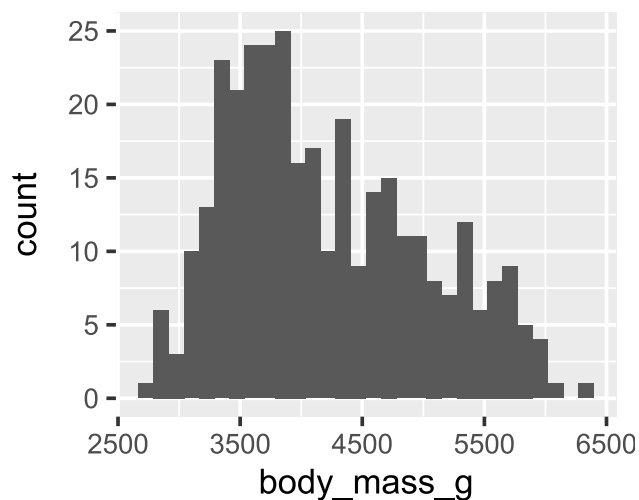


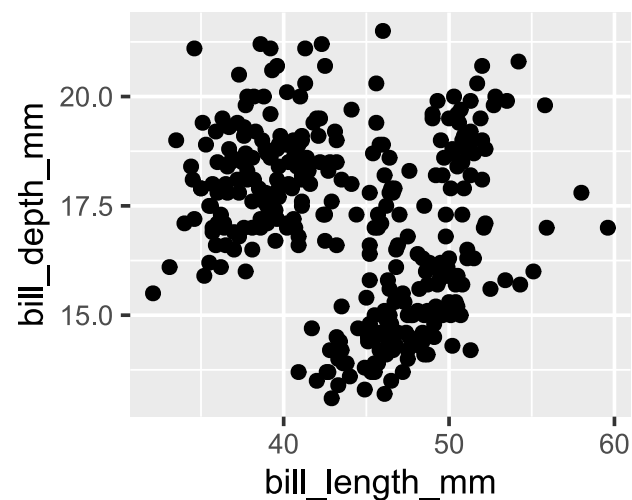
Main geoms and their application

П'ять основних геометрій для опису даних

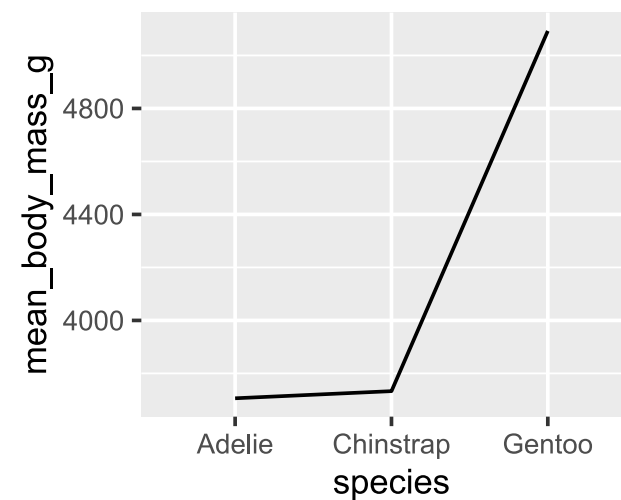
Histogram: geom_histogram



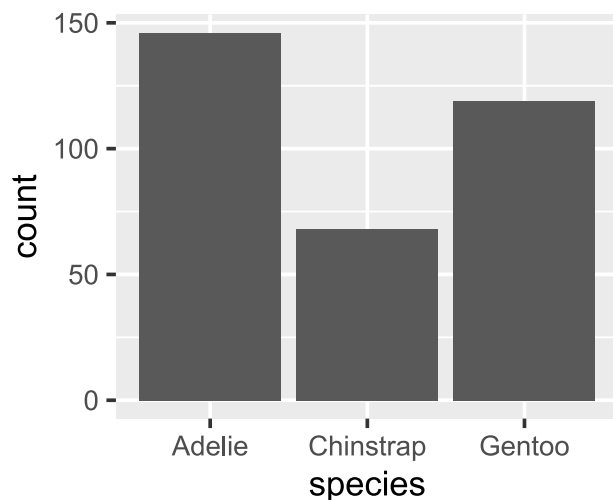
Scatterplot: geom_point



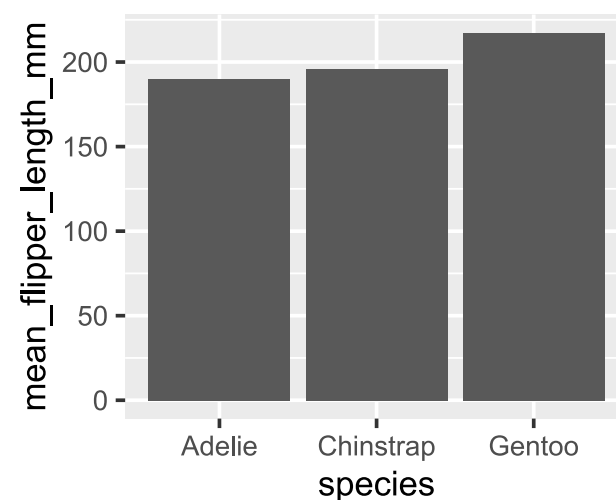
Linechart: geom_line



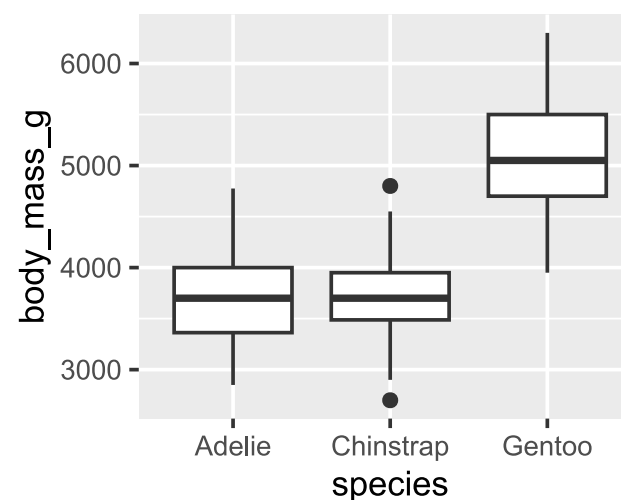
Barplot (quantity): geom_bar



Barplot (quality): geom_col



Boxplot: geom_boxplot

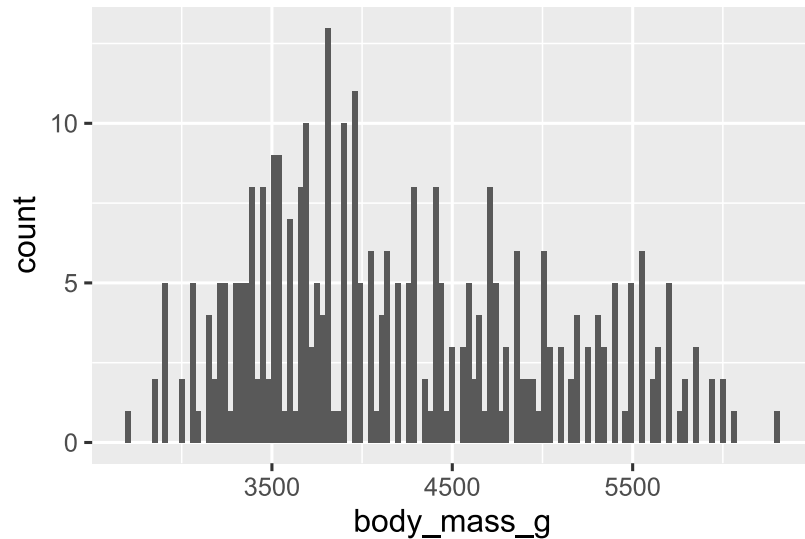


Histograms

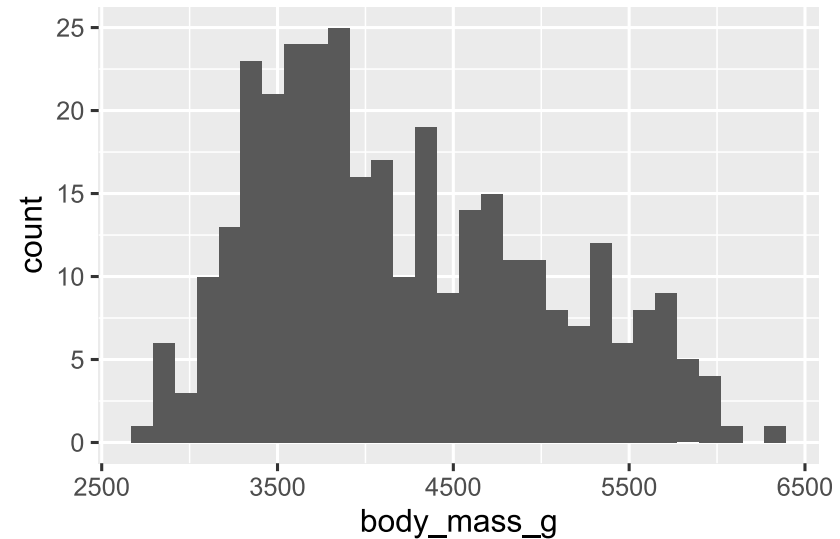
Використовують для: демонстрації розподілу *одної неперервної чисельної змінної*

Двома важливими аргументами функції `geom_histogram()` є `binwidth` та `bins` які дозволяють задати ширину біну (ящику) гістограми та їх кількість відповідно (використовується або той або інший аргумент, але не обидва!)

```
1 penguins |>  
2   ggplot(aes(body_mass_g)) +  
3   geom_histogram(binwidth = 30)
```



```
1 penguins |>  
2   ggplot(aes(body_mass_g)) +  
3   geom_histogram(bins = 30)
```



