Programmieren in Anwendungen

Jan Stärz

Technische Universität Kaiserslautern

j_staerz11@cs.uni-kl.de

20.06.2013

Überblick

Einleitung

Was ist ggplot2?

- Ein R-Package für Datenvisualisierung
- Unkompliziert, intuitive Benutzung, ermöglicht Modifikation der Plotkomponenten selbst auf hoher Abstraktionsebene
- Hauptfunktionen sind qplot und ggplot
- Arbeitet mit Layern (Schichten)

- Installation via *install.packages("ggplot2")*
- Einbindung in den Workspace via *library(ggplot2)*

Die Funktionen qplot und ggplot

qplot - Quick plot

qplot(data, x, y = NULL, ..., facets = NULL,
margins = FALSE, geom = "auto", stat = list(NULL),
position = list(NULL), xlim = c(NA, NA),
ylim = c(NA, NA), log = "", main = NULL,
xlab = deparse(substitute(x)),
ylab = deparse(substitute(y)), asp = NA)

- Meistgenutzte Parameter in rot!

ggplot - komplexere Plots

ggplot(data = NULL, aes)

- aes(x, y, color, alpha, size,...)
  - Wie Datenvariablen auf visuelle Eigenschaften der benutzten Geometrie übertragen werden (Ästhetik)
  - Für alle folgenden Layer gültig
- Inkrementelle Ploterstellung
  - Addition von Anweisungen mithilfe von +
  - Z.B. geom_point(), geom_bar(), xlab("X-Achse"), facet_grid(...)

Data frame “diamonds“

Struktur von “diamonds“

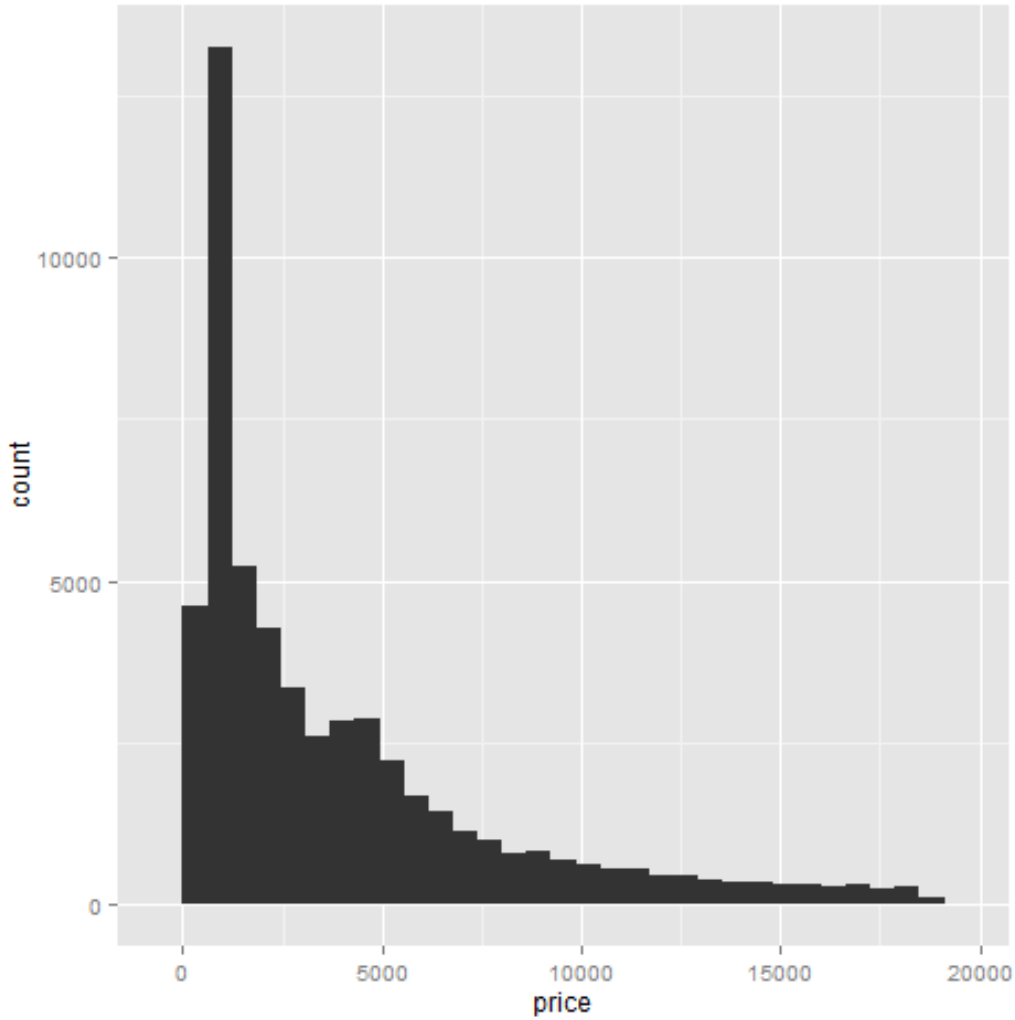
- Standardmäßig im ggplot2-Package enthalten
- Beinhaltet Eigenschaften zu fast 54.000 Diamanten

```
> head(diamonds)
  carat       cut color clarity depth table price   ...
1  0.23     Ideal     E     SI2  61.5    55   326   ...
2  0.21   Premium     E     SI1  59.8    61   326   ...
3  0.23      Good     E     VS1  56.9    65   327   ...
4  0.29   Premium     I     VS2  62.4    58   334   ...
5  0.31      Good     J     SI2  63.3    58   335   ...
6  0.24 Very Good     J    VVS2  62.8    57   336   ...
```

