The Grammar of Graphics in Python with `plotnine`

Hassan Kibirige

@posit::conf(2023)

Part 1: Origin

How it comes about.

Motivation

Motivation

> **E.C**: And the winner is [redacted].

> **Your Trusted Source**: Here is how the country voted …

> **Truth Everyday**: Here is how the country voted …

> **Hassan**: Please Media, I want some more.

Part 2: Meaning

What is it?

What is *The Grammar of Graphics*

What is *The Grammar of Graphics*

A language and rules to make statistical graphics

What is *The Grammar of Graphics*

A language and rules to make statistical graphics

! ! !

Components of the language

Components of the language

- Data

Components of the language

- Data
- Aesthetics

Components of the language

- Data
- Aesthetics
- Scales

Components of the language

- Data
- Aesthetics
- Scales
- Statistics

Components of the language

- Data
- Aesthetics
- Scales
- Statistics
- Positions

Components of the language

- Data
- Aesthetics
- Scales
- Statistics
- Positions
- Geometric Objects

Components of the language

- Data
- Aesthetics
- Scales
- Statistics
- Positions
- Geometric Objects
- Facets

Components of the language

- Data
- Aesthetics
- Scales
- Statistics
- Positions
- Geometric Objects
- Facets
- Coordinate System

Components of the language

- Data
- Aesthetics
- Scales
- Statistics
- Positions
- Geometric Objects
- Facets
- Coordinate System
- Theme

The Data

The Data

```
1 import pandas as pd
2 import numpy as np
3
4 penguins = pd.read_parquet("data/penguins.parquet")
5 penguins.sample(5)
```

The Data

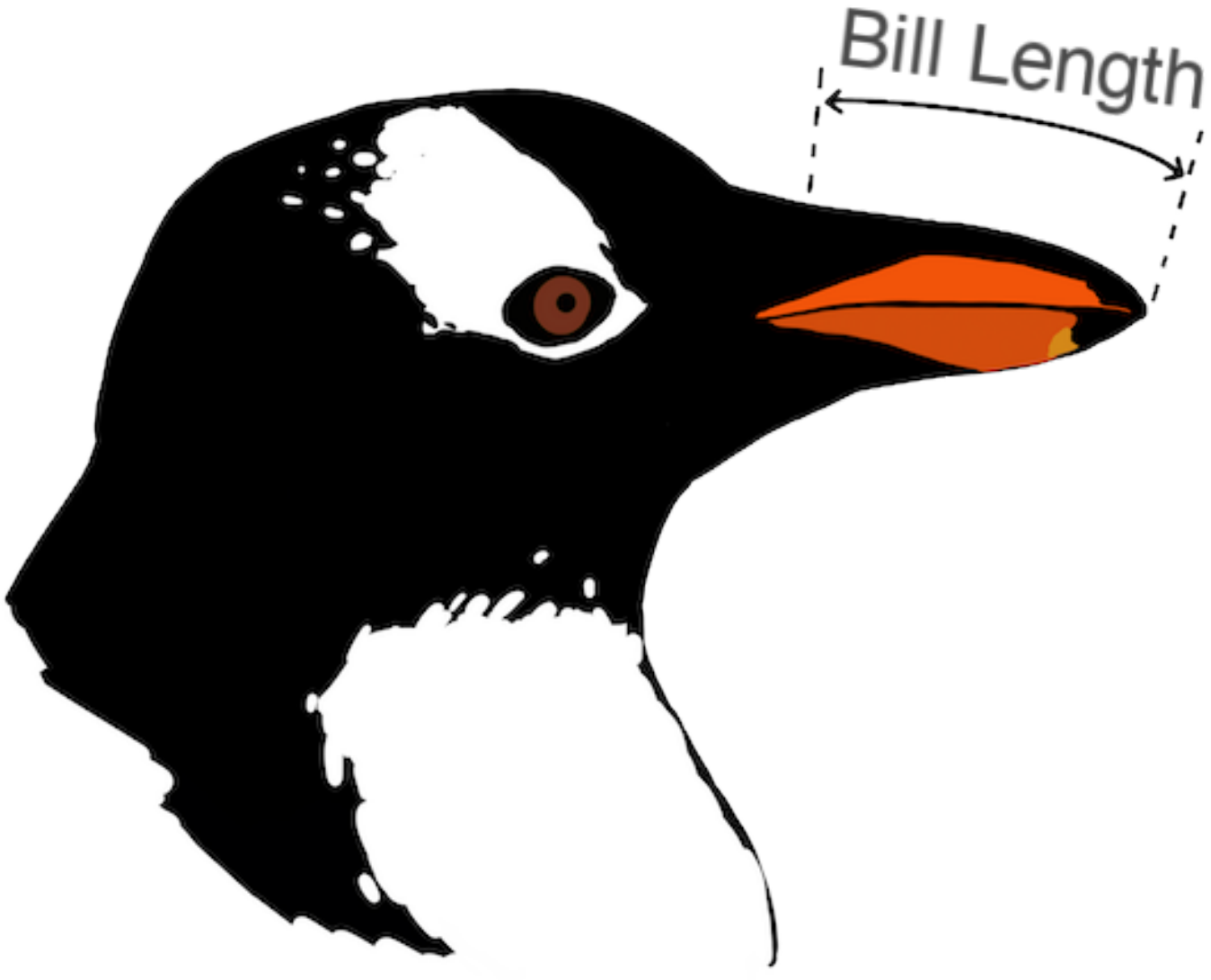
```
1 import pandas as pd
2 import numpy as np
3
4 penguins = pd.read_parquet("data/penguins.parquet")
5 penguins.sample(5)
```

The Data

```python
import pandas as pd
import numpy as np

penguins = pd.read_parquet("data/penguins.parquet")
penguins.sample(5)
```

species	bill_length	sex
Gentoo	46.2	female
Gentoo	47.2	female
Chinstrap	54.2	male
Adelie	37.0	female
Gentoo	45.3	female

Bill Length



One Boxplot

One Boxplot

```
1 from plotnine import *
2
3 (ggplot(penguins, aes(y="bill_length"))
4  + geom_boxplot()
5 )
```

One Boxplot

```
from plotnine import *

(ggplot(penguins, aes(y="bill_length"))
 + geom_boxplot()
)
```

One Boxplot

```
1 from plotnine import *
2
3 (ggplot(penguins, aes(y="bill_length"))
4  + geom_boxplot()
5 )
```

One Boxplot

```
from plotnine import *

(ggplot(penguins, aes(y="bill_length"))
 + geom_boxplot()
)
```

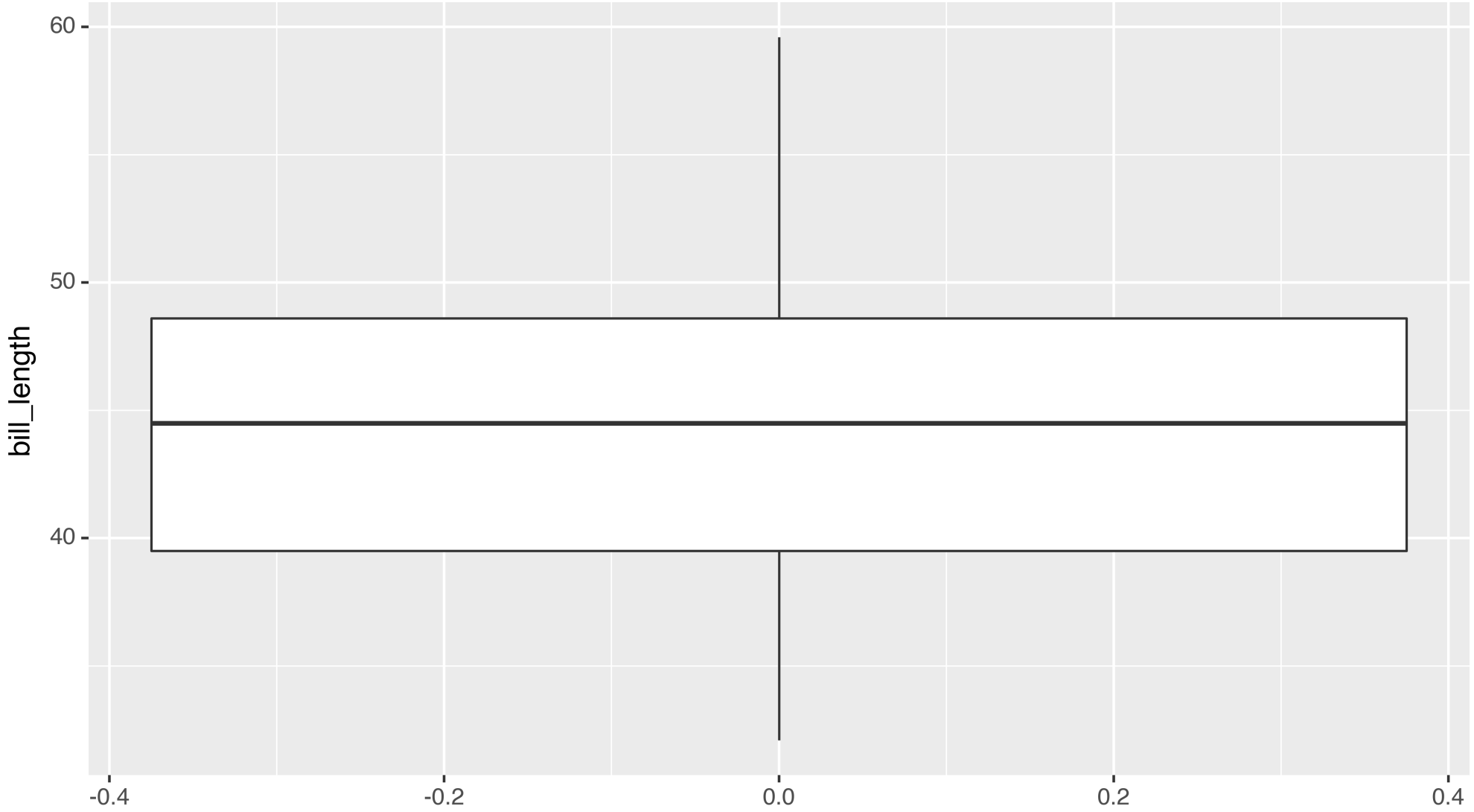
One Boxplot

```
1 from plotnine import *
2
3 (ggplot(penguins, aes(y="bill_length"))
4  + geom_boxplot()
5 )
```

One Boxplot

```
1 from plotnine import *
2
3 (ggplot(penguins, aes(y="bill_length"))
4  + geom_boxplot()
5 )
```

One Boxplot



Boxplot (by species)

Boxplot (by species)

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

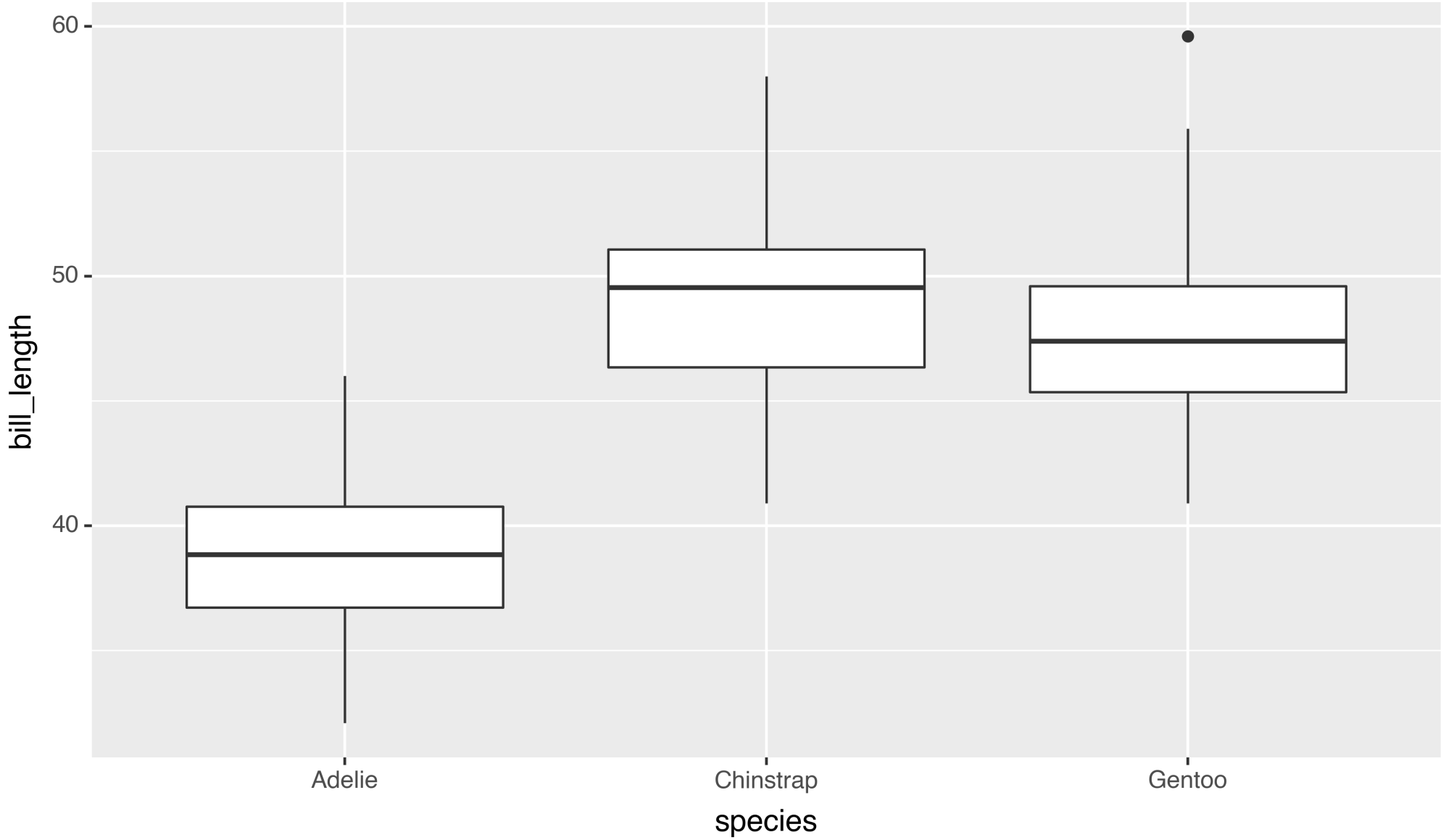
Boxplot (by species)

