

Visualizing in R

– advanced plotting

Norsk statistikermøte, Halden, 10. juni 2013

Elisabeth Orskaug
Thordis Thorarinsdottir

Norsk Regnesentral

3. Introduction to ggplot2

ggplot2 is built on a specific *grammar of graphics* that maps

data \longrightarrow $(color, shape, size)$ aesthetic attributes of $(points, lines, bars)$ geometric objects

Its main elements are

layers first a layer of raw data, then additional layers of annotations and statistical summaries

faceting breaking the data into subsets and displaying those

It has two main functions that replace `plot()`

`qplot()` quick plot, similar to `plot()`, not generic

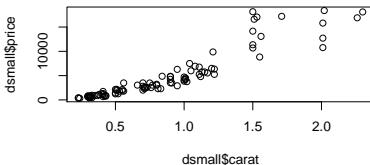
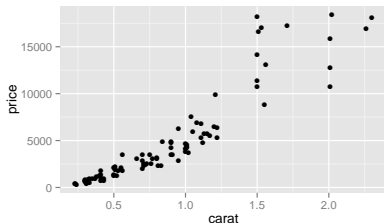
`ggplot()` more involved, generic

Example dataset

```
> data(diamonds)
> head(diamonds)
```

	carat	cut	color	clarity	depth	table	price	x	y	z
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	4.20	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

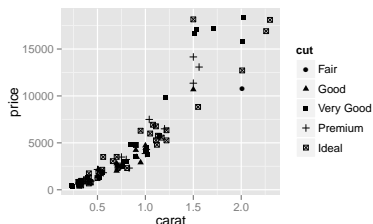
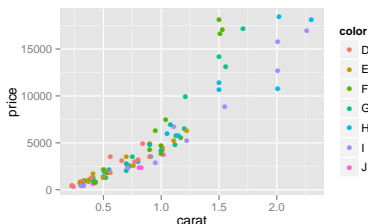
Simple comparison of `qplot()` and `plot()`



```
qplot(carat, price, data=dsmall)
plot(dsmall$carat, dsmall$price)
```

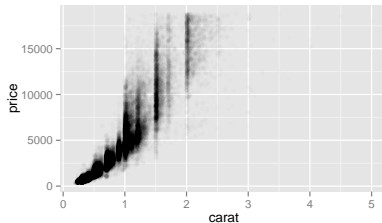
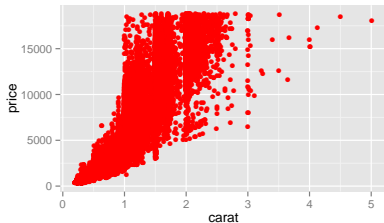
Note that data must be a data frame object.

Adding aesthetic attributes



```
qplot(carat, price, data=dsmall, colour=color)
ggsave("fig//diamonds3.pdf", width=5, height=3)
qplot(carat, price, data=dsmall, shape=cut)
ggsave("fig//diamonds4.pdf", width=5, height=3)
```

Manually setting aesthetics



```
qplot(carat, price, data=diamond, colour=I("red"))  
qplot(carat, price, data=diamond, alpha=I(1/100))
```

Changing the geometric object

`geom` describes the type of object used to display the data

“point” scatterplot, the default

“smooth” fits a smoother and displays it with standard error

“boxplot” box-and-whisker plot (e.g. catagorical vs. continuous)

“jitter” plots jittered points (e.g. catagorical vs. continuous)

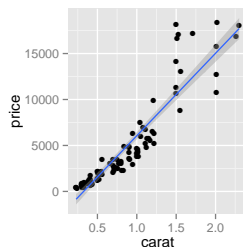
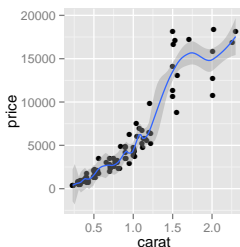
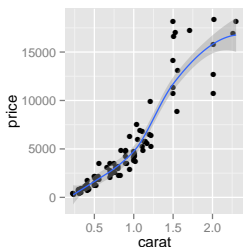
“line” draws lines between the data points that travel from left to right

“path” lines that go in any direction

We can use more than one at a time with e.g.

```
geom = c(“point”, “smooth”).
```

Example: Adding a smoother



```
qplot(carat, price, data=dsmall, geom=c("point","smooth"))
qplot(carat, price, data=dsmall, geom=c("point","smooth"),
      span=0.2)
qplot(carat, price, data=dsmall, geom=c("point","smooth"),
      method="lm", formula=y~x)
```


One-dimensional geometric objects

Same set-up can display only one data vector.