Der Umgang mit qplot

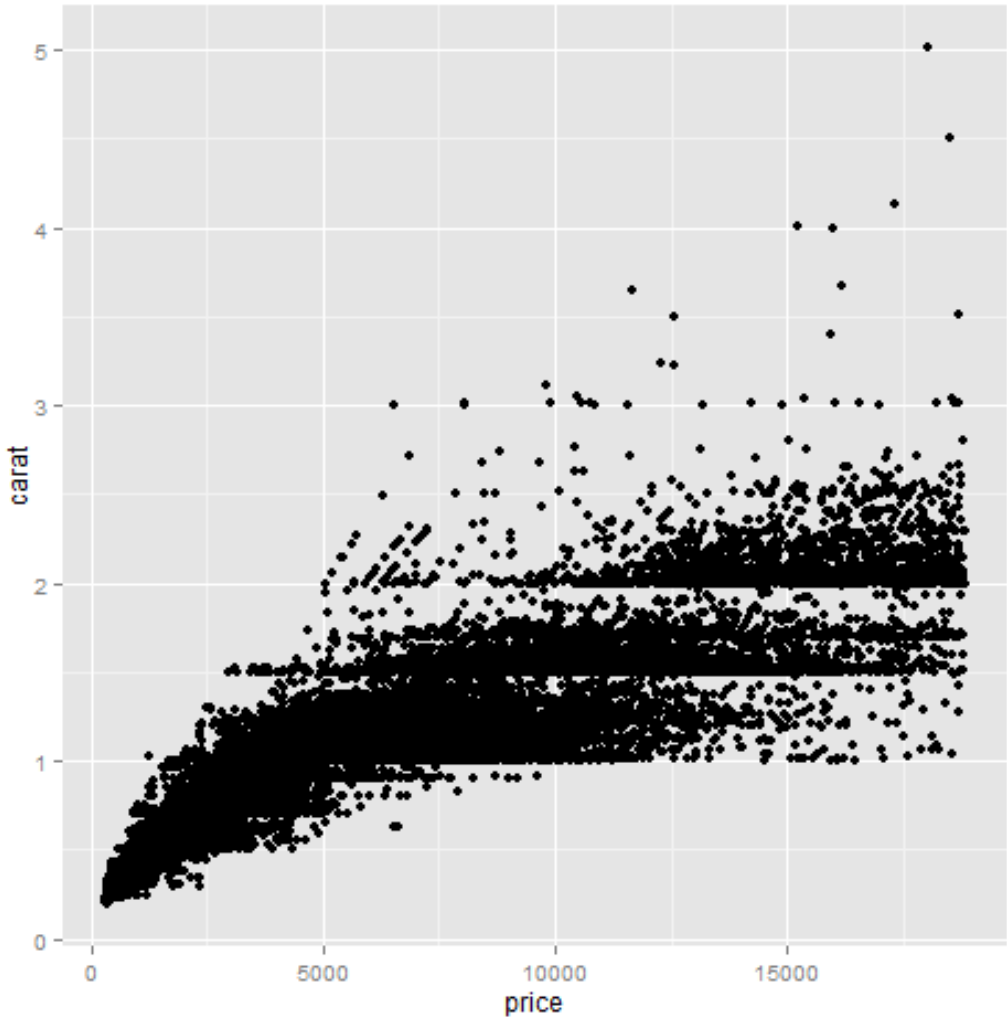
Der erste Plot

```
qplot(data=diamonds, price)
```



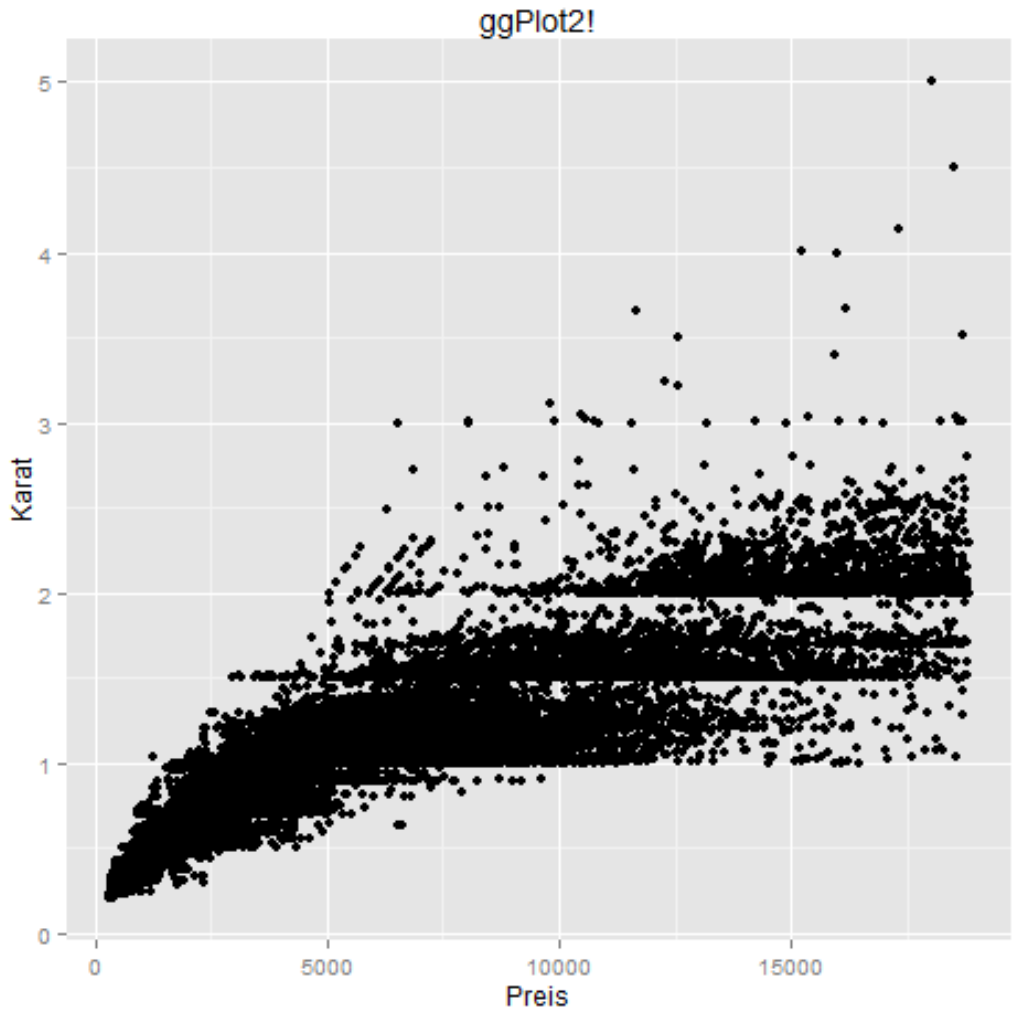
Der erste Plot - Version 2

```
qplot(data=diamonds, price, carat)
```



Der erste Plot - Version 3

```
qplot(data=diamonds, price, carat, main="ggPlot2!",
      xlab="Preis", ylab="Karat")
```