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

Boxplot (by species)

```
(ggplot(penguins, aes("species", "bill_length"))
 + geom_boxplot()
)
```

Boxplot (by species)



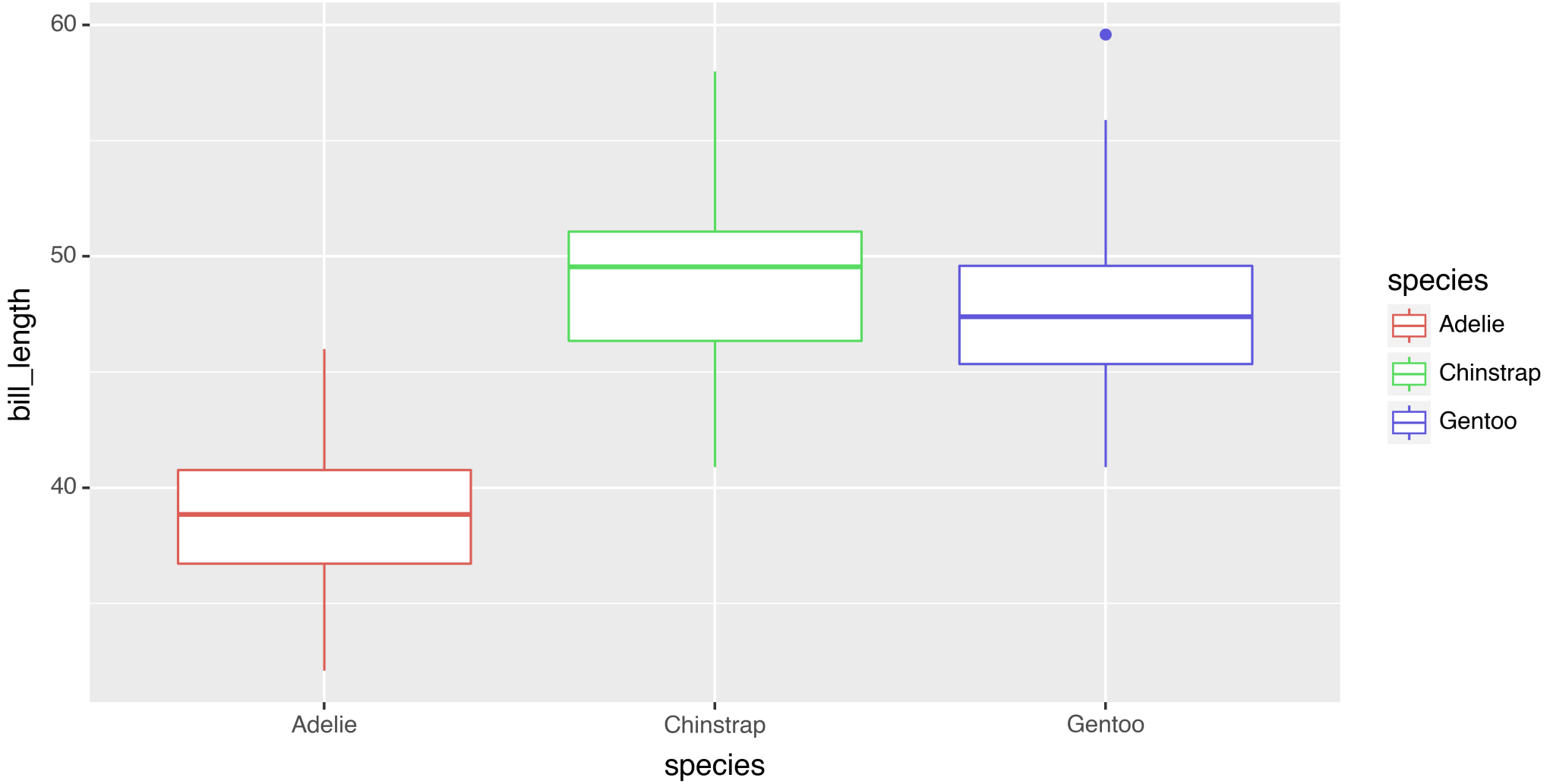
Boxplot + color

```
1 (ggplot(penguins, aes("species", "bill_length", color="species"))
2  + geom_boxplot()
3 )
```

Boxplot + color

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_boxplot()
)
```

Boxplot + `color`



Boxplot + mean

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_boxplot()
)
```

Boxplot + mean

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_boxplot()
)
```

Boxplot + mean

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_boxplot()
)
```

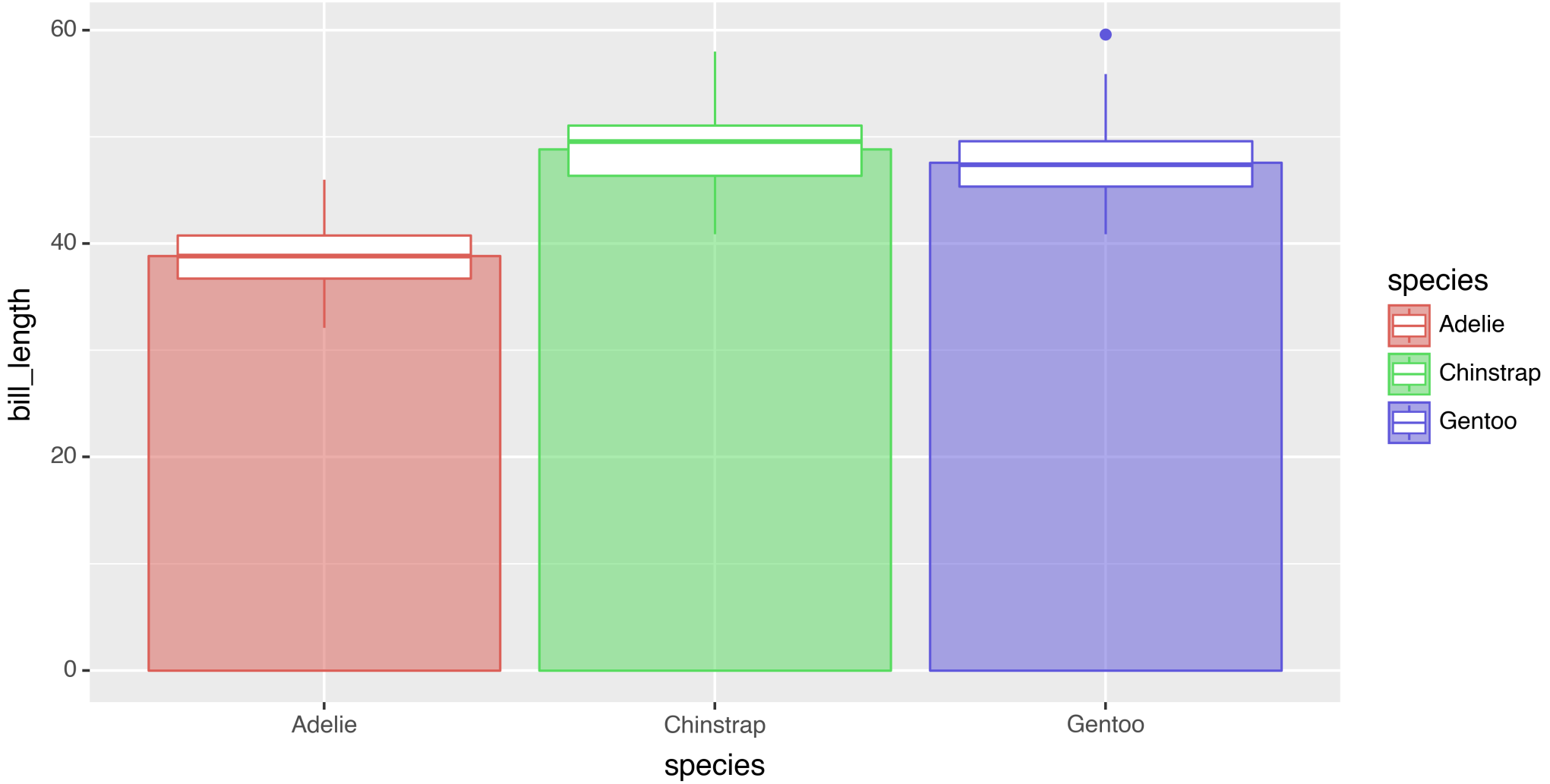
Boxplot + mean

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_boxplot()
)
```

Boxplot + mean

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_boxplot()
)
```

Boxplot + mean



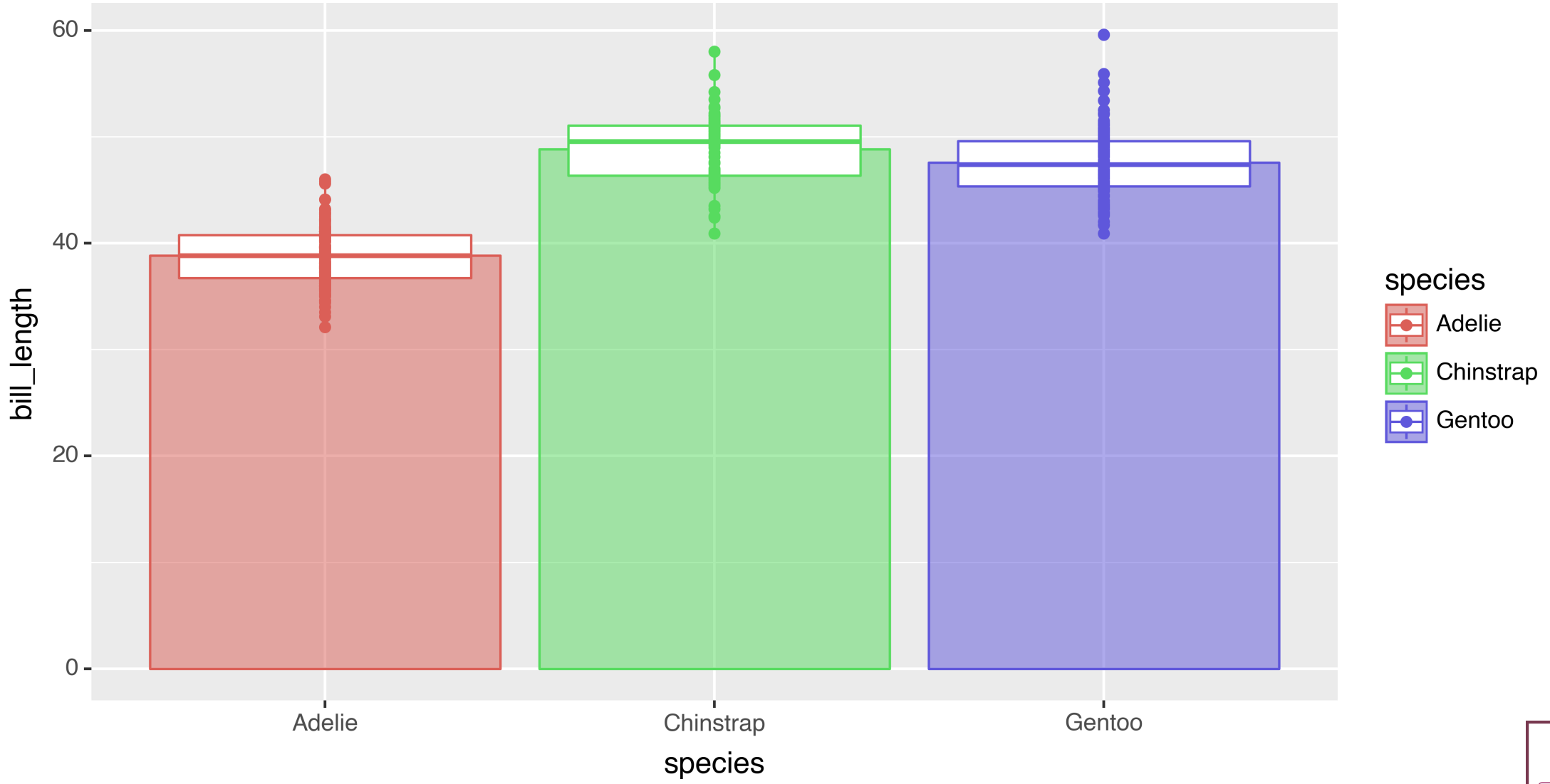
Boxplot + mean & points

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_boxplot()
 + geom_point()
)
```

Boxplot + mean & points

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_boxplot()
 + geom_point()
)
```

