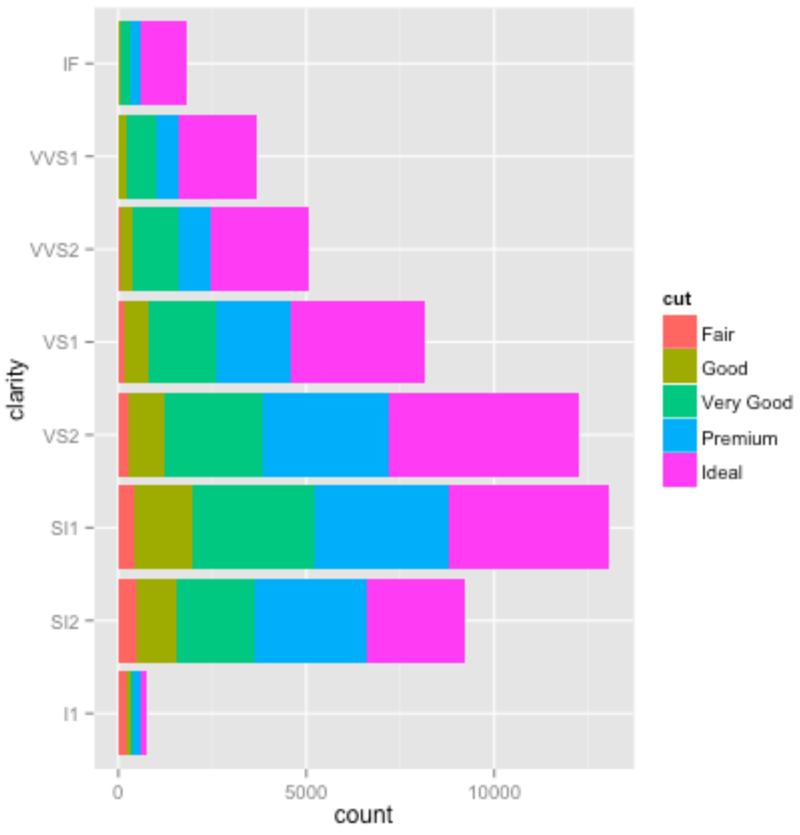


```
## Red minor panel grid
p.breaks + theme(panel.grid.minor = element_line(color =
"red"))
```



