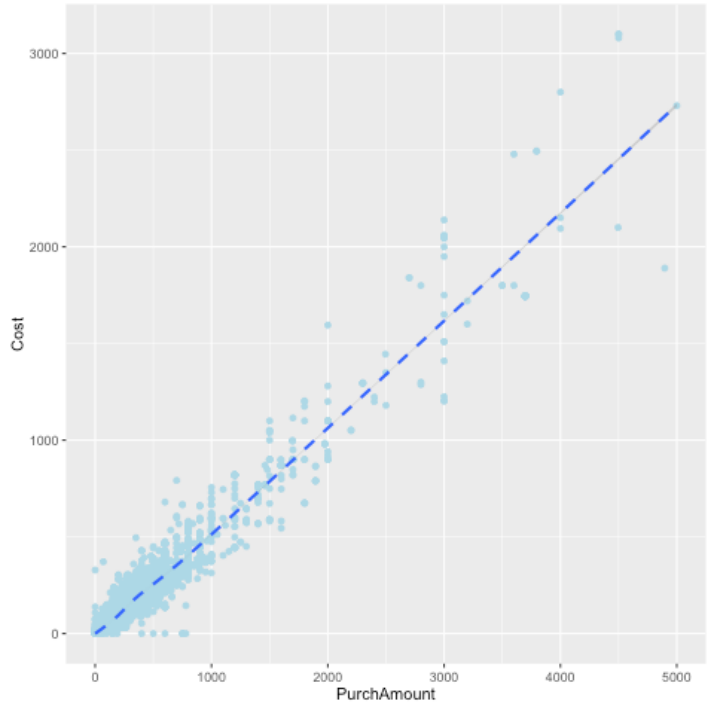
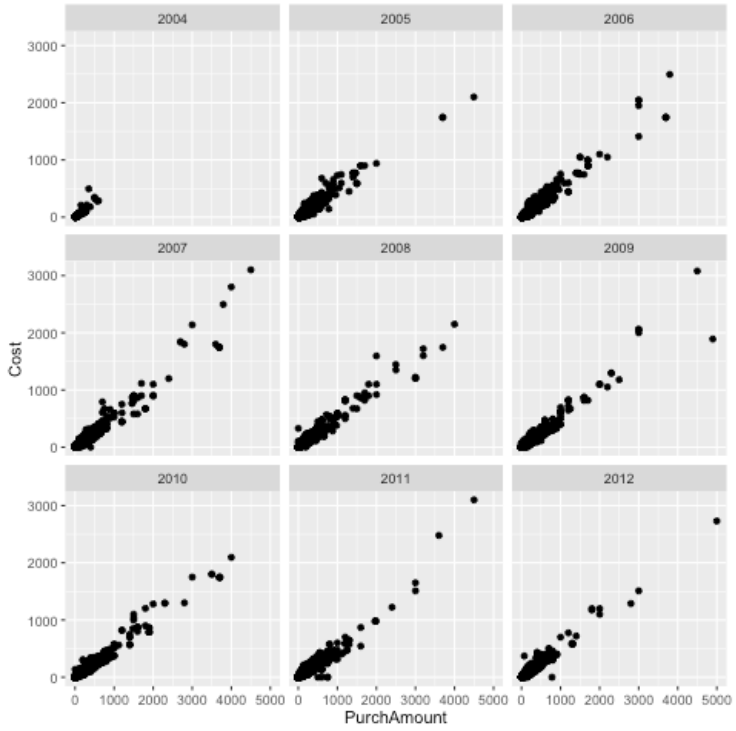
Very advanced plotting options

Overlay plots on the same axes (i.e., overlay geoms)




```
ggplot(myData, aes(PurchAmount, Cost, color=Cost)) + …… +
                geom_point(color="lightblue") +
                geom_smooth(linetype="dashed")
```

Facets split up your data by one or more variables and plot the subsets of data together

