

ggplot2 ecosystem & designing visualizations

Lecture 11

Dr. Colin Rundel

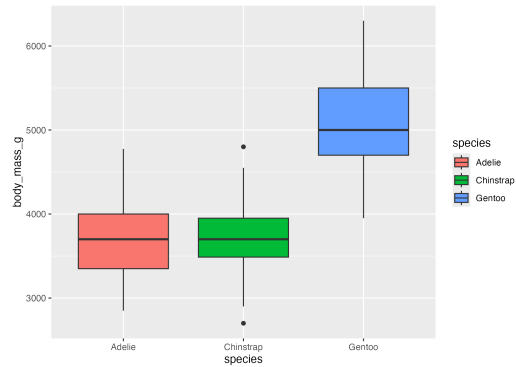
The wider ggplot2 ecosystem

ggthemes

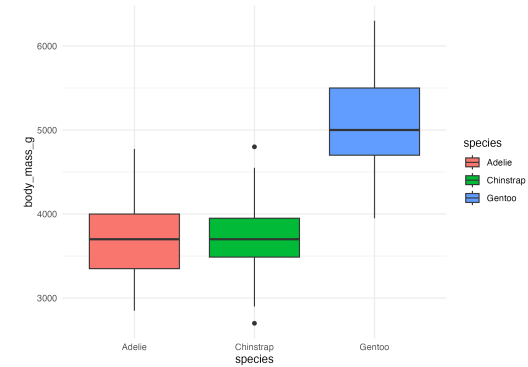
ggplot2 themes

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

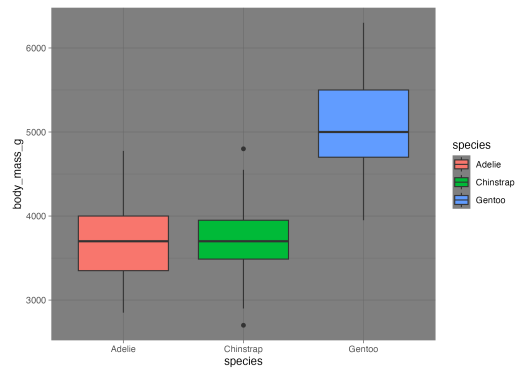
```
1 g
```



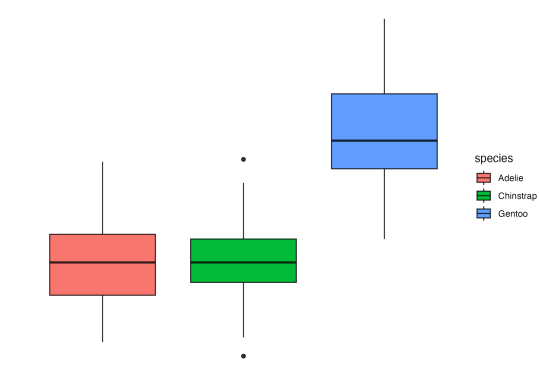
```
1 g + theme_minimal()
```



```
1 g + theme_dark()
```

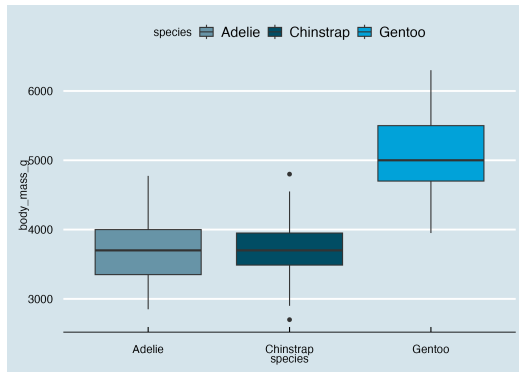


```
1 g + theme_void()
```

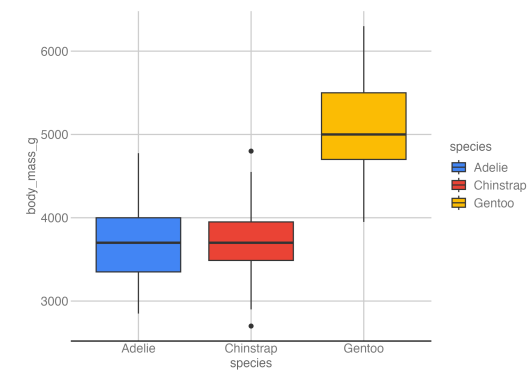


ggthemes

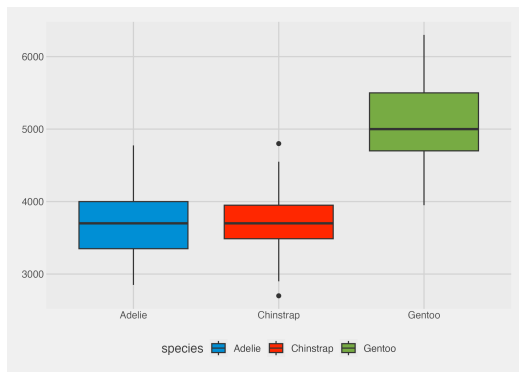
```
1 g + ggthemes::theme_economist() +
2   ggthemes::scale_fill_economist()
```



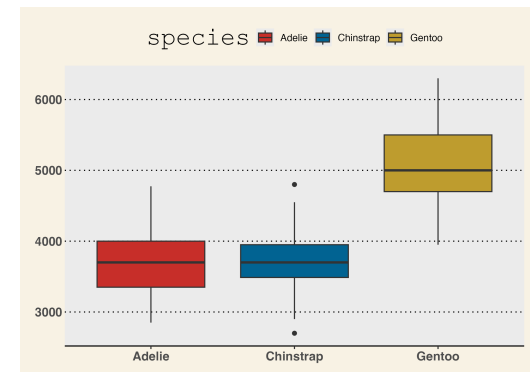
```
1 g + ggthemes::theme_gdocs() +
2   ggthemes::scale_fill_gdocs()
```



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

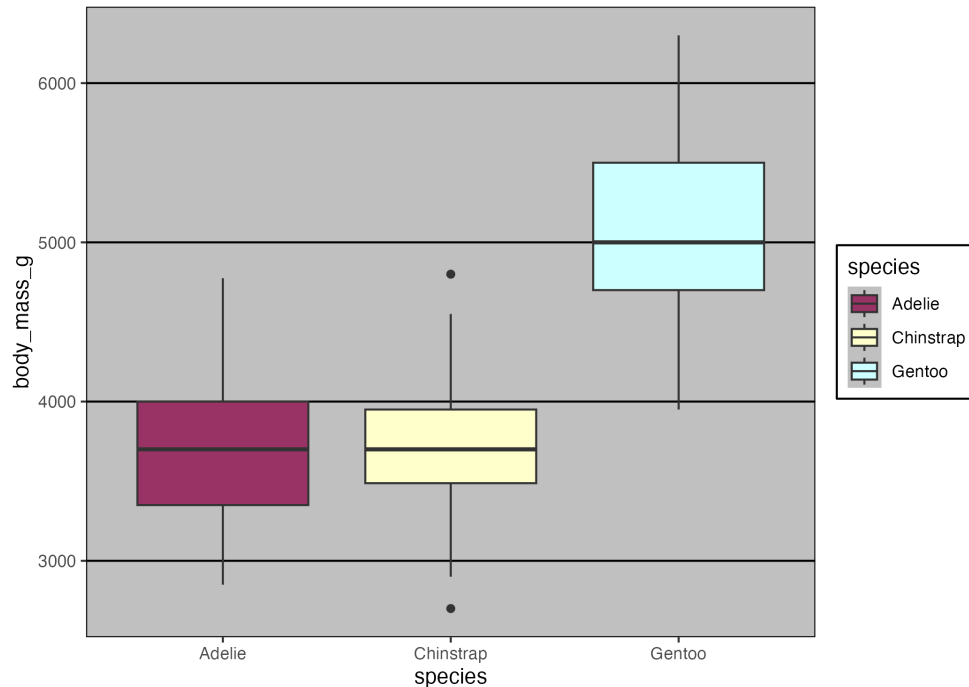


```
1 g + ggthemes::theme_wsj() +
2   ggthemes::scale_fill_wsj()
```

