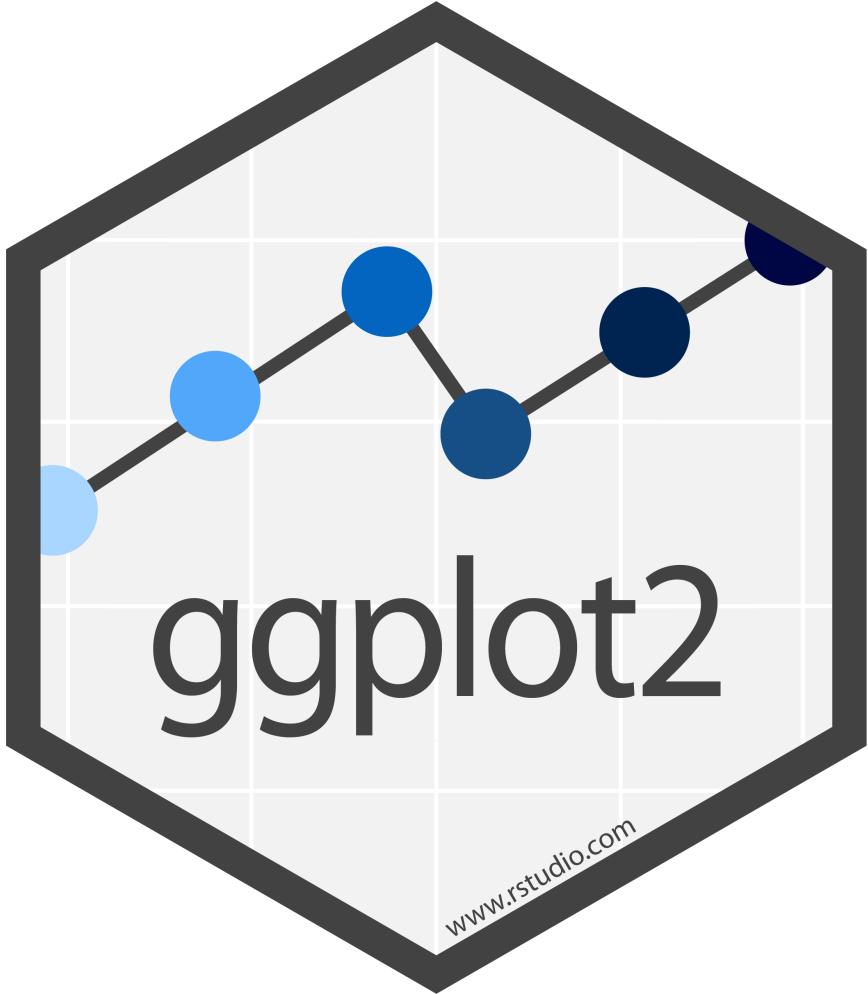


Visualization

Colin Rundel

2019-10-14



The Grammar of Graphics

- Visualisation concept created by Leland Wilkinson (1999)
 - to define the basic elements of a statistical graphic
- Adapted for R by Hadley Wickham (2009)
 - consistent and compact syntax to describe statistical graphics
 - highly modular as it breaks up graphs into semantic components
- It is not meant as a guide to which graph to use and how to best convey your data (more on that later).

Terminology

A statistical graphic is a...

- mapping of **data**
- which may be **statistically transformed** (summarised, log-transformed, etc.)
- to **aesthetic attributes** (color, size, xy-position, etc.)
- using **geometric objects** (points, lines, bars, etc.)
- and mapped onto a specific **facet** and **coordinate system**

Anatomy of a ggplot call

```
ggplot(  
  data = [dataframe],  
  mapping = aes(  
    x = [var_x], y = [var_y],  
    color = [var_for_color],  
    shape = [var_for_shape],  
    ...  
  )  
) +  
  geom_[some_geom]()  
  mapping = aes(  
    color = [var_for_geom_color],  
    ...  
  )  
) +  
... # other geometries  
scale_[some_axis]_[some_scale]() +  
facet_[some_facet]([formula]) +  
... # other options
```

Diamonds

```
set.seed(20191014)
diamonds = sample_n(ggplot2::diamonds, 1000)
diamonds

## # A tibble: 1,000 x 10
##   carat cut      color clarity depth table price     x     y     z
##   <dbl> <ord>    <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 0.39 Ideal     I     SI1     62.4    54    534  4.67  4.69  2.92
## 2 0.41 Premium   E     SI1     61.1    62    904  4.83  4.76  2.93
## 3 0.51 Very Good D     VS2     62.2    57   1678  5.11  5.14  3.19
## 4 0.34 Ideal     G     VS2     62.4    57    517  4.46  4.48  2.79
## 5 1.26 Very Good G     SI1     63.3    58   7910  6.88  6.84  4.34
## 6 0.43 Good      F     SI1     63.6    53    948  4.83  4.79  3.06
## 7 0.93 Premium   I     VS2     62.7    61   4073  6.24  6.14  3.88
## 8 1.14 Premium   I     SI1     62.2    58   4788  6.7   6.67  4.16
## 9 0.36 Fair       F     VS1     55.3    67    810  4.79  4.72  2.63
## 10 0.33 Ideal     F    VVS2     61.4   57    824  4.43  4.46  2.73
## # ... with 990 more rows
```

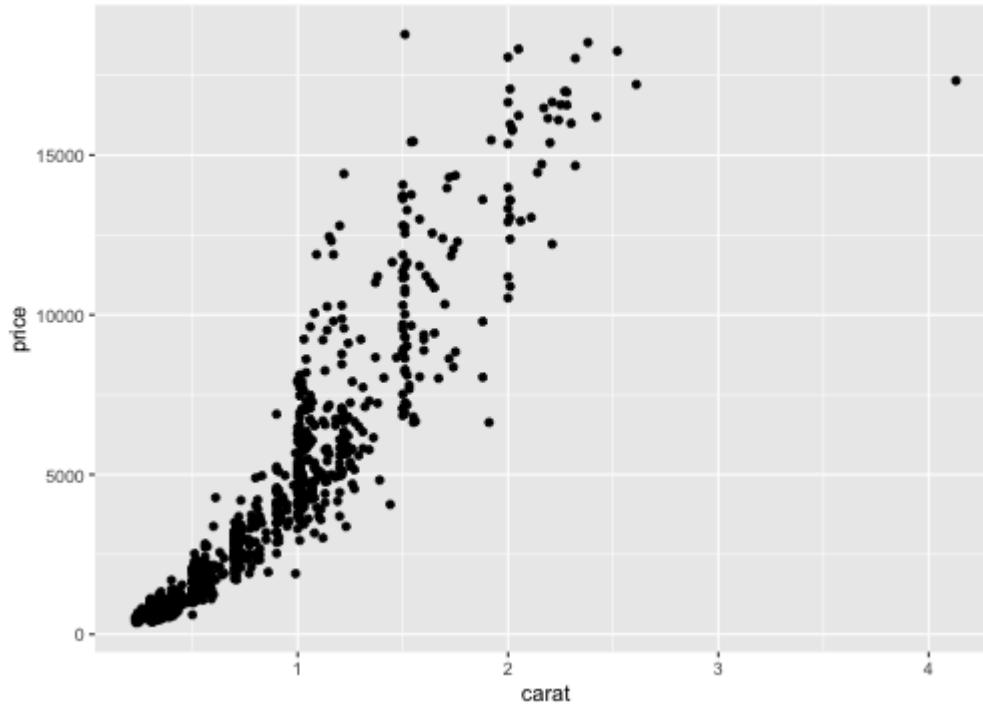
```
head(diamonds$cut)
```

```
## [1] Ideal      Premium   Very Good  Ideal      Very Good Good
## Levels: Fair < Good < Very Good < Premium < Ideal
```

```
head(diamonds$color)
```

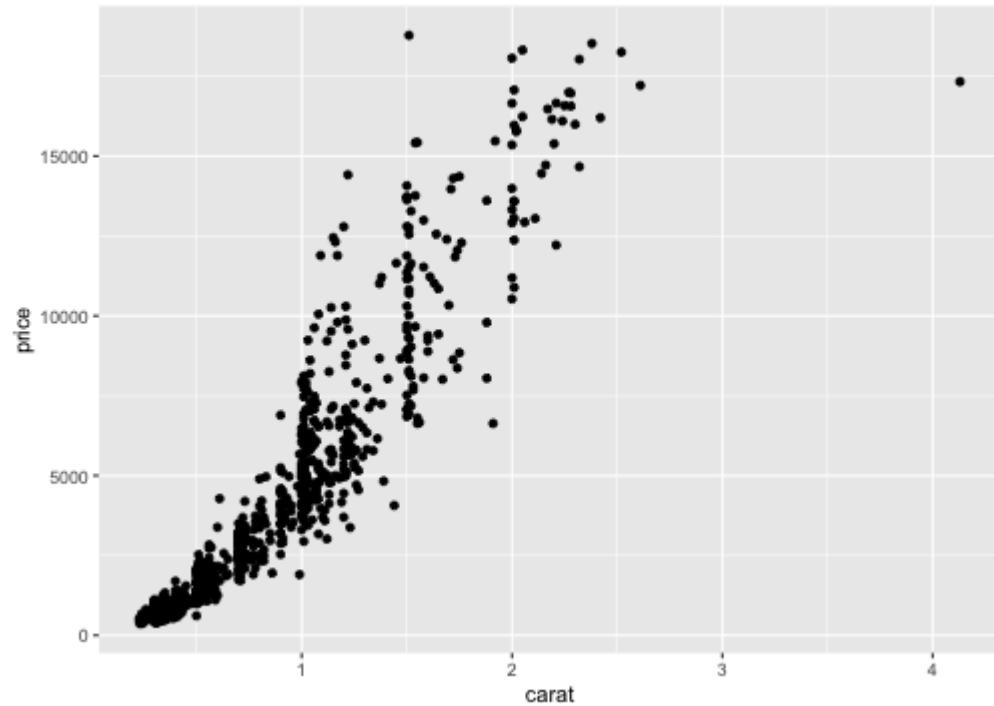
```
## [1] I E D G G F
## Levels: D < E < F < G < H < I < J
```

Example 1

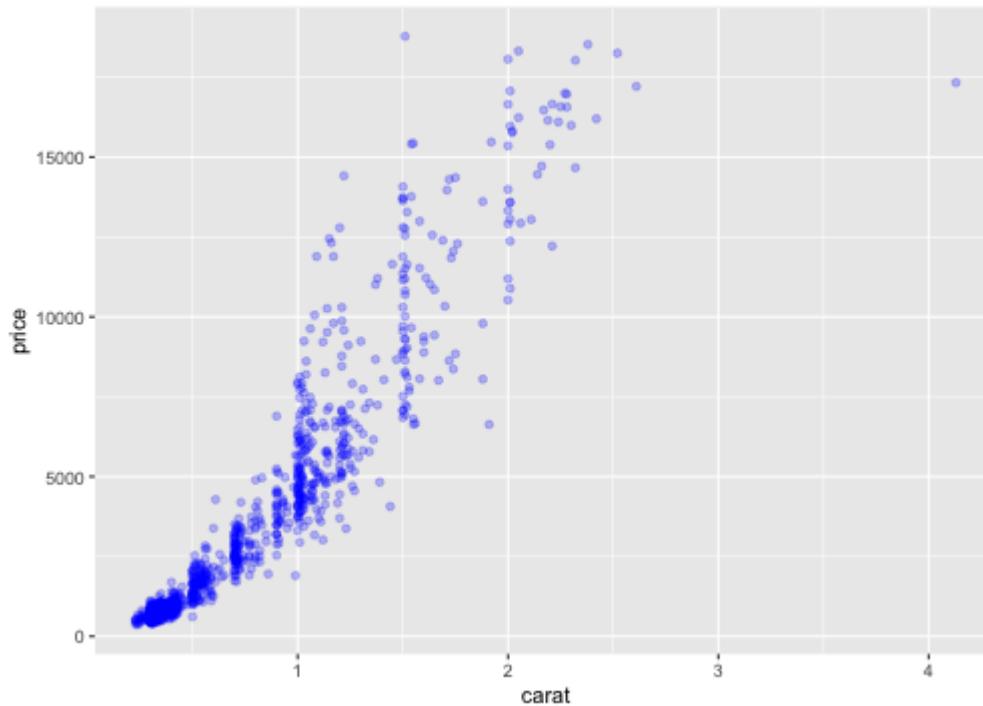


- Which data is used as an input?
- Are the variables statistically transformed before plotting?
- What geometric objects are used to represent the data?
- What variables are mapped onto which aesthetic attributes?
- What type of scales are used to map data to aesthetics?

```
ggplot(data = diamonds, aes(x = carat, y = price)) +  
  geom_point()
```

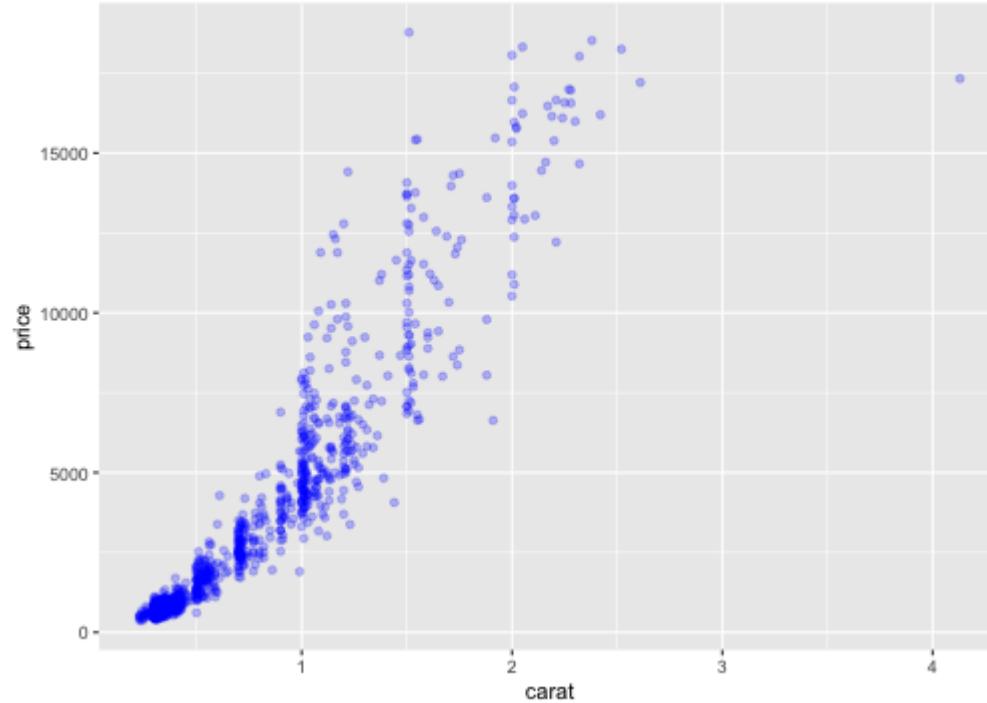


Altering aesthetics

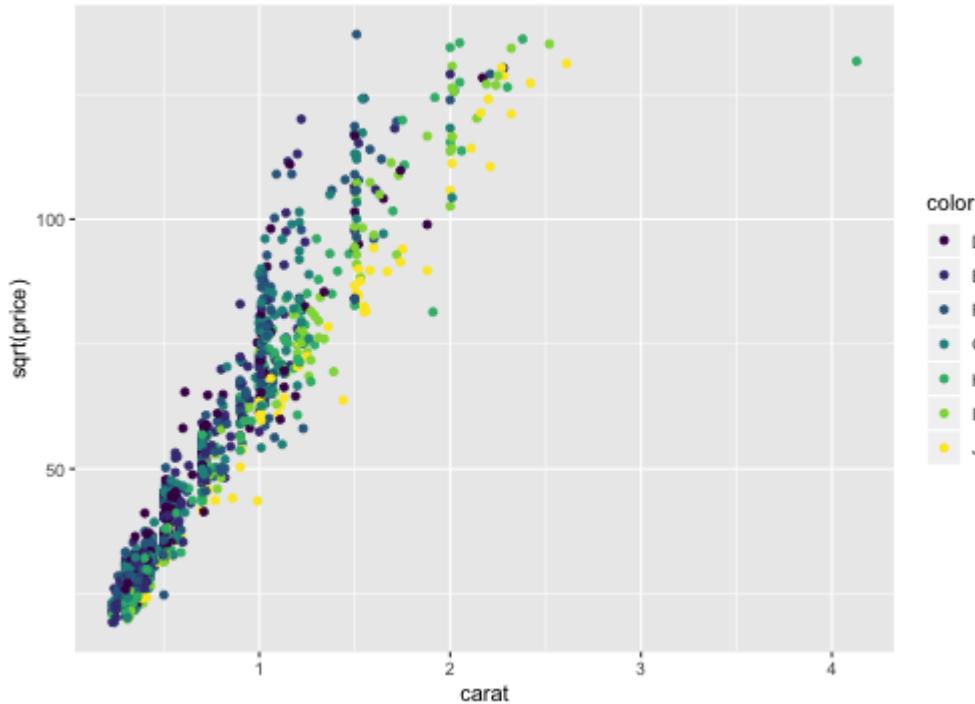


- How did the plot change?
- Are these changes based on data or are the changes based on stylistic choices for the geometric objects?

```
ggplot(data = diamonds, aes(x = carat, y = price)) +  
  geom_point(alpha = 0.25, color = "blue")
```

