6_Summarized_Data_Distributions

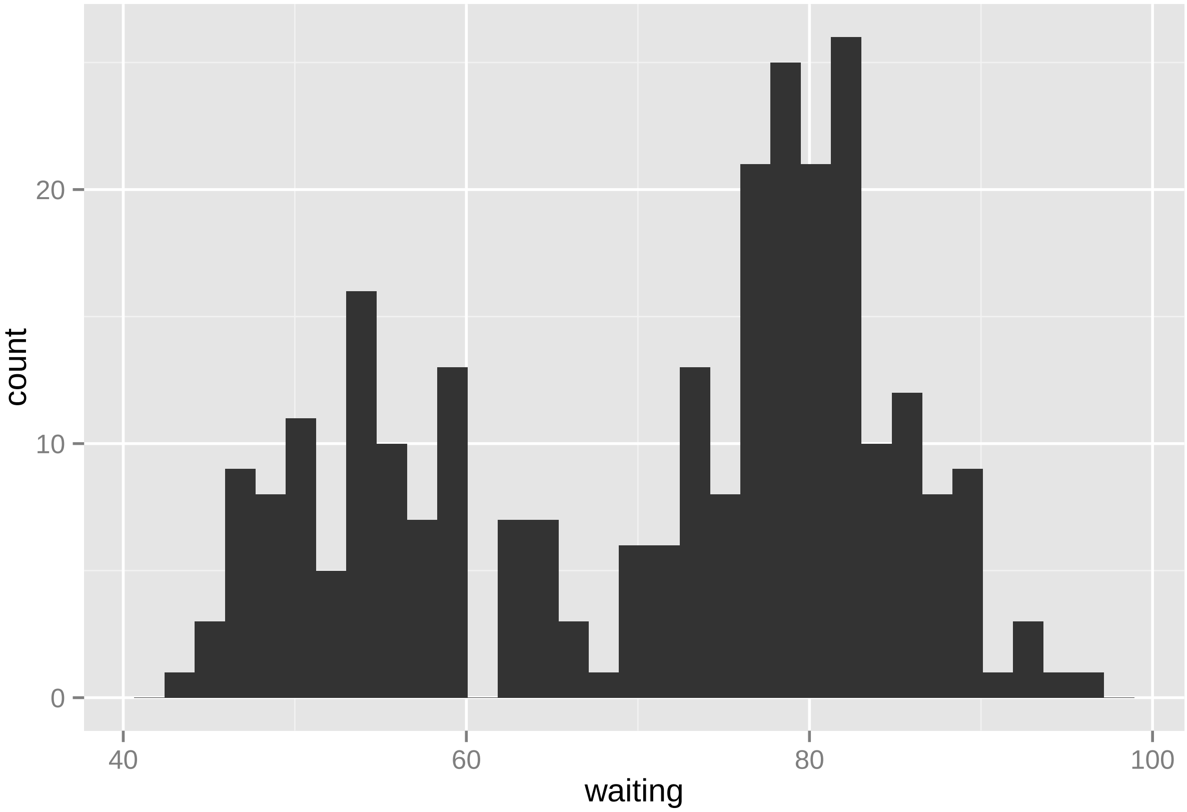
Gino Tesei

December 13, 2014

1. Making a Basic Histogram

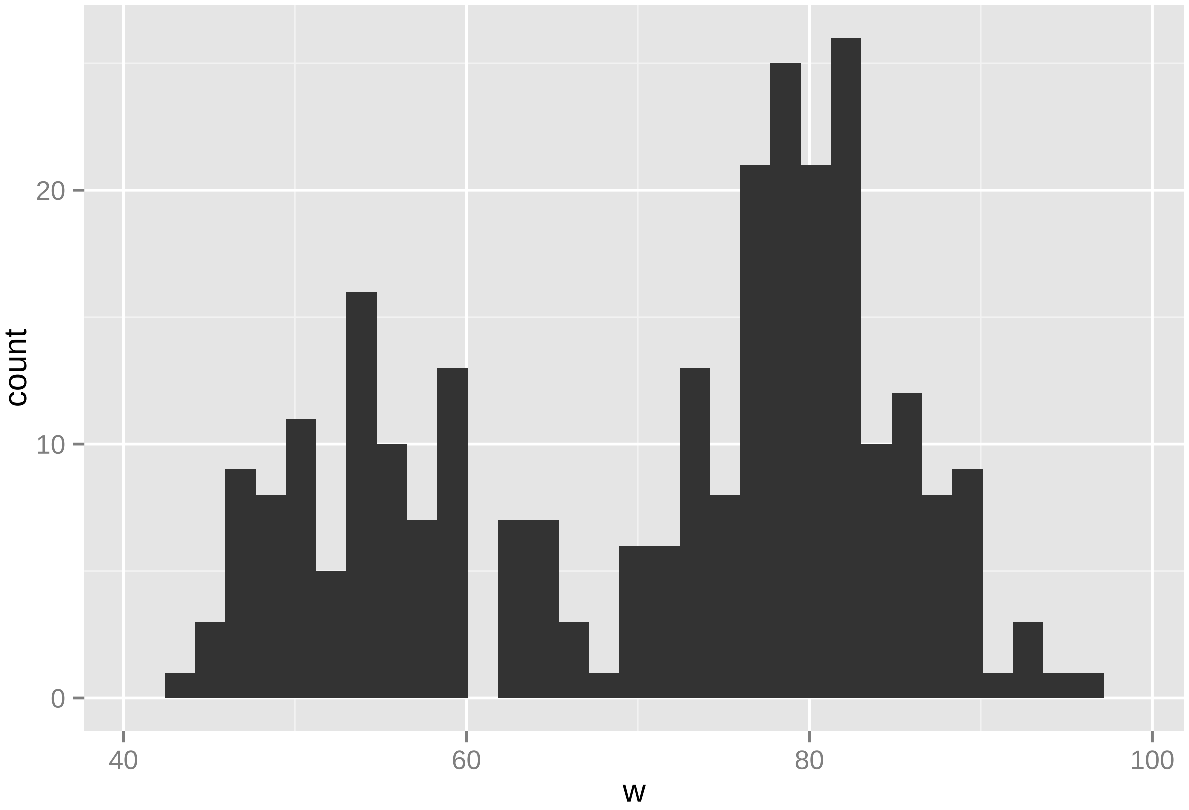
```
library(ggplot2)
library(gcookbook) # For the data set

library(plyr) ##

ggplot( faithful, aes( x = waiting)) + geom_histogram()
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

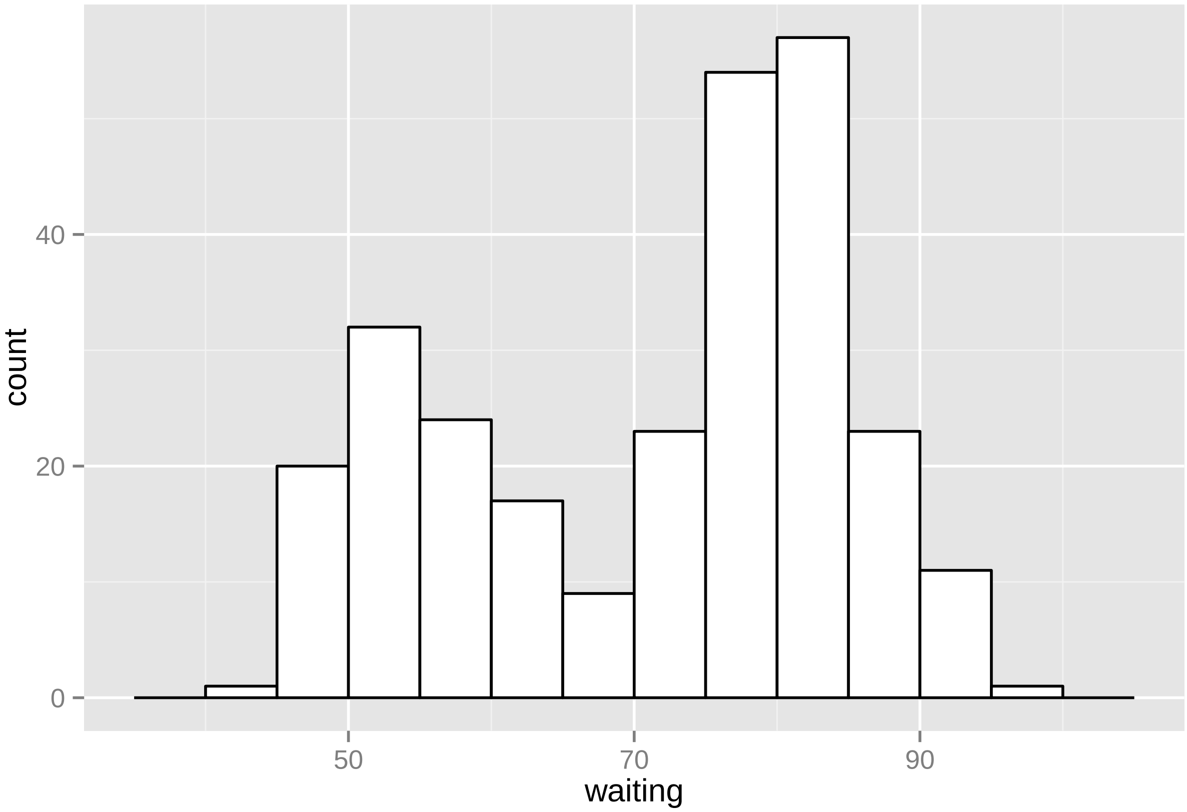