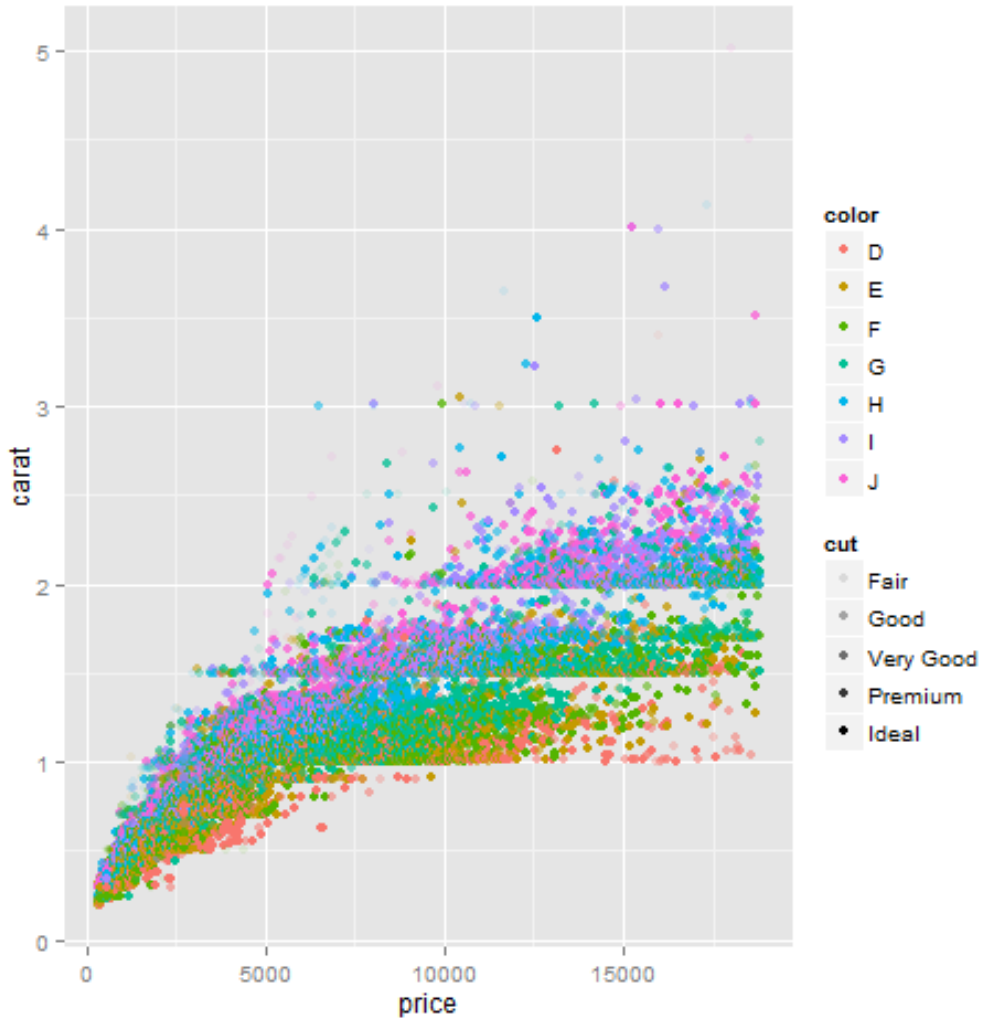


Der erste Plot - Version 4

```
qplot(data=diamonds, price, carat, color=color)
```

Der erste Plot - Version 5

```
qplot(data=diamonds, price, carat, color=color,
      alpha=cut)
```



