

$$R\left(\quad \right)$$

“R”

R

R

R

CC BY-NC-ND 3.0
Recovery kākāpō

Amazon
248

Kākāpō

•

<https://mine-cetinkaya-rundel.github.io/r4ds-solutions>

Contributor Code of Conduct

<https://www.netlify.com>

“R”
Çetinkaya-Rundel

Posit(RStudio)

Mine

- “ ” Whole game “ ”
- “ ” Visualize ggplot2 ,
- “ ” Transform
- “ ” Import
- “ ” Program tidyverse tidy
base R
- “ ” Modeling tidymodels Max
Kuhn Julia Silge Tidy Modeling with R
- “ ” Communicate Quarto R Markdown Quarto

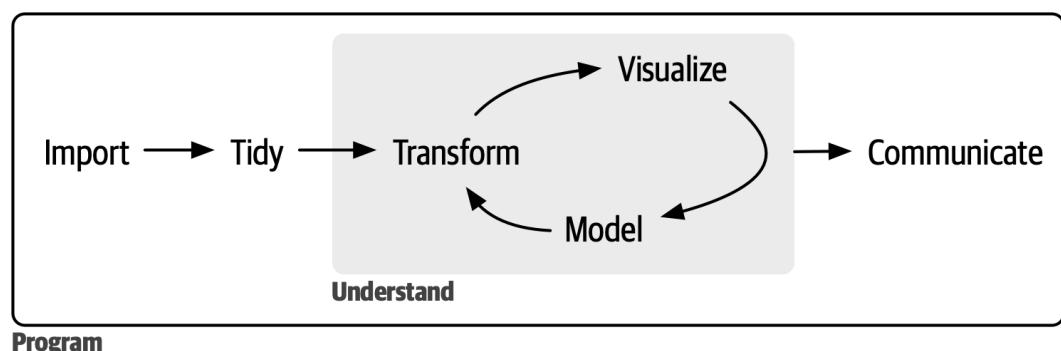
R

R

R

R

??



1

(import) R

Web (API)

R

R

(tidy)

(transform)
(wrangling)

! ()

()

()

10

(Visualization)

(Models)

(communication)

(programming)

80/20

80% 20%

()

80% 20%

ge Tidy Modeling with R

tidymodels

Max Kuhn Julia Sil-
tidyverse

GB Parquet

(10-100GB) data.table

tidyverse

Python Julia

Python Julia
R

R Python

Programming with R

R RStudio tidyverse R

Garrett Hands on

R

R

R CRAN (the **c**omprehensive **R** archive **n**etwork) <https://cloud.r-project.org>
 R 2-3 R4.2.0

RStudio

RStudio	https://posit.co/download/rstudio-desktop/
RStudio	RStudio 2022.02.0
RStudio ??	R
	¹

Tidyverse

R R	R	R	tidyverse	t idyverse	R
tidyverse					

```
install.packages("tidyverse")
```

enter R CRAN

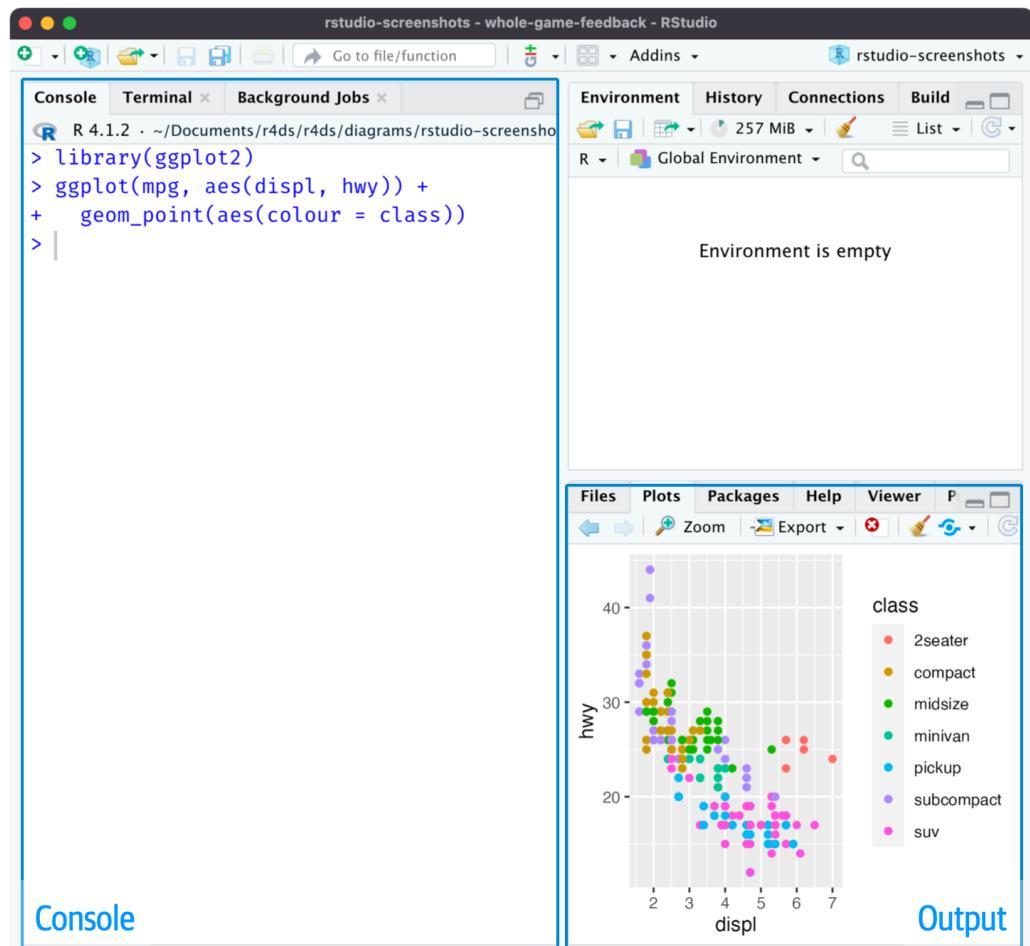
library()	library()
-----------	-----------

```
library(tidyverse)
#> -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
#> v dplyr     1.1.4    v readr     2.1.5
#> vforcats   1.0.0    v stringr   1.5.1
#> v ggplot2   3.5.0    v tibble    3.2.1
#> v lubridate 1.9.3    v tidyr     1.3.1
#> v purrr    1.0.2
#> -- Conflicts ----- tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()   masks stats::lag()
#> i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

tidyverse 9 : dplyrforcatsggplot2lubridpurrrreaderstringrtibbletidyrtidyverse

tidyverse tidyverse_update()

¹If you'd like a comprehensive overview of all of RStudio's features, see the RStudio User Guide at <https://docs.posit.co/ide/user>.



2: RStudio IDE

R

tidyverse verse	messyverse	universes	R
tidyverse		R	tidy-

```
install.packages(
  c("arrow", "babynames", "curl", "duckdb", "gapminder",
    "ggrepel", "ggridges", "ggthemes", "hexbin", "janitor", "Lahman",
    "leaflet", "maps", "nycflights13", "openxlsx", "palmerpenguins",
    "repurrrsive", "tidymodels", "writexl")
)
```

```
library(ggrepel)
#> Error in library(ggrepel) : there is no package called 'ggrepel'
```

```
install.packages("ggrepel")
```

R

R

```
1 + 2
#> [1] 3
```

:

```
> 1 + 2
[1] 3
```

```
> > #>
```

- `sum()` `mean()`
- R () , `flights` `x`.
- to make it clear which package an object comes from,
`dplyr::mutate()` `nycflights13::flights` R

Hadley Garrett R () !
 pull requests GitHub 259 (): @a-rosenberg,
 Tim Becker (@a2800276), Abinash Satapathy (@Abinashbunty), Adam Gruer
 (@adam-gruer), adi pradhan (@adidoit), A. s. (@Adrianzo), Aep Hidayatuloh
 (@aephidayatuloh), Andrea Gilardi (@agila5), Ajay Deonarine (@ajay-d),
 @AlanFeder, Daihe Sui (@alansuidaihe), @alberto-agudo, @AlbertRapp,
 @aleloj, pete (@alonzi), Alex (@ALShum), Andrew M. (@amacfarland),
 Andrew Landgraf (@andland), @andyhuynh92, Angela Li (@angela-li), Antti
 Rask (@AnttiRask), LOU Xun (@aquahead), @ariespirgel, @august-18,
 Michael Henry (@aviast), Azza Ahmed (@azzaea), Steven Moran (@bam-
 booforest), Brian G. Barkley (@BarkleyBG), Mara Averick (@batpigandme),
 Oluwafemi OYEDELE (@BB1464), Brent Brewington (@bbrewington), Bill
 Behrman (@behrman), Ben Herbertson (@benherbertson), Ben Marwick
 (@benmarwick), Ben Steinberg (@bensteinberg), Benjamin Yeh (@ben-
 tyeh), Betul Turkoglu (@betulturkoglu), Brandon Greenwell (@bgreenwell),
 Bianca Peterson (@BinxiePeterson), Birger Niklas (@BirgerNi), Brett Klamer
 (@bklamer), @boardtc, Christian (@c-hoh), Caddy (@caddycarine), Camille V
 Leonard (@camilleleonard), @canovasjm, Cedric Batailler (@cedricbatailler),
 Christina Wei (@christina-wei), Christian Mongeau (@chrMongeau), Cooper
 Morris (@coopermor), Colin Gillespie (@csgillespie), Rademeyer Vermaak
 (@csrvermaak), Chloe Thierstein (@cthierst), Chris Saunders (@ctsa), Abhinav
 Singh (@curious-abhinav), Curtis Alexander (@curtisalexander), Christian
 G. Warden (@cwarden), Charlotte Wickham (@cwickham), Kenny Darrell
 (@darrkj), David Kane (@davidkane9), David (@davidrsch), David Rubinger
 (@davidrubinger), David Clark (@DDClark), Derwin McGeary (@derwinm-
 geary), Daniel Gromer (@dgromer), @Divider85, @djbirke, Danielle Navarro
 (@djnavarro), Russell Shean (@DOH-RPS1303), Zhuoer Dong (@dongzhuoer),
 Devin Pastoor (@dpastoor), @DSGeoff, Devarshi Thakkar (@dthakkar09),
 Julian During (@duju211), Dylan Cashman (@dylancashman), Dirk Ed-
 delbuettel (@eddelbuettel), Edwin Thoen (@EdwinTh), Ahmed El-Gabbas
 (@elgabbas), Henry Webel (@enryH), Ercan Karadas (@ercan7), Eric Kitaif
 (@EricKit), Eric Watt (@ericwatt), Erik Erhardt (@erikerhardt), Etienne
 B. Racine (@etiennebr), Everett Robinson (@evjrob), @fellennert, Flemming
 Miguel (@flemmingmiguel), Floris Vanderhaeghe (@florisvdh), @funkybluehen,
 @gabrivera, Garrick Aden-Buie (@gadenbuie), Peter Ganong (@ganong123),
 Gerome Meyer (@GeroVanMi), Gleb Ebert (@gl-eb), Josh Goldberg (@Gold-
 bergData), bahadir cankardes (@gridgrad), Gustav W Delius (@gustavdelius),
 Hao Chen (@hao-trivago), Harris McGehee (@harrismcgehee), @hendrikweisser,
 Hengni Cai (@hengnicai), Iain (@Iain-S), Ian Sealy (@iansealy), Ian Lytle
 (@ijlyttle), Ivan Krukova (@ivan-krukova), Jacob Kaplan (@jacobkap), Jazz
 Weisman (@jazzlw), John Blischak (@jdblischak), John D. Storey (@jdstorey),
 Gregory Jefferis (@jefferis), Jeffrey Stevens (@JeffreyRStevens),
 (@Jel-
 dorPKU), Jennifer (Jenny) Bryan (@jennybc), Jen Ren (@jenren), Jeroen

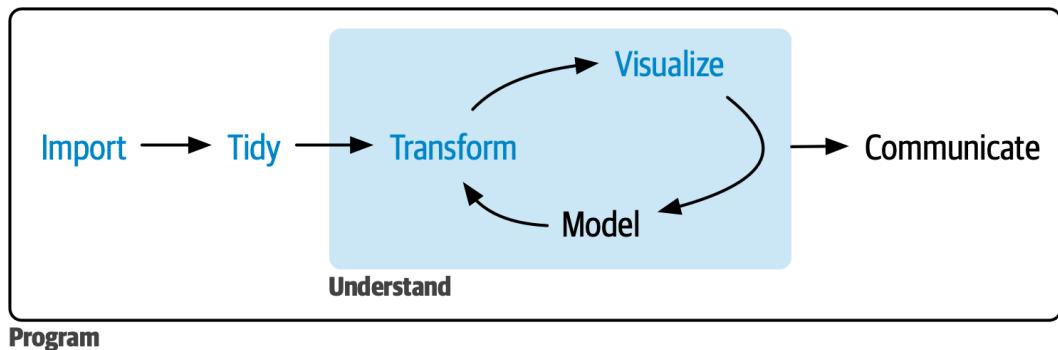
Janssens (@jeroenjanssens), @jeromecholewa, Janet Wesner (@jilmun), Jim Hester (@jimhester), JJ Chen (@jjchern), Jacek Kolacz (@jkolacz), Joanne Jang (@joannejang), @johannes4998, John Sears (@johnsears), @jonathanflint, Jon Calder (@jonmcalder), Jonathan Page (@jonpage), Jon Harmon (@jonthegEEK), JooYoung Seo (@jooyoungseo), Justinas Petuchovas (@jpetuchovas), Jordan (@jrdnbradford), Jeffrey Arnold (@jrnold), Jose Roberto Ayala Solares (@jroberayalas), Joyce Robbins (@jtr13), @juandering, Julia Stewart Lowndes (@jules32), Sonja (@kaetschap), Kara Woo (@karawoo), Katrin Leinweber (@katrinleinweber), Karandeep Singh (@kdpsingh), Kevin Perese (@kevinx-perese), Kevin Ferris (@kferris10), Kirill Sevastyanenko (@kirillseva), Jonathan Kitt (@KittJonathan), @koalabearski, Kirill Müller (@krmlr), Rafal Kucharski (@kucharsky), Kevin Wright (@kwstat), Noah Landesberg (@landesbergn), Lawrence Wu (@lawwu), @lindbrook, Luke W Johnston (@lwjohnst86), Kara de la Marck (@MarckK), Kunal Marwaha (@marwahaha), Matan Hakim (@matanhakim), Matthias Liew (@MatthiasLiew), Matt Wittbrodt (@MattWittbrodt), Mauro Lepore (@maurolepore), Mark Beveridge (@mbeveridge), @mcewenkhundi, mcsnowface, PhD (@mcsnowface), Matt Herman (@mfherman), Michael Boerman (@michaelboerman), Mitsuo Shiota (@mitsuoxv), Matthew Hendrickson (@mjhendrickson), @MJMarshall, Misty Knight-Finley (@mkfin7), Mohammed Hamdy (@mmhamdy), Maxim Nazarov (@mnazarov), Maria Paula Caldas (@mpaulacaldas), Mustafa Ascha (@mustafaascha), Nelson Areal (@nareal), Nate Olson (@nate-dolson), Nathanael (@nateaff), @nattalides, Ned Western (@NedJWestern), Nick Clark (@nickclark1000), @nickelas, Nirmal Patel (@nirmalpatel), Nischal Shrestha (@nischalshrestha), Nicholas Tierney (@njtierney), Jakub Nowosad (@Nowosad), Nick Pullen (@nstjhp), @olivier6088, Olivier Cailloux (@oliviercailloux), Robin Penfold (@p0bs), Pablo E. Garcia (@pabloedug), Paul Adamson (@padamson), Penelope Y (@penelopeysm), Peter Hurford (@peterhurford), Peter Baumgartner (@petzi53), Patrick Kennedy (@pkq), Pooya Taherkhani (@pooyataher), Y. Yu (@PursuitOfDataScience), Radu Grosu (@radugrosu), Ranae Dietzel (@Ranae), Ralph Straumann (@rastrau), Rayna M Harris (@raynamharris), @Reece-Goding, Robin Gertenbach (@rgertenbach), Jajo (@RIngyao), Riva Quiroga (@rivaquiropa), Richard Knight (@RJHKnight), Richard Zijdeman (@rlzijdemann), @robertchu03, Robin Kohrs (@RobinKohrs), Robin (@Robinlovelace), Emily Robinson (@robinsones), Rob Tenorio (@robtenorio), Rod Mazloomi (@RodAli), Rohan Alexander (@RohanAlexander), Romero Morais (@Romero-Barata), Albert Y. Kim (@rudeboybert), Saghir (@saghirb), Hojjat Salmasian (@salmasian), Jonas (@sauercrowd), Vebash Naidoo (@sciencificity), Seamus McKinsey (@seamus-mckinsey), @seanpwiliams, Luke Smith (@seasmith), Matthew Sedaghatfar (@sedaghatfar), Sebastian Kraus (@sekR4), Sam Firke (@sfirke), Shannon Ellis (@ShanEllis), @shoili, Christian Heinrich (@Shurakai), S'busiso Mkhondwane (@sibusiso16), SM Raiyyan (@sm-raiyyan), Jakob Krigovsky (@sonicdoe), Stephan Koenig (@stephan-koenig), Stephen Balogun (@stephenbalogun), Steven M. Mortimer (@StevenMMortimer), Stéphane Guillou (@stragu), Sulgi Kim (@sulgik), Sergiusz Bleja (@svenski), Tal Galili (@talgalili), Alec Fisher (@Taurenamo), Todd Gerarden (@tgerarden), Tom

Godfrey (@thomasggodfrey), Tim Broderick (@timbroderick), Tim Waterhouse (@timwaterhouse), TJ Mahr (@tjmahr), Thomas Klebel (@tklebel), Tom Prior (@tomjamesprior), Terence Teo (@tteo), @twgardner2, Ulrik Lyngs (@ulyngs), Shinya Uryu (@uribo), Martin Van der Linden (@vanderlindenma), Walter Somerville (@waltersom), @werkstattcodes, Will Beasley (@wibeasley), Yihui Xie (@yihui), Yiming (Paul) Li (@yimingli), @yingxingwu, Hiroaki Yutani (@yutannihilation), Yu Yu Aung (@yuyu-aung), Zach Bogart (@zachbogart), @zeal626, Zeki Akyol (@zekiakyol).

r4ds <https://r4ds.hadley.nz> <https://github.com/hadley/>
Quarto

Part I

?? “ ” ()



3:

- R ?? ggplot2
- ??
- ??
- R ?? .csv R
- R ?? ?? ?? R
- ??

Chapter 1

1.1

“ . ”

— John Tukey

R , ggplot2 ggplot2 ggplot2
ggplot2 ggplot2

1.1.1

ggplot2 tidyverse tidyverse

```
library(tidyverse)
#> -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
#> v dplyr     1.1.4     v readr     2.1.5
#> vforcats   1.0.0     v stringr   1.5.1
#> v ggplot2   3.5.0     v tibble    3.2.1
#> v lubridate 1.9.3     v tidyrr    1.3.1
#> v purrr    1.0.2
#> -- Conflicts ----- tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()   masks stats::lag()
#> i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

tidyverse

tidyverse

base R

1

1 conflicted package

<https://conflicted.r-lib.org>

```
there is no package called 'tidyverse'           library()
```

```
install.packages("tidyverse")
library(tidyverse)
```

```
tidyverse,    palmerpenguins    penguins    Palmer      ggthemes
```

```
library(palmerpenguins)
library(ggthemes)
```

1.2

1.2.1 penguins

```
palmerpenguins palmerpenguins::penguins  penguins
( ) ( )  penguins  344  Kristen Gorman      2
```

-
-
-
-
- “ ”

```
R          tibble  tidyverse    tibbles
```

```
penguins
#> # A tibble: 344 x 8
#>   species     island   bill_length_mm bill_depth_mm flipper_length_mm
#>   <fct>     <fct>        <dbl>        <dbl>            <int>
#> 1 Adelie    Torgersen       39.1        18.7            181
```

²Horst AM, Hill AP, Gorman KB (2020). palmerpenguins: Palmer Archipelago (Antarctica) penguin data. R package version 0.1.0. <https://allisonhorst.github.io/palmerpenguins/>. doi: 10.5281/zenodo.3960218.

```
#> 2 Adelie Torgersen      39.5      17.4      186
#> 3 Adelie Torgersen      40.3       18      195
#> 4 Adelie Torgersen      NA        NA      NA
#> 5 Adelie Torgersen      36.7      19.3      193
#> 6 Adelie Torgersen      39.3      20.6      190
#> # i 338 more rows
#> # i 3 more variables: body_mass_g <int>, sex <fct>, year <int>
```

```
8      glimpse()          RStudio View(penguins)
```

```
glimpse(penguins)
#> Rows: 344
#> Columns: 8
#> $ species      <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, A~
#> $ island        <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torge~
#> $ bill_length_mm <dbl> 39.1, 39.5, 40.3, NA, 36.7, 39.3, 38.9, 39.2, 34.~
#> $ bill_depth_mm  <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.~
#> $ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, ~
#> $ body_mass_g    <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 347~
#> $ sex            <fct> male, female, female, NA, female, male, female, m~
#> $ year           <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2~
```

penguins

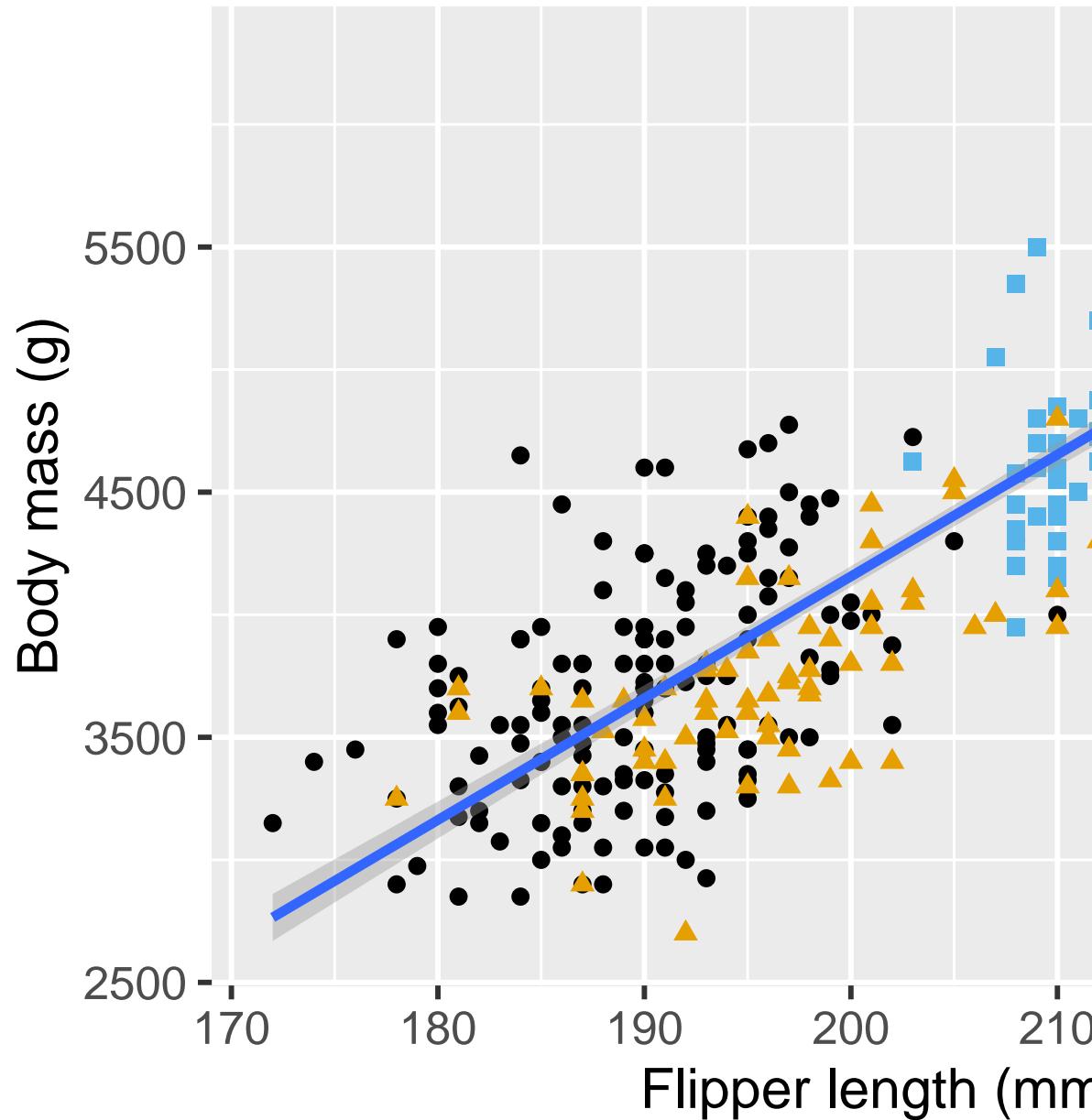
1. species Adelie, Chinstrap, or Gentoo
 2. flipper_length_mm:
 3. body_mass_g:

penguins ?penguins

1.2.2

Body mass and flipper length

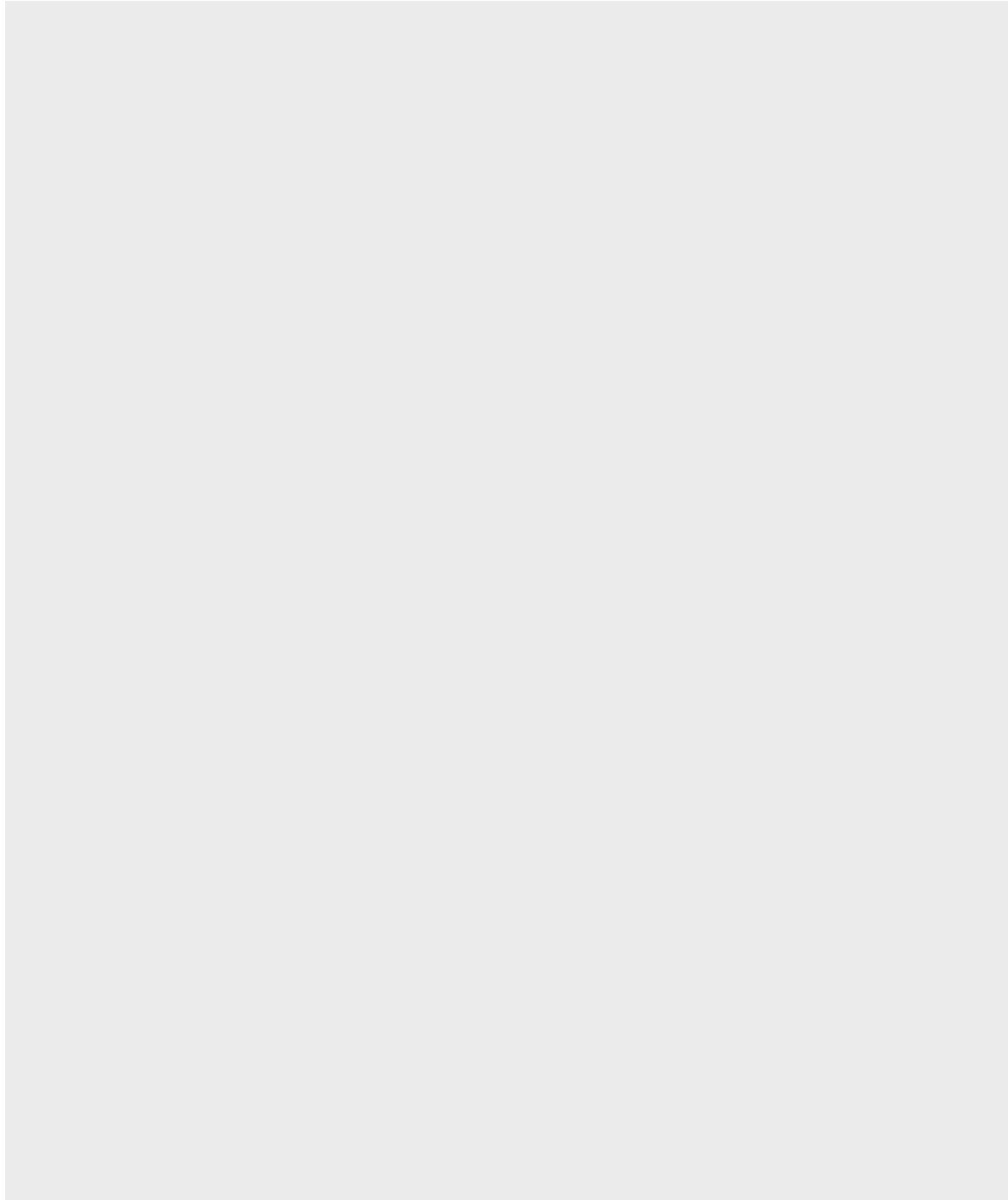
Dimensions for Adelie, Chinstrap, and Gentoo penguins



1.2.3 ggplot

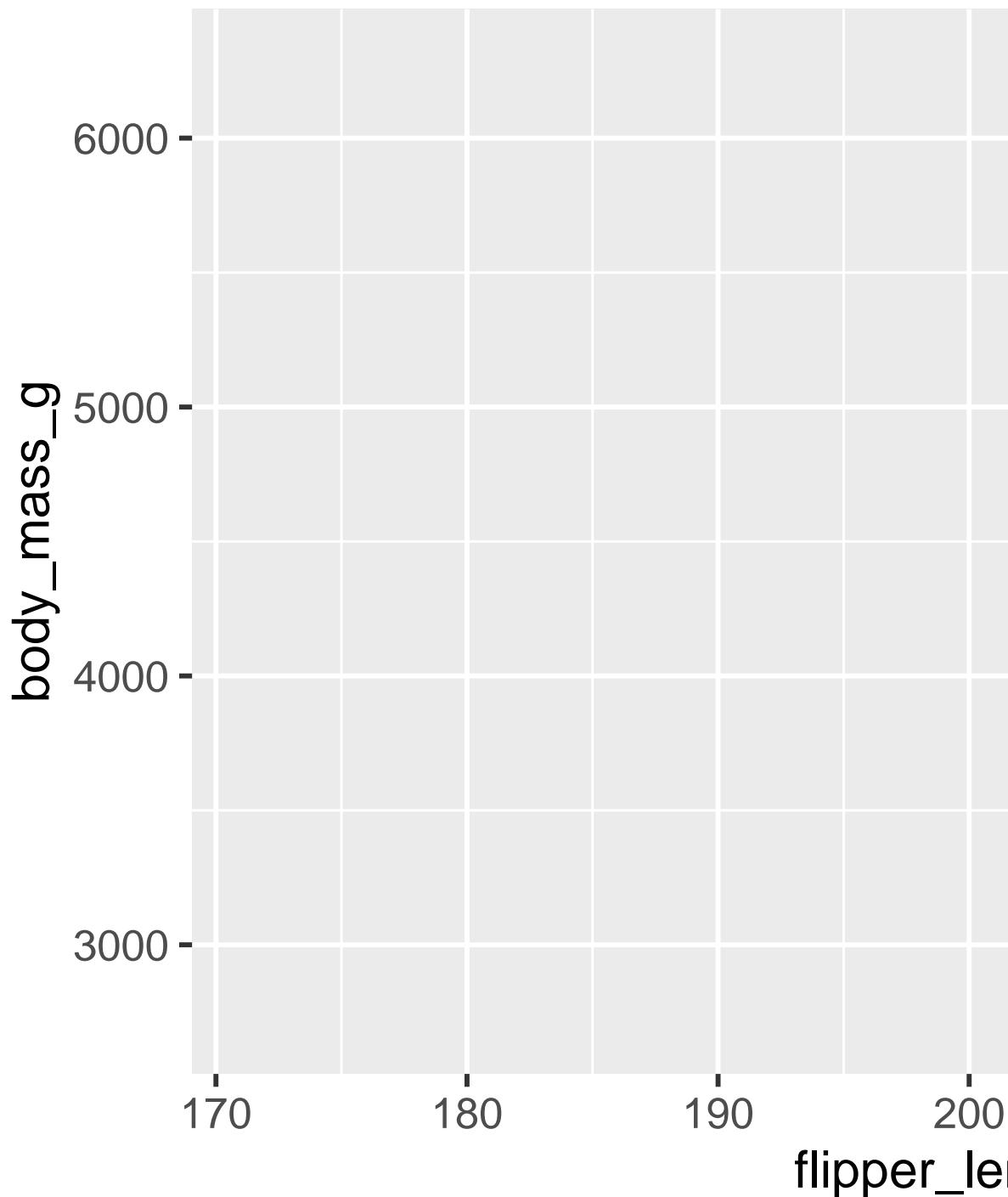
```
ggplot2  ggplot()  
penguins)      penguins  
ggplot()  
ggplot(data =
```

```
ggplot(data = penguins)
```



```
ggplot()      ggplot()  mapping      aesthetics
mapping  aes()    aes() x y      x y      x       y       g
gplot2        penguins
```

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
)
```

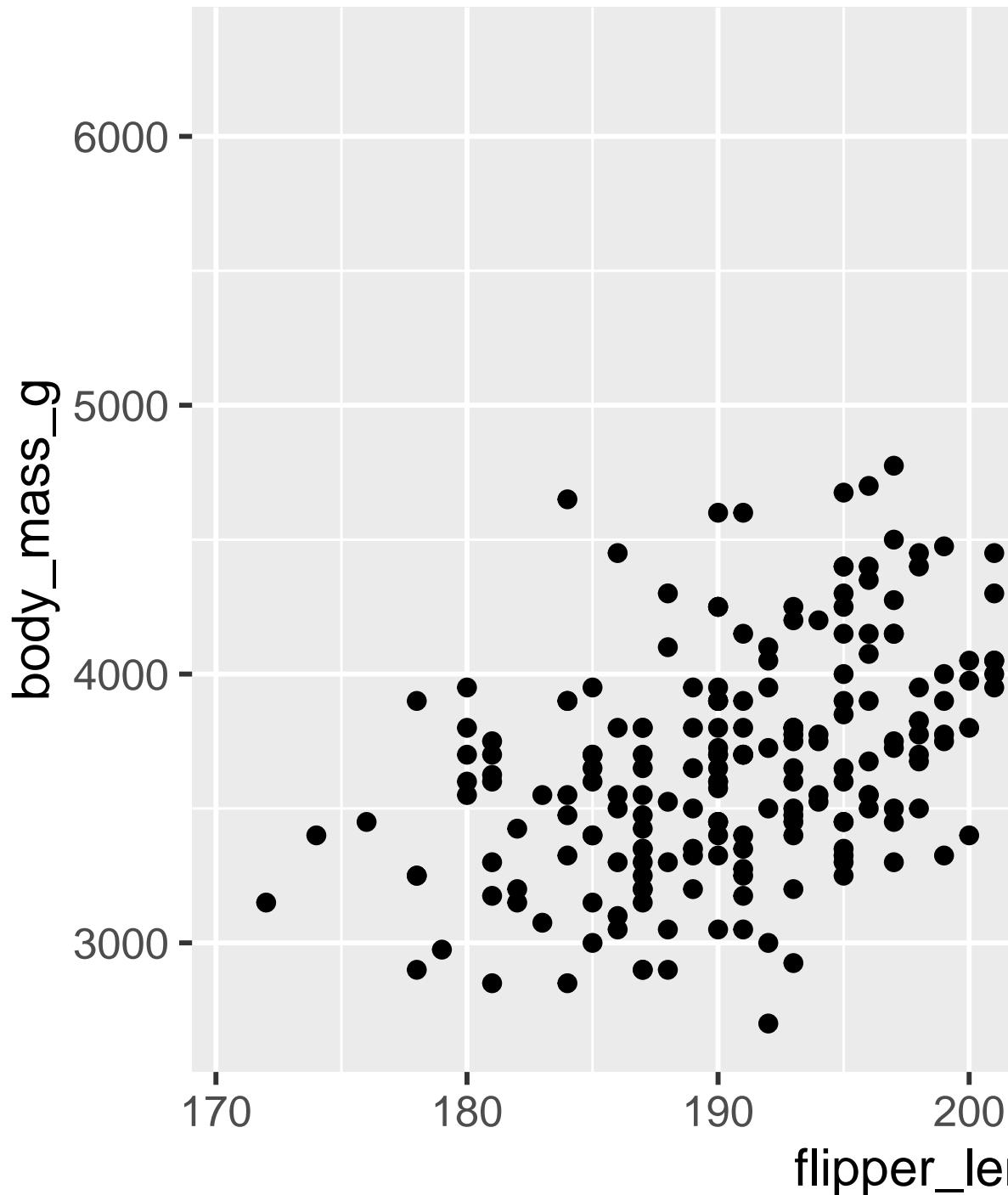


```
x      y    penguins
```

```
geom   geom      geom ggplot2  geom_
geom_bar()  geom_line()  geom_boxplot()  geom_point()
```

```
geom_point()          g gplot2    geom      geom      ??
```

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point()
#> Warning: Removed 2 rows containing missing values or values outside the scale range
#> (`geom_point()`).
```



```
“ ” “ ” “ ” ?” )
```

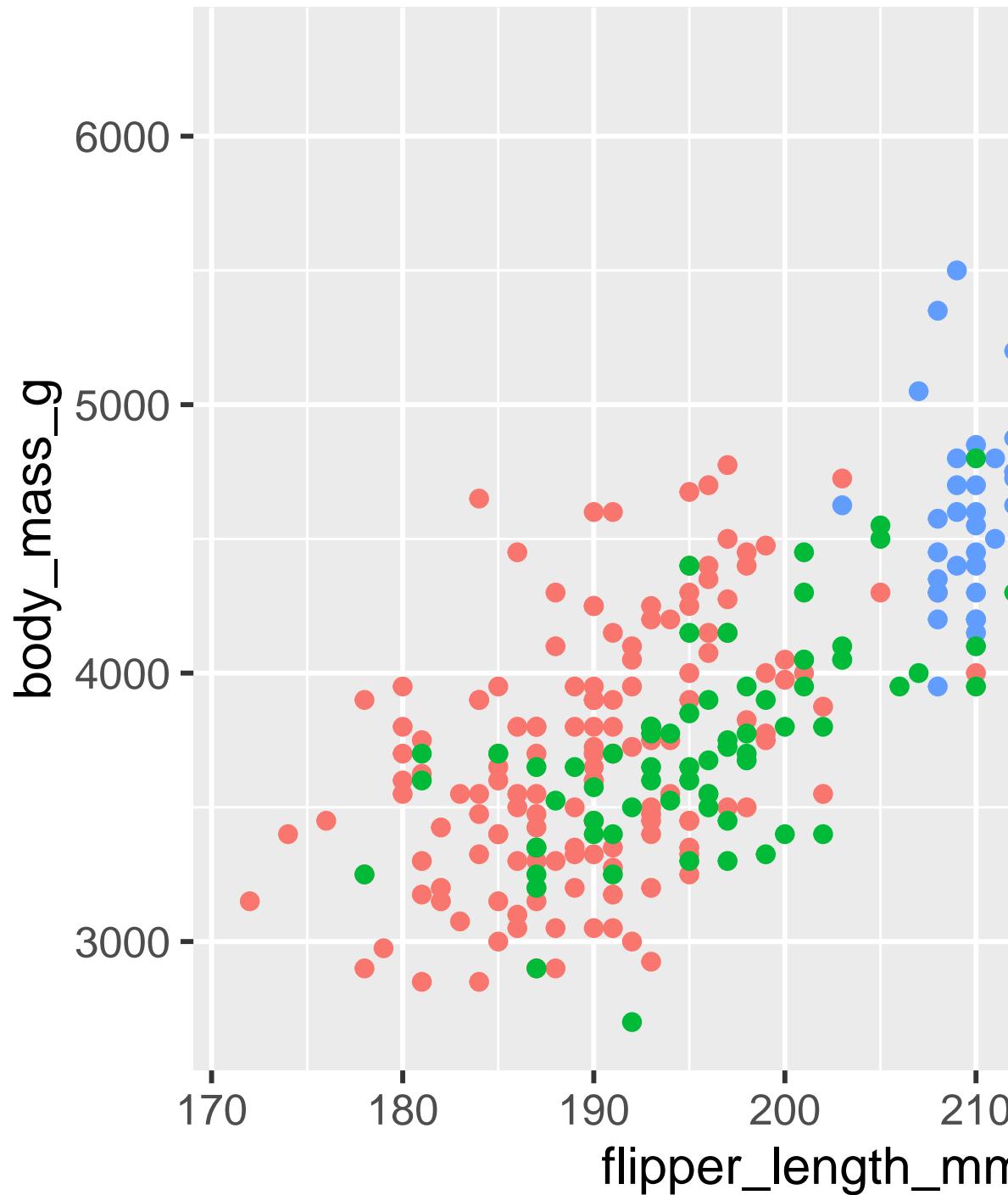
Removed 2 rows containing missing values (geom_point()).

```
ggplot2 R ggplot2  
??
```

1.2.4

```
geom aes() ggplot2 gg-  
plot
```

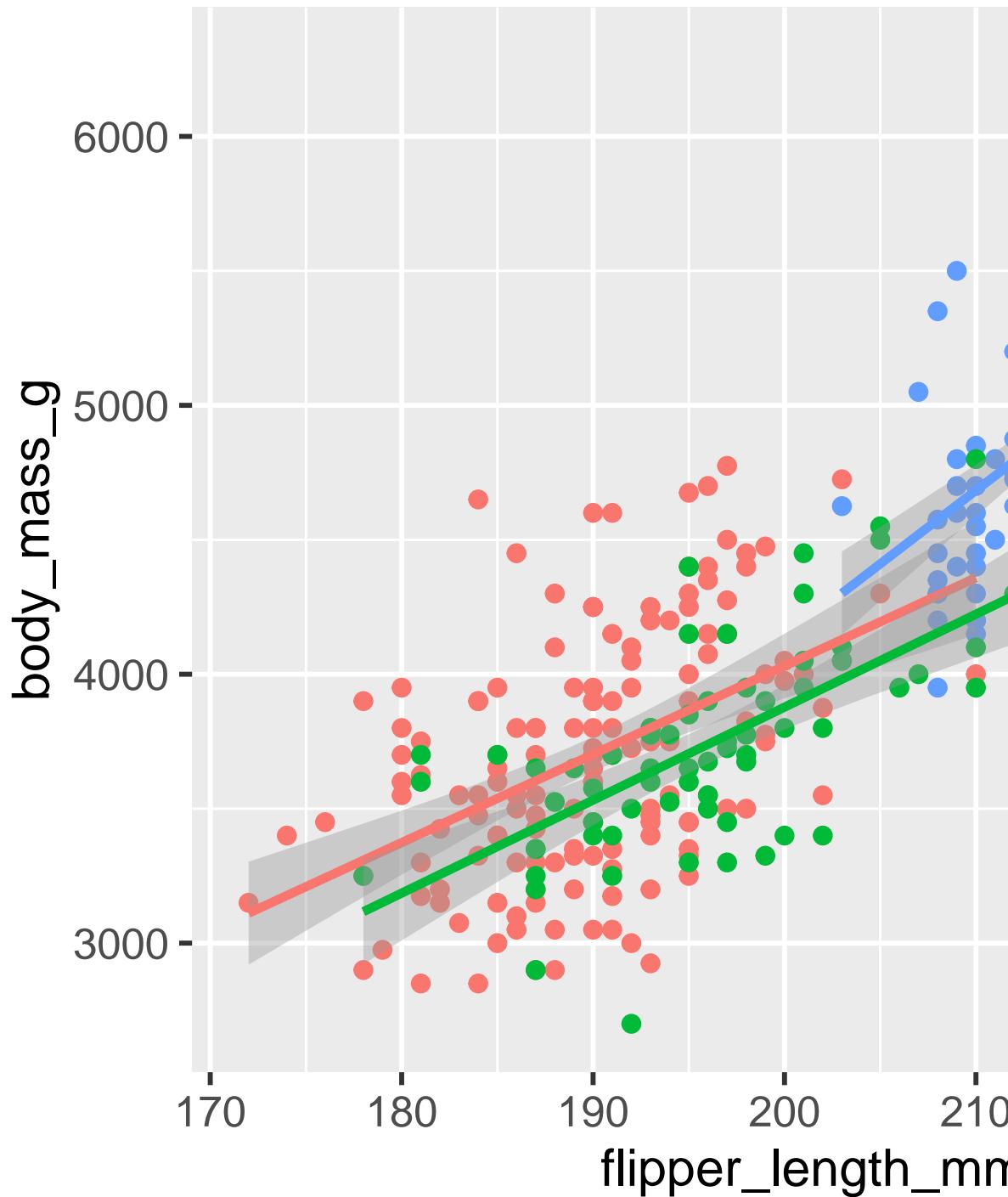
```
ggplot(  
  data = penguins,  
  mapping = aes(x = flipper_length_mm, y = body_mass_g, color = species)  
) +  
  geom_point()
```



```
ggplot2           scaling g gplot2
```

```
geom           geom   geom_smooth()    method = "lm"    linear  
model
```

```
ggplot(  
  data = penguins,  
  mapping = aes(x = flipper_length_mm, y = body_mass_g, color = species)  
) +  
  geom_point() +  
  geom_smooth(method = "lm")
```



??

```
ggplot()      aesthetic mappings      geom layers      ggplot2      -  
geom         mapping argument       points        species      lines      geom_point()  
= species
```

```
ggplot(  
  data = penguins,  
  mapping = aes(x = flipper_length_mm, y = body_mass_g)  
) +  
  geom_point(mapping = aes(color = species)) +  
  geom_smooth(method = "lm")
```



1.2.

33

!

```
ggplot(  
  data = penguins,  
  mapping = aes(x = flipper_length_mm, y = body_mass_g)  
) +  
  geom_point(mapping = aes(color = species, shape = species)) +  
  geom_smooth(method = "lm")
```

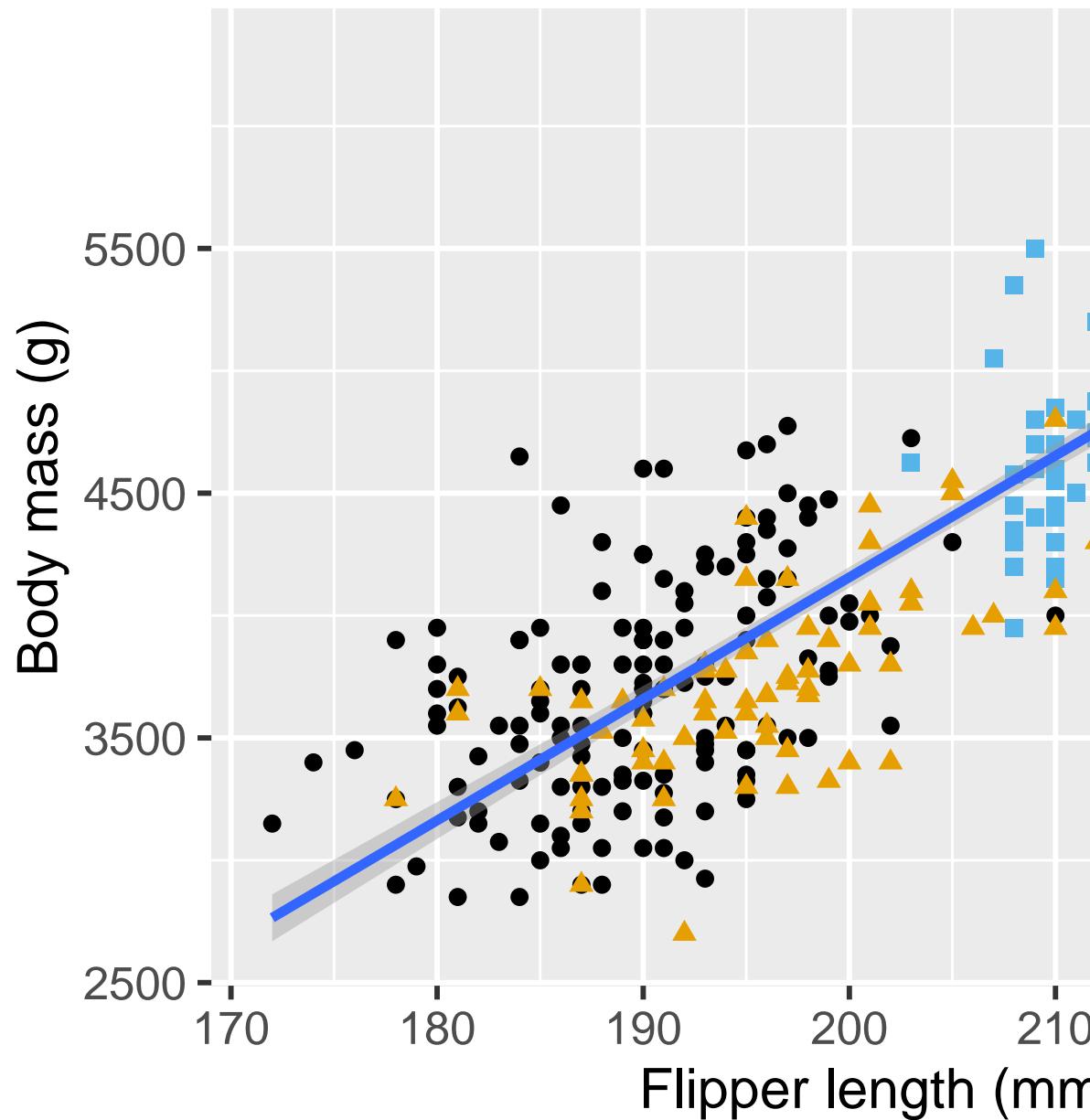


```
labs()      labs()      title   subtitle      x x   y y   color shape
ggthemes scale_colorblind()

ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point(aes(color = species, shape = species)) +
  geom_smooth(method = "lm") +
  labs(
    title = "Body mass and flipper length",
    subtitle = "Dimensions for Adelie, Chinstrap, and Gentoo Penguins",
    x = "Flipper length (mm)", y = "Body mass (g)",
    color = "Species", shape = "Species"
  ) +
  scale_color_colorblind()
```

Body mass and flipper length

Dimensions for Adelie, Chinstrap, and Gentoo penguins



“ ” !

1.2.5

1. penguins

2. penguins bill_depth_mm ?penguins

3. bill_depth_mm bill_length_mm y bill_depth_mm,
x bill_length_mm

4. species bill_depth_mm

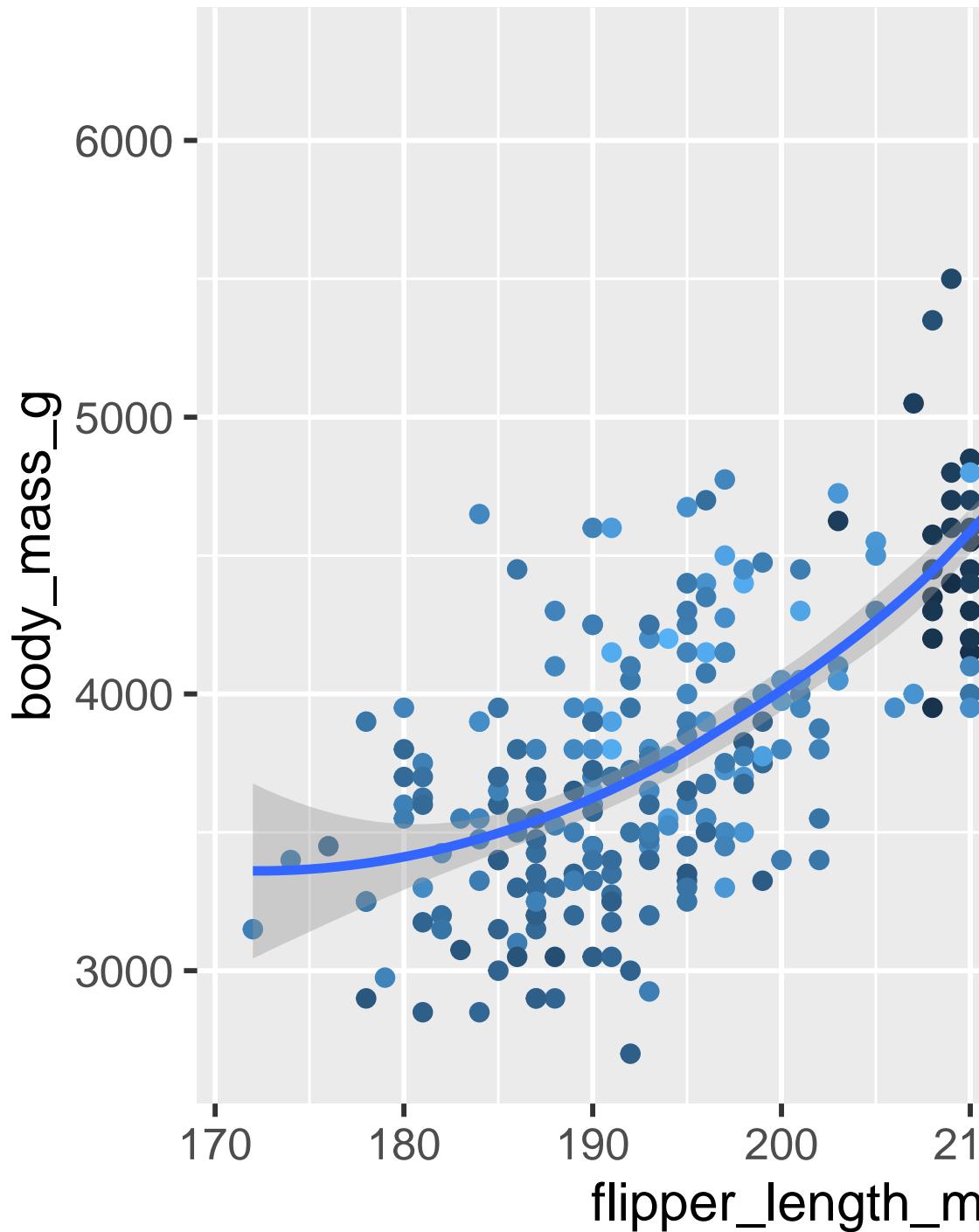
5. ?

```
ggplot(data = penguins) +  
  geom_point()
```

6. na.rm geom_point() TRUE

7. “ palmerpenguins ” labs()

8. bill_depth_mm



9. R

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g, color = island)
) +
  geom_point() +
  geom_smooth(se = FALSE)
```

10. /

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point() +
  geom_smooth()

ggplot() +
  geom_point(
    data = penguins,
    mapping = aes(x = flipper_length_mm, y = body_mass_g)
  ) +
  geom_smooth(
    data = penguins,
    mapping = aes(x = flipper_length_mm, y = body_mass_g)
  )
```

1.3 ggplot2

ggplot2

```
ggplot(
  data = penguins,
  mapping = aes(x = flipper_length_mm, y = body_mass_g)
) +
  geom_point()
```

```
ggplot()      data mapping
??
```

```
ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point()
```

```
|>          :
```

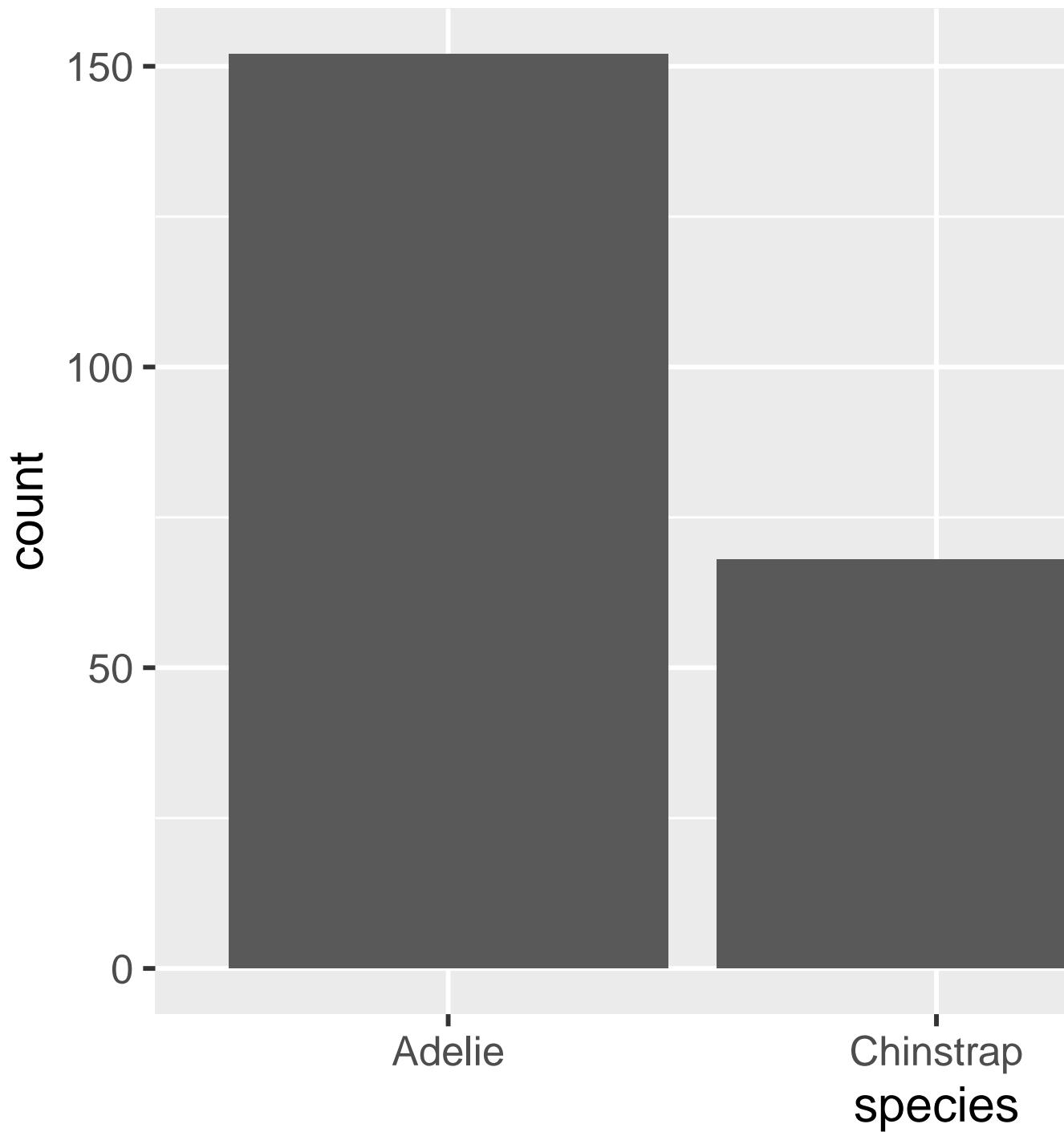
```
penguins |>  
  ggplot(aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point()
```

1.4

1.4.1

x

```
ggplot(penguins, aes(x = species)) +  
  geom_bar()
```

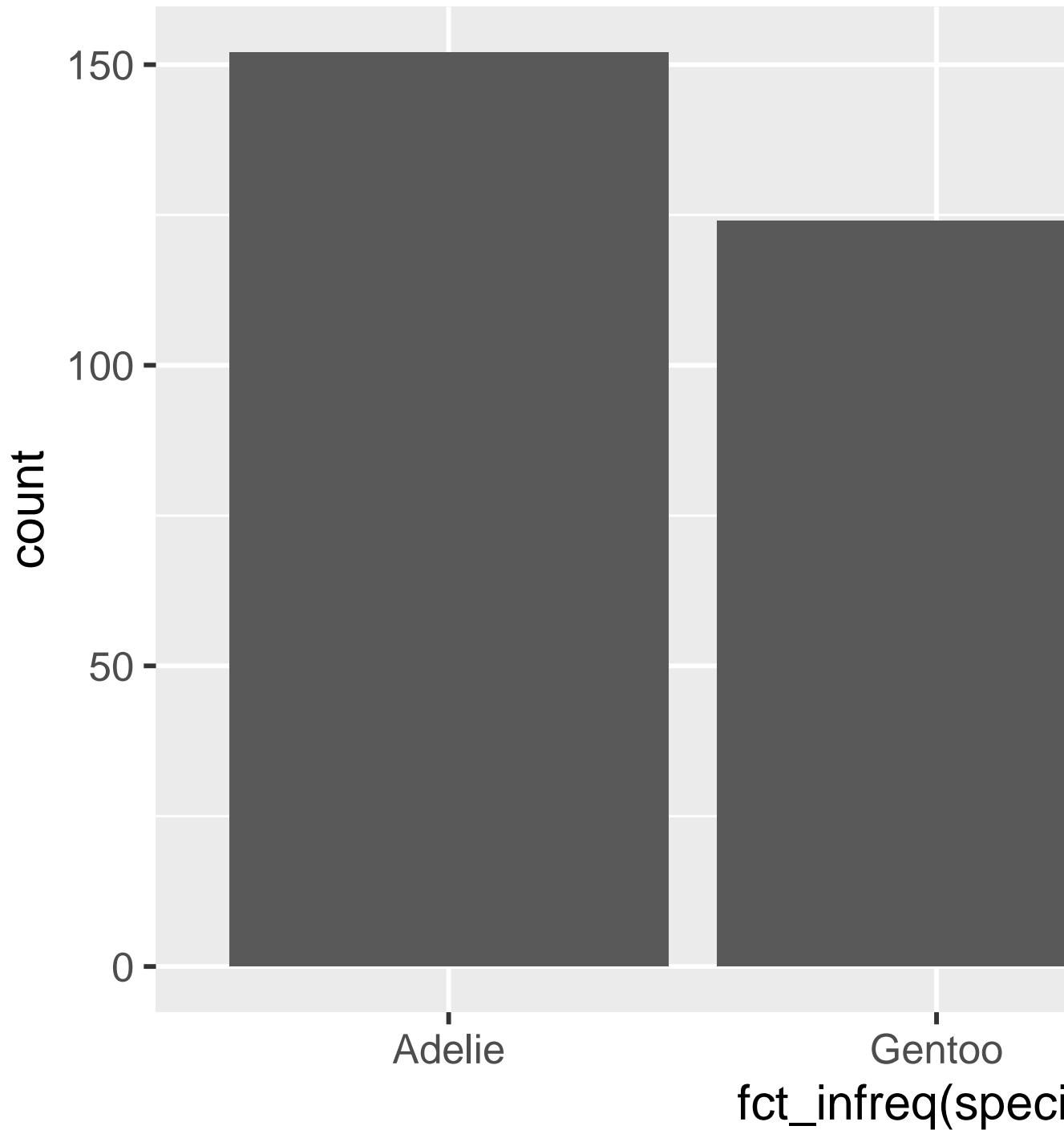


(R)

```
ggplot(penguins, aes(x = fct_infreq(species))) +  
  geom_bar()
```

1.4.

43



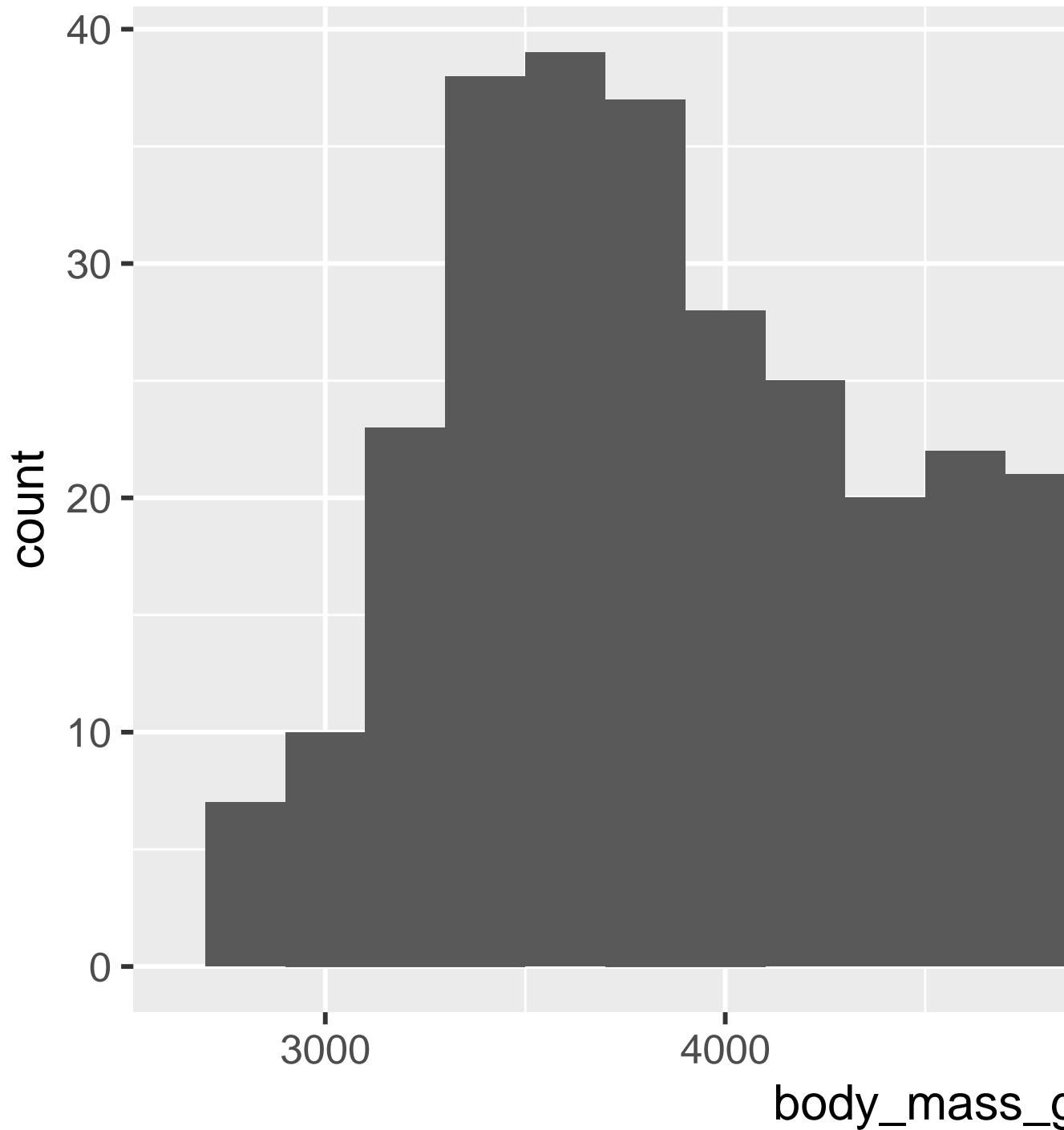
```
@sec-factors          fct_infreq()
```

1.4.2

```
ggplot(penguins, aes(x = body_mass_g)) +  
  geom_histogram(binwidth = 200)
```

1.4.

45



```
x      " "
39    body_mass_g 3,500 3,700
```

binwidth	x	20
2,000	200	

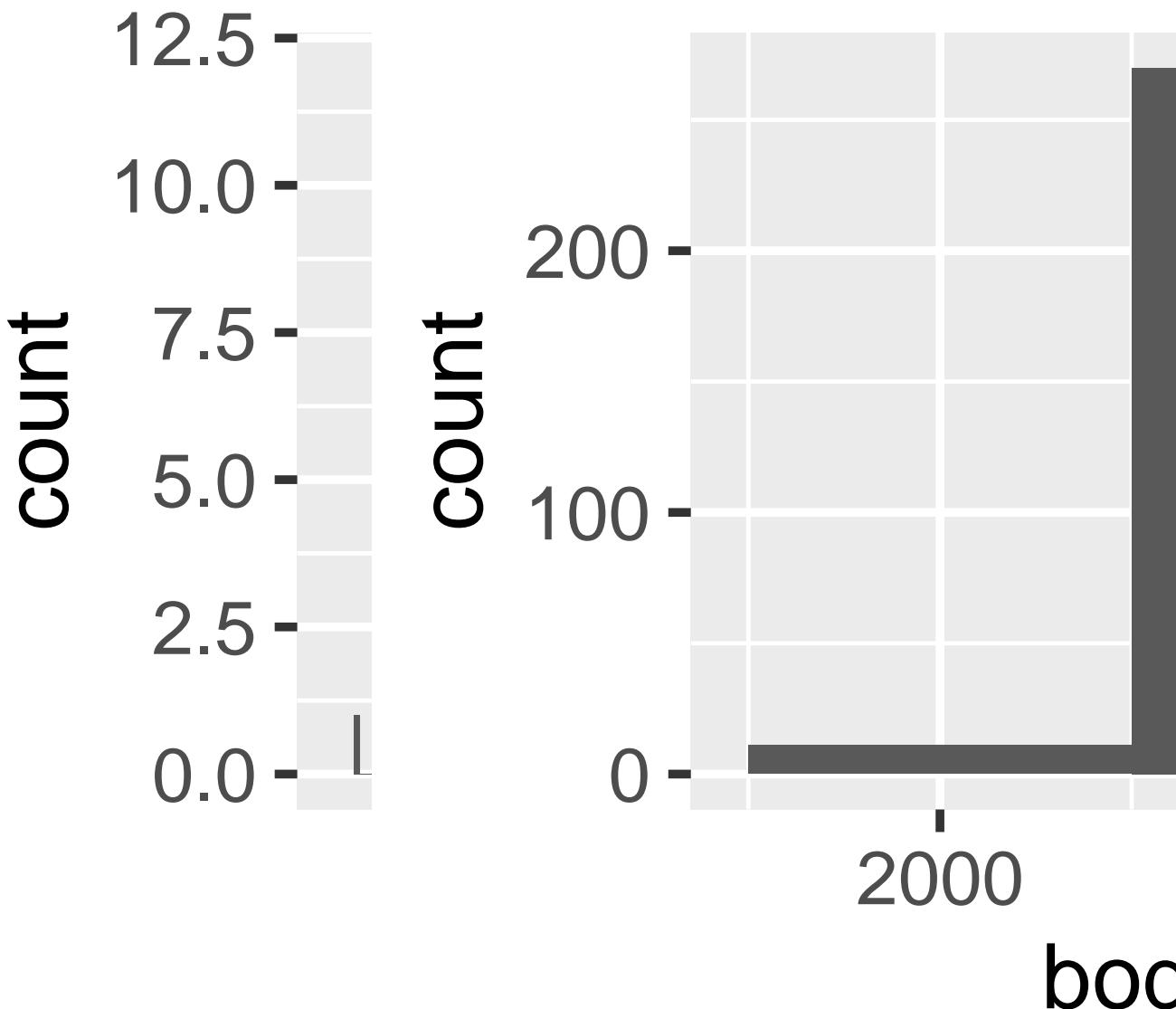
```
ggplot(penguins, aes(x = body_mass_g)) +
  geom_histogram(binwidth = 20)
ggplot(penguins, aes(x = body_mass_g)) +
  geom_histogram(binwidth = 2000)
```

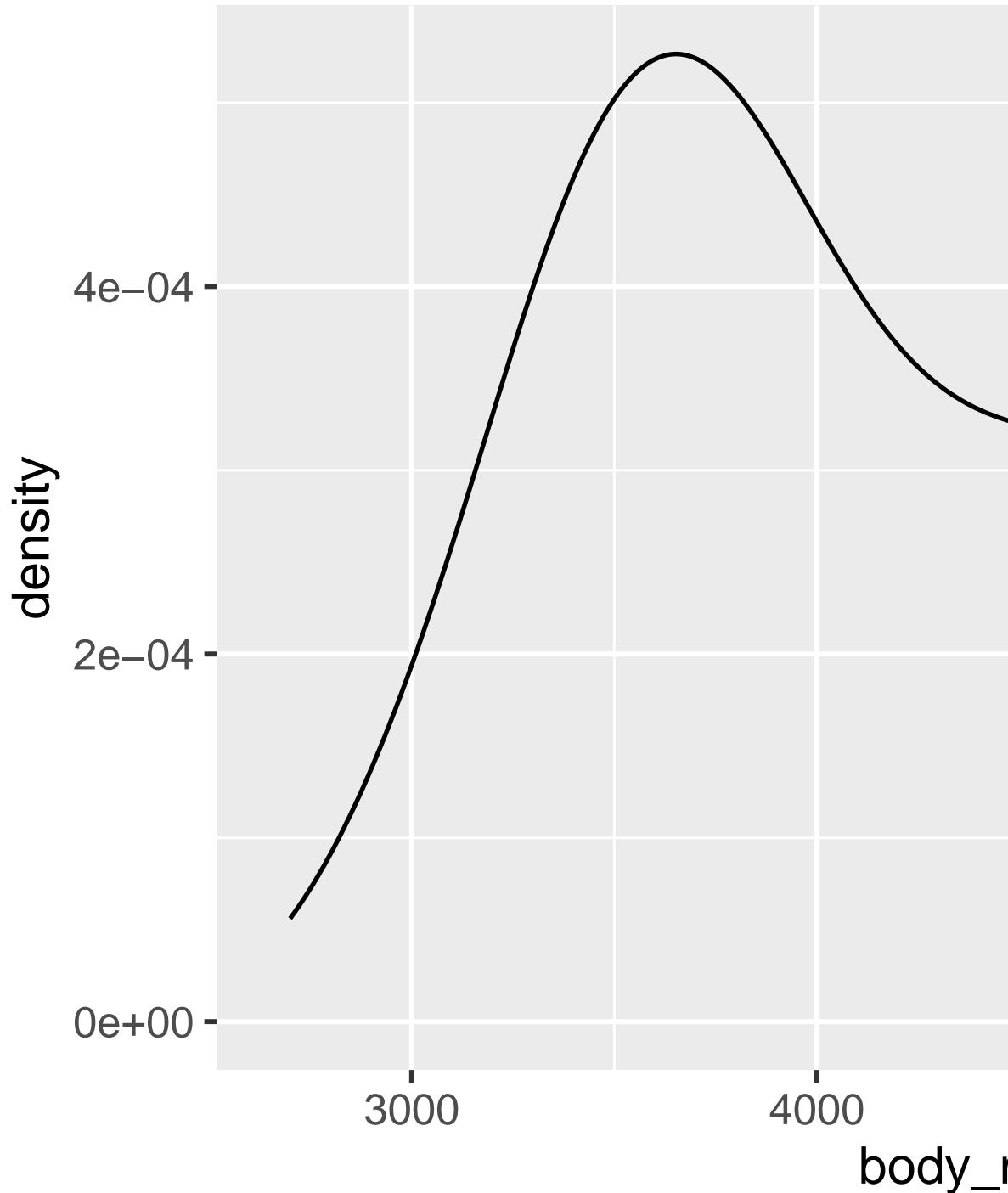
```
geom_density()
```

```
ggplot(penguins, aes(x = body_mass_g)) +
  geom_density()
#> Warning: Removed 2 rows containing non-finite outside the scale range
#> (`stat_density()`).
```

1.4.

47





1.4.3

1. y

2.

```
ggplot(penguins, aes(x = species)) +  
  geom_bar(color = "red")  
  
ggplot(penguins, aes(x = species)) +  
  geom_bar(fill = "red")
```

3. `geom_histogram()` `bins`

4. tidyverse diamonds carat

1.5

To visualize a relationship we need to have at least two variables mapped to aesthetics of a plot. In the following sections you will learn about commonly used plots for visualizing relationships between two or more variables and the geoms used for creating them.

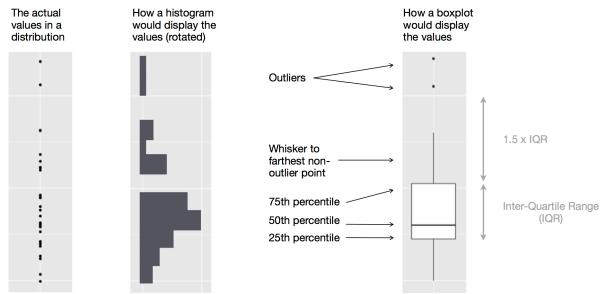
1.5.1

??