Flip X/Y axes

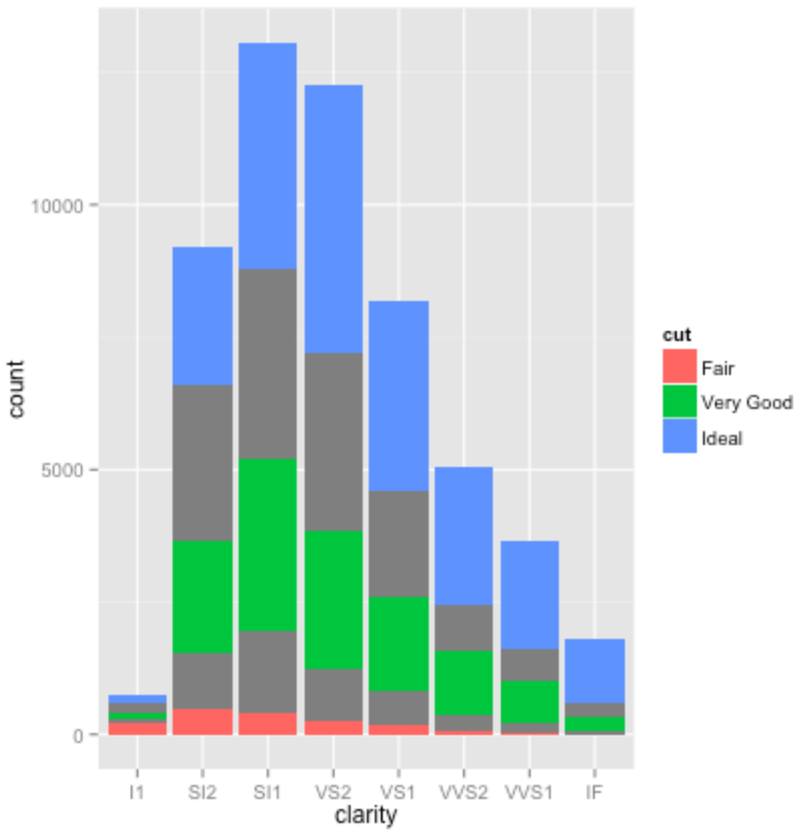
```
p + coord_flip()
```



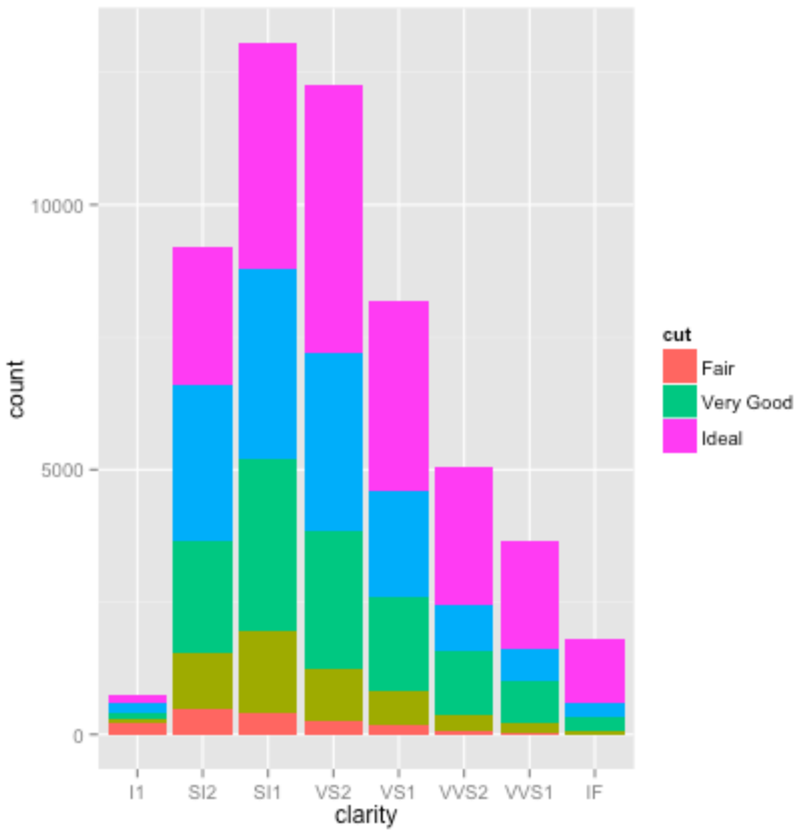
Manipulate color/fill scale

Discrete fill

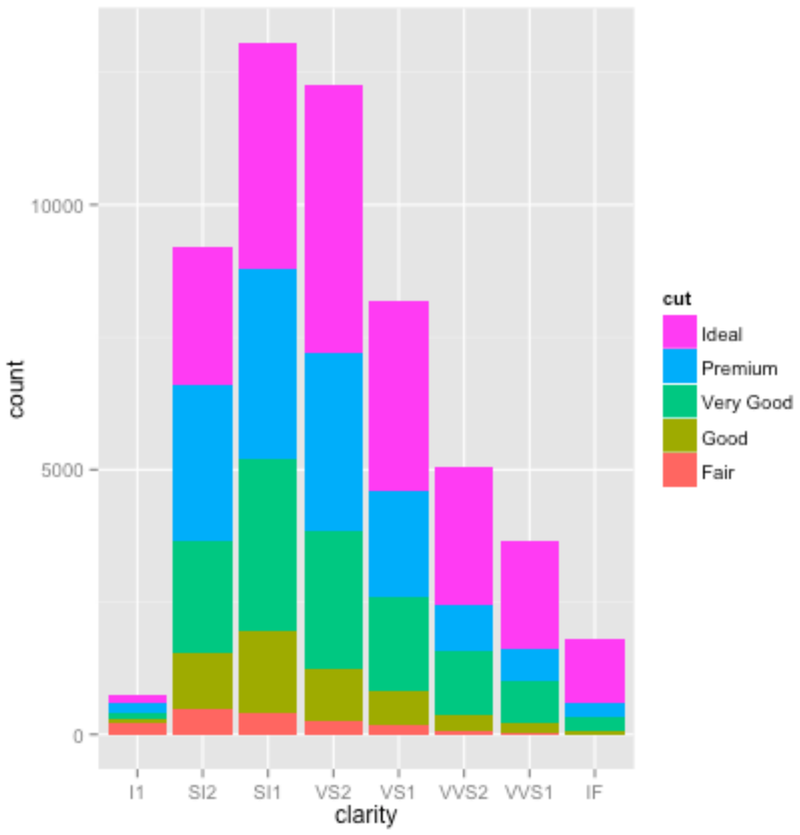
```
## limits affect the plot
p + scale_fill_hue(limits = c("Fair","Very Good","Ideal"))
```



```
## breaks affect the legend
p + scale_fill_hue(breaks = c("Fair","Very Good","Ideal"))
```