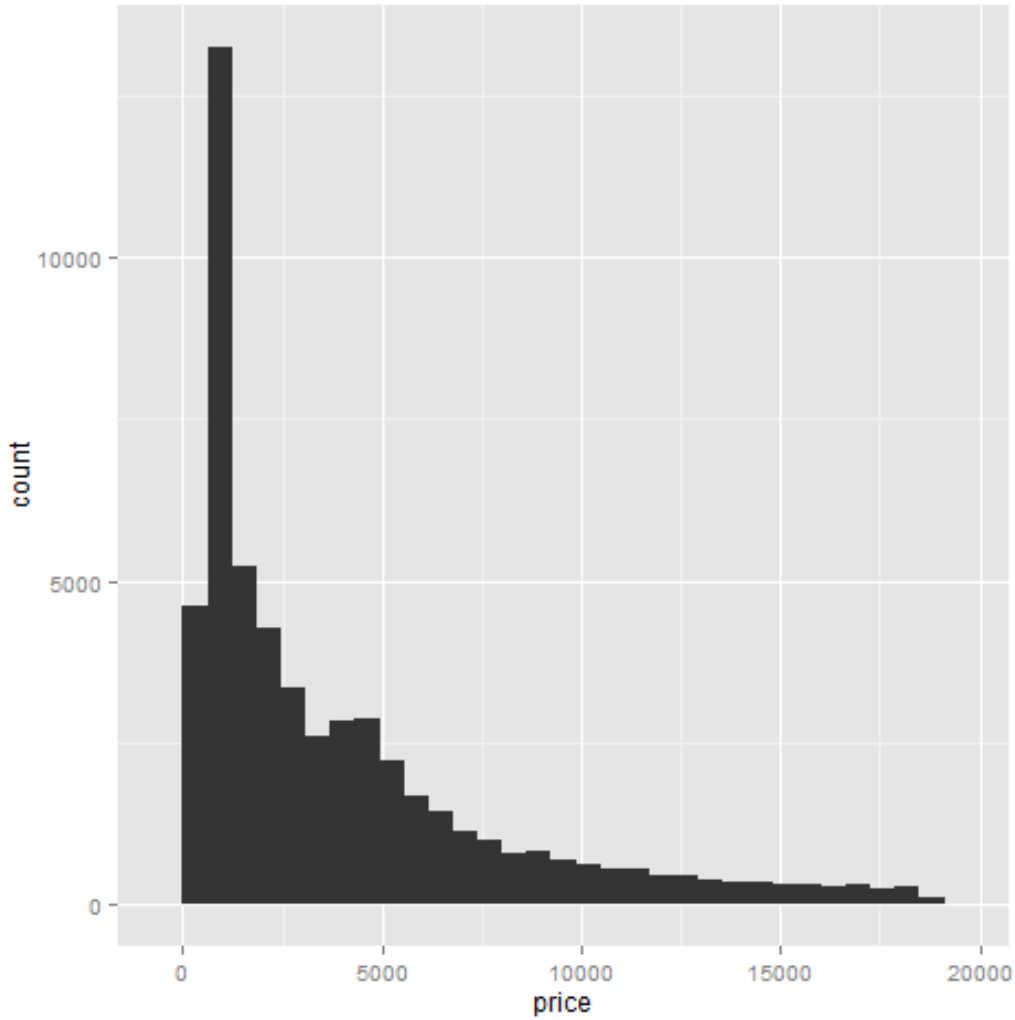
Der Umgang mit ggplot

“No layers in plot“

```
myPlot <- ggplot(data=diamonds, aes(x=price))
```

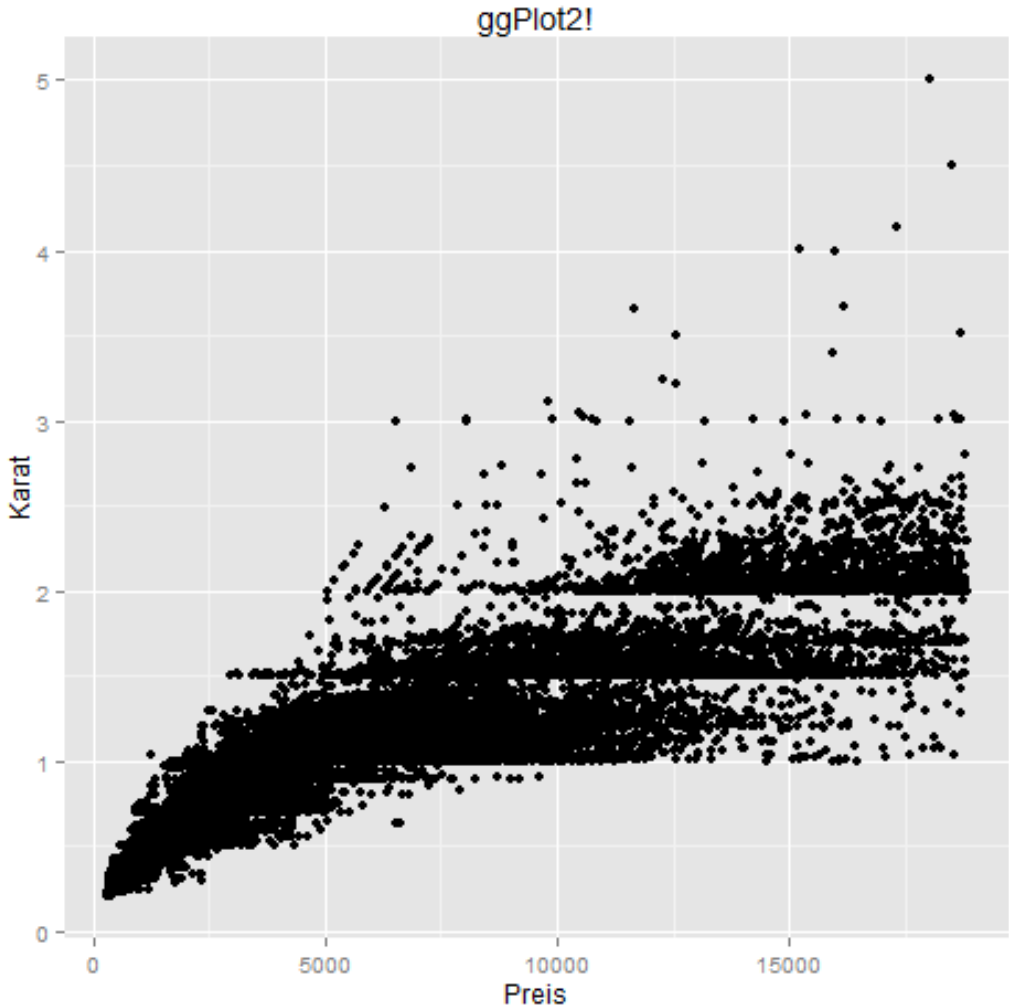
Der analoge ggPlot

```
myPlot <- myPlot + geom_bar()
```



Der analoge ggPlot - Version 2

```
myPlot <- ggplot(data=diamonds, aes(x=price, y=carat))
myPlot + geom_point()
       + ggtitle("ggPlot2") + xlab("Preis") + ylab("Karat")
```

