

Lec 10 - ggplot2 ecosystem & designing visualization

Statistical Programming

Sem 1, 2020

Dr. Colin Rundel

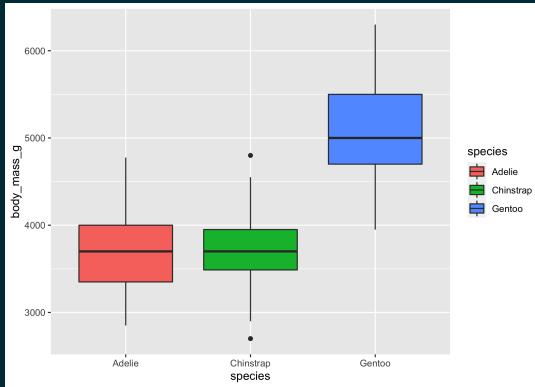
The ggplot2 ecosystem

ggthemes

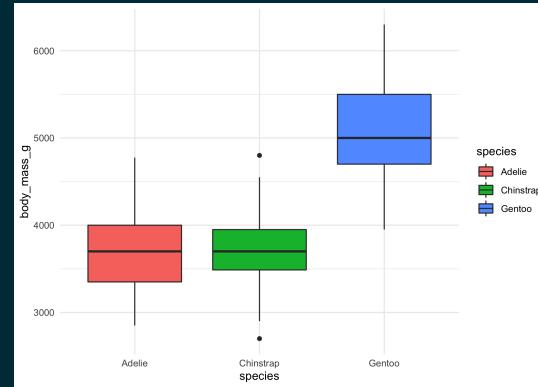
ggplot2 themes

```
g = ggplot(palmerpenguins::penguins, aes(x=species, y=body_mass_g, fill=species)) + geom_boxplot()
```

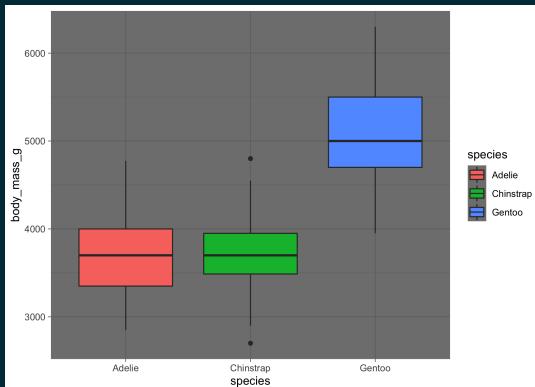
g



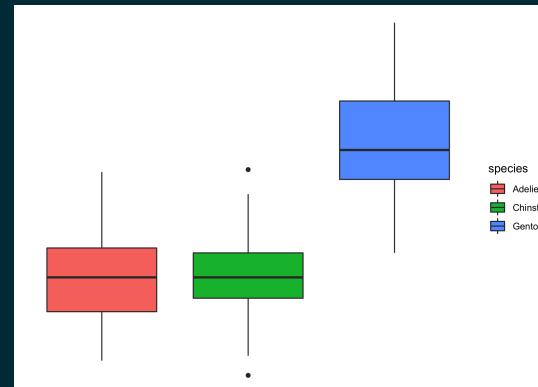
g + theme_minimal()



g + theme_dark()

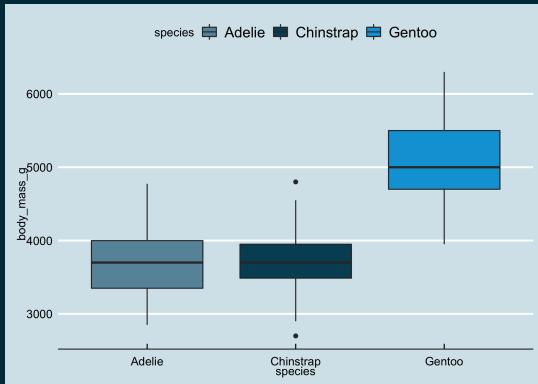


g + theme_void()

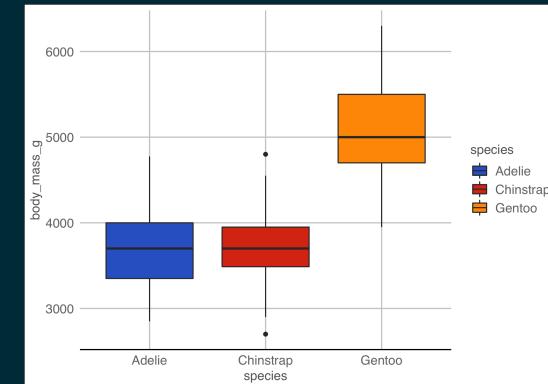


ggthemes

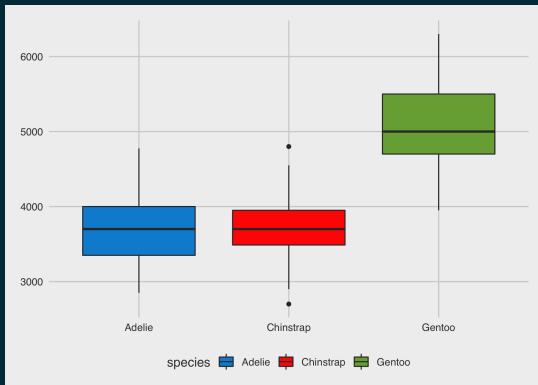
```
g + ggthemes::theme_economist() +  
  ggthemes::scale_fill_economist()
```



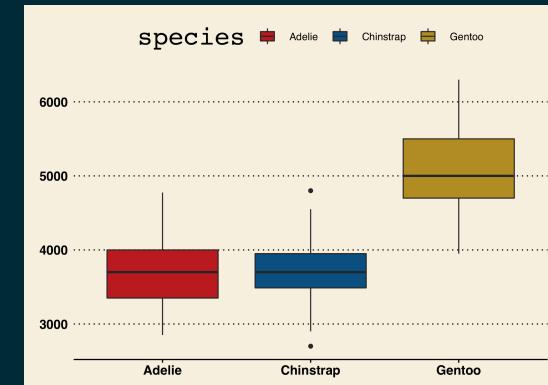
```
g + ggthemes::theme_gdocs() +  
  ggthemes::scale_fill_gdocs()
```



```
g + ggthemes::theme_fivethirtyeight() +  
  ggthemes::scale_fill_fivethirtyeight()
```

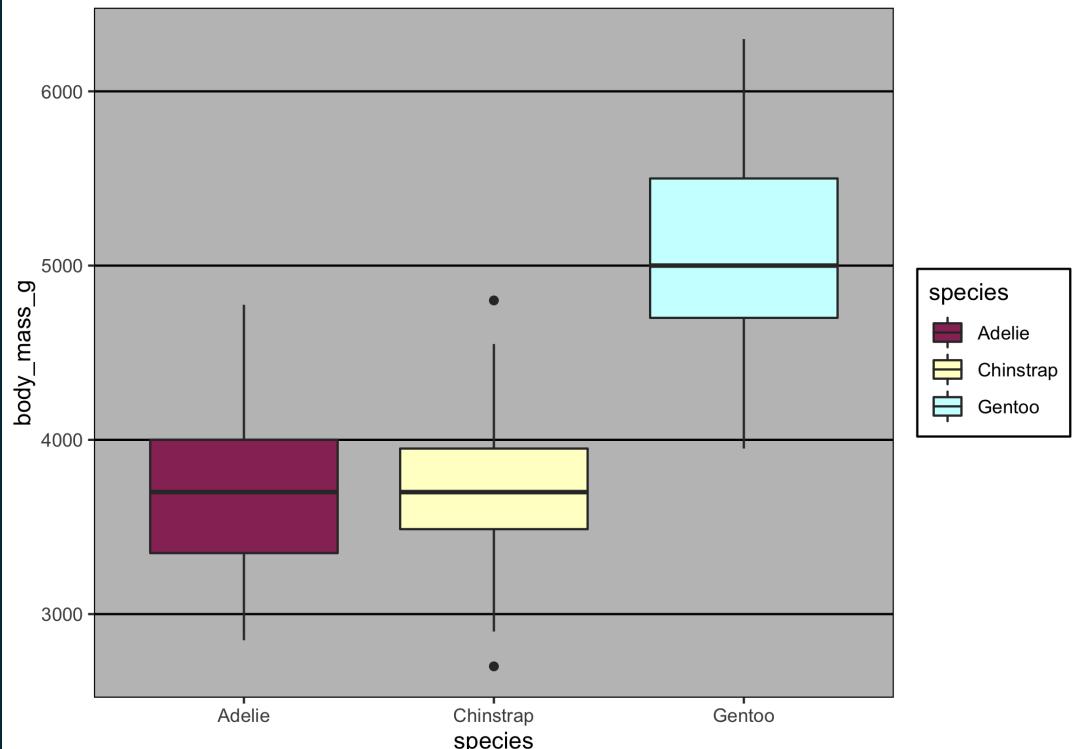


```
g + ggthemes::theme_wsj() +  
  ggthemes::scale_fill_wsj()
```

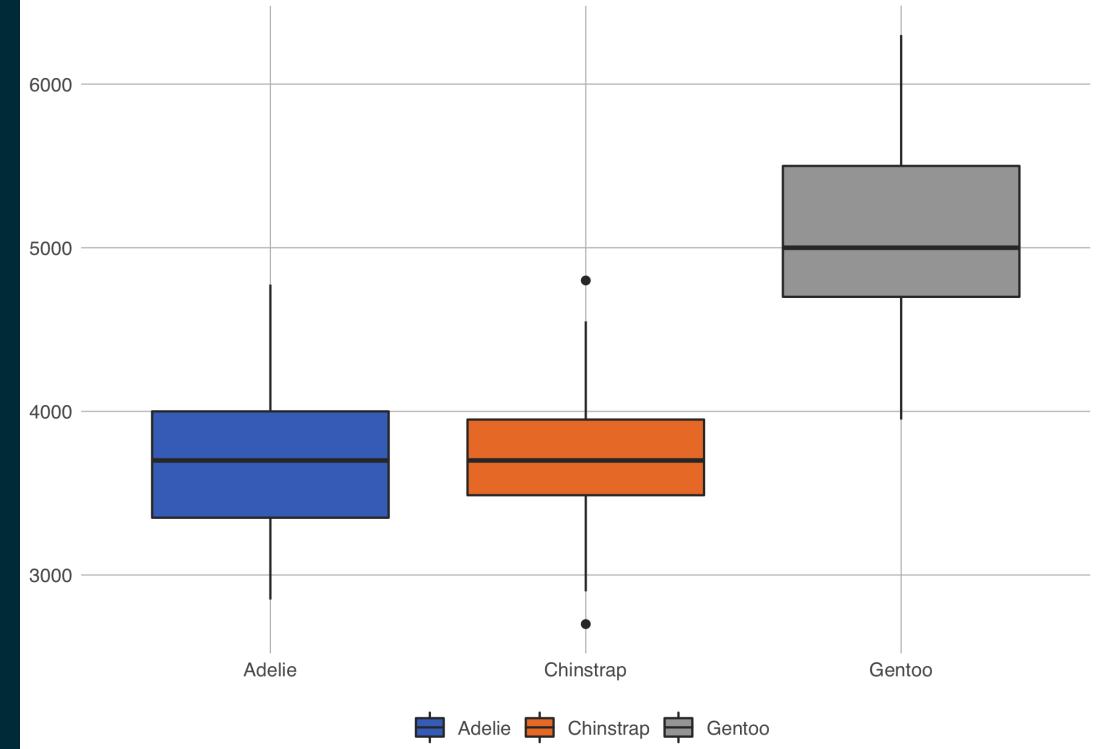


And for those who miss Excel

```
g + ggthemes::theme_excel() +  
  ggthemes::scale_fill_excel()
```

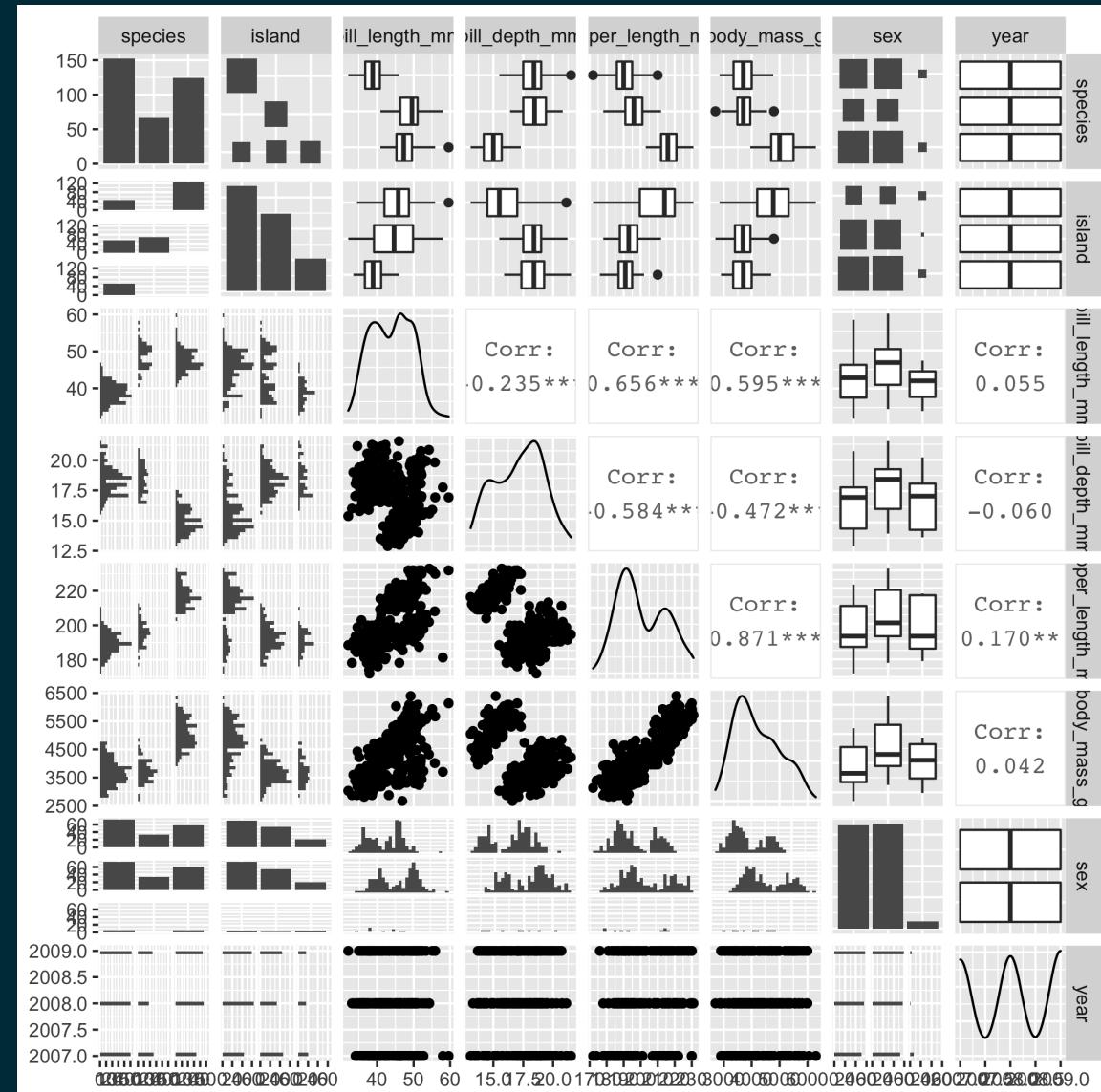


```
g + ggthemes::theme_excel_new() +  
  ggthemes::scale_fill_excel_new()
```