Boxplot + `mean & points`



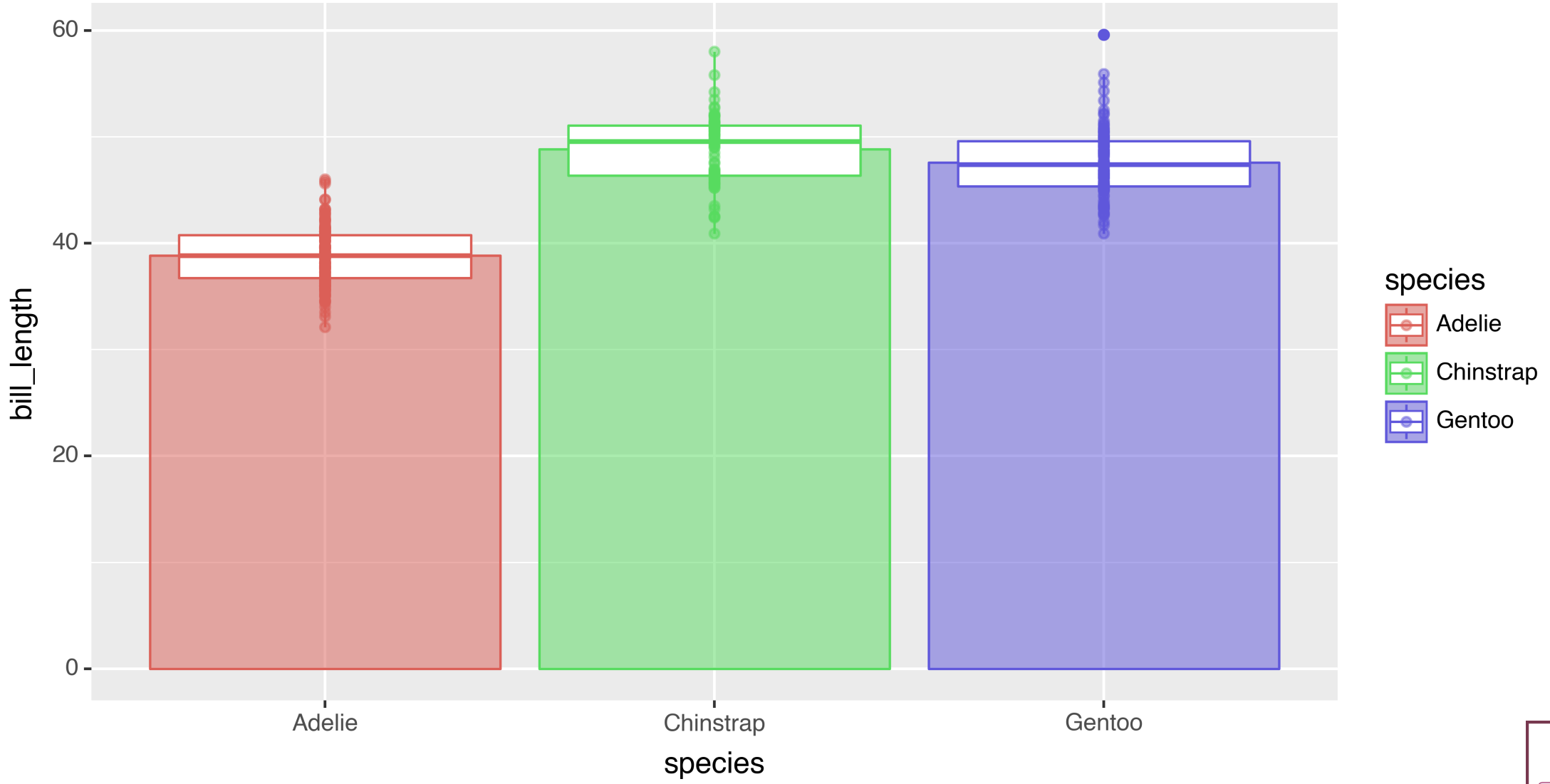
Boxplot + mean & alpha points

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=0.5)
 + geom_boxplot()
 + geom_point(alpha=0.5)
)
```

Boxplot + mean & alpha points

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=0.5)
 + geom_boxplot()
 + geom_point(alpha=0.5)
)
```

Boxplot + `mean & alpha points`



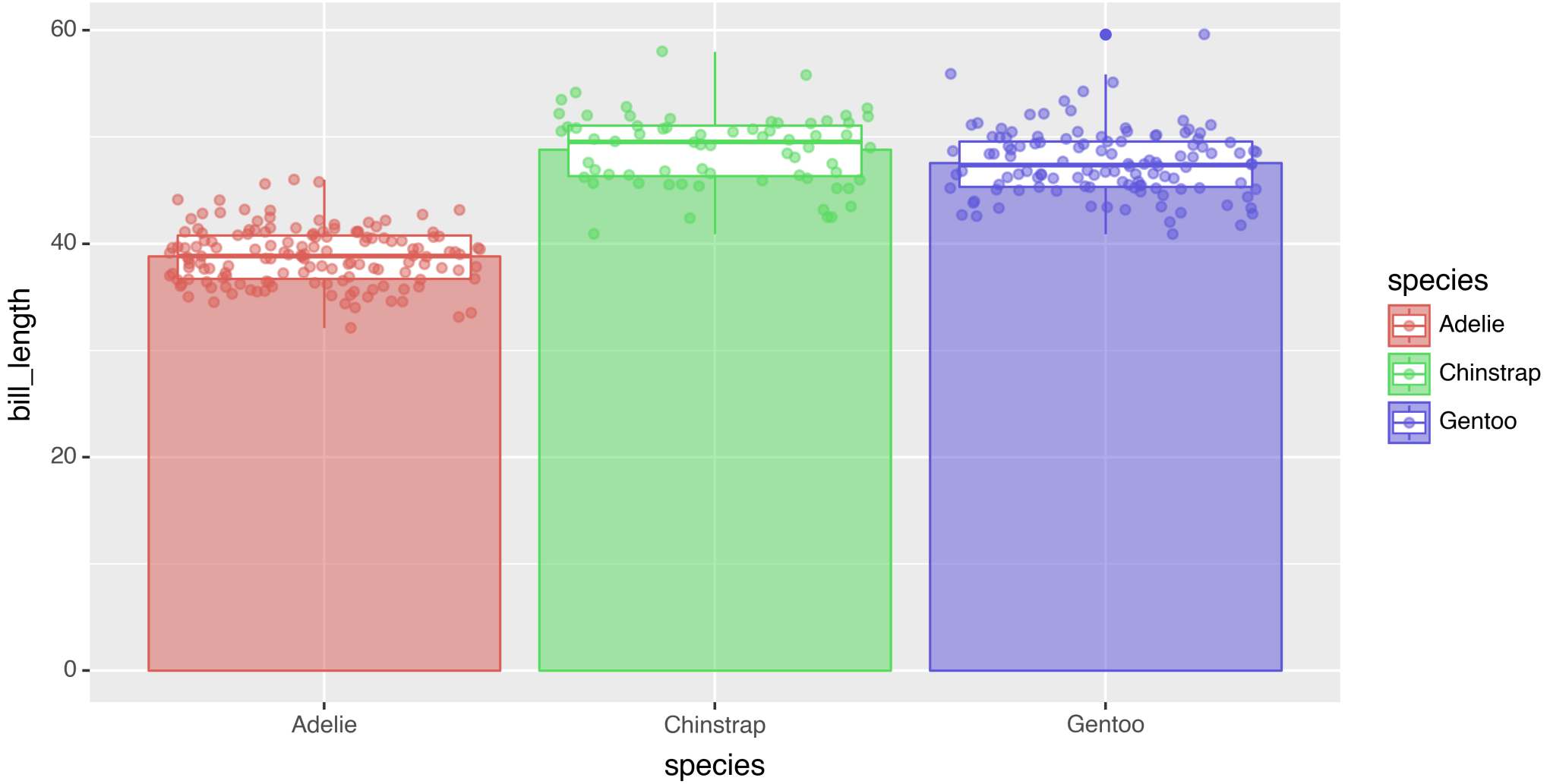
Boxplot + mean & jittered alpha points

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_boxplot()
 + geom_point(alpha=0.5, position="jitter")
)
```

Boxplot + mean & jittered alpha points

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_boxplot()
 + geom_point(alpha=0.5, position="jitter")
)
```

Boxplot + mean & jittered alpha points



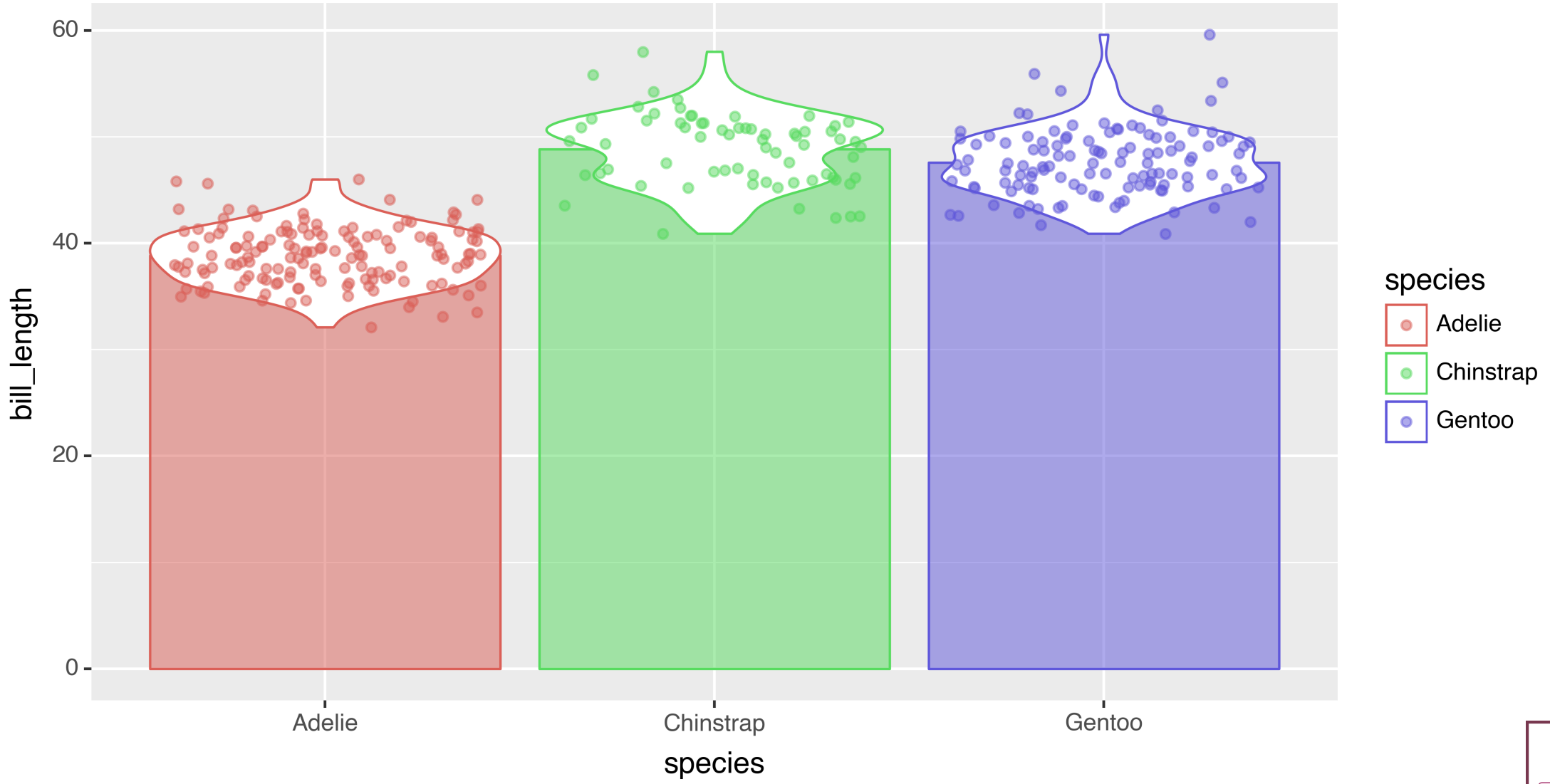
Violinplot + mean & jittered alpha points