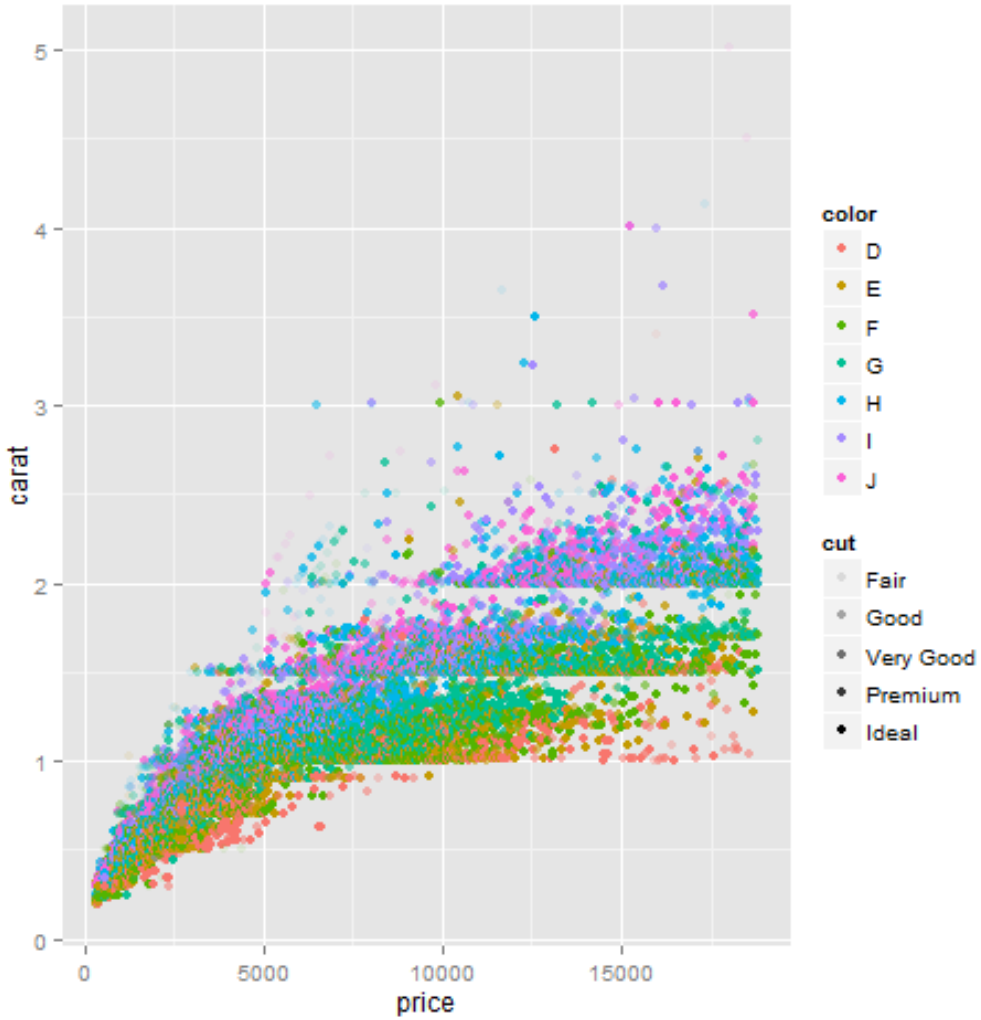


Der analoge ggPlot - Version 3

```
myPlot + geom_point(aes(color=color))
```

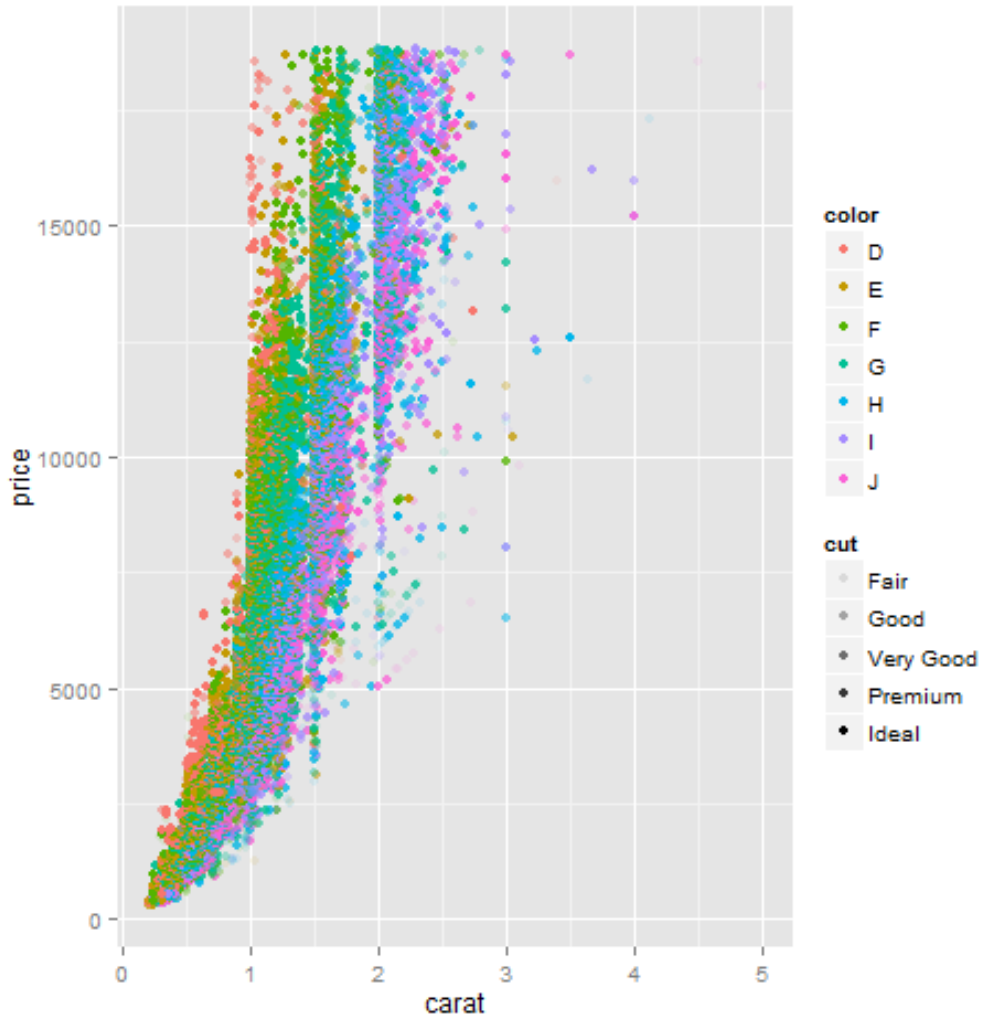
Der analoge ggPlot - Version 4

```
myPlot + geom_point(aes(color=color, alpha=cut))
```



Der analoge ggPlot - Version 5

```
myPlot + geom_point(aes(color=color, alpha=cut))
       + coord_flip()
```



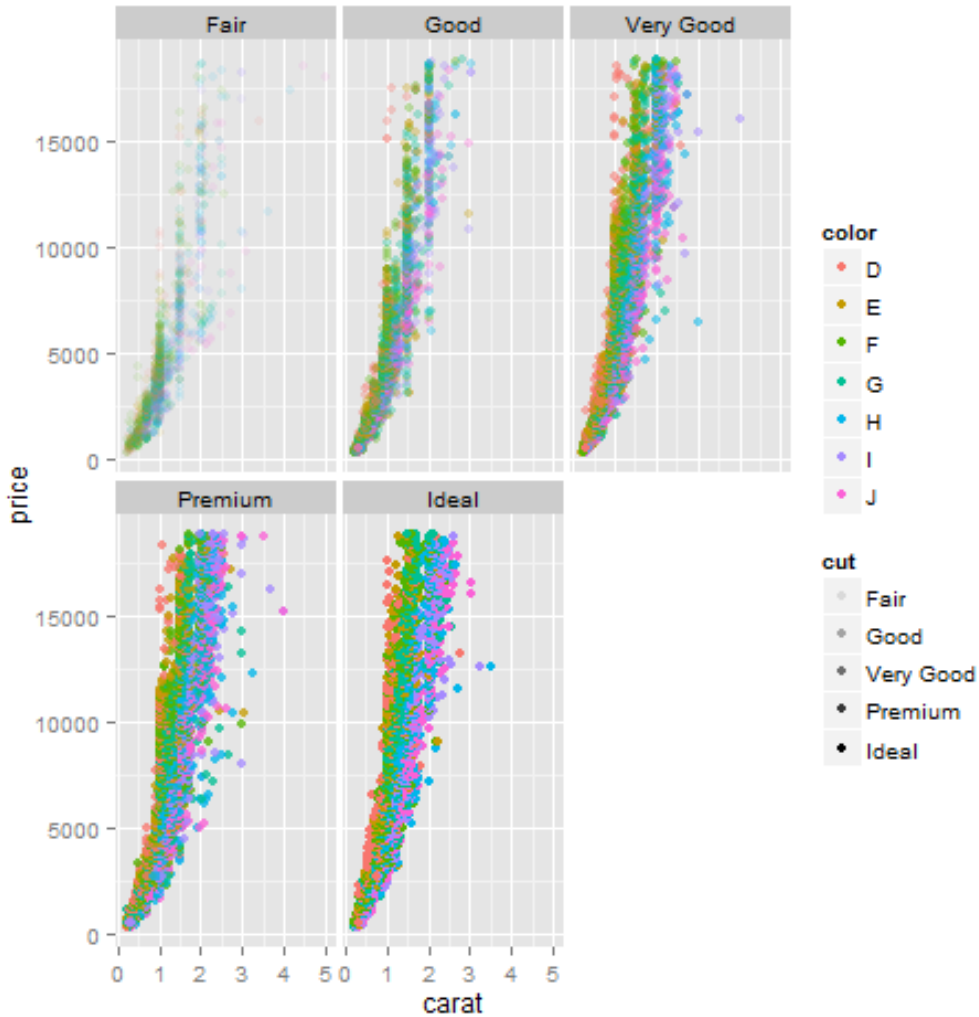
Einschub - Facetten

Facetten dienen der Aufsplitterung von Diagrammen in die einzelnen Kategorien eines Parameters:

+ **facet_grid(horizontal ~ vertical)**
  - Gruppiere in der Horizontalen nach den Werten von cut *facet_grid(cut ~ .)*
  - Gruppiere in der Vertikalen nach den Werten von cut *facet_grid(. ~ cut)*
+ **facet_wrap(~ vertical)**
  - Automatische Platzierung der Facetten neben- und untereinander
  - Parameter *ncol=x* begrenzt das Wrapping horizontal