```
# If you just want to get a quick look at some data that isn't in a data frame, you can get the same re
w <- faithful$waiting
ggplot( NULL, aes( x = w)) + geom_histogram()
```

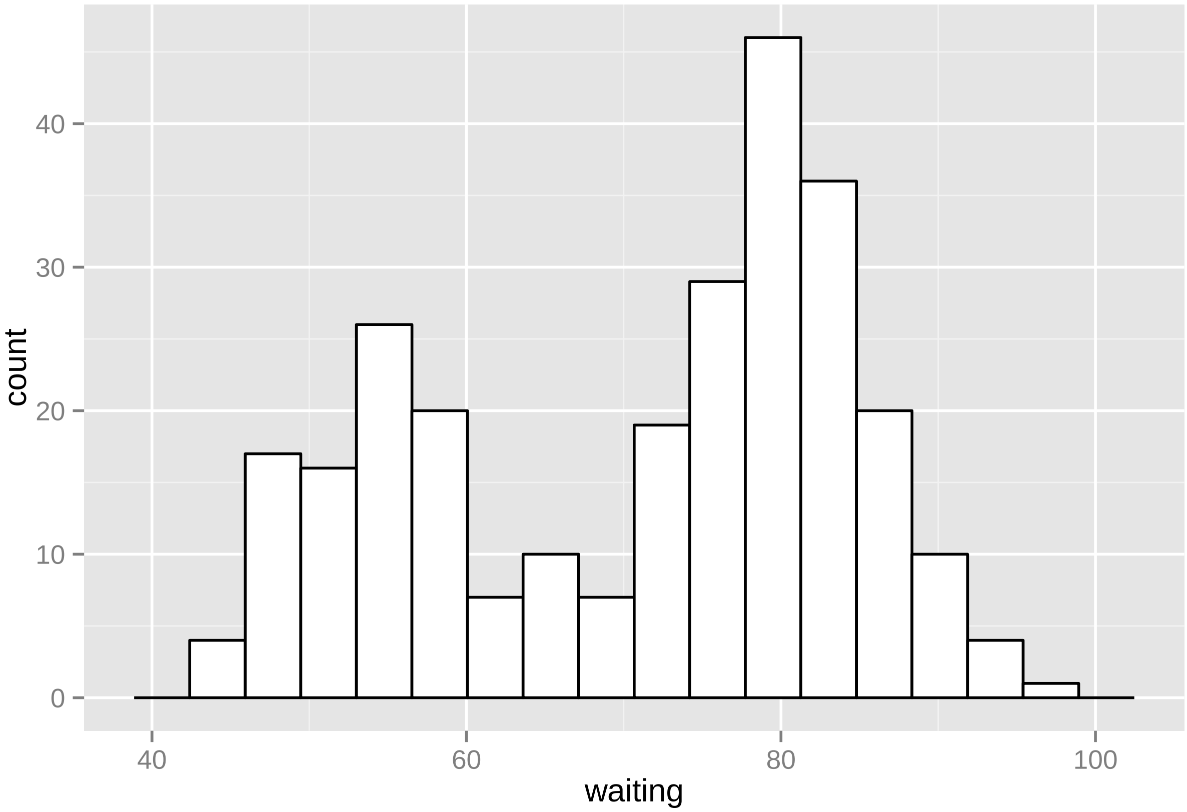
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



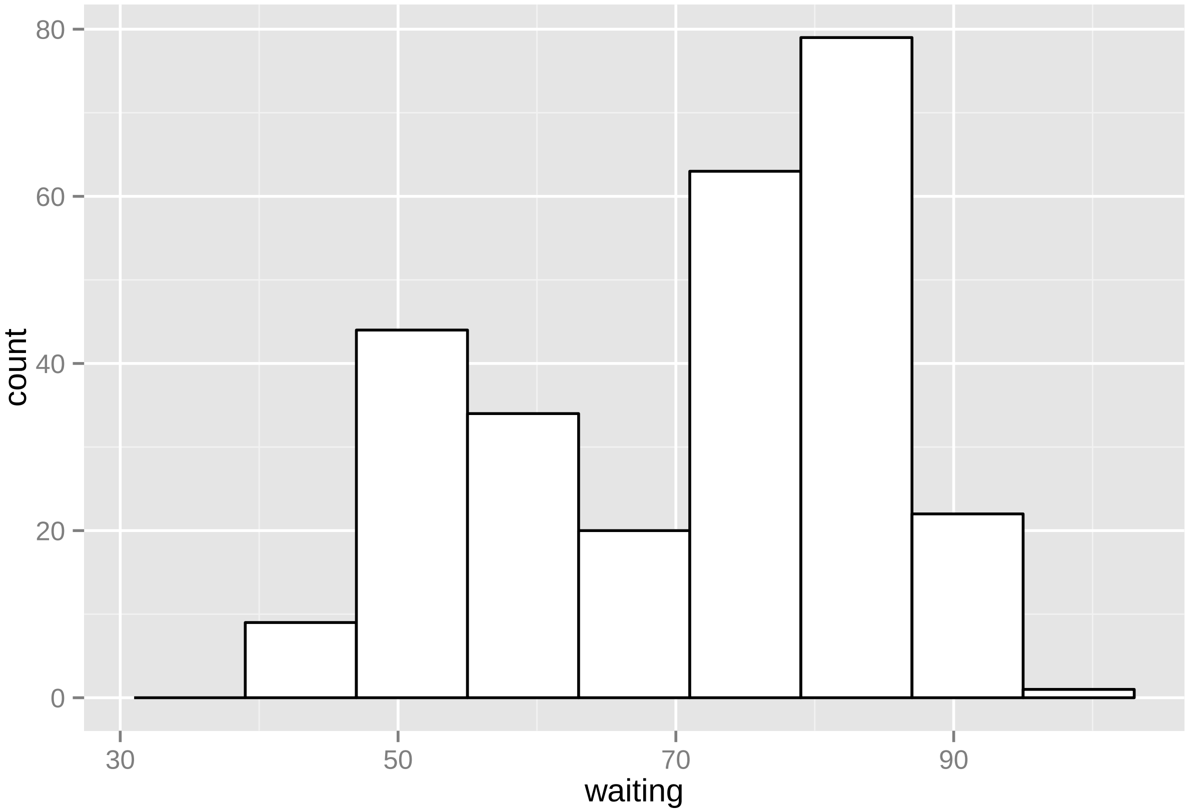
```
# You can change the size of the bins by using binwidth,
# Set the width of each bin to 5
ggplot( faithful, aes( x = waiting)) +
  geom_histogram( binwidth = 5, fill ="white", colour ="black")
```



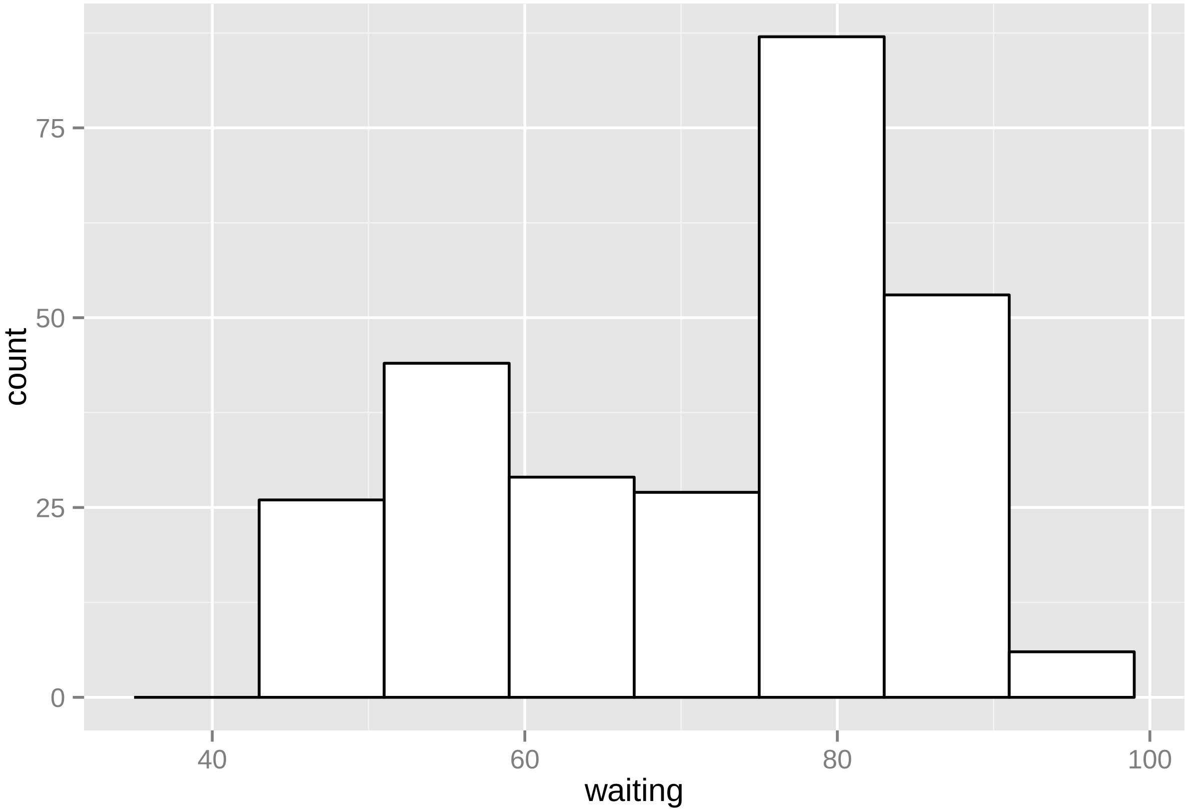
```
# Divide the x range into 15 bins
binsize <- diff( range( faithful$waiting))/ 15
ggplot( faithful, aes( x = waiting)) + geom_histogram( binwidth = binsize, fill ="white", colour ="blac
```