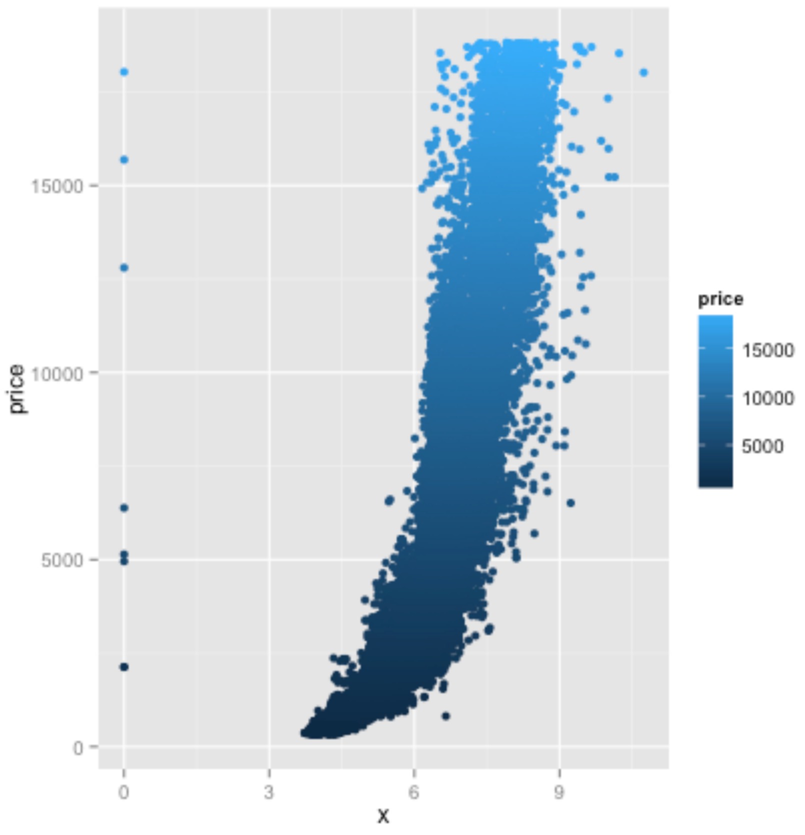


```
## breaks can be used to reverse the lengend ordering
p + scale_fill_hue(breaks = rev(levels(diamonds$cut)))
```