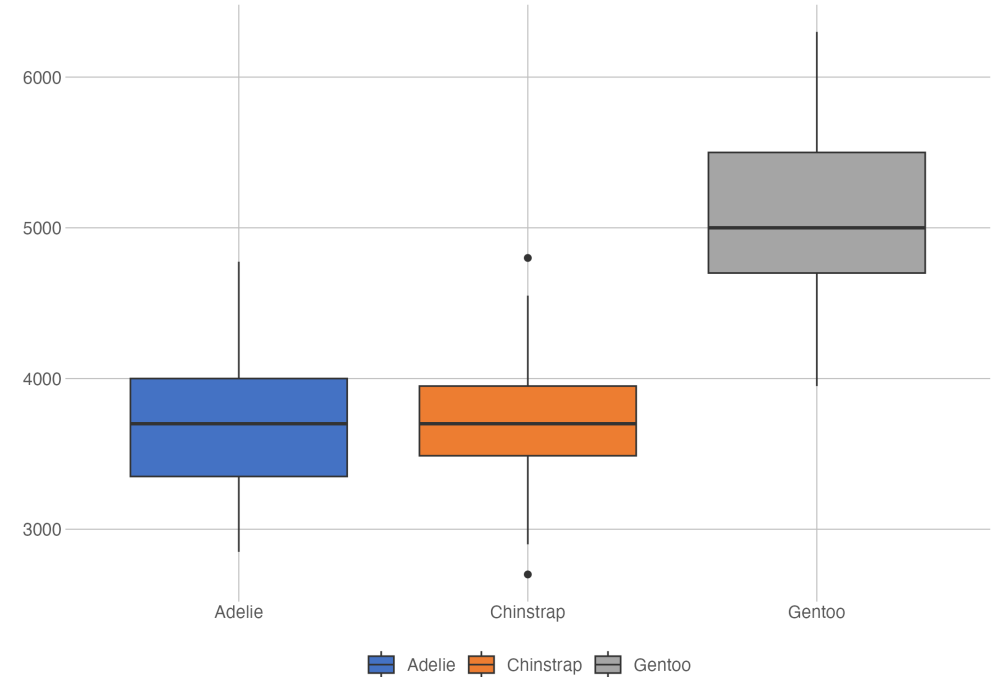


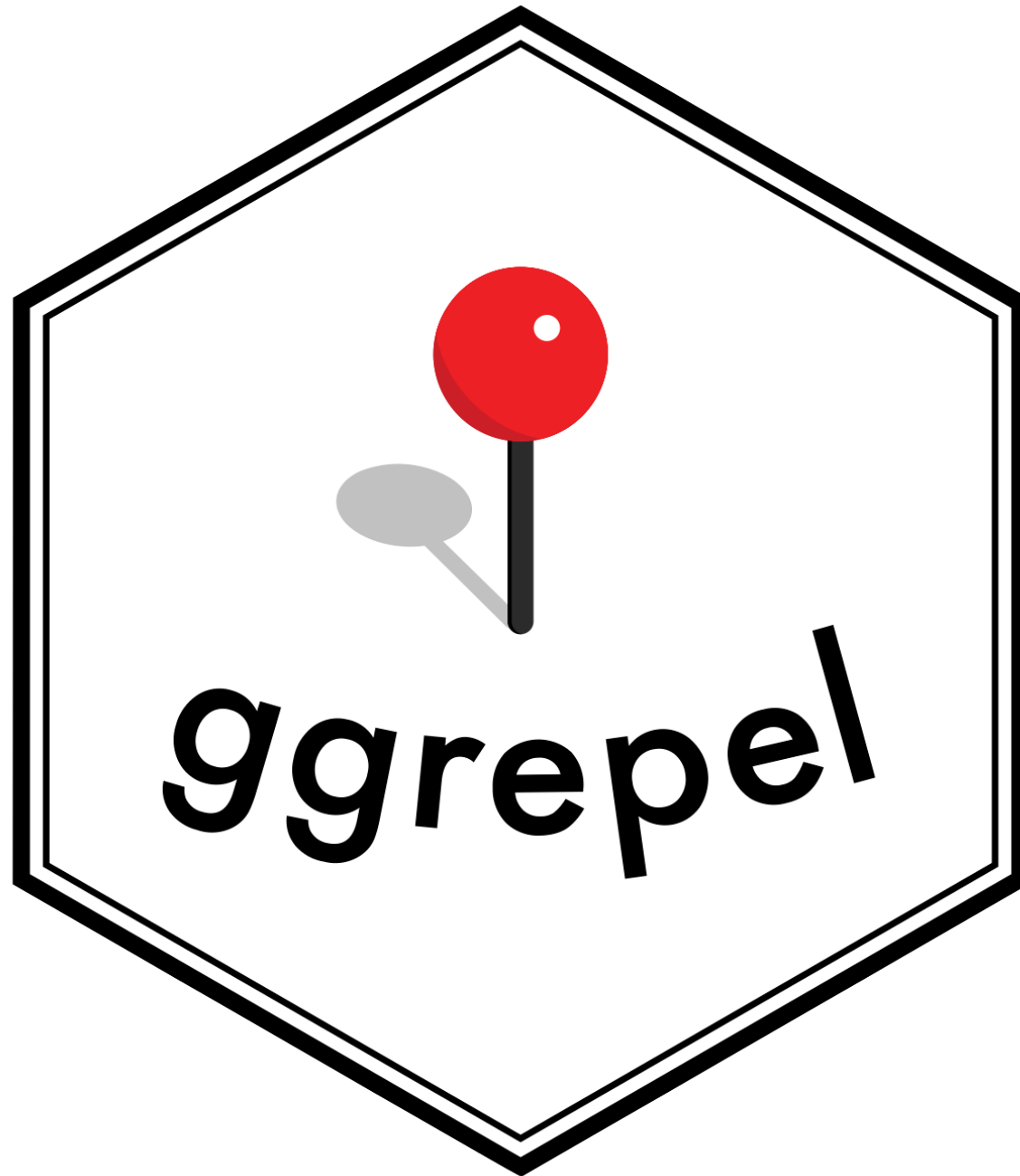
And for those who miss Excel

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



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





```
1 d = tibble(  
2   car = rownames(mtcars),  
3   weight = mtcars$wt,  
4   mpg = mtcars$mpg  
5 ) |>  
6   filter(weight > 2.75, weight < 3.45)
```

```

1 ggplot(d, aes(x=weight, y=mpg)) +
2   geom_point(color="red") +
3   geom_text(
4     aes(label = car)
5   )

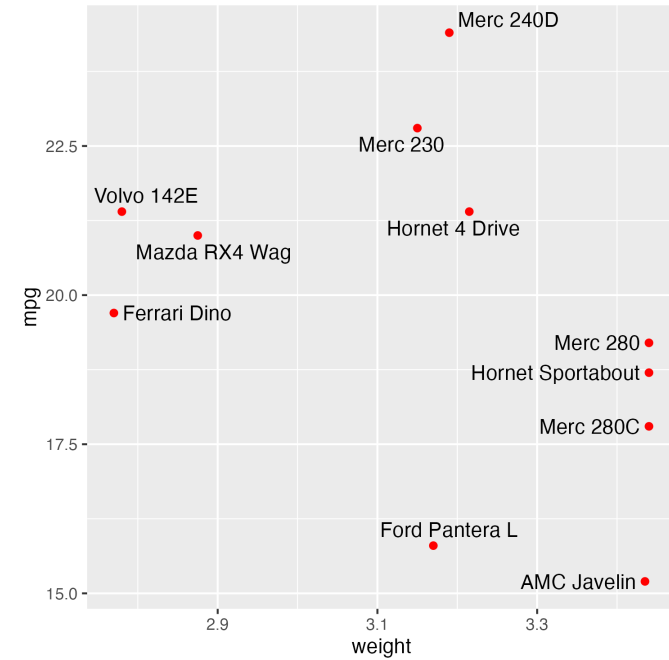
```



```

1 ggplot(d, aes(x=weight, y=mpg)) +
2   geom_point(color="red") +
3   ggrepel::geom_text_repel(
4     aes(label = car)
5   )

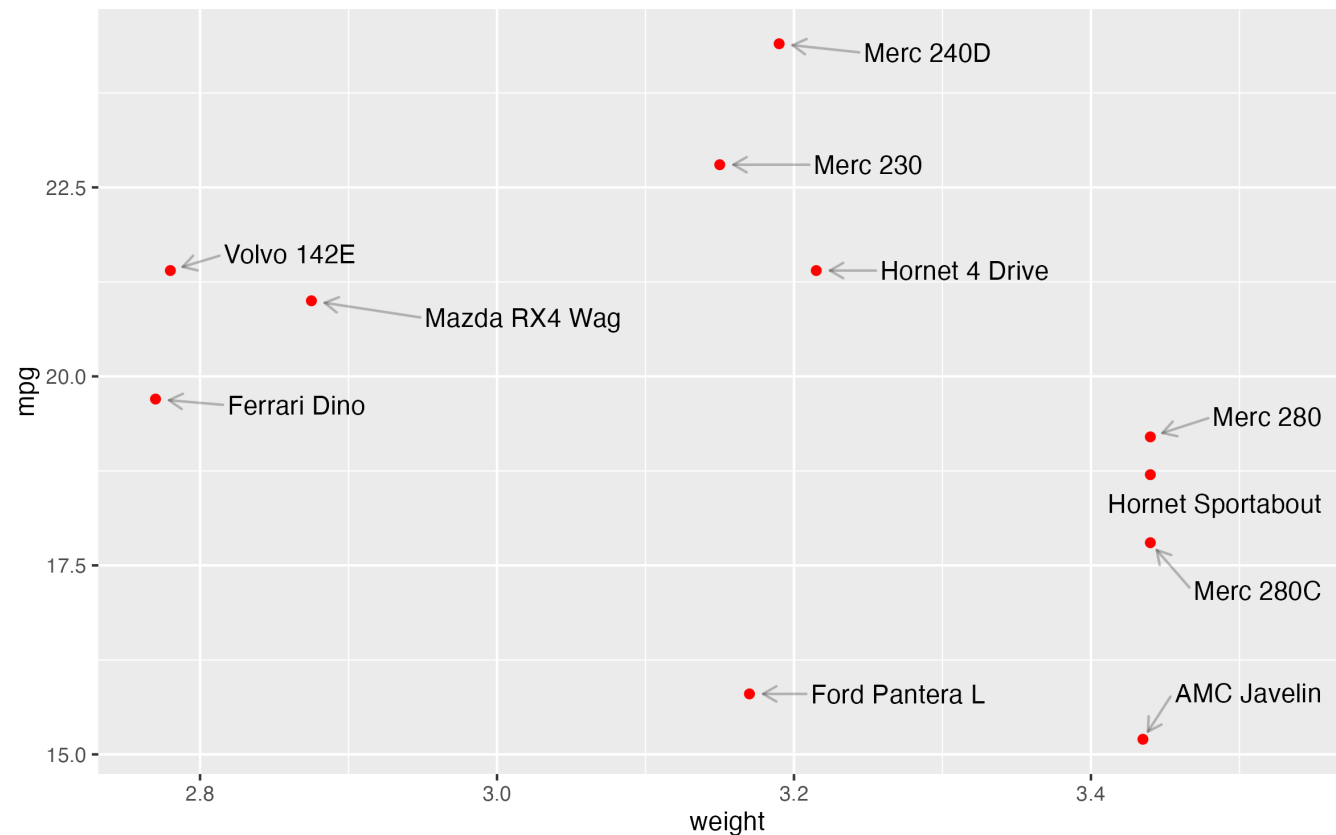
```



```

1 ggplot(d, aes(x=weight, y=mpg)) +
2   geom_point(color="red") +
3   ggrepel::geom_text_repel(
4     aes(label = car),
5     nudge_x = .1, box.padding = 1, point.padding = 0.6,
6     arrow = arrow(length = unit(0.02, "npc")), segment.alpha = 0.25
7   )

```





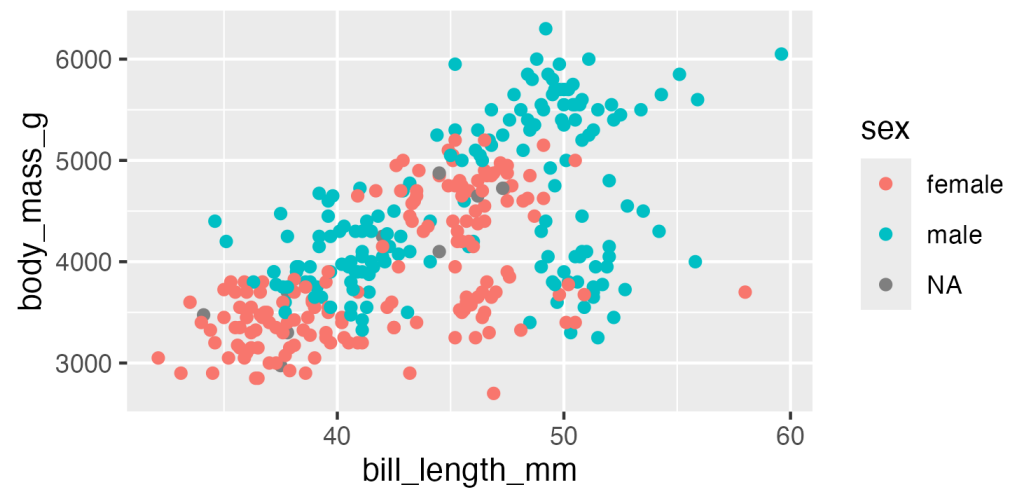
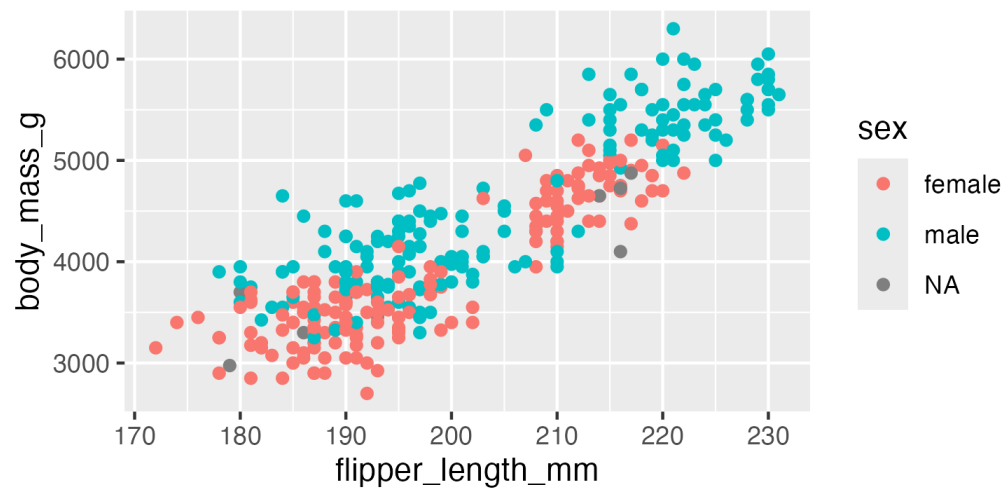
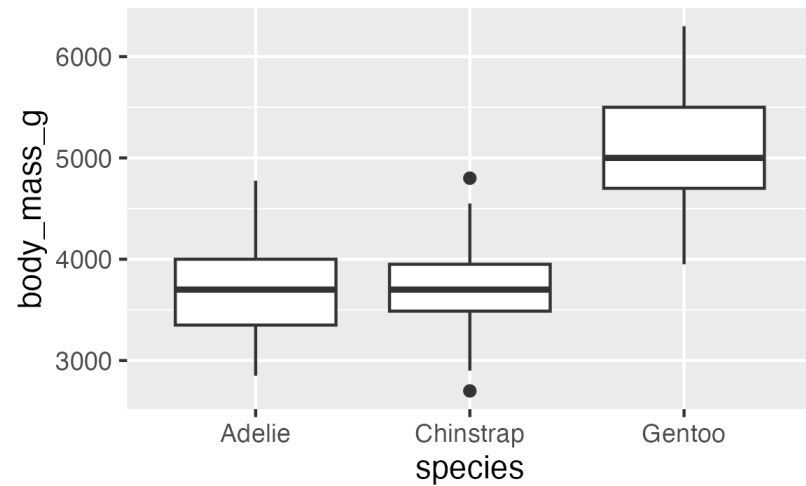
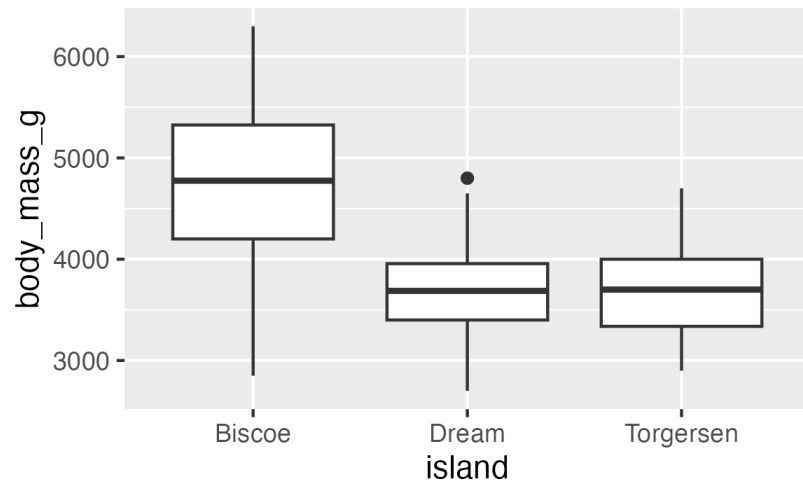
ggplot objects

```
1 library(patchwork)
2
3 p1 = ggplot(palmerpenguins::penguins) +
4   geom_boxplot(aes(x = island, y = body_mass_g))
5
6 p2 = ggplot(palmerpenguins::penguins) +
7   geom_boxplot(aes(x = species, y = body_mass_g))
8
9 p3 = ggplot(palmerpenguins::penguins) +
10  geom_point(aes(x = flipper_length_mm, y = body_mass_g, color = sex))
11
12 p4 = ggplot(palmerpenguins::penguins) +
13  geom_point(aes(x = bill_length_mm, y = body_mass_g, color = sex))
```

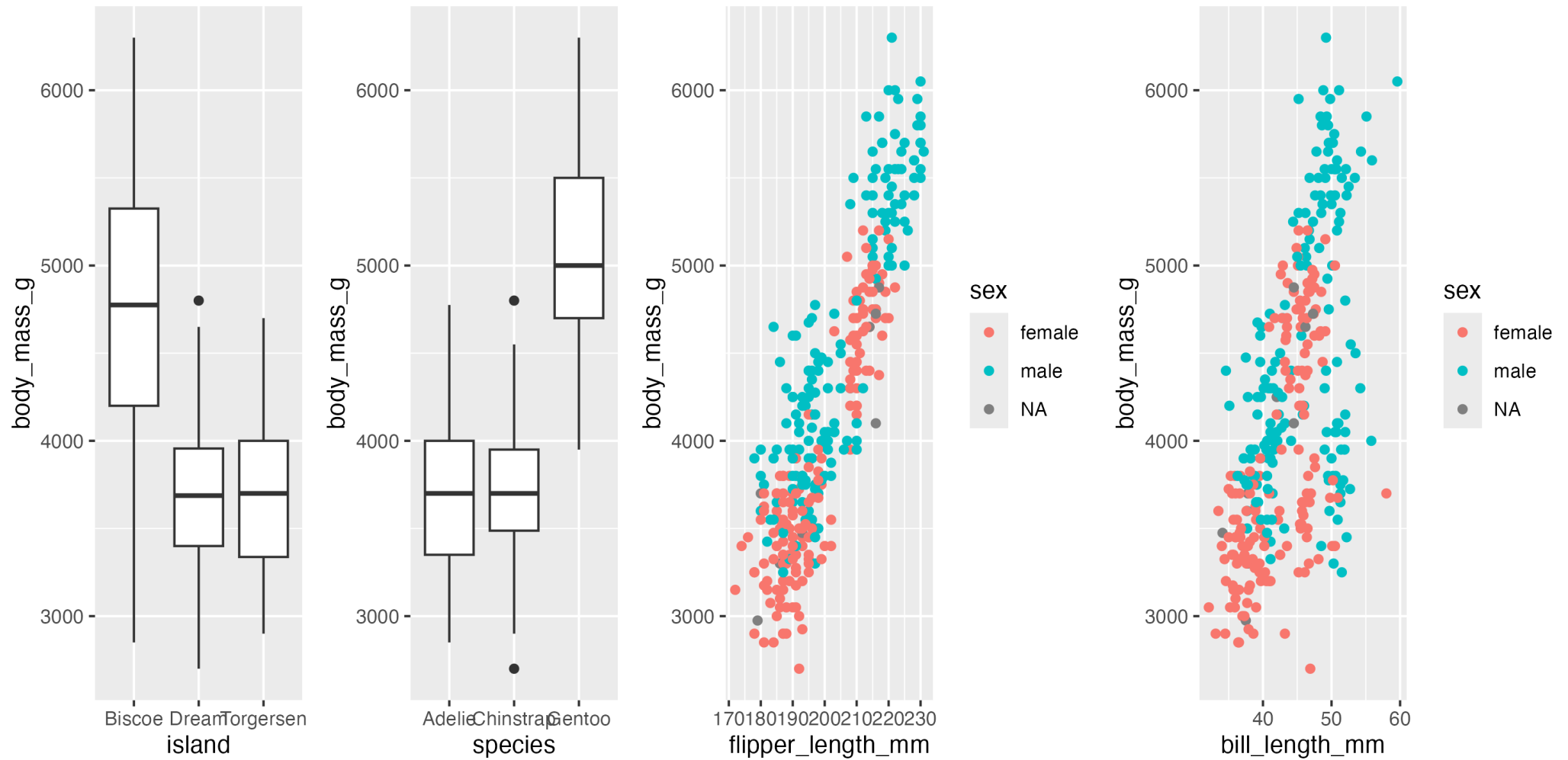
```
1 class(p1)
```

```
[1] "ggplot2::ggplot" "ggplot"          "ggplot2::gg"     "S7_object"
[5] "gg"
```