```
h <- ggplot( faithful, aes( x = waiting))
# Save the base object for reuse
h + geom_histogram( binwidth = 8, fill ="white", colour ="black", origin = 31)
```



```
h + geom_histogram( binwidth = 8, fill ="white", colour ="black", origin = 35)
```

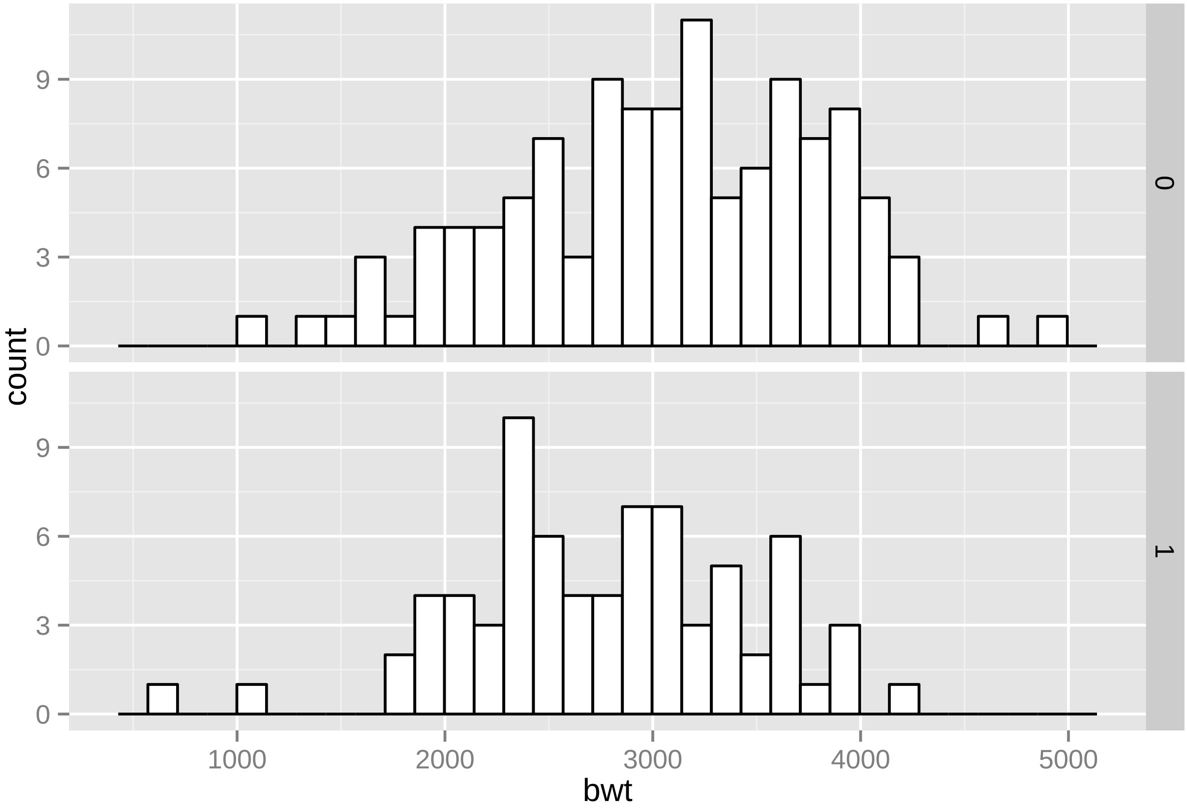


2. Making Multiple Histograms from Grouped Data

You want to make histograms of multiple groups of data.

```
library( MASS) # For the data set # Use smoke as the faceting variable
ggplot( birthwt, aes( x = bwt)) +
  geom_histogram( fill ="white", colour ="black") + facet_grid( smoke ~ .)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

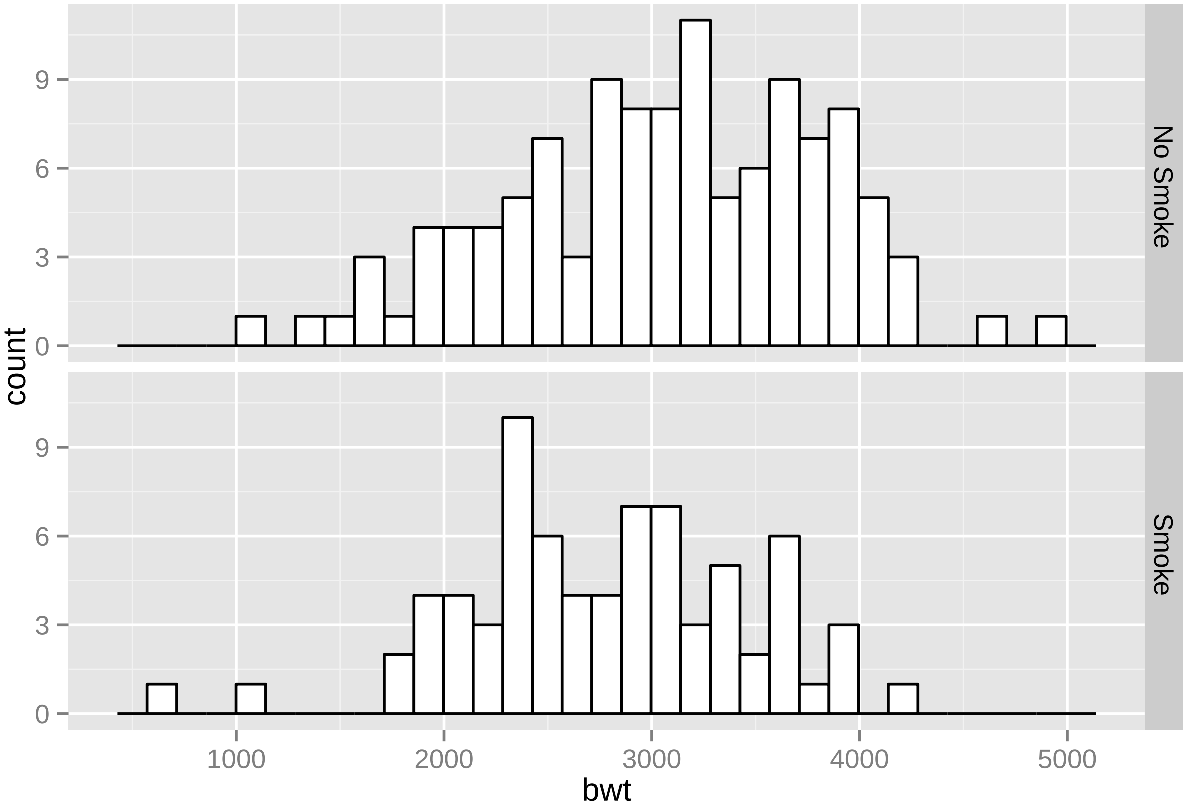


```
## One problem with the faceted graph is that the facet labels are just 0 and 1, and there's no label i
birthwt1 <- birthwt # Make a copy of the data # Convert smoke to a factor
birthwt1$smoke <- factor( birthwt1$smoke)
levels( birthwt1$smoke)
```

```
## [1] "0" "1"
```