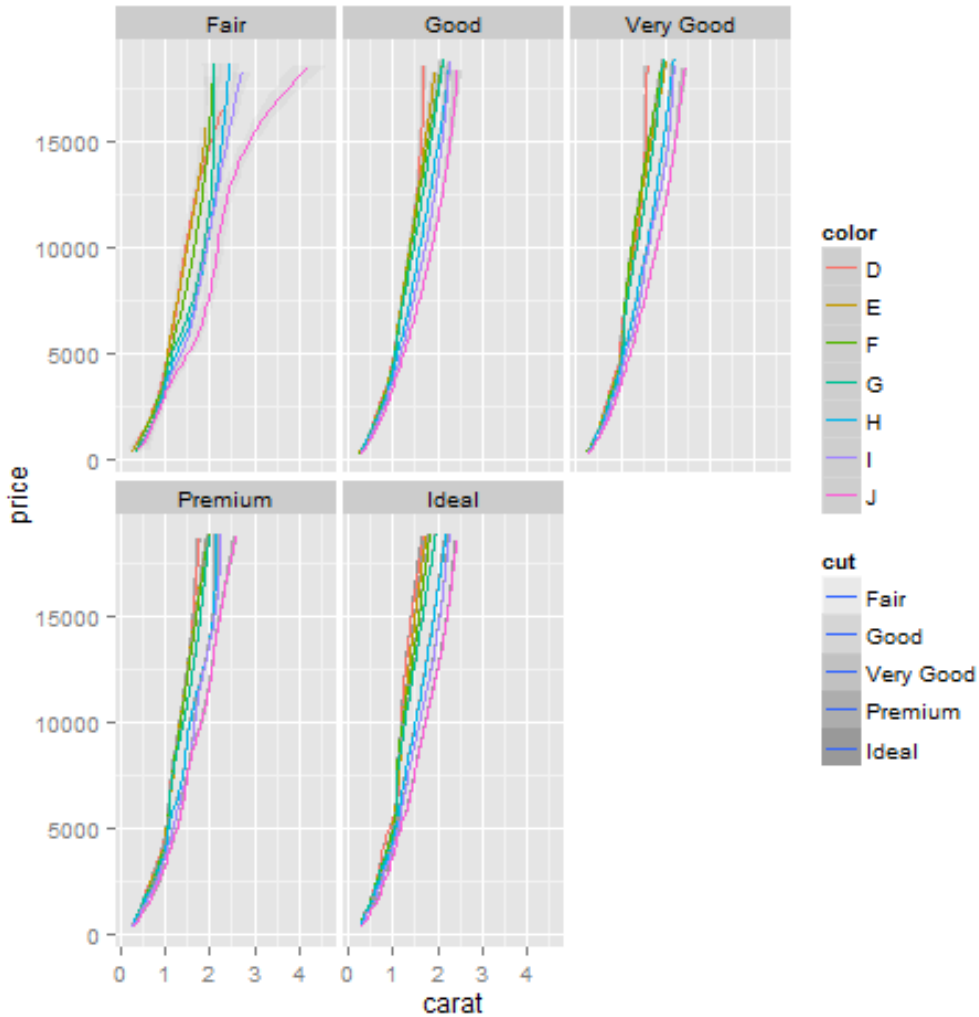
Der analoge ggPlot - Version 6

```
myPlot + geom_point(aes(color=color, alpha=cut))
       + coord_flip() + facet_wrap(~ cut)
```



Der analoge ggPlot - Version 7

```
myPlot + geom_smooth(aes(color=color, alpha=cut))
       + coord_flip() + facet_wrap(~ cut)
```



ggplot - Nützliche Geometrien

Säulendiagramm

```
myPlot2 <- ggplot(data=diamonds, aes(x=clarity))
```

+ geom_bar(

  data=w, (Definition eigener Data frame)

  width=x, (Einstellung der Breite)

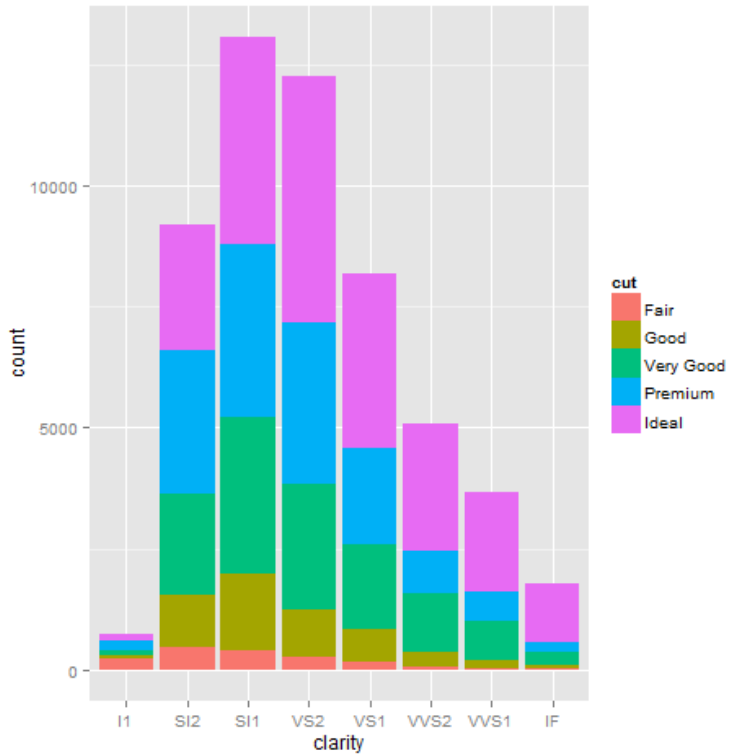
  aes(fill=y), (Füllfarbe der Balken abhängig von y)

  position="z" (Anordnung der gefärbten Balken)

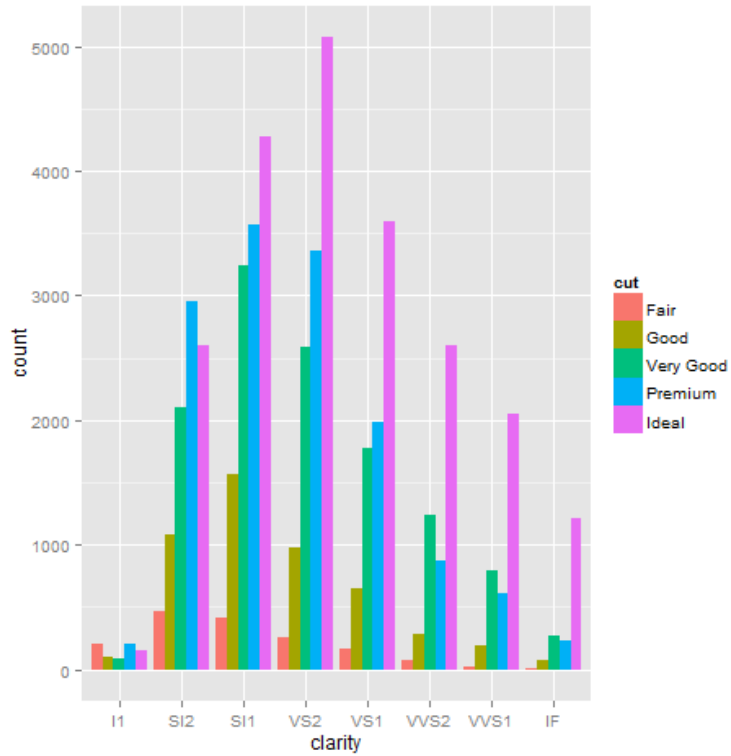
  - Möglich sind "stack" (*default*), "dodge", "fill" und "identity"

)

