
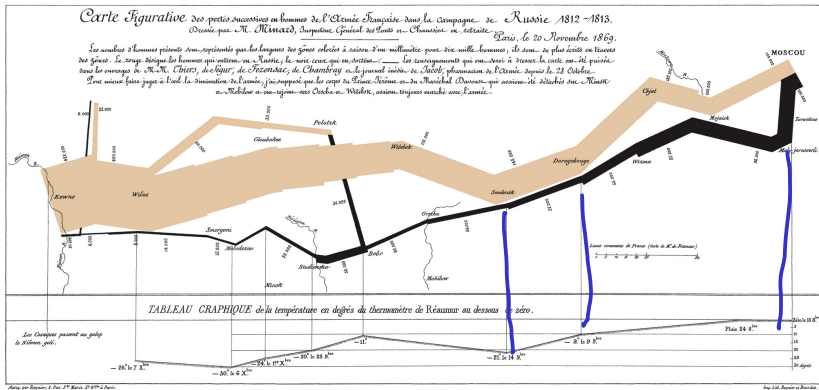


Putting Graphs Together



Introduction

In multivariate data visualizations, it can be good design to place multiple graphs over common axes.



We'll look at data on number of Atlantic hurricanes in our examples.

The data set contains counts of named tropical/subtropical storms, hurricanes and severe hurricanes by year. We'll create these variables:

- **Year:** The Year
- **All:** All named Atlantic storms, hurricane or not.
- **Storms:** All named storms that weren't hurricanes. ←
- **Hurr12:** Category 1 or 2 hurricanes. ←
- **Hurr345:** Category 3-5 hurricanes (the more intense storms). ←

```
hur <- read.csv("data/atlantic_hurricanes.csv") %>%  
  filter(Year <= 2017) %>%  
  mutate(  
    All = Named.Storms,  
    Storms = Named.Storms - Hurricanes,  
    Hurr12 = Hurricanes - Major.Hurricanes,  
    Hurr345 = Major.Hurricanes  
  ) %>%  
  select(Year, All, Storms, Hurr12, Hurr345)
```



Combining Graphs in Base R Graphics

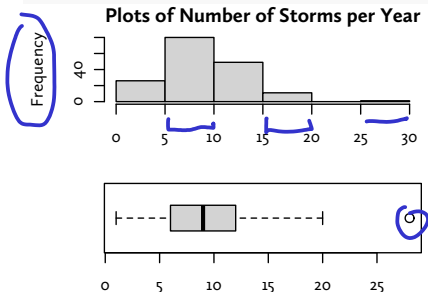
In base R, graphs can be combined with the `par` command and `mfrow` or `mfcol` options.

For example, we might do a box plot and histogram to show the distribution of number of named storms per year.

- `mfrow` and `mfcol` both take an argument `c(nrow, ncol)` and create a grid of graphs.
- They differ on whether the grid is filled in by row first or by column first.
- Default margins often seem too large, and can be set with `mar=c(bot, left, top, right)`.

Two rows, one column.

```
par(mfrow=c(2, 1), mar=c(2, 4, 2, 0))  
hist(hur$All,  
     main="Plots of Number of Storms per Year",  
     xlab="")  
boxplot(hur$All, horizontal=TRUE,  
        xlab="Number of Storms")
```