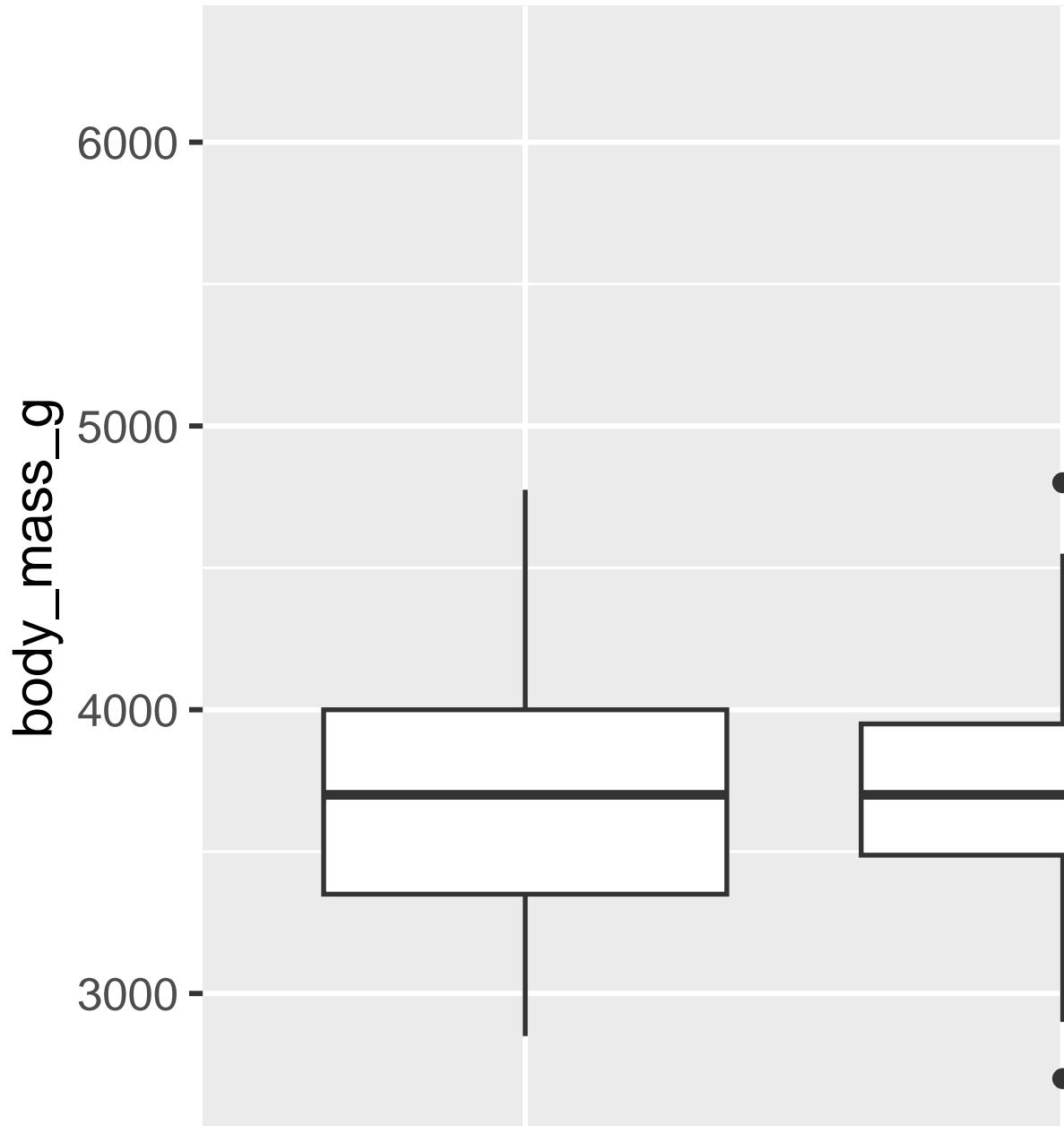
- 25% 75% **interquartile range IQR** 50%
 • 1.5 IQR
 • “ ”

`geom_boxplot()`

```
ggplot(penguins, aes(x = species, y = body_mass_g)) +  
  geom_boxplot()
```



1.1: Diagram depicting how a boxplot is created.

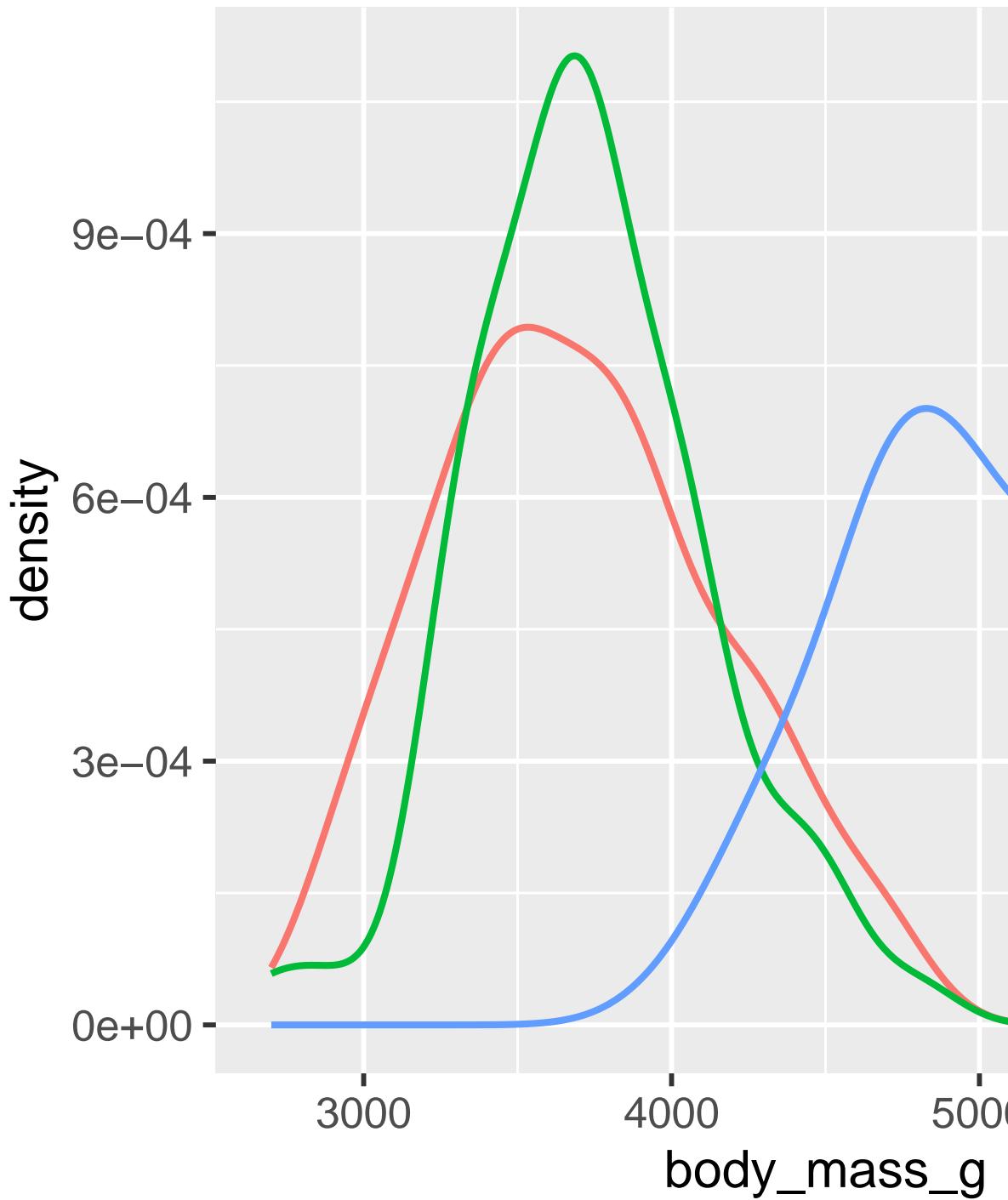


1.5.

51

```
geom_density()
```

```
ggplot(penguins, aes(x = body_mass_g, color = species)) +  
  geom_density(linewidth = 0.75)
```



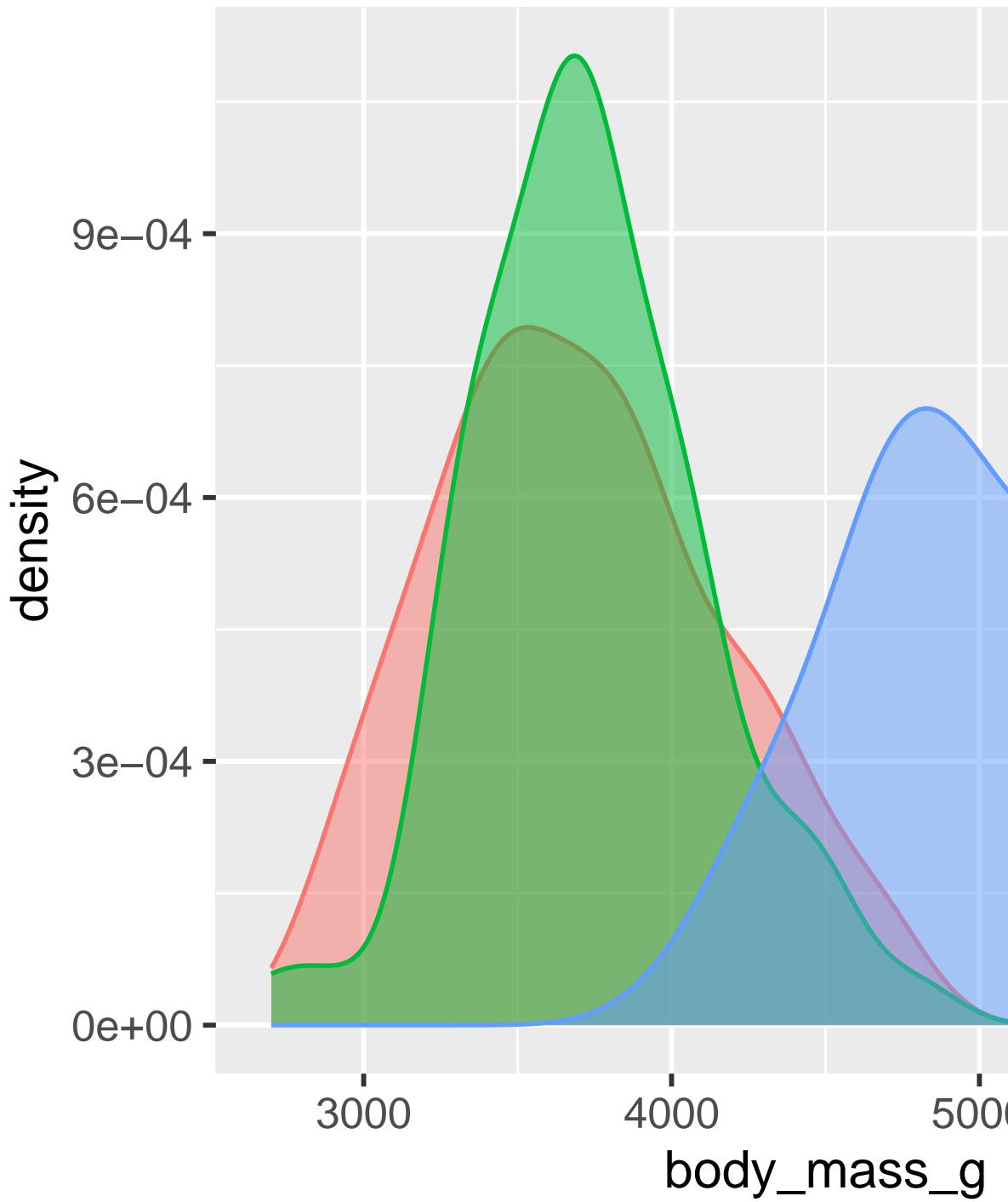
1.5.

53

linewidth

alpha 0 1 0.5

```
ggplot(penguins, aes(x = body_mass_g, color = species, fill = species)) +  
  geom_density(alpha = 0.5)
```



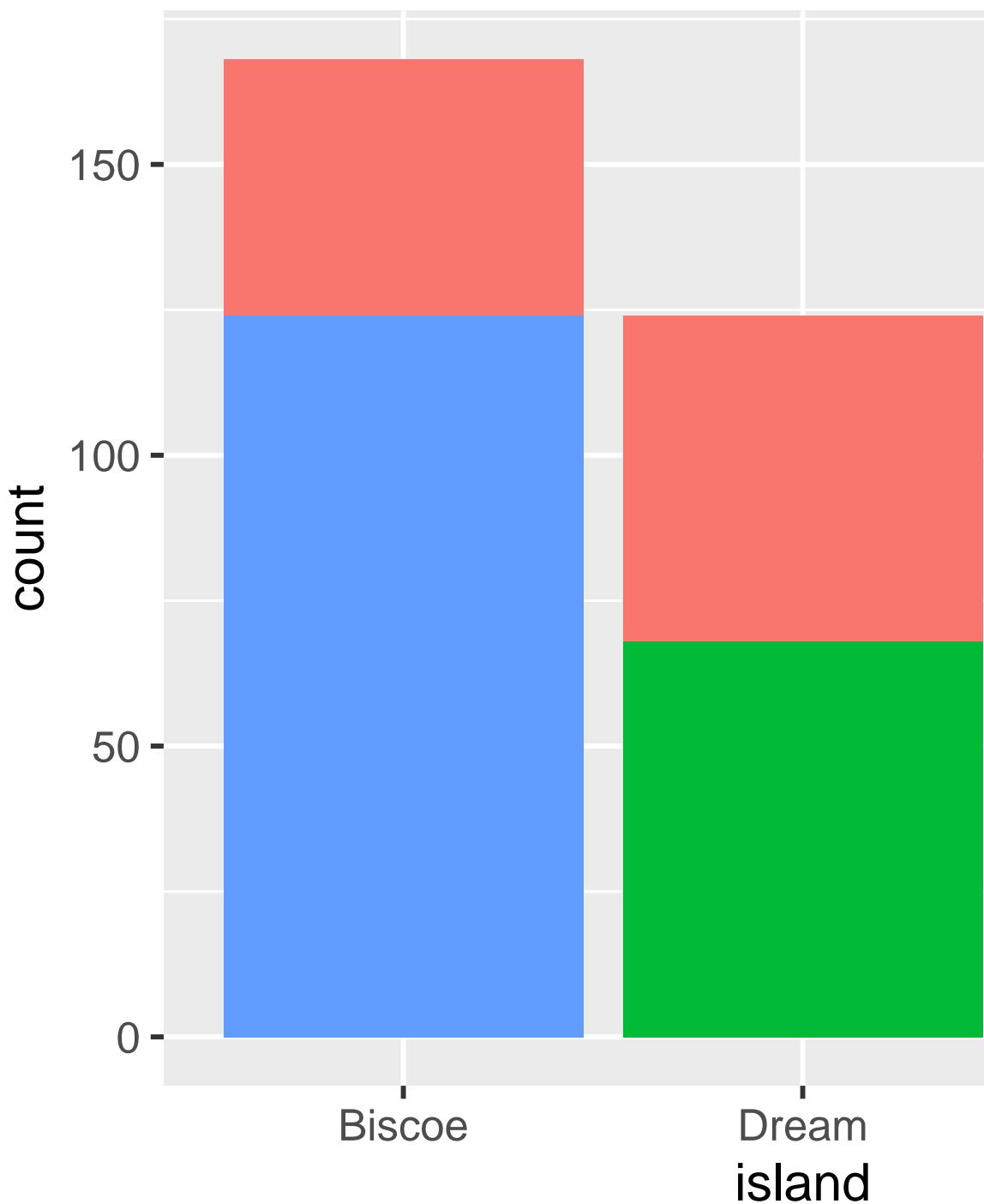
:

-
-

1.5.2

Adelies

```
ggplot(penguins, aes(x = island, fill = species)) +  
  geom_bar()
```



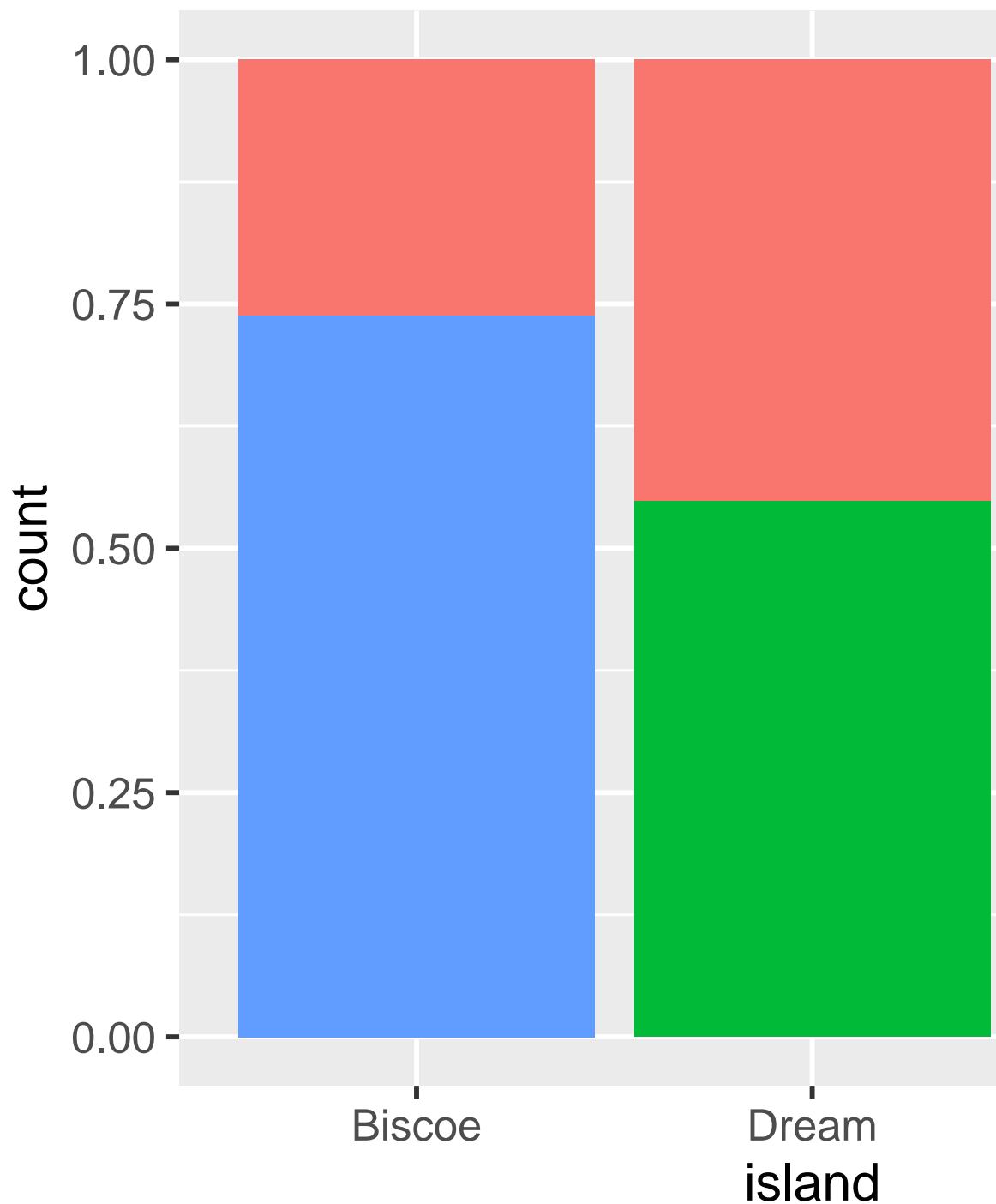
1.5.

57

```
geom position = "fill"
```

Gentoo	Biscoe	75% Chinstrap	Dr
--------	--------	---------------	----

```
ggplot(penguins, aes(x = island, fill = species)) +  
  geom_bar(position = "fill")
```



1.5.

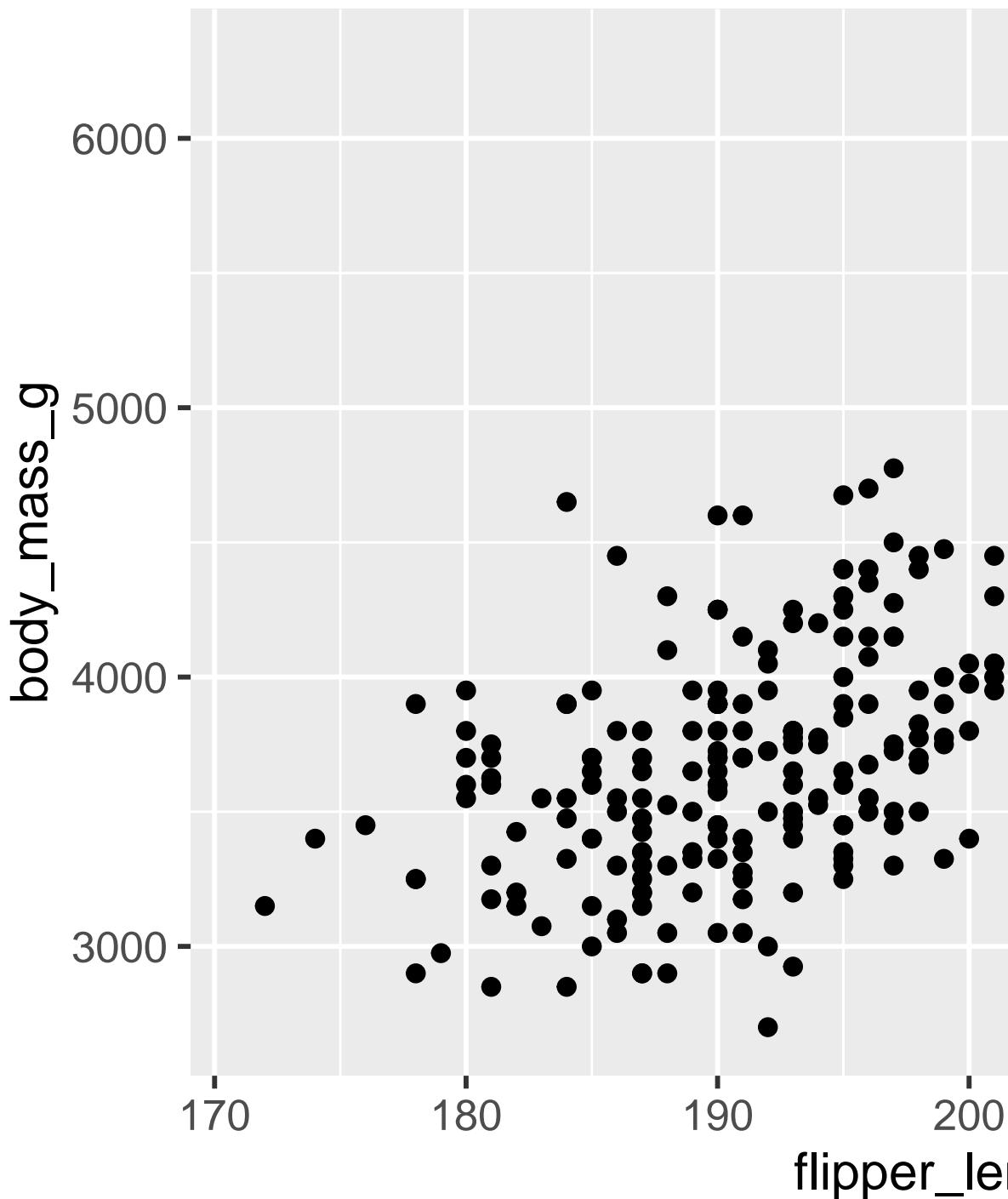
59

x fill

1.5.3

geom_point() geom_smooth()

```
ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point()
```



1.5.4

@sec-adding-aesthetics-layers

```
ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point(aes(color = species, shape = island))
```

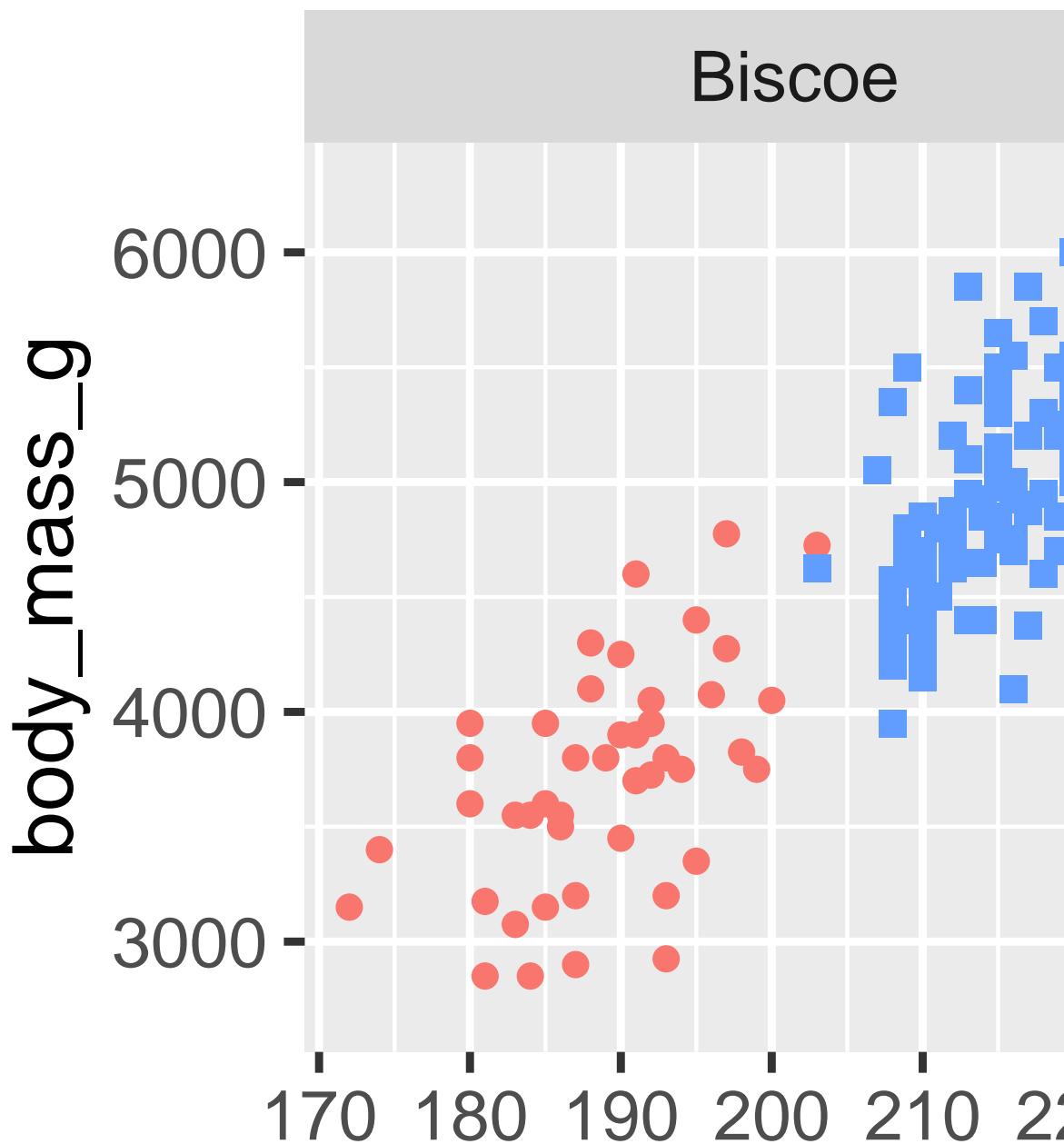


facets

`facet_wrap()` `facet_wrap()` formula³ ~ `facet_wrap()`

```
ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +  
  geom_point(aes(color = species, shape = species)) +  
  facet_wrap(~island)
```

³ “formula” ~ “equation”



```
@sec-layers           geoms
```

1.5.5

```
1. ggplot2   mpg      234     38    mpg      ?mpg      mpg
2. mpg     hwy displ      color    size    color size    shape
3. hwy displ      linewidth
4.
5. bill_depth_mm bill_length_mm      species
   species
6.

ggplot(
  data = penguins,
  mapping = aes(
    x = bill_length_mm, y = bill_depth_mm,
    color = species, shape = species
  )
) +
  geom_point() +
  labs(color = "Species")

7.

ggplot(penguins, aes(x = island, fill = species)) +
  geom_bar(position = "fill")
ggplot(penguins, aes(x = species, fill = island)) +
  geom_bar(position = "fill")
```

1.6

```
R           ggsave()
```

```
ggplot(penguins, aes(x = flipper_length_mm, y = body_mass_g)) +
  geom_point()
ggsave(filename = "penguin-plot.png")
```

```
,    @sec-workflow-scripts-projects
width height          ggsave()
Quarto      Quarto          @sec-quarto
Quarto
```

1.6.1

1. mpg-plot.png

```
ggplot(mpg, aes(x = class)) +
  geom_bar()
ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point()
ggsave("mpg-plot.png")
```

2. PDF PNG ggsave()

1.7

R

R

R
ESC

() "

+ R

ggplot2 +

```
ggplot(data = mpg)
+ geom_point(mapping = aes(x = displ, y = hwy))
```

?function_name R RStudio F1

R

Google

1.8

```
ggplot2      ggplot2
/      faceting
@sec-layers @sec-communication      ggplot2
R
```

Chapter 2

R

R

R

R

RStudio

2.1

R

```
1 / 200 * 30
#> [1] 0.15
(59 + 73 + 2) / 3
#> [1] 44.66667
sin(pi / 2)
#> [1] 1
```

<-:

```
x <- 3 * 4
```

x ,

x

c()

```
primes <- c(2, 3, 5, 7, 11, 13)
```

```
primes * 2
#> [1] 4 6 10 14 22 26
primes - 1
#> [1] 1 2 4 6 10 12
```

R :
object_name <- value
“ ”
<- RStudio Alt + - RStudio <-
giveyoureyesabreak

2.2

R # R

```
# create vector of primes
primes <- c(2, 3, 5, 7, 11, 13)

# multiply primes by 2
primes * 2
#> [1] 4 6 10 14 22 26
```

why	how	what
	geom_smooth()	span
0.9		span 0.75

2.3

- . snake_case -
i_use_snake_case
otherPeopleUseCamelCase
some.people.use.periods
And_aFew.People_RENOUNCEconvention

??

```
x
#> [1] 12
```

```
this_is_a_really_long_name <- 2.5
```

RStudio “this” TAB
 this_is_a_really_long_name 3.5 2.5 ↑
 “this” Cmd/Ctrl + ↑ Enter
 2.5 3.5

```
r_rocks <- 2^3
```

```
r_rock
#> Error: object 'r_rock' not found
R_rocks
#> Error: object 'R_rocks' not found
```

R R r_rock r_rocks R
 R_rocks r_rocks R

2.4

R :
 function_name(argument1 = value1, argument2 = value2, ...)

seq() RStudio se TAB q seq()
 ↑/↓ F1 () from 1 to 10
 TAB RStudio

```
seq(from = 1, to = 10)
#> [1] 1 2 3 4 5 6 7 8 9 10
```

:

```
seq(1, 10)
#> [1] 1 2 3 4 5 6 7 8 9 10
```

RStudio

```
x <- "hello world"
```

R Studio

R +

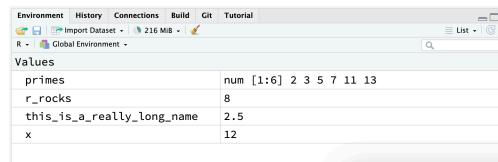
```
> x <- "hello
+ 
```

+ R

")

ESCAPE

Environment



2.5

1.

```
my_variable <- 10
my_variable
#> Error in eval(expr, envir, enclos): object 'my_variable' not found
```

2. R

```
library(tidyverse)

ggplot(dTA = mpg) +
  geom_point(mapping = aes(x = displ y = hwy)) +
  geom_smooth(method = "lm")
```

3. Option + Shift + K / Alt + Shift + K

4. ?? mpg-plot.png

```
my_bar_plot <- ggplot(mpg, aes(x = class)) +
  geom_bar()
my_scatter_plot <- ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point()
ggsave(filename = "mpg-plot.png", plot = my_bar_plot)
```

2.6

R

dplyr tidyverse

Chapter 3

3.1

dplyr 2013

pipe

3.1.1

dplyr tidyverse nycflights13 dplyr ggplot2

```
library(nycflights13)
library(tidyverse)
#> -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
#> v dplyr     1.1.4     v readr     2.1.5
#> v forcats   1.0.0     v stringr   1.5.1
#> v ggplot2   3.5.0     v tibble    3.2.1
#> v lubridate 1.9.3     v tidyr    1.3.1
#> v purrr    1.0.2
#> -- Conflicts ----- tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()   masks stats::lag()
#> i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

tidyverse dplyr R dplyr stats::filter()
stats::lag() R packagename::functionname()

3.1.2 nycflights13

```
dplyr      nycflights13::flights      r  format(nrow(nycflights13::flights),
big.mark = ",")2013          Bureau of Transportation Statistics    ?
lights
```

	flights	# A tibble: 336,776 x 19						
		year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time						
		<int> <int> <int> <int> <int> <dbl> <int> <int>						
#> 1	2013	1	1	517	515	2	830	819
#> 2	2013	1	1	533	529	4	850	830
#> 3	2013	1	1	542	540	2	923	850
#> 4	2013	1	1	544	545	-1	1004	1022
#> 5	2013	1	1	554	600	-6	812	837
#> 6	2013	1	1	554	558	-4	740	728
		# i 336,770 more rows						
		# i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...						

flights tibble tibble tidyverse tibble tibble
RStudio View(flights) print(flights, width
= Inf) glimpse()

```
glimpse(flights)
#> Rows: 336,776
#> Columns: 19
#> $ year      <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013~
#> $ month     <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
#> $ day       <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
#> $ dep_time   <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 55~
#> $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 60~
#> $ dep_delay   <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -2~
#> $ arr_time    <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 8~
#> $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 8~
#> $ arr_delay   <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7, ~
#> $ carrier     <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6"~
#> $ flight       <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301~
#> $ tailnum     <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N~
#> $ origin       <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LG~
#> $ dest         <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IA~
#> $ air_time     <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149~
#> $ distance     <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 73~
#> $ hour          <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6~
#> $ minute        <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59~
#> $ time_hour    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-0~
```

<int>	<dbl>	<chr>	<dttm>
“ ”			

3.1.3 dplyr

dplyr

- 1.
- 2.
- 3.

```
|> f(y)  f(x, y)  x |> f(y) |> g(z)  g(f(x, y), z) |>    “then”      x
flights |>
  filter(dest == "IAH") |>
  group_by(year, month, day) |>
  summarize(
    arr_delay = mean(arr_delay, na.rm = TRUE)
  )
```

dplyr	rows	columns	groups	tables	@sec-joins
!					

3.2

filter()	arrange()	dis-
tinct()	arrange() filter()	

3.2.1 filter()

Filter() 1 120 (2) :

```
flights |>
  filter(dep_delay > 120)
#> # A tibble: 9,723 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>     <int>       <int>     <dbl>   <int>       <int>
#> 1  2013     1     1      848        1835      853     1001       1950
#> 2  2013     1     1      957        733       144     1056       853

```

1 slice_*

```
#> 3 2013 1 1 1114 900 134 1447 1222
#> 4 2013 1 1 1540 1338 122 2020 1825
#> 5 2013 1 1 1815 1325 290 2120 1542
#> 6 2013 1 1 1842 1422 260 1958 1535
#> # i 9,717 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

```
| >      >=      <    <=      ==      !=      & ,      “ ”
| “ ”
```

```
# Flights that departed on January 1
flights |>
  filter(month == 1 & day == 1)
#> # A tibble: 842 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>    <int>        <int>     <dbl>    <int>        <int>
#> 1 2013     1     1      517        515       2     830        819
#> 2 2013     1     1      533        529       4     850        830
#> 3 2013     1     1      542        540       2     923        850
#> 4 2013     1     1      544        545      -1    1004       1022
#> 5 2013     1     1      554        600      -6     812        837
#> 6 2013     1     1      554        558      -4     740        728
#> # i 836 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

Flights that departed in January or February

```
flights |>
  filter(month == 1 | month == 2)
#> # A tibble: 51,955 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>    <int>        <int>     <dbl>    <int>        <int>
#> 1 2013     1     1      517        515       2     830        819
#> 2 2013     1     1      533        529       4     850        830
#> 3 2013     1     1      542        540       2     923        850
#> 4 2013     1     1      544        545      -1    1004       1022
#> 5 2013     1     1      554        600      -6     812        837
#> 6 2013     1     1      554        558      -4     740        728
#> # i 51,949 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

```
| ==      %in%
```

```
# A shorter way to select flights that departed in January or February
flights |>
```

```

filter(month %in% c(1, 2))
#> # A tibble: 51,955 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>     <int>        <dbl>    <int>        <int>
#> 1 2013     1     1      517         515       2     830        819
#> 2 2013     1     1      533         529       4     850        830
#> 3 2013     1     1      542         540       2     923        850
#> 4 2013     1     1      544         545      -1    1004       1022
#> 5 2013     1     1      554         600      -6     812        837
#> 6 2013     1     1      554         558      -4     740        728
#> # i 51,949 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...

```

@sec-logicals

filter() dplyr flights dplyr <-

```

jan1 <- flights |>
  filter(month == 1 & day == 1)

```

3.2.2

R = == filter()

```

flights |>
  filter(month = 1)
#> Error in `filter()`:
#> ! We detected a named input.
#> i This usually means that you've used `=` instead of `==`.
#> i Did you mean `month == 1`?

```

“ ” :

```

flights |>
  filter(month == 1 | 2)

```

“ ” | month == 1 2 2 ??

3.2.3 arrange()

arrange()

```

flights |>
  arrange(year, month, day, dep_time)
#> # A tibble: 336,776 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
#> 1  2013     1     1      517            515       2     830          819
#> 2  2013     1     1      533            529       4     850          830
#> 3  2013     1     1      542            540       2     923          850
#> 4  2013     1     1      544            545      -1    1004         1022
#> 5  2013     1     1      554            600      -6     812          837
#> 6  2013     1     1      554            558      -4     740          728
#> # i 336,770 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...

```

arrange() desc()

```

flights |>
  arrange(desc(dep_delay))
#> # A tibble: 336,776 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
#> 1  2013     1     9      641            900      1301     1242        1530
#> 2  2013     6    15     1432           1935      1137     1607        2120
#> 3  2013     1    10     1121           1635      1126     1239        1810
#> 4  2013     9    20     1139           1845      1014     1457        2210
#> 5  2013     7    22      845            1600      1005     1044        1815
#> 6  2013     4    10     1100           1900      960      1342        2211
#> # i 336,770 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...

```

3.2.4 distinct()

distinct()

```

# Remove duplicate rows, if any
flights |>
  distinct()
#> # A tibble: 336,776 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
#> 1  2013     1     1      517            515       2     830          819

```

```

#> 2 2013 1 1 533 529 4 850 830
#> 3 2013 1 1 542 540 2 923 850
#> 4 2013 1 1 544 545 -1 1004 1022
#> 5 2013 1 1 554 600 -6 812 837
#> 6 2013 1 1 554 558 -4 740 728
#> # i 336,770 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...

# Find all unique origin and destination pairs
flights |>
  distinct(origin, dest)
#> # A tibble: 224 x 2
#>   origin dest
#>   <chr>  <chr>
#> 1 EWR    IAH
#> 2 LGA    IAH
#> 3 JFK    MIA
#> 4 JFK    BQN
#> 5 LGA    ATL
#> 6 EWR    ORD
#> # i 218 more rows

.flip .keep_all = TRUE

flights |>
  distinct(origin, dest, .keep_all = TRUE)
#> # A tibble: 224 x 19
#>   year month day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int> <int> <int> <dbl> <int> <int>
#> 1 2013 1 1 517 515 2 830 819
#> 2 2013 1 1 533 529 4 850 830
#> 3 2013 1 1 542 540 2 923 850
#> 4 2013 1 1 544 545 -1 1004 1022
#> 5 2013 1 1 554 600 -6 812 837
#> 6 2013 1 1 554 558 -4 740 728
#> # i 218 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...

1 1   distinct()

distinct()  count()  sort = TRUE          ??      count

flights |>
  count(origin, dest, sort = TRUE)
#> # A tibble: 224 x 3

```

```
#>   origin dest    n
#>   <chr>  <chr> <int>
#> 1 JFK    LAX   11262
#> 2 LGA    ATL   10263
#> 3 LGA    ORD   8857
#> 4 JFK    SFO   8204
#> 5 LGA    CLT   6168
#> 6 EWR    ORD   6100
#> # i 218 more rows
```

3.2.5

- 1.
 -
 - IAH HOU
 - United American , Delta
 -
 - 30
 2. flights
 3. flights :
 4. 2013
 - 5.
 6. filter() arrange()

3.3

`mutate()` `select()` `rename()` `relocate()`

3.3.1 mutate()

`mutate()` gain

```
flights |>  
  mutate(  
    gain = dep_delay - arr_delay,  
    speed = distance / air_time * 60
```

```

)
#> # A tibble: 336,776 x 21
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>     <int>           <int>     <dbl>     <int>           <int>
#> 1 2013     1     1      517            515       2     830           819
#> 2 2013     1     1      533            529       4     850           830
#> 3 2013     1     1      542            540       2     923           850
#> 4 2013     1     1      544            545      -1    1004          1022
#> 5 2013     1     1      554            600      -6     812           837
#> 6 2013     1     1      554            558      -4     740           728
#> # i 336,770 more rows
#> # i 13 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...

```

`mutate()` .before

By default, `mutate()` adds new columns on the right hand side of your dataset, which makes it difficult to see what's happening here. We can use the `.before` argument to instead add the variables to the left hand side²:

```

flights |>
  mutate(
    gain = dep_delay - arr_delay,
    speed = distance / air_time * 60,
    .before = 1
  )
#> # A tibble: 336,776 x 21
#>   gain speed year month   day dep_time sched_dep_time dep_delay arr_time
#>   <dbl> <dbl> <int> <int> <int>           <int>     <dbl>     <int>
#> 1   -9   370. 2013     1     1      517            515       2     830
#> 2  -16   374. 2013     1     1      533            529       4     850
#> 3  -31   408. 2013     1     1      542            540       2     923
#> 4   17   517. 2013     1     1      544            545      -1    1004
#> 5   19   394. 2013     1     1      554            600      -6     812
#> 6  -16   288. 2013     1     1      554            558      -4     740
#> # i 336,770 more rows
#> # i 12 more variables: sched_arr_time <int>, arr_delay <dbl>, ...

```

. .before day	.after	.before .after
------------------	--------	----------------

```

flights |>
  mutate(
    gain = dep_delay - arr_delay,

```

²Remember that in RStudio, the easiest way to see a dataset with many columns is `View()`.

```

    speed = distance / air_time * 60,
    .after = day
)
  .keep      "used"      mutate()
  dep_delay arr_delay air_time gain hours
flights |>
  mutate(
    gain = dep_delay - arr_delay,
    hours = air_time / 60,
    gain_per_hour = gain / hours,
    .keep = "used"
)
  flights      gain hours gain_per_hour
  delay_gain      flights

```

3.3.2 select()

s elect()

- :

```

flights |>
  select(year, month, day)

```

- year day :

```

flights |>
  select(year:day)

```

- year day

```

flights |>
  select(!year:day)

```

- ! - ! “ ” & |

-

```

flights |>
  select(where(is.character))

```

```
select()
```

- starts_with("abc"): "abc"
- ends_with("xyz"): "xyz"
- contains("ijk"): "ijk"
- num_range("x", 1:3): x1 x2 x3

?s elect	??	matches()
select()	=	=

```
flights |>
  select(tail_num = tailnum)
#> # A tibble: 336,776 x 1
#>   tail_num
#>   <chr>
#> 1 N14228
#> 2 N24211
#> 3 N619AA
#> 4 N804JB
#> 5 N668DN
#> 6 N39463
#> # i 336,770 more rows
```

3.3.3 rename()

```
rename() select():
```

```
flights |>
  rename(tail_num = tailnum)
#> # A tibble: 336,776 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>     <int>        <int>     <dbl>    <int>        <int>
#> 1 2013     1     1      517          515       2     830         819
#> 2 2013     1     1      533          529       4     850         830
#> 3 2013     1     1      542          540       2     923         850
#> 4 2013     1     1      544          545      -1    1004        1022
#> 5 2013     1     1      554          600      -6     812         837
#> 6 2013     1     1      554          558      -4     740         728
#> # i 336,770 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

```
janitor::clean_names()
```

3.3.4 relocate()

```

relocate()          relocate()      :

flights |>
  relocate(time_hour, air_time)
#> # A tibble: 336,776 x 19
#>   time_hour           air_time   year month   day dep_time sched_dep_time
#>   <dttm>             <dbl> <int> <int> <int>    <int>        <int>
#> 1 2013-01-01 05:00:00     227  2013     1     1      517        515
#> 2 2013-01-01 05:00:00     227  2013     1     1      533        529
#> 3 2013-01-01 05:00:00     160  2013     1     1      542        540
#> 4 2013-01-01 05:00:00     183  2013     1     1      544        545
#> 5 2013-01-01 06:00:00     116  2013     1     1      554        600
#> 6 2013-01-01 05:00:00     150  2013     1     1      554        558
#> # i 336,770 more rows
#> # i 12 more variables: dep_delay <dbl>, arr_time <int>, ...

```

```

.before .after      mutate() :

flights |>
  relocate(year:dep_time, .after = time_hour)
flights |>
  relocate(starts_with("arr"), .before = dep_time)

```

3.3.5

1. dep_time sched_dep_time dep_delay
2. flights dep_time dep_delay arr_time arr_delay
3. select()
4. any_of() ?

```
variables <- c("year", "month", "day", "dep_delay", "arr_delay")
```

5. select()

```
flights |> select(contains("TIME"))
```

6. air_time air_time_min
7. error ?

```

flights |>
  select(tailnum) |>
  arrange(arr_delay)
#> Error in `arrange()`:
#> i In argument: `..1 = arr_delay`.
#> Caused by error:
#> ! object 'arr_delay' not found

```

3.4

IAH filter() mutate() select() arrange():

```

flights |>
  filter(dest == "IAH") |>
  mutate(speed = distance / air_time * 60) |>
  select(year:day, dep_time, carrier, flight, speed) |>
  arrange(desc(speed))
#> # A tibble: 7,198 x 7
#>   year month   day dep_time carrier flight   speed
#>   <int> <int> <int>    <int> <chr>   <int> <dbl>
#> 1  2013     7     9      707  UA       226   522.
#> 2  2013     8     27     1850  UA      1128   521.
#> 3  2013     8     28      902  UA      1711   519.
#> 4  2013     8     28     2122  UA      1022   519.
#> 5  2013     6     11     1628  UA      1178   515.
#> 6  2013     8     27     1017  UA       333   515.
#> # i 7,192 more rows

```

flights

```

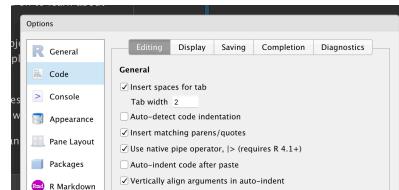
arrange(
  select(
    mutate(
      filter(
        flights,
        dest == "IAH"
      ),
      speed = distance / air_time * 60
    ),
    year:day, dep_time, carrier, flight, speed
  ),

```

```
desc(speed)
)
```

```
flights1 <- filter(flights, dest == "IAH")
flights2 <- mutate(flights1, speed = distance / air_time * 60)
flights3 <- select(flights2, year:day, dep_time, carrier, flight, speed)
arrange(flights3, desc(speed))
```

Ctrl/Cmd + Shift + M |> %>% RStudio ??
%>%



3.1: To insert |>, make sure the “Use native pipe operator” option is checked.

3.5 Magrittr

tidyverse magrittr %>% m agritr tidyverse tidyverse %>%

```
library(tidyverse)

mtcars %>%
  group_by(cyl) %>%
  summarize(n = n())
```

2014 |> %>% |> 2021 |> R 4.1.0 |> tidyverse |> %>% %>%

3.6

dplyr group_by() summarize() slice

3.6.1 group_by()

```
group_by()          :  
  
flights |>  
  group_by(month)  
#> # A tibble: 336,776 x 19  
#> # Groups:   month [12]  
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time  
#>   <int> <int> <int>   <int>           <int>     <dbl>   <int>           <int>  
#> 1 2013     1     1      517            515       2     830           819  
#> 2 2013     1     1      533            529       4     850           830  
#> 3 2013     1     1      542            540       2     923           850  
#> 4 2013     1     1      544            545      -1    1004          1022  
#> 5 2013     1     1      554            600      -6     812           837  
#> 6 2013     1     1      554            558      -4     740           728  
#> # i 336,770 more rows  
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

group_by() “ ” Groups: month [12] “ ” g
roup_by()

3.6.2 summarize()

```
dplyr      summarize()3  
  
flights |>  
  group_by(month) |>  
  summarize(  
    avg_delay = mean(dep_delay)  
  )  
#> # A tibble: 12 x 2  
#>   month avg_delay  
#>   <int>     <dbl>  
#> 1     1       NA  
#> 2     2       NA  
#> 3     3       NA  
#> 4     4       NA  
#> 5     5       NA  
#> 6     6       NA  
#> # i 6 more rows
```

NA	“N-A”	R	NA
??	mean()	na.rm	TRUE

³ `summarise()`

```

flights |>
  group_by(month) |>
  summarize(
    avg_delay = mean(dep_delay, na.rm = TRUE)
  )
#> # A tibble: 12 x 2
#>   month avg_delay
#>   <int>     <dbl>
#> 1     1      10.0
#> 2     2      10.8
#> 3     3      13.2
#> 4     4      13.9
#> 5     5      13.0
#> 6     6      20.8
#> # i 6 more rows

```

<pre> summarize()</pre>	<pre>n()</pre>
-------------------------	----------------

```

flights |>
  group_by(month) |>
  summarize(
    avg_delay = mean(dep_delay, na.rm = TRUE),
    n = n()
  )
#> # A tibble: 12 x 3
#>   month avg_delay     n
#>   <int>     <dbl> <int>
#> 1     1      10.0 27004
#> 2     2      10.8 24951
#> 3     3      13.2 28834
#> 4     4      13.9 28330
#> 5     5      13.0 28796
#> 6     6      20.8 28243
#> # i 6 more rows

```

Means counts !

3.6.3 slice_

:

- df |> slice_head(n = 1) t
- df |> slice_tail(n = 1)
- df |> slice_min(x, n = 1) x

- df |> slice_max(x, n = 1) x
- df |> slice_sample(n = 1)

n n= prop = 0.1 10%

```
flights |>
  group_by(dest) |>
  slice_max(arr_delay, n = 1) |>
  relocate(dest)
#> # A tibble: 108 x 19
#> # Groups:   dest [105]
#>   dest    year month   day dep_time sched_dep_time dep_delay arr_time
#>   <chr> <int> <int> <int>     <int>      <dbl>     <int>
#> 1 ABQ    2013     7    22    2145      2007      98     132
#> 2 ACK    2013     7    23    1139      800      219    1250
#> 3 ALB    2013     1    25    123       2000      323     229
#> 4 ANC    2013     8    17    1740      1625      75    2042
#> 5 ATL    2013     7    22    2257      759      898     121
#> 6 AUS    2013     7    10    2056      1505      351    2347
#> # i 102 more rows
#> # i 11 more variables: sched_arr_time <int>, arr_delay <dbl>, ...
```

105 108 slice_min() slice_max() n = 1
 with_ties = FALSE
 summarize()

3.6.4

You can create groups using more than one variable. For example, we could make a group for each date.

```
daily <- flights |>
  group_by(year, month, day)
daily
#> # A tibble: 336,776 x 19
#> # Groups:   year, month, day [365]
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>     <int>      <dbl>     <int>      <int>
#> 1 2013     1     1     517      515      2     830      819
#> 2 2013     1     1     533      529      4     850      830
#> 3 2013     1     1     542      540      2     923      850
#> 4 2013     1     1     544      545     -1    1004     1022
```

```
#> 5 2013 1 1 554 600 -6 812 837
#> 6 2013 1 1 554 558 -4 740 728
#> # i 336,770 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

tibble

dplyr

```
daily_flights <- daily |>
  summarise(n = n())
#> `summarise()` has grouped output by 'year', 'month'. You can override using
#> the `.`groups` argument.
```

:

```
daily_flights <- daily |>
  summarise(
    n = n(),
    .groups = "drop_last"
  )
```

“drop” “keep”

3.6.5

summarize() ungroup()

```
daily |>
  ungroup()
#> # A tibble: 336,776 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>     <int>        <int>     <dbl>     <int>        <int>
#> 1 2013     1     1      517        515       2     830        819
#> 2 2013     1     1      533        529       4     850        830
#> 3 2013     1     1      542        540       2     923        850
#> 4 2013     1     1      544        545      -1    1004       1022
#> 5 2013     1     1      554        600      -6     812        837
#> 6 2013     1     1      554        558      -4     740        728
#> # i 336,770 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

```
daily |>
  ungroup() |>
  summarize(
    avg_delay = mean(dep_delay, na.rm = TRUE),
    flights = n()
  )
#> # A tibble: 1 x 2
#>   avg_delay flights
#>       <dbl>     <int>
#> 1      12.6     336776
```

dplyr

3.6.6 .by

dplyr 1.1.0 .by group_by() ungroup() .by

```
flights |>
  summarize(
    delay = mean(dep_delay, na.rm = TRUE),
    n = n(),
    .by = month
  )
```

```
flights |>
  summarize(
    delay = mean(dep_delay, na.rm = TRUE),
    n = n(),
    .by = c(origin, dest)
  )
```

.by .groups ungroup()
dplyr 1.1.0

3.6.7

1. / : flights |> summarize(n())
2.

3.

4. slice_min() n
 5. dplyr count() count() sort
 6. :

```
df <- tibble(
  x = 1:5,
  y = c("a", "b", "a", "a", "b"),
  z = c("K", "K", "L", "L", "K")
)
```

a. group_by()

```
df |>
  group_by(y)
```

b. arrange() a group_by()

```
df |>
  arrange(y)
```

c.

```
df |>
  group_by(y) |>
  summarize(mean_x = mean(x))
```

d.

```
df |>
  group_by(y, z) |>
  summarize(mean_x = mean(x))
```

e. (d) ?

```
df |>
  group_by(y, z) |>
  summarize(mean_x = mean(x), .groups = "drop")
```

f. ?

```
df |>
  group_by(y, z) |>
  summarize(mean_x = mean(x))

df |>
```

```
group_by(y, z) |>
  mutate(mean_x = mean(x))
```

3.7 :

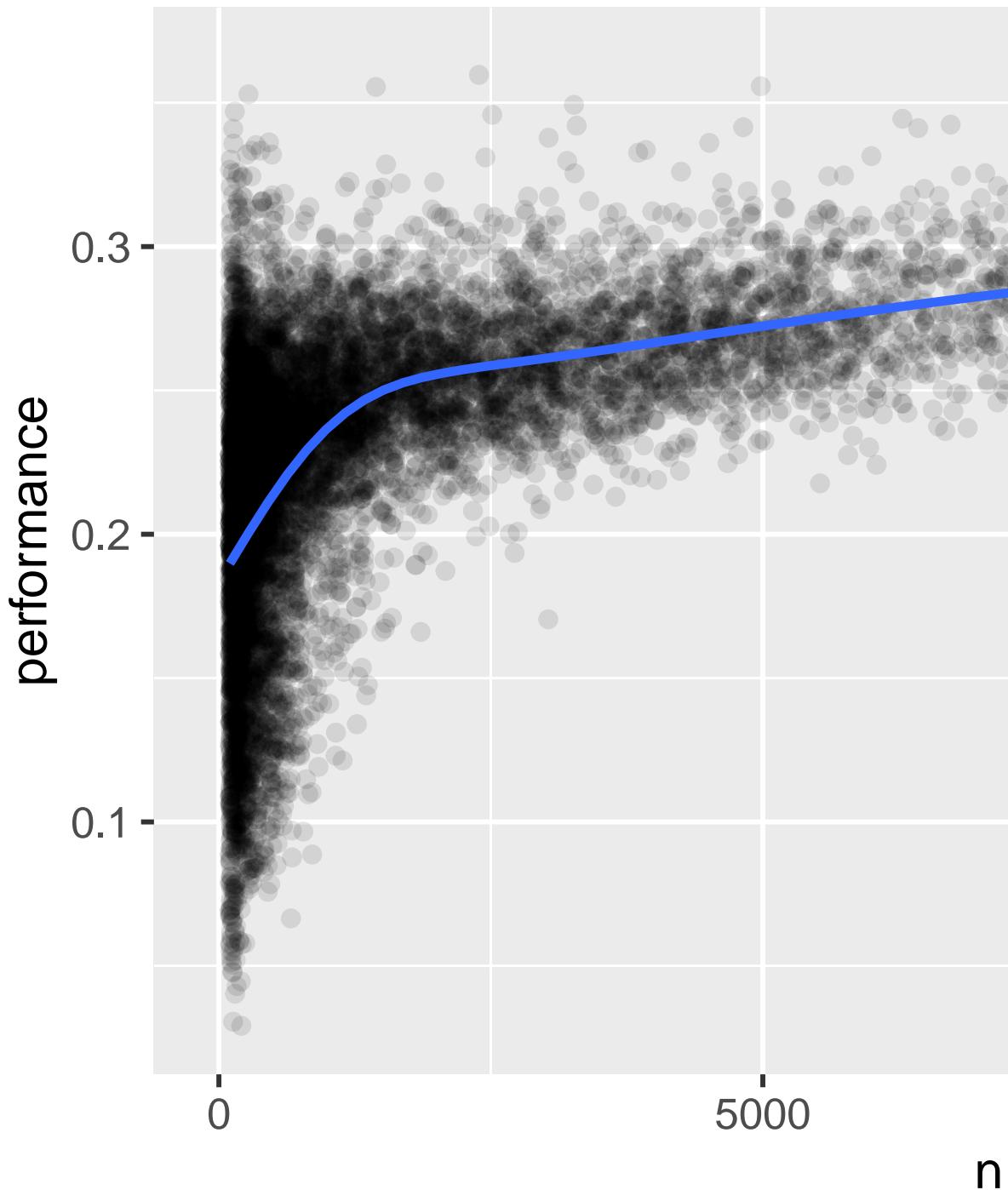
n()	Lahman	H	AB
-----	---------------	---	----

```
batters <- Lahman::Batting |>
  group_by(playerID) |>
  summarize(
    performance = sum(H, na.rm = TRUE) / sum(AB, na.rm = TRUE),
    n = sum(AB, na.rm = TRUE)
  )
batters
#> # A tibble: 20,469 x 3
#>   playerID  performance     n
#>   <chr>      <dbl> <int>
#> 1 aardsda01      0.305 12364
#> 2 aaronha01      0.229  944
#> 3 aaronto01      0.0952   21
#> 4 aasedo01       0.111    9
#> 5 abadan01      0.0952   21
#> 6 abadfe01      0.111    9
#> # i 20,463 more rows
```

performance	n
-------------	---

1.	4
2. performance	n

```
batters |>
  filter(n > 100) |>
  ggplot(aes(x = n, y = performance)) +
  geom_point(alpha = 1 / 10) +
  geom_smooth(se = FALSE)
```



```
ggplot2 dplyr          |>      +
  desc(performance)
```

```
batters |>
  arrange(desc(performance))
#> # A tibble: 20,469 x 3
#>   playerID  performance     n
#>   <chr>        <dbl> <int>
#> 1 abramge01      1     1
#> 2 alberan01      1     1
#> 3 banisje01      1     1
#> 4 bartocl01      1     1
#> 5 bassdo01      1     1
#> 6 birasst01      1     2
#> # i 20,463 more rows
```

http://varianceexplained.org/r/empirical_bayes_baseball/
<https://www.evanmiller.org/how-not-to-sort-by-average-rating.html>.

3.8

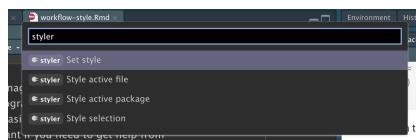
dplyr	filter() arrange()	select() mutate()	group_by() summarize()
“ ”			

Chapter 4

•

tidyverse

install.packages("styler") dio	RStudio Cmd/Ctrl + Shift + P	Lorenz “styler”	Walthert styler RStu- ??
-----------------------------------	---------------------------------	--------------------	---------------------------------------



4.1: RStudio’s command palette makes it easy to access every RStudio command using only the keyboard.

tidyverse nycflights13

```
library(tidyverse)
library(nycflights13)
```

4.1

```
@sec-whats-in-a-name           <- mutate()          - - 
# Strive for:
short_flights <- flights |> filter(air_time < 60)

# Avoid:
SHORTFLIGHTS <- flights |> filter(air_time < 60)
```

4.2

\wedge (- == < ...) (<-)

```
# Strive for  
z <- (a + b)^2 / d  
  
# Avoid  
z<-( a + b ) ^ 2/d
```

```
# Strive for  
mean(x, na.rm = TRUE)  
  
# Avoid  
mean (x ,na.rm=TRUE)
```

mutate() = 1

```
flights |>
  mutate(
    speed      = distance / air_time,
    dep_hour   = dep_time %/% 100,
    dep_minute = dep_time %% 100
  )
```

4.3

|>

```
# Strive for
flights |>
  filter(!is.na(arr_delay), !is.na(tailnum)) |>
  count(dest)

# Avoid
flights|>filter(!is.na(arr_delay), !is.na(tailnum))|>count(dest)
```

¹ dep_time HMM HHMM (%) (%) () (%)

```
mutate() summarize()          select() filter()

# Strive for
flights |>
  group_by(tailnum) |>
  summarize(
    delay = mean(arr_delay, na.rm = TRUE),
    n = n()
  )

# Avoid
flights |>
  group_by(
    tailnum
  ) |>
  summarize(delay = mean(arr_delay, na.rm = TRUE), n = n())

|RStudio|> )
```

```
# Strive for
flights |>
  group_by(tailnum) |>
  summarize(
    delay = mean(arr_delay, na.rm = TRUE),
    n = n()
  )

# Avoid
flights|>
  group_by(tailnum) |>
  summarize(
    delay = mean(arr_delay, na.rm = TRUE),
    n = n()
  )

# Avoid
flights|>
  group_by(tailnum) |>
  summarize(
    delay = mean(arr_delay, na.rm = TRUE),
    n = n()
  )
```

```
# This fits compactly on one line
df |> mutate(y = x + 1)

# While this takes up 4x as many lines, it's easily extended to
# more variables and more steps in the future
df |>
  mutate(
    y = x + 1
  )
```

10-15

4.4 ggplot2

```
ggplot2 + |>

flights |>
  group_by(month) |>
  summarize(
    delay = mean(arr_delay, na.rm = TRUE)
  ) |>
  ggplot(aes(x = month, y = delay)) +
  geom_point() +
  geom_line()

:

flights |>
  group_by(dest) |>
  summarize(
    distance = mean(distance),
    speed = mean(distance / air_time, na.rm = TRUE)
  ) |>
  ggplot(aes(x = distance, y = speed)) +
  geom_smooth(
    method = "loess",
    span = 0.5,
    se = FALSE,
    color = "white",
    linewidth = 4
  ) +
  geom_point()
```

```
|> + ggplot2
```

4.5

sectioning comments

```
# Load data -----  
# Plot data -----
```

RStudio (Cmd/Ctrl Shift R) ??



4.2: After adding sectioning comments to your script, you can easily navigate to them using the code navigation tool in the bottom-left of the script editor.

4.6

1.

```
flights|>filter(dest=="IAH")|>group_by(year,month,day)|>summarize(n=n(),  
delay=mean(arr_delay,na.rm=TRUE))|>filter(n>10)  
  
flights|>filter(carrier=="UA",dest%in%c("IAH","HOU"),sched_dep_time>  
0900,sched_arr_time<2000)|>group_by(flight)|>summarize(delay=mean(  
arr_delay,na.rm=TRUE),cancelled=sum(is.na(arr_delay)),n=n())|>filter(n>10)
```

4.7

styler

tidyverse
tidyR

tidyverse

Chapter 5

5.1

“ ”
— .
“ ”
— .
R tidy data tidy-
verse data pivoting

5.1.1

tidyverse

```
library(tidyverse)
```

```
library(tidyverse)
```

5.2

country year population tuberculosis TB cases

```

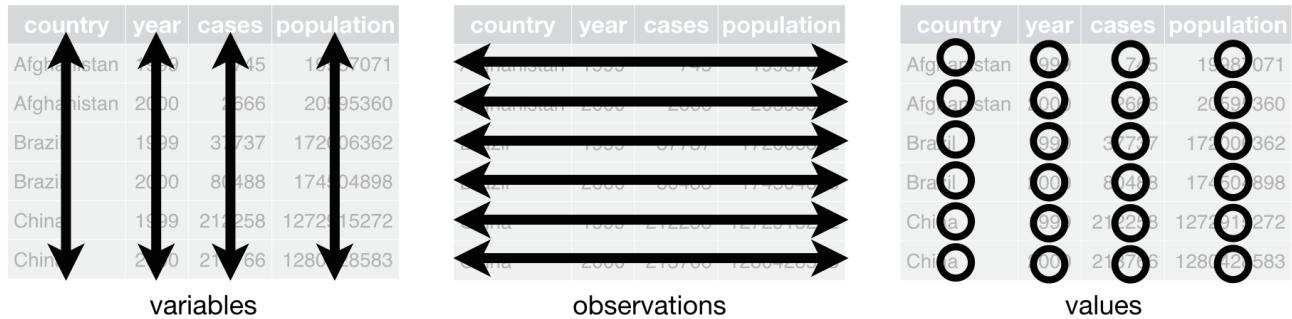


```

table1 tidyverse tidy

- 1.
- 2.
- 3.

??



5.1: The following three rules make a dataset tidy: variables are columns, observations are rows, and values are cells.

1.

2.

R

@sec-mutate @sec-summarize

R

dplyr ggplot2 tidyverse table1

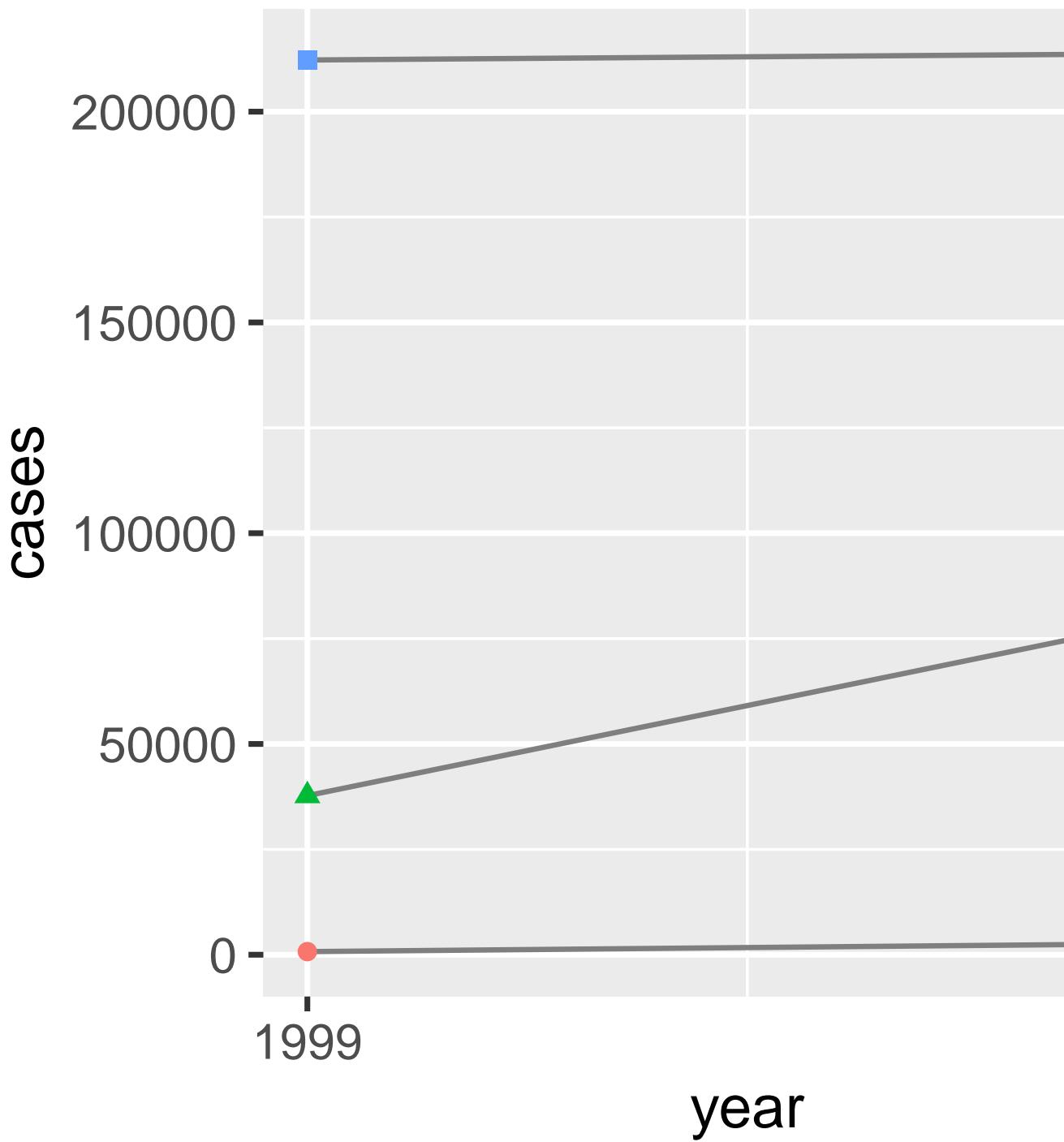
```
# Compute rate per 10,000
table1 |>
  mutate(rate = cases / population * 10000)
#> # A tibble: 6 x 5
#>   country     year    cases population    rate
#>   <chr>      <dbl>    <dbl>        <dbl>    <dbl>
#> 1 Afghanistan 1999     745    19987071  0.373
#> 2 Afghanistan 2000    2666    20595360  1.29
#> 3 Brazil       1999   37737   172006362  2.19
#> 4 Brazil       2000   80488   174504898  4.61
#> 5 China        1999  212258  1272915272  1.67
#> 6 China        2000  213766  1280428583  1.67

# Compute total cases per year
table1 |>
  group_by(year) |>
  summarise(total_cases = sum(cases))
#> # A tibble: 2 x 2
#>   year total_cases
#>   <dbl>      <dbl>
#> 1 1999      250740
#> 2 2000      296920
```

```
# Visualize changes over time
ggplot(table1, aes(x = year, y = cases)) +
  geom_line(aes(group = country), color = "grey50") +
  geom_point(aes(color = country, shape = country)) +
  scale_x_continuous(breaks = c(1999, 2000)) # x-axis breaks at 1999 and 2000
```

5.2.

107



5.2.1

- 1.
 2. **table2 table3 rate**
- a. TB
 - b.
 - c. 10000
 - d. .

5.3 —

- 1.
- 2.

```
pivot
tidyr      pivot_longer() pivot_wider()    pivot_longer()
```

5.3.1

```
billboard 2000 Billboard :
```

```
billboard
#> # A tibble: 317 x 79
#>   artist      track      date.entered    wk1    wk2    wk3    wk4    wk5
#>   <chr>     <chr>     <date>      <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
#> 1 2 Pac     Baby Don't Cry (Ke~ 2000-02-26     87     82     72     77     87
#> 2 2Ge+her   The Hardest Part 0~ 2000-09-02     91     87     92     NA     NA
#> 3 3 Doors Down Kryptonite    2000-04-08     81     70     68     67     66
#> 4 3 Doors Down Loser       2000-10-21     76     76     72     69     67
#> 5 504 Boyz   Wobble Wobble    2000-04-15     57     34     25     17     17
#> 6 98^0       Give Me Just One N~ 2000-08-19     51     39     34     26     26
#> # i 311 more rows
#> # i 71 more variables: wk6 <dbl>, wk7 <dbl>, wk8 <dbl>, wk9 <dbl>, ...
```

```

      artist track  date.entered      76 wk1-wk76      1
week          rank

pivot_longer():

billboard |>
pivot_longer(
  cols = starts_with("wk"),
  names_to = "week",
  values_to = "rank"
)
#> # A tibble: 24,092 x 5
#>   artist track           date.entered week   rank
#>   <chr>  <chr>          <date>       <chr> <dbl>
#> 1 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk1     87
#> 2 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk2     82
#> 3 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk3     72
#> 4 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk4     77
#> 5 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk5     87
#> 6 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk6     94
#> 7 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk7     99
#> 8 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk8     NA
#> 9 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk9     NA
#> 10 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk10    NA
#> # i 24,082 more rows

```

- cols select() !c(artist, track, date.entered) starts_with("wk")
- names_to week
- values_to rank

```

"week" "rank" pivot_longer()
100    76      2 Pac Baby Don't Cry      100    7
NA          pivot_longer()  values_drop_na = TRUE

```

```

billboard |>
pivot_longer(
  cols = starts_with("wk"),
  names_to = "week",
  values_to = "rank",
  values_drop_na = TRUE

```

```

)
#> # A tibble: 5,307 x 5
#>   artist track           date.entered week  rank
#>   <chr>  <chr>          <date>      <chr> <dbl>
#> 1 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk1    87
#> 2 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk2    82
#> 3 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk3    72
#> 4 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk4    77
#> 5 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk5    87
#> 6 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk6    94
#> # i 5,301 more rows

```

```

NA
100  76
mutate() readr::parse_number() week      parse_number()

```

```

billboard_longer <- billboard |>
  pivot_longer(
    cols = starts_with("wk"),
    names_to = "week",
    values_to = "rank",
    values_drop_na = TRUE
  ) |>
  mutate(
    week = parse_number(week)
  )
billboard_longer
#> # A tibble: 5,307 x 5
#>   artist track           date.entered week  rank
#>   <chr>  <chr>          <date>      <dbl> <dbl>
#> 1 2 Pac Baby Don't Cry (Keep... 2000-02-26 1    87
#> 2 2 Pac Baby Don't Cry (Keep... 2000-02-26 2    82
#> 3 2 Pac Baby Don't Cry (Keep... 2000-02-26 3    72
#> 4 2 Pac Baby Don't Cry (Keep... 2000-02-26 4    77
#> 5 2 Pac Baby Don't Cry (Keep... 2000-02-26 5    87
#> 6 2 Pac Baby Don't Cry (Keep... 2000-02-26 6    94
#> # i 5,301 more rows

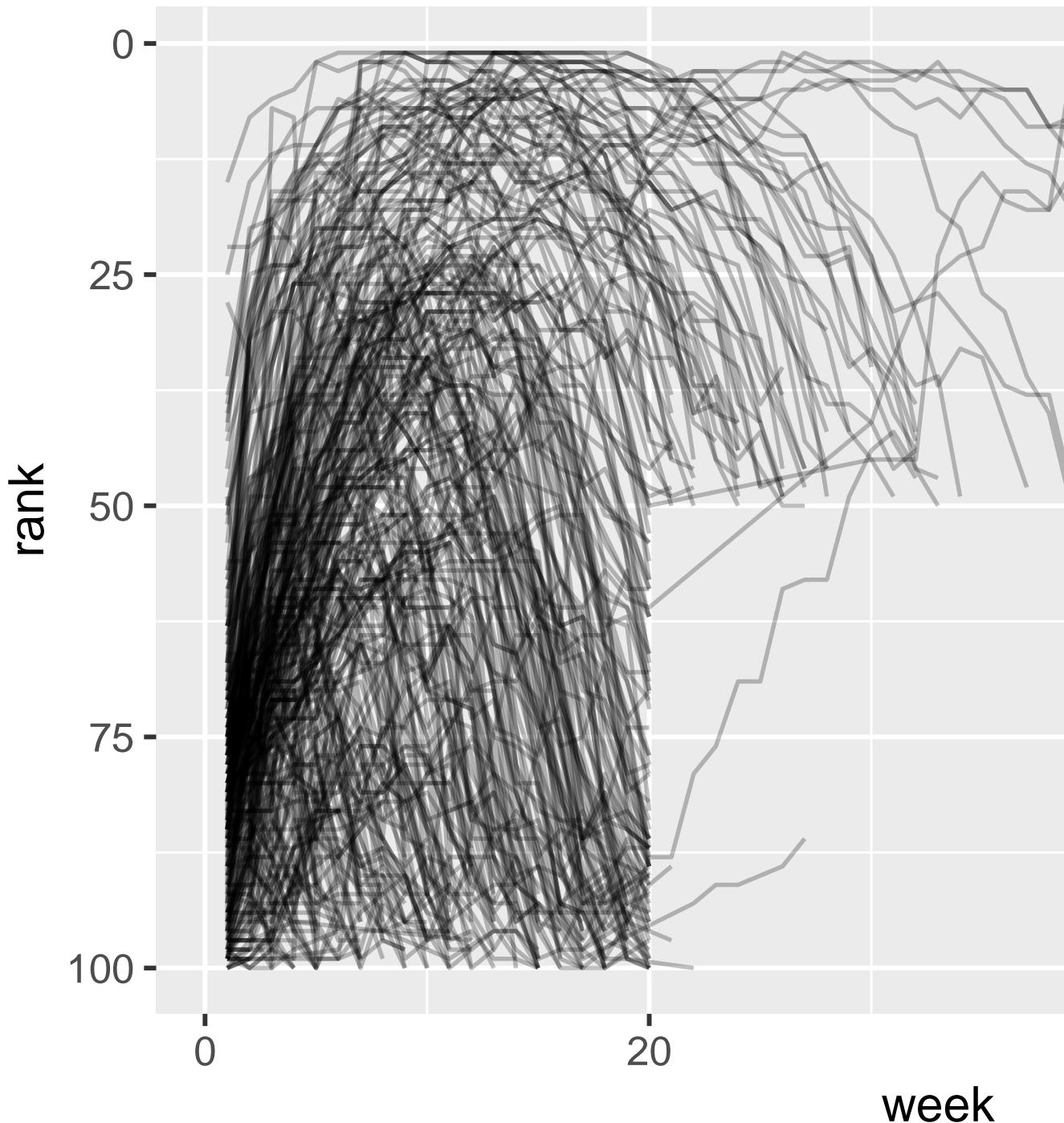
```

@fig-billboard-ranks 100 20

```

billboard_longer |>
  ggplot(aes(x = week, y = rank, group = track)) +
  geom_line(alpha = 0.25) +
  scale_y_reverse()

```



5.2: A line plot showing how the rank of a song changes over time.