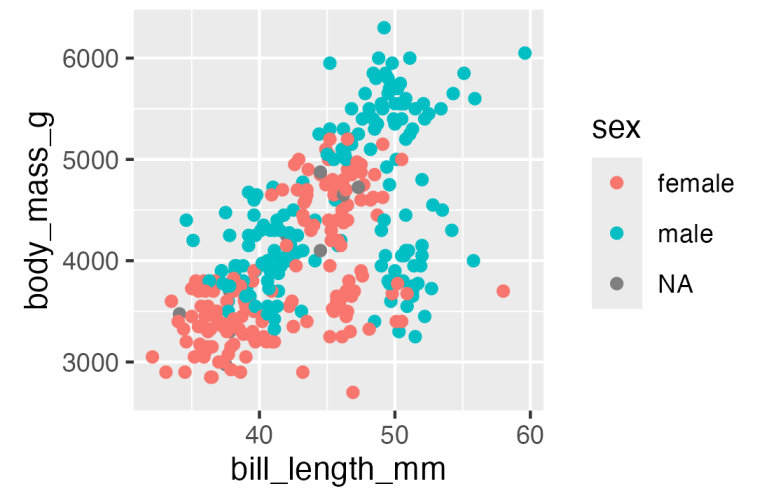
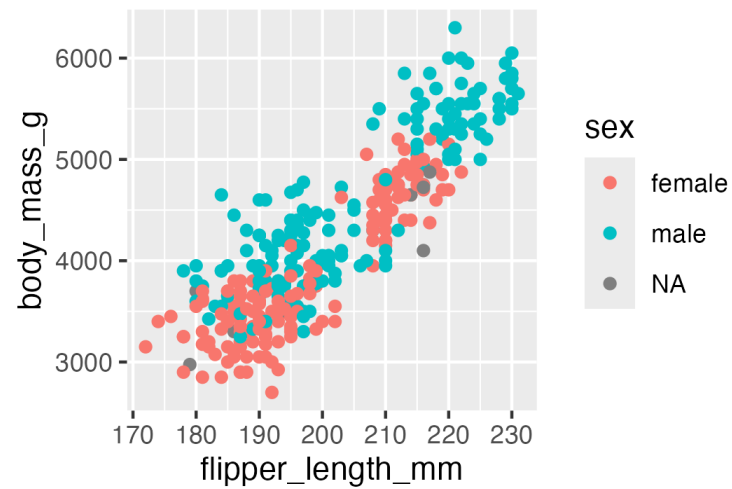
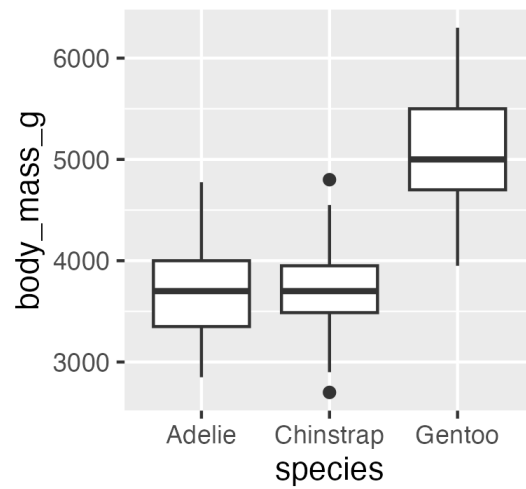
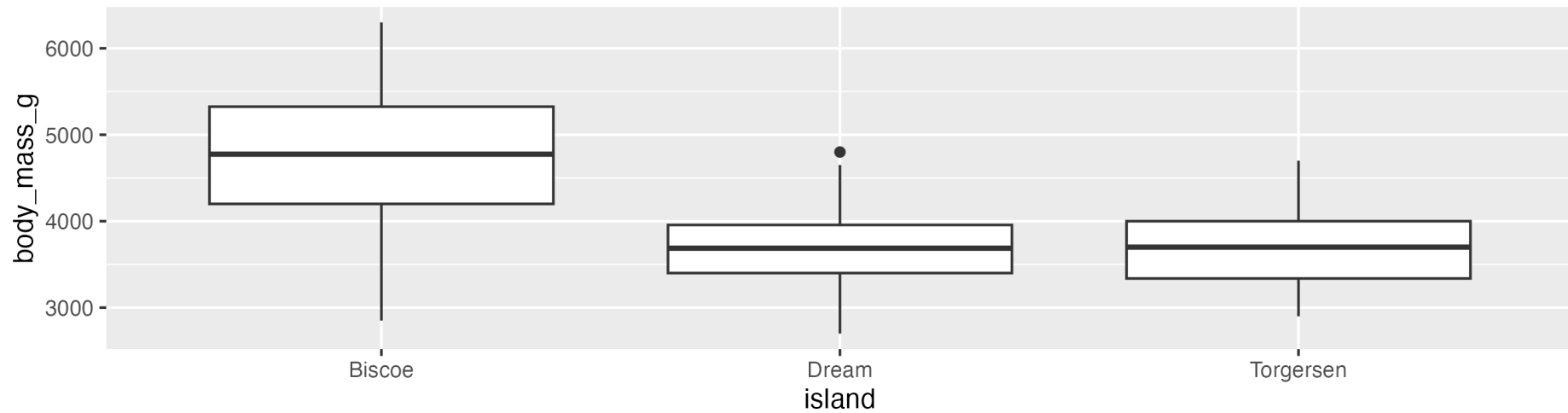
1 p1 + p2 + p3 + p4



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

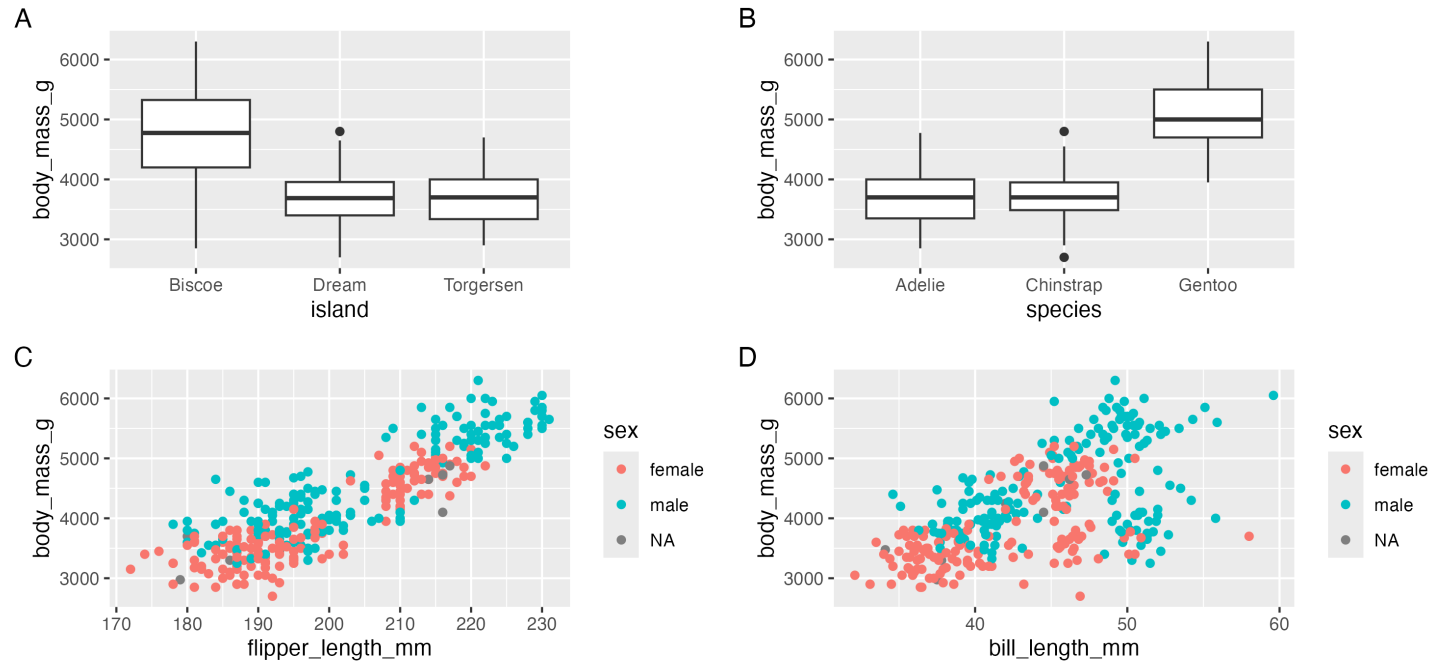


1 p1 / (p2 + p3 + p4)



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

Palmer Penguins



```

1 p1 + {
2   p2 + {
3     p3 + p4 + plot_layout(ncol = 1) + plot_layout(tag_level = 'new')
4   }
5 } +
6 plot_layout(ncol = 1) +
7 plot_annotation(tag_levels = c("1","a"), tag_prefix = "Fig ")

```

Fig 1

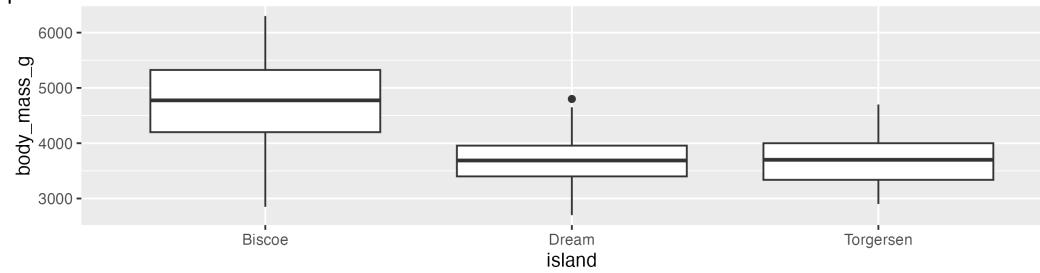


Fig 2

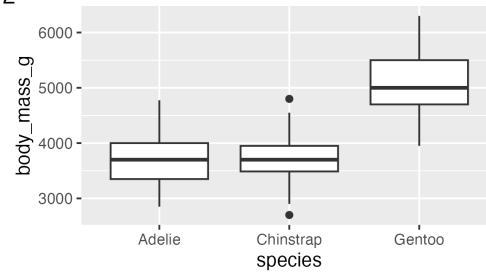


Fig 3a

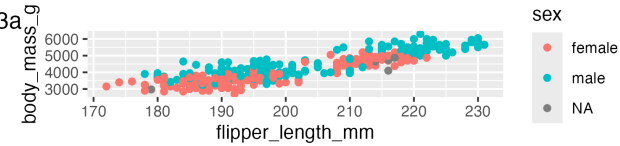
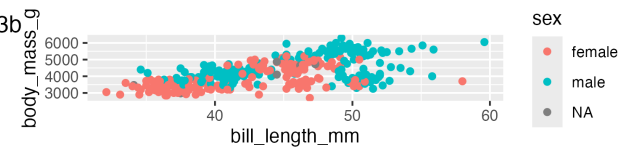
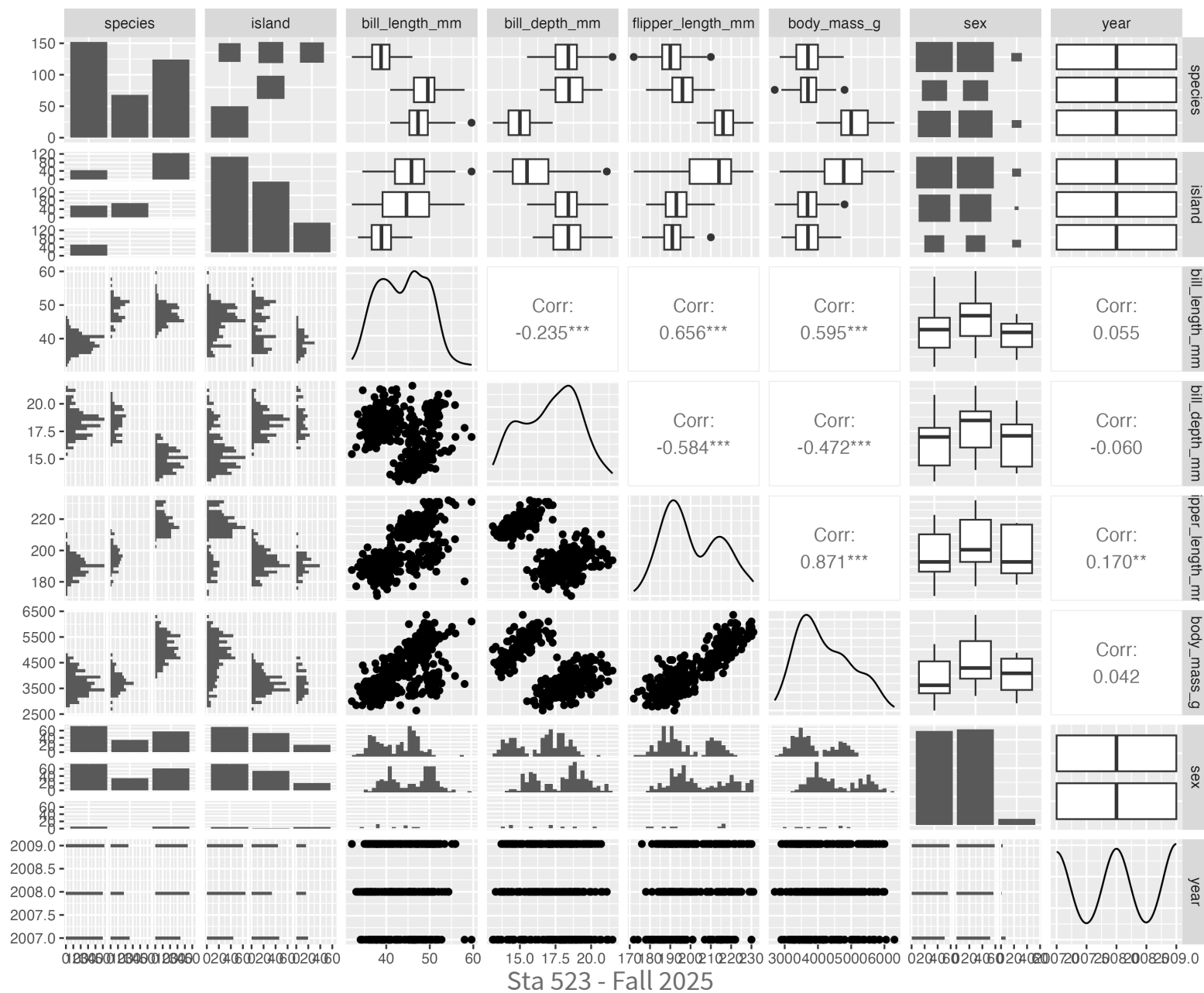