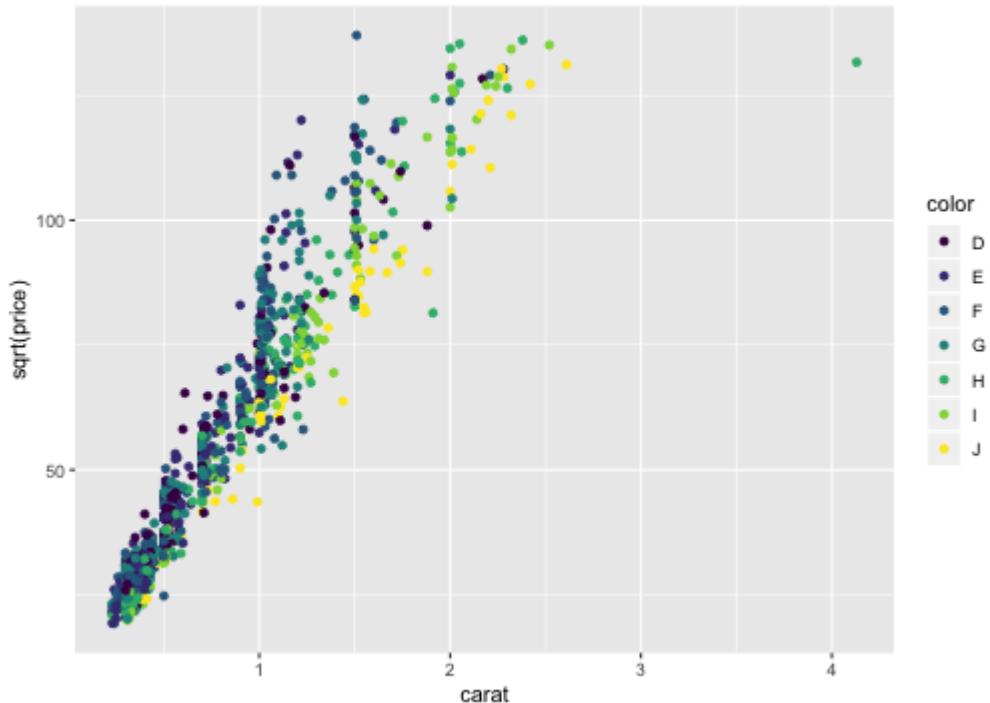


Example 2

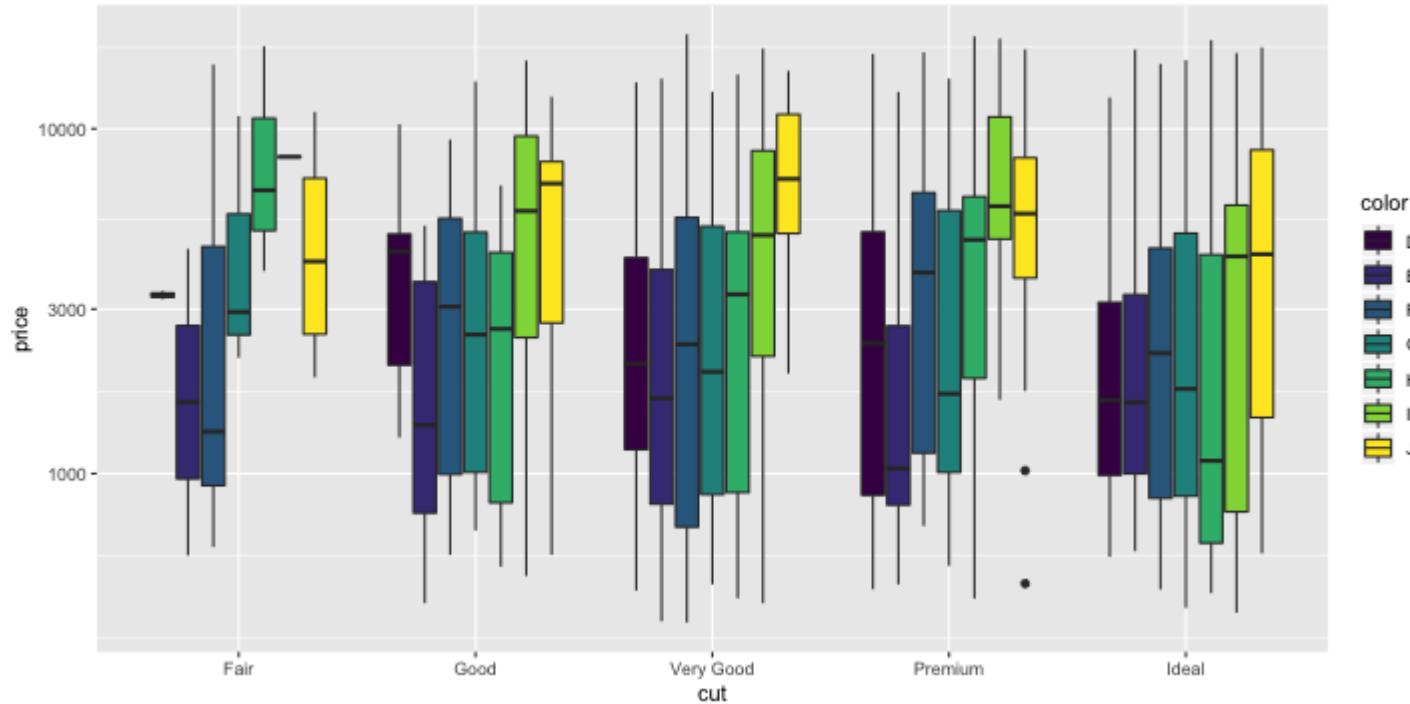


- Which data is used as an input?
- Are the variables statistically transformed before plotting?
- What geometric objects are used to represent the data?
- What variables are mapped onto which aesthetic attributes?
- What type of scales are used to map data to aesthetics?

```
ggplot(data = diamonds, aes(x = carat, y = sqrt(price), color = color)) +  
  geom_point()
```

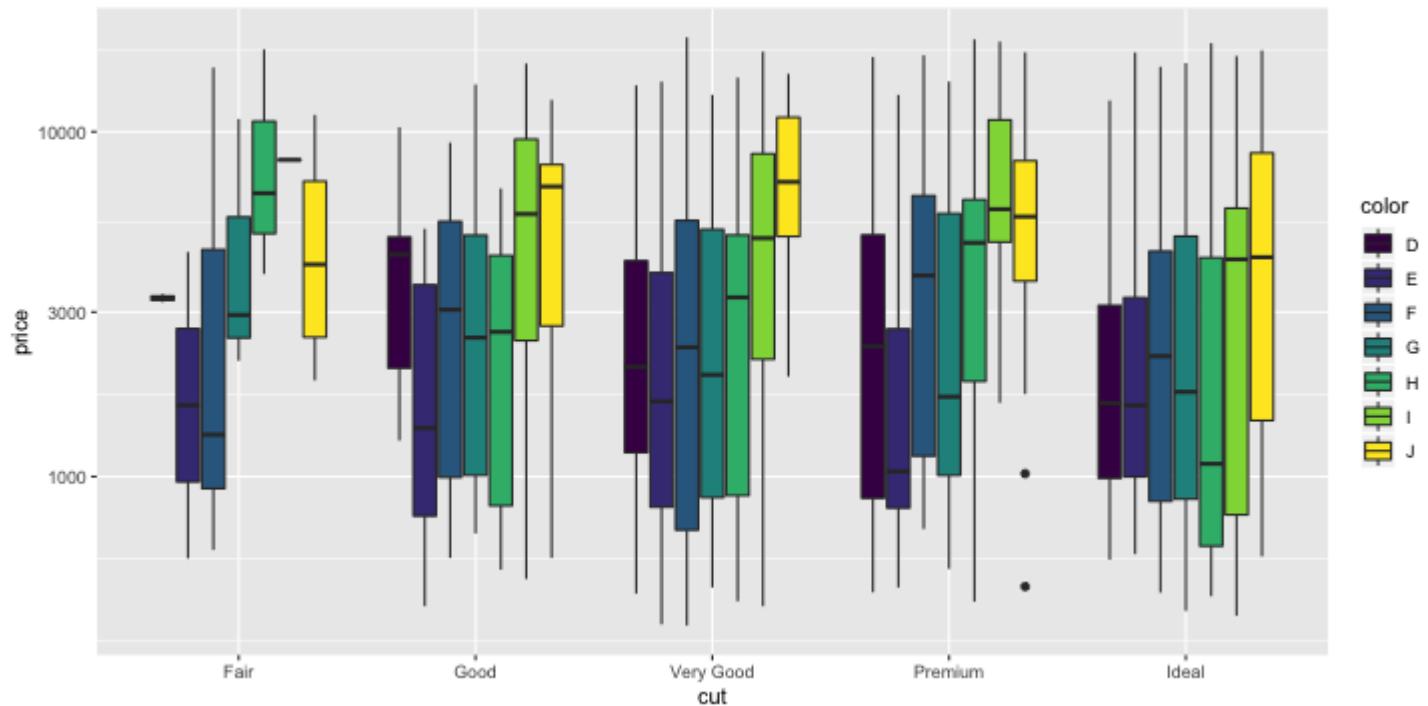


Example 3

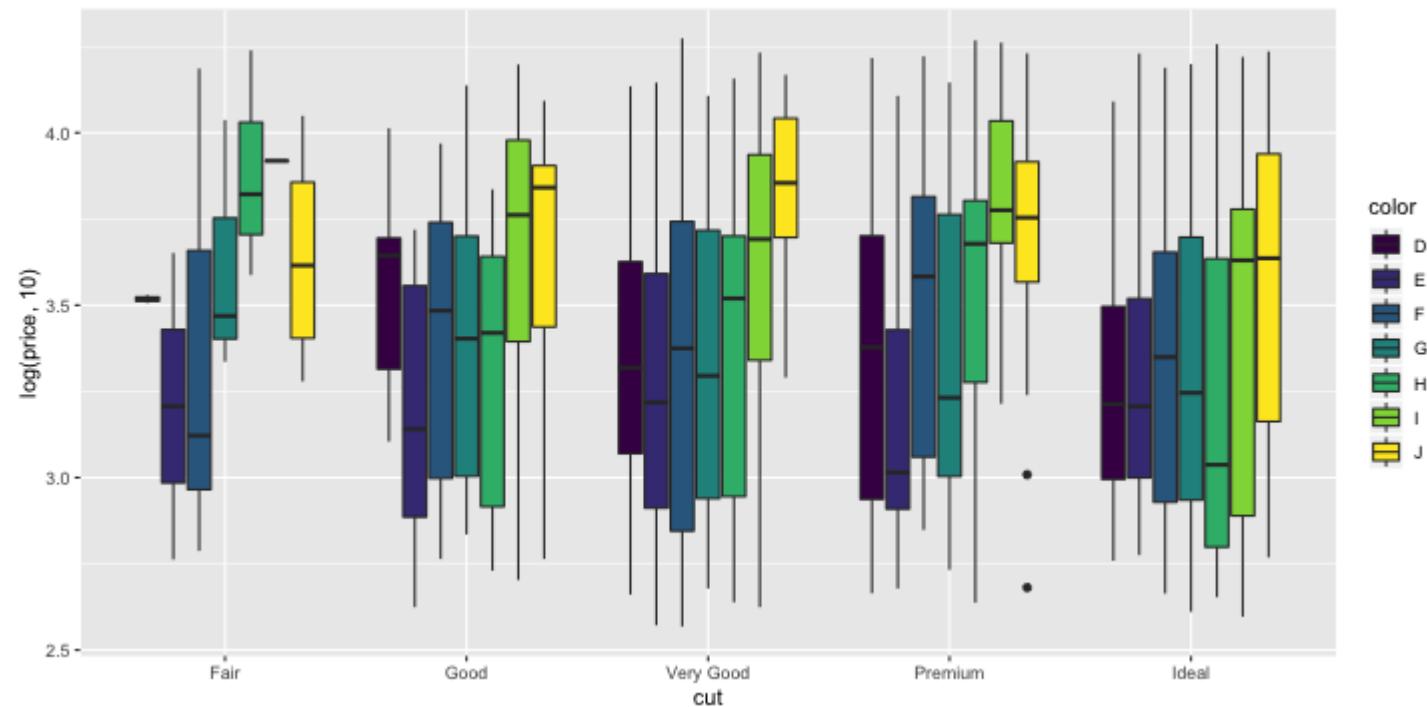


- Which data is used as an input?
- Are the variables statistically transformed before plotting?
- What geometric objects are used to represent the data?
- What variables are mapped onto which aesthetic attributes?
- What type of scales are used to map data to aesthetics?

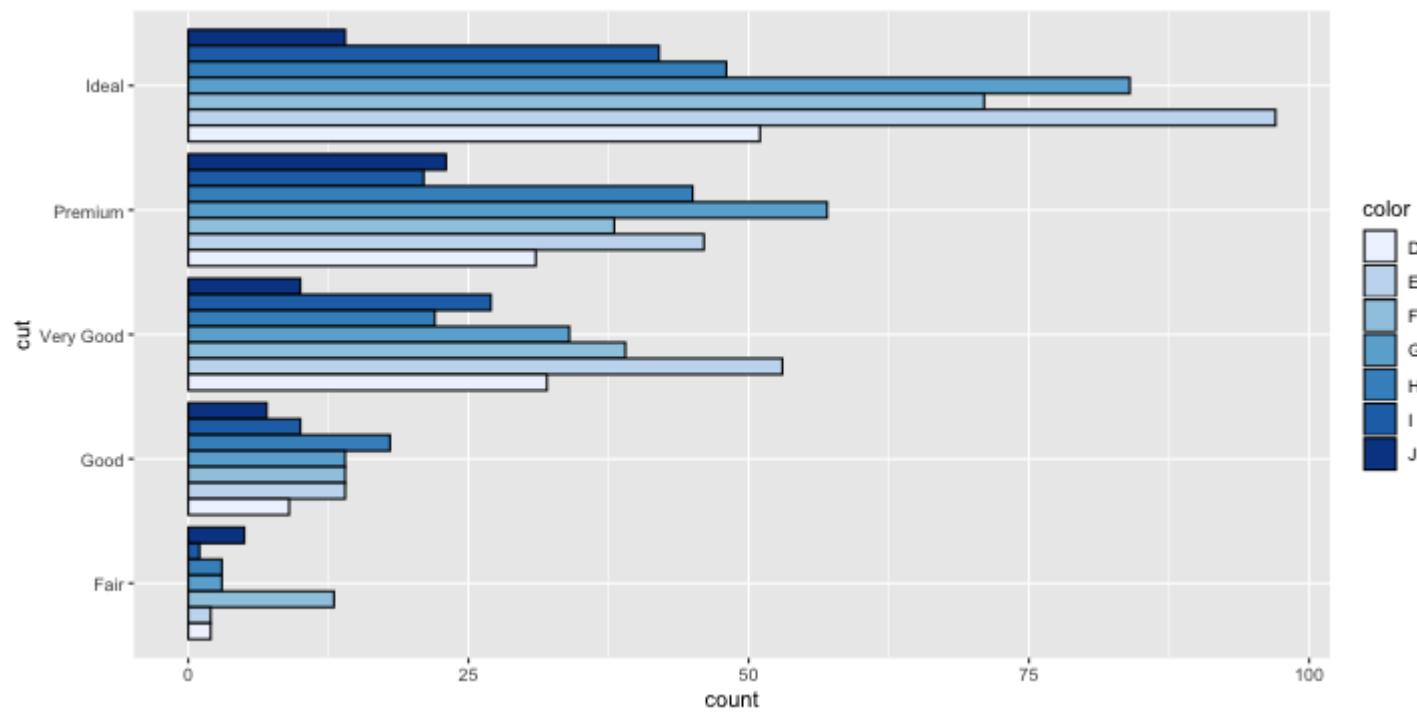
```
ggplot(data = diamonds, aes(x = cut, y = price, fill = color)) +  
  geom_boxplot() +  
  scale_y_log10()
```