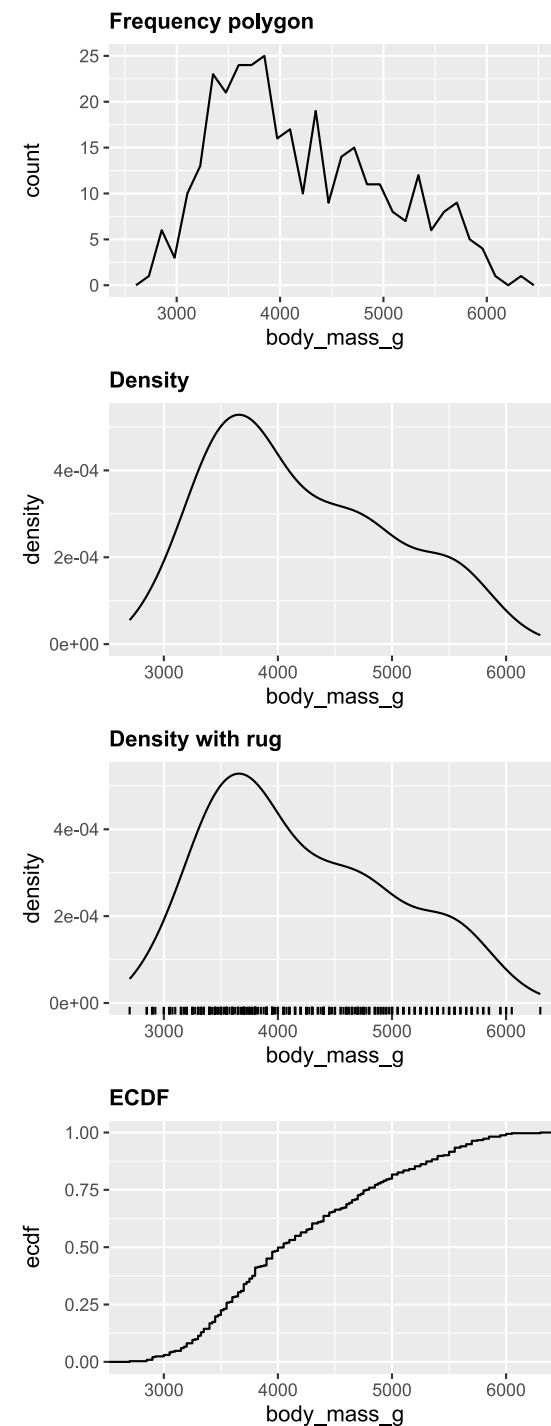
Маніпулювати шириною бінів також можливо через аргумент `breaks`, що дозволяє задати проміжок від і до, завдяки чому можливо наприклад отримати гістограму з бінами різної ширини

Histogram-related

Спорідненими до гістограм є:

- Frequency polygons з `geom_freqpoly()` — повний аналог гістограми, що використовує іншу геометрію для візуального відображення виконаної статистичної трансформації
- KDE з `geom_density()` або `stat_density()` — ядрова оцінка густини розподілу, “гладенька” версія гістограми. Підходить для відображення чисельних значень, які походять з *неперервного* розподілу. Графік густини розподілу часто можна побачити комбінованим з килимковим графіком (rug plot)
- ECDF з `stat_ecdf()` — розрахована функція кумулятивної щільності, альтернативний варіант зображення розподілу, на відміну від інших трьох також сприймає категоріальні змінні

`geom_histogram()` та `geom_freqpoly()` використовують `stat_bin()` для статистичної трансформації даних



Barplots

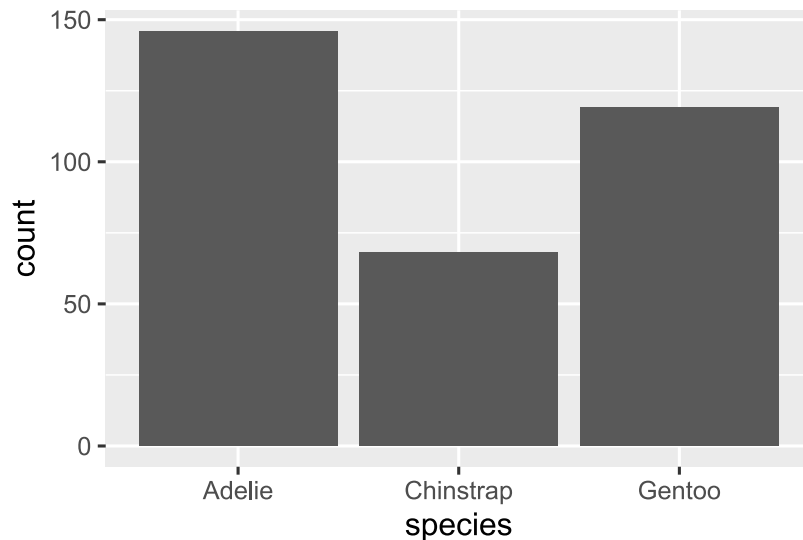
Використовують для: демонстрації відношень рівнів категоріальної змінної до сукупної статистичної оцінки чисельної змінної, демонстрації розподілу рівнів категоріальної змінної

Дефолтна поведінка `geom_bar()` є подібною до `geom_histogram()`, функція приймає специфікацію одної осі для категоріальної змінної, і внутрішньо викликає `stat_count()`

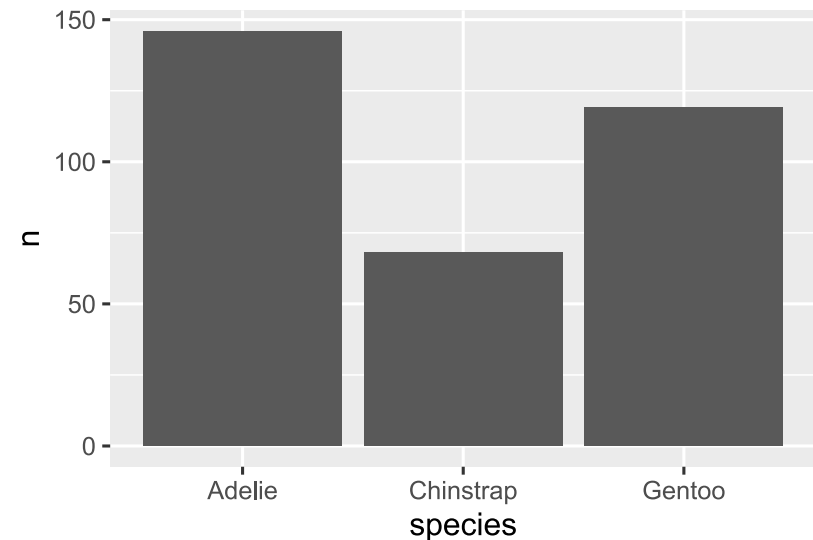
Функція `geom_col()` натомість приймає специфікацію для обох осей — категоріальної та чисельної. Кожному рівню категоріальної змінної має відповідати одне єдине чисельне значення.

```
1 penguins_count <- penguins |> group_by(species) |> count()
```

```
1 penguins |>  
2   ggplot(aes(species)) +  
3   geom_bar()
```



```
1 penguins_count |>  
2   ggplot(aes(species, n)) +  
3   geom_col()
```

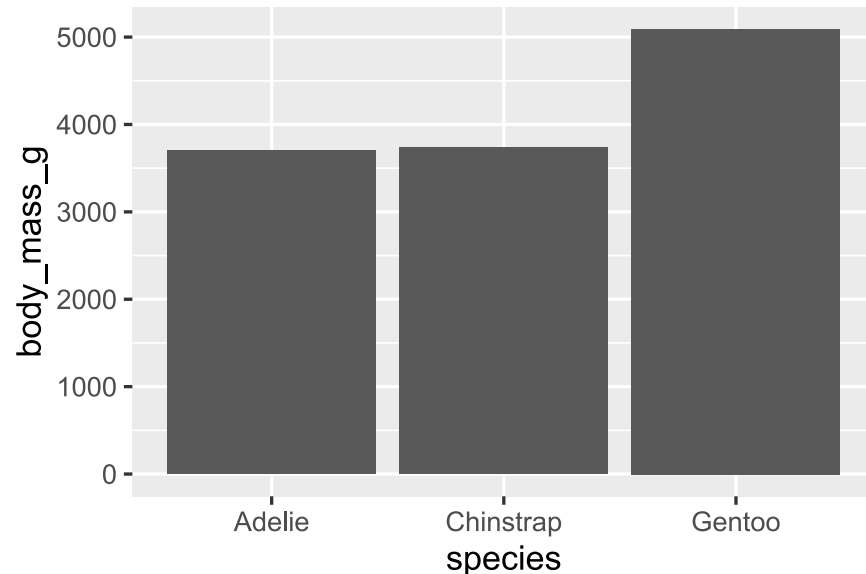


Barplots

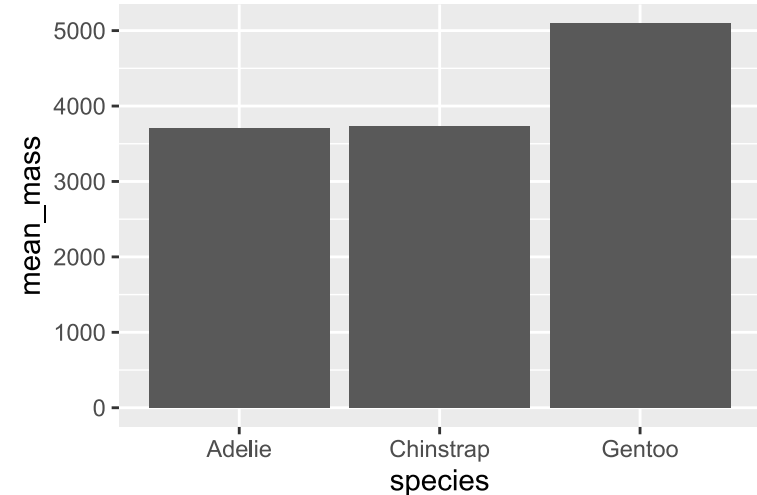
Обравши інше значення аргументу `stat` функцію `geom_bar()` можливо змусити приймати специфікацію для оХ та оУ одночасно. При `stat = "identity"` функція буде поводити себе як `geom_col()`, проте можливо використання і інших трансформації, зокрема `stat = "summary"` дозволяє розрахувати певний статистичний підсумок безпосередньо при створенні графіку

```
1 penguins_mass <- penguins |> group_by(species) |> summarise(mean_mass = mean(body_mass_g))
```

```
1 penguins |>  
2   ggplot(aes(species, body_mass_g)) +  
3   geom_bar(stat = "summary", fun = "mean")
```



```
1 penguins_mass |>  
2   ggplot(aes(species, mean_mass)) +  
3   geom_col()
```