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_violin()
 + geom_point(alpha=0.5, position="jitter")
)
```

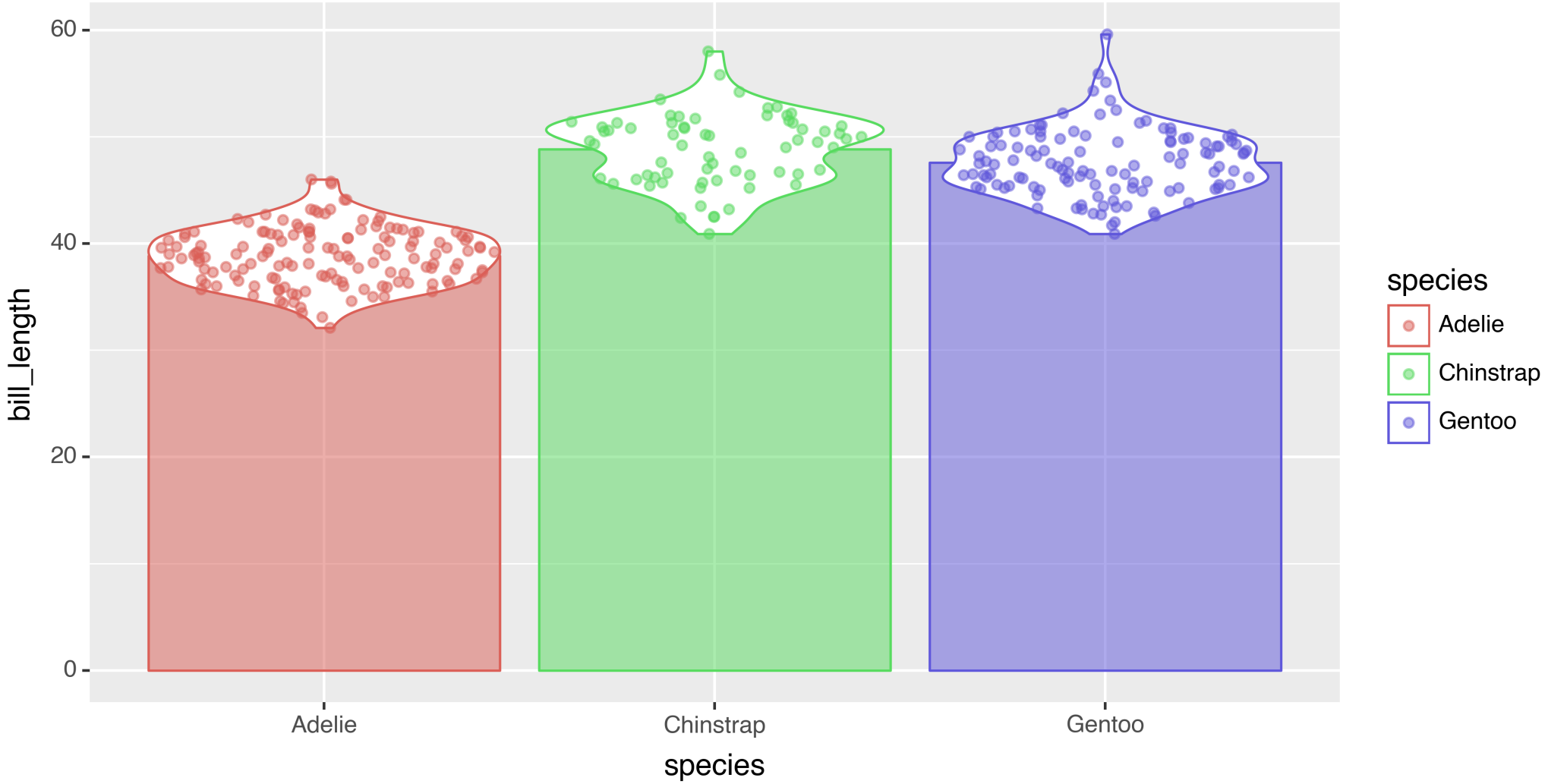
Violinplot + mean & jittered alpha points

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_violin()
 + geom_point(alpha=0.5, position="jitter")
)
```

Violinplot + `mean & jittered alpha points`



Restrict the points to within the violin



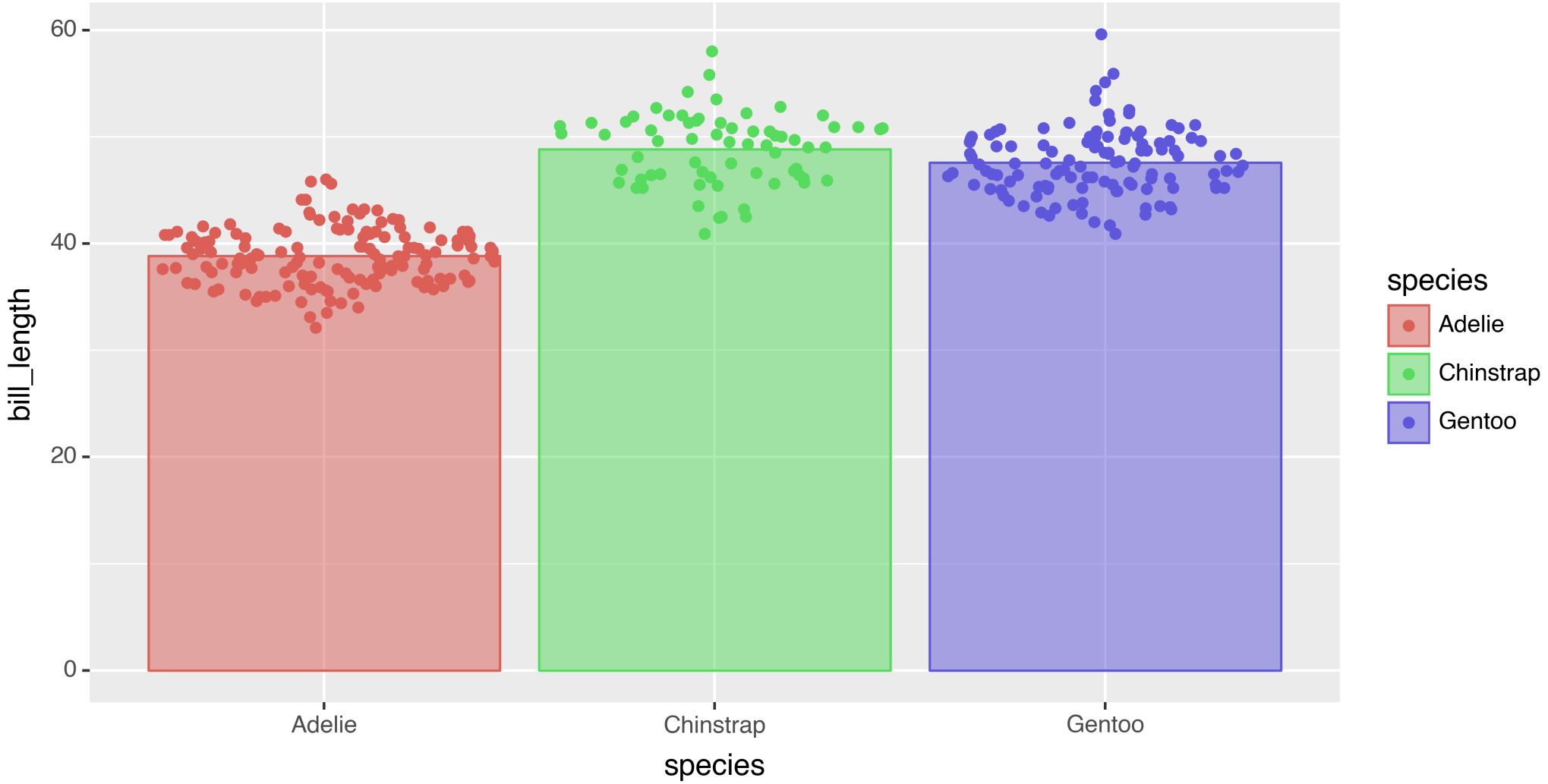
Sinaplot + `mean column`

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_sina()
)
```

Sinaplot + `mean column`

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_sina()
)
```

Sinaplot + `mean column`



Sinaplot + `mean_column & mean_segment`

Sinaplot + mean column & mean segment

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

Sinaplot + mean column & mean segment

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

Sinaplot + mean column & mean segment

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

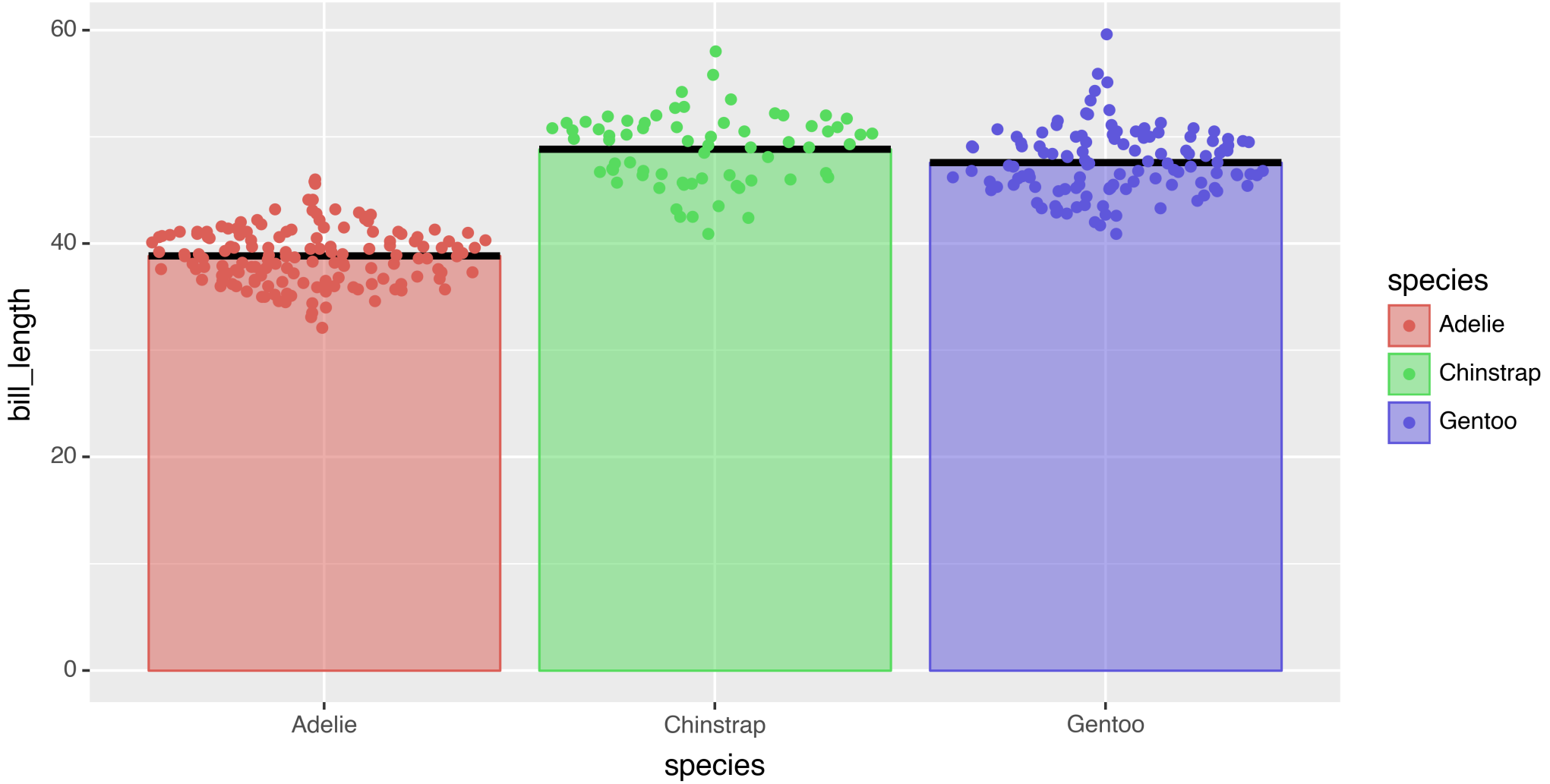
Sinaplot + mean column & mean segment

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

Sinaplot + mean column & mean segment

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_col(aes(fill="species"), stat="summary", fun_y=np.mean, alpha=.5)
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