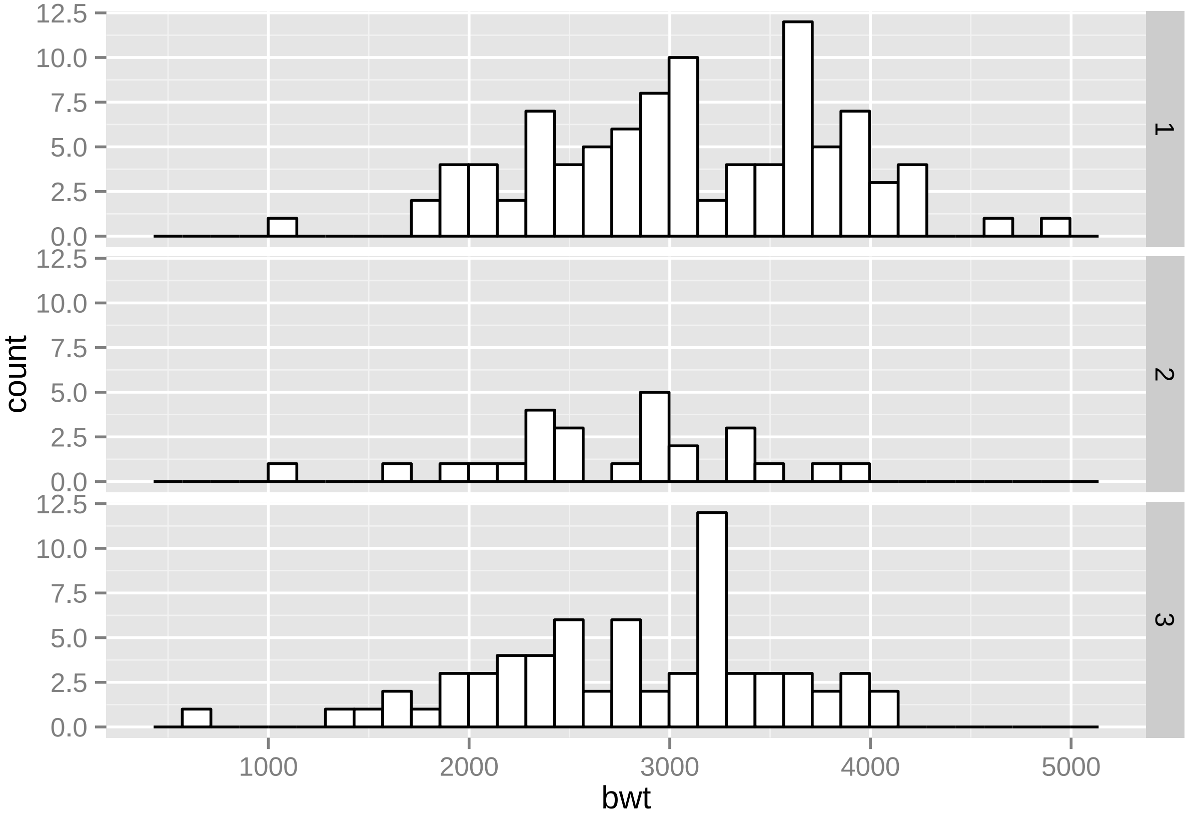
```
birthwt1$smoke <- revalue( birthwt1$smoke, c("0" =" No Smoke", "1" =" Smoke"))

ggplot( birthwt1, aes( x = bwt)) +
  geom_histogram( fill ="white", colour ="black") +
  facet_grid( smoke ~ .)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



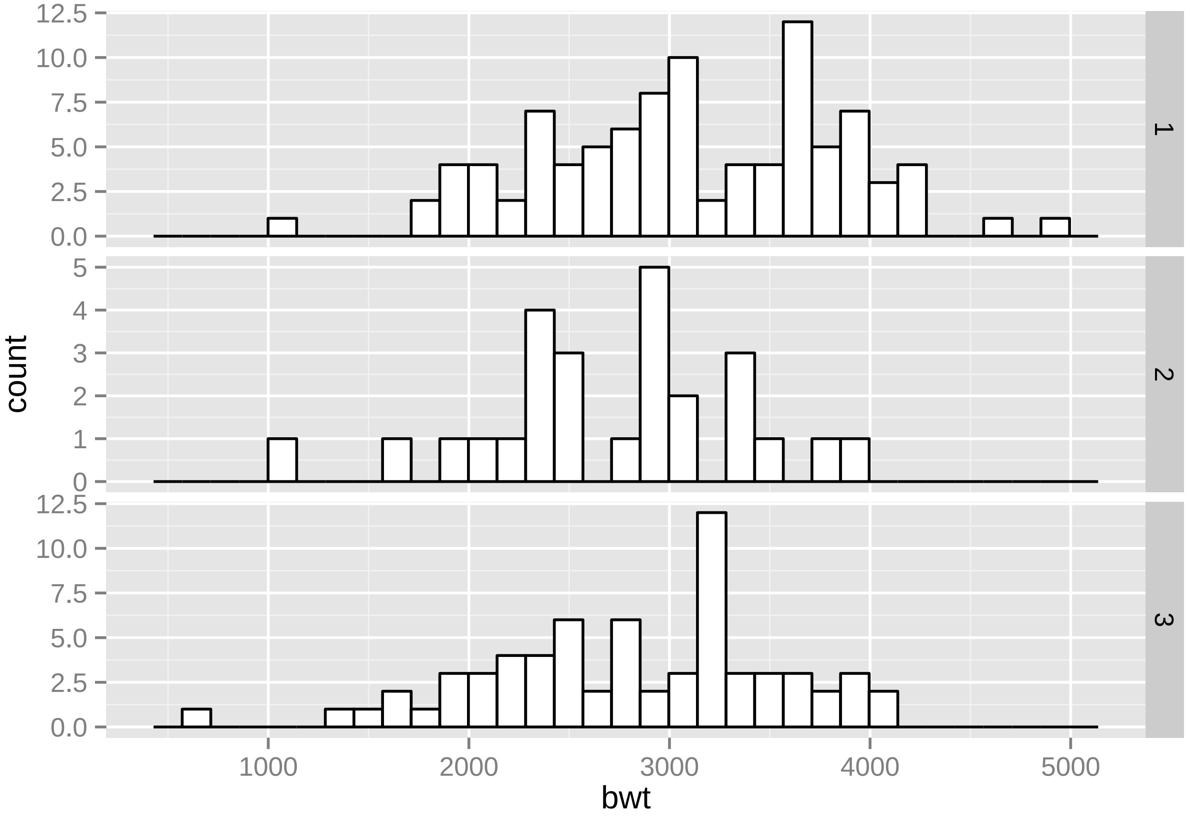
```
ggplot( birthwt, aes( x = bwt)) +
  geom_histogram( fill ="white", colour ="black") +
  facet_grid( race ~ .)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



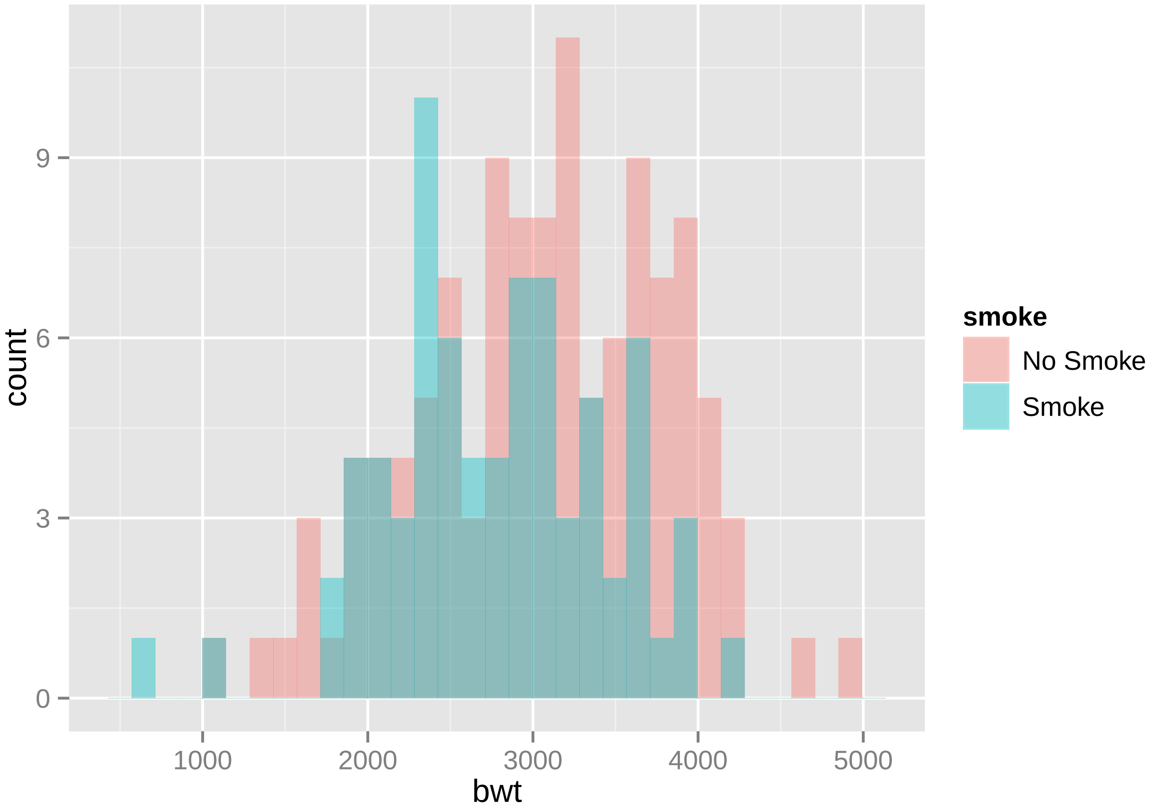
```
# To allow the y scales to be resized independently use scales =" free".
ggplot( birthwt, aes( x = bwt)) +
  geom_histogram( fill ="white", colour ="black") +
  facet_grid( race ~ ., scales ="free")
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



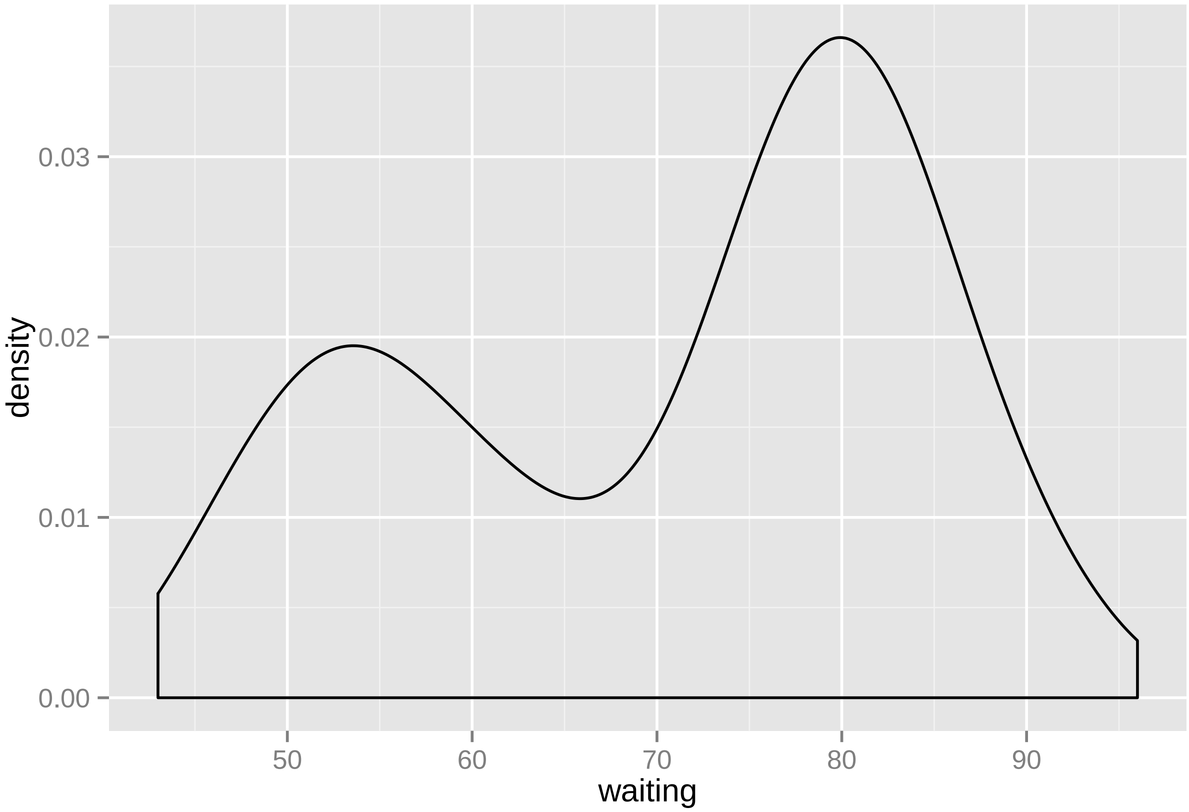
```
# Convert smoke to a factor
birthwt1$smoke <- factor( birthwt1$smoke)