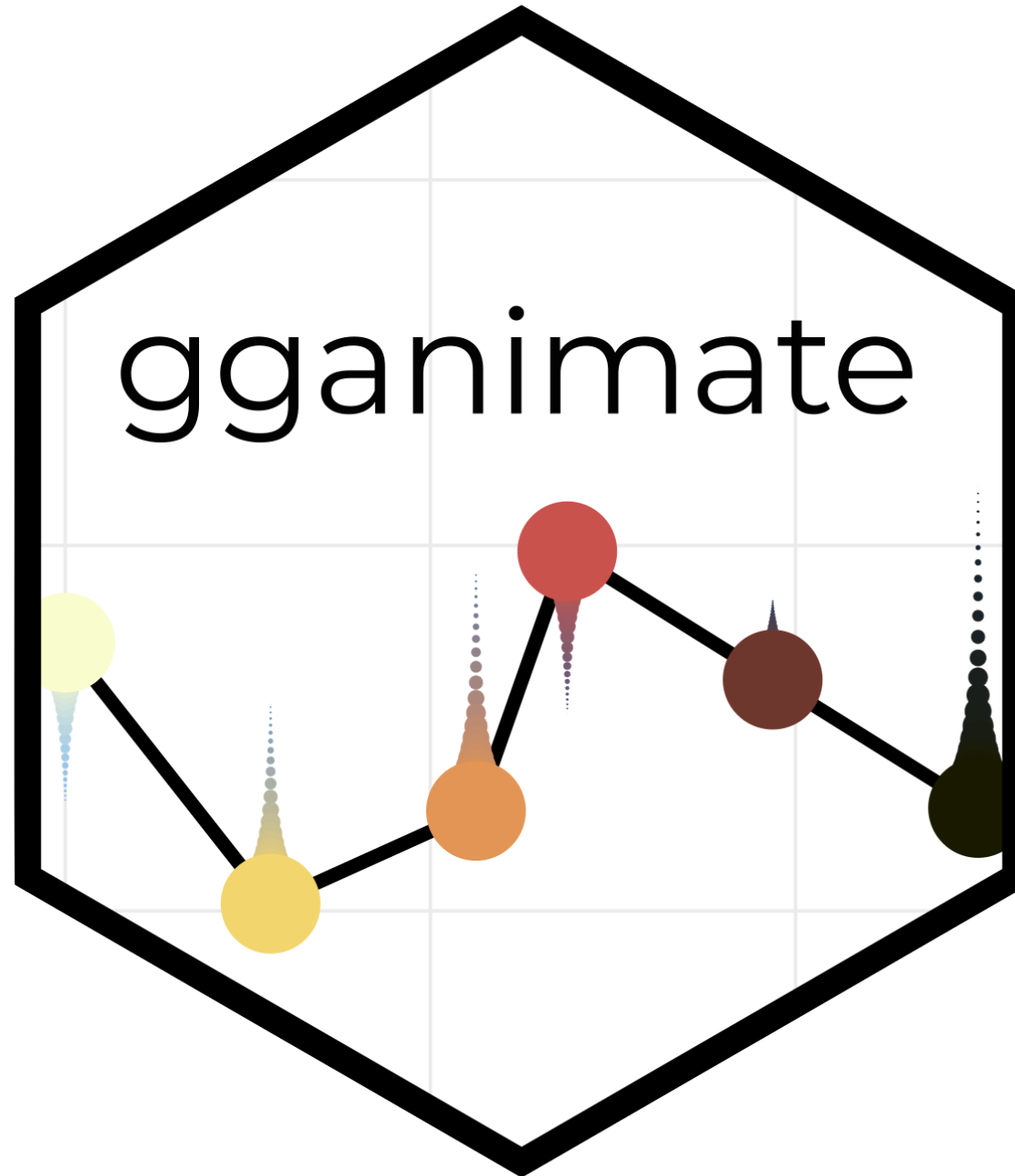
Fig 3b



GGally

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

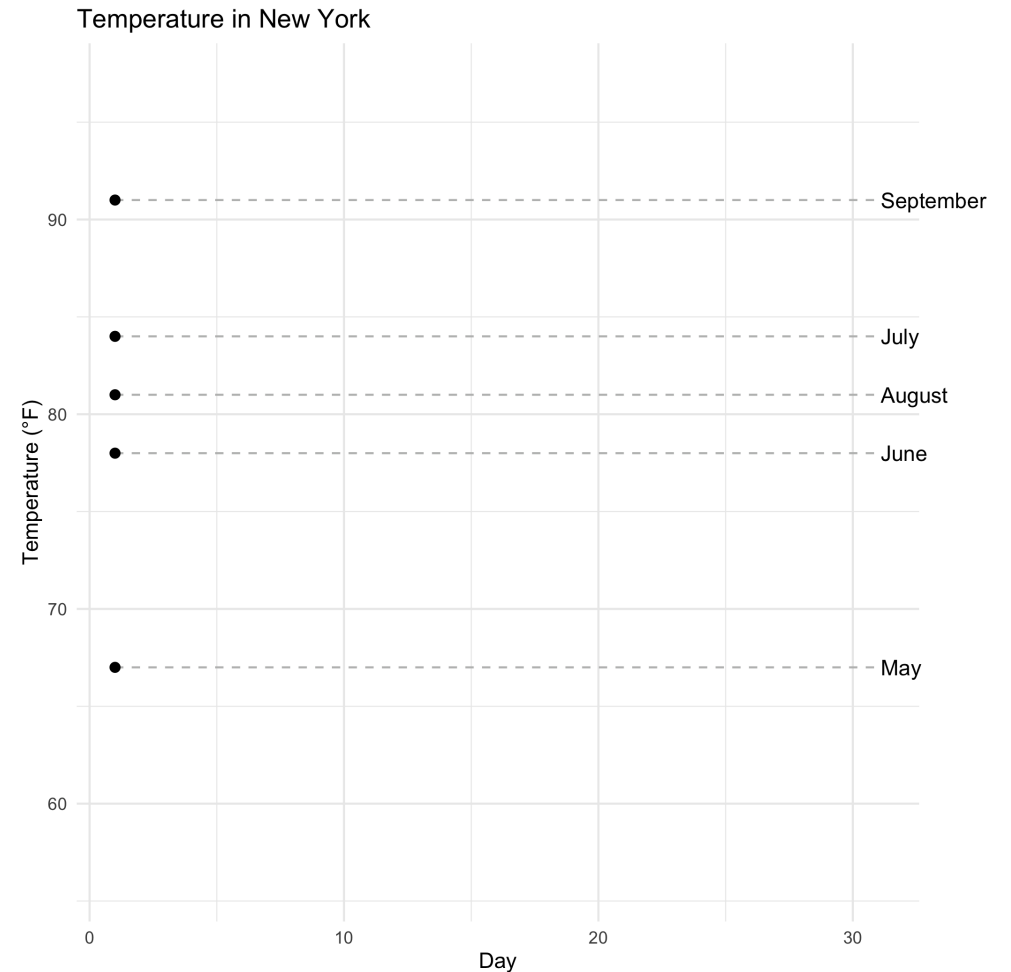




```

1 airq = airquality
2 airq$Month = month.name[airq$Month]
3
4 ggplot(
5   airq,
6   aes(Day, Temp, group = Month)
7 ) +
8   geom_line() +
9   geom_segment(
10    aes(xend = 31, yend = Temp),
11    linetype = 2,
12    colour = 'grey'
13  ) +
14  geom_point(size = 2) +
15  geom_text(
16    aes(x = 31.1, label = Month),
17    hjust = 0
18  ) +
19  gganimate::transition_reveal(Day) +
20  coord_cartesian(clip = 'off') +

```



Some other notable packages

- `marquee` - add rendered markdown to your plots
- `thematic` & `brand.yml` - automatic theming of plots to match your app / site
- `ggridges` - creates ridgeline plots (stacked density plots)
- `ggdist` - visualizations and utilities for distributions and uncertainty (think bayesian model output)
- `legendary` - adds addition guides (legends) to ggplot2

More extensions

exts.ggplot2.tidyverse.org/gallery/

Why do we visualize?

Asncombe's Quartet

```
1 datasets::anscombe |> as_tibble()
```

```
# A tibble: 11 × 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

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

A tibble: 44 × 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

i 34 more rows

```
1 tidy_anscombe |>
2   group_by(group) |>
3   summarize(
4     mean_x = mean(x), mean_y = mean(y),
5     sd_x = sd(x), sd_y = sd(y),
6     cor = cor(x,y), .groups = "drop"
7   )
```

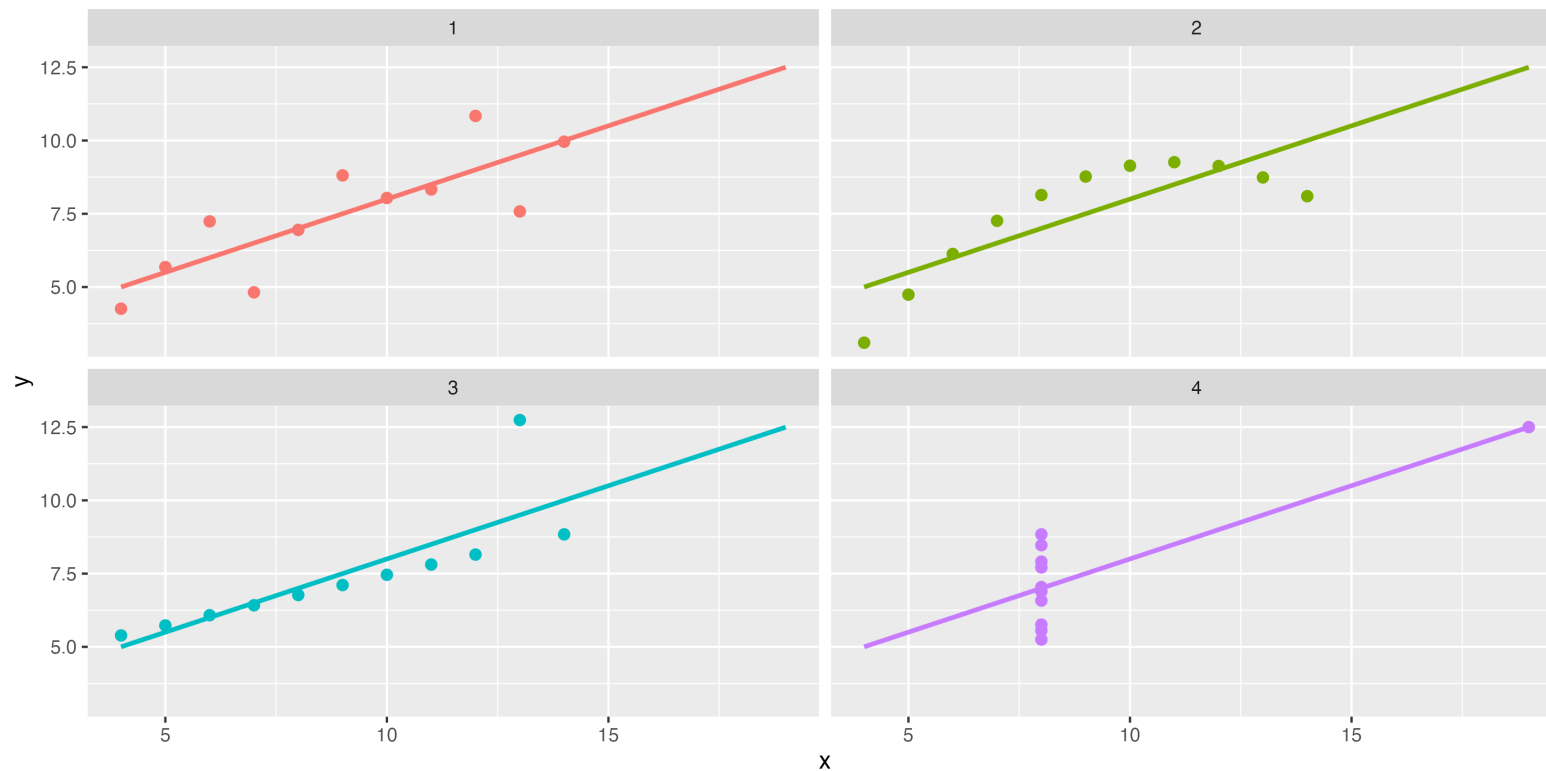
A tibble: 4 × 6

	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

```

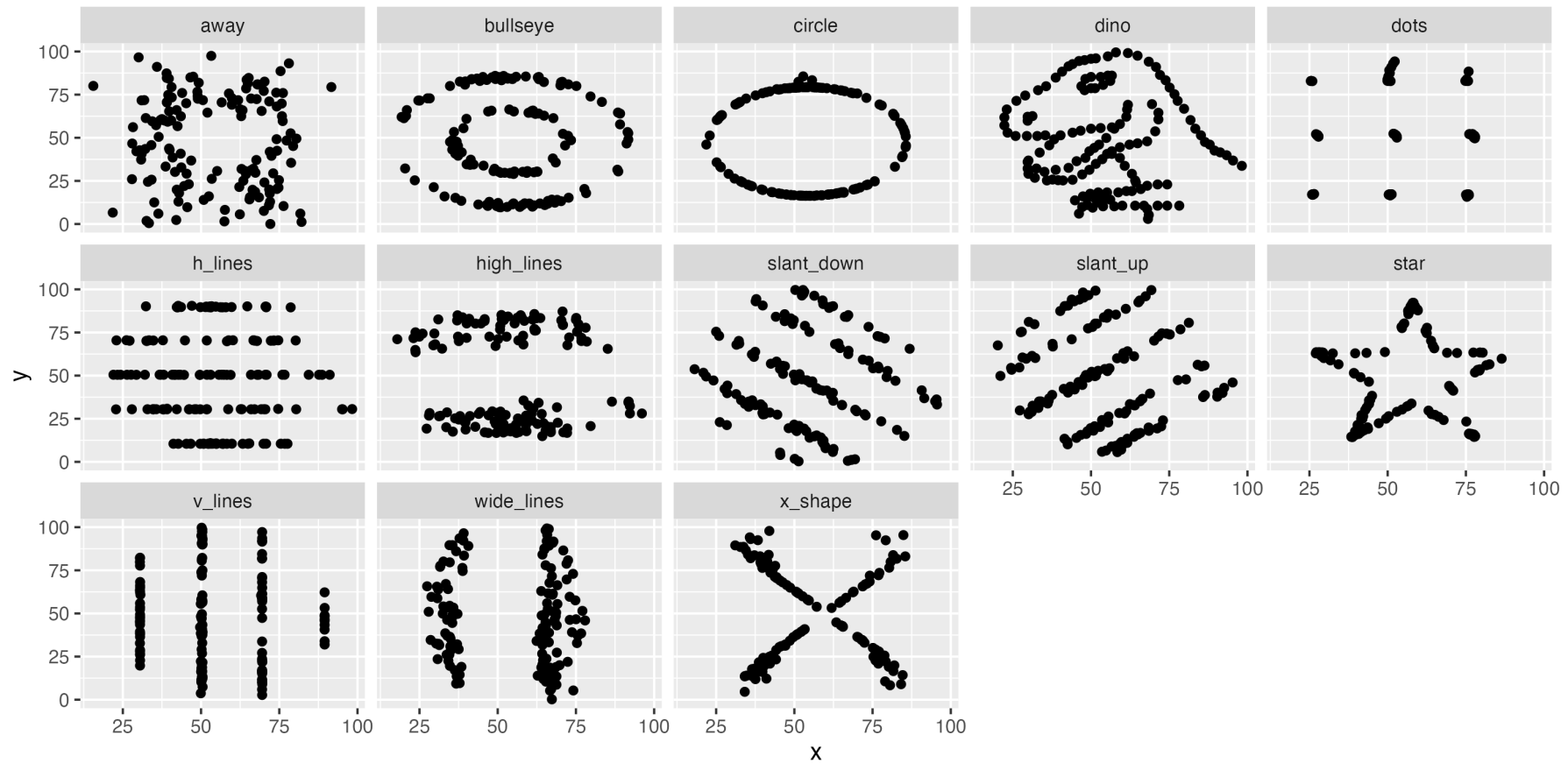
1 ggplot(tidy_anscombe, aes(x = x, y = y, color = as.factor(group)))
2   geom_point(size=2) +
3   facet_wrap(~group) +
4   geom_smooth(method="lm", se=FALSE, fullrange=TRUE, formula = y~x)
5   guides(color="none")