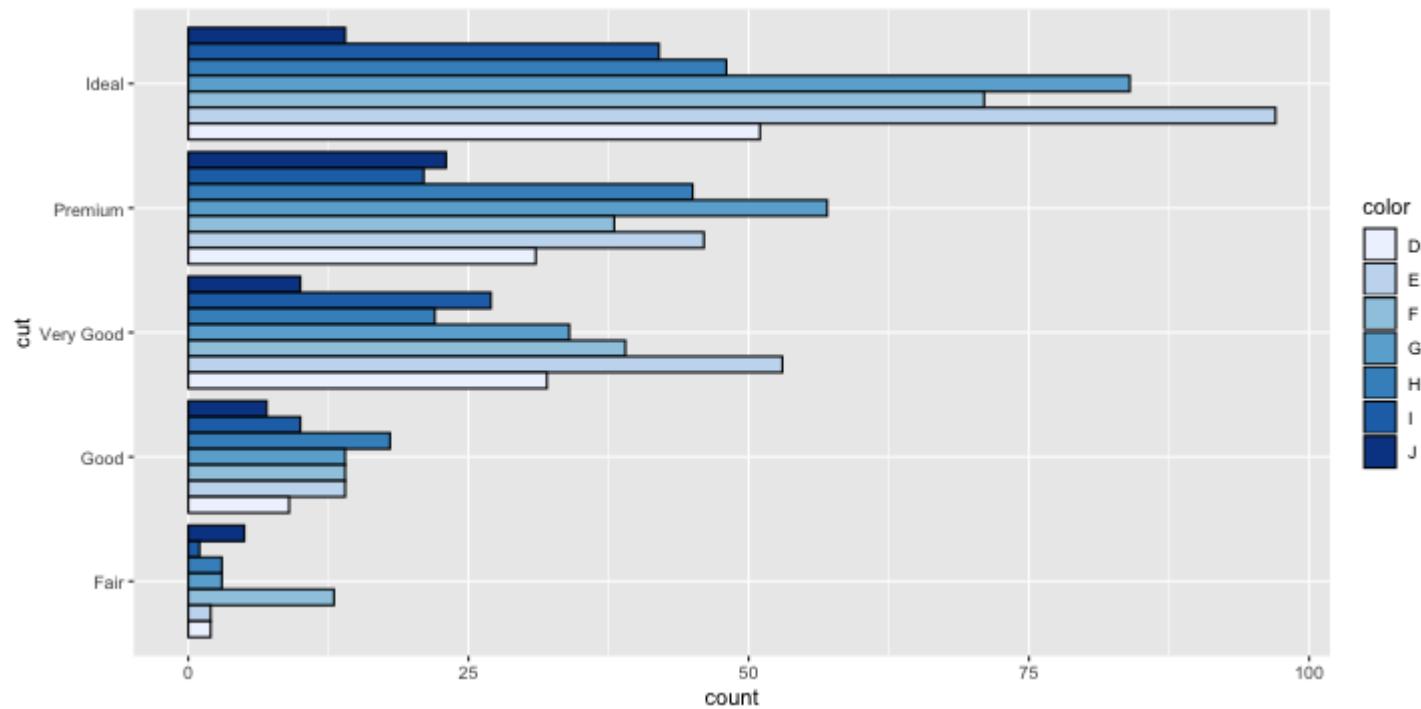
```
ggplot(data = diamonds, aes(x = cut, y = log(price,10), fill = color)) +  
  geom_boxplot() +  
  scale_y_continuous()
```



Example 4

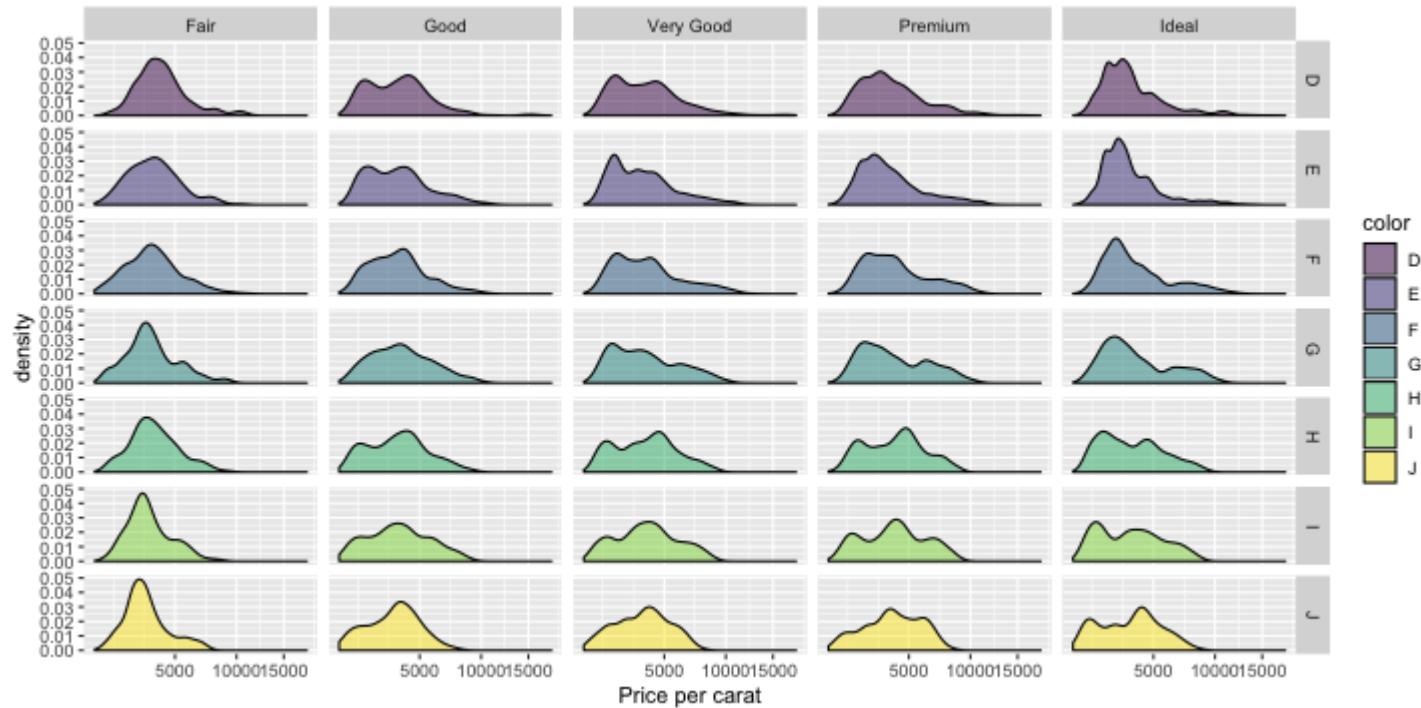


```
ggplot(data = diamonds, aes(x = cut, fill=color)) +
  geom_bar(position = "dodge", color = "black") +
  coord_flip() +
  scale_fill_brewer(palette = "Blues")
```



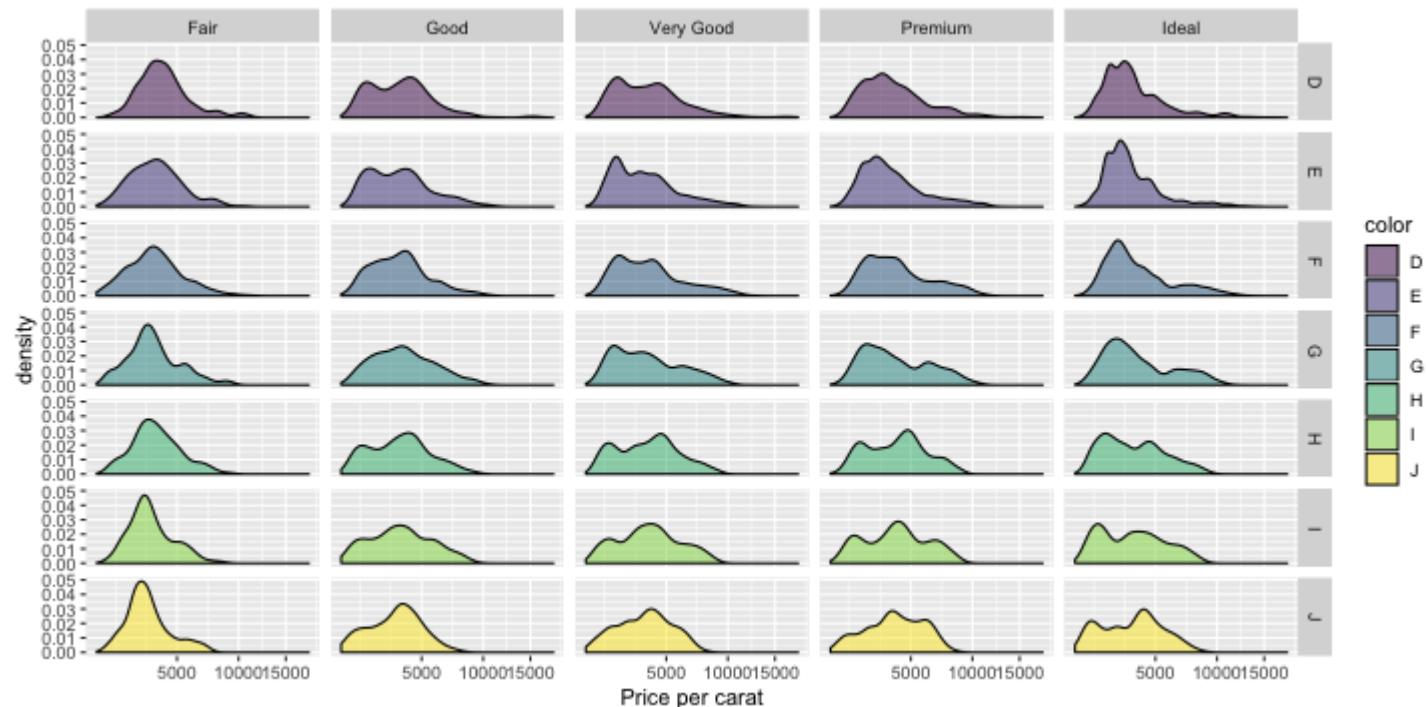
- Which data is used as an input?
- Are the variables statistically transformed before plotting?
- What geometric objects are used to represent the data?
- What variables are mapped onto which aesthetic attributes?
- What type of scales are used to map data to aesthetics?

Example 5



- Which data is used as an input?
- Are the variables statistically transformed before plotting?
- What geometric objects are used to represent the data?
- What variables are mapped onto which aesthetic attributes?
- What type of scales are used to map data to aesthetics?

```
ggplot(data = ggplot2::diamonds, aes(x = price/carat, fill=color)) +  
  geom_density(alpha=0.5) +  
  facet_grid(rows = vars(color), cols = vars(cut)) +  
  scale_x_sqrt() +  
  labs(x = "Price per carat")
```



More ggplot2 resources

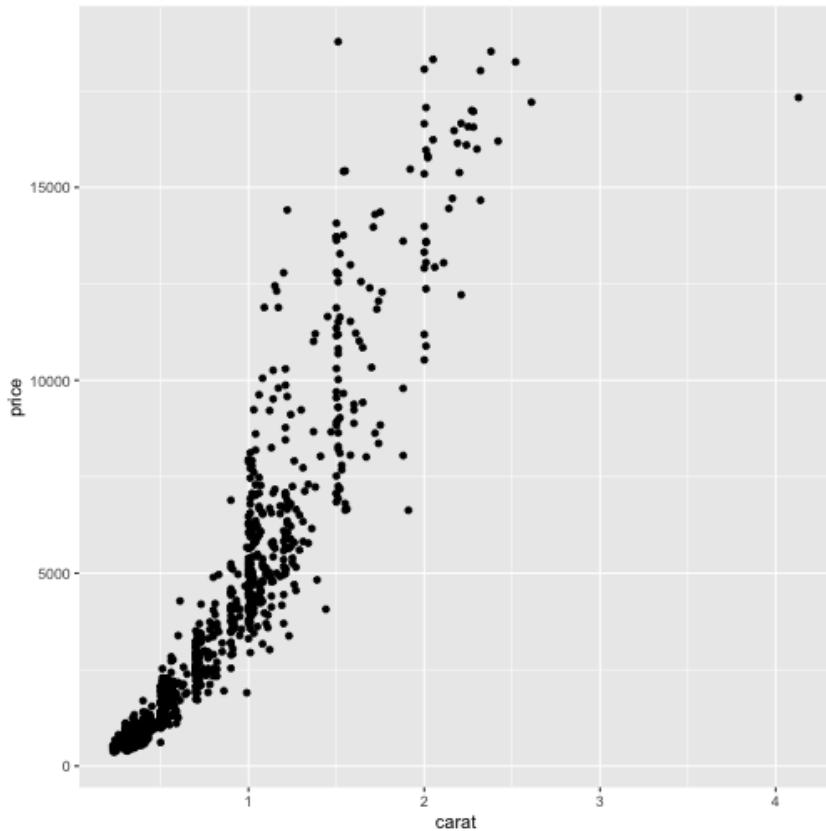
- Visit <https://ggplot2.tidyverse.org/index.html> for ggplot2 documentation and helpful articles. Reference section contains lots of examples for each geometry type.
- Refer to the ggplot2 cheatsheet:
<https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf>

ggplot objects

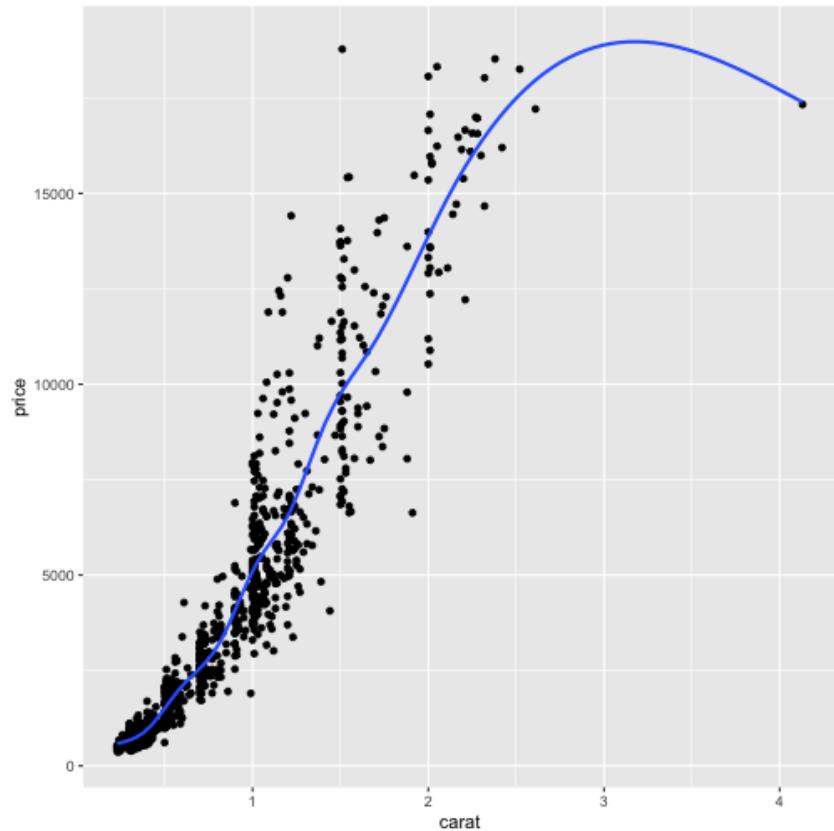
```
g = ggplot(diamonds, aes(x = carat, y = price)) + geom_point()  
class(g)
```

```
## [1] "gg"      "ggplot"
```