GGally

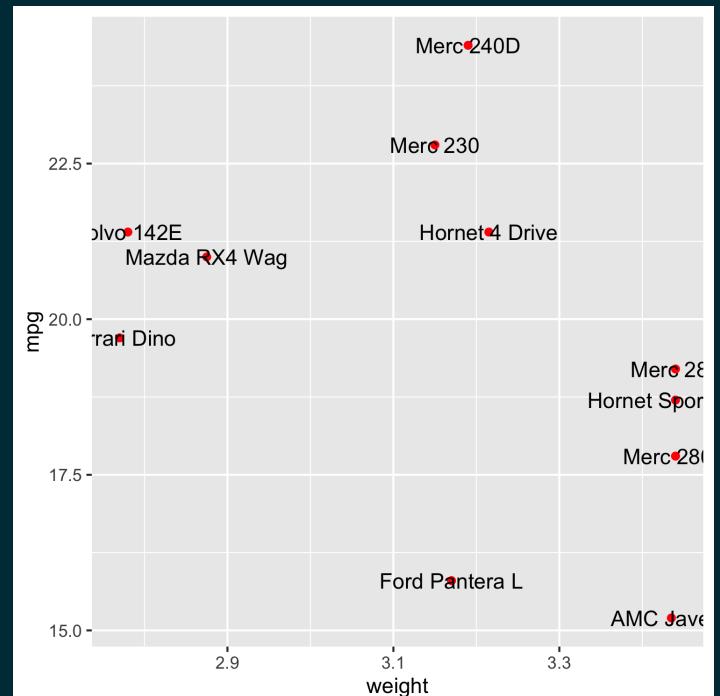
```
GGally::ggpairs(palmerpenguins::penguins)
```





```
d = tibble(  
  car = rownames(mtcars),  
  weight = mtcars$wt,  
  mpg = mtcars$mpg  
) %>%  
  filter(weight > 2.75, weight < 3.45)
```

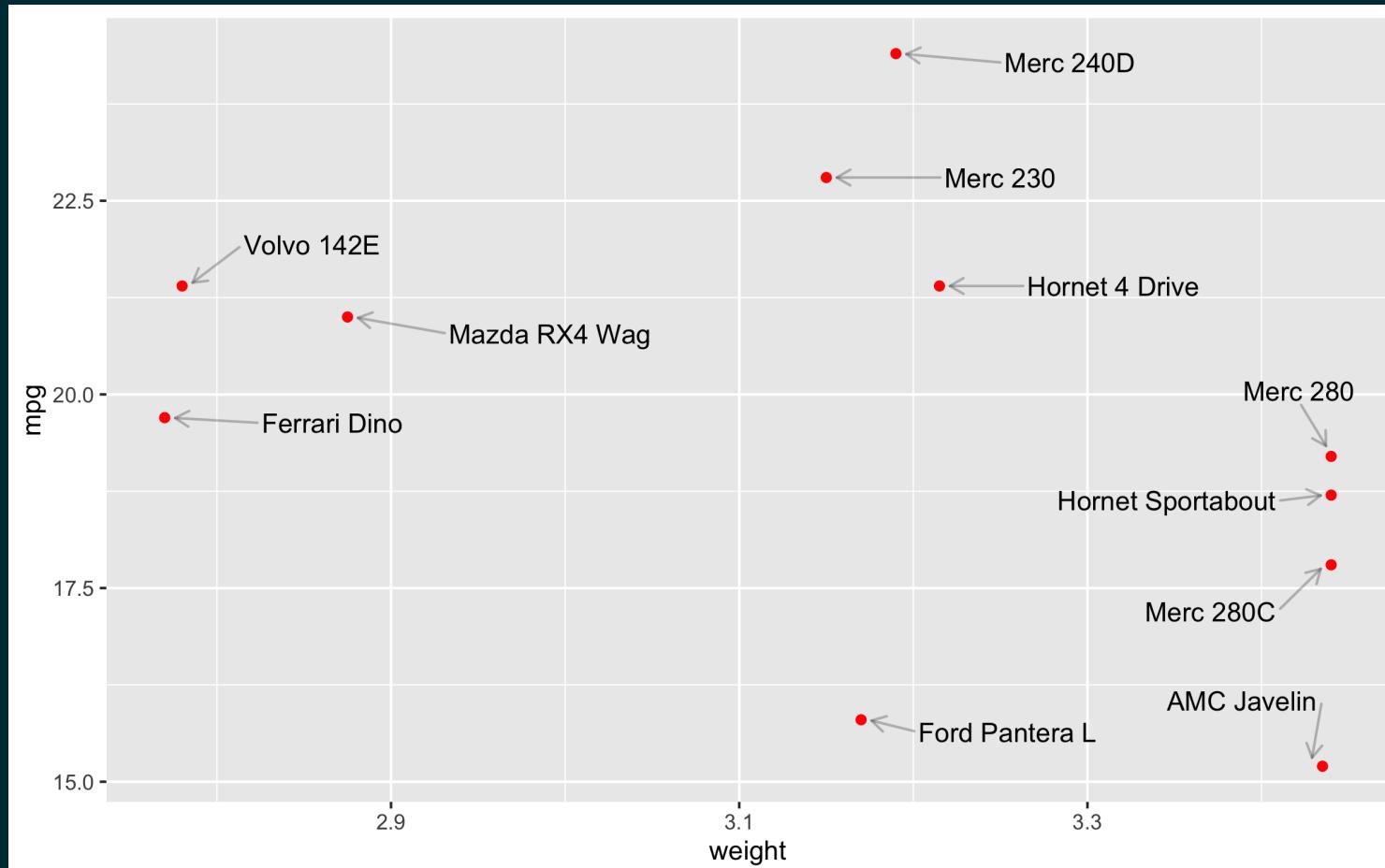
```
ggplot(d, aes(x=weight, y=mpg)) +  
  geom_point(color="red") +  
  geom_text(  
    aes(label = car)  
)
```



```
ggplot(d, aes(x=weight, y=mpg)) +  
  geom_point(color="red") +  
  ggrepel::geom_text_repel(  
    aes(label = car)  
)
```



```
ggplot(d, aes(x=weight, y=mpg)) +  
  geom_point(color="red") +  
  ggrepel::geom_text_repel(  
    aes(label = car),  
    nudge_x = .1, box.padding = 1, point.padding = 0.6,  
    arrow = arrow(length = unit(0.02, "npc")), segment.alpha = 0.25  
)
```





Plot objects

```
library(patchwork)

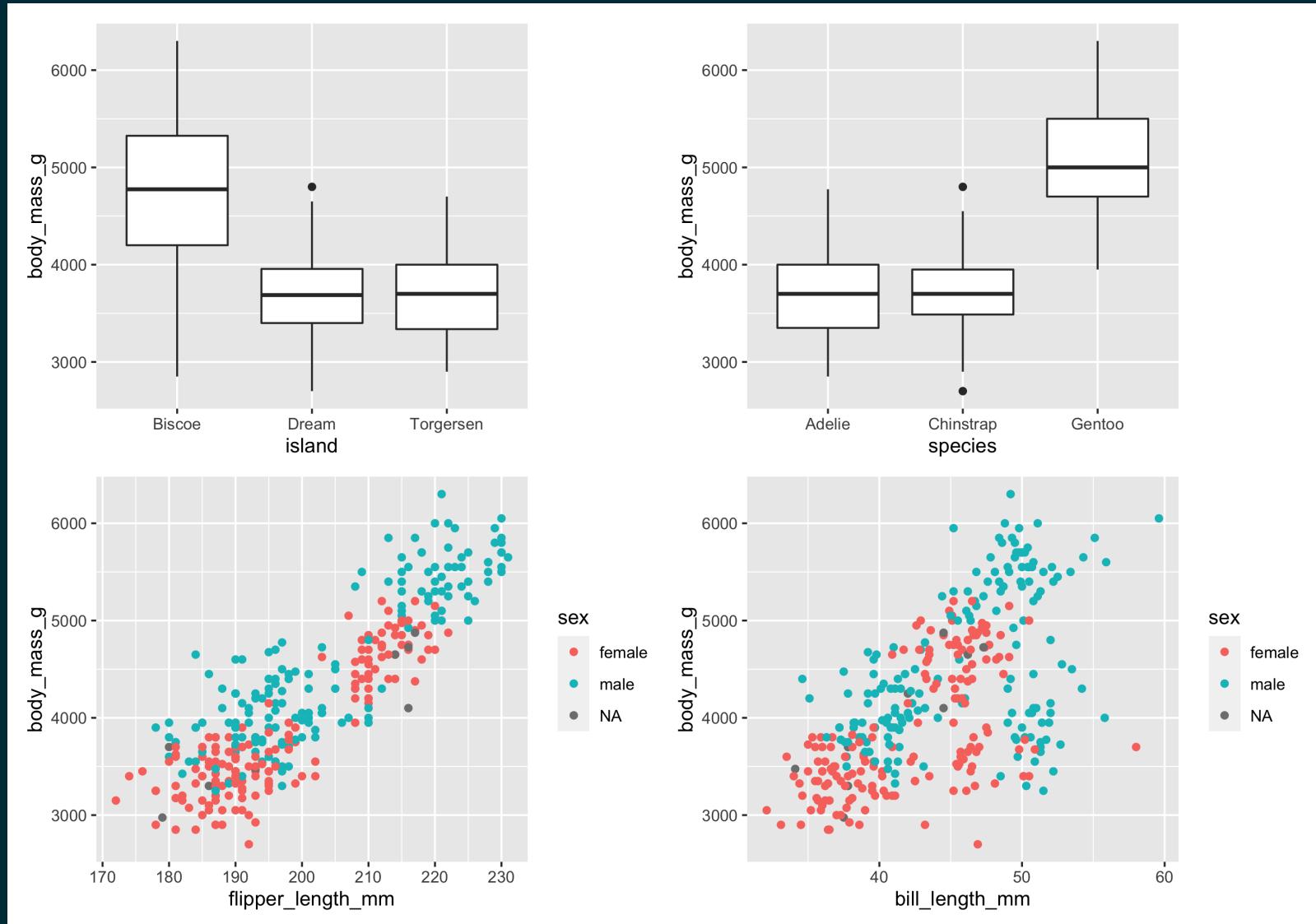
p1 = ggplot(palmerpenguins::penguins) +
  geom_boxplot(aes(x = island, y = body_mass_g))

p2 = ggplot(palmerpenguins::penguins) +
  geom_boxplot(aes(x = species, y = body_mass_g))

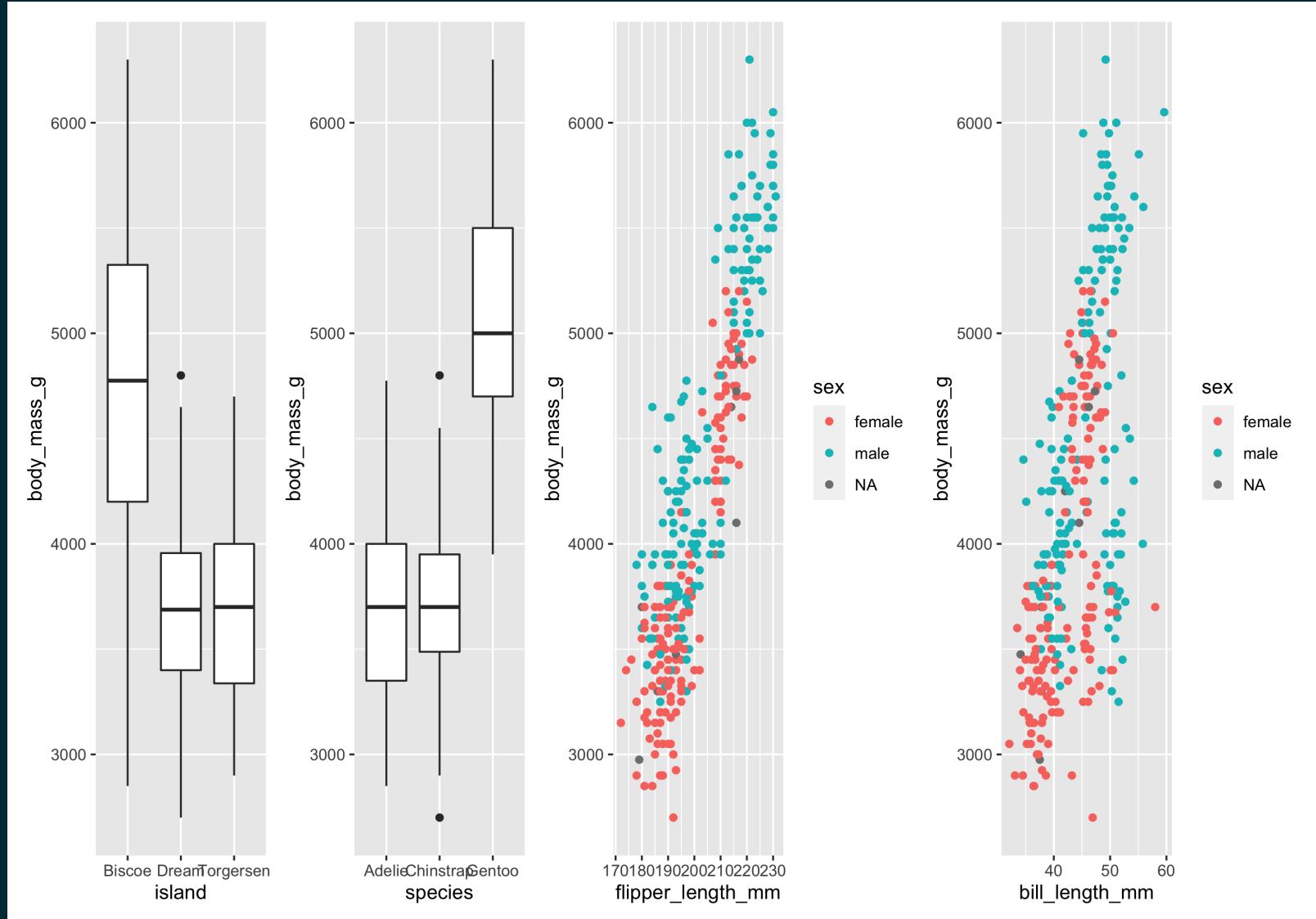
p3 = ggplot(palmerpenguins::penguins) +
  geom_point(aes(x = flipper_length_mm, y = body_mass_g, color = sex))

p4 = ggplot(palmerpenguins::penguins) +
  geom_point(aes(x = bill_length_mm, y = body_mass_g, color = sex))
```