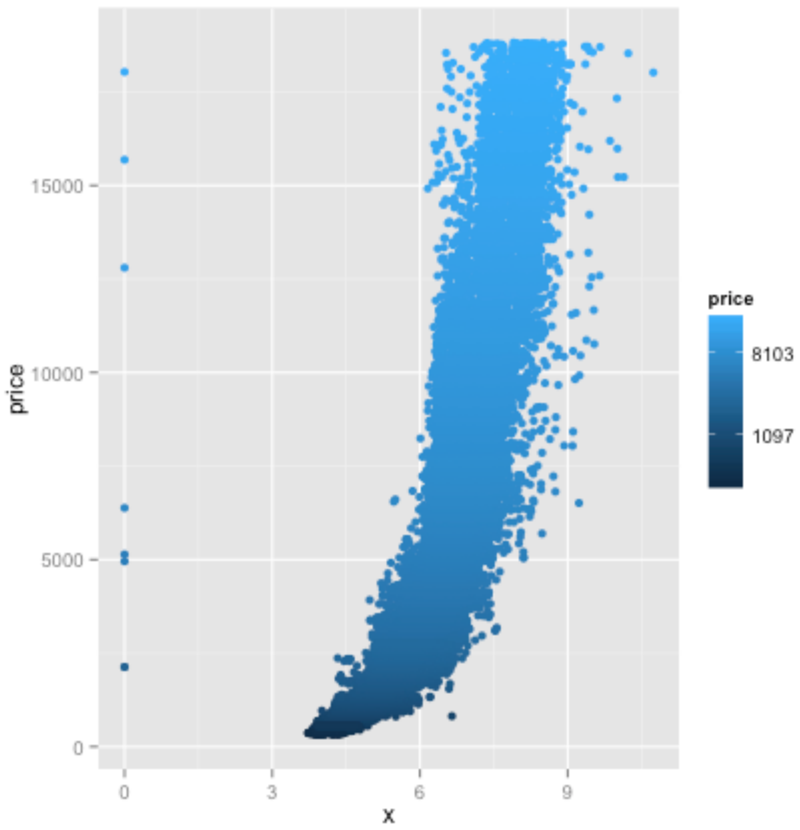


Continuous color

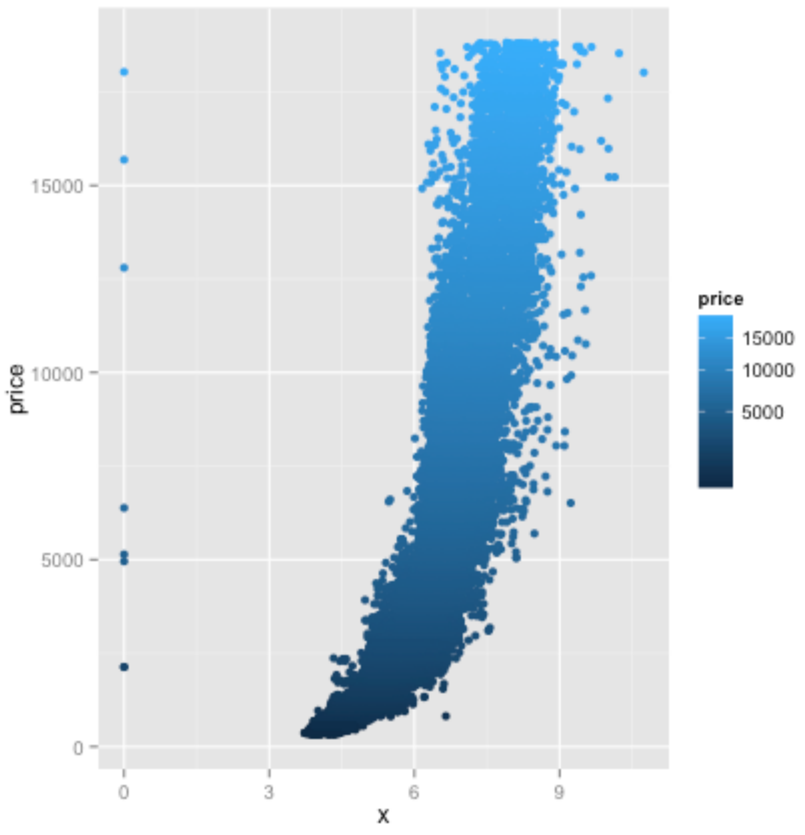
```
p.color <- ggplot(data = diamonds, mapping = aes(x = x, y =
price, color = price)) +
    layer(geom = "point")

## Default
p.color + scale_color_gradient()
```



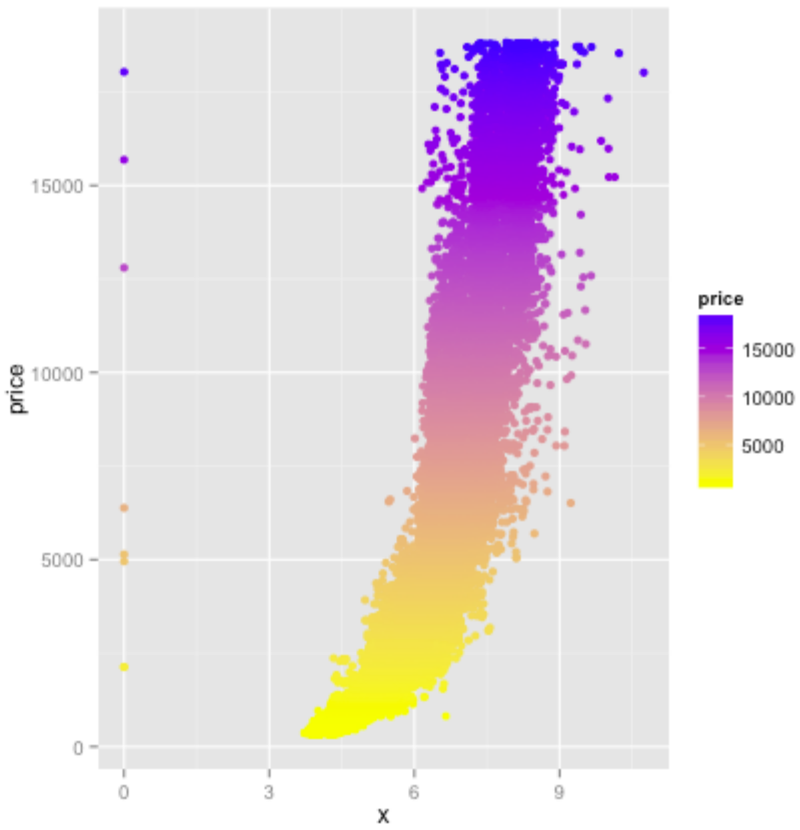
```
## log transformation
p.color + scale_color_gradient(trans = "log")
```