5.3.2 pivoting ?

```

          pivoting      pivoting      id  A B C
tribble()    tribble()    tibble

df <- tribble(
  ~id, ~bp1, ~bp2,
  "A", 100, 120,
  "B", 140, 115,
  "C", 120, 125
)

          id   measurement   value      df  pivot

df |>
  pivot_longer(
    cols = bp1:bp2,
    names_to = "measurement",
    values_to = "value"
  )
#> # A tibble: 6 x 3
#>   id   measurement value
#>   <chr> <chr>     <dbl>
#> 1 A     bp1        100
#> 2 A     bp2        120
#> 3 B     bp1        140
#> 4 B     bp2        115
#> 5 C     bp1        120
#> 6 C     bp2        125

          ??           id
names_to  @fig-pivot-names
values_to @ fig-pivot-values

```

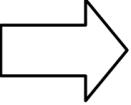
5.3.3

```

          table1      who2

who2
#> # A tibble: 7,240 x 58
#>   country      year sp_m_014 sp_m_1524 sp_m_2534 sp_m_3544 sp_m_4554
#>   <chr>       <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
#> 1 Afghanistan 1980      NA      NA      NA      NA      NA

```

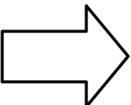


The diagram illustrates the process of pivoting a dataset. On the left, a wide table has three columns: **id**, **bp1**, and **bp2**. The rows are labeled **A**, **B**, and **C**, with corresponding values 100, 120; 140, 115; and 120, 125 respectively. An arrow points to the right, leading to a long table where each row corresponds to a combination of **id** and **measurement**. The **measurement** column contains **bp1** and **bp2**, and the **value** column contains the respective numerical values: 100, 120 for id A; 140, 115 for id B; and 120, 125 for id C.

id	bp1	bp2
A	100	120
B	140	115
C	120	125

id	measurement	value
A	bp1	100
A	bp2	120
B	bp1	140
B	bp2	115
C	bp1	120
C	bp2	125

5.3: Columns that are already variables need to be repeated, once for each column that is pivoted.



The diagram illustrates the process of pivoting a dataset. On the left, a wide table has three columns: **id**, **bp1**, and **bp2**. The rows are labeled **A**, **B**, and **C**, with corresponding values 100, 120; 140, 115; and 120, 125 respectively. An arrow points to the right, leading to a long table where each row corresponds to a combination of **id** and **measurement**. The **measurement** column contains **bp1** and **bp2**, and the **value** column contains the respective numerical values: 100, 120 for id A; 140, 115 for id B; and 120, 125 for id C.

id	bp1	bp2
A	100	120
B	140	115
C	120	125

id	measurement	value
A	bp1	100
A	bp2	120
B	bp1	140
B	bp2	115
C	bp1	120
C	bp2	125

5.4: The column names of pivoted columns become values in a new column. The values need to be repeated once for each row of the original dataset.

The diagram illustrates a data transformation process. On the left, there is a wide-format table with three columns: 'id' (A, B, C), 'bp1' (100, 140, 120), and 'bp2' (120, 115, 125). On the right, after the arrow, is a long-format table with three columns: 'id' (A, A, B, B, C, C), 'measurement' (bp1, bp2, bp1, bp2, bp1, bp2), and 'value' (100, 120, 140, 115, 120, 125).

id	bp1	bp2
A	100	120
B	140	115
C	120	125

id	measurement	value
A	bp1	100
A	bp2	120
B	bp1	140
B	bp2	115
C	bp1	120
C	bp2	125

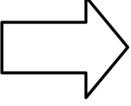
5.5: The number of values is preserved (not repeated), but unwound row-by-row.

```
#> 2 Afghanistan 1981      NA      NA      NA      NA      NA
#> 3 Afghanistan 1982      NA      NA      NA      NA      NA
#> 4 Afghanistan 1983      NA      NA      NA      NA      NA
#> 5 Afghanistan 1984      NA      NA      NA      NA      NA
#> 6 Afghanistan 1985      NA      NA      NA      NA      NA
#> # i 7,234 more rows
#> # i 51 more variables: sp_m_5564 <dbl>, sp_m_65 <dbl>, sp_f_014 <dbl>, ...
,           country year  56 sp_m_014 ep_m_4554 rel_m_3544
,           -       sp/rel/ep      m/f          014/1524/2534/3544/4554/5564/
14
who2      country year
pivot_longer()  name
```

```
who2 |>
pivot_longer(
  cols = !(country:year),
  names_to = c("diagnosis", "gender", "age"),
  names_sep = "_",
  values_to = "count"
)
#> # A tibble: 405,440 x 6
#>   country      year diagnosis gender age   count
#>   <chr>        <dbl> <chr>     <chr> <chr> <dbl>
#> 1 Afghanistan  1980    sp         m     014     NA
```

```
#> 2 Afghanistan 1980 sp      m    1524    NA
#> 3 Afghanistan 1980 sp      m    2534    NA
#> 4 Afghanistan 1980 sp      m    3544    NA
#> 5 Afghanistan 1980 sp      m    4554    NA
#> 6 Afghanistan 1980 sp      m    5564    NA
#> # i 405,434 more rows
```

names_sep names_pattern @sec-regular-expressions
@ sec-regular-expressions



id	x_1	y_2
A	1	2
B	3	4
C	5	6

id	name	number	value
A	x	1	1
A	y	2	2
B	x	1	3
B	y	2	4
C	x	1	5
C	y	2	6

5.6: Pivoting columns with multiple pieces of information in the names means that each column name now fills in values in multiple output columns.

5.3.4

household

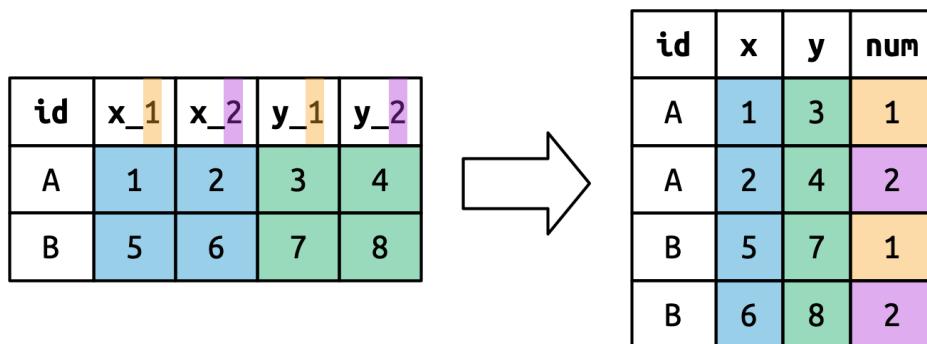
```
household
#> # A tibble: 5 x 5
#>   family dob_child1 dob_child2 name_child1 name_child2
#>   <int> <date>     <date>     <chr>     <chr>
#> 1      1 1998-11-26 2000-01-29 Susan      Jose
#> 2      2 1996-06-22 NA          Mark      <NA>
#> 3      3 2002-07-11 2004-04-05 Sam       Seth
#> 4      4 2004-10-10 2009-08-27 Craig     Khai
#> 5      5 2000-12-05 2005-02-28 Parker   Gracie
```

```
values_to

household |>
  pivot_longer(
    cols = !family,
    names_to = c(".value", "child"),
    names_sep = "_",
    values_drop_na = TRUE
  )
#> # A tibble: 9 x 4
#>   family child   dob      name
#>   <int> <chr> <date>    <chr>
#> 1     1 child1 1998-11-26 Susan
#> 2     1 child2 2000-01-29 Jose
#> 3     2 child1 1996-06-22 Mark
#> 4     3 child1 2002-07-11 Sam
#> 5     3 child2 2004-04-05 Seth
#> 6     4 child1 2004-10-10 Craig
#> # i 3 more rows
```

".value

	values_drop_na = TRUE	NA
??	names_to ".value"	



5.7: Pivoting with `names_to = c(".value", "num")` splits the column names into two components: the first part determines the output column name (`x` or `y`), and the second part determines the value of the `num` column.

5.4 —

`pivot_longer()`

HA HA

```
pivot    " "    pivot_wider()
```

`cms_patient_experience`

Centers of Medicare and Medi-

caid Services

```
cms_patient_experience
#> # A tibble: 500 x 5
#>   org_pac_id org_nm               measure_cd  measure_title  prf_rate
#>   <chr>      <chr>              <chr>       <chr>          <dbl>
#> 1 0446157747 USC CARE MEDICAL GROUP INC CAHPS_GRP_1 CAHPS for MIPS~ 63
#> 2 0446157747 USC CARE MEDICAL GROUP INC CAHPS_GRP_2 CAHPS for MIPS~ 87
#> 3 0446157747 USC CARE MEDICAL GROUP INC CAHPS_GRP_3 CAHPS for MIPS~ 86
#> 4 0446157747 USC CARE MEDICAL GROUP INC CAHPS_GRP_5 CAHPS for MIPS~ 57
#> 5 0446157747 USC CARE MEDICAL GROUP INC CAHPS_GRP_8 CAHPS for MIPS~ 85
#> 6 0446157747 USC CARE MEDICAL GROUP INC CAHPS_GRP_12 CAHPS for MIPS~ 24
#> # i 494 more rows
```

```
distinct()  measure_cd measure_title
```

```
cms_patient_experience |>
  distinct(measure_cd, measure_title)
#> # A tibble: 6 x 2
#>   measure_cd  measure_title
#>   <chr>        <chr>
#> 1 CAHPS_GRP_1 CAHPS for MIPS SSM: Getting Timely Care, Appointments, and In~
#> 2 CAHPS_GRP_2 CAHPS for MIPS SSM: How Well Providers Communicate
#> 3 CAHPS_GRP_3 CAHPS for MIPS SSM: Patient's Rating of Provider
#> 4 CAHPS_GRP_5 CAHPS for MIPS SSM: Health Promotion and Education
#> 5 CAHPS_GRP_8 CAHPS for MIPS SSM: Courteous and Helpful Office Staff
#> 6 CAHPS_GRP_12 CAHPS for MIPS SSM: Stewardship of Patient Resources
```

measure_cd	measure_title	measure_cd	values_from	names_from
pivot_wider()	pivot_longer()			

```
cms_patient_experience |>
  pivot_wider(
    names_from = measure_cd,
    values_from = prf_rate
  )
#> # A tibble: 500 x 9
#>   org_pac_id org_nm               measure_title  CAHPS_GRP_1  CAHPS_GRP_2
#>   <chr>      <chr>              <chr>          <dbl>        <dbl>
#> 1 0446157747 USC CARE MEDICAL GROUP ~ CAHPS for MIPS~       63          NA
#> 2 0446157747 USC CARE MEDICAL GROUP ~ CAHPS for MIPS~       NA          87
#> 3 0446157747 USC CARE MEDICAL GROUP ~ CAHPS for MIPS~       NA          NA
#> 4 0446157747 USC CARE MEDICAL GROUP ~ CAHPS for MIPS~       NA          NA
```

```
#> 5 0446157747 USC CARE MEDICAL GROUP ~ CAHPS for MIPS~ NA NA
#> 6 0446157747 USC CARE MEDICAL GROUP ~ CAHPS for MIPS~ NA NA
#> # i 494 more rows
#> # i 4 more variables: CAHPS_GRP_3 <dbl>, CAHPS_GRP_5 <dbl>, ...
#>
#> pivot_wider() "org"
#>
#> cms_patient_experience |>
#>   pivot_wider(
#>     id_cols = starts_with("org"),
#>     names_from = measure_cd,
#>     values_from = prf_rate
#>   )
#> # A tibble: 95 x 8
#>   org_pac_id org_nm      CAHPS_GRP_1 CAHPS_GRP_2 CAHPS_GRP_3 CAHPS_GRP_5
#>   <chr>       <chr>        <dbl>       <dbl>       <dbl>       <dbl>
#> 1 0446157747 USC CARE MEDICA~       63          87          86          57
#> 2 0446162697 ASSOCIATION OF ~      59          85          83          63
#> 3 0547164295 BEAVER MEDICAL ~     49          NA          75          44
#> 4 0749333730 CAPE PHYSICIANS~    67          84          85          65
#> 5 0840104360 ALLIANCE PHYSIC~    66          87          87          64
#> 6 0840109864 REX HOSPITAL INC    73          87          84          67
#> # i 89 more rows
#> # i 2 more variables: CAHPS_GRP_8 <dbl>, CAHPS_GRP_12 <dbl>
```

5.4.1 pivot_wider()

	id	A	B	A	B
pivot_wider()					

```
df <- tribble(
  ~id, ~measurement, ~value,
  "A",      "bp1",    100,
  "B",      "bp1",    140,
  "B",      "bp2",    115,
  "A",      "bp2",    120,
  "A",      "bp3",    105
)
```

value	measurement	:

```

df |>
  pivot_wider(
    names_from = measurement,
    values_from = value
  )
#> # A tibble: 2 x 4
#>   id      bp1    bp2    bp3
#>   <chr> <dbl> <dbl> <dbl>
#> 1 A        100    120    105
#> 2 B        140    115     NA

pivot_wider()               measurement

df |>
  distinct(measurement) |>
  pull()
#> [1] "bp1" "bp2" "bp3"

id_cols

df |>
  select(-measurement, -value) |>
  distinct()
#> # A tibble: 2 x 1
#>   id
#>   <chr>
#> 1 A
#> 2 B

pivot_wider()      :
df |>
  select(-measurement, -value) |>
  distinct() |>
  mutate(x = NA, y = NA, z = NA)
#> # A tibble: 2 x 4
#>   id      x      y      z
#>   <chr> <lgl> <lgl> <lgl>
#> 1 A      NA     NA     NA
#> 2 B      NA     NA     NA

B                         ??  pivot_wider() “ ”
id “A” measurement “bp1”

```

```

df <- tribble(
  ~id, ~measurement, ~value,
  "A",      "bp1",    100,
  "A",      "bp1",    102,
  "A",      "bp2",    120,
  "B",      "bp1",    140,
  "B",      "bp2",    115
)

#> Warning: Values from `value` are not uniquely identified; output will contain
#> list-cols.
#> * Use `values_fn = list` to suppress this warning.
#> * Use `values_fn = {summary_fun}` to summarise duplicates.
#> * Use the following dplyr code to identify duplicates.
#>   {data} |>
#>     dplyr::summarise(n = dplyr::n(), .by = c(id, measurement)) |>
#>     dplyr::filter(n > 1L)
#> # A tibble: 2 x 3
#>   id     bp1     bp2
#>   <chr> <list>   <list>
#> 1 A      <dbl [2]> <dbl [1]>
#> 2 B      <dbl [1]> <dbl [1]>

:

df |>
  group_by(id, measurement) |>
  summarize(n = n(), .groups = "drop") |>
  filter(n > 1)
#> # A tibble: 1 x 3
#>   id     measurement     n
#>   <chr> <chr>       <int>
#> 1 A      bp1            2

```

5.5

```
tidyverse
pivot_longer() pivot_wider()
package = "tidyr" vignette("pivot",
                           " ")
vignette
```

Journal of Statistical Software Tidy Data

R

Chapter 6

:

6.1

ggplot2 dplyr
“ ” “ ” “R ” Cmd/Ctrl + Shift + N ??

6.1.1

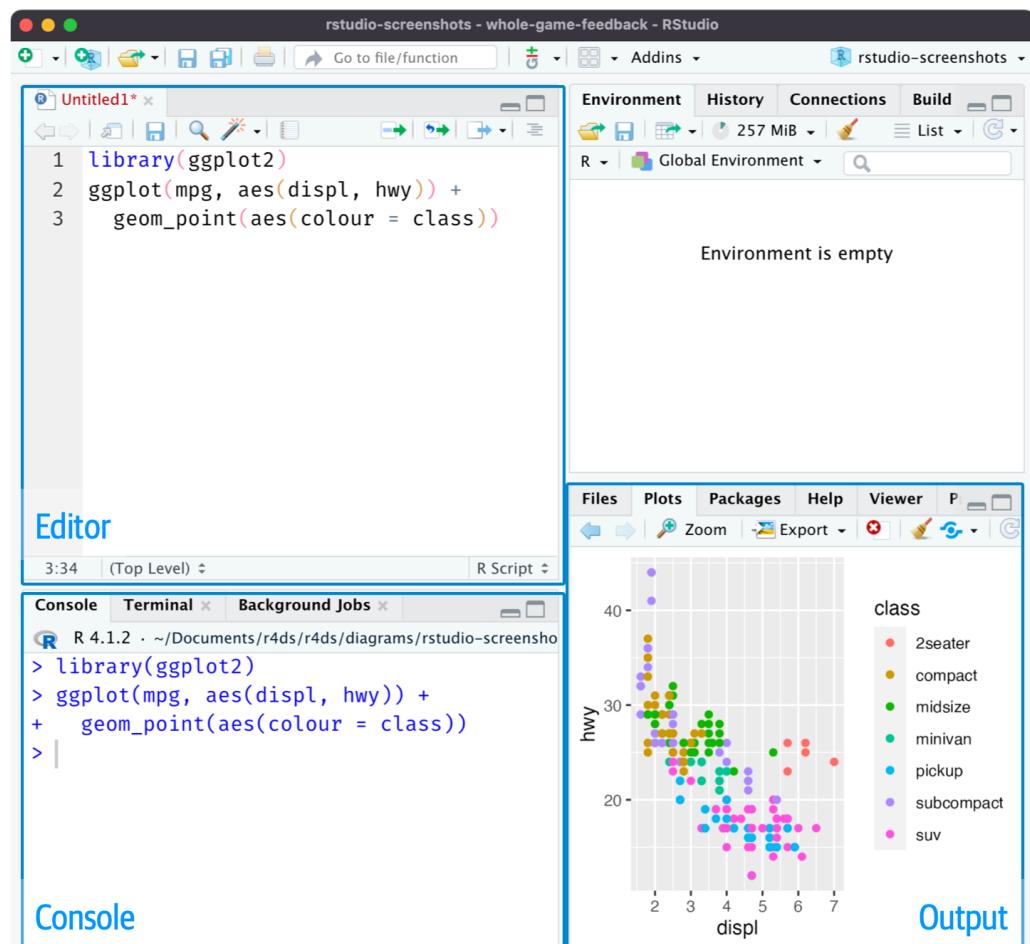
ggplot2 dplyr Cmd/Ctrl + Enter R

```
library(dplyr)
library(nycflights13)

not_cancelled <- flights |>
  filter(!is.na(dep_delay) , !is.na(arr_delay))

not_cancelled |>
  group_by(year, month, day) |>
  summarize(mean = mean(dep_delay))
```

|> Cmd/Ctrl + Enter not_cancelled
Cmd/Ctrl + Enter
Cmd/Ctrl + Shift + S



6.1: Opening the script editor adds a new pane at the top-left of the IDE.

```
install.packages()
```

6.1.2 RStudio

RStudio

A screenshot of the RStudio interface. In the code editor, there is a syntax error highlighted with a red circle and a tooltip. The code contains an assignment operator '←' which is not valid in R. The tooltip shows the error message: 'unexpected token 'y''. The code snippet is:

```
3 x y ← 10
```

RStudio

A screenshot of the RStudio interface. A warning message is displayed in the status bar at the bottom of the screen. It says: 'use 'is.na' to check whether expression evaluates to NA'. The code snippet is:

```
3 == NA
```

6.1.3

RStudio

“Untitled1” “Untitled2” “Untitled3”

code.R myscript.R

- 1.
- 2.
- 3.

alternative model.R
 code for exploratory analysis.r
 finalreport.qmd
 FinalReport.qmd
 fig 1.png
 Figure_02.png
 model_first_try.R
 run-first.r
 temp.txt

<pre>finalreport FinalReport¹ temp</pre>	<pre>run-first</pre>
--	----------------------

¹ “final” Piled Higher and Deeper

```
01-load-data.R
02-exploratory-analysis.R
03-model-approach-1.R
04-model-approach-2.R
fig-01.png
fig-02.png
report-2022-03-20.qmd
report-2022-04-02.qmd
report-draft-notes.txt
```

```
temp report-draft-notes
```

6.2

R

R R

1.

2.

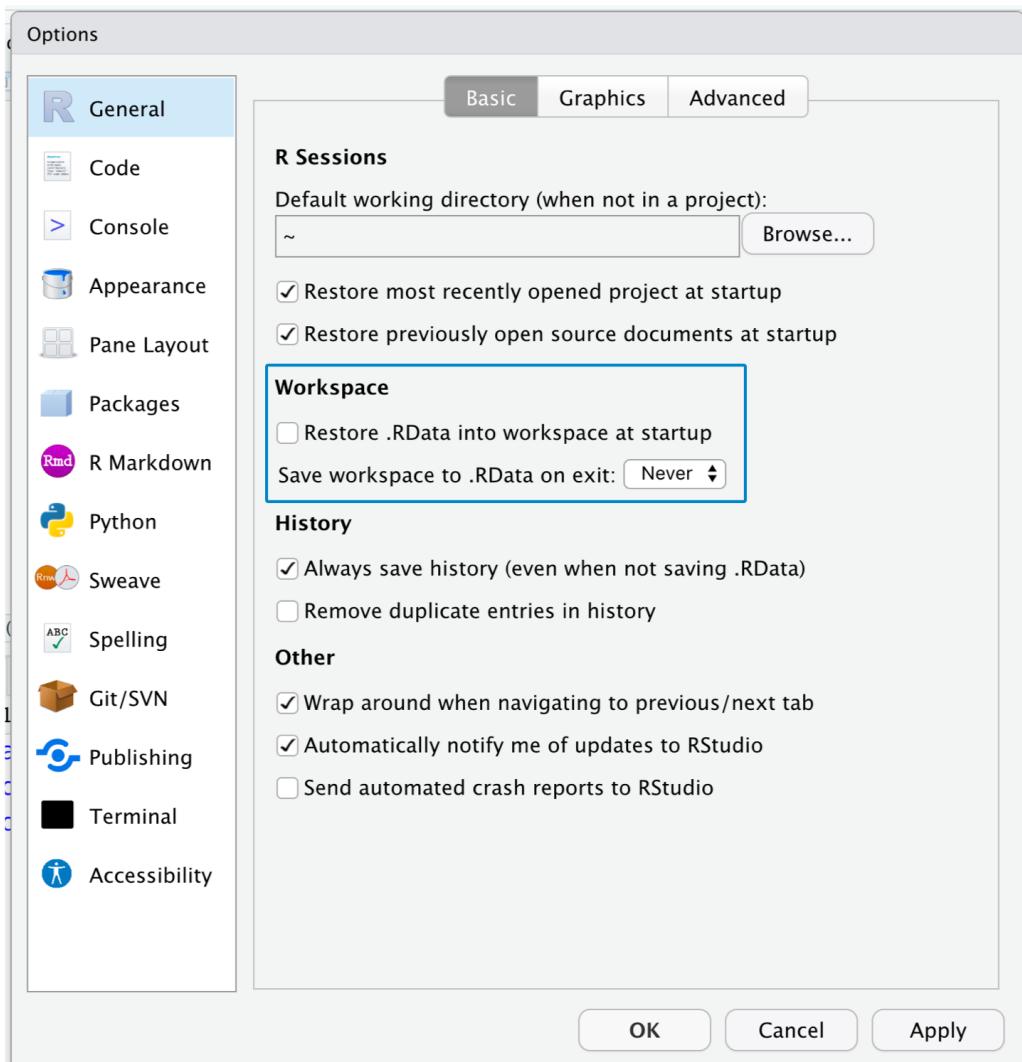
6.2.1

Environment R	R	R	R
R ??	RStudio	workspace RStudio	<code>usethis::use_blank_slate()</code> ²

1. Cmd/Ctrl + Shift + 0/F10 R
2. Cmd/Ctrl + Shift + S

“ ”>“ R”

² usethis install.packages("usethis")



6.2: Copy these options in your RStudio options to always start your RStudio session with a clean slate.



6.2.2 ?

R R R Studio

Console Terminal Find in Files
R 4.1.2 - ~/Documents/r4ds/

`getwd()` R : :

```
getwd()
#> [1] "/Users/hadley/Documents/r4ds"
```

R “ ” Hadley Documents r4ds Hadley

R R

R

```
setwd("/path/to/my/CoolProject")
```

R RStudio

6.2.3 RStudio

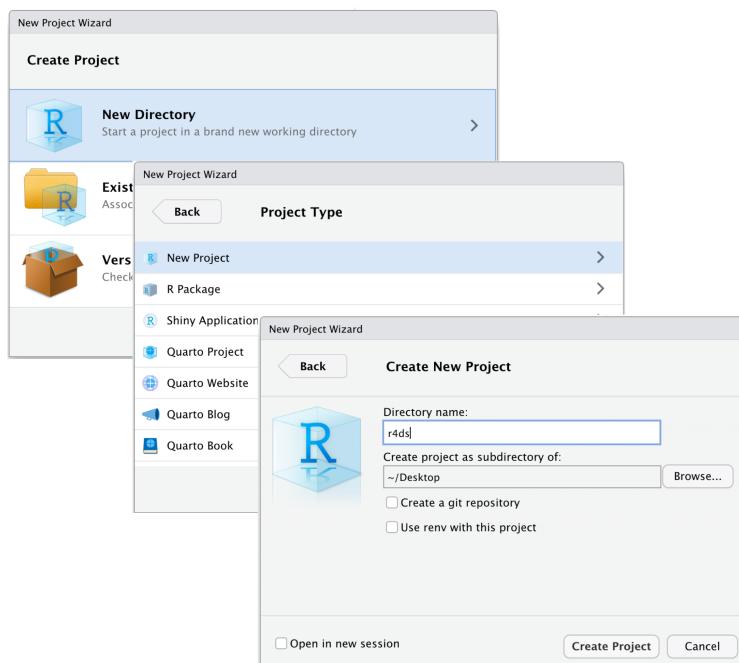
R RStudio
“ ”>“ ” @fig-new-project

r4ds

RStudio “ ”

```
getwd()
#> [1] /Users/hadley/Documents/r4ds
```

“diamonds.R” “data” RStudio “ ” “ ”
PNG CSV



6.3: To create new project: (top) first click New Directory, then (middle) click New Project, then (bottom) fill in the directory (project) name, choose a good subdirectory for its home and click Create Project.

```
library(tidyverse)

ggplot(diamonds, aes(x = carat, y = price)) +
  geom_hex()
ggsave("diamonds.png")

write_csv(diamonds, "data/diamonds.csv")
```

RStudio — .Rproj

diamonds.png	PNG	diamonds.R
R		

6.2.4

Mine R	Hadley /Users/Mine/Documents/r4ds Windows	data/diamonds.csv /Users/Mine/Documents/r4ds/data/diamonds.csv	C: \servername	Mac/Linux "/" /user
	Mac Linux R	data/diamonds.csv Windows	data\diamonds.csv Linux/Mac	R

6.3

1. RStudio Tips Twitter <https://twitter.com/rstudiotips>
2. RStudio <https://support.posit.co/hc/en-us/articles/205753617-Code-Diagnostics>

6.4

- RStudio
- R
-

R readr R

Chapter 7

7.1

```
R           R  
          R           R
```

7.1.1

```
readr R     readr   tidyverse
```

```
library(tidyverse)
```

7.2

```
CSV Comma-Separated Values      CSV
```

```
Student ID,Full Name,favourite.food,mealPlan,AGE  
1,Sunil Huffmann,Strawberry yoghurt,Lunch only,4  
2,Barclay Lynn,French fries,Lunch only,5  
3,Jayendra Lyne,N/A,Breakfast and lunch,7  
4,Leon Rossini,Anchovies,Lunch only,  
5,Chidiegwu Dunkel,Pizza,Breakfast and lunch,five  
6,Güvenç Attila,Ice cream,Lunch only,6
```

```
??
```

7.1: Data from the students.csv file as a table.

Student ID	Full Name	favourite.food	mealPlan	AGE
1	Sunil Huffmann	Strawberry yoghurt	Lunch only	4
2	Barclay Lynn	French fries	Lunch only	5
3	Jayendra Lyne	N/A	Breakfast and lunch	7
4	Leon Rossini	Anchovies	Lunch only	NA
5	Chidiegwu Dunkel	Pizza	Breakfast and lunch	five
6	Güvenç Attila	Ice cream	Lunch only	6

```
read_csv()      R           students.csv  data

students <- read_csv("data/students.csv")
#> Rows: 6 Columns: 5
#> -- Column specification --
#> Delimiter: ","
#> chr (4): Full Name, favourite.food, mealPlan, AGE
#> dbl (1): Student ID
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

data students.csv <https://pos.it/r4ds-students-csv>
 students.csv URL

```
students <- read_csv("https://pos.it/r4ds-students-csv")
```

read_csv() readr @sec-
 col-types

7.2.1

students

```
students
#> # A tibble: 6 x 5
#>   `Student ID` `Full Name`    favourite.food    mealPlan      AGE
#>       <dbl> <chr>          <chr>            <chr>        <chr>
```

```
#> 1      1 Sunil Huffmann    Strawberry yoghurt Lunch only      4
#> 2      2 Barclay Lynn     French fries      Lunch only      5
#> 3      3 Jayendra Lyne   N/A                  Breakfast and lunch 7
#> 4      4 Leon Rossini    Anchovies       Lunch only      <NA>
#> 5      5 Chidiegwu Dunkel Pizza            Breakfast and lunch five
#> 6      6 Güvenç Attila    Ice cream       Lunch only      6

favourite.food      N/A      R      " "  NA      na      read_csv()      ("")
NAs      "N/A"

students <- read_csv("data/students.csv", na = c("N/A", ""))

students
#> # A tibble: 6 x 5
#>   `Student ID` `Full Name`  favourite.food  mealPlan      AGE
#>   <dbl> <chr>        <chr>          <chr>        <chr>
#> 1      1 Sunil Huffmann  Strawberry yoghurt Lunch only      4
#> 2      2 Barclay Lynn    French fries      Lunch only      5
#> 3      3 Jayendra Lyne  <NA>            Breakfast and lunch 7
#> 4      4 Leon Rossini   Anchovies       Lunch only      <NA>
#> 5      5 Chidiegwu Dunkel Pizza          Breakfast and lunch five
#> 6      6 Güvenç Attila   Ice cream       Lunch only      6
```

Student ID Full Name R non-syntactic

```
students |>
  rename(
    student_id = `Student ID`,
    full_name = `Full Name`
  )
#> # A tibble: 6 x 5
#>   student_id full_name      favourite.food mealPlan     AGE
#>   <dbl> <chr>          <chr>           <chr>        <chr>
#> 1       1 Sunil Huffmann Strawberry yoghurt Lunch only    4
#> 2       2 Barclay Lynn    French fries      Lunch only    5
#> 3       3 Jayendra Lyne  <NA>             Breakfast and lunch 7
#> 4       4 Leon Rossini   Anchovies        Lunch only    <NA>
#> 5       5 Chidiegwu Dunkel Pizza           Breakfast and lunch five
#> 6       6 Güvenç Attila  Ice cream        Lunch only    6
```

```
janitor::clean_names() snake case1
```

`janitor tidyverse` |>

```
students |> janitor::clean_names()
#> # A tibble: 6 x 5
#>   student_id full_name      favourite_food    meal_plan     age
#>   <dbl> <chr>           <chr>            <chr>          <chr>
#> 1       1 Sunil Huffmann  Strawberry yoghurt Lunch only    4
#> 2       2 Barclay Lynn    French fries      Lunch only    5
#> 3       3 Jayendra Lyne  <NA>              Breakfast and lunch 7
#> 4       4 Leon Rossini   Anchovies        Lunch only    <NA>
#> 5       5 Chidiegwu Dunkel Pizza            Breakfast and lunch five
#> 6       6 Güvenç Attila   Ice cream        Lunch only    6
```

meal_plan R

```
students |>
  janitor::clean_names() |>
  mutate(meal_plan = factor(meal_plan))
#> # A tibble: 6 x 5
#>   student_id full_name      favourite_food    meal_plan     age
#>   <dbl> <chr>            <chr>           <fct>        <chr>
#> 1       1 Sunil Huffmann  Strawberry yoghurt Lunch only    4
#> 2       2 Barclay Lynn    French fries      Lunch only    5
#> 3       3 Jayendra Lyne  <NA>             Breakfast and lunch 7
#> 4       4 Leon Rossini   Anchovies       Lunch only    <NA>
#> 5       5 Chidiegwu Dunkel Pizza          Breakfast and lunch five
#> 6       6 Güvenç Attila   Ice cream       Lunch only    6
```

meal_plan <chr> <fct> ??

age id age five 5 ??

```
students <- students |>
  janitor::clean_names() |>
  mutate(
    meal_plan = factor(meal_plan),
    age = parse_number(if_else(age == "five", "5", age))
  )

students
#> # A tibble: 6 x 5
#>   student_id full_name      favourite_food    meal_plan      age
#>   <dbl> <chr>            <chr>           <fct>          <dbl>
#> 1       1 Sunil Huffmann  Strawberry yoghurt Lunch only     4
#> 2       2 Barclay Lynn    French fries      Lunch only     5
#> 3       3 Jayendra Lyne  <NA>             Breakfast and lunch 7
#> 4       4 Leon Rossini   Anchovies       Lunch only     NA
```

```
#> 5      5 Chidiegwu Dunkel Pizza      Breakfast and lunch      5
#> 6      6 Güvenç Attila    Ice cream      Lunch only      6

if_else()      test      test TRUE      yes      FALSE      no
age  "five"    "5"          ??      if_else()
```

7.2.2

`read_csv()` CSV

```
read_csv(
  "a,b,c
  1,2,3
  4,5,6"
)
#> # A tibble: 2 x 3
#>   a     b     c
#>   <dbl> <dbl> <dbl>
#> 1     1     2     3
#> 2     4     5     6

  read_csv()                               skip = n  n  comment =
"#"  #

read_csv(
  "The first line of metadata
  The second line of metadata
  x,y,z
  1,2,3",
  skip = 2
)
#> # A tibble: 1 x 3
#>   x     y     z
#>   <dbl> <dbl> <dbl>
#> 1     1     2     3

  read_csv(
    "# A comment I want to skip
    x,y,z
    1,2,3",
    comment = "#"
)
#> # A tibble: 1 x 3
#>   x     y     z
```

```
#>   <dbl> <dbl> <dbl>
#> 1     1     2     3
```

```
col_names = FALSE  read_csv()      X1 Xn
```

In other cases, the data might not have column names. You can use `col_names = FALSE` to tell `read_csv()` not to treat the first row as headings and instead label them sequentially from `X1` to `Xn`:

```
read_csv(
  "1,2,3
  4,5,6",
  col_names = FALSE
)
#> # A tibble: 2 x 3
#>       X1     X2     X3
#>   <dbl> <dbl> <dbl>
#> 1     1     2     3
#> 2     4     5     6
```

```
col_names

read_csv(
  "1,2,3
  4,5,6",
  col_names = c("x", "y", "z")
)
#> # A tibble: 2 x 3
#>       x     y     z
#>   <dbl> <dbl> <dbl>
#> 1     1     2     3
#> 2     4     5     6
```

CSV .csv read_csv()

7.2.3

- `read_csv()` `readr`
- `read_csv2()` ; , , ,
- `read_tsv()` tab-delimited
- `read_delim()`

- `read_fwf()` `fwf_widths()` `fwf_positions()`
- `read_table()`
- `read_log()` Apache

7.2.4

1. “|”
2. `file skip comment read_csv() read_tsv()`
3. `read_fwf()`
4. CSV " " `read_csv()`
 " `read_csv()`
- "x,y\n1,'a,b'"
5. CSV

```
read_csv("a,b\n1,2,3\n4,5,6")
read_csv("a,b,c\n1,2\n1,2,3,4")
read_csv("a,b\n\"1")
read_csv("a,b\n1,2\nna,b")
read_csv("a;b\n1;3")
```
6.
 - a. 1
 - b. 1 2
 - c. 3 2 1
 - d. one, two three.

```
annoying <- tibble(
  `1` = 1:10,
  `2` = `1` * 2 + rnorm(length(`1`))
)
```

7.3

CSV `readr`
 `readr`

7.3.1

readr 1,000²

- F T FALSE TRUE
- 1 -4.5 5e6 Inf
- ISO8601 - @sec-creating-datetime -
-

```
read_csv("logical,numeric,date,string
TRUE,1,2021-01-15,abc
false,4.5,2021-02-15,def
T,Inf,2021-02-16,ghi
")
#> # A tibble: 3 x 4
#>   logical numeric date      string
#>   <lgl>     <dbl> <date>    <chr>
#> 1 TRUE       1 2021-01-15 abc
#> 2 FALSE      4.5 2021-02-15 def
#> 3 TRUE       Inf 2021-02-16 ghi
```

7.3.2

readr NA
CSV

```
simple_csv <- "
x
10
.
20
30"
```

x

² guess_max 1000

```

read_csv(simple_csv)
#> # A tibble: 4 x 1
#>   x
#>   <chr>
#> 1 10
#> 2 .
#> 3 20
#> 4 30

```

readr x

```

.
col_types      CSV

```

```

df <- read_csv(
  simple_csv,
  col_types = list(x = col_double())
)
#> Warning: One or more parsing issues, call `problems()` on your data frame for
#> details, e.g.:
#>   dat <- vroom(...)
#>   problems(dat)

```

```

read_csv()      problems()    :

```

```

problems(df)
#> # A tibble: 1 x 5
#>   row     col expected actual file
#>   <int> <int> <chr>    <chr>  <chr>
#> 1     3     1 a double . /private/var/folders/2m/th7w53zx2fx6gl3g1jcj4l~

```

3 1 readr . . na = "."

```

read_csv(simple_csv, na = ".")
#> # A tibble: 4 x 1
#>   x
#>   <dbl>
#> 1    10
#> 2    NA
#> 3    20
#> 4    30

```

7.3.3

readr :

```

• col_logical() col_double()           readr
• col_integer()
• col_character()
• col_factor(), col_date() col_datetime()      -    @sec-factors
  ???
• col_number()                      @sec-numbers
• col_skip()                         CSV

list()  cols() .default

another_csv <- "
x,y,z
1,2,3"

read_csv(
  another_csv,
  col_types = cols(.default = col_character()))
)
#> # A tibble: 1 x 3
#>   x     y     z
#>   <chr> <chr> <chr>
#> 1 1     2     3

```

Another useful helper is `cols_only()` which will read in only the columns you specify:

```

read_csv(
  another_csv,
  col_types = cols_only(x = col_character()))
)
#> # A tibble: 1 x 1
#>   x
#>   <chr>
#> 1 1

```

7.4

```

1   01-sales.csv 2 02-sales.csv 3 03-sales.csv
read_csv()

sales_files <- c("data/01-sales.csv", "data/02-sales.csv", "data/03-sales.csv")
read_csv(sales_files, id = "file")
#> # A tibble: 19 x 6

```

```
#>   file      month    year brand item    n
#>   <chr>     <chr>    <dbl> <dbl> <dbl> <dbl>
#> 1 data/01-sales.csv January 2019 1 1234 3
#> 2 data/01-sales.csv January 2019 1 8721 9
#> 3 data/01-sales.csv January 2019 1 1822 2
#> 4 data/01-sales.csv January 2019 2 3333 1
#> 5 data/01-sales.csv January 2019 2 2156 9
#> 6 data/01-sales.csv January 2019 2 3987 6
#> # i 13 more rows

  data CSV https://pos.it/r4ds-01-sales, https://pos.it/r4ds-02-sales https://pos.it/r4ds-03-sales :

sales_files <- c(
  "https://pos.it/r4ds-01-sales",
  "https://pos.it/r4ds-02-sales",
  "https://pos.it/r4ds-03-sales"
)
read_csv(sales_files, id = "file")

id      file
list.files() @sec-regular-
expressions

sales_files <- list.files("data", pattern = "sales\\*.csv$", full.names = TRUE)
sales_files
#> [1] "data/01-sales.csv" "data/02-sales.csv" "data/03-sales.csv"
```

7.5

```
readr      write_csv() write_tsv()      x( ) file( )
na

write_csv(students, "students.csv")

csv      CSV      :
students
#> # A tibble: 6 x 5
#>   student_id full_name      favourite_food    meal_plan      age
#>   <dbl> <chr>        <chr>            <fct>          <dbl>
#> 1       1 Sunil Huffmann Strawberry yoghurt Lunch only      4
```

```
#> 2      2 Barclay Lynn   French fries    Lunch only      5
#> 3      3 Jayendra Lyne <NA>           Breakfast and lunch 7
#> 4      4 Leon Rossini  Anchovies     Lunch only      NA
#> 5      5 Chidiegwu Dunkel Pizza    Breakfast and lunch 5
#> 6      6 Güvenç Attila   Ice cream    Lunch only      6
write_csv(students, "students-2.csv")
read_csv("students-2.csv")
#> # A tibble: 6 x 5
#>   student_id full_name   favourite_food meal_plan      age
#>   <dbl> <chr>        <chr>          <chr>        <dbl>
#> 1      1 Sunil Huffmann Strawberry yoghurt Lunch only      4
#> 2      2 Barclay Lynn   French fries    Lunch only      5
#> 3      3 Jayendra Lyne <NA>           Breakfast and lunch 7
#> 4      4 Leon Rossini  Anchovies     Lunch only      NA
#> 5      5 Chidiegwu Dunkel Pizza    Breakfast and lunch 5
#> 6      6 Güvenç Attila   Ice cream    Lunch only      6
```

CSV

1. write_rds() read_rds()	readRDS() saveRDS()	R	RDS
	R		

```
write_rds(students, "students.rds")
read_rds("students.rds")
#> # A tibble: 6 x 5
#>   student_id full_name   favourite_food meal_plan      age
#>   <dbl> <chr>        <chr>          <fct>        <dbl>
#> 1      1 Sunil Huffmann Strawberry yoghurt Lunch only      4
#> 2      2 Barclay Lynn   French fries    Lunch only      5
#> 3      3 Jayendra Lyne <NA>           Breakfast and lunch 7
#> 4      4 Leon Rossini  Anchovies     Lunch only      NA
#> 5      5 Chidiegwu Dunkel Pizza    Breakfast and lunch 5
#> 6      6 Güvenç Attila   Ice cream    Lunch only      6
```

2. arrow	parquet	??	arrow
----------	---------	----	-------

```
library(arrow)
write_parquet(students, "students.parquet")
read_parquet("students.parquet")
#> # A tibble: 6 x 5
#>   student_id full_name   favourite_food meal_plan      age
#>   <dbl> <chr>        <chr>          <fct>        <dbl>
#> 1      1 Sunil Huffmann Strawberry yoghurt Lunch only      4
#> 2      2 Barclay Lynn   French fries    Lunch only      5
```

#> 3	3 Jayendra Lyne	NA	Breakfast and lunch	7
#> 4	4 Leon Rossini	Anchovies	Lunch only	NA
#> 5	5 Chidiegwu Dunkel	Pizza	Breakfast and lunch	5
#> 6	6 Güvenç Attila	Ice cream	Lunch only	6

Parquet RDS R arrow

7.6

R tibble tibble tibble()

```
tibble(
  x = c(1, 2, 5),
  y = c("h", "m", "g"),
  z = c(0.08, 0.83, 0.60)
)
#> # A tibble: 3 x 3
#>   x     y     z
#>   <dbl> <chr> <dbl>
#> 1 1     h     0.08
#> 2 2     m     0.83
#> 3 5     g     0.6
```

tribble() transposed tibble tribble() ~

```
tribble(
  ~x, ~y, ~z,
  1, "h", 0.08,
  2, "m", 0.83,
  5, "g", 0.60
)
#> # A tibble: 3 x 3
#>   x     y     z
#>   <dbl> <chr> <dbl>
#> 1 1     h     0.08
#> 2 2     m     0.83
#> 3 5     g     0.6
```

7.7

<code>read_csv()</code>	CSV	<code>tibble()</code>	<code>tribble()</code>	CSV
Excel	Google	??	??	Parquet
JSON	@sec-scraping			??
		reprex		R

Chapter 8

R

8.1

```
Google      "R"          R      "tidyverse" "ggplot2"
oogle      Sys.setenv(LANGUAGE
= "en")  
Google      Stack Overflow [R]      R  
" R      " R
```

8.2 reprexx

```
Google      reprexx reproducible example      reprexx
reprexx
```

- library() reprexx
- R

- 80% reprexx
- 20%

```
reprexx      reprexx      tidyverse      RStudio
Server Cloud
```

```
y <- 1:4
mean(y)
```

reprex() GitHub :

```
reprex::reprex()
```

RStudio HTML RStudio Viewer
Server Cloud r eprex

RStudio

```
``` r
y <- 1:4
mean(y)
#> [1] 2.5
```

```

Markdown StackOverflow Github
down Github

Mark-

```
y <- 1:4
mean(y)
#> [1] 2.5
```

1.

tidyverse

tidyverse_update()

2. dput() R R mtcars

1. R dput(mtcars)
2. t
3. reprex mtcars <-

3.

-
-
-

R

replices

replices

R

8.3

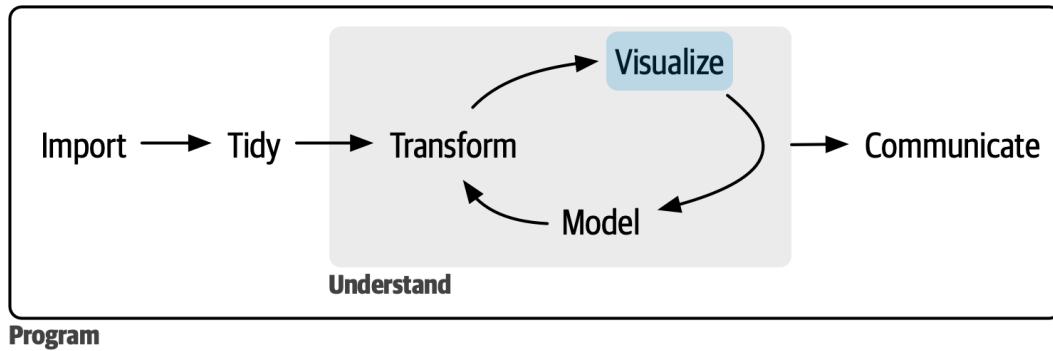
Weekly R tidyverse tidyverse R R

8.4

Whole Game

ggplot2

Part II



8.1: Data visualization is often the first step in data exploration.

- ??
- ??
- , ??

ggplot2 *ggplot2: Elegant graphics for data analysis.*
geoms scales ggplot2 ggplot2 https://exts.ggplot2.tidyverse.org/gallery/

Chapter 9

9.1

```
?? ,           ggplot2  
                  ggplot2  
  
ggplot2           ggplot2
```

9.1.1

```
ggplot2           tidyverse:
```

```
library(tidyverse)
```

9.2

“ ”

— John Tukey

```
ggplot2     mpg      38      234
```

```
mpg  
#> # A tibble: 234 x 11  
#>   manufacturer model displ year cyl trans      drv      cty      hwy fl  
#>   <chr>        <chr> <dbl> <int> <int> <chr>    <chr> <int> <int> <chr>
```

```
#> 1 audi      a4     1.8 1999    4 auto(15)   f     18  29 p
#> 2 audi      a4     1.8 1999    4 manual(m5) f     21  29 p
#> 3 audi      a4     2   2008    4 manual(m6) f     20  31 p
#> 4 audi      a4     2   2008    4 auto(av)   f     21  30 p
#> 5 audi      a4     2.8 1999    6 auto(15)   f     16  26 p
#> 6 audi      a4     2.8 1999    6 manual(m5) f     18  26 p
#> # i 228 more rows
#> # i 1 more variable: class <chr>
```

mpg :

1. displ:

2. hwy: mpg

3. class:

| class | displ | hwy | x | y | color | shape |
|-------|-------|-----|---|---|-------|-------|
|-------|-------|-----|---|---|-------|-------|

```
# Left
ggplot(mpg, aes(x = displ, y = hwy, color = class)) +
  geom_point()

# Right
ggplot(mpg, aes(x = displ, y = hwy, shape = class)) +
  geom_point()
#> Warning: The shape palette can deal with a maximum of 6 discrete values because more
#> than 6 becomes difficult to discriminate
#> i you have requested 7 values. Consider specifying shapes manually if you
#> need that many have them.
#> Warning: Removed 62 rows containing missing values or values outside the scale range
#> (`geom_point()`).
```

class shape

1: 6 6 7

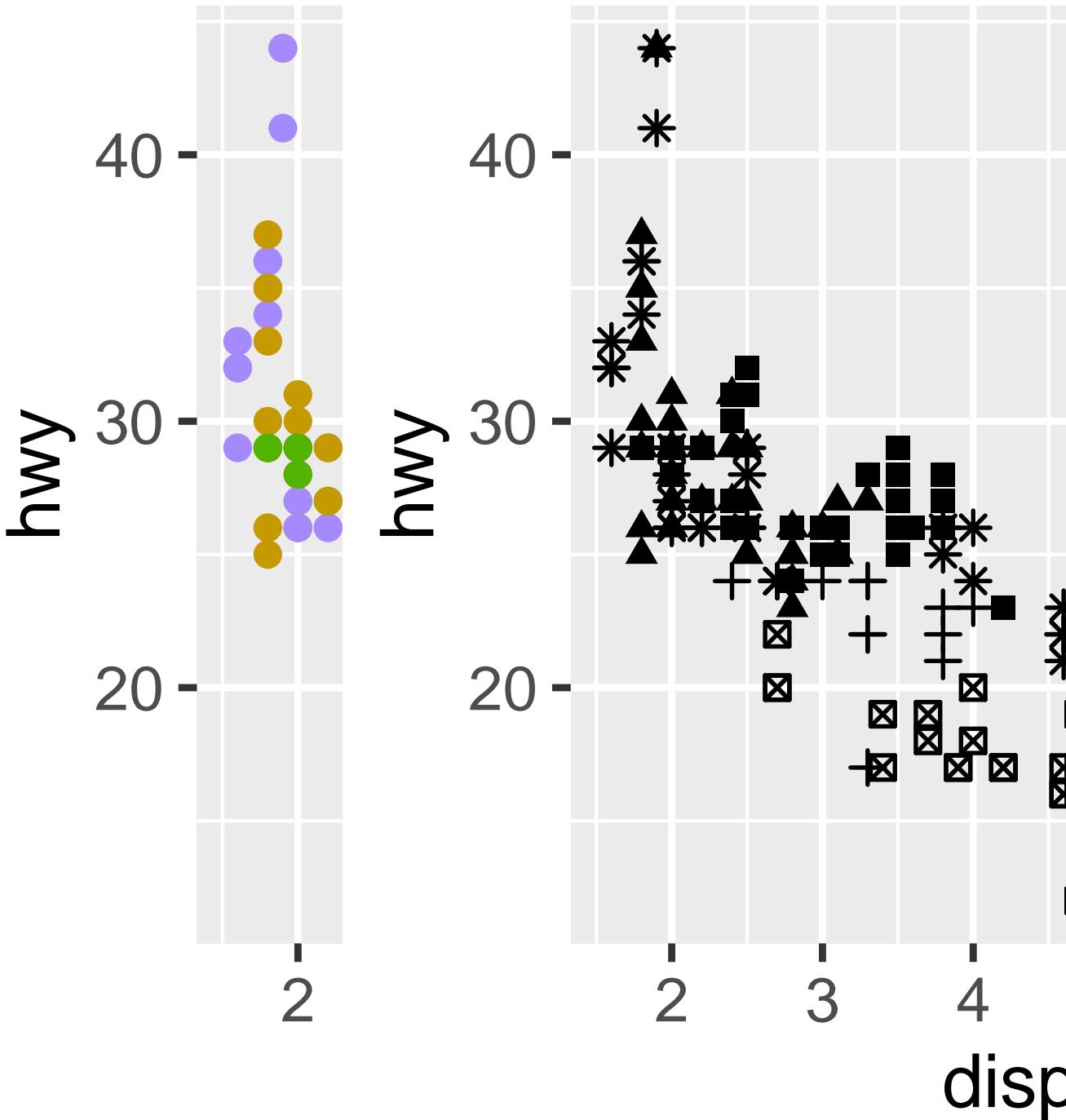
2: 62 geom_point()

ggplot2 — 62 SUV

class size alpha

9.2.

157



```
# Left
ggplot(mpg, aes(x = displ, y = hwy, size = class)) +
  geom_point()
#> Warning: Using size for a discrete variable is not advised.

# Right
ggplot(mpg, aes(x = displ, y = hwy, alpha = class)) +
  geom_point()
#> Warning: Using alpha for a discrete variable is not advised.
```

2

alpha

class size alpha

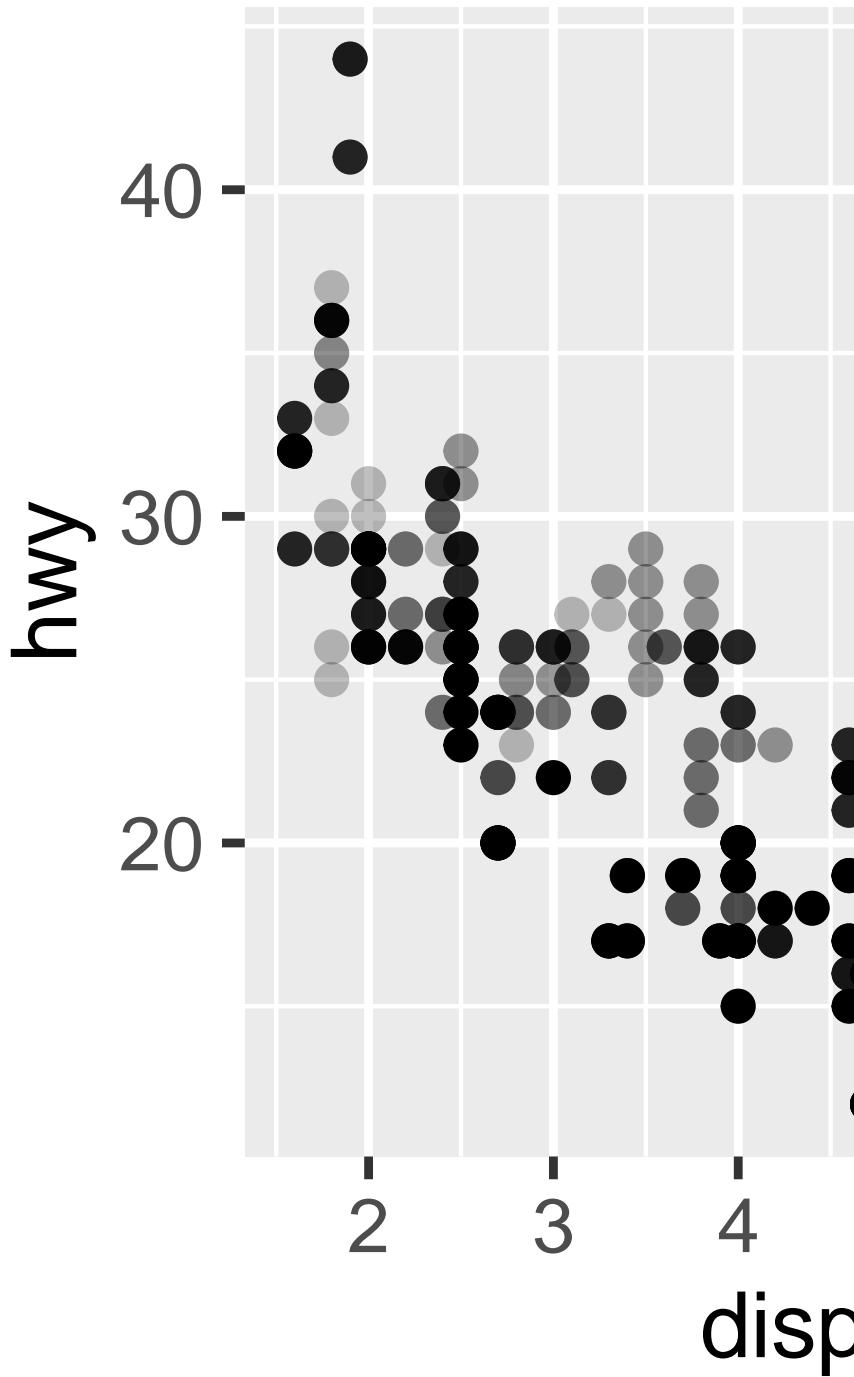
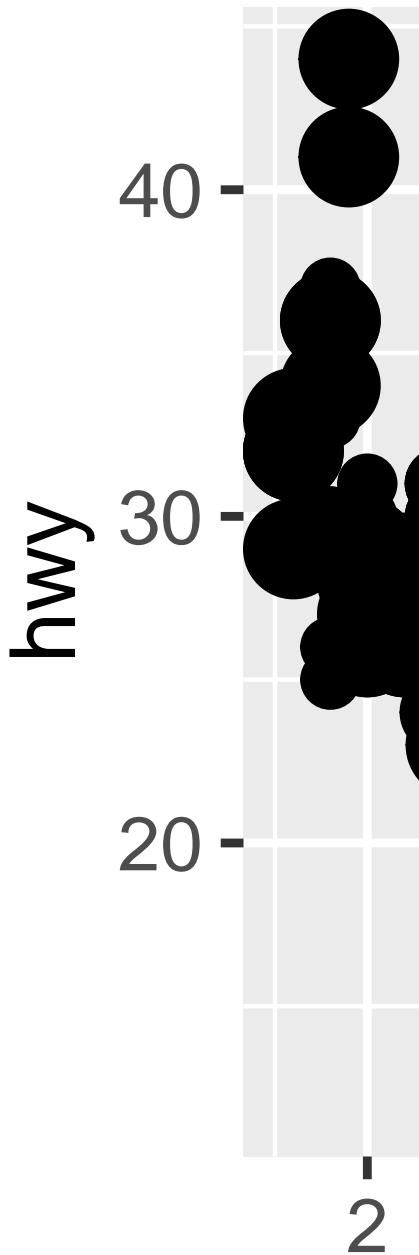
ggplot2 x y ggplot2

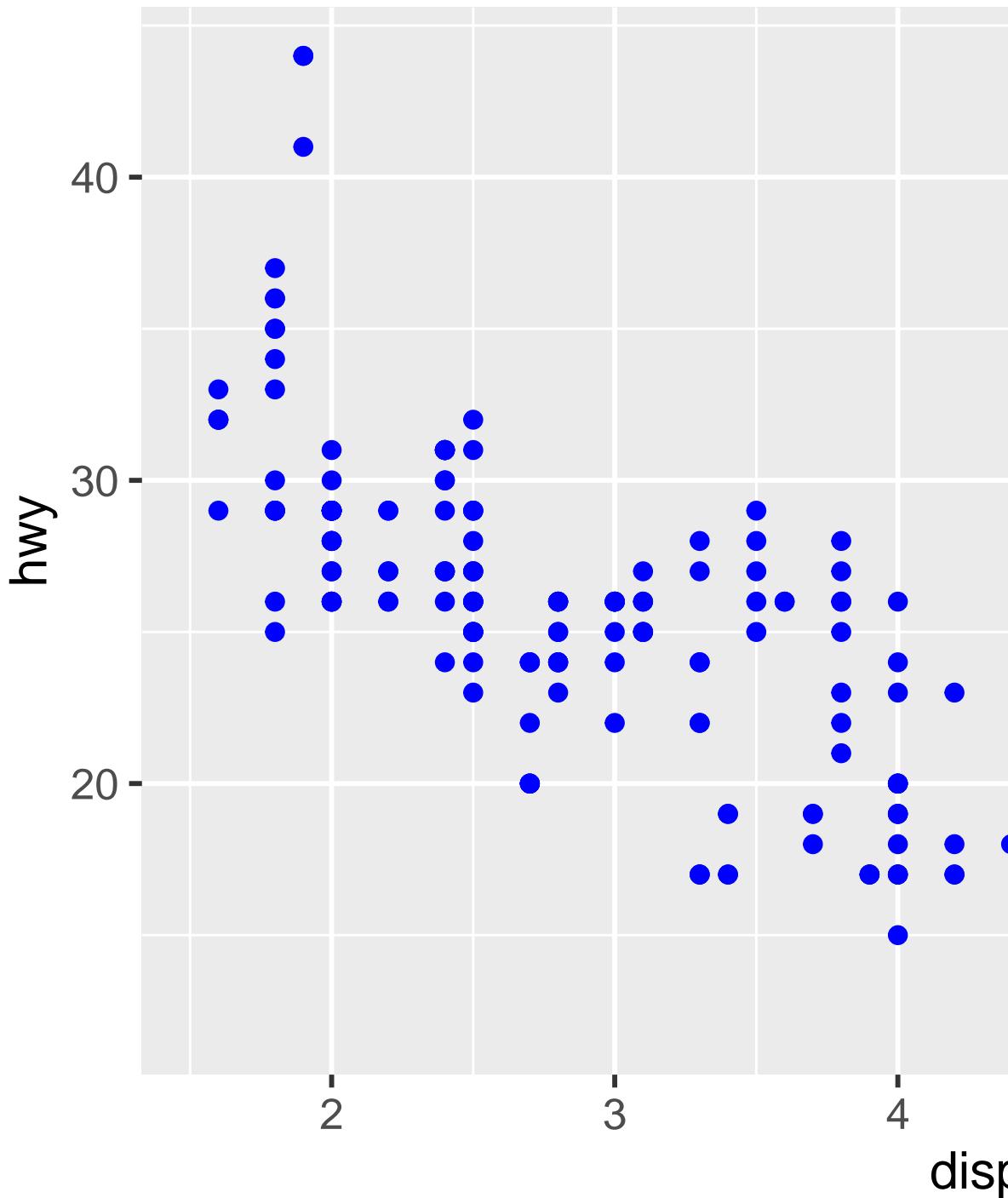
`aes()`

```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point(color = "blue")
```

9.2.

159





```

•      , color = "blue"
•      size = 1
•      shape = 1, ??
```

point geom https://ggplot2.tidyverse.org/articles/ggplot2-specs.html

geom

9.2.1

1. hwy displ

2.

```
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy, color = "blue"))
```

3. stroke aesthetic ?geom_point

4. aes(color = displ < 5) x y

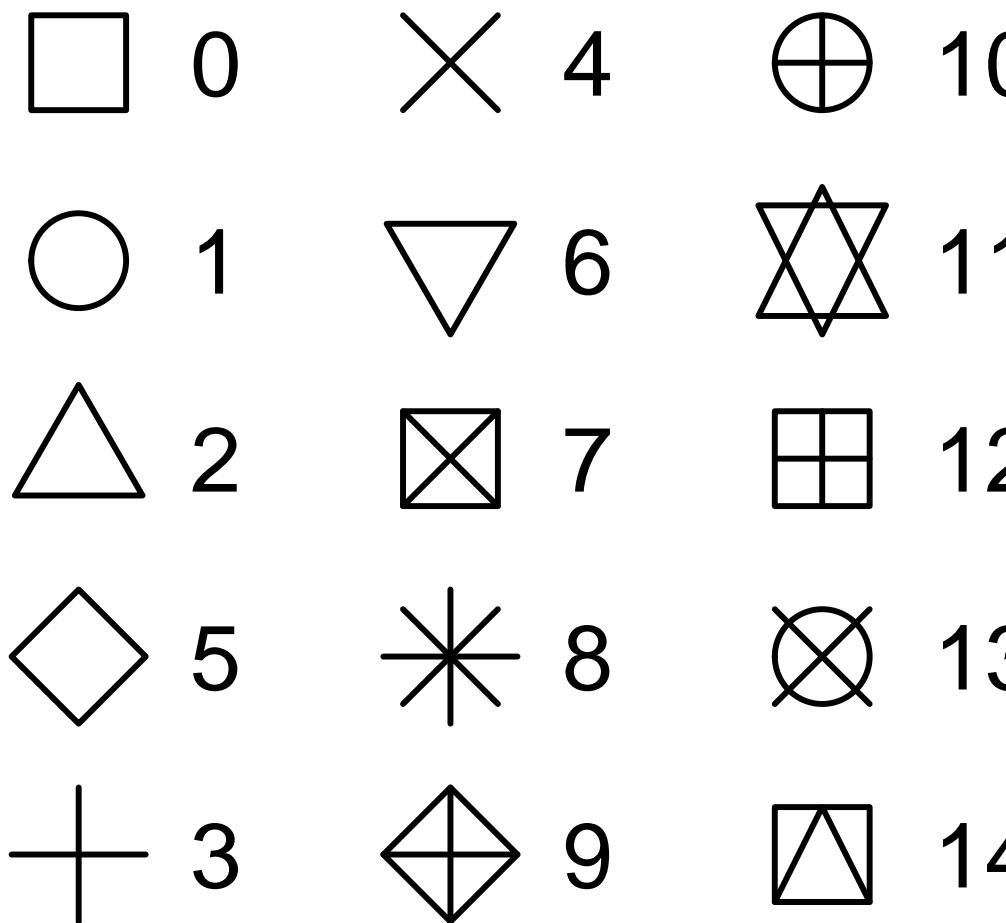
9.3

| | | | |
|------|----------|------------|--------|
| x y | geom | point geom | smooth |
| geom | geom | geom | |
| | ggplot() | | |

```
# Left
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point()

# Right
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_smooth()
#> `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

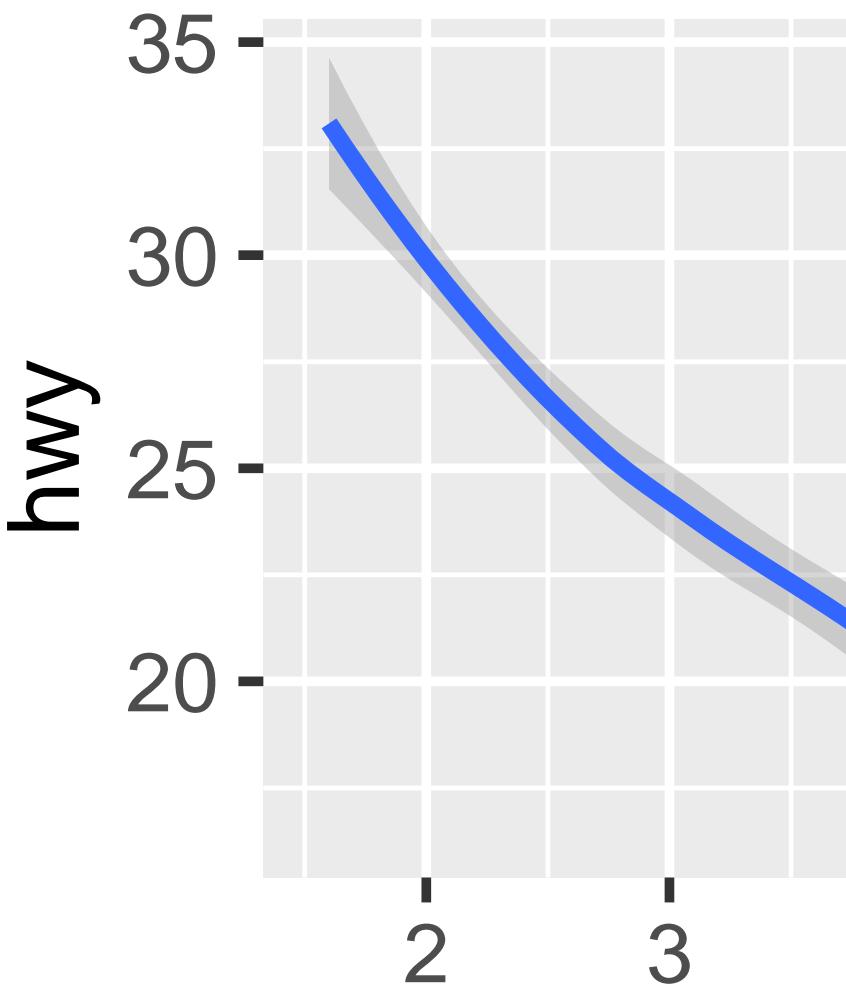
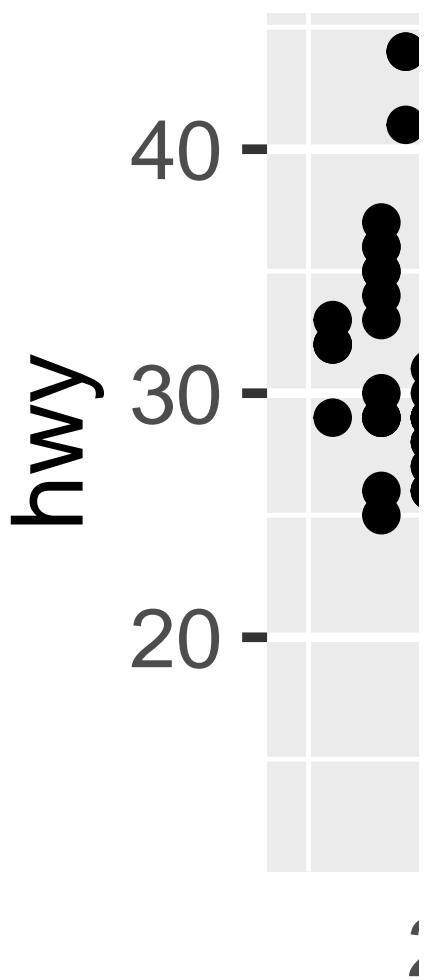
| | | | | | |
|---------|---------|------|---------------|-----------|------|
| ggplot2 | geom | geom | ggplot() | aesthetic | geom |
| " " | ggplot2 | | geom_smooth() | | |



9.1: R has 25 built-in shapes that are identified by numbers. There are some seeming duplicates: for example, 0, 15, and 22 are all squares. The difference comes from the interaction of the `color` and `fill` aesthetics. The hollow shapes (0–14) have a border determined by `color`; the solid shapes (15–20) are filled with `color`; the filled shapes (21–24) have a border of `color` and are filled with `fill`. Shapes are arranged to keep similar shapes next to each other.

9.3.

163



```
# Left
ggplot(mpg, aes(x = displ, y = hwy, shape = drv)) +
  geom_smooth()

# Right
ggplot(mpg, aes(x = displ, y = hwy, linetype = drv)) +
  geom_smooth()
```

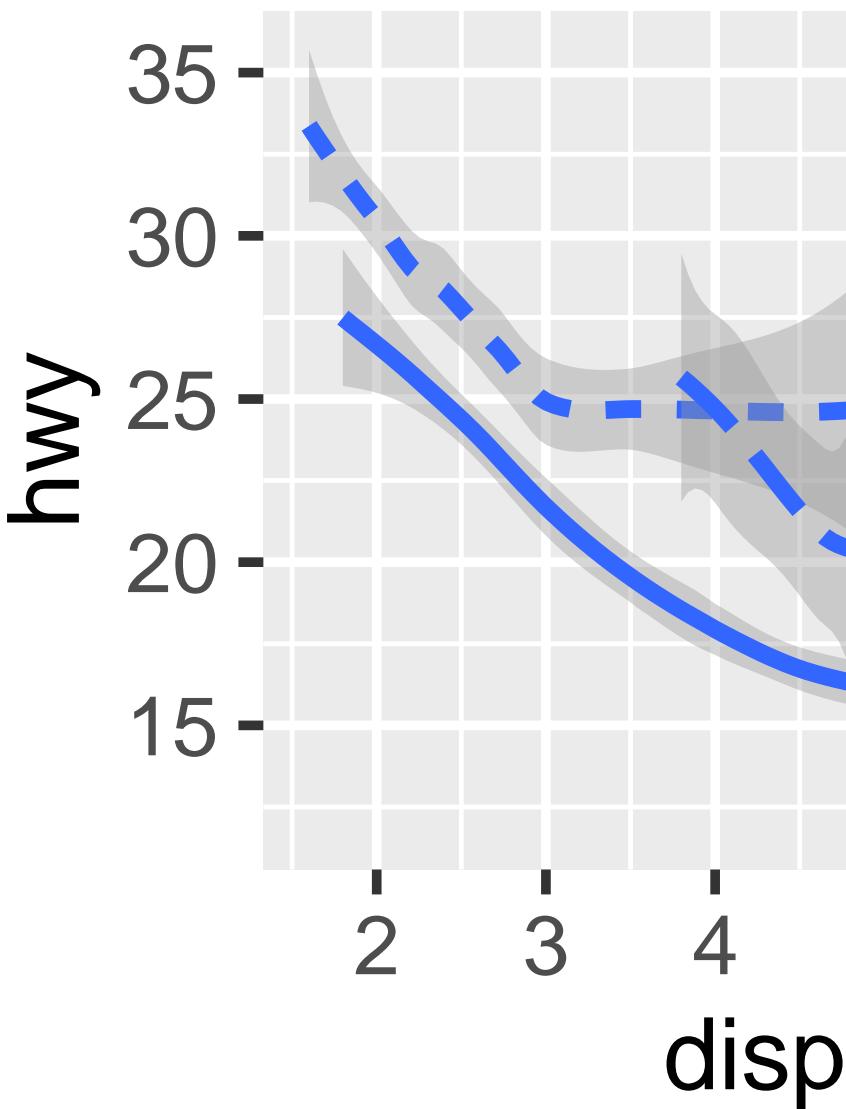
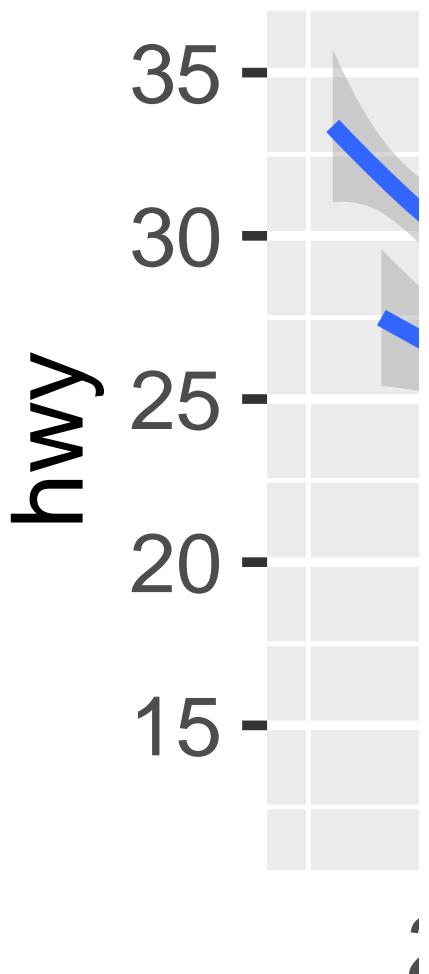
geom_smooth() drv
4 f
4 r

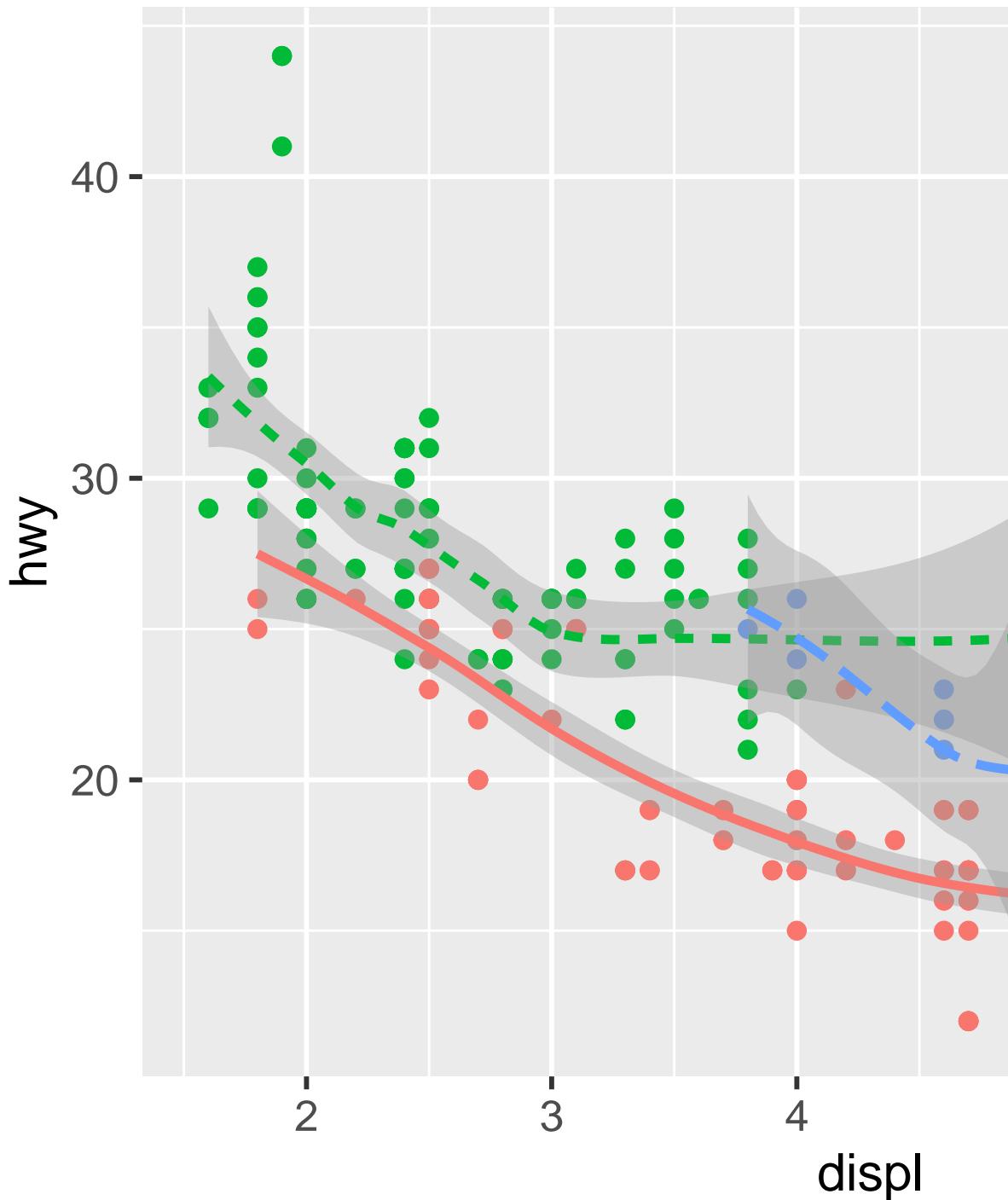
drv

```
ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point() +
  geom_smooth(aes(linetype = drv))
```

9.3.