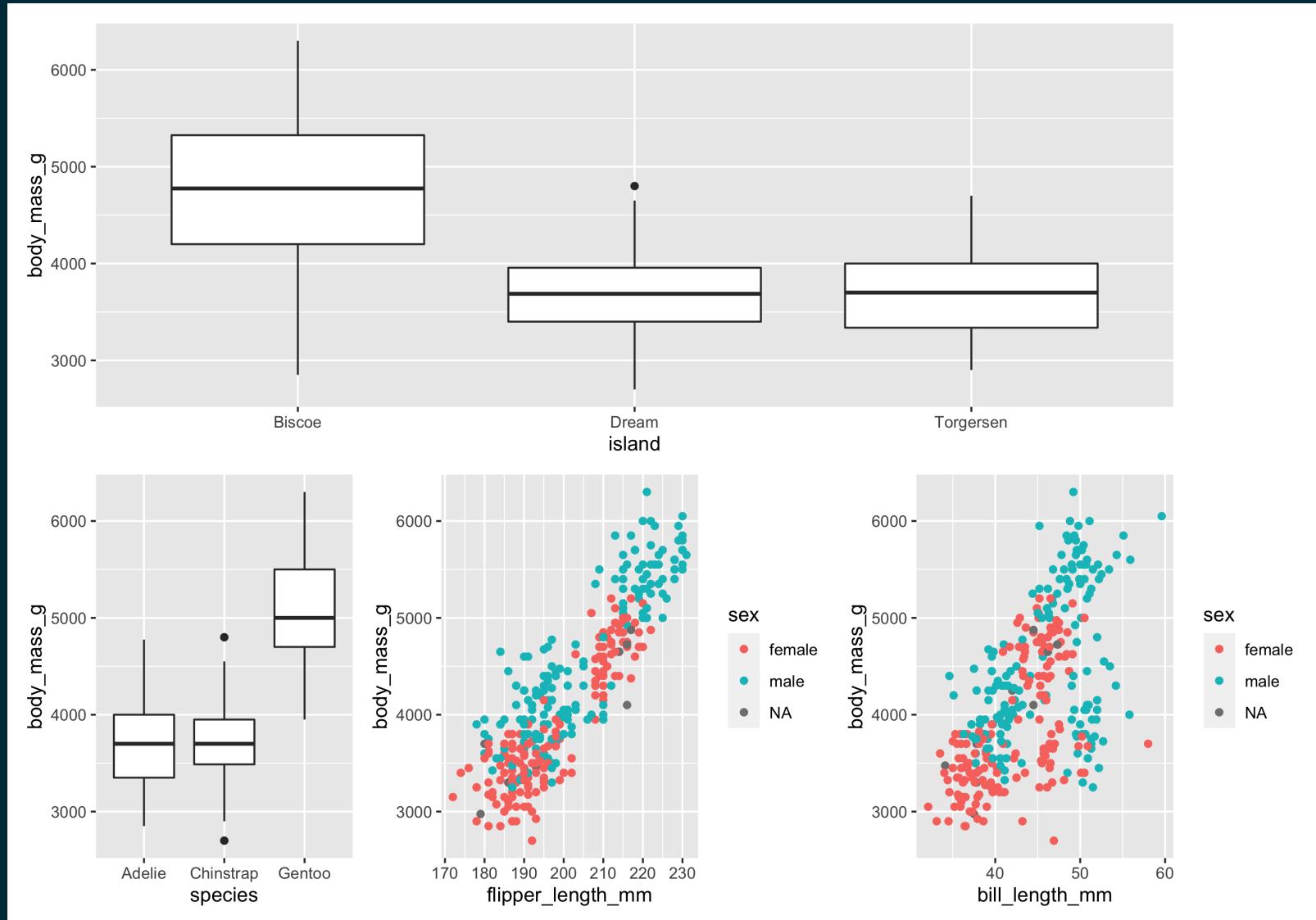
p1 + p2 + p3 + p4



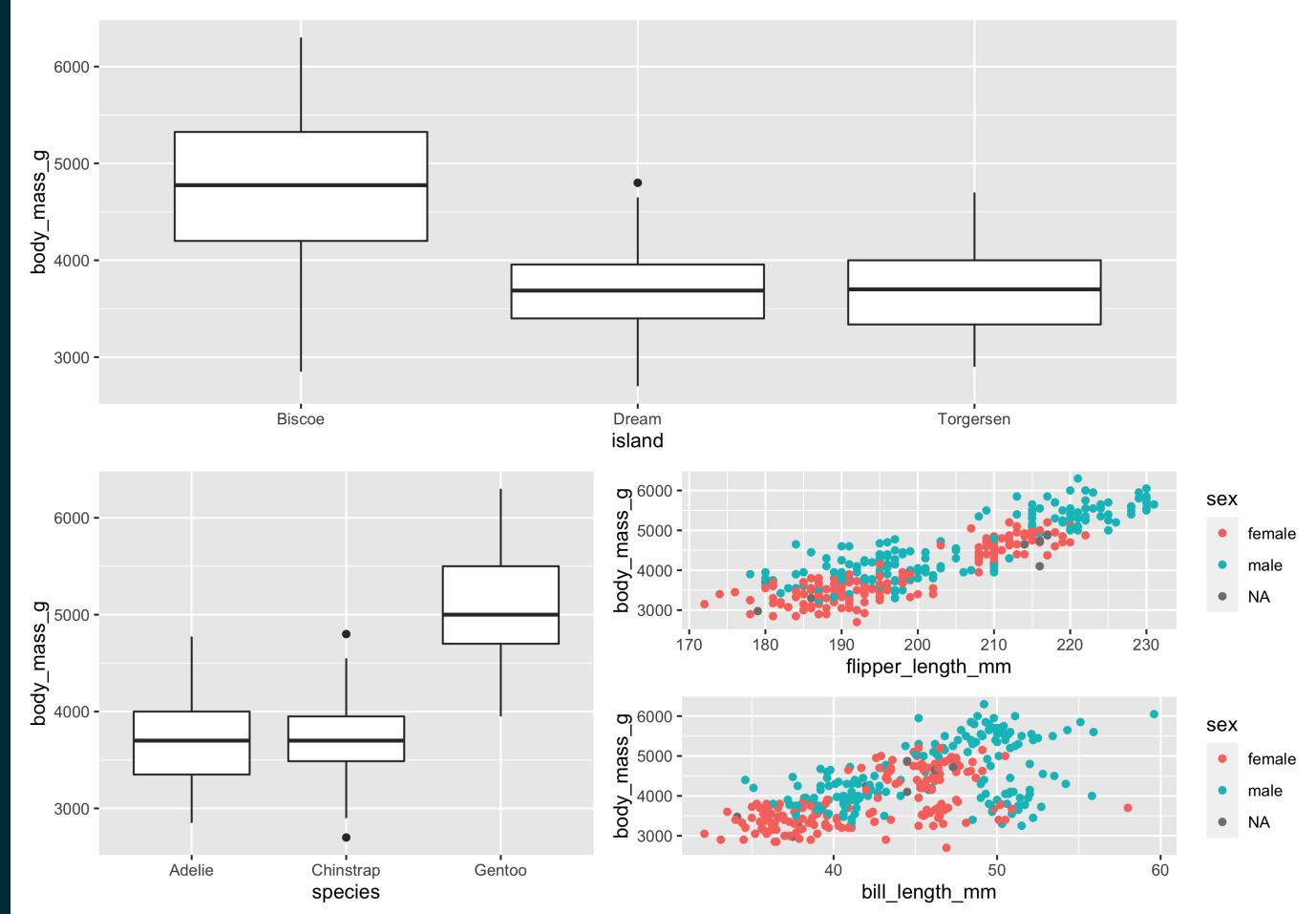
p1 + p2 + p3 + p4 + plot_layout(nrow=1)



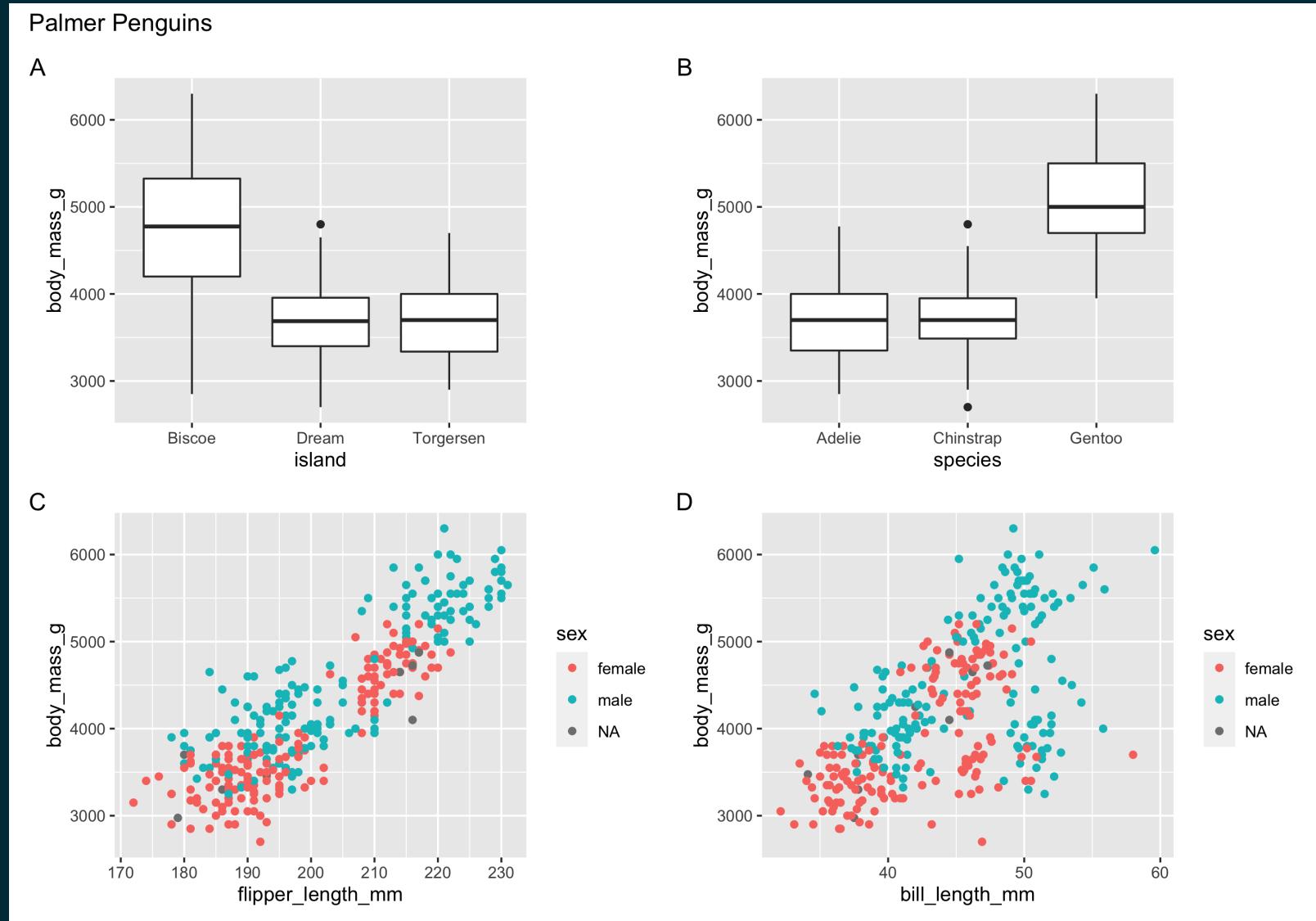
p1 / (p2 + p3 + p4)



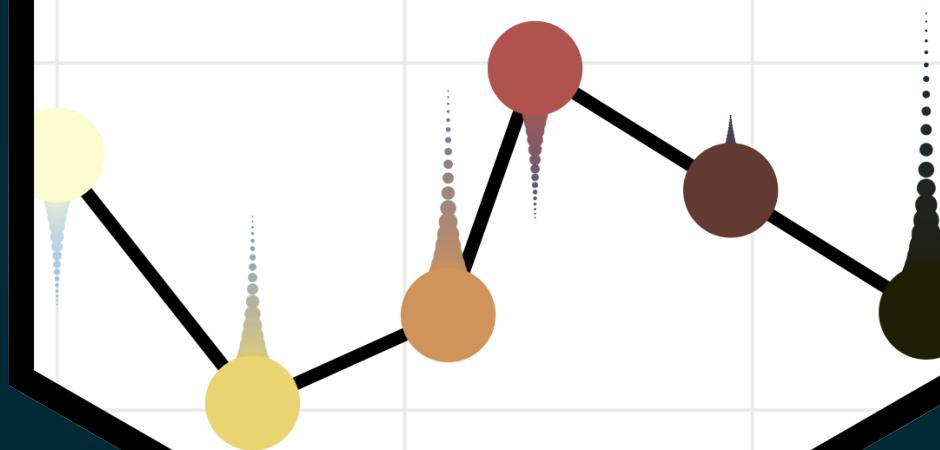
```
p1 + {  
  p2 + {  
    p3 + p4 + plot_layout(ncol = 1)  
  }  
} + plot_layout(ncol = 1)
```



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



ganimate

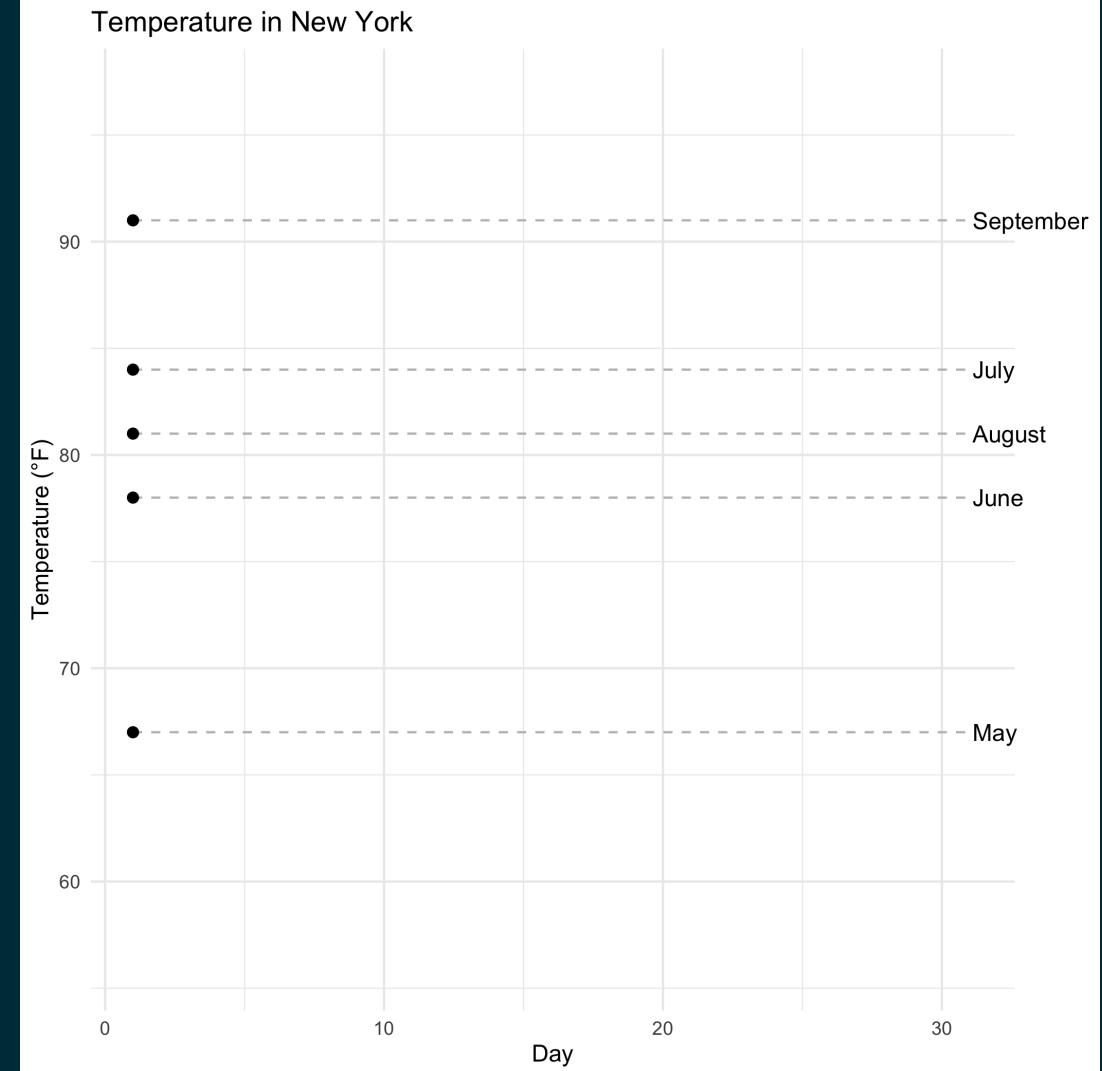


```

airq = airquality
airq$Month = month.name[airq$Month]

ggplot(
  airq,
  aes(Day, Temp, group = Month)
) +
  geom_line() +
  geom_segment(
    aes(xend = 31, yend = Temp),
    linetype = 2,
    colour = 'grey'
  ) +
  geom_point(size = 2) +
  geom_text(
    aes(x = 31.1, label = Month),
    hjust = 0
  ) +
  gganimate::transition_reveal(Day) +
  coord_cartesian(clip = 'off') +
  labs(
    title = 'Temperature in New York',
    y = 'Temperature (°F)'
  ) +
  theme_minimal() +
  theme(plot.margin = margin(5.5, 40, 5.5, 5.5))