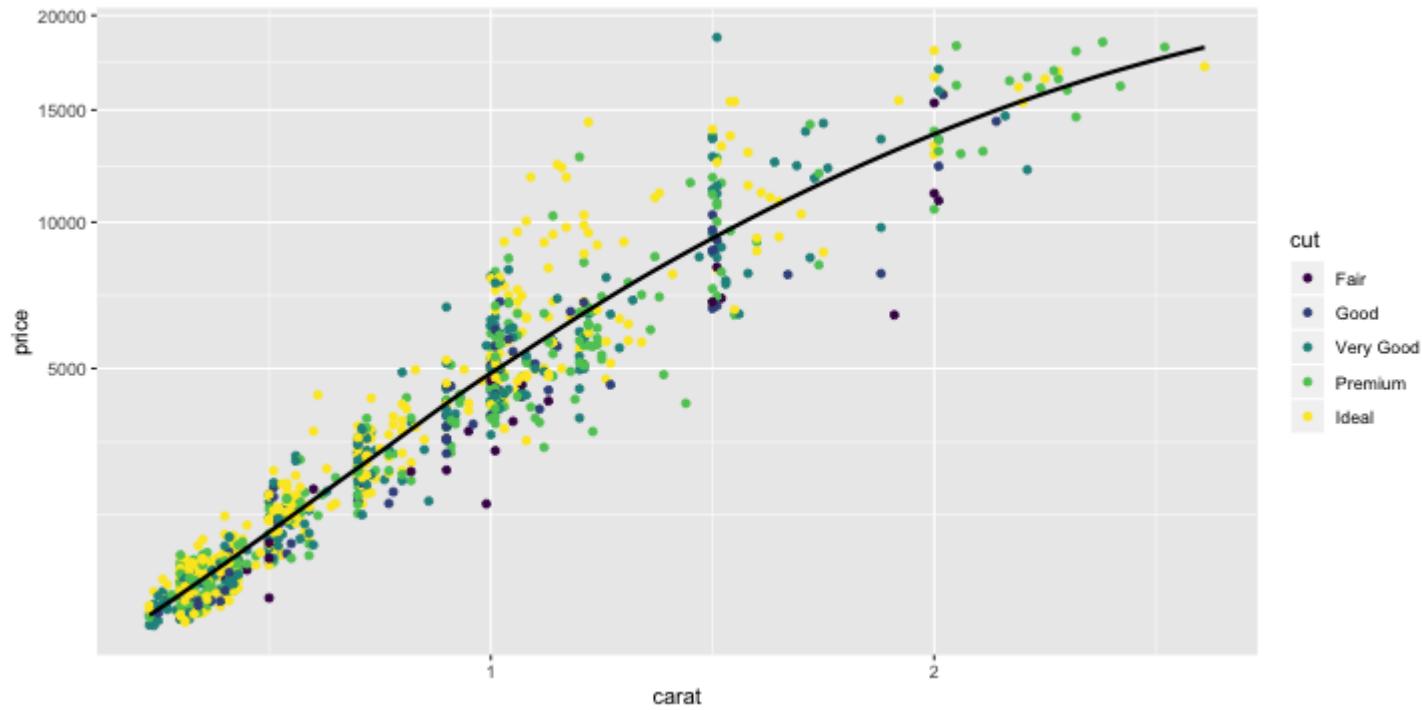
```
g
```



```
g + geom_smooth(se=FALSE)
```

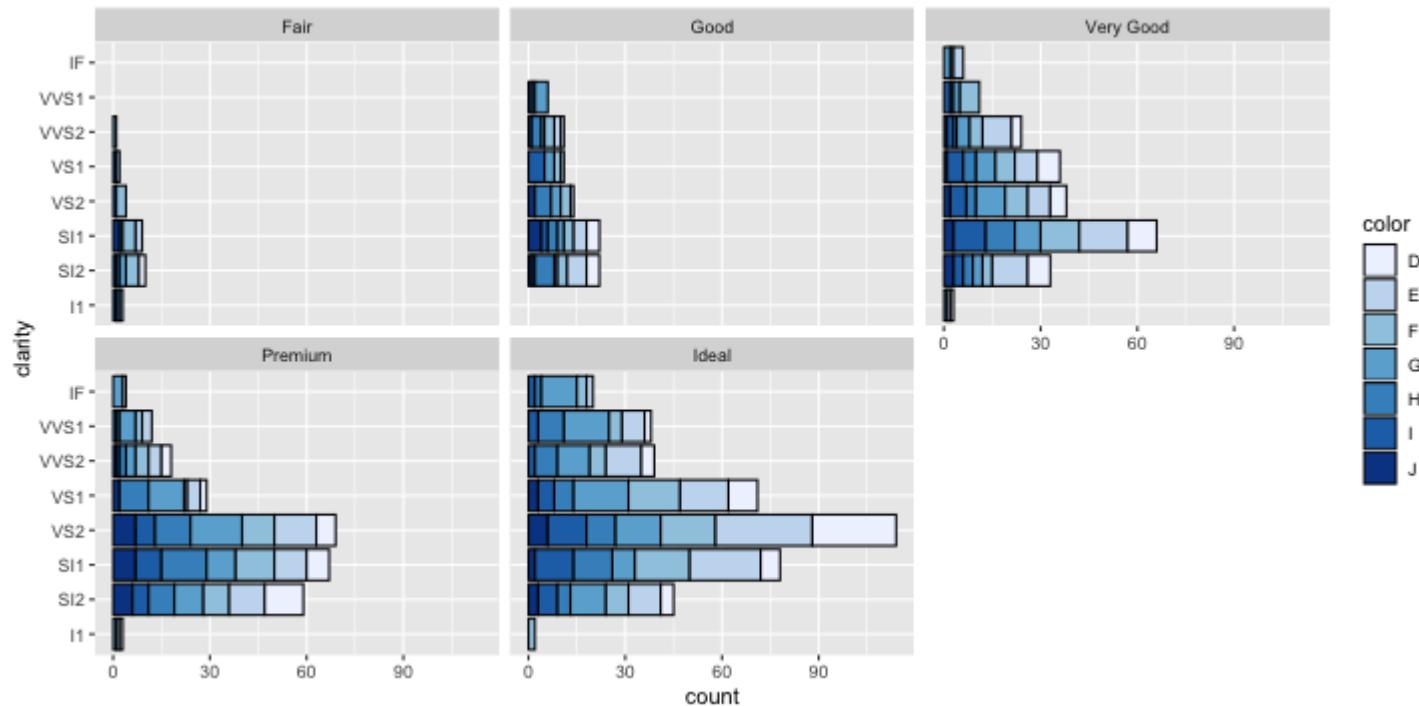


Exercise 1



Not the presence, or lack thereof, of any outliers.

Exercise 2





```
remotes::install_github("thomasp85/patchwork")
```

Plots

```
library(patchwork)

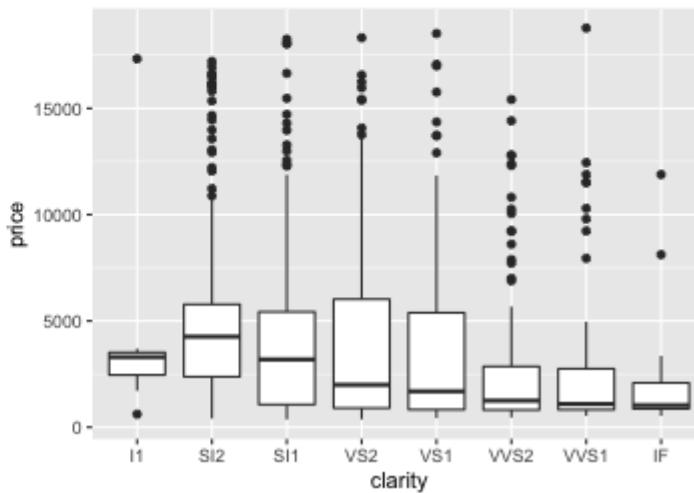
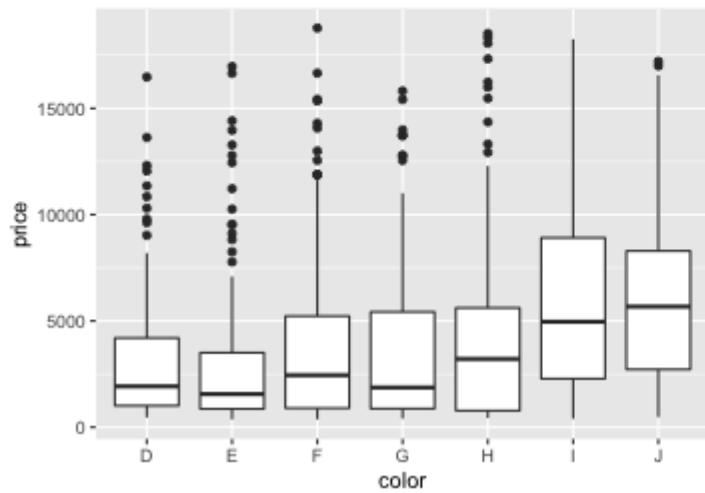
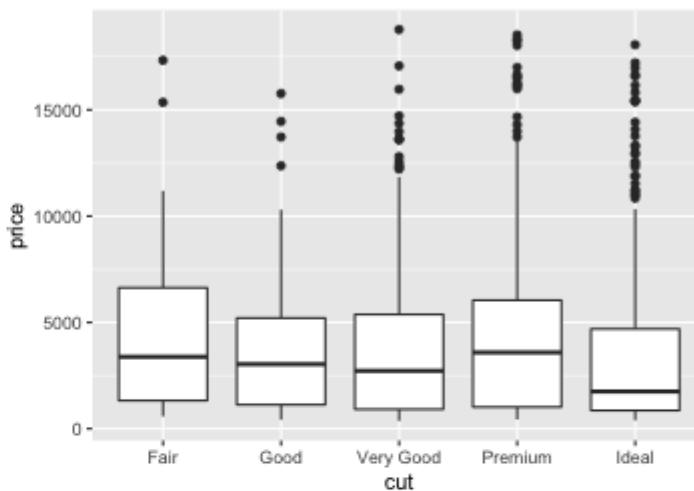
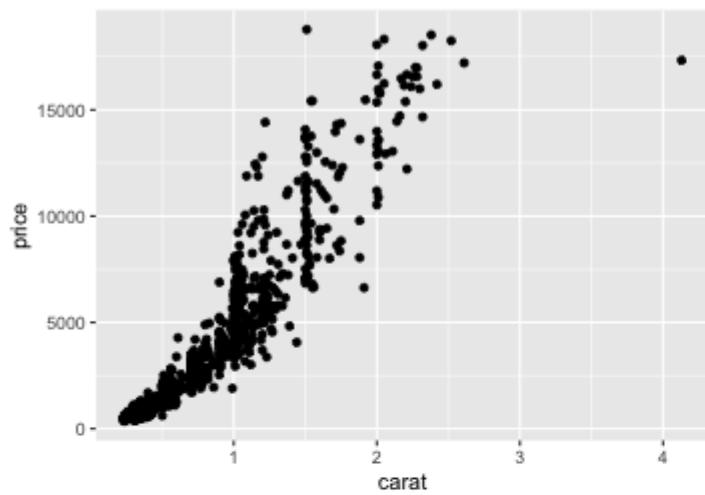
p1 = ggplot(diamonds) + geom_point(aes(x = carat, y = price))

p2 = ggplot(diamonds) + geom_boxplot(aes(x = cut, y = price))

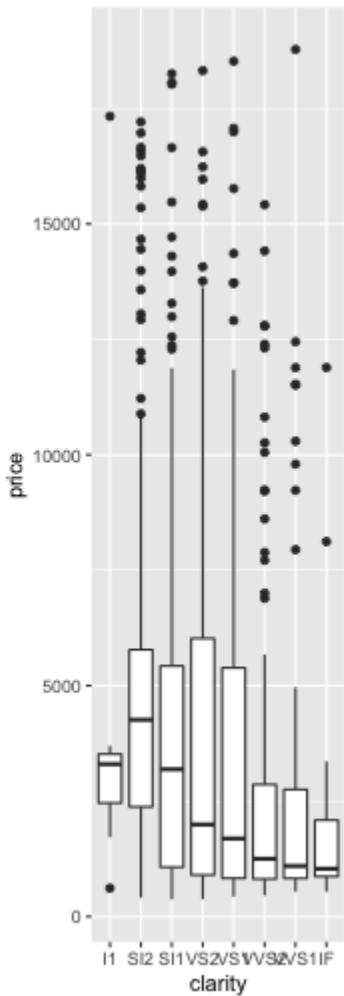
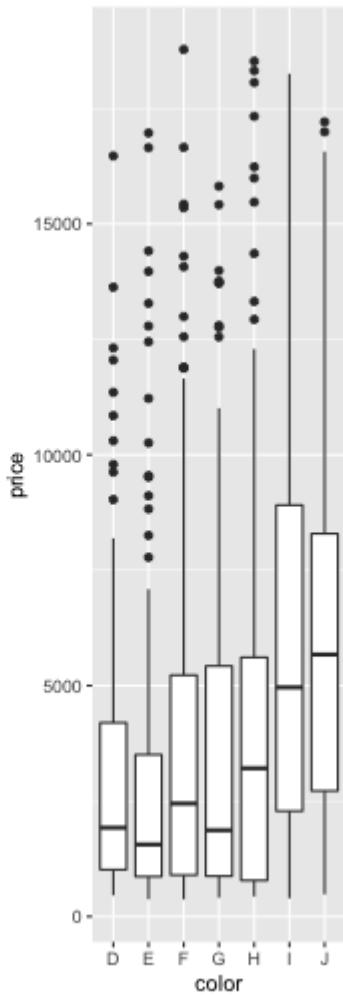
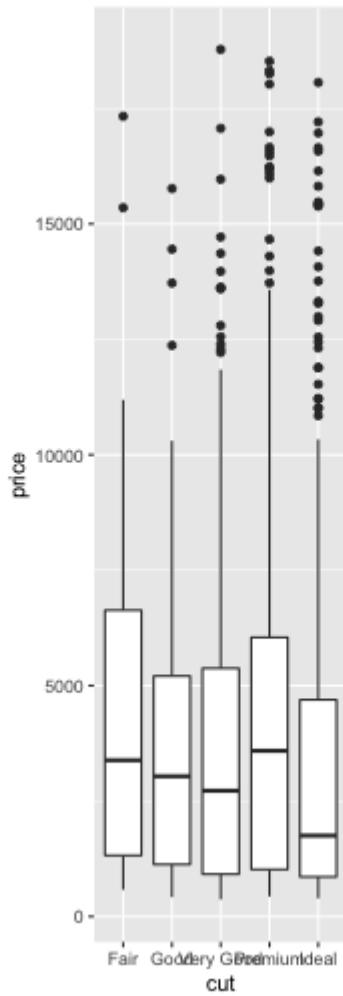
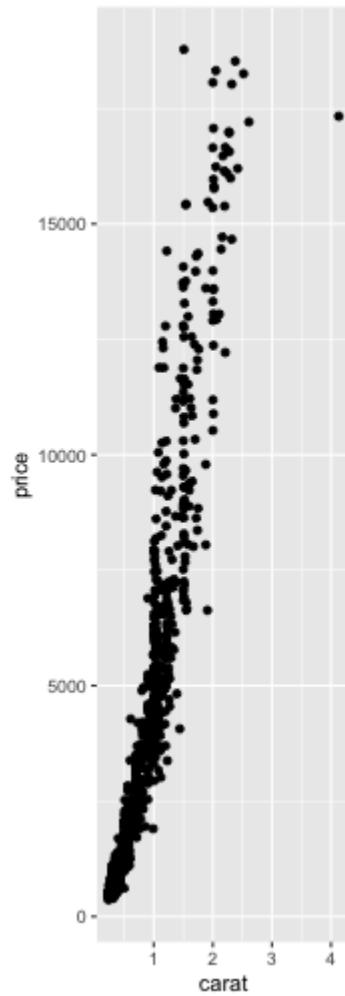
p3 = ggplot(diamonds) + geom_boxplot(aes(x = color, y = price))

p4 = ggplot(diamonds) + geom_boxplot(aes(x = clarity, y = price))
```