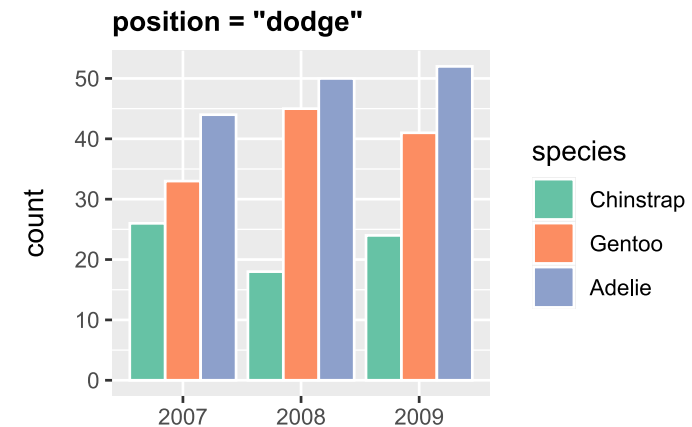
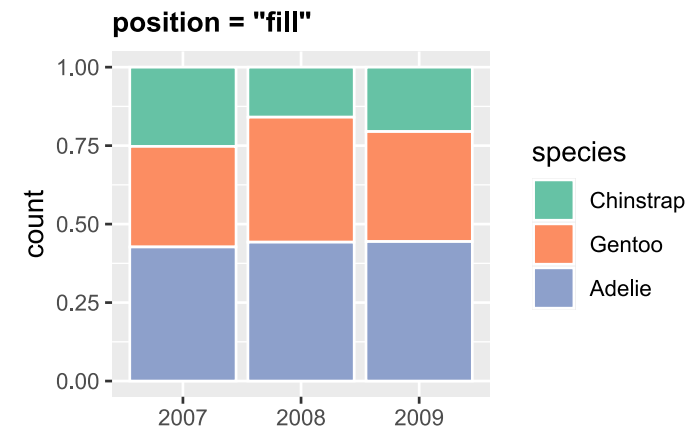
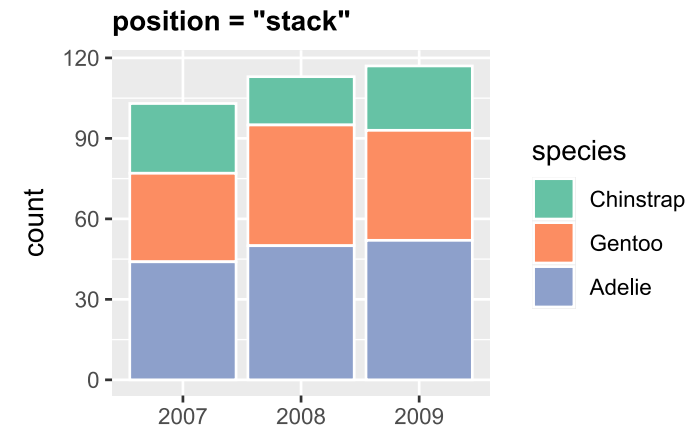


Параметр позиції

Естетичний параметр `position` визначає розташування окремих елементів геометрій по відношенню один до одного. Найчастіше використовується з барплотами та гістограмами, але може застосовуватися і до інших геометрій.

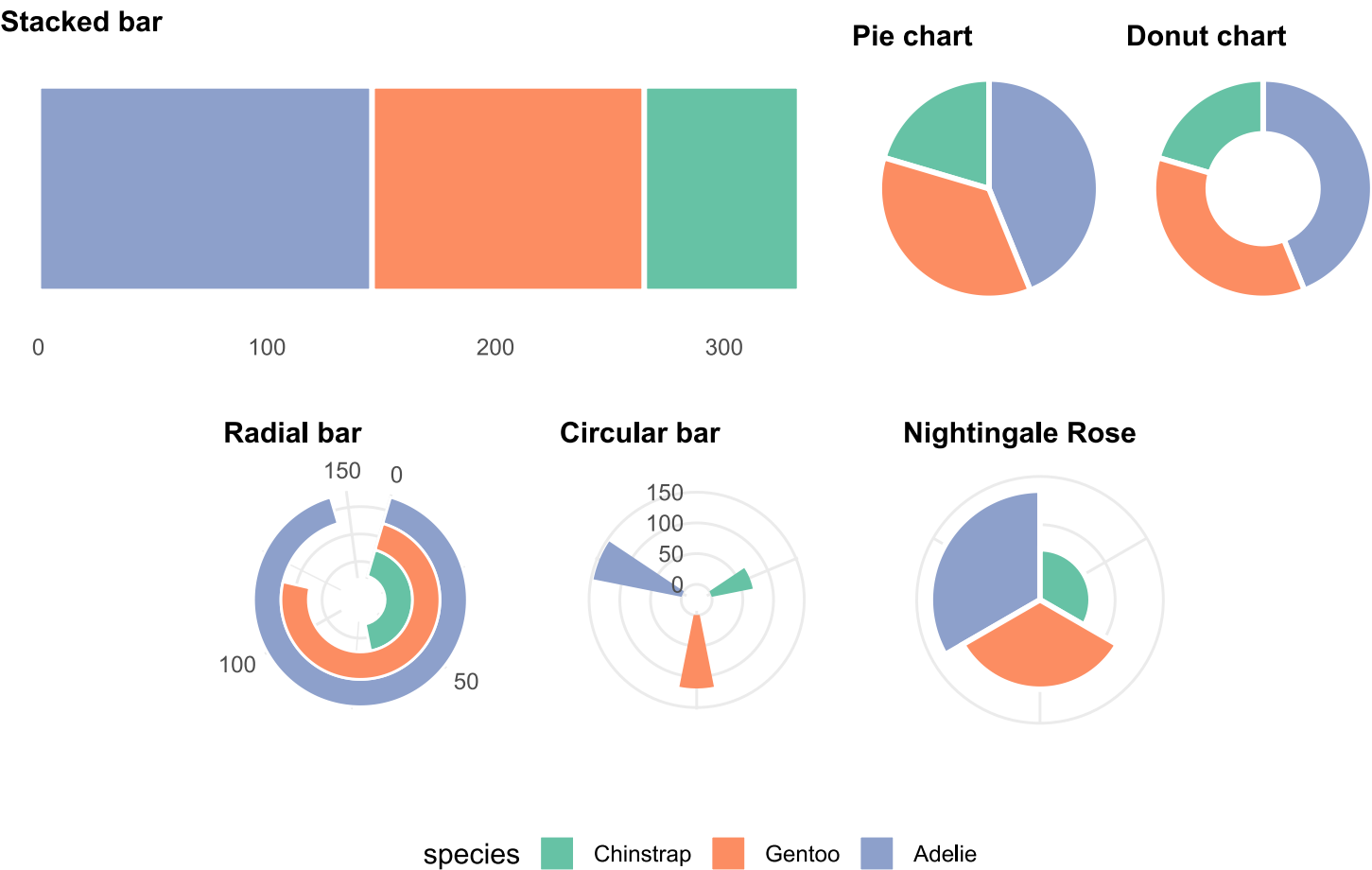
Позиція може бути задана через вказання її назви, е.г. `position = "dodge"` або через виклик відповідної функції, е.г. `position = position_dodge()`. Окрім представлених справа варіантів також є варіанти `identity`, `dodge2`, `jitter`, `jitterdodge` та `nudge`.

Варіант `jitter` додає рандомний шум до координат розташування геометрії, варіант `nudge` дозволяє мануально зсунути геометрію на певну фіксовану дистанцію.



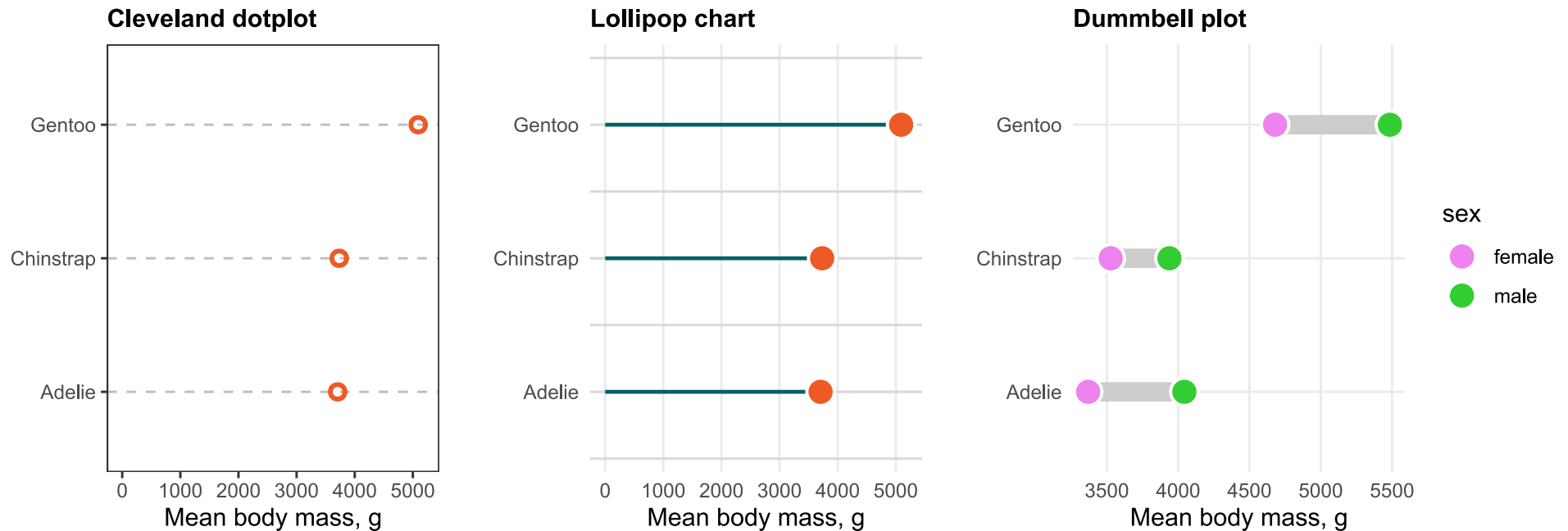
Barplot-related

При зміні евклідової системи координат на полярну або радіальну можливо отримати різноманітні циркулярні діаграми. Пайчарт та донатчарт є повними аналогами стеку і слугують для відображення відношення частини до цілого. Радіальний, циркулярний та чарт Розе можуть, натомість, відображати будь який статистичний підсумок



Barplot-related

Іншими, візуально “легшими”, альтернативами барплотам є дотплот Клівленда та “льодяниковий” чарт, що у `ggplot2` використовують `geom_point()` та `geom_segment()` або `geom_linerange()`. Підвидом даних графіків також є так званий dumbbell-графік, що використовується для демонстрації певних логічних пар значень



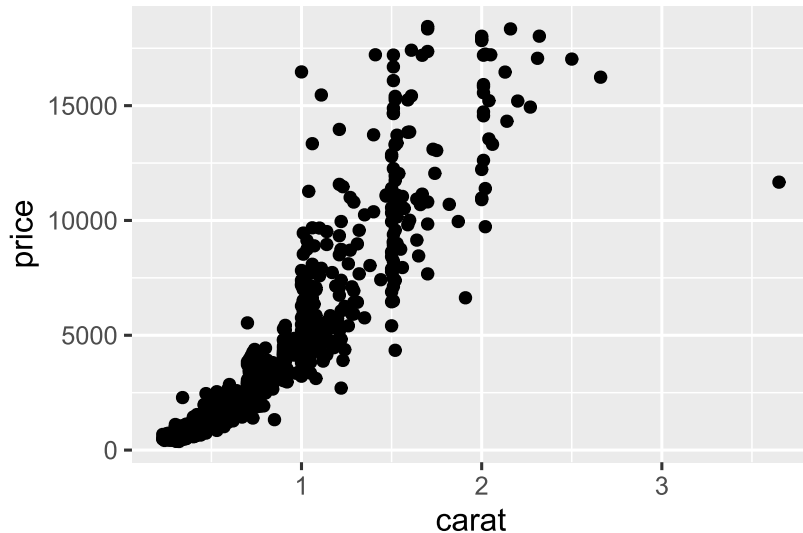
Scatterplots

Використовують для: демонстрації відносин між двома (зазвичай) неперервними чисельними змінними