Säulendiagramm, Beispiel



(a) aes(fill=cut), position="stack"



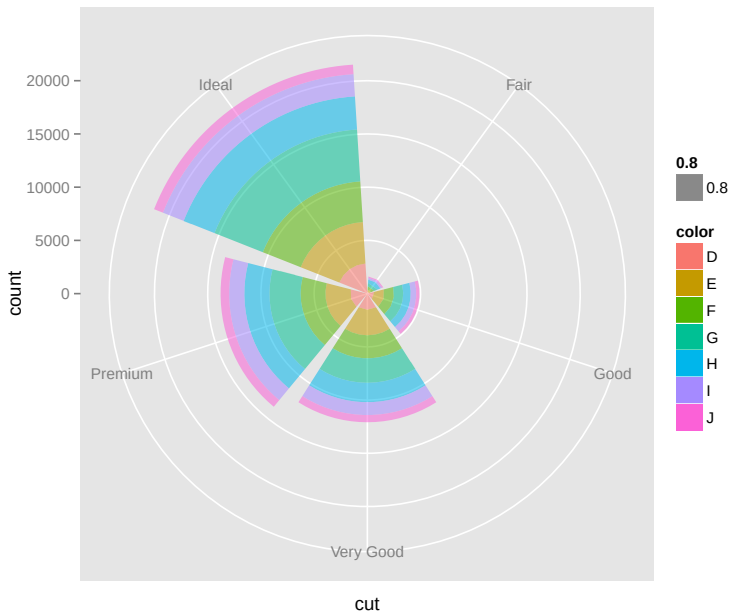
(b) aes(fill=cut), position="dodge"

Kreisdiagramme, spezielle Säulendiagramme

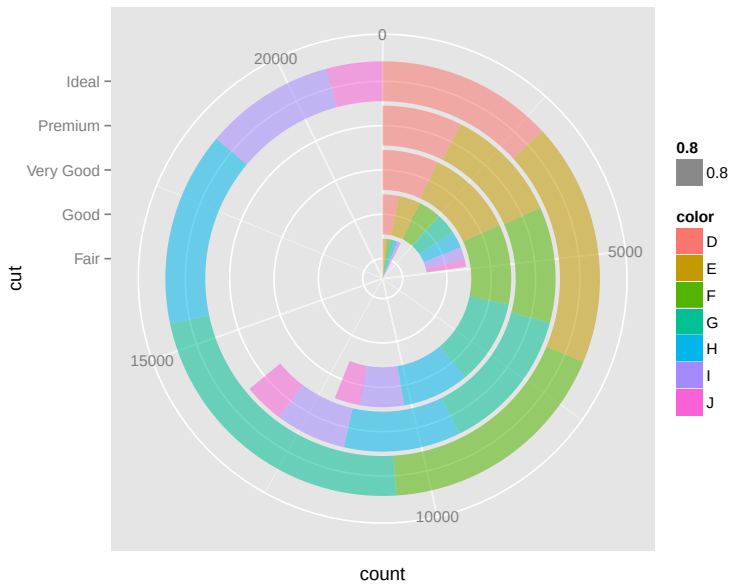
+ geom_bar() + coord_polar(

theta="x" (welche Achse den Kreisumfang bildet, "x"/"y")

)



(c)



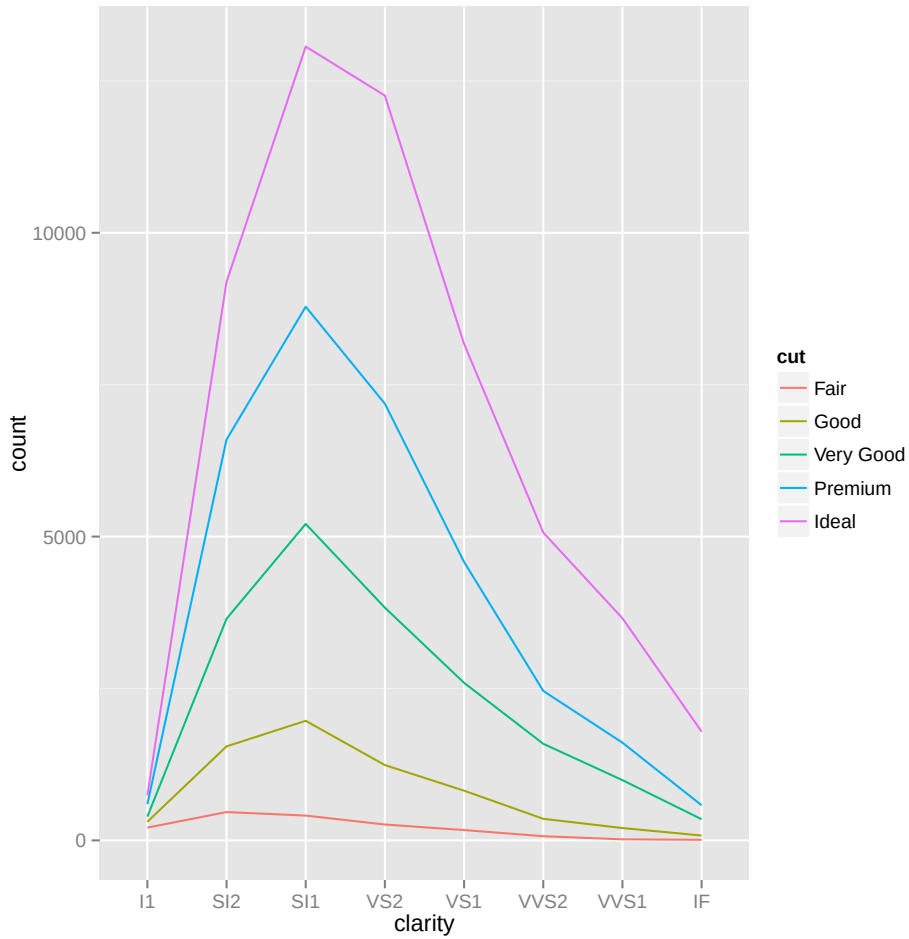
(d) theta="y"

Frequency polygon

```
geom_freqpoly(
    data=w, (Definition eigener Data frame)
    aes(group=x, (Linien nach Kriterium x)
    color=x), (Farben der Kriterien)
    position="z" (Anordnung der Linien)
    )
```