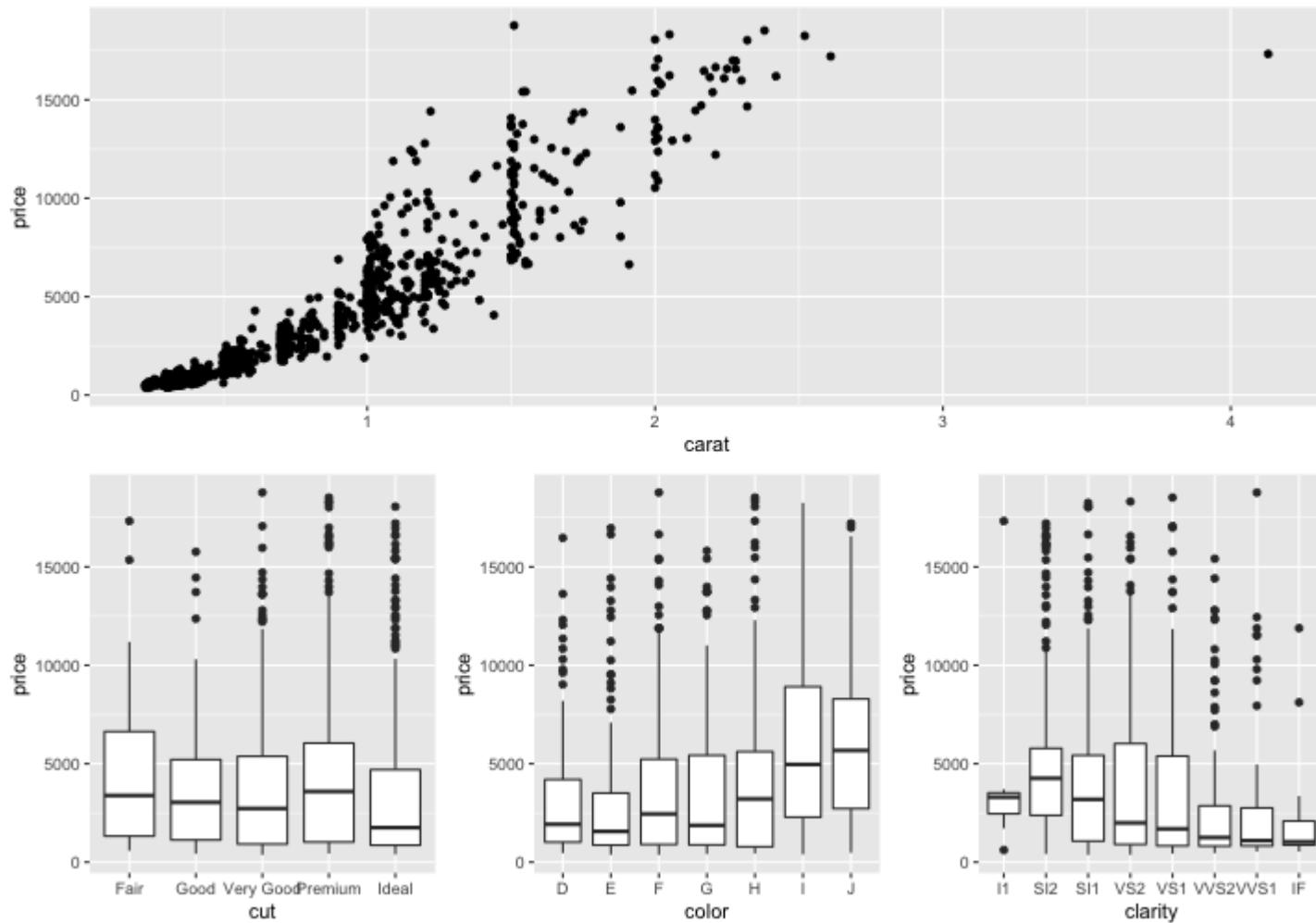
p1 + p2 + p3 + p4



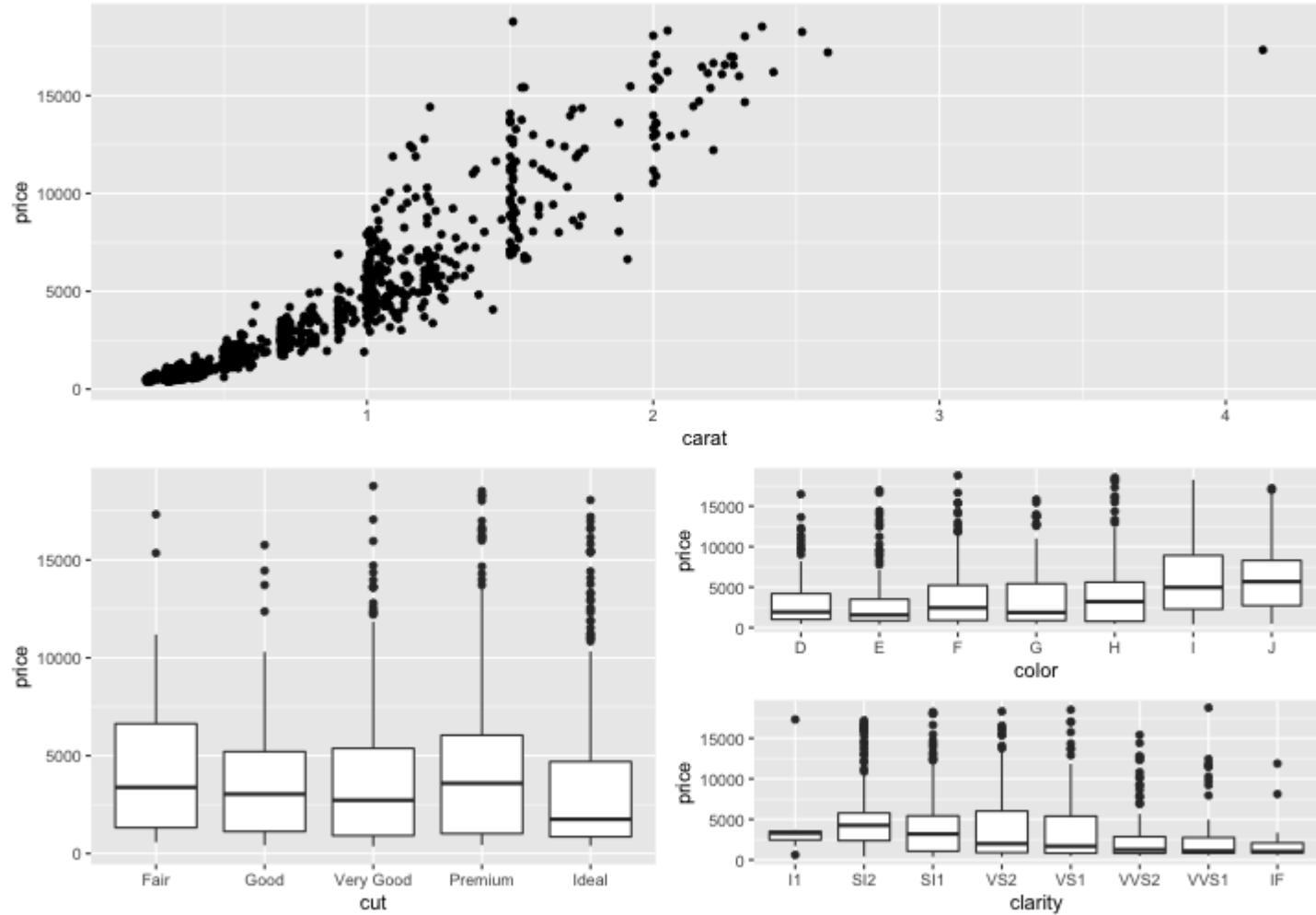
```
p1 + p2 + p3 + p4 + plot_layout(nrow=1)
```



$$p1 / (p2 + p3 + p4)$$



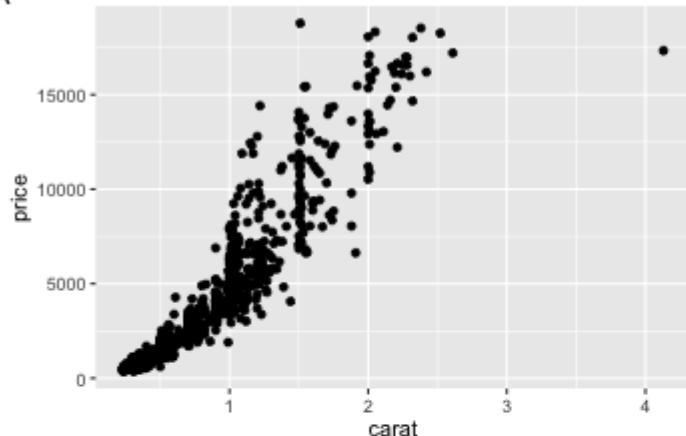
```
p1 + {  
  p2 + {  
    p3 + p4 + plot_layout(ncol = 1)  
  }  
} + plot_layout(ncol = 1)
```



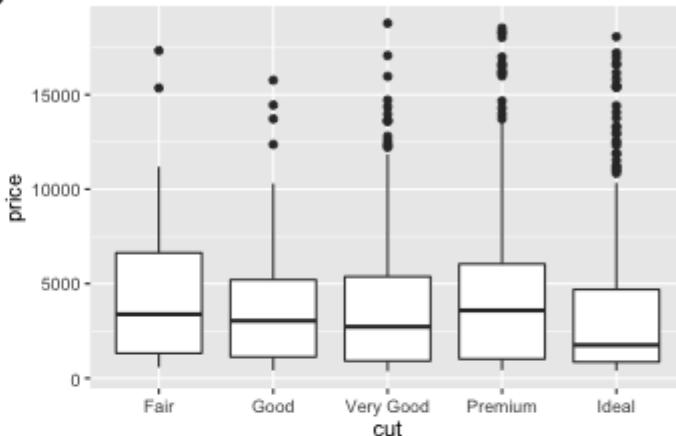
```
p1 + p2 + p3 + p4 + plot_annotation(title = "Diamonds data", tag_levels = c("A", "1"))
```

Diamonds data

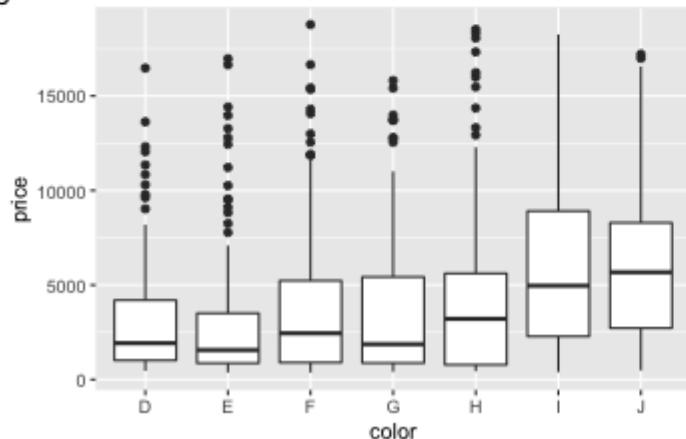
A



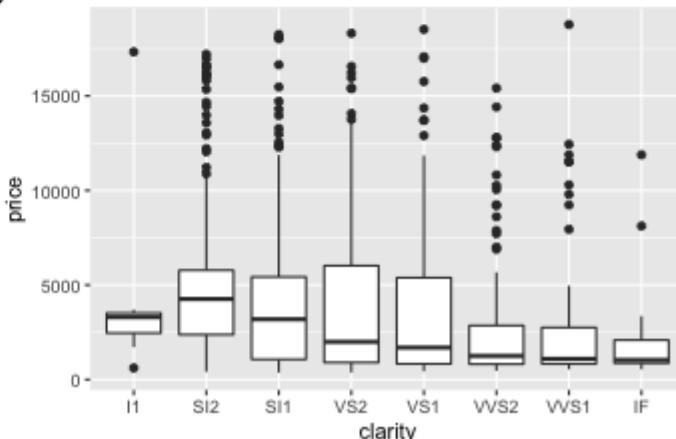
B



C



D



Why do we visualize?

Anscombe's Quartet

```
datasets::anscombe %>% as_tibble()
```

```
## # A tibble: 11 x 8
##       x1     x2     x3     x4     y1     y2     y3     y4
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1     10     10     10      8    8.04    9.14    7.46    6.58
## 2      8      8      8      8    6.95    8.14    6.77    5.76
## 3     13     13     13      8    7.58    8.74   12.7    7.71
## 4      9      9      9      8    8.81    8.77    7.11    8.84
## 5     11     11     11      8    8.33    9.26    7.81    8.47
## 6     14     14     14      8    9.96    8.1     8.84    7.04
## 7      6      6      6      8    7.24    6.13    6.08    5.25
## 8      4      4      4     19    4.26    3.1     5.39   12.5
## 9     12     12     12      8   10.8    9.13    8.15    5.56
## 10     7      7      7      8    4.82    7.26    6.42    7.91
## 11     5      5      5      8    5.68    4.74    5.73    6.89
```

Tidy anscombe

```
tidy_anscombe = datasets::anscombe %>%
  pivot_longer(everything(), names_sep = 1, names_to = c("var", "group")) %>%
  group_by(group, var) %>%
  mutate(index = seq_len(n())) %>%
  ungroup() %>%
  arrange(group, index) %>%
  pivot_wider(id_cols = c("group", "index"), names_from = var, values_from = value)
```

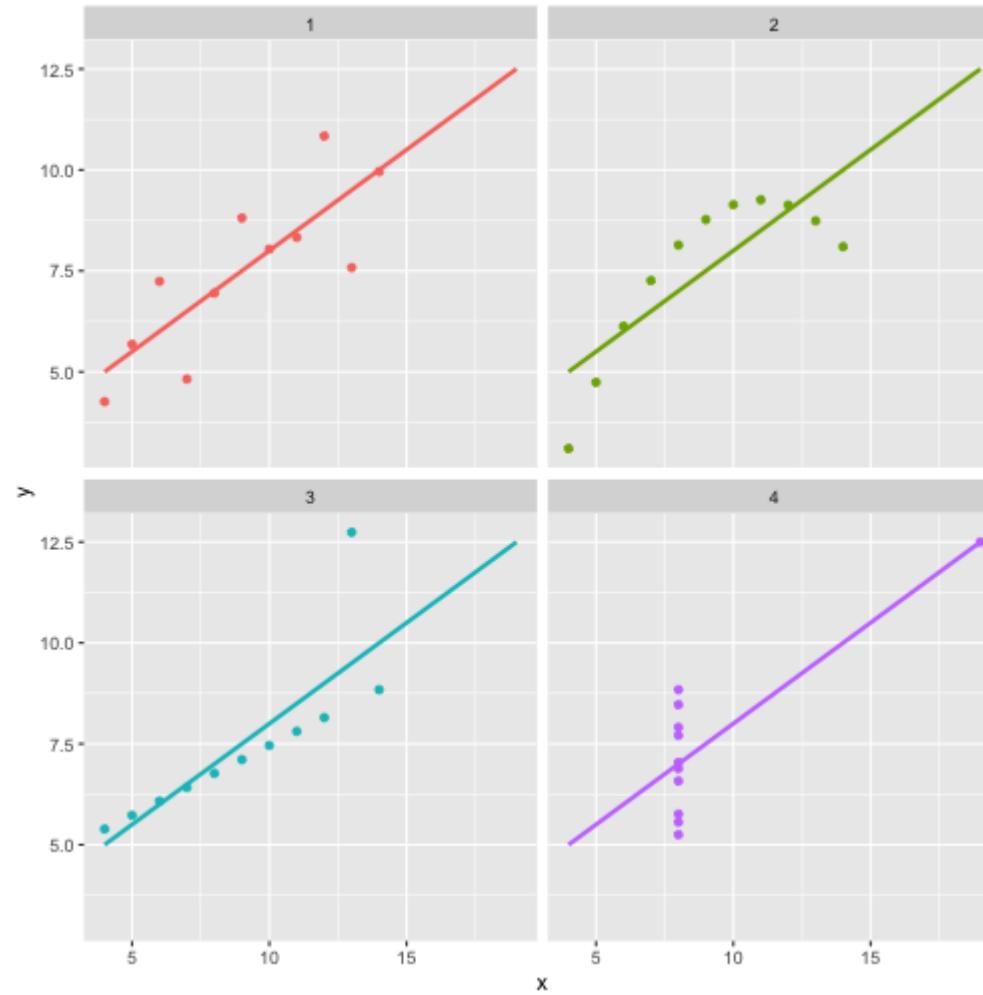
```
tidy_anscombe
```

```
## # A tibble: 44 x 4
##   group index     x     y
##   <chr> <int> <dbl> <dbl>
## 1 1         1    10  8.04
## 2 1         2     8  6.95
## 3 1         3    13  7.58
## 4 1         4     9  8.81
## 5 1         5    11  8.33
## 6 1         6    14  9.96
## 7 1         7     6  7.24
## 8 1         8     4  4.26
## 9 1         9    12 10.8 
## 10 1        10    7  4.82
## # ... with 34 more rows
```

```
tidy_anscombe %>%
  group_by(group) %>%
  summarize(mean_x = mean(x), mean_y = mean(y), sd_x = sd(x), sd_y = sd(y), cor = cor(x,y))

## # A tibble: 4 x 6
##   group mean_x mean_y   sd_x   sd_y   cor
##   <chr>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 1        9     7.50   3.32   2.03  0.816
## 2 2        9     7.50   3.32   2.03  0.816
## 3 3        9     7.5    3.32   2.03  0.816
## 4 4        9     7.50   3.32   2.03  0.817
```

```
ggplot(tidy_anscombe, aes(x = x, y = y, color = as.factor(group))) +  
  geom_point() +  
  facet_wrap(vars(group)) +  
  geom_smooth(method="lm", se=FALSE, fullrange=TRUE) +  
  guides(color=FALSE)
```



