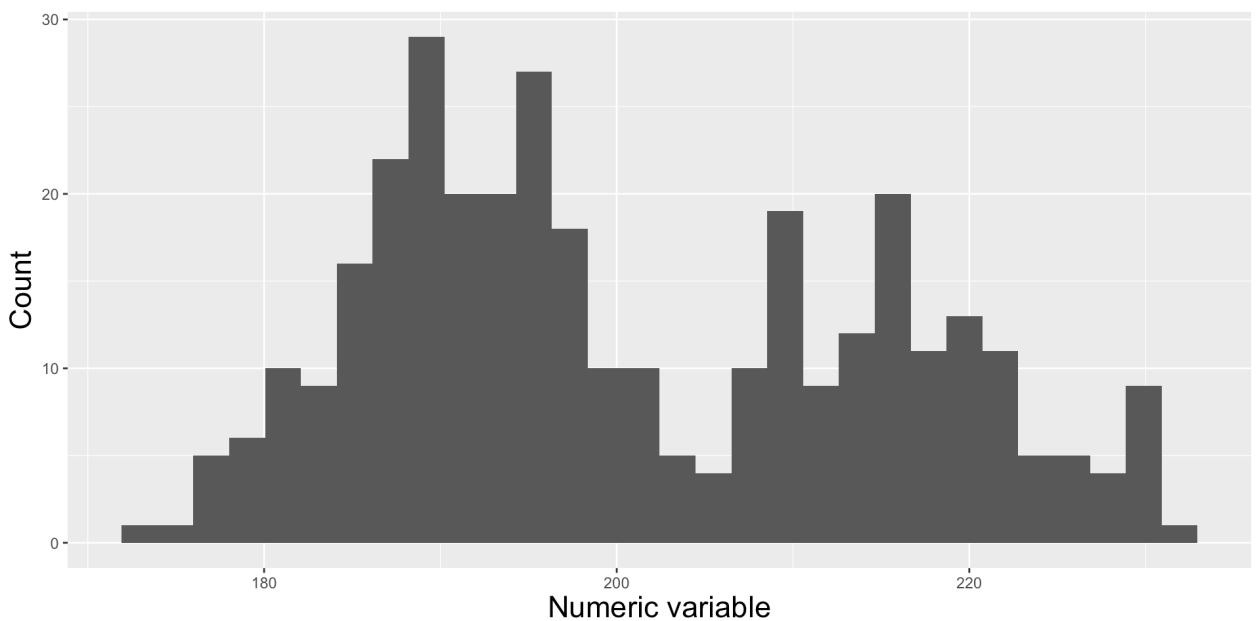
EDS 240: Lecture 2.2

Visualizing distributions

Week 2 | January 13th, 2024

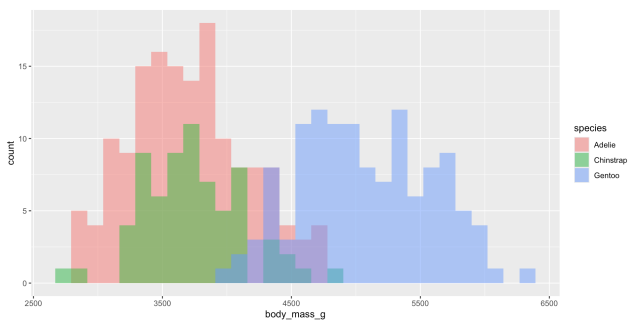
Visualizing data *distribution*?

Visualizing the **spread** of a numeric variable(s)