# Map smoke to fill, make the bars NOT stacked, and make them semitransparent
ggplot( birthwt1, aes( x = bwt, fill = smoke)) +
  geom_histogram( position ="identity", alpha = 0.4)
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```

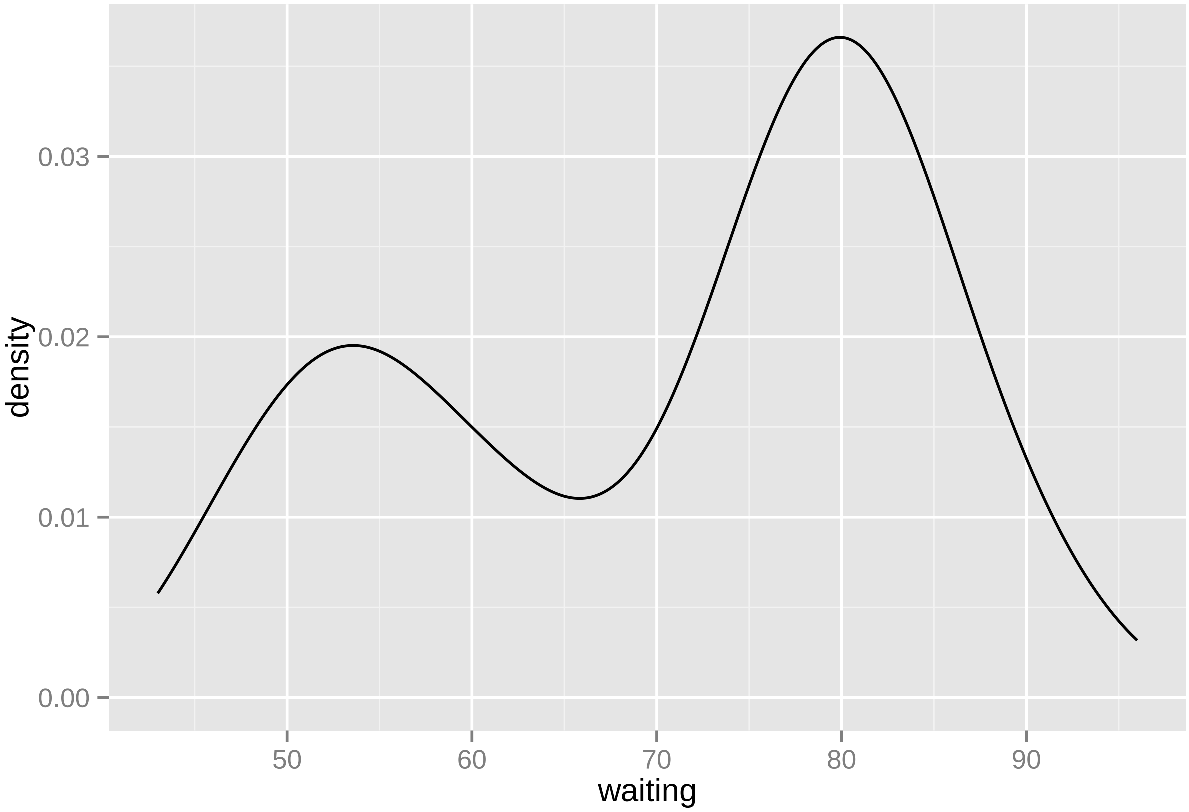


3. Making a Density Curve

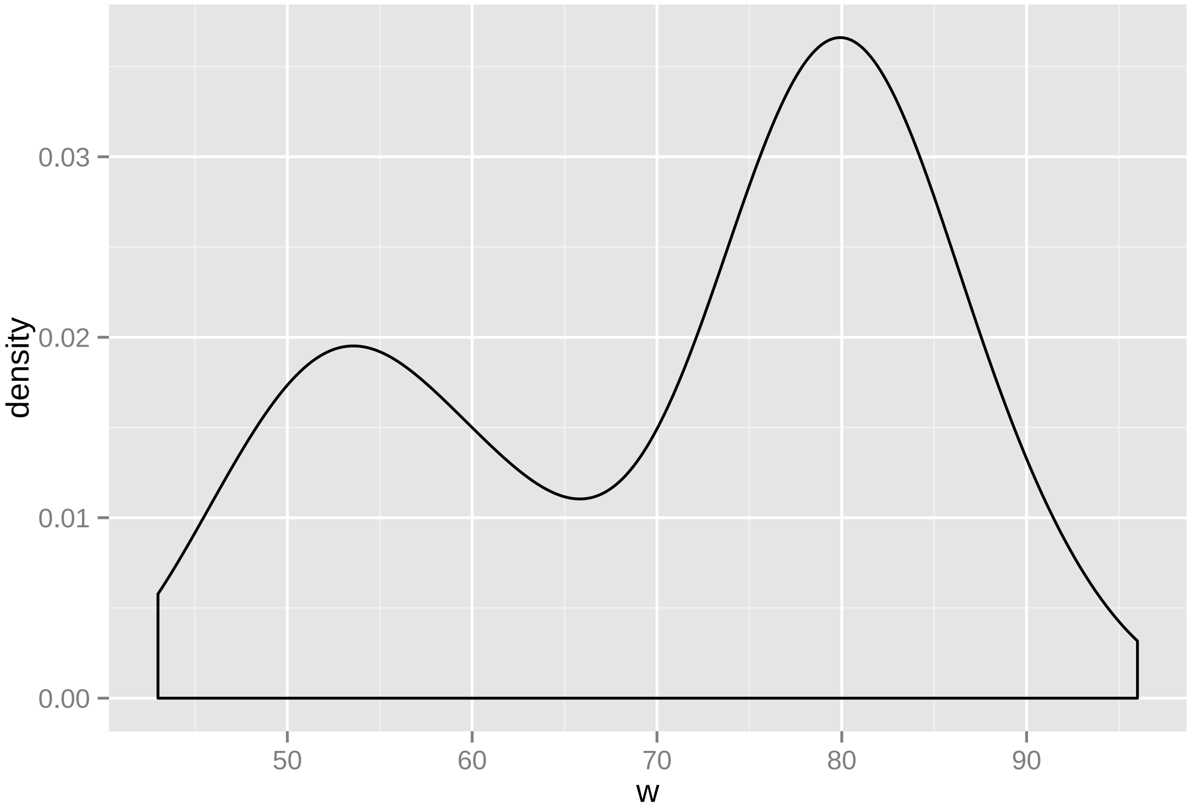
```
# Use geom_density() and map a continuous variable to x
ggplot( faithful, aes( x = waiting)) +
  geom_density()
```