DatasauRus

```
datasauRus::datasaurus_dozen
```

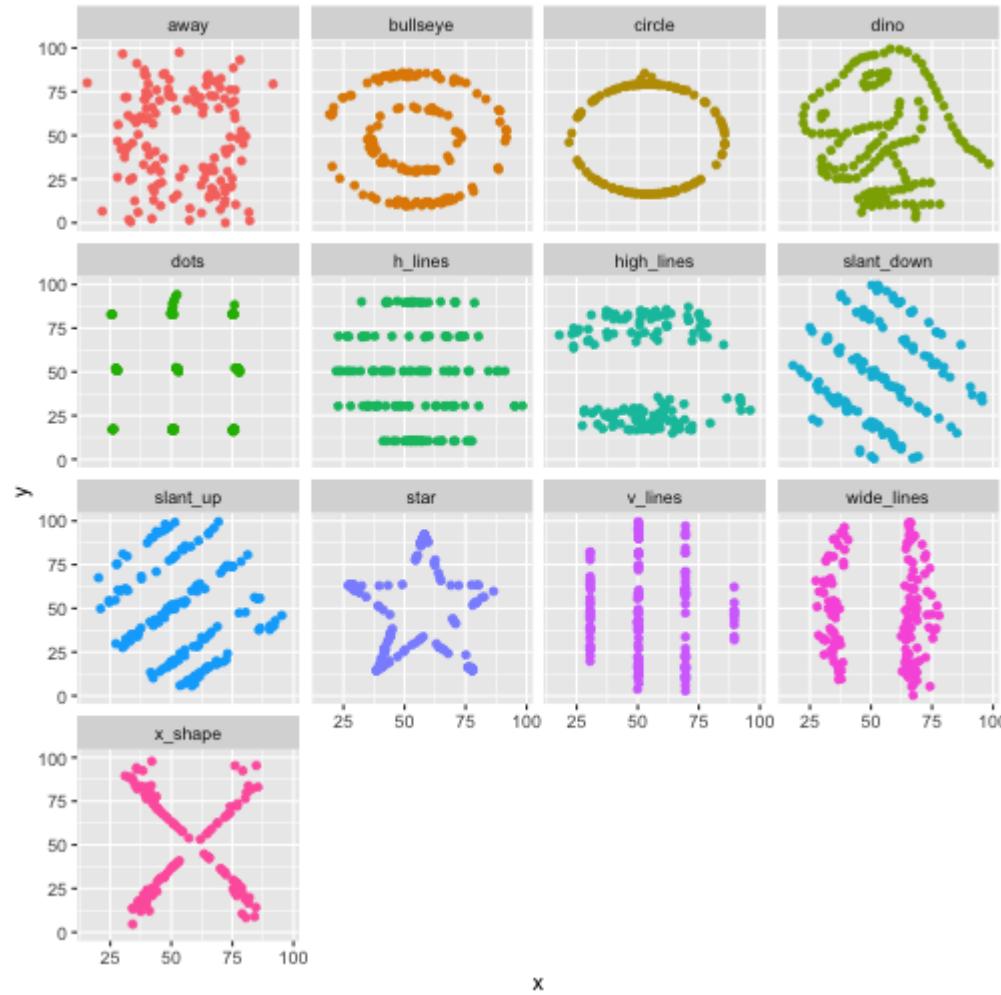
```
## # A tibble: 1,846 x 3
##   dataset      x     y
##   <chr>    <dbl> <dbl>
## 1 dino      55.4  97.2
## 2 dino      51.5  96.0
## 3 dino      46.2  94.5
## 4 dino      42.8  91.4
## 5 dino      40.8  88.3
## 6 dino      38.7  84.9
## 7 dino      35.6  79.9
## 8 dino      33.1  77.6
## 9 dino      29.0  74.5
## 10 dino     26.2  71.4
## # ... with 1,836 more rows
```

```
datasauRus::datasaurus_dozen %>%
  group_by(dataset) %>%
  summarize(mean_x = mean(x), mean_y = mean(y),
           sd_x = sd(x), sd_y = sd(y),
           cor = cor(x,y))
```

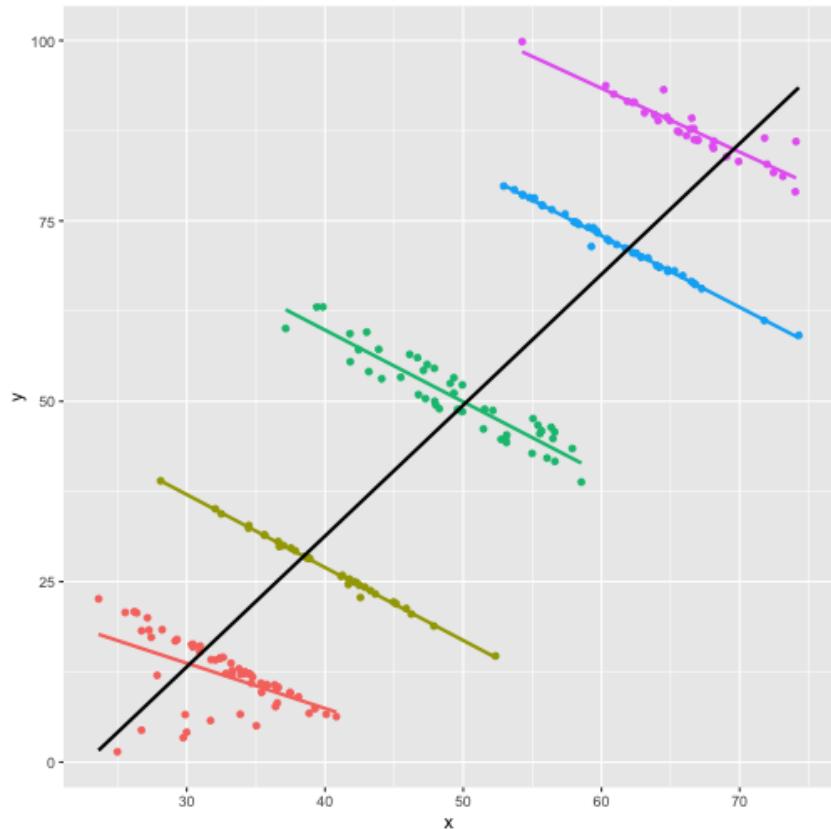
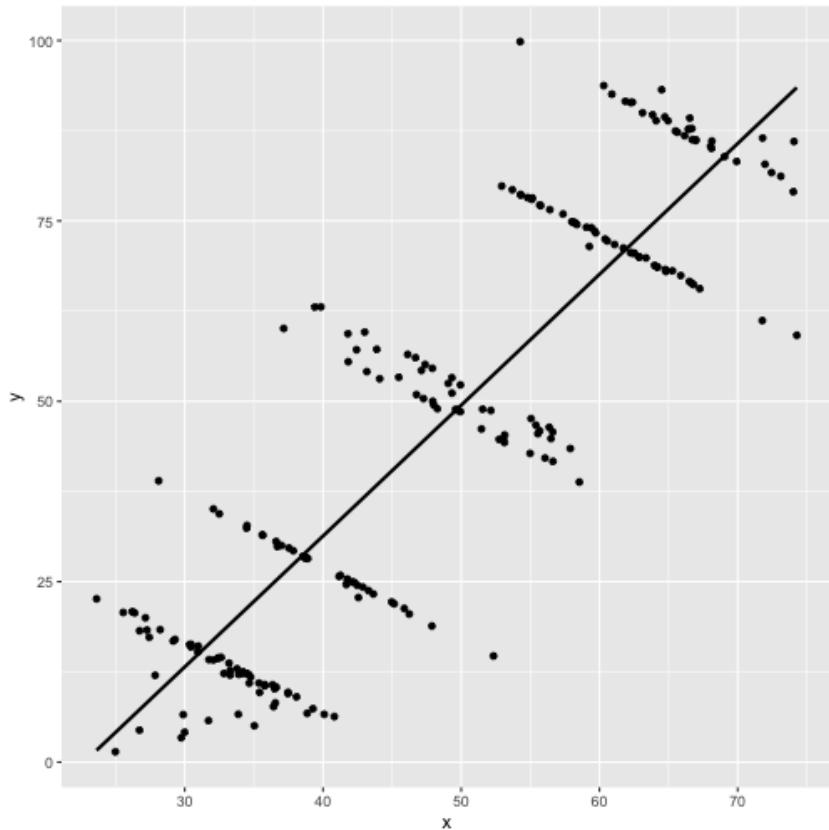
A tibble: 13 x 6

dataset	mean_x	mean_y	sd_x	sd_y	cor
1 away	54.3	47.8	16.8	26.9	-0.0641
2 bullseye	54.3	47.8	16.8	26.9	-0.0686
3 circle	54.3	47.8	16.8	26.9	-0.0683
4 dino	54.3	47.8	16.8	26.9	-0.0645
5 dots	54.3	47.8	16.8	26.9	-0.0603
6 h_lines	54.3	47.8	16.8	26.9	-0.0617
7 high_lines	54.3	47.8	16.8	26.9	-0.0685
8 slant_down	54.3	47.8	16.8	26.9	-0.0690
9 slant_up	54.3	47.8	16.8	26.9	-0.0686
10 star	54.3	47.8	16.8	26.9	-0.0630
11 v_lines	54.3	47.8	16.8	26.9	-0.0694
12 wide_lines	54.3	47.8	16.8	26.9	-0.0666
13 x_shape	54.3	47.8	16.8	26.9	-0.0656

```
ggplot(datasauRus::datasaurus_dozen, aes(x = x, y = y, color = dataset)) +  
  geom_point() +  
  facet_wrap(vars(dataset)) +  
  guides(color=FALSE)
```

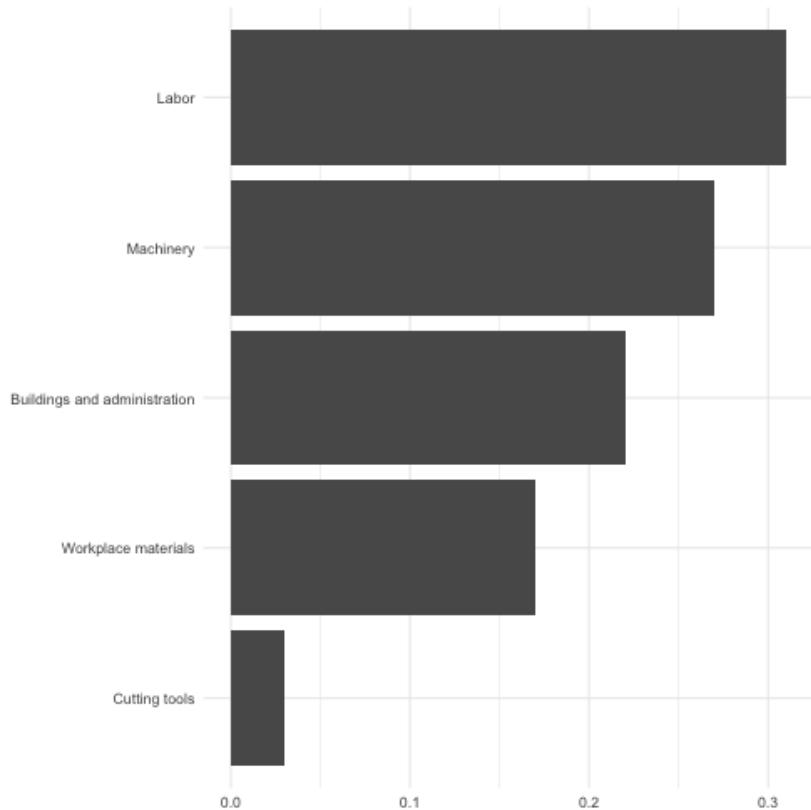
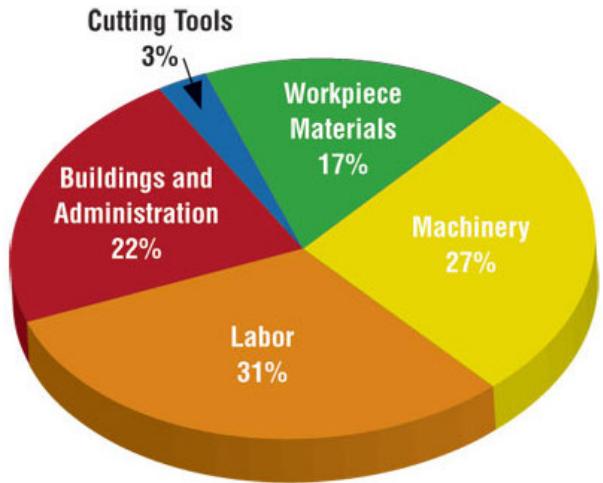


Simpson's Paradox

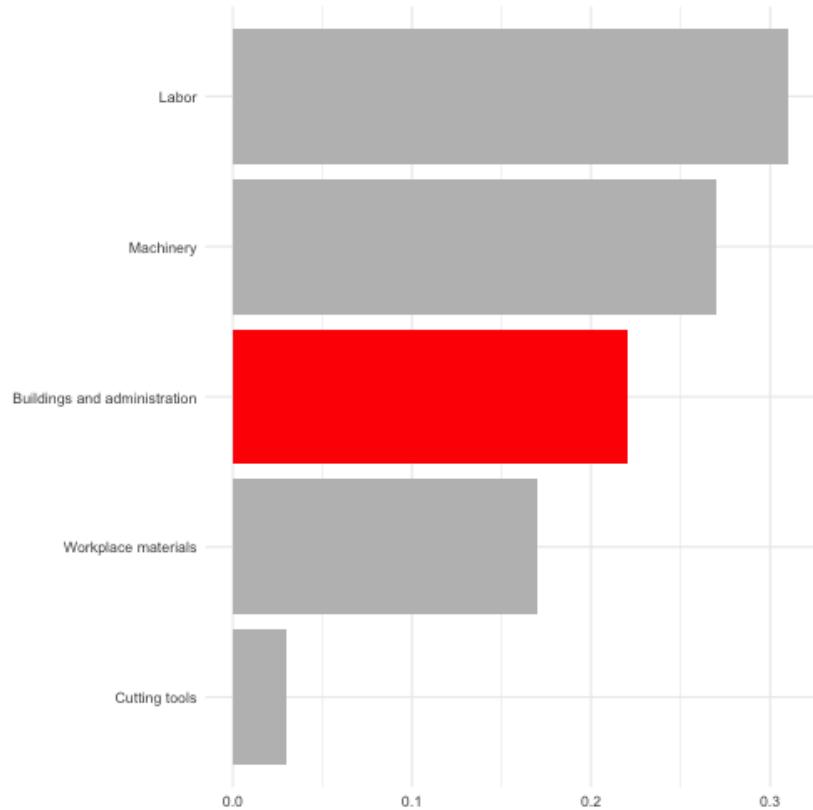
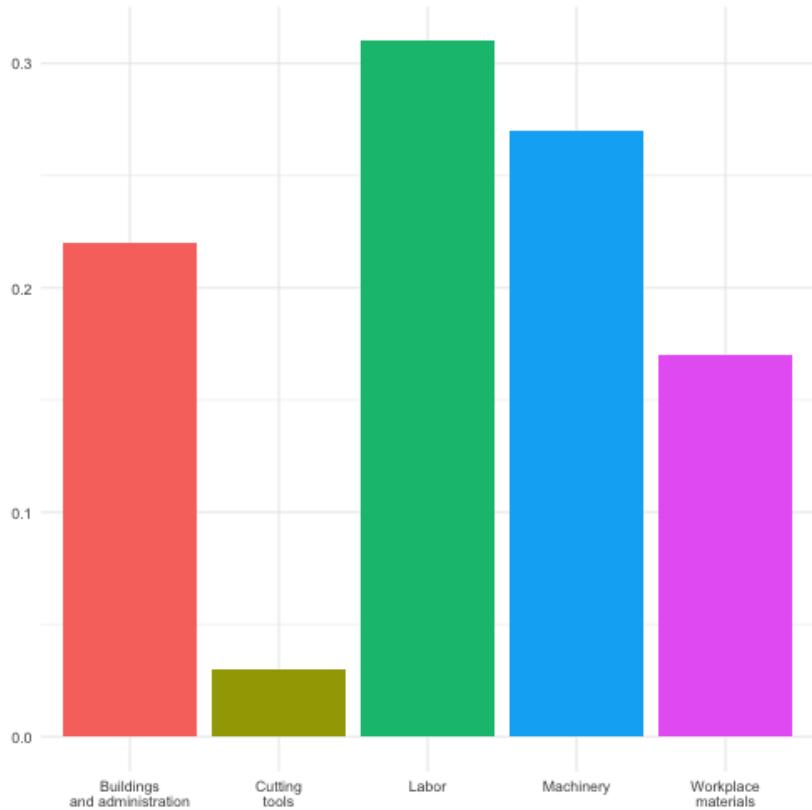