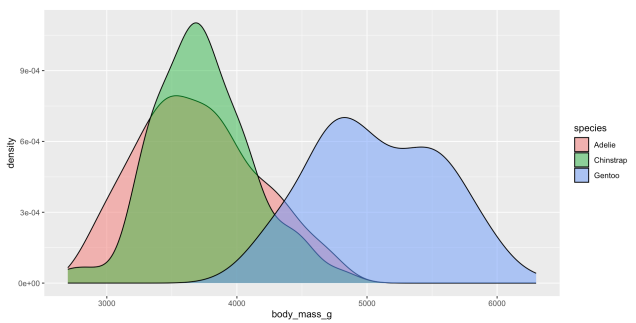


“Core” distribution chart types

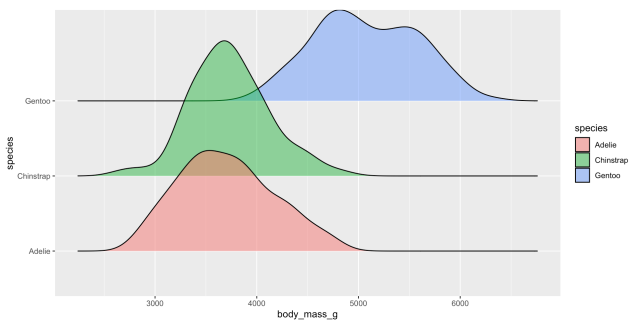
Histograms



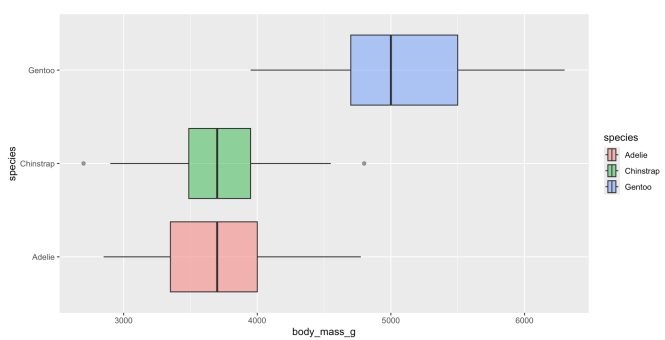
Density plots



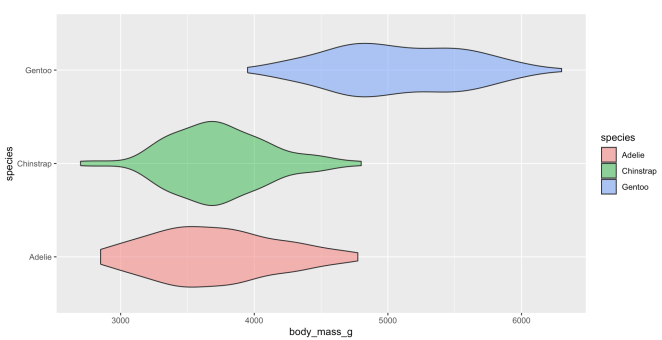
Ridgeline plots



Box plots



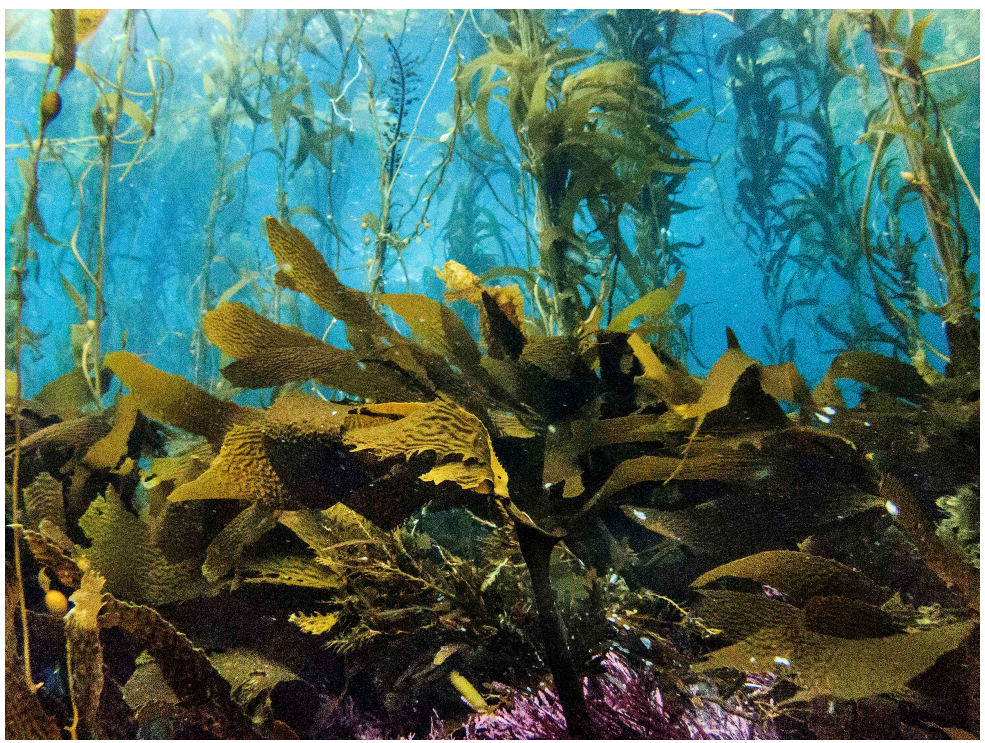
Violin plots



Examples show the distribution of penguin body masses (g) for Adelie, Chinstrap & Gentoo penguins.

The data: bottom temperatures at Mohawk Reef

The Santa Barbara Coastal Long Term Ecolgical Research (SBC LTER) site was established in 2000 to understand the ecology of coastal kelp forest ecosystems. A number of coastal rocky reef sites are outfitted with instrumentation that collect long-term monitoring data.



We'll be exploring **bottom temperatures recorded at Mohawk Reef**, a near-shore rocky reef and one of the Santa Barbara Coastal (SBC) LTER research sites.

Data wrangling

Data are imported directly from the EDI Data Portal. Explore the metadata package online to learn more about these data.

```
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
##                                  setup                                  ----
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

#.........................load packages.........................
library(tidyverse)
library(chron)
library(naniar)
library(palmerpenguins) # for some minimal examples

#.........................import data...........................
mko <- read_csv("https://portal.edirepository.org/nis/dataviewer?packageid=knb-lter

##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
##                               wrangle data                              ----
##~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

mko_clean <- mko |>

  # keep only necessary columns ----
```

Histograms - `ggplot2::geom_histogram()`

What are they?

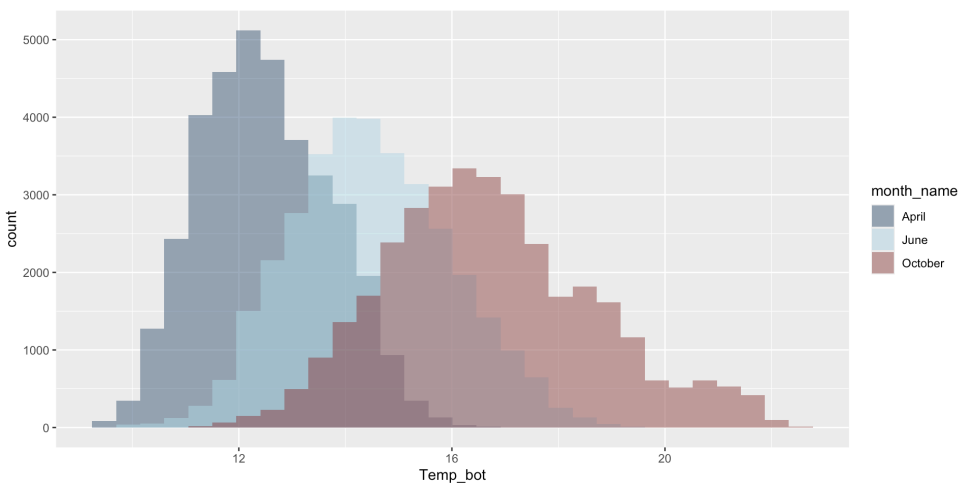
- Histograms are used to represent the **distribution of a *numeric* variable(s)**, which is **cut into several bins**. The number of observations per bin is represented by the height of the bar.

Need:

- a numeric variable with lots of values
- meaningful differences between values

Important considerations:

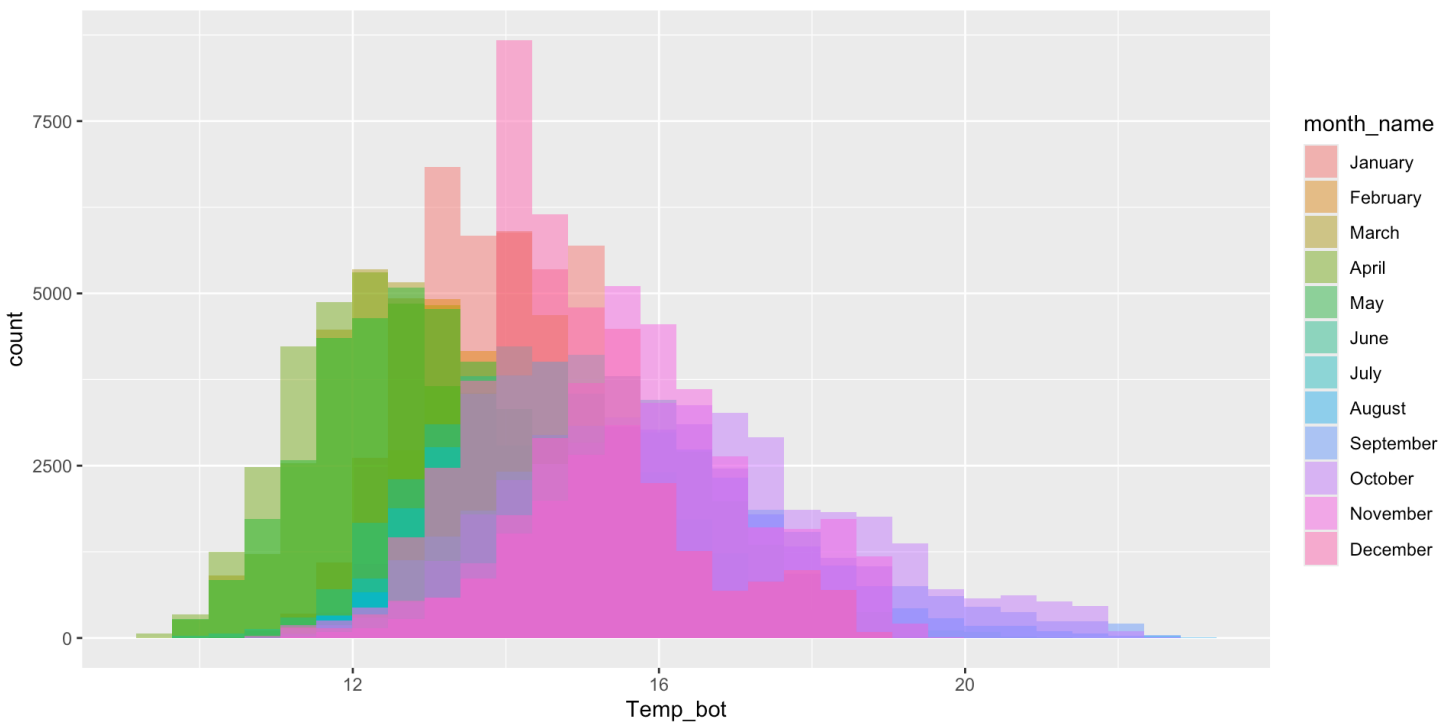
- bin width (30 **bins** by default)
- too few / too many bins



Histograms - avoid plotting too many groups

Twelve groups (`month_name`) is too many groups – especially when the range of temperature values for each of our groups largely overlap:

```
mko_clean |>
  mutate(month_name = factor(month_name, levels = month.name)) |>
  ggplot(aes(x = Temp_bot, fill = month_name)) +
  geom_histogram(position = "identity", alpha = 0.5)
```

