



DATA VISUALIZATION WITH GGPLOT2

# Case Study I

## Bag Plot

# ggplot2 2.0

- Write your own extensions
- Extremely flexible
- Create bag plot
  - John Tukey (box plots)
  - 2D box plot

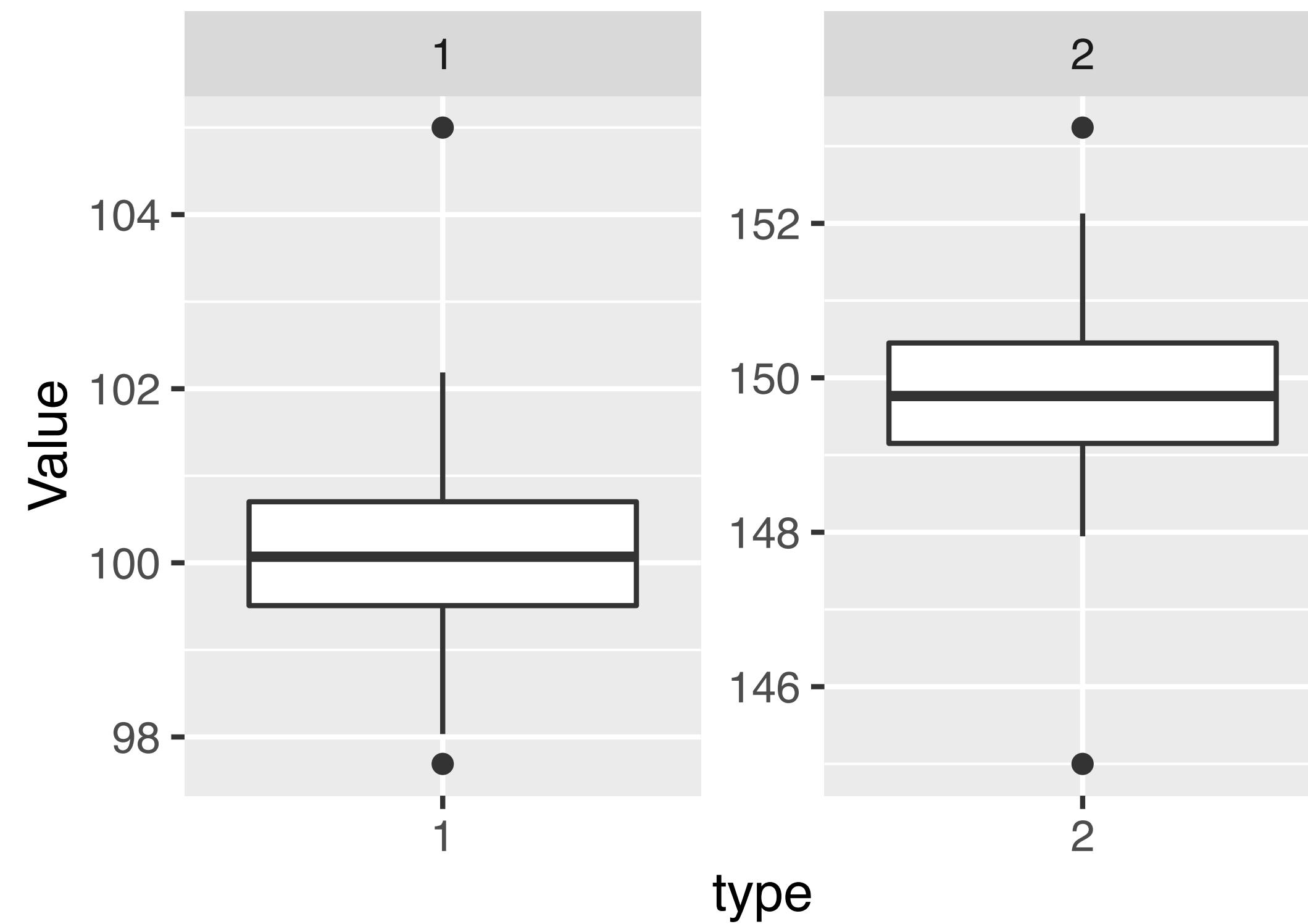
# data set

```
> dim(df)
[1] 202   2

> head(df)
  type      value
1    1 99.43952
2    1 99.76982
3    1 101.55871
4    1 100.07051
5    1 100.12929
6    1 101.71506
```

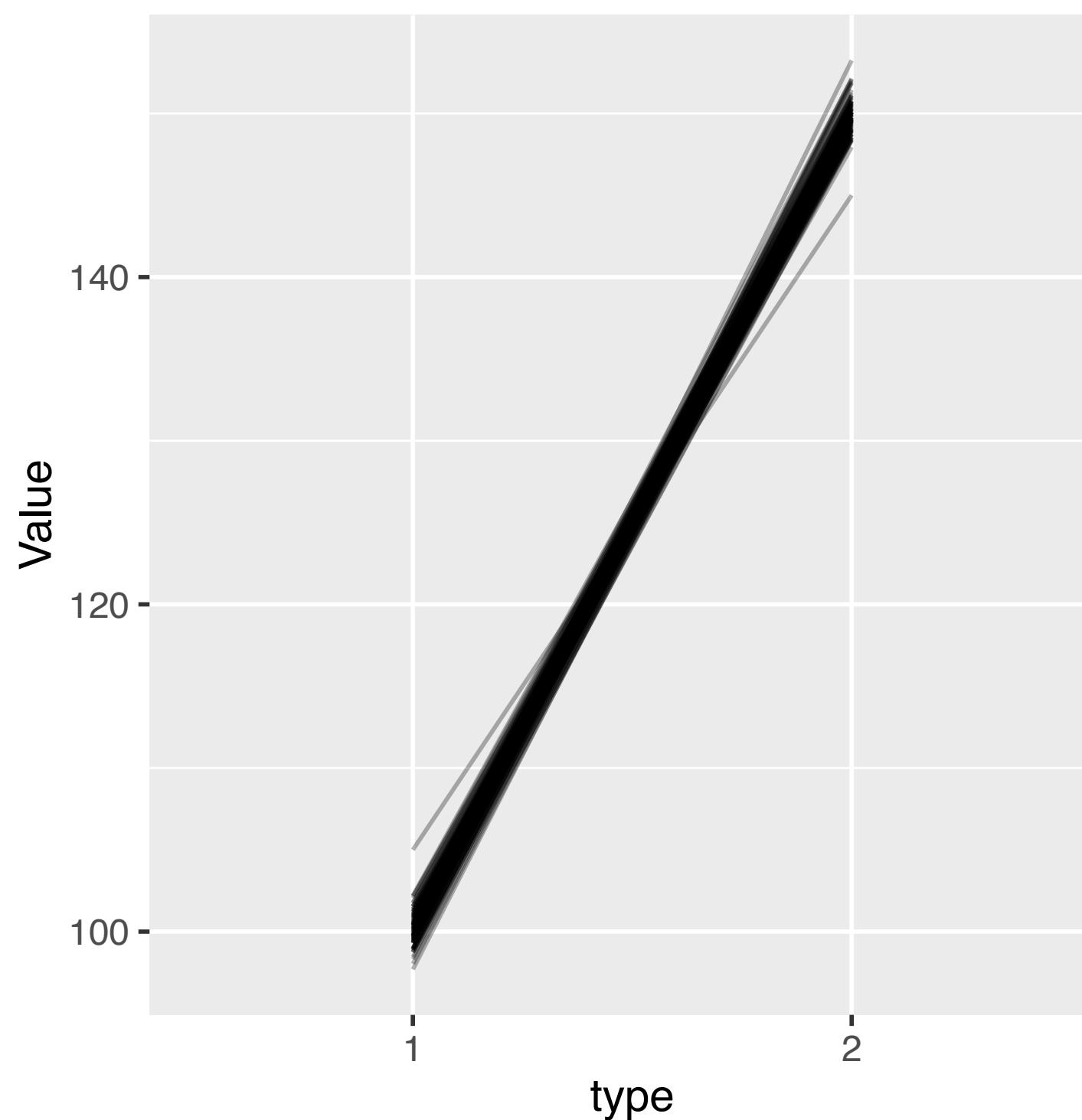
# 2 box plots

```
> ggplot(df, aes(x = type, Value)) +  
  geom_boxplot() +  
  facet_wrap(~type, ncol = 2, scales = "free")
```

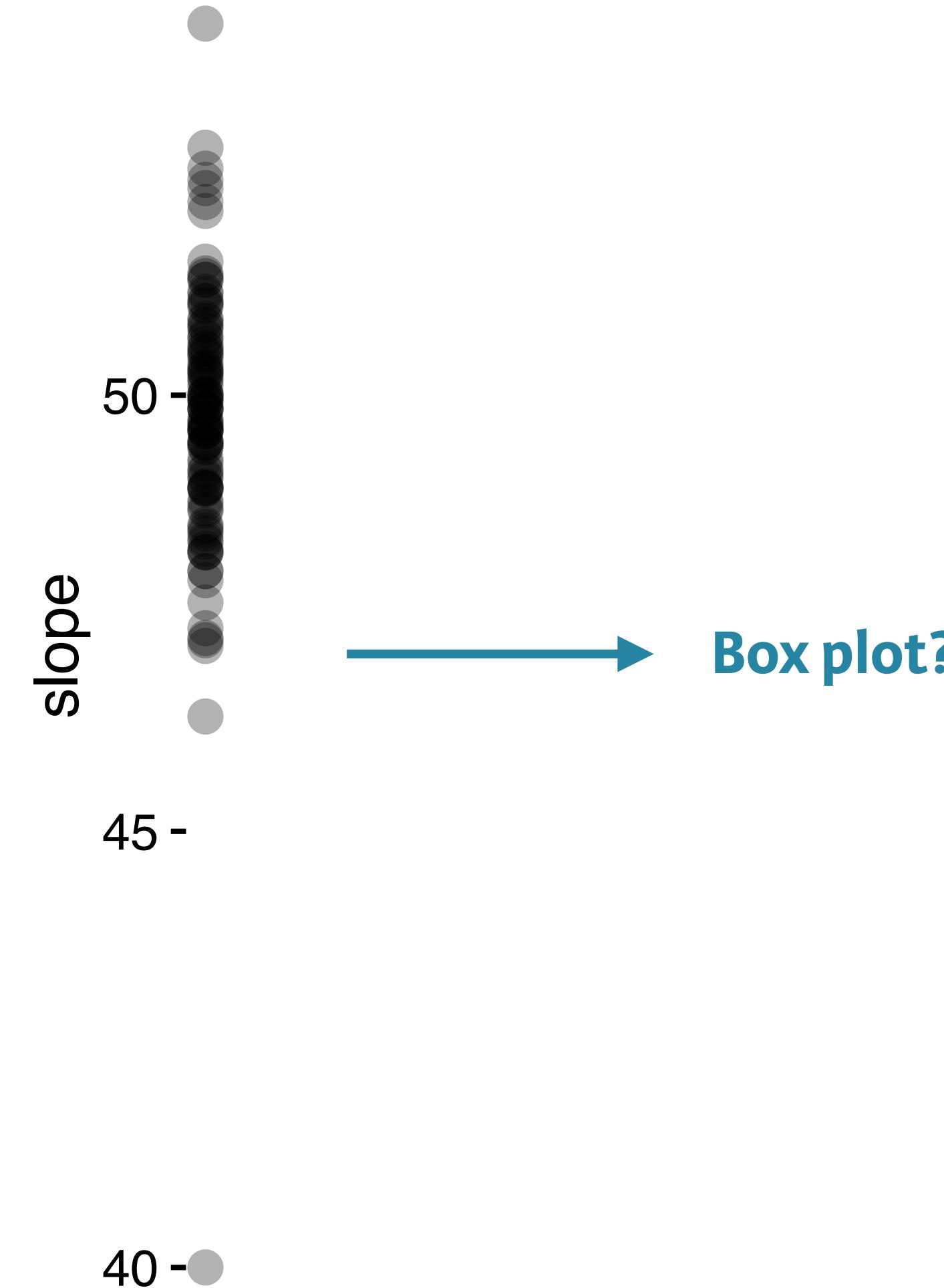


# slope plot

```
> df$ID <- seq_len(nrow(df) / 2)
> ggplot(df, aes(x = type, Value, group = ID)) +
  geom_line(alpha = 0.3)
```



# Distribution of slope

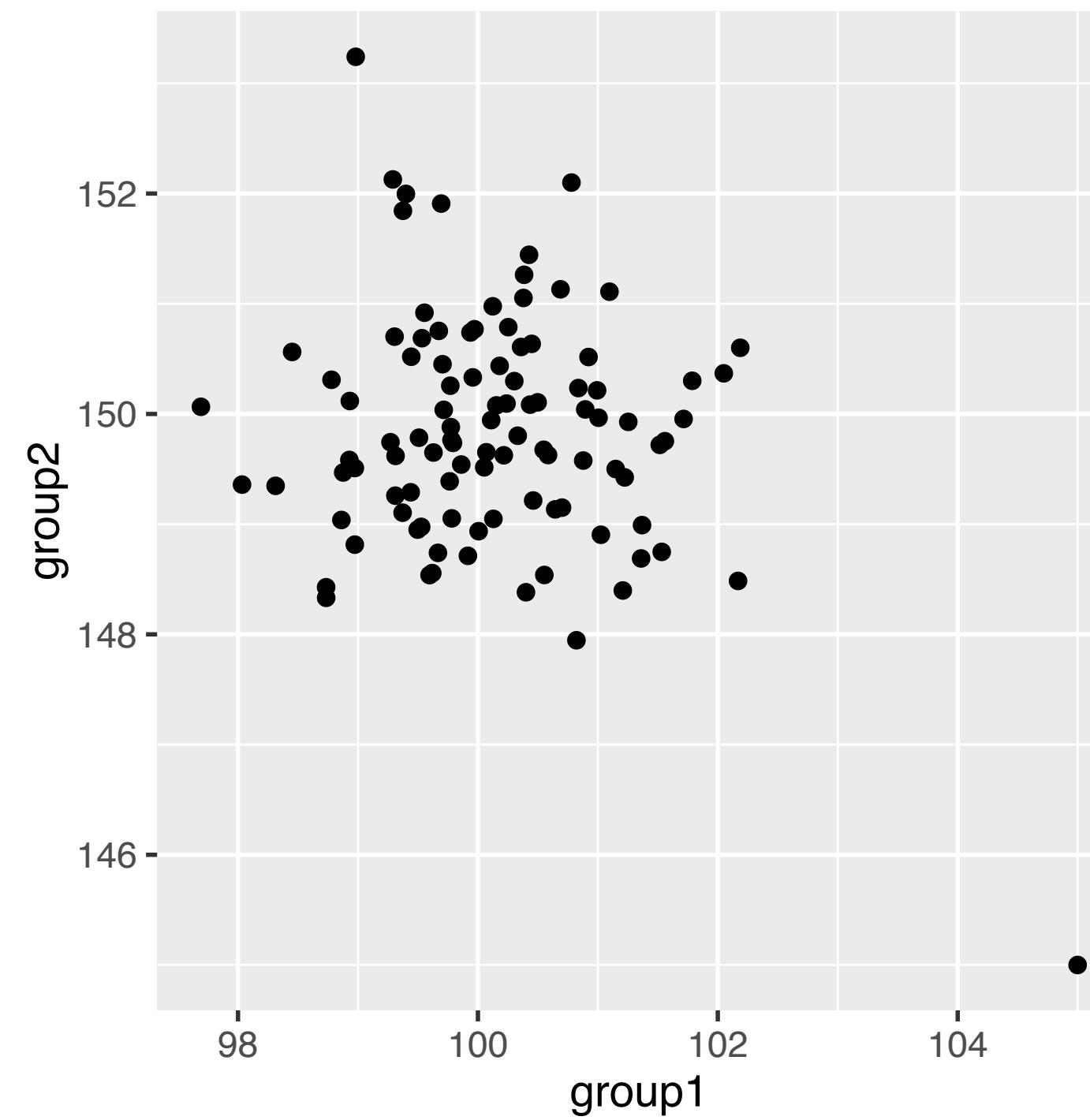


# 2 distinct variables

```
> head(dat)
  group1    group2
1 99.43952 149.2896
2 99.76982 150.2569
3 101.55871 149.7533
4 100.07051 149.6525
5 100.12929 149.0484
6 101.71506 149.9550
```