```



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

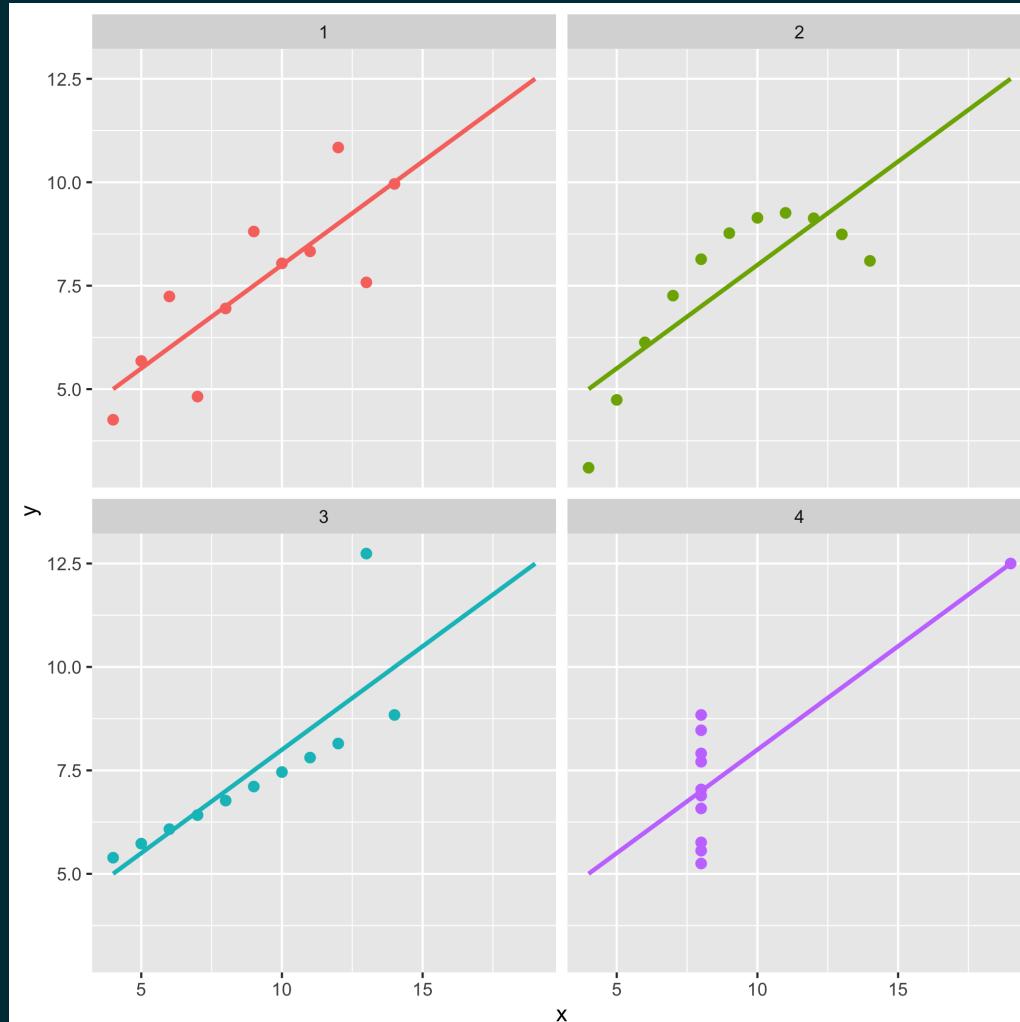
```
(tidy_anscombe = datasets::anscombe %>%
  pivot_longer(everything(), names_sep = 1, names_to = c("var", "group")) %>%
  pivot_wider(id_cols = group, names_from = var,
              values_from = value, values_fn = list(value = list)) %>%
  unnest(cols = c(x,y)))
```

```
## # A tibble: 44 x 3
##   group     x     y
##   <chr> <dbl> <dbl>
## 1 1       10  8.04
## 2 1       8   6.95
## 3 1       13  7.58
## 4 1       9   8.81
## 5 1       11  8.33
## 6 1       14  9.96
## 7 1       6   7.24
## 8 1       4   4.26
## 9 1       12  10.8
## 10 1      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), .groups = "drop"
  )
```

```
## # 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(size=2) +  
  facet_wrap(~group) +  
  geom_smooth(method="lm", se=FALSE, fullrange=TRUE, formula = y~x) +  
  guides(color=FALSE)
```

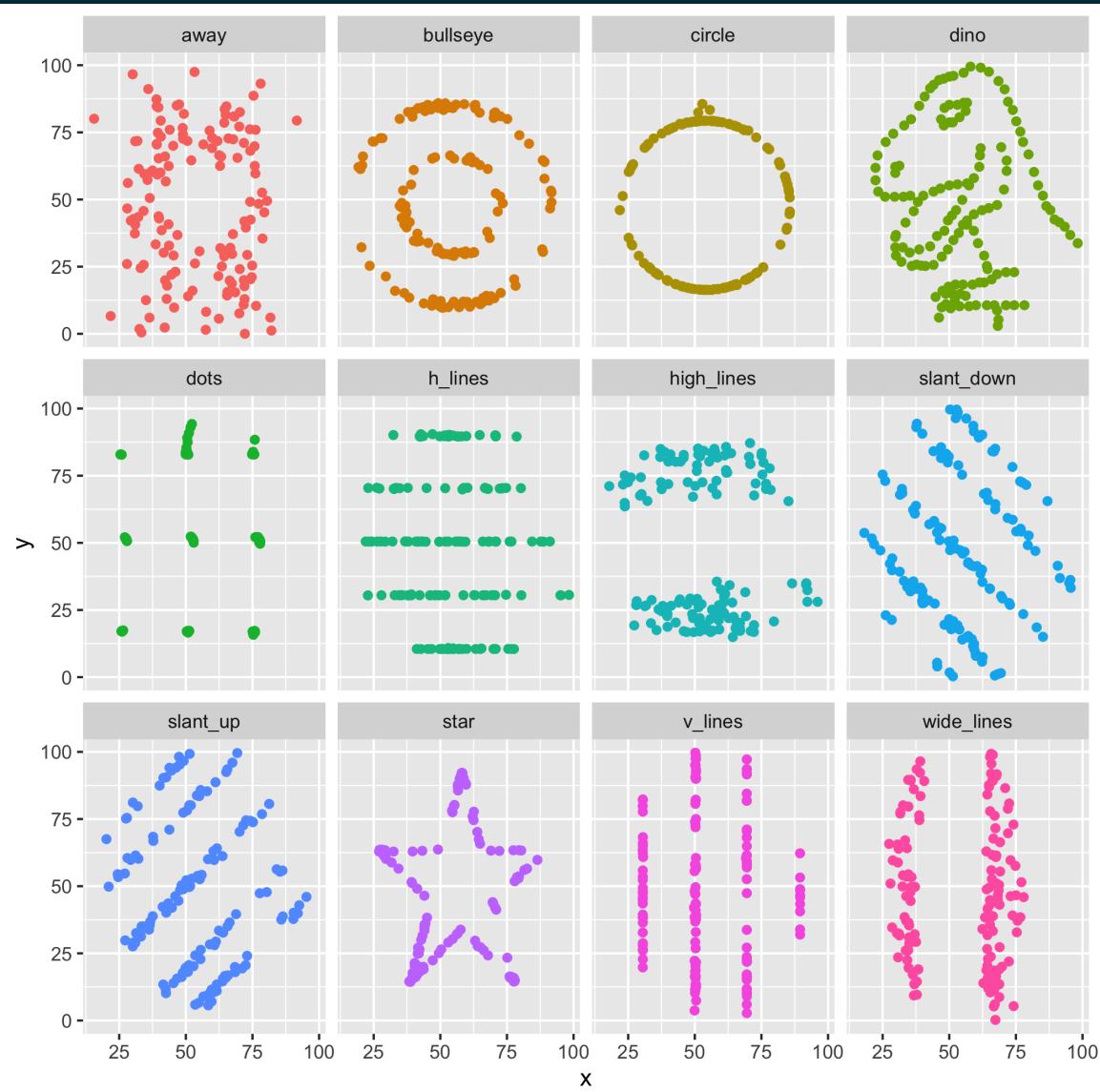


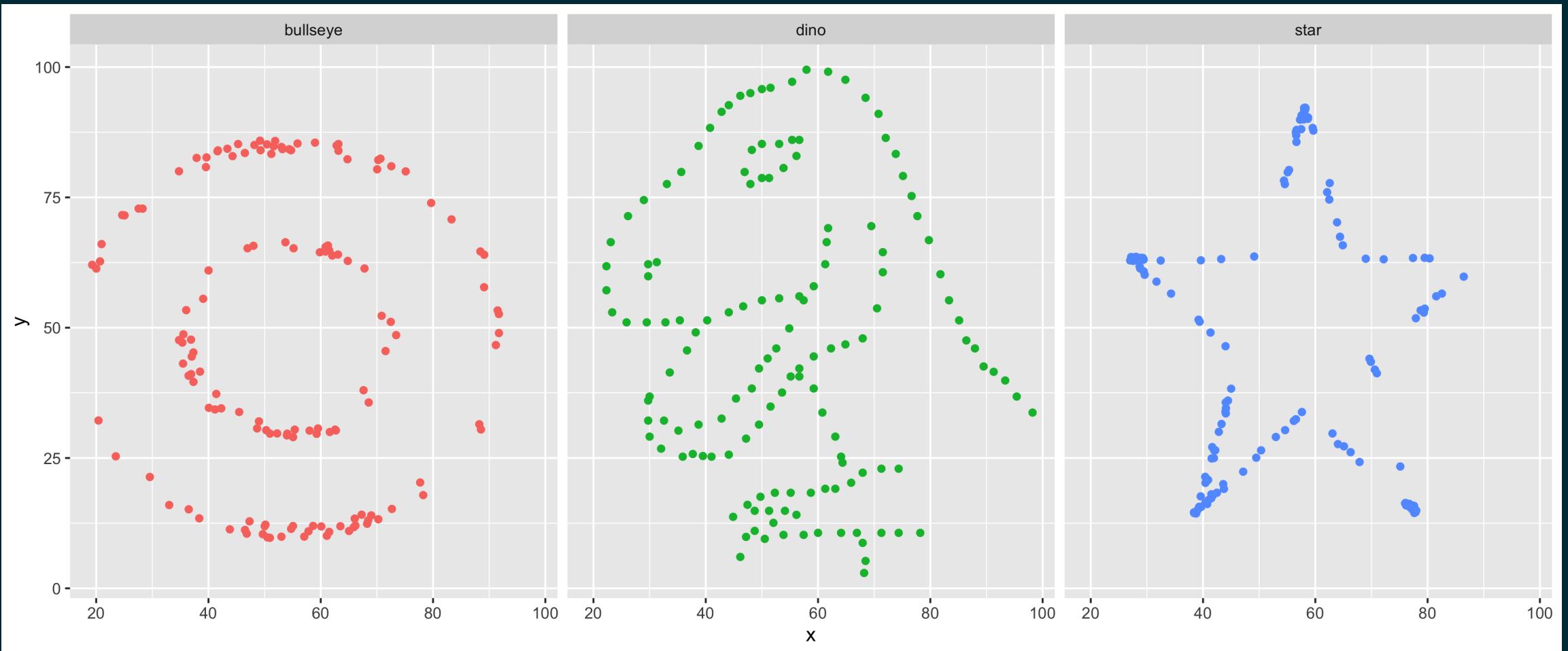
DatasauRus

```
library(datasauRus)
```

```
##  
## Attaching package: 'datasauRus'  
## The following object is masked _by_ '.GlobalEnv':  
##  
##     datasaurus_dozen
```

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





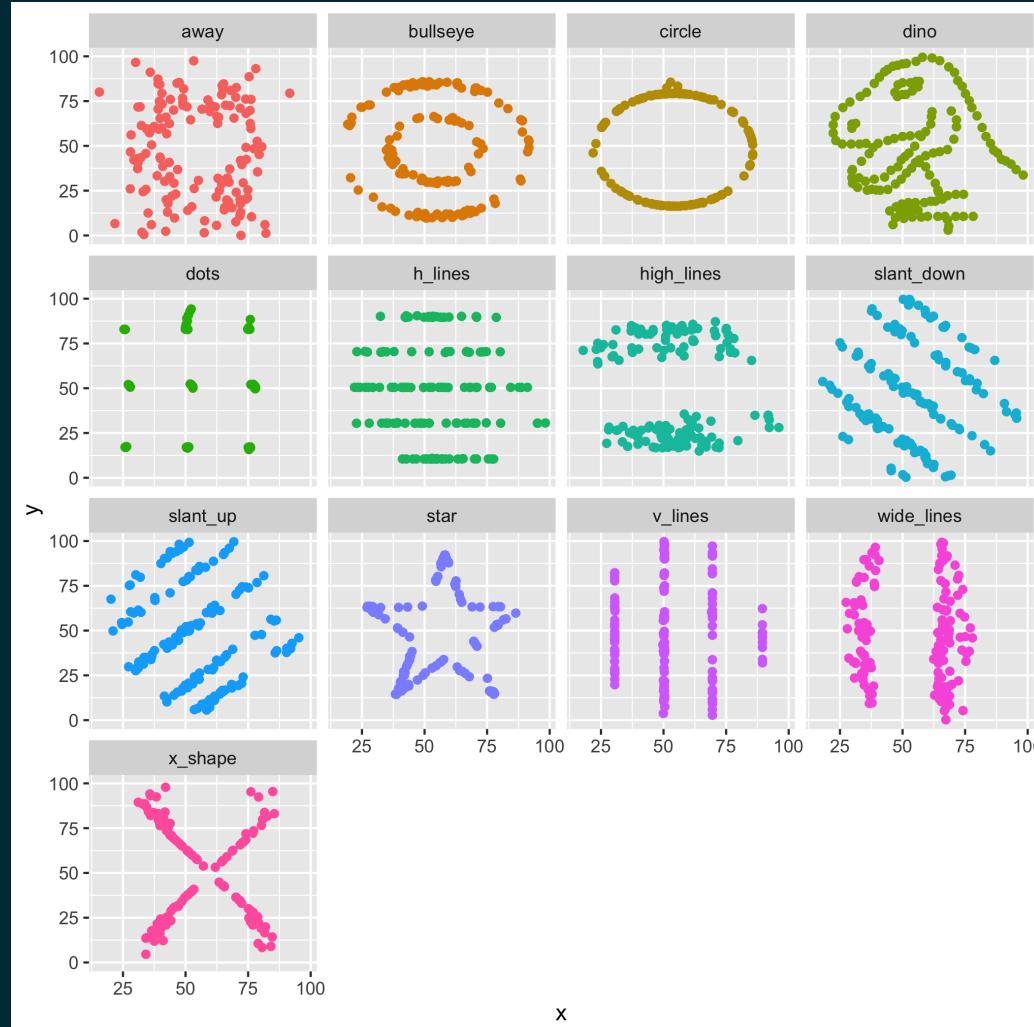
```
datasauRus::datasaurus_dozen
```

```
## # 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
```