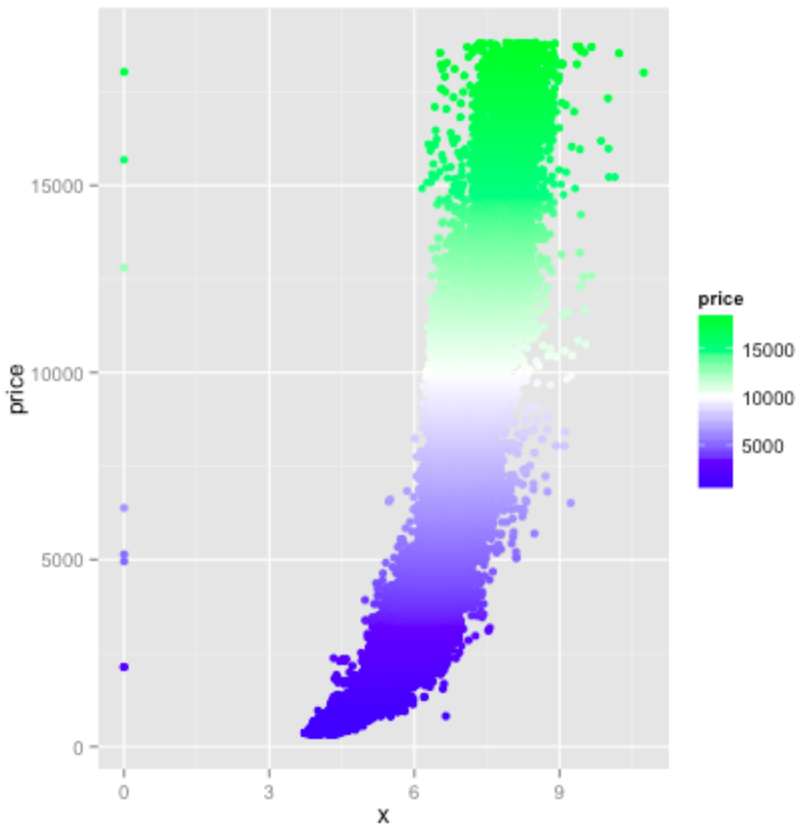
```
## sqrt transformation
p.color + scale_color_gradient(trans = "sqrt")
```



```
## Specify starting color and ending color of a gradient
p.color + scale_color_gradient(low = "yellow", high = "blue")
```

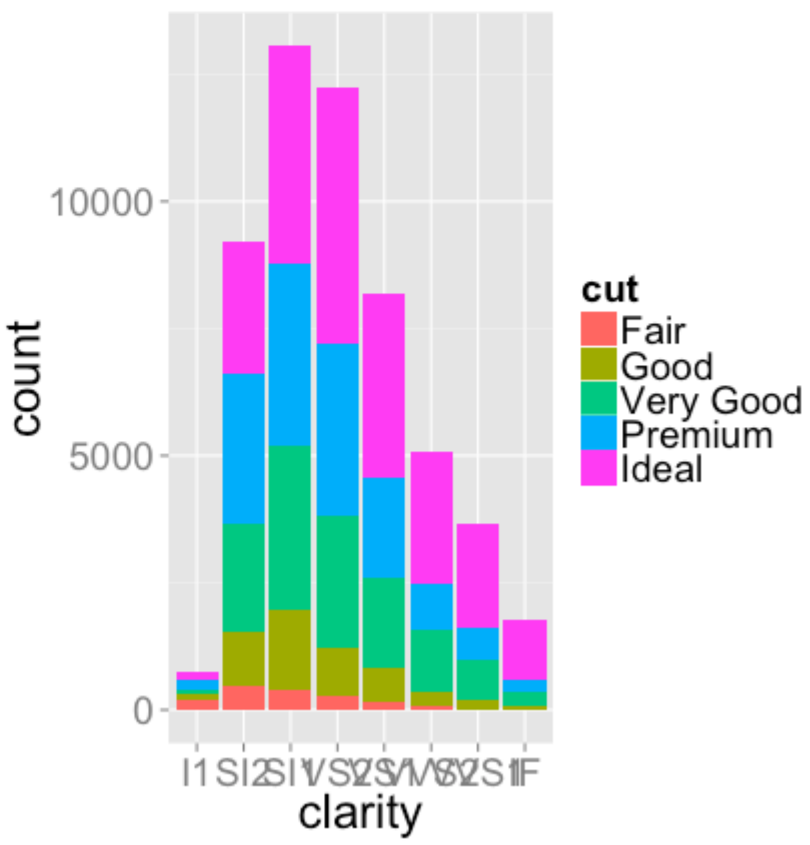


```
## Diverging colour gradient with scale_color_gradient2()
p.color + scale_color_gradient2(low = "blue", mid = "white",
high = "green", midpoint = 10000)
```