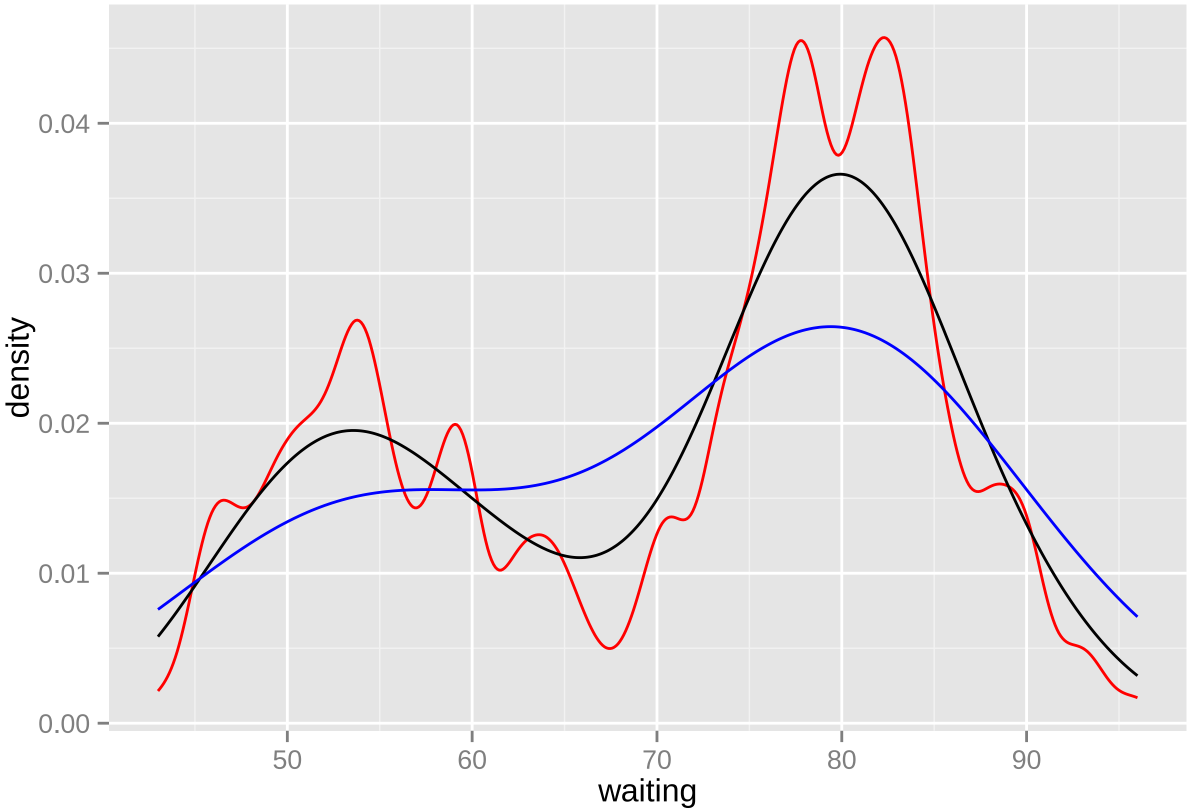
```
# The expand_limits() increases the y range to include the value 0
ggplot( faithful, aes( x = waiting)) +
  geom_line( stat ="density") +
  expand_limits( y = 0)
```



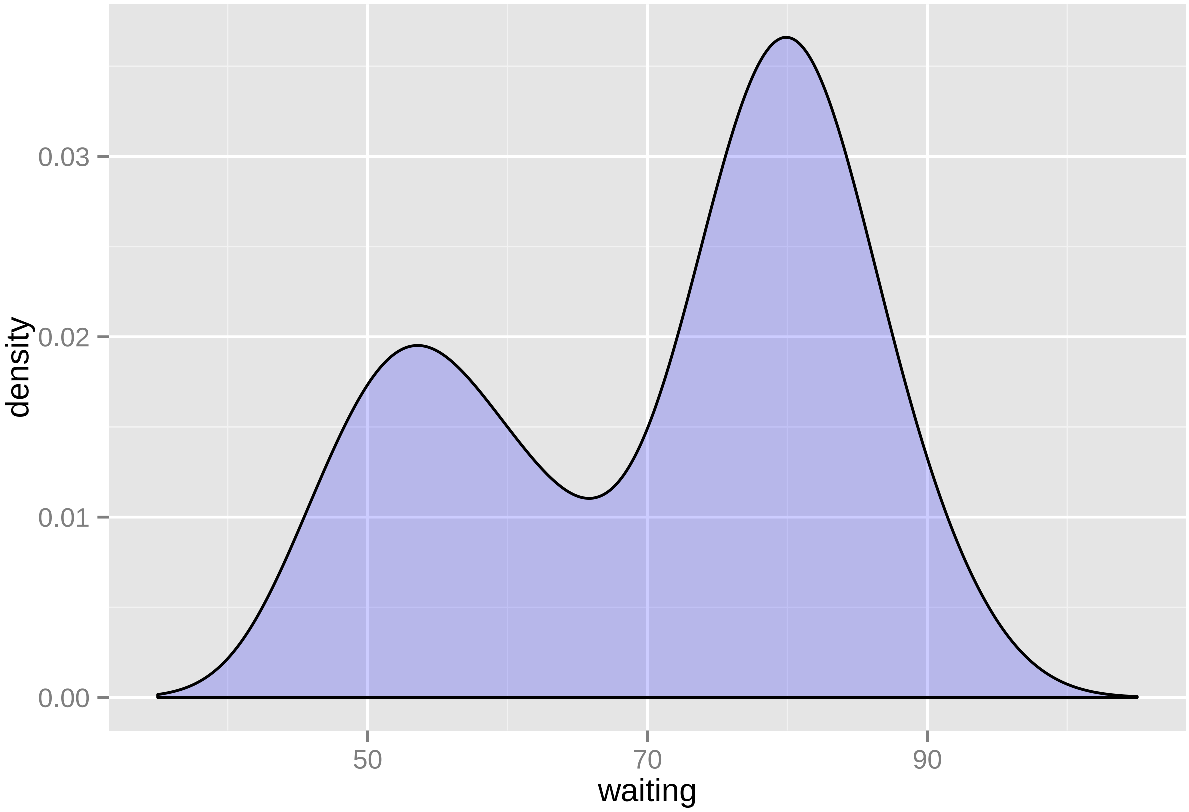
```
# Store the values in a simple vector
w <- faithful$waiting
ggplot( NULL, aes( x = w)) +
  geom_density()
```



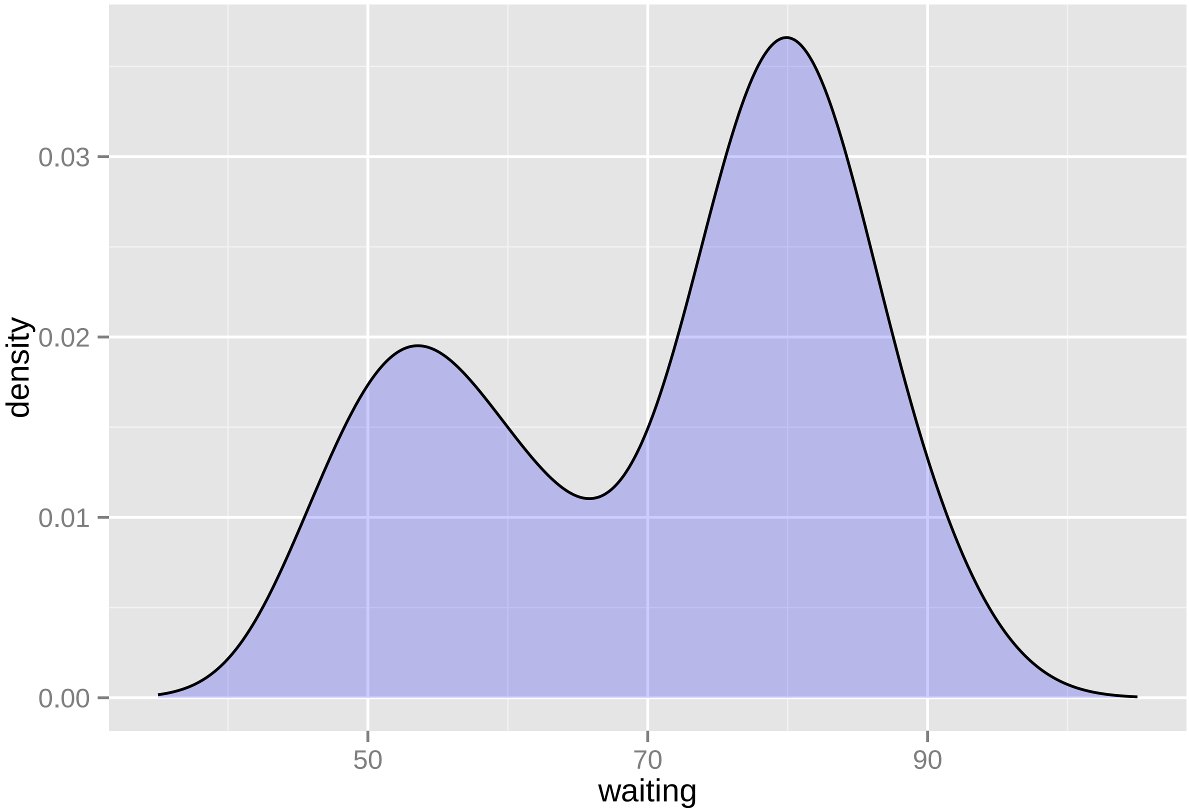
```
# kernel density curve is an estimate of the population distribution, based on the sample data. The amo
ggplot( faithful, aes( x = waiting)) +
  geom_line( stat ="density", adjust =.25, colour ="red") +
  geom_line( stat ="density") +
  geom_line( stat ="density", adjust = 2, colour ="blue")
```



```
# area
ggplot( faithful, aes( x = waiting)) +
  geom_density( fill ="blue", alpha =.2) +
  xlim( 35, 105)
```