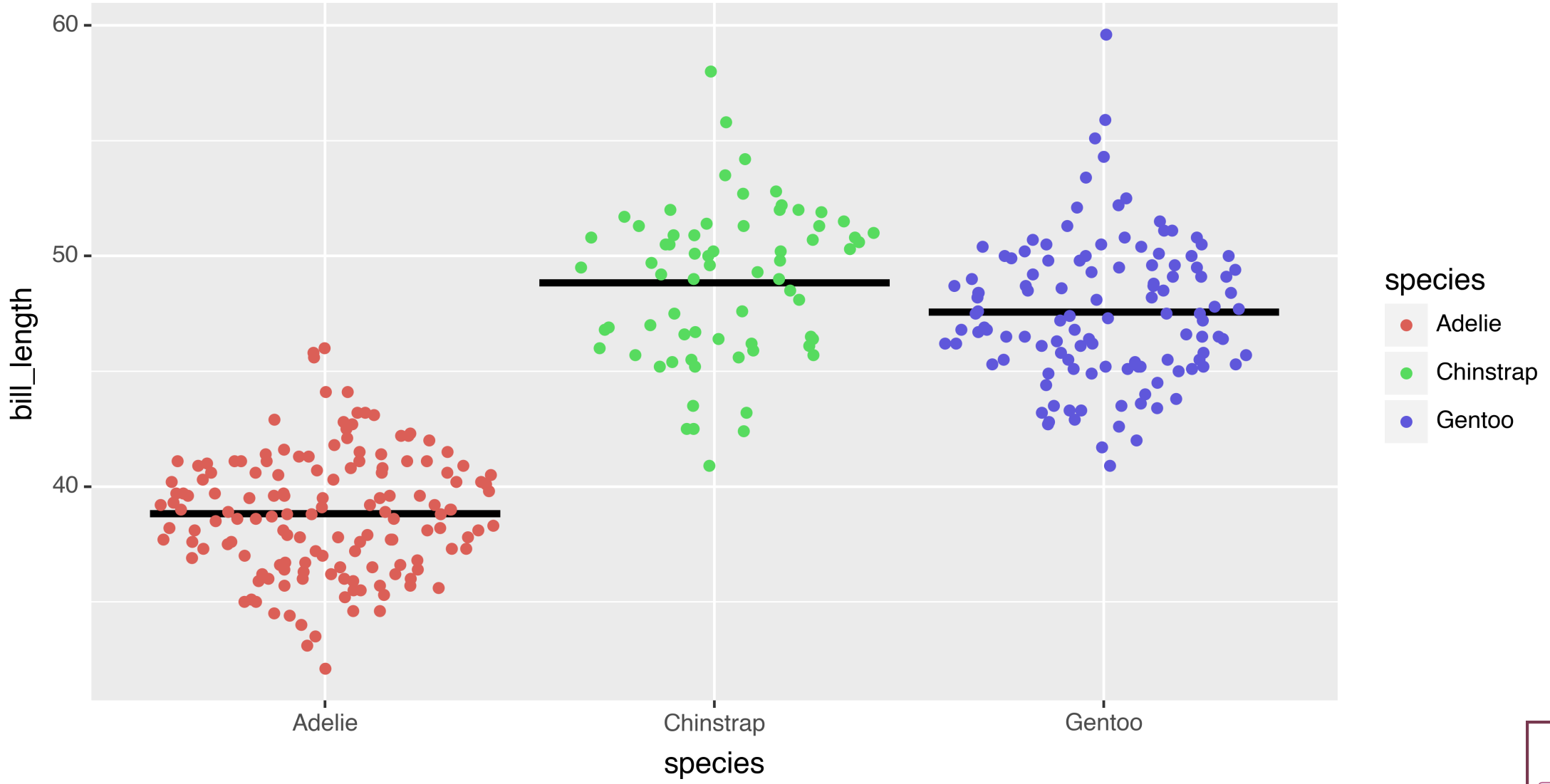
Sinaplot + `mean column & mean segment`



Sinaplot + mean

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

Sinaplot + `mean`



Sinaplot (per sub-group) + `mean`

Sinaplot (per sub-group) + mean

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

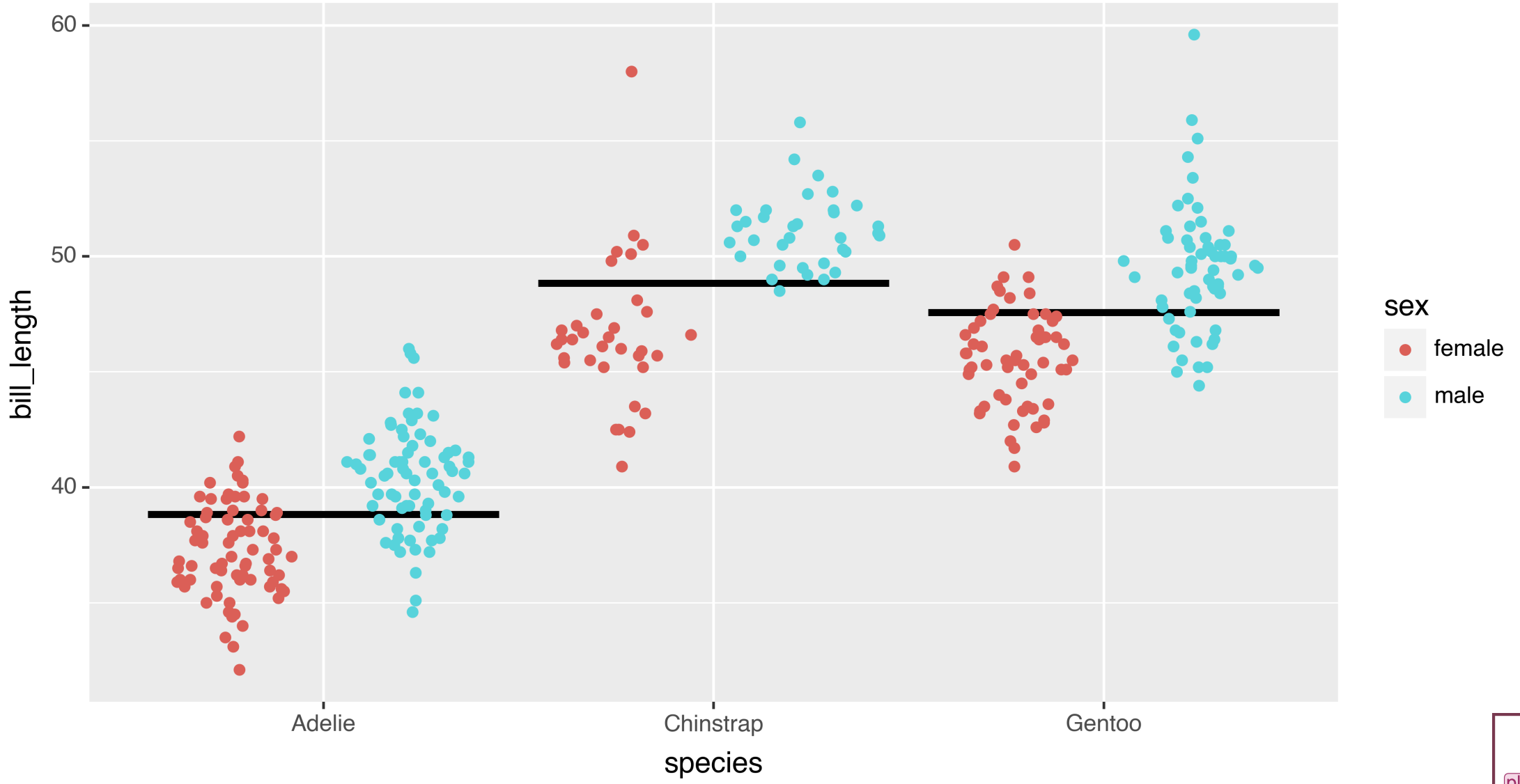
Sinaplot (per sub-group) + mean

```
(ggplot(penguins, aes("species", "bill_length", color="species"))
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

Sinaplot (per sub-group) + mean

```
(ggplot(penguins, aes("species", "bill_length", color="sex"))
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

Sinaplot (per sub-group) + `mean`



Sinaplot + `global & sub-group means I`

Sinaplot + global & sub-group means I

```
(ggplot(penguins, aes("species", "bill_length", color="sex"))
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

Sinaplot + global & sub-group means I

```
(ggplot(penguins, aes("species", "bill_length", color="sex"))
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_sina()
)
```

Sinaplot + global & sub-group means I

```
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
```

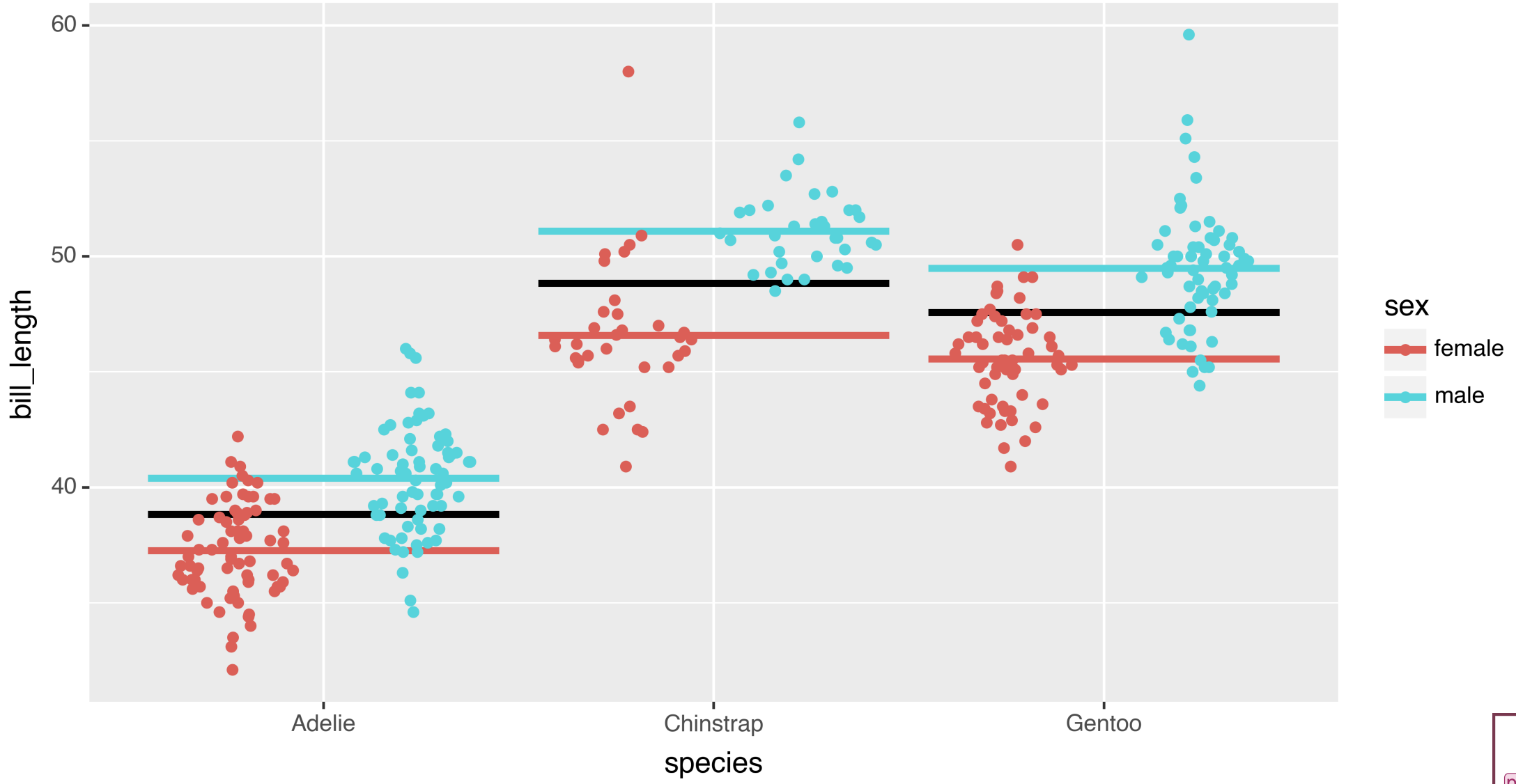
Sinaplot + global & sub-group means I

```
  + geom_segment(
      aes(
          x=stage("species", after_stat="x-.45"),
          xend=stage("species", after_stat="x+.45"),
          yend=after_stat("y"),
      ),
      stat="summary",
      fun_y=np.mean,
      color="black",
      size=1.5
  )
  + geom_segment(
      aes(
          x=stage("species", after_stat="x-.45"),
          xend=stage("species", after_stat="x+.45"),
          yend=after_stat("y"),
      ),
      stat="summary",
      fun_y=np.mean,
      color="black",
      size=1.5
  )
```

Sinaplot + global & sub-group means I