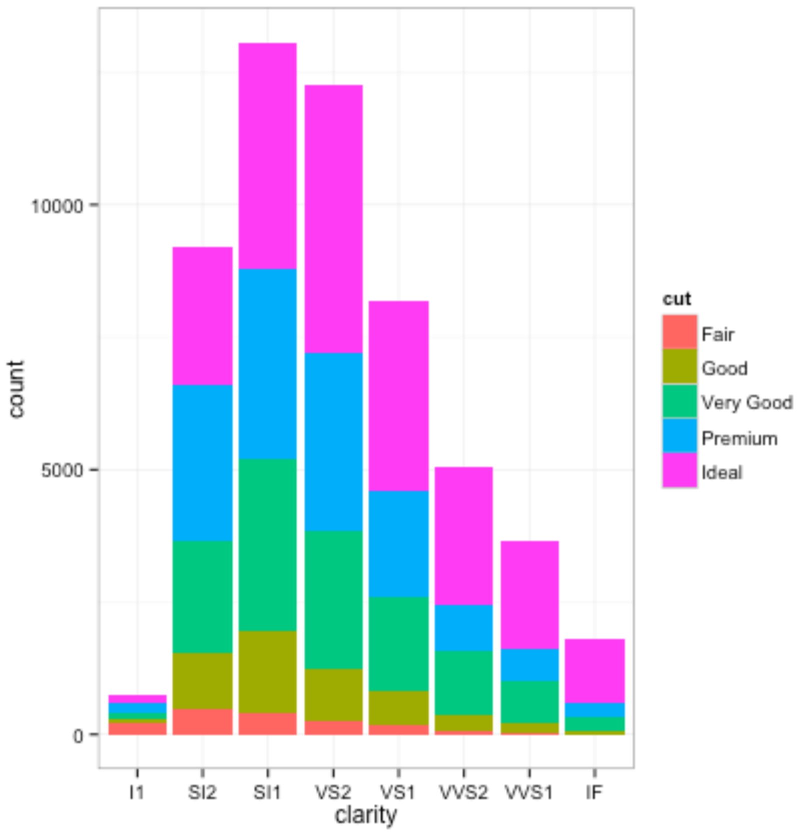


Themes

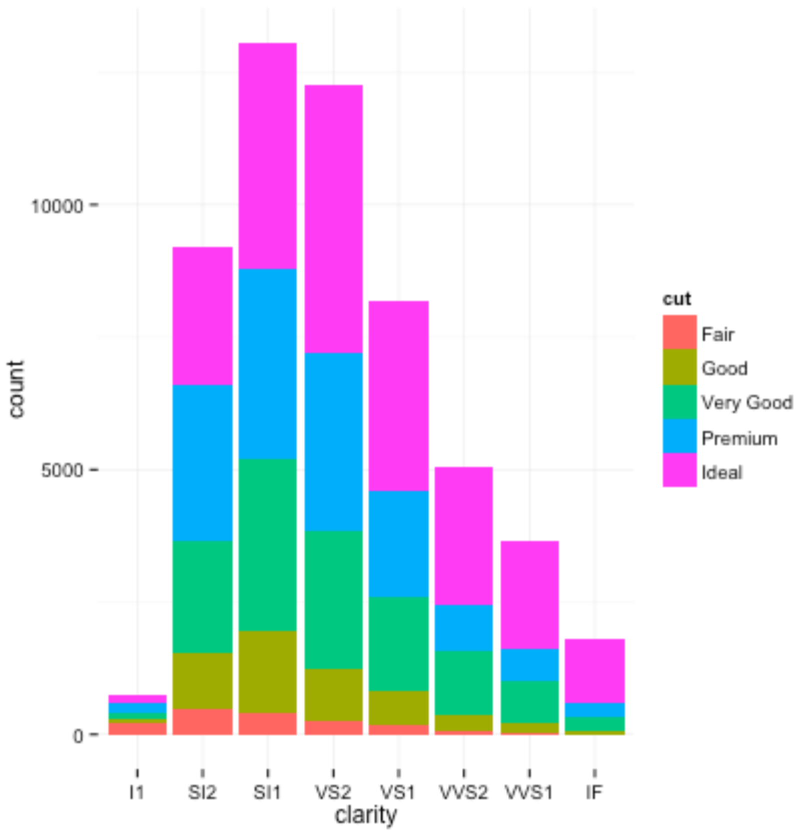
```
## A theme with grey background and white gridlines (default).
Altered font size.
p + theme_grey(base_size = 24)
```



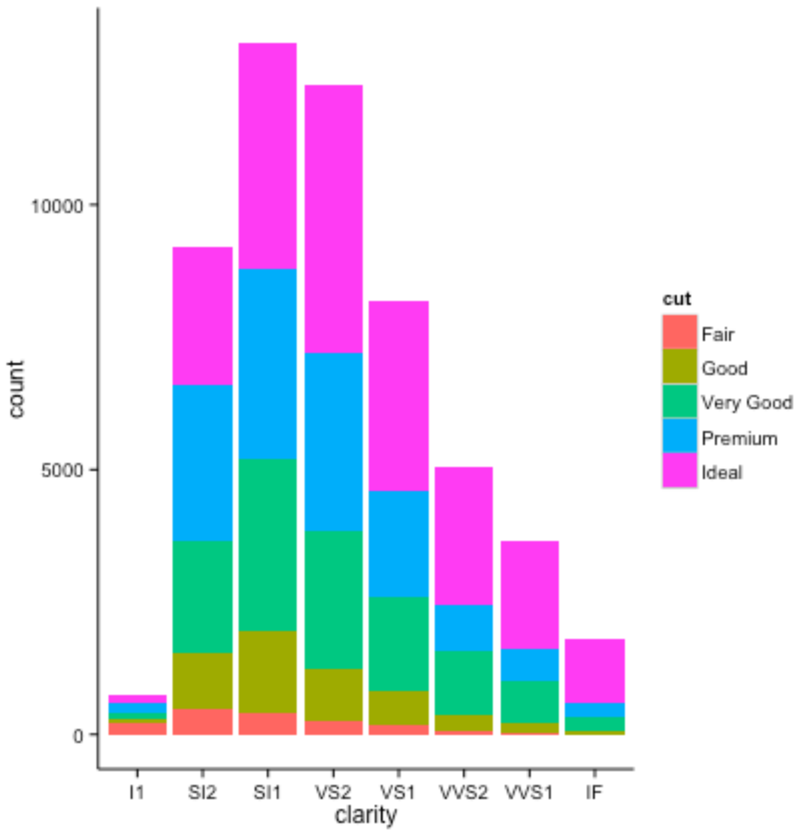
```
## A theme with white background and black gridlines.
p + theme_bw()
```



```
## A minimalistic theme with no background annotations.
p + theme_minimal()
```



```
## A classic-looking theme, with x and y axis lines and no
gridlines.
p + theme_classic()
```



Theme elements:

element_line

Theme element: line.

```
  colour: line colour
    size: line size
linetype: line type
 lineend: line end
   color: an alias for ‘colour’
```

element_rect

Most often used for backgrounds and borders.

```
    fill: fill colour
  colour: border colour
    size: border size
linetype: border linetype
   color: an alias for ‘colour’
```

element_text

Used for text manipulation.

```
  family: font family
    face: font face ("plain", "italic", "bold", "bold.italic")
  colour: text colour
    size: text size (in pts)
   hjust: horizontal justification (in [0, 1])
   vjust: vertical justification (in [0, 1])
   angle: angle (in [0, 360])
lineheight: line height
   color: an alias for ‘colour’
```