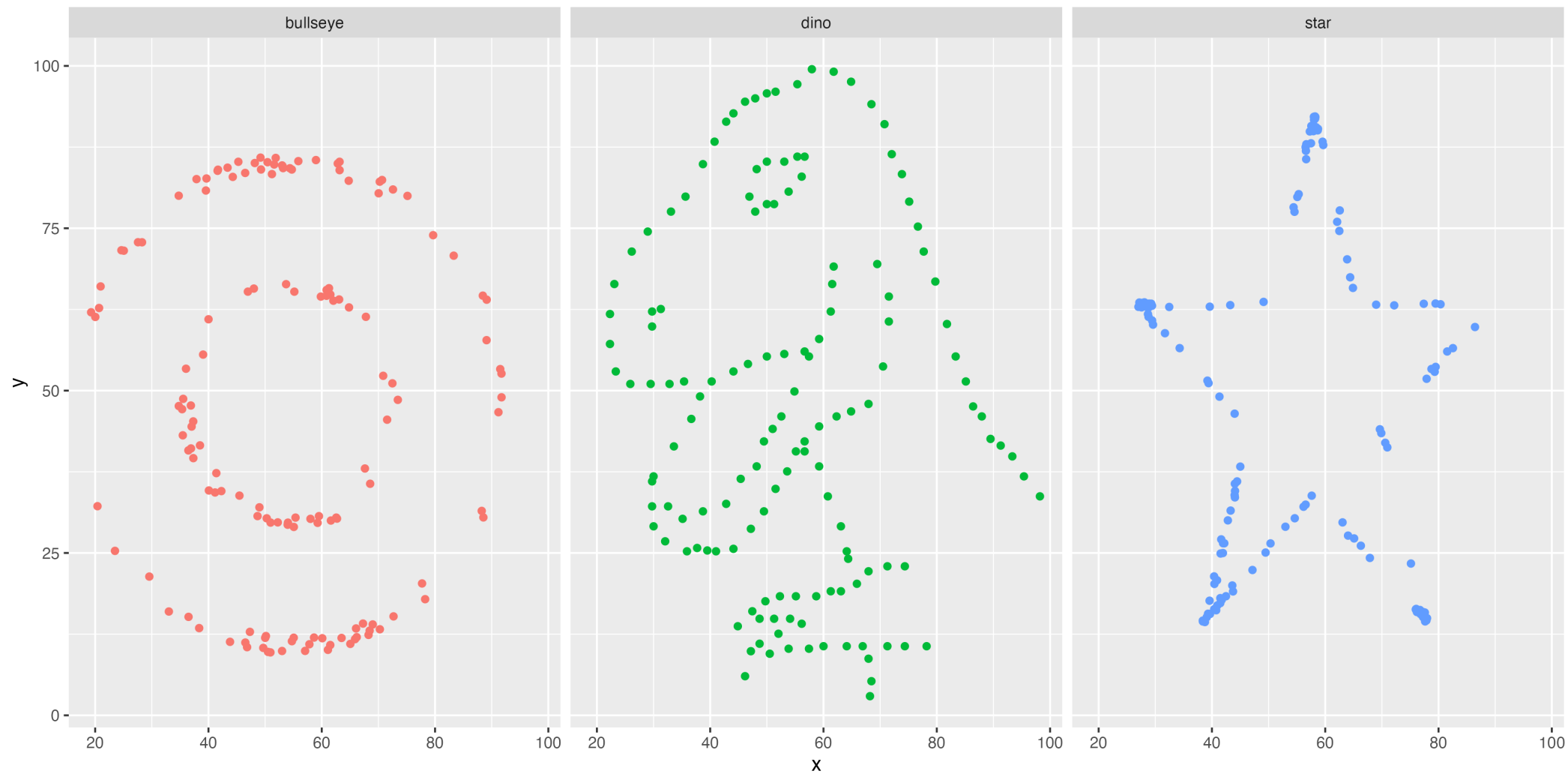
```



DatasauRus

```
1 ggplot(datasauRus::datasaurus_dozen, aes(x = x, y = y)) +  
2   geom_point() +  
3   facet_wrap(~dataset, ncol=5)
```





```
1 datasauRus::datasaurus_
```

```
# A tibble: 1,846 × 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
# i 1,836 more rows
```

```
1 datasauRus::datasaurus_dozen |>
2   group_by(dataset) |>
3   summarize(mean_x = mean(x), mean_y = mean(y),
4             sd_x = sd(x), sd_y = sd(y),
5             cor = cor(x,y), .groups = "drop")
```

```
# A tibble: 13 × 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
13 x_shape   54.3  47.8  16.8  26.9 -0.0656
```

Simpson's Paradox

```
1 lm(y~x, data=simpsons) |>  
2 summary()
```

Call:

```
lm(formula = y ~ x, data = simpsons)
```

Residuals:

Min	1Q	Median	3Q	Max
-38.988	-10.208	-0.707	9.874	42.642

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-41.20220	3.51007	-11.74	<2e-16 ***
x	1.81324	0.06993	25.93	<2e-16 ***

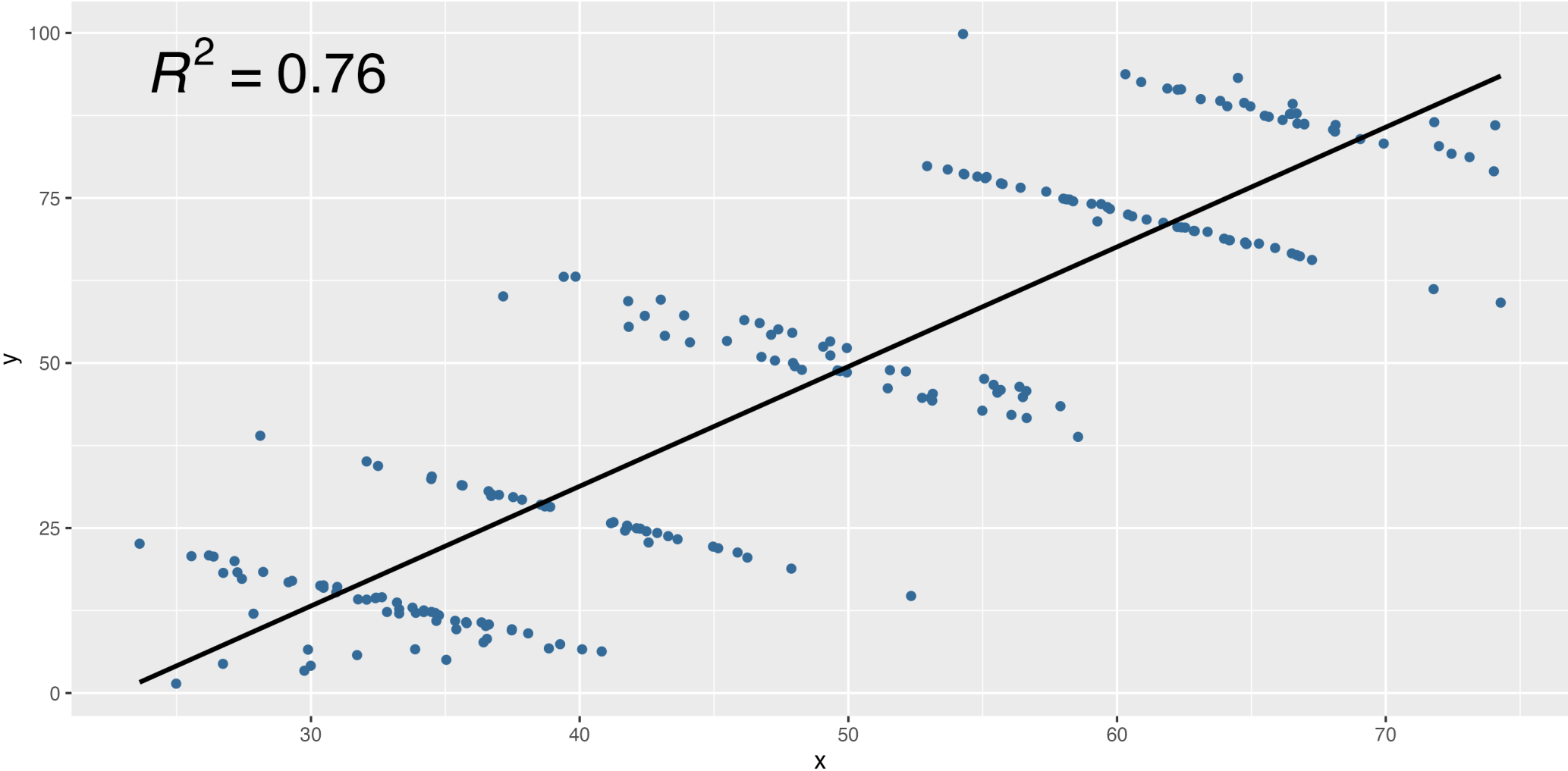
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.93 on 215 degrees of freedom

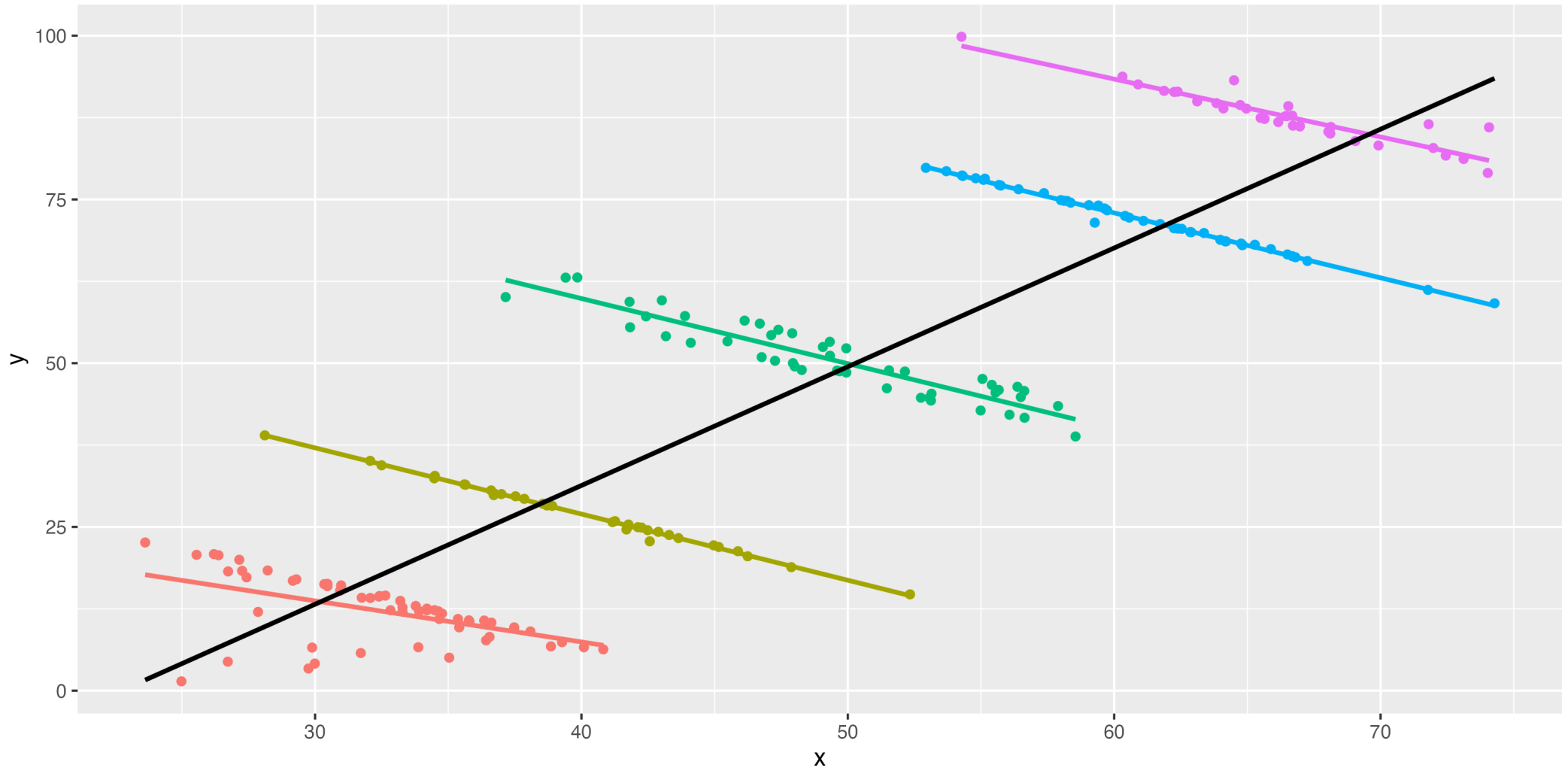
Multiple R-squared: 0.7577, Adjusted R-squared: 0.7566

F-statistic: 672.2 on 1 and 215 DF, p-value: < 2.2e-16

Simpson's Paradox Visually



Simpson's Paradox with groups



Revised model

```
1 lm(y~x*group-1, data=simpsons) |>
2   summary()
```

Call:

```
lm(formula = y ~ x * group - 1, data = simpsons)
```

Residuals:

Min	1Q	Median	3Q	Max
-15.4264	-0.6137	0.0811	1.0448	5.0613

Coefficients:

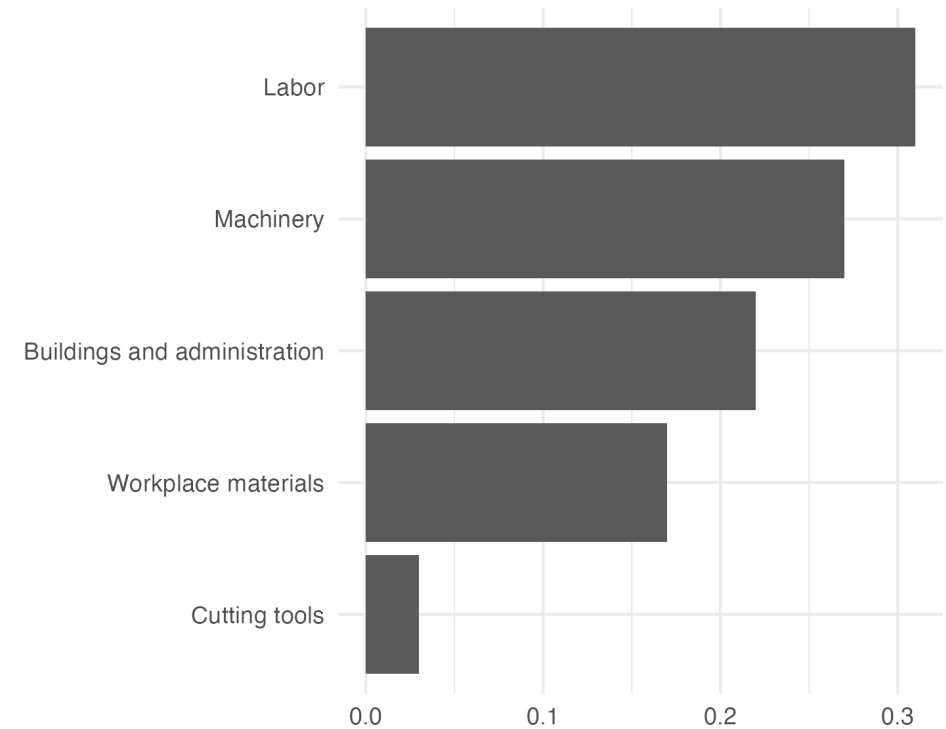
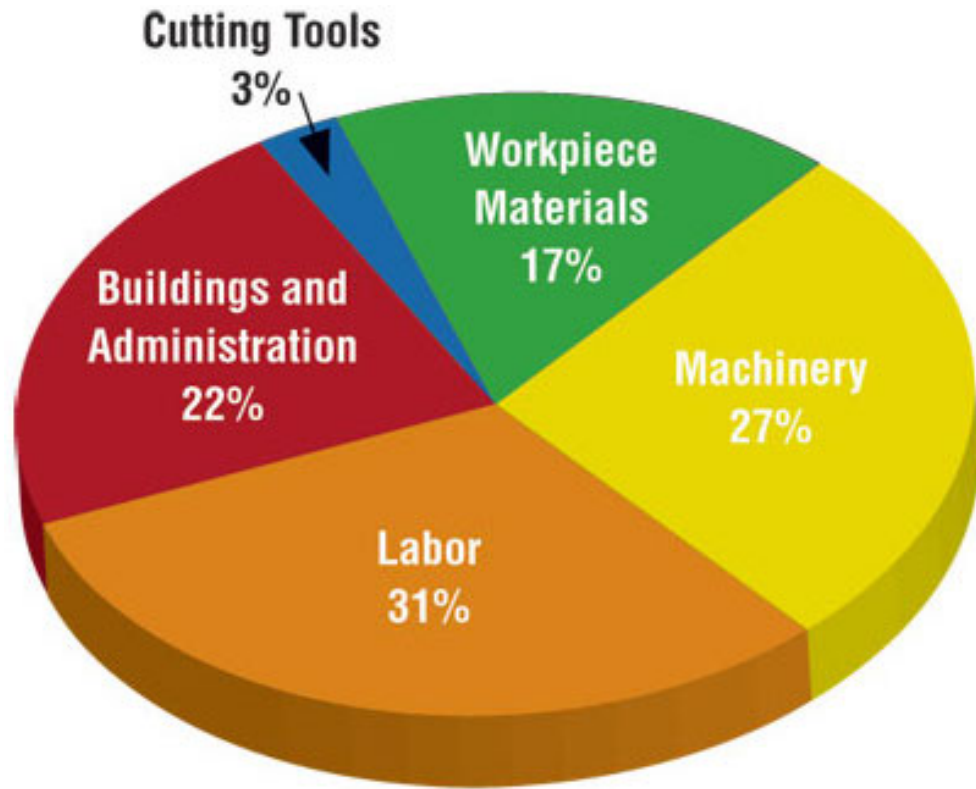
	Estimate	Std. Error	t value	Pr(> t)	
x	-0.62658	0.07987	-7.846	2.27e-13	***
group1	32.50512	2.61640	12.424	< 2e-16	***
group2	67.38858	3.47010	19.420	< 2e-16	***
group3	99.63330	3.34565	29.780	< 2e-16	***
group4	132.39316	4.76158	27.804	< 2e-16	***
group5	146.36456	6.78530	21.571	< 2e-16	***
x:group2	-0.38394	0.11747	-3.268	0.001267	**
x:group3	-0.36743	0.10440	-3.519	0.000532	***
x:group4	-0.36425	0.11146	-3.268	0.001268	**
x:group5	-0.25654	0.12950	-1.981	0.048917	*

Designing effective visualizations

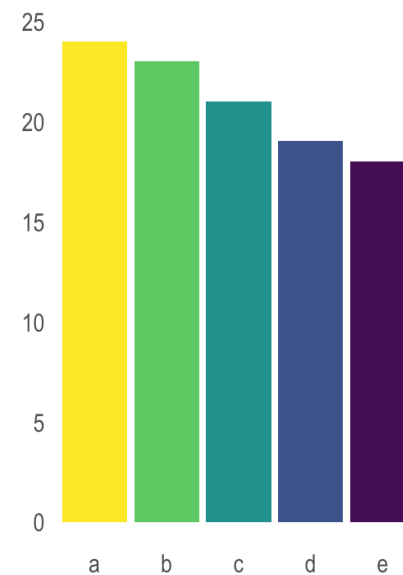
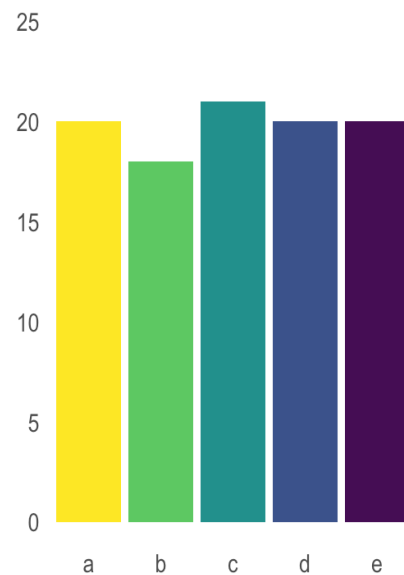
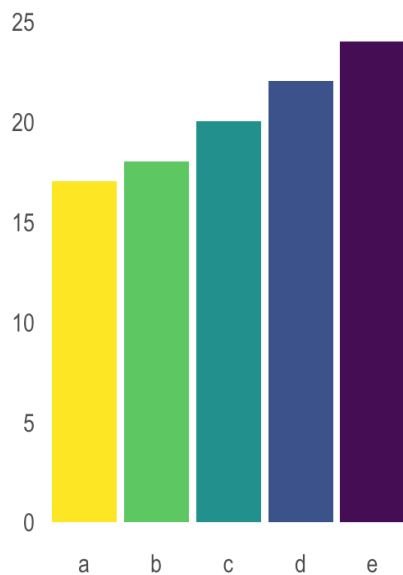
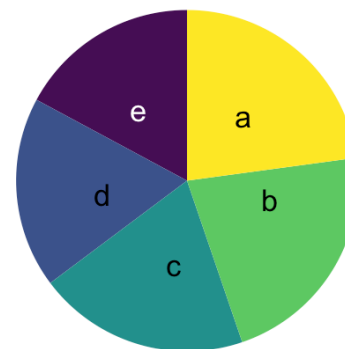
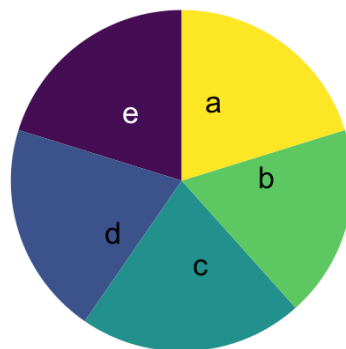
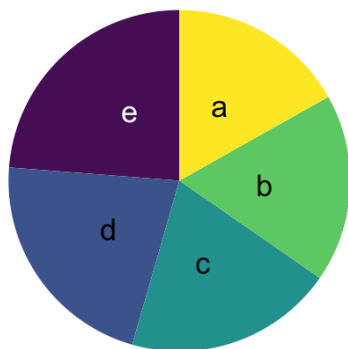
Gapminder

gapminder.org/dollar-street

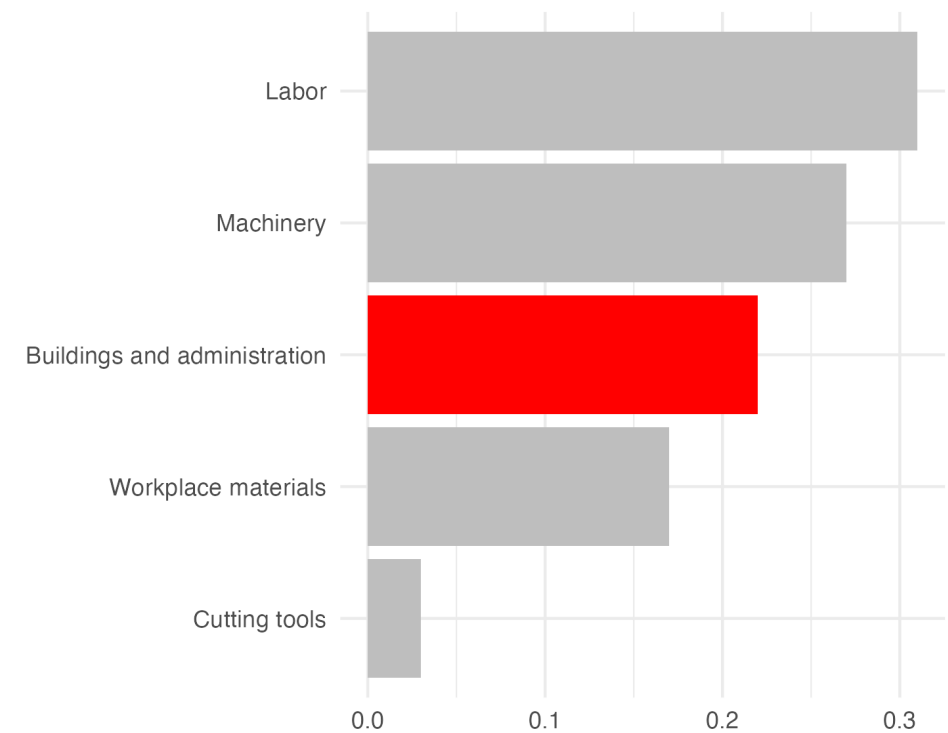
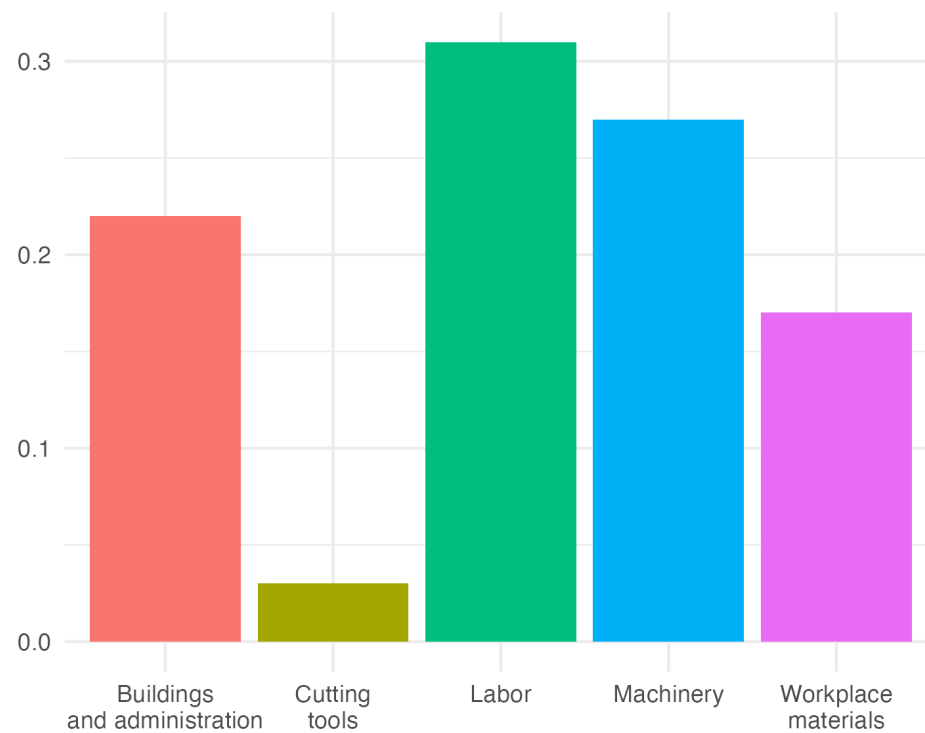
Keep it simple



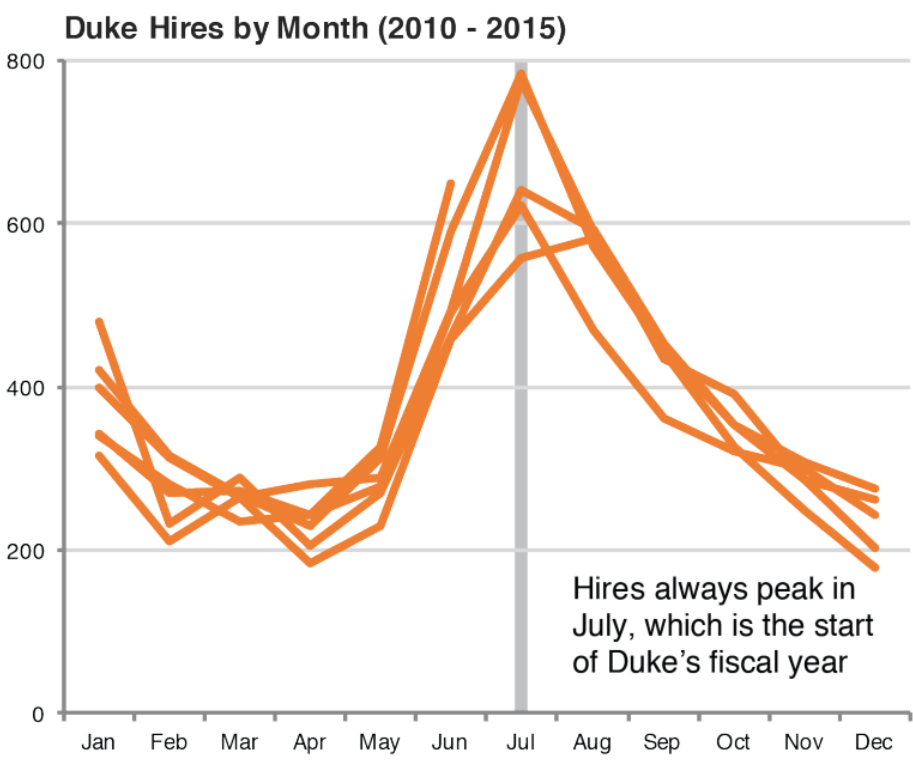
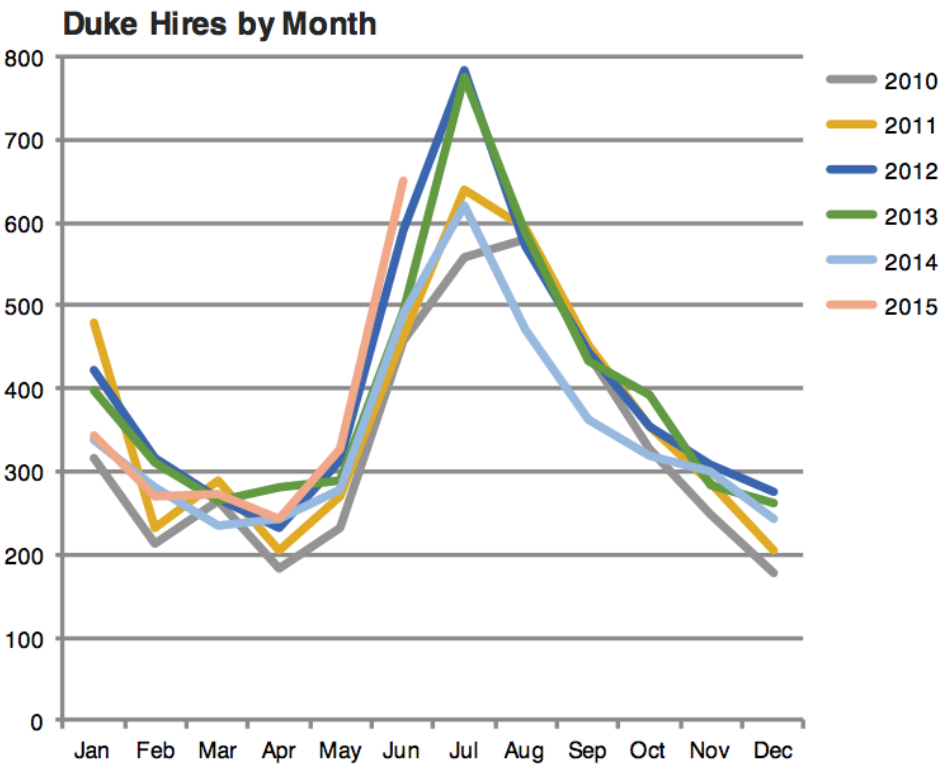
Judging relative area



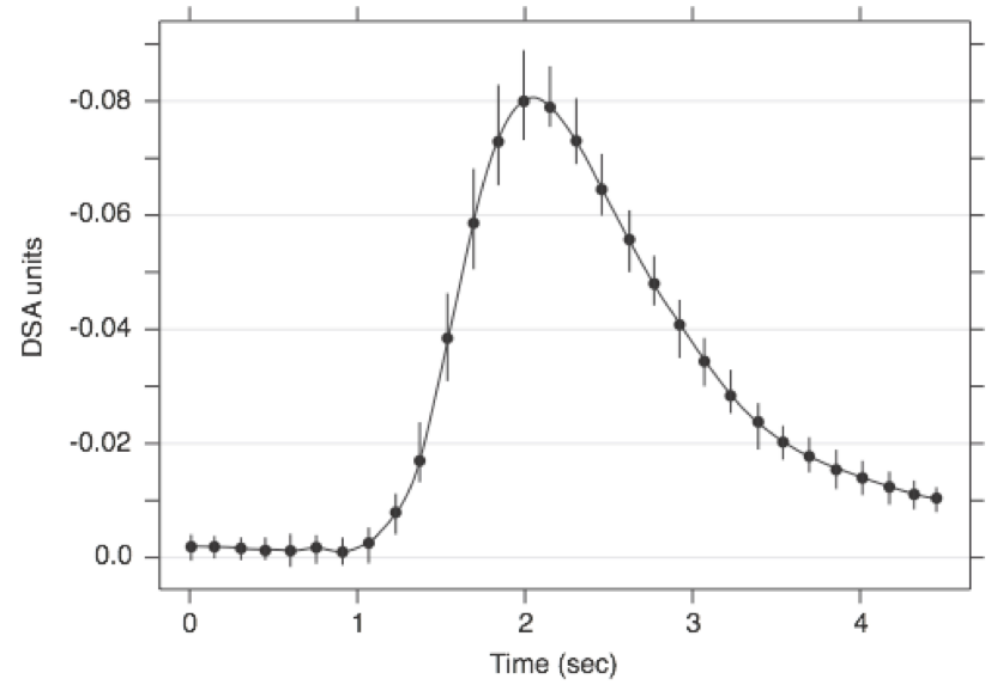
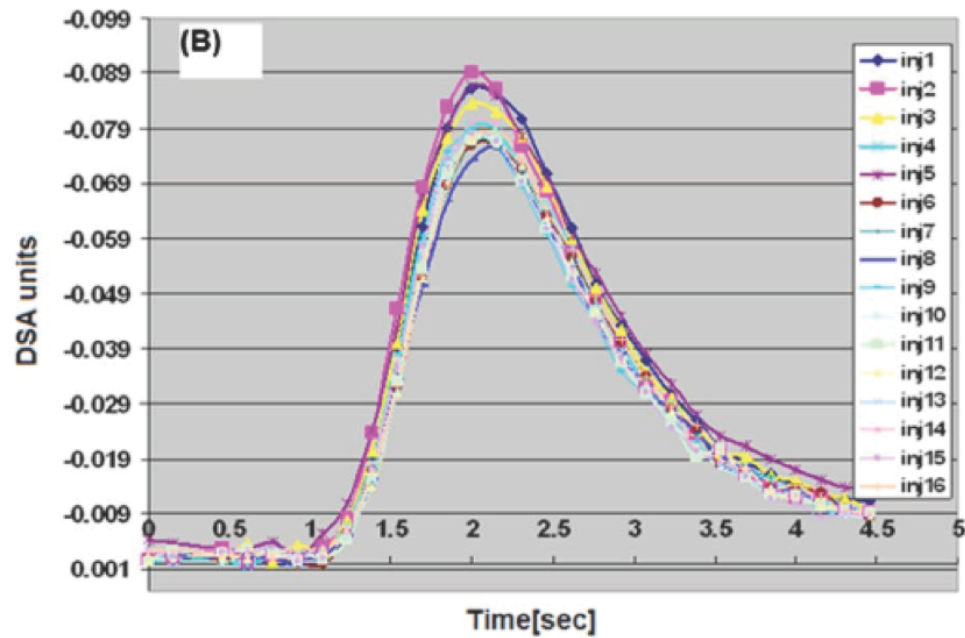