165





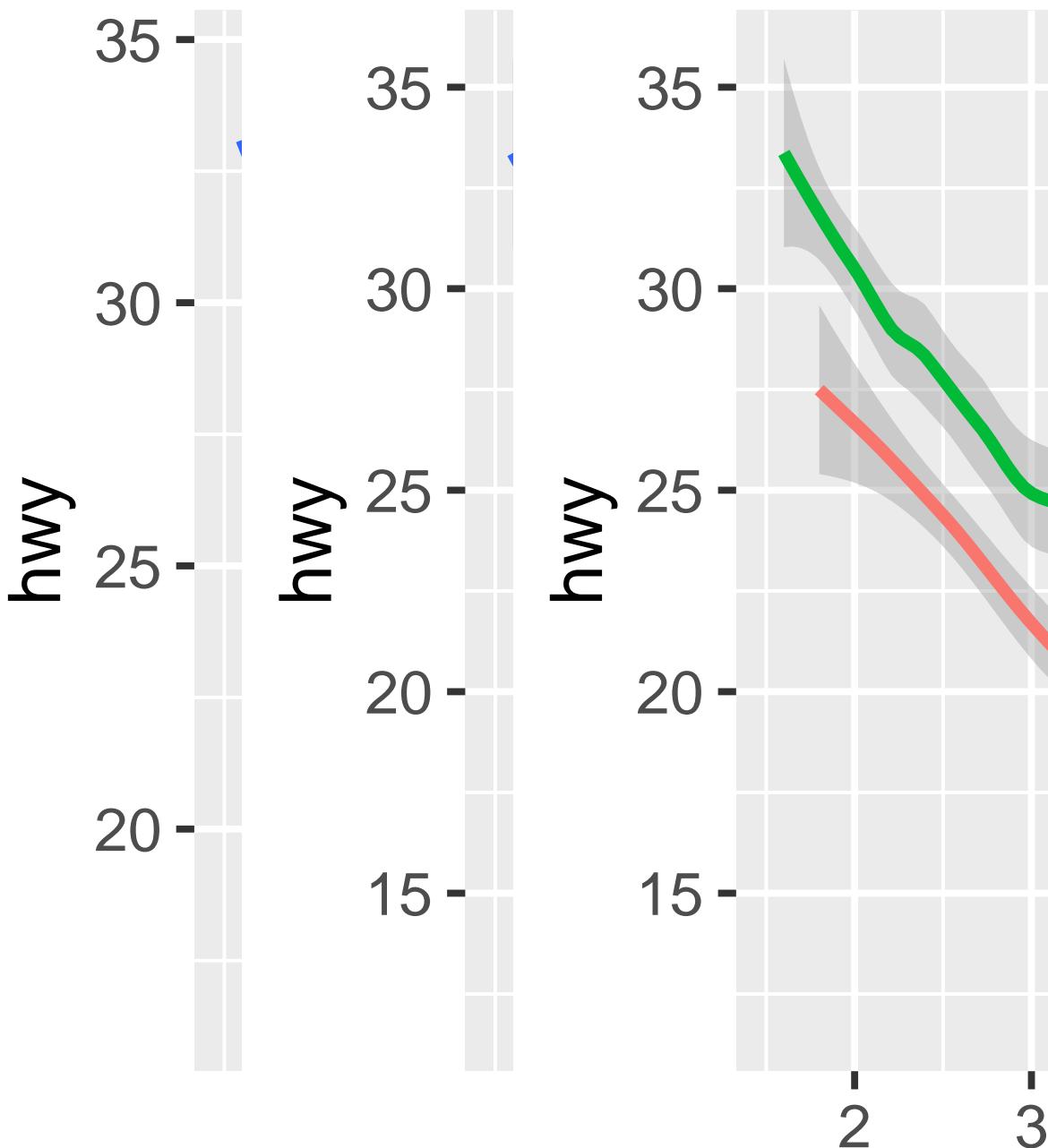
geoms

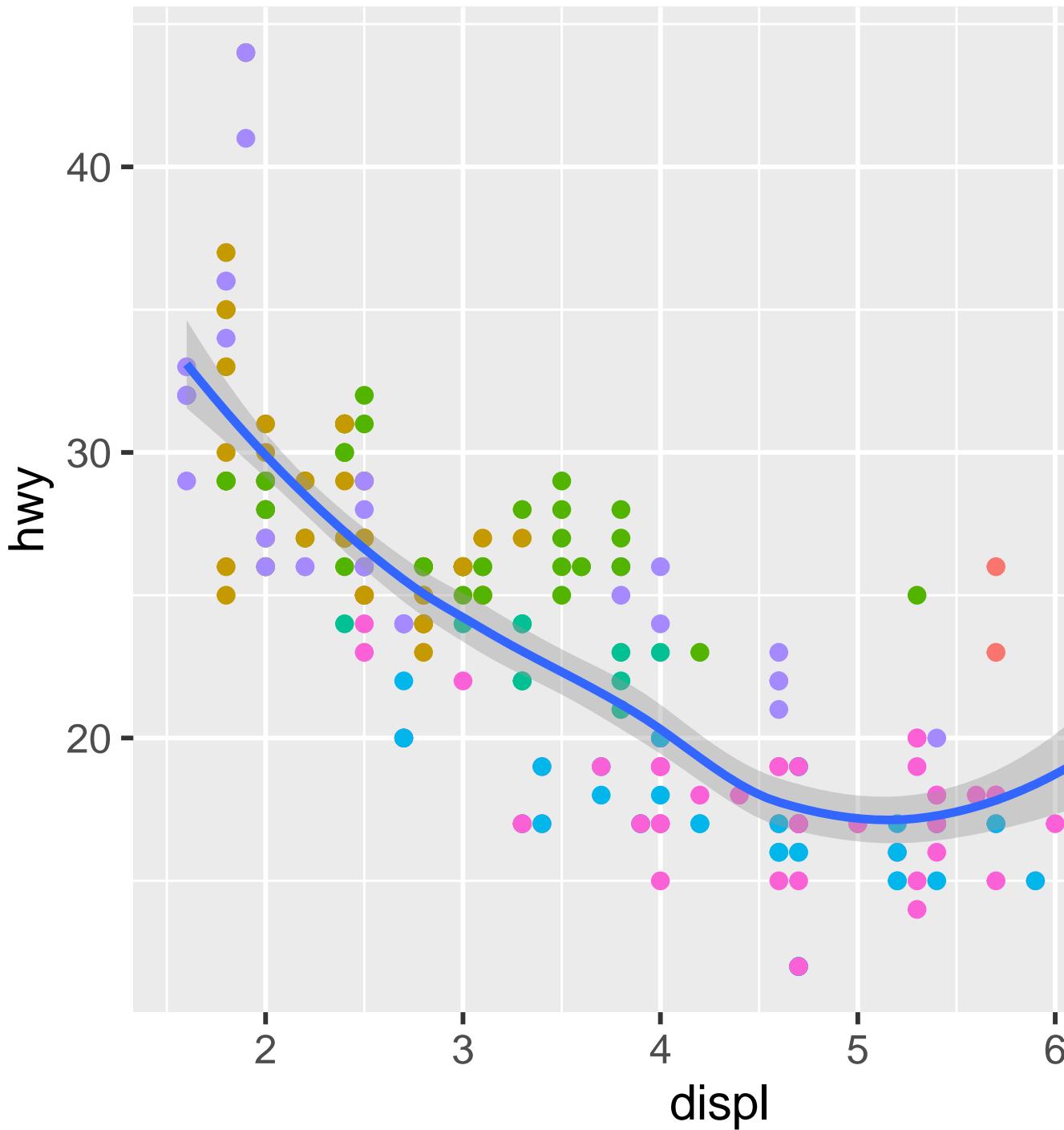
```
geom_smooth()  
ggplot2          group      g gplot2  
                    group
```

```
# Left  
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_smooth()  
  
# Middle  
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_smooth(aes(group = drv))  
  
# Right  
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_smooth(aes(color = drv), show.legend = FALSE)
```

geom ggplot2

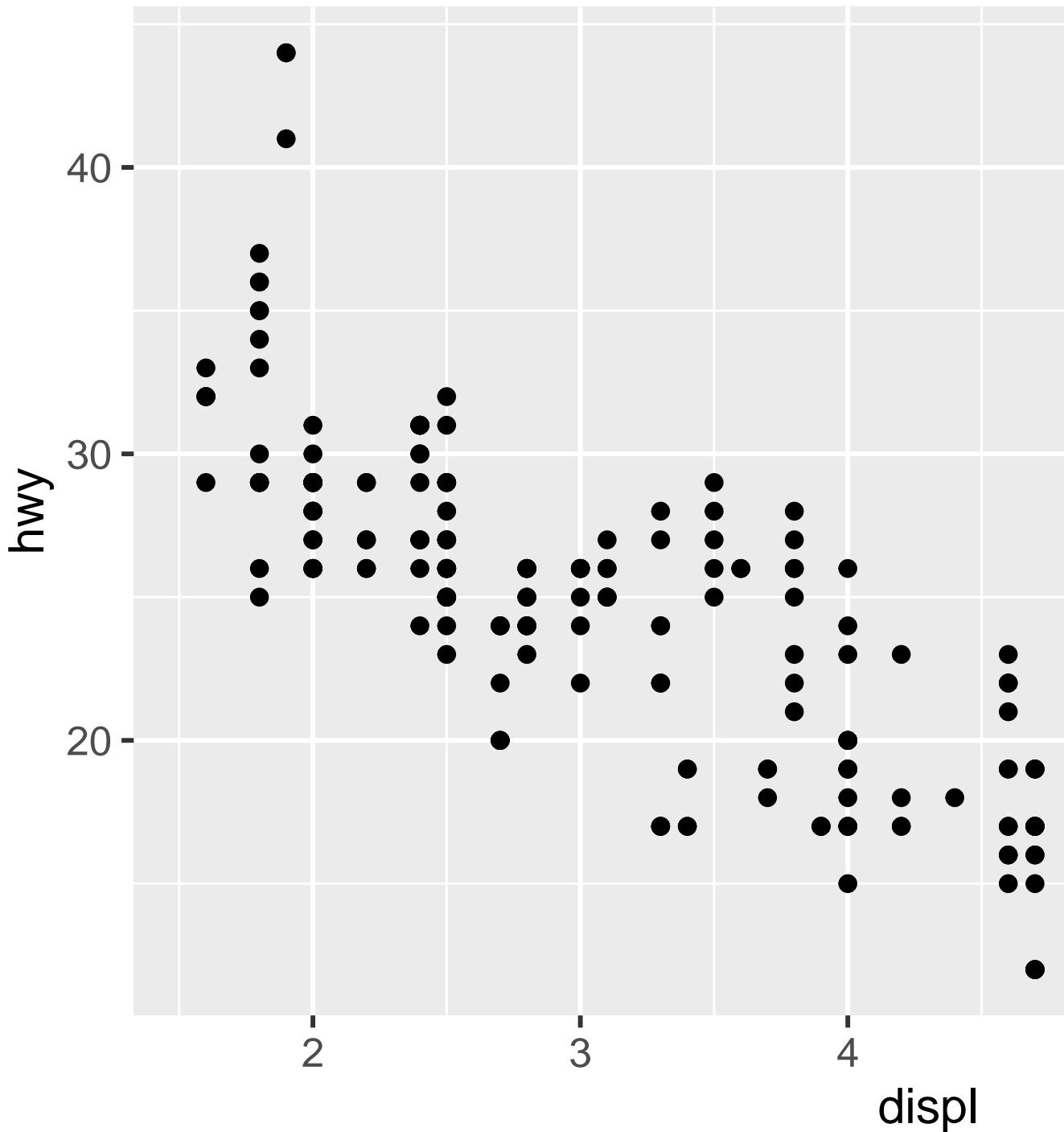
```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point(aes(color = class)) +  
  geom_smooth()
```





ment ggplot() global data argument

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  geom_point(
    data = mpg |> filter(class == "2seater"),
    color = "red"
  ) +
  geom_point(
    data = mpg |> filter(class == "2seater"),
    shape = "circle open", size = 3, color = "red"
  )
}
```



```
geoms ggplot2           geom           geoms
```

```
# Left
ggplot(mpg, aes(x = hwy)) +
  geom_histogram(binwidth = 2)

# Middle
ggplot(mpg, aes(x = hwy)) +
  geom_density()

# Right
ggplot(mpg, aes(x = hwy)) +
  geom_boxplot()
```

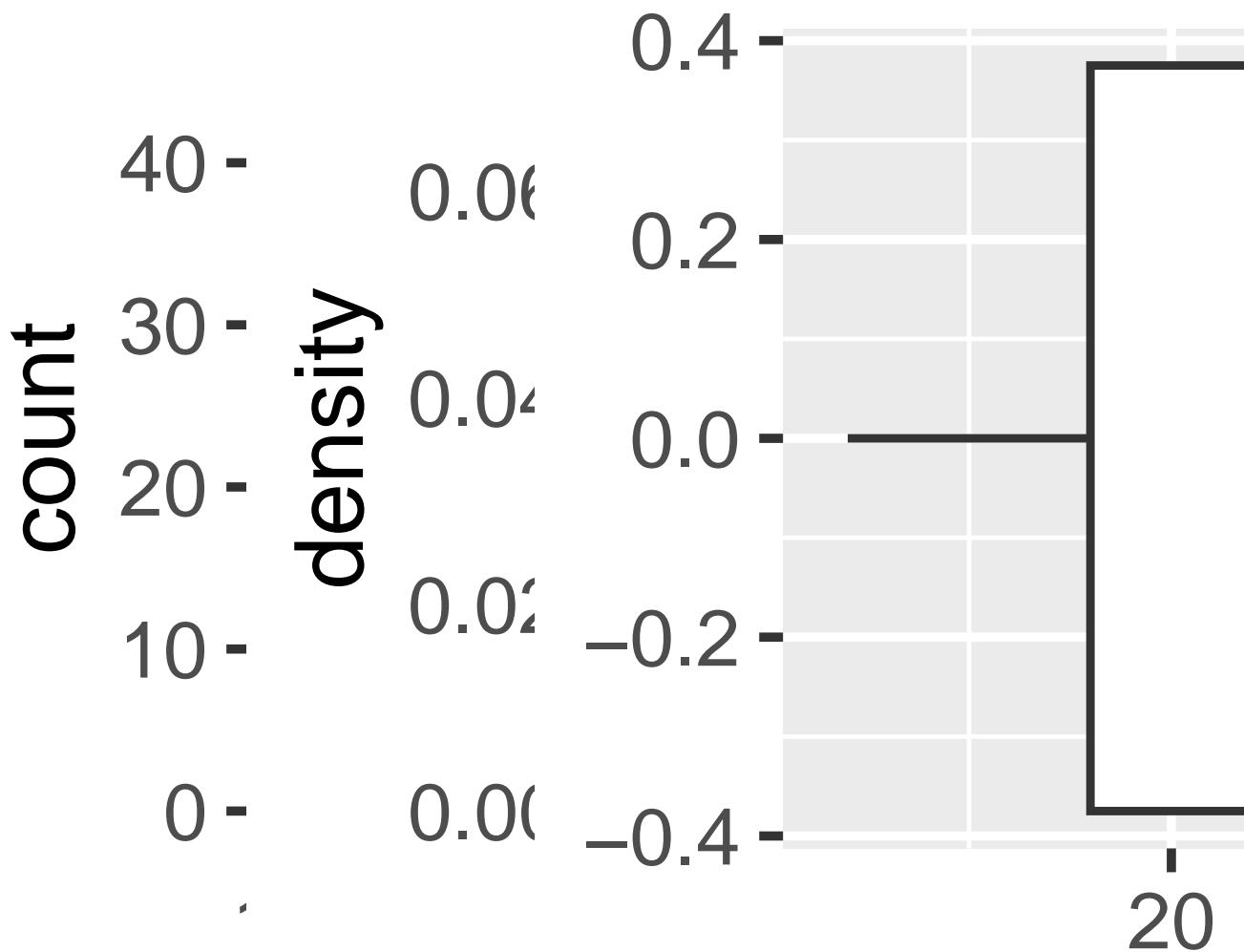
| | | | | | |
|--------------------------------|----|-------|----------|-------------------------------|------------------|
| ggplot2 | 40 | geoms | geom | https://exts. | |
| ggplot2.tidyverse.org/gallery/ | | | ggridges | https://wilkelab.org/ggridges | ridgeline |
| plots | | | geom | geom_density_ridges() | drv y fill color |
| = 0.5 | | | | | |

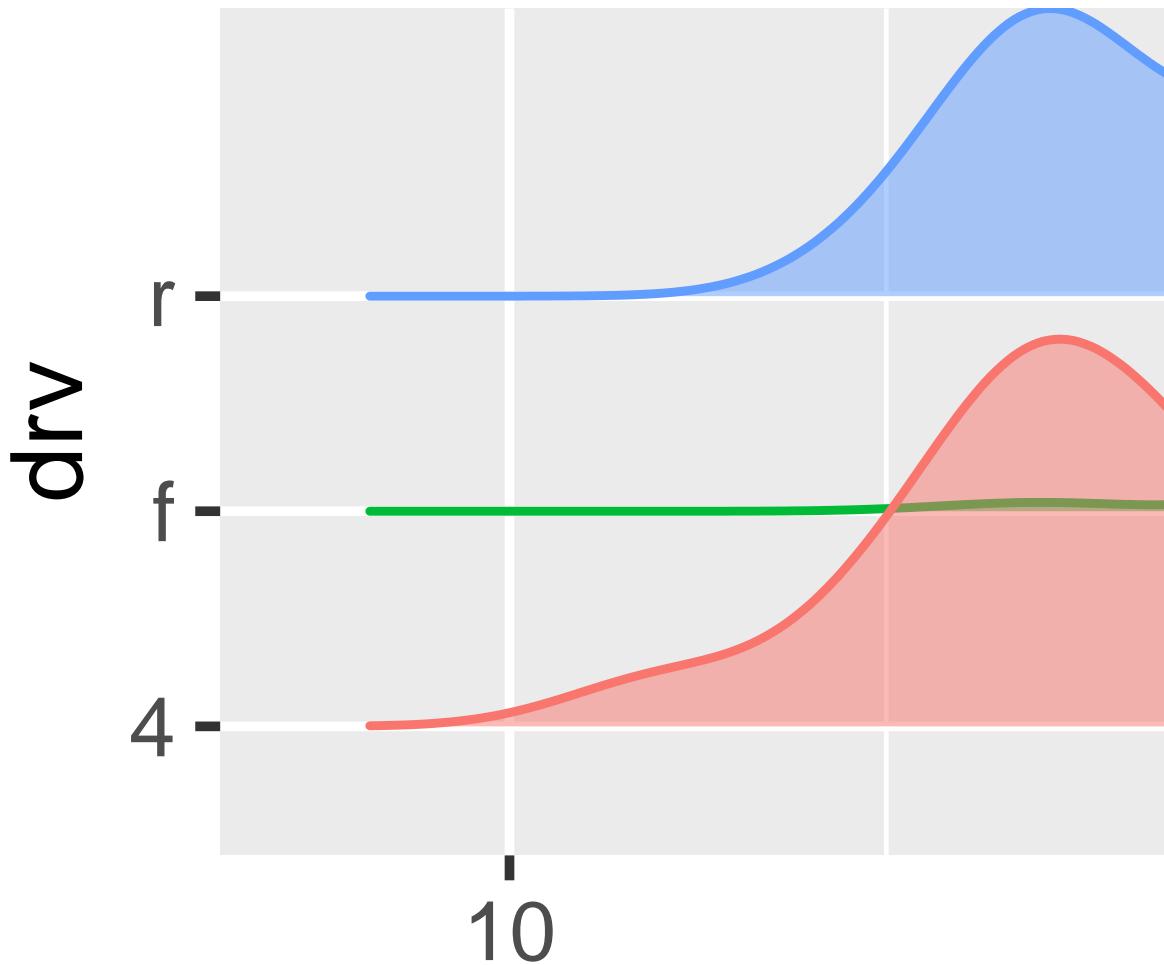
```
library(ggridges)

ggplot(mpg, aes(x = hwy, y = drv, fill = drv, color = drv)) +
  geom_density_ridges(alpha = 0.5, show.legend = FALSE)
#> Picking joint bandwidth of 1.28
```

9.3.

173





`ggplot2` `geoms` <https://ggplot2.tidyverse.org/reference/geom.html>

9.3.1

1. geom
 2. show.legend

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_smooth(aes(color = drv), show.legend = FALSE)
```

show.legend = FALSE ? ? ?

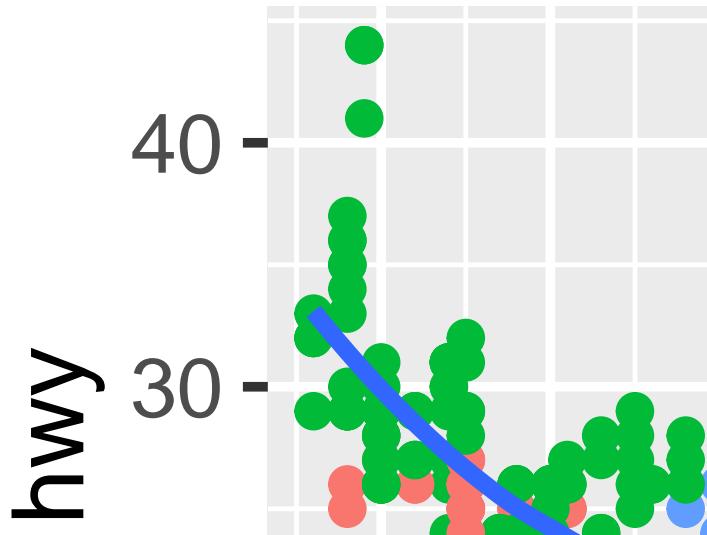
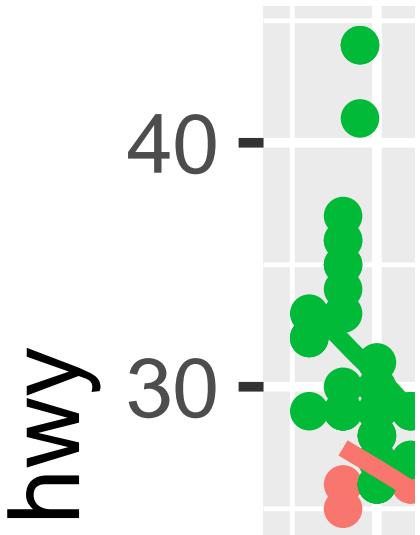
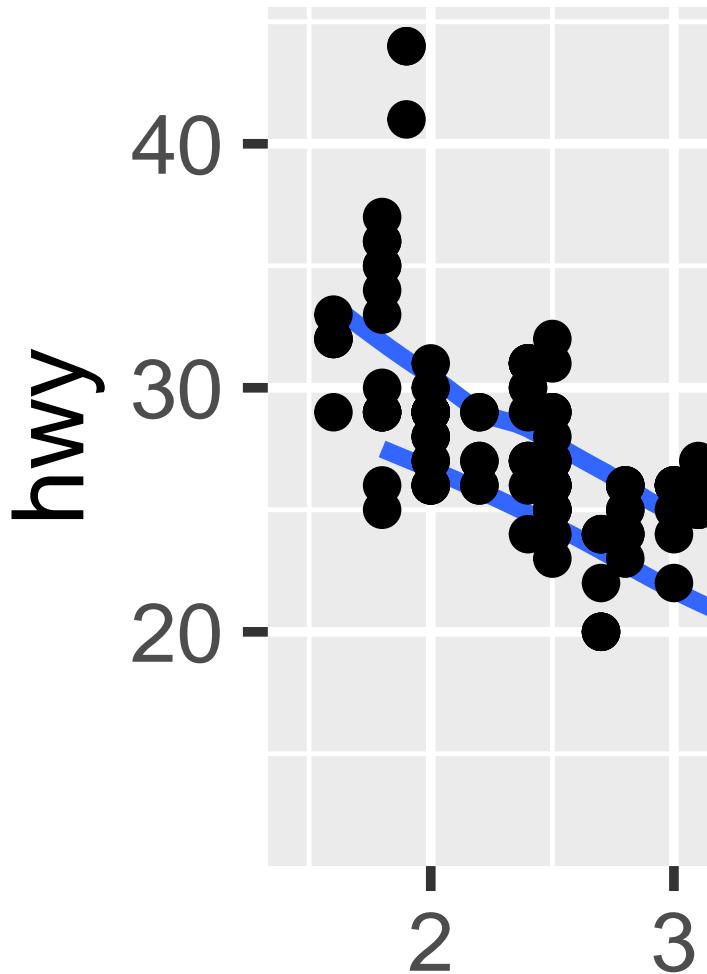
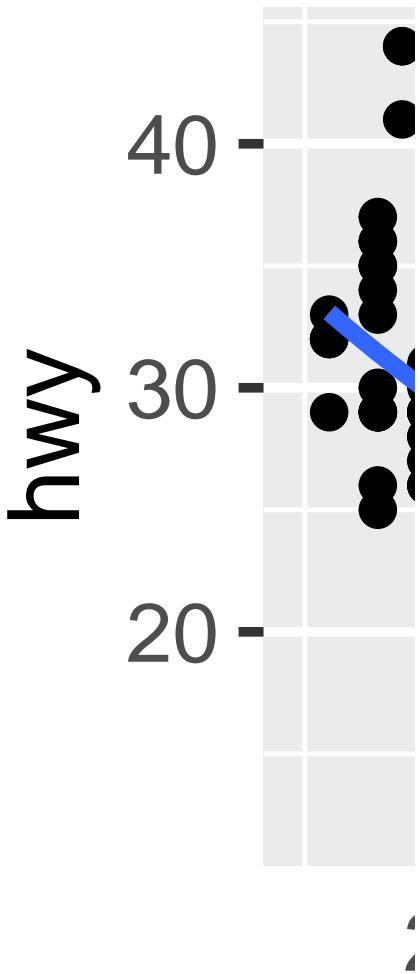
3. geom_smooth() se ?

4. Rdrv

9.4

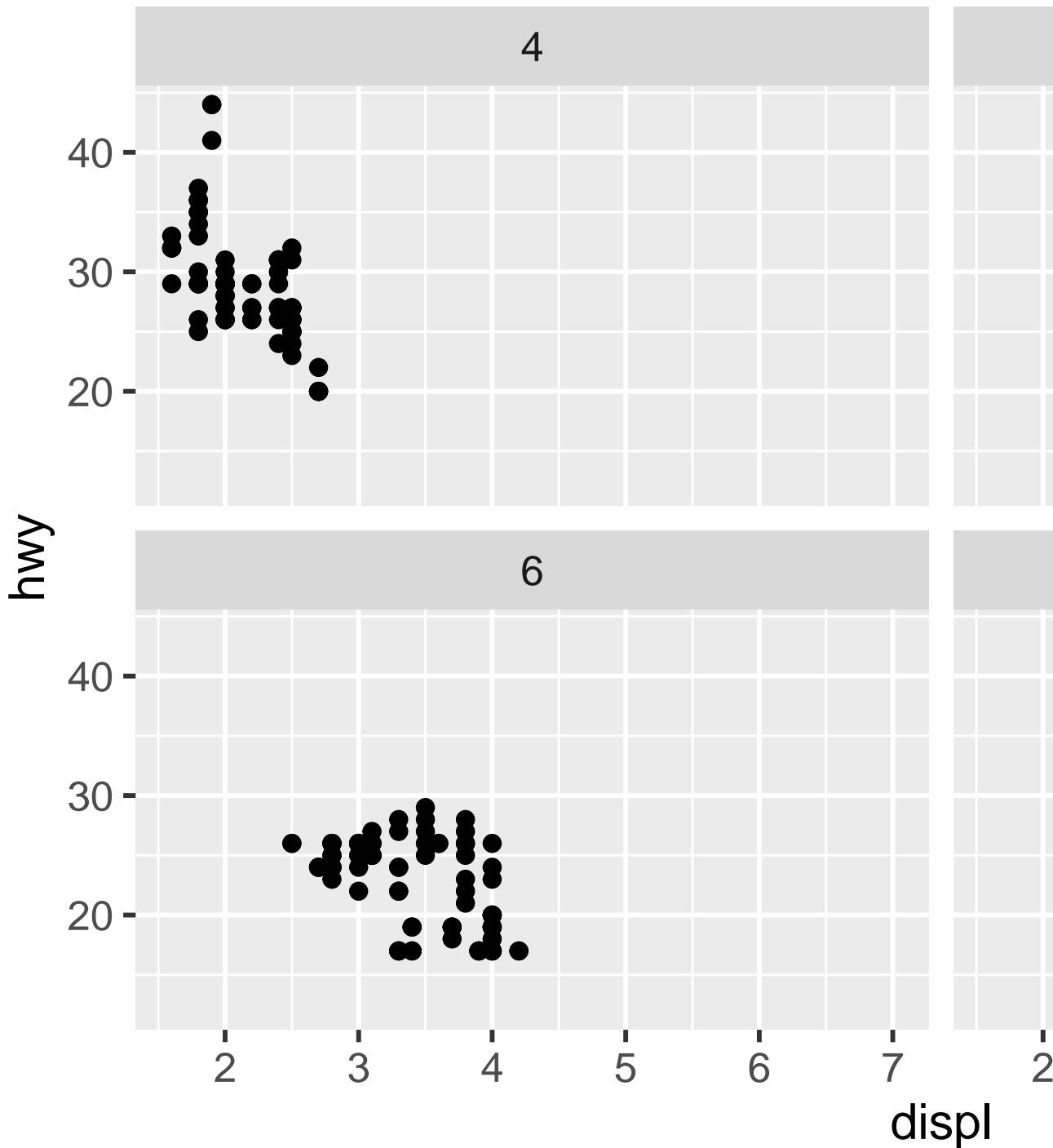
?? facet_wrap()

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  facet_wrap(~cyl)
```



9.4.

177

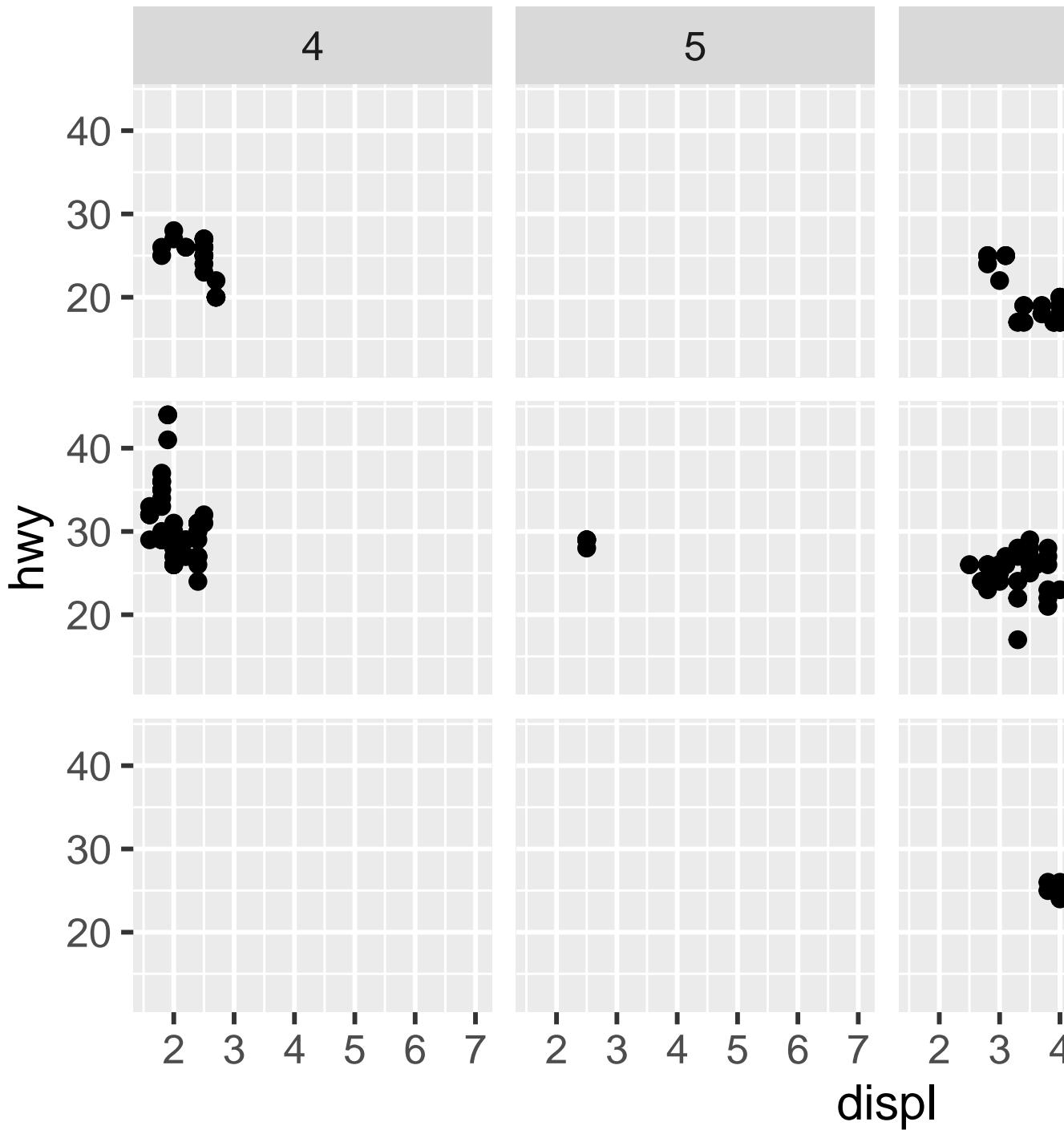


```
facet_wrap()  facet_grid() facet_grid()  
~ cols                                                 rows
```

```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point() +  
  facet_grid(drv ~ cyl)
```

9.4.

179

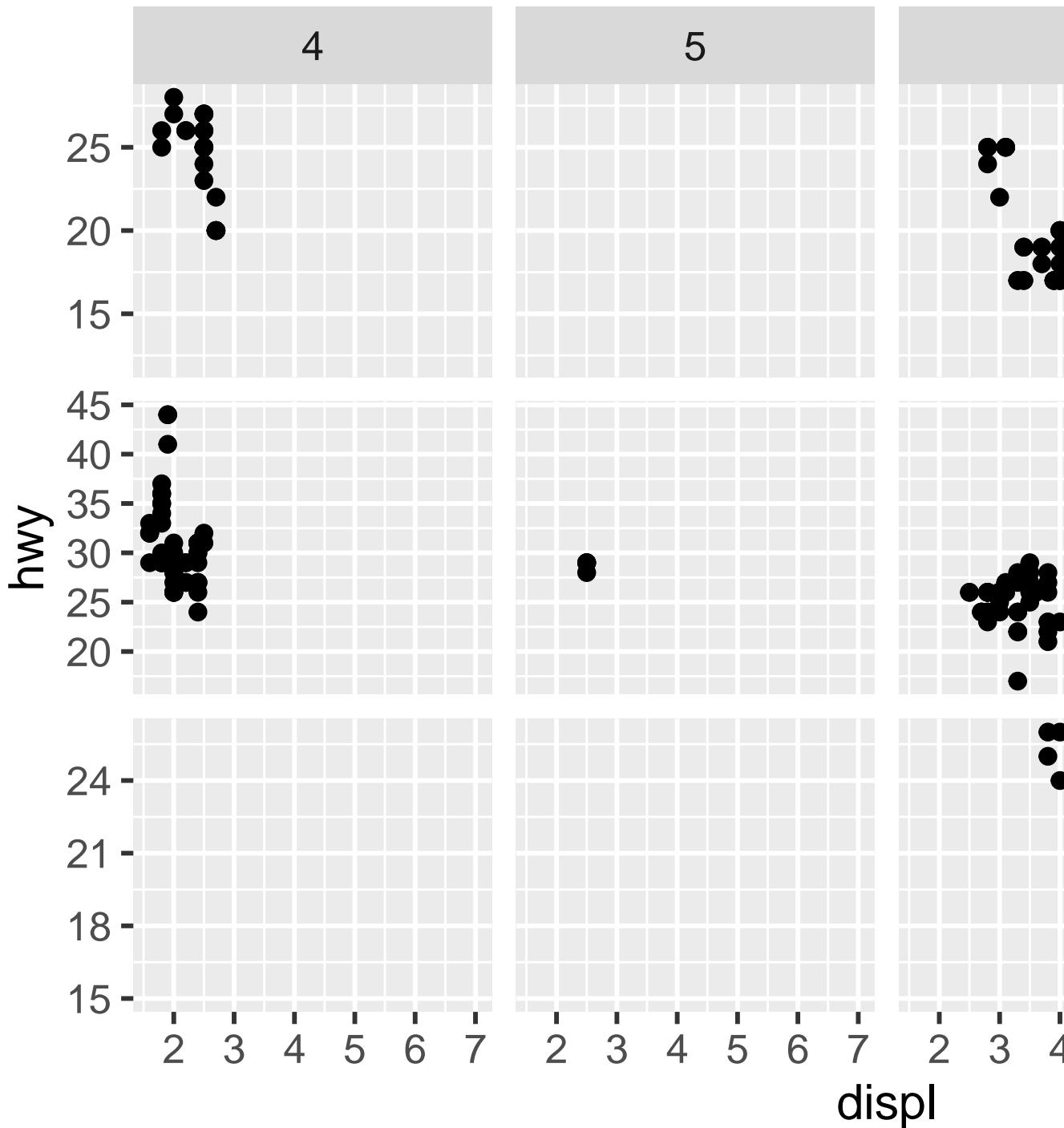


```
x y           scales "free"      "free_x"    "fri
```

```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point() +  
  facet_grid(drv ~ cyl, scales = "free_y")
```

9.4.

181



9.4.1

1.
 2. facet_grid(drv ~ cyl)

```
ggplot(mpg) +
  geom_point(aes(x = drv, y = cyl))
```

3. .

```
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy)) +
  facet_grid(drv ~ .)

ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy)) +
  facet_grid(. ~ cyl)
```

4.

```
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy)) +
  facet_wrap(~ class, nrow = 2)
```

 faceting color aesthetic

5. ?facet_wrap nrow n col facet_grid() nrow ncol
 6. displ

```
ggplot(mpg, aes(x = displ)) +
  geom_histogram() +
  facet_grid(drv ~ .)

ggplot(mpg, aes(x = displ)) +
  geom_histogram() +
  facet_grid(. ~ drv)
```

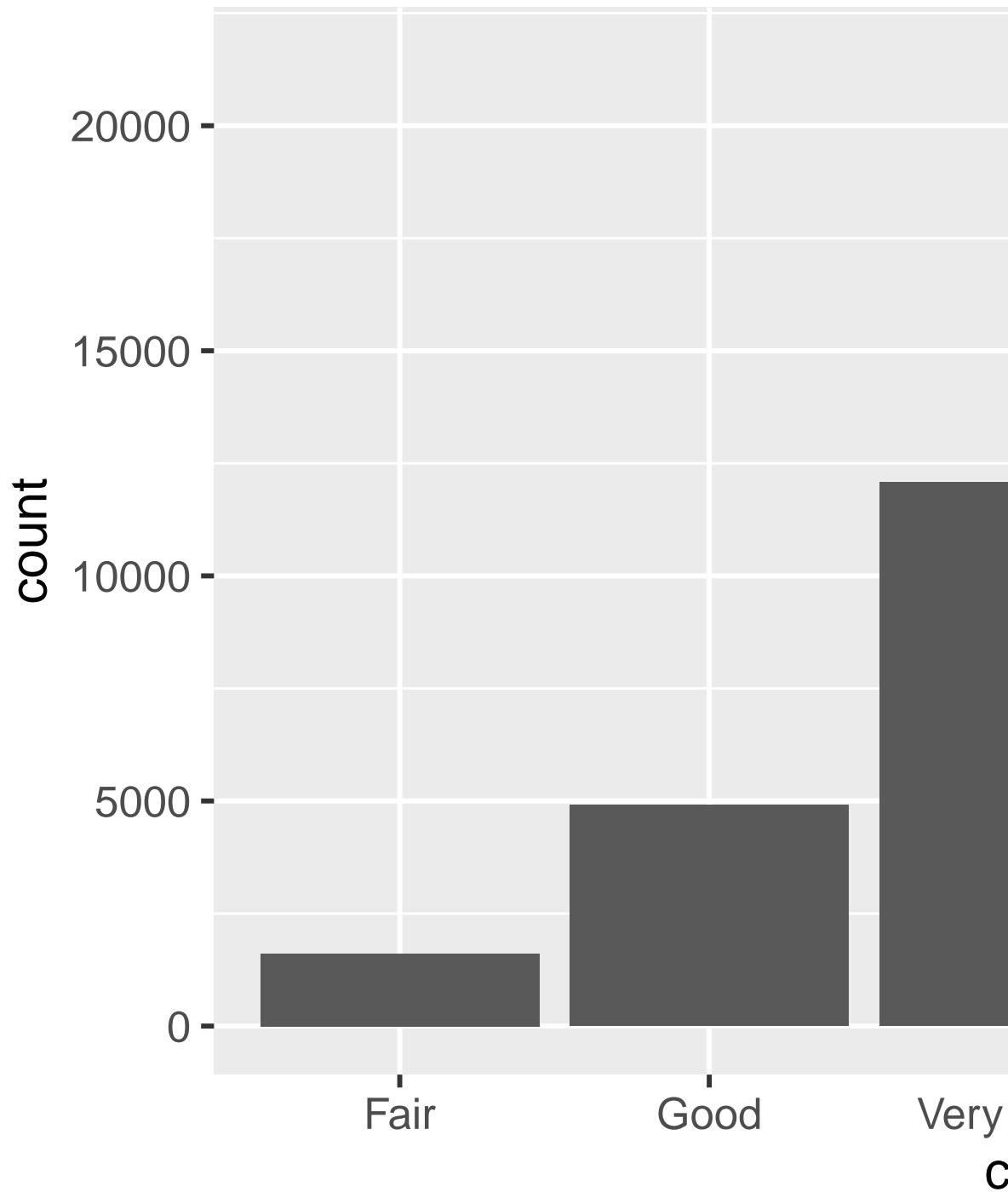
7. facet_wrap()

```
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy)) +
  facet_grid(drv ~ .)
```

9.5

```
geom_bar() geom_col()      diamonds   cut      diamonds   ggplot2
54,000        price    carat    color    clarity   cut
```

```
ggplot(diamonds, aes(x = cut)) +
  geom_bar()
```



```
x      diamonds      cut   y      count  count  diamonds      c ount
```

-
- smoothers
- five-number summary

stat statistical transformation @ fig-vis-stat-bar **geom_bar()**

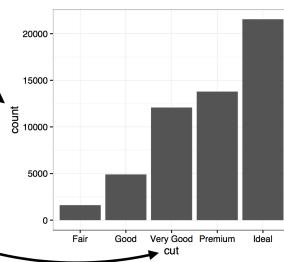
1. **geom_bar()** begins with the **diamonds** data set

| carat | cut | color | clarity | depth | table | price | x | y | z |
|-------|---------|-------|---------|-------|-------|-------|------|------|------|
| 0.23 | Ideal | E | SI2 | 61.5 | 55 | 326 | 3.95 | 3.98 | 2.43 |
| 0.21 | Premium | E | SI1 | 59.8 | 61 | 326 | 3.89 | 3.84 | 2.31 |
| 0.23 | Good | E | VS1 | 56.9 | 65 | 327 | 4.05 | 4.07 | 2.31 |
| 0.29 | Premium | I | VS2 | 62.4 | 58 | 334 | 4.20 | 4.23 | 2.63 |
| 0.31 | Good | J | SI2 | 63.3 | 58 | 335 | 4.34 | 4.35 | 2.75 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

2. **geom_bar()** transforms the data with the "count" stat, which returns a data set of cut values and counts.

| cut | count | prop |
|-----------|-------|------|
| Fair | 1610 | 1 |
| Good | 4906 | 1 |
| Very Good | 12082 | 1 |
| Premium | 13791 | 1 |
| Ideal | 21551 | 1 |

3. **geom_bar()** uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.



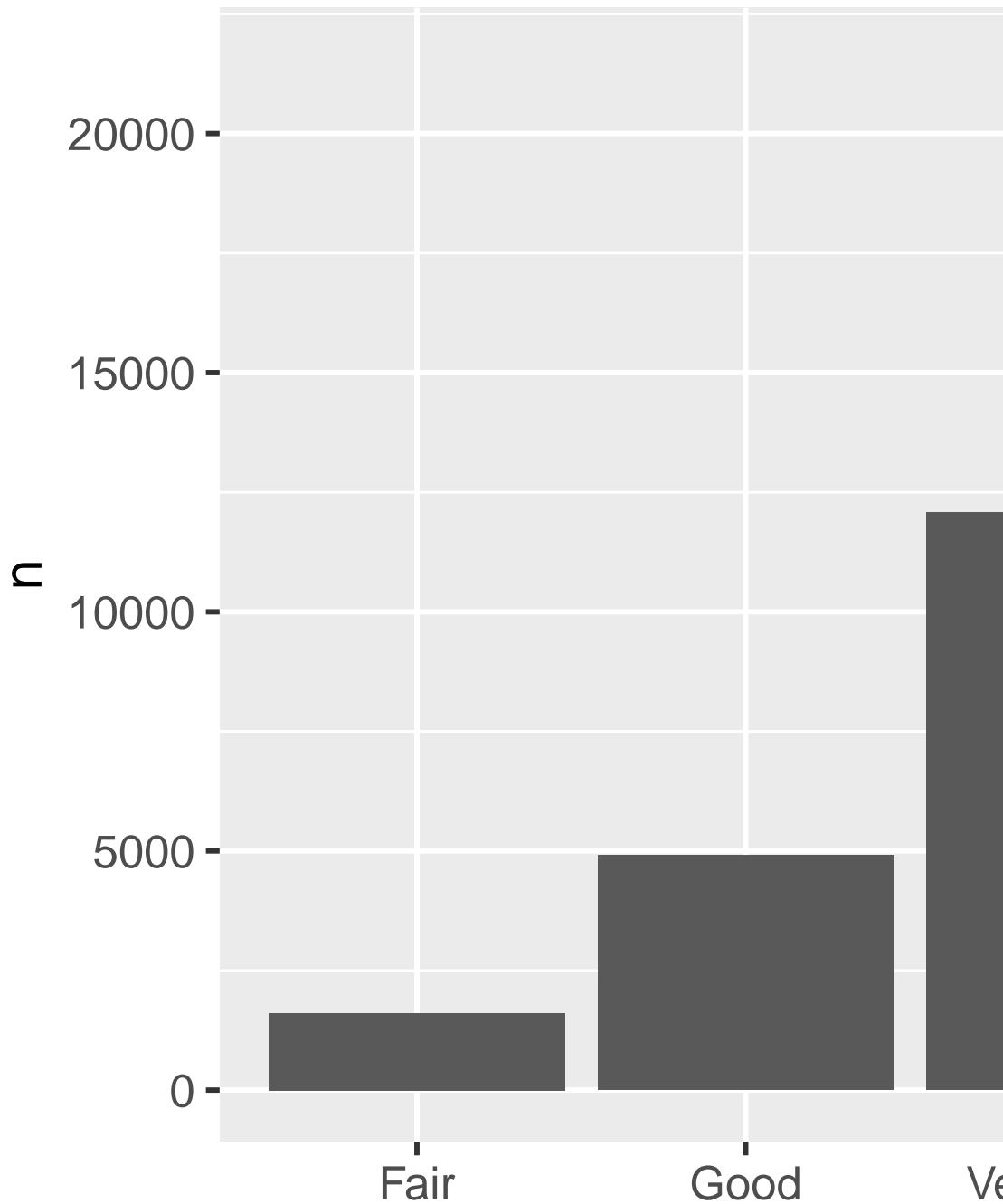
9.2: When creating a bar chart we first start with the raw data, then aggregate it to count the number of observations in each bar, and finally map those computed variables to plot aesthetics.

```
stat      geom      ?geom_bar stat    "count"  geom_bar()  stat_count() s
stat_count()  geom_bar()           "Computed variables"  count prop
```

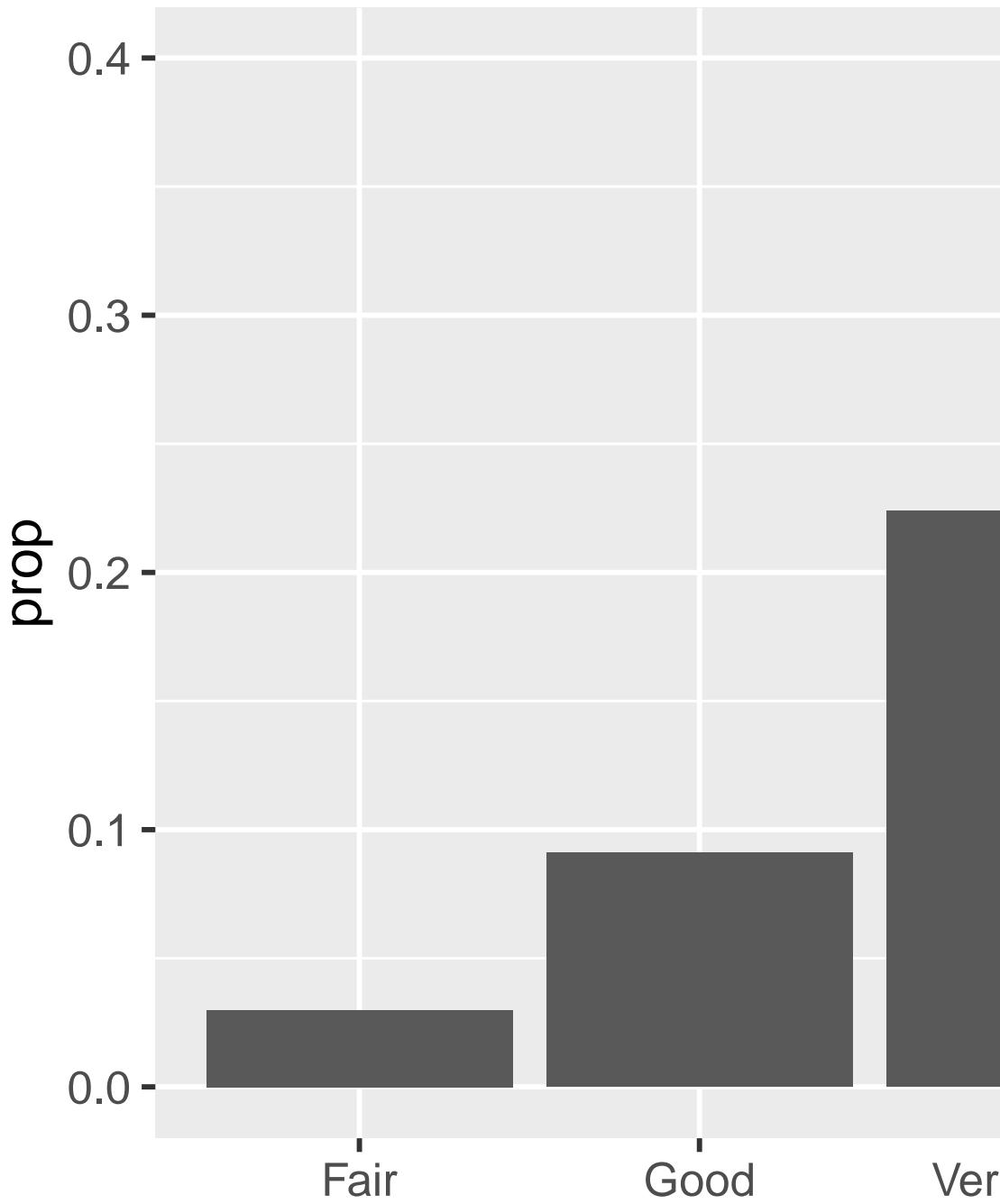
```
geom      geom      geoms
```

1. **geom_bar()** count identity y

```
diamonds |>
  count(cut) |>
  ggplot(aes(x = cut, y = n)) +
  geom_bar(stat = "identity")
```



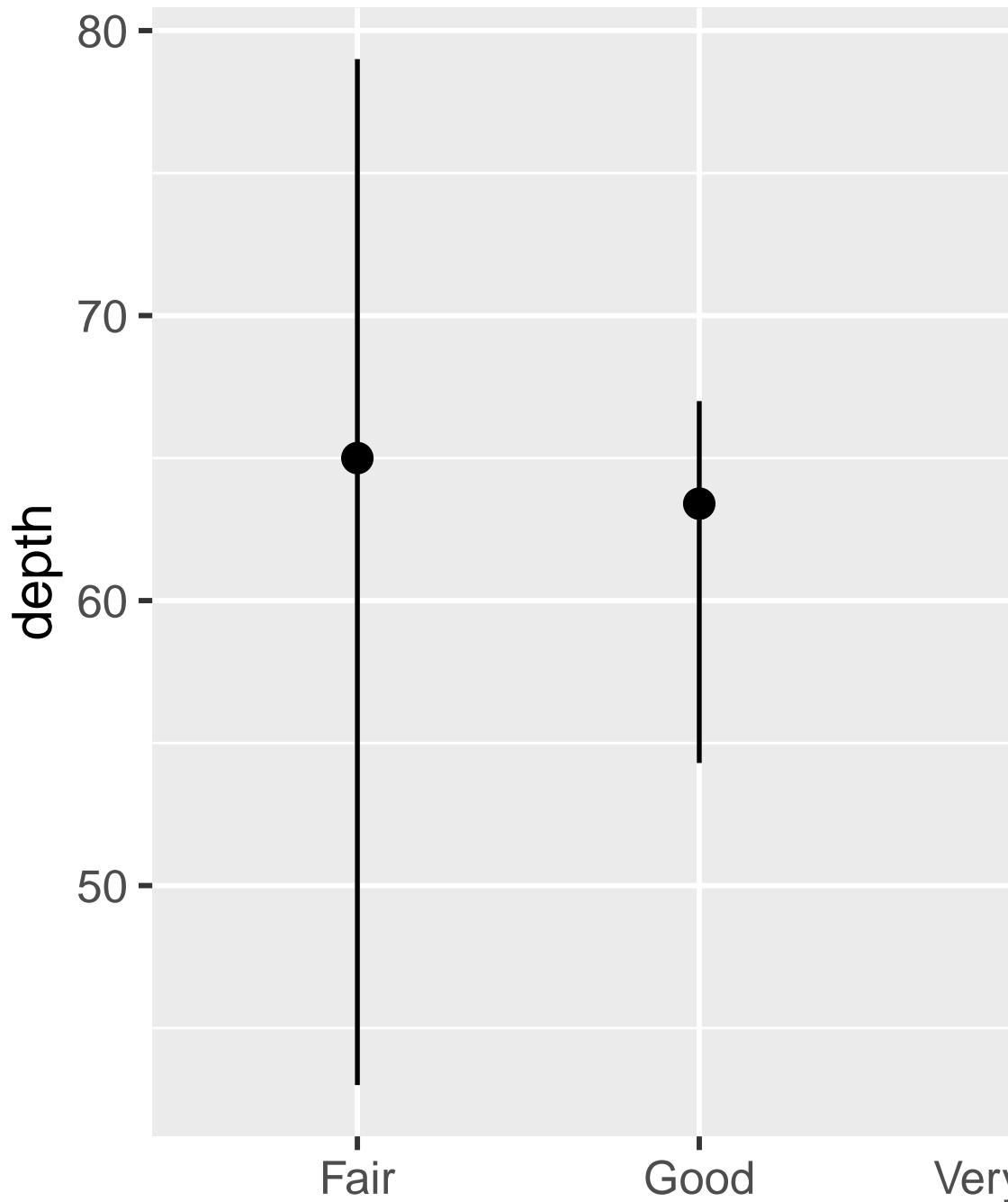
```
ggplot(diamonds, aes(x = cut, y = after_stat(prop), group = 1)) +  
  geom_bar()
```



geom_bar() “computed variables”

3. stat_summary() x y

```
ggplot(diamonds) +  
  stat_summary(  
    aes(x = cut, y = depth),  
    fun.min = min,  
    fun.max = max,  
    fun = median  
)
```



```
ggplot2      20          ?stat_bin
```

9.5.1

1. stat_summary() geom geom stat

2. geom_col() geom_bar()

3. geoms stats

4. stat_smooth()

5. group = 1

```
ggplot(diamonds, aes(x = cut, y = after_stat(prop))) +
  geom_bar()
ggplot(diamonds, aes(x = cut, fill = color, y = after_stat(prop))) +
  geom_bar()
```

9.6

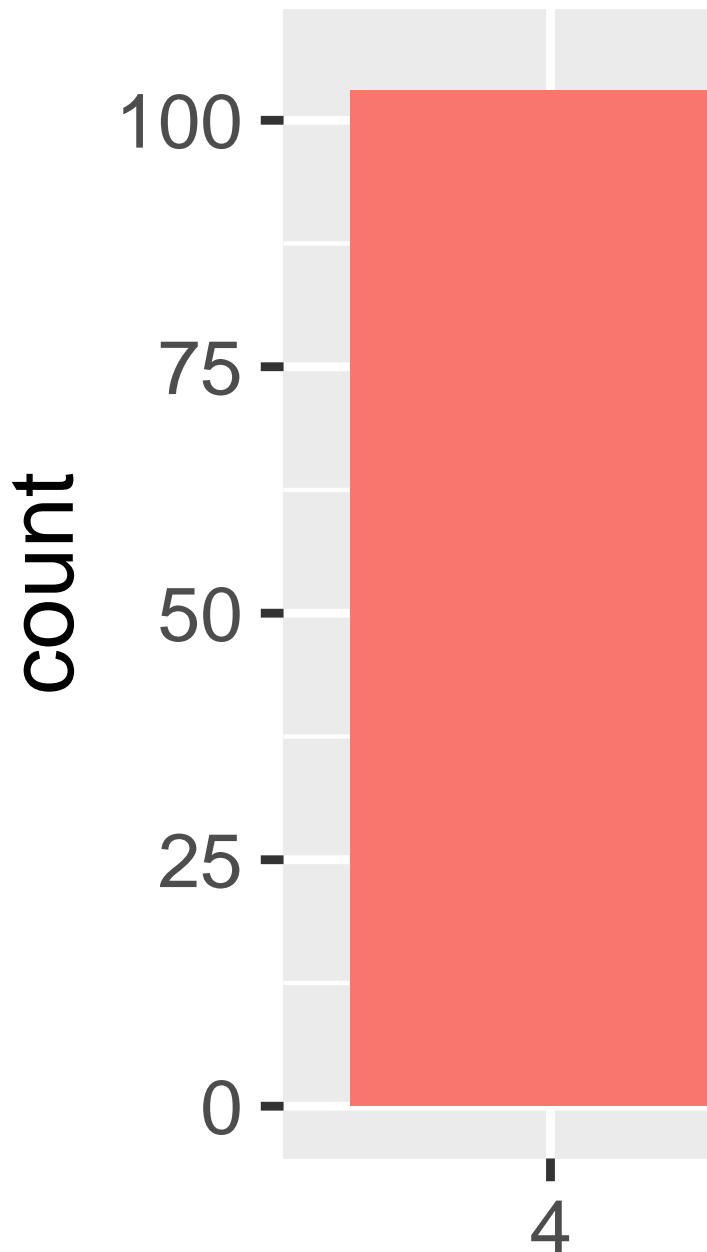
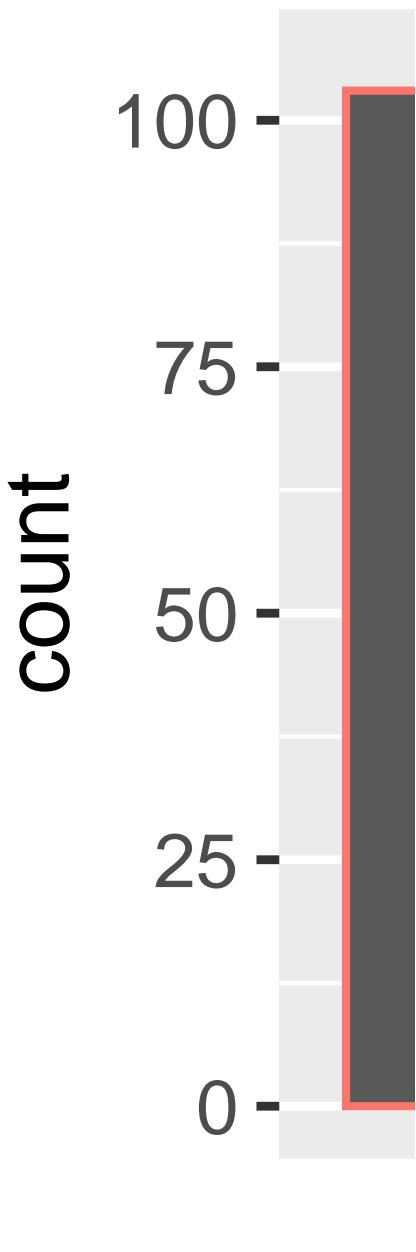
color aesthetic fill aesthetic

```
# Left
ggplot(mpg, aes(x = drv, color = drv)) +
  geom_bar()

# Right
ggplot(mpg, aes(x = drv, fill = drv)) +
  geom_bar()
```

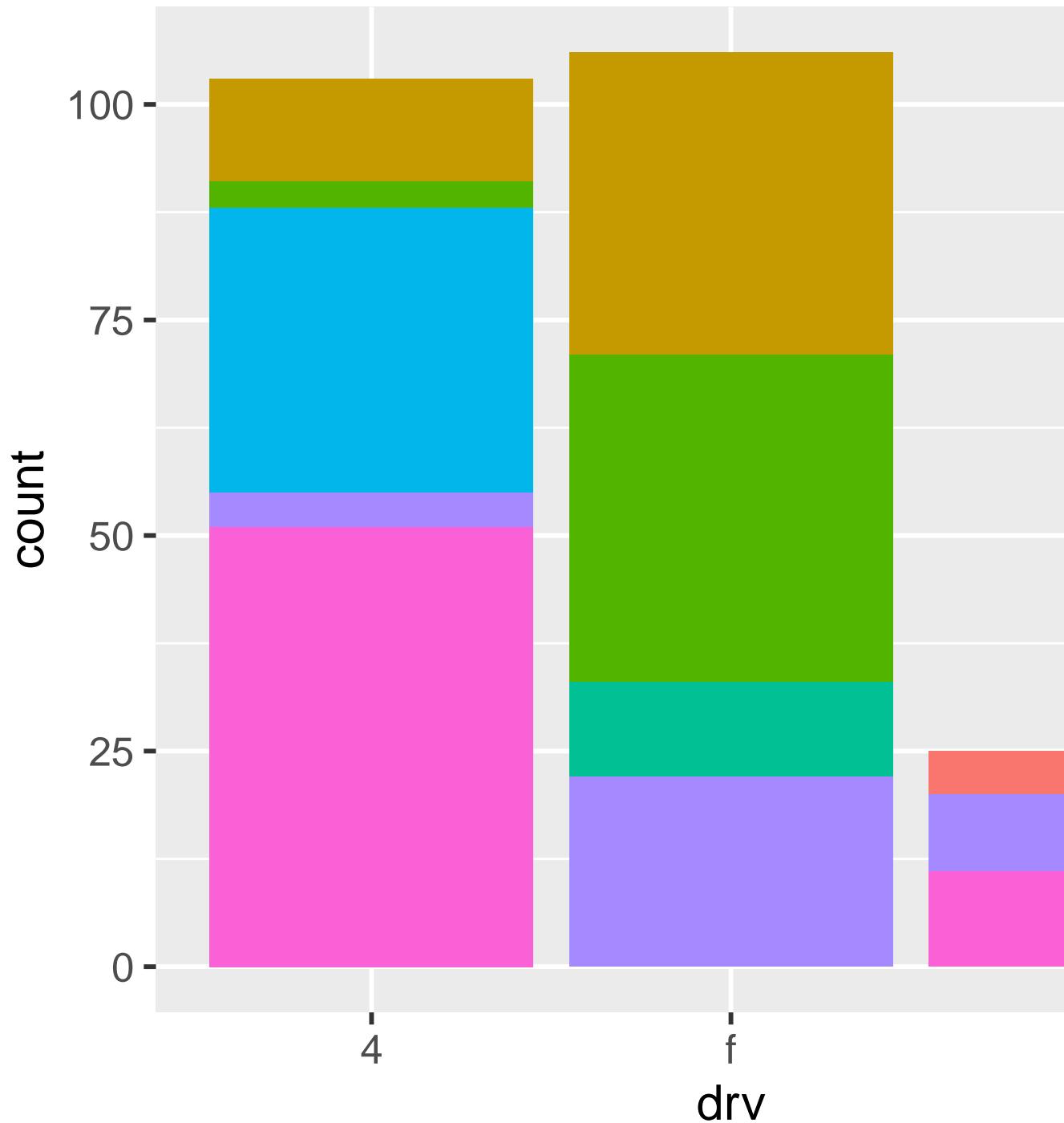
class drv class

```
ggplot(mpg, aes(x = drv, fill = class)) +
  geom_bar()
```



9.6.

193



```

      position      position adjustment      "identity" "dodge"
"fill"                                     alpha      fill

• position = "identity"                      alpha      fill
      = NA

```

```

# Left
ggplot(mpg, aes(x = drv, fill = class)) +
  geom_bar(alpha = 1/5, position = "identity")

# Right
ggplot(mpg, aes(x = drv, color = class)) +
  geom_bar(fill = NA, position = "identity")

```

"identity"

- position = "fill"

- position = "dodge"

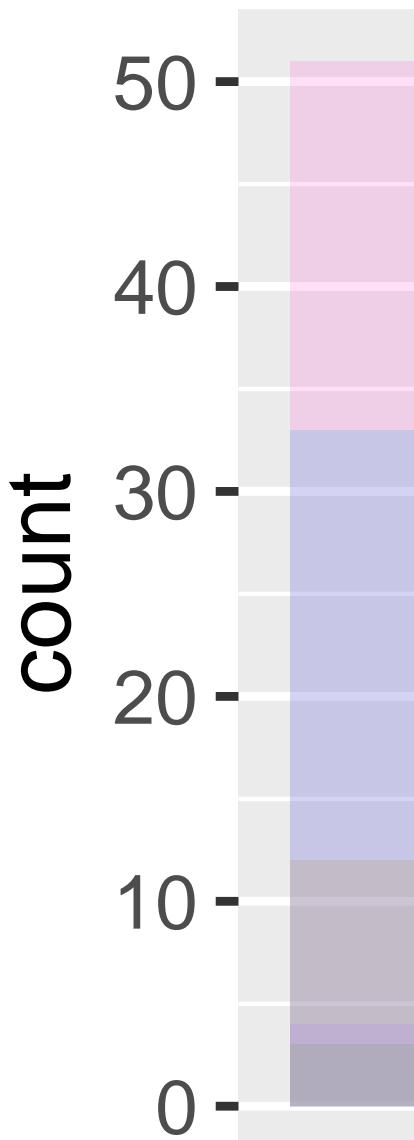
```

# Left
ggplot(mpg, aes(x = drv, fill = class)) +
  geom_bar(position = "fill")

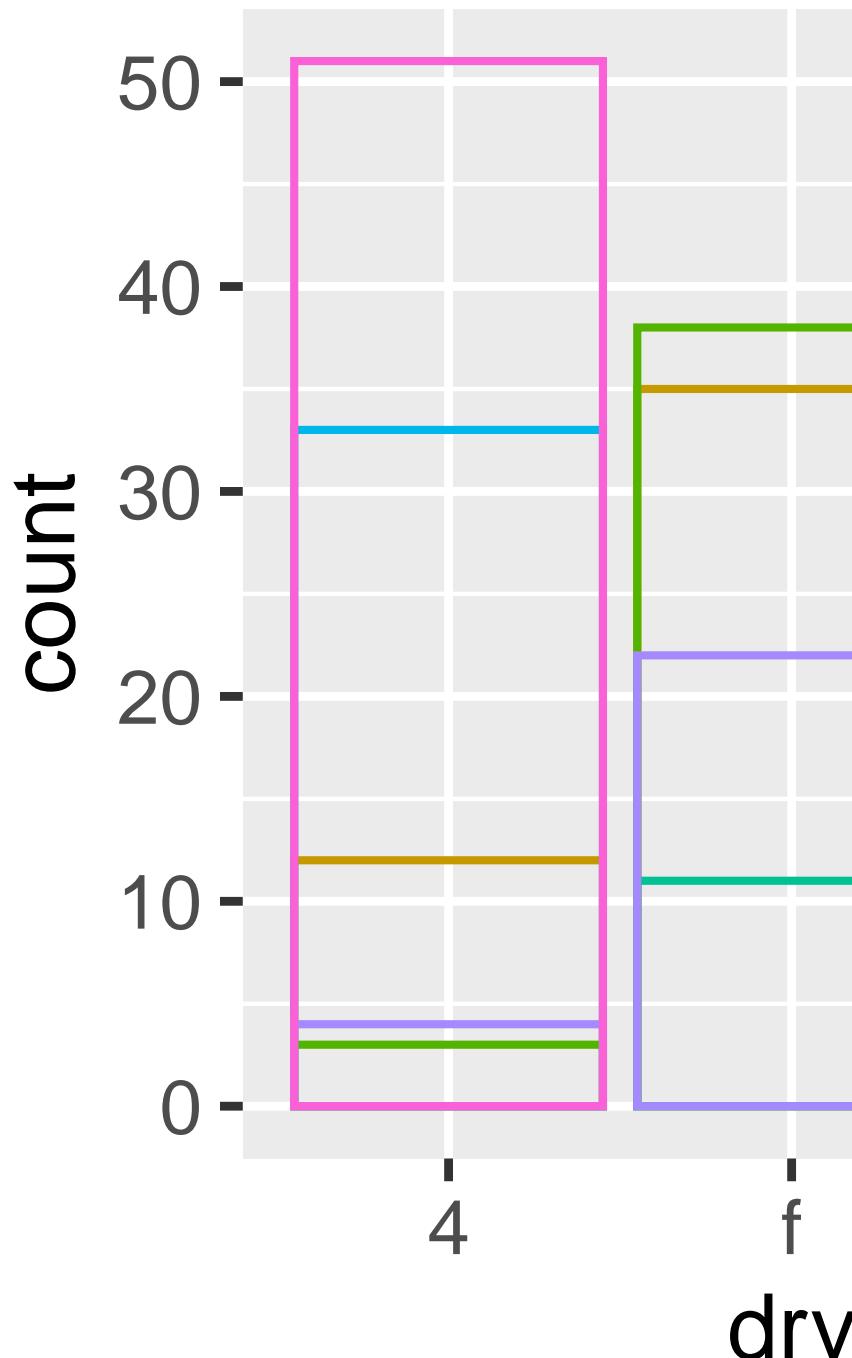
# Right
ggplot(mpg, aes(x = drv, fill = class)) +
  geom_bar(position = "dodge")

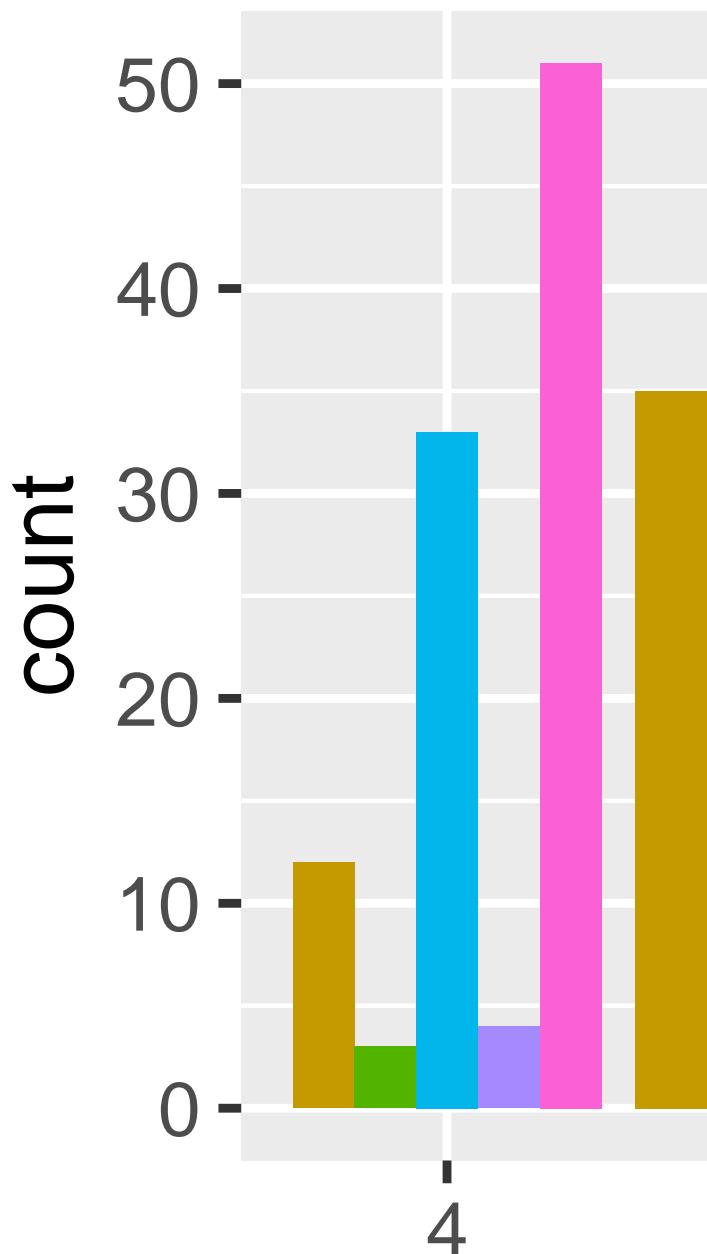
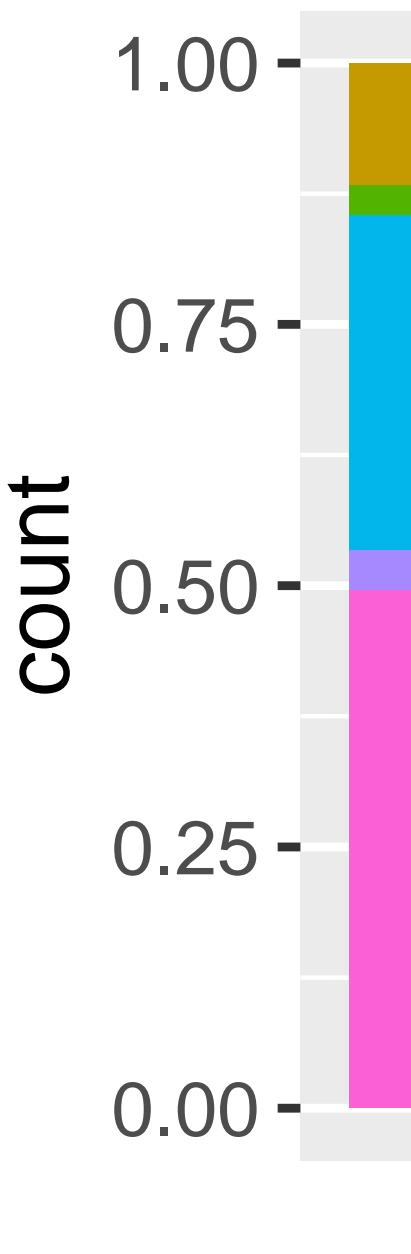
```

9.6.



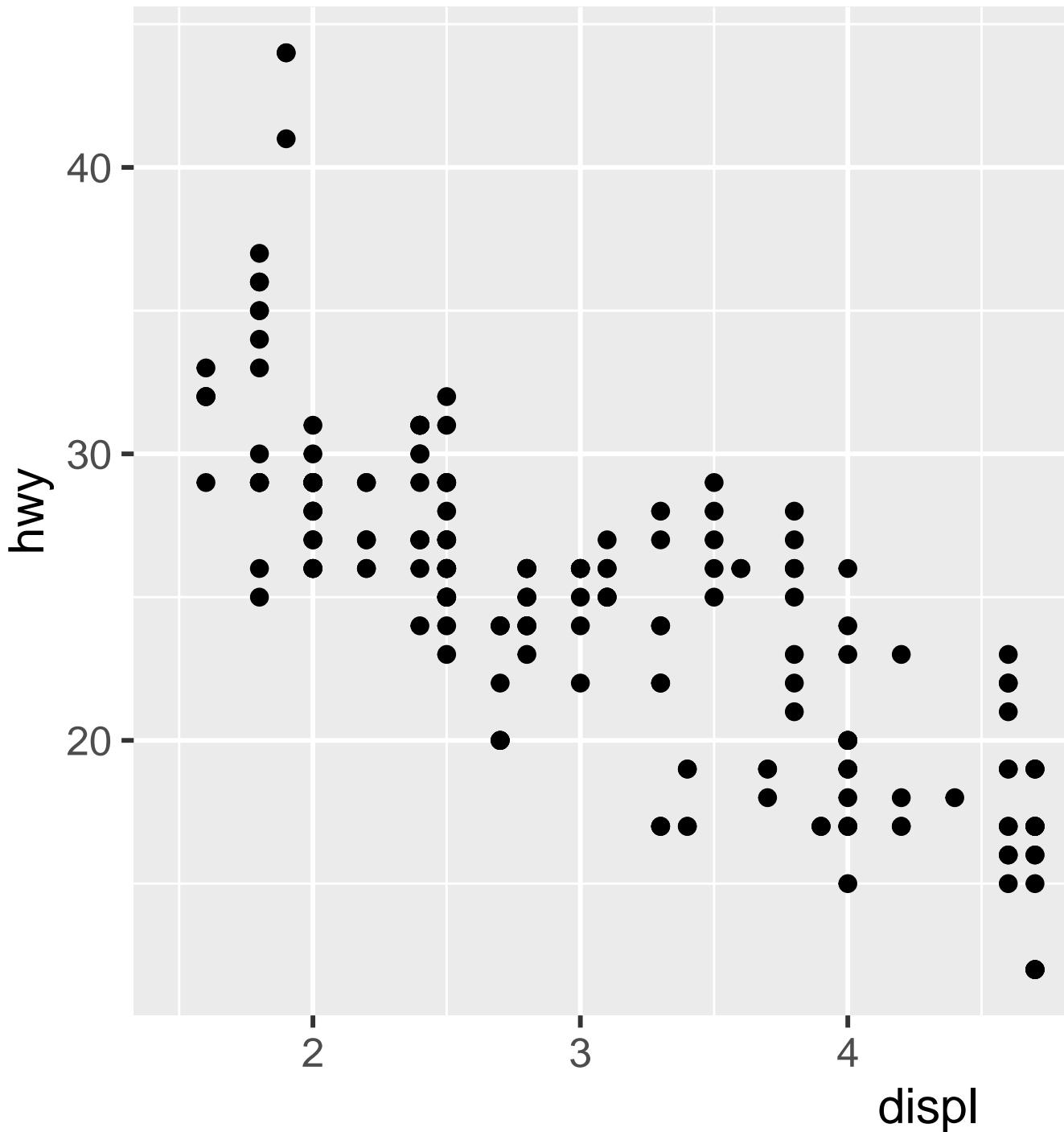
195





9.6.

197



hwy displ

overplotting

hwy displ

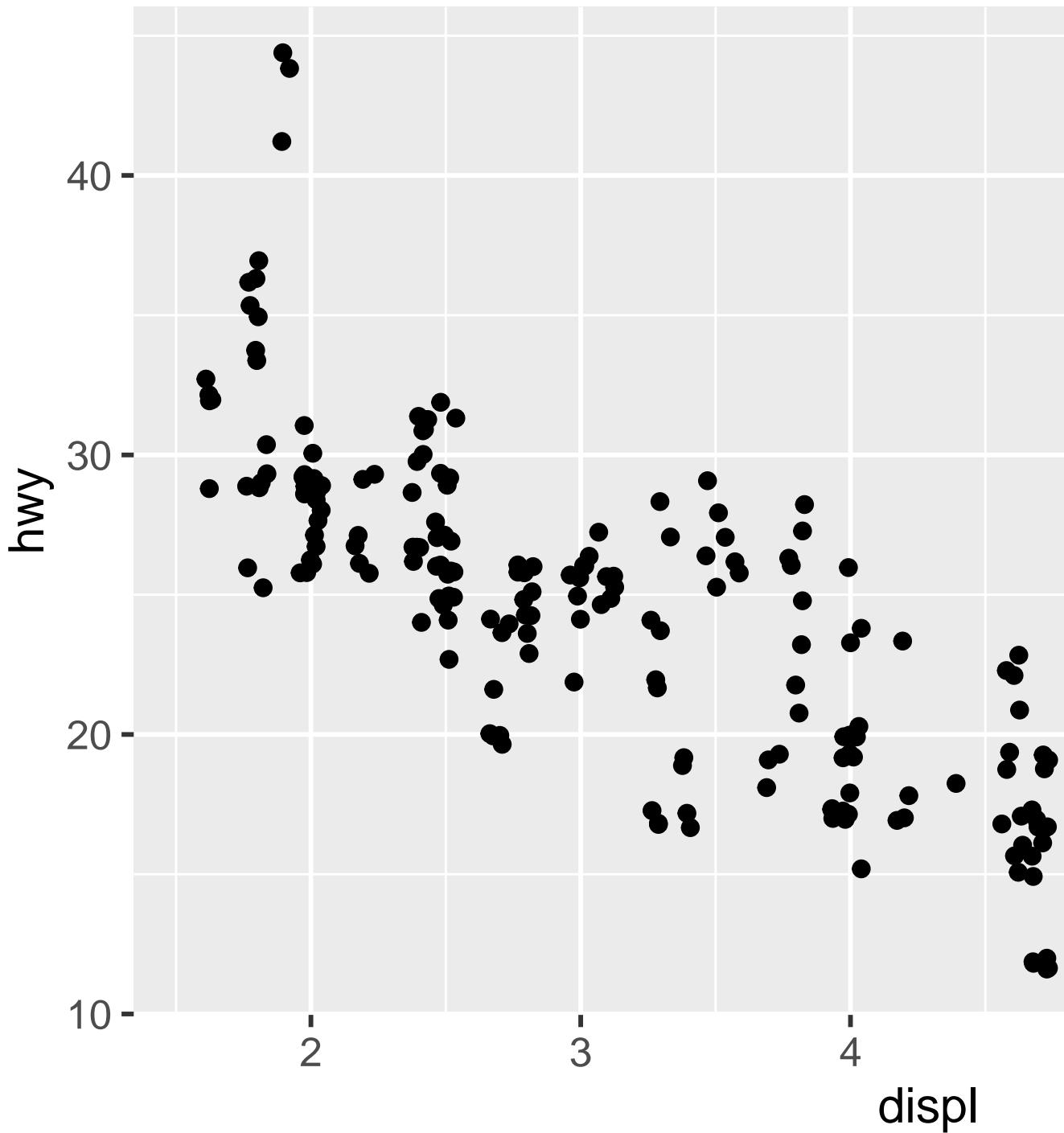
109

“jitter” position = “jitter”

```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point(position = "jitter")
```

9.6.

199



```
ggplot2 geom_point(position =
"jitter")      geom_jitter()
?position_dodge ?position_fill ?position_identity ?position_jitter ?pos
```

9.6.1

1.