Elements

```
     The individual theme elements are:
       line                   all line elements('element_line')
       rect                   all rectangluarelements
('element_rect')
       text                   all textelements ('element_text')
       title                  all title
                              elements: plot, axes, legends
('element_text';
                              inherits from 'text')
       axis.title             label of axes
('element_text';inherits from 'text')
       axis.title.x           x axis
                              label ('element_text'; inherits
from
                              'axis.title')
       axis.title.y           y axis label
                              ('element_text'; inherits from
'axis.title')
       axis.text              tick labels along axes
                              ('element_text'; inherits from
'text')
       axis.text.x            x axis tick labels
('element_text';
                              inherits from 'axis.text')
       axis.text.y            y
                              axis tick labels ('element_text';
inherits from
                              'axis.text')
       axis.ticks             tick marks along
                              axes ('element_line'; inherits
from 'line')
       axis.ticks.x           x axis tick marks ('element_line';
                              inherits from 'axis.ticks')
       axis.ticks.y           y
                              axis tick marks ('element_line';
inherits from
                              'axis.ticks')
       axis.ticks.length      length oftick marks ('unit')
       axis.ticks.margin      spacebetween tick mark and tick
label ('unit')
```

```
       axis.line             lines along axes
('element_line';inherits from 'line')
       axis.line.x           line
                             along x axis ('element_line';
inherits from
                             'axis.line')
       axis.line.y           line along y axis
                             ('element_line'; inherits from
'axis.line')
       legend.background     background of legend
                             ('element_rect'; inherits from
'rect')
       legend.margin         extra space added around
legend('unit')
       legend.key            background underneath
                             legend keys ('element_rect';
inherits from
                             'rect')
       legend.key.size       size of legend keys
                             ('unit'; inherits from
'legend.key.size')
       legend.key.height     key background height
                             ('unit'; inherits from
'legend.key.size')
       legend.key.width      key background width ('unit';
                             inherits from 'legend.key.size')
       legend.text           legend item labels
('element_text'; inherits
                             from 'text')
       legend.text.align     alignment of
                             legend labels (number from 0
(left) to 1 (right))
       legend.title          title of legend
('element_text';inherits from 'title')
       legend.title.align    alignment of legend title (number
from 0 (left) to 1
                             (right))
       legend.position       the position of
                             legends.  ("left", "right",
"bottom", "top", or
                             two-element numeric vector)
       legend.direction      layout of items in legends
("horizontal" or "vertical")
       legend.justification  anchor point for
                             positioning legend inside plot
("center" or two-element
                             numeric vector)
       legend.box            arrangement of
                             multiple legends ("horizontal" or
"vertical")
       panel.background      background of plotting area, drawn
                             underneath plot (‘element_rect’;
inherits from
                             ‘rect’)
       panel.border          border around plotting
                             area, drawn on top of plot so that
it covers tick marks
                             and grid lines. This should be
used with ‘fill=NA’
                             (‘element_rect’; inherits from
‘rect’)
       panel.margin          margin around facet panels(‘unit’)
       panel.grid            grid lines(‘element_line’;
inherits from ‘line’)
       panel.grid.major      major grid lines
                             (‘element_line’; inherits from
‘panel.grid’)
       panel.grid.minor      minor grid lines
                             (‘element_line’; inherits from
‘panel.grid’)
       panel.grid.major.x    vertical major grid lines
                             (‘element_line’; inherits from
                             ‘panel.grid.major’)
       panel.grid.major.y    horizontal major grid lines
(‘element_line’;
                             inherits from ‘panel.grid.major’)
       panel.grid.minor.x    vertical minor grid lines
                             (‘element_line’; inherits from
                             ‘panel.grid.minor’)
       panel.grid.minor.y    horizontal minor grid lines
(‘element_line’;
                             inherits from ‘panel.grid.minor’)
       plot.background       background of the entire plot
                             (‘element_rect’; inherits from
‘rect’)
       plot.title            plot title (text appearance)
                             (‘element_text’; inherits from
‘title’)
       plot.margin           margin around entire plot (‘unit’
                             with the sizes of the top, right,
bottom, and left
                             margins)
       strip.background      background of facet labels
                             (‘element_rect’; inherits from
‘rect’)
       strip.text            facet labels
(‘element_text’;inherits from ‘text’)
       strip.text.x          facet
                             labels along horizontal direction
(‘element_text’;
                             inherits from ‘strip.text’)
       strip.text.y          facet labels along vertical
direction
                             (‘element_text’; inherits from
‘strip.text’)
```