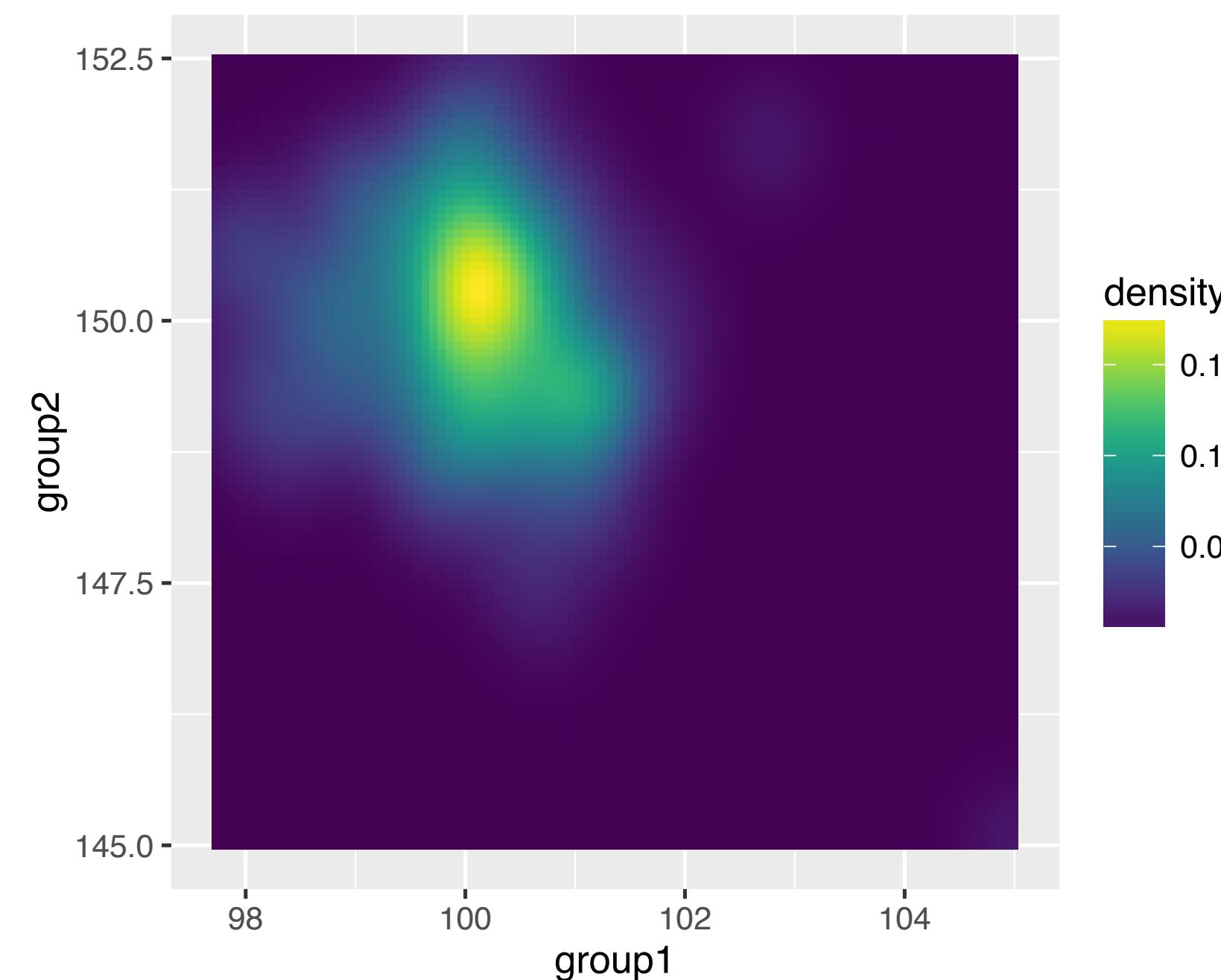
# Scatter plot

```
> ggplot(dat, aes(x = group1, y = group2)) +  
  geom_point()
```



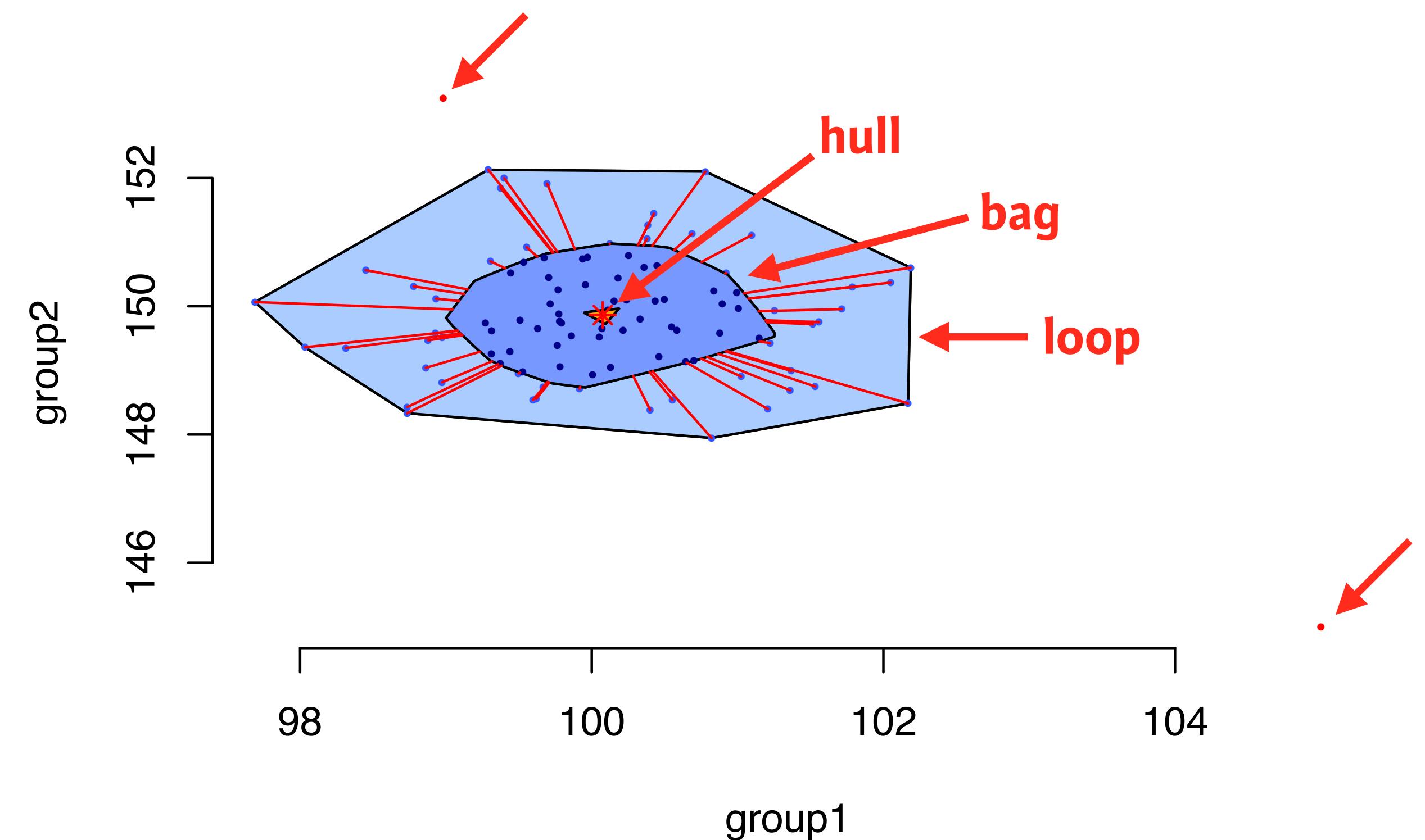
# 2D density plot

```
> library(viridis)
> ggplot(dat, aes(x = group1, y = group2)) +
  stat_density_2d(geom = "tile", aes(fill = ..density..),
                  contour = FALSE) +
  scale_fill_viridis()
```



# Bag plot

```
> library(aplpack)
> bagplot(dat[1:2])
```



# aplpack

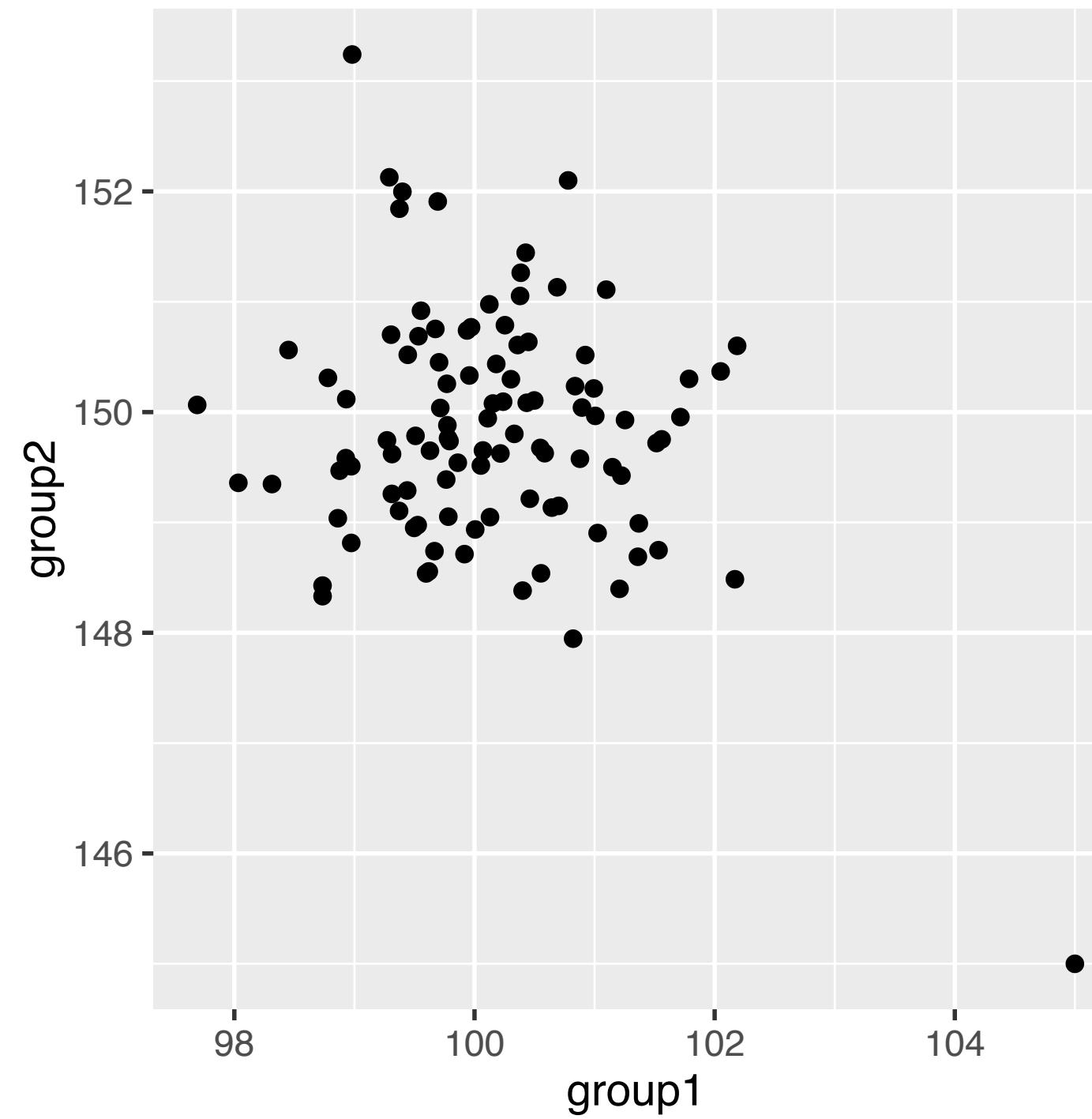
```
> library(aplpack)

> plot_data <- compute.bagplot(x = dat$group1, y = dat$group2)

> names(plot_data)
[1] "center"      "hull.center"  "hull.bag"     "hull.loop"
[5] "pxy.bag"     "pxy.outer"    "pxy.outlier"  "hdepths"
[9] "is.one.dim"  "prdata"       "xy"           "xydata"
```

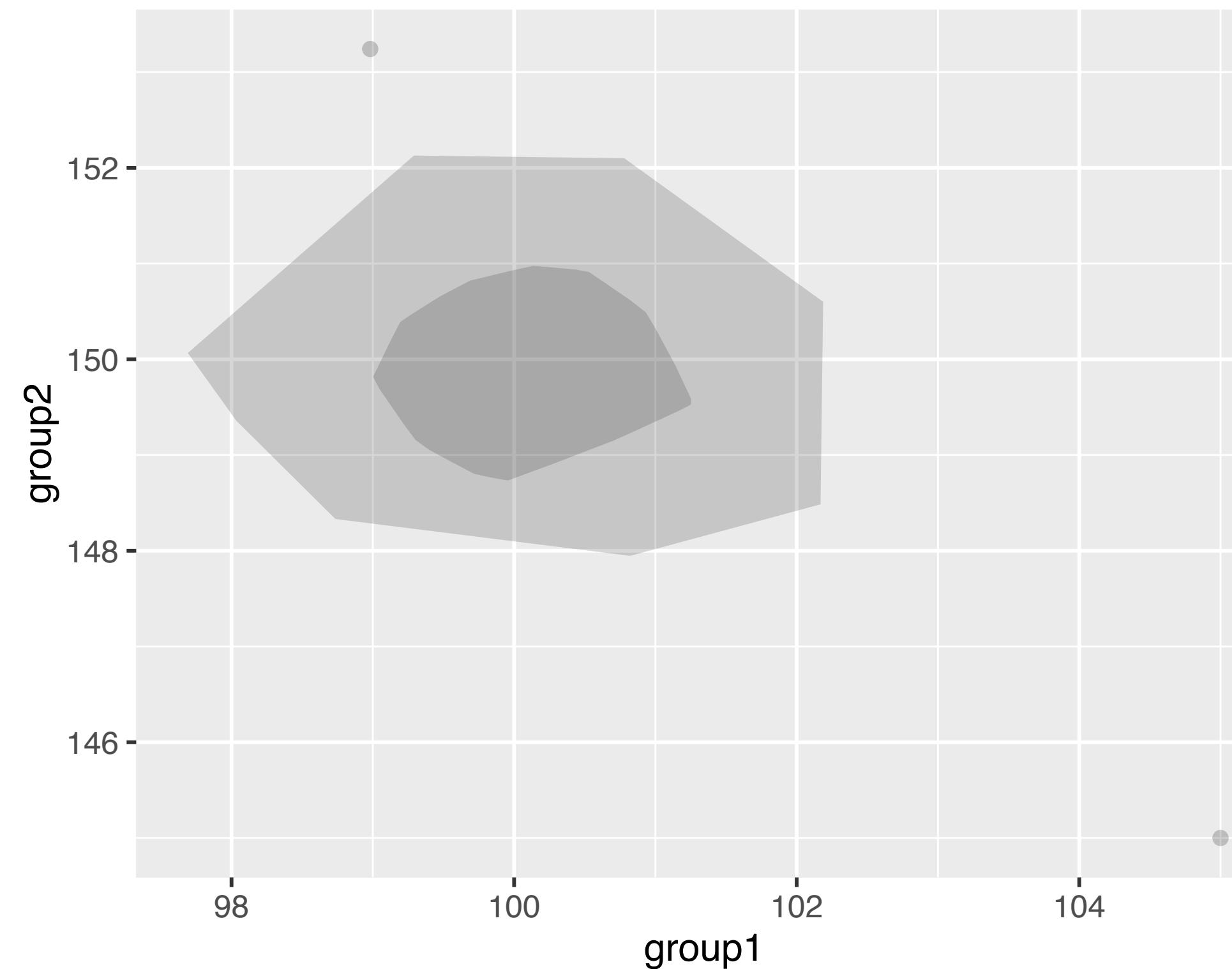
# ggplot2

```
> ggplot(dat, aes(x = group1, y = group2)) +  
  geom_point()
```