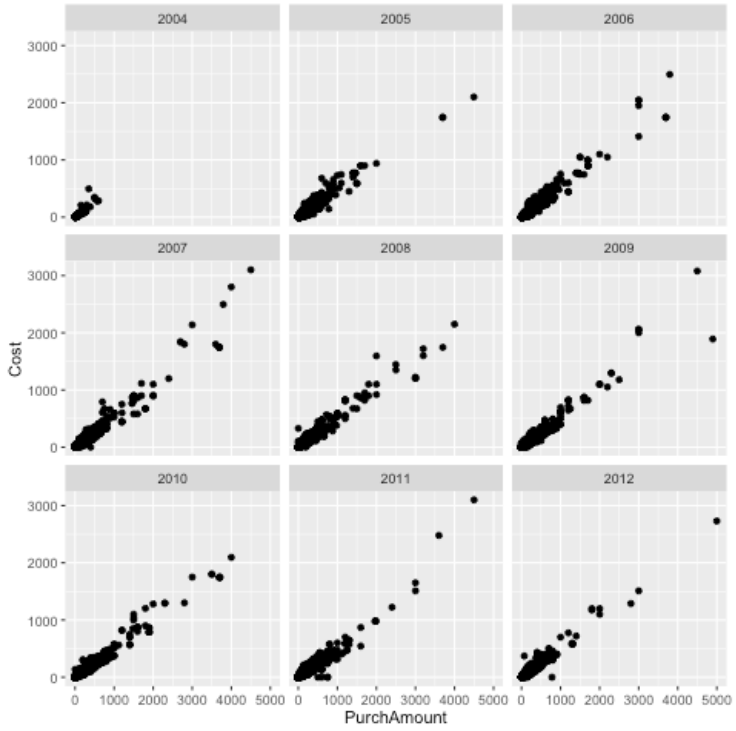



```
myData[,Year:= year(TransDate)]
ggplot(myData, aes(PurchAmount, Cost)) + geom_point() +
  facet_wrap(~Year, ncol=3) + theme_few()
```

1 Formula (~ -operator) specifying variables to facet by

2 Function `year()` from the `lubridate` package extracts the year from a date

Facets split up your data by one or more variables and plot the subsets of data together

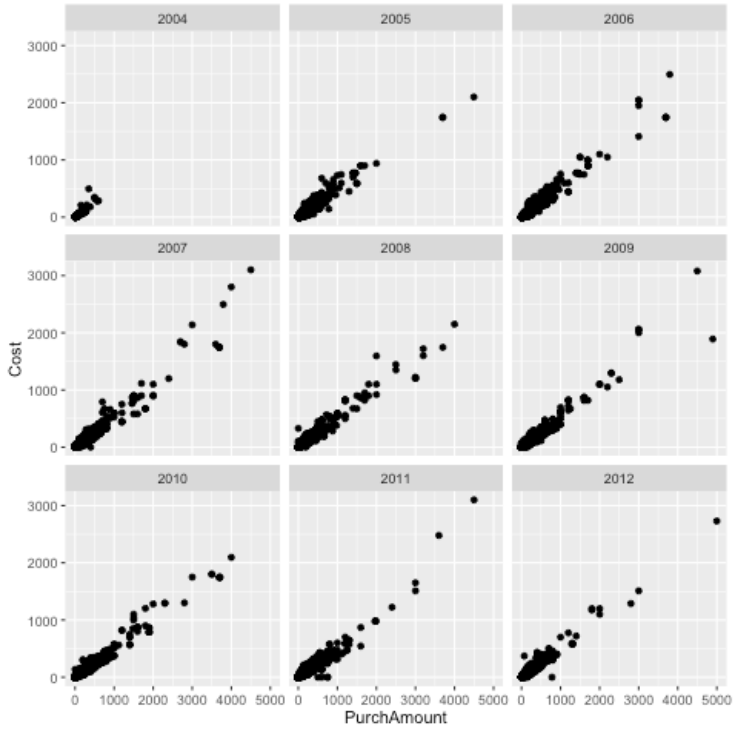



```
myData[,Year:= year(TransDate)]
ggplot(myData, aes(PurchAmount, Cost)) + geom_point() +
  facet_wrap(~Year, ncol=3) + theme_few()
```

Function `year()` from the `lubridate` package extracts the year from a date (2)

Formula (~ -operator) specifying variables to facet by (1)

Facets split up your data by one or more variables and plot the subsets of data together