```
(ggplot(penguins, aes("species", "bill_length", color="sex"))
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     color="black",
     size=1.5
 )
 + geom_segment(
     aes(
         x=stage("species", after_stat="x-.45"),
         xend=stage("species", after_stat="x+.45"),
         yend=after_stat("y"),
     ),
     stat="summary",
     fun_y=np.mean,
     size=1.5
 )
 + geom_sina()
```

Sinaplot + `global & sub-group means I`



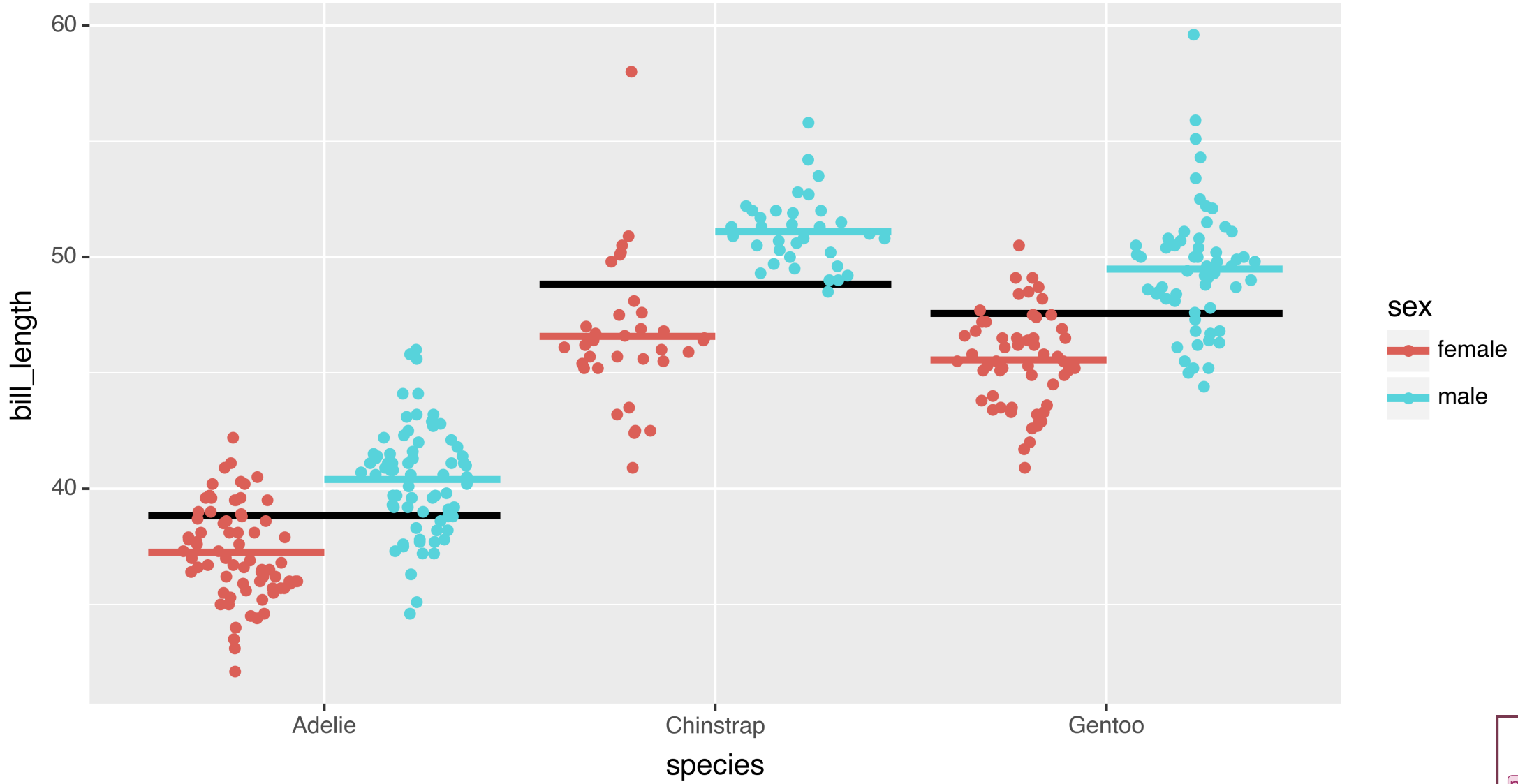
Sinaplot + global & sub-group means II

```
        ),
        stat="summary",
        fun_y=np.mean,
        color="black",
        size=1.5
    )
    + geom_segment(
        aes(
            x=stage("species", after_stat="x-.45"),
            xend=stage("species", after_stat="x+.45"),
            yend=after_stat("y"),
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_sina()
)
```

Sinaplot + global & sub-group means II

```
        ),
        stat="summary",
        fun_y=np.mean,
        color="black",
        size=1.5
    )
    + geom_segment(
        aes(
            x=stage("species", after_stat="x-.45"),
            xend=stage("species", after_stat="x+.45"),
            yend=after_stat("y"),
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_sina()
)
```

Sinaplot + global & sub-group means II



Sinaplot + global & sub-group means & counts

```
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
            label=after_stat("n")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_ymin=np.min,
        format_string="n = {}",
        size=9,
    )
    + geom_sina()
)
```

Sinaplot + global & sub-group means & counts

```
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
            label=after_stat("n")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_ymin=np.min,
        format_string="n = {}",
        size=9,
    )
    + geom_sina()
)
```

Sinaplot + global & sub-group means & counts

```
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
            label=after_stat("n")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_ymin=np.min,
        format_string="n = {}",
        size=9,
    )
    + geom_sina()
)
```

Sinaplot + global & sub-group means & counts

```
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
            label=after_stat("n")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_ymin=np.min,
        format_string="n = {}",
        size=9,
    )
    + geom_sina()
)
```

Sinaplot + global & sub-group means & counts

```
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
            label=after_stat("n")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_ymin=np.min,
        format_string="n = {}",
        size=9,
    )
    + geom_sina()
)
```

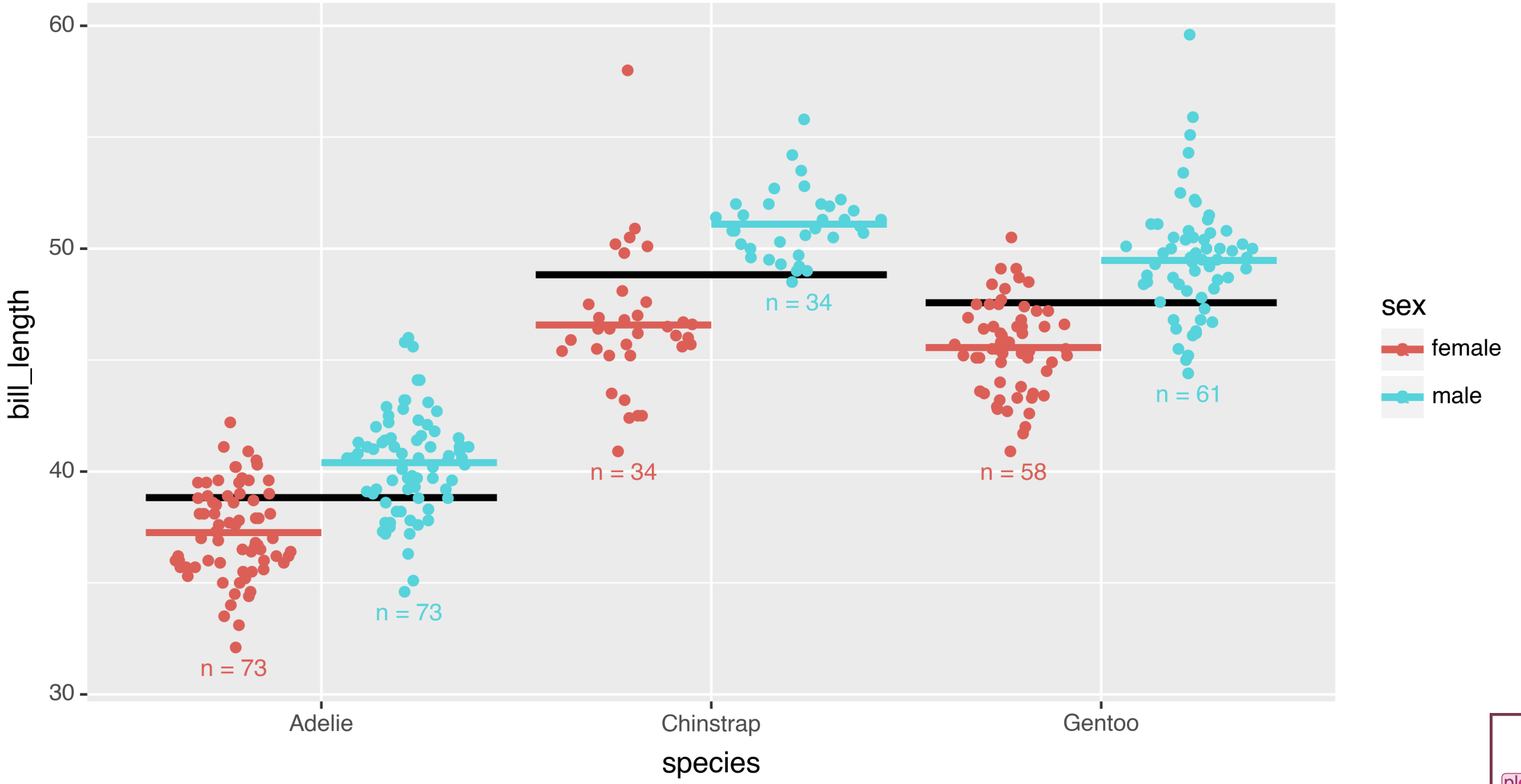
Sinaplot + global & sub-group means & counts

```
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
            label=after_stat("n")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_ymin=np.min,
        format_string="n = {}",
        size=9,
    )
    + geom_sina()
)
```

Sinaplot + global & sub-group means & counts

```
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
            label=after_stat("n")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_ymin=np.min,
        format_string="n = {}",
        size=9,
    )
    + geom_sina()
)
```

Sinaplot + global & sub-group means & counts



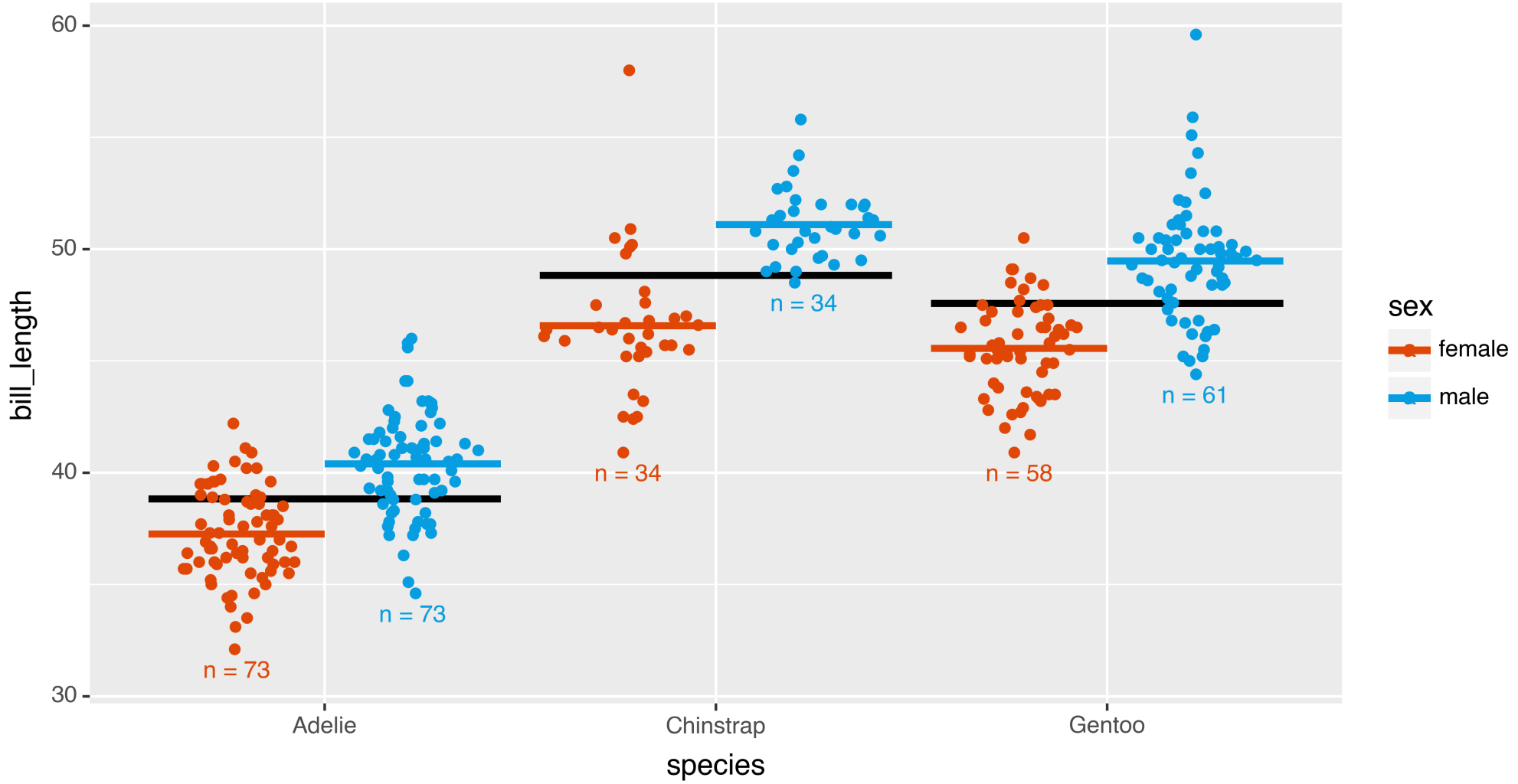
Vibrant Sinaplot

```
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
            label=after_stat("n")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_ymin=np.min,
        format_string="n = {}",
        size=9,
    )
    + geom_sina()
    + scale_color_hue(h=0.05, s=0.95, l=0.45)
)
```

Vibrant Sinaplot