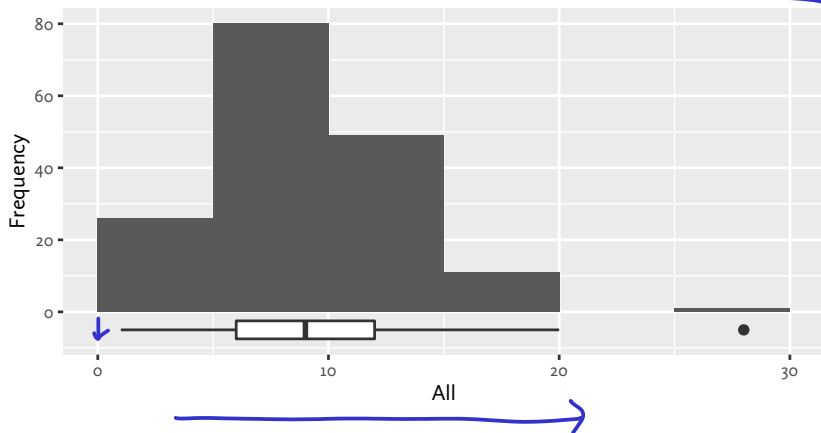




Combining Graphs in ggplot

The `position_nudge` function can help you combine plots in ggplot.

```
ggplot(hur) + ylab("Frequency") +  
→ geom_histogram(aes(x=All), breaks=seq(0, 30, 5)) +  
→ geom_boxplot(aes(x=All), position=position_nudge(x=0, y=-5), width=5)
```



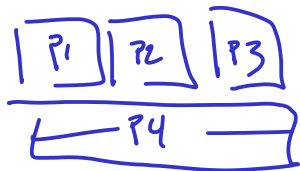
Several R packages allow you to combine plots.

- gridExtra (I've known this the longest.)
 - grid.arrange command
 - Flexible grid, but doesn't align plot areas.
- cowplot
 - plot_grid command
 - Does align plot areas.
- patchwork (My new favorite, I think.)
 - Uses mathematical operators for nice, short code.
 - Add extra design options using plot_layout.
- Each might fail in specific circumstances. Good to have choice.

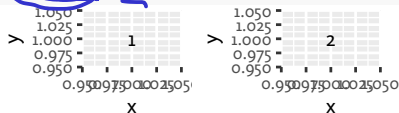
patchwork combines plots using mathematical operators.

- "+" for side-by-side.
- "/" for stacked.
- "|" for side-by-side nested.

— one col.
(p1 | p2 | p3) / p4



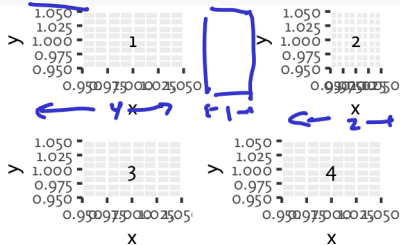
```
library(patchwork)
df <- data.frame(x=1, y=1)
p1 <- ggplot(df, aes(x=x, y=y)) +
  geom_text(label="1")
p2 <- ggplot(df, aes(x=x, y=y)) +
  geom_text(label="2")
p3 <- ggplot(df, aes(x=x, y=y)) +
  geom_text(label="3")
p4 <- ggplot(df, aes(x=x, y=y)) +
  geom_text(label="4")
(p1 + p2) / (p3 + p4)
```



patchwork combines plots using mathematical operators.

- “+” for side-by-side.
- “/” for stacked.
- “|” for side-by-side nested.
- plot_spacer() add empty space.
- plot_layout adds options.
 - You can use `nrow` and `ncol` with just “+” operator to get a grid.
 - `widths` and `heights` change scales of columns and rows.

```
(p1 + plot_spacer() + p2 +  
  plot_layout(widths=c(4, 1, 2))) /  
(p3 | p4)
```

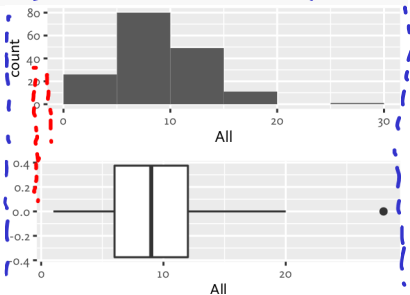


```
p1 + p2 + p3 + p4 +  
  plot_layout(ncol=2)
```

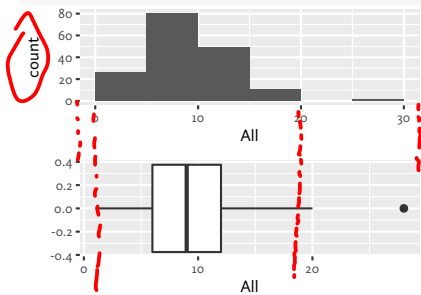


A comparison of grid.arrange and patchwork.

```
library(gridExtra)
g1 <- ggplot(hur) +
  geom_histogram(aes(x=All),
    breaks=seq(0, 30, 5))
g2 <- ggplot(hur) +
  geom_boxplot(aes(x=All))
grid.arrange(g1, g2, nrow=2)
```