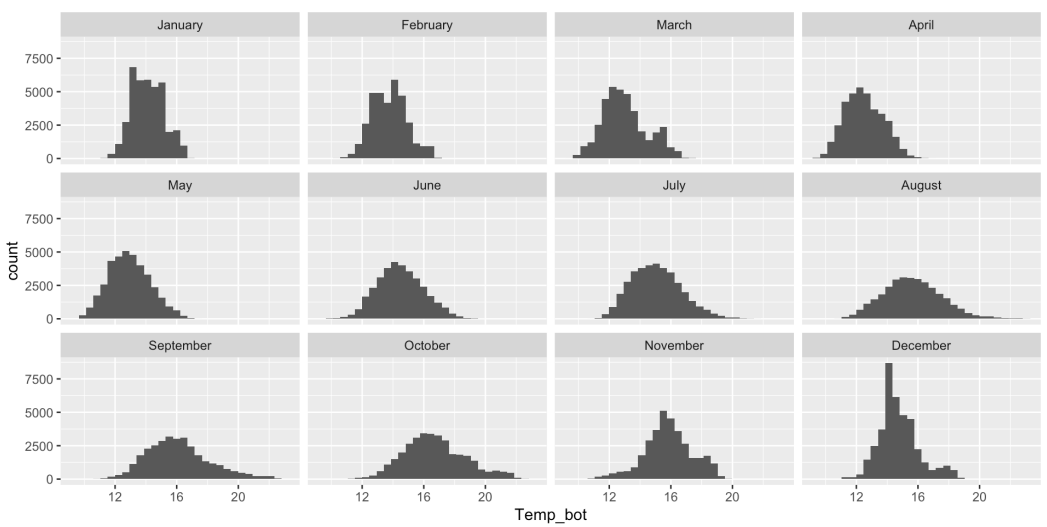


Histograms - adjustments

Small multiples | Fewer groups | Adjust colors | Modify bin widths

If you want to plot all groups, consider splitting them into small multiples. If so, does color add any valuable information? Remove if not:

```
mko_clean |>
  mutate(month_name = factor(month_name, levels = month.name)) |>
  ggplot(aes(x = Temp_bot)) +
  geom_histogram() +
  facet_wrap(~month_name)
```



Density plots - `ggplot2::geom_density()`

What are they?

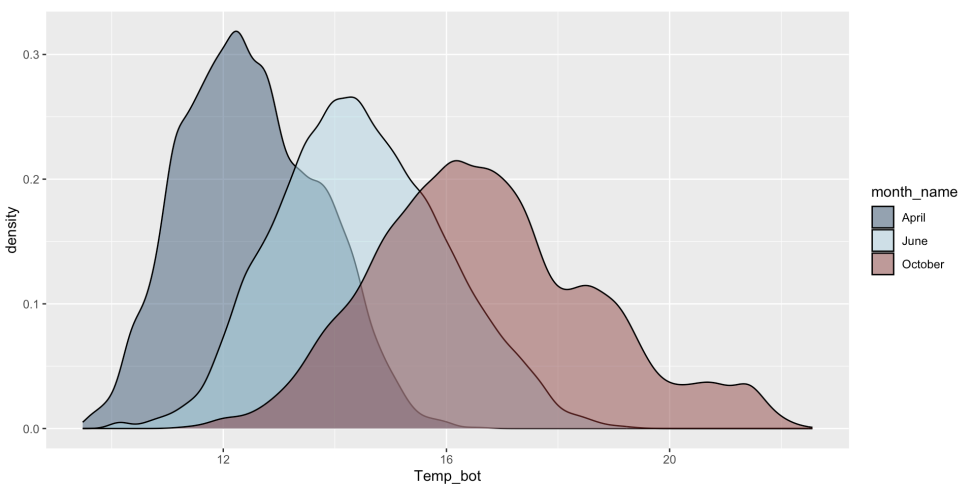
A smoothed version of a histogram. Density plots are representations of the **distribution of a *numeric* variable(s)**, which uses a kernel density estimate (KDE) to show the probability density function of the variable. **The y-axis represents the estimated density, i.e. the relative likelihood of a value occurring. The area under each curve is equal to 1.** *Use a density plot when you are most concerned with the shape of the distribution.*

Need:

- a numeric variable with lots of values

Important considerations:

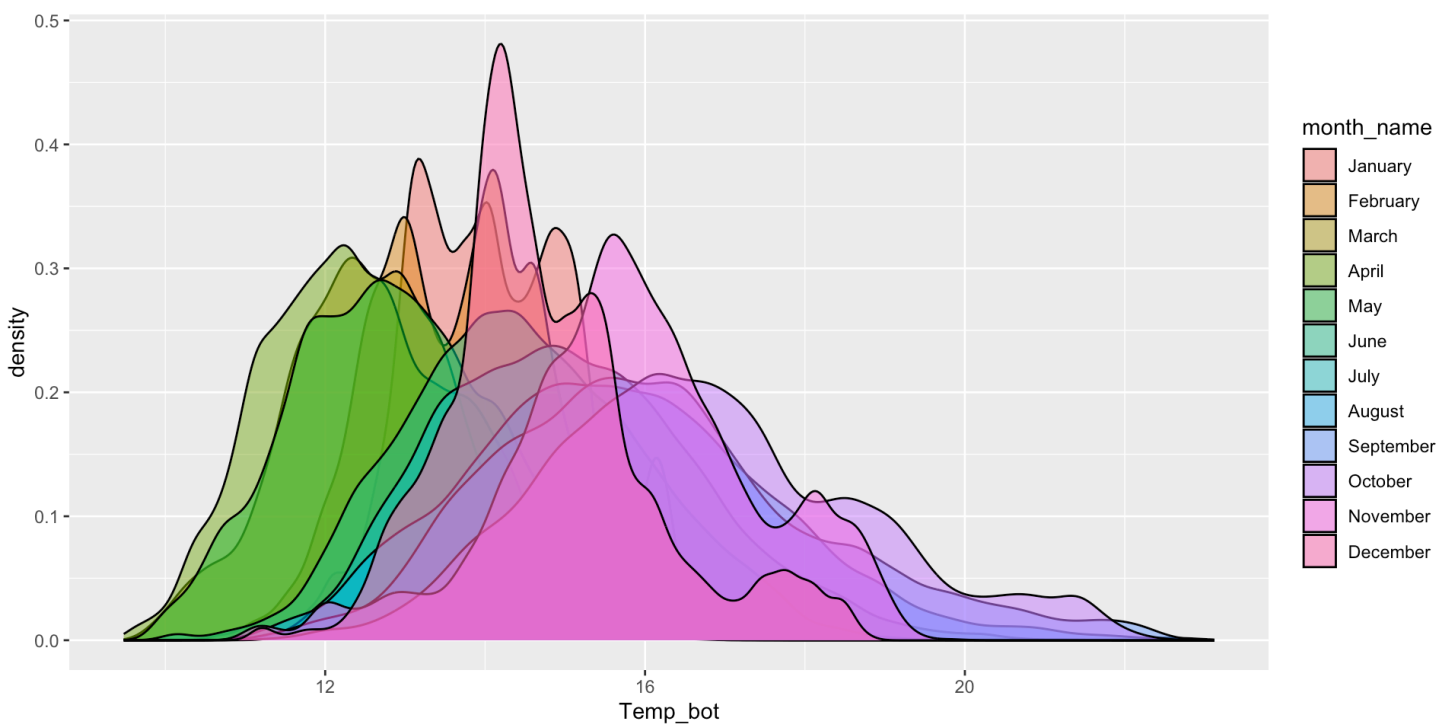
- useful when you want to visualize the shape of your data (not affected by bin number)
- does not indicate sample size
- can be misleading with small data sets
- band width, which affects level of smoothing