```
ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point()
```

2.

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point()
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(position = "identity")
```

3. `geom_jitter()` amount of jittering

4. `geom_jitter()` `geom_count()`

5. `geom_boxplot()` mpg

9.7

`ggplot2` x y

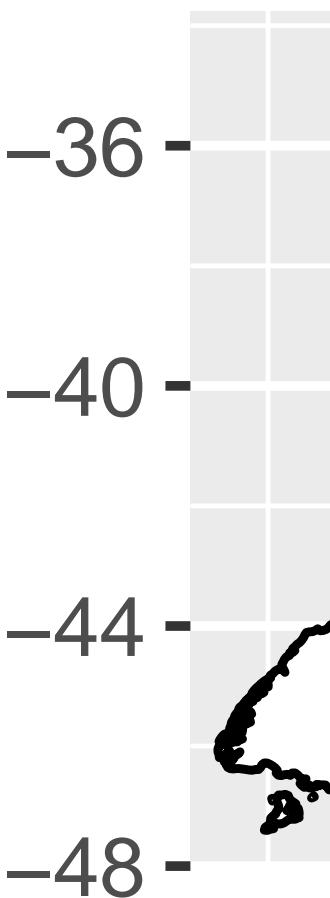
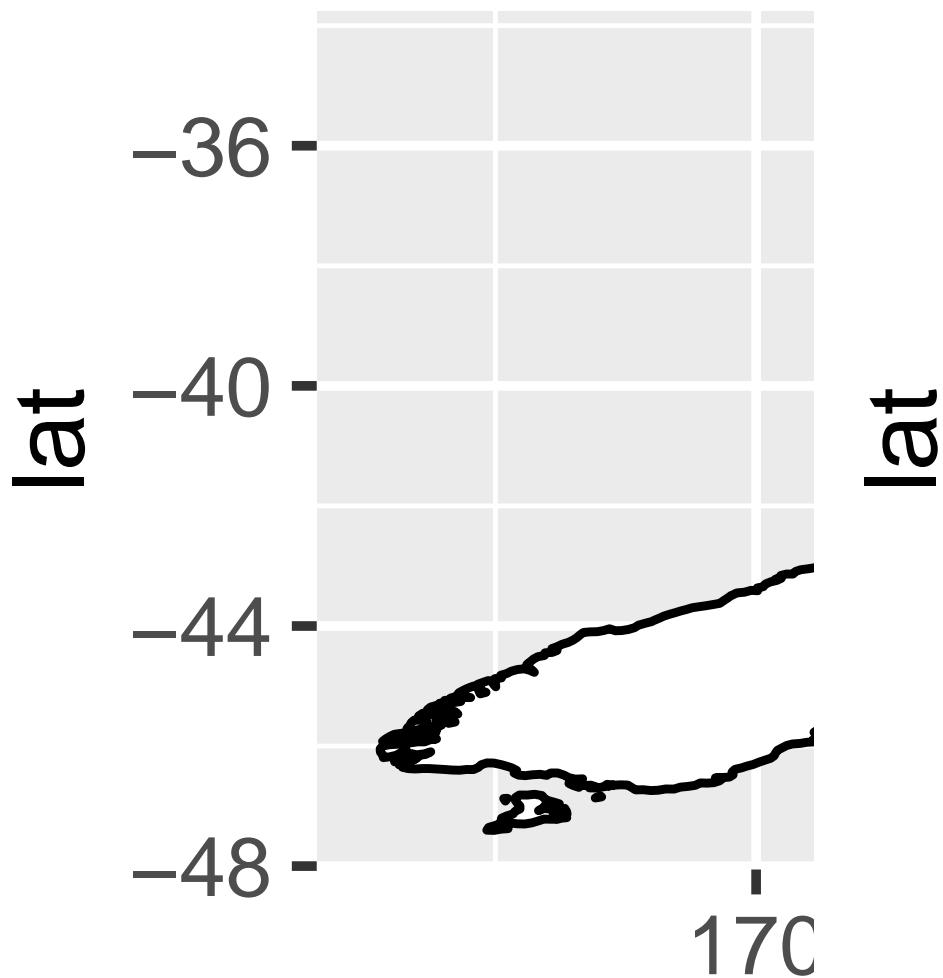
- `coord_quickmap()` ggplot2
Elegant graphics for data analysis Maps chapter

```
nz <- map_data("nz")
ggplot(nz, aes(x = long, y = lat, group = group)) +
  geom_polygon(fill = "white", color = "black")
ggplot(nz, aes(x = long, y = lat, group = group)) +
  geom_polygon(fill = "white", color = "black") +
  coord_quickmap()
```

- `coord_polar()` Coxcomb chart

9.7.

201



```
bar <- ggplot(data = diamonds) +  
  geom_bar(  
    mapping = aes(x = clarity, fill = clarity),  
    show.legend = FALSE,  
    width = 1  
) +  
  theme(aspect.ratio = 1)  
  
bar + coord_flip()  
bar + coord_polar()
```

9.7.1

1. coord_polar()
 2. coord_quickmap() coord_map()
 3. city mpg coord_fixed() geom_abline()

```
ggplot(data = mpg, mapping = aes(x = cty, y = hwy)) +  
  geom_point() +  
  geom_abline() +  
  coord_fixed()
```

9.8

@sec-ggplot2-calls

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(  
    mapping = aes(<MAPPINGS>),  
    stat = <STAT>,  
    position = <POSITION>  
) +  
  <COORDINATE_FUNCTION> +  
  <FACET FUNCTION>
```

ggplot2 geom

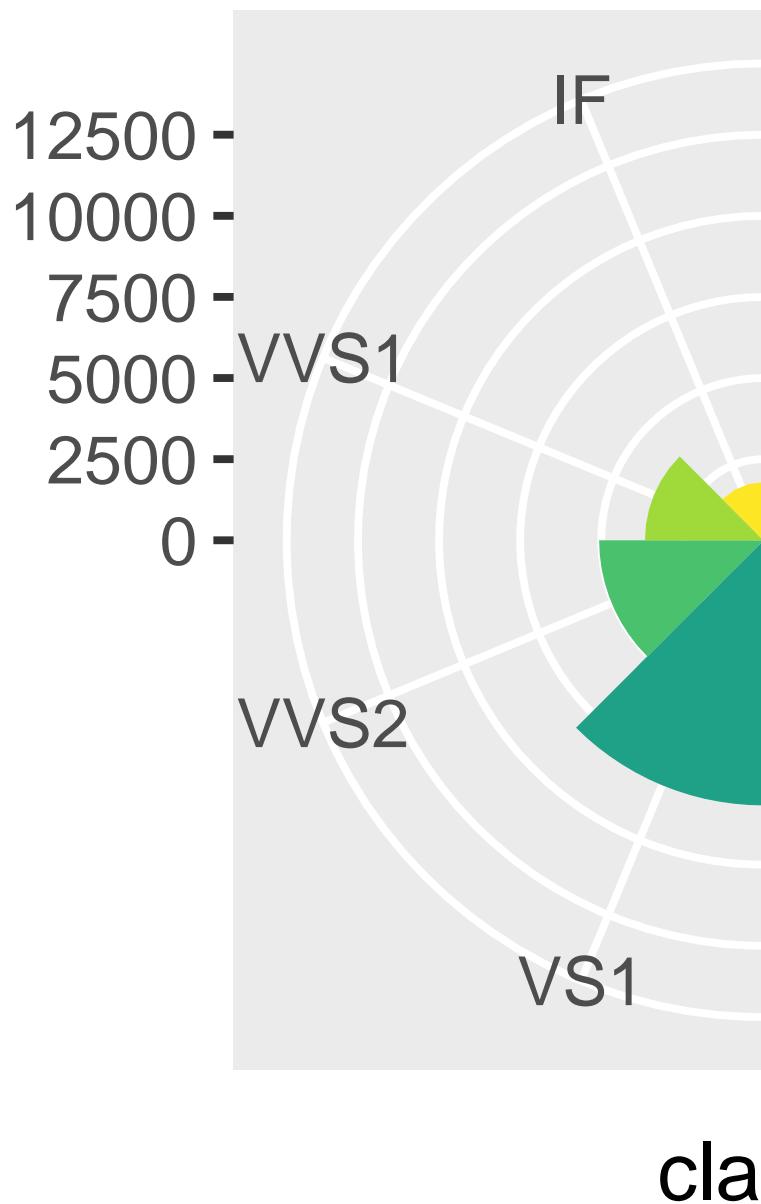
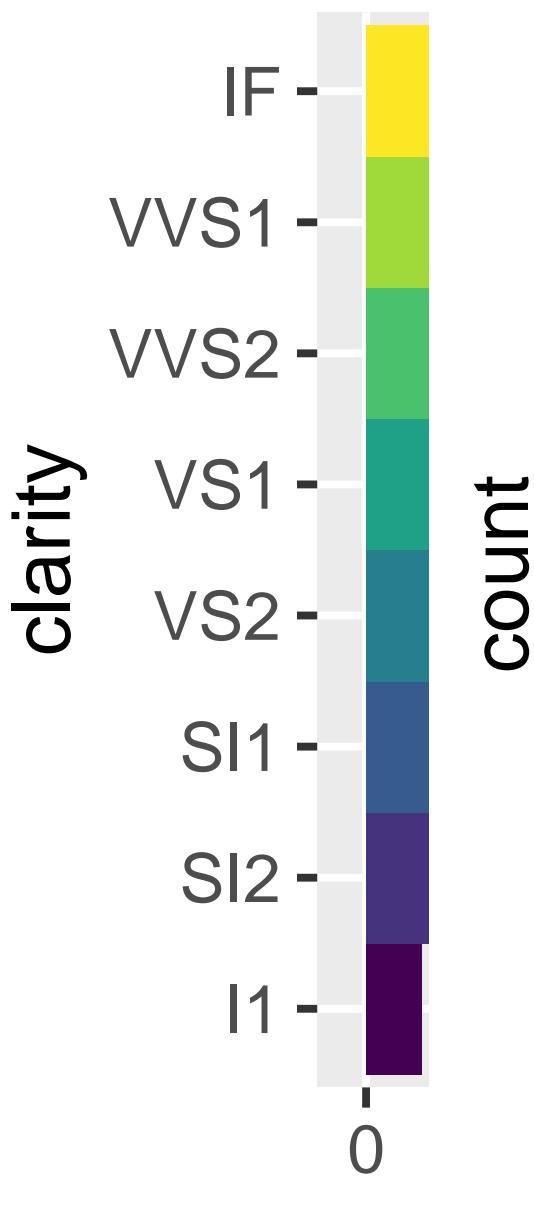
grammar of graphics

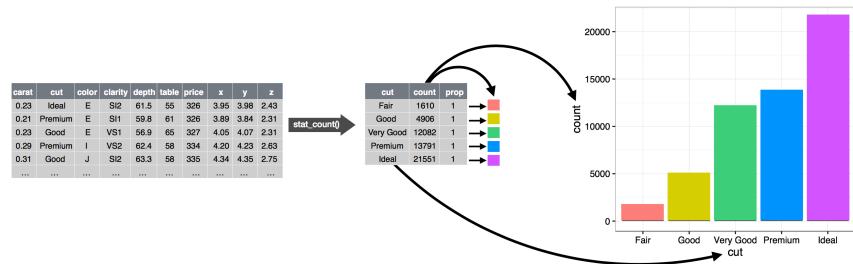
geom

geom
?? geom x y

9.8.

203





9.3: Steps for going from raw data to a table of frequencies to a bar plot where the heights of the bar represent the frequencies.

9.9

geom
geom
geom
x
y
theme
??
ggplot2
ggplot2
https://posit.co/resources/cheatsheets
ggplot2
https://ggplot2.tidyverse.org/)
ggplot2
geom
geom
geom
ggplot2

Chapter 10

10.1

Exploratory Data Analysis EDA E DA

- 1.
- 2.
- 3.

EDA EDA
EDA EDA EDA

10.1.1

dplyr ggplot2

```
library(tidyverse)
```

10.2

“ ” — Sir David Cox

“ ” — John Tukey

EDA

EDA

1.

2.

10.3

variation

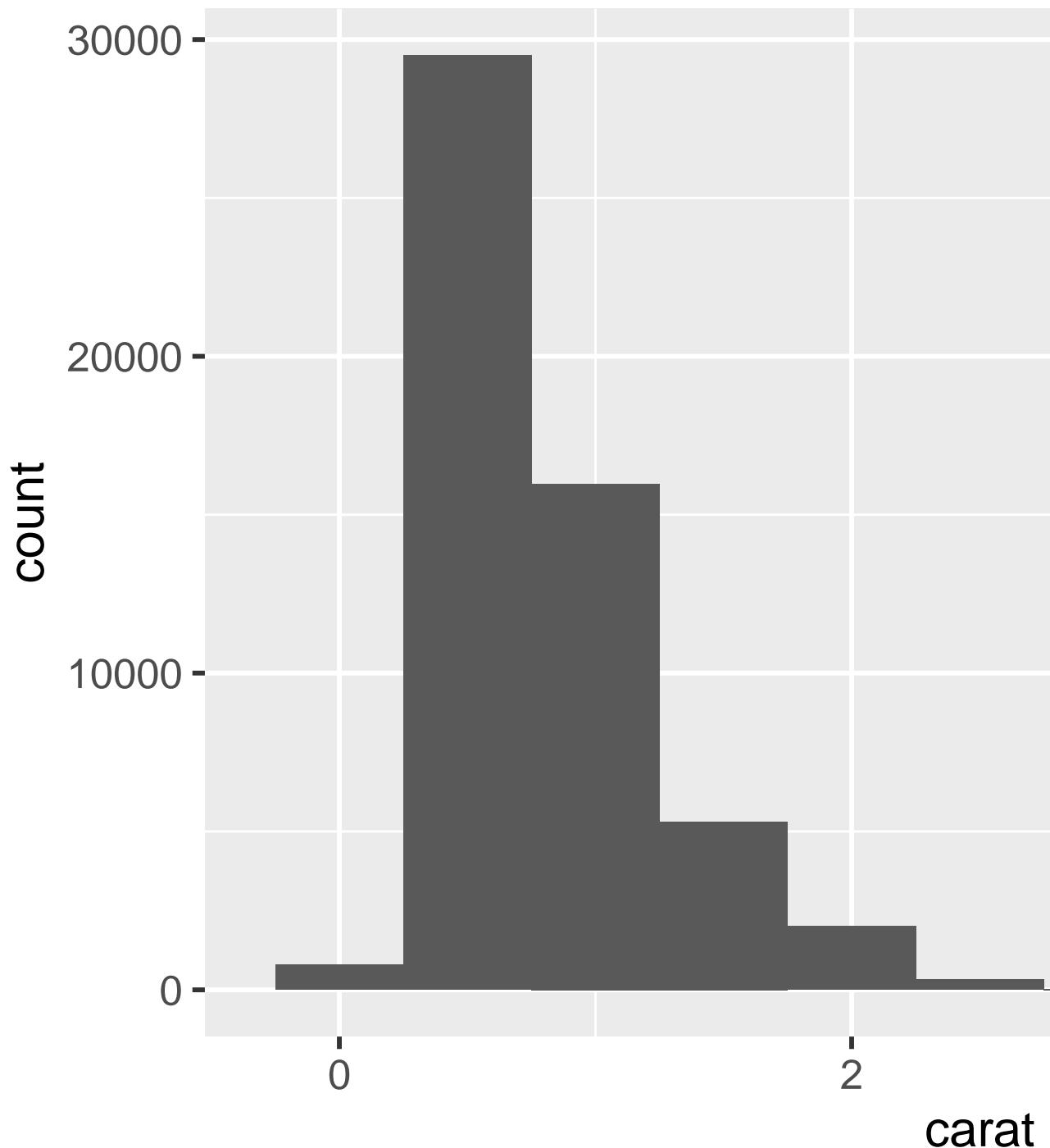
??

diamonds 54,000 carat carat

```
ggplot(diamonds, aes(x = carat)) +  
  geom_histogram(binwidth = 0.5)
```

10.3.

207



10.3.1

•
•
•

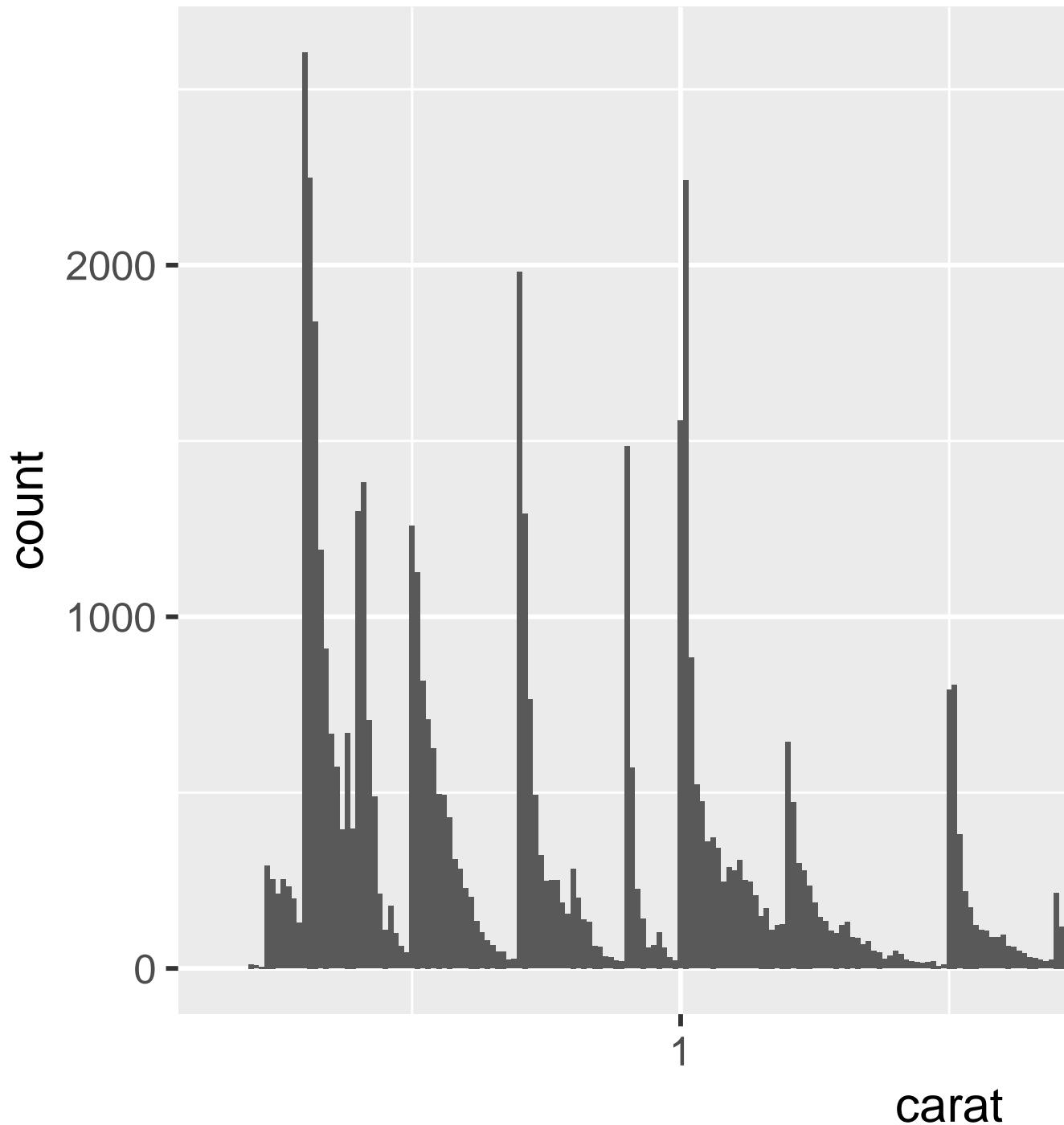
carat

```
smaller <- diamonds |>
  filter(carat < 3)

ggplot(smaller, aes(x = carat)) +
  geom_histogram(binwidth = 0.01)
```

10.3.

209



•
•
•
•
•
•
•
•

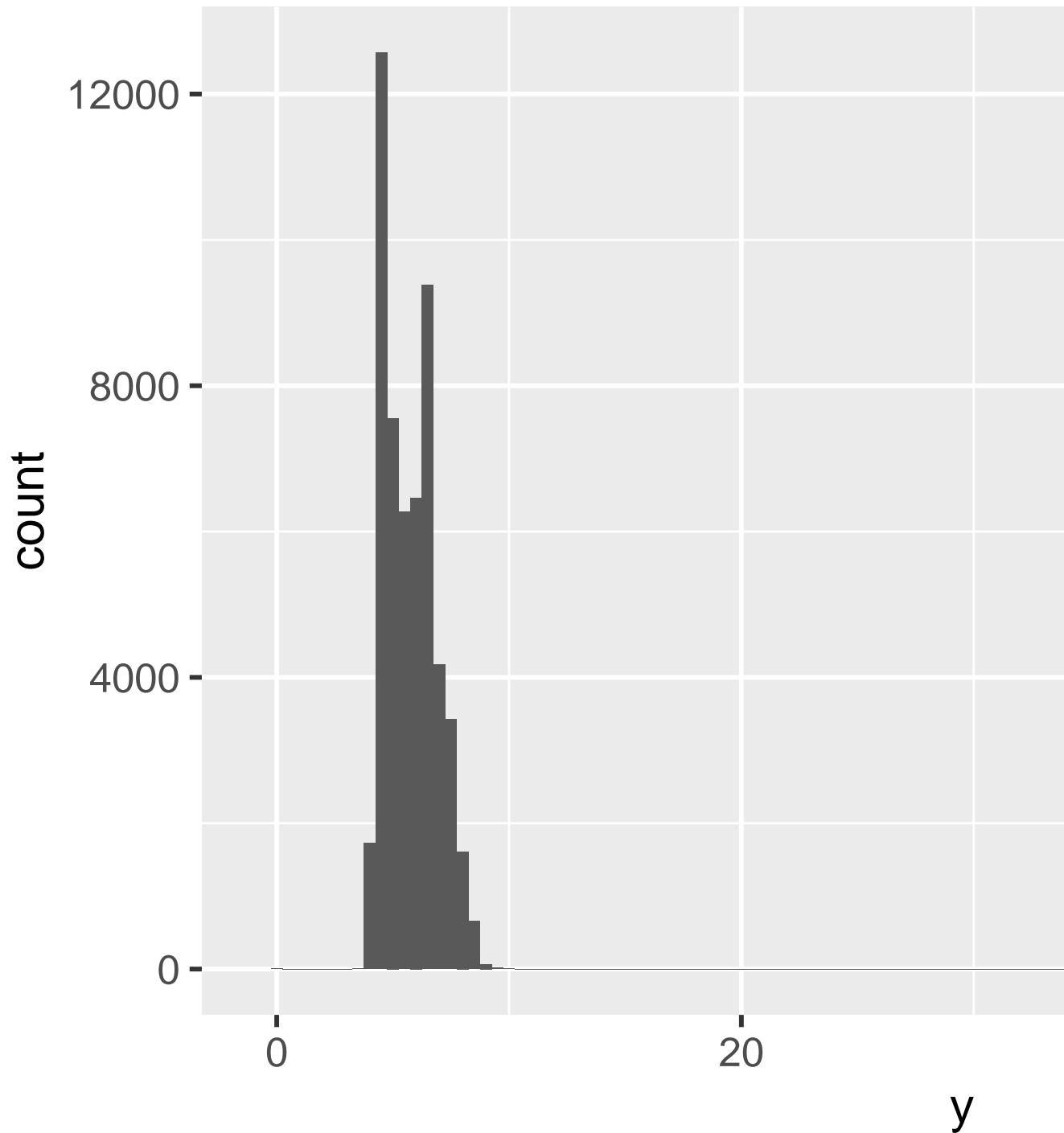
10.3.2

y x

```
ggplot(diamonds, aes(x = y)) +  
  geom_histogram(binwidth = 0.5)
```

10.3.

211

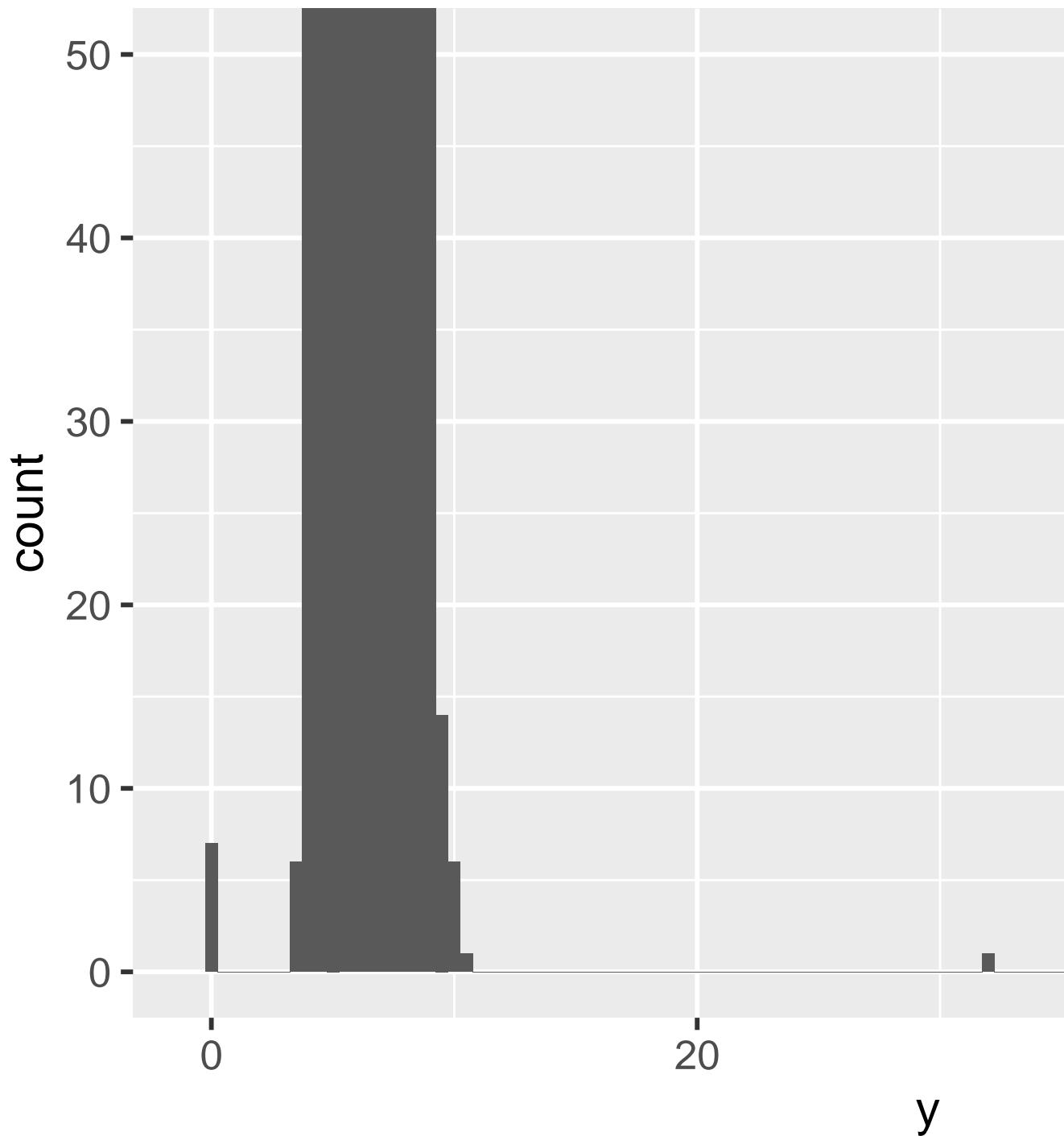


```
y    0          coord_cartesian()  
y
```

```
ggplot(diamonds, aes(x = y)) +  
  geom_histogram(binwidth = 0.5) +  
  coord_cartesian(ylim = c(0, 50))
```

10.3.

213



```

coord_cartesian() xlim()      x      ggplot2 xlim() ylim()
0 ~30 ~60    dplyr

unusual <- diamonds |>
  filter(y < 3 | y > 20) |>
  select(price, x, y, z) |>
  arrange(y)
unusual
#> # A tibble: 9 x 4
#>   price     x     y     z
#>   <int> <dbl> <dbl> <dbl>
#> 1 5139     0     0     0
#> 2 6381     0     0     0
#> 3 12800    0     0     0
#> 4 15686    0     0     0
#> 5 18034    0     0     0
#> 6 2130     0     0     0
#> 7 2130     0     0     0
#> 8 2075     5.15  31.8  5.12
#> 9 12210    8.09  58.9  8.06

```

| | | | |
|---|----------|---------------|---------|
| y | 0
NAs | ED
A
32 | 0
59 |
|---|----------|---------------|---------|

10.3.3

1. diamonds x y z
2. price binwidth
3. 0.99 1
4. coord_cartesian() xlim() ylim() binwidth

10.4

- 1.

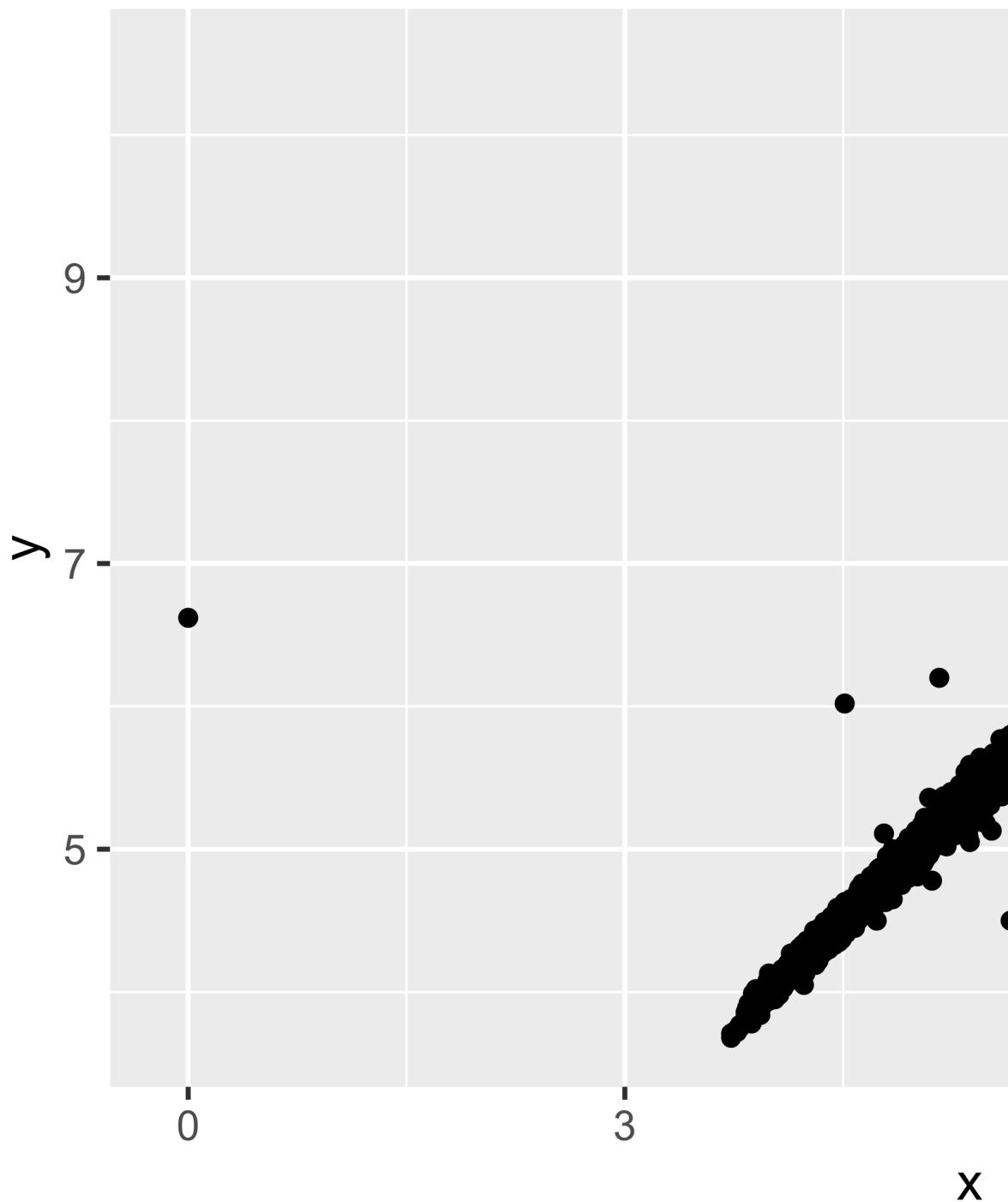
```
diamonds2 <- diamonds |>  
  filter(between(y, 3, 20))
```

2. `mutate()` `if_else()` `NA`

```
diamonds2 <- diamonds |>  
  mutate(y = if_else(y < 3 | y > 20, NA, y))
```

ggplot2

```
ggplot(diamonds2, aes(x = x, y = y)) +  
  geom_point()  
#> Warning: Removed 9 rows containing missing values or values outside the scale range  
#> (`geom_point()`).
```



```
na.rm = TRUE:
```

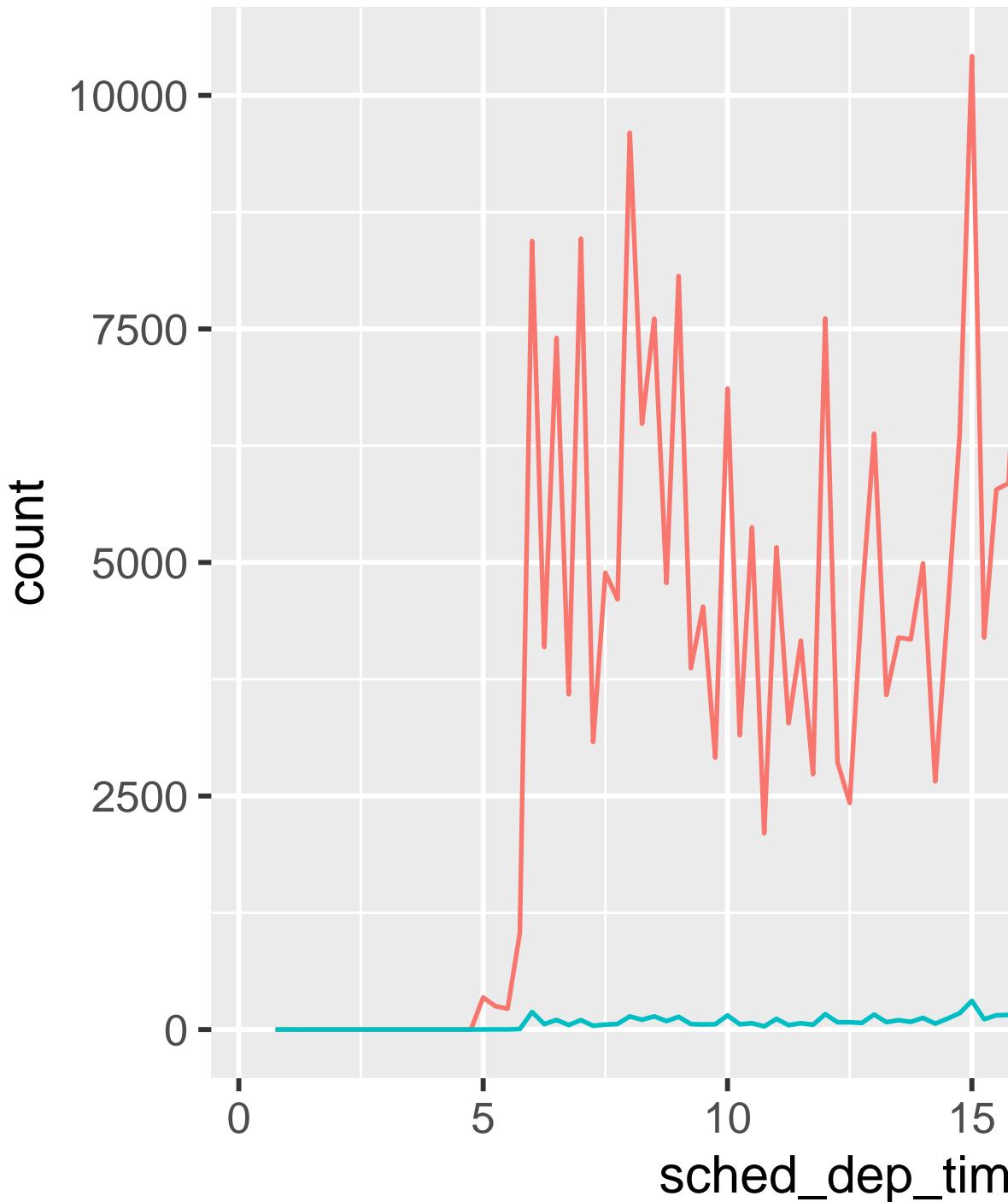
```
ggplot(diamonds2, aes(x = x, y = y)) +
  geom_point(na.rm = TRUE)
```

```
nycflights13::flights1  dep_time
  is.na()  dep_time
```

```
nycflights13::flights |>
  mutate(
    cancelled = is.na(dep_time),
    sched_hour = sched_dep_time %% 100,
    sched_min = sched_dep_time %% 100,
    sched_dep_time = sched_hour + (sched_min / 60)
  ) |>
  ggplot(aes(x = sched_dep_time)) +
  geom_freqpoly(aes(color = cancelled), binwidth = 1/4)
```

¹

package::function() package::dataset



10.4.1

1.

2. `mean()` `sum()` `na.rm = TRUE`

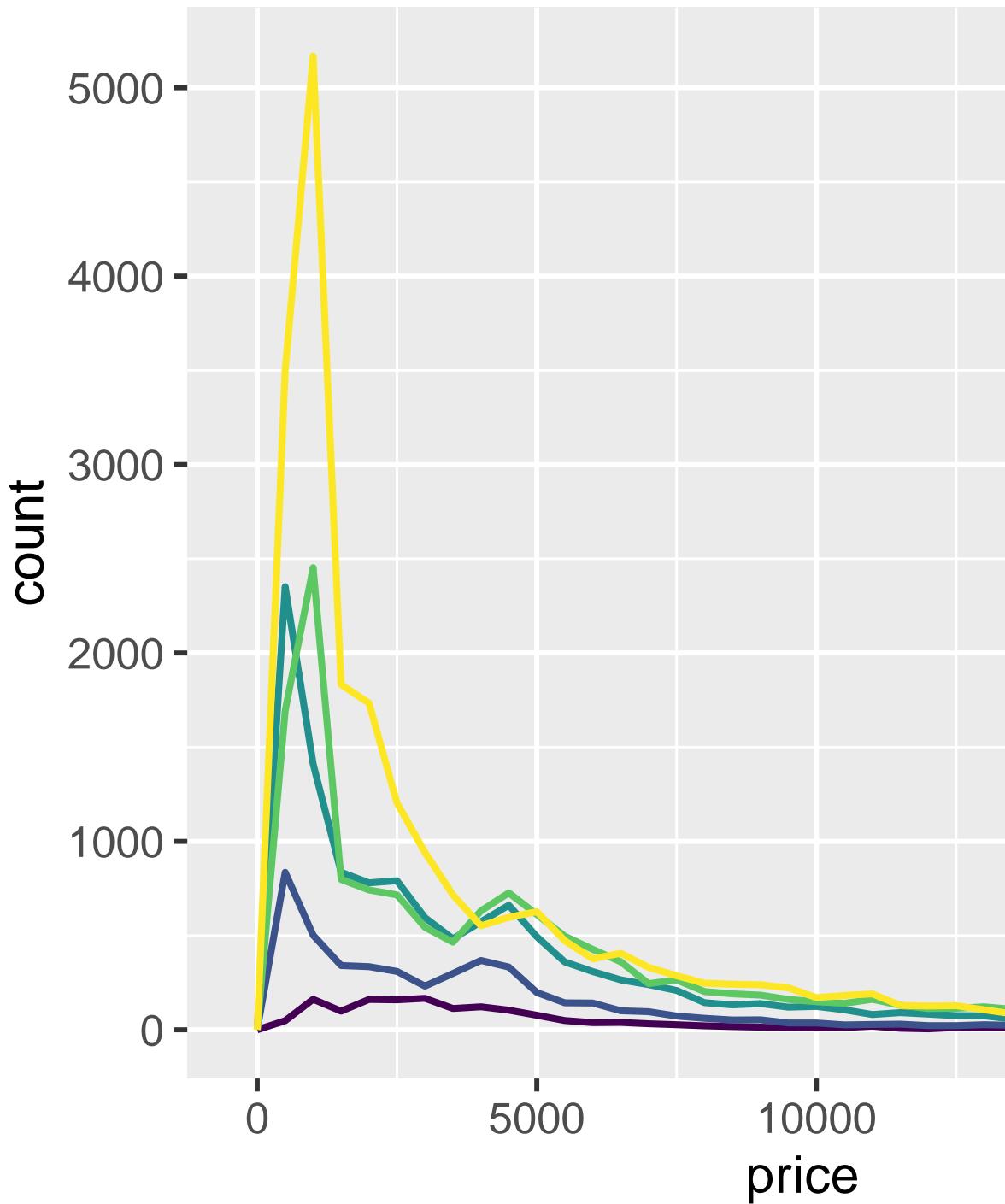
3. `scheduled_dep_time` `cancelled` `scales`

10.5 covariation

10.5.1

`geom_freqpoly()` `(cut)` :

```
ggplot(diamonds, aes(x = price)) +
  geom_freqpoly(aes(color = cut), binwidth = 500, linewidth = 0.75)
```

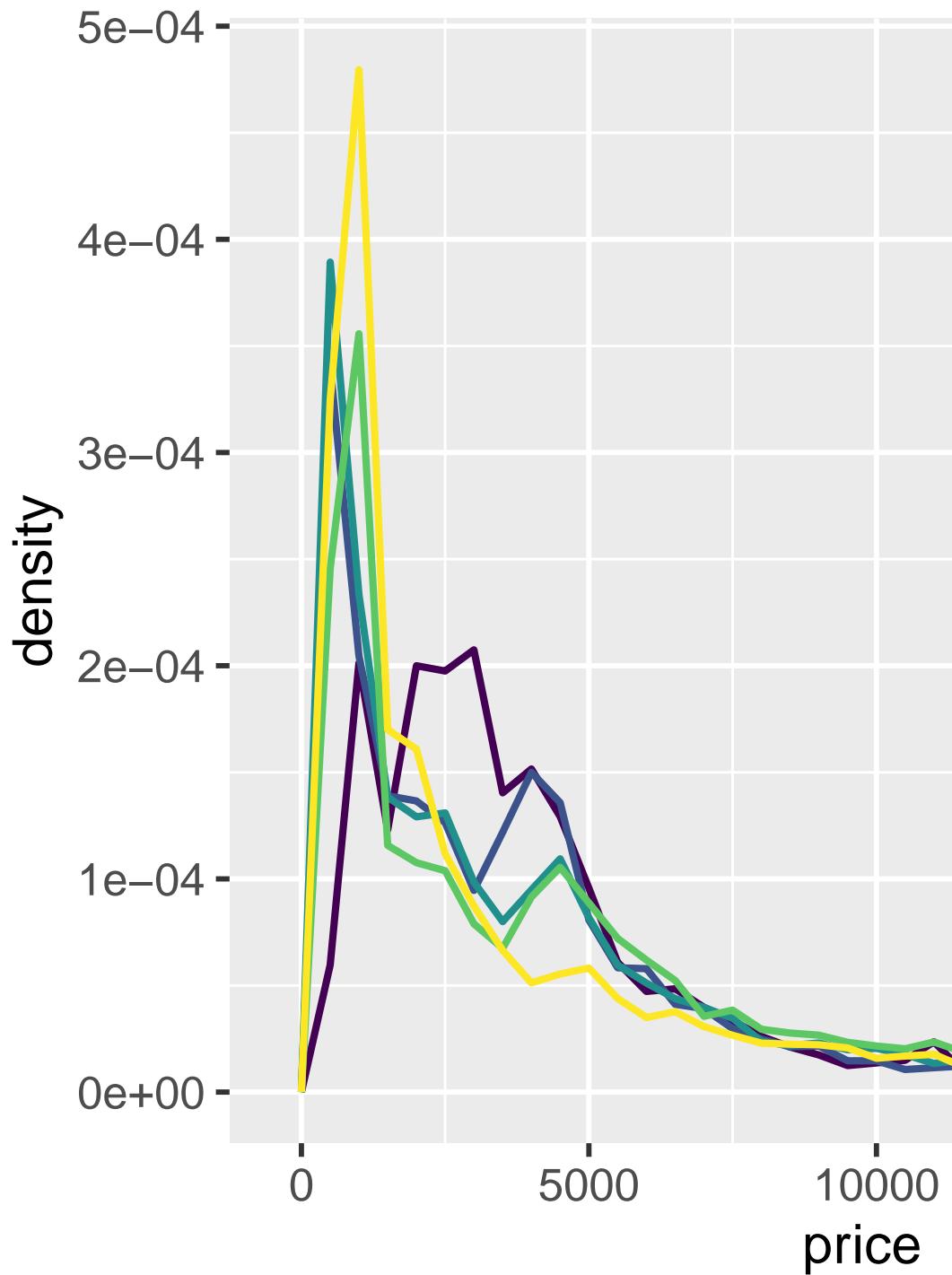


```
ggplot2 cut          ??
```

```
geom_freqpoly()      cut
```

```
y           density      1
```

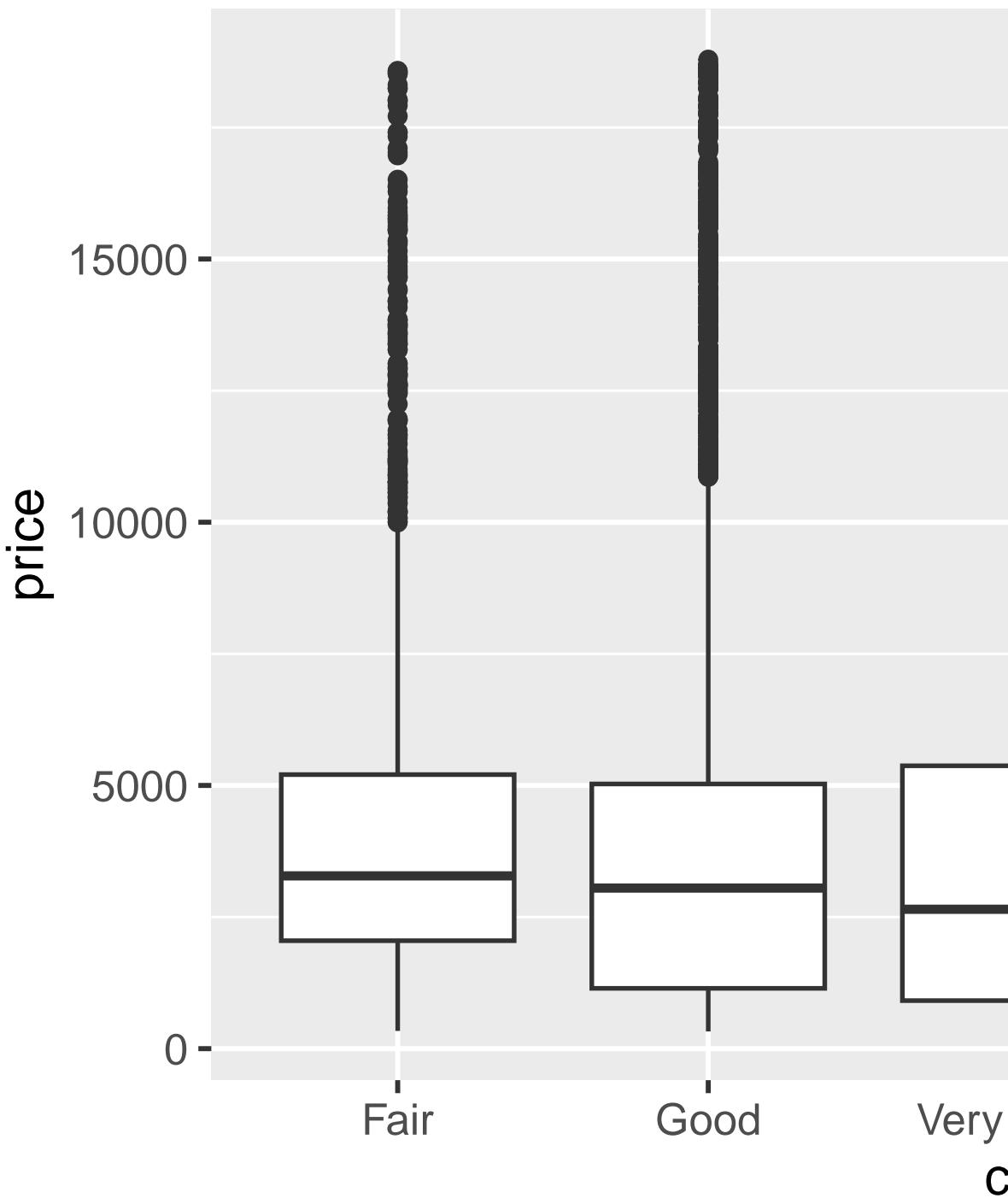
```
ggplot(diamonds, aes(x = price, y = after_stat(density))) +  
  geom_freqpoly(aes(color = cut), binwidth = 500, linewidth = 0.75)
```



```
density    y    density  diamonds      after_stat()
```

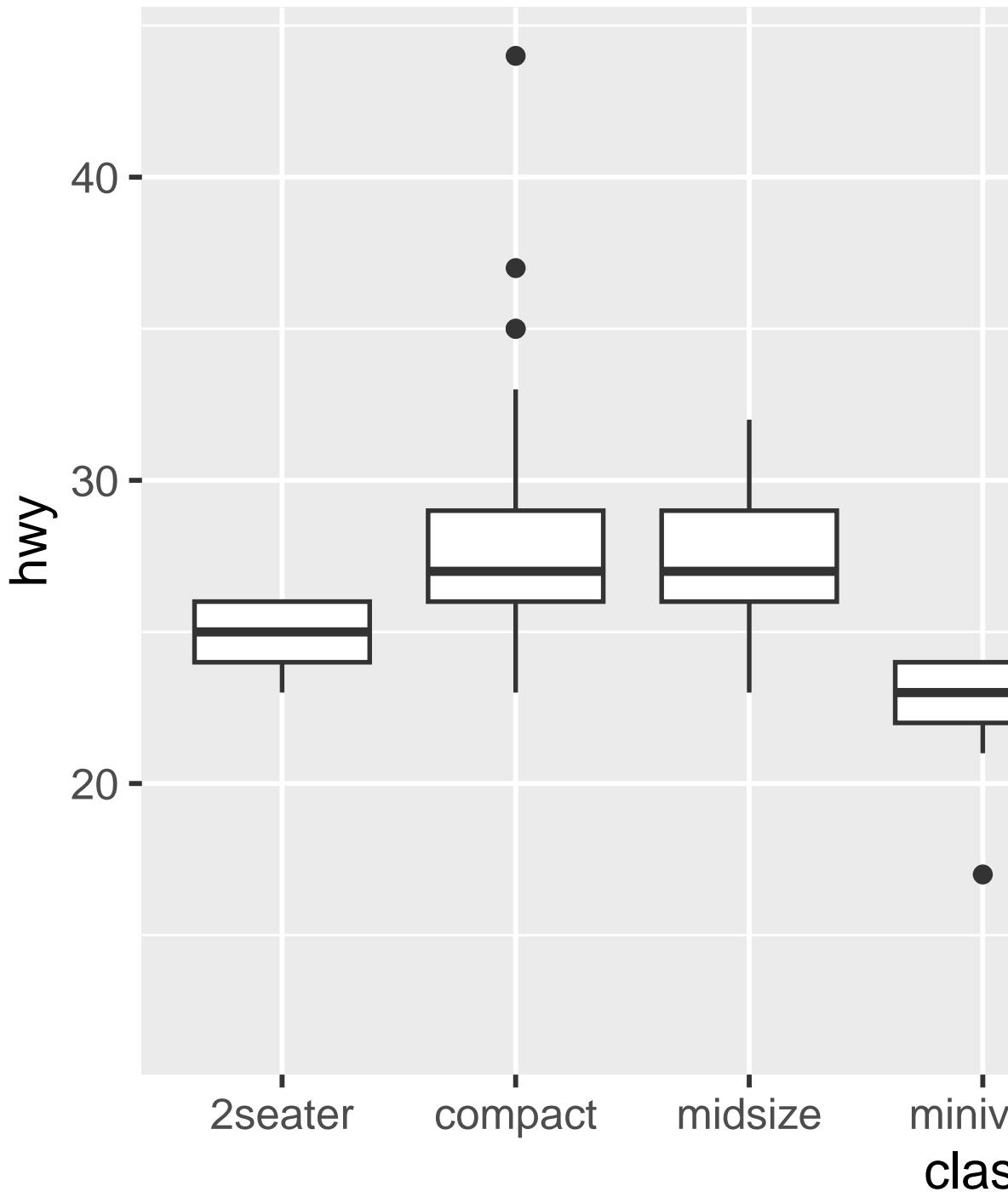
—

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



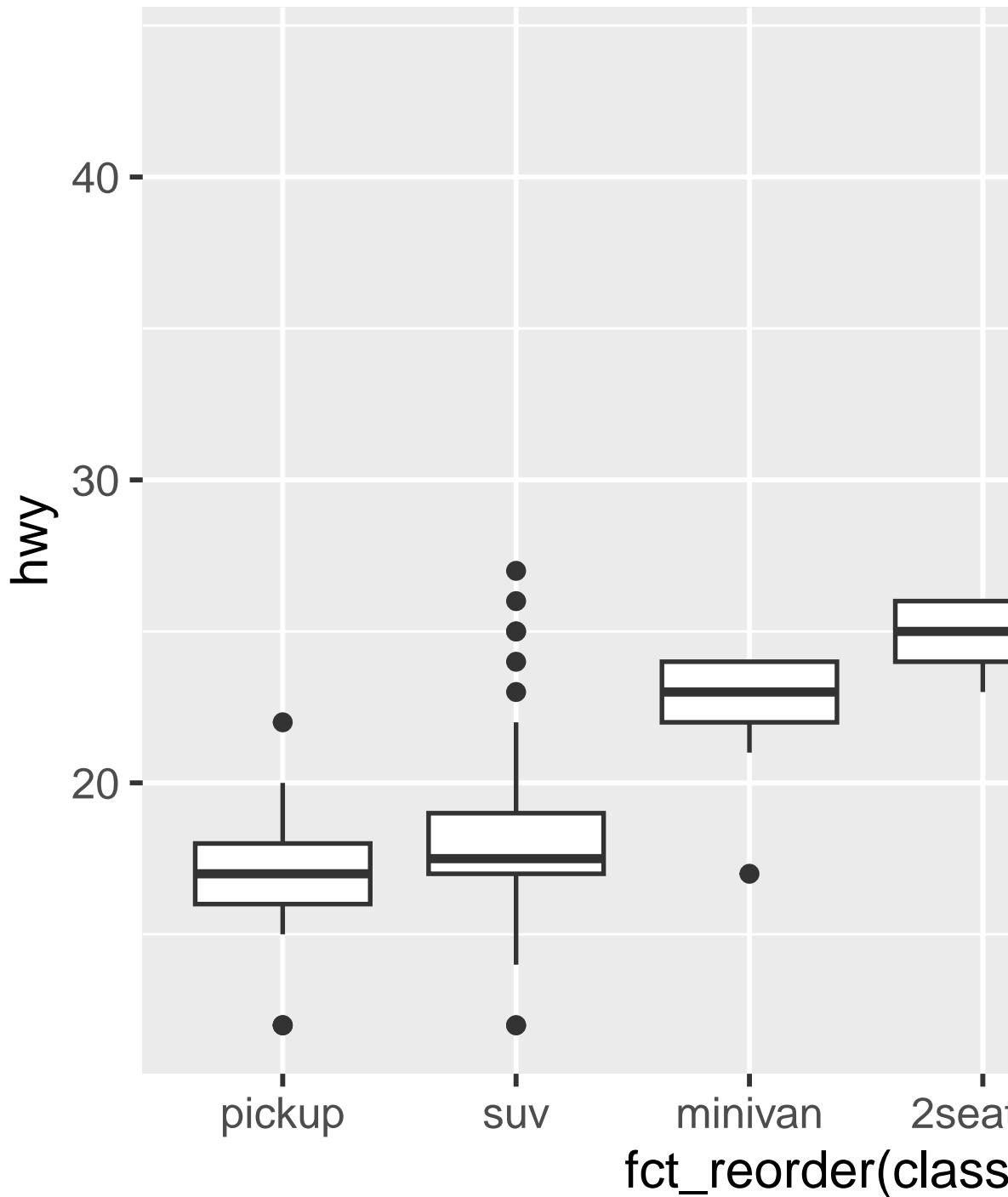
```
cut      fair     good    good    very good  
fct_reorder()        ??  
mpg      class
```

```
ggplot(mpg, aes(x = class, y = hwy)) +  
  geom_boxplot()
```



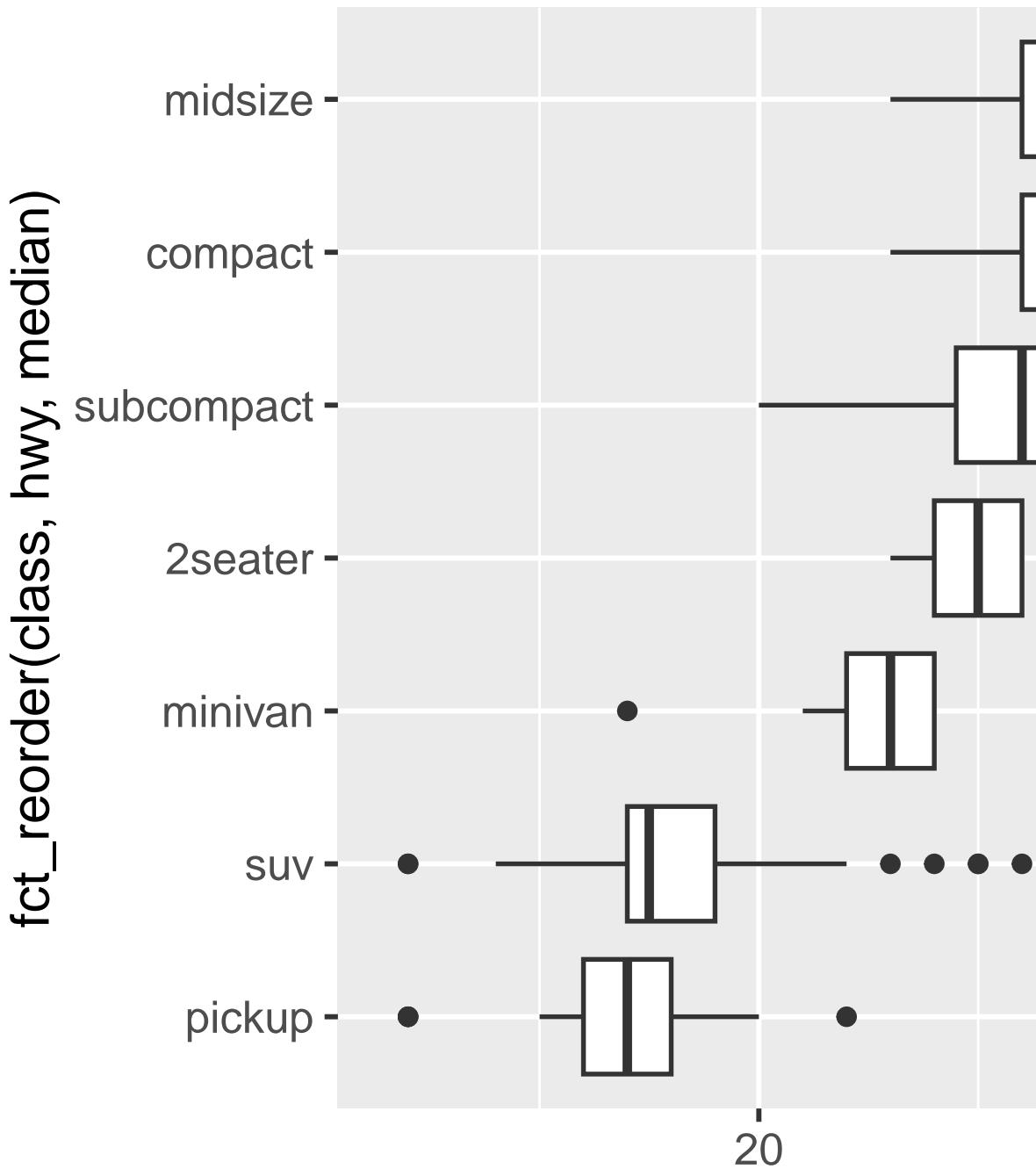
```
hwy    class  :
```

```
ggplot(mpg, aes(x = fct_reorder(class, hwy, median), y = hwy)) +  
  geom_boxplot()
```



```
geom_boxplot() 90°      x  y
```

```
ggplot(mpg, aes(x = hwy, y = fct_reorder(class, hwy, median))) +  
  geom_boxplot()
```



10.5.1.1

1.

2. EDA diamonds cut

3. x y coord_flip()

4. “ ” lvplot geom_lv() price cut

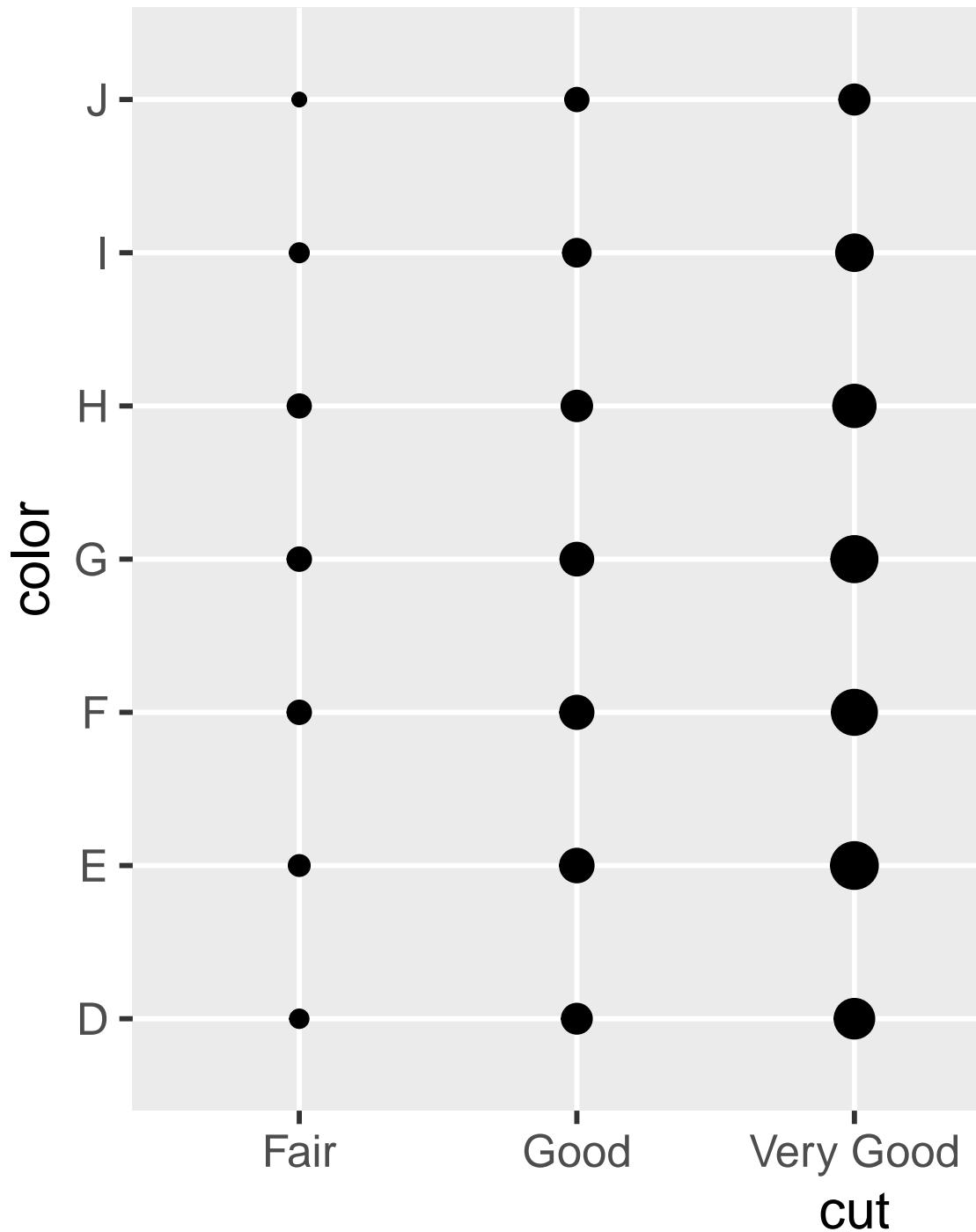
`geom_violin()` `geom_histogram()` `geom_freqpoly()` `geom_density()`

6. `geom_jitter()` `geom_beeswarm()` `geom_jitter()`

10.5.2

`geom_count()`

```
ggplot(diamonds, aes(x = cut, y = color)) +  
  geom_count()
```



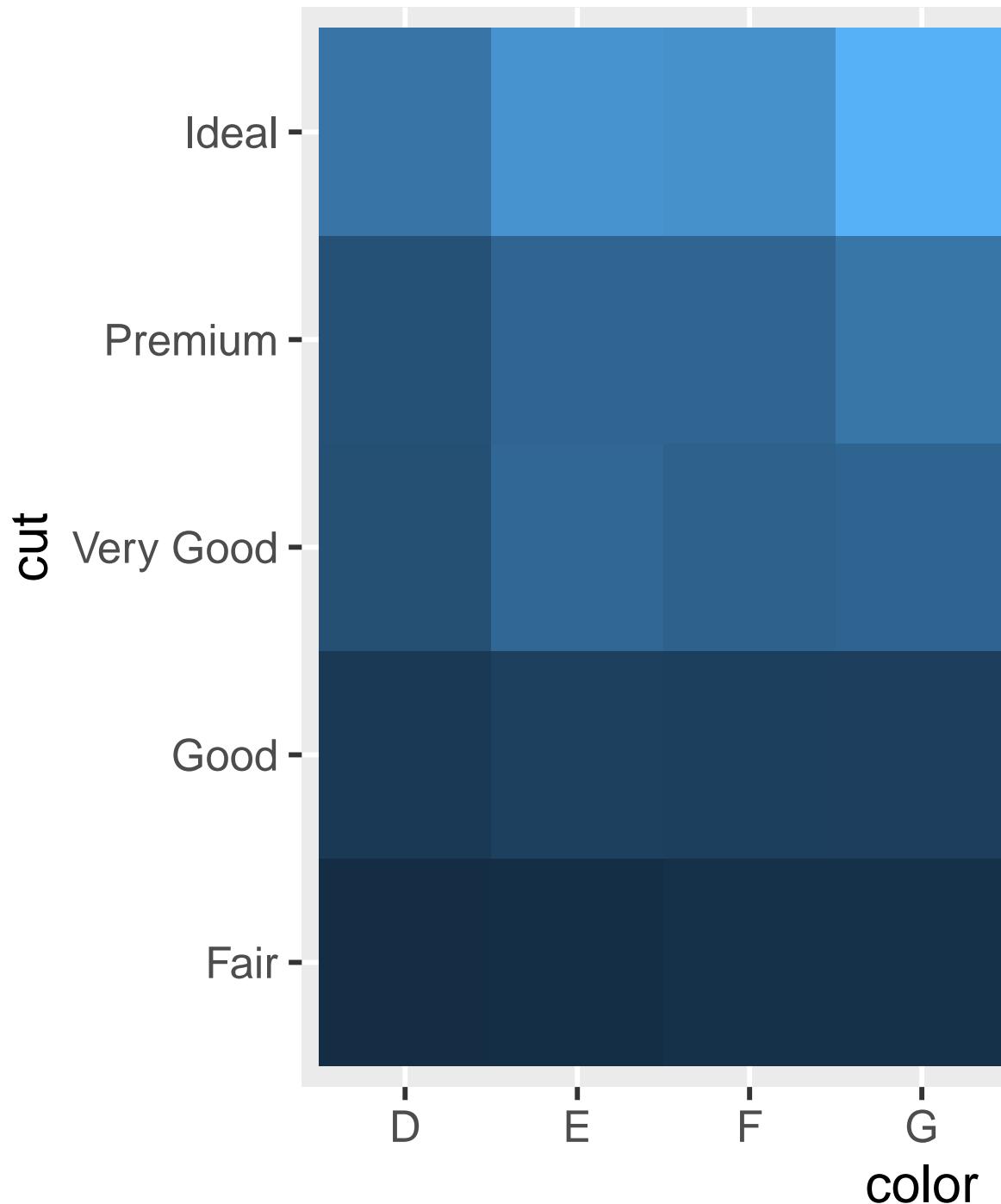
```
x    y
```

```
dplyr
```

```
diamonds |>
  count(color, cut)
#> # A tibble: 35 x 3
#>   color     cut       n
#>   <ord> <ord>     <int>
#> 1 D       Fair      163
#> 2 D       Good      662
#> 3 D       Very Good 1513
#> 4 D       Premium   1603
#> 5 D       Ideal     2834
#> 6 E       Fair      224
#> # i 29 more rows
```

```
geom_tile()      :
```

```
diamonds |>
  count(color, cut) |>
  ggplot(aes(x = color, y = cut)) +
  geom_tile(aes(fill = n))
```



```
seriation           heatmaply
```

10.5.2.1

1. color cut cut color

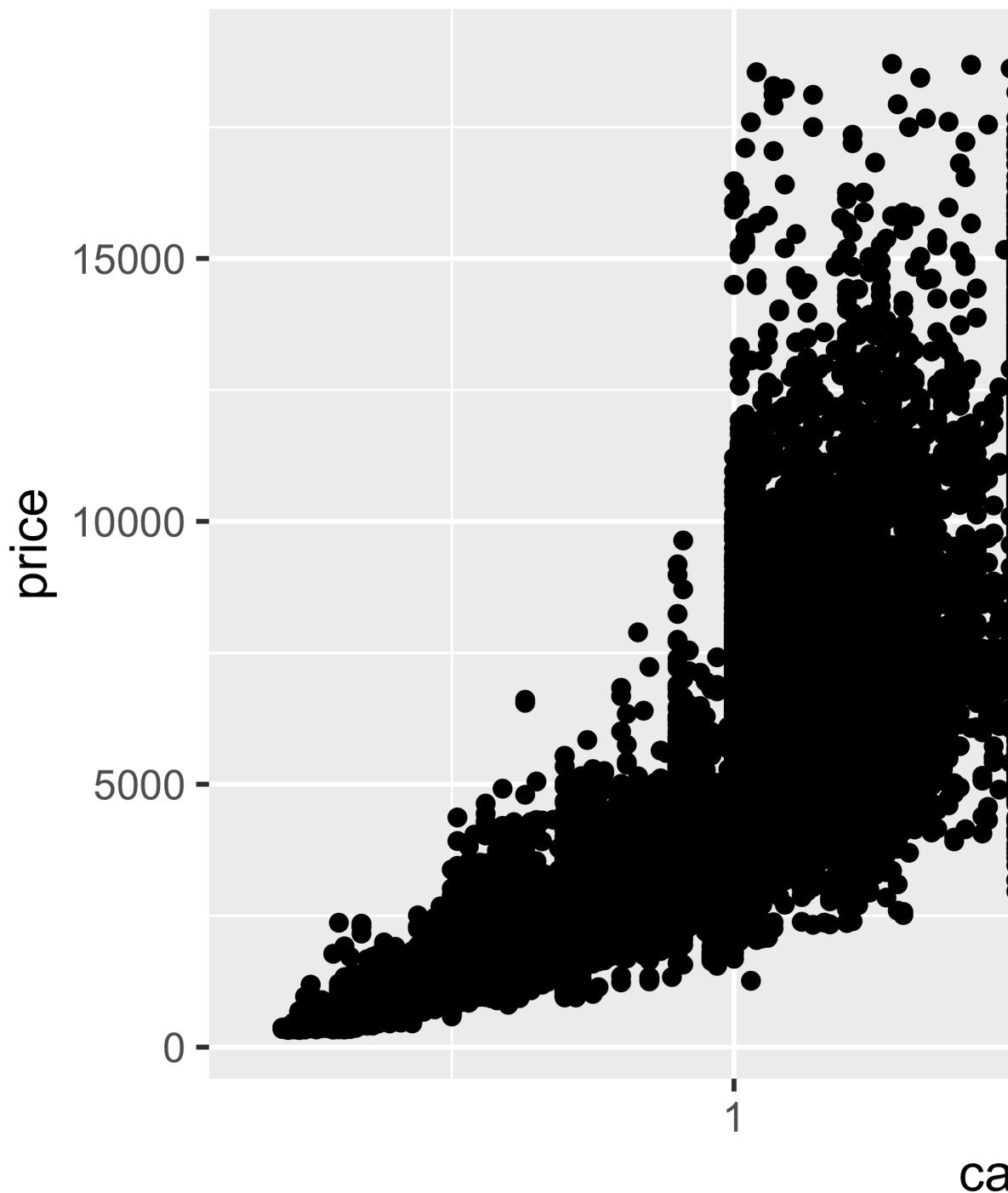
2. color x cut fill

3. geom_tile() dplyr

10.5.3