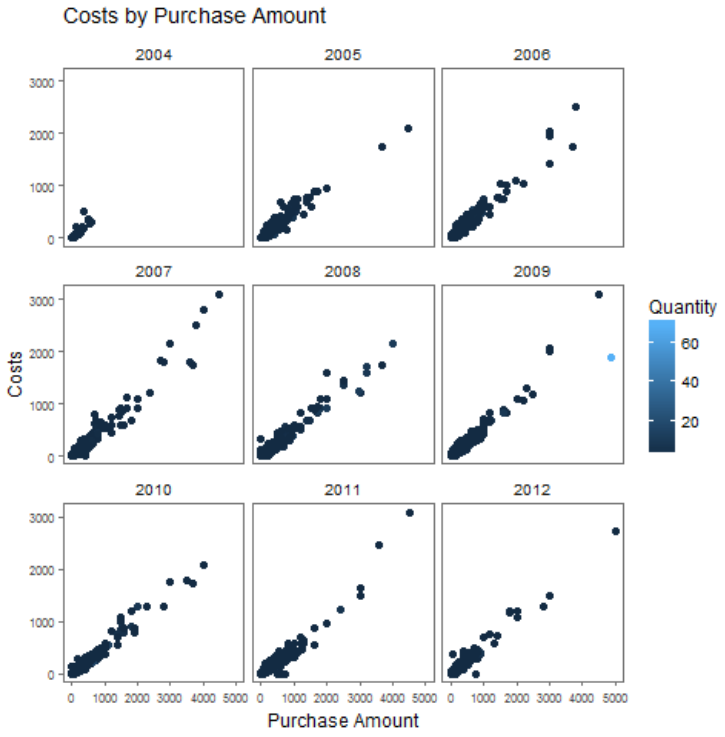



```
myData[,Year:= year(TransDate)]
ggplot(myData, aes(PurchAmount, Cost)) + geom_point() +
  facet_wrap(~Year, ncol=3) + theme_few()
```

2 Function `year()` from the `lubridate` package extracts the year from a date

1 Formula (~ -operator) specifying variables to facet by

With facets you can plot up to four dimensions in one single figure

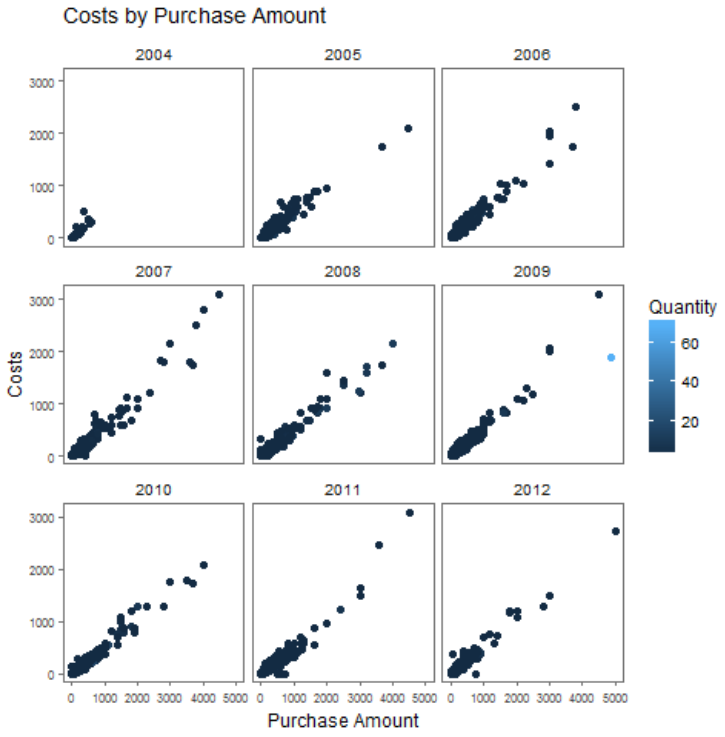



```
myData[,Year:= year(TransDate)]
ggplot(myData, aes(PurchAmount, Cost, color=Quantity)) + geom_point()
+.....+ facet_wrap(facets=~Year, ncol=3) + theme_few() +
theme(axis.text=element_text(size=8),text=element_text(size=12))
```