Designing effective visualizations

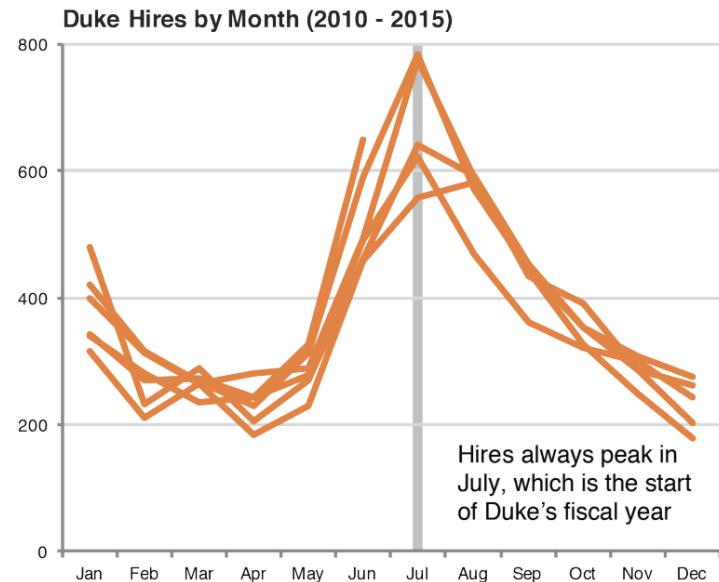
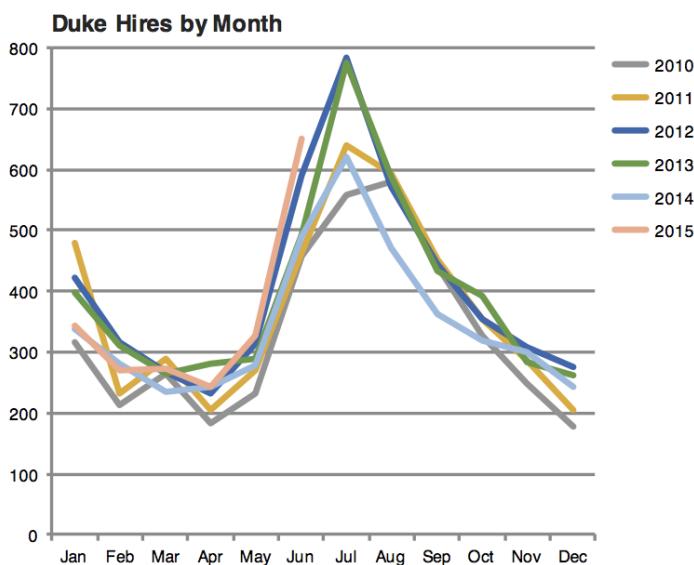
Keep it simple



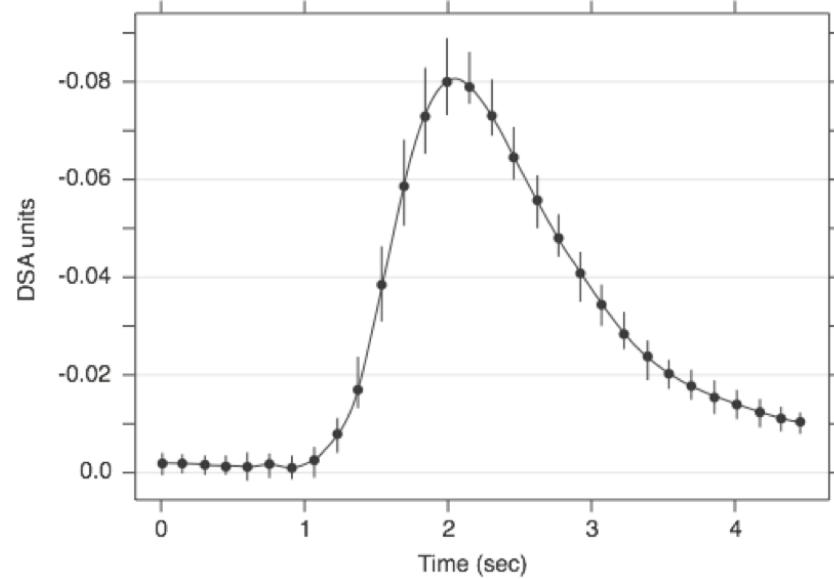
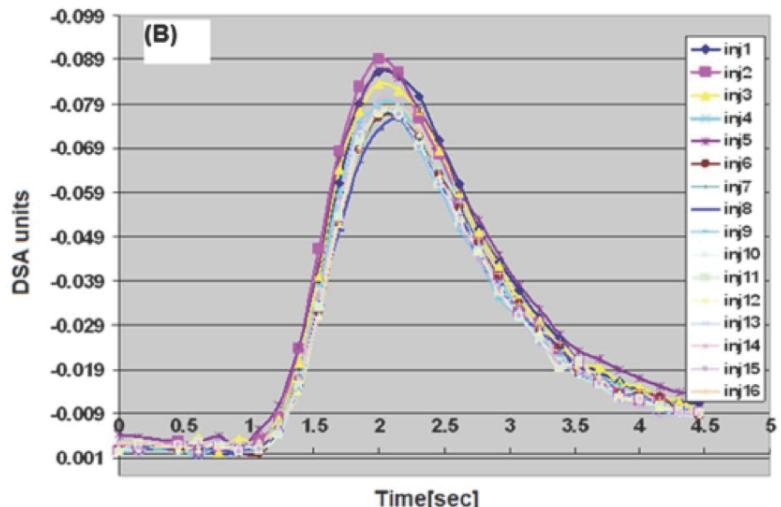
Use color to draw attention



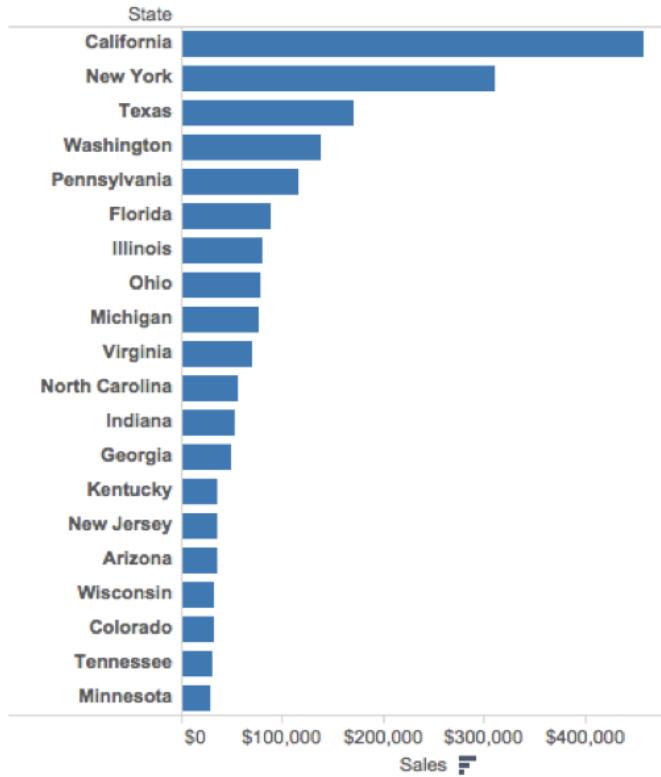
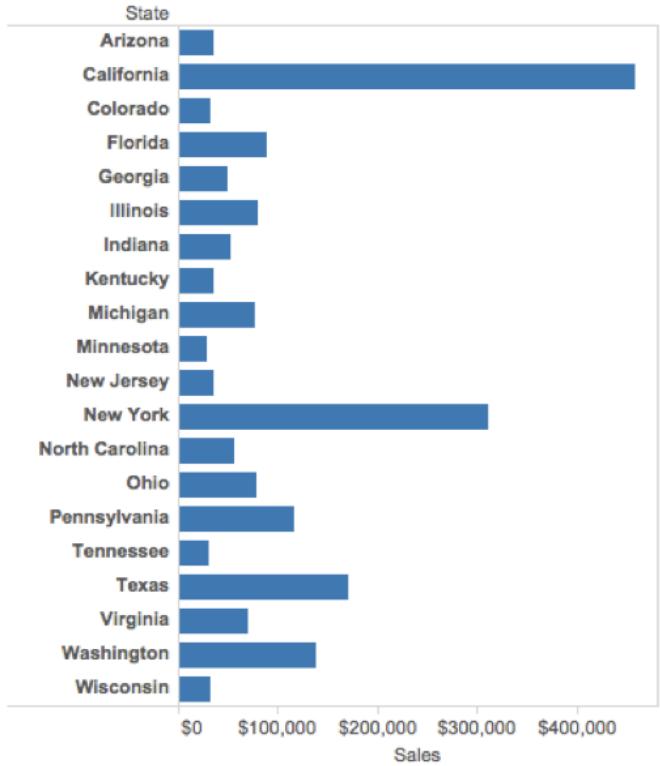
Tell a story



Leave out non-story details



Order / usage matters



Be clear about missing data

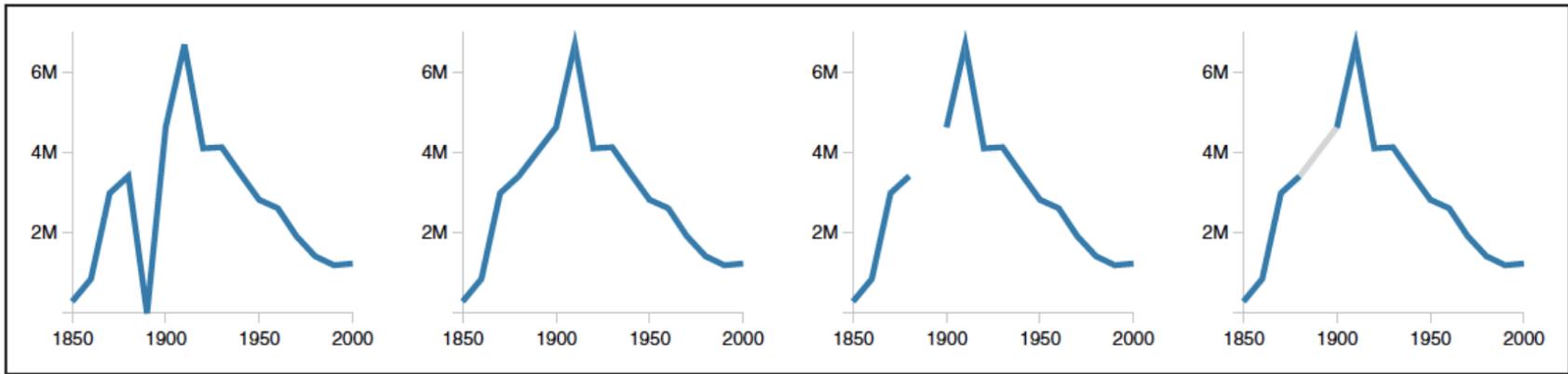
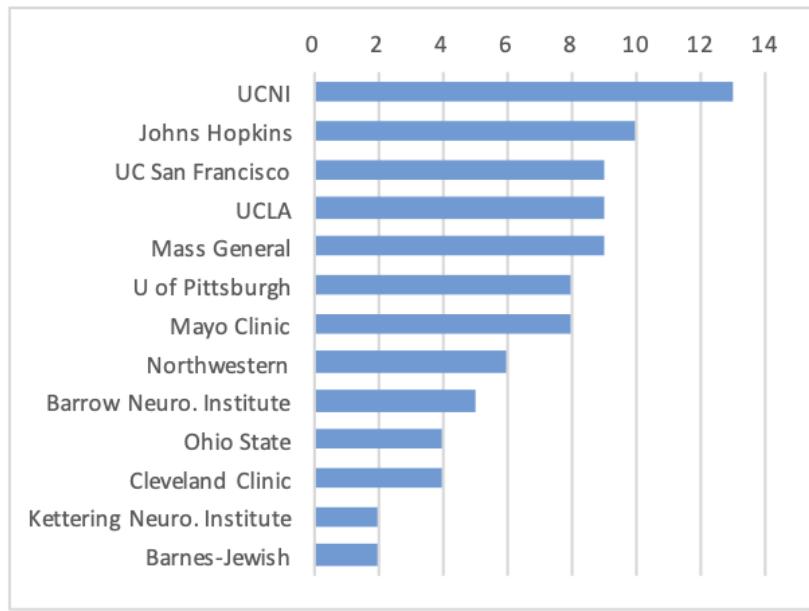
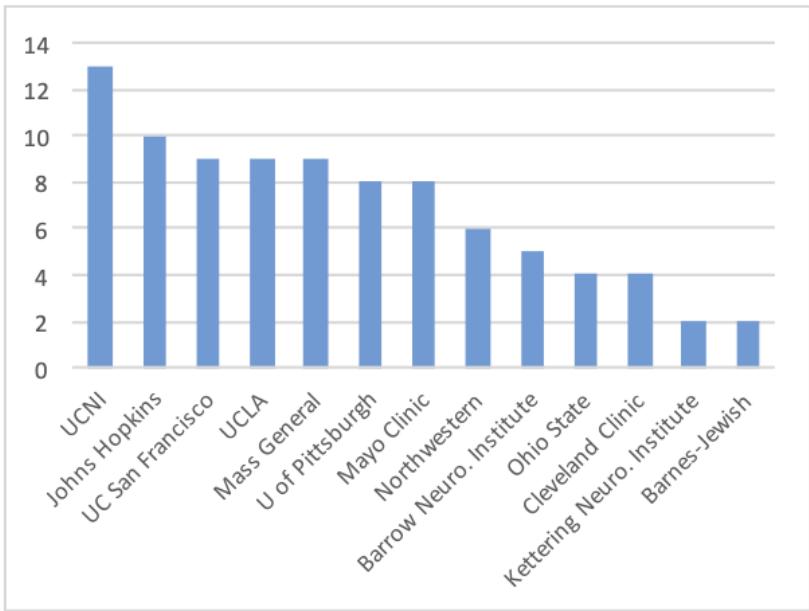


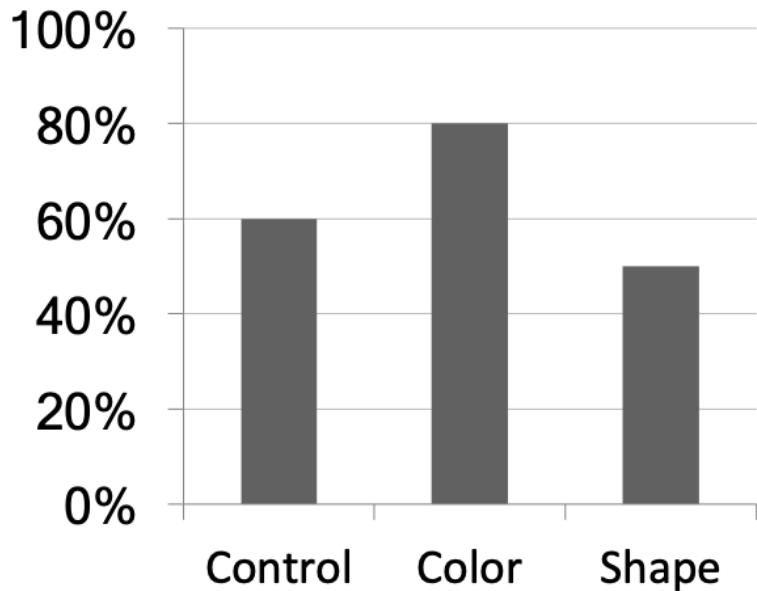
Figure 4. Alternative representations of missing data in a line chart. The data are U.S. census counts of people working as 'Farm Laborers'; values from 1890 are missing due to records being burned in a fire. (a) Missing data is treated as a zero value. (b) Missing data is ignored, resulting in a line segment that interpolates the missing value. (c) Missing data is omitted from the chart. (d) Missing data is explicitly interpolated and rendered in gray.