```
geom_point()
```

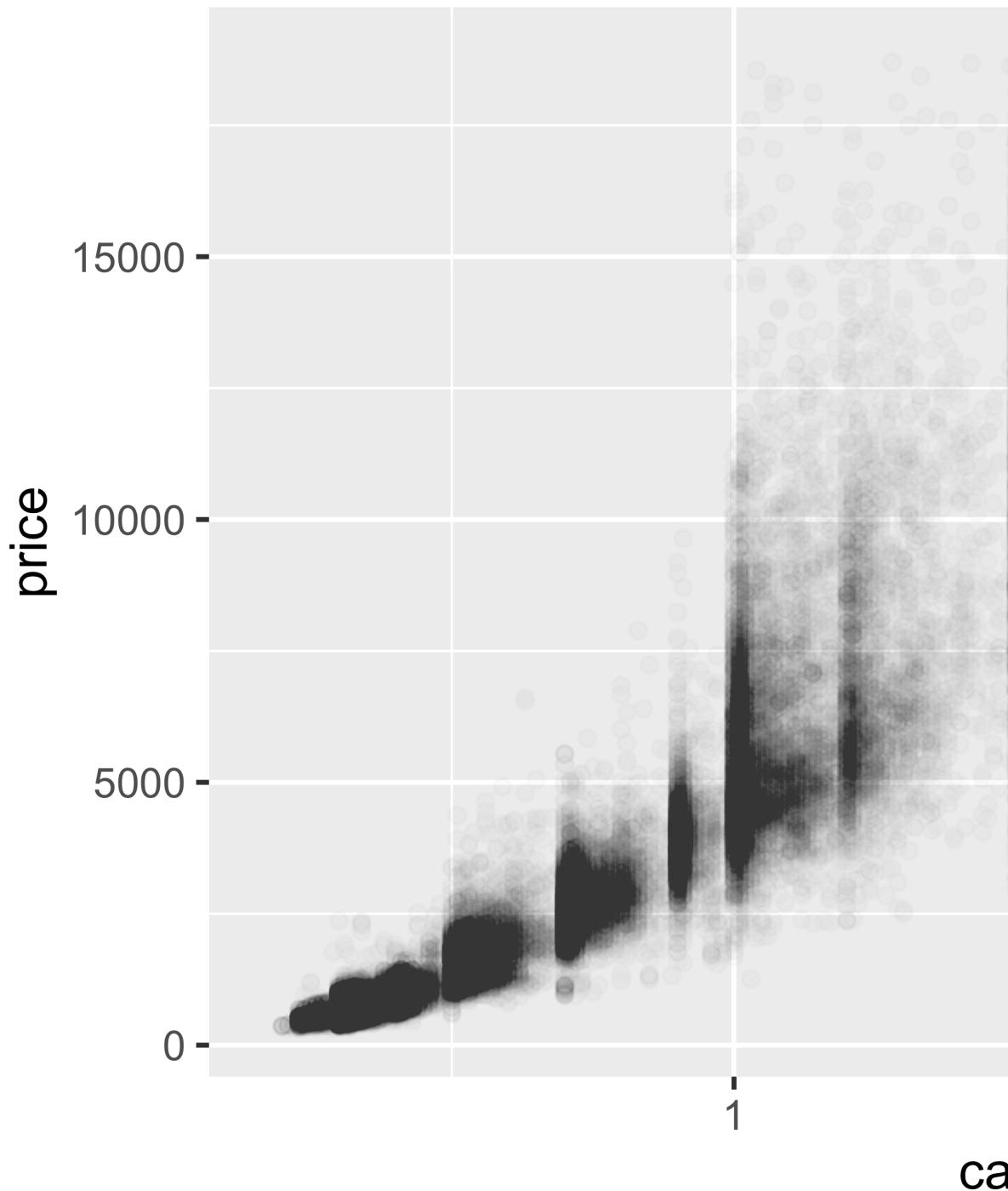
```
ggplot(smaller, aes(x = carat, y = price)) +  
  geom_point()
```



```
(    smaller      3  )
```

```
alpha
```

```
ggplot(smaller, aes(x = carat, y = price)) +  
  geom_point(alpha = 1 / 100)
```



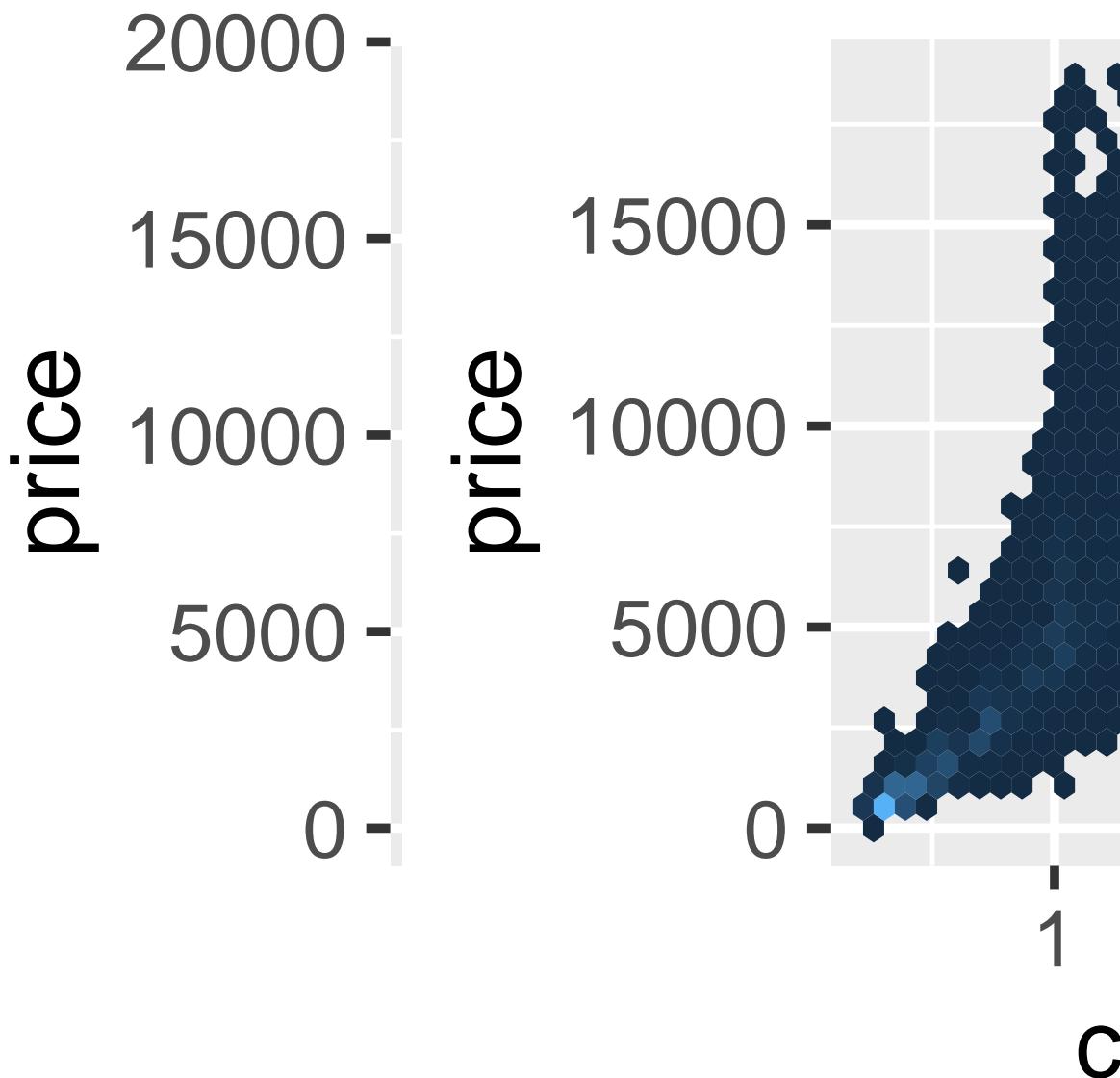
```
bin      geom_histogram() geom_freqpoly()  
geom_bin2d() geom_hex()
```

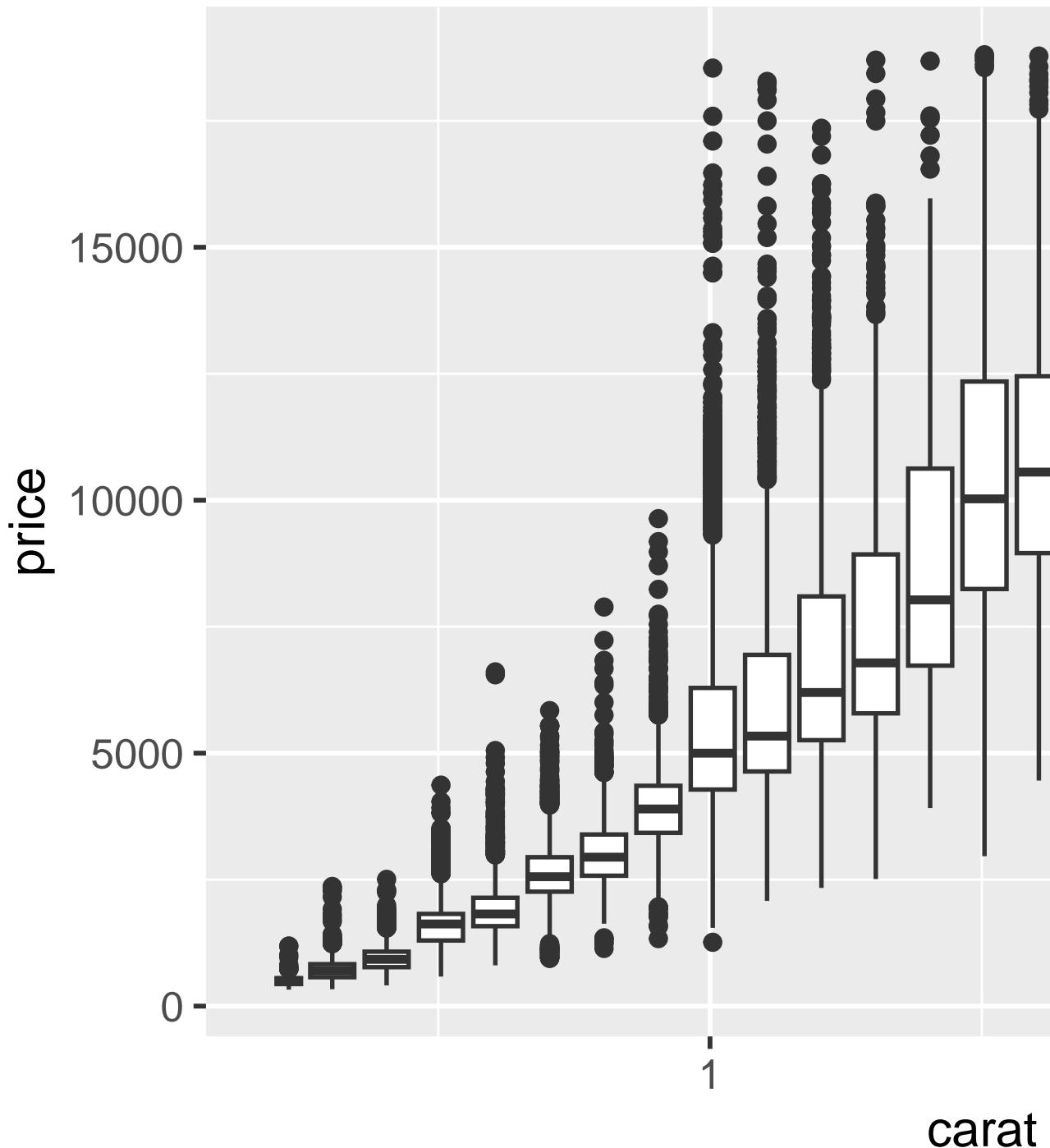
```
geom_bin2d() geom_hex()      bins      g  geom_bin2d()      g  
geom_hex()      geom_hex()    hexbin
```

```
ggplot(smaller, aes(x = carat, y = price)) +  
  geom_bin2d()  
  
# install.packages("hexbin")  
ggplot(smaller, aes(x = carat, y = price)) +  
  geom_hex()
```

carat

```
ggplot(smaller, aes(x = carat, y = price)) +  
  geom_boxplot(aes(group = cut_width(carat, 0.1)))
```





```
cut_width(x, width) x    width
                  varwidth = TRUE
```

10.5.3.1

```
1.                      cut_width() cut_number()      carat price
2.  price  carat
3.
4.          cut carat price
5.          x  y          x  y

diamonds |>
  filter(x >= 4) |>
  ggplot(aes(x = x, y = y)) +
  geom_point() +
  coord_cartesian(xlim = c(4, 11), ylim = c(4, 11))

6.  cut_number()      cut_width()

ggplot(smaller, aes(x = carat, y = price)) +
  geom_boxplot(aes(group = cut_number(carat, 20)))
```

10.6

-
-
-
-
-

carat price

price carat

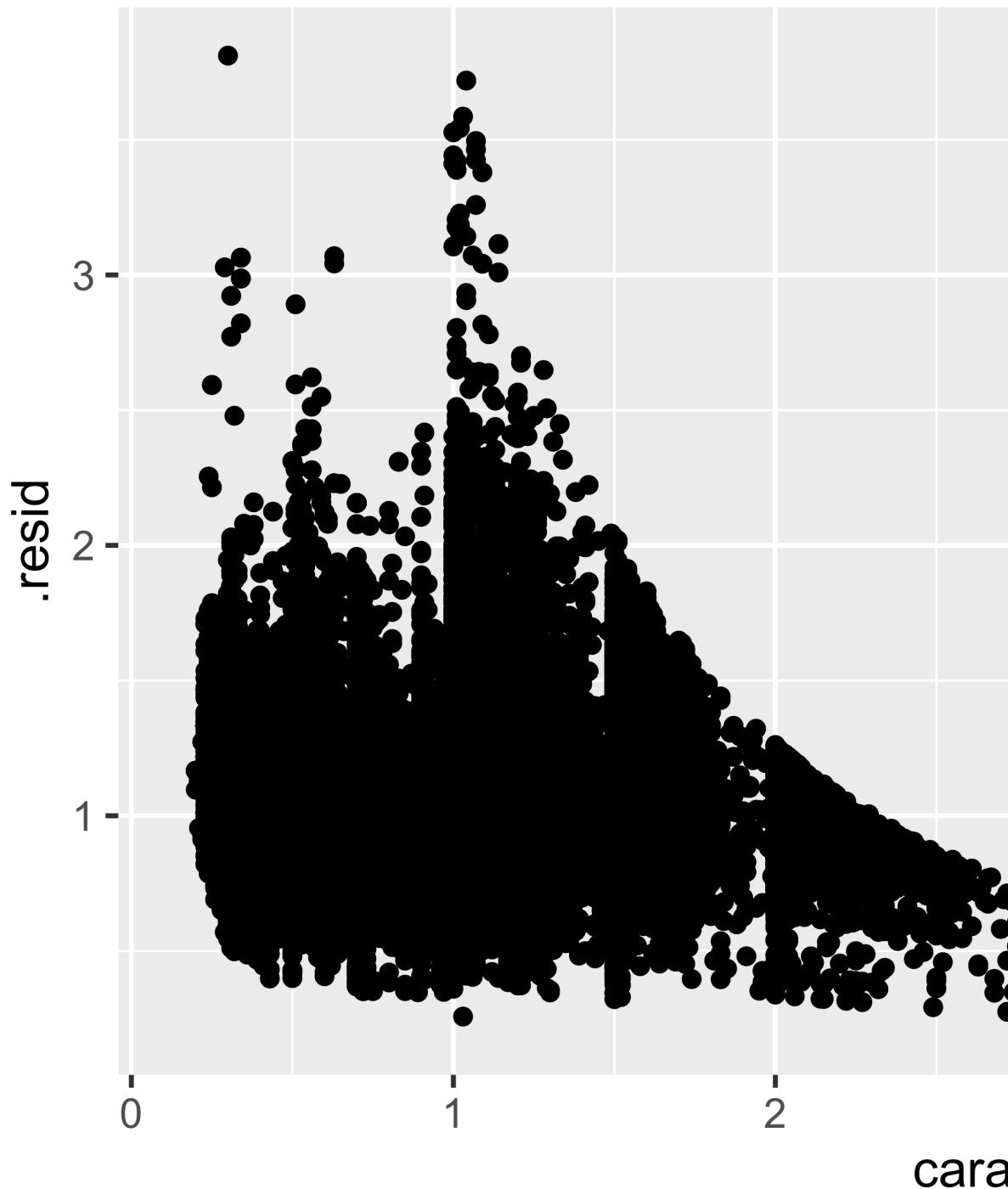
```
library(tidymodels)

diamonds <- diamonds |>
  mutate(
    log_price = log(price),
    log_carat = log(carat)
  )

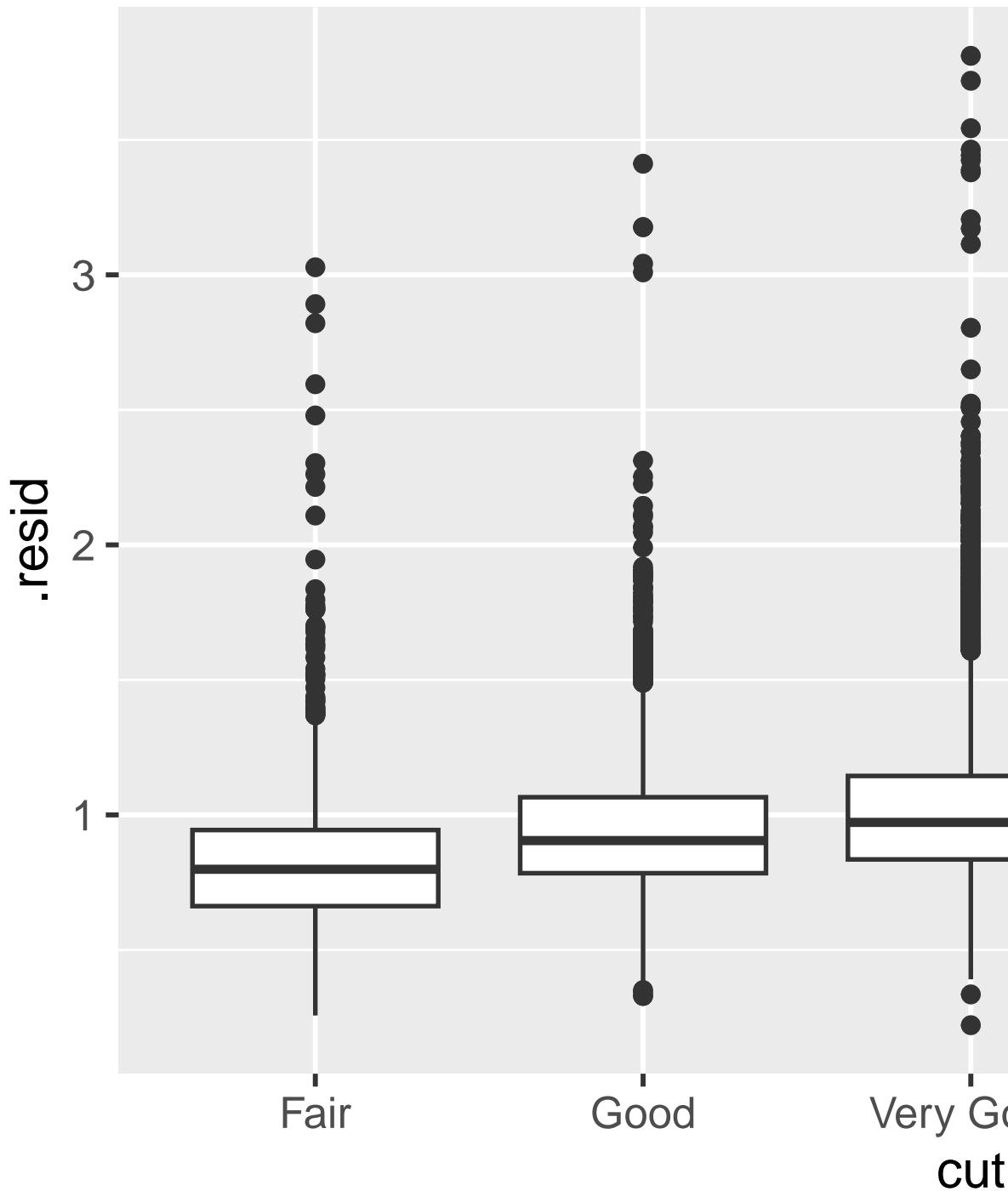
diamonds_fit <- linear_reg() |>
  fit(log_price ~ log_carat, data = diamonds)

diamonds_aug <- augment(diamonds_fit, new_data = diamonds) |>
  mutate(.resid = exp(.resid))

ggplot(diamonds_aug, aes(x = carat, y = .resid)) +
  geom_point()
```



```
ggplot(diamonds_aug, aes(x = cut, y = .resid)) +  
  geom_boxplot()
```



10.7.

247

10.7

Chapter 11

11.1

??
ggplot2
Albert Cairo The Truthful
Art

11.1.1

ggplot2 dplyr scales ggplot2 Kamil
Slowikowski ggrepel https://ggrepel.slowkow.com Thomas Lin Peder-
sen patchwork https://patchwork.data-imaginist.com install.packages()

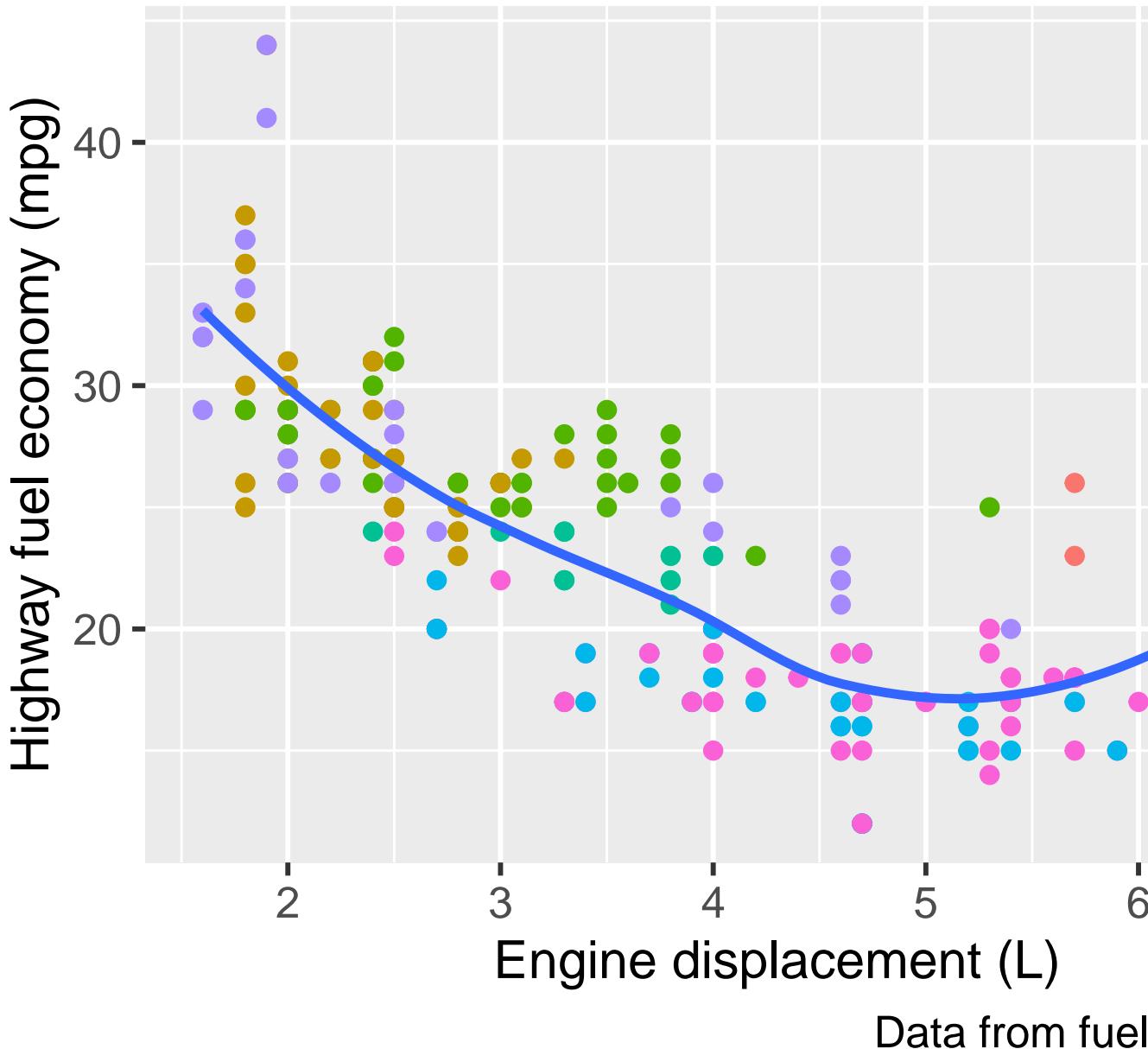
```
library(tidyverse)
library(scales)
library(ggrepel)
library(patchwork)
```

11.2

```
labs()  
  
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point(aes(color = class)) +  
  geom_smooth(se = FALSE) +
```

```
labs(  
  x = "Engine displacement (L)",  
  y = "Highway fuel economy (mpg)",  
  color = "Car type",  
  title = "Fuel efficiency generally decreases with engine size",  
  subtitle = "Two seaters (sports cars) are an exception because of their light weight",  
  caption = "Data from fueleconomy.gov"  
)
```

Fuel efficiency generally decreases with engine size.
Two seaters (sports cars) are an exception



```

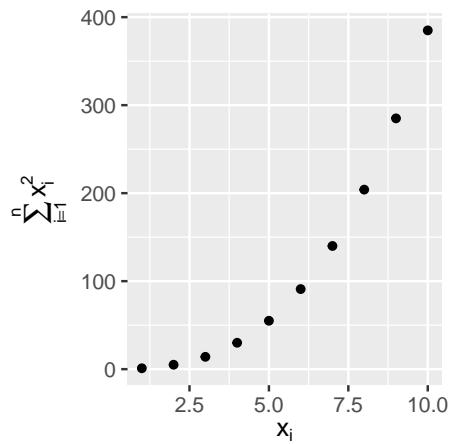
title           ""
subtitle        ""
caption         ""
labs()          ""

""  quote()  ?plotmath

df <- tibble(
  x = 1:10,
  y = cumsum(x^2)
)

ggplot(df, aes(x, y)) +
  geom_point() +
  labs(
    x = quote(x[i]),
    y = quote(sum(x[i] ^ 2, i == 1, n))
  )

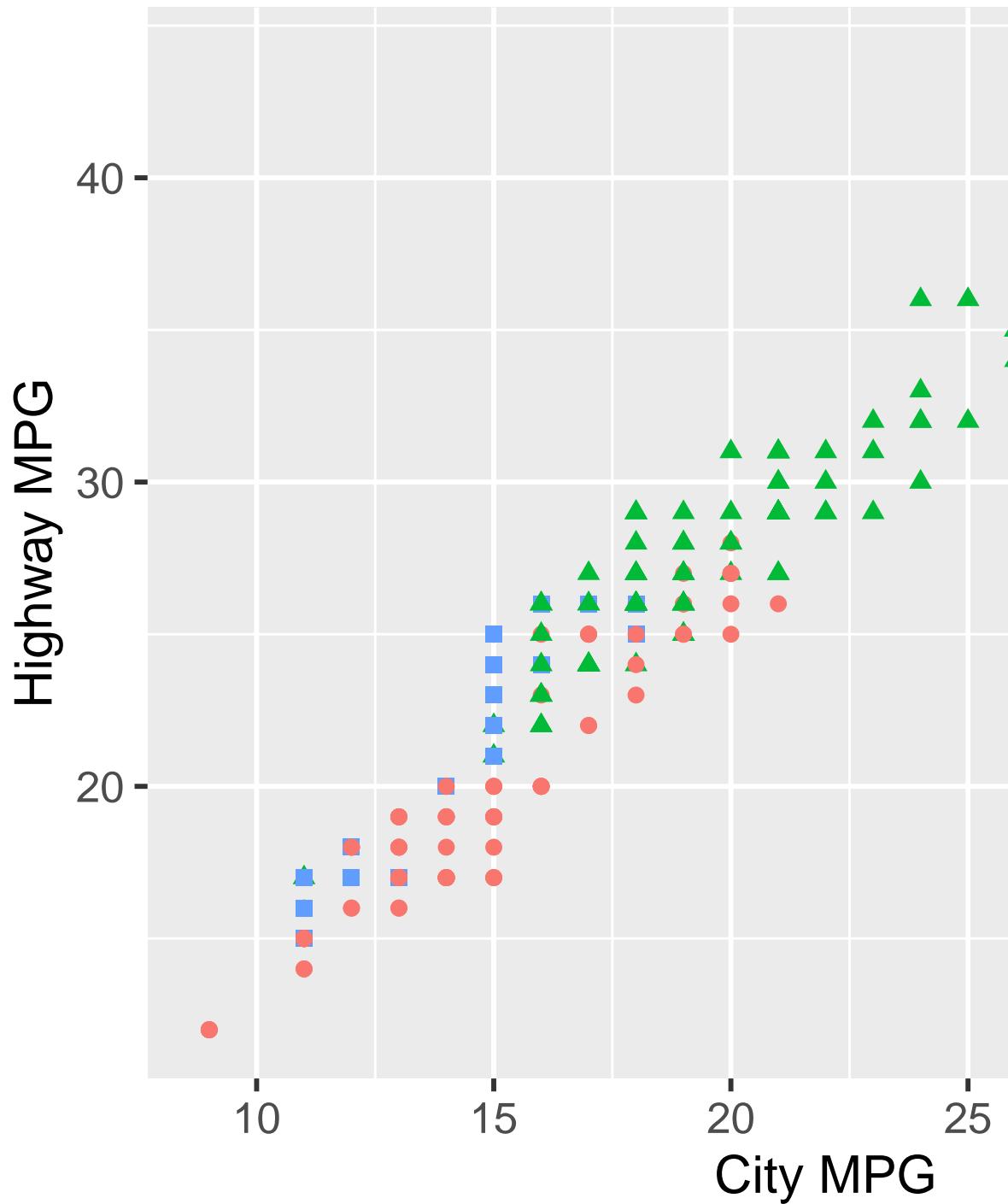
```



11.2.1

1. Create one plot on the fuel economy data with customized `title`, `subtitle`, `caption`, `x`, `y`, and `color` labels.

- 2.



3.

11.3

```

geom_text() geom_text() geom_point()      label
tibble          label_info

label_info <- mpg |>
  group_by(drv) |>
  arrange(desc(displ)) |>
  slice_head(n = 1) |>
  mutate(
    drive_type = case_when(
      drv == "f" ~ "front-wheel drive",
      drv == "r" ~ "rear-wheel drive",
      drv == "4" ~ "4-wheel drive"
    )
  ) |>
  select(displ, hwy, drv, drive_type)

label_info
#> # A tibble: 3 x 4
#> # Groups:   drv [3]
#>   displ   hwy   drv   drive_type
#>   <dbl> <int> <chr> <chr>
#> 1     6.5     17   f     4-wheel drive
#> 2     5.3     25   r     front-wheel drive
#> 3     7.0     24   r     rear-wheel drive

```

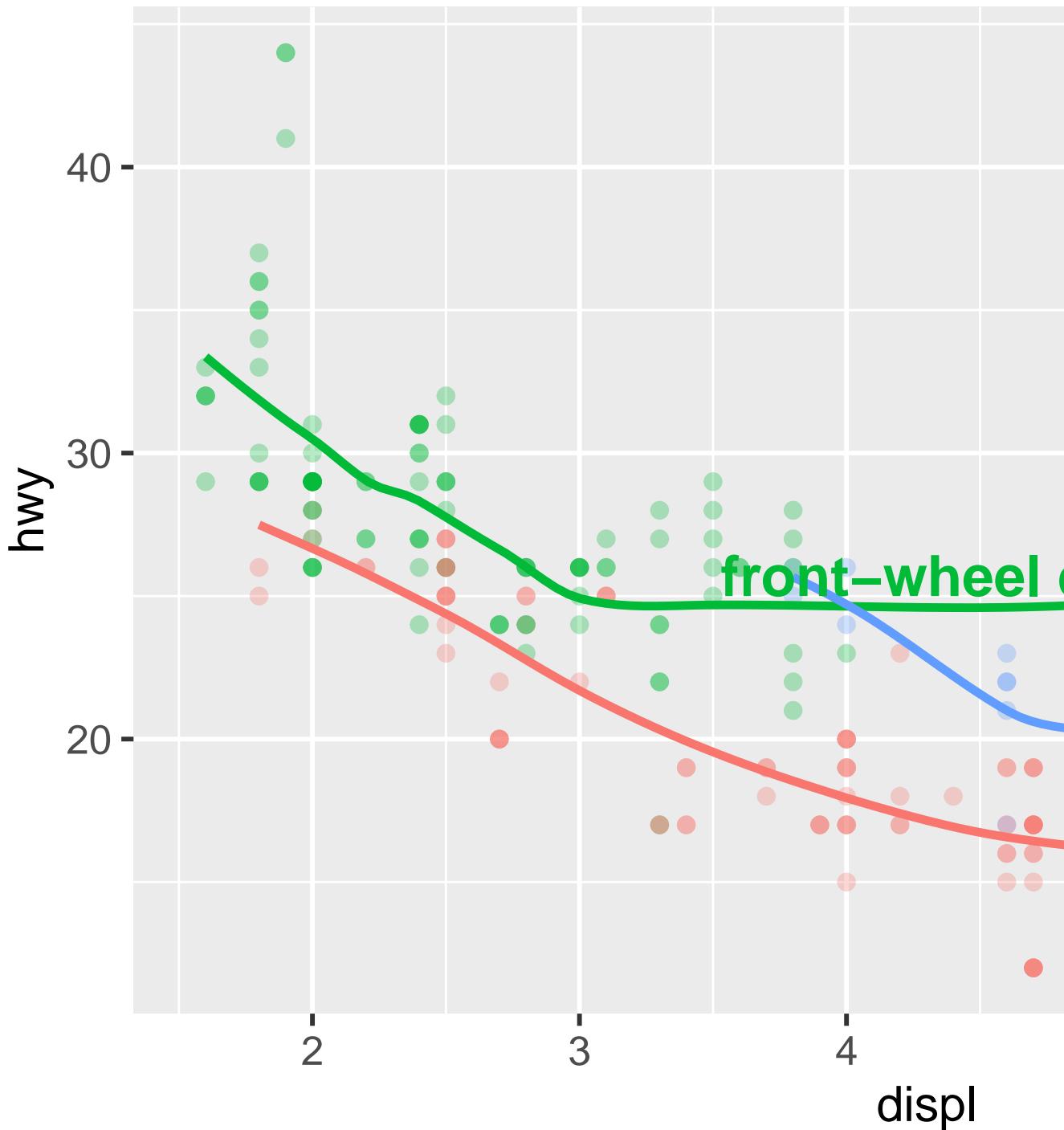


```

fontface size      theme(legend.position
= "none"

ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point(alpha = 0.3) +
  geom_smooth(se = FALSE) +
  geom_text(
    data = label_info,
    aes(x = displ, y = hwy, label = drive_type),
    fontface = "bold", size = 5, hjust = "right", vjust = "bottom"
  ) +
  theme(legend.position = "none")
#> `geom_smooth()` using method = 'loess' and formula = 'y ~ x'

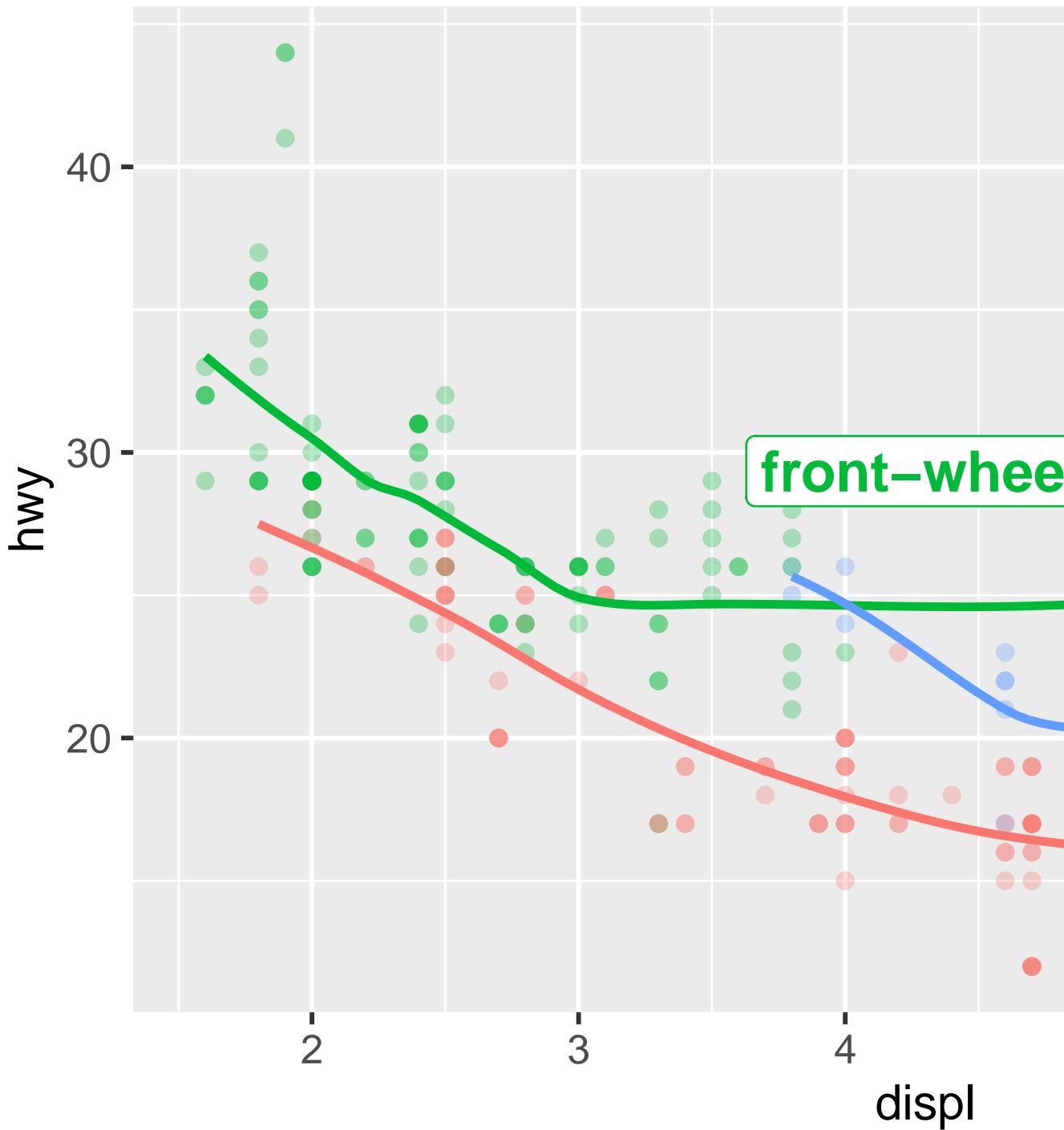
```



```
hjust      vjust
```

```
ggrepel geom_label_repel()
```

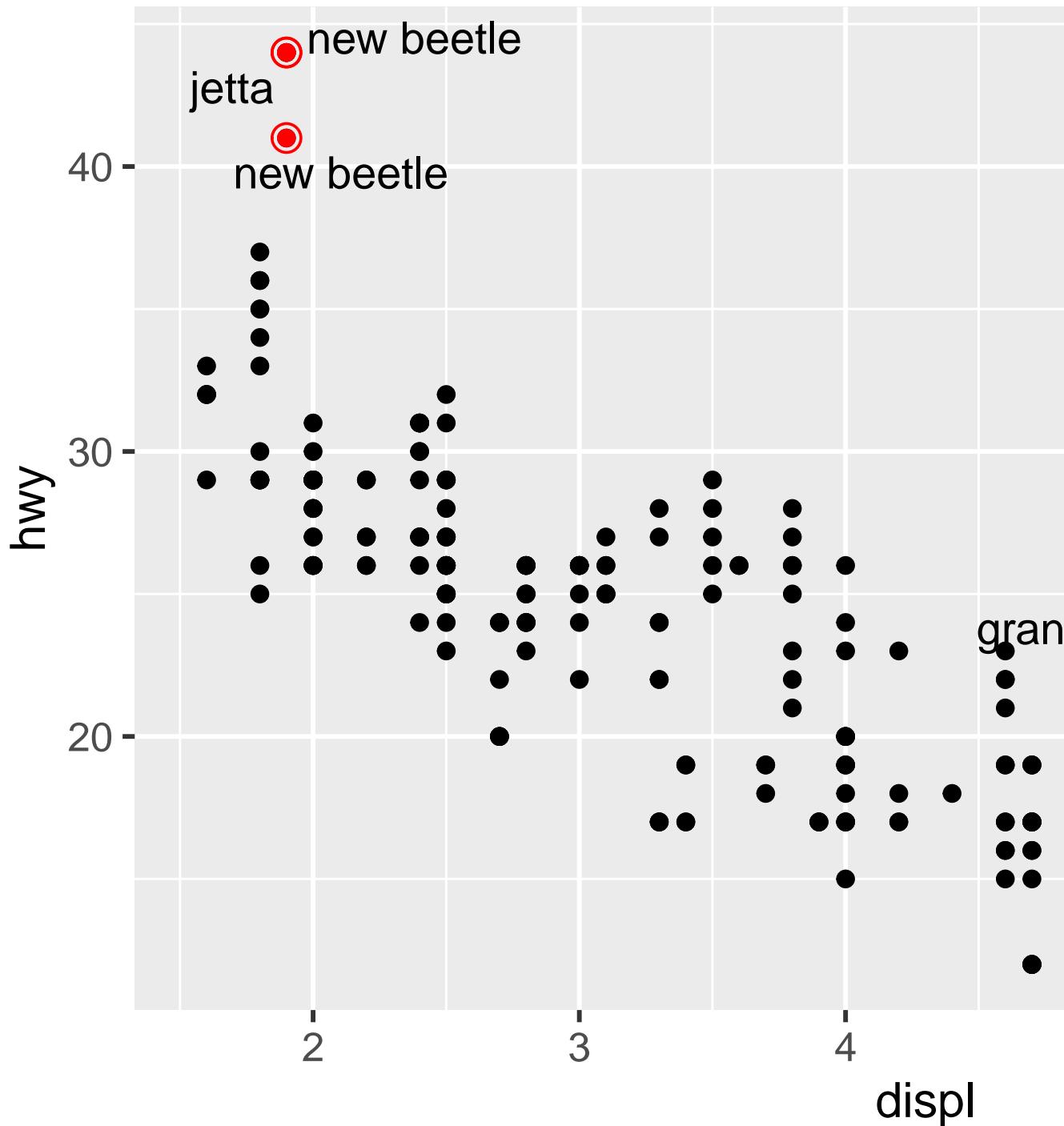
```
ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point(alpha = 0.3) +
  geom_smooth(se = FALSE) +
  geom_label_repel(
    data = label_info,
    aes(x = displ, y = hwy, label = drive_type),
    fontface = "bold", size = 5, nudge_y = 2
  ) +
  theme(legend.position = "none")
#> `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



```
ggrepel geom_text_repel()
```

```
potential_outliers <- mpg |>
  filter(hwy > 40 | (hwy > 20 & displ > 5))

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  geom_text_repel(data = potential_outliers, aes(label = model)) +
  geom_point(data = potential_outliers, color = "red") +
  geom_point(
    data = potential_outliers,
    color = "red", size = 3, shape = "circle open"
  )
```



```

geom_text() geom_label() ggplot2     geoms

• geom_hline() geom_vline()           linewidth = 2   color =
white

• geom_rect()                      xmin, xmax, ymin, ymax      ggforce  geom_mark_hull()

• arrow  geom_segment()            x y      xend yend

annotate()      geoms      annotate()

annotate()      stringr::str_wrap()

trend_text <- "Larger engine sizes tend to have lower fuel economy." |>
  str_wrap(width = 30)
trend_text
#> [1] "Larger engine sizes tend to\nhave lower fuel economy."

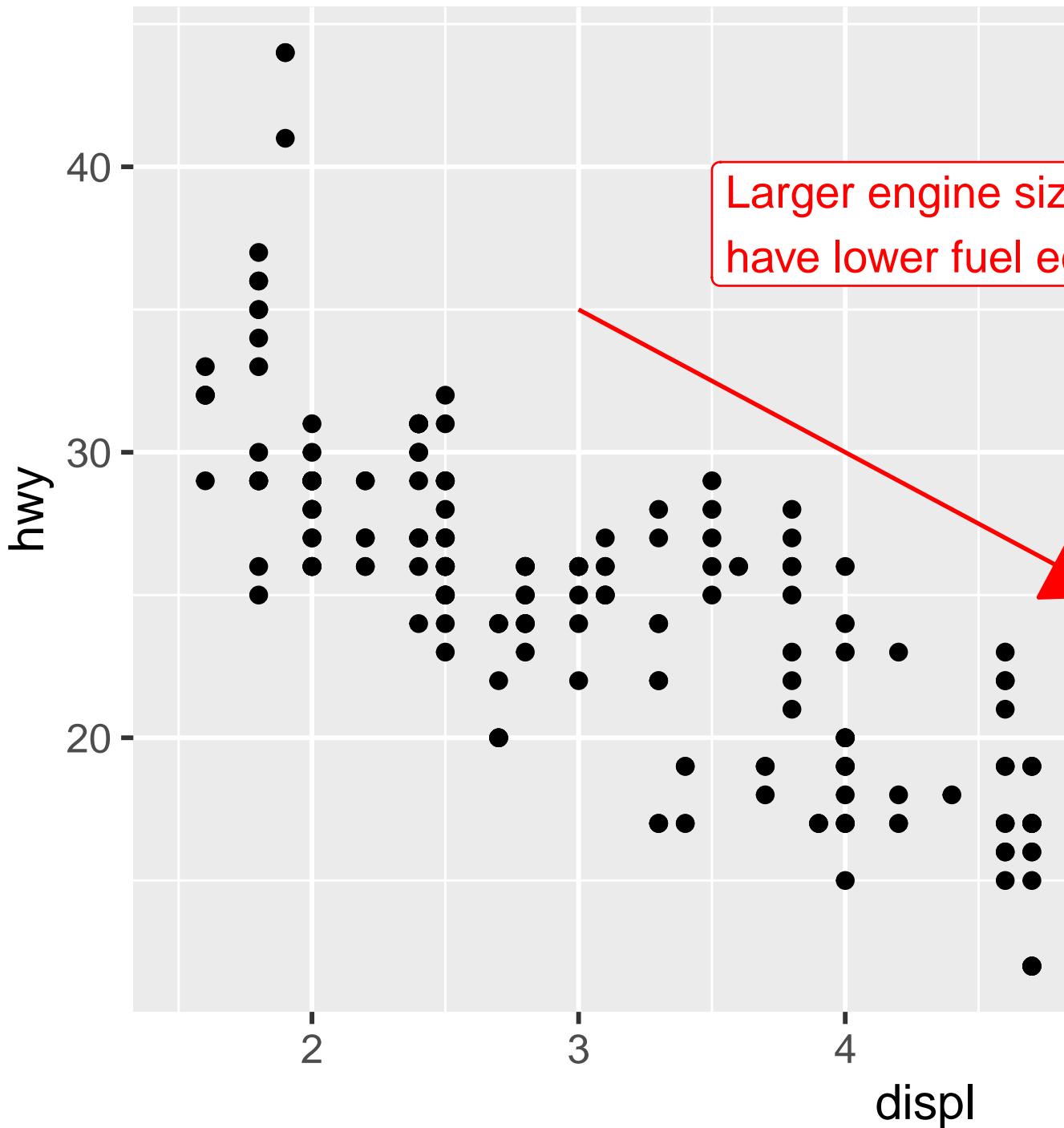
```

```

geom      geom      x y      xend yend

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  annotate(
    geom = "label", x = 3.5, y = 38,
    label = trend_text,
    hjust = "left", color = "red"
  ) +
  annotate(
    geom = "segment",
    x = 3, y = 35, xend = 5, yend = 25, color = "red",
    arrow = arrow(type = "closed")
  )

```



11.3.1

1. `geom_text()`
2. `annotate()` tibble
3. `geom_text()` `geom_text()`
4. `geom_label()`
5. `arrow()`

11.4

11.4.1

`ggplot2`

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class))
```

`ggplot2` :

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class)) +
  scale_x_continuous() +
  scale_y_continuous() +
  scale_color_discrete()
```

```
x   scale_color_discrete()                                scale_x_continuous() displ
```

•
•

```
“ ”      x  y
      breaks    labels b  reaks      l  abels      /      b
reaks
```

11.4.2

“ ” x y

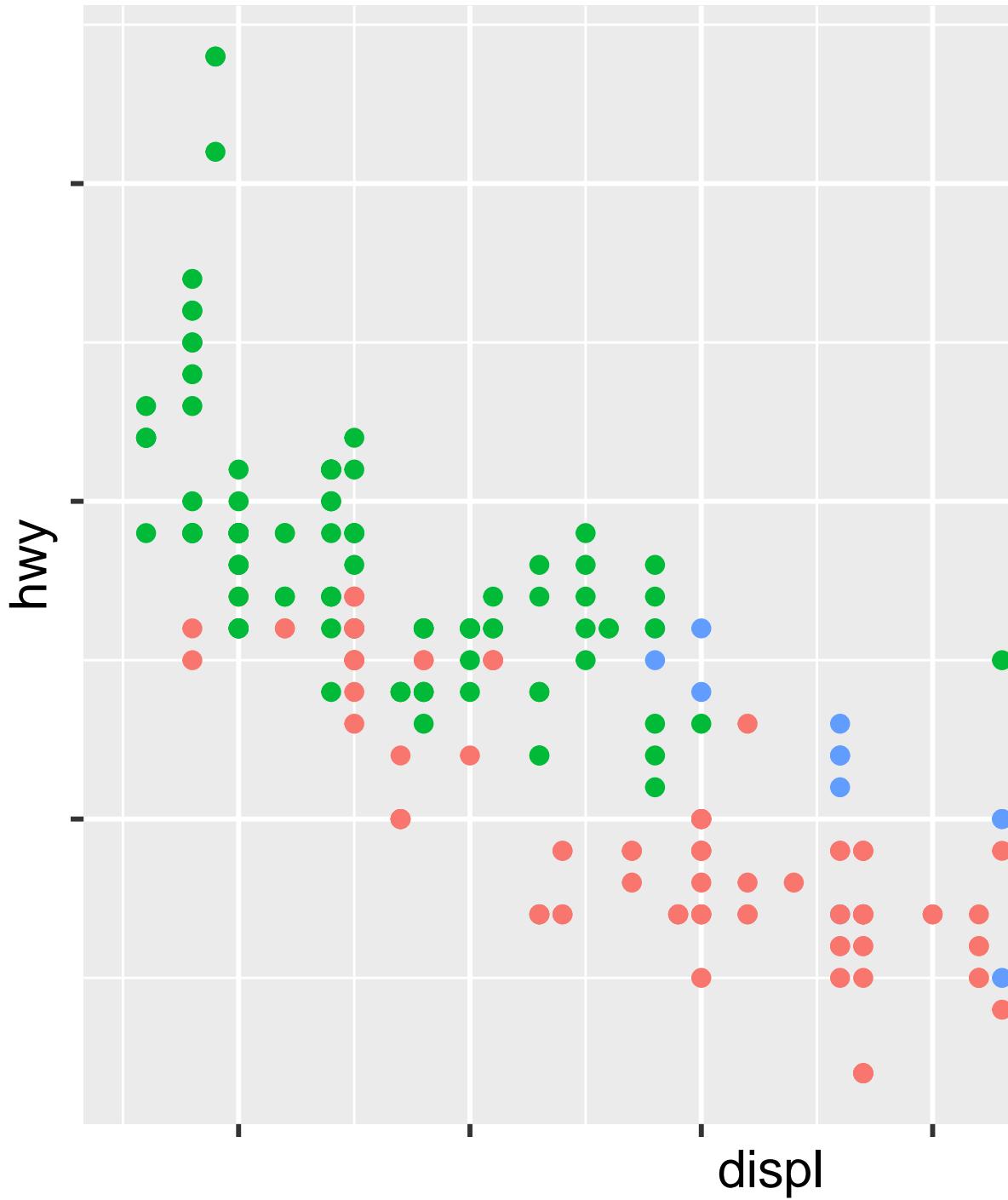
breaks labels breaks labels / breaks

```
ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +  
  geom_point() +  
  scale_y_continuous(breaks = seq(15, 40, by = 5))
```



```
  labels  breaks      NULL  
breaks labels      labels
```

```
ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +  
  geom_point() +  
  scale_x_continuous(labels = NULL) +  
  scale_y_continuous(labels = NULL) +  
  scale_color_discrete(labels = c("4" = "4-wheel", "f" = "front", "r" = "rear"))
```



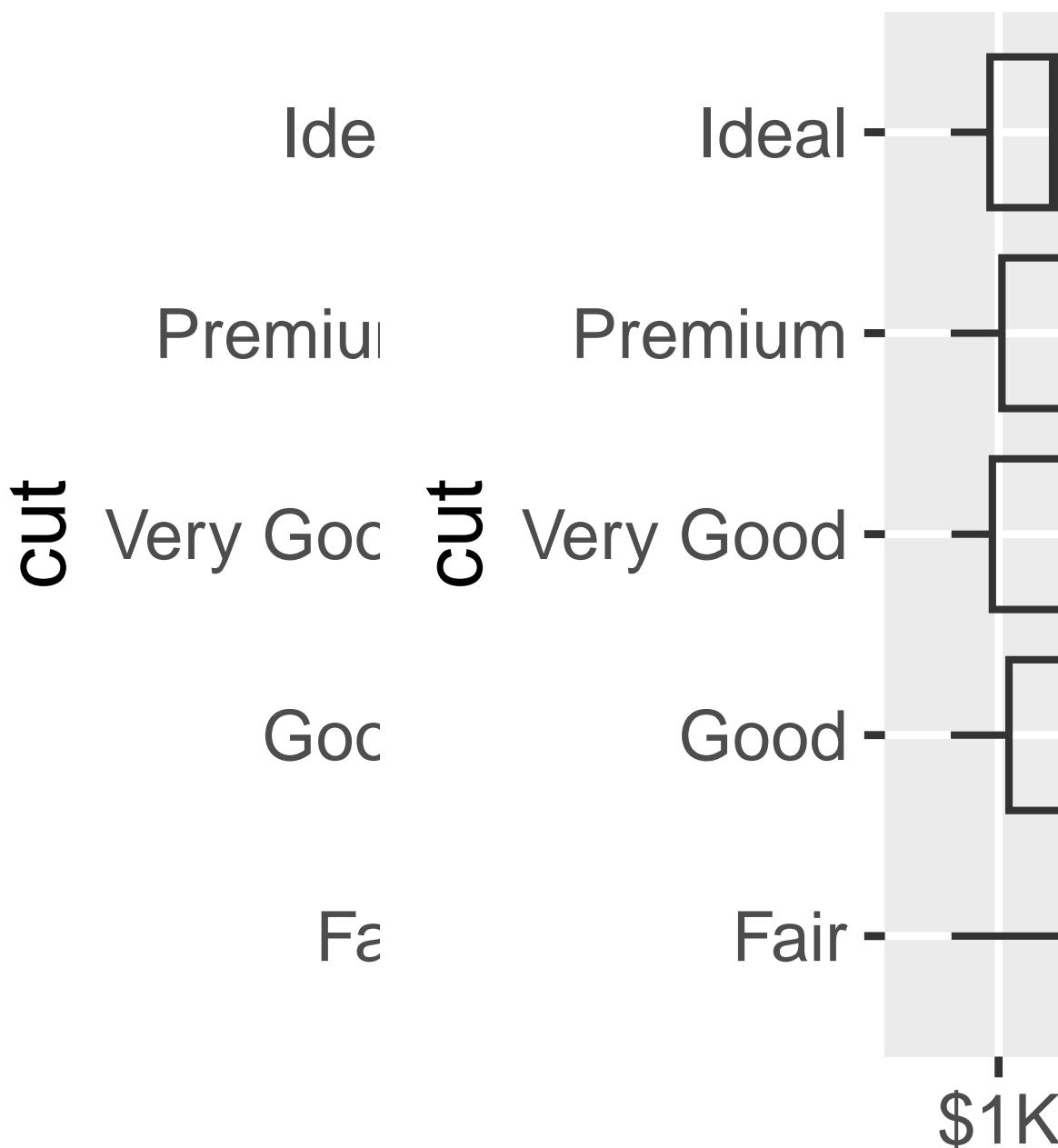
```
labels    scales                      label_dollar()  
        1,000   "K"  ""                 breaks
```

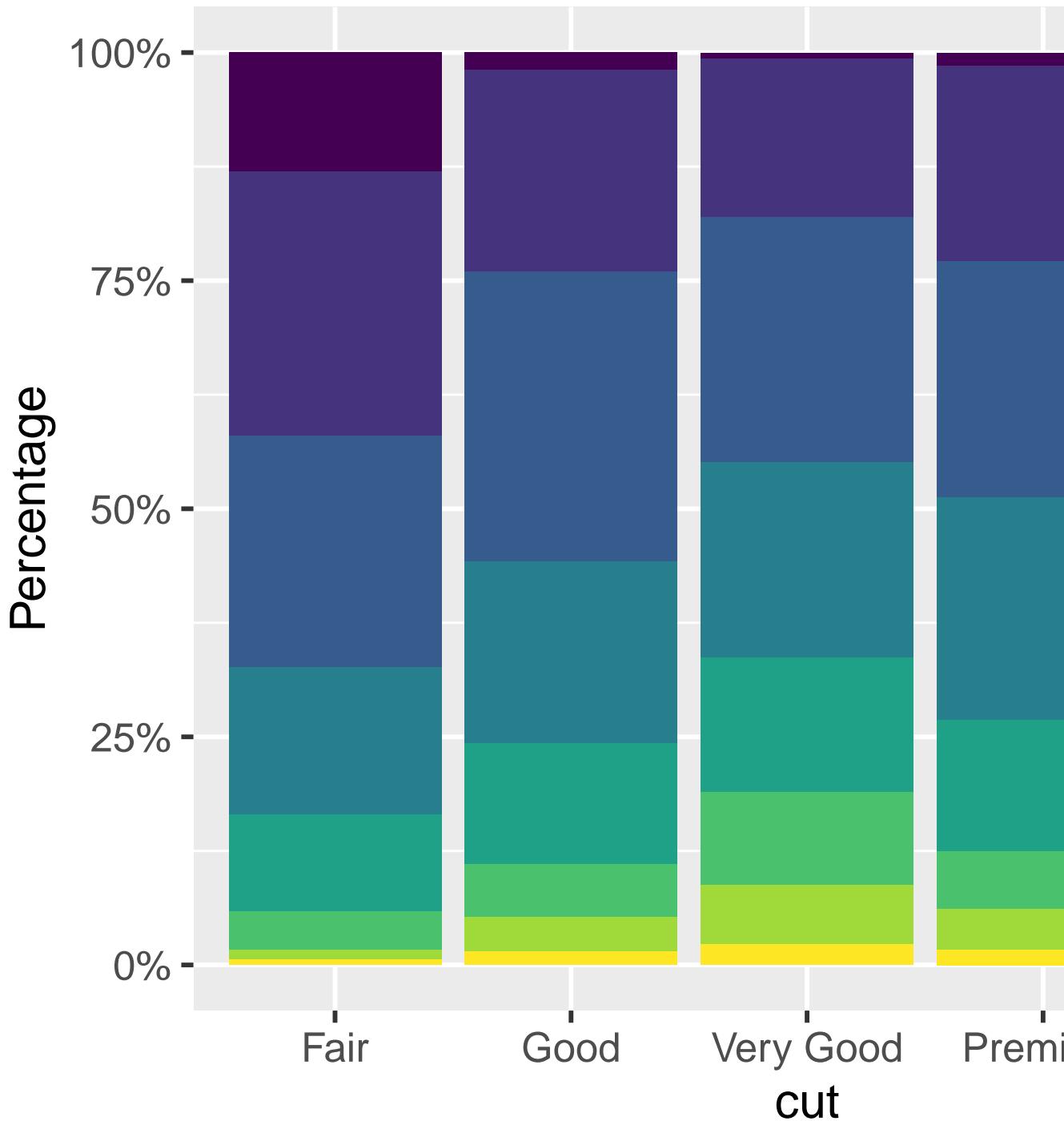
```
# Left
ggplot(diamonds, aes(x = price, y = cut)) +
  geom_boxplot(alpha = 0.05) +
  scale_x_continuous(labels = label_dollar())

# Right
ggplot(diamonds, aes(x = price, y = cut)) +
  geom_boxplot(alpha = 0.05) +
  scale_x_continuous(
    labels = label_dollar(scale = 1/1000, suffix = "K"),
    breaks = seq(1000, 19000, by = 6000)
  )
```

```
label_percent():
```

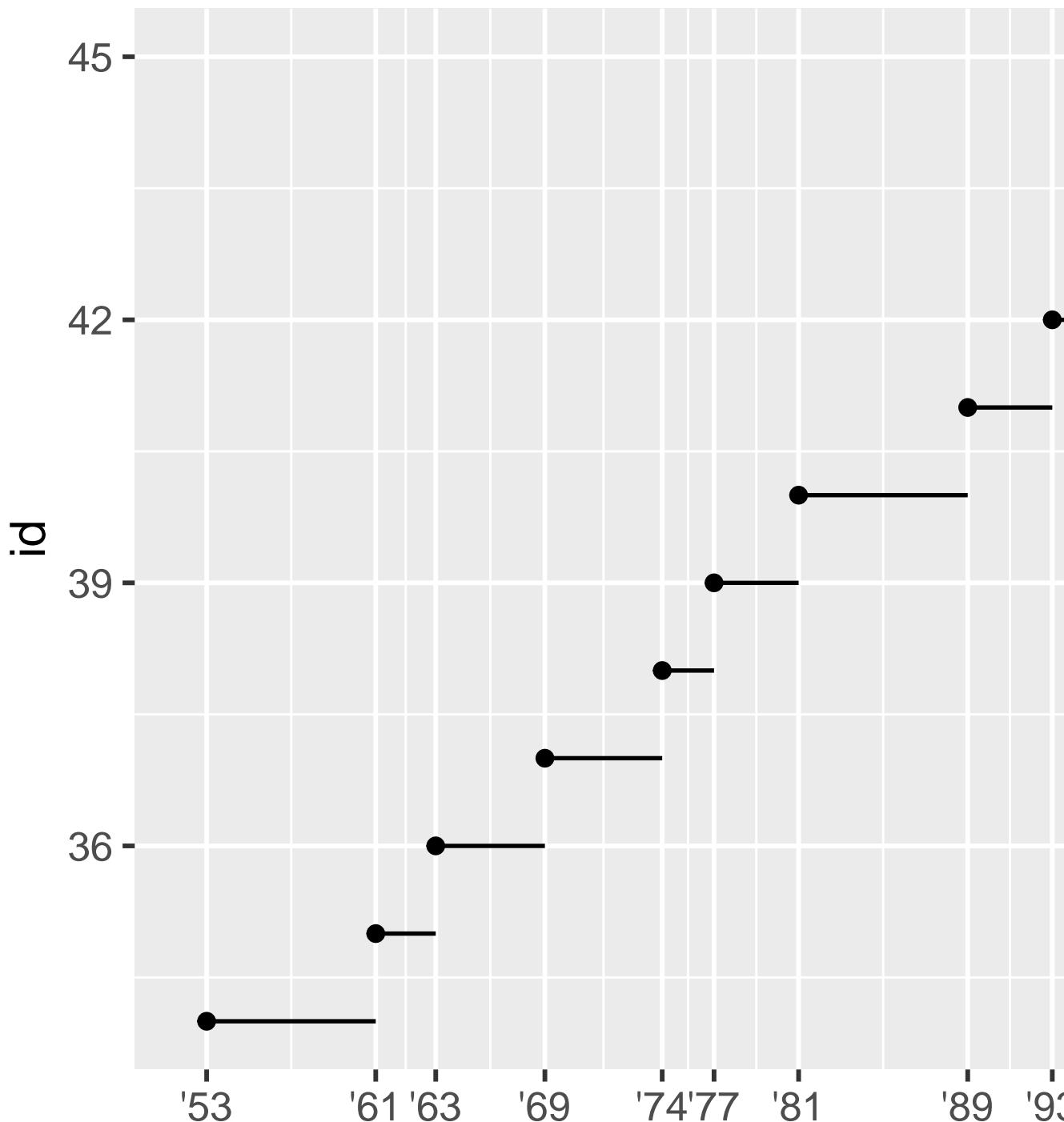
```
ggplot(diamonds, aes(x = cut, fill = clarity)) +  
  geom_bar(position = "fill") +  
  scale_y_continuous(name = "Percentage", labels = label_percent())
```





breaks

```
presidential |>
  mutate(id = 33 + row_number()) |>
  ggplot(aes(x = start, y = id)) +
  geom_point() +
  geom_segment(aes(xend = end, yend = id)) +
  scale_x_date(name = NULL, breaks = presidential$start, date_labels = "'%y")
```



```

breaks      start      presidential$start           breaks labels

• date_labels      parse_datetime()

• date_breaks     "2 days"  "1 month"

```

11.4.3

```

breaks labels

theme()           legend.position

```

```

base <- ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class))

base + theme(legend.position = "right") # the default
base + theme(legend.position = "left")
base +
  theme(legend.position = "top") +
  guides(color = guide_legend(nrow = 3))
base +
  theme(legend.position = "bottom") +
  guides(color = guide_legend(nrow = 3))

```

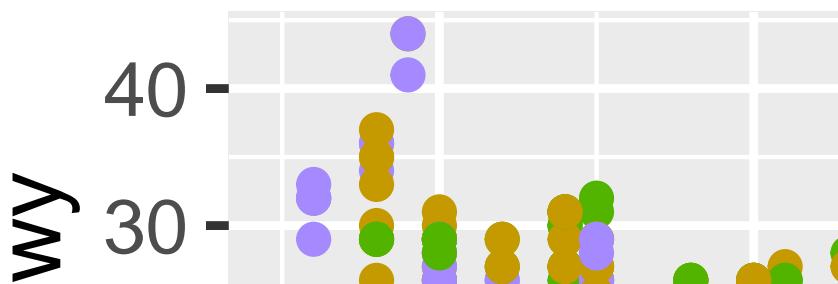
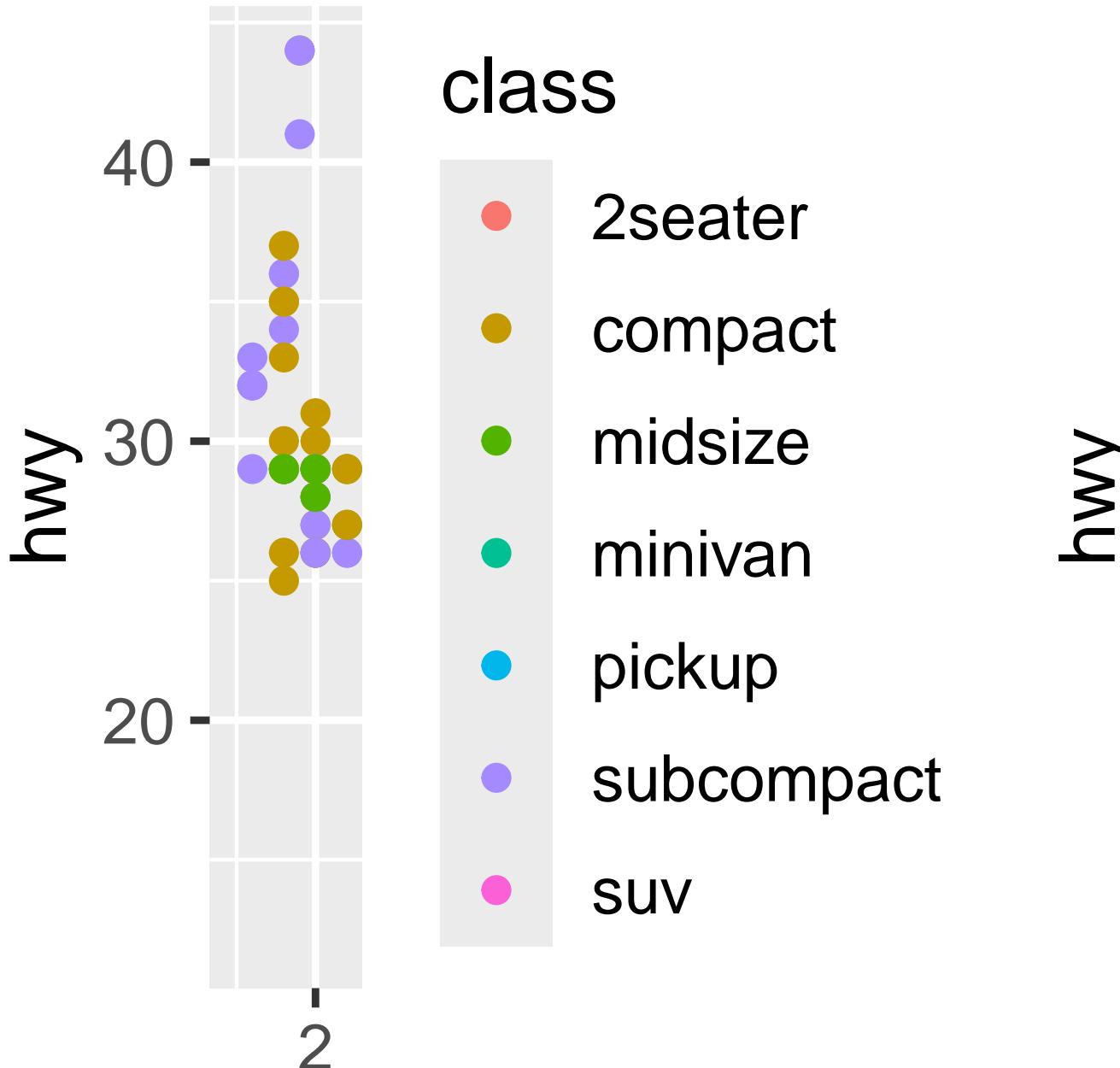
```

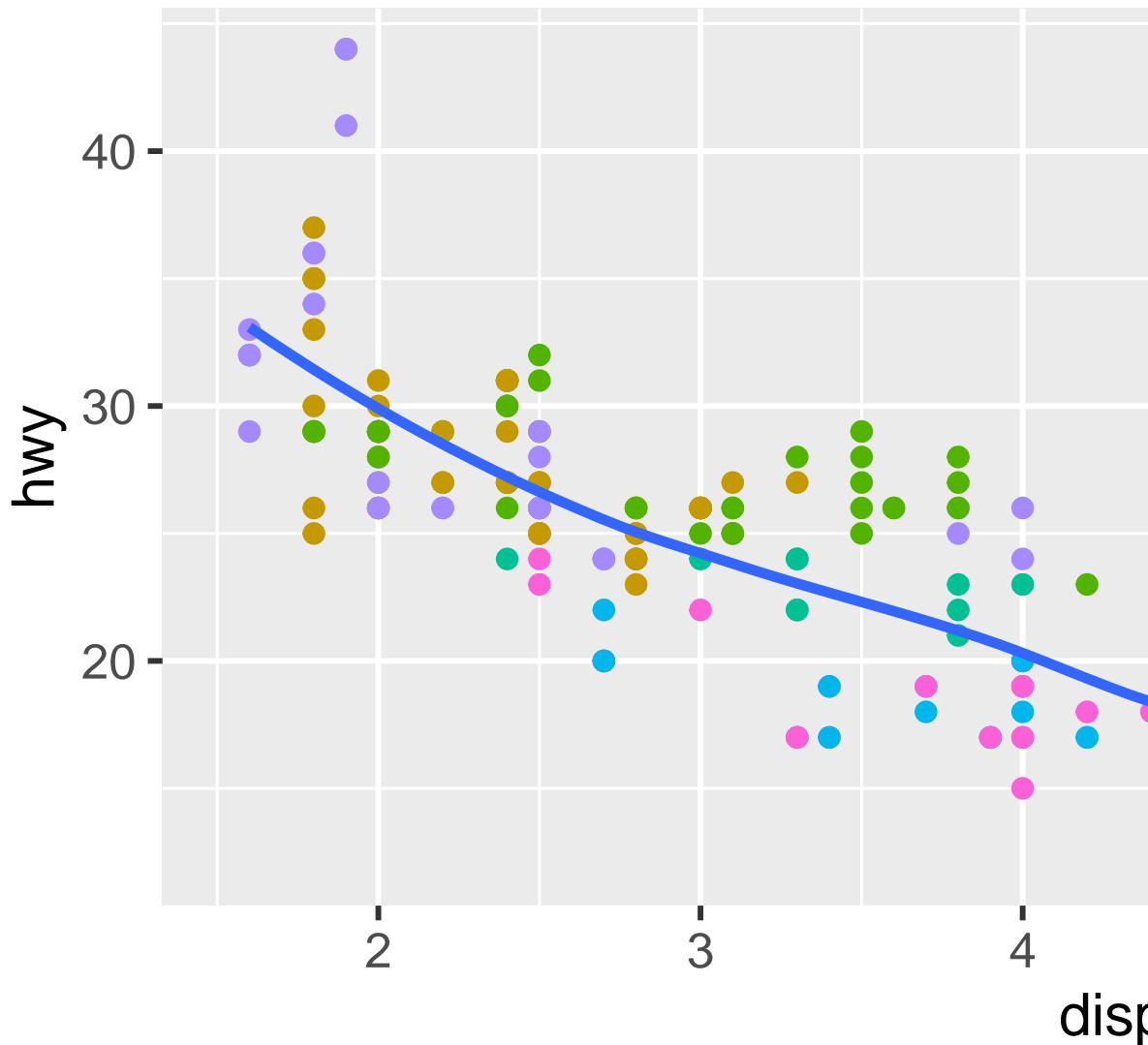
legend.position = "none"

guides()  guide_legend()  guide_colorbar()          nrow
alpha

ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  theme(legend.position = "bottom") +
  guides(color = guide_legend(nrow = 2, override.aes = list(size = 4)))
#> `geom_smooth()` using method = 'loess' and formula = 'y ~ x'

```





class



2seater



compact



midsized



minivan

```
guides()           labs()
```

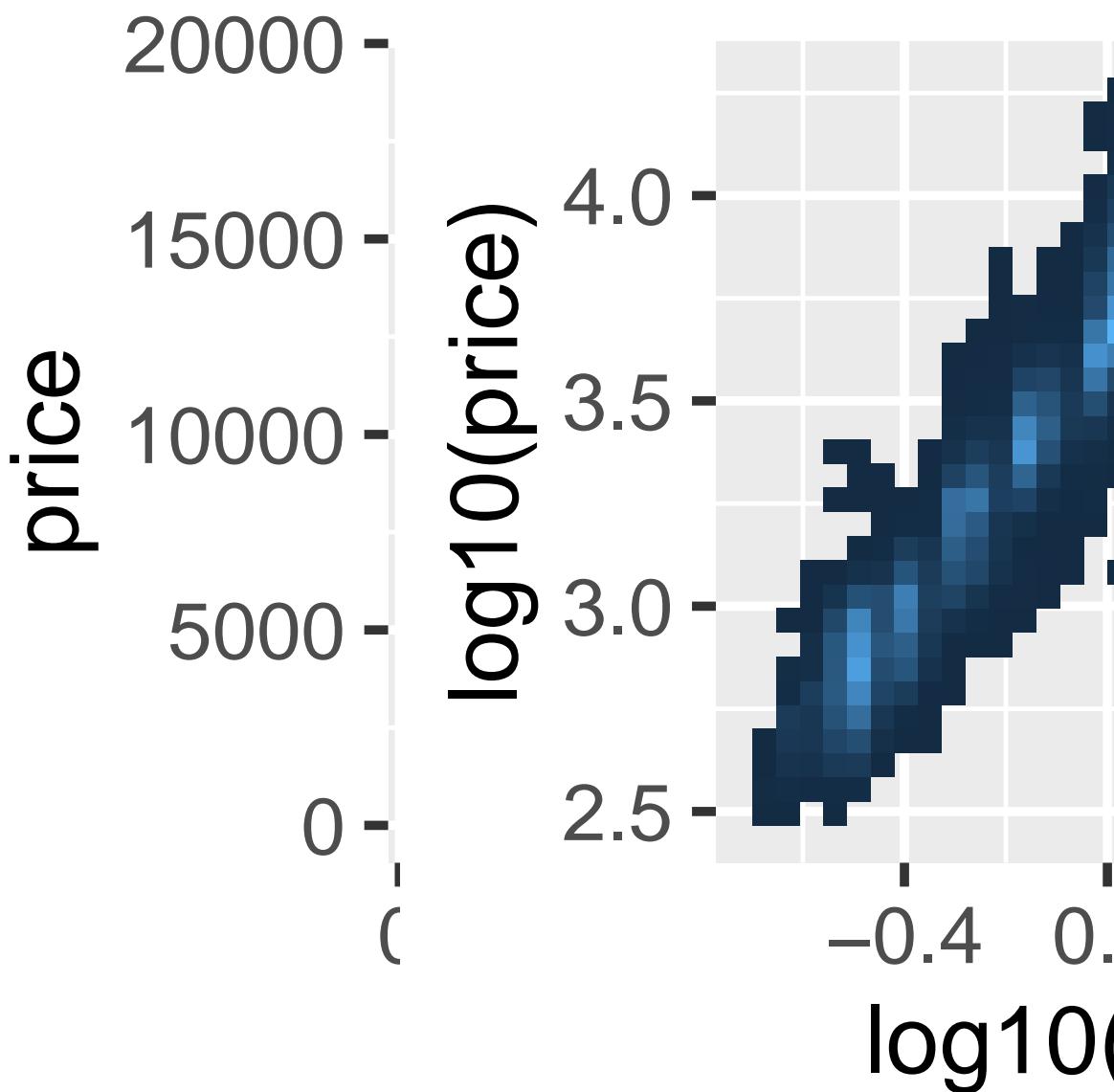
11.4.4

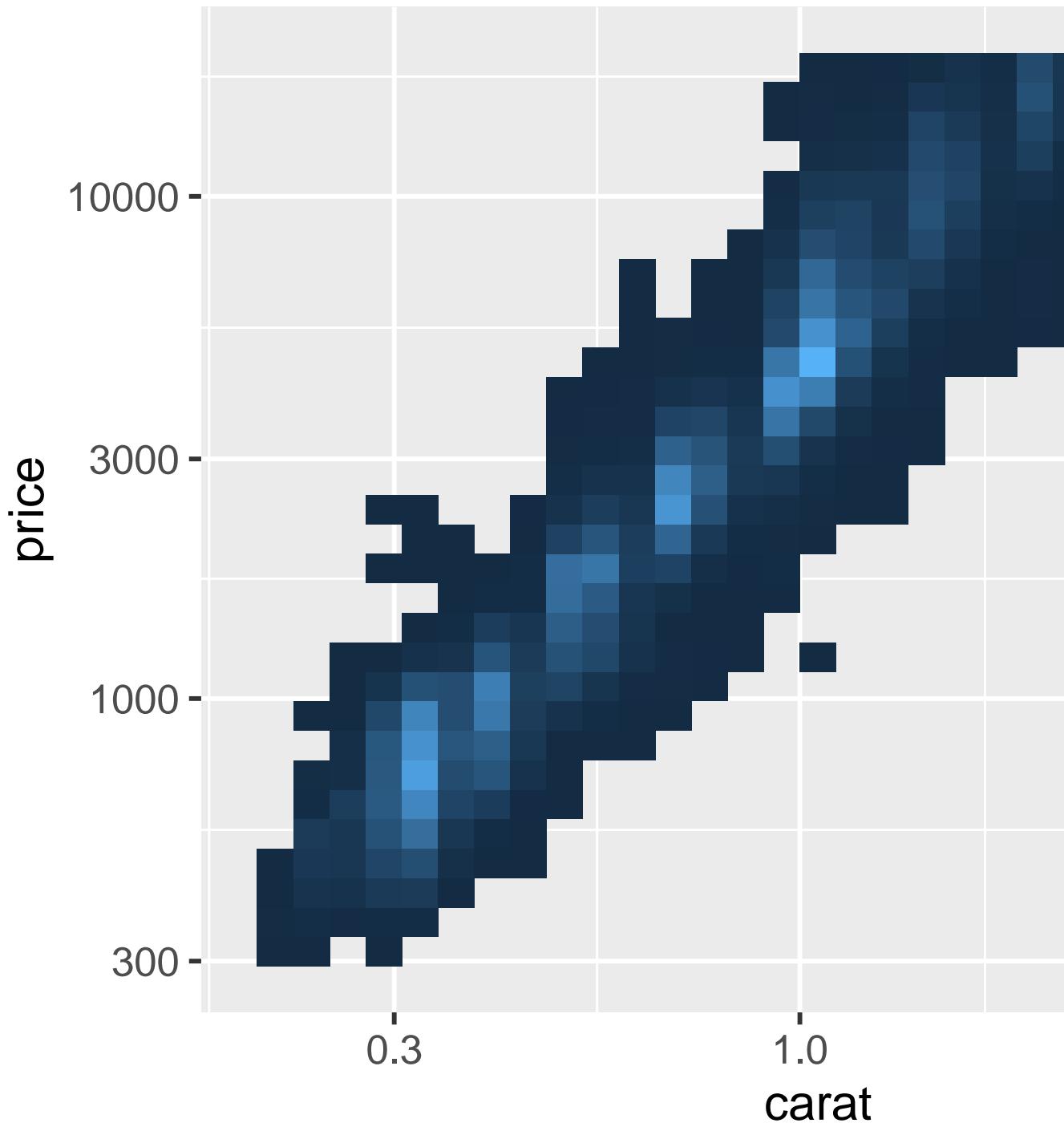
```
carat   price
```

```
# Left
ggplot(diamonds, aes(x = carat, y = price)) +
  geom_bin2d()

# Right
ggplot(diamonds, aes(x = log10(carat), y = log10(price))) +
  geom_bin2d()
```

```
ggplot(diamonds, aes(x = carat, y = price)) +
  geom_bin2d() +
  scale_x_log10() +
  scale_y_log10()
```





ColorBrewer

1

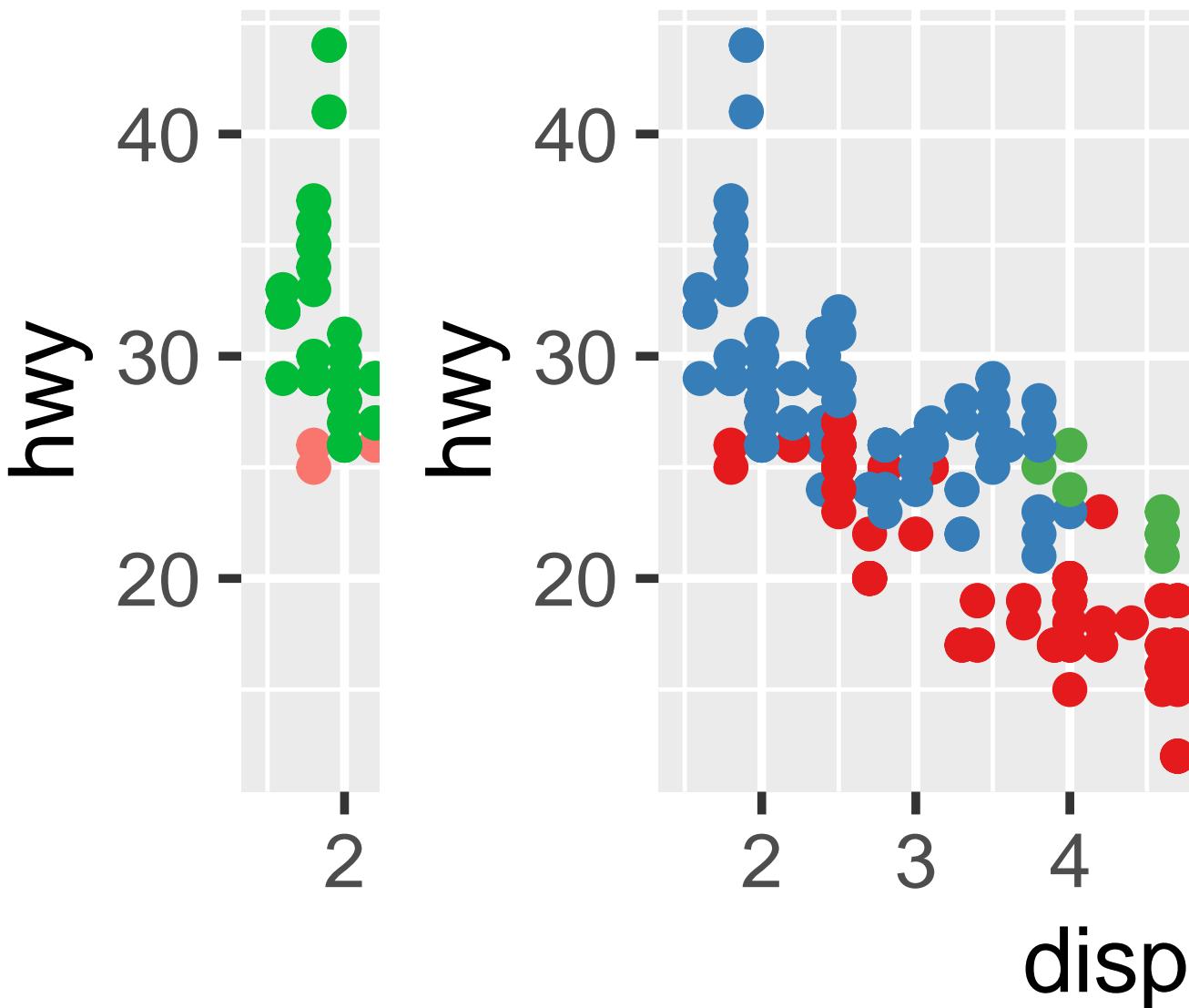
```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point(aes(color = drv))  
  
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point(aes(color = drv)) +  
  scale_color_brewer(palette = "Set1")
```

```
ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point(aes(color = drv, shape = drv)) +  
  scale_color_brewer(palette = "Set1")
```

¹ SimDaltonism

11.4.