1 Add a 4th dimension

2 Adjust layout to fit text in plot

With facets you can plot up to four dimensions in one single figure

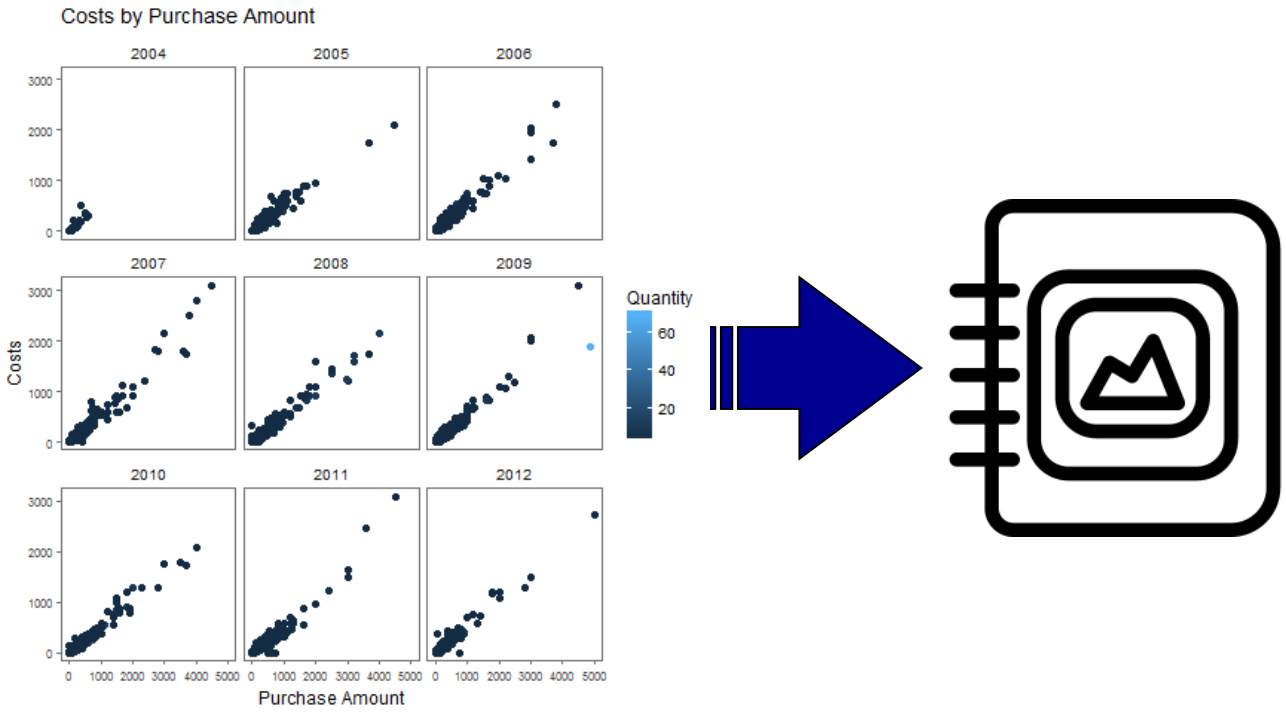



```
                    myData[,Year:= year(TransDate)]
ggplot(myData, aes(PurchAmount, Cost, color=Quantity)) + geom_point()
          +.....+ facet_wrap(facets=~Year, ncol=3) + theme_few() +
  theme(axis.text=element_text(size=8),text=element_text(size=12))
```