Density plots - avoid plotting too many groups

Similar to the histogram, twelve groups (`month_name`) is too many groups! Consider small multiples (using `facet_wrap()`) if you want to keep all groups.

```
mko_clean |>
  mutate(month_name = factor(x = month_name, levels = month.name)) |>
  ggplot(aes(x = Temp_bot, fill = month_name)) +
  geom_density(alpha = 0.5)
```

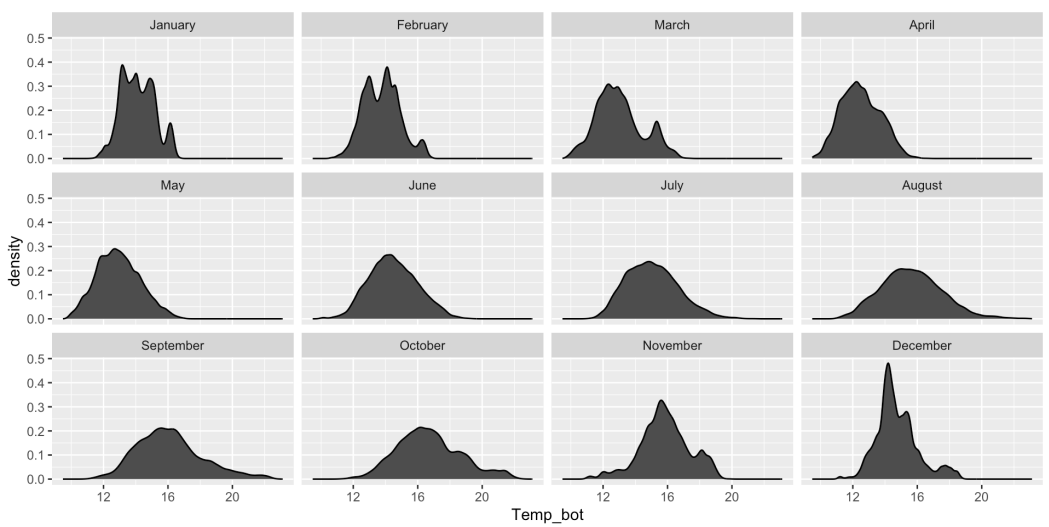


Density plots - adjustments

Small multiples | Fewer groups | Modify band widths

If you want to plot all groups, consider splitting them into small multiples. If so, does color add any valuable information? Remove if not:

```
mko_clean |>
  mutate(month_name = factor(month_name, levels = month.name)) |>
  ggplot(aes(x = Temp_bot)) +
  geom_density(fill = "gray30") +
  facet_wrap(~month_name)
```



An important distinction

Histograms show us the **counts** (frequency) of values in each range (bin), represented by the height of the bars.

Density plots show the **proportion** of values in each range (area under the curve equal 1; peaks indicate where more values are concentrated, but it does *not* tell us anything about the *the number* of observations).

We'll use some dummy data to demonstrate how this differs visually:

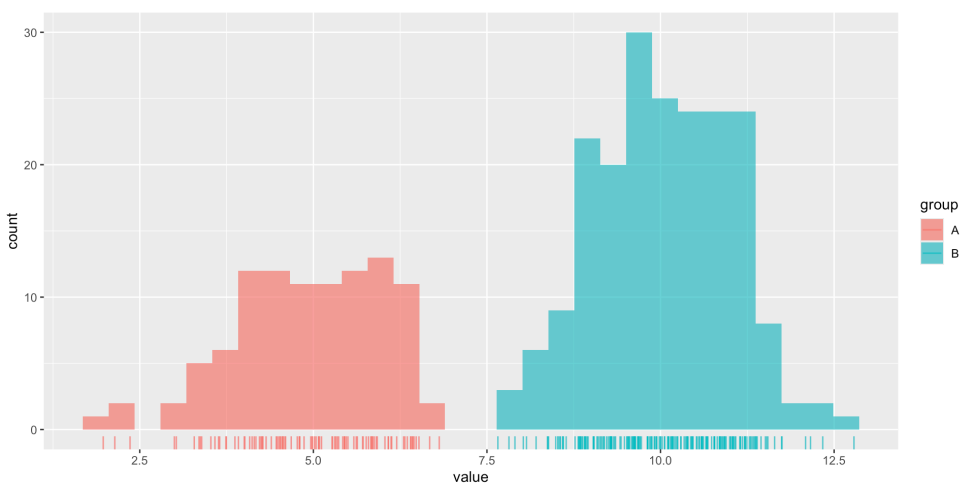
```
dummy_data <- data.frame(value = c(rnorm(n = 100, mean = 5),
                                   rnorm(n = 200, mean = 10)),
                         group = rep(c("A", "B"),
                                     times = c(100, 200)))
```

Here, we have two groups (`A`, `B`) of values which are normally distributed, but with different means. Group `A` also has a smaller sample size (100) than group `B` (200).

An important distinction

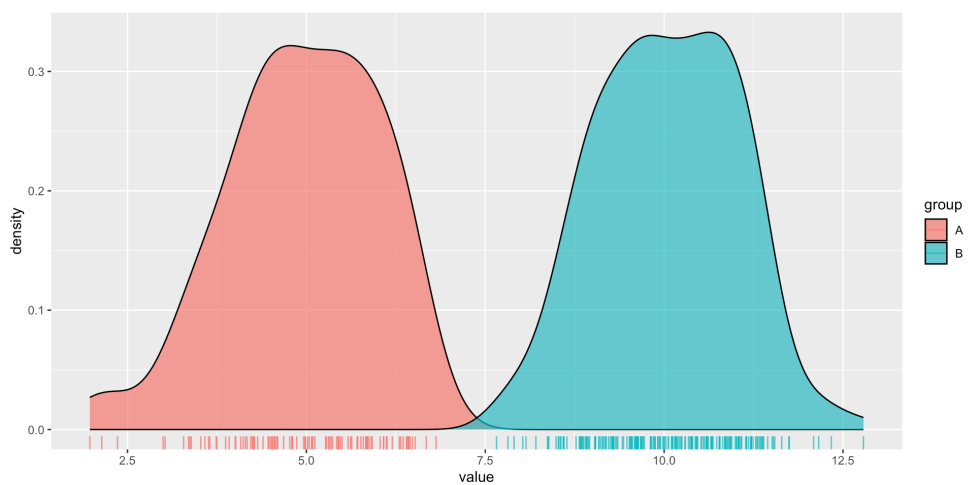
It's easy to see that group `B` has a larger sample size than group `A` when looking at our histogram. Additionally, we can get a good sense of our data distribution. *But* what happens when you reduce the number of `bins` (e.g. set `bins = 4`)?

```
ggplot(dummy_data, aes(x = value, fill
  geom_histogram(position = "identity"
  geom_rug(aes(color = group), alpha =
```



We lose information about sample size in our density plot (note that both curves are ~the same height, despite group `B` having 2x as many observations). However, they're great for visualizing the shape of our distributions since they are unaffected by the number of bins.