```
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
            label=after_stat("n")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_ymin=np.min,
        format_string="n = {}",
        size=9,
    )
    + geom_sina()
    + scale_color_hue(h=0.05, s=0.95, l=0.45)
)
```

Vibrant Sinaplot

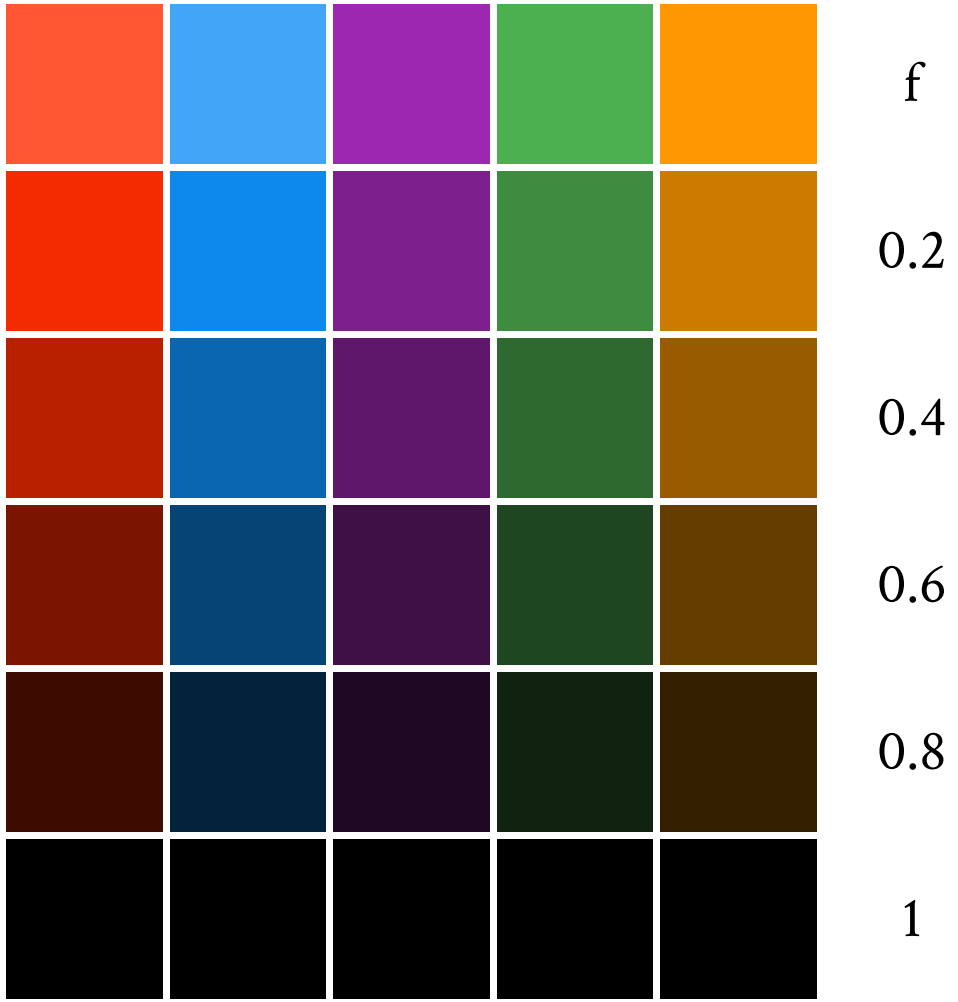


darken function

```
def darken(hex_colors, f):
    """
    Make colors darker by a factor f
    """
    from colorsys import rgb_to_hls, hls_to

    def _hex_to_int(hx):
        return int(hx, 16) / 255

    def _hex_to_rgb(hex_color):
        hxs = (hex_color[i:i+2] for i in (1
        return tuple(_hex_to_int(hx) for hx

    def _rgb_to_hex(rgb_color):
         ints = (round(c * 255) for c in rg
         hxs = ("{:#04X}".format(i)[2:] for
         return f"#{''.join(hxs)}"

    def _darken(hex_color):
```

darken function

```
def darken(hex_colors, f):
    """
    Make colors darker by a factor f
    """
    from colorsys import rgb_to_hls, hls_to

    def _hex_to_int(hx):
        return int(hx, 16) / 255

    def _hex_to_rgb(hex_color):
        hxs = (hex_color[i:i+2] for i in (1
        return tuple(_hex_to_int(hx) for hx

    def _rgb_to_hex(rgb_color):
        ints = (round(c * 255) for c in rg
        hxs = ("{:#04X}".format(i)[2:] for
        return f"#{''.join(hxs)}"

    def _darken(hex_color):
```



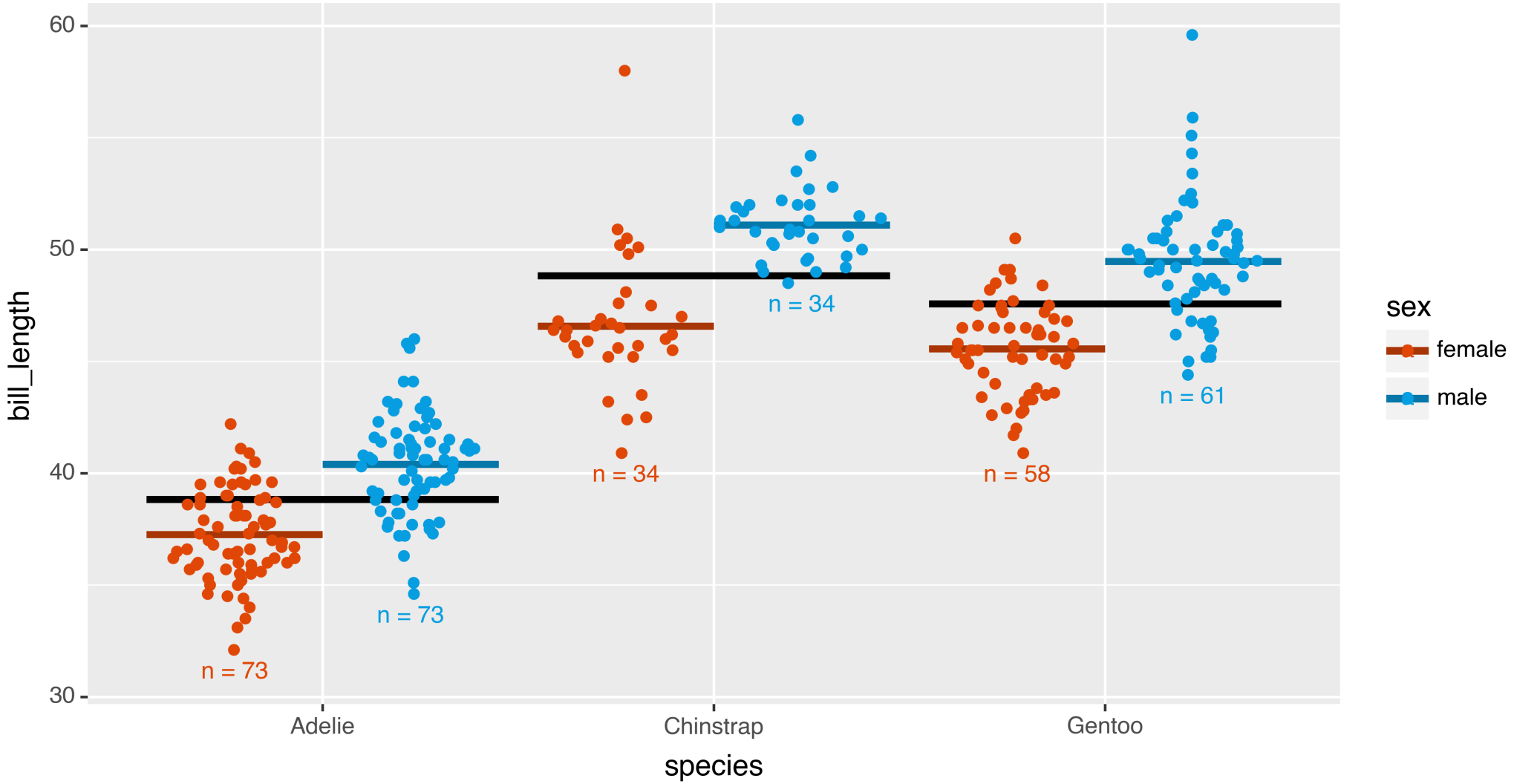
Vibrant Sinaplot + darker lines

```
        fun_y=np.mean,
        color="black",
        size=1.5
    )
    + geom_segment(
        aes(
            x=stage("species", after_stat="x-.45"),
            xend=stage("species", after_stat="x+.45"),
            yend=after_stat("y"),
            color=stage("sex", after_scale="darken(color, .25)")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
```

Vibrant Sinaplot + darker lines

```
        fun_y=np.mean,
        color="black",
        size=1.5
    )
    + geom_segment(
        aes(
            x=stage("species", after_stat="x-.45"),
            xend=stage("species", after_stat="x+.45"),
            yend=after_stat("y"),
            color=stage("sex", after_scale="darken(color, .25)")
        ),
        stat="summary",
        position=position_dodge(width=0.9),
        fun_y=np.mean,
        size=1.5
    )
    + geom_text(
        aes(
            y=stage("bill_length", after_stat="ymin-1"),
```

Vibrant Sinaplot + `darker lines`



Part 3: What Next

What comes after familiarity?

The Board, The Strings, The Math

Chord Data

String	Fret
1	-1
2	3
3	2
4	0
5	1
6	0

Plottable Chord Data

```
def as_plottable_chord(chord):
    """Make chord plottable"""
    df = chord.copy()
    fret = df["Fret"]
    df["state"] = "on"
    df.loc[fret == 0, "state"] = "open"
    df.loc[fret == -1, "state"] = "clos
    df["Fret"] = (fret - 0.5).clip(lowe

    # Sequence based on no. of notes in
    df["sequence"] = range(1, len(df)+1
    return df

data = as_plottable_chord(c_chord)
data
```

Plottable Chord Data

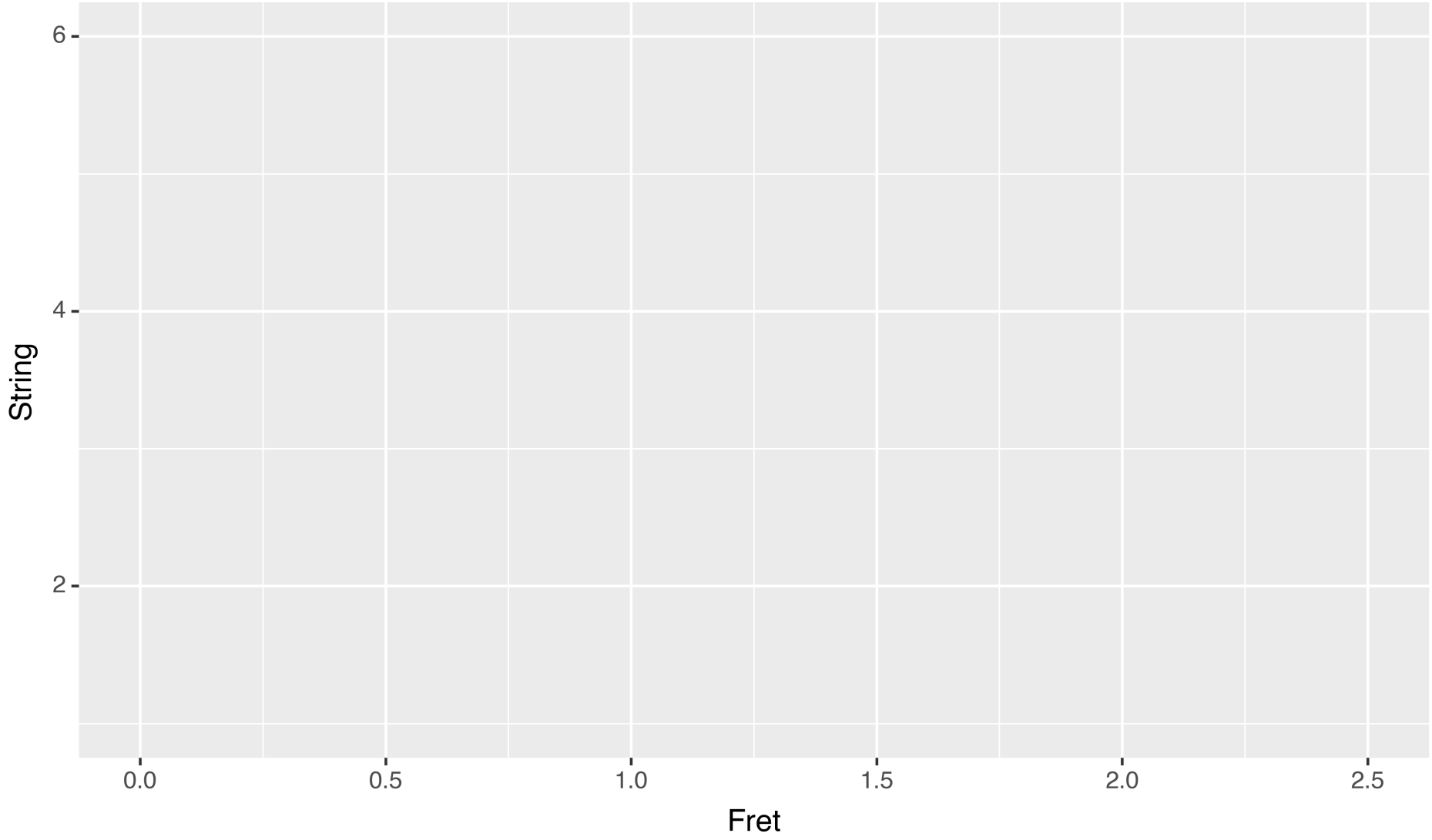
```
def as_plottable_chord(chord):
    """Make chord plottable"""
    df = chord.copy()
    fret = df["Fret"]
    df["state"] = "on"
    df.loc[fret == 0, "state"] = "open"
    df.loc[fret == -1, "state"] = "clos
    df["Fret"] = (fret - 0.5).clip(lowe