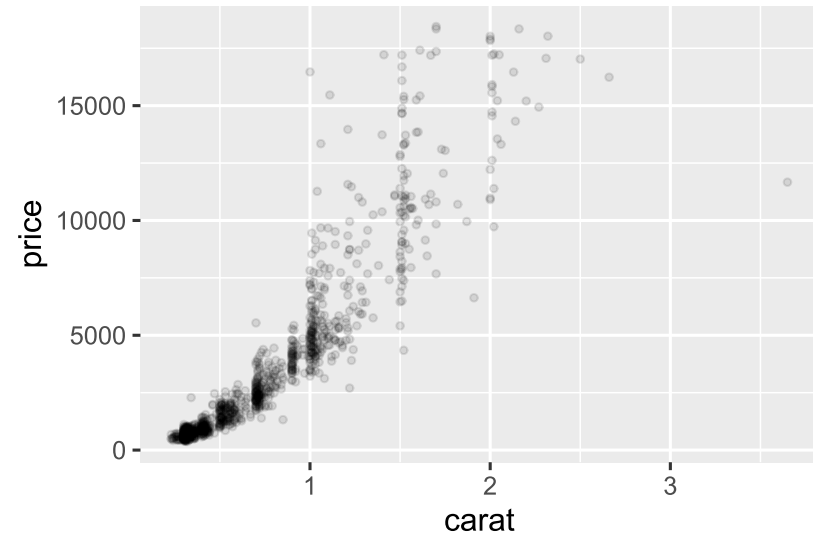
Основна проблема при використанні графіків розсіяння є так званий overplotting — перекриття значень на графіку один одним. Аби уникнути оверплотингу можливо вдається до модифікації параметру форми, прозорості або позиції

```
1 set.seed(18475)
2 dsmall <- sample_n(diamonds, 1000)
```

```
1 dsmall |>
2   ggplot(aes(carat, price)) +
3   geom_point()
```

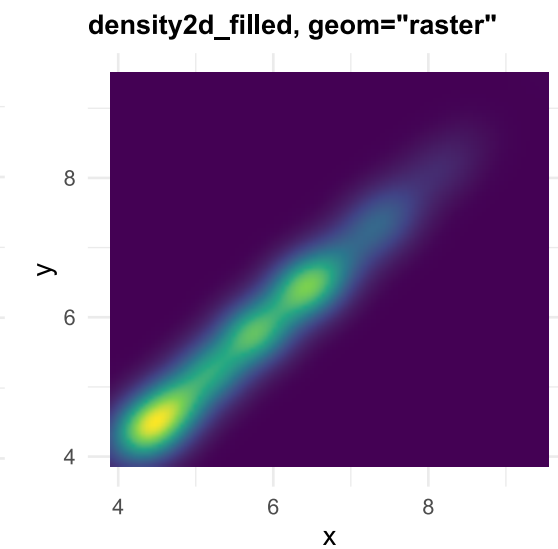
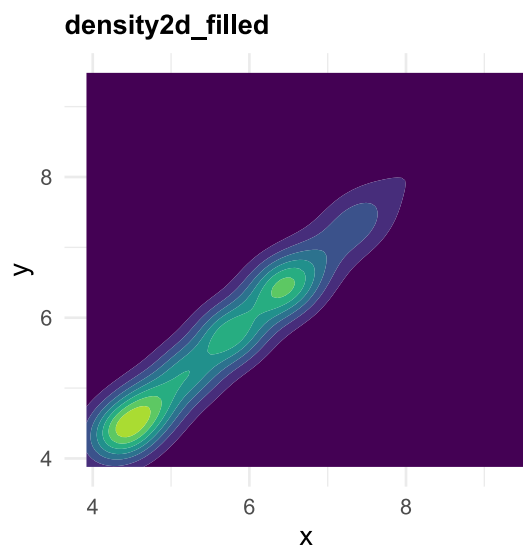
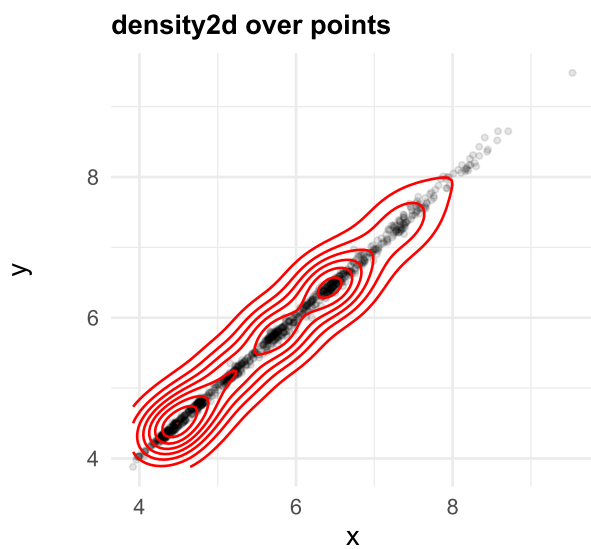
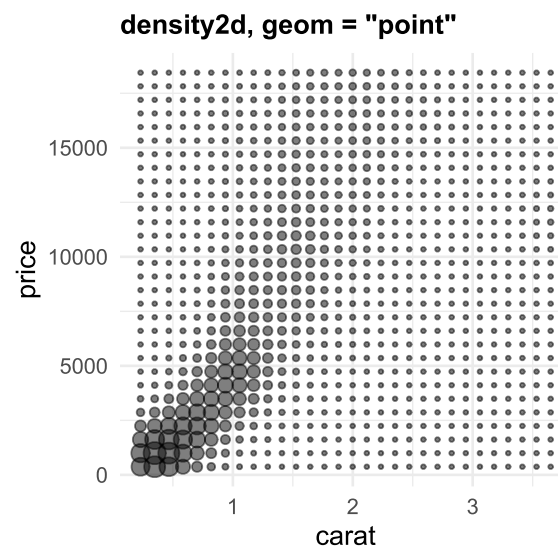
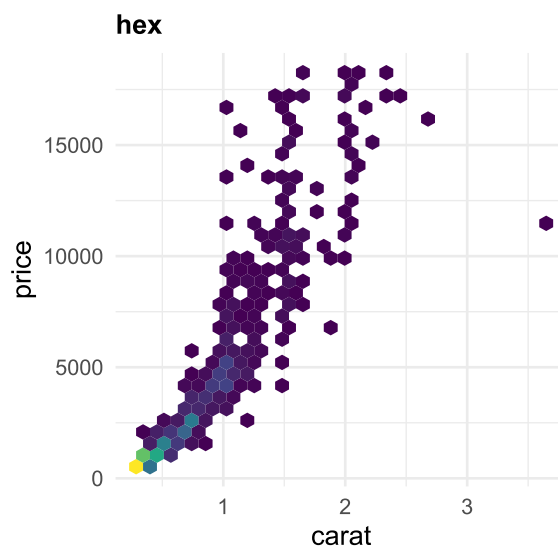
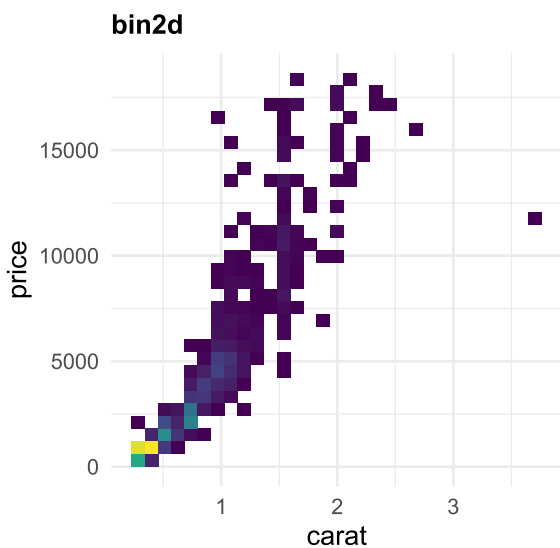


```
1 dsmall |>
2   ggplot(aes(carat, price)) +
3   geom_point(shape = 20, alpha = .1)
```



Scatterplot-related

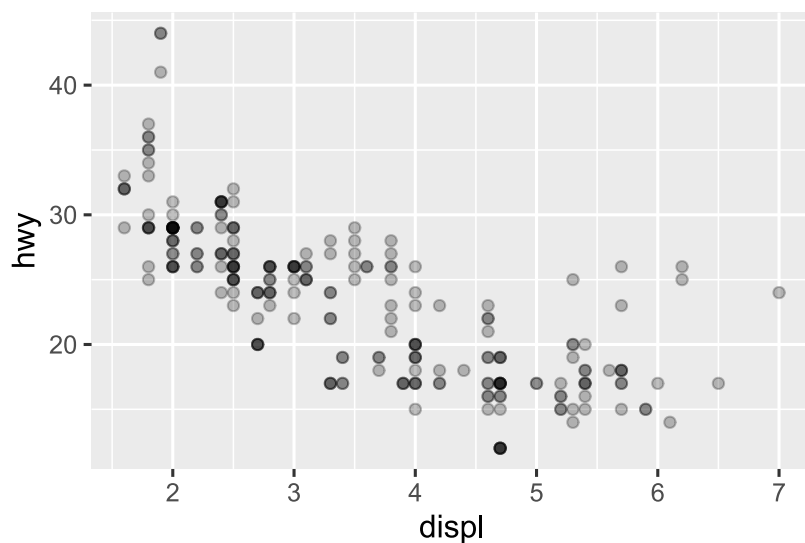
Також для уникнення оверплотингу можливо вдається до використання 2d бінів/гістограм



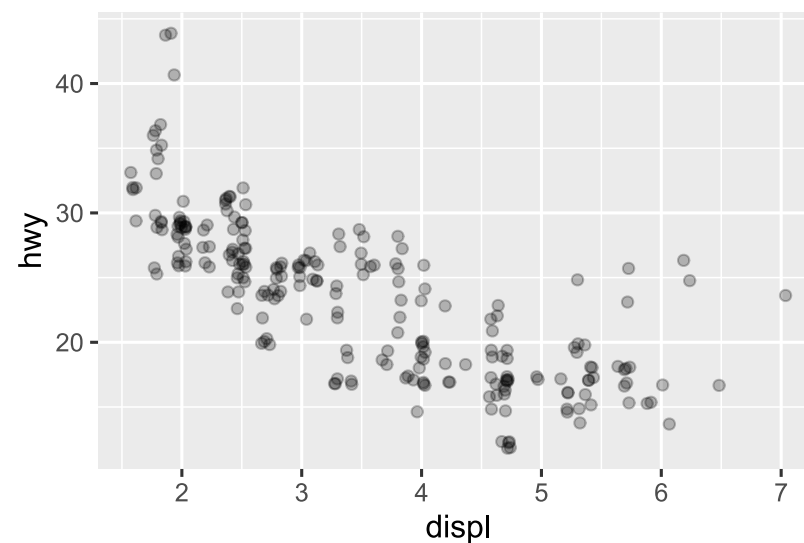
Jitterplot

Джитеринг — додавання випадкових шумів малих значень до координат поїнтів на графіку, також один із способів боротьби з оверплотингом. Дозволяє підвищити читабельність графіків у випадку коли значна частина даних має близько-ідентичні значення, проте одночасно знижує точність відображення цих даних на графіку

```
1 mpg |>  
2   ggplot(aes(displ, hwy)) +  
3   geom_point(alpha = .25)
```



```
1 mpg |>  
2   ggplot(aes(displ, hwy)) +  
3   geom_jitter(alpha = .25)
```