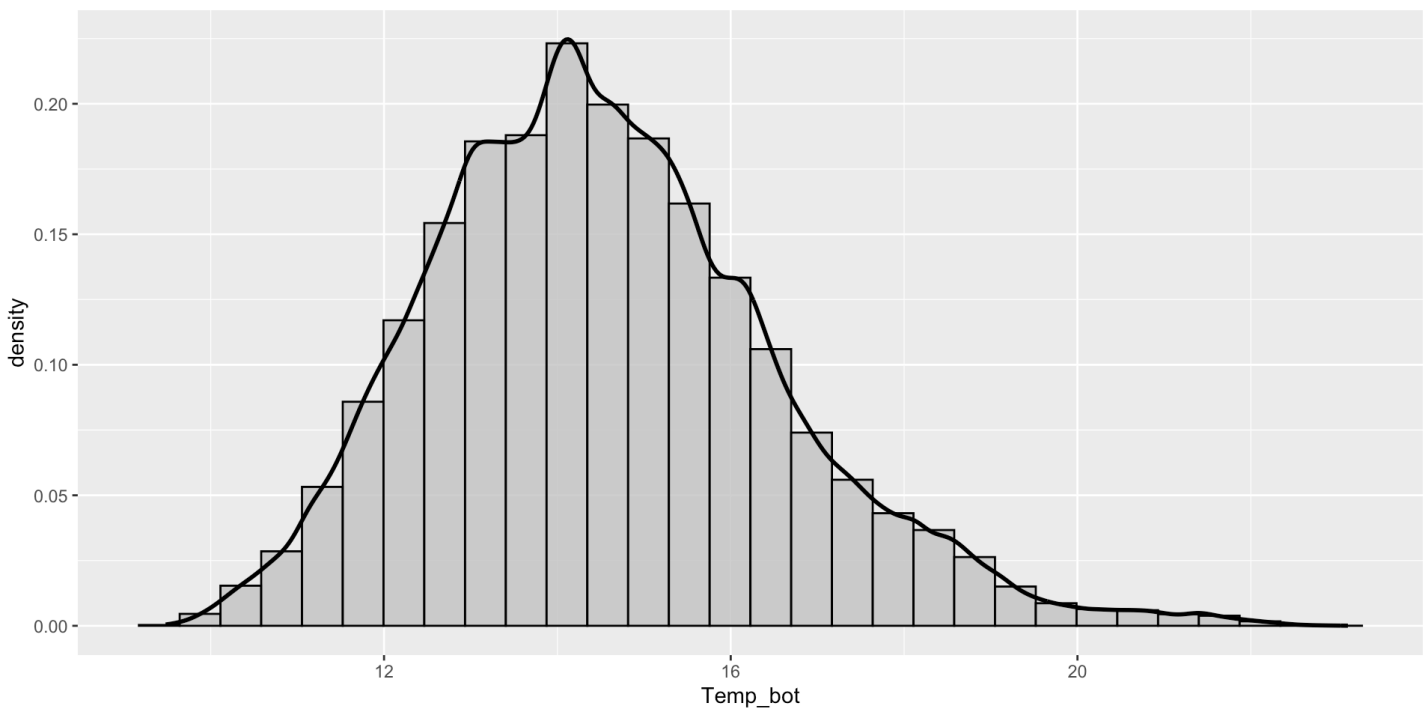
```
ggplot(dummy_data, aes(x = value, fill
  geom_density(alpha = 0.7) +
  geom_rug(aes(color = group), alpha =
```



Combining geoms - histogram & density plot

Overlaying a histogram and density plot requires scaling down the histogram to match the density curve scale. Adding `y = after_stat(density)` within the `aes()` function rescales the histogram counts so that bar areas integrate to 1:

```
ggplot(mko_clean, aes(x = Temp_bot, y = after_stat(density))) + # scale down hist t
  geom_histogram(fill = "gray", color = "black", alpha = 0.75) +
  geom_density(size = 1)
```



Scaled density plots for comparing groups to a whole

In a normal density plot, the area under the curve(s) is equal to 1. **In a scaled density plot, the area under the curve reflects the number of observations for each group.**

We can use scaled density plots to compare individual group distributions to the total distribution. Demonstrated here, using the `palmerpenguins::penguins` data set:

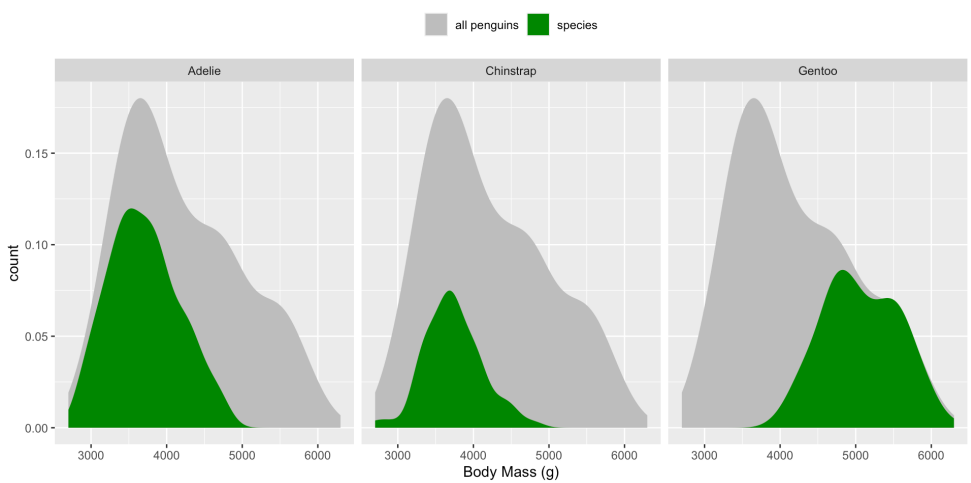
```
# use `after_stat(count)` to plot dens
ggplot(penguins, aes(x = body_mass_g,

  # plot full distribution curve with
  geom_density(data = select(penguins,
                aes(fill = "all penguin

  # plot second curve with label "spec
  geom_density(aes(fill = "species"),

  # facet wrap by species ----
  facet_wrap(~species, nrow = 1) +

  # update colors, x-axis label, legen
  scale_fill_manual(values = c("grey",
  labs(x = "Body Mass (g)") +
  theme(legend.position = "top")
```



Ridgeline plots - {ggridges}

What are they?

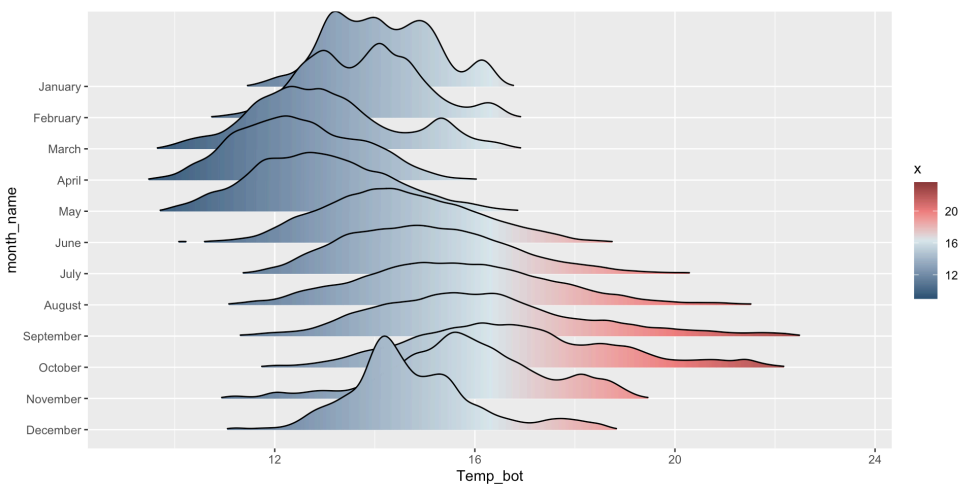
- Ridgeline plots show the **distribution of a *numeric* variable** for **multiple groups**.