279





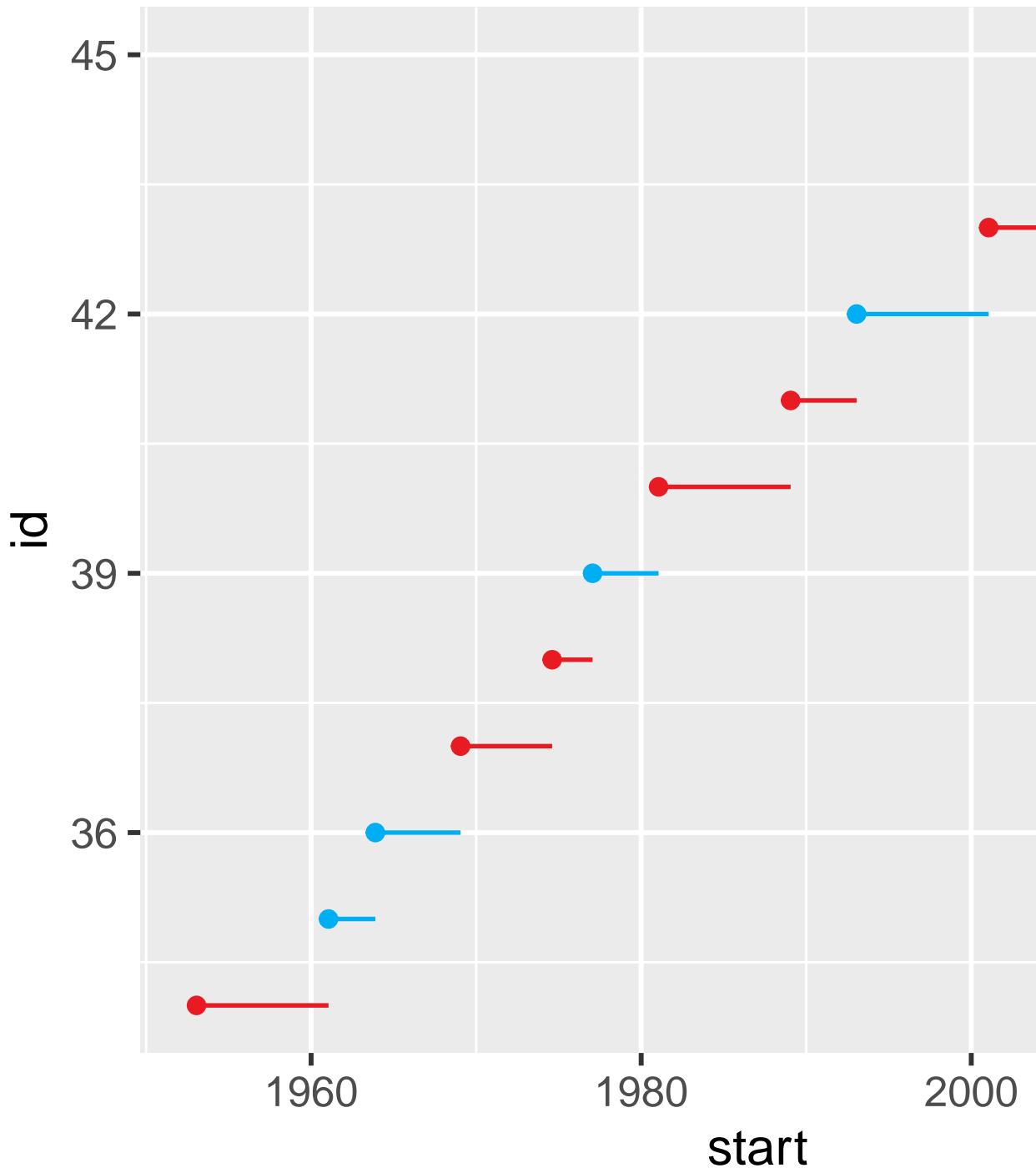
```
ColorBrewer           https://colorbrewer2.org/      Erich  Neuwirth
RColorBrewer    R      @ fig-brewer          “ ”
cut()
```

```
scale_color_manual()
```

```
presidential |>
  mutate(id = 33 + row_number()) |>
  ggplot(aes(x = start, y = id, color = party)) +
  geom_point() +
  geom_segment(aes(xend = end, yend = id)) +
  scale_color_manual(values = c(Republican = "#E81B23", Democratic = "#00AEF3"))
```



11.1: All colorBrewer scales.



```

scale_color_gradient() scale_fill_gradient()           scale_color_gradient2()

viridis      Nathaniel Smith  Stéfan van der Walt
ggplot2      c    d    b

df <- tibble(
  x = rnorm(10000),
  y = rnorm(10000)
)

ggplot(df, aes(x, y)) +
  geom_hex() +
  coord_fixed() +
  labs(title = "Default, continuous", x = NULL, y = NULL)

ggplot(df, aes(x, y)) +
  geom_hex() +
  coord_fixed() +
  scale_fill_viridis_c() +
  labs(title = "Viridis, continuous", x = NULL, y = NULL)

ggplot(df, aes(x, y)) +
  geom_hex() +
  coord_fixed() +
  scale_fill_viridis_b() +
  labs(title = "Viridis, binned", x = NULL, y = NULL)

scale_color_*(*) scale_fill_*(*)           color scales

```

11.4.5

- 1.
- 2.
3. coord_cartesian() xlim ylim

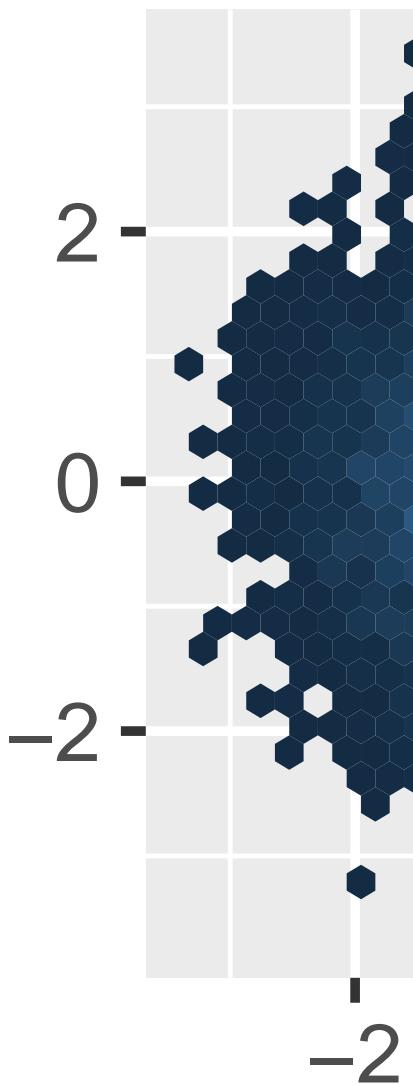
```
x  y
```

```

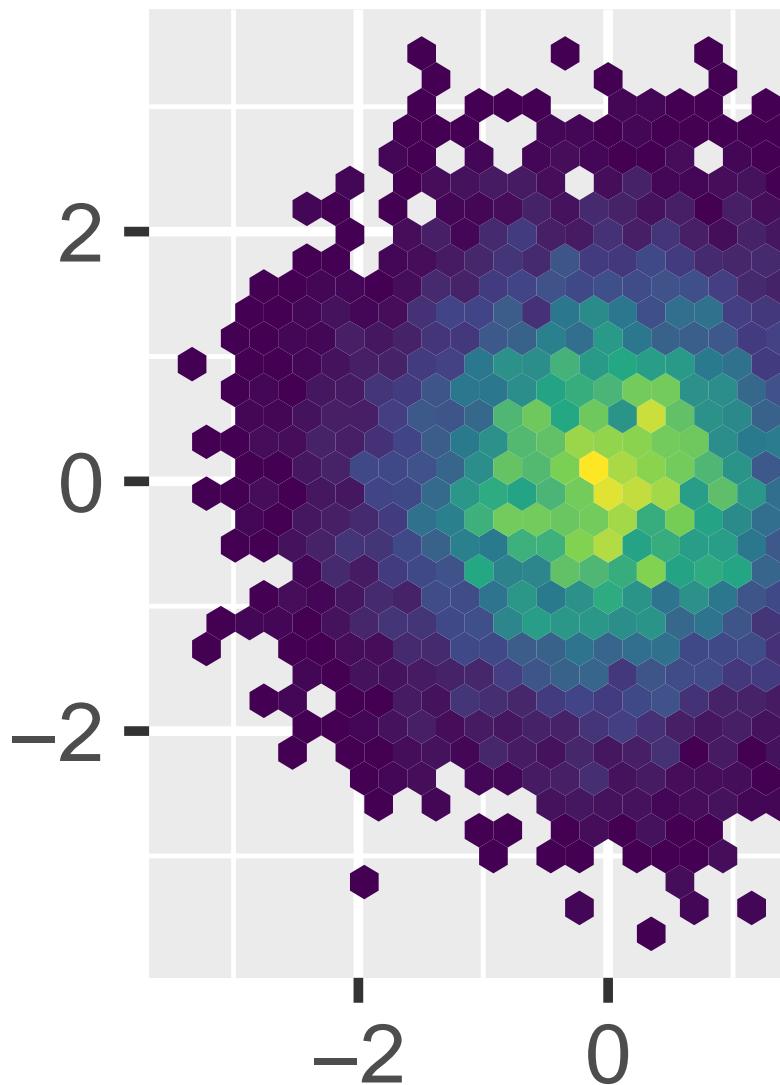
# Left
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) +
  geom_smooth()

```

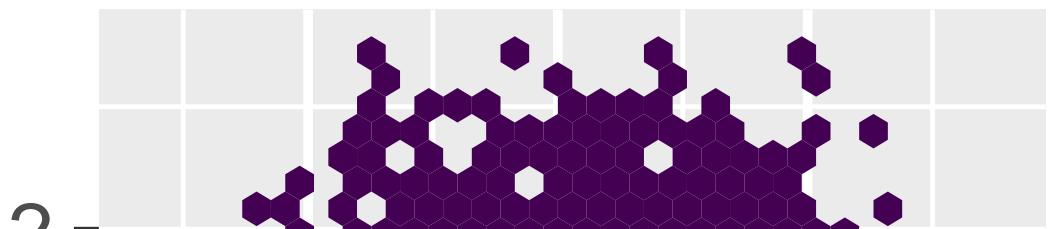
Default



Viridis, conti



Viridis, binned



```
# Right
mpg |>
  filter(displ >= 5 & displ <= 6 & hwy >= 10 & hwy <= 25) |>
  ggplot(aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) +
  geom_smooth()
```

coord_cartesian()

coord_cartesian()

```
# Left
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) +
  geom_smooth() +
  scale_x_continuous(limits = c(5, 6)) +
  scale_y_continuous(limits = c(10, 25))

# Right
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = drv)) +
  geom_smooth() +
  coord_cartesian(xlim = c(5, 6), ylim = c(10, 25))
```

| | | |
|--------|---|---|
| limits | x | y |
|--------|---|---|

```
suv <- mpg |> filter(class == "suv")
compact <- mpg |> filter(class == "compact")

# Left
ggplot(suv, aes(x = displ, y = hwy, color = drv)) +
  geom_point()

# Right
ggplot(compact, aes(x = displ, y = hwy, color = drv)) +
  geom_point()
```

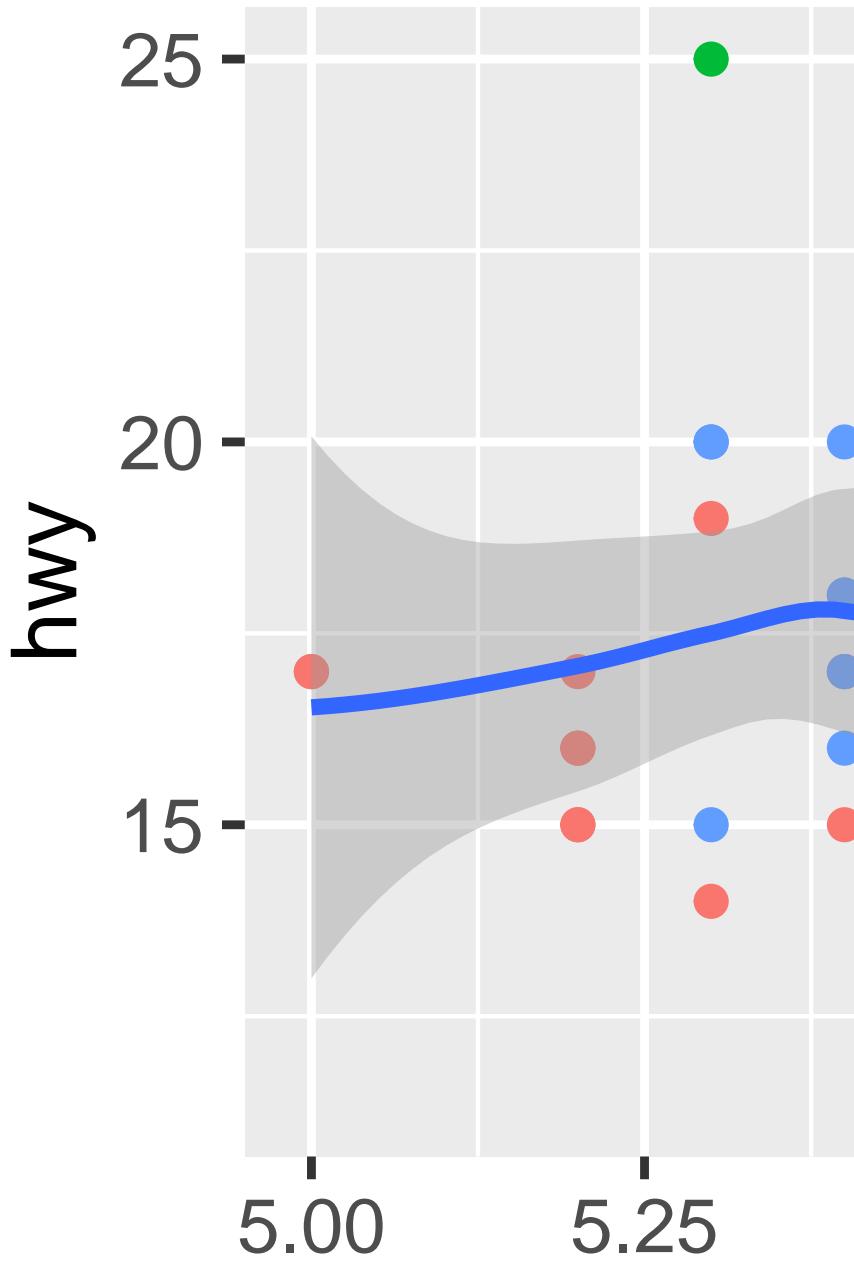
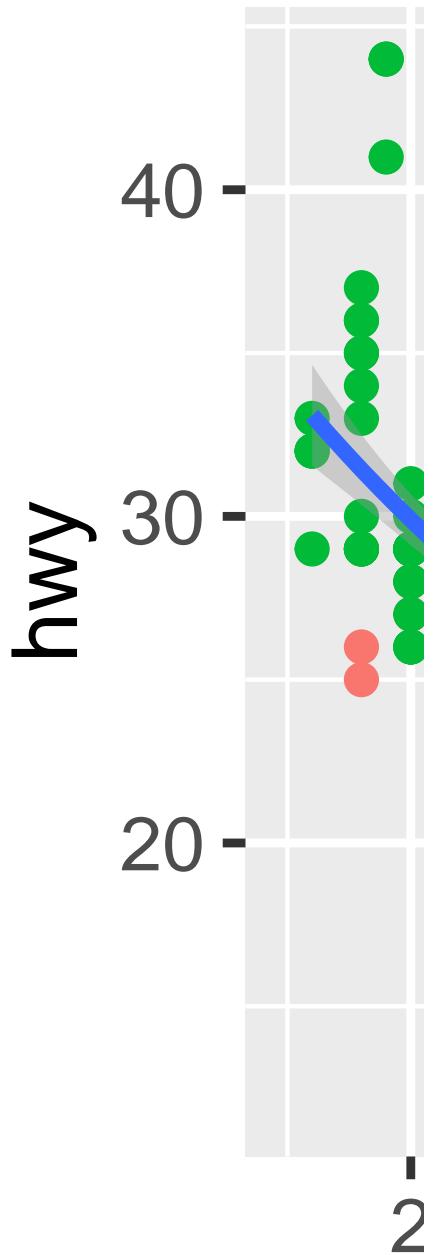
limits

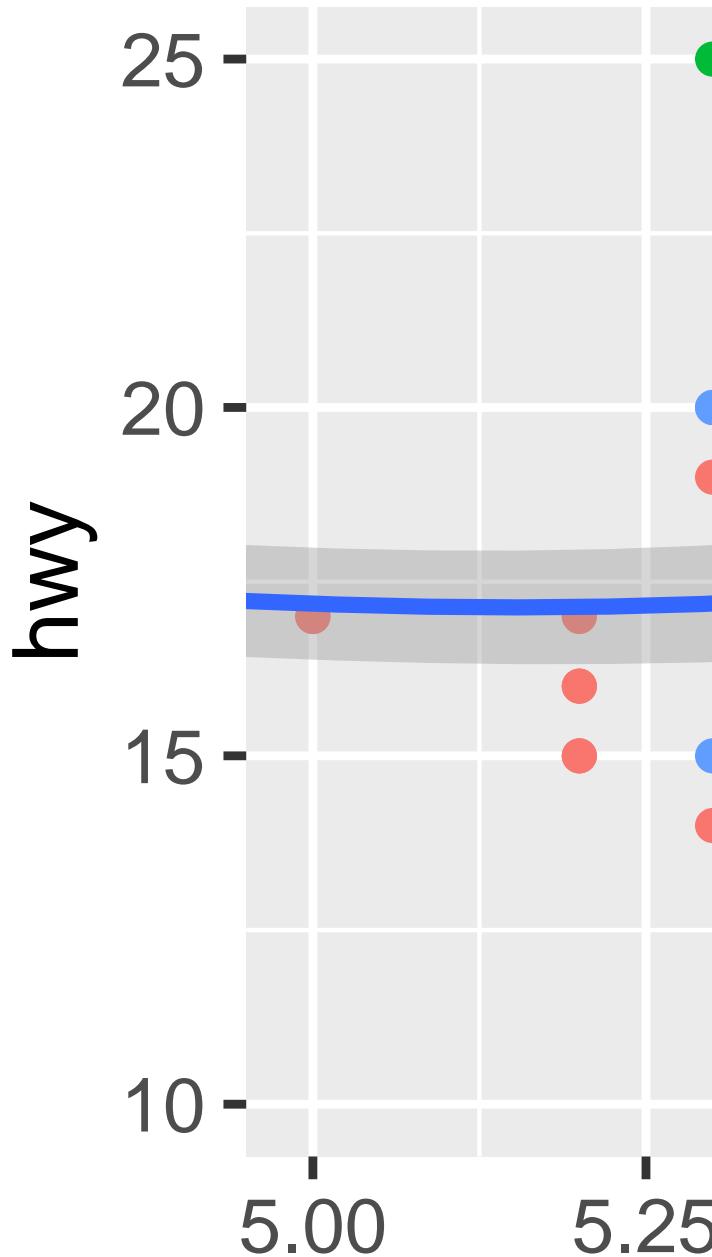
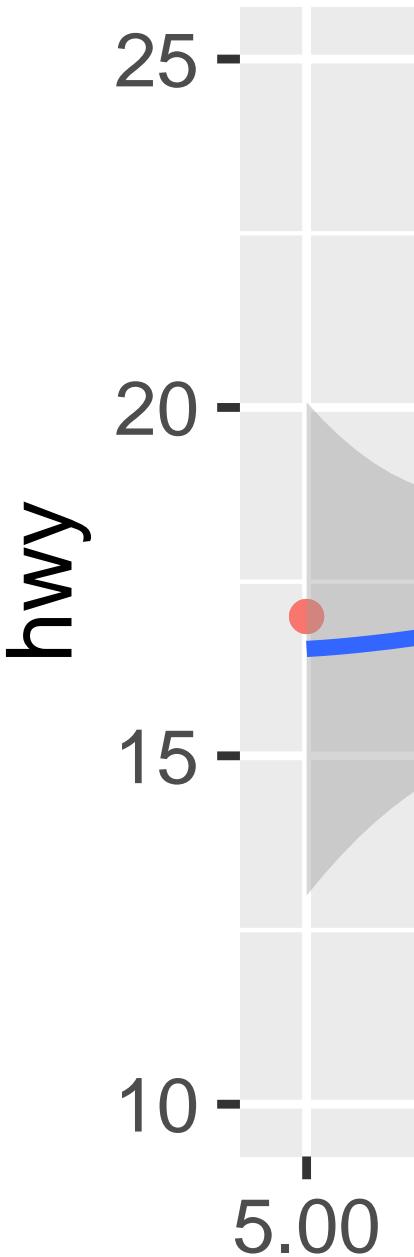
```
x_scale <- scale_x_continuous(limits = range(mpg$displ))
y_scale <- scale_y_continuous(limits = range(mpg$hwy))
col_scale <- scale_color_discrete(limits = unique(mpg$drv))

# Left
ggplot(suv, aes(x = displ, y = hwy, color = drv)) +
```

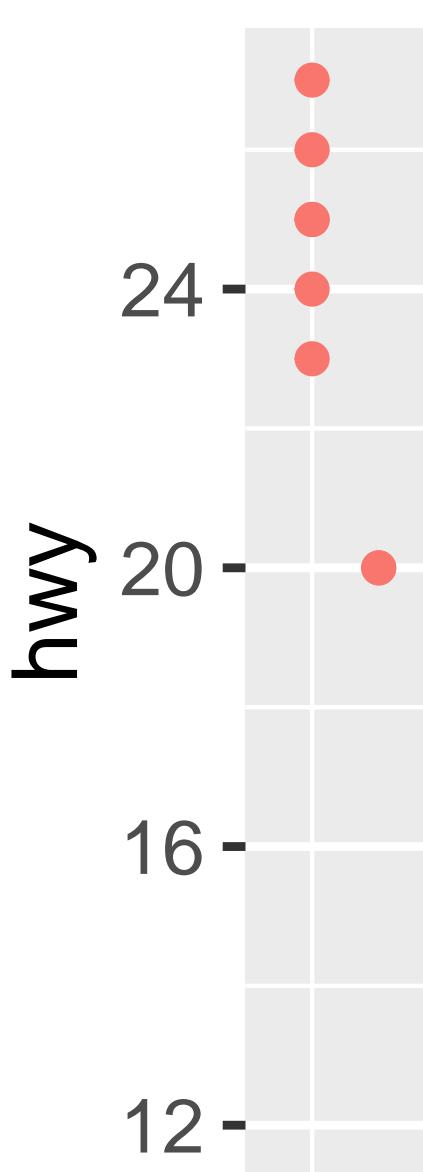
11.4.

287





11.4.



289



```

geom_point() +
x_scale +
y_scale +
col_scale

# Right
ggplot(compact, aes(x = displ, y = hwy, color = drv)) +
  geom_point() +
  x_scale +
  y_scale +
  col_scale

```

faceting

11.4.6

1. ?

```

df <- tibble(
  x = rnorm(10000),
  y = rnorm(10000)
)

ggplot(df, aes(x, y)) +
  geom_hex() +
  scale_color_gradient(low = "white", high = "red") +
  coord_fixed()

```

2. labs()

3.

- a. x
- b. y
- c.
- d.
- e. 4

4. override.aes

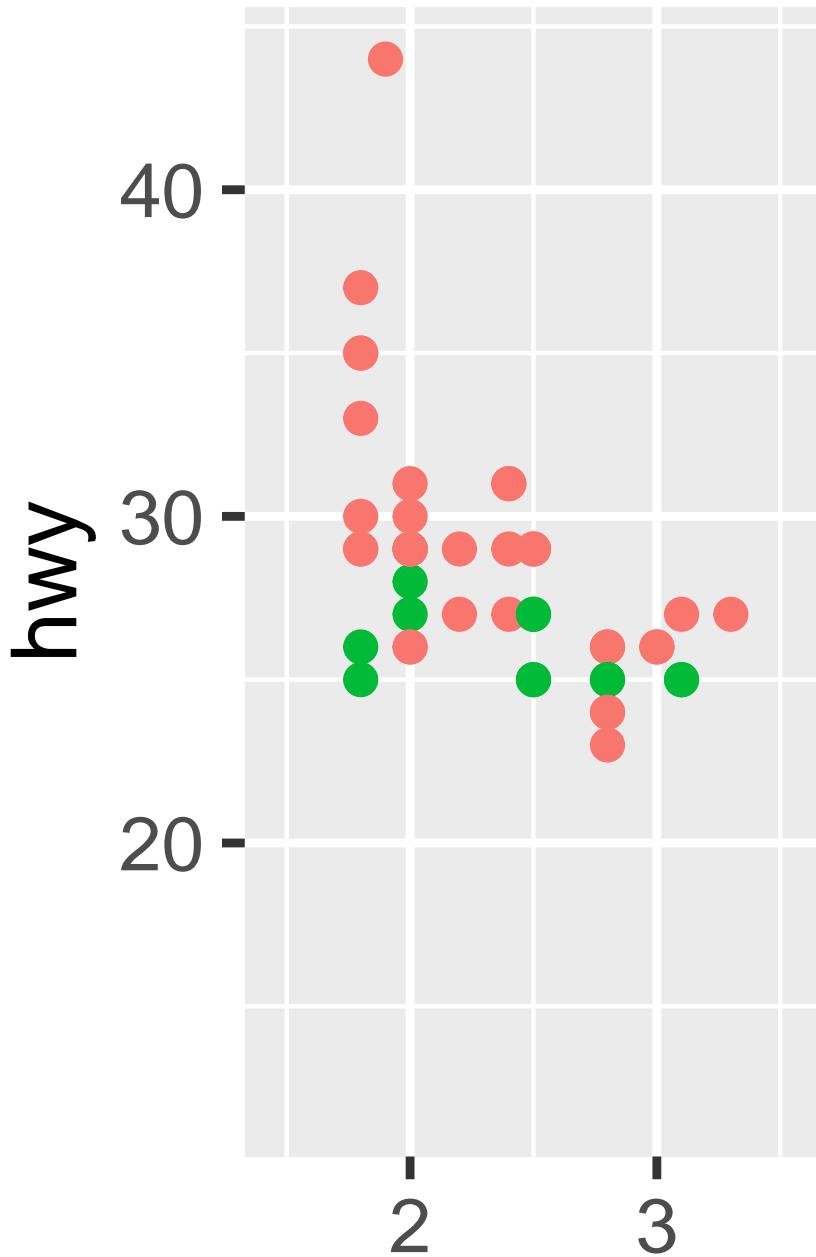
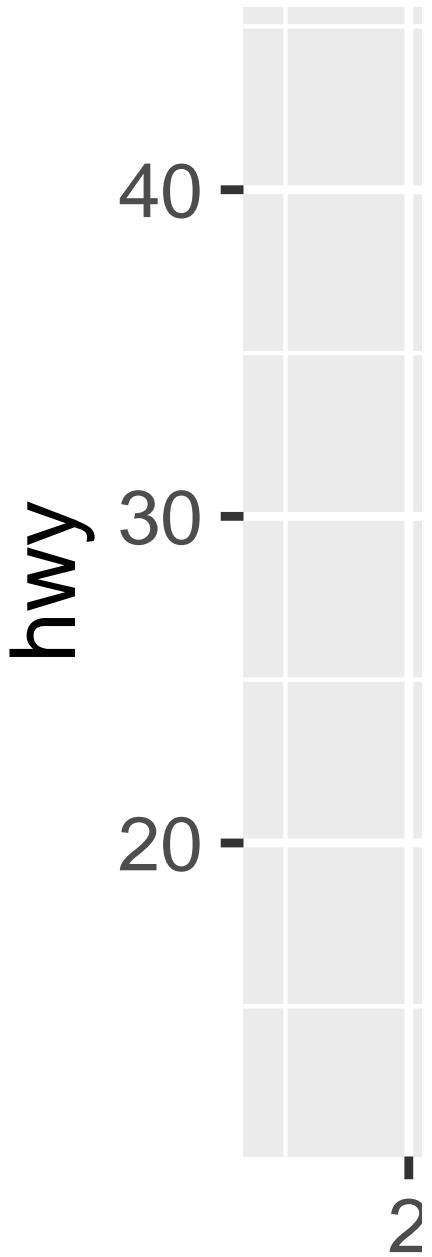
```

ggplot(diamonds, aes(x = carat, y = price)) +
  geom_point(aes(color = cut), alpha = 1/20)

```

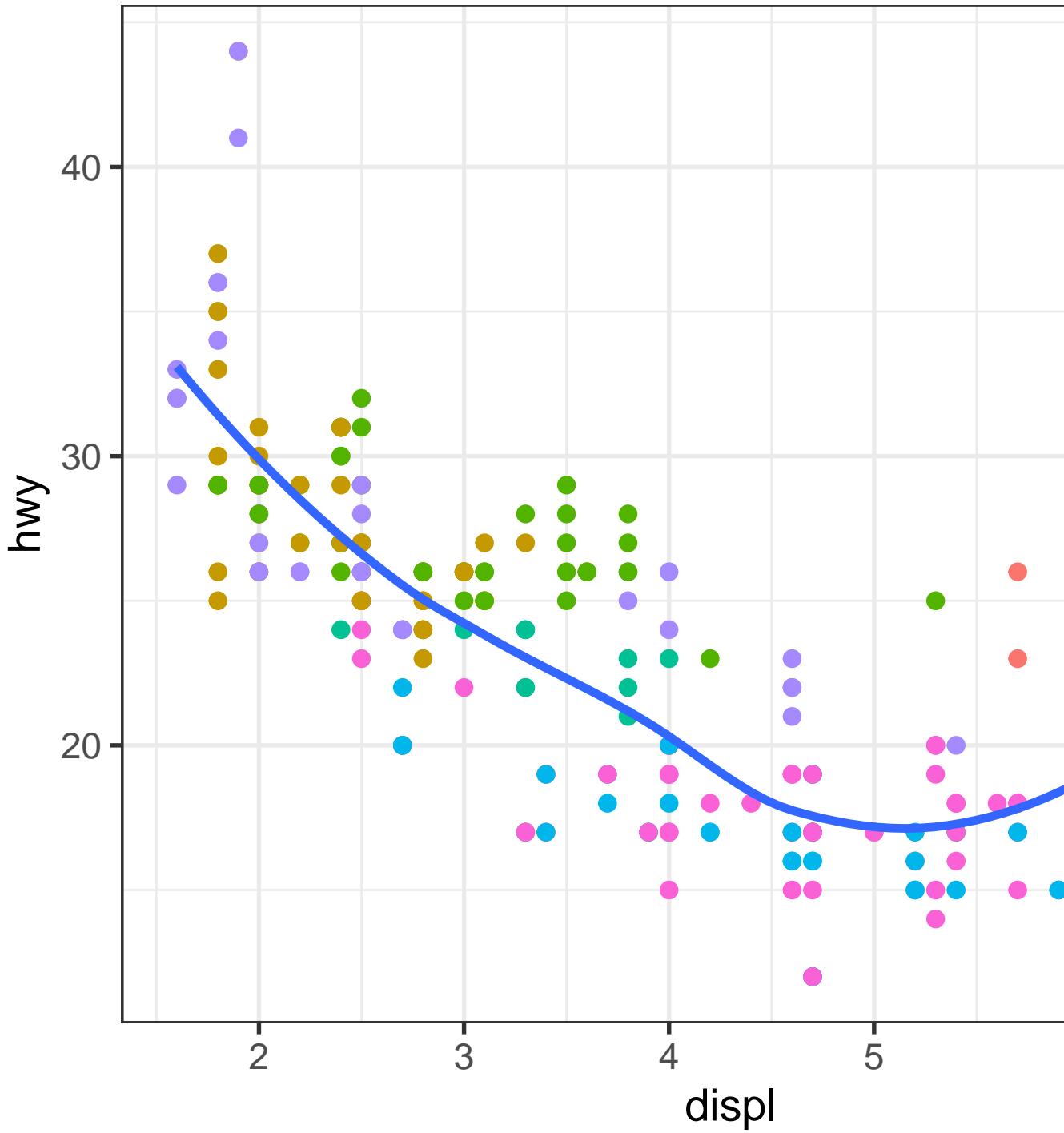
11.4.

291

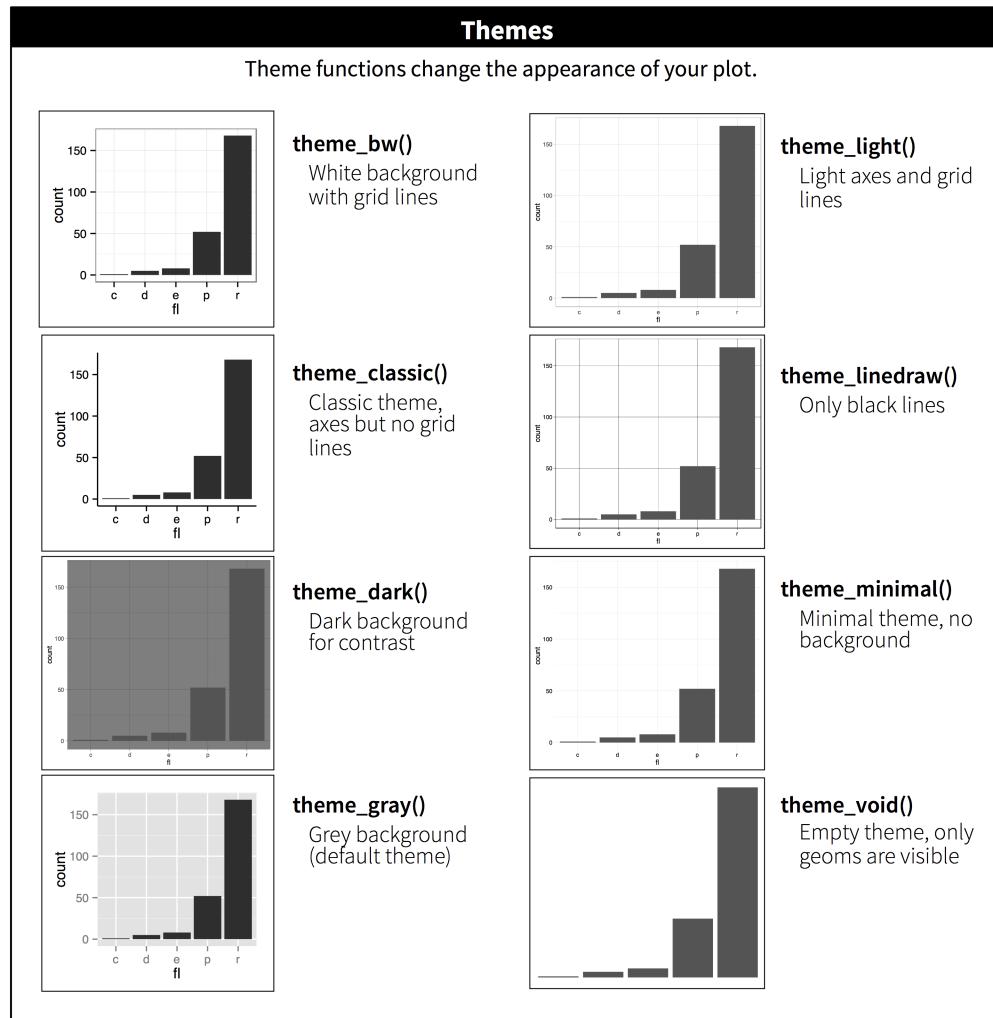


11.5

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth(se = FALSE) +
  theme_bw()
```



```
ggplot2      ??      theme_gray()      ggthemes https://jrnold.github.io/
ggthemes      Jeffrey Arnold
```

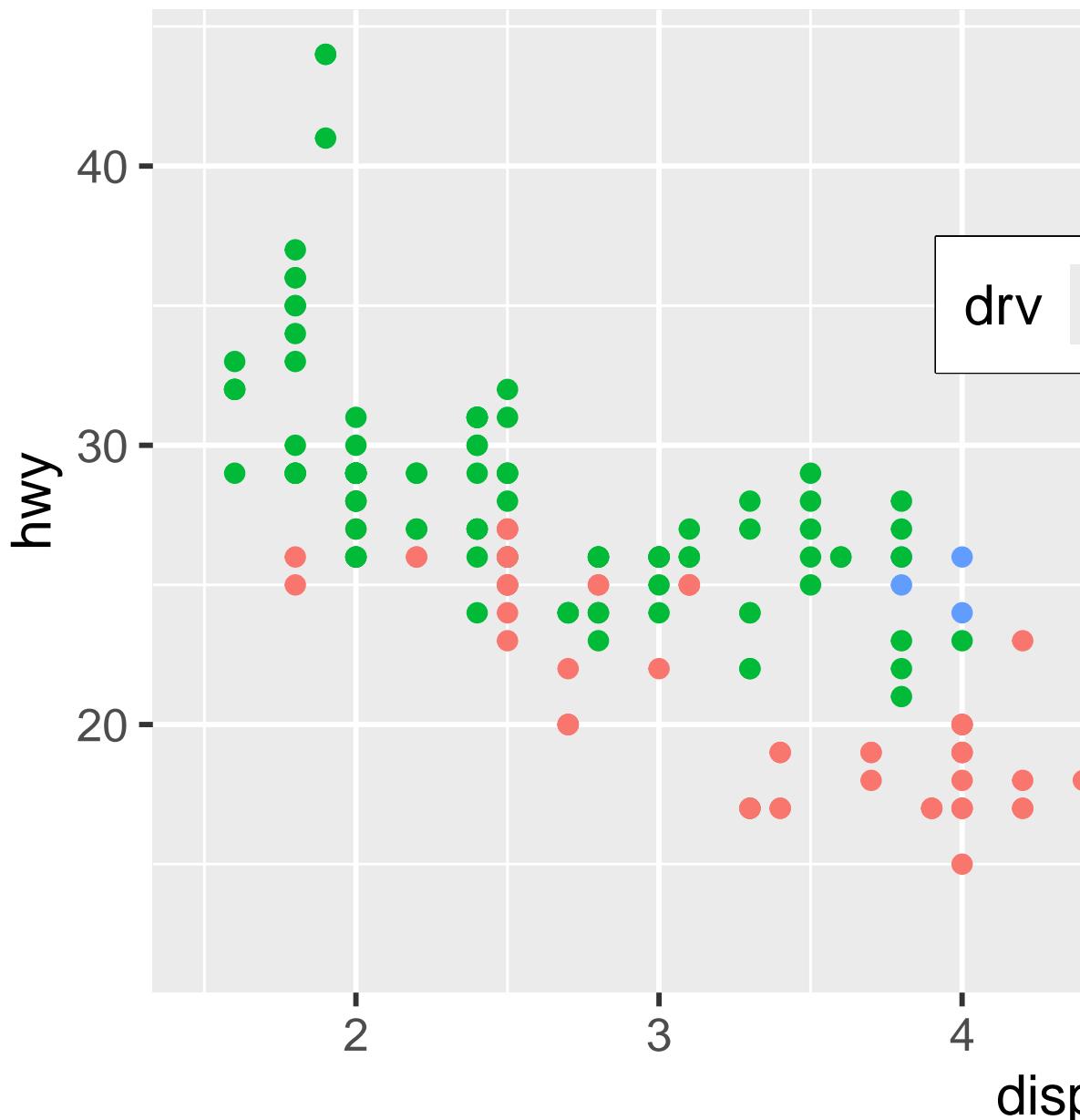


11.2: The eight themes built-in to ggplot2.

```
y          legend.position      theme()
           element_*( )
plot.title.position plot.caption.position   element_text() face
           theme()                           "plot"
```

```
ggplot(mpg, aes(x = displ, y = hwy, color = drv)) +
  geom_point() +
  labs(
    title = "Larger engine sizes tend to have lower fuel economy",
    caption = "Source: https://fueleconomy.gov."
  ) +
  theme(
    legend.position = c(0.6, 0.7),
    legend.direction = "horizontal",
    legend.box.background = element_rect(color = "black"),
    plot.title = element_text(face = "bold"),
    plot.title.position = "plot",
    plot.caption.position = "plot",
    plot.caption = element_text(hjust = 0)
  )
#> Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2
#> 3.5.0.
#> i Please use the `legend.position.inside` argument of `theme()` instead.
```

Larger engine sizes tend to have lower fuel economy



Source: <https://fueleconomy.gov>.

```
theme()      ?theme    g gplot2 book
```

For an overview of all `theme()` components, see help with `?theme`. The `gplot2` book is also a great place to go for the full details on theming.

11.5.1

1. `ggthemes`
- 2.

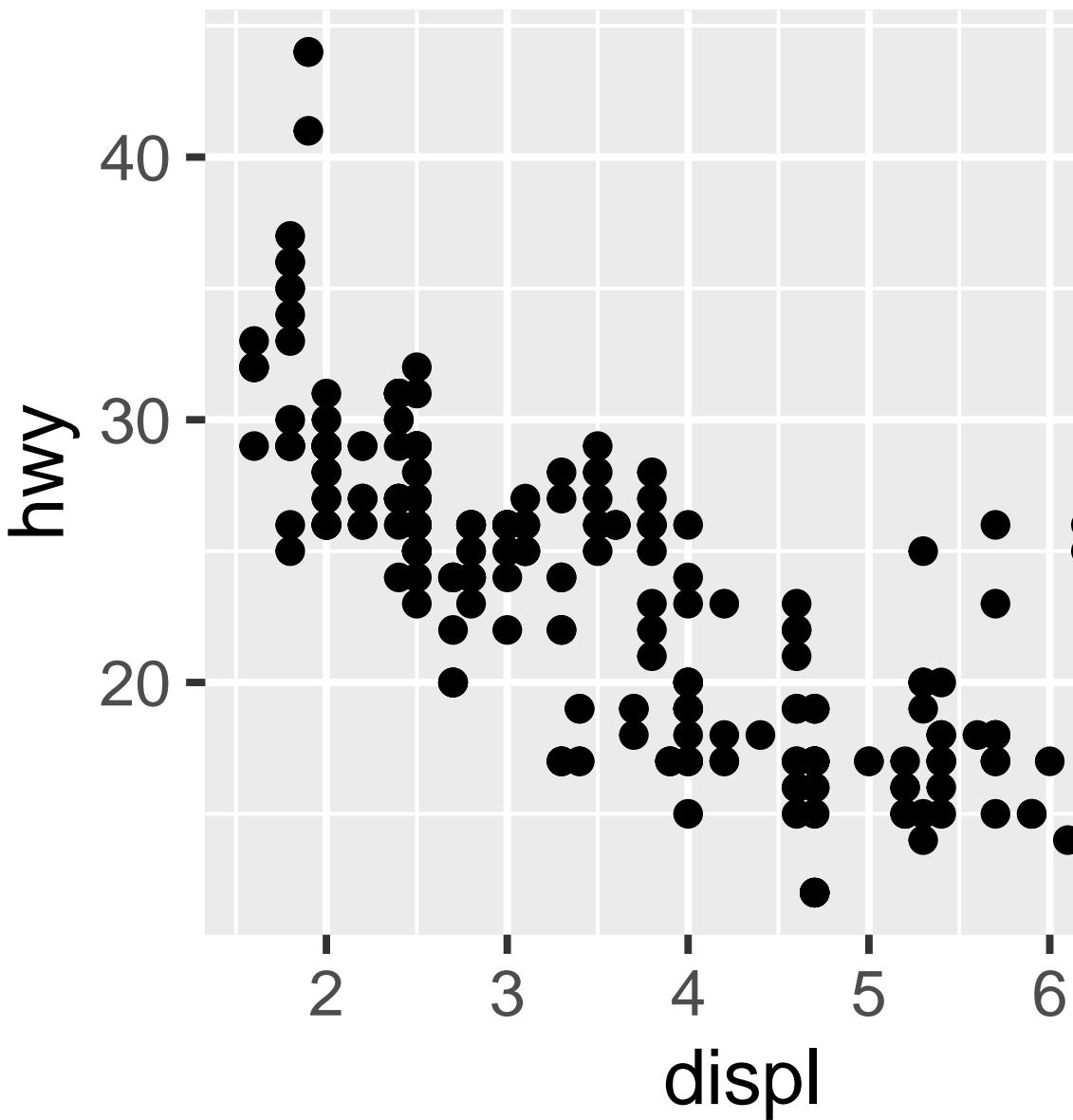
11.6

```
p patchwork
```

```
p1 p2      +
```

```
p1 <- ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  labs(title = "Plot 1")
p2 <- ggplot(mpg, aes(x = drv, y = hwy)) +
  geom_boxplot() +
  labs(title = "Plot 2")
p1 + p2
```

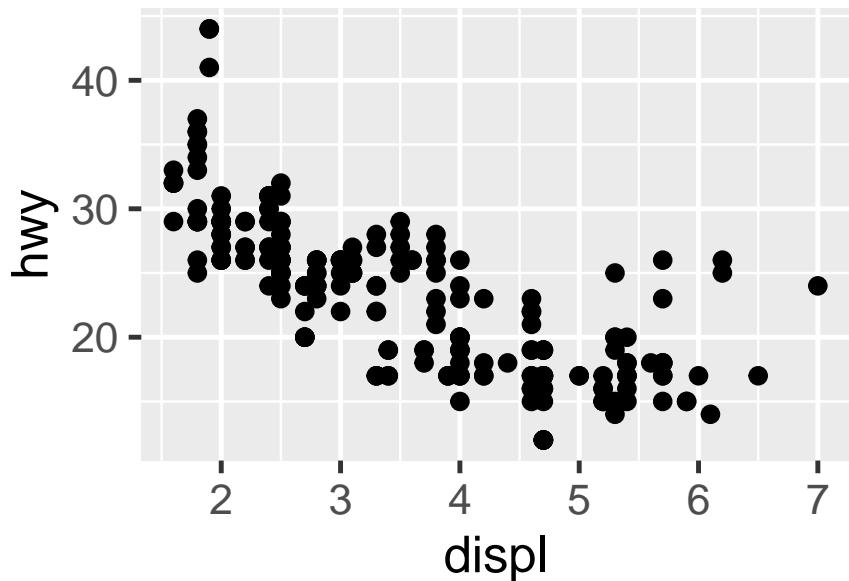
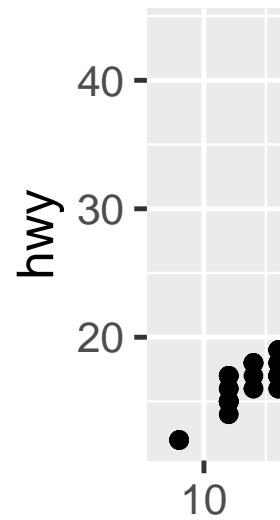
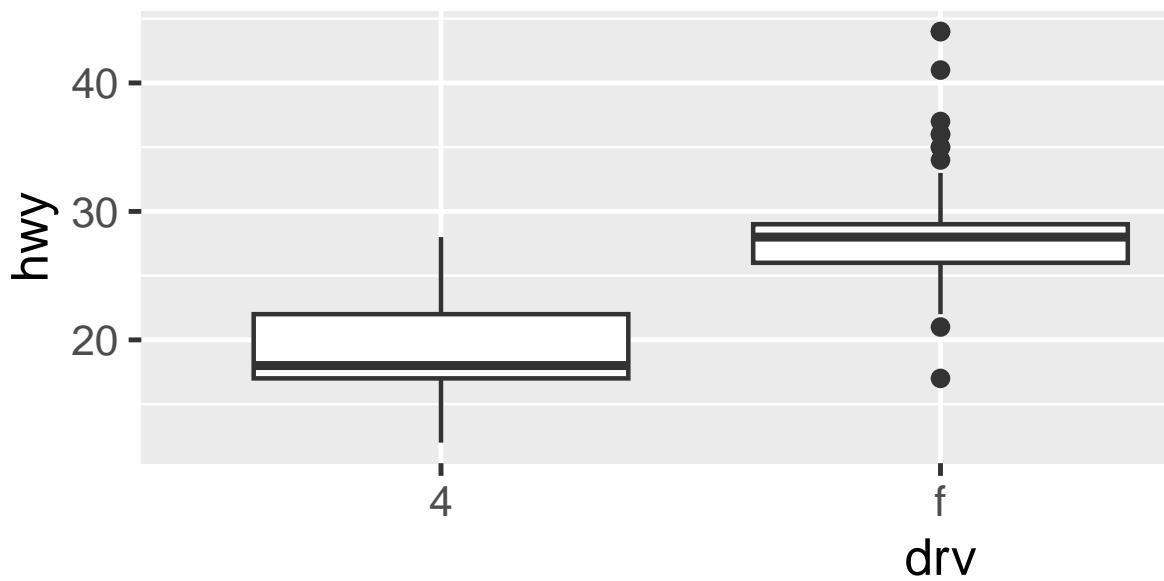
Plot 1



```
patchwork      +
```

```
patchwork      | p1 p3    / p2
```

```
p3 <- ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point() +
  labs(title = "Plot 3")
(p1 | p3) / p2
```

Plot 1**Plot 3****Plot 2**

```
patchwork
theme(legend.position = "top")      &      +      5      &
guide_area()           patchwork      1     3     2      patchwork      ggplot
                                         4 p
patchwork
```



```
p1 <- ggplot(mpg, aes(x = drv, y = cty, color = drv)) +
  geom_boxplot(show.legend = FALSE) +
  labs(title = "Plot 1")

p2 <- ggplot(mpg, aes(x = drv, y = hwy, color = drv)) +
  geom_boxplot(show.legend = FALSE) +
  labs(title = "Plot 2")

p3 <- ggplot(mpg, aes(x = cty, color = drv, fill = drv)) +
  geom_density(alpha = 0.5) +
  labs(title = "Plot 3")

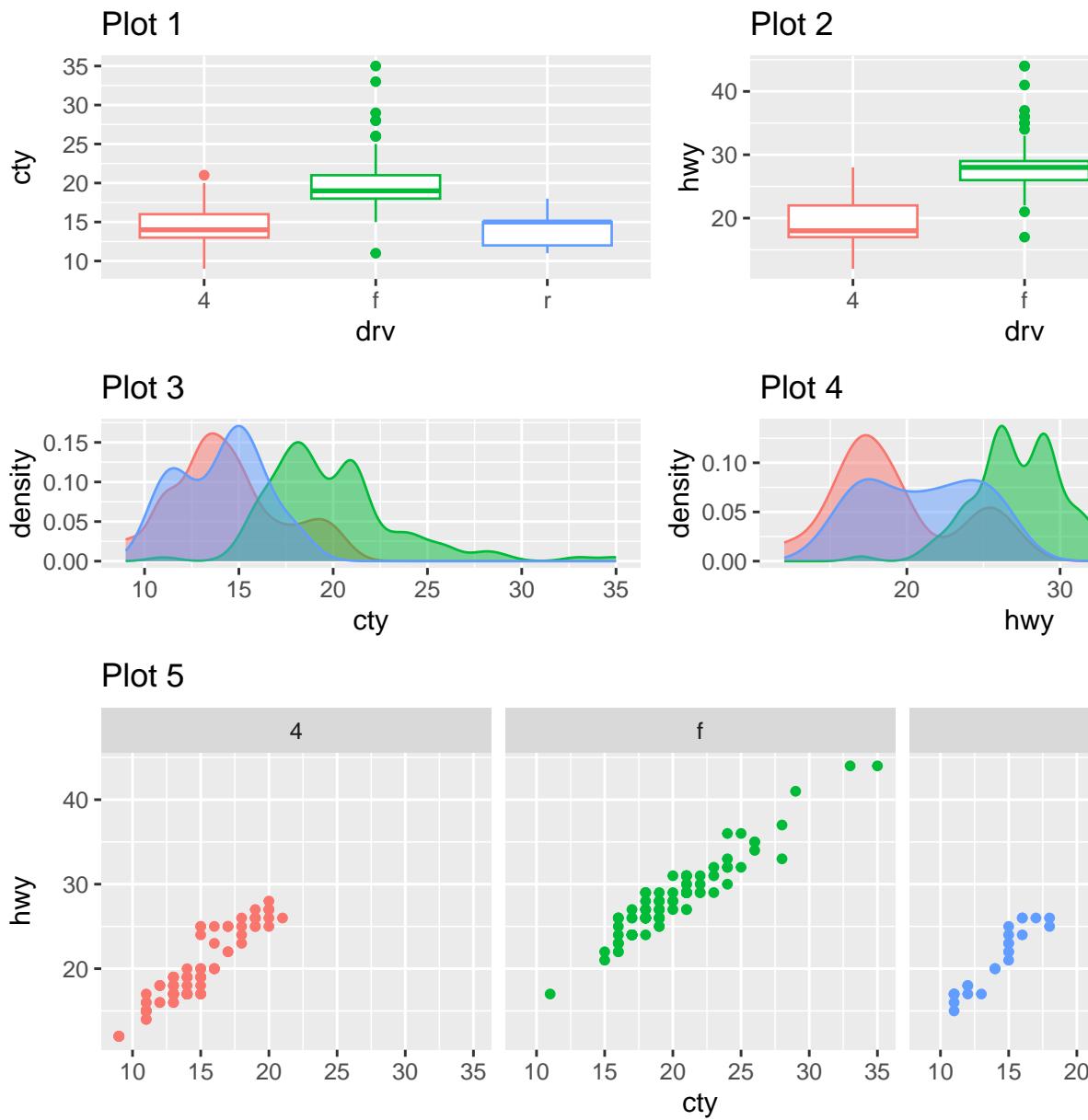
p4 <- ggplot(mpg, aes(x = hwy, color = drv, fill = drv)) +
  geom_density(alpha = 0.5) +
  labs(title = "Plot 4")

p5 <- ggplot(mpg, aes(x = cty, y = hwy, color = drv)) +
  geom_point(show.legend = FALSE) +
  facet_wrap(~drv) +
  labs(title = "Plot 5")

(guide_area() / (p1 + p2) / (p3 + p4) / p5) +
  plot_annotation(
    title = "City and highway mileage for cars with different drive trains",
    caption = "Source: https://fueleconomy.gov.'
  ) +
  plot_layout(
    guides = "collect",
    heights = c(1, 3, 2, 4)
  ) &
  theme(legend.position = "top")
```

City and highway mileage for cars with different drive trains

drv 4 f r



Source: <https://www.r-project.org>

patchwork

<https://patchwork.data-imaginist.com>

11.6.1

1.

```
p1 <- ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  labs(title = "Plot 1")
p2 <- ggplot(mpg, aes(x = drv, y = hwy)) +
  geom_boxplot() +
  labs(title = "Plot 2")
p3 <- ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point() +
  labs(title = "Plot 3")

(p1 | p2) / p3
```

2.

patchwork

Fig. A:
Plot 1

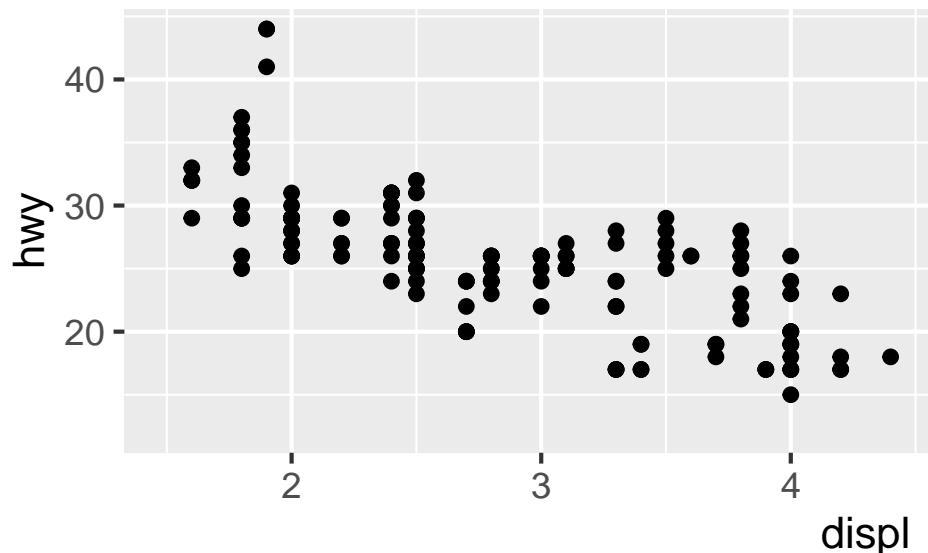


Fig. B:
Plot 2

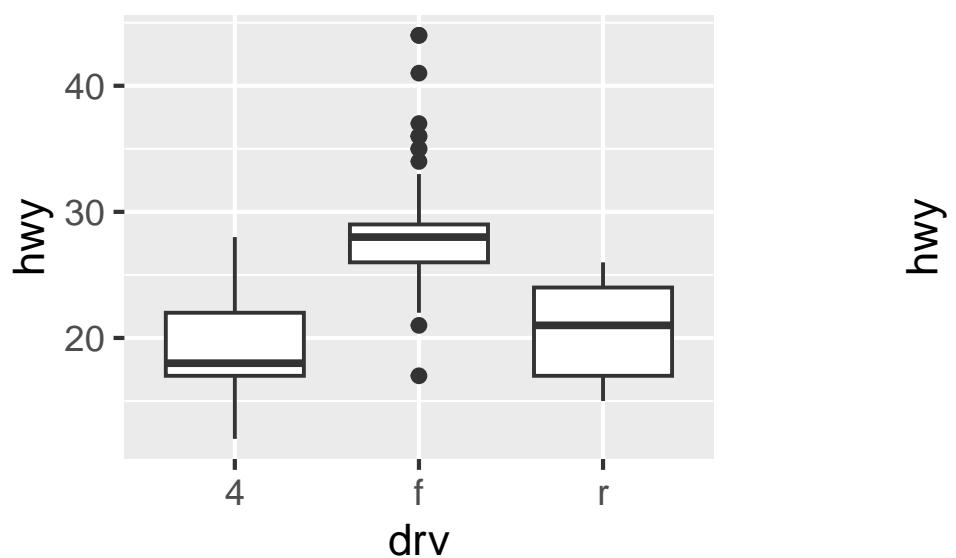
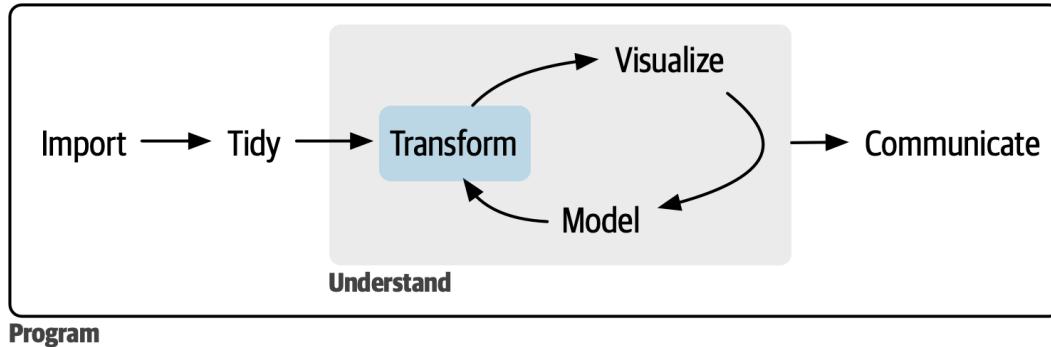


Fig. C:

11.7

ggplot2
Elegant Graphics for Data Analysis Winston Chang *R Graphics Cookbook* Claus Wilke *Fundamentals of Data Visualization*

Part III



Program

11.3: The options for data transformation depends heavily on the type of data involved, the subject of this part of the book.

- ??
- ??
- ?? stringr tidyr
- ?? R
- ?? lubridate
- ?? join keys
- ??

Chapter 12

12.1

```
TRUE  FALSE  NA  
if_else() case_when()
```

12.1.1

```
R      tidyverse      mutate() filter()      nycflights13::flights
```

```
library(tidyverse)  
library(nycflights13)
```

```
c()      :
```

```
x <- c(1, 2, 3, 5, 7, 11, 13)  
x * 2  
#> [1] 2 4 6 10 14 22 26
```

```
mutate()
```

```
df <- tibble(x)  
df |>  
  mutate(y = x * 2)  
#> # A tibble: 7 x 2  
#>       x     y  
#>   <dbl> <dbl>
```

```
#> 1     1     2
#> 2     2     4
#> 3     3     6
#> 4     5    10
#> 5     7    14
#> 6    11    22
#> # i 1 more row
```

12.2

```
< <= > >= != ==           filter()           filter()

flights |>
  filter(dep_time > 600 & dep_time < 2000 & abs(arr_delay) < 20)
#> # A tibble: 172,286 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>     <int>          <int>     <dbl>     <int>      <int>
#> 1  2013     1     1       601            600        1     844        850
#> 2  2013     1     1       602            610       -8     812        820
#> 3  2013     1     1       602            605       -3     821        805
#> 4  2013     1     1       606            610       -4     858        910
#> 5  2013     1     1       606            610       -4     837        845
#> 6  2013     1     1       607            607        0     858        915
#> # i 172,280 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...

mutate()

flights |>
  mutate(
    daytime = dep_time > 600 & dep_time < 2000,
    approx_ontime = abs(arr_delay) < 20,
    .keep = "used"
  )
#> # A tibble: 336,776 x 4
#>   dep_time arr_delay daytime approx_ontime
#>   <int>     <dbl> <lgl>    <lgl>
#> 1     517      11 FALSE    TRUE
#> 2     533      20 FALSE   FALSE
#> 3     542      33 FALSE   FALSE
#> 4     544     -18 FALSE    TRUE
#> 5     554     -25 FALSE   FALSE
#> 6     554      12 FALSE    TRUE
#> # i 336,770 more rows
```

```
filter

flights |>
  mutate(
    daytime = dep_time > 600 & dep_time < 2000,
    approx_ontime = abs(arr_delay) < 20,
  ) |>
  filter(daytime & approx_ontime)
```

12.2.1

`==` 1 2

```
x <- c(1 / 49 * 49, sqrt(2) ^ 2)
x
#> [1] 1 2
```

FALSE:

```
x == c(1, 2)
#> [1] FALSE FALSE
```

```
1/49 sqrt(2)           print()   digits1
print(x, digits = 16)
#> [1] 0.9999999999999999 2.0000000000000004
```

R

`==` `dplyr::near()`

```
near(x, c(1, 2))
#> [1] TRUE TRUE
```

12.2.2

“ ”

¹R `print x print(x)` `print`

```
NA > 5
#> [1] NA
10 == NA
#> [1] NA
```

```
NA == NA
#> [1] NA
```

```
# We don't know how old Mary is
age_mary <- NA

# We don't know how old John is
age_john <- NA

# Are Mary and John the same age?
age_mary == age_john
#> [1] NA
# We don't know!
```

```
dep_time           dep_time == NA      NA filter()

flights |>
  filter(dep_time == NA)
#> # A tibble: 0 x 19
#> #   i 19 variables: year <int>, month <int>, day <int>, dep_time <int>,
#> #     sched_dep_time <int>, dep_delay <dbl>, arr_time <int>, ...
#> #   ...
```

: is.na().

12.2.3 is.na()

```
is.na(x)          TRUE      FALSE

is.na(c(TRUE, NA, FALSE))
#> [1] FALSE  TRUE FALSE
is.na(c(1, NA, 3))
#> [1] FALSE  TRUE FALSE
is.na(c("a", NA, "b"))
#> [1] FALSE  TRUE FALSE
```

```

is.na()      dep_time

flights |>
  filter(is.na(dep_time))
#> # A tibble: 8,255 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>    <int>        <int>     <dbl>    <int>        <int>
#> 1 2013     1     1       NA        1630      NA      NA        1815
#> 2 2013     1     1       NA        1935      NA      NA        2240
#> 3 2013     1     1       NA        1500      NA      NA        1825
#> 4 2013     1     1       NA        600       NA      NA        901
#> 5 2013     1     2       NA        1540      NA      NA        1747
#> 6 2013     1     2       NA        1620      NA      NA        1746
#> # i 8,249 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...

```



```

is.na() arrange()      arrange()          is.na()

flights |>
  filter(month == 1, day == 1) |>
  arrange(dep_time)
#> # A tibble: 842 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>    <int>        <int>     <dbl>    <int>        <int>
#> 1 2013     1     1       517        515       2       830        819
#> 2 2013     1     1       533        529       4       850        830
#> 3 2013     1     1       542        540       2       923        850
#> 4 2013     1     1       544        545      -1      1004       1022
#> 5 2013     1     1       554        600      -6       812        837
#> 6 2013     1     1       554        558      -4       740        728
#> # i 836 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...

flights |>
  filter(month == 1, day == 1) |>
  arrange(desc(is.na(dep_time)), dep_time)
#> # A tibble: 842 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>    <int>        <int>     <dbl>    <int>        <int>
#> 1 2013     1     1       NA        1630      NA      NA        1815
#> 2 2013     1     1       NA        1935      NA      NA        2240
#> 3 2013     1     1       NA        1500      NA      NA        1825
#> 4 2013     1     1       NA        600       NA      NA        901
#> 5 2013     1     1       517        515       2       830        819
#> 6 2013     1     1       533        529       4       850        830

```

```
#> # i 836 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

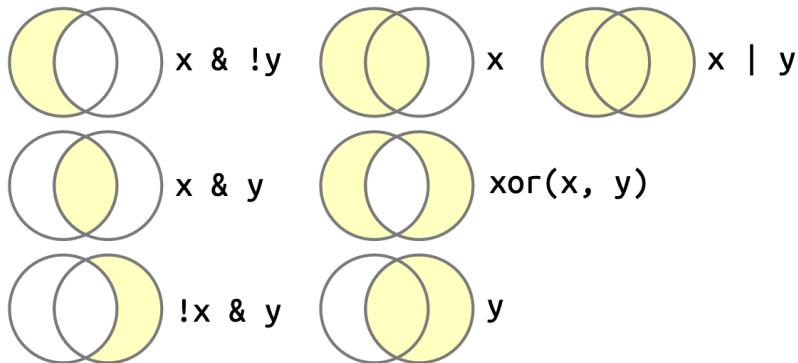
??

12.2.4

```
1. dplyr::near()      near      sqrt(2)^2  2
2.   mutate(),is.na(), count()  dep_time sched_dep_time dep_delay
```

12.3

```
R & " " | " " ! " " xor() ^ 2  df %>% filter(!is.na(x))  x  df
%>% filter(x < -10 | x > 0)  x -10  0  ??
```



12.1: The complete set of Boolean operations. x is the left-hand circle, y is the right-hand circle, and the shaded region show which parts each operator selects.

| | |
|--|-------------------------|
| <code>& R && dplyr</code> | <code>TRUE FALSE</code> |
|--|-------------------------|

As well as `&` and `|`, R also has `&&` and `||`. Don't use them in dplyr functions! These are called short-circuiting operators and only ever return a single `TRUE` or `FALSE`. They're important for programming, not data science.