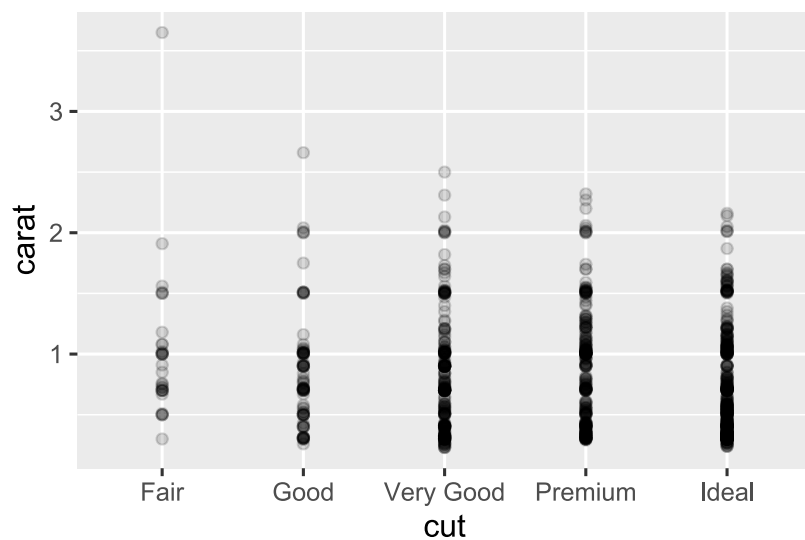


Того ж самого можливо досягти через вказання `position = position_jitter()` у `geom_point()`

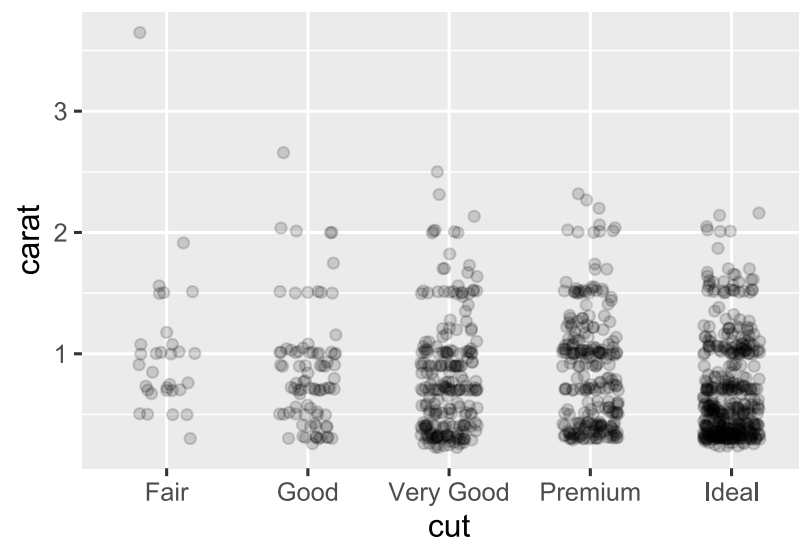
Jitterplot

Якщо одна з осей мапована на дискретні або категоріальні дані, то джитеринг дозволяє створювати т.к. стріпчарти, що можуть виступати альтернативою або доповненням до боксплотів

```
1 dsmall |>  
2   ggplot(aes(cut, carat)) +  
3   geom_point(alpha = .15)
```



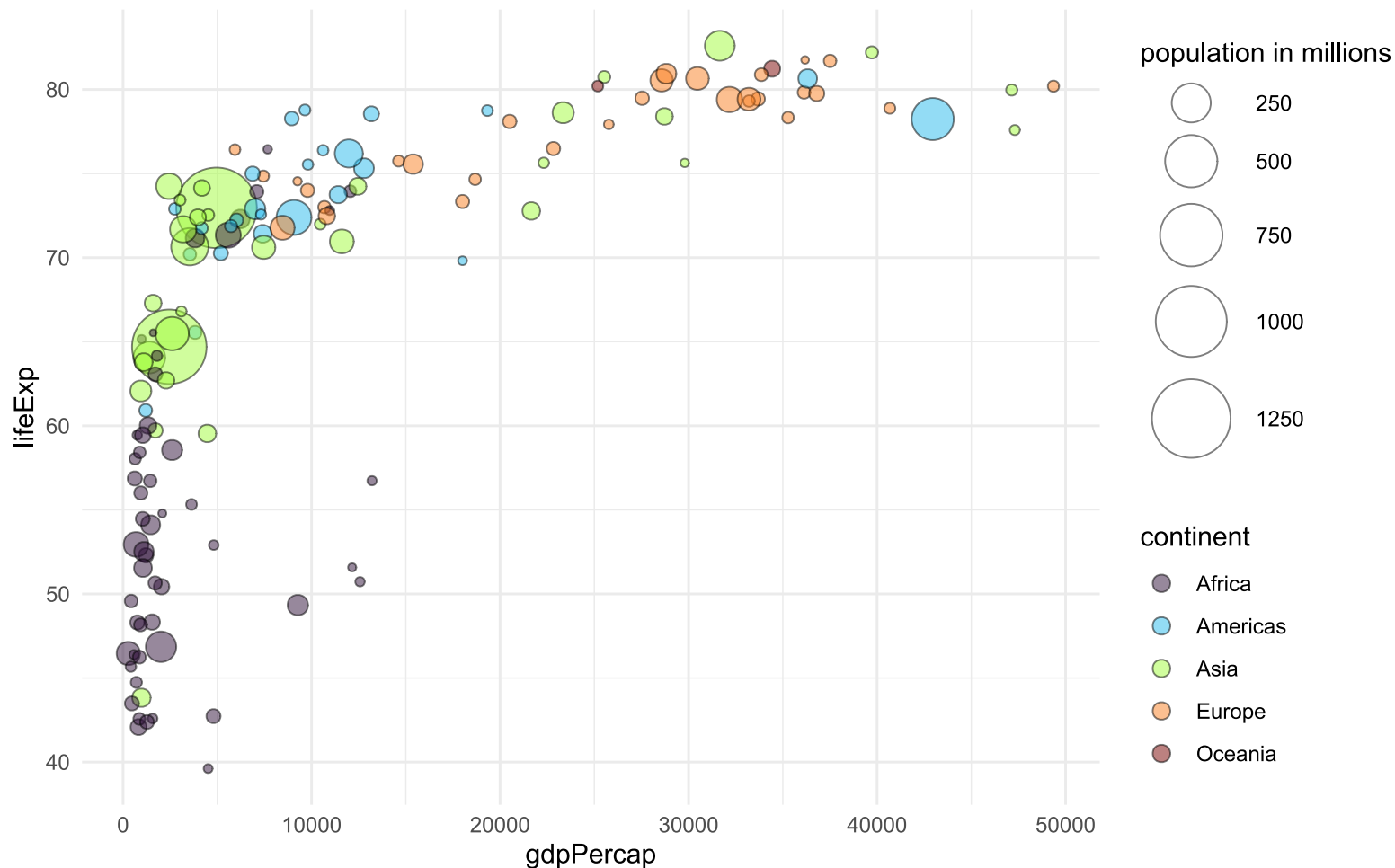
```
1 dsmall |>  
2   ggplot(aes(cut, carat)) +  
3   geom_jitter(alpha = .15, width = .2)
```



Bubblechart

Баблчарт є варіантом графіка розсіяння у якому третя чисельна змінна є мапованою на естетику розміру поїнта. Співвідношення між розмірами поїнтів додатково може бути контрольоване через шар `scale_size()`

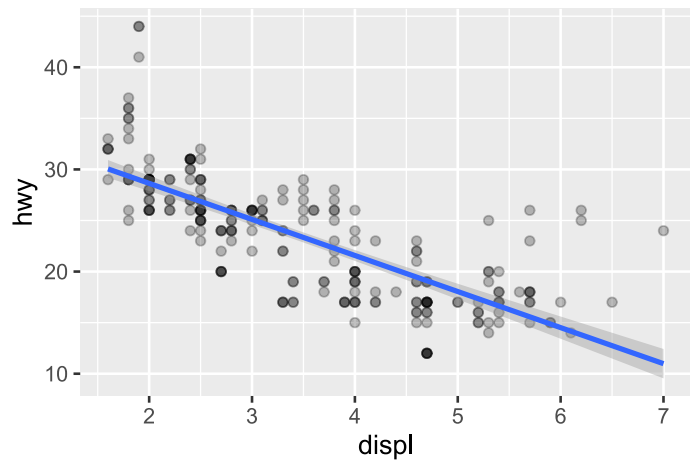
На графіку нижче популяція країни є мапованою на розмір поїнту, e.g. `geom_point(aes(size = pop))`



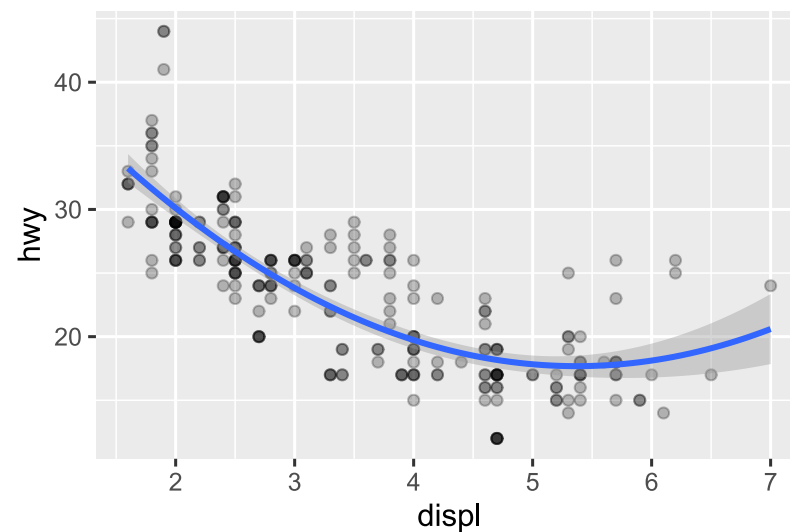
Curve fit over points

Функція `geom/stat_smooth()` або `geom_line(stat = "smooth")` дозволяє продемонструвати наявні тренди у даних за допомогою неперервної математичної функції. Окрім лінійної моделі з методів також є доступними генералізована лінійна модель (`glm`), генералізована адитивна модель (`gam`) та локальна поліноміальна регресія (`loess`)

```
1 mpg |>  
2   ggplot(aes(displ, hwy)) +  
3   geom_point(alpha = .25) +  
4   geom_smooth(method = lm)
```