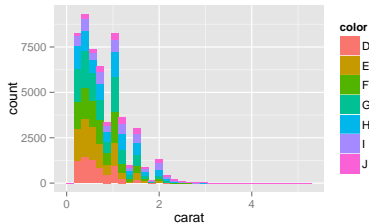
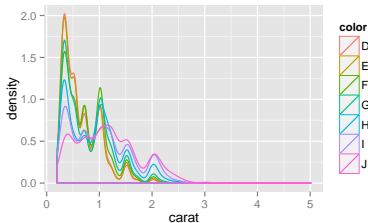
Here, values for `geom` include

“`histogram`” histogram, the default

“`density`” density plot

“`bar`” bar chart

Example: Densities vs. histograms



```
qplot(carat, data=diamonds, geom="density", colour=color)  
qplot(carat, data=diamonds, geom="histogram", fill=color)
```

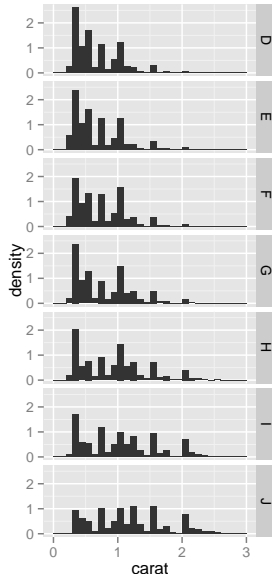
Now it is your turn...

Use the `qplot()` options we have just discussed to visualize the data set `mpg`.

```
> data(mpg)
> head(mpg)
  manif model displ year  cyl      trans drv  cty  hwy fl  class
1  audi   a4    1.8 1999   4   auto(l5)  f   18   29  p compact
2  audi   a4    1.8 1999   4 manual(m5)  f   21   29  p compact
3  audi   a4    2.0 2008   4 manual(m6)  f   20   31  p compact
4  audi   a4    2.0 2008   4   auto(av)   f   21   30  p compact
5  audi   a4    2.8 1999   6   auto(l5)  f   16   26  p compact
6  audi   a4    2.8 1999   6 manual(m5)  f   18   26  p compact
> ?mpg
```

Faceting creates a table of graphics

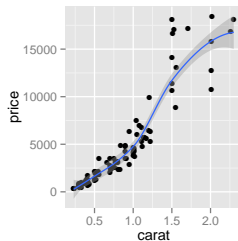
```
qplot(carat, ..density..,  
      data=diamonds,  
      facets=color ~ .,  
      geom="histogram",  
      binwidth=0.1,  
      xlim=c(0,3))
```



ggplot() and layers

ggplot() has two arguments, **data** and aesthetic **mapping**.

To see a plot, we need to add layers.

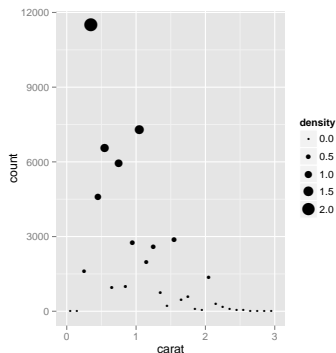


```
> qplot(carat, price, data=dsmall, geom=c("point","smooth"))
> ggplot(dsmall, aes(carat, price)) +
  layer(geom="point") + layer(geom="smooth")
> ggplot(dsmall, aes(carat, price)) +
  geom_point() + geom_smooth()
> qplot(carat, price, data=dsmall, geom="point") +
  geom_smooth()
```