Use color to draw attention



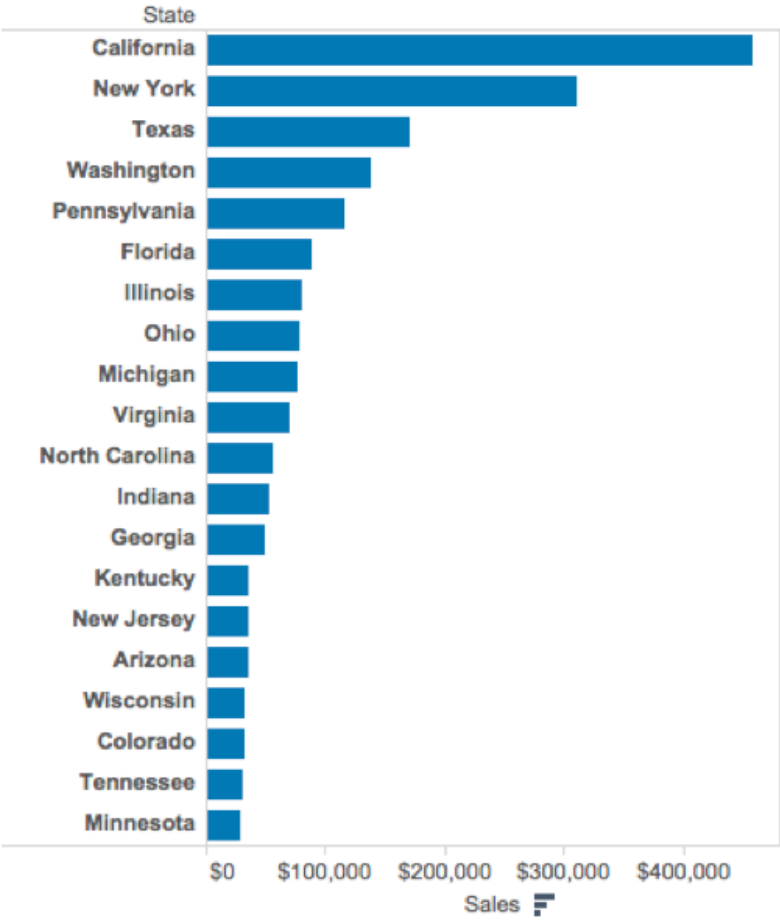
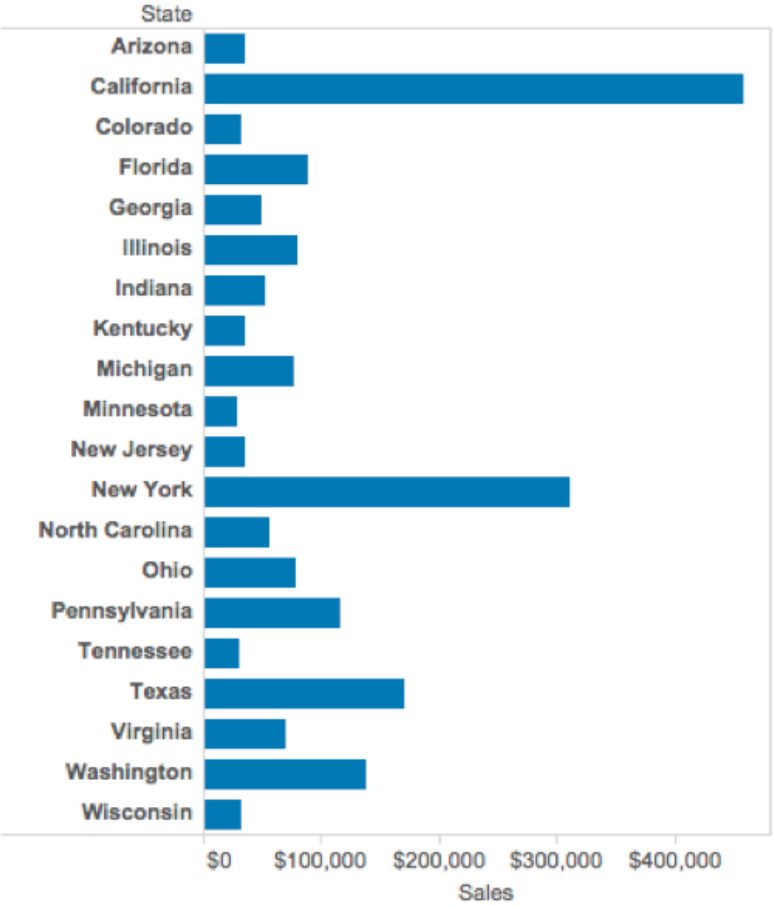
Tell a story



Leave out non-story details



Ordering matter



Clearly indicate missing data

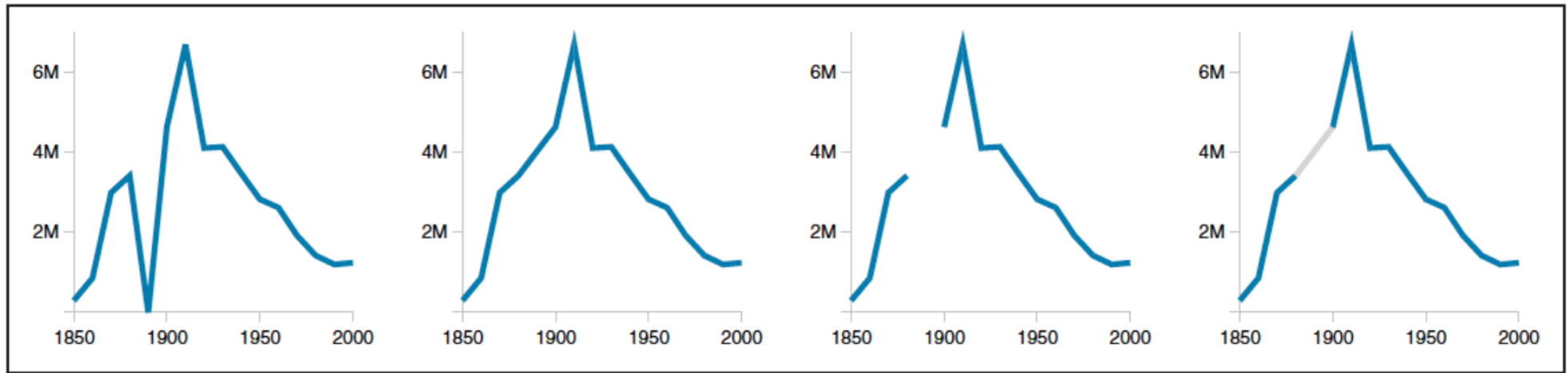
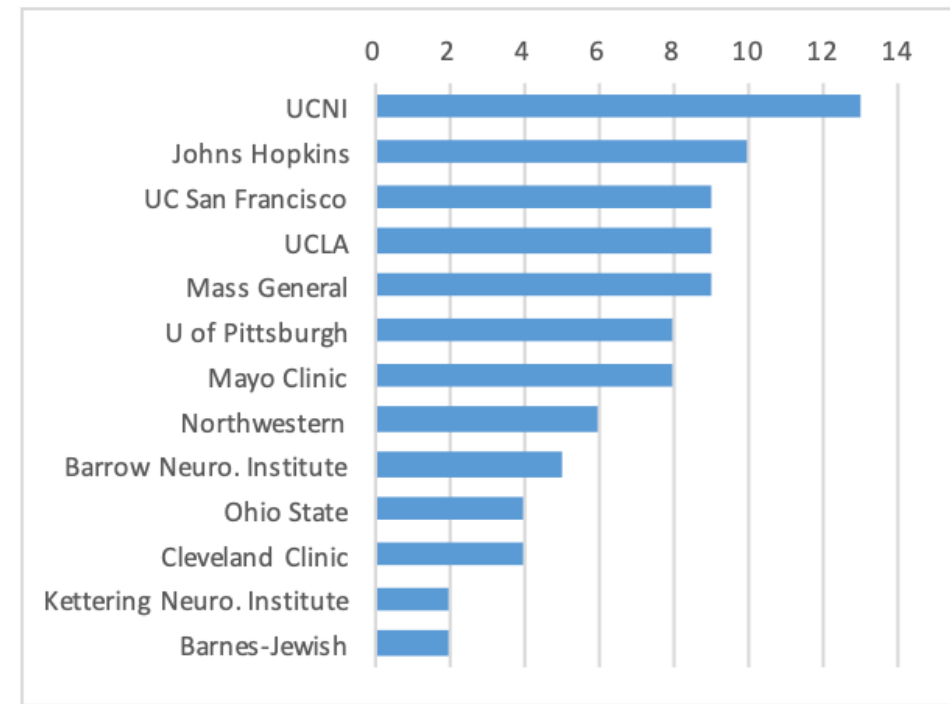
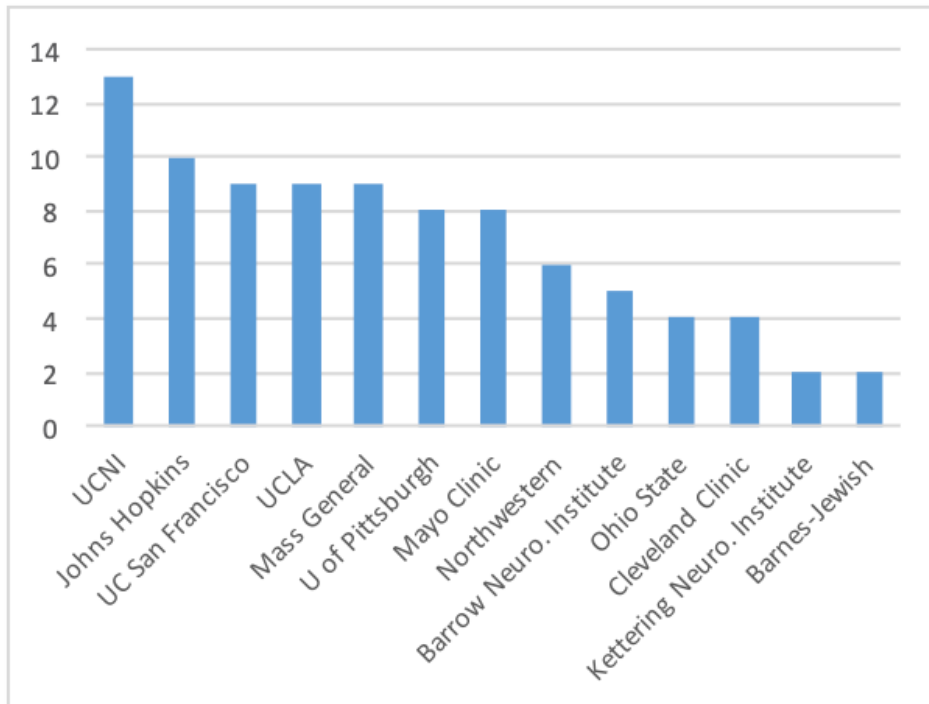


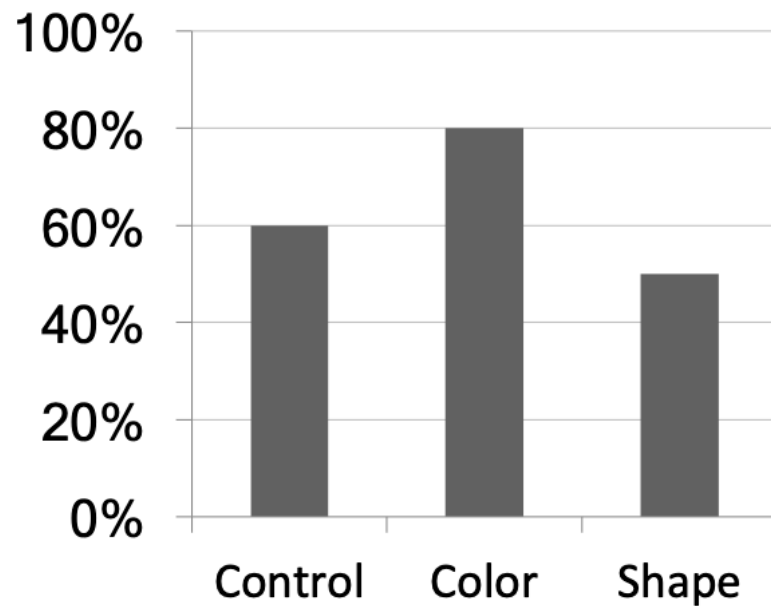
Figure 4. Alternative representations of missing data in a line chart. The data are U.S. census counts of people working as 'Farm Laborers'; values from 1890 are missing due to records being burned in a fire. (a) Missing data is treated as a zero value. (b) Missing data is ignored, resulting in a line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

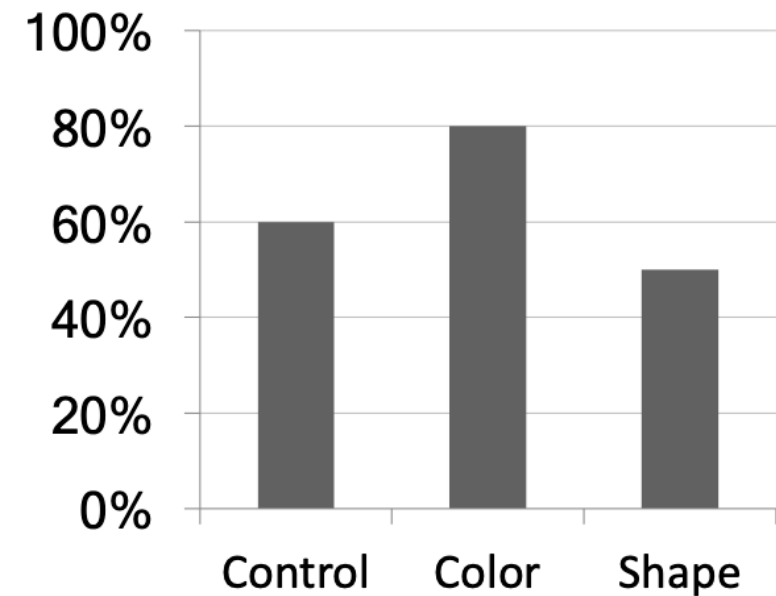


Use descriptive titles

**Accuracy versus
Color and Shape**



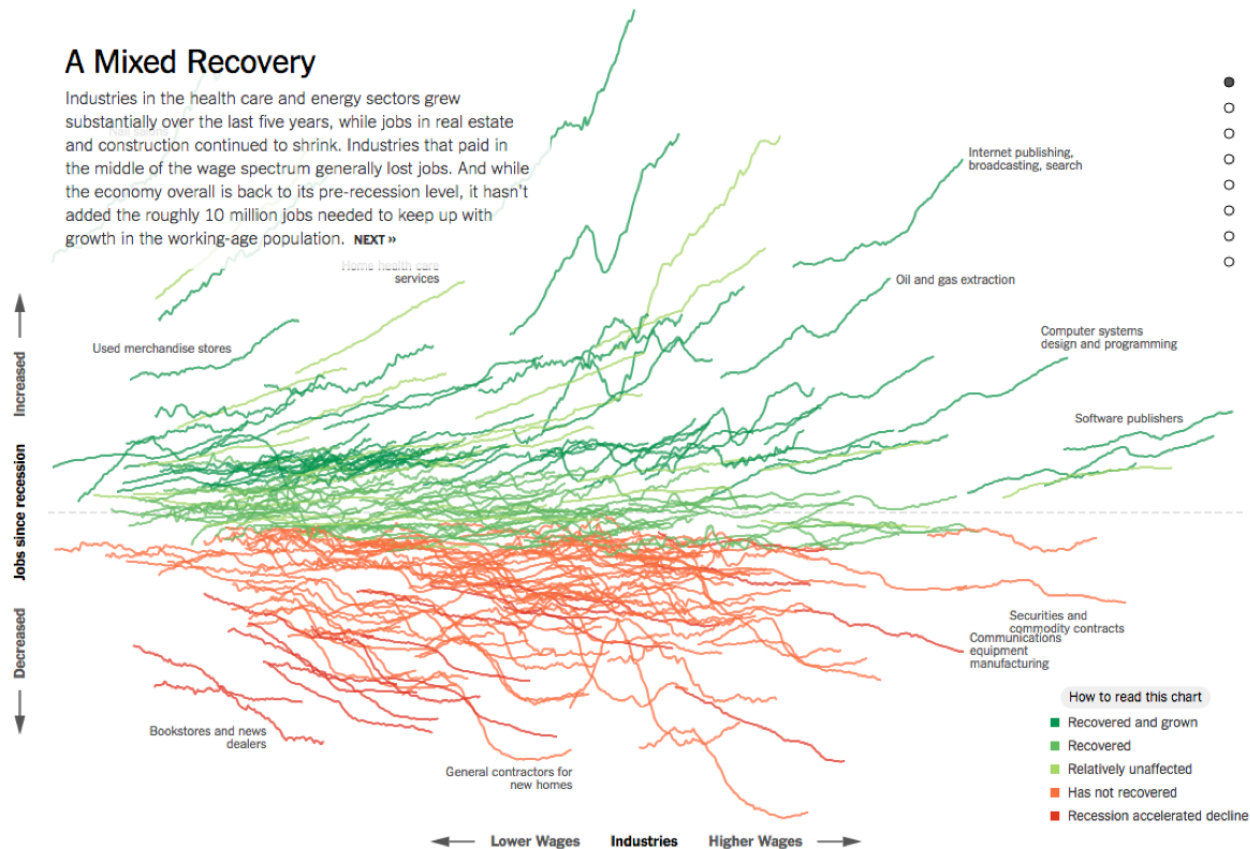
**Accuracy Improved by
Color, not Shape**



Annotate figures



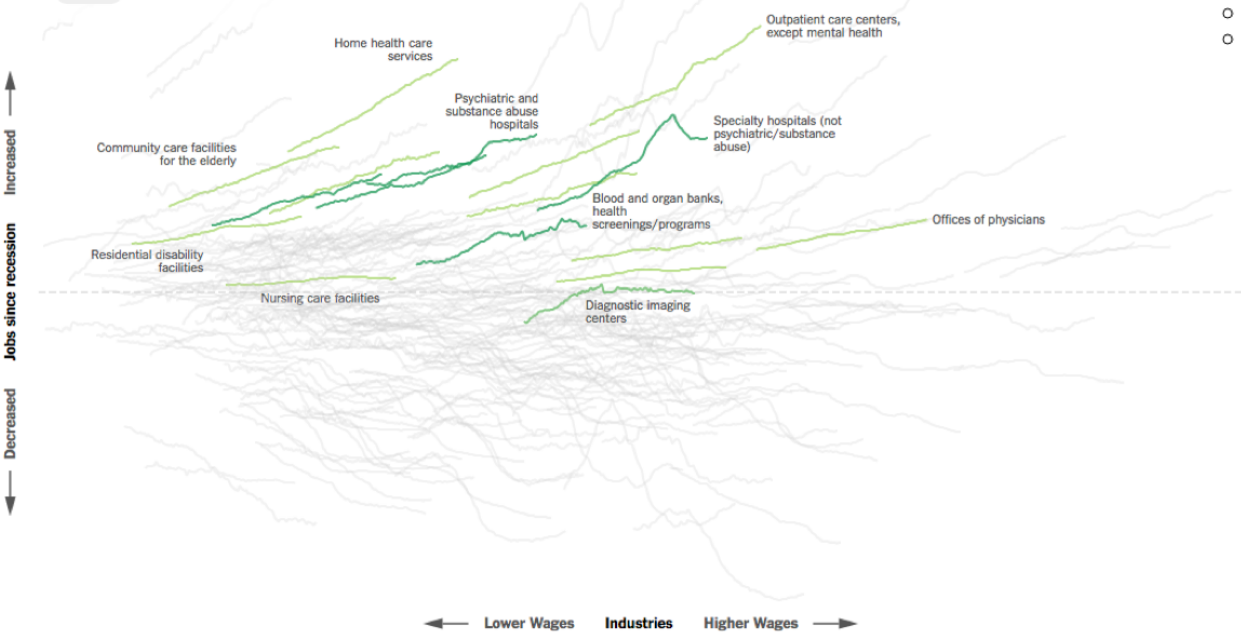
All of the data doesn't tell a story



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 »**

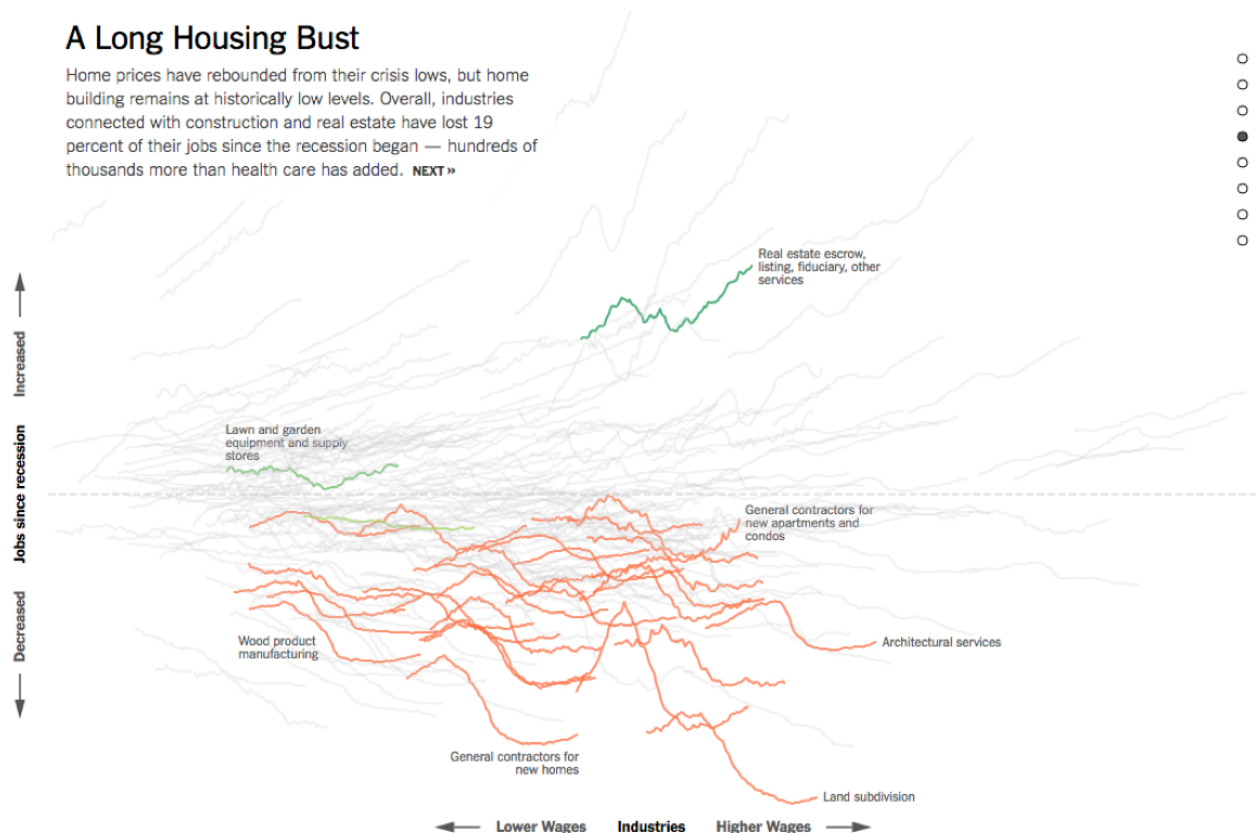


○
○
●
○
○
○
○
○

All of the data doesn't tell a story

A Long Housing Bust

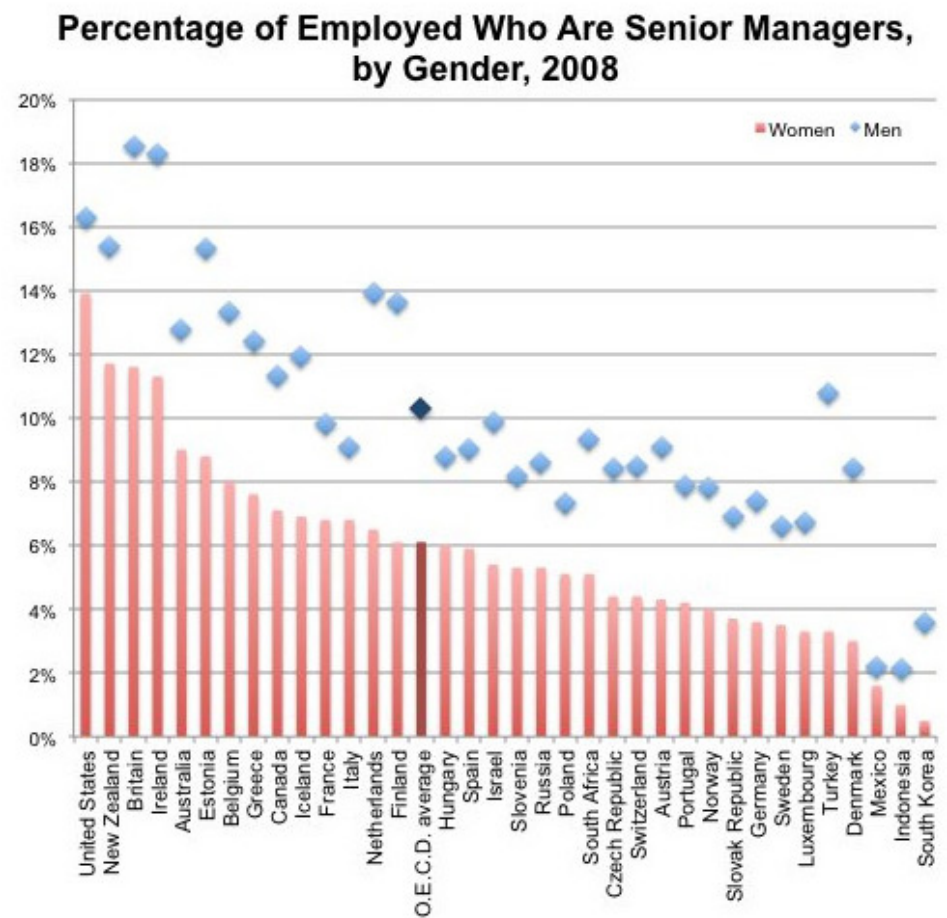