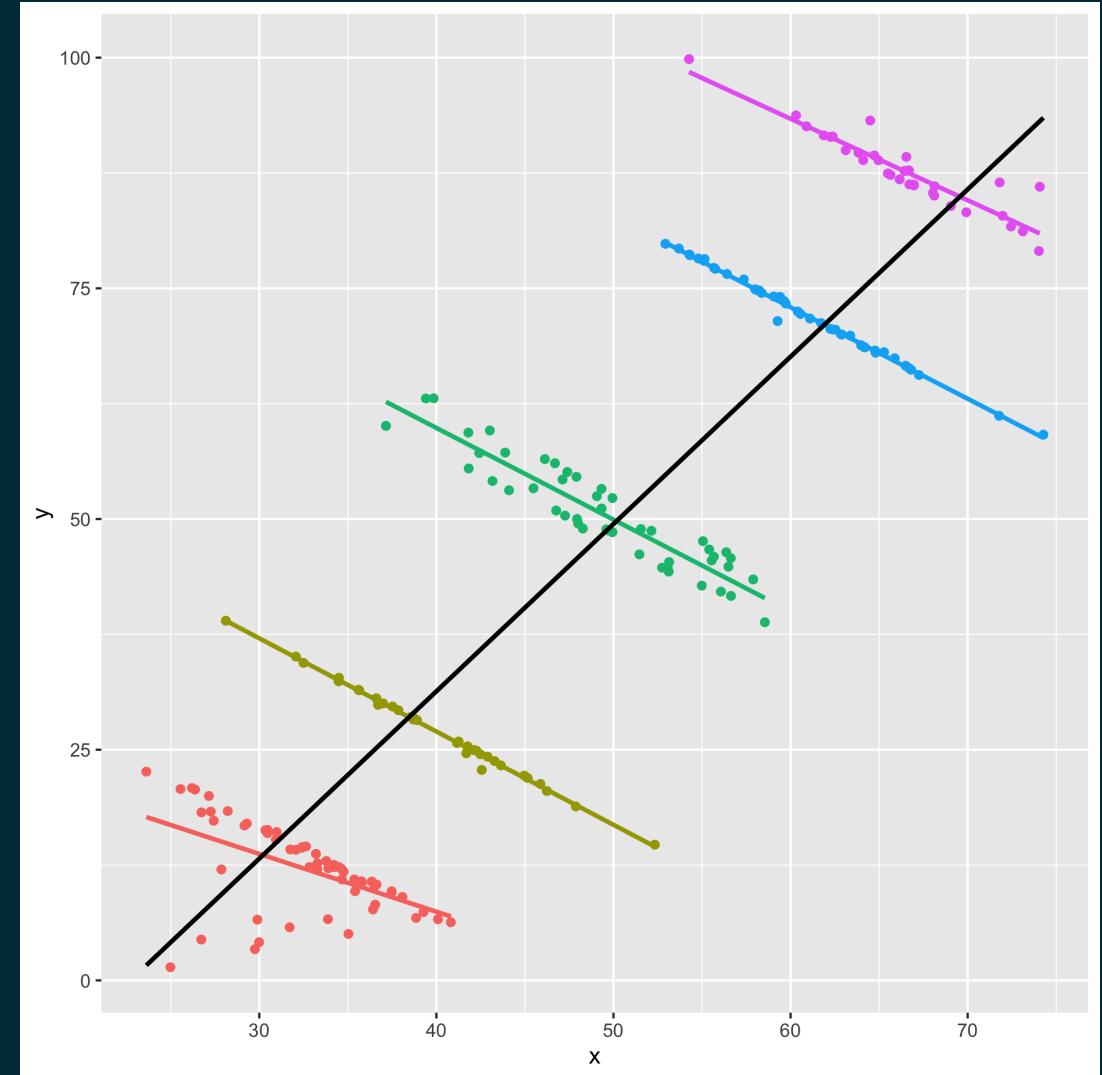
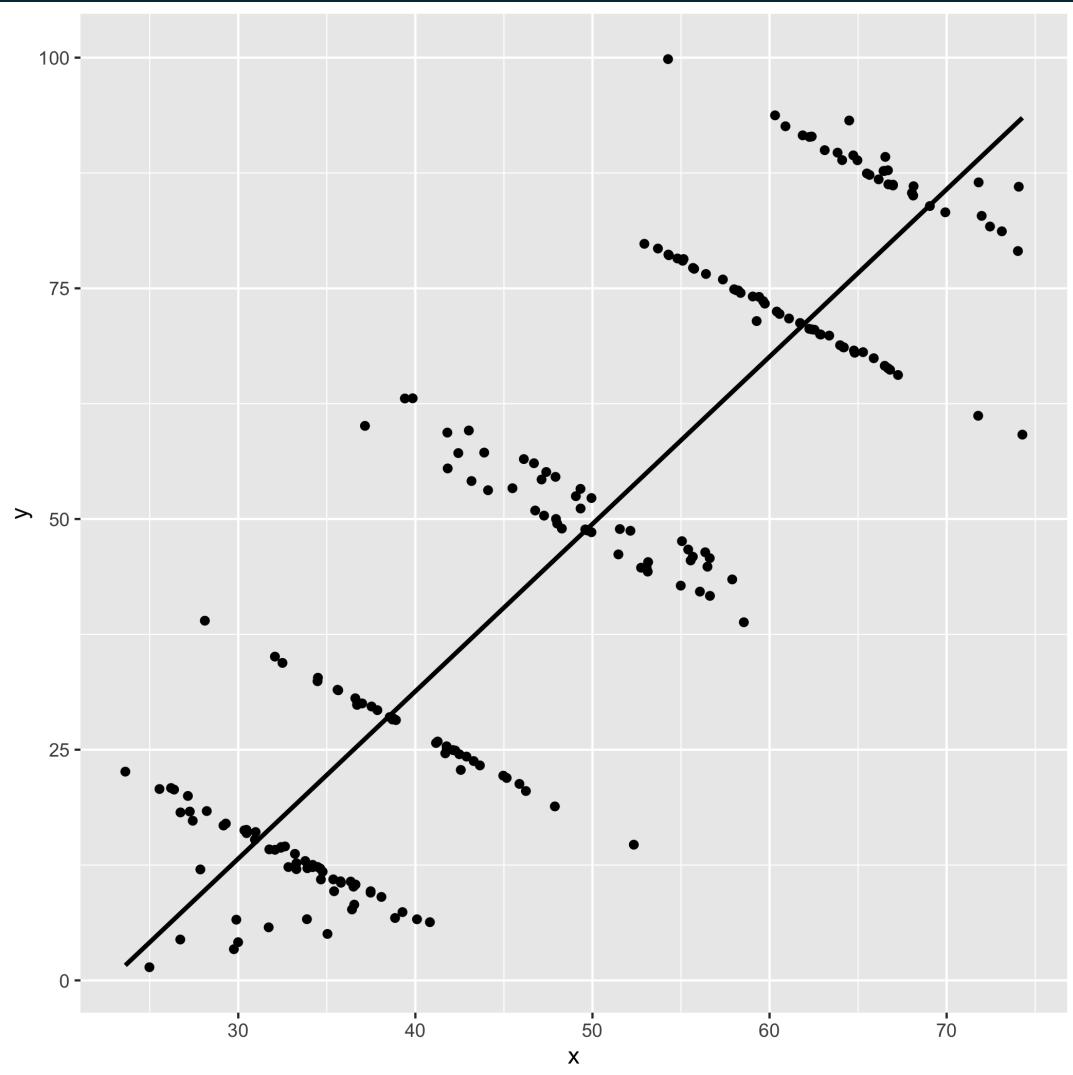
```
datasaurus_dozen %>%
  group_by(dataset) %>%
  summarize(mean_x = mean(x), mean_y = mean(y),
            sd_x = sd(x), sd_y = sd(y),
            cor = cor(x,y), .groups = "drop")
```

```
## # A tibble: 12 x 6
##   dataset    mean_x  mean_y   sd_x   sd_y     cor
##   <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 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
```

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

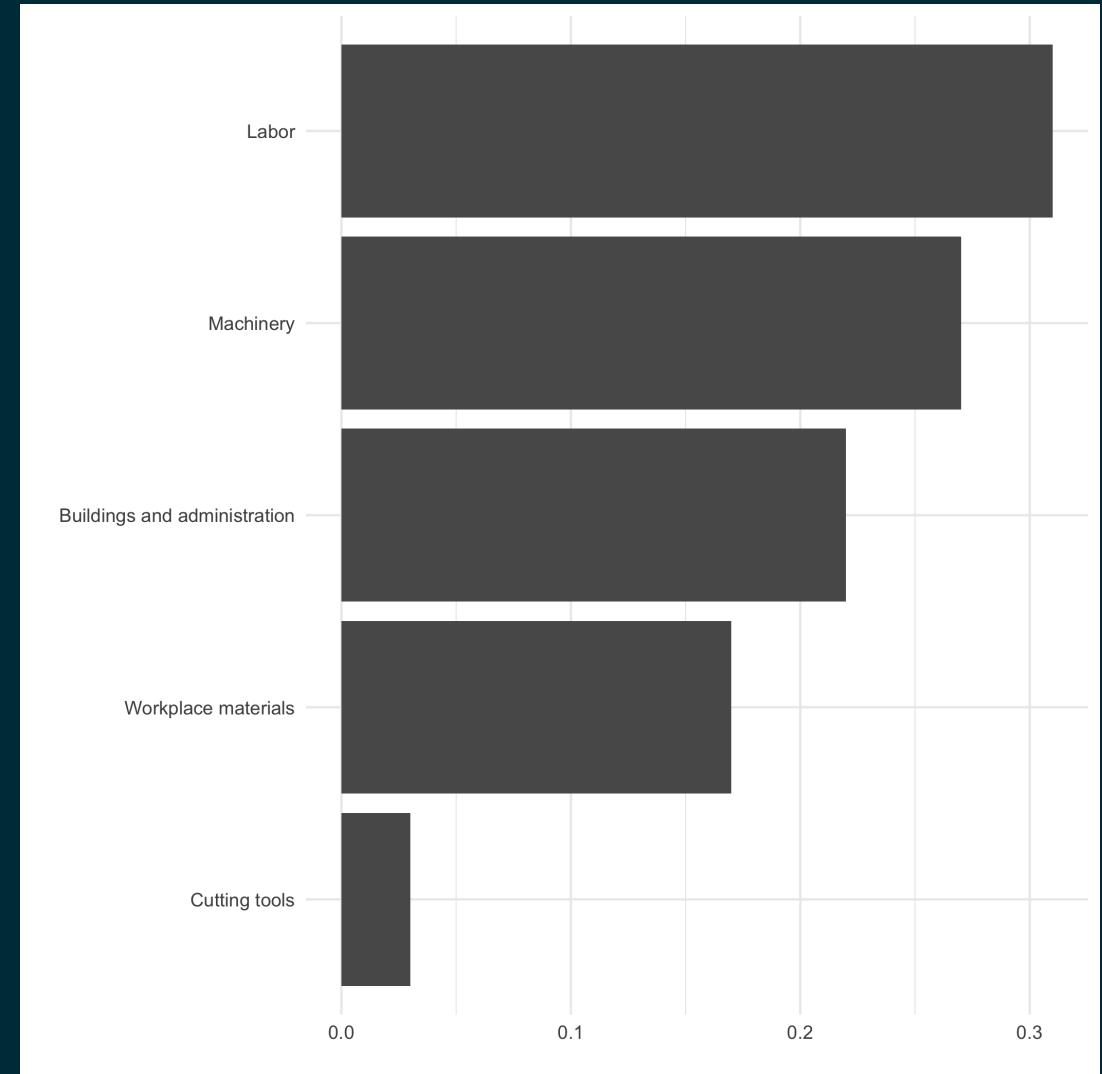


Simpson's Paradox

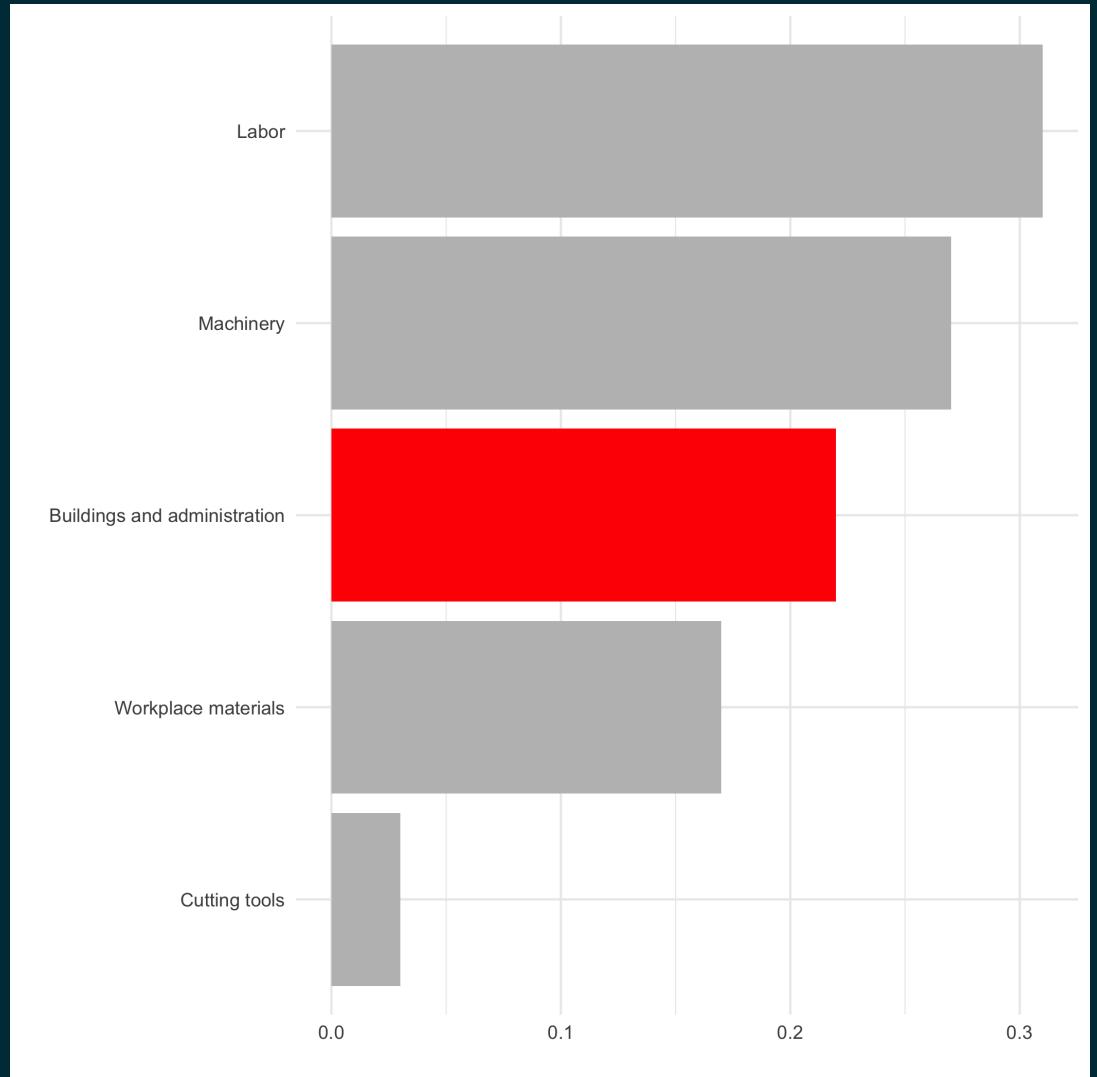
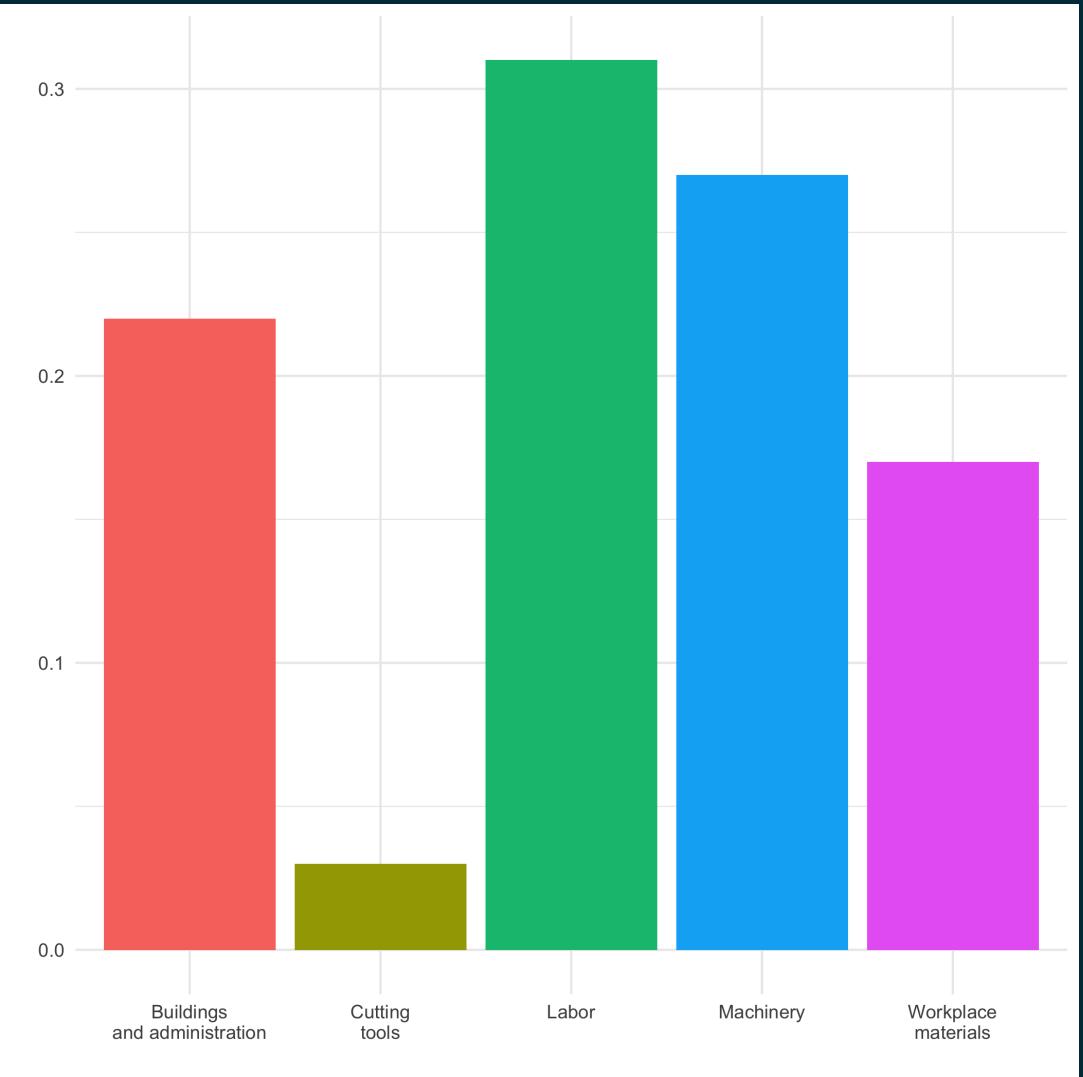