Add a 4th dimension 1

Adjust layout to fit text in plot 2

Save your ggplot with `ggsave()` or via the point-and-click method

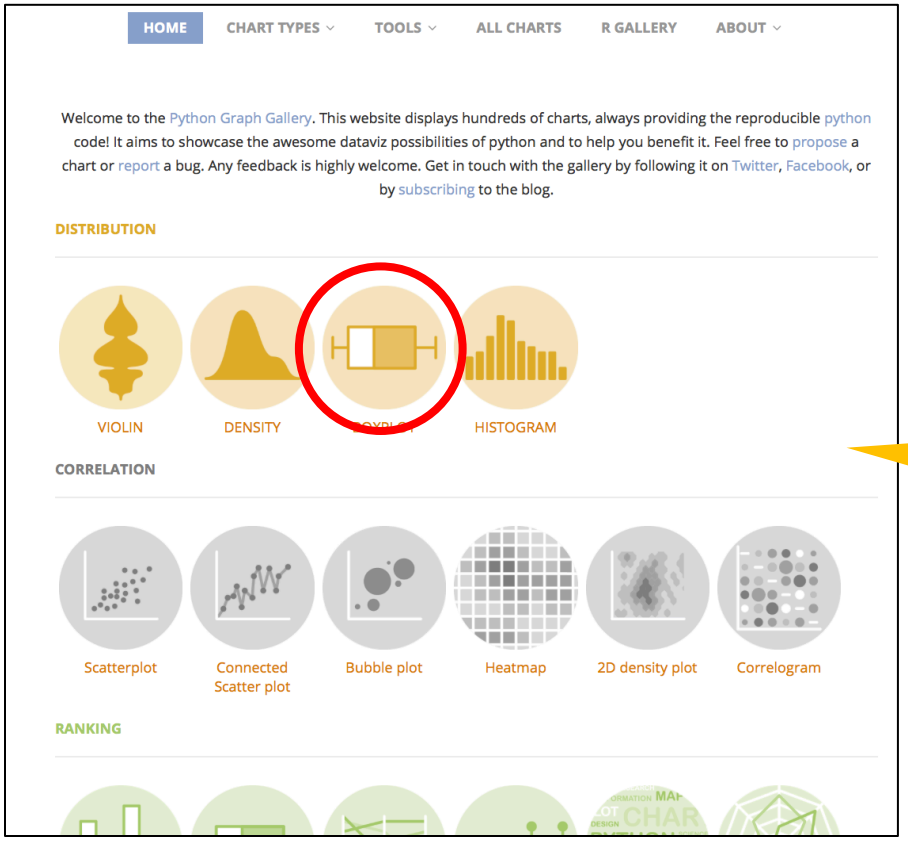



```
ggsave("myPlot.png")
```

Saves last plot

Visualize data with https://r-graph-gallery.com

Step 1: Explore functions



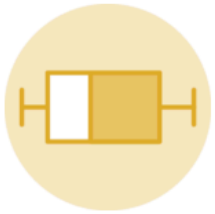
Get a smart overview over possibilities to visualize data with reproducible R code at https://r-graph-gallery.com

Get inspired by the graphs on the home page

Visualize data with https://r-graph-gallery.com

Step 2: Understand your function

BOXPLOT



Boxplot is probably one of the most common type of graphic. It gives a nice **summary** of one or several **numeric variables**. The line that divides the box into 2 parts represents the **median** of the data. The end of the box shows the upper and lower **quartiles**. The extreme lines shows the highest and lowest value excluding **outliers**. Note that boxplot hide the number of values existing behind the variable. Thus, it is highly advised to print the number of observation, add unique observation with jitter or use a violinplot if you have many observations.