Home prices have rebounded from their crisis lows, but home building remains at historically low levels. Overall, industries connected with construction and real estate have lost 19 percent of their jobs since the recession began — hundreds of thousands more than health care has added. **NEXT »**



-
-
-
-
-
-
-
-

Chart Remakes / Makeovers

The Why Axis - Gender Gap



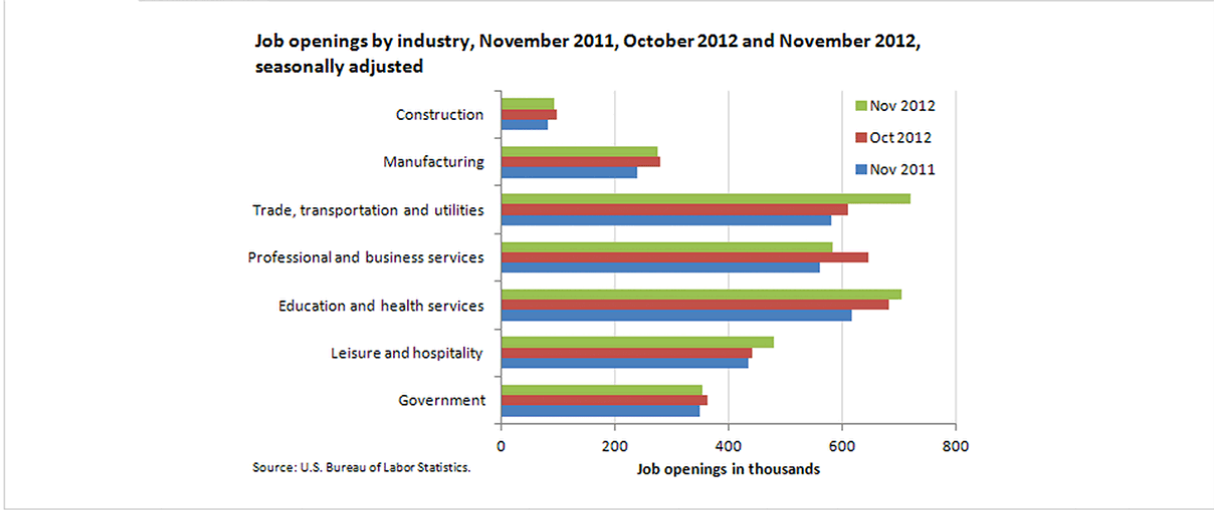
The Why Axis - BLS

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



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).

Other Resources

- Duke Library - Center for Data and Visualization Sciences - <https://library.duke.edu/data/>
- Tidy tuesday - <https://github.com/rfordatascience/tidytuesday>
- Twitter / Bluesky / Mastodon - #dataviz, #tidytuesday
- Books:
 - Wickham, Navarro, Pedersen. *ggplot2: Elegant Graphics for Data Analysis*. 3rd edition. Faller, 2021.
 - Wilke. *Fundamentals of Data Visualization*. O'Reilly Media, 2019.
 - Healy. *Data Visualization: A Practical Introduction*. Princeton University Press, 2018.
 - Tufte. *The visual display of quantitative information*. 2nd edition. Connecticut Graphics Press, 2015.

Acknowledgments

Above materials are derived in part from the following sources:

- sVisualization training materials originally developed by Angela Zoss and Eric Monson
- Duke Center for Data and Visualization Sciences