Visualizing distributions 1

Claus O. Wilke

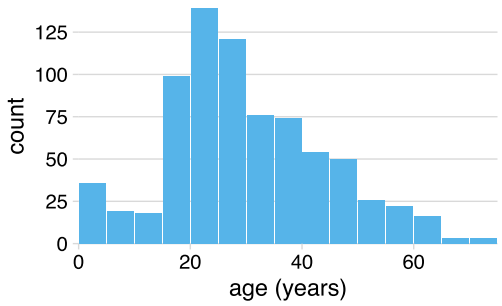
last updated: 2021-01-18

Passengers on the Titanic

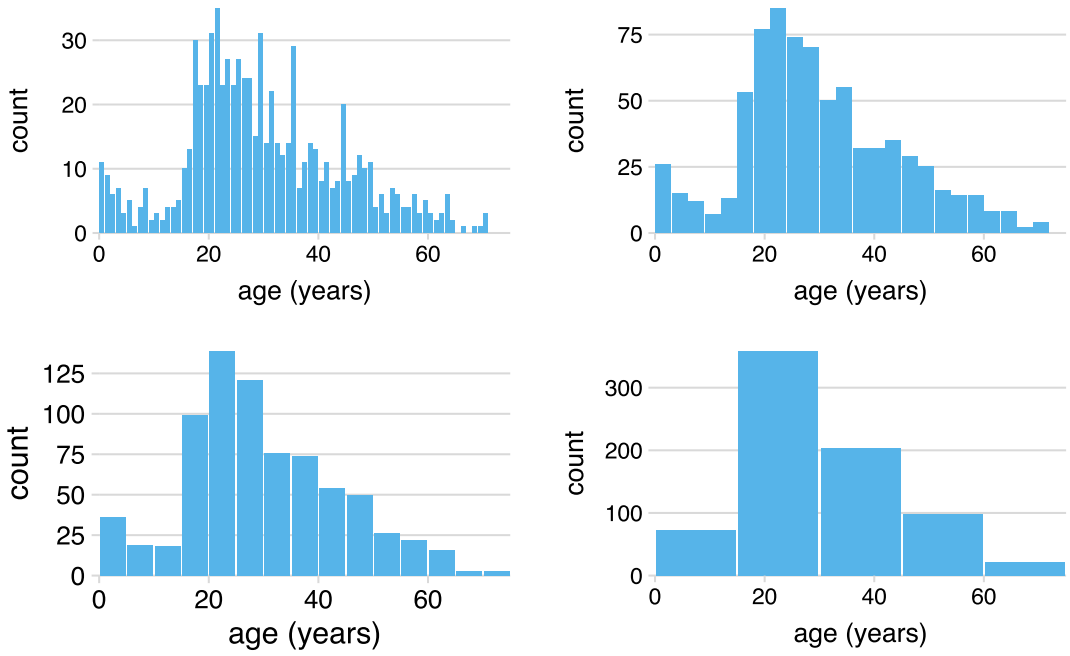
age	sex	class	survived
0.17	female	3rd	survived
0.33	male	3rd	died
0.80	male	2nd	survived
0.83	male	2nd	survived
0.83	male	3rd	survived
0.92	male	1st	survived
1.00	female	2nd	survived
1.00	female	3rd	survived
1.00	male	2nd	survived
1.00	male	2nd	survived

Histogram: Define bins and count cases

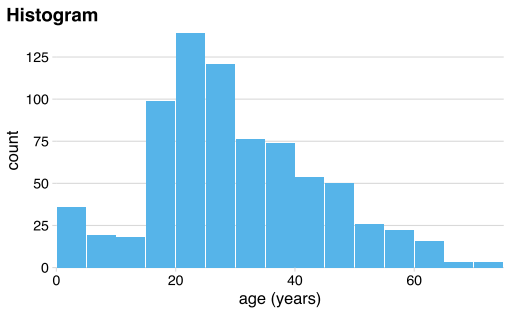
age range	count
0–5	36
6–10	19
11–15	18
16–20	99
21–25	139
26–30	121
31–35	76
36–40	74

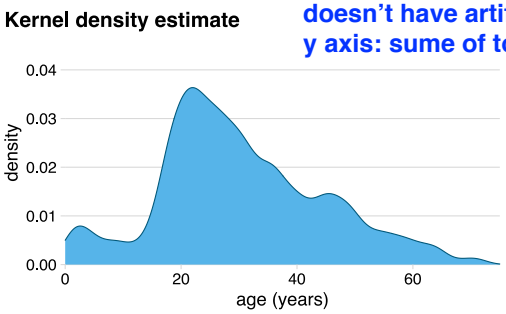


Histograms depend on the chosen bin width



Alternative to histogram: Kernel density estimate (KDE)