Need:

- a numeric variable with lots of values

Important considerations:

- work best when you have > 6 groups
- works well when there is a clear pattern in the result (e.g. if there is an obvious ranking in groups) and / or when visualizing changes in distributions over time or space

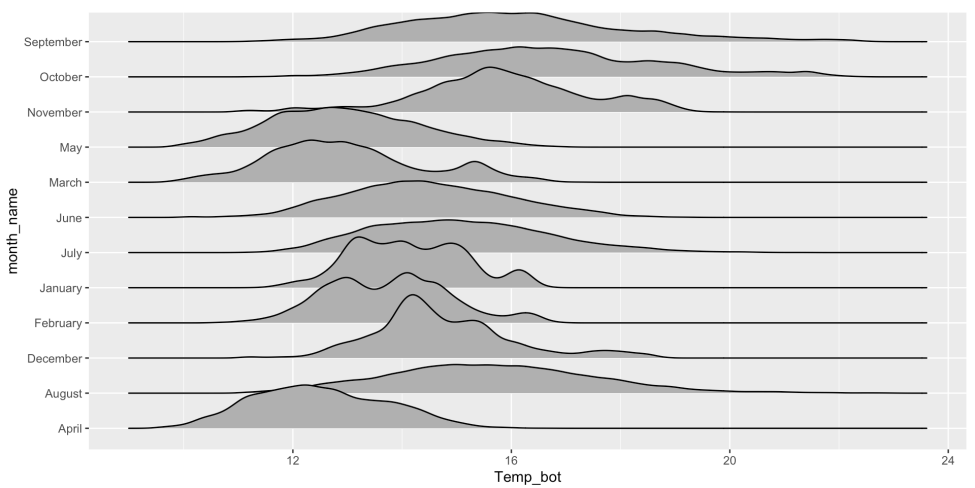


Ridgeline plots - good for multiple groups

The `{ggridges}` package has a number of different geoms for creating ridgeline plots that work well for data sets with larger group numbers (e.g. months). Two great geoms to explore (to start):

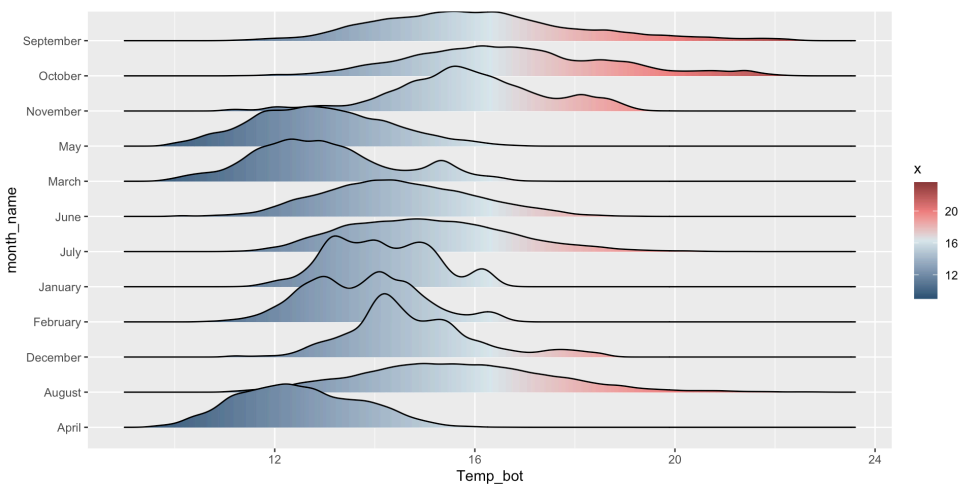
`geom_density_ridges()` to create a basic ridgeline plot:

```
1  ggplot(mko_clean, aes(x = Temp_bot, y
2    ggridges::geom_density_ridges()
```



`geom_density_ridges_gradient()` to fill with a color gradient:

```
1  ggplot(mko_clean, aes(x = Temp_bot, y
2    ggridges::geom_density_ridges_gradie
3    scale_fill_gradientn(colors = c("#2C
```

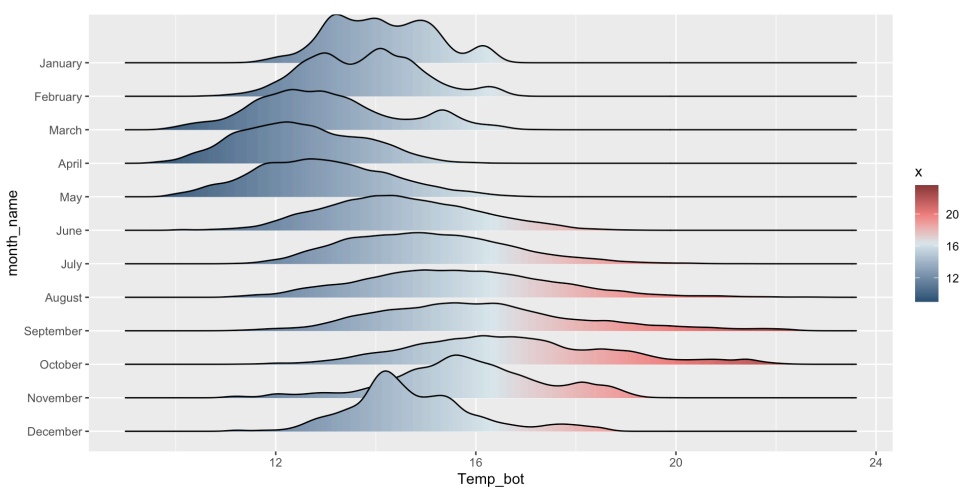


Ridgeline plots - adjustments

Group order | Overlap & tails | Quantiles | Jitter raw data

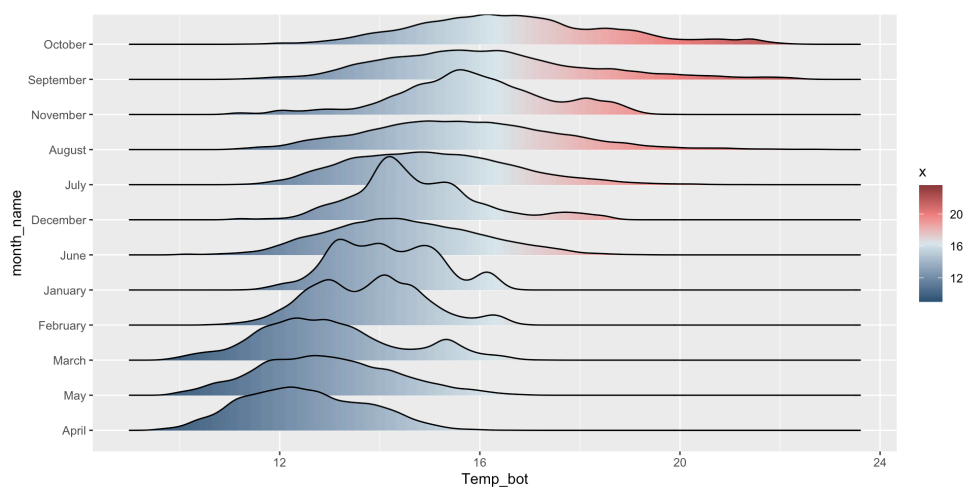
Order by month (ideal, since months have an inherent order):

```
1  ggplot(mko_clean, aes(x = Temp_bot, y
2    ggridges::geom_density_ridges_gradie
3    scale_y_discrete(limits = rev(month.
4    scale_fill_gradientn(colors = c("#2C
```



Order by mean or median (makes more sense when you have unordered groups):

```
1  mko_clean |>
2    mutate(month_name = fct_reorder(mont
3    ggplot(mko_clean, mapping = aes(x =
4    ggridges::geom_density_ridges_gradie
5    scale_fill_gradientn(colors = c("#2C
```



Box plots - `ggplot2::geom_boxplot()`

What are they?