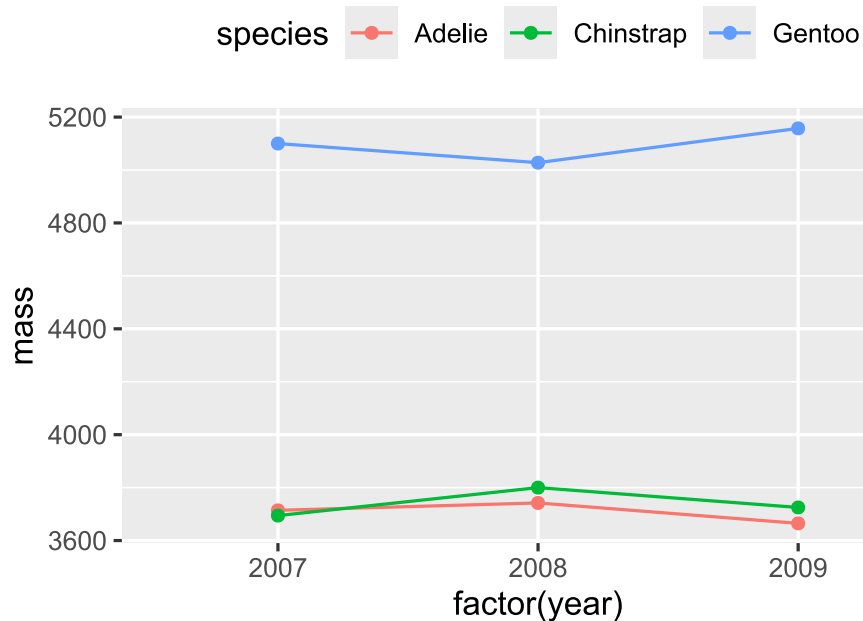
```
1 mpg |>  
2   ggplot(aes(displ, hwy)) +  
3   geom_point(alpha = .25) +  
4   geom_smooth(method = lm, formula = y ~ poly(x, 2))
```



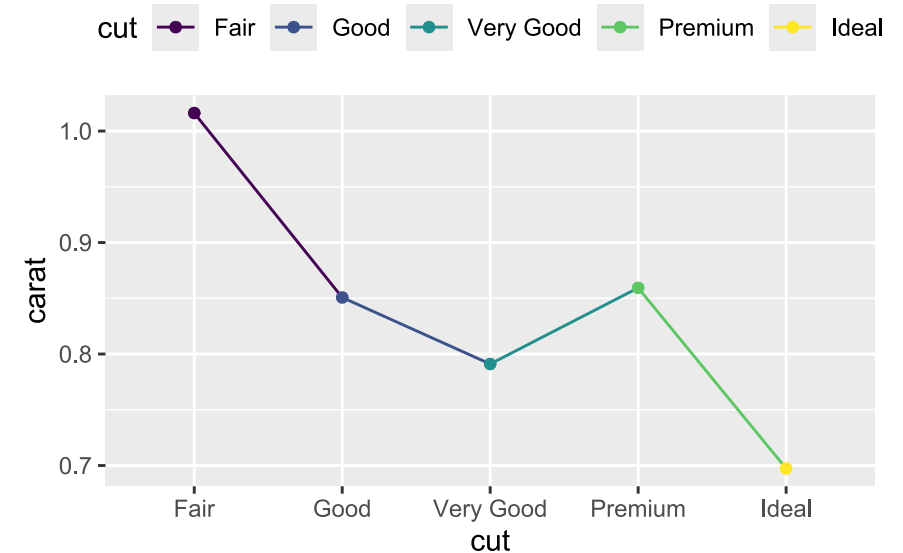
Linecharts

Використовують для: демонстрації часових рядів та наявних трендів у даних

```
1 penguins |> group_by(species, year) |>
2   summarise(mass = mean(body_mass_g)) |>
3   ggplot(aes(factor(year), mass,
4             color = species)) +
5   geom_line(aes(group = species)) +
6   geom_point() +
7   theme(legend.position = "top")
```




```
1 dsmall |> group_by(cut) |>
2   summarise(carat = mean(carat)) |>
3   ggplot(aes(cut, carat,
4             color = cut)) +
5   geom_line(aes(group = 1)) +
6   geom_point() +
7   theme(legend.position = "top")
```

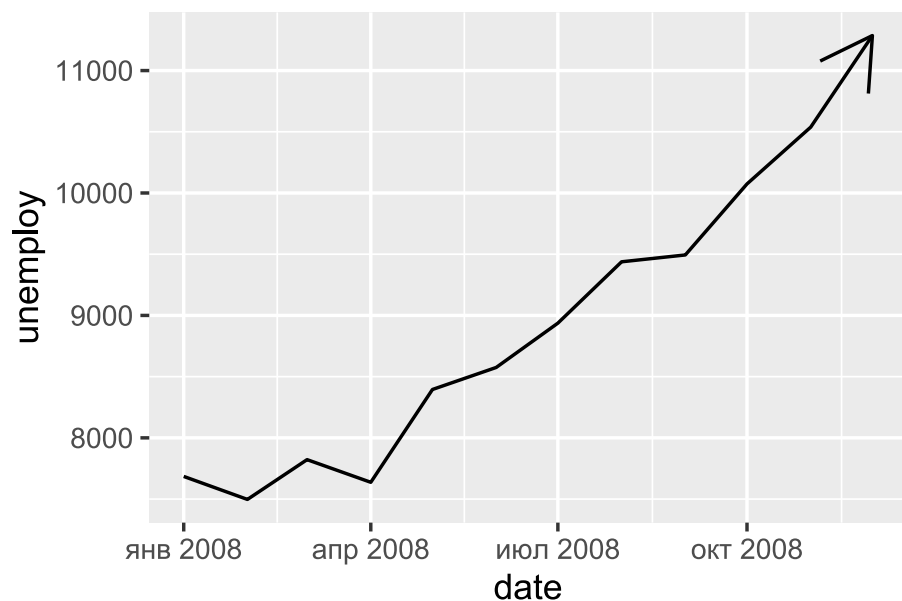



Linecharts

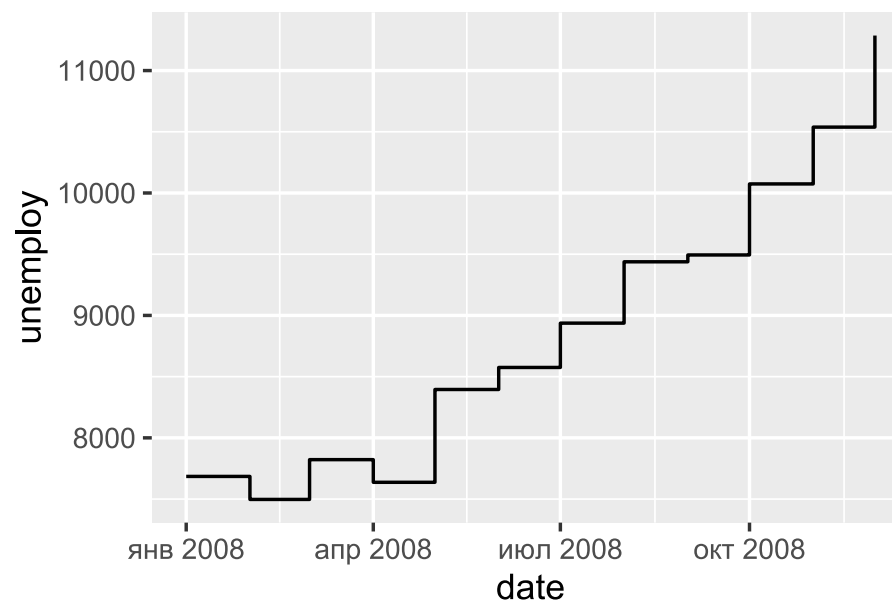
Окрім `geom_line()` також є `geom_step()`, яка малює ступінчатий графік та `geom_path()`, яка поєднує спостереження у тому порядку, у якому вони є представлені у даних

```
1 econ <- economics |> filter(between(date, as.Date("2008-01-01"), as.Date("2008-12-01"))) 
```

```
1 econ |>  
2   ggplot(aes(date, unemployment)) +  
3   geom_line(arrow = arrow()) 
```



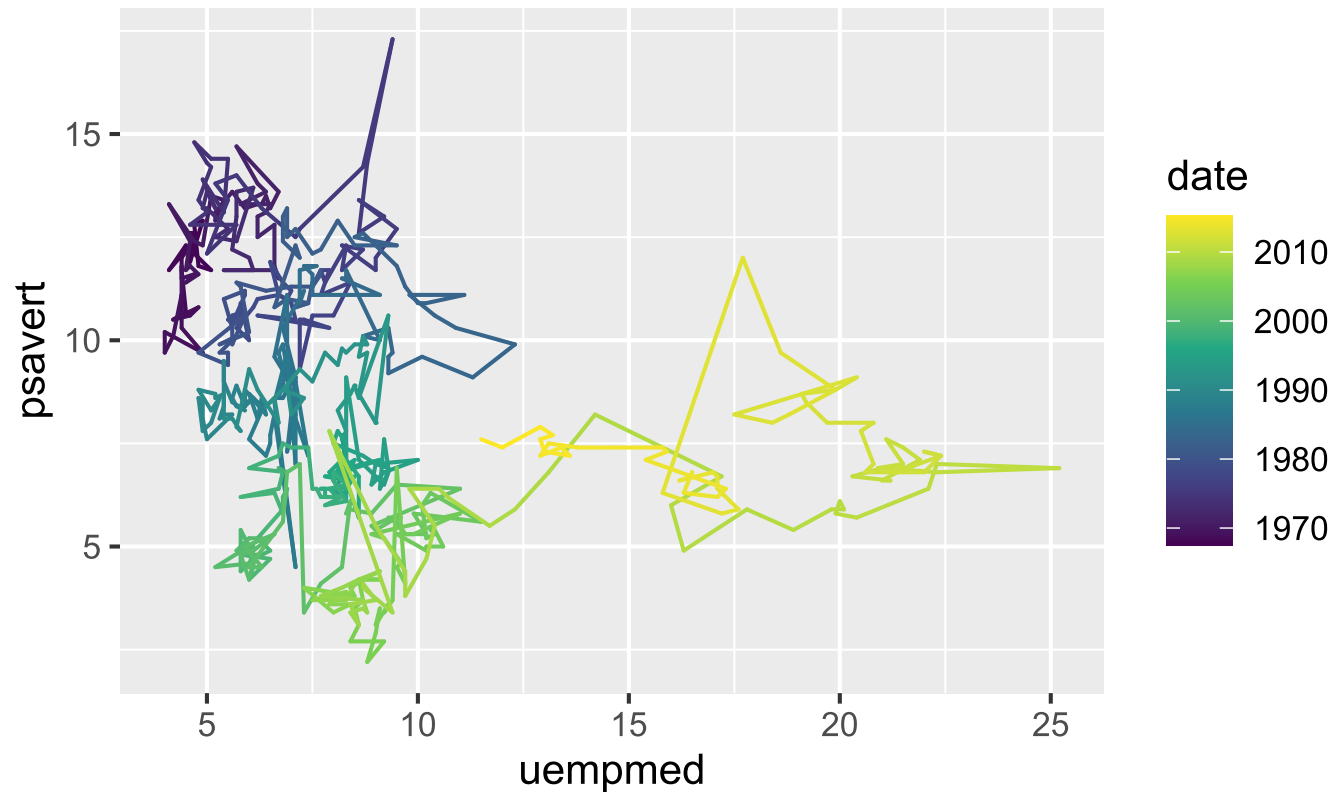
```
1 econ |>  
2   ggplot(aes(date, unemployment)) +  
3   geom_step() 
```