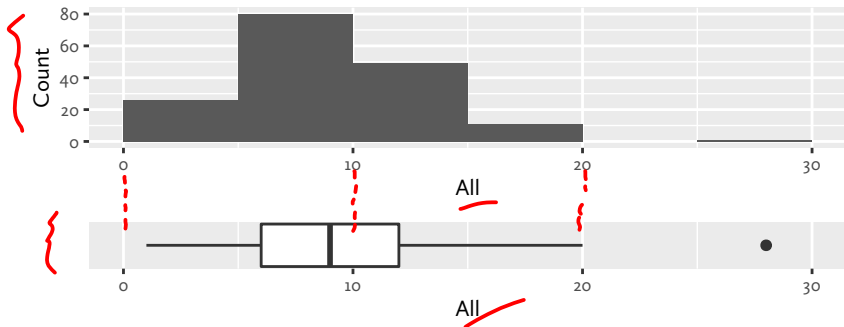
```
library(patchwork)
g1 <- ggplot(hur) +
  geom_histogram(aes(x=All),
    breaks=seq(0, 30, 5))
g2 <- ggplot(hur) +
  geom_boxplot(aes(x=All))
g1 / g2
```



You can edit the coordinate limits to align graphs exactly.
(Remember editing limits in scale can drop data points.)

scale_x_continuous

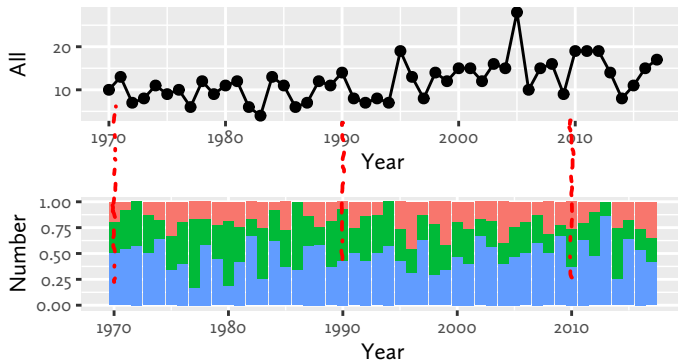
```
library(patchwork)
g1 <- ggplot(hur) + geom_histogram(aes(x=All), breaks=seq(0, 30, 5)) +
→ coord_cartesian(xlim=c(0, 30)) + ylab("Count")
g2 <- ggplot(hur) + geom_boxplot(aes(x=All)) +
→ coord_cartesian(xlim=c(0, 30)) + scale_y_continuous(breaks=NULL)
g1 / g2 + plot_layout(heights=c(3,1))
```



Another example pairs a line graph of total storms vs. a bar graph showing distribution of storm types.

A first pass gets the basics down.

```
g1 <- hur %>% filter(Year >= 1970) %>% ggplot() +  
  geom_point(aes(x=Year, y=All)) +  
  geom_line(aes(x=Year, y=All))  
g2 <- hur %>% filter(Year >= 1970) %>% select(Year, Hurr345, Hurr12, Storms) %>%  
  gather(key="Type", value="Number", -Year, factor_key=TRUE) %>%  
  ggplot() + geom_col(aes(x=Year, y=Number, fill=Type), position="fill")  
g1 / g2
```



	All	Strong	H2	4th
1970	20	10	7	3
1971	18	6	12	0
.

gather

Year	Type	Num
1970	Stirn	10
1970	H12	7
1970	H345	3
:	:	:
:	:	:

Type

Hurr345
Hurr12
Storms

1st
2nd
3rd

Another example pairs a line graph of total storms vs. a bar graph showing distribution of storm types.

```
# Now we'll worry about the details.
```

```
g1 <- g1 + xlab(NULL) +  
coord_cartesian(xlim=c(1969, 2018)) +  
ggtitle("Storm Number and Distribution by Year")  
g2 <- g2 + ylab("Percent") +  
scale_fill_viridis_d(end=0, begin=1) +  
coord_cartesian(xlim=c(1969, 2018))  
g1 / g2 + plot_layout(heights=c(2,1), guides="collect")
```