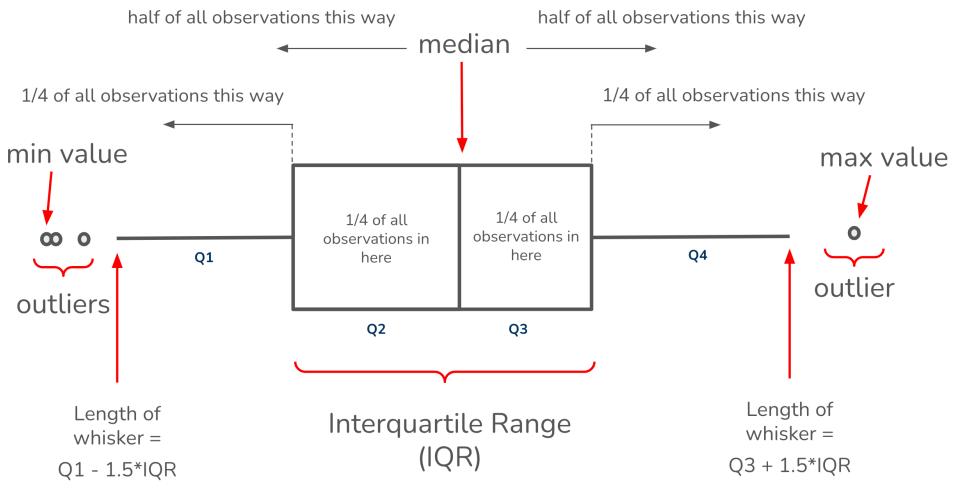
- Box plots **summarize the distribution of a numeric variable for one or several groups**.

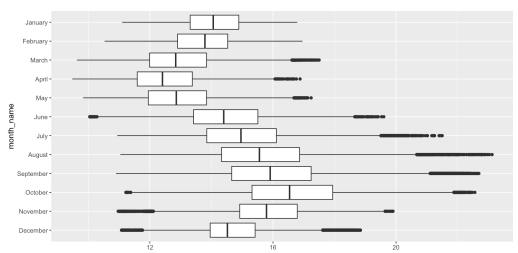
Need:

- a numeric variable, often with multiple groups

Important considerations:

- box plots *summarize* data, meaning we can't see the underlying shape of the distribution or sample size
- add jittered points on top, or if large sample size, consider a violin plot

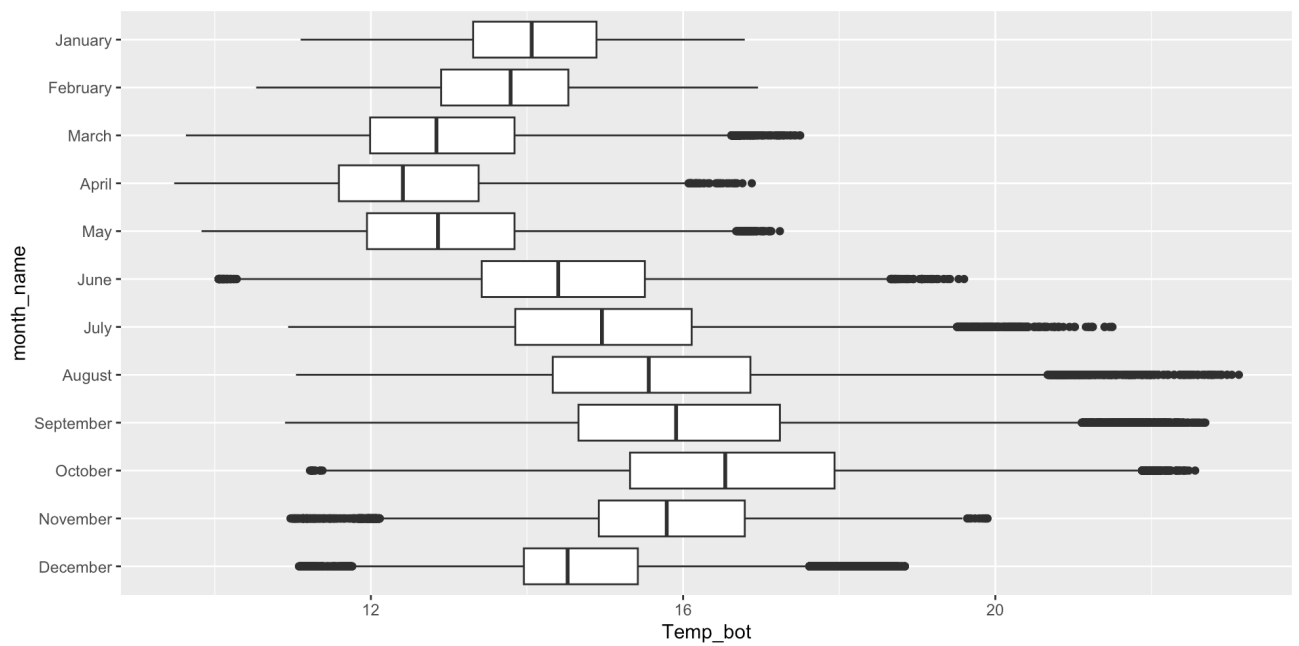




Box plots - good for multiple groups

Box plots are great for a few to multiple groups (too many boxes just results in a lot of information to synthesize, as a viewer). If your x-axis text is long, consider flipping your axes to make them less crunched:

```
ggplot(mko_clean, aes(x = month_name, y = Temp_bot)) +
  geom_boxplot() +
  scale_x_discrete(limits = rev(month.name)) +
  coord_flip()
```

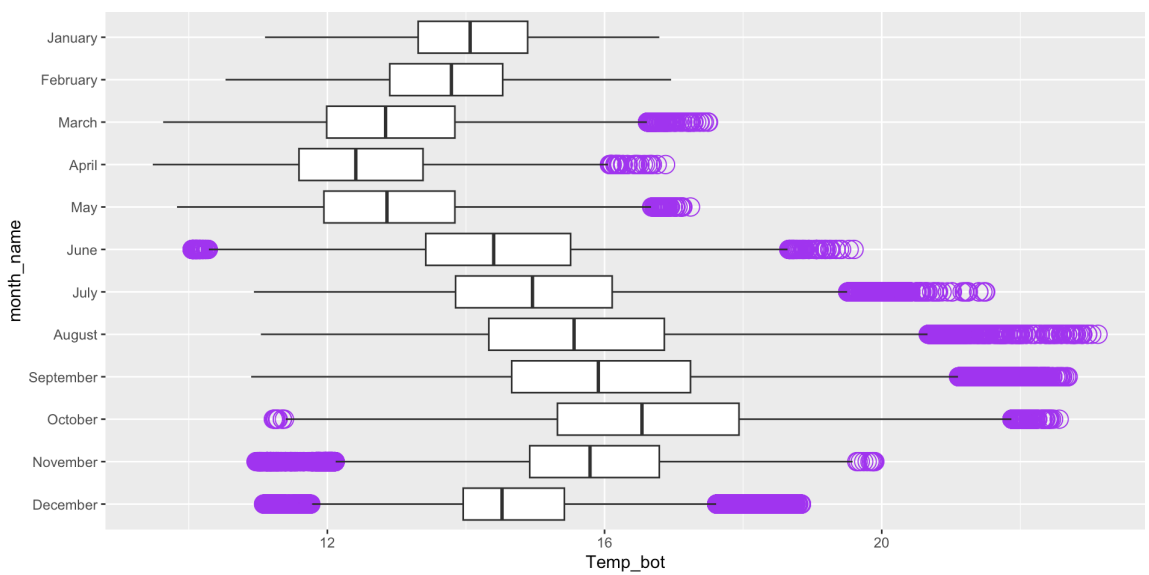


Box plots - adjustments

Outliers | Highlight a group(s) | Jitter raw data | Dodged groups | Overlay beeswarm

You can modify outlier aesthetics inside `geom_boxplot()`:

```
1  ggplot(mko_clean, aes(x = month_name, y = Temp_bot)) +
2    geom_boxplot(outlier.color = "purple", outlier.shape = "circle open", outlier.siz
3    scale_x_discrete(limits = rev(month.name)) +
4    coord_flip()
```



Violin plots - `ggplot2::geom_violin()`

What are they?