# ggplot2

```
> ggplot(dat, aes(x = group1, y = group2)) +  
  stat_bag(alpha = 0.2)
```



# Remarks

- Useful but not popular
- Poorly understood
- Learn to use ggplot2 extensions



DATA VISUALIZATION WITH GGPLOT2

**Let's practice!**

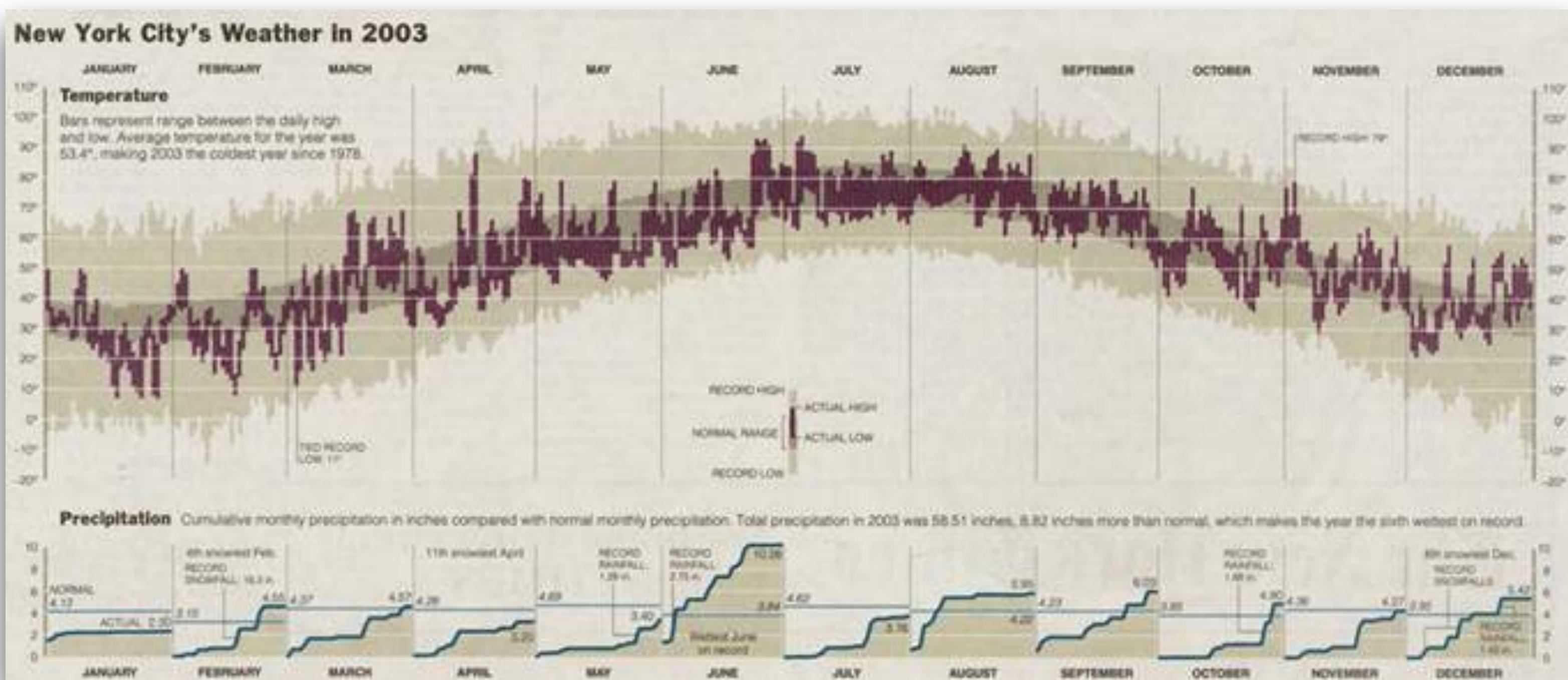


DATA VISUALIZATION WITH GGPLOT2

# Case Study II Weather (Part 1)



# Weather



# present

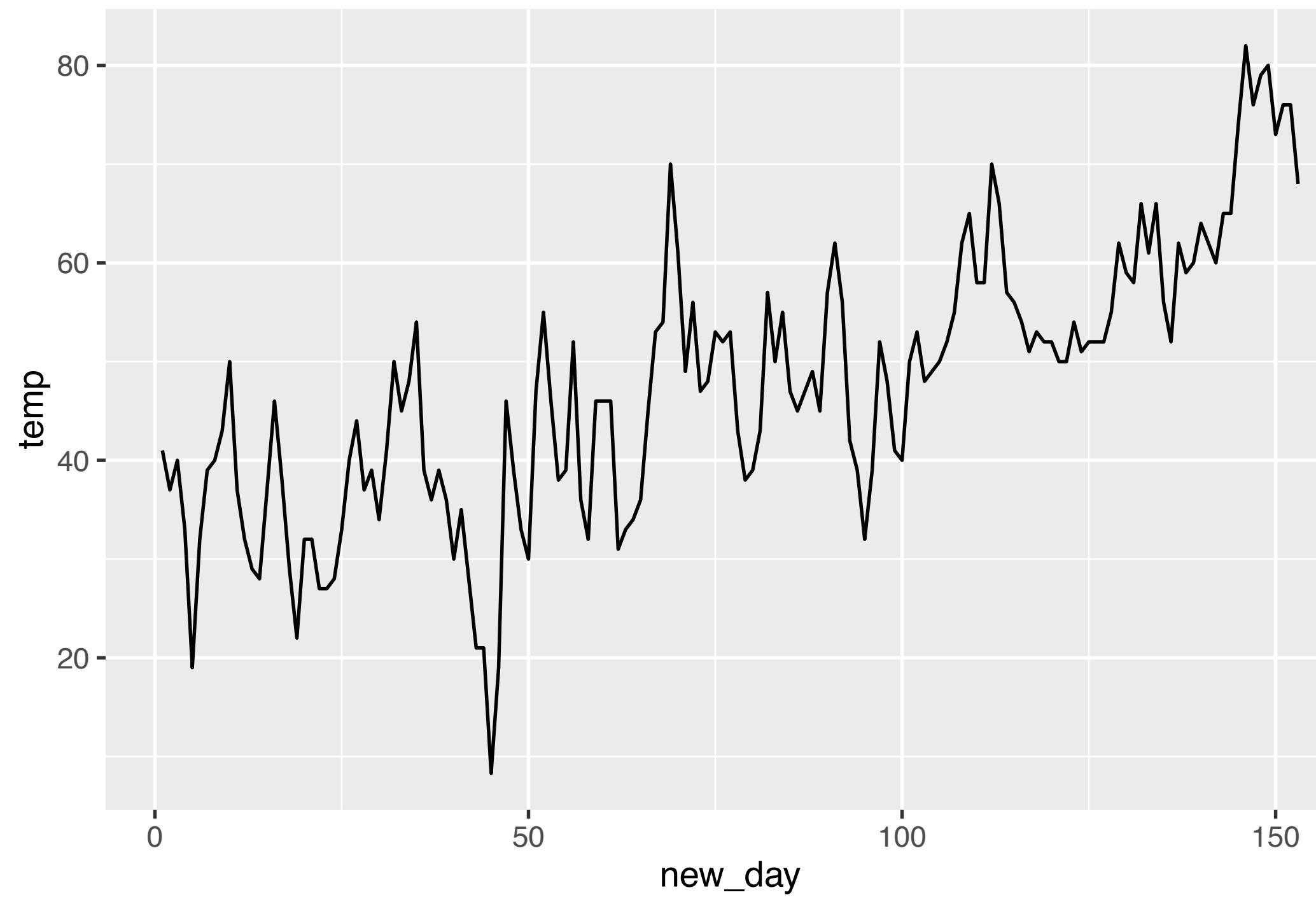
```
> dim(present)
[1] 153   5

> head(present, n = 4)
  month day year temp new_day
1     1   1 2016   41       1
2     1   2 2016   37       2
3     1   3 2016   40       3
4     1   4 2016   33       4

> tail(present, n = 4)
  month day year temp new_day
148     5   28 2016   79      148
149     5   29 2016   80      149
150     5   30 2016   73      150
151     5   31 2016   76      151
```

# Time series

```
> ggplot(present, aes(x = new_day, y = temp)) +  
  geom_line()
```

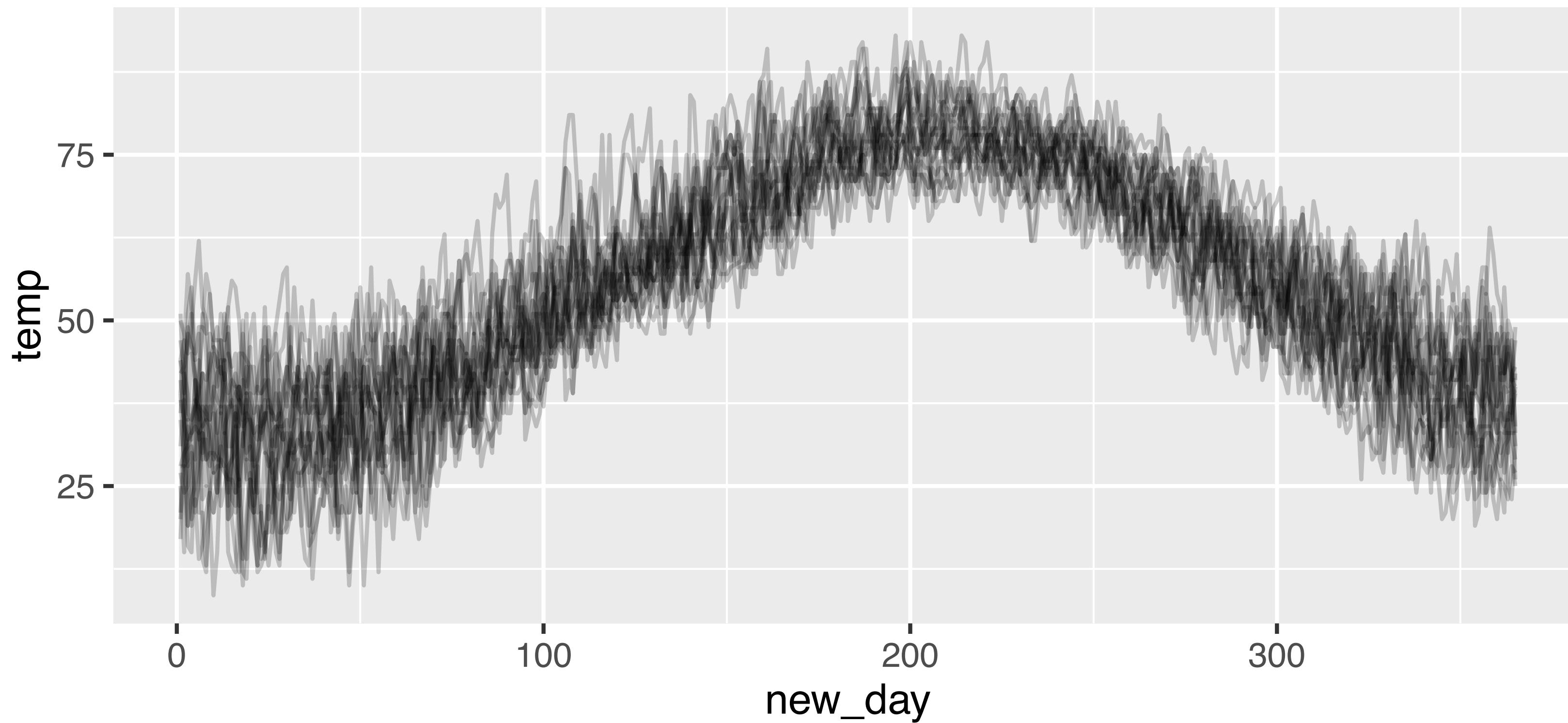


# past

```
> str(past)
'data.frame': 7645 obs. of 11 variables:
 $ month     : num  1 1 1 1 1 1 1 1 1 1 ...
 $ day        : num  1 2 3 4 5 6 7 8 9 10 ...
 $ year       : num  1995 1995 1995 1995 1995 ...
 $ temp       : num  44 41 28 31 21 27 42 35 34 29 ...
 $ new_day    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ upper      : num  51 48 57 55 56 62 52 57 54 47 ...
 $ lower      : num  17 15 16 15 21 14 14 12 21 8.5 ...
 $ avg        : num  35.6 35.4 34.9 35.1 35.9 ...
 $ se         : num  2.19 1.83 2.46 2.53 1.92 ...
 $ avg_upper : num  40.2 39.2 40 40.5 39.9 ...
 $ avg_lower : num  31 31.5 29.7 29.8 31.9 ...
```

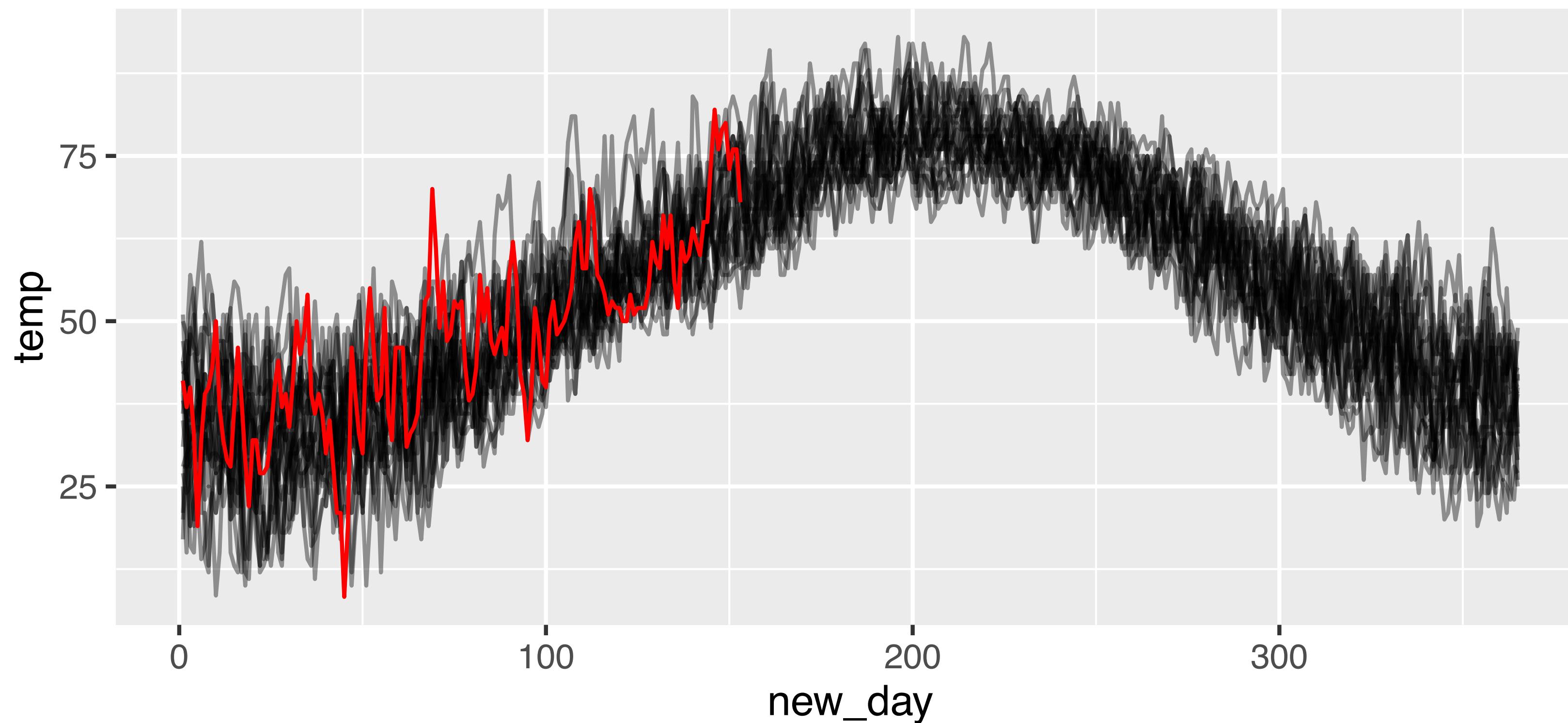
# Each year separately

```
> ggplot(past, aes(x = new_day, y = temp, group = year)) +  
  geom_line(alpha = 0.2)
```



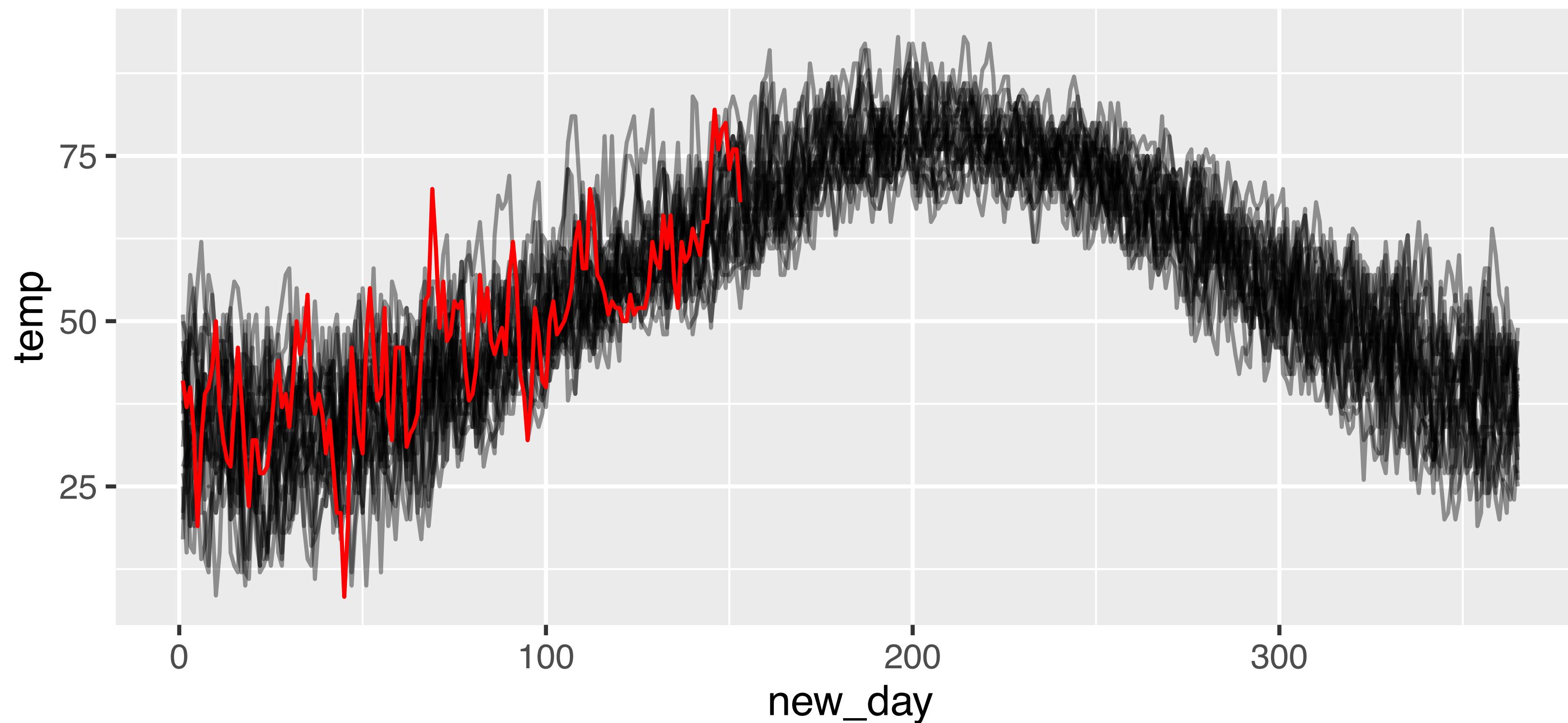
# present + past

```
> ggplot(past, aes(x = new_day, y = temp, group = year)) +  
  geom_line(alpha = 0.4) +  
  geom_line(data = present, aes(group = 1), col = "red")
```

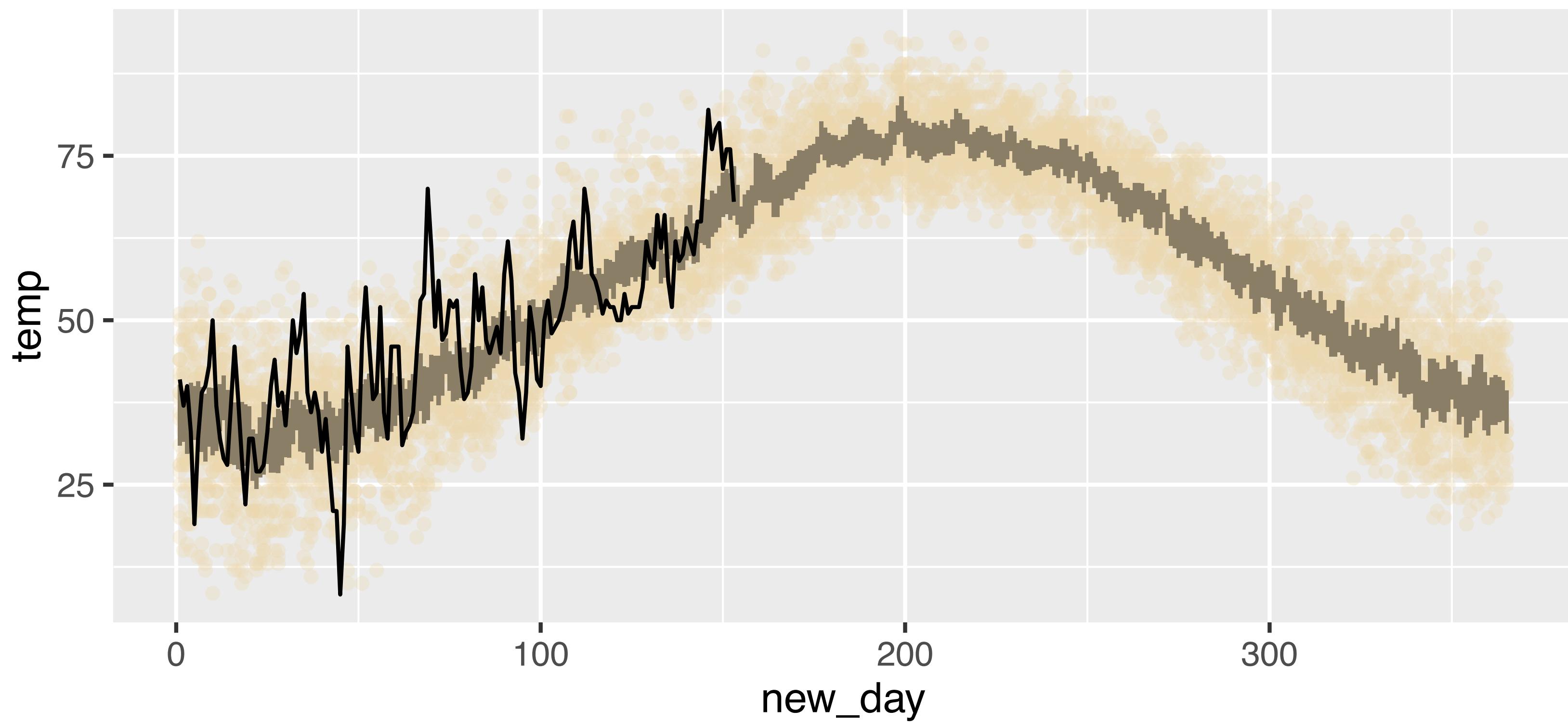


# present + past

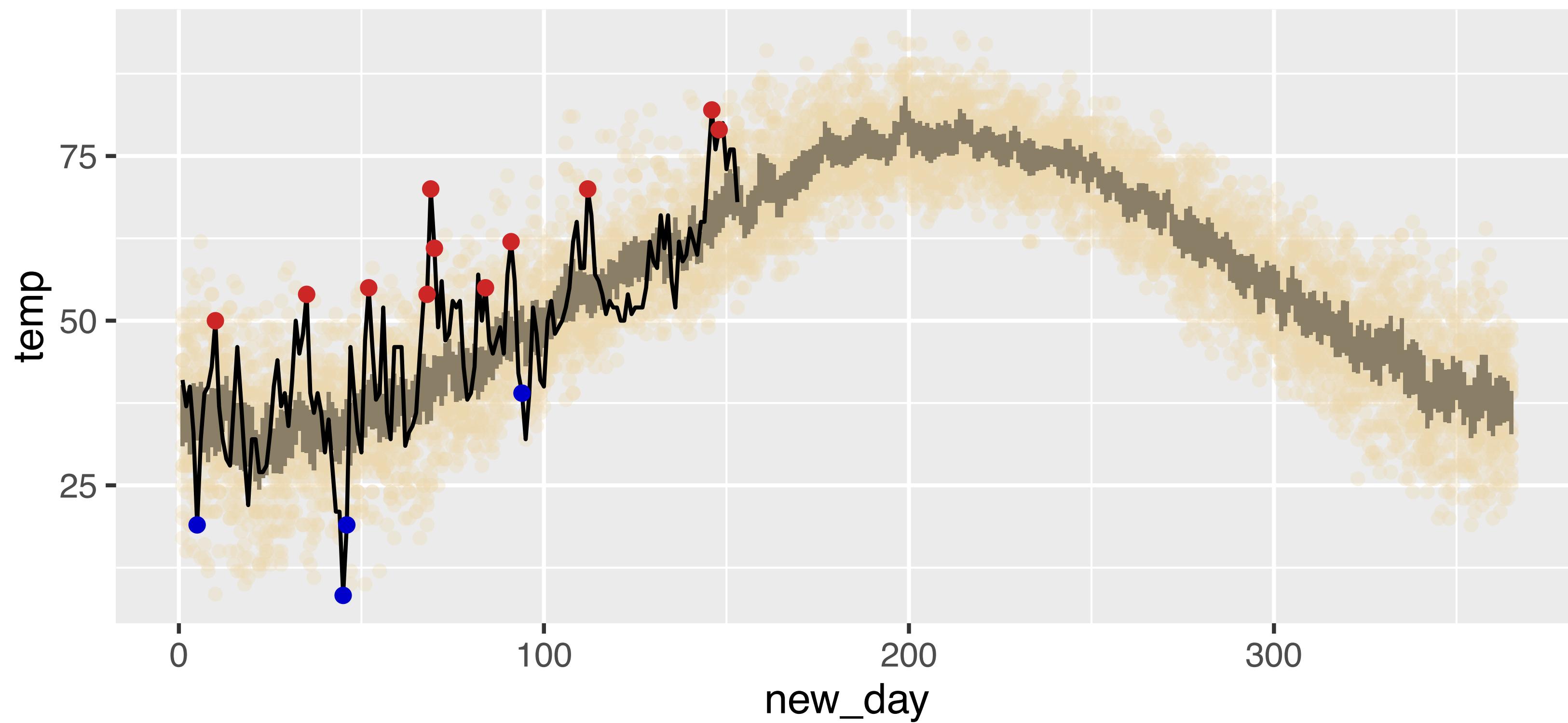
```
> ggplot(past, aes(x = new_day, y = temp, group = year)) +  
  geom_line(alpha = 0.4) +  
  geom_line(data = present, aes(group = 1), col = "red")
```



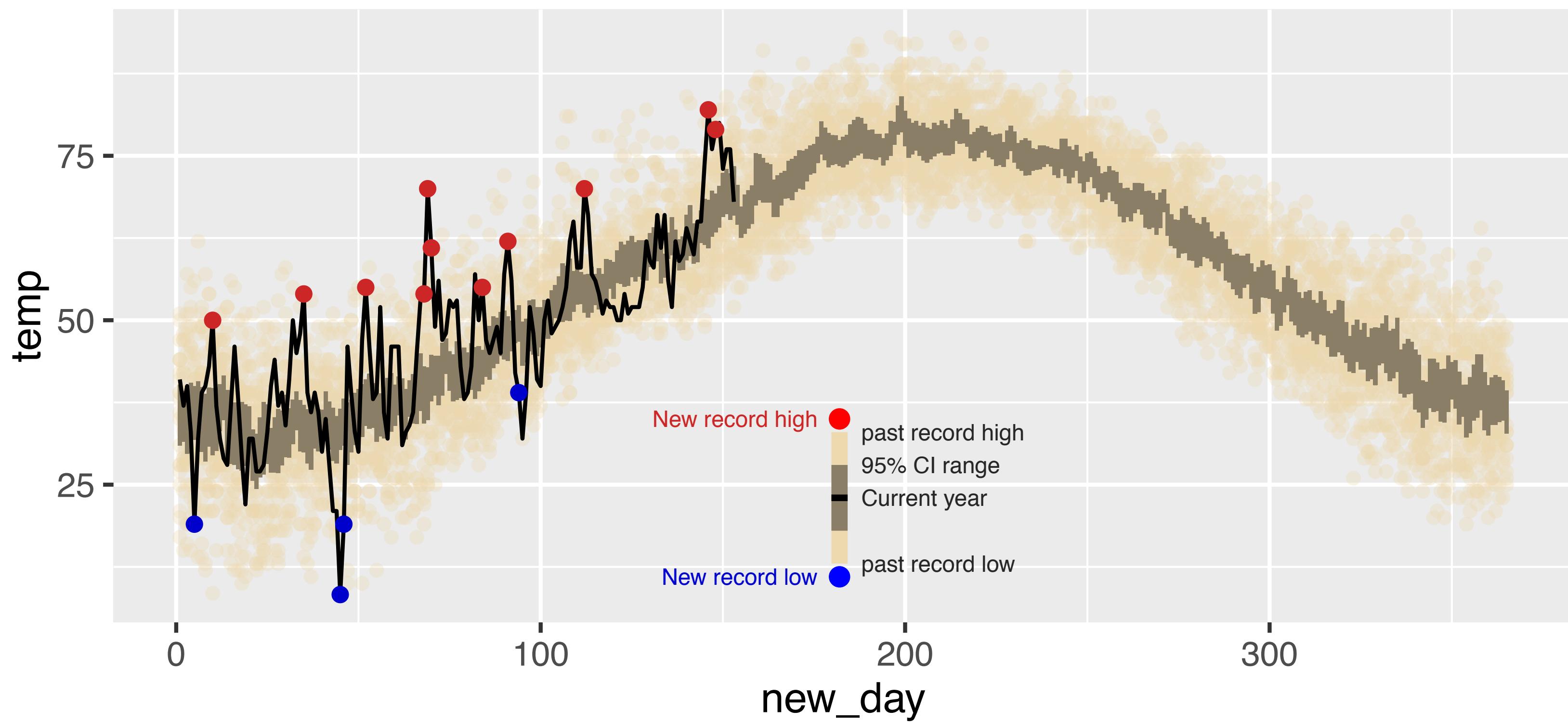
# Linerange



# Records



# Custom legend





DATA VISUALIZATION WITH GGPLOT2

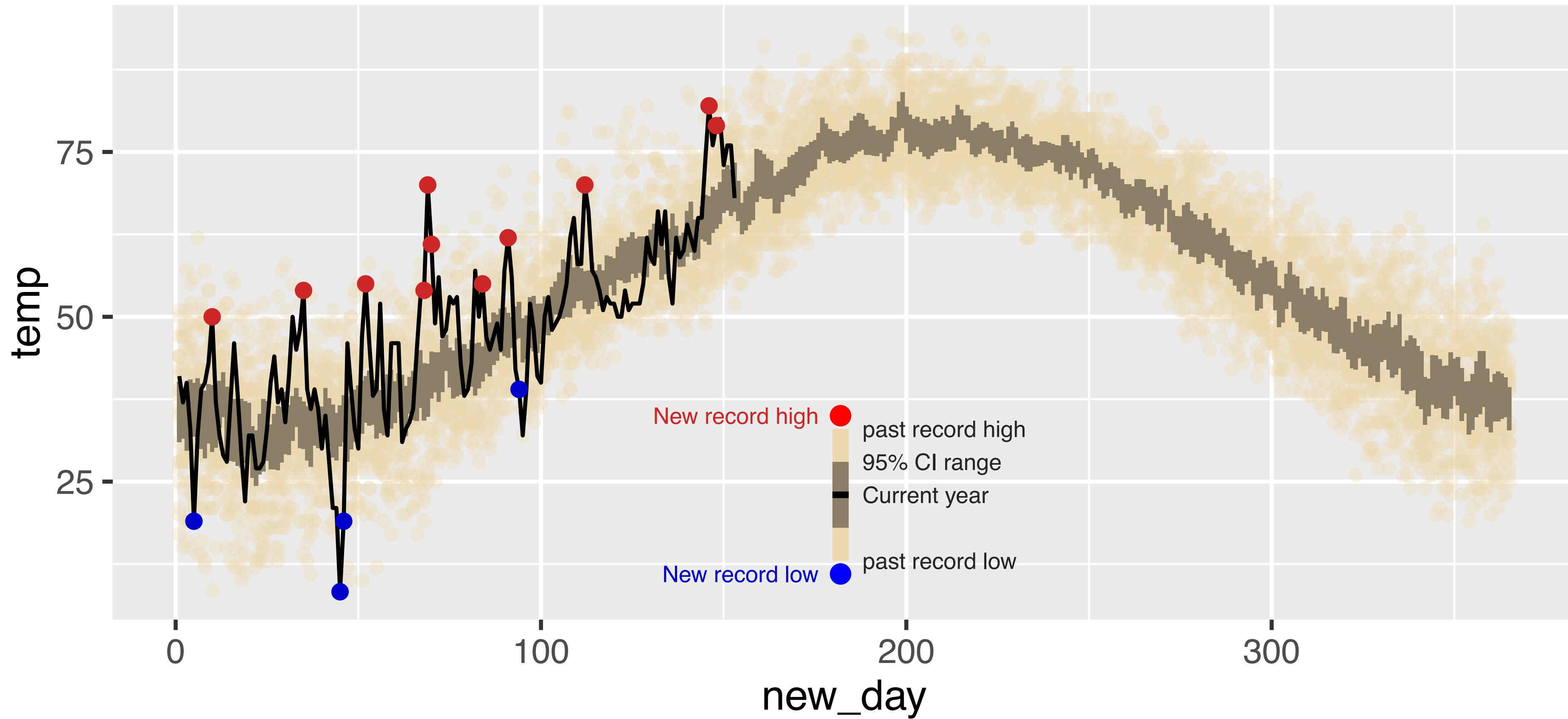
**Let's practice!**



DATA VISUALIZATION WITH GGPLOT2

# Case Study II Weather (Part 2)

# Up to now

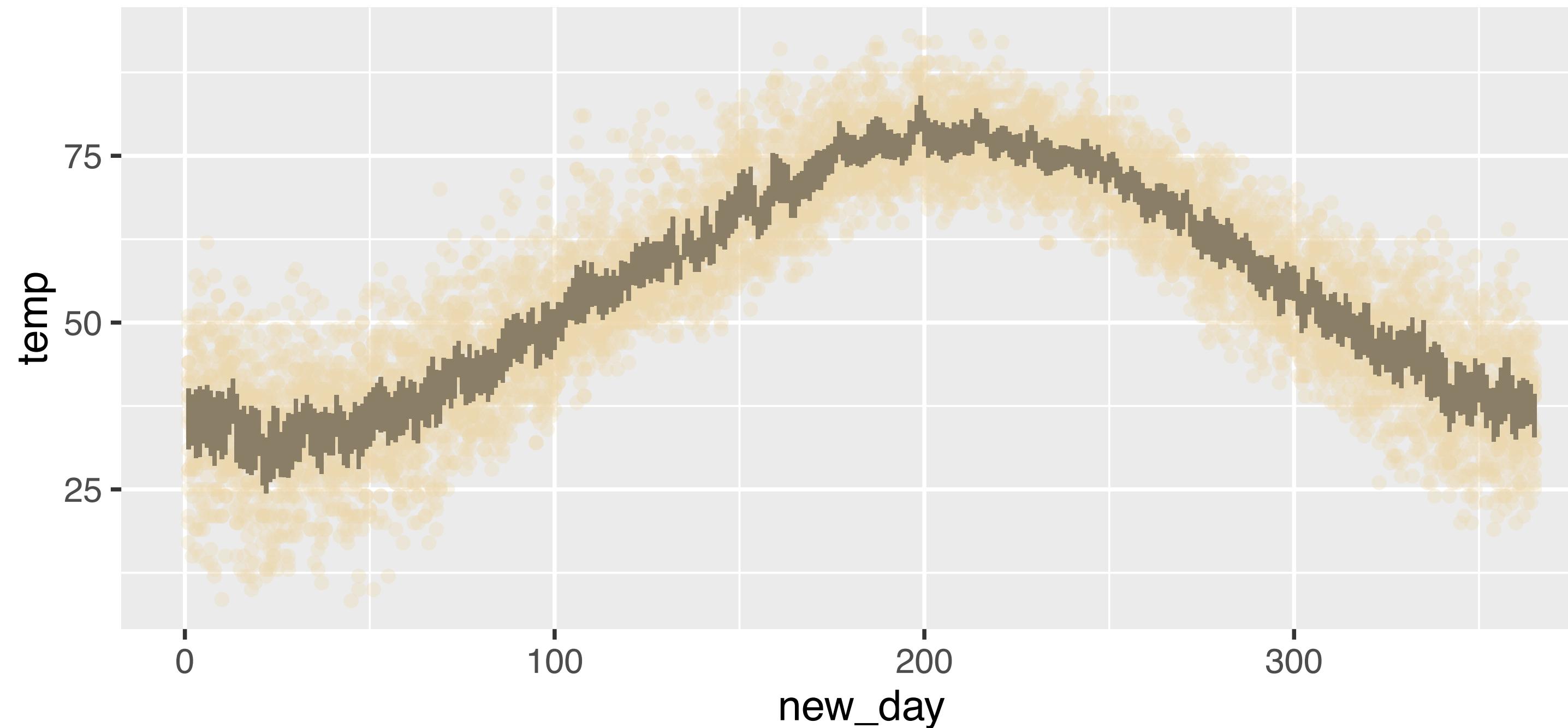


# Situation

- Many data frames
- Plot summary data frame as a layer
- `stat_summary()`

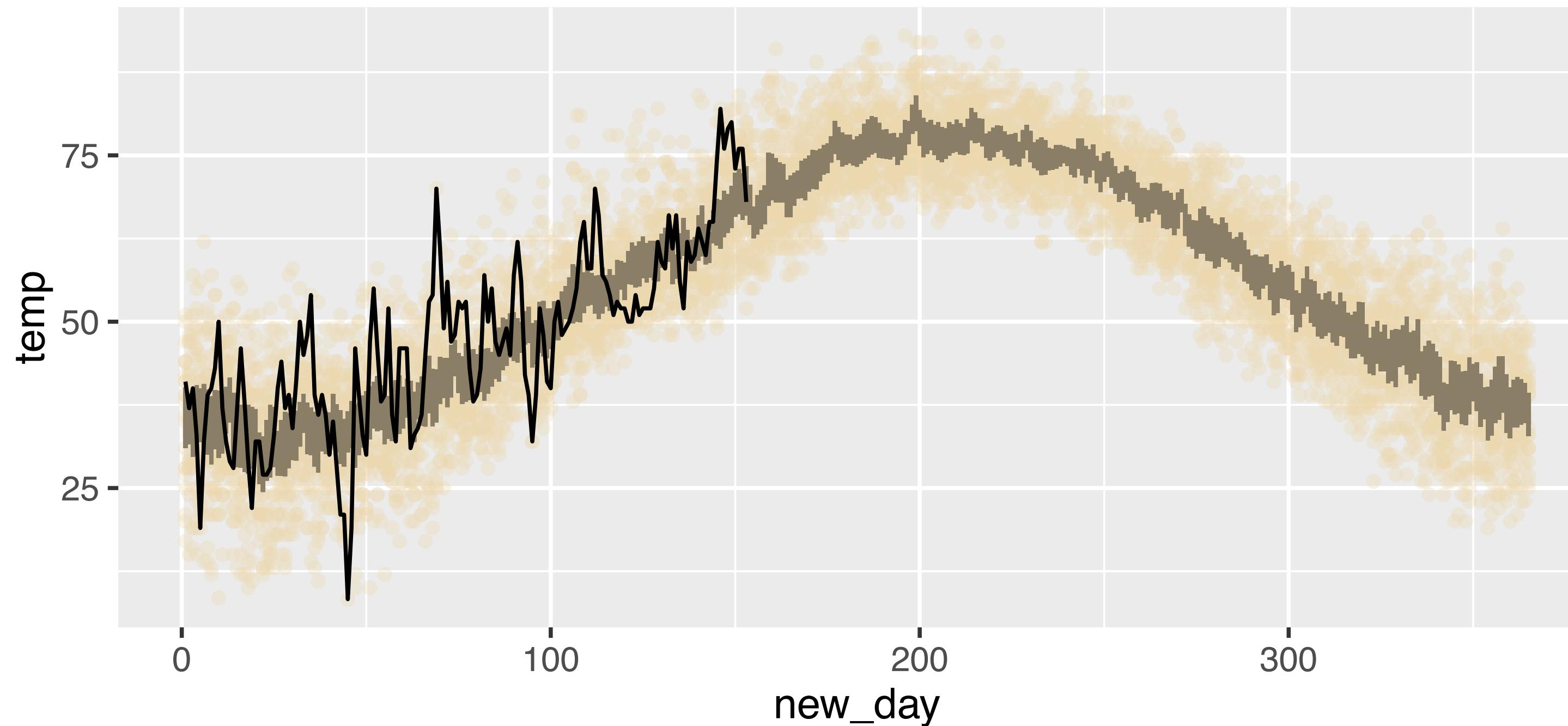
# stat\_histirical()

```
> ggplot(my_data, aes(x = new_day, y = temp, fill = year)) +  
  stat_histirical()
```



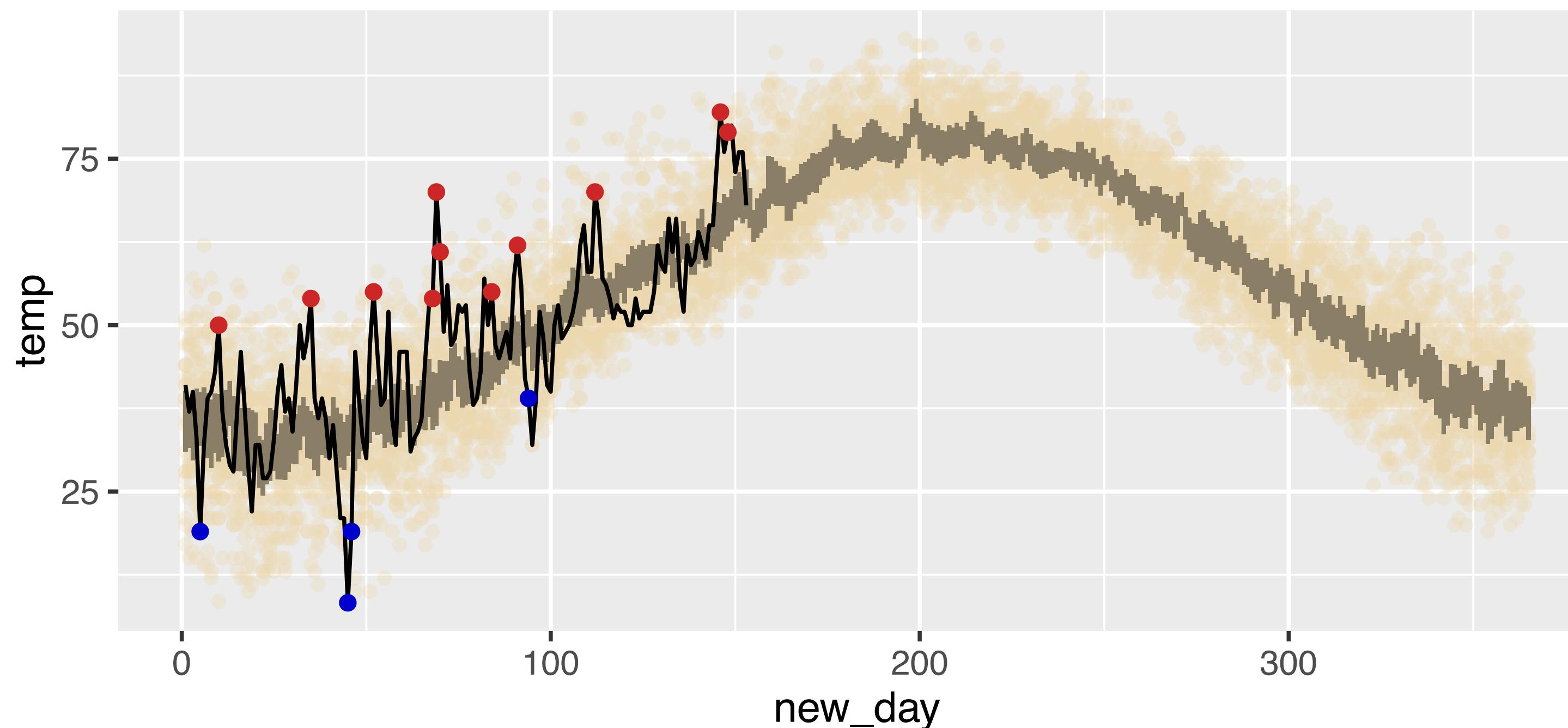
# stat\_present()

```
> ggplot(my_data, aes(x = new_day, y = temp, fill = year)) +  
  stat_historical() +  
  stat_present()
```



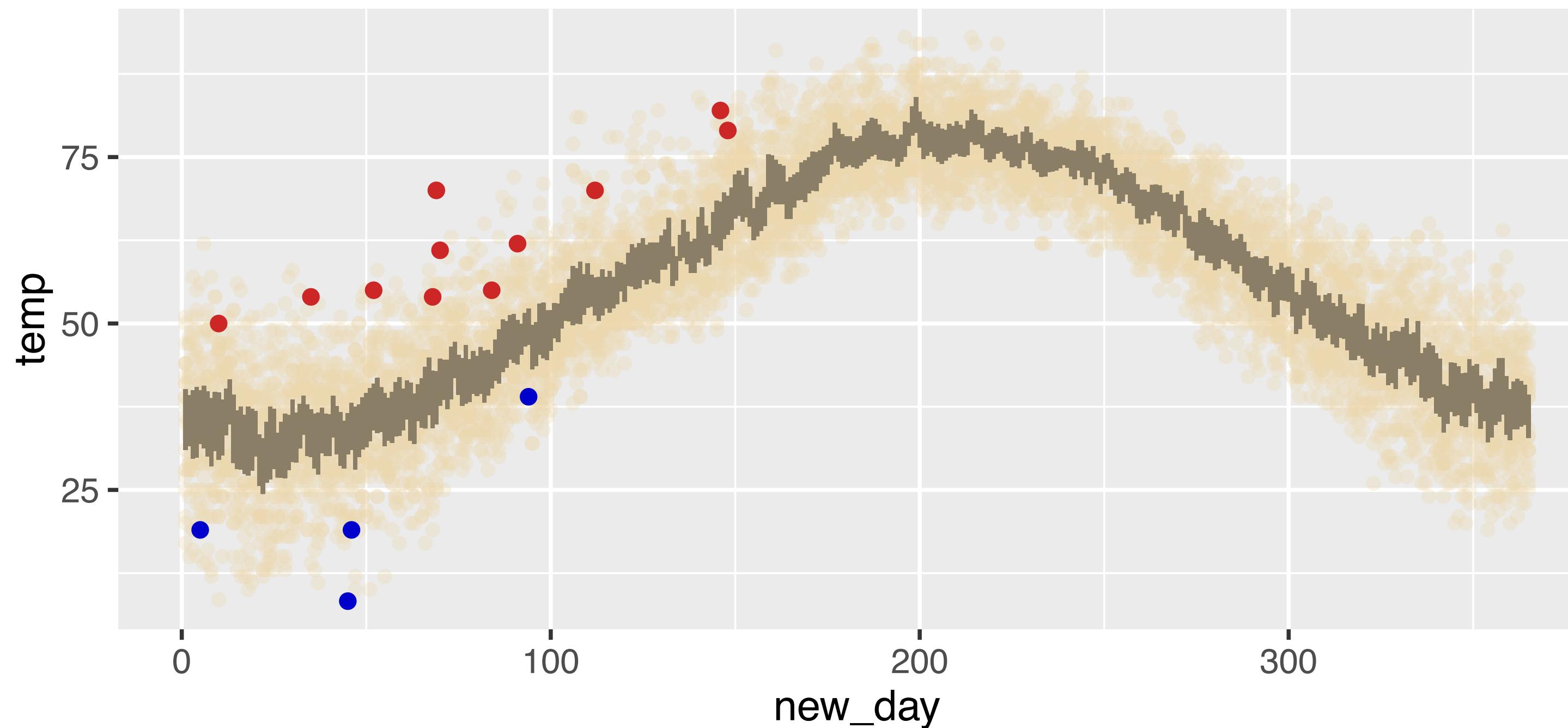
# stat\_extremes()

```
> ggplot(my_data, aes(new_day, temp, fill = year)) +  
  stat_historical() +  
  stat_present() +  
  stat_extremes(aes(colour = ..record..))
```

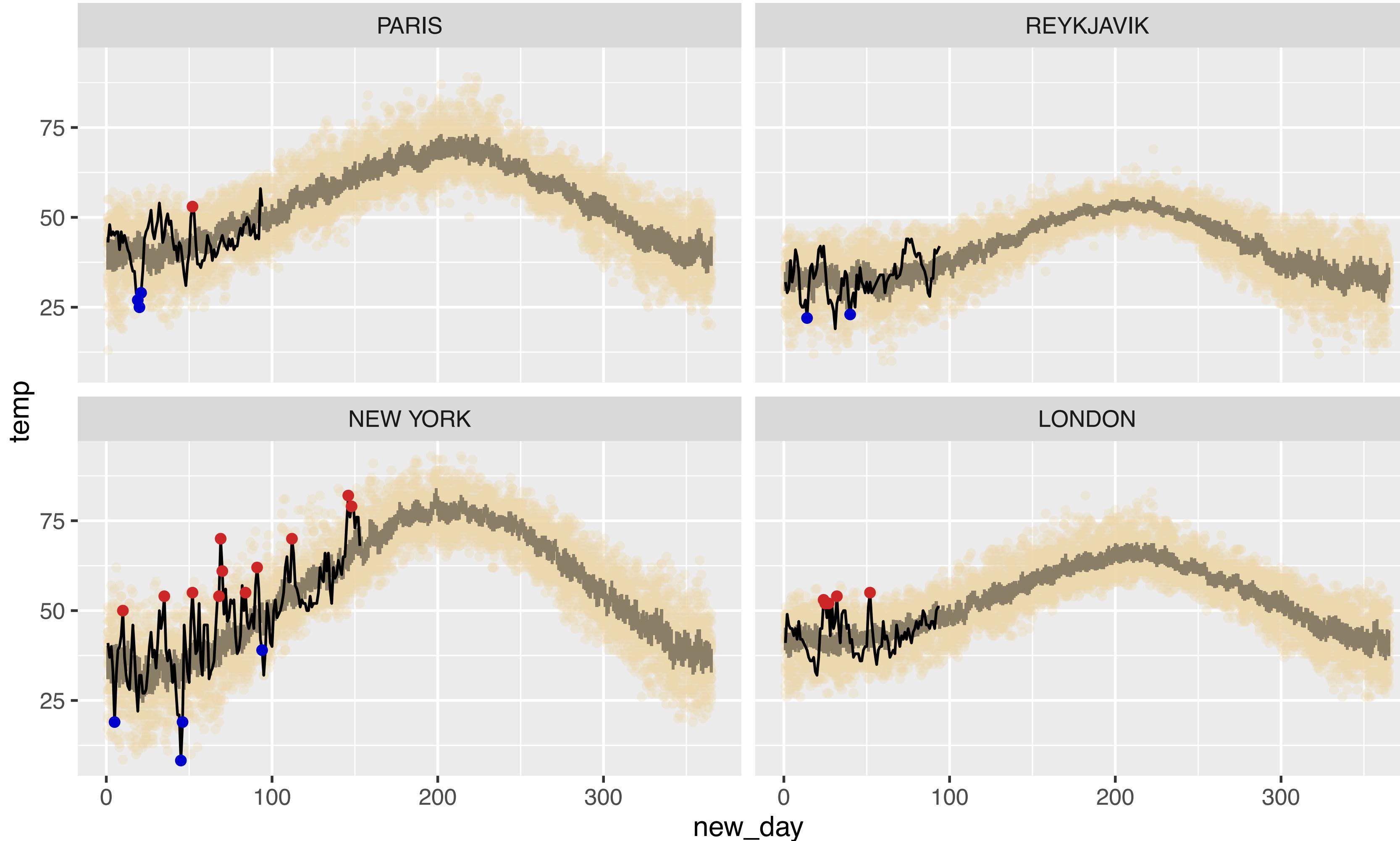


# Specific layers

```
> ggplot(my_data, aes(new_day, temp, fill = year)) +  
  stat_historical() +  
  # stat_present() +  
  stat_extremes(aes(colour = ..record..))
```



# Facetting





DATA VISUALIZATION WITH GGPLOT2

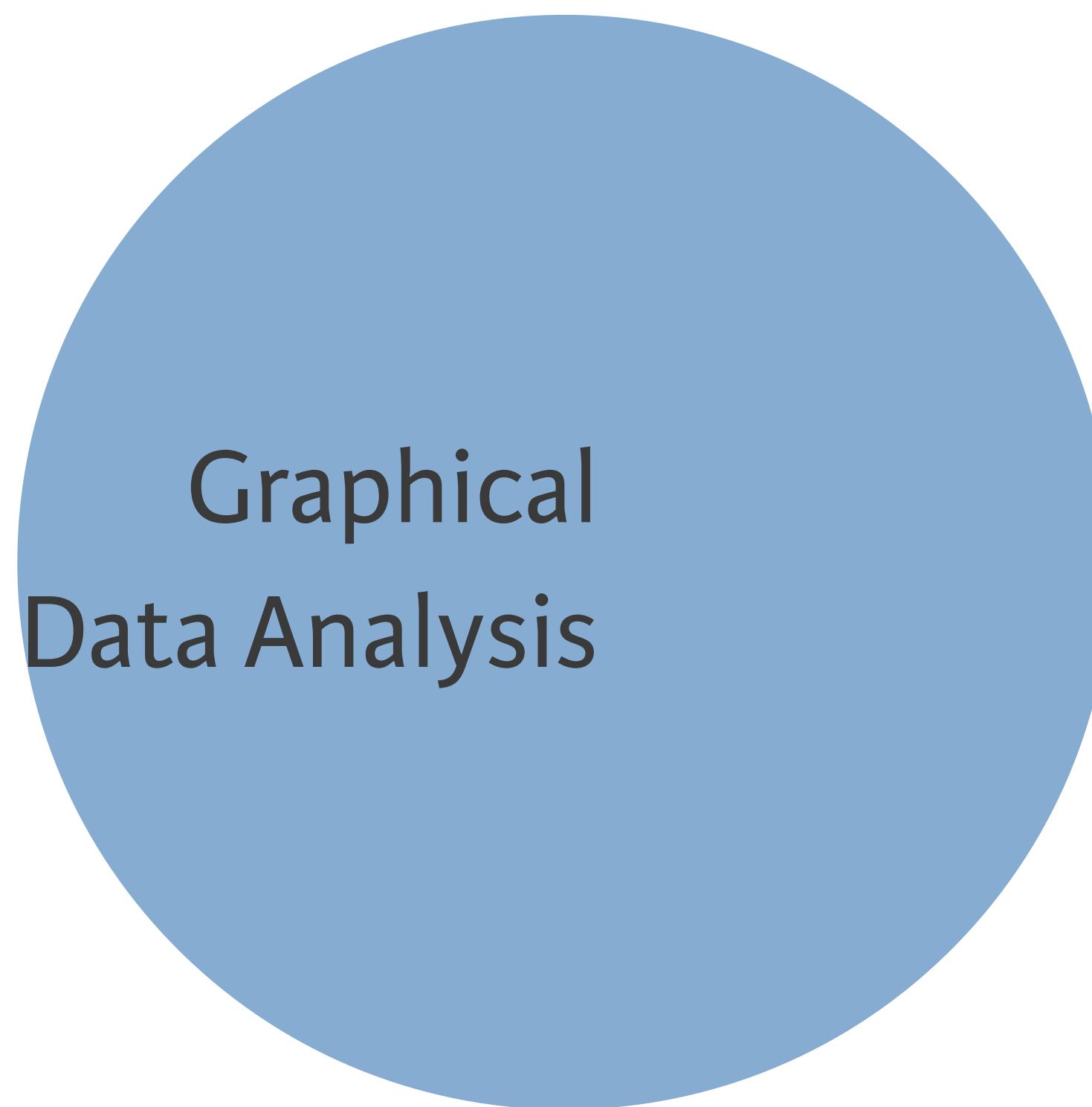
**Let's practice!**



DATA VISUALIZATION WITH GGPLOT2

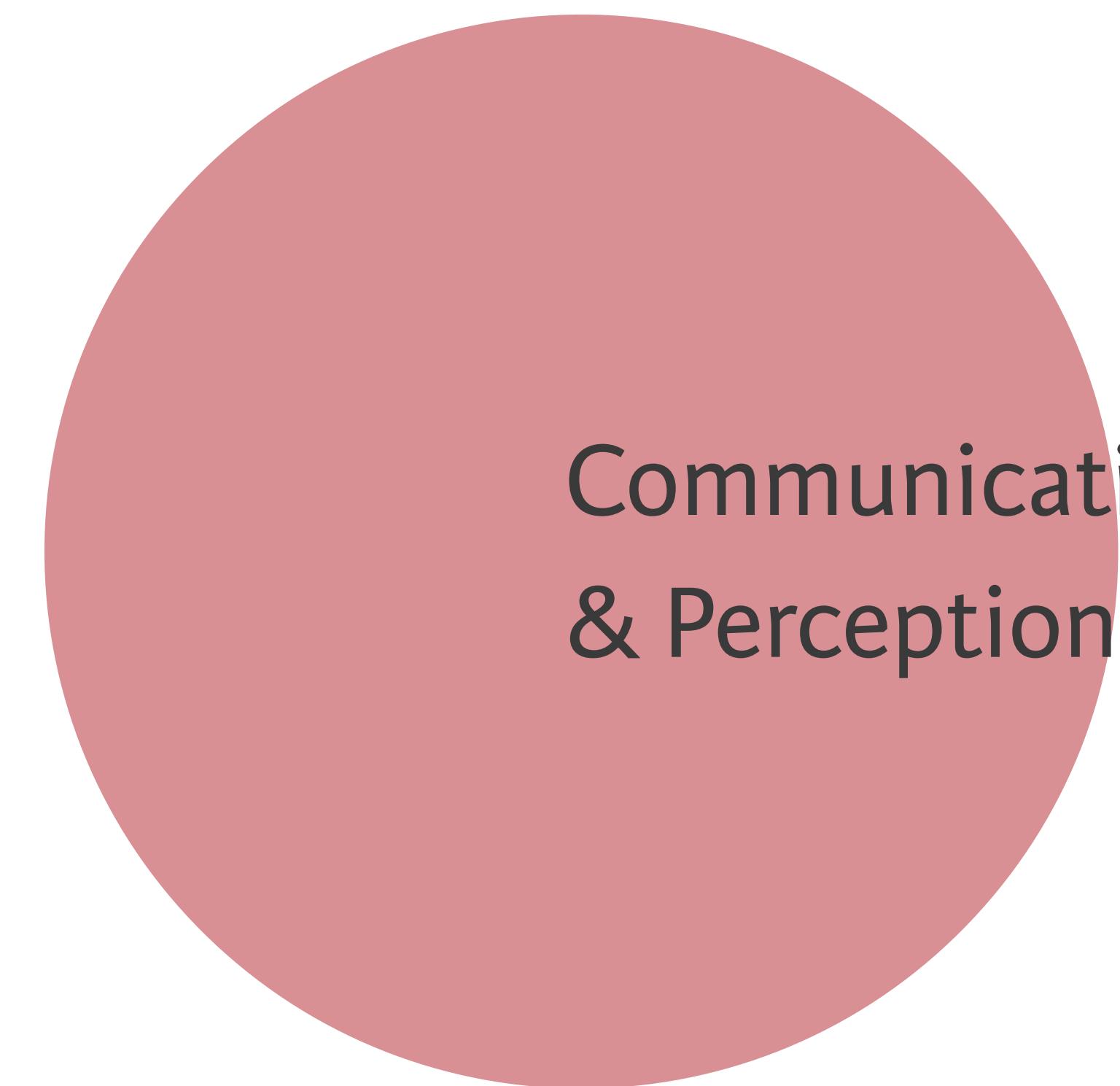
# Wrap-up

Statistics

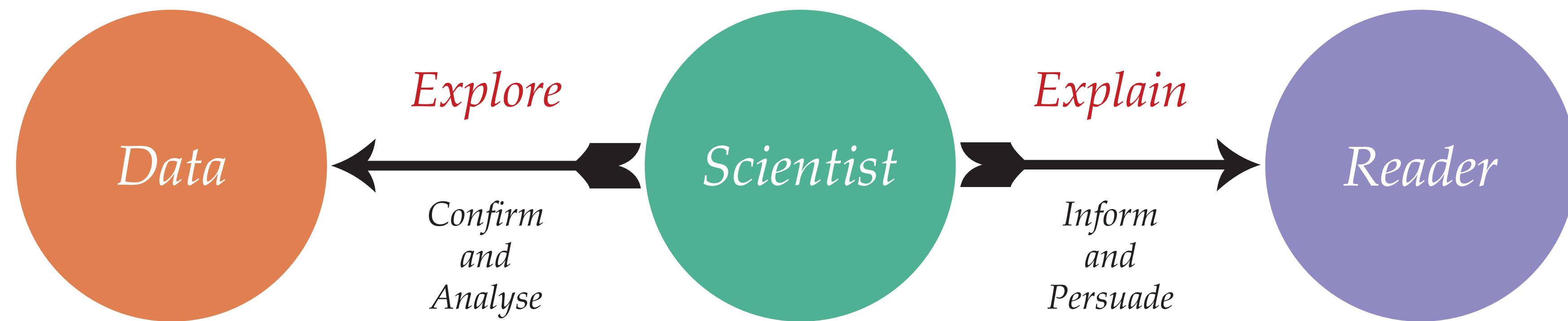


Graphical  
Data Analysis

Design

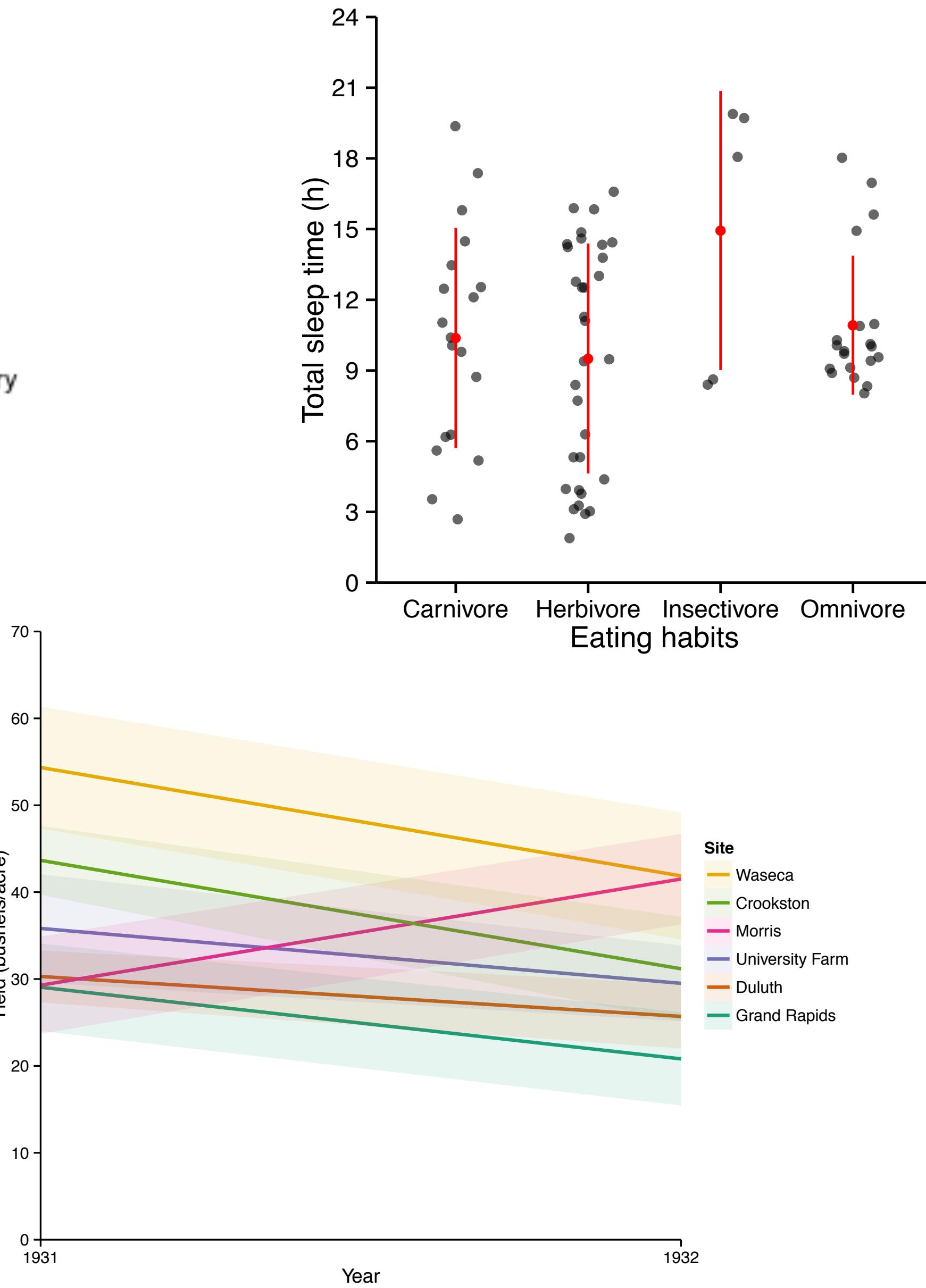
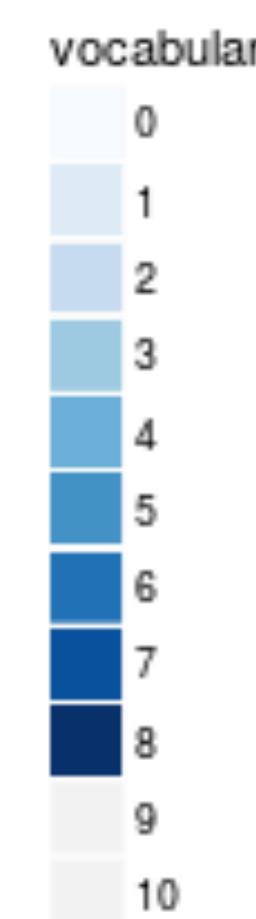
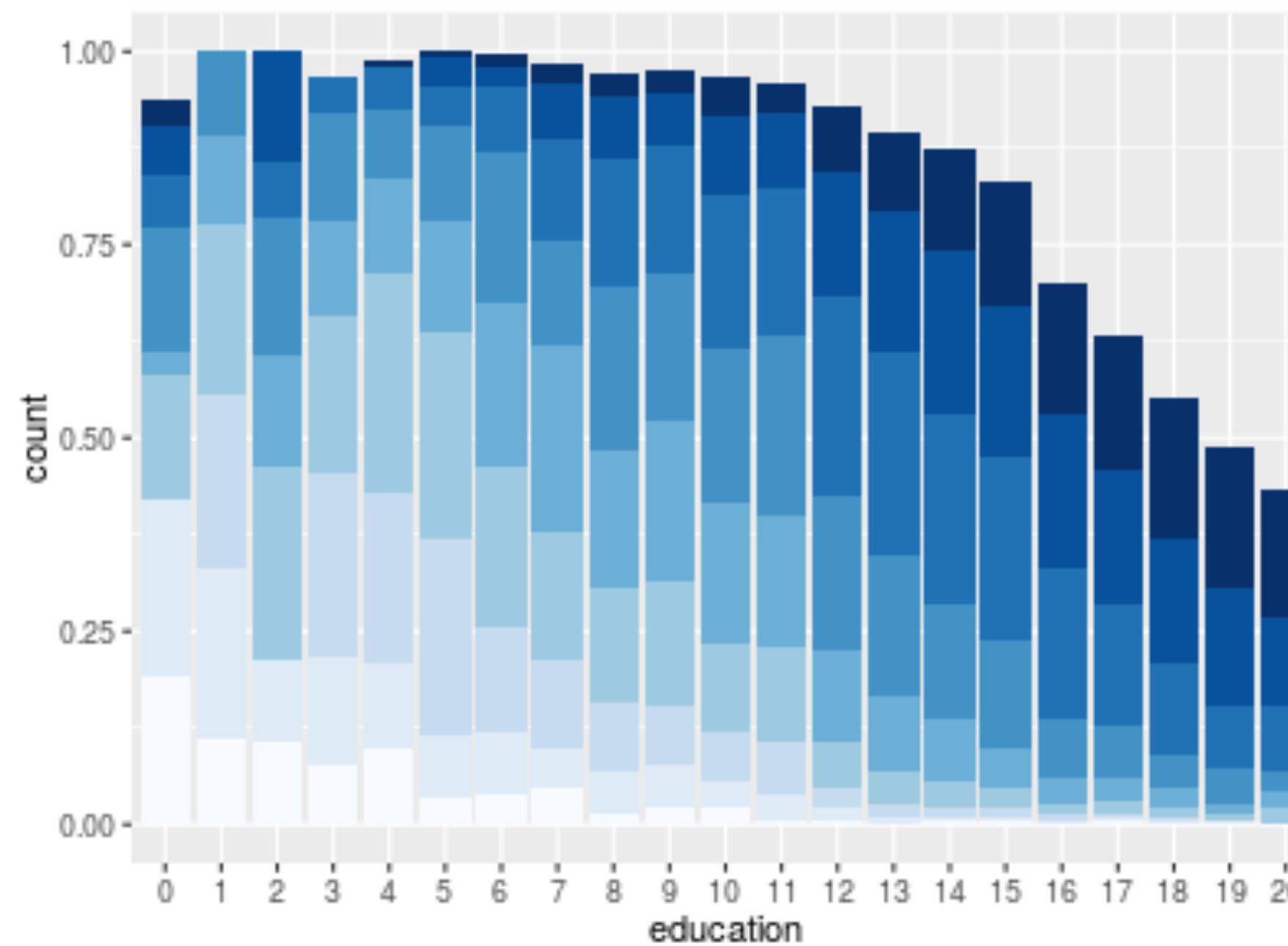


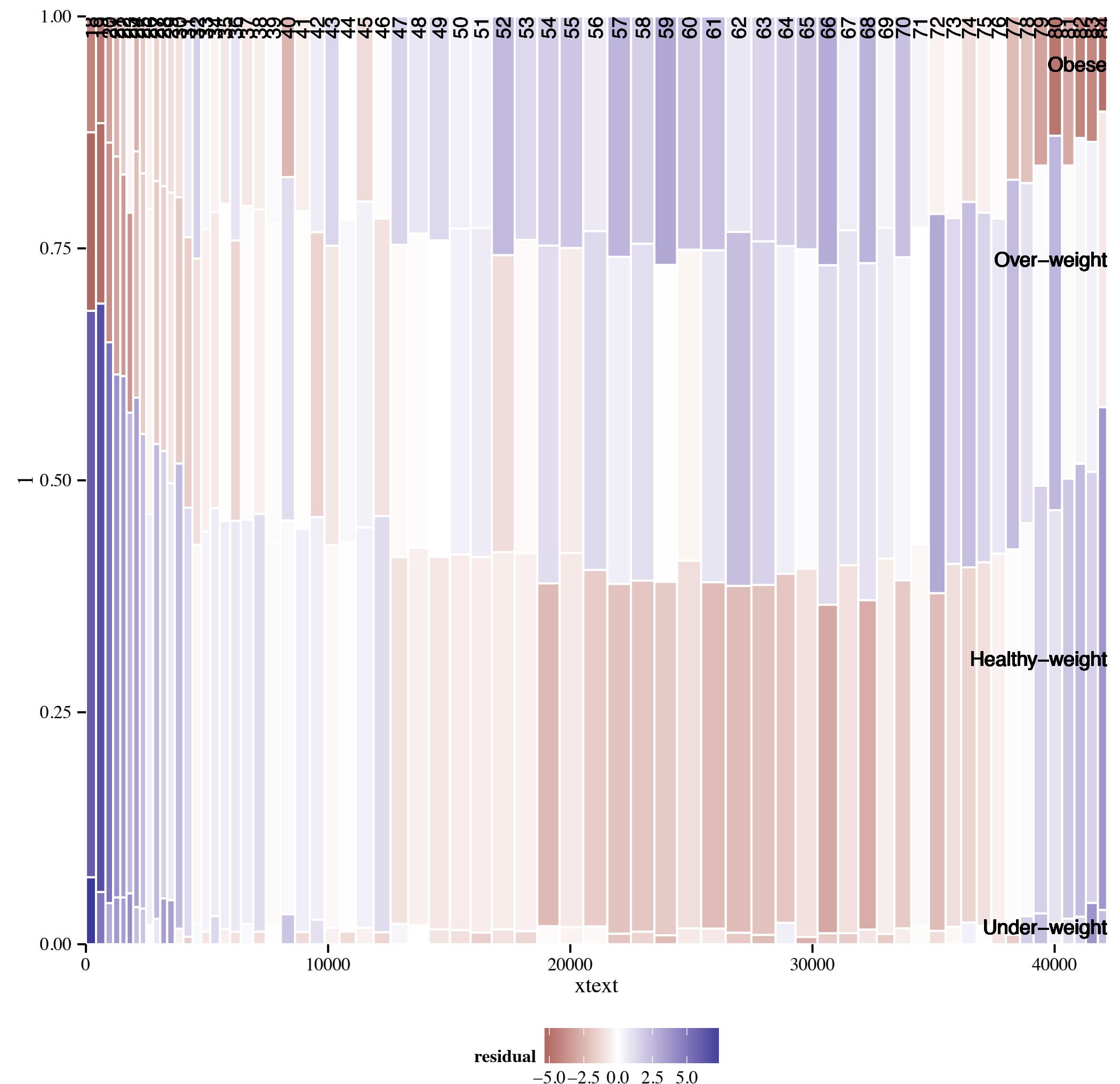
Communication  
& Perception

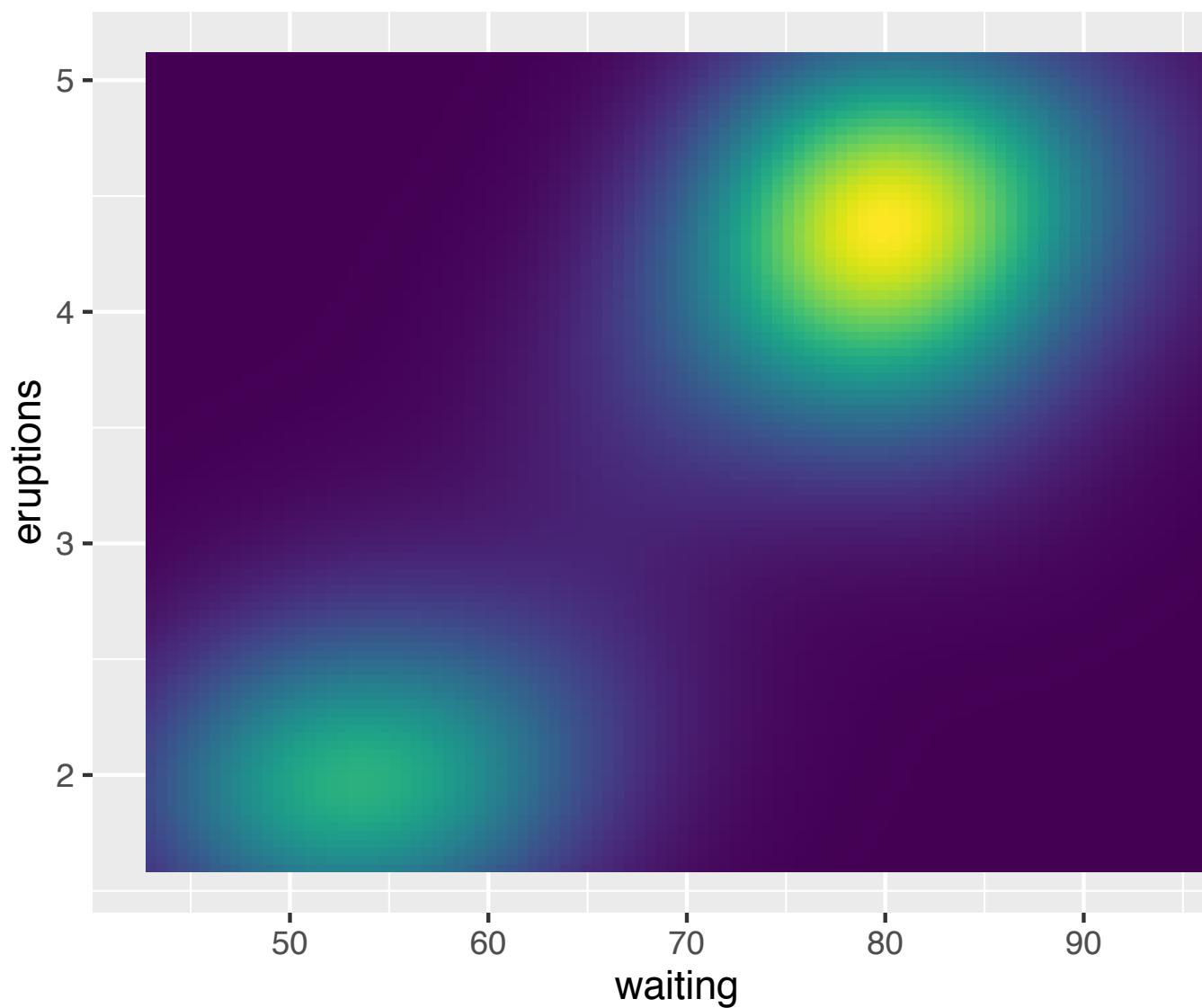


Element	Description
Data	The dataset being plotted.
Aesthetics	The scales onto which we <i>map</i> our data.
Geometries	The visual elements used for our data.

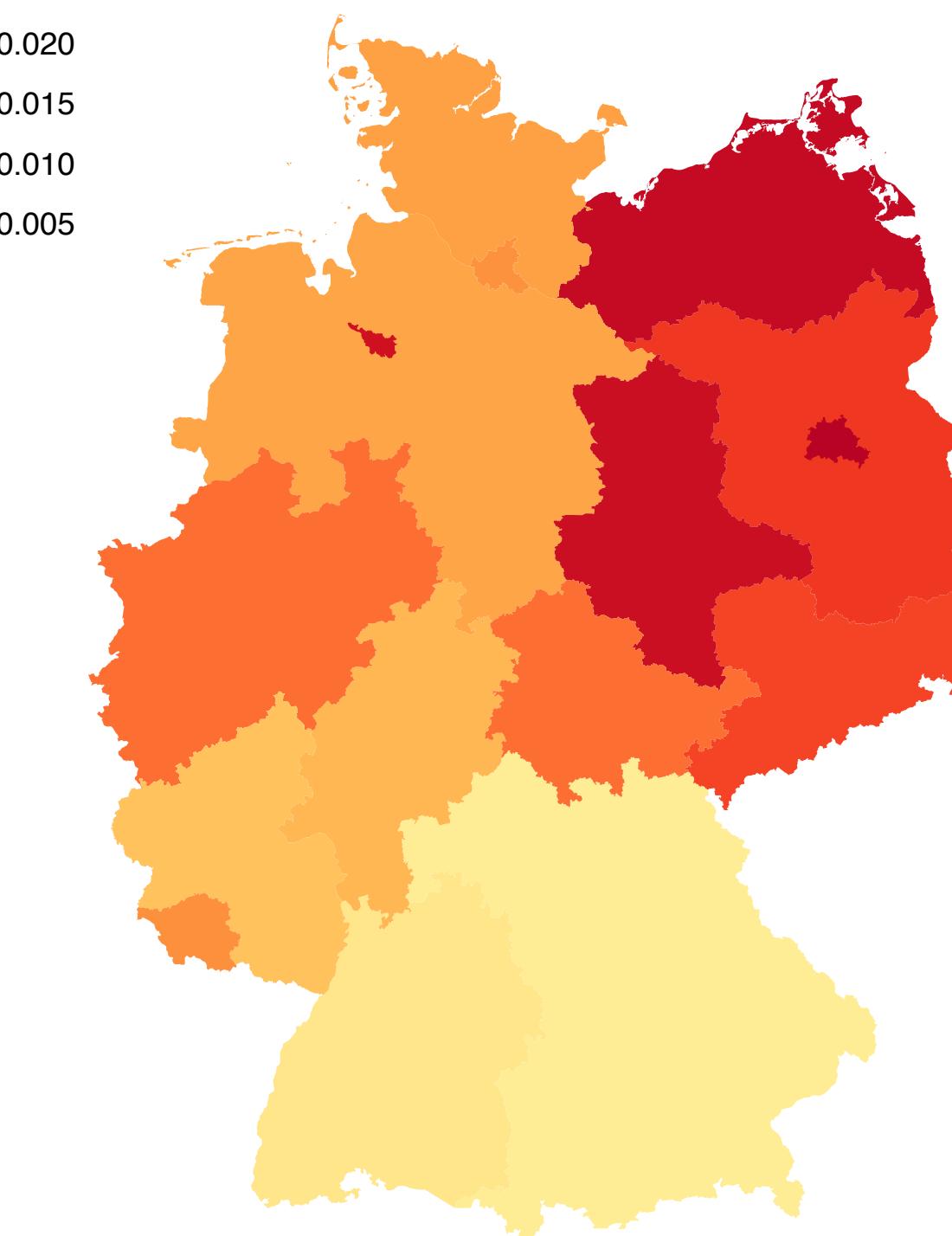
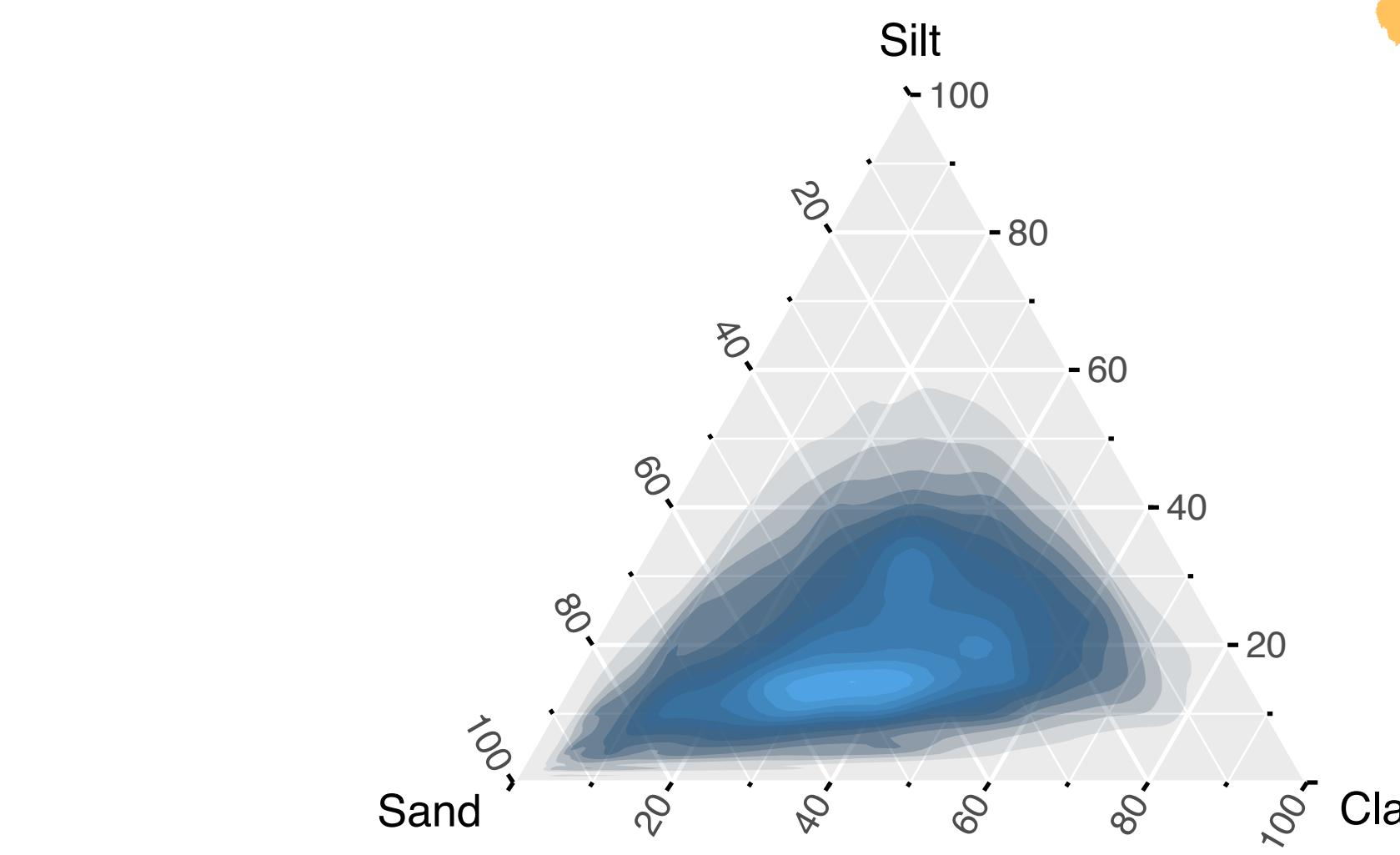
Element	Description
Data	The dataset being plotted.
Aesthetics	The scales onto which we <i>map</i> our data.
Geometries	The visual elements used for our data.
Facets	Plotting small multiples.
Statistics	Representations of our data to aid understanding.
Coordinates	The space on which the data will be plotted.
Themes	All non-data ink.



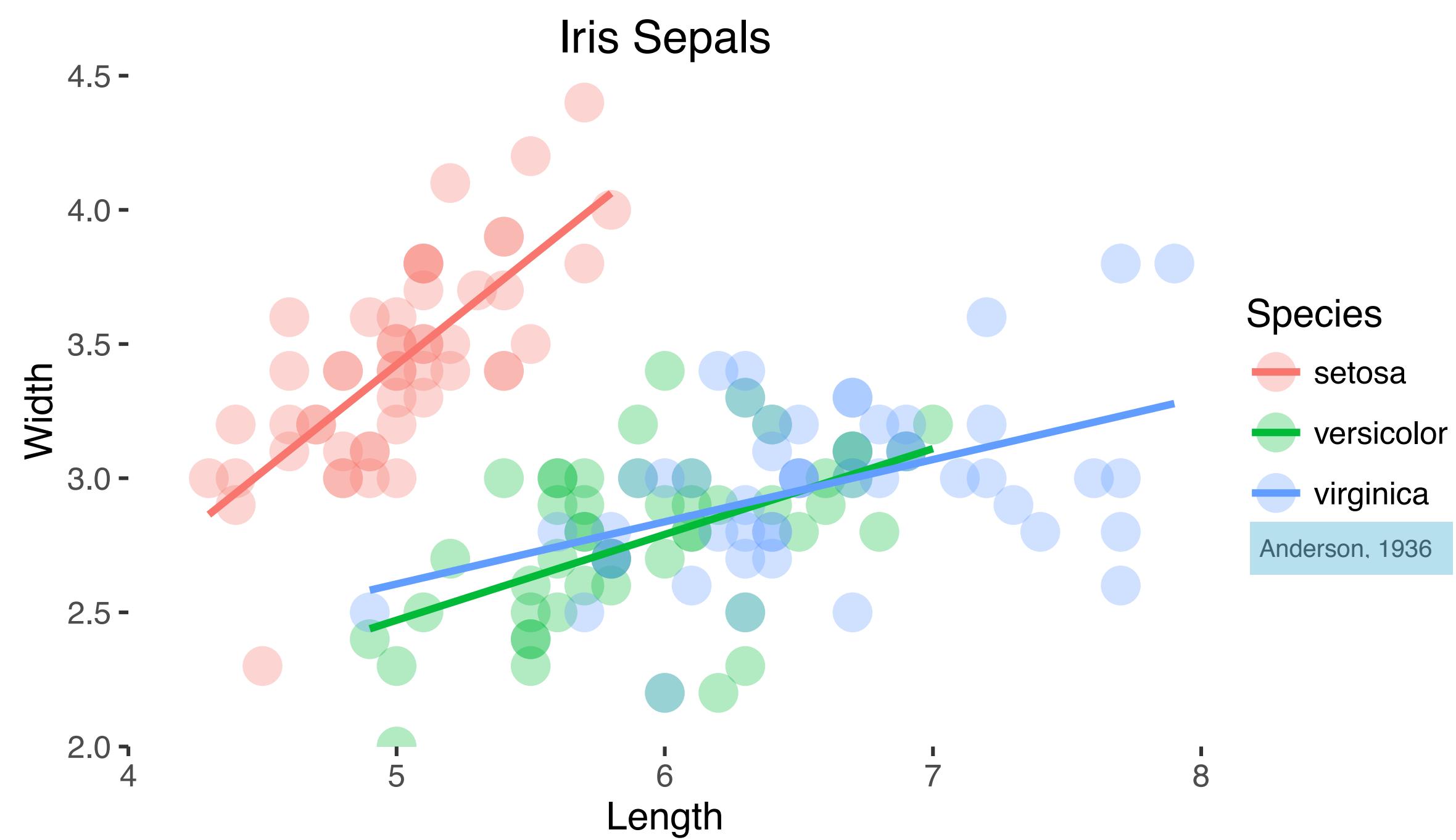


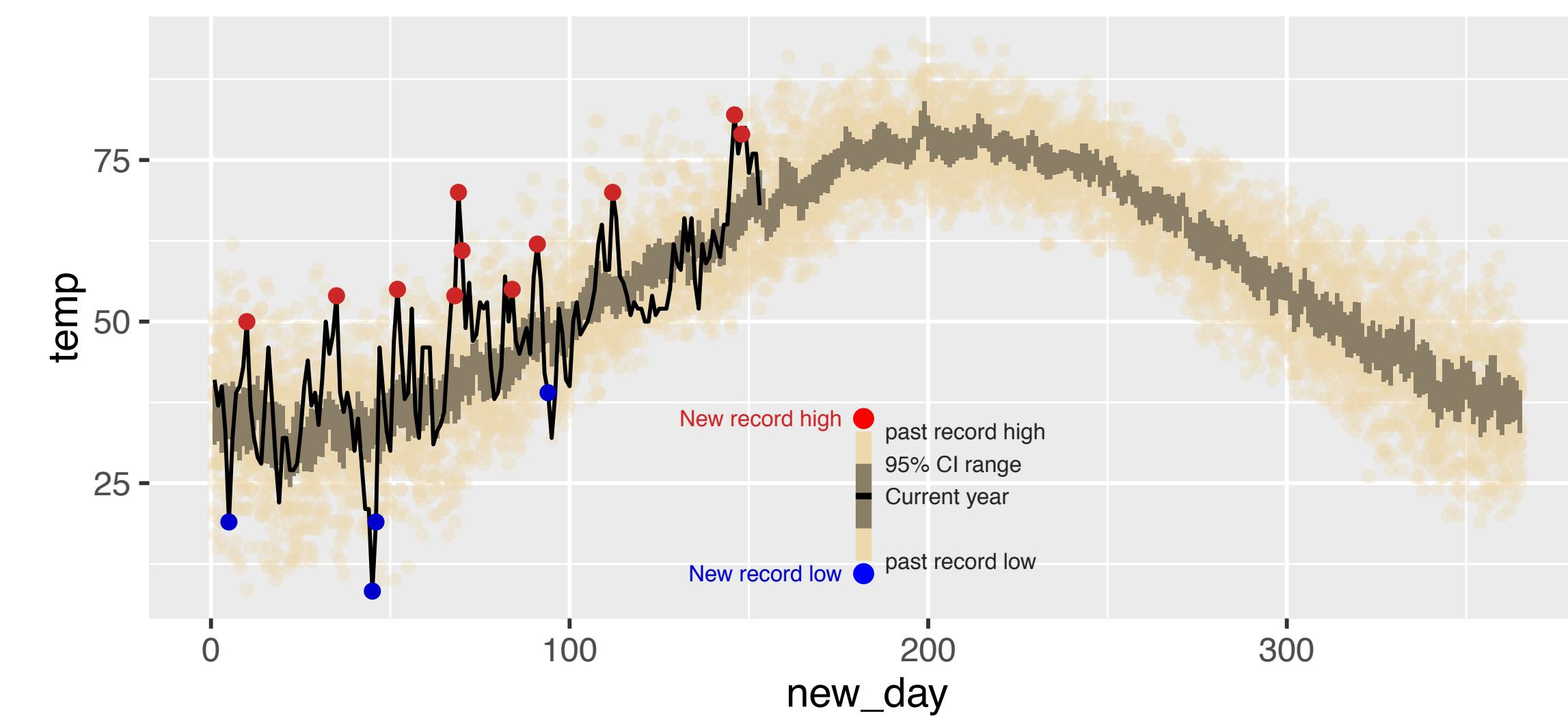
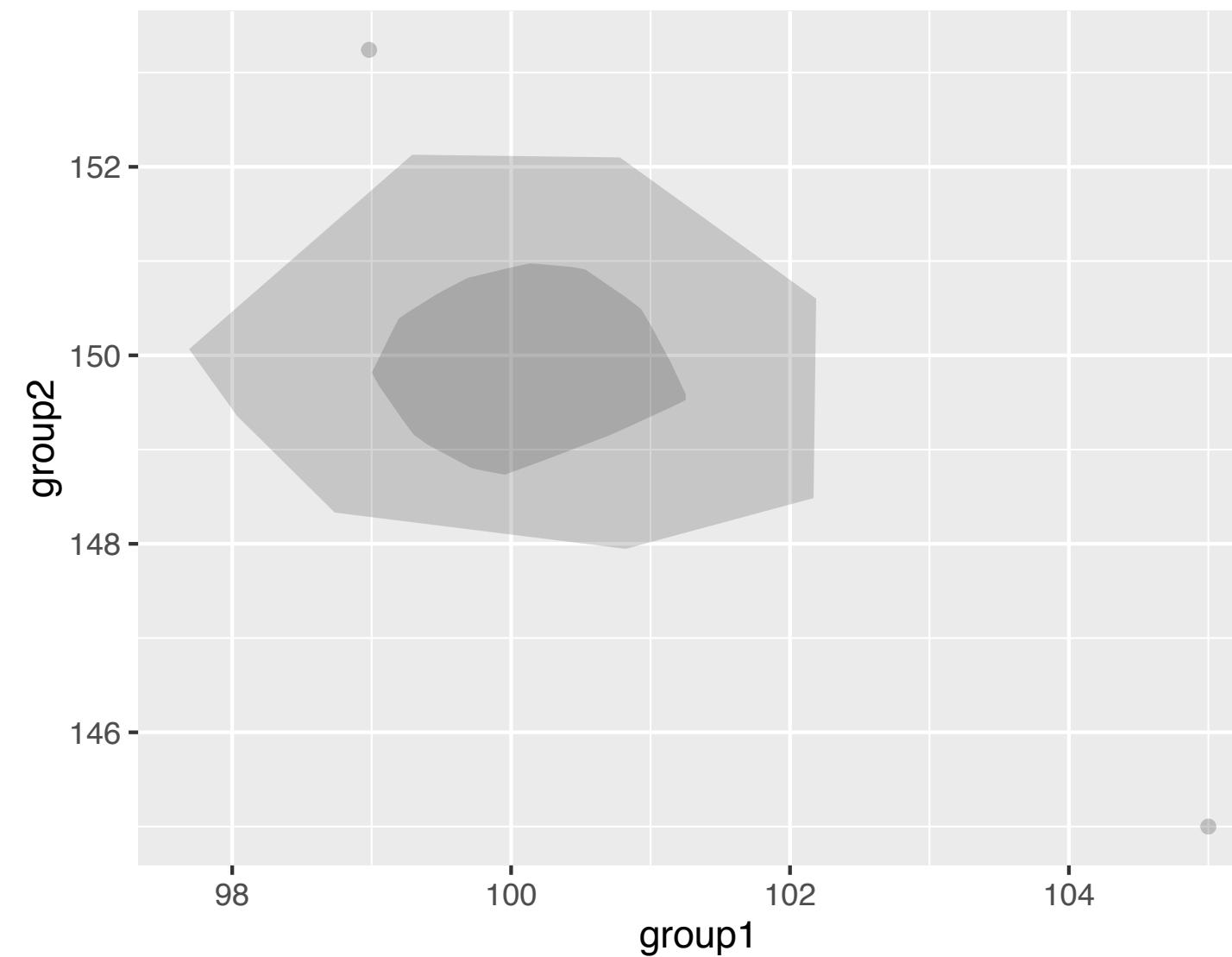


density  
0.025  
0.020  
0.015  
0.010  
0.005



Unemployment (%)  
12  
9  
6  
3







DATA VISUALIZATION WITH GGPLOT2

**Thank you!**