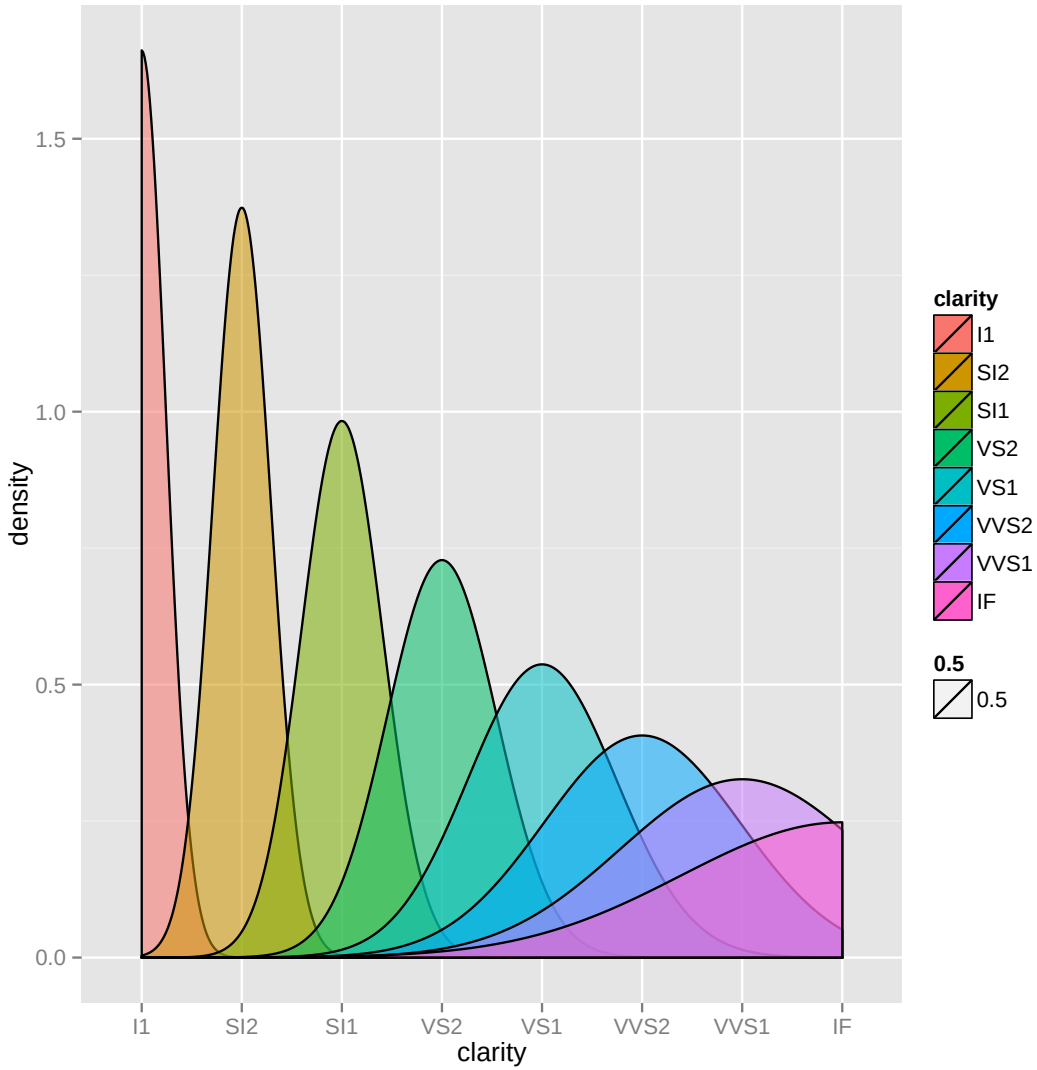
Frequency polygon, Beispiel

```
myPlot2 + geom_freqpoly(aes(group=cut, color=cut),
                        position="stack")
```

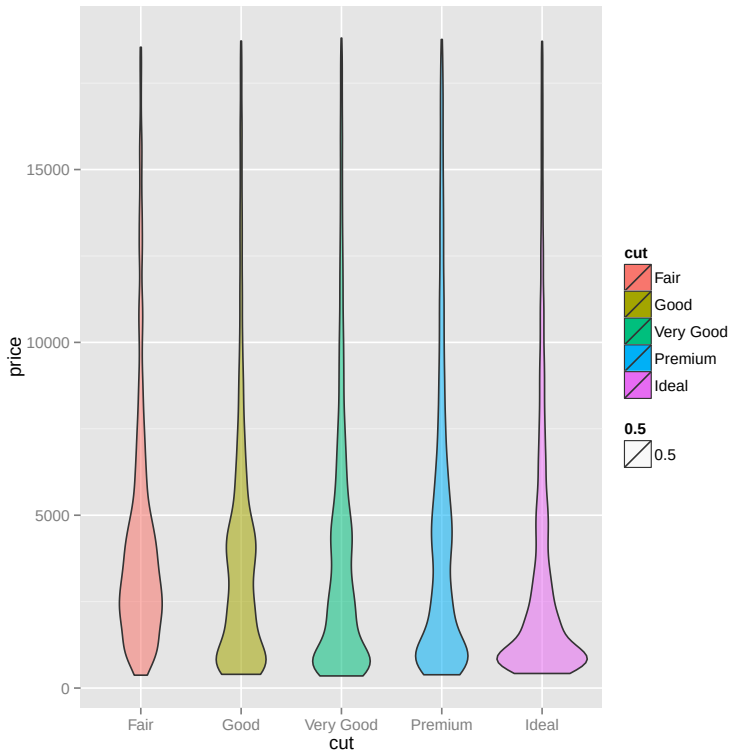


Dichteauswertung “density“

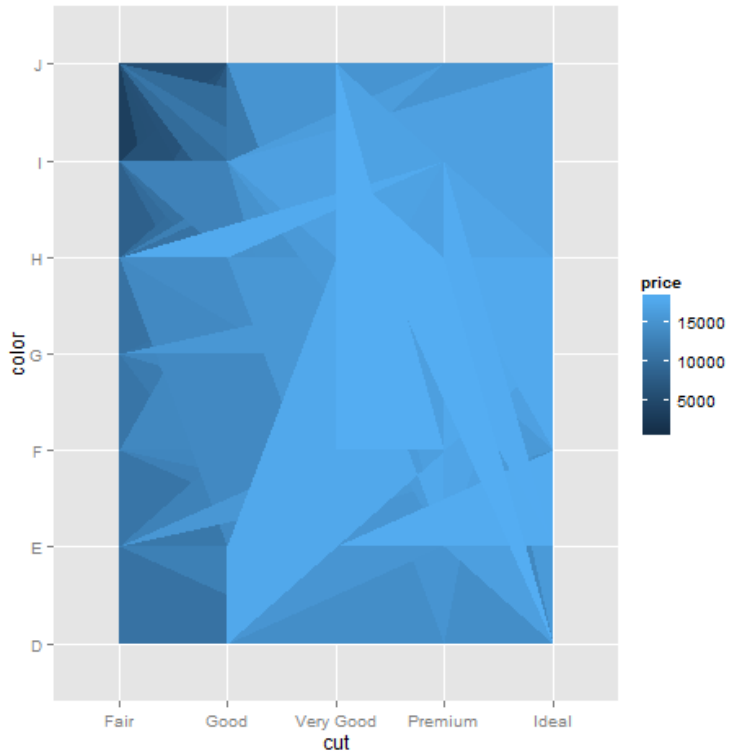
```
myPlot2 + geom_density(aes(fill=clarity, alpha=0.5))
```



...und viele mehr



(e) geom_violin()



(f) geom_polygon()