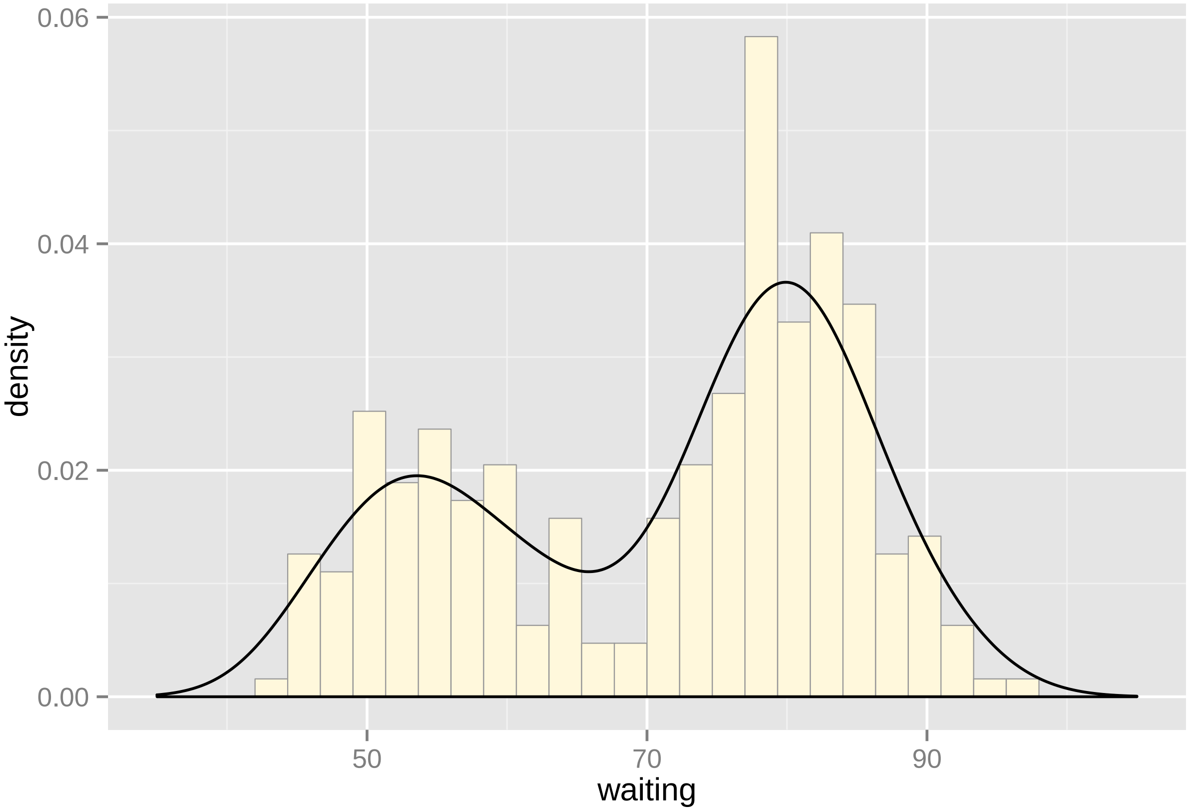


```
# This draws a blue polygon with geom_density(), then adds a line on top
ggplot( faithful, aes( x = waiting)) +
  geom_density( fill ="blue", colour = NA, alpha =.2) +
  geom_line( stat ="density") +
  xlim( 35, 105)
```



```r
# To compare the theoretical and observed distributions, you can overlay the density curve with the his
ggplot( faithful, aes( x = waiting, y = ..density.. )) +
  geom_histogram( fill ="cornsilk", colour ="grey60", size =.2) +
  geom_density() +
  xlim( 35, 105)
```

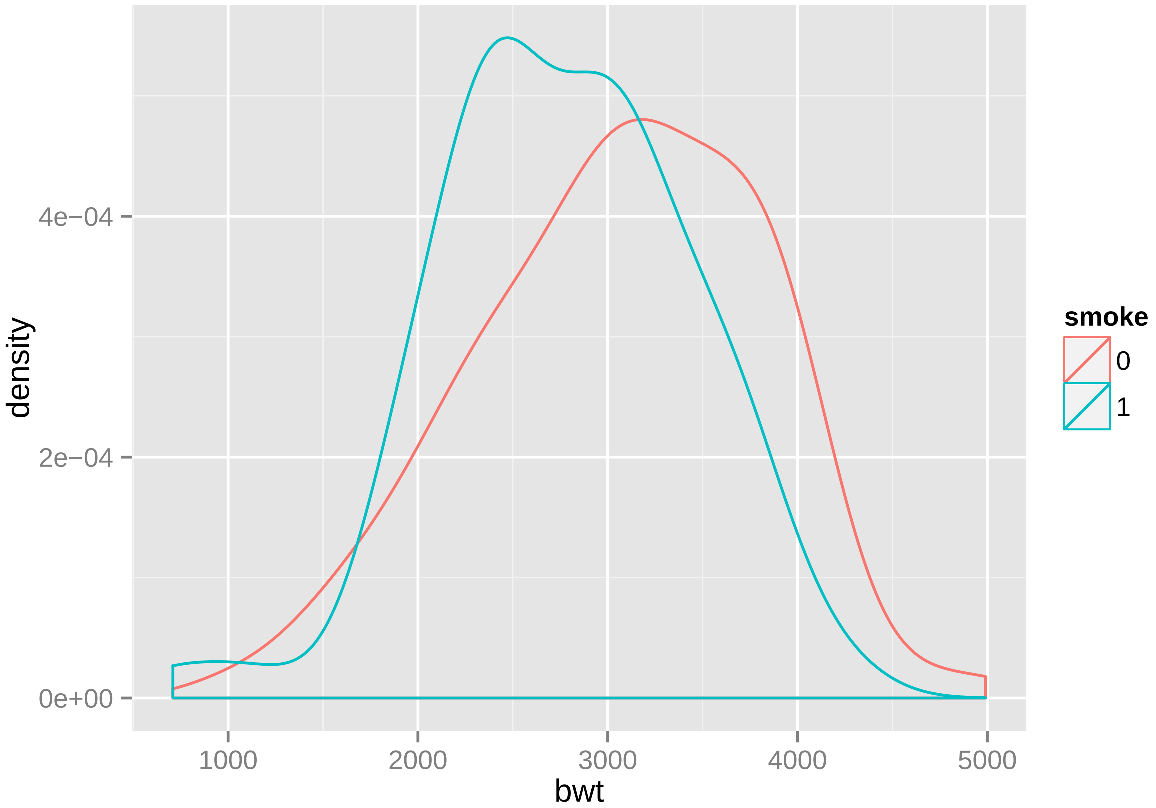
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



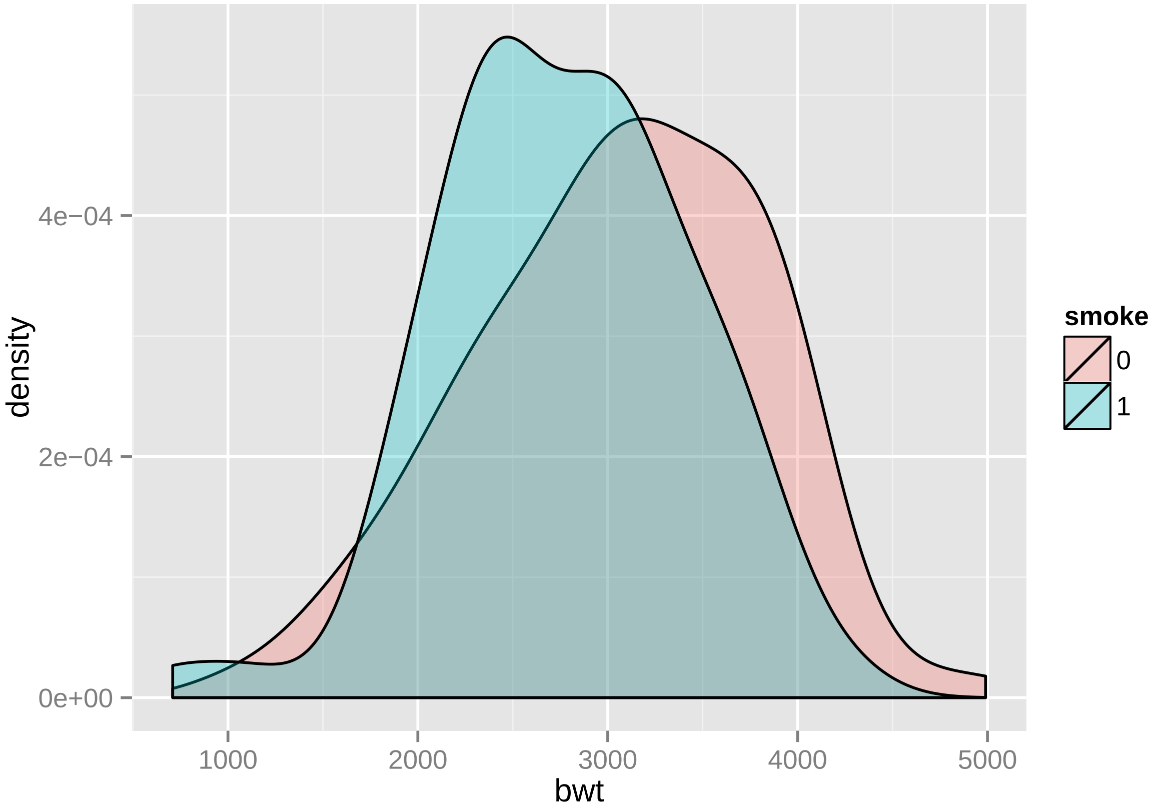
4. Making Multiple Density Curves from Grouped Data

You want to make density curves of multiple groups of data.