Linecharts

Приклад з `geom_path()`, графік демонструє зв'язок рівня особистих заощаджень американців (psavert, заощадження як відсоток від наявного особистого доходу) та медіанної тривалості безробіття у тижнях (uempmed) по роках

```
1 economics |>  
2   ggplot(aes(uempmed, psavert)) +  
3   geom_path(aes(color = date), lineend = "round") +  
4   scale_color_viridis_c(trans = "date")
```



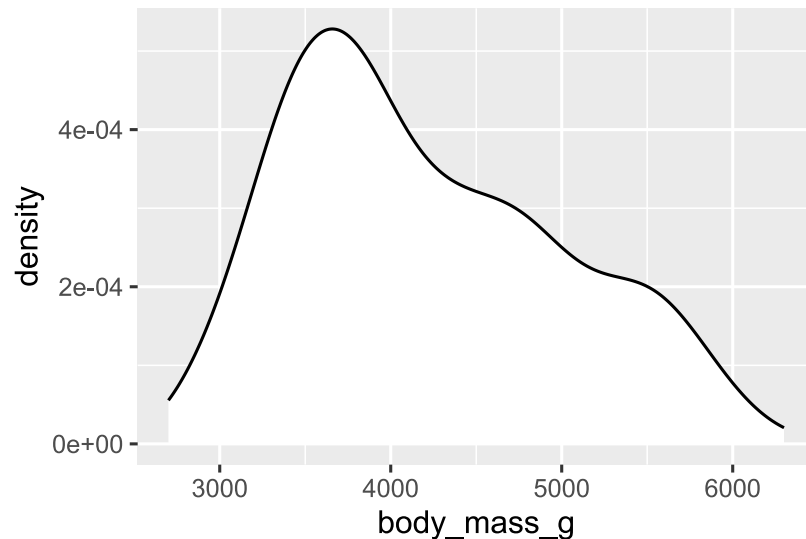
Boxplots

Використовуються для: демонстрації розподілу та відношень між чисельною та категоріальною змінною

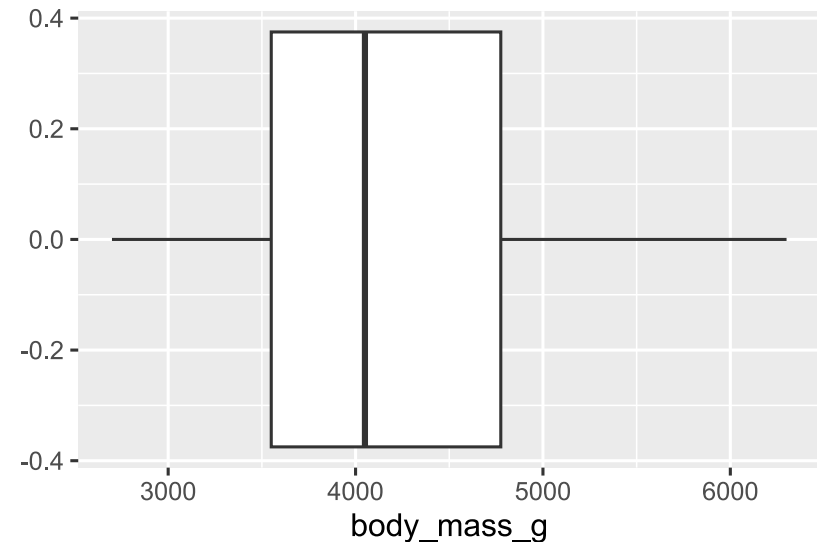
Коробкові графіки/діаграми розмаху/“ящики з вусами” демонструють розподіл статистичної вибірки через візуалізацію квантилів

```
1 summary(penguins$body_mass_g)
2 #>      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
3 #>    2700    3550    4050    4207    4775    6300
```

```
1 penguins |>
2   ggplot(aes(body_mass_g)) +
3   geom_density(fill = "white")
```

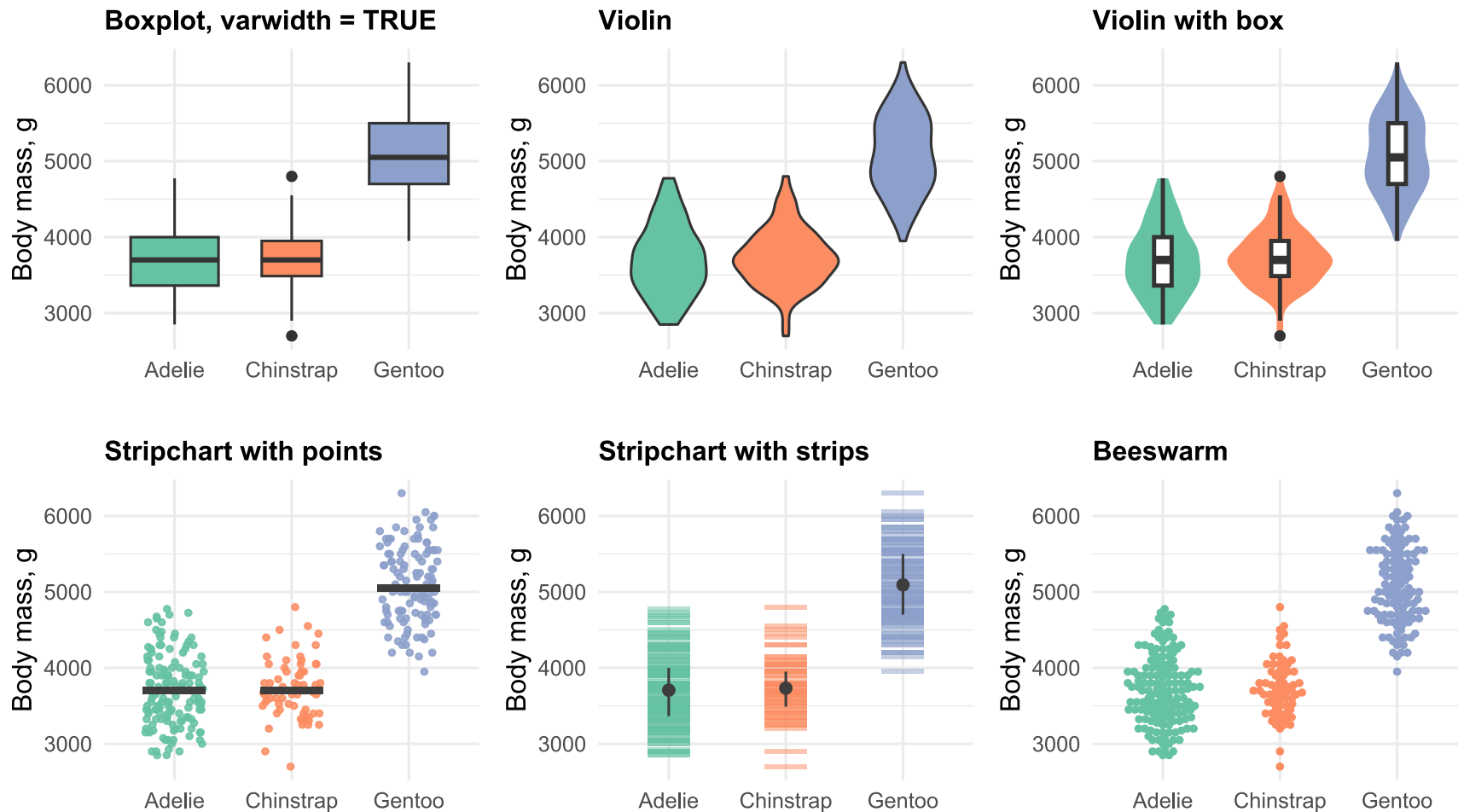


```
1 penguins |>
2   ggplot(aes(body_mass_g)) +
3   geom_boxplot()
```



Boxplot-related

Одним із недоліків боксплотів є те, що вони формально приховують “реальну” форму розподілу значень. Їх альтернативами є графіки-“скрипки” (`geom_violin()`), стріпчарти (`geom_point()`) та різні варіанти дотплотів (`geom_dotplot()`) або функції з пакету [ggbeeswarm](#)



Boxplot-related

Риджлайн графіки з [ggridges](#) для одночасної демонстрації розподілів рівнів категоріальної змінної без використання фасетингу