Designing effective visualizations

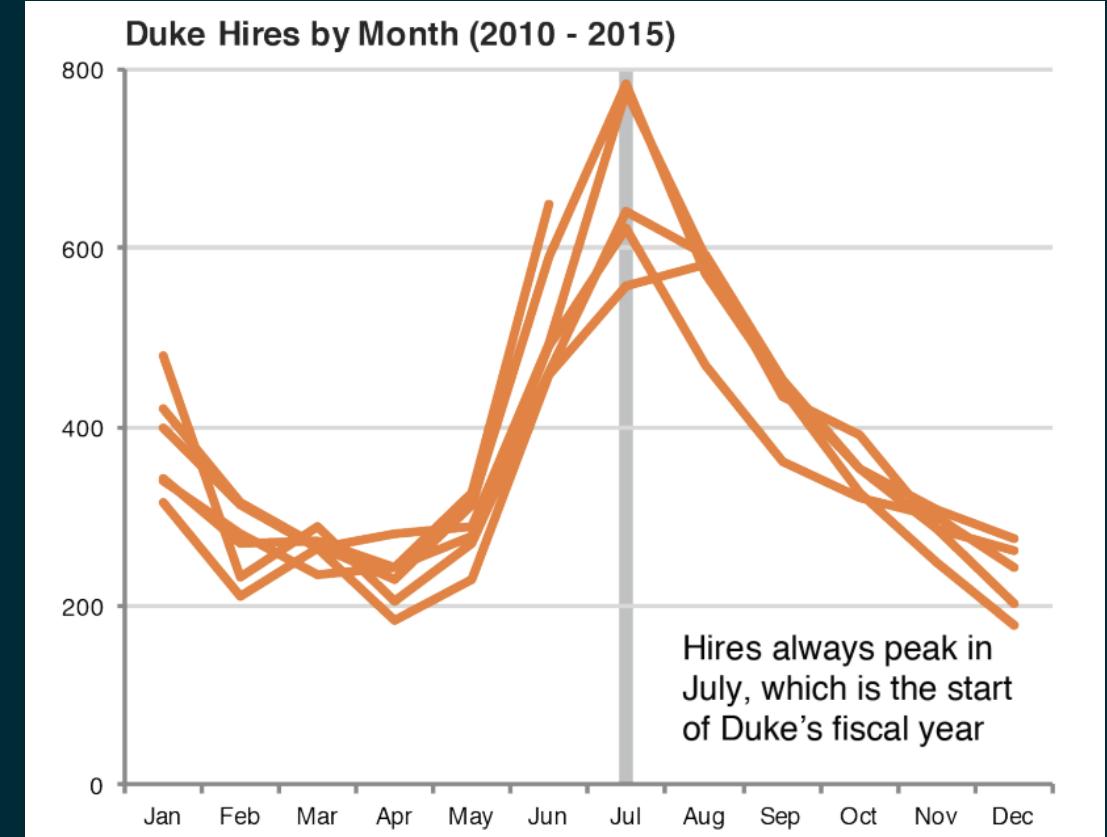
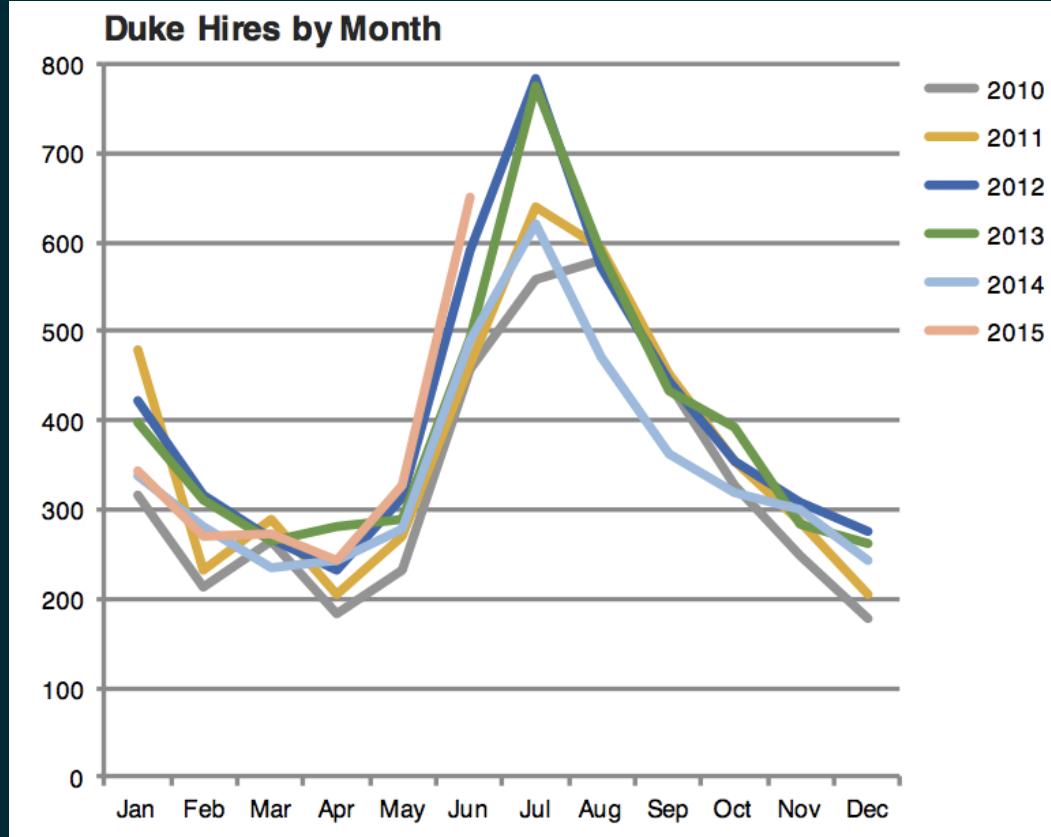
Keep it simple



Use color to draw attention

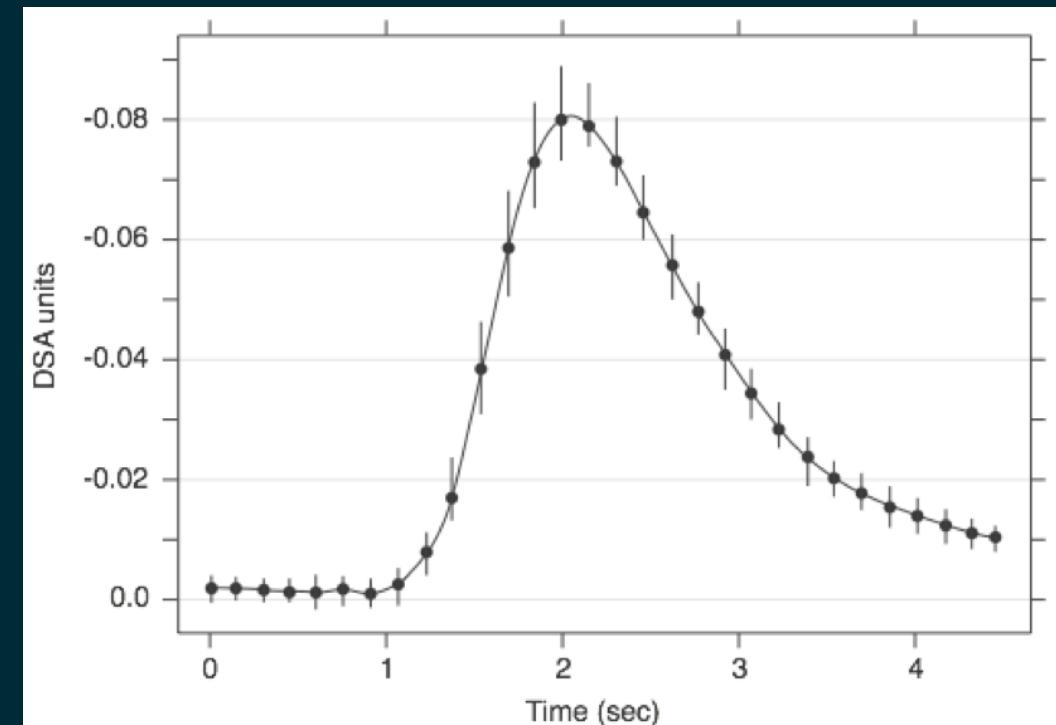
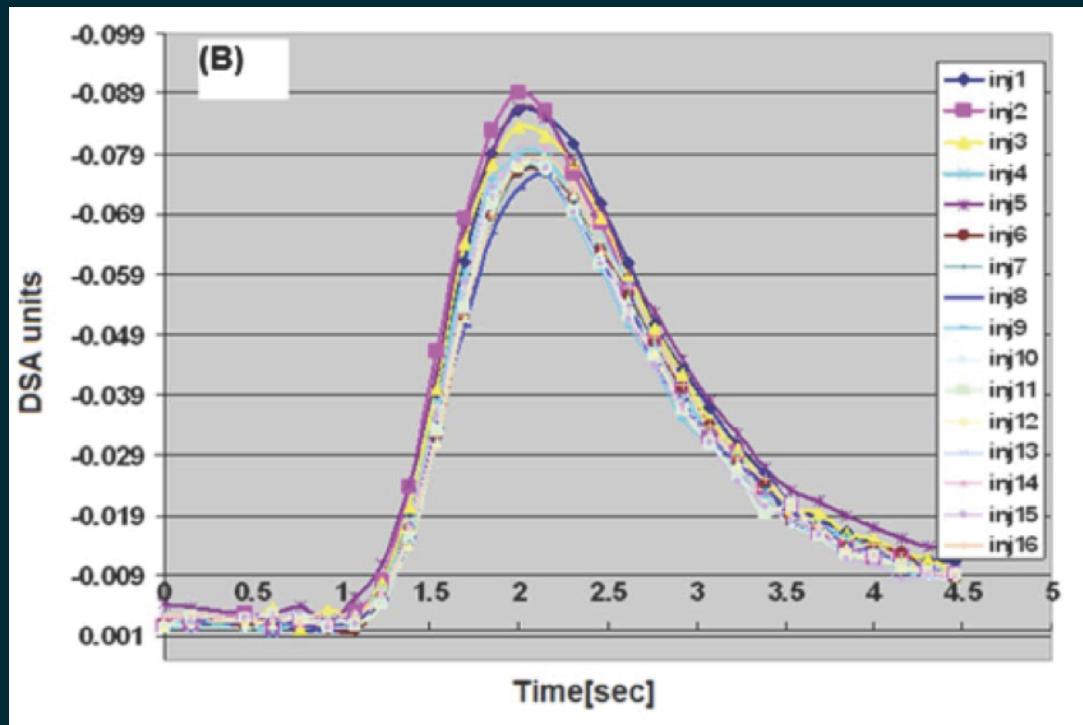


Tell a story



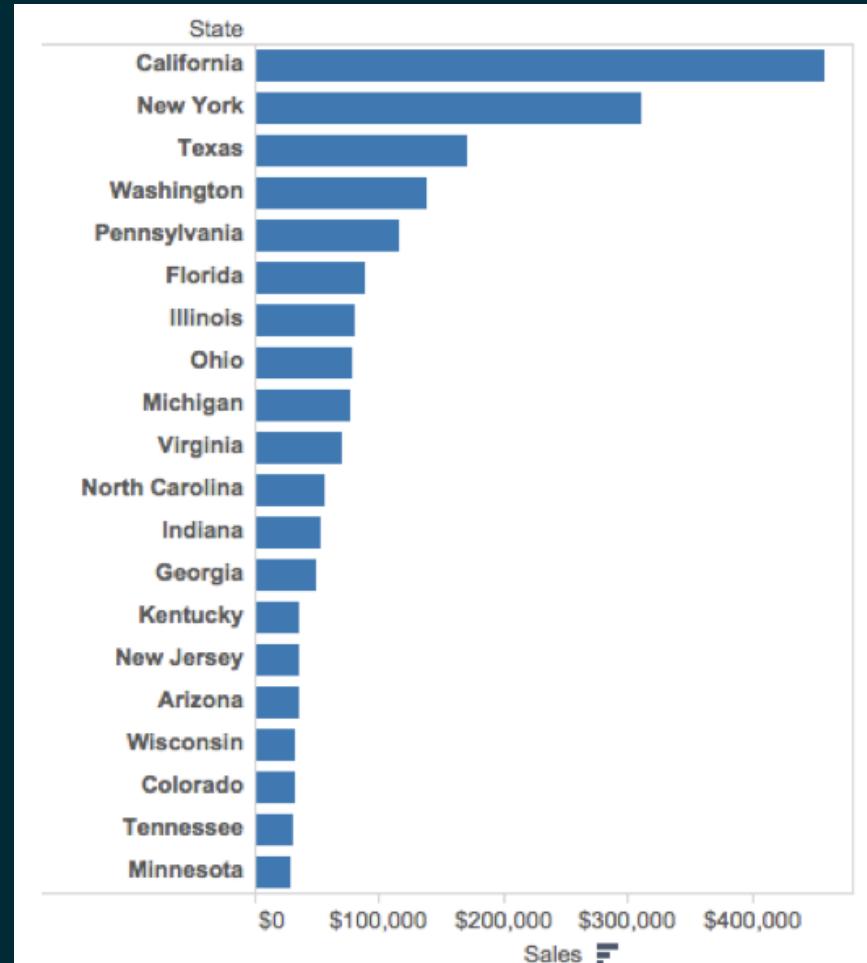
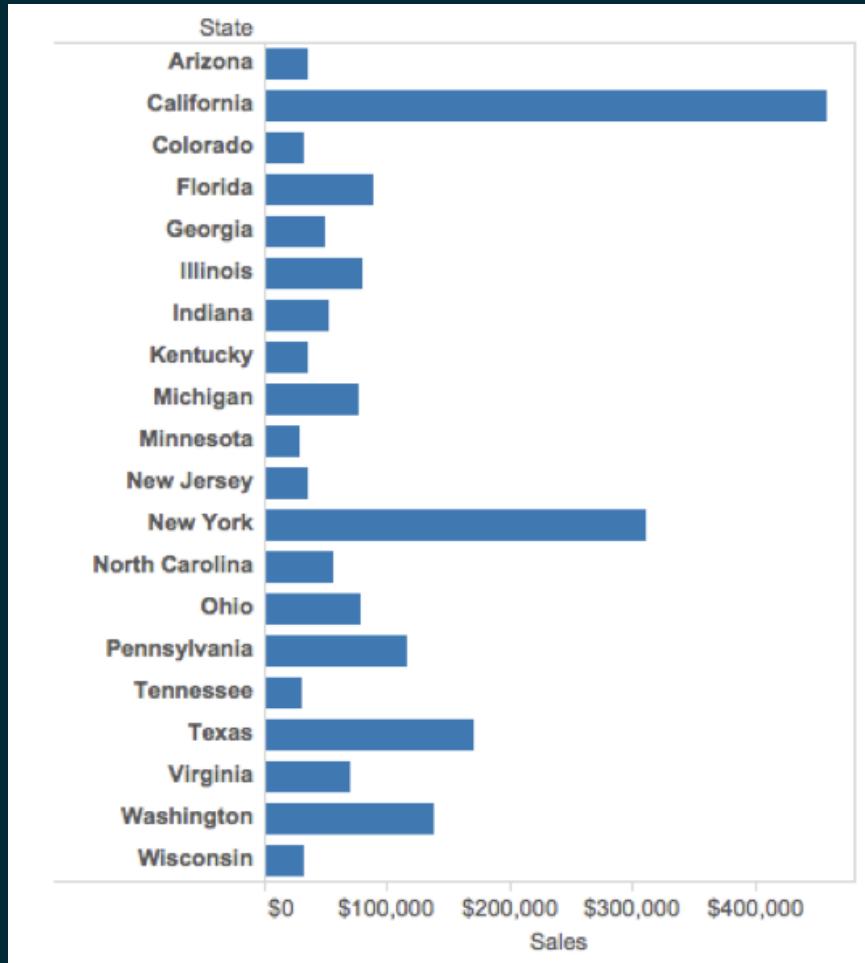
Credit: Angela Zoss and Eric Monson, Duke DVS

