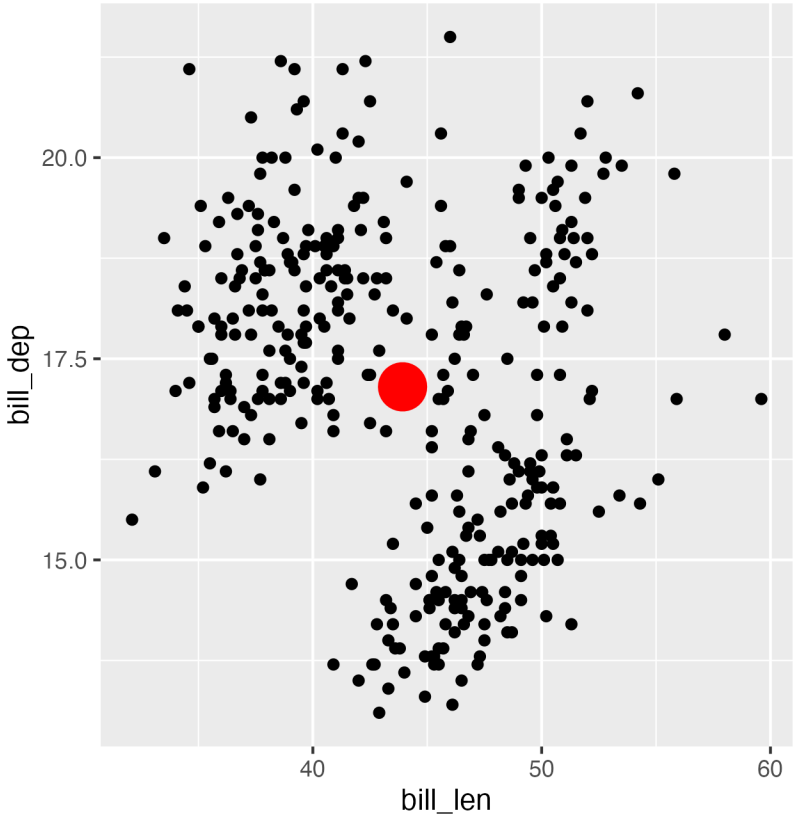
Posit DS Lab Notebook: 2026 JUL 21 - Extending ggplot2 and plotnine

Step 0. Get the job done with 'base' ggplot2.

```
penguins_means <- penguins |>
  summarise(
      mean_bill_len = mean(bill_len
      mean_bill_dep = mean(bill_dep
      )

penguins |>
  ggplot() +
  aes(x = bill_len,
      y = bill_dep) +
  geom_point() +
  geom_point(
    data = penguins_means,
    mapping = aes(
        x = mean_bill_len,
        y = mean_bill_dep
        ),
    color = "red",
    size = 8)
```



Posit DS Lab Notebook: 2026 JUL 21 - Extending ggplot2 and plotnine

Step 1. Define Compute. Test.

```
compute_group_means <-
     function(data, scales){

  data |>
    summarise(
      x = mean(x, na.rm = T),
      y = mean(y, na.rm = T)
      )

}

penguins |>
  select(x = bill_len,
         y = bill_dep) |>
  compute_group_means()
```

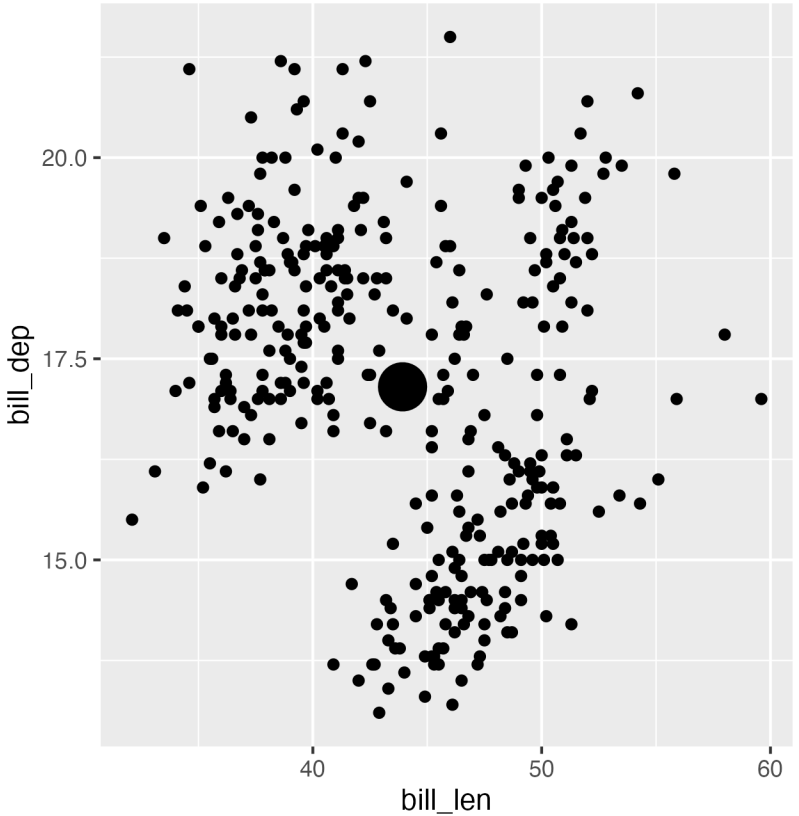
x	y
43.92193	17.15117

Posit DS Lab Notebook: 2026 JUL 21 - Extending ggplot2 and plotnine

Step 2. Define StatMeans. Test.

```
StatMeans <-
  ggproto(
      `_class` = "StatMeans",
      `_inherit` = Stat,
      compute_group =
           compute_group_means,
      required_aes = c("x", "y")
      )


ggplot(penguins) +
  aes(x = bill_len,
      y = bill_dep) +
  geom_point() +
  geom_point(
      stat = StatMeans,
      size = 8
      )
```

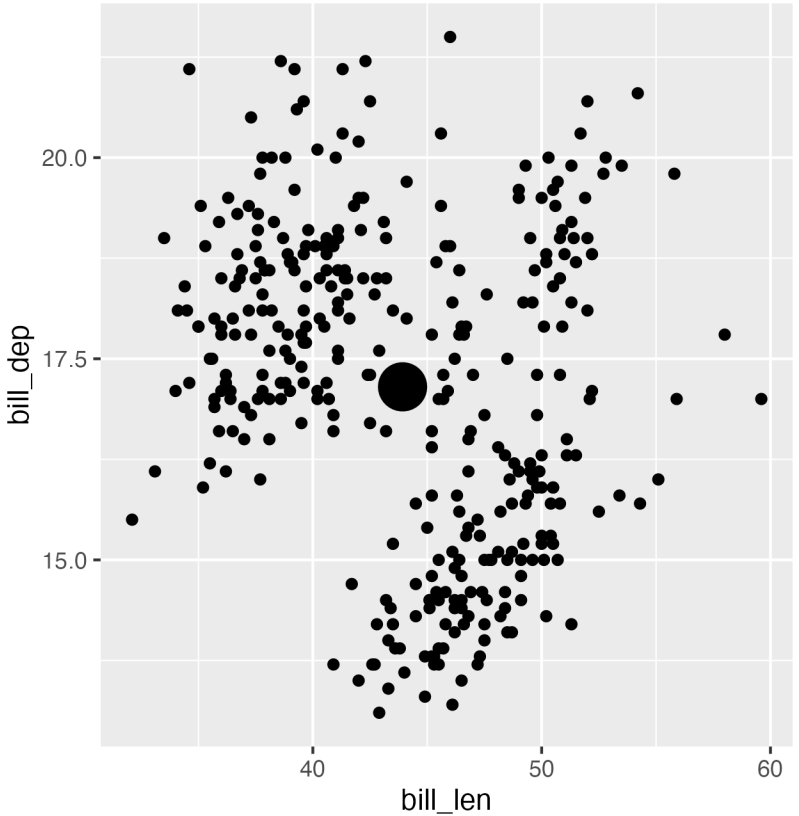


Posit DS Lab Notebook: 2026 JUL 21 - Extending ggplot2 and plotnine

Step 4. Define user-facing function. Enjoy!

```
geom_means <-
  make_constructor( # >v4.0 ggplot2
    x = GeomPoint,
    stat = "means"
    )


ggplot(penguins) +
  aes(bill_len, bill_dep) +
  geom_point() +
  geom_means(size = 8)
```



Posit DS Lab Notebook: 2026 JUL 21 - Extending ggplot2 and plotnine

Step 4.b Test out group-wise compute (for free!)

```
ggplot(penguins) +
  aes(bill_len, bill_dep) +
  geom_point() +
  geom_means(size = 8) +
  aes(color = species)
```