12.3.1

| | | |
|----------------------|------------------------|----------------------|
| <code>^ 2 x y</code> | <code>xor(x, y)</code> | <code>" " " "</code> |
|----------------------|------------------------|----------------------|

```
df <- tibble(x = c(TRUE, FALSE, NA))

df |>
  mutate(
    and = x & NA,
    or = x | NA
  )
#> # A tibble: 3 x 3
#>   x     and    or
#>   <lgl> <lgl> <lgl>
#> 1 TRUE  NA    TRUE
#> 2 FALSE FALSE NA
#> 3 NA    NA    NA
```

12.3.2

```
flights |>  
  filter(month == 11 | month == 12)
```

“Find all flights that departed in November or December.”:

```
flights |>
  filter(month == 11 | 12)
#> # A tibble: 336,776 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>     <int>           <int>     <dbl>     <int>           <int>
#> 1 2013     1     1       517             515      2         830            819
#> 2 2013     1     1       533             529      4         850            830
#> 3 2013     1     1       542             540      2         923            850
#> 4 2013     1     1       544             545     -1        1004           1022
#> 5 2013     1     1       554             600     -6        812            837
#> 6 2013     1     1       554             558     -4        740            728
#> # i 336,770 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

```
          R      month == 11      nov      nov | 12
0    TRUE    nov | TRUE TRUE | TRUE TRUE
```

```

flights |>
  mutate(
    nov = month == 11,
    final = nov | 12,
    .keep = "used"
  )
#> # A tibble: 336,776 x 3
#>   month nov   final
#>   <int> <lgl> <lgl>
#> 1     1 FALSE TRUE
#> 2     1 FALSE TRUE
#> 3     1 FALSE TRUE
#> 4     1 FALSE TRUE
#> 5     1 FALSE TRUE
#> 6     1 FALSE TRUE
#> # i 336,770 more rows

```

12.3.3 %in%

```

==s |s      %in% x %in% y   x      x   y      TRUE
1:12 %in% c(1, 5, 11)
#> [1] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE
letters[1:10] %in% c("a", "e", "i", "o", "u")
#> [1] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE
11 12 : 
flights |>
  filter(month %in% c(11, 12))

```

%in% NA == NA %in% NA TRUE

Note that %in% obeys different rules for NA to ==, as NA %in% NA is TRUE.

```

c(1, 2, NA) == NA
#> [1] NA NA NA
c(1, 2, NA) %in% NA
#> [1] FALSE FALSE TRUE
:
```

```

flights |>
  filter(dep_time %in% c(NA, 0800))
#> # A tibble: 8,803 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>    <int>        <int>     <dbl>    <int>        <int>
#> 1 2013     1     1      800        800       0    1022        1014
#> 2 2013     1     1      800        810      -10     949        955
#> 3 2013     1     1       NA       1630       NA       NA        1815
#> 4 2013     1     1       NA       1935       NA       NA        2240
#> 5 2013     1     1       NA       1500       NA       NA        1825
#> 6 2013     1     1       NA       600        NA       NA        901
#> # i 8,797 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...

```

12.3.4

| | | | | | |
|----|-----------|-----------|----------|----------------|-----------|
| 1. | arr_delay | dep_delay | arr_time | sched_arr_time | arr_delay |
| 2. | dep_time | | | | |
| 3. | dep_time | | | | |

12.4

12.4.1 logical summaries

```

any() all() any(x)  |  x  TRUE      TRUE all(x)  &  x  TRUE      TRUE
NA          na.rm = TRUE

all() any()                      group_by()

flights |>
  group_by(year, month, day) |>
  summarize(
    all_delayed = all(dep_delay <= 60, na.rm = TRUE),
    any_long_delay = any(arr_delay >= 300, na.rm = TRUE),
    .groups = "drop"
  )
#> # A tibble: 365 x 5
#>   year month   day all_delayed any_long_delay
#>   <int> <int> <int> <lgl>        <lgl>
#> 1 2013     1     1 FALSE        TRUE

```

```
#> 2 2013 1 2 FALSE TRUE
#> 3 2013 1 3 FALSE FALSE
#> 4 2013 1 4 FALSE FALSE
#> 5 2013 1 5 FALSE TRUE
#> 6 2013 1 6 FALSE FALSE
#> # i 359 more rows
```

| | | |
|-------------|------------|-------------------|
| any() all() | TRUE FALSE | numeric summaries |
|-------------|------------|-------------------|

12.4.2

| | | | | | | |
|---------|---------|--------|-------|----------|--------|------|
| TRUE | 1 FALSE | 0 | sum() | mean() | sum(x) | TRUE |
| mean(x) | TRUE | mean() | sum() | length() | | |

```
flights |>
  group_by(year, month, day) |>
  summarize(
    proportion_delayed = mean(dep_delay <= 60, na.rm = TRUE),
    count_long_delay = sum(arr_delay >= 300, na.rm = TRUE),
    .groups = "drop"
  )
#> # A tibble: 365 x 5
#>   year month   day proportion_delayed count_long_delay
#>   <int> <int> <int>             <dbl>           <int>
#> 1 2013     1     1            0.939            3
#> 2 2013     1     2            0.914            3
#> 3 2013     1     3            0.941            0
#> 4 2013     1     4            0.953            0
#> 5 2013     1     5            0.964            1
#> 6 2013     1     6            0.959            0
#> # i 359 more rows
```

12.4.3 logical subsetting

[“subset” ??

```
flights |>
  filter(arr_delay > 0) |>
  group_by(year, month, day) |>
  summarize(
```

```

behind = mean(arr_delay),
n = n(),
.groups = "drop"
)
#> # A tibble: 365 x 5
#>   year month   day behind     n
#>   <int> <int> <int>  <dbl> <int>
#> 1  2013     1     1    32.5    461
#> 2  2013     1     2    32.0    535
#> 3  2013     1     3    27.7    460
#> 4  2013     1     4    28.3    297
#> 5  2013     1     5    22.6    238
#> 6  2013     1     6    24.4    381
#> # i 359 more rows

```

```
[ arr_delay[arr_delay
> 0]
```

```

flights |>
  group_by(year, month, day) |>
  summarize(
    behind = mean(arr_delay[arr_delay > 0], na.rm = TRUE),
    ahead = mean(arr_delay[arr_delay < 0], na.rm = TRUE),
    n = n(),
    .groups = "drop"
)
#> # A tibble: 365 x 6
#>   year month   day behind ahead     n
#>   <int> <int> <int>  <dbl> <dbl> <int>
#> 1  2013     1     1    32.5 -12.5    842
#> 2  2013     1     2    32.0 -14.3    943
#> 3  2013     1     3    27.7 -18.2    914
#> 4  2013     1     4    28.3 -17.0    915
#> 5  2013     1     5    22.6 -14.0    720
#> 6  2013     1     6    24.4 -13.6    832
#> # i 359 more rows

```

n() n()

12.4.4

1. sum(is.na(x)) mean(is.na(x))
2. prod() min()

12.5

| | | |
|---|---|-----------------------|
| x | y | if_else() case_when() |
|---|---|-----------------------|

12.5.1 if_else()

| | | | |
|-------------------|---------------|---------------------------|--------------------------|
| TRUE
condition | FALSE
true | dplyr::if_else()
false | if_else()
“+ve” “-ve” |
|-------------------|---------------|---------------------------|--------------------------|

```
x <- c(-3:3, NA)
if_else(x > 0, "+ve", "-ve")
#> [1] "-ve" "-ve" "-ve" "-ve" "+ve" "+ve" "+ve" NA
```

There's an optional fourth argument, `missing` which will be used if the input is `NA`:

```
if_else(x > 0, "+ve", "-ve", "??")
#> [1] "-ve" "-ve" "-ve" "-ve" "+ve" "+ve" "+ve" "??"
```

| | | |
|------------|-------|------------|
| true false | abs() | coalesce() |
|------------|-------|------------|

```
if_else(x < 0, -x, x)
#> [1] 3 2 1 0 1 2 3 NA
```

| | | |
|-----------|--------------------|--|
| if_else() | dplyr::case_when() | |
|-----------|--------------------|--|

```
if_else(x == 0, "0", if_else(x < 0, "-ve", "+ve"), "??")
#> [1] "-ve" "-ve" "-ve" "0" "+ve" "+ve" "+ve" "??"
```

12.5.2 `case_when()`

```
dplyr case_when() SQL CASE tidyverse
  condition ~ output  condition TRUE  output
  if_else()
```

```
x <- c(-3:3, NA)
case_when(
  x == 0 ~ "0",
  x < 0 ~ "-ve",
  x > 0 ~ "+ve",
  is.na(x) ~ "???"
```

#> [1] "-ve" "-ve" "-ve" "0" "+ve" "+ve" "+ve" "???"

```
case_when() NA
```

```
case_when(
  x < 0 ~ "-ve",
  x > 0 ~ "+ve"
)
#> [1] "-ve" "-ve" "-ve" NA    "+ve" "+ve" "+ve" NA
```

```
“ ” .default
```

```
case_when(
  x < 0 ~ "-ve",
  x > 0 ~ "+ve",
  .default = "???"
```

#> [1] "-ve" "-ve" "-ve" "???" "+ve" "+ve" "+ve" "???"

```
:
```

```
case_when(
  x > 0 ~ "+ve",
  x > 2 ~ "big"
)
#> [1] NA    NA    NA    NA    "+ve" "+ve" "+ve" NA
```

```
if_else() ~ case_when()
```

```

flights |>
  mutate(
    status = case_when(
      is.na(arr_delay) ~ "cancelled",
      arr_delay < -30 ~ "very early",
      arr_delay < -15 ~ "early",
      abs(arr_delay) <= 15 ~ "on time",
      arr_delay < 60 ~ "late",
      arr_delay < Inf ~ "very late",
    ),
    .keep = "used"
  )
#> # A tibble: 336,776 x 2
#>   arr_delay status
#>       <dbl> <chr>
#> 1        11 on time
#> 2        20 late
#> 3        33 late
#> 4       -18 early
#> 5       -25 early
#> 6        12 on time
#> # i 336,770 more rows

```

case_when() < >

12.5.3 compatible types

```

if_else() case_when()

if_else(TRUE, "a", 1)
#> Error in `if_else()`:
#> ! Can't combine `true` <character> and `false` <double>.

case_when(
  x < -1 ~ TRUE,
  x > 0 ~ now()
)
#> Error in `case_when()`:
#> ! Can't combine `..1 (right)` <logical> and `..2 (right)` <datetime<local>>.

```

- ??
- - ??
- NA,

tidyverse

12.5.4

```

1.      2      R      x %% 2 == 0      if_else()  0 20
2.      x <- c("Monday", "Saturday", "Wednesday") if_else()
3. if_else()      x
4. case_when()   flights   month day      7 4
                  TRUE FALSE          NA

```

12.6

```

TRUE FALSE NA      > < <= >= == != is.na()      ! & |      any() all() sum() mean()
if_else() case_when()
@sec-strings    str_detect(x, pattern)      x      TRUE @sec-
dates-and-times

```


Chapter 13

13.1

```
R  
  count()      mutate()  
summarize()      mutate()
```

13.1.1

```
R      tidyverse      tidyverse      mutate() filter()  
nycflights13      c() tribble()
```

```
library(tidyverse)  
library(nycflights13)
```

13.2

```
R  
readr      parse_double() parse_number()      parse_double()  
  
x <- c("1.2", "5.6", "1e3")  
parse_double(x)  
#> [1] 1.2 5.6 1000.0
```

```
parse_number()
```

```
x <- c("$1,234", "USD 3,513", "59%")
parse_number(x)
#> [1] 1234 3513   59
```

13.3

```
dplyr    count()

flights |> count(dest)
#> # A tibble: 105 x 2
#>   dest      n
#>   <chr> <int>
#> 1 ABQ      254
#> 2 ACK      265
#> 3 ALB      439
#> 4 ANC       8
#> 5 ATL     17215
#> 6 AUS     2439
#> # i 99 more rows

??          count()
sort = TRUE:

flights |> count(dest, sort = TRUE)
#> # A tibble: 105 x 2
#>   dest      n
#>   <chr> <int>
#> 1 ORD     17283
#> 2 ATL     17215
#> 3 LAX     16174
#> 4 BOS     15508
#> 5 MCO     14082
#> 6 CLT     14064
#> # i 99 more rows

|> View() |> print(n = Inf)

group_by() summarize() n()“ ”

flights |>
  group_by(dest) |>
  summarize(
    n = n(),
```

```

delay = mean(arr_delay, na.rm = TRUE)
)
#> # A tibble: 105 x 3
#>   dest      n delay
#>   <chr> <int> <dbl>
#> 1 ABQ      254  4.38
#> 2 ACK      265  4.85
#> 3 ALB      439 14.4
#> 4 ANC       8 -2.5
#> 5 ATL     17215 11.3
#> 6 AUS     2439  6.02
#> # i 99 more rows

n()           “ ”      dplyr

n()
#> Error in `n()`:
#> ! Must only be used inside data-masking verbs like `mutate()`,
#> `filter()`, and `group_by()`.

n() count()

• n_distinct(x)

flights |>
  group_by(dest) |>
  summarize(carriers = n_distinct(carrier)) |>
  arrange(desc(carriers))
#> # A tibble: 105 x 2
#>   dest    carriers
#>   <chr>     <int>
#> 1 ATL        7
#> 2 BOS        7
#> 3 CLT        7
#> 4 ORD        7
#> 5 TPA        7
#> 6 AUS        6
#> # i 99 more rows

•           “ ”

flights |>
  group_by(tailnum) |>
  summarize(miles = sum(distance))

```

```
#> # A tibble: 4,044 x 2
#>   tailnum    miles
#>   <chr>     <dbl>
#> 1 D942DN     3418
#> 2 NOEGMQ    250866
#> 3 N10156    115966
#> 4 N102UW     25722
#> 5 N103US     24619
#> 6 N104UW     25157
#> # i 4,038 more rows
count()    wt
flights |> count(tailnum, wt = distance)
```

- sum() is.na() flights

```
flights |>
  group_by(dest) |>
  summarize(n_cancelled = sum(is.na(dep_time)))
#> # A tibble: 105 x 2
#>   dest  n_cancelled
#>   <chr>     <int>
#> 1 ABQ        0
#> 2 ACK        0
#> 3 ALB       20
#> 4 ANC        0
#> 5 ATL      317
#> 6 AUS       21
#> # i 99 more rows
```

13.3.1

1. count()
2. count() group_by() summarize() arrange():
 1. flights |> count(dest, sort = TRUE)
 2. flights |> count(tailnum, wt = distance)

13.4

`mutate()`

R

R

13.4.1

```
??      +, -, *, /, ^
rules          recycling
60)    /  336,776
```

R

```
x <- c(1, 2, 10, 20)
x / 5
#> [1] 0.2 0.4 2.0 4.0
# is shorthand for
x / c(5, 5, 5, 5)
#> [1] 0.2 0.4 2.0 4.0
```

1 R

R

```
x * c(1, 2)
#> [1] 1 4 10 40
x * c(1, 2, 3)
#> Warning in x * c(1, 2, 3): longer object length is not a multiple of shorter
#> object length
#> [1] 1 4 30 20
```

```
==, <, <=, >, >=, !=      == %in%
1 2
```

```
flights |>
  filter(month == c(1, 2))
#> # A tibble: 25,977 x 19
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>   <int>           <int>     <dbl>   <int>           <int>
#> 1  2013     1     1     517            515       2     830            819
#> 2  2013     1     1     542            540       2     923            850
#> 3  2013     1     1     554            600      -6     812            837
#> 4  2013     1     1     555            600      -5     913            854
#> 5  2013     1     1     557            600      -3     838            846
#> 6  2013     1     1     558            600      -2     849            851
#> # i 25,971 more rows
#> # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

1 2 flights

tidyverse

R == filter()

13.4.2

```
pmin() pmax()

df <- tribble(
  ~x, ~y,
  1, 3,
  5, 2,
  7, NA,
)

df |>
  mutate(
    min = pmin(x, y, na.rm = TRUE),
    max = pmax(x, y, na.rm = TRUE)
  )
#> # A tibble: 3 x 4
#>   x     y     min     max
#>   <dbl> <dbl> <dbl> <dbl>
#> 1 1     3     1     3
#> 2 5     2     2     5
#> 3 7     NA    7     7
```

```
min() max()

df |>
  mutate(
    min = min(x, y, na.rm = TRUE),
    max = max(x, y, na.rm = TRUE)
  )
#> # A tibble: 3 x 4
#>   x     y     min     max
#>   <dbl> <dbl> <dbl> <dbl>
#> 1 1     3     1     7
#> 2 5     2     1     7
#> 3 7     NA    1     7
```

13.4.3

| | | |
|--------------------|---|-------------|
| modular arithmetic | R | %/% %% |
|--------------------|---|-------------|

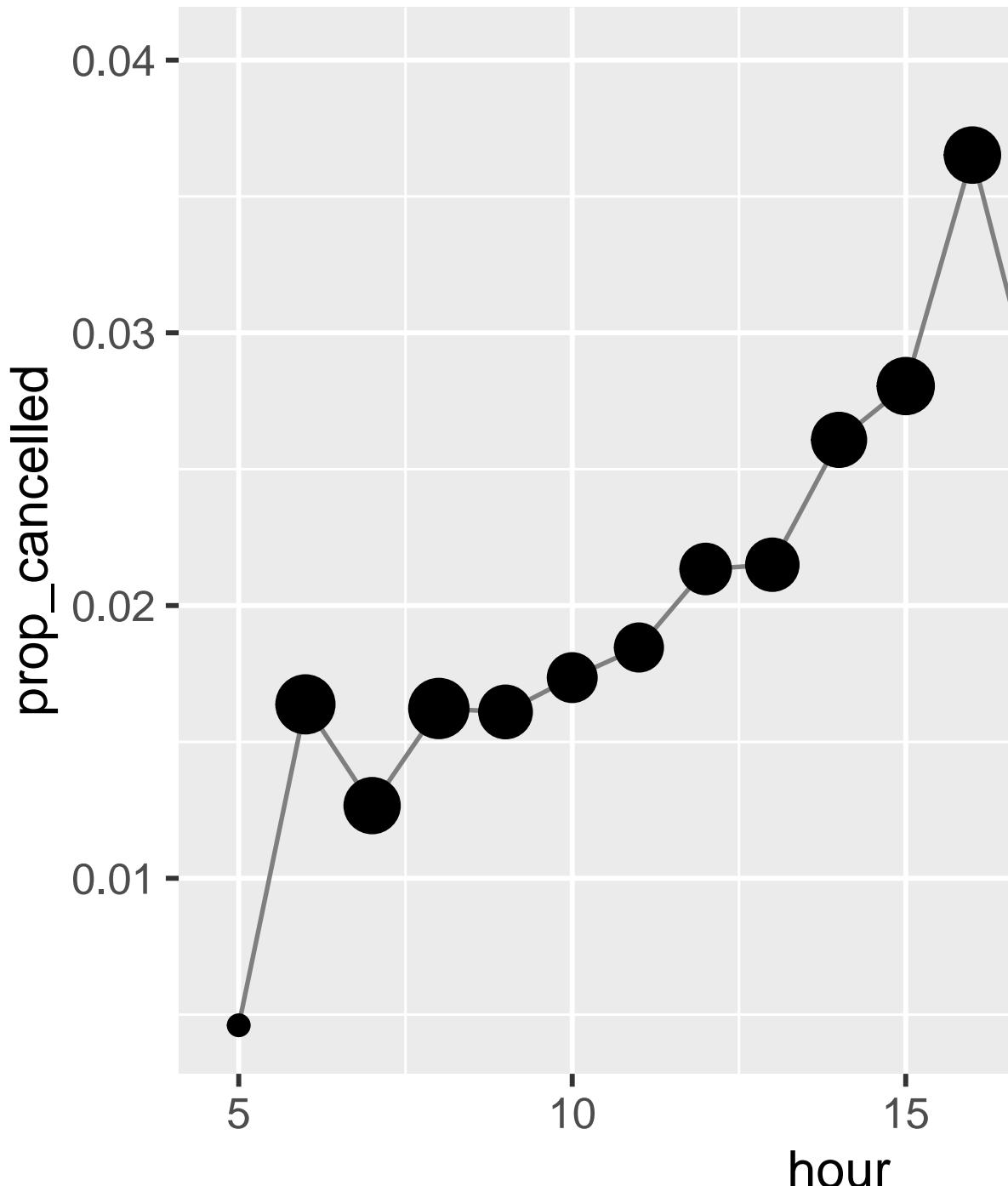
```
1:10 %/% 3
#> [1] 0 0 1 1 1 2 2 2 3 3
1:10 %% 3
#> [1] 1 2 0 1 2 0 1 2 0 1
```

```
flights      sched_dep_time  hour minute
```

```
flights |>
  mutate(
    hour = sched_dep_time %/%
    minute = sched_dep_time %% 100,
    .keep = "used"
  )
#> # A tibble: 336,776 x 3
#>   sched_dep_time  hour minute
#>   <int> <dbl> <dbl>
#> 1 515     5     15
#> 2 529     5     29
#> 3 540     5     40
#> 4 545     5     45
#> 5 600     6     0
#> 6 558     5     58
#> # i 336,770 more rows
```

```
?? mean(is.na(x)) ??
```

```
flights |>
  group_by(hour = sched_dep_time %/%
            100) |>
  summarize(prop_cancelled = mean(is.na(dep_time)), n = n()) |>
  filter(hour > 1) |>
  ggplot(aes(x = hour, y = prop_cancelled)) +
  geom_line(color = "grey50") +
  geom_point(aes(size = n))
```



13.1: A line plot with scheduled departure hour on the x-axis, and proportion of cancelled flights on the y-axis. Cancellations seem to accumulate over the course of the day until 8pm, very late flights are much less likely to be cancelled.

13.4.4

| | | | | | | | |
|---|--------|---------|-------|--------|---------|--------------|---------------|
| | R | log() | e | log2() | 2 | log10() | 10 |
| | log2() | log10() | 1 | -1 | log10() | 3 | 10 |
| 3 | 1000 | log() | exp() | log2() | log10() | 2^{\wedge} | 10^{\wedge} |

13.4.5

```
round(x)          :
round(123.456)
#> [1] 123

digits      round(x, digits) x      10^-n digits = 2 x
0.01        round(x, -3) x

round(123.456, 2) # two digits
#> [1] 123.46
round(123.456, 1) # one digit
#> [1] 123.5
round(123.456, -1) # round to nearest ten
#> [1] 120
round(123.456, -2) # round to nearest hundred
#> [1] 100

round()

round(c(1.5, 2.5))
#> [1] 2 2

round()      "   " "   "
round() floor() ceiling()  floor()    ceiling()

x <- 123.456

floor(x)
#> [1] 123
ceiling(x)
#> [1] 124

floor() ceiling() digits
```

```
# Round down to nearest two digits
floor(x / 0.01) * 0.01
#> [1] 123.45

# Round up to nearest two digits
ceiling(x / 0.01) * 0.01
#> [1] 123.46

round()

# Round to nearest multiple of 4
round(x / 4) * 4
#> [1] 124

# Round to nearest 0.25
round(x / 0.25) * 0.25
#> [1] 123.5
```

13.4.6

`cut()`¹

```
x <- c(1, 2, 5, 10, 15, 20)
cut(x, breaks = c(0, 5, 10, 15, 20))
#> [1] (0,5]   (0,5]   (0,5]   (5,10]  (10,15] (15,20]
#> Levels: (0,5] (5,10] (10,15] (15,20]
```

breaks

```
cut(x, breaks = c(0, 5, 10, 100))
#> [1] (0,5]   (0,5]   (0,5]   (5,10]   (10,100] (10,100]
#> Levels: (0,5] (5,10] (10,100]
```

labels labels breaks

```
cut(x,
  breaks = c(0, 5, 10, 15, 20),
  labels = c("sm", "md", "lg", "xl")
)
#> [1] sm sm sm md lg xl
#> Levels: sm md lg xl
```

¹ggplot2 cut_interval(),cut_number() cut_width()
ggplot2 tidyverse

```

breaks      NA

y <- c(NA, -10, 5, 10, 30)
cut(y, breaks = c(0, 5, 10, 15, 20))
#> [1] <NA>    <NA>    (0,5]   (5,10]  <NA>
#> Levels: (0,5] (5,10] (10,15] (15,20]

right include.lowest      [a, b)  (a, b]      [a, b]

```

13.4.7

R cumsum(),cumprod(),cummin(),cummax()
cummean()

```

x <- 1:10
cumsum(x)
#> [1] 1 3 6 10 15 21 28 36 45 55

```

slider

13.4.8

1. @fig-prop-cancelled
2. R
3. dep_time sched_dep_time

```

flights |>
  filter(month == 1, day == 1) |>
  ggplot(aes(x = sched_dep_time, y = dep_delay)) +
  geom_point()

```

4. dep_time arr_time

13.5

13.5.1 Ranks

```
dplyr      SQL          dplyr::min_rank()           1  2  2  4
x <- c(1, 2, 2, 3, 4, NA)
min_rank(x)
#> [1] 1 2 2 4 5 NA

desc(x)

min_rank(desc(x))
#> [1] 5 3 3 2 1 NA

min_rank()           dplyr::row_number() dplyr::dense_rank() dplyr::percent_rank() dplyr::rank()

df <- tibble(x = x)
df |>
  mutate(
    row_number = row_number(x),
    dense_rank = dense_rank(x),
    percent_rank = percent_rank(x),
    cume_dist = cume_dist(x)
  )
#> # A tibble: 6 x 5
#>   x   row_number   dense_rank   percent_rank   cume_dist
#>   <dbl>     <int>       <int>        <dbl>      <dbl>
#> 1 1         1          1          1          0        0.2
#> 2 2         2          2          2          0.25     0.6
#> 3 2         3          3          2          0.25     0.6
#> 4 3         4          4          3          0.75     0.8
#> 5 4         5          5          4          1        1
#> 6 NA        NA         NA         NA         NA

ties.method      R      rank()           na.last = "keep"  NAs
NA

row_number()  dplyr           " "      %% %/%
df <- tibble(id = 1:10)

df |>
  mutate(
    row0 = row_number() - 1,
    three_groups = row0 %% 3,
```

```

    three_in_each_group = row0 %/% 3
  )
#> # A tibble: 10 x 4
#>   id    row0 three_groups three_in_each_group
#>   <int> <dbl>      <dbl>              <dbl>
#> 1     1     0          0                 0
#> 2     2     1          1                 0
#> 3     3     2          2                 0
#> 4     4     3          0                 1
#> 5     5     4          1                 1
#> 6     6     5          2                 1
#> # i 4 more rows

```

13.5.2 Offsets

```
dplyr::lead() dplyr::lag()   " "           NAs
```

```

x <- c(2, 5, 11, 11, 19, 35)
lag(x)
#> [1] NA 2 5 11 11 19
lead(x)
#> [1] 5 11 11 19 35 NA

```

- `x - lag(x)`

```

x - lag(x)
#> [1] NA 3 6 0 8 16

```

- `x == lag(x)`

```

x == lag(x)
#> [1] NA FALSE FALSE TRUE FALSE FALSE

```

n

13.5.3

```
sessions      x
```

```

events <- tibble(
  time = c(0, 1, 2, 3, 5, 10, 12, 15, 17, 19, 20, 27, 28, 30)
)

```

```
events <- events |>
  mutate(
    diff = time - lag(time, default = first(time)),
    has_gap = diff >= 5
  )
events
#> # A tibble: 14 x 3
#>   time   diff has_gap
#>   <dbl> <dbl> <lgl>
#> 1     0     0 FALSE
#> 2     1     1 FALSE
#> 3     2     1 FALSE
#> 4     3     1 FALSE
#> 5     5     2 FALSE
#> 6    10     5 TRUE
#> # i 8 more rows
```

group_by() cumsum() ?? has_gap TRUE @sec-
numeric-summaries-of-logicals

```
events |> mutate(
  group = cumsum(has_gap)
)
#> # A tibble: 14 x 4
#>   time   diff has_gap group
#>   <dbl> <dbl> <lgl>   <int>
#> 1     0     0 FALSE      0
#> 2     1     1 FALSE      0
#> 3     2     1 FALSE      0
#> 4     3     1 FALSE      0
#> 5     5     2 FALSE      0
#> 6    10     5 TRUE       1
#> # i 8 more rows
```

consecutive_id() this stackoverflow question

```
df <- tibble(
  x = c("a", "a", "a", "b", "c", "c", "d", "e", "a", "a", "b", "b"),
  y = c(1, 2, 3, 2, 4, 1, 3, 9, 4, 8, 10, 199)
)
```

x group_by() consecutive_id() slice_head()

```
df |>
  group_by(id = consecutive_id(x)) |>
  slice_head(n = 1)
#> # A tibble: 7 x 3
#> # Groups:   id [7]
#>   x     y   id
#>   <chr> <dbl> <int>
#> 1 a      1     1
#> 2 b      2     2
#> 3 c      4     3
#> 4 d      3     4
#> 5 e      9     5
#> 6 a      4     6
#> # i 1 more row
```

13.5.4

1. 10 min_rank()
2. tailnum
- 3.
4. flights |> group_by(dest) |> filter(row_number() < 4) f lights |> group_by(dest) |> filter(row_number(dep_delay) < 4)
- 5.

6. lag()

```
flights |>
  mutate(hour = dep_time %/% 100) |>
  group_by(year, month, day, hour) |>
  summarize(
    dep_delay = mean(dep_delay, na.rm = TRUE),
    n = n(),
    .groups = "drop"
  ) |>
  filter(n > 5)
```

7.

8.

13.6

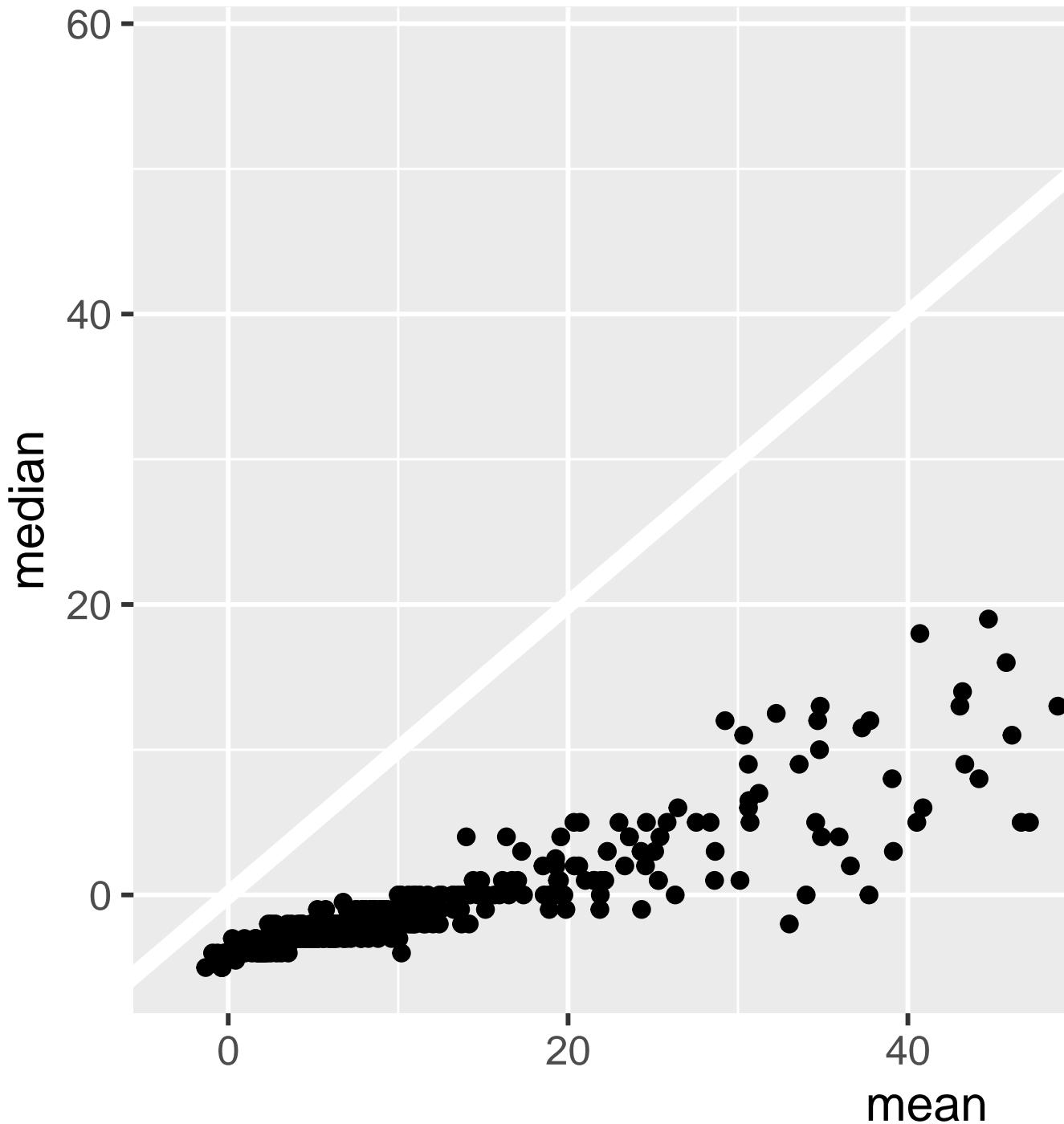
counts means sums R

13.6.1

`mean()` @sec-sample-size `median()` “ ” 50%

??

```
flights |>
  group_by(year, month, day) |>
  summarize(
    mean = mean(dep_delay, na.rm = TRUE),
    median = median(dep_delay, na.rm = TRUE),
    n = n(),
    .groups = "drop"
  ) |>
  ggplot(aes(x = mean, y = median)) +
  geom_abline(slope = 1, intercept = 0, color = "white", linewidth = 2) +
  geom_point()
```



13.2: A scatterplot showing the differences of summarizing daily departure delay with median instead of mean.

mode
R 2

13.6.2

```

      m   in() max()           quantile()    quantile(x,
0.25)  25% x quantile(x, 0.5)  quantile(x, 0.95)  95% x
flights      95%           5%
flights |>
  group_by(year, month, day) |>
  summarize(
    max = max(dep_delay, na.rm = TRUE),
    q95 = quantile(dep_delay, 0.95, na.rm = TRUE),
    .groups = "drop"
  )
#> # A tibble: 365 x 5
#>   year month   day   max   q95
#>   <int> <int> <int> <dbl> <dbl>
#> 1  2013     1     1   853  70.1
#> 2  2013     1     2   379   85
#> 3  2013     1     3   291   68
#> 4  2013     1     4   288   60
#> 5  2013     1     5   327   41
#> 6  2013     1     6   202   51
#> # i 359 more rows

```

13.6.3 Spread

```

      sd(x)   IQR()       sd()       IQR()       quantile(x,
0.75) - quantile(x, 0.25)  50%
flights

```

EGE

```

flights |>
  group_by(origin, dest) |>
  summarize(
    distance_iqr = IQR(distance),
    n = n(),
    .groups = "drop"
  ) |>
  filter(distance_iqr > 0)

```

²mode() !

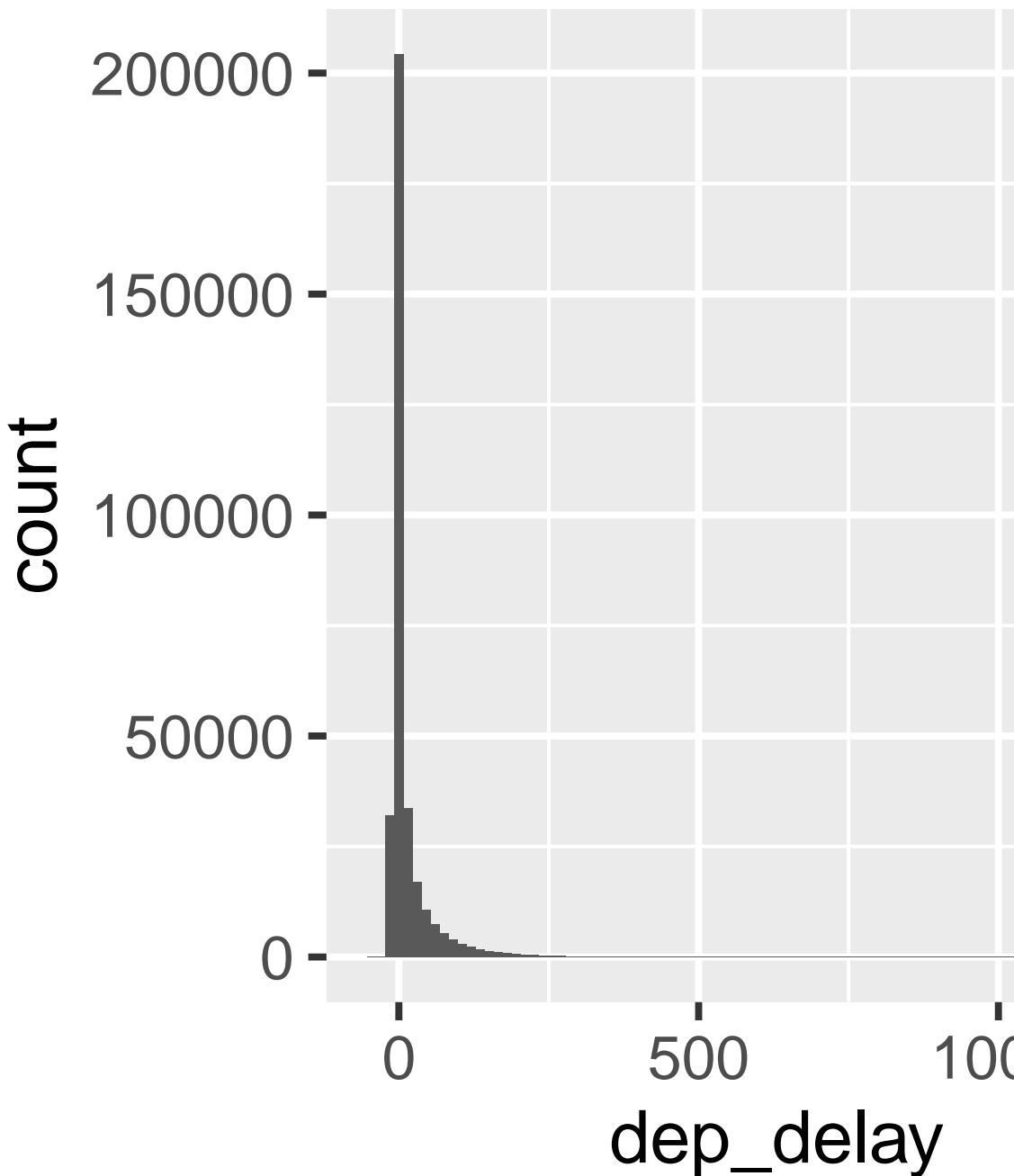
```
#> # A tibble: 2 x 4
#>   origin dest  distance_iqr     n
#>   <chr>   <chr>      <dbl> <int>
#> 1 EWR     EGE        1    110
#> 2 JFK     EGE        1    103
```

13.6.4

??

dep_delay 365 frequency polygons

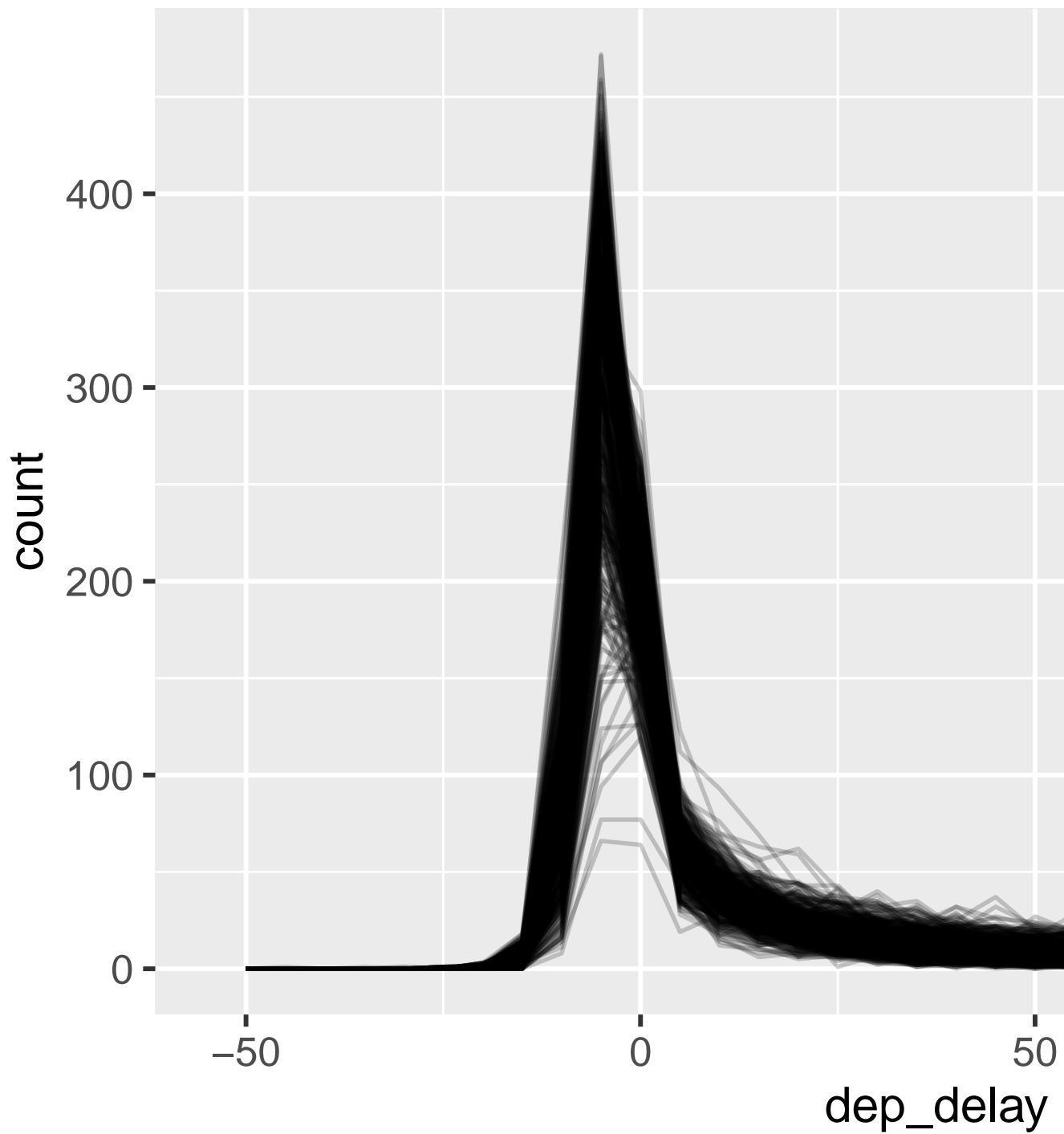
```
flights |>
  filter(dep_delay < 120) |>
  ggplot(aes(x = dep_delay, group = interaction(day, month))) +
  geom_freqpoly(binwidth = 5, alpha = 1/5)
```



13.3: (Left) The histogram of the full data is extremely skewed making it hard to get any details. (Right) Zooming into delays of less than two hours makes it possible to see what's happening with the bulk of the observations.

13.6.

347



@sec-sample-size

13.6.5

```
first(x) last(x) nth(x, n)
```

```
flights |>
  group_by(year, month, day) |>
  summarise(
    first_dep = first(dep_time, na_rm = TRUE),
    fifth_dep = nth(dep_time, 5, na_rm = TRUE),
    last_dep = last(dep_time, na_rm = TRUE)
  )
#> `summarise()` has grouped output by 'year', 'month'. You can override using
#> the ` `.groups` argument.
#> # A tibble: 365 x 6
#> # Groups:   year, month [12]
#>   year month   day first_dep fifth_dep last_dep
#>   <int> <int> <int>     <int>     <int>     <int>
#> 1  2013     1     1       517       554      2356
#> 2  2013     1     2        42       535      2354
#> 3  2013     1     3        32       520      2349
#> 4  2013     1     4        25       531      2358
#> 5  2013     1     5        14       534      2357
#> 6  2013     1     6        16       555      2355
#> # i 359 more rows
```

```
(  dplyr _           na_rm  na.rm
[  @sec-subset-many          order_by      na_rm
```

```
flights |>
  group_by(year, month, day) |>
  mutate(r = min_rank(sched_dep_time)) |>
  filter(r %in% c(1, max(r)))
#> # A tibble: 1,195 x 20
#> # Groups:   year, month, day [365]
#>   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#>   <int> <int> <int>     <int>           <int>     <dbl>     <int>           <int>
#> 1  2013     1     1       517            515        2     830            819
#> 2  2013     1     1      2353           2359       -6     425            445
```

```
#> 3 2013     1     1    2353      2359     -6    418     442
#> 4 2013     1     1    2356      2359     -3    425     437
#> 5 2013     1     2     42      2359     43    518     442
#> 6 2013     1     2    458      500     -2    703     650
#> # i 1,189 more rows
#> # i 12 more variables: arr_delay <dbl>, carrier <chr>, flight <int>, ...
```

13.6.6 `mutate()`

`summarize()` @sec-recycling `mutate()`

- `x / sum(x)` x
- $(x - \text{mean}(x)) / \text{sd}(x)$ Z-score $x \quad 0 \quad 1$
- $(x - \text{min}(x)) / (\text{max}(x) - \text{min}(x))$ $x \quad [0, 1]$
- `x / first(x)`

13.6.7

1. 5 `mean()` `median()`
planes
- 2.
3. EGE

13.7

R

`stringr`

Chapter 14

14.1

14.1.1

```
stringr      stringr tidyverse      babynames
```

```
library(tidyverse)
library(babynames)
```

```
stringr          stringr  str_      RStudio      str_
> str_c           (string) | str_c(..., sep = "", collapse = NULL)
> str_conv        (string) | To understand how str_c works, you need to know that you are
> str_count       (string) | building up a matrix of strings. Each input argument needs a
> str_detect      (string) | column. This is expanded to the length of the longest argument,
> str_dup          (string) | using the usual recycling rules. The sep string is inserted between
> str_extract     (string) | each column. If collapse is NULL, each row is collapsed into a single
> str_extract_all (string) | string. If collapse is not NULL, that string is inserted at the end of each row,
> str_             (string) | and the entire matrix collapsed to a single string.
> str_             (string) | Press F1 for additional help
> str_
```

14.2

```
  "      tidyverse      "      "
```

```
string1 <- "This is a string"
string2 <- 'If I want to include a "quote" inside a string, I use single quotes'
```

+

```
> "This is a string without a closing quote
+
+
+ HELP I'M STUCK IN A STRING
```

Esc

14.2.1 Escapes

\ “ ”

```
double_quote <- "\\" # or ''
single_quote <- '\'' # or '''
```

:\"\\":

```
backslash <- "\\\"
```

`str_view()`¹

```
x <- c(single_quote, double_quote, backslash)
x
#> [1] "" "\'" "\\"

str_view(x)
#> [1] |
#> [2] |
#> [3] | \
```

14.2.2

double_quote single_quote

¹ R `writeLines()`.

```
tricky <- "double_quote <- \"\\\\\" # or '\"'
single_quote <- '\\'' # or '\"\\\"'
str_view(tricky)
#> [1] | double_quote <- "\"" # or """
#>     | single_quote <- '\\'' # or """

```

!()²:

```
tricky <- r"(double_quote <- \"\" # or """
single_quote <- '\\'' # or '\"")"
str_view(tricky)
#> [1] | double_quote <- "\"" # or """
#>     | single_quote <- '\\'' # or """

```

r"(" ")")" r"[]" r"{}" "r"--()--" r"---()---"

14.2.3

| | | | |
|----------|-------|---------------|---------|
| \" \' \\ | \n \t | \u \U Unicode | ?Quotes |
|----------|-------|---------------|---------|

```
x <- c("one\ntwo", "one\ttwo", "\u00b5", "\u0001f604")
x
#> [1] "one\ntwo" "one\ttwo" "μ"      ""
str_view(x)
#> [1] | one
#>     | two
#> [2] | one{\t}two
#> [3] | μ
#> [4] |
```

str_view()³

14.2.4

1. :

1. He said "That's amazing!"
2. \a\b\c\d
3. \\\\\\

² R 4.0.0

³ str_view()

```
2. R           "\u00a0"   s tr_view()
```

```
x <- "This\u00a0is\u00a0tricky"
```

14.3

| | |
|--------------------|-------------|
| “ ” | “Hello” |
| str_c() str_glue() | mutate() |
| | summarize() |
| | stringr |
| | str_fl |

14.3.1 str_c()

```
str_c()          :
```

```
str_c("x", "y")
#> [1] "xy"
str_c("x", "y", "z")
#> [1] "xyz"
str_c("Hello ", c("John", "Susan"))
#> [1] "Hello John"  "Hello Susan"
```

| | | | | | |
|----------------|------------|----------|-----------|-----------|-------------|
| str_c() | R paste0() | mutate() | tidyverse | recycling | propagating |
| missing values | | | | | |

```
df <- tibble(name = c("Flora", "David", "Terra", NA))
df |> mutate(greeting = str_c("Hi ", name, "!"))
#> # A tibble: 4 x 2
#>   name  greeting
#>   <chr> <chr>
#> 1 Flora Hi Flora!
#> 2 David Hi David!
#> 3 Terra Hi Terra!
#> 4 <NA>  <NA>
```

| | | |
|------------|---------|------------|
| coalesce() | str_c() | coalesce() |
|------------|---------|------------|

```
df |>
  mutate(
    greeting1 = str_c("Hi ", coalesce(name, "you"), "!"),
    greeting2 = coalesce(str_c("Hi ", name, "!"), "Hi!")
  )
#> # A tibble: 4 x 3
#>   name  greeting1 greeting2
```

```
#> <chr> <chr> <chr>
#> 1 Flora Hi Flora! Hi Flora!
#> 2 David Hi David! Hi David!
#> 3 Terra Hi Terra! Hi Terra!
#> 4 <NA> Hi you! Hi!
```

14.3.2 str_glue()

```
str_c()           "s           glue       str_glue()4      {}

df |> mutate(greeting = str_glue("Hi {name}!"))
#> # A tibble: 4 x 2
#>   name  greeting
#>   <chr> <glue>
#> 1 Flora Hi Flora!
#> 2 David Hi David!
#> 3 Terra Hi Terra!
#> 4 <NA>  Hi NA!

str_glue()      "NA"    str_c()
{ }             glue      \
df |> mutate(greeting = str_glue("{Hi {name}!}"))
#> # A tibble: 4 x 2
#>   name  greeting
#>   <chr> <glue>
#> 1 Flora {Hi Flora!}
#> 2 David {Hi David!}
#> 3 Terra {Hi Terra!}
#> 4 <NA>  {Hi NA!}
```

14.3.3 str_flatten()

```
str_c() str_glue() mutate()           summarize()           str_flatten()5

str_flatten(c("x", "y", "z"))
#> [1] "xyz"
str_flatten(c("x", "y", "z"), ", ")
#> [1] "x, y, z"
str_flatten(c("x", "y", "z"), ", ", last = ", and ")
#> [1] "x, y, and z"
```

⁴ stringr::glue()

⁵The base R equivalent is paste() used with the collapse argument.

```

summarize()

df <- tribble(
  ~ name, ~ fruit,
  "Carmen", "banana",
  "Carmen", "apple",
  "Marvin", "nectarine",
  "Terence", "cantaloupe",
  "Terence", "papaya",
  "Terence", "mandarin"
)
df |>
  group_by(name) |>
  summarize(fruits = str_flatten(fruit, ", "))
#> # A tibble: 3 x 2
#>   name    fruits
#>   <chr>   <chr>
#> 1 Carmen  banana, apple
#> 2 Marvin  nectarine
#> 3 Terence cantaloupe, papaya, mandarin

```

14.3.4

1. paste0() str_c() :

```

str_c("hi ", NA)
str_c(letters[1:2], letters[1:3])

```
2. paste() paste0() str_c() paste()
3. str_c() str_glue() :
 - a. str_c("The price of ", food, " is ", price)
 - b. str_glue("I'm {age} years old and live in {country}")
 - c. str_c("\section{", title, "}")

14.4

tidyr

- df |> separate_longer_delim(col, delim)
- df |> separate_longer_position(col, width)
- df |> separate_wider_delim(col, delim, names)

- `df |> separate_wider_position(col, widths)`

- `separate_ longer wider _`

- `pivot_longer() pivot_wider() _longer _wider`
- `delim ", " " " position c(3, 5, 2)`

- `@sec-regular-expressions` `separate_wider_regex()` `wider`

- `wider`

14.4.1

```
separate_longer_delim()

df1 <- tibble(x = c("a,b,c", "d,e", "f"))
df1 |>
  separate_longer_delim(x, delim = ",")
#> # A tibble: 6 x 1
#>   x
#>   <chr>
#> 1 a
#> 2 b
#> 3 c
#> 4 d
#> 5 e
#> 6 f
```



```
separate_longer_position()

df2 <- tibble(x = c("1211", "131", "21"))
df2 |>
  separate_longer_position(x, width = 1)
#> # A tibble: 9 x 1
#>   x
#>   <chr>
#> 1 1
#> 2 2
#> 3 1
#> 4 1
#> 5 1
#> 6 3
#> # i 3 more rows
```

14.4.2

```

longer           x      "."
separate_wider_delim()

df3 <- tibble(x = c("a10.1.2022", "b10.2.2011", "e15.1.2015"))
df3 |>
  separate_wider_delim(
    x,
    delim = ".",
    names = c("code", "edition", "year")
  )
#> # A tibble: 3 x 3
#>   code  edition year
#>   <chr> <chr>   <chr>
#> 1 a10   1       2022
#> 2 b10   2       2011
#> 3 e15   1       2015

```

NA

```

df3 |>
  separate_wider_delim(
    x,
    delim = ".",
    names = c("code", NA, "year")
  )
#> # A tibble: 3 x 2
#>   code  year
#>   <chr> <chr>
#> 1 a10   2022
#> 2 b10   2011
#> 3 e15   2015

```

separate_wider_position()

```

df4 <- tibble(x = c("202215TX", "202122LA", "202325CA"))
df4 |>
  separate_wider_position(
    x,
    widths = c(year = 4, age = 2, state = 2)
  )
#> # A tibble: 3 x 3
#>   year  age  state

```

```
#> <chr> <chr> <chr>
#> 1 2022 15 TX
#> 2 2021 22 LA
#> 3 2023 25 CA
```

14.4.3

```
separate_wider_delim()6          separate_wider_delim()  too_few too_many
      too_few

df <- tibble(x = c("1-1-1", "1-1-2", "1-3", "1-3-2", "1"))

df |>
  separate_wider_delim(
    x,
    delim = "-",
    names = c("x", "y", "z")
  )
#> Error in `separate_wider_delim()`:
#> ! Expected 3 pieces in each element of `x`.
#> ! 2 values were too short.
#> i Use `too_few = "debug"` to diagnose the problem.
#> i Use `too_few = "align_start"/"align_end"` to silence this message.
```

```
debug <- df |>
  separate_wider_delim(
    x,
    delim = "-",
    names = c("x", "y", "z"),
    too_few = "debug"
  )
#> Warning: Debug mode activated: adding variables `x_ok`, `x_pieces`, and
#> `x_remainder`.
debug
#> # A tibble: 5 x 6
#>   x     y     z     x_ok x_pieces x_remainder
#>   <chr> <chr> <chr> <lgl>    <int> <chr>
#> 1 1-1-1 1     1     TRUE        3  ""
#> 2 1-1-2 1     2     TRUE        3  ""
#> 3 1-3   3     <NA> FALSE       2  ""
```

⁶ `separate_wider_position()` `separate_wider_regex()`

```

#> 4 1-3-2 3      2      TRUE      3 ""
#> 5 1      <NA>  <NA> FALSE      1 ""

x_ok x_pieces x_remainder          x_ok

debug |> filter(!x_ok)
#> # A tibble: 2 x 6
#>   x     y     z   x_ok   x_pieces x_remainder
#>   <chr> <chr> <chr> <lgl>    <int> <chr>
#> 1 1-3   3      <NA> FALSE      2 ""
#> 2 1      <NA>  <NA> FALSE      1 ""

x_pieces      3  names      x_remainder
                           too_few = "debug"
NAAs           too_few = "align_start" too_few = "align_end"
NAAs

df |>
  separate_wider_delim(
    x,
    delim = "-",
    names = c("x", "y", "z"),
    too_few = "align_start"
  )
#> # A tibble: 5 x 3
#>   x     y     z
#>   <chr> <chr> <chr>
#> 1 1     1     1
#> 2 1     1     2
#> 3 1     3     <NA>
#> 4 1     3     2
#> 5 1     <NA> <NA>

:

df <- tibble(x = c("1-1-1", "1-1-2", "1-3-5-6", "1-3-2", "1-3-5-7-9"))

df |>
  separate_wider_delim(
    x,
    delim = "-",
    names = c("x", "y", "z")
  )

```

```
#> Error in `separate_wider_delim()`:
#> ! Expected 3 pieces in each element of `x`.
#> ! 2 values were too long.
#> i Use `too_many = "debug"` to diagnose the problem.
#> i Use `too_many = "drop"/"merge"` to silence this message.

x_remainder

debug <- df |>
  separate_wider_delim(
    x,
    delim = "-",
    names = c("x", "y", "z"),
    too_many = "debug"
  )
#> Warning: Debug mode activated: adding variables `x_ok`, `x_pieces`, and
#> `x_remainder`.
debug |> filter(!x_ok)
#> # A tibble: 2 x 6
#>   x     y     z   x_ok x_pieces x_remainder
#>   <chr> <chr> <chr> <lgl>   <int> <chr>
#> 1 1-3-5-6 3     5     FALSE      4 -6
#> 2 1-3-5-7-9 3     5     FALSE      5 -7-9

" "      " "

df |>
  separate_wider_delim(
    x,
    delim = "-",
    names = c("x", "y", "z"),
    too_many = "drop"
  )
#> # A tibble: 5 x 3
#>   x     y     z
#>   <chr> <chr> <chr>
#> 1 1     1     1
#> 2 1     1     2
#> 3 1     3     5
#> 4 1     3     2
#> 5 1     3     5

df |>
```

```
separate_wider_delim(
  x,
  delim = "-",
  names = c("x", "y", "z"),
  too_many = "merge"
)
#> # A tibble: 5 x 3
#>   x     y     z
#>   <chr> <chr> <chr>
#> 1 1     1     1
#> 2 1     1     2
#> 3 1     3     5-6
#> 4 1     3     2
#> 5 1     3     5-7-9
```

14.5

14.5.1

```
str_length()      :
str_length(c("a", "R for data science", NA))
#> [1] 1 18 NA
```

| | | |
|---------|----------|------|
| count() | filter() | 15 7 |
|---------|----------|------|

```
babynames |>
  count(length = str_length(name), wt = n)
#> # A tibble: 14 x 2
#>   length     n
#>   <int>    <int>
#> 1      2  338150
#> 2      3  8589596
#> 3      4  48506739
#> 4      5  87011607
#> 5      6  90749404
#> 6      7  72120767
#> # i 8 more rows
```

```

babynames |>
  filter(str_length(name) == 15) |>
  count(name, wt = n, sort = TRUE)
#> # A tibble: 34 x 2
#>   name              n
#>   <chr>            <int>
#> 1 Franciscojavier    123
#> 2 Christopherjohn    118
#> 3 Johnchristopher    118
#> 4 Christopherjame    108
#> 5 Christophermich     52
#> 6 Ryanchristopher    45
#> # i 28 more rows

```

14.5.2

| | start | end | start | end | start | end |
|-----------------------------|-------|-----|-------|-----|-------|-----|
| str_sub(string, start, end) | start | end | start | end | start | end |
| - start + 1 | | | | | | |

```

x <- c("Apple", "Banana", "Pear")
str_sub(x, 1, 3)
#> [1] "App" "Ban" "Pea"

```

-1 -2

```

str_sub(x, -3, -1)
#> [1] "ple" "ana" "ear"

```

```

str_sub()

```

```

str_sub("a", 1, 5)
#> [1] "a"

```

```

str_sub() mutate()

```

```

babynames |>
  mutate(
    first = str_sub(name, 1, 1),
    last = str_sub(name, -1, -1)
  )
#> # A tibble: 1,924,665 x 7
#>   year sex   name      n  prop first last
#>   <dbl> <chr> <chr>    <int> <dbl> <chr> <chr>

```

```
#> 1 1880 F Mary 7065 0.0724 M y
#> 2 1880 F Anna 2604 0.0267 A a
#> 3 1880 F Emma 2003 0.0205 E a
#> 4 1880 F Elizabeth 1939 0.0199 E h
#> 5 1880 F Minnie 1746 0.0179 M e
#> 6 1880 F Margaret 1578 0.0162 M t
#> # i 1,924,659 more rows
```

14.5.3

1. wt = n
2. str_length() str_sub()
3. babynames

14.6

26

14.6.1

R charToRaw()

```
charToRaw("Hadley")
#> [1] 48 61 64 6c 65 79
```

| | | |
|---|--------------------------|----------------------------------|
| 48 H 61 a | ASCII A SCII | American |
| Standard Code for Information Interchange | | |
| ISO-8859-2 | Latin1 b1 “±” Latin2 “ä” | Latin1 ISO-8859-1 Latin2 UTF-8 U |
| TF-8 | | |
| readr | UTF-8 | UTF-8 |
| | | CSV ⁸ |

```
x1 <- "text\nEl Ni\xf1o was particularly bad this year"
read_csv(x1)$text
#> [1] "El Ni\xf1o was particularly bad this year"

x2 <- "text\n\x82\xb1\x82\xf1\x82\xc9\x82\xbf\x82\xcd"
```

⁸ \x

```
read_csv(x2)$text
#> [1] "\x82\xb1\x82\xf1\x82 \xbf\x82\xcd"

locale   :
read_csv(x1, locale = locale(encoding = "Latin1"))$text
#> [1] "El Ni  o was particularly bad this year"

read_csv(x2, locale = locale(encoding = "Shift-JIS"))$text
#> [1] " "
```

readr guess_encoding()

<http://kunststube.net/encoding/>

14.6.2

```
str_length() str_sub()                  u      "
  

u <- c("\u00fc", "u\u0308")
str_view(u)
#> [1] |   
#> [2] |   

:
str_length(u)
#> [1] 2
str_sub(u, 1, 1)
#> [1] "  " "u"

==           stringr str_equal()

u[[1]] == u[[2]]
#> [1] FALSE

str_equal(u[[1]], u[[2]])
#> [1] TRUE
```

14.6.3

```

    stringr      locale
“en”   “en_GB”   “en_US”           stringi::stri_locale_list()  stringr
R          R                               stringr “en”      locale
                                         i

str_to_upper(c("i", "I"))
#> [1] "I" "I"
str_to_upper(c("i", "I"), locale = "tr")
#> [1] "İ" "I"

9      “ch”      h

str_sort(c("a", "c", "ch", "h", "z"))
#> [1] "a"  "c"  "ch" "h"  "z"
str_sort(c("a", "c", "ch", "h", "z"), locale = "cs")
#> [1] "a"  "c"  "h"  "ch" "z"

```

This also comes up when sorting strings with `dplyr::arrange()`, which is why it also has a `locale` argument.

14.7

`stringr`

Chapter 15

15.1

```
??          regular expression
" "        "regex"1 "regexp"
           stringr           stringr
tidyverse base R           " "
```

15.1.1

```
tidyverse  stringr  tidyverse  babynames
library(tidyverse)
library(babynames)
```

```
babynames      stringr
• fruit  80
• words  980
• sentences 720
```

15.2

```
str_view()
str_view()           str_view()
----- < >
1   g reg-x   g rej-x
```

```
str_view(fruit, "berry")
#> [6] | bil<berry>
#> [7] | black<berry>
#> [10] | blue<berry>
#> [11] | boysen<berry>
#> [19] | cloud<berry>
#> [21] | cran<berry>
#> ... and 8 more
```

| | | | |
|---|--------------------|-------------|--------------------------------|
| | literal characters | . + * [] ? | ²
metacharacters |
| . | 3 "a."
"a" | | |

```
str_view(c("a", "ab", "ae", "bd", "ea", "eab"), "a.")
#> [2] | <ab>
#> [3] | <ae>
#> [6] | e<ab>
```

“a” “e”

```
str_view(fruit, "a...e")
#> [1] | <apple>
#> [7] | bl<ackbe>rry
#> [48] | mand<arine>
#> [51] | nect<arine>
#> [62] | pine<apple>
#> [64] | pomegr<anate>
#> ... and 2 more
```

Quantifiers

- ? 0 1
- +
- * 0

```
# ab? matches an "a", optionally followed by a "b".
str_view(c("a", "ab", "abb"), "ab?")
#> [1] | <a>
#> [2] | <ab>
#> [3] | <ab>b
```

² ??
³ \n

```
# ab+ matches an "a", followed by at least one "b".
str_view(c("a", "ab", "abb"), "ab+")
#> [2] | <ab>
#> [3] | <abb>

# ab* matches an "a", followed by any number of "b"s.
str_view(c("a", "ab", "abb"), "ab*")
#> [1] | <a>
#> [2] | <ab>
#> [3] | <abb>
```

Character classes [] [abcd] “a” “b” “c” “d” ^ [^abcd] “a” “b” “c” “d”
“x” “y”

```
str_view(words, "[aeiou]x[aeiou]")
#> [284] | <exa>ct
#> [285] | <exa>mple
#> [288] | <exe>rcise
#> [289] | <exi>st
str_view(words, "[^aeiou]y[^aeiou]")
#> [836] | <sys>tem
#> [901] | <typ>e
```

alternation | “apple” “melon” “nut”

```
str_view(fruit, "apple|melon|nut")
#> [1] | <apple>
#> [13] | canary <melon>
#> [20] | coco<nut>
#> [52] | <nut>
#> [62] | pine<apple>
#> [72] | rock <melon>
#> ... and 1 more
str_view(fruit, "aa|ee|ii|oo|uu")
#> [9] | bl<oo>d orange
#> [33] | g<oo>seberry
#> [47] | lych<ee>
#> [66] | purple mangost<ee>n
```

stringr

15.3

stringr tidyR

15.3.1

```
str_detect()          TRUE FALSE
```

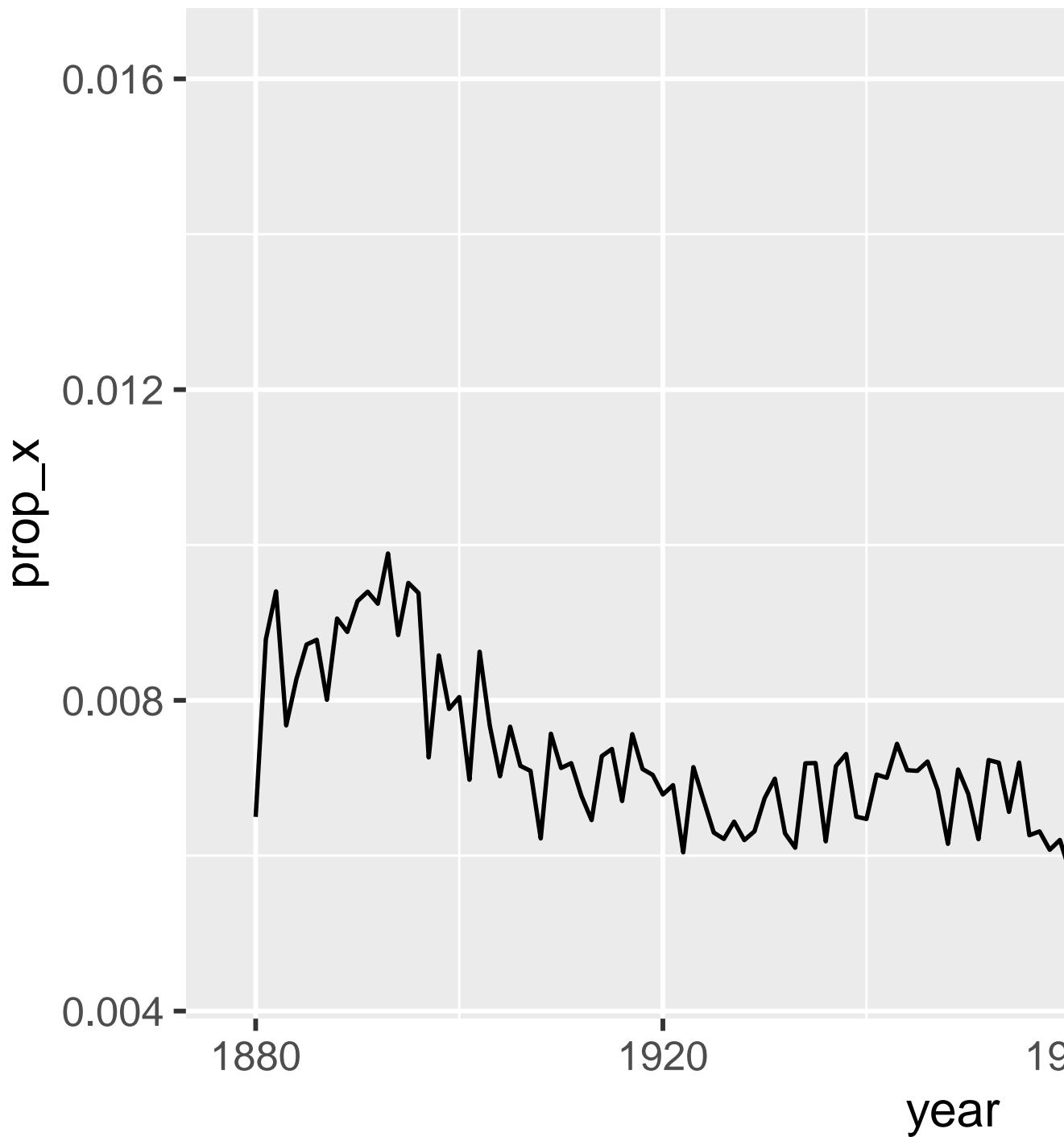
```
str_detect(c("a", "b", "c"), "[aeiou]")
#> [1] TRUE FALSE FALSE
```

```
str_detect()          filter()        "x"
```

```
babynames |>
  filter(str_detect(name, "x")) |>
  count(name, wt = n, sort = TRUE)
#> # A tibble: 974 x 2
#>   name      n
#>   <chr>    <int>
#> 1 Alexander 665492
#> 2 Alexis    399551
#> 3 Alex       278705
#> 4 Alexandra 232223
#> 5 Max        148787
#> 6 Alexa     123032
#> # i 968 more rows
```

```
str_detect() summarize()  sum() mean() s      um(str_detect(x,
pattern))      mean(str_detect(x, pattern))
"X"                                     4
```

```
babynames |>
  group_by(year) |>
  summarize(prop_x = mean(str_detect(name, "x"))) |>
  ggplot(aes(x = year, y = prop_x)) +
  geom_line()
```



15.3.2

str_detect() str_count()

```
x <- c("apple", "banana", "pear")
str_count(x, "p")
#> [1] 2 0 1
```

"abababa" "aba"

```
str_count("abababa", "aba")
#> [1] 2
str_view("abababa", "aba")
#> [1] | <aba>b<aba>
```

```
str_count() mutate() str_count()
```

```
babynames |>
  count(name) |>
  mutate(
    vowels = str_count(name, "[aeiou]"),
    consonants = str_count(name, "[^aeiou]"))
  )
#> # A tibble: 97,310 x 4
#>   name          n vowels consonants
#>   <chr>     <int>  <int>      <int>
#> 1 Aaban        10      2          3
#> 2 Aabha         5      2          3
#> 3 Aabid         2      2          3
#> 4 Aabir         1      2          3
#> 5 Aabriella     5      4          5
#> 6 Aada          1      2          2
#> # i 97,304 more rows
```

“Aaban” “a”

- `str_count(name, "[aeiouAEIOU] ")`
- `str_count(name, regex("[aeiou]", ignore_case = TRUE))`
??

- `str_to_lower()`
- ```
str_count(str_to_lower(name), "[aeiou]")
```
- 

```
babynames |>
 count(name) |>
 mutate(
 name = str_to_lower(name),
 vowels = str_count(name, "[aeiou]"),
 consonants = str_count(name, "[^aeiou]")
)
#> # A tibble: 97,310 x 4
#> name n vowels consonants
#> <chr> <int> <int> <int>
#> 1 aaban 10 3 2
#> 2 aabha 5 3 2
#> 3 aabid 2 3 2
#> 4 aabir 1 3 2
#> 5aabriella 5 5 4
#> 6 aada 1 3 1
#> # i 97,304 more rows
```

### 15.3.3

<code>str_replace()</code>	<code>str_replace_all()</code>	<code>str_replace()</code>	<code>str_replace_all()</code>
----------------------------	--------------------------------	----------------------------	--------------------------------

```
x <- c("apple", "pear", "banana")
str_replace_all(x, "[aeiou]", "-")
#> [1] "-ppl-" "p--r" "b-n-n-"
```

`str_remove()` and `str_remove_all()` are handy shortcuts for `str_replace(x, pattern, "")`:

```
x <- c("apple", "pear", "banana")
str_remove_all(x, "[aeiou]")
#> [1] "ppl" "pr" "bnn"
```

```
mutate()
```

### 15.3.4

```

 separate_wider_regex() ?? separate_wider_position() separate_wider_delim()
tidyR 5

babynames

df <- tribble(
 ~str,
 "<Sheryl>-F_34",
 "<Kisha>-F_45",
 "<Brandon>-N_33",
 "<Sharon>-F_38",
 "<Penny>-F_58",
 "<Justin>-M_41",
 "<Patricia>-F_84",
)
separate_wider_regex()

df |>
 separate_wider_regex(
 str,
 patterns = c(
 "<",
 name = "[A-Za-z]+",
 ">-",
 gender = ".",
 "_",
 age = "[0-9]+"
)
)
#> # A tibble: 7 x 3
#> name gender age
#> <chr> <chr> <chr>
#> 1 Sheryl F 34
#> 2 Kisha F 45
#> 3 Brandon N 33
#> 4 Sharon F 38
#> 5 Penny F 58
#> 6 Justin M 41
#> # i 1 more row
too_few = "debug" separate_wider_delim() separate_wider_position()

```

---

5

### 15.3.5

- 1.
2. "a/b/c/d/e" / \
3. str\_replace\_all() str\_to\_lower()
- 4.

## 15.4

stringr tidyR anchors quantifiers grouping	character operator precedence	escaping classes
-----------------------------------------------------	----------------------------------	---------------------

### 15.4.1

```

. 6 . \. \
\.. "\\".."

To create the regular expression \., we need to use \\..
dot <- "\\.."

But the expression itself only contains one \
str_view(dot)
#> [1] | \.

And this tells R to look for an explicit .
str_view(c("abc", "a.c", "bef"), "a\\\.c")
#> [2] | <a.c>

\.. "\\".."
\.. \.. \\.. \.. \.. "\\\\.."—
—
```

```

x <- "a\\b"
str_view(x)
#> [1] | a\b
str_view(x, "\\\\..")
#> [1] | a<\>b

```

---

6 .~\$\\|\*+?{}[]()

??

```
str_view(x, r"{{\\}}")
#> [1] | a<\>b

. $ | * + ? { } () . $ | ...

str_view(c("abc", "a.c", "a*c", "a c"), "a[.]c")
#> [2] | <a.c>
str_view(c("abc", "a.c", "a*c", "a c"), ".[*]c")
#> [3] | <a*c>
```

### 15.4.2

^ \$

```
str_view(fruit, "^a")
#> [1] | <a>pple
#> [2] | <a>pricot
#> [3] | <a>ocado
str_view(fruit, "a$")
#> [4] | banan<a>
#> [15] | cherimoy<a>
#> [30] | feijo<a>
#> [36] | guav<a>
#> [56] | papay<a>
#> [74] | satsum<a>
```

\$

^ \$

```
str_view(fruit, "apple")
#> [1] | <apple>
#> [62] | pine<apple>
str_view(fruit, "^apple$")
#> [1] | <apple>
```

\b

RStudio

sum()

\bsum\b

summarize summary rowsum

```
x <- c("summary(x)", "summarize(df)", "rowsum(x)", "sum(x)")
str_view(x, "sum")
#> [1] | <sum>mary(x)
#> [2] | <sum>marize(df)
```

```
#> [3] | row<sum>(x)
#> [4] | <sum>(x)
str_view(x, "\\bsum\\b")
#> [4] | <sum>(x)
```

```
str_view("abc", c("$", "^", "\\b"))
#> [1] | abc<>
#> [2] | <>abc
#> [3] | <>abc<>
```

```
str_replace_all("abc", c("$", "^", "\\b"), "--")
#> [1] "abc--" "--abc" "--abc--"
```

### 15.4.3

$\cdot$ $\cdot$ $\cdot$	$[]$ $^ []$ $- [a-z]$ $\backslash [\wedge \backslash - \backslash]$	$[a]$ $[0-9]$ $^ - ]$	$[abc]$ $[^abc]$ $"a" "b" "c"$ $"a" "b" "c"$
-------------------------------	------------------------------------------------------------------------------	-----------------------------	-------------------------------------------------------

```
x <- "abcd ABCD 12345 -!@#%. "
str_view(x, "[abc]+")
#> [1] | <abc>d ABCD 12345 -!@#%.
str_view(x, "[a-z]+")
#> [1] | <abcd> ABCD 12345 -!@#%.
str_view(x, "[^a-z0-9]+")
#> [1] | abcd< ABCD >12345< -!@#%.>

You need an escape to match characters that are otherwise
special inside of []
str_view("a-b-c", "[a-c]")
#> [1] | <a>--<c>
str_view("a-b-c", "[a\\-c]")
#> [1] | <a><->b<-><c>
```