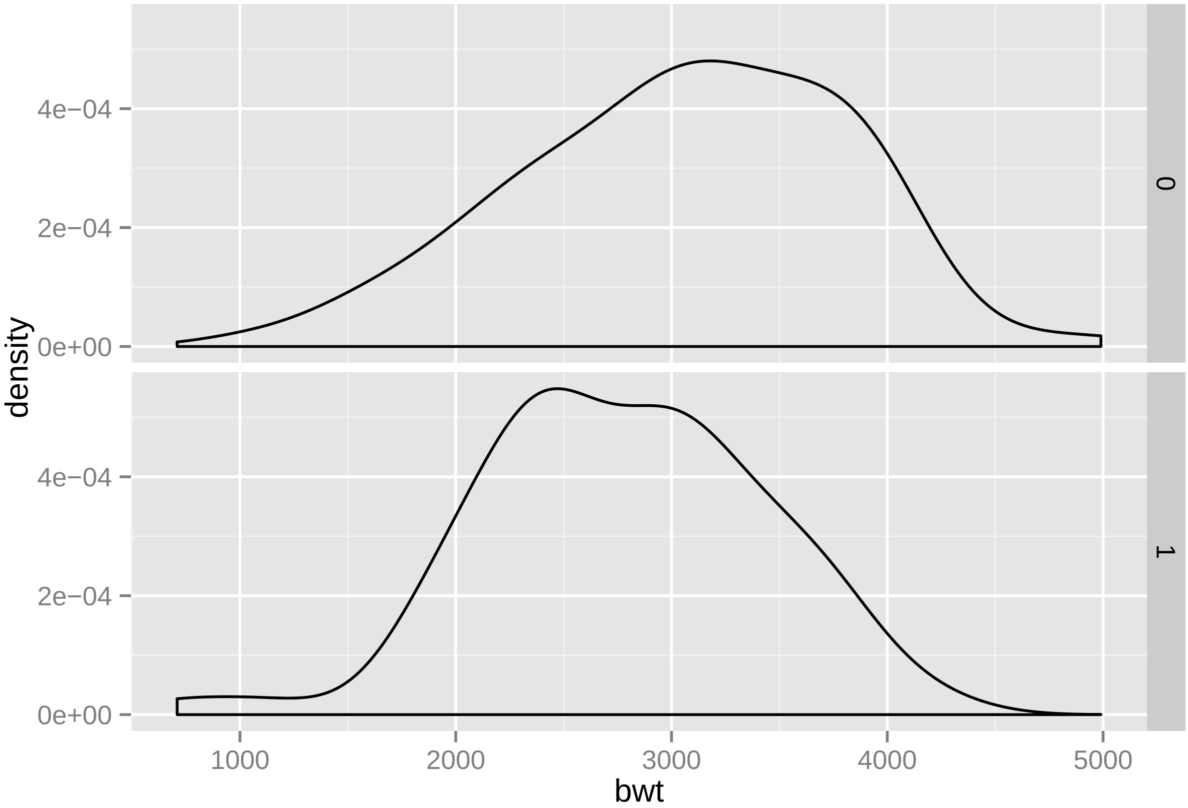
```
library( MASS) # For the data set

# Make a copy of the data
birthwt1 <- birthwt

# Convert smoke to a factor
birthwt1$smoke <- factor( birthwt1$smoke)

# Map smoke to colour
ggplot( birthwt1, aes( x = bwt, colour = smoke)) +
  geom_density()
```



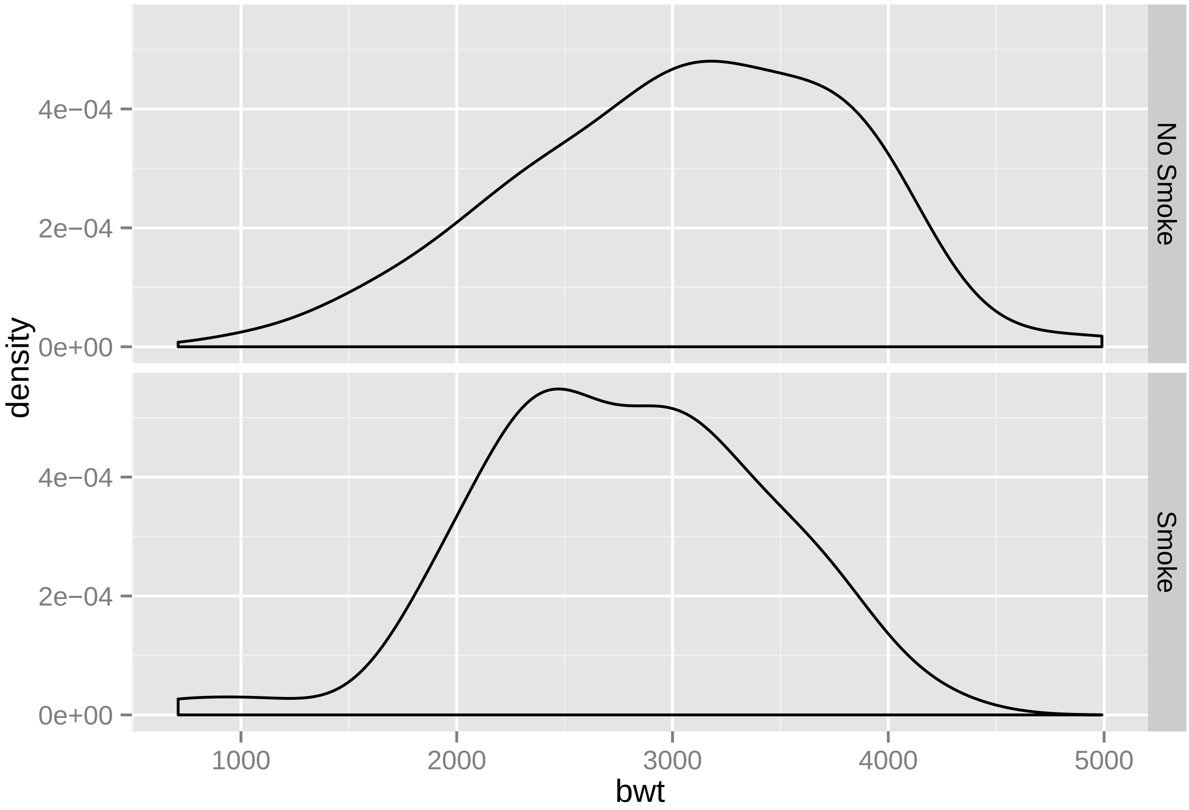
```
# Map smoke to fill and make the fill semitransparent by setting alpha
ggplot( birthwt1, aes( x = bwt, fill = smoke)) + geom_density( alpha =.3)
```



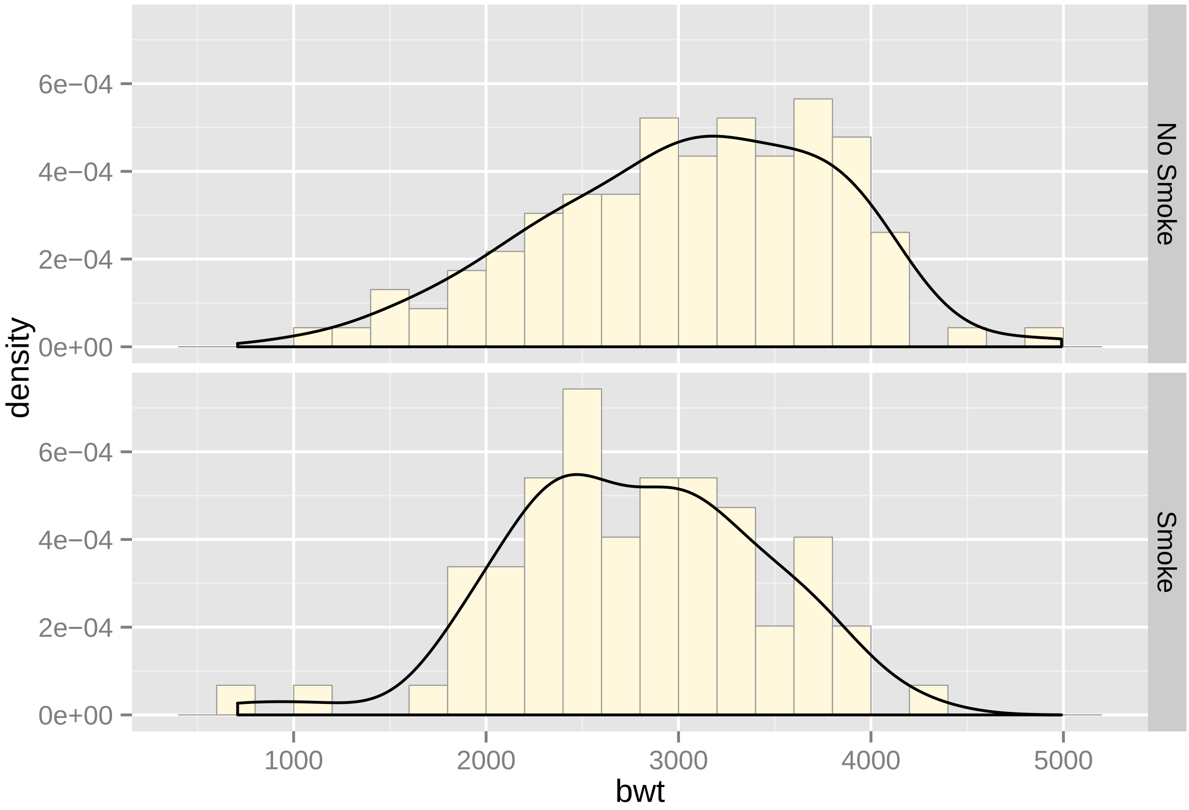
```
# Another method for visualizing the distributions is to use facets,
ggplot( birthwt1, aes( x = bwt)) +
  geom_density() +
  facet_grid( smoke ~ .)
```



```
# to rename 0/1 into Smoke/No Smoke
birthwt1$smoke <- revalue( birthwt1$smoke, c("0" ="No Smoke", "1" ="Smoke"))
ggplot( birthwt1, aes( x = bwt)) +
  geom_density() +
  facet_grid( smoke ~ .)
```