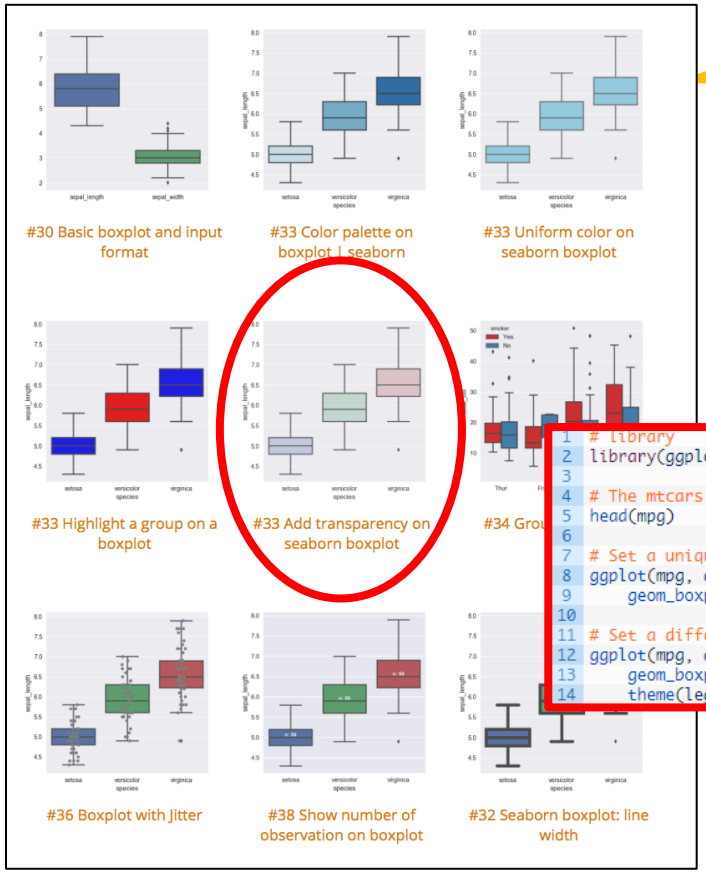
Input format

Variable 1	Group
1.3	A

Find a short function summary and for what data the plot is suited

Visualize data with https://r-graph-gallery.com

Step 3: Choose your favorite and get the code



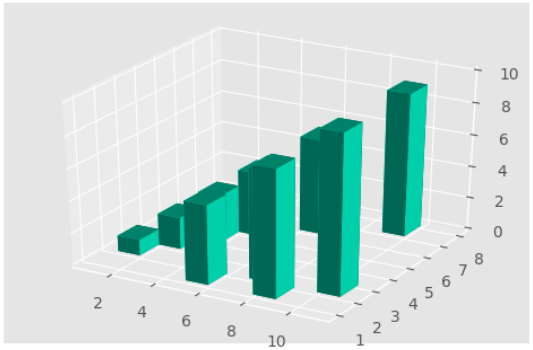
1 Choose one of the templates

```r
# library
library(ggplot2)

# The mtcars dataset is proposed in R
head(mpg)

# Set a unique color with fill, colour, and alpha
ggplot(mpg, aes(x=class, y=hwy)) +
    geom_boxplot(color="red", fill="orange", alpha=0.2)

# Set a different color for each group
ggplot(mpg, aes(x=class, y=hwy, fill=class)) +
    geom_boxplot(alpha=0.3) +
    theme(legend.position="none")
```

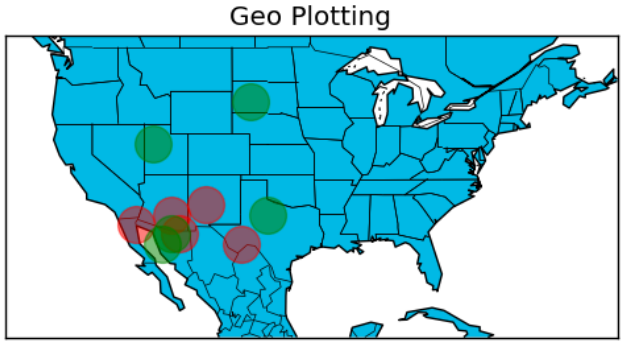
2 Get R code which you can reproduce immediately

Summing up: endless possibilities enable users to create professional data visualization

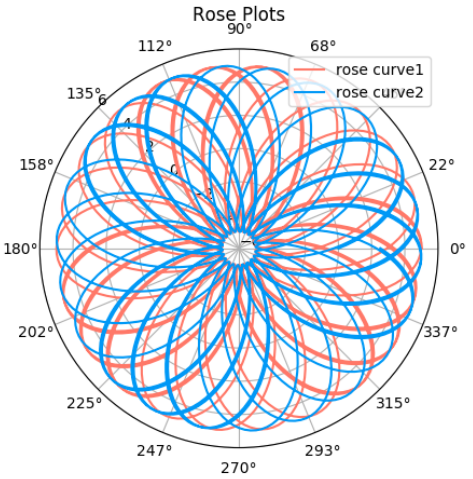
There are (almost) no restrictions in data visualization. Watch the examples and understand why and how to create even better plots and maps!



1 3D barplots



2 "Basemap" functionality with adjusted markersizes



3 Illustration of polar coordinates