Low-level drawing functions are equivalent to adding layers in ggplot2

Base function	ggplot2 layer
<code>curve()</code>	<code>geom_curve()</code>
<code>lines()</code>	<code>geom_line()</code>
<code>points()</code>	<code>geom_point()</code>
<code>polygon()</code>	<code>geom_polygon()</code>
<code>curve()</code>	<code>geom_curve()</code>
<code>segments()</code>	<code>geom_segment()</code>
<code>text()</code>	<code>geom_text()</code>
<code>abline(lm($y \sim x$))</code>	<code>geom_smooth(method='lm')</code>
<code>lines(density(x))</code>	<code>geom_density()</code>
<code>lines(loess(x, y))</code>	<code>geom_smooth()</code>

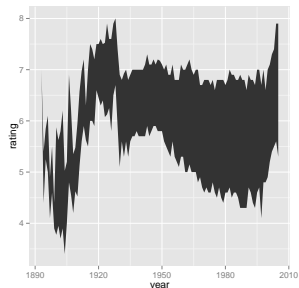
On using transformed data



```
d <- ggplot(diamonds, aes(carat)) + xlim(0, 3)
d + stat_bin(aes(size = ..density..),
             binwidth=0.1, geom="point")
```

Displaying uncertainty

```
data(movies)
m <- ggplot(movies,
            aes(year, rating))
iqr <- function(x, ...) {
  qs <- quantile(as.numeric(x),
                 c(0.25, 0.75),
                 na.rm = T)
  names(qs) <- c("ymin", "ymax")
  qs
}
m + stat_summary(fun.data = "iqr",
                 geom="ribbon")
```



Assignment: Displaying uncertainty

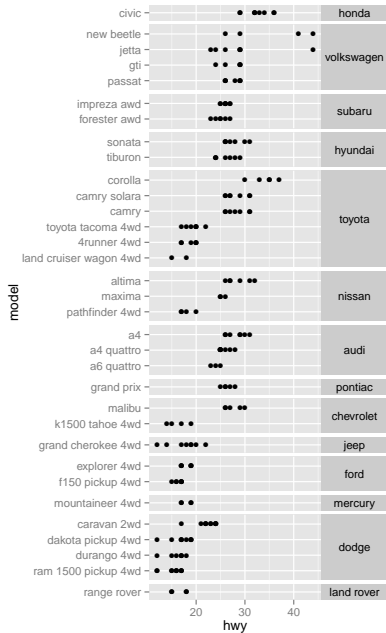
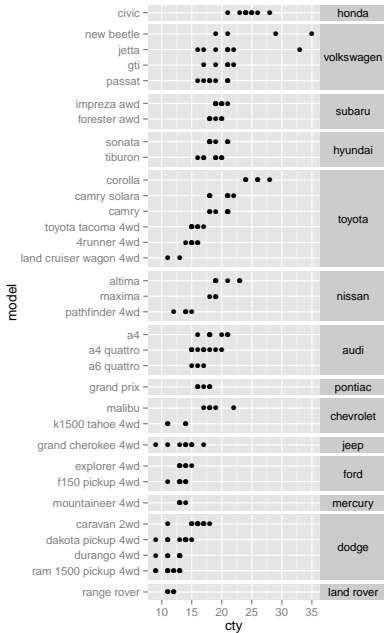
- ▶ Run the additional code for the `movie` data set that is in the script and compare the various results.
- ▶ Use this code to investigate the uncertainty in the diamonds data

More information on `ggplot2` is available on

<http://ggplot2.org/>

including extensive documentation.

Example: The car data set



Example: The car data set

```
data(mpg)

mpg2 <- subset(mpg, cyl != 5 & drv %in% c("4","f"))

mpg3 <- within(mpg2, {
  model <- reorder(model, cty)
  manufacturer <- reorder(manufacturer, -cty)
})

models <- qplot(cty, model, data=mpg3)
models + facet_grid(manufacturer~., scales="free", space="free")
      + theme(strip.text.y=element_text(angle=360))
ggsave("fig//cars1.pdf", width=5, height=8)

models2 <- qplot(hwy, model, data=mpg3)
models2 + facet_grid(manufacturer~., scales="free", space="free")
      + theme(strip.text.y=element_text(angle=360))
ggsave("fig//cars2.pdf", width=5, height=8)
```