Reduce cognitive burden

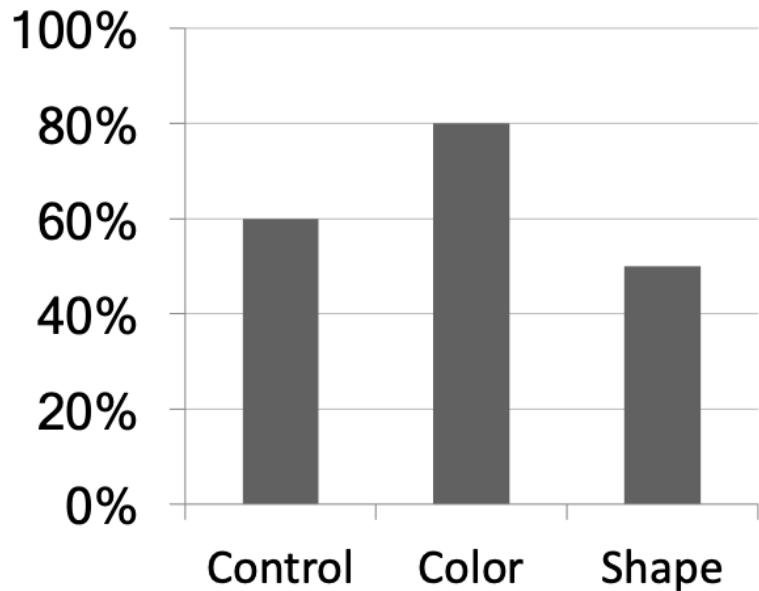


Use descriptive titles

Accuracy versus Color and Shape



Accuracy Improved by Color, not Shape

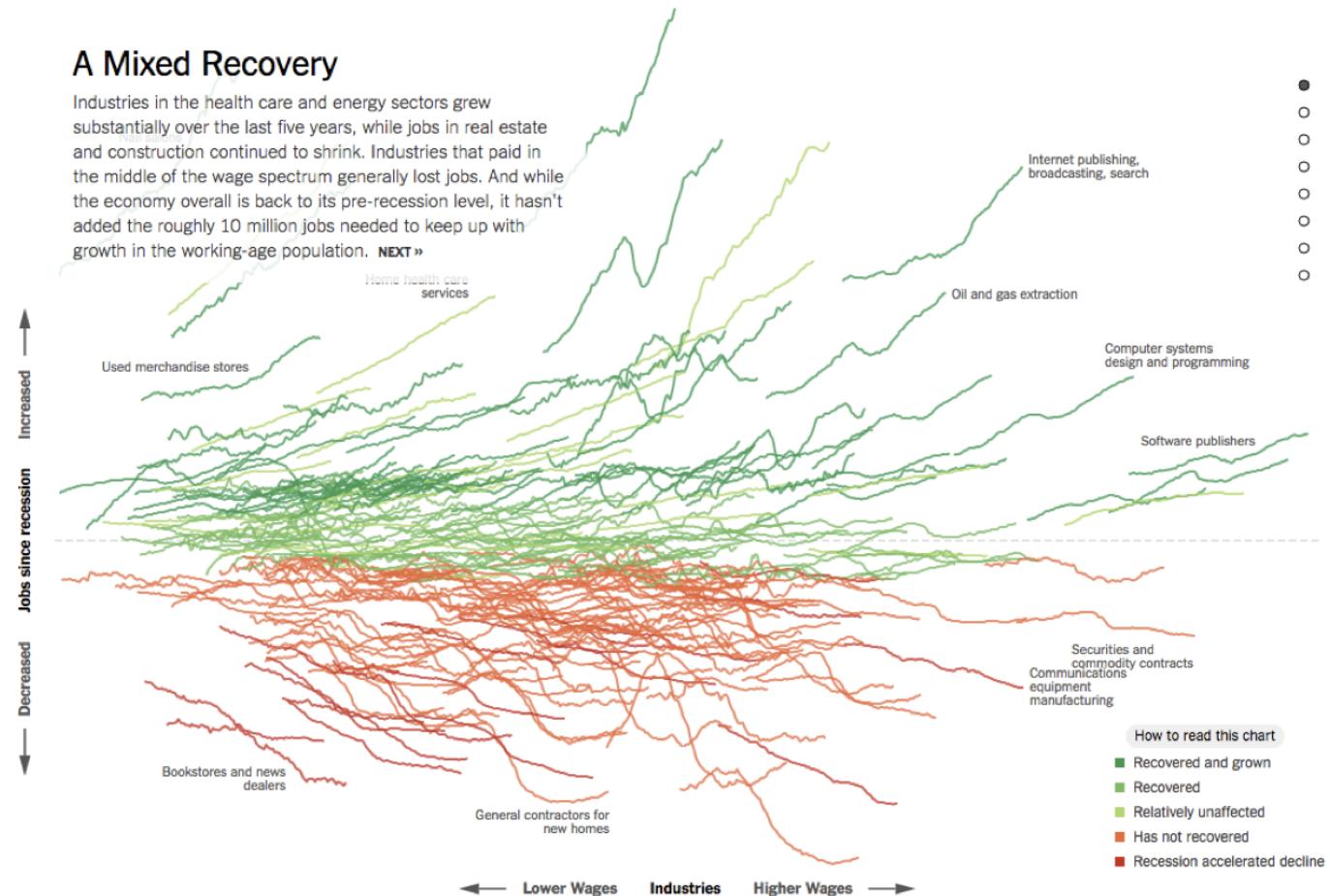


Annotate figures directly

AAPL stock example



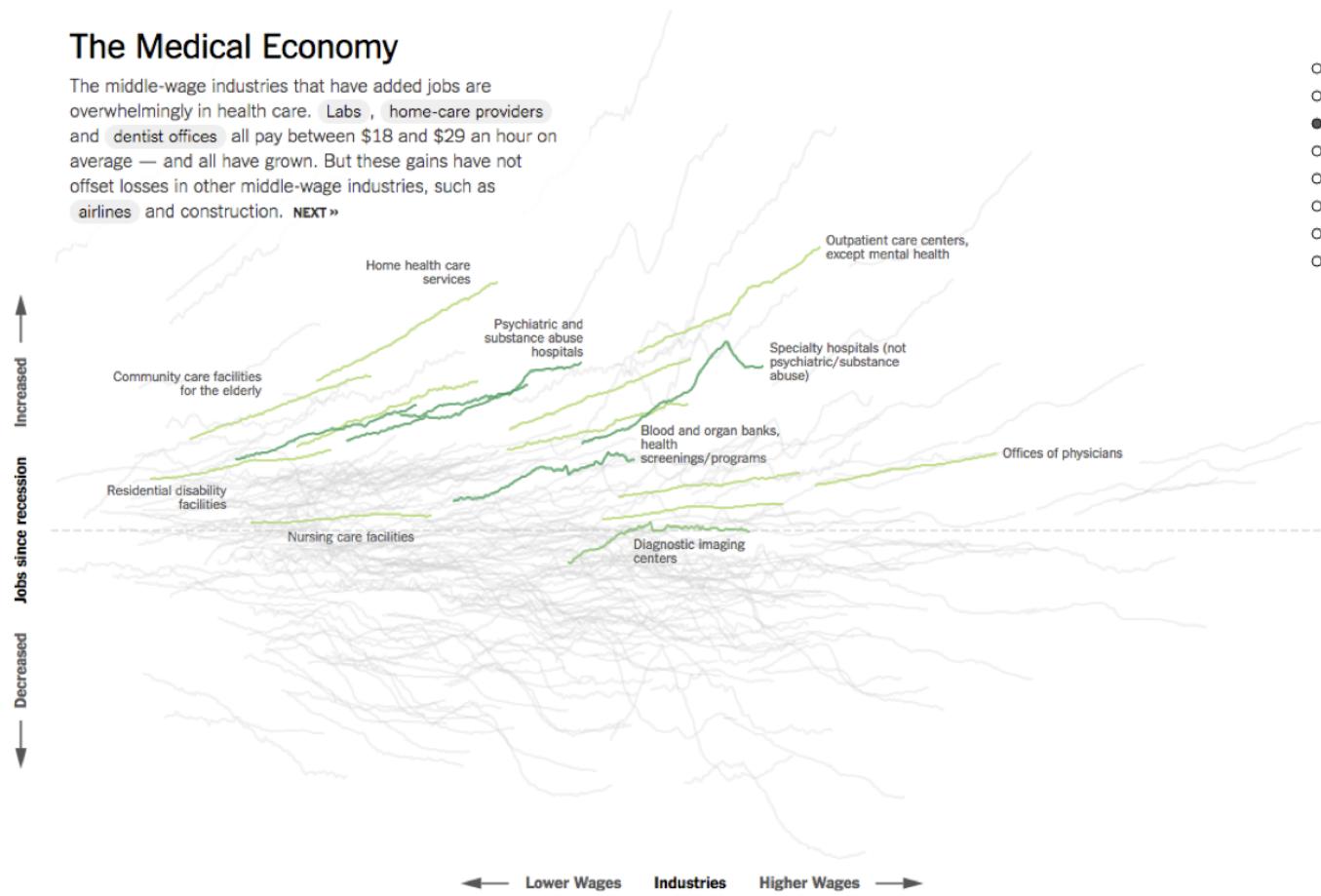
All of the data doesn't tell a story



All of the data doesn't tell a story

The Medical Economy

The middle-wage industries that have added jobs are overwhelmingly in health care. Labs, home-care providers and dentist offices all pay between \$18 and \$29 an hour on average — and all have grown. But these gains have not offset losses in other middle-wage industries, such as airlines and construction. [NEXT»](#)



All of the data doesn't tell a story

A Long Housing Bust

Home prices have rebounded from their crisis lows, but home building remains at historically low levels. Overall, industries connected with construction and real estate have lost 19 percent of their jobs since the recession began — hundreds of thousands more than health care has added. [NEXT »](#)

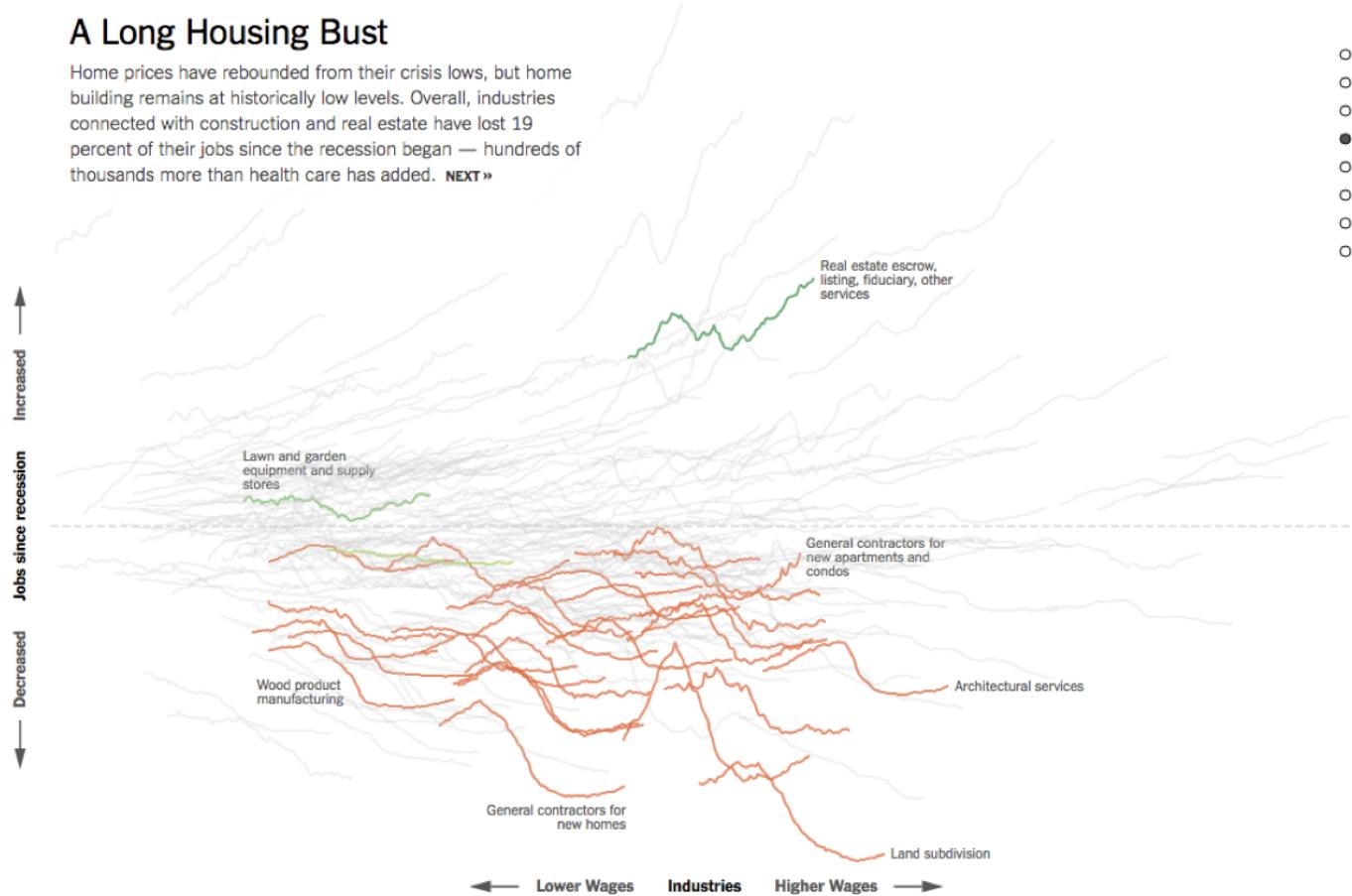


Chart Remakes / Makeovers

The Why Axis - BLS

Job openings in November 2012

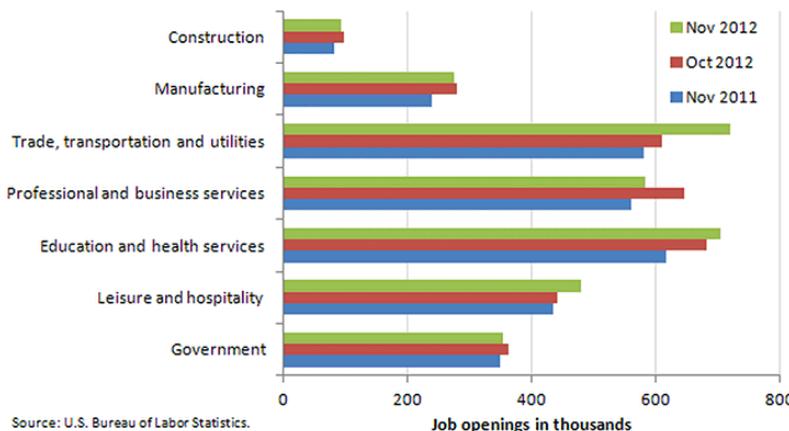
JANUARY 11, 2013

There were 3.7 million job openings on the last business day of November 2012, unchanged from October 2012. In November 2011 there were 3.3 million job openings.

CHART IMAGE

CHART DATA

Job openings by industry, November 2011, October 2012 and November 2012, seasonally adjusted



Source: U.S. Bureau of Labor Statistics.

Job openings in thousands

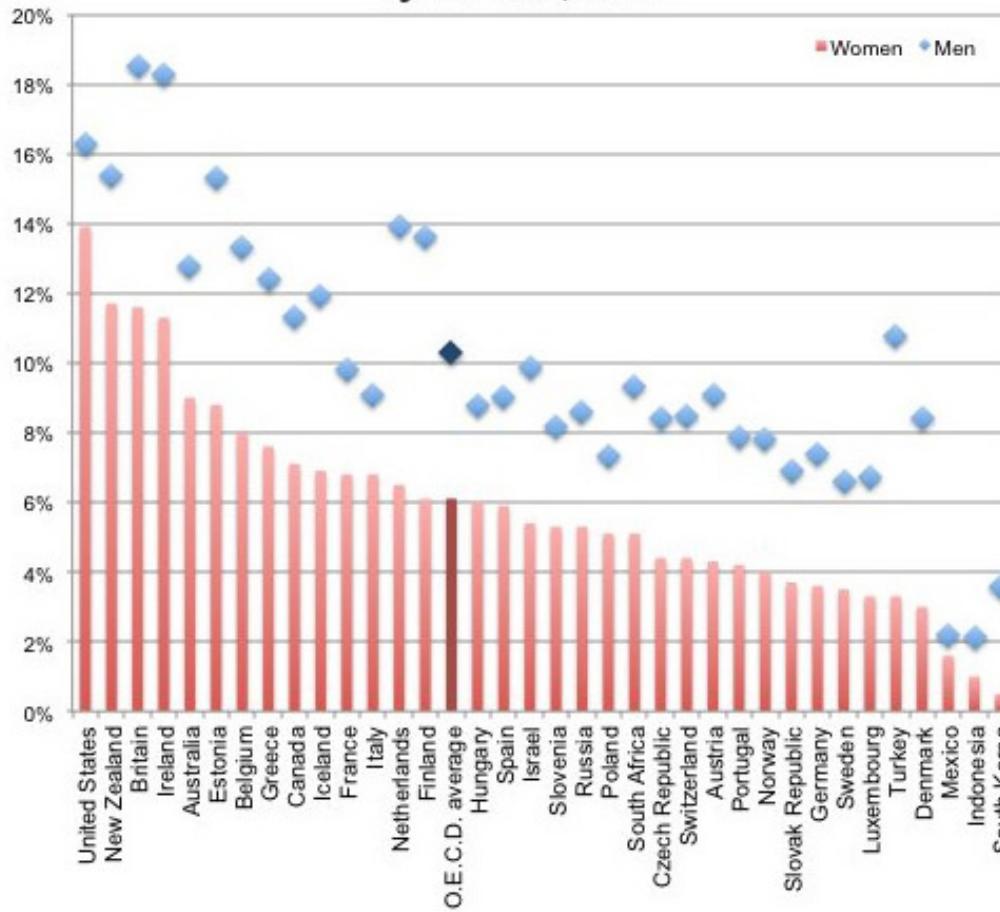
From November 2011 to November 2012, job openings increased most in retail trade (144,000, within the trade, transportation and utilities industry) and health care and social assistance (91,000, within the education and health services industry).

Government job openings increased the least, by 6,000.

These data are from the [Job Openings and Labor Turnover Survey](#). Data for the most recent month are preliminary and subject to revision. For additional information, see "Job Openings and Labor Turnover — November 2012" ([HTML](#)) ([PDF](#)), news release USDL-13-0015. More charts featuring data on job openings, hires, and employment separations can be found in [Job Openings and Labor Turnover Survey Highlights: November 2012](#) ([PDF](#)).

The Why Axis - Gender Gap

Percentage of Employed Who Are Senior Managers,
by Gender, 2008



Acknowledgments

Acknowledgments

Above materials are derived in part from the following sources:

- Hadley Wickham - R for Data Science & Elegant Graphics for Data Analysis
- ggplot2 website
- Visualization training materials developed by Angela Zoss and Eric Monson, Duke DVS