    # Sequence based on no. of notes in
    df["sequence"] = range(1, len(df)+1
    return df

data = as_plottable_chord(c_chord)
data
```

String	Fret	state	sequence
1	0.0	close	1
2	2.5	on	2
3	1.5	on	3
4	0.0	open	4
5	0.5	on	5
6	0.0	open	6

Fretboard

```
1 (ggplot(data, aes("Fret", "String", color="sequence"))
2 )
```

Fretboard



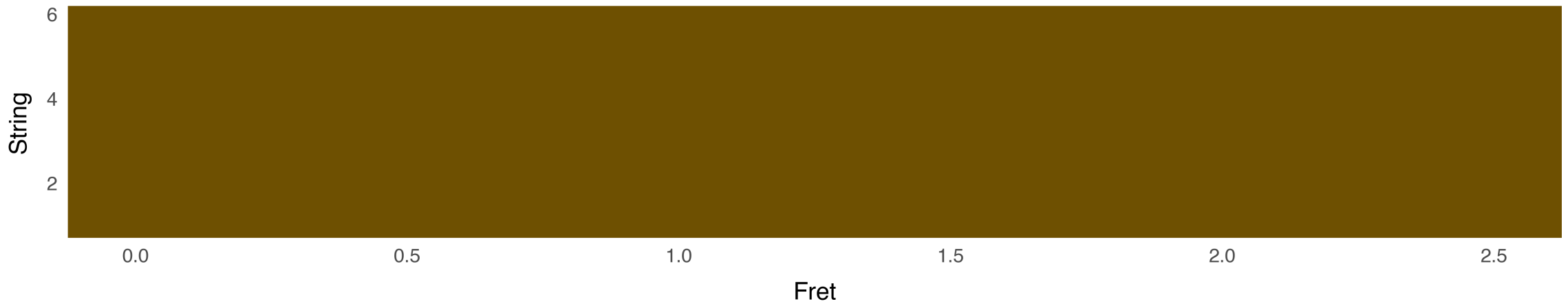
Fretboard + theme

```
fretboard_theme = theme_gray() + theme(
    figure_size=(10, 2),
    panel_background=element_rect(fill="#6e5001"),
    panel_grid=element_blank(),
)

(ggplot(data, aes("Fret", "String", color="sequence"))
 + fretboard_theme
)
```

Fretboard + theme

```
fretboard_theme = theme_gray() + theme(
    figure_size=(10, 2),
    panel_background=element_rect(fill="#6e5001"),
    panel_grid=element_blank(),
)

(ggplot(data, aes("Fret", "String", color="sequence"))
 + fretboard_theme
)
```

Fretboard + `theme`



Fretboard + theme, annotations

```
from mizani.bounds import rescale

# Standard markings for a Stratocaster
mx = [2.5, 4.5, 6.5, 8.5, 11.5, 11.5, 14.5, 16.5, 18.5, 20.5]
my = [3.5, 3.5, 3.5, 3.5, 2, 5, 3.5, 3.5, 3.5, 3.5]

# 10 Gauge (Medium String)
gauge = np.array([.010, .014, .017, .026, .036, .046])
gauge = rescale(gauge, (.6, 1.5))

fretboard_annotations = [
  annotate("point", x=mx, y=my, fill="black", color="gray", size=5, stroke=.2),
  annotate("vline", xintercept=range(0, 23), color="black", size=1.5*1.8),
  annotate("vline", xintercept=range(0, 23), color="lightgray", size=1.5),
  annotate("hline", yintercept=range(1, 7), color="gray", size=gauge)
]

(ggplot(data, aes("Fret", "String", color="sequence"))
 + fretboard_theme
```

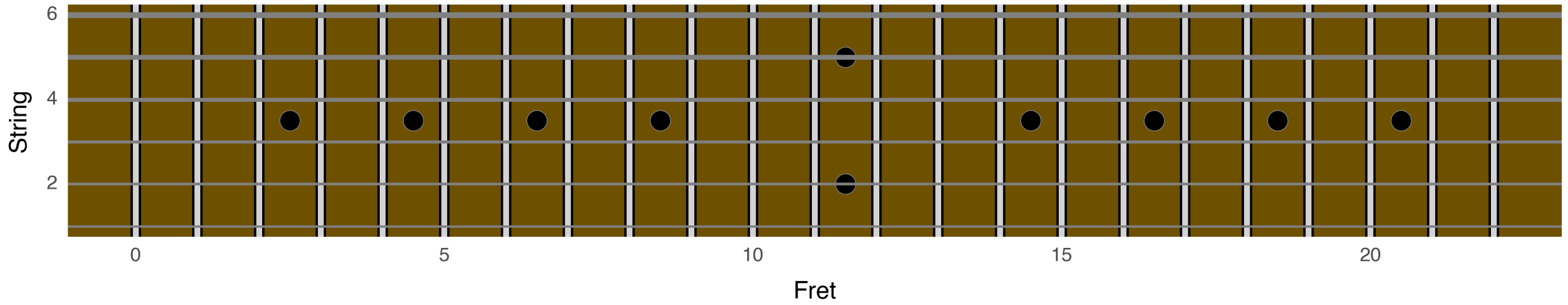
Fretboard + theme, annotations

```
# Standard markings for a Stratocaster
mx = [2.5, 4.5, 6.5, 8.5, 11.5, 11.5, 14.5, 16.5, 18.5, 20.5]
my = [3.5, 3.5, 3.5, 3.5, 2, 5, 3.5, 3.5, 3.5, 3.5]

# 10 Gauge (Medium String)
gauge = np.array([.010, .014, .017, .026, .036, .046])
gauge = rescale(gauge, (.6, 1.5))

fretboard_annotations = [
  annotate("point", x=mx, y=my, fill="black", color="gray", size=5, stroke=.2),
  annotate("vline", xintercept=range(0, 23), color="black", size=1.5*1.8),
  annotate("vline", xintercept=range(0, 23), color="lightgray", size=1.5),
  annotate("hline", yintercept=range(1, 7), color="gray", size=gauge)
]

(ggplot(data, aes("Fret", "String", color="sequence"))
 + fretboard_theme
 + fretboard_annotations
)
```

Fretboard + theme, annotations

```
# Standard markings for a Stratocaster
mx = [2.5, 4.5, 6.5, 8.5, 11.5, 11.5, 14.5, 16.5, 18.5, 20.5]
my = [3.5, 3.5, 3.5, 3.5, 2, 5, 3.5, 3.5, 3.5, 3.5]

# 10 Gauge (Medium String)
gauge = np.array([.010, .014, .017, .026, .036, .046])
gauge = rescale(gauge, (.6, 1.5))

fretboard_annotations = [
  annotate("point", x=mx, y=my, fill="black", color="gray", size=5, stroke=.2),
  annotate("vline", xintercept=range(0, 23), color="black", size=1.5*1.8),
  annotate("vline", xintercept=range(0, 23), color="lightgray", size=1.5),
  annotate("hline", yintercept=range(1, 7), color="gray", size=gauge)
]

(ggplot(data, aes("Fret", "String", color="sequence"))
 + fretboard_theme
 + fretboard_annotations
)
```

Fretboard + `theme, annotations`



A Custom Scale

A Custom Scale

Domain

A Custom Scale

Domain

$$(0, \text{number\_of\_frets})$$

A Custom Scale

Domain

$$(0, 22)$$

A Custom Scale

Domain

$$(0, 22)$$

Transformation

$$25.5 - \frac{25.5}{2^{\frac{x}{12}}}$$

A Custom Scale

Domain

$$(0, 22)$$

Transformation

$$25.5 - \frac{25.5}{2^{\frac{x}{12}}}$$

Inverse

$$12 \log_2(\frac{25.5}{25.5 - x})$$

A Custom Scale

Domain

$$(0, 22)$$

Transformation

$$25.5 - \frac{25.5}{2^{\frac{x}{12}}}$$

Inverse

$$12 \log_2\left(\frac{25.5}{25.5 - x}\right)$$

Breaks

$$(0, 22)$$

A trans class

```
    """
    Frets Transformation
    """
    number_of_frets = 22
    domain = (0, number_of_frets)

    @staticmethod
    def transform(x):
        x = np.asarray(x)
        return -(25.5 - (25.5 / (2 ** (x/12))))

    @staticmethod
    def inverse(x):
        x = -np.asarray(x)
        return (12 * np.log2(25.5/(25.5-x)))

    @classmethod
    def breaks_(cls, limits):
        return cls.domain
```

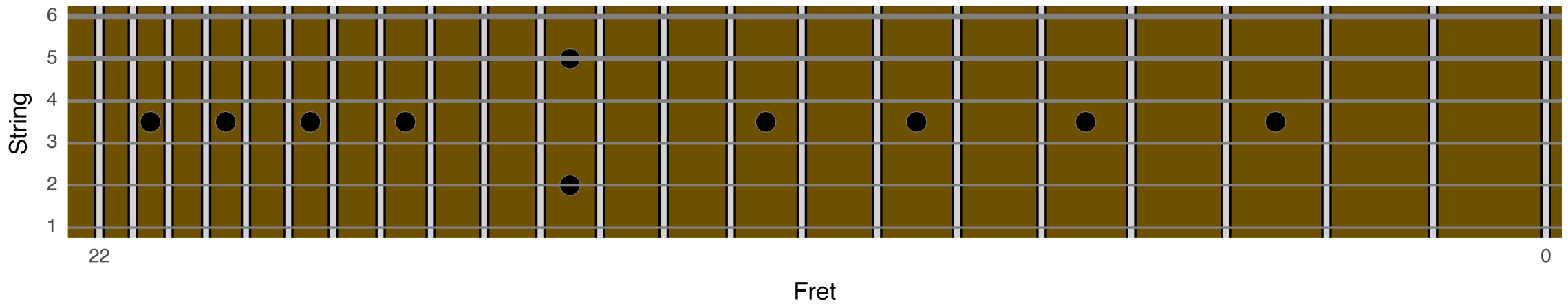
Fret Board + theme, annotations, scales

```
fretboard_scales = [
    scale_x_continuous(trans=frets_trans),
    scale_y_continuous(breaks=range(1, 7)),
    scale_fill_manual(values=["red", "lime", "white"]),
    guides(color=False, fill=False)
]

(ggplot(data, aes("Fret", "String", color="sequence"))
 + fretboard_theme
 + fretboard_annotations
 + fretboard_scales
)
```

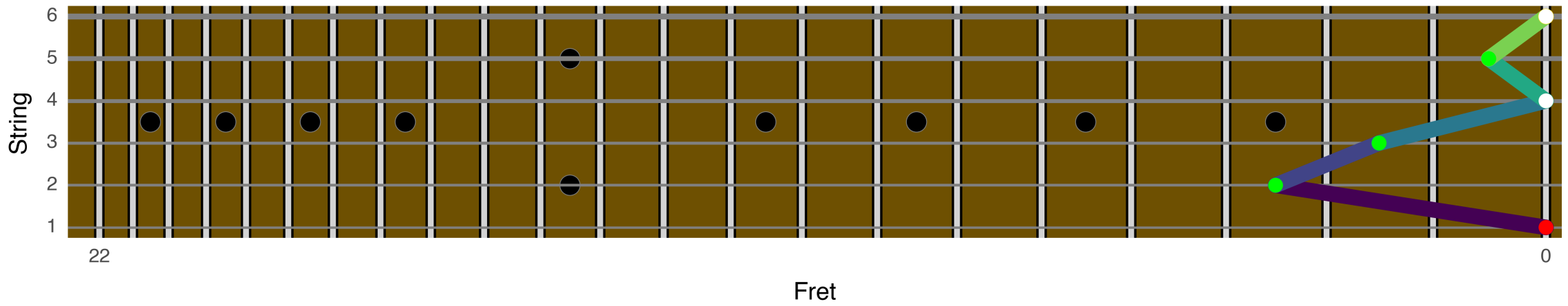
Fret Board + theme, annotations, scales

```
fretboard_scales = [
    scale_x_continuous(trans=frets_trans),
    scale_y_continuous(breaks=range(1, 7)),
    scale_fill_manual(values=["red", "lime", "white"]),
    guides(color=False, fill=False)
]

(ggplot(data, aes("Fret", "String", color="sequence"))
 + fretboard_theme
 + fretboard_annotations
 + fretboard_scales
)
```

Fret Board + `theme, annotations, scales`



Fretboard Chord



Thank You