Leave out non-story details



Credit: Angela Zoss and Eric Monson, Duke DVS

Ordering matter



Credit: Angela Zoss and Eric Monson, Duke DVS

Clearly indicate missing data

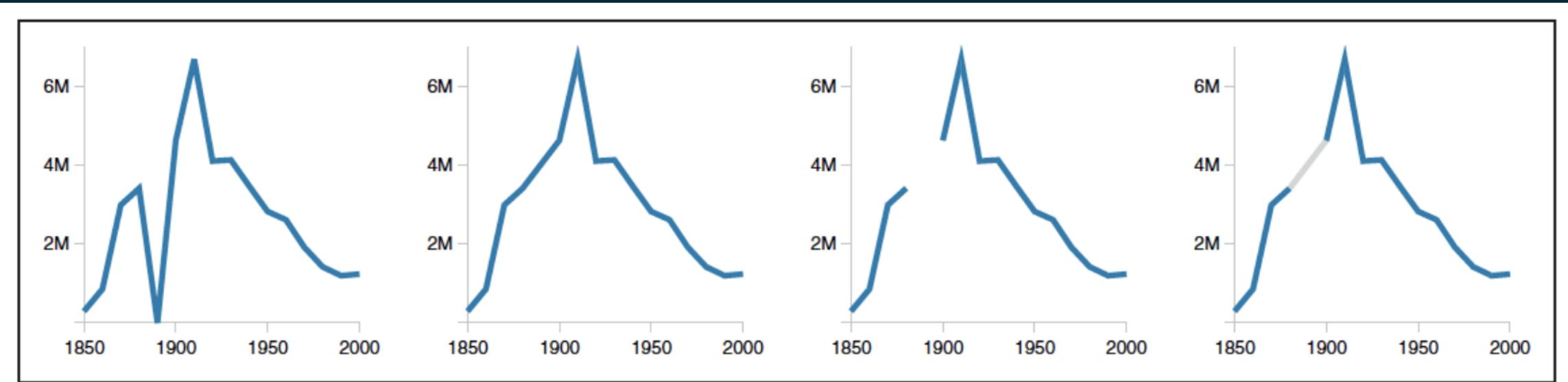
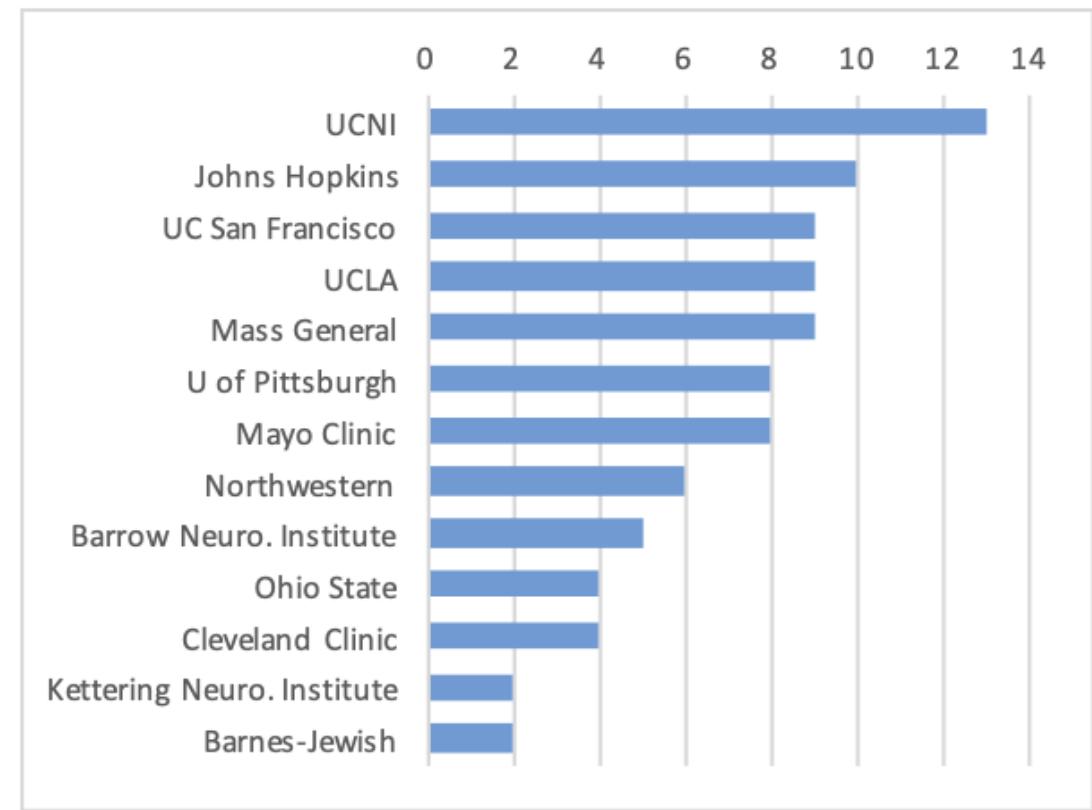
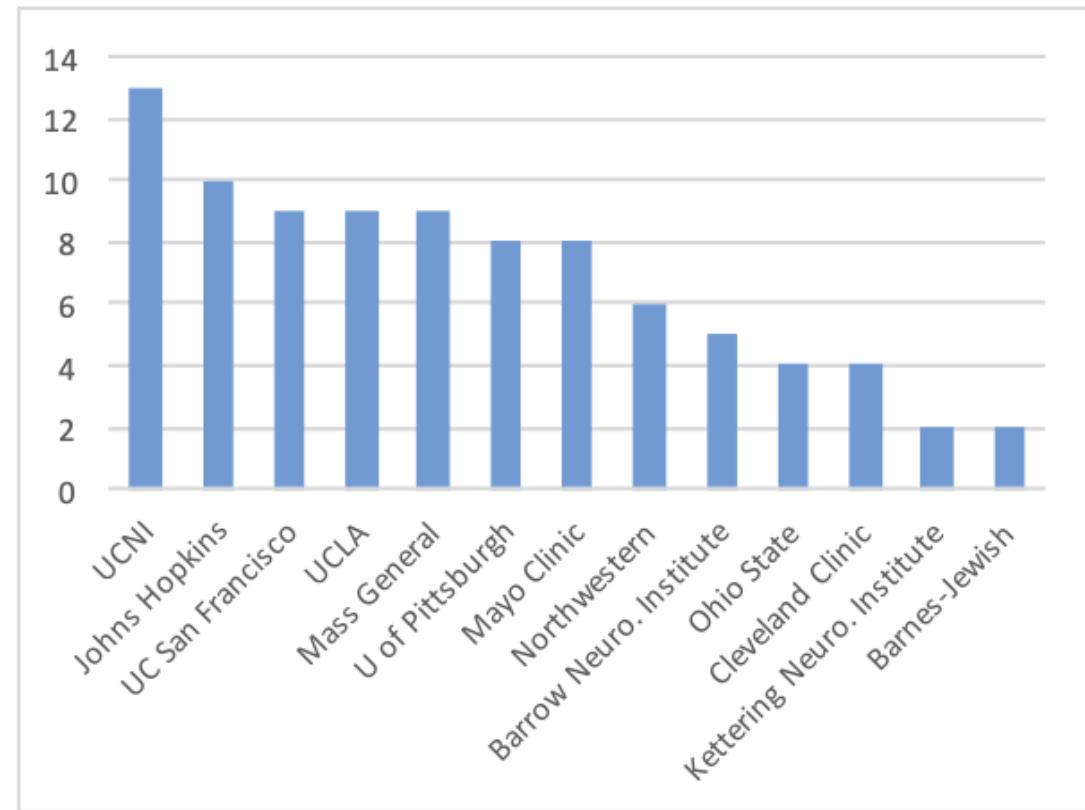


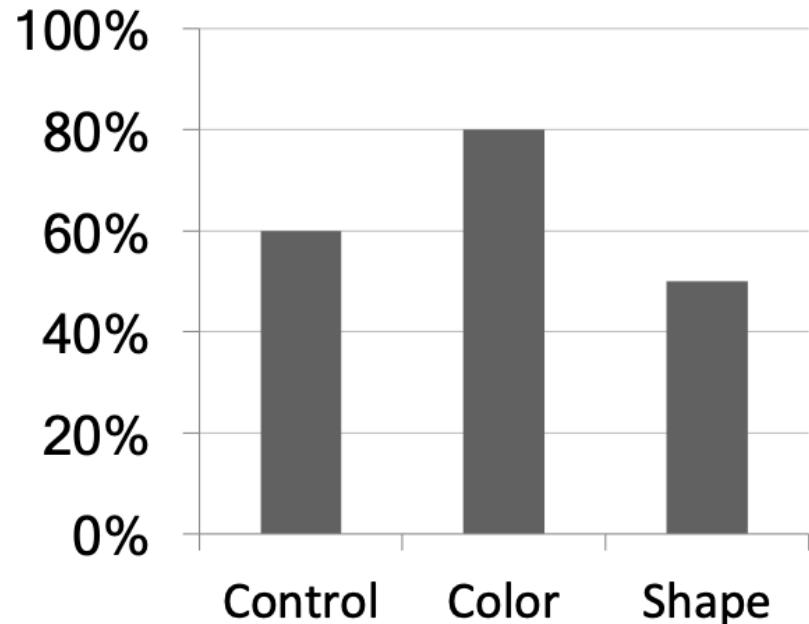
Figure 4. Alternative representations of missing data in a line chart. The data are U.S. census counts of people working as 'Farm Laborers'; values from 1890 are missing due to records being burned in a fire. (a) Missing data is treated as a zero value. (b) Missing data is ignored, resulting in a line 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 load