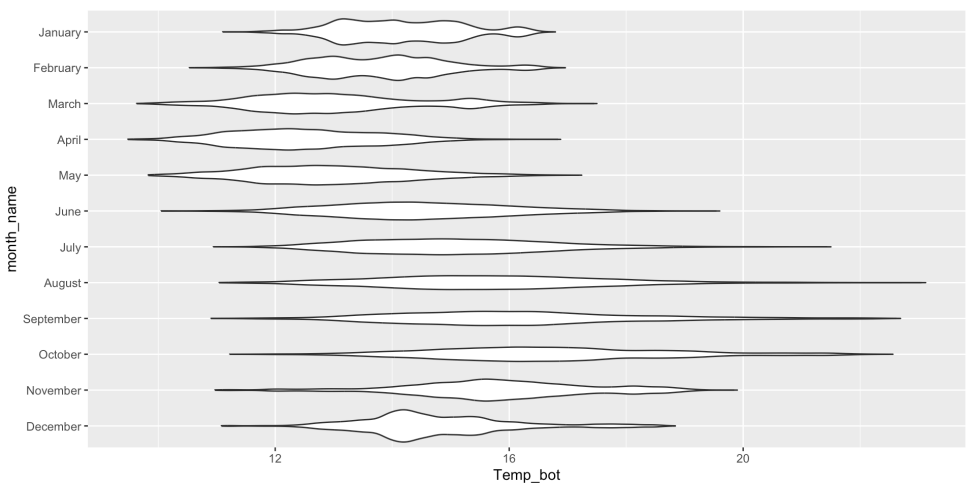
- Violin plots visualize the **distribution of a numeric variable for one or several groups**, where the shape of the violin represents the density estimate of the variable (i.e. the more data points in a specific range, the larger the violin is for that range). They provide more information about the underlying distribution than a box plot.

Need:

- a numeric variable, often with multiple groups

Important considerations:

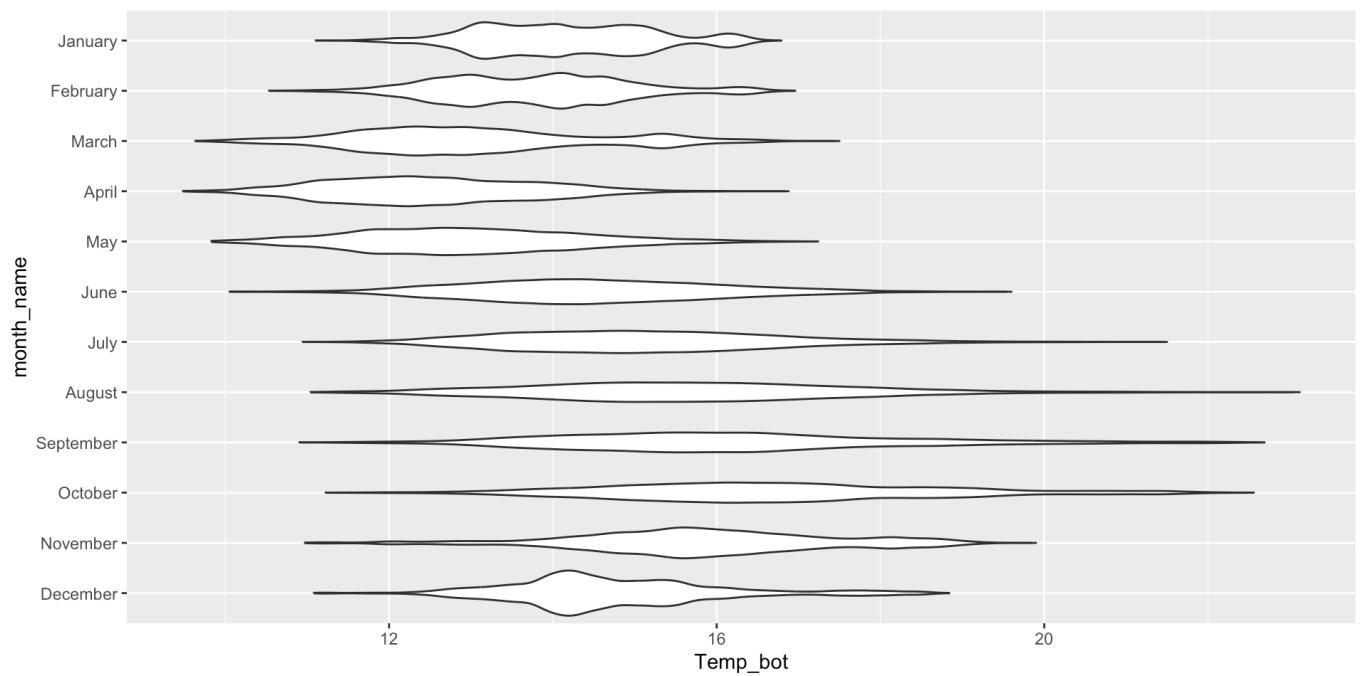
- ordering groups by median value can make it easier to understand
- show sample size when comparing groups with very different distributions (e.g. half violin plot)



Violin plots - good for multiple groups with lots of data

Violin plots are great for a few to multiple groups, and are often a better choice than box plots when you have a very large data set (and overlaying jittered points looks busy or downright unreasonable). If your x-axis text is long, consider flipping your axes to make them less crunched:

```
ggplot(mko_clean, aes(x = month_name, y = Temp_bot)) +
  geom_violin() +
  scale_x_discrete(limits = rev(month.name)) +
  coord_flip()
```

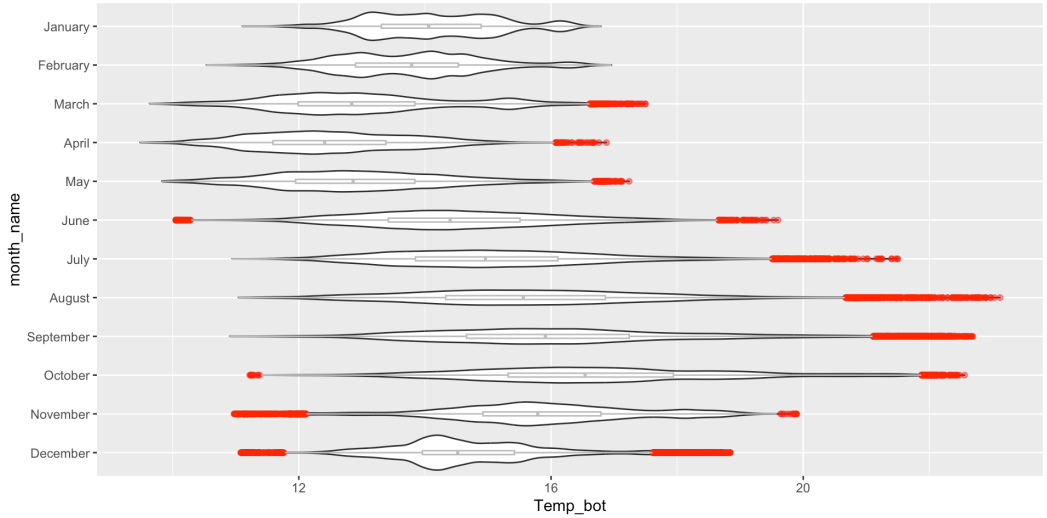


Combining geoms - adjustments

Overlay boxplot | Half-violin half-dot plot

Overlaying a box plot inside a violin plot can be helpful in providing your audience with summary stats in a compact form:

```
ggplot(mko_clean, aes(x = month_name, y = Temp_bot)) +
  geom_violin() +
  geom_boxplot(width = 0.1, color = "gray", alpha = 0.5,
               outlier.color = "red") +
  scale_x_discrete(limits = rev(month.name)) +
  coord_flip()
```


January
February
March
April
May
June
July
August
September
October
November
December
month_name
12
16
20
Temp_bot

Take a Break

~ This is the end of Lesson 2 (of 3) ~

05:00