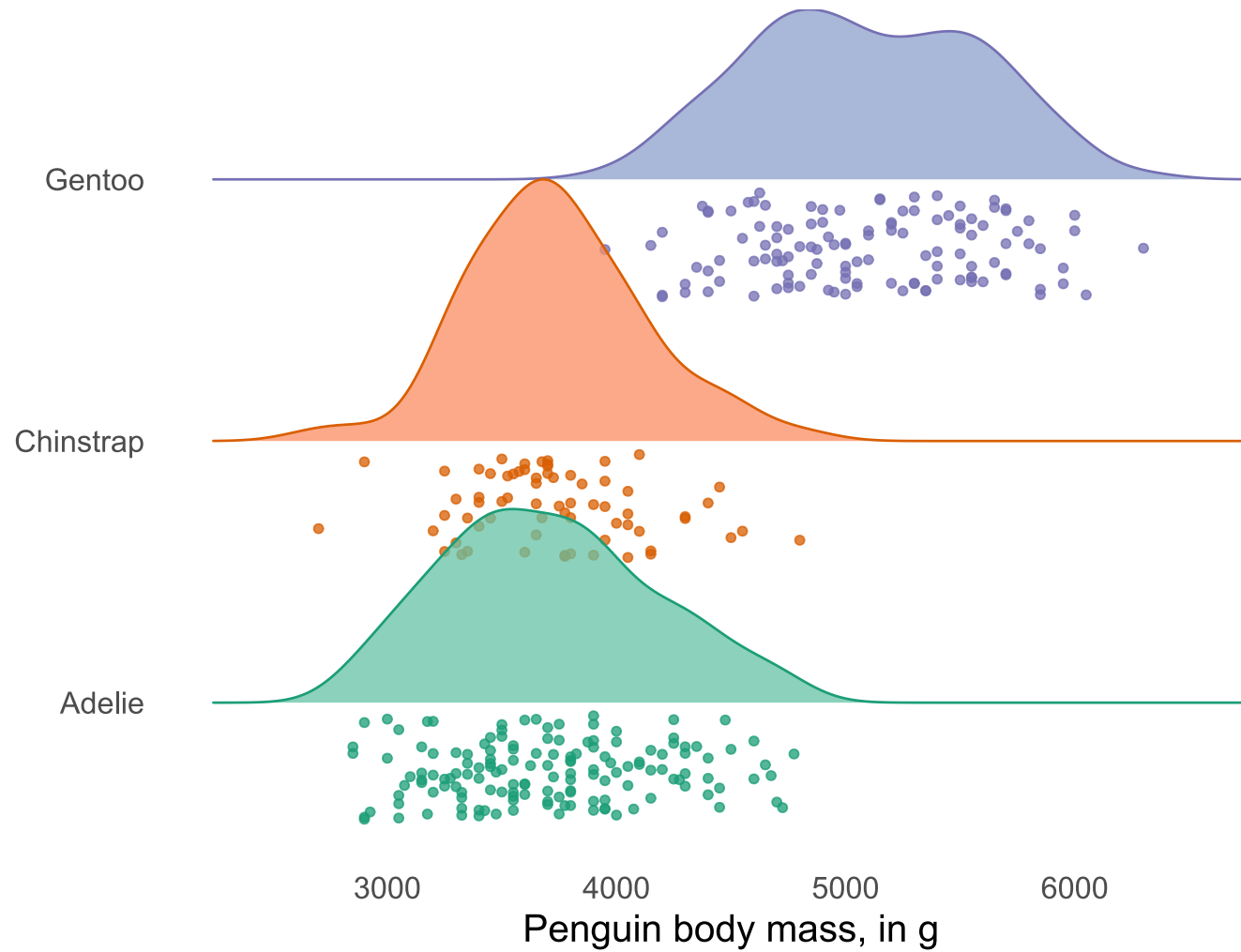
```
1 penguins |>
2   ggplot(aes(body_mass_g, y = species)) +
3     ggridges::geom_density_ridges(
4       aes(fill = species, color = species),
5       scale = 1, alpha = .75,
6       jittered_points = TRUE,
7       position = "raincloud",
8       show.legend = FALSE
9     ) +
10    scale_fill_brewer(palette = "Set2") +
11    scale_color_brewer(palette = "Dark2") +
12    labs(x = "Penguin body mass, in g", y = "",
13         title = "Raincloud plot", subtitle = "created with ggridges") +
14    theme_minimal(base_size = 16) +
15    theme(panel.grid = element_blank(),
16         plot.title = element_text(face = "bold"))
```

Також щодо цього рекомендую переглянути можливості, що надаються пакетами [gghalves](#) та [ggdist](#)

Boxplot-related

Raincloud plot

created with ggridges



Quick summary

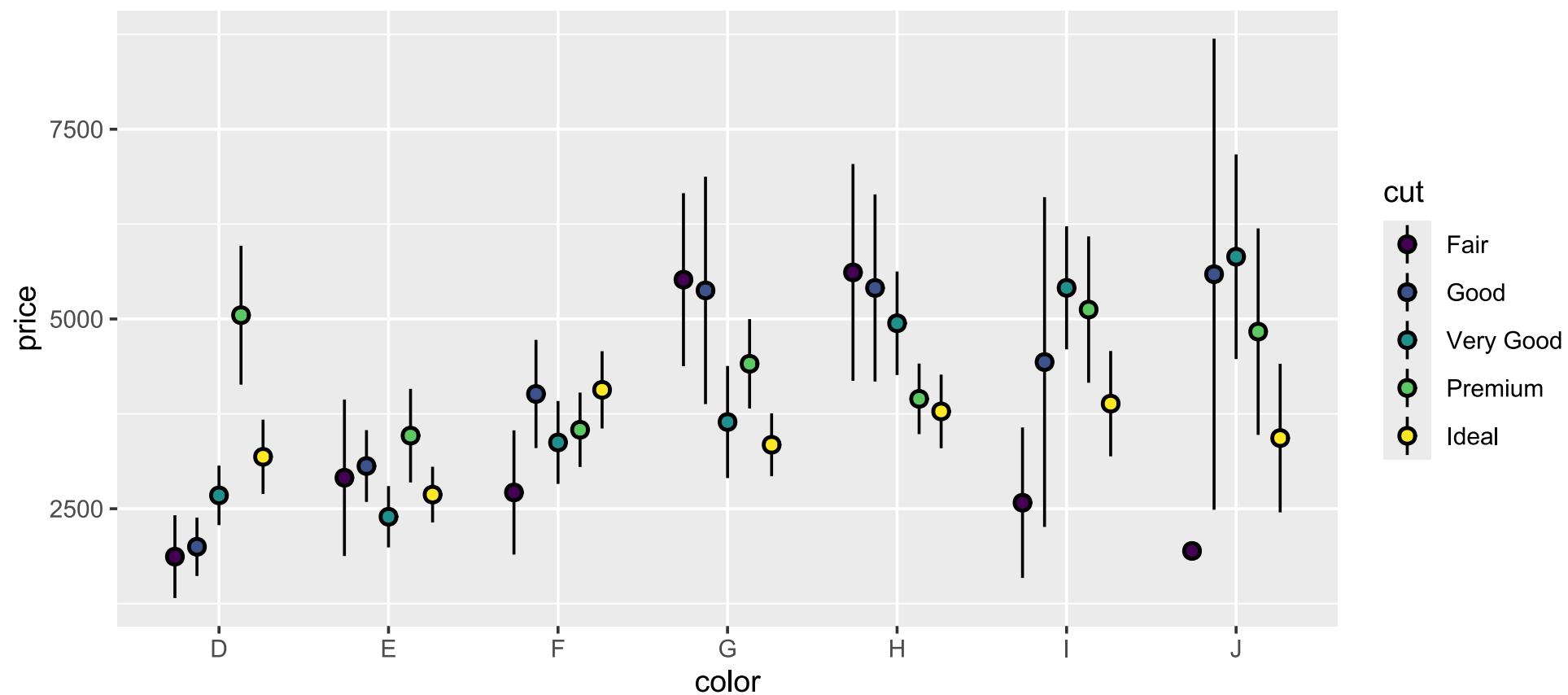
Функція `stat_summary` дозволяє швидко створювати графіки, що відображують певний статистичний підсумок. Має декілька можливих готових параметрів для аргументу `fun.data`, що повертає три значення для побудови графіку:

- `"mean_se"` — середнє зі стандартною похибкою
- `"mean_sdl"` — середнє зі стандартним відхиленням
- `"mean_cl_normal"` — середнє з 95% інтервалами достовірності для нормального розподілу
- `"mean_cl_boot"` — середнє з 95% інтервалами достовірності на основі бутстрепа
- `"median_hilow"` — медіана, 2.5 та 97.5 перцентиль

Можливо також викликати власну функцію через аргумент `fun`, або набір з трьох аргументів `fun`, `fun.max`, `fun.min`. Дефолтно використовує геометрію `pointrange`, що потребує трьох значень

Quick summary

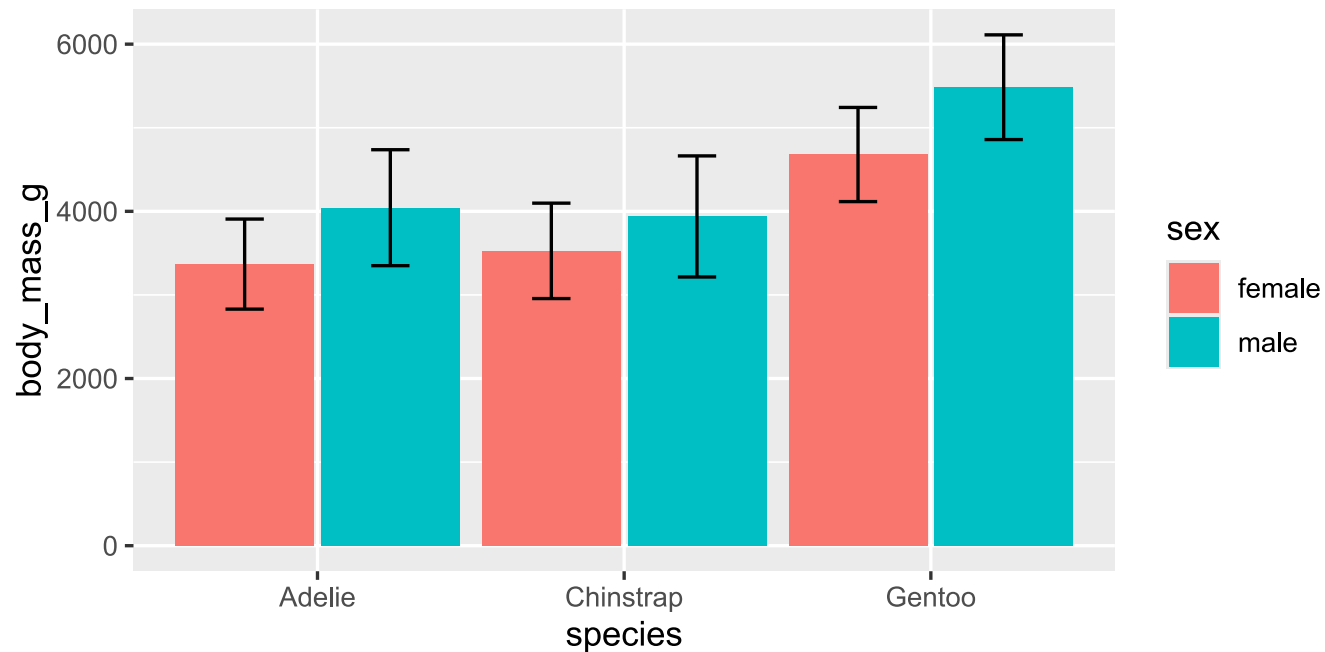
```
1 dsmall |>
2   ggplot(aes(color, price, fill = cut)) +
3   stat_summary(
4     fun.data = "mean_se",
5     position = position_dodge(width = .65),
6     shape = 21
7   )
```



Quick summary

Приклад побудови барпловтів з анотацією стандартного відхилення у вибірці через `stat_summary`

```
1 penguins |>
2   ggplot(aes(species, body_mass_g, group = sex)) +
3   stat_summary(aes(fill = sex),
4               fun = "mean",
5               geom = "bar",
6               position = position_dodge(width = 0.95)) +
7   stat_summary(fun.data = "mean_sdl",
8               geom = "errorbar",
9               width = .25,
10              position = position_dodge(width = 0.95))
```



Pairwise plots

Функція `ggpairs` з додаткового пакету [GGally](#), дозволяє створити парний графік аналогічний `pairs` з базового графічного пакету R. Докладніше про налаштування [до документації](#)

```
1 penguins |>
2   GGally::ggpairs(
3     columns = 3:7,
4     lower = list(
5       mapping = aes(color = species, fill = species, alpha = .85)
6     ),
7     upper = list(
8       mapping = aes(color = species)
9     )
10  ) +
11  scale_color_brewer(palette = "Set2") +
12  scale_fill_brewer(palette = "Pastel2")
```

Pairwise plots