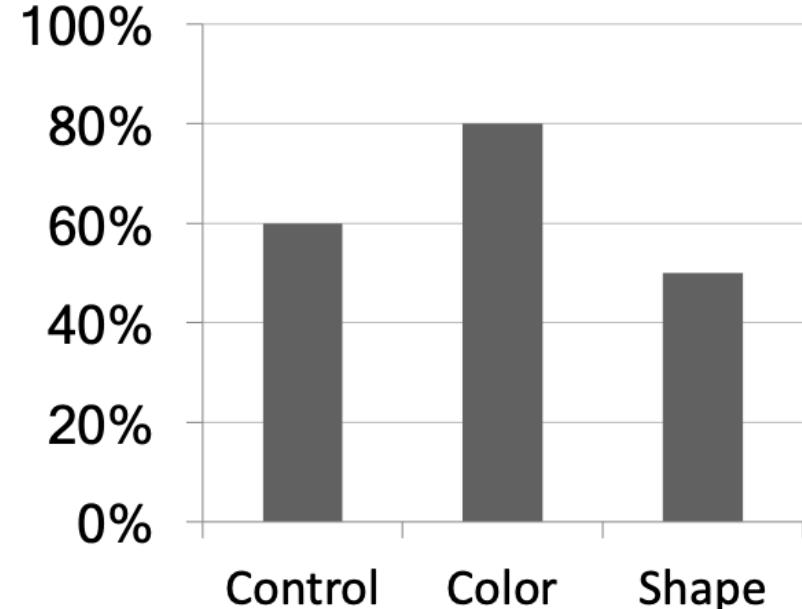


Use descriptive titles

**Accuracy versus
Color and Shape**



**Accuracy Improved by
Color, not Shape**

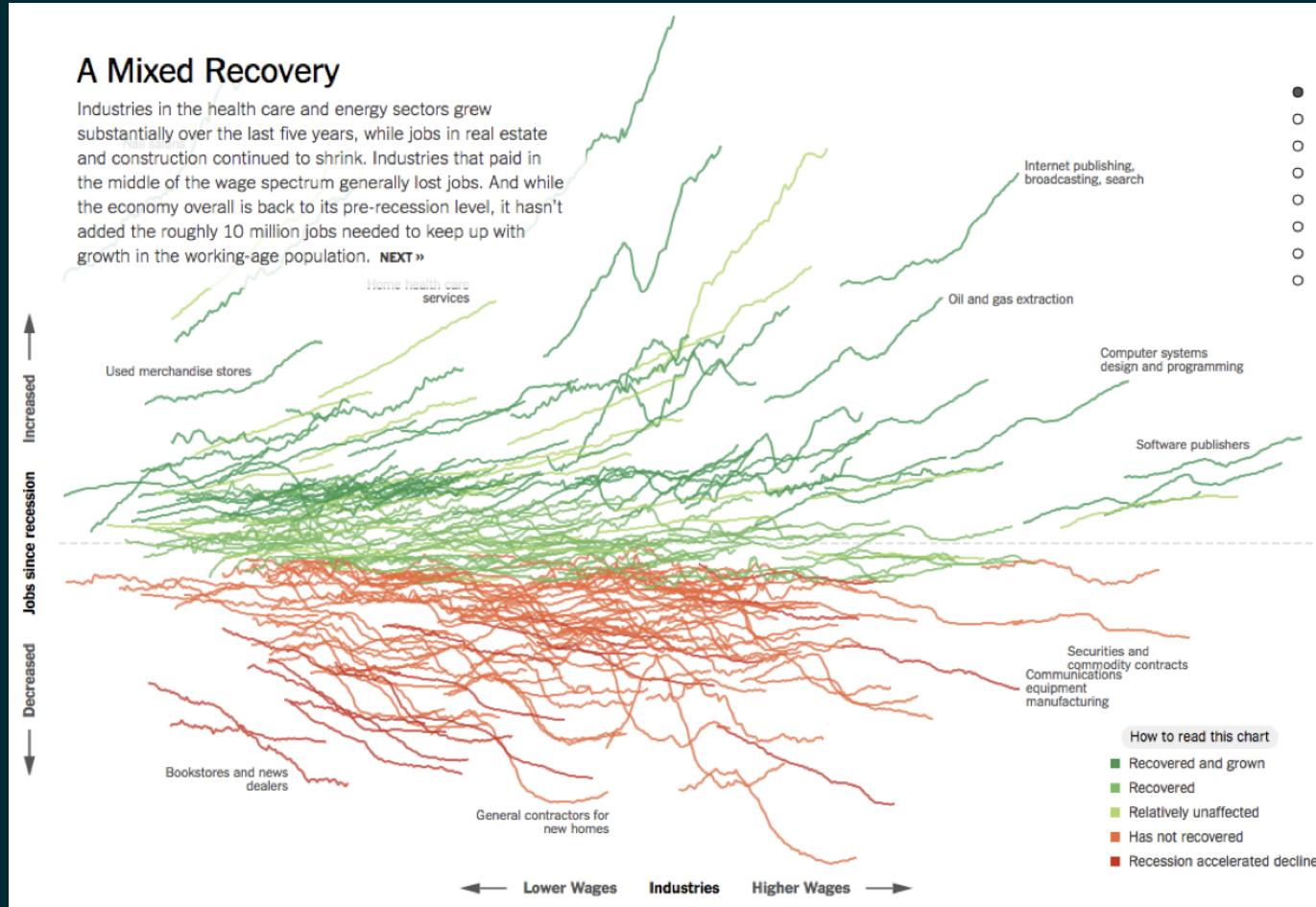


Credit: Angela Zoss and Eric Monson, Duke DVS

Annotate figures directly

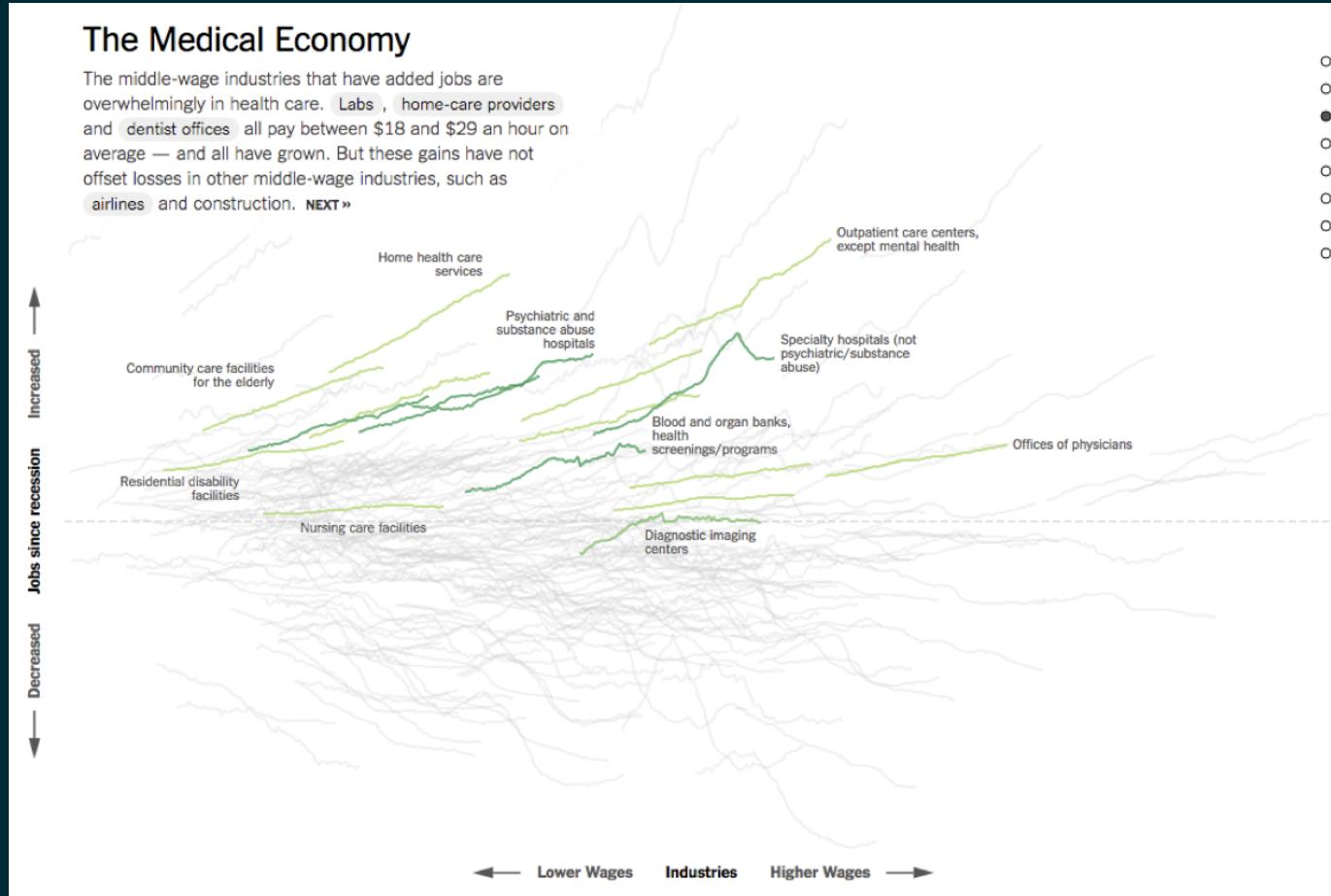


All of the data doesn't tell a story



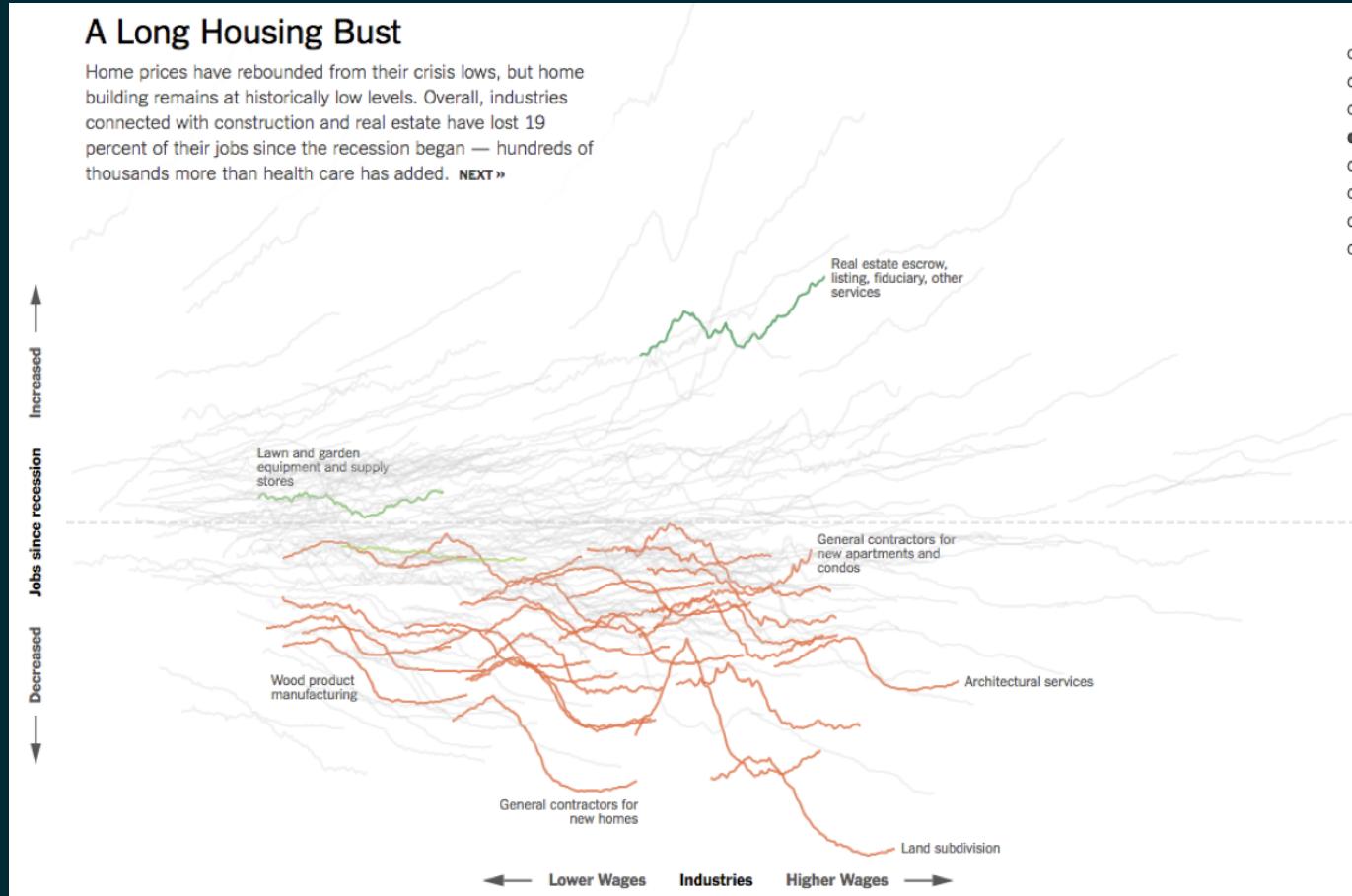
<http://www.nytimes.com/interactive/2014/06/05/upshot/how-the-recession-reshaped-the-economy-in-255-charts.html>

All of the data doesn't tell a story



<http://www.nytimes.com/interactive/2014/06/05/upshot/how-the-recession-reshaped-the-economy-in-255-charts.html>

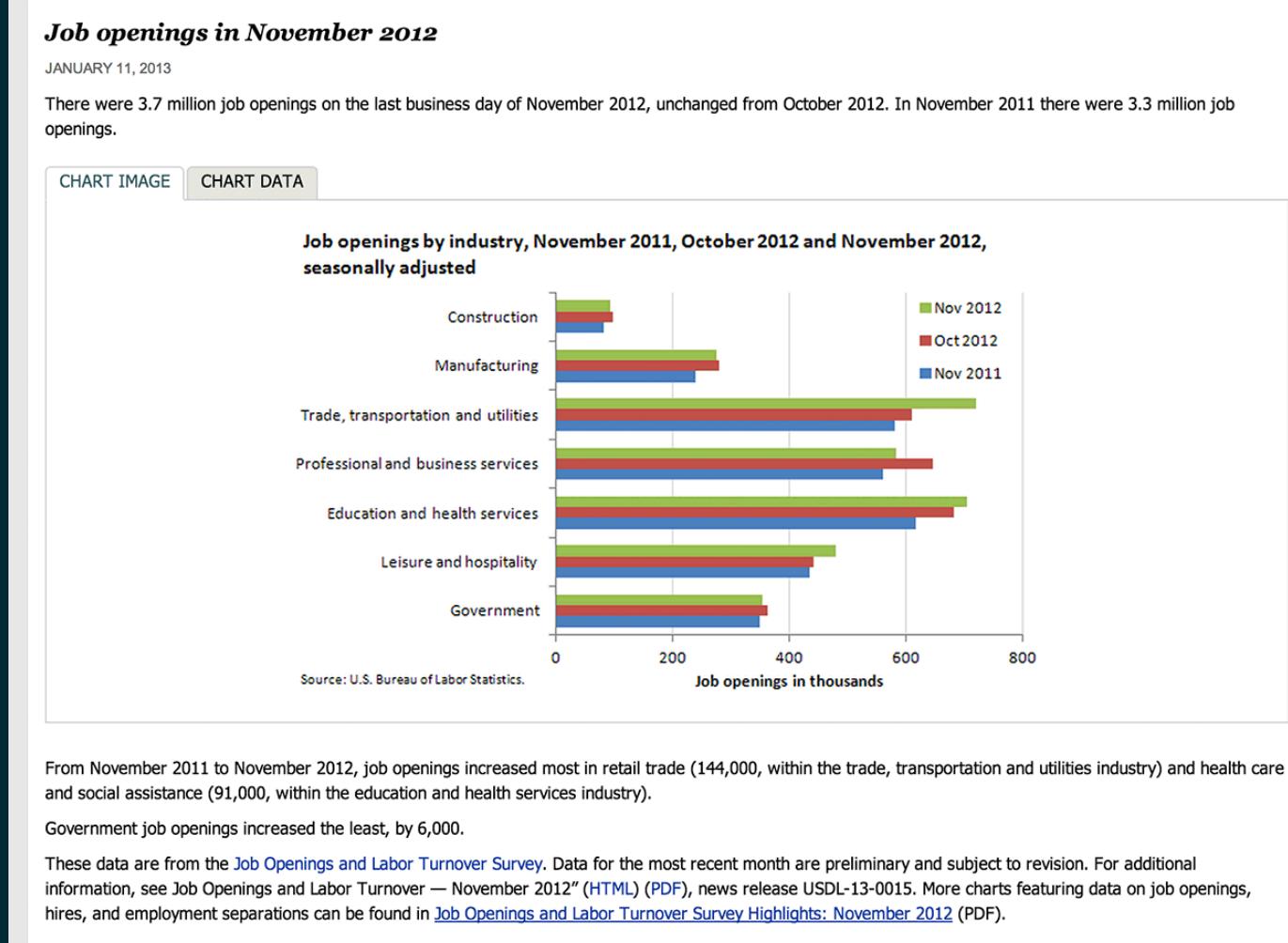
All of the data doesn't tell a story



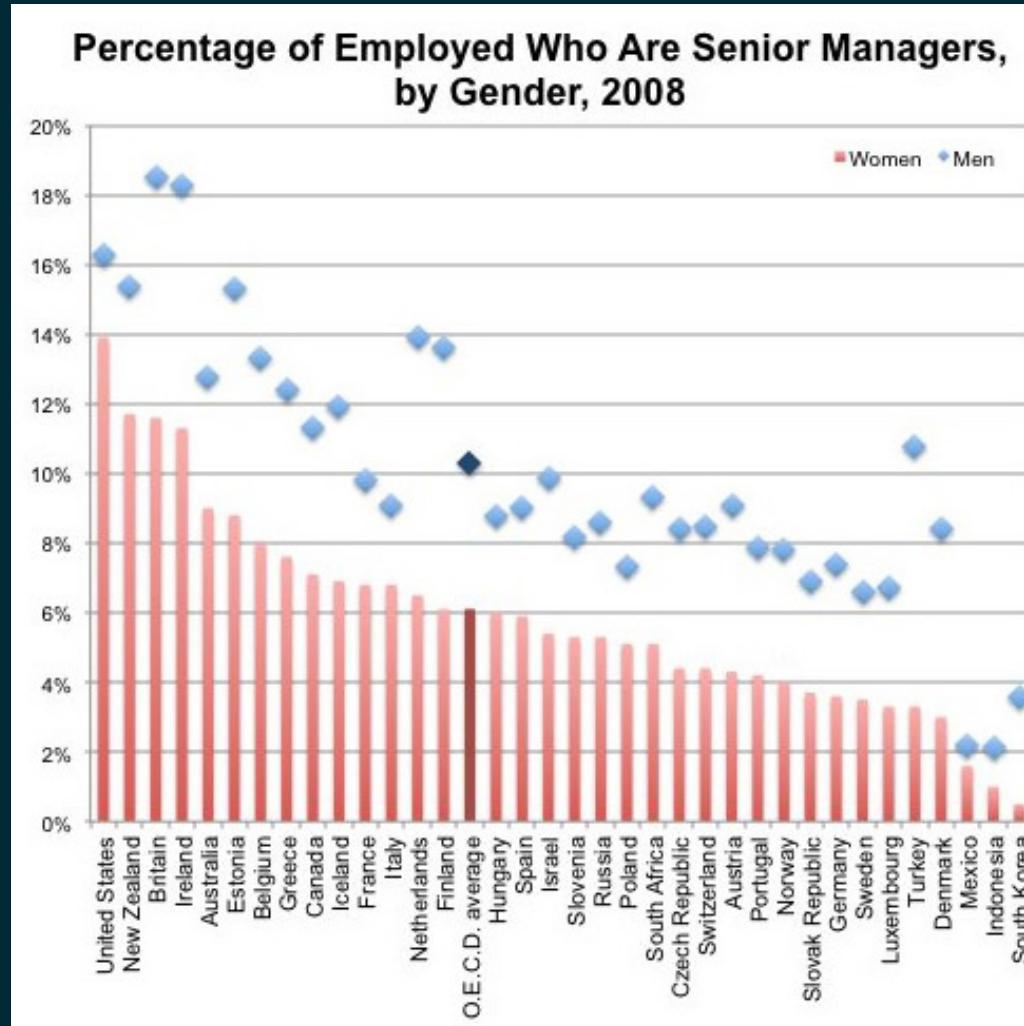
<http://www.nytimes.com/interactive/2014/06/05/upshot/how-the-recession-reshaped-the-economy-in-255-charts.html>

Chart Remakes / Makeovers

The Why Axis - BLS



The Why Axis - Gender Gap



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