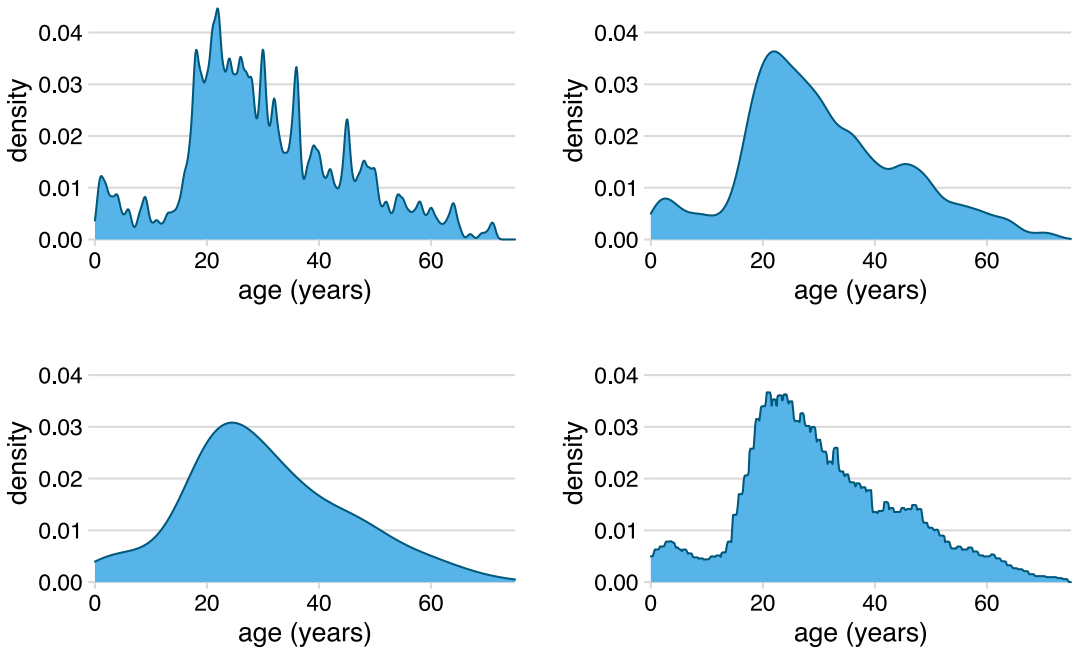




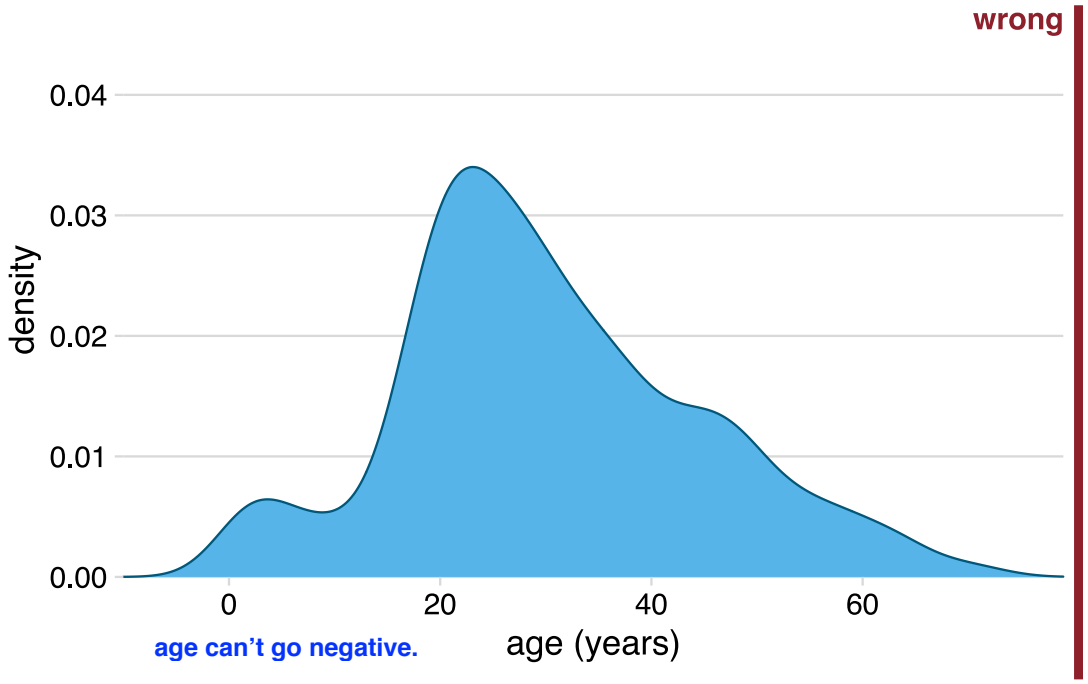
Continuous histogram doesn't have artificial blocking
y axis: sume of total area = 1

Histograms show raw counts, KDEs show proportions. (Total area = 1)

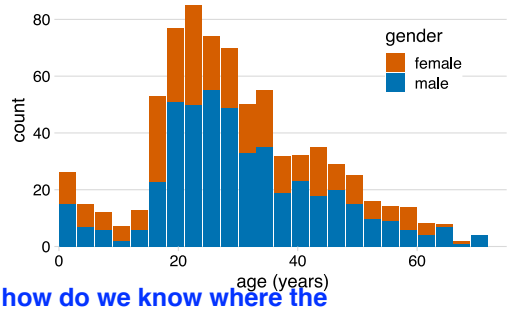
KDEs also depend on parameter settings



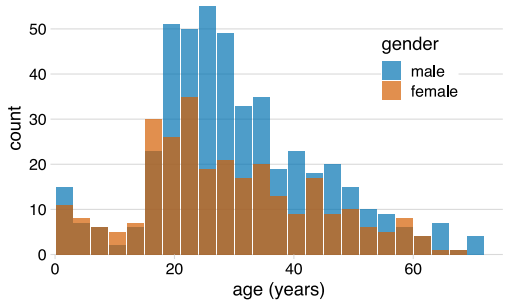
Careful: KDEs can show non-sensical data



Careful: Are bars stacked or overlapping?



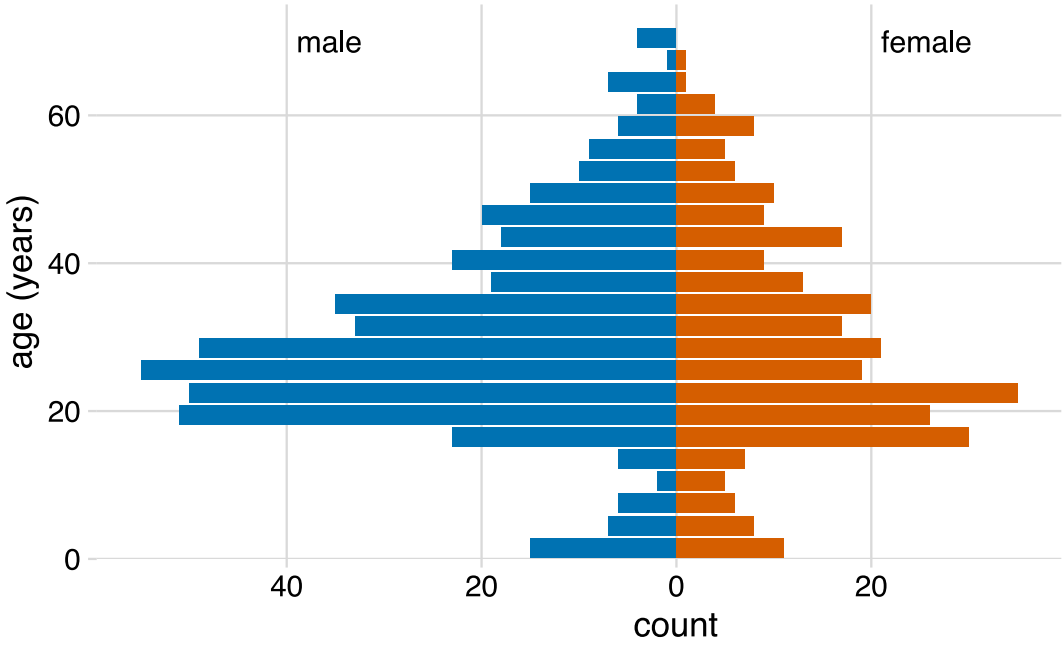
Stacked: how do we know where the female starts? can't see the female distribution



Overlapping

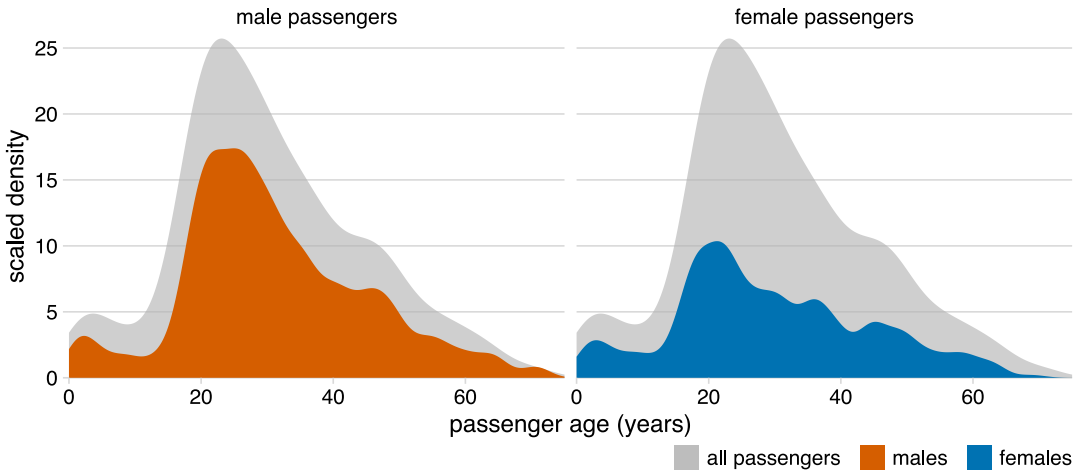
Stacked or overlapping histograms are rarely a good choice.

Alternatively: Age pyramid



hgistogram: works with two variables

Alternatively: KDEs showing proportions of total

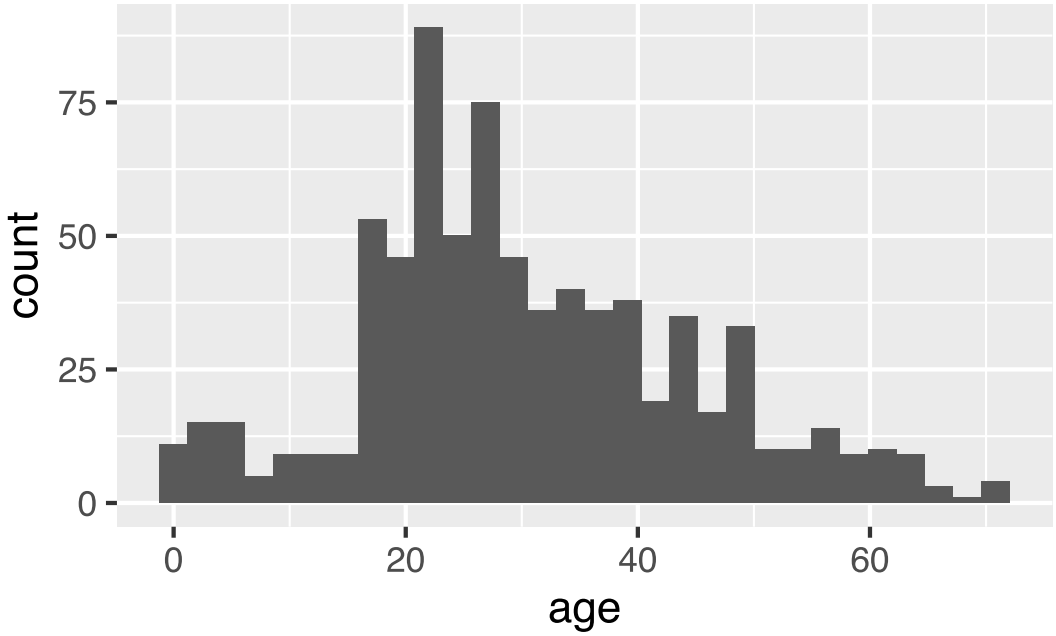


scaled y axis: so total number of passangers

Making histograms with ggplot: `geom_histogram()`

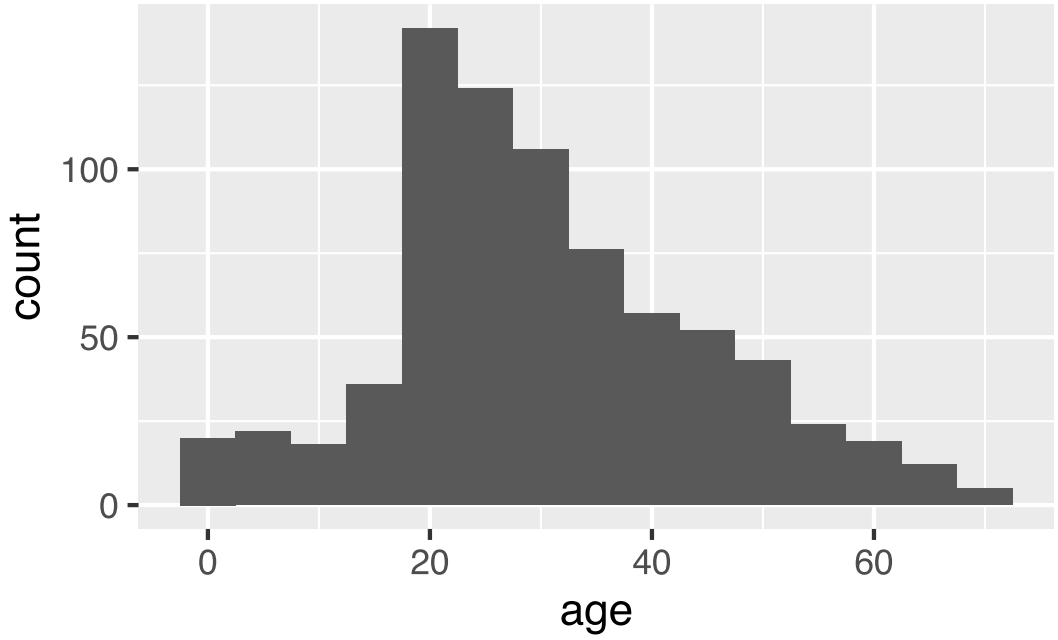
```
ggplot(titanic, aes(age)) +
  geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better val
```



Setting the bin width

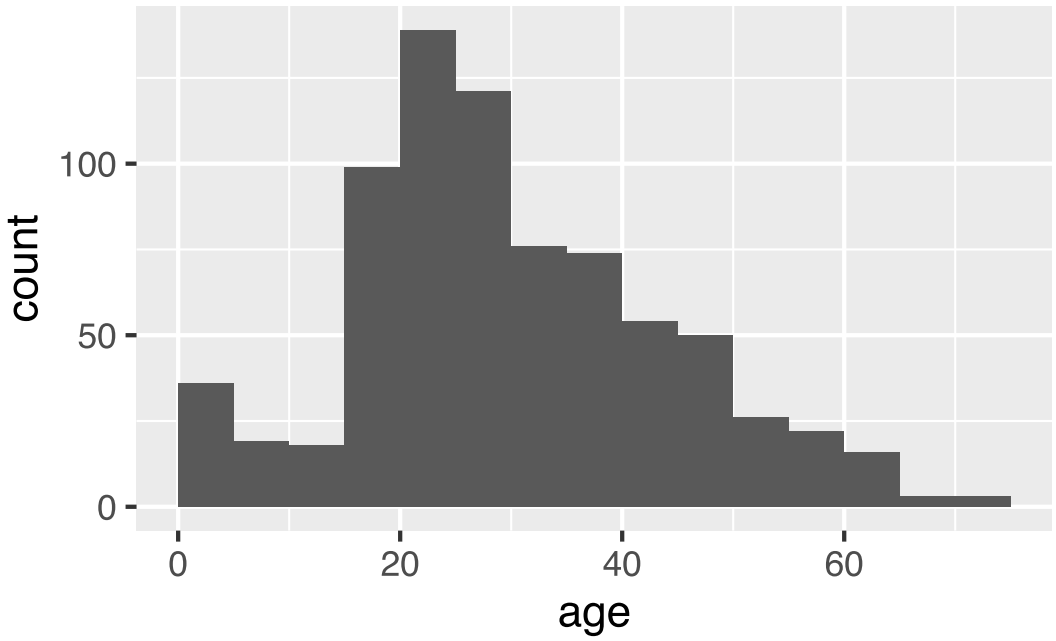
```
ggplot(titanic, aes(age)) +
  geom_histogram(binwidth = 5)
```



Do you like the bin placement?

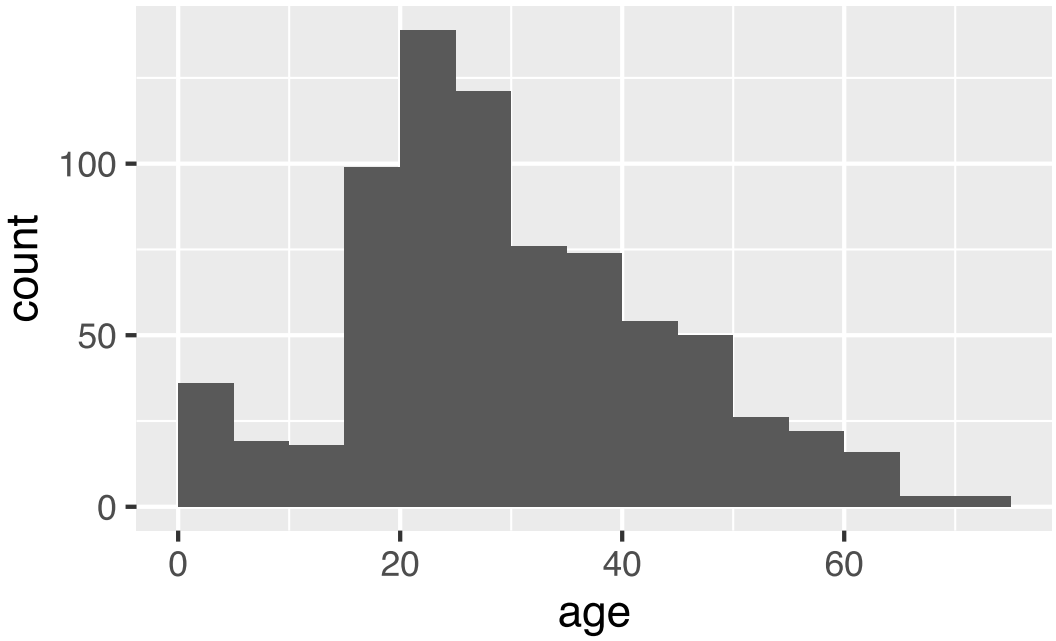
Always set the center as well

```
ggplot(titanic, aes(age)) +
  geom_histogram(
    binwidth = 5,   # width of the bins
    center = 2.5    # center of the bin
  )
```



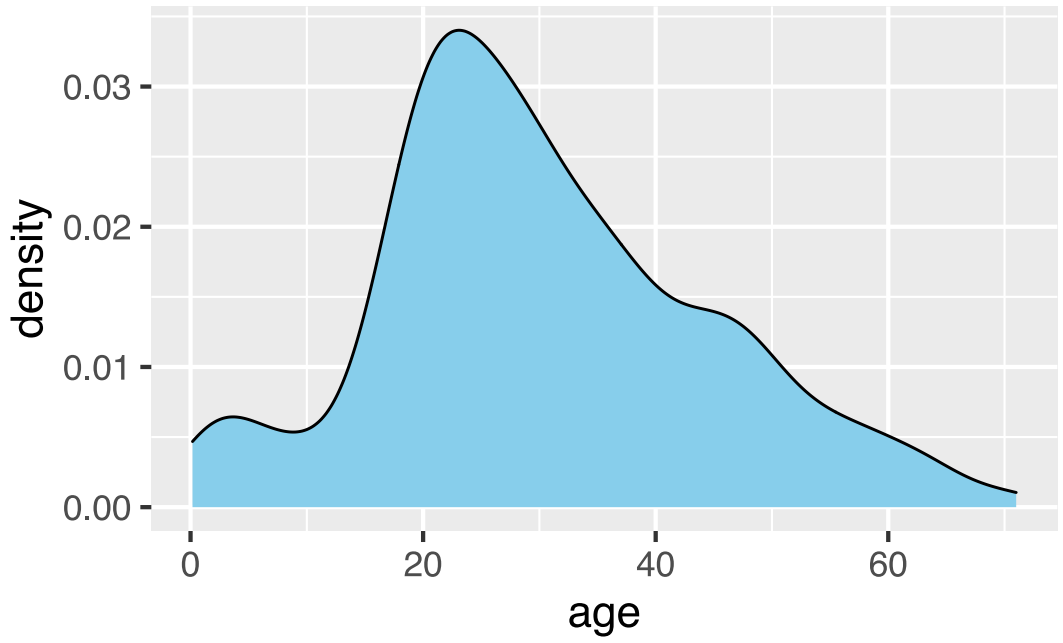
Always set the center as well

```
ggplot(titanic, aes(age)) +
  geom_histogram(
    binwidth = 5,   # width of the bins
    center = 10.5   # center of the bin
  )
```



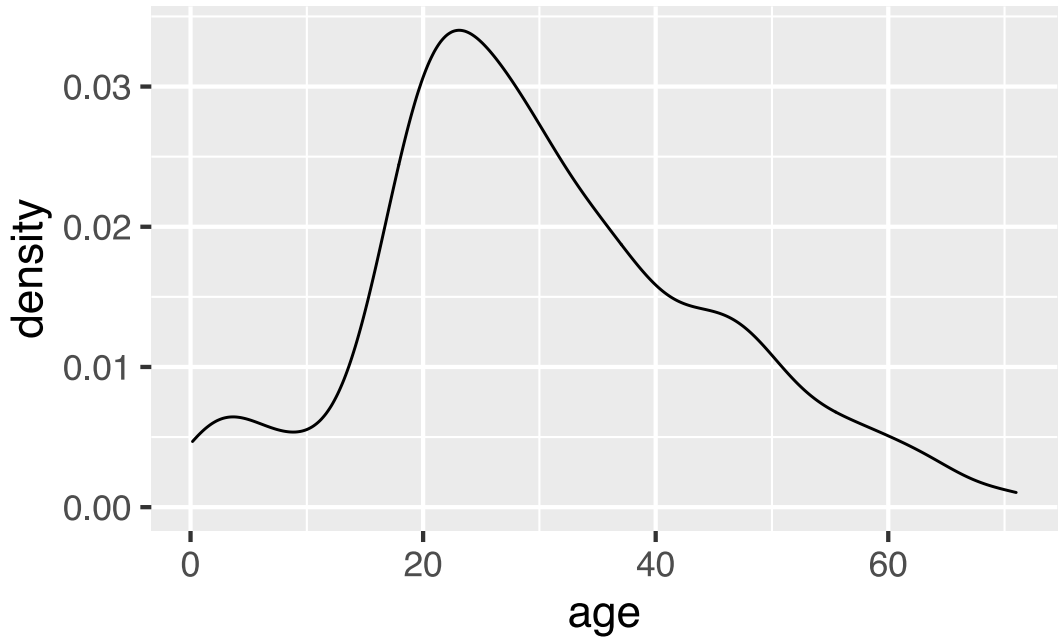
Making density plots with ggplot: `geom_density()`

```
ggplot(titanic, aes(age)) +
  geom_density(fill = "skyblue")
```



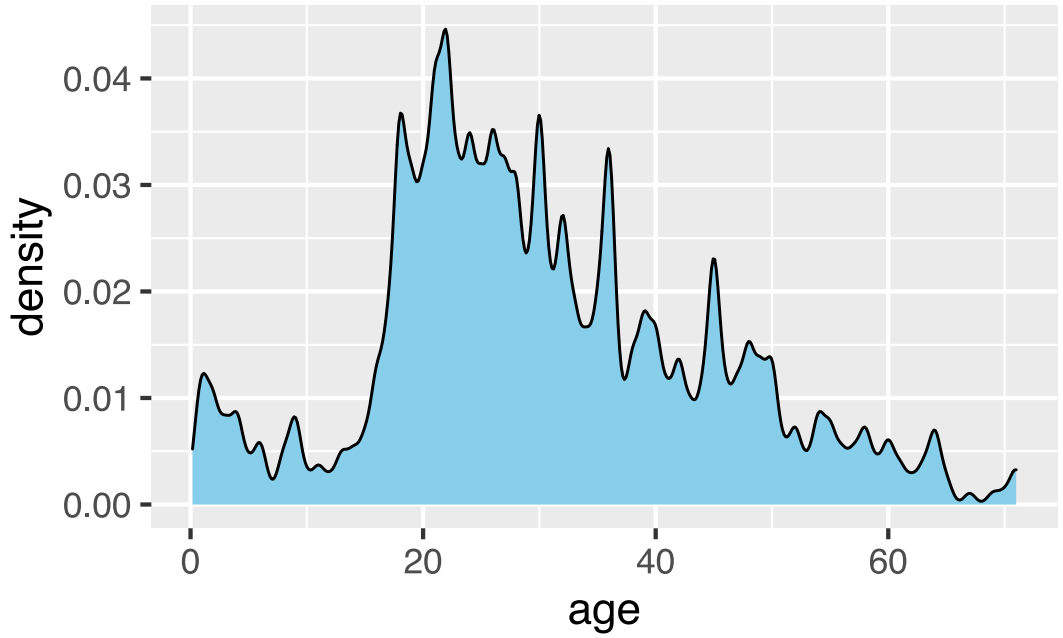
Making density plots with ggplot: `geom_density()`

```
ggplot(titanic, aes(age)) +
  geom_density() # without fill
```



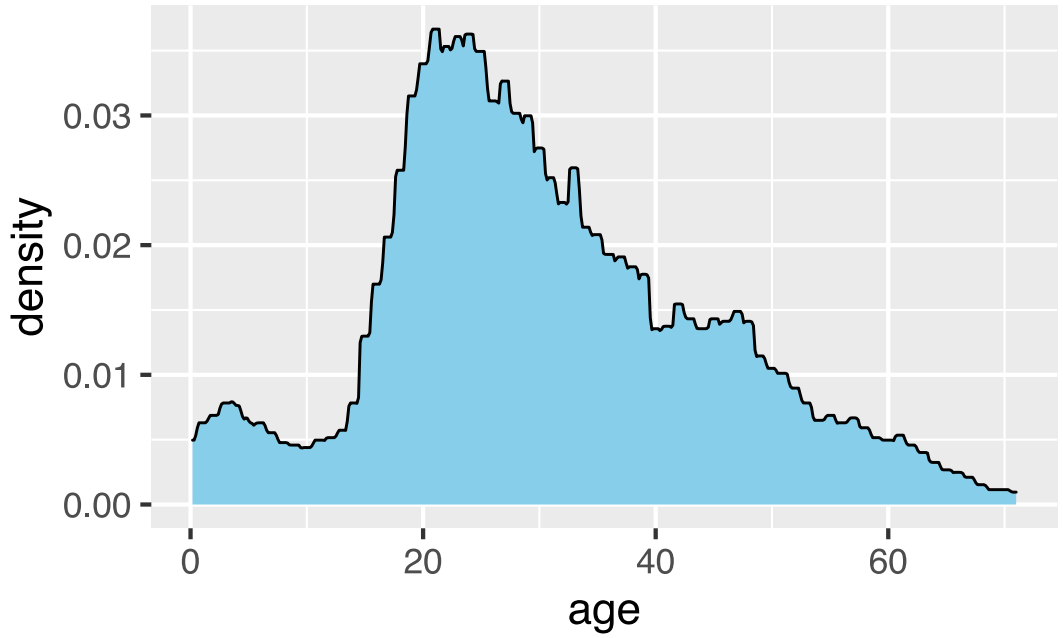
Modifying bandwidth (bw) and kernel parameters

```
ggplot(titanic, aes(age)) +
  geom_density(
    fill = "skyblue",
    bw = 0.5,                # a small bandwidth
    kernel = "gaussian"      # Gaussian kernel (the
  )
```



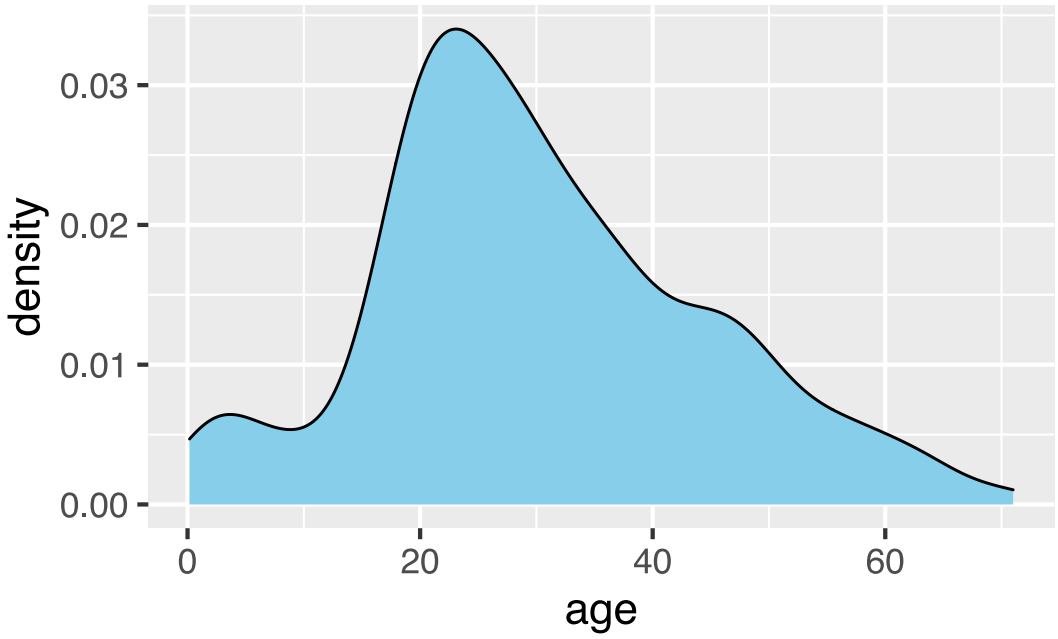
Modifying bandwidth (bw) and kernel parameters

```
ggplot(titanic, aes(age)) +
  geom_density(
    fill = "skyblue",
    bw = 2,                    # a moderate bandwidt
    kernel = "rectangular"   # rectangular kernel
  )
```



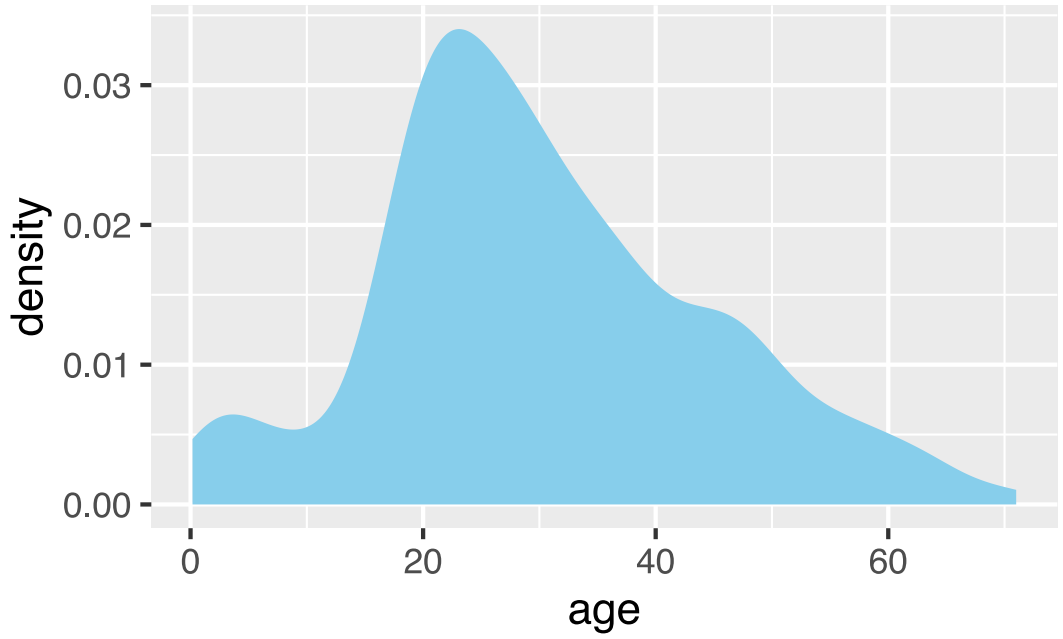
Statistical transformations (stats) can be set explicitly

```
ggplot(titanic, aes(age)) +
  geom_density(
    stat = "density",    # the default for geom_d
    fill = "skyblue"
  )
```



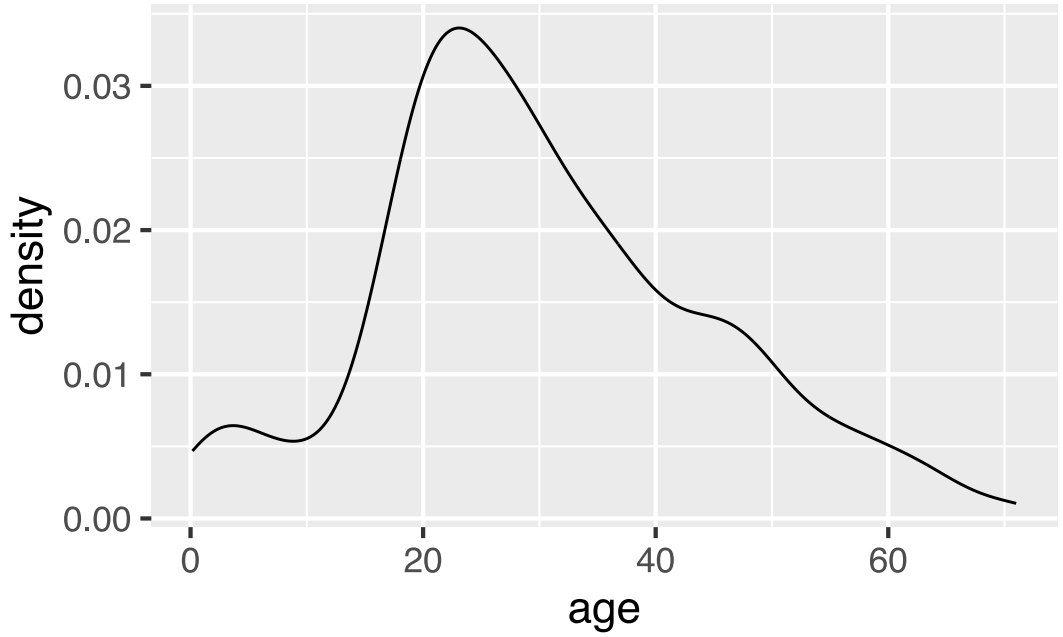
Statistical transformations (stats) can be set explicitly

```
ggplot(titanic, aes(age)) +
  geom_area(  # geom_area() does not normally use
    stat = "density",
    fill = "skyblue"
  )
```



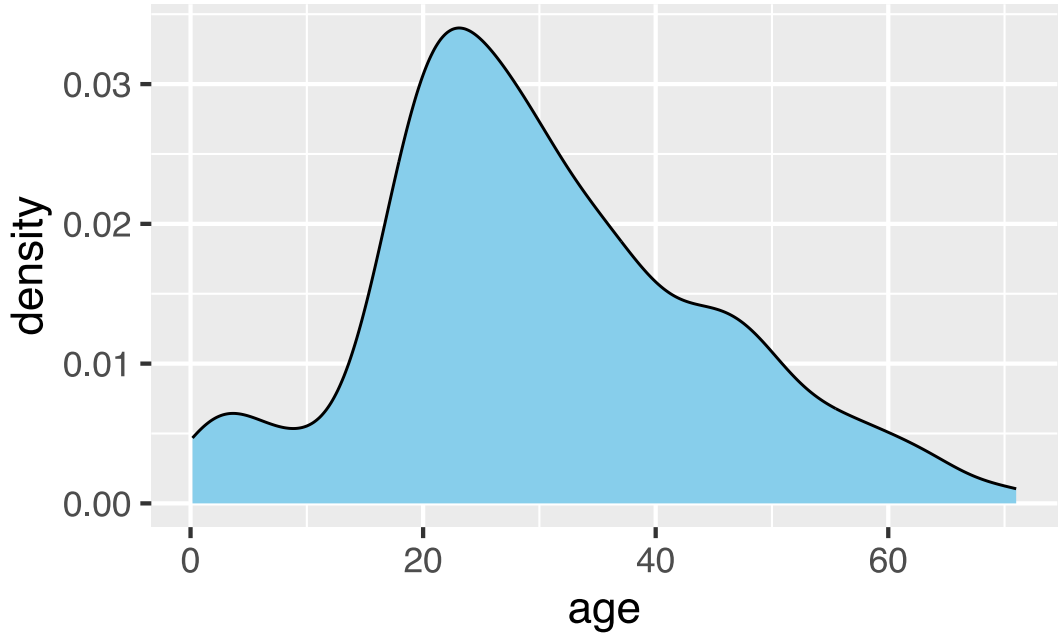
Statistical transformations (stats) can be set explicitly

```
ggplot(titanic, aes(age)) +
  geom_line(  # neither does geom_line()
    stat = "density"
  )
```



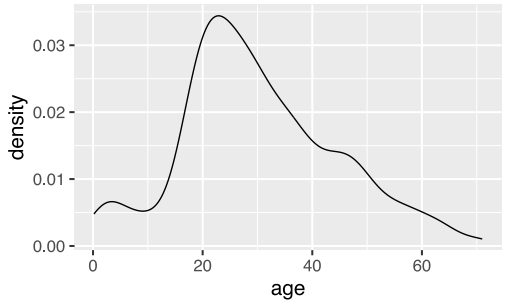
Statistical transformations (stats) can be set explicitly

```
ggplot(titanic, aes(age)) +
  # we can use multiple geoms on top of each othe
  geom_area(stat = "density", fill = "skyblue") +
  geom_line(stat = "density")
```

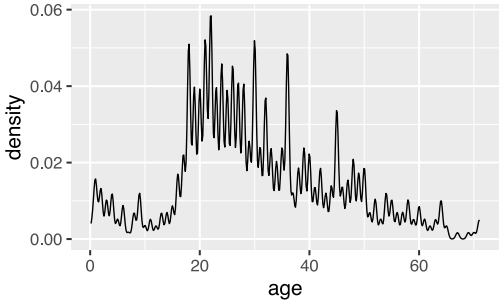


Parameters are handed through to the stat

```
ggplot(titanic, aes(age
  geom_line(stat = "den
```



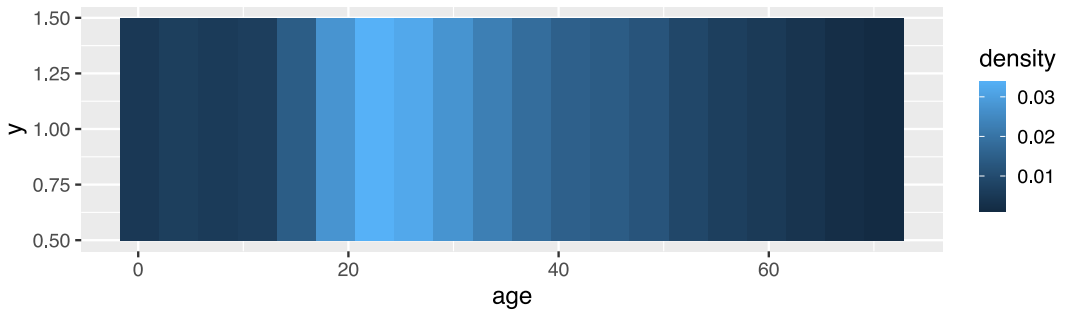

```
ggplot(titanic, aes(age
  geom_line(stat = "den
```



Here, `bw` is a parameter of `stat_density()`, not of `geom_line()`.

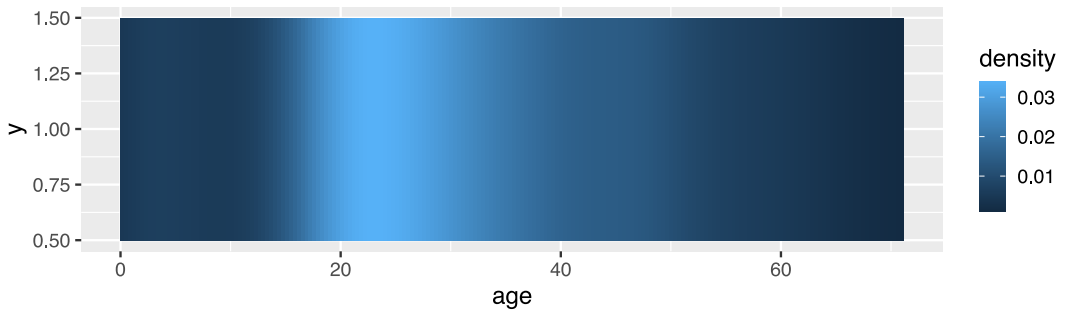
We can explicitly map results from stat computations

```
ggplot(titanic, aes(age)) +
  geom_tile( # geom_tile() draws rectangular colo
    aes(
      y = 1, # draw all tiles at the same y locat
      fill = after_stat(density)  # use computed
    ),
    stat = "density",
    n = 20    # number of points calculated by sta
  )
```



We can explicitly map results from stat computations

```
ggplot(titanic, aes(age)) +
  geom_tile( # geom_tile() draws rectangular colo
    aes(
      y = 1, # draw all tiles at the same y locat
      fill = after_stat(density)  # use computed
    ),
    stat = "density",
    n = 200   # number of points calculated by st
  )
```



Further reading

- Fundamentals of Data Visualization: Chapter 7: Visualizing distributions
- Data Visualization—A Practical Introduction: 4.6 Histograms and density plots
- **ggplot2** reference documentation: `geom_histogram()`
- **ggplot2** reference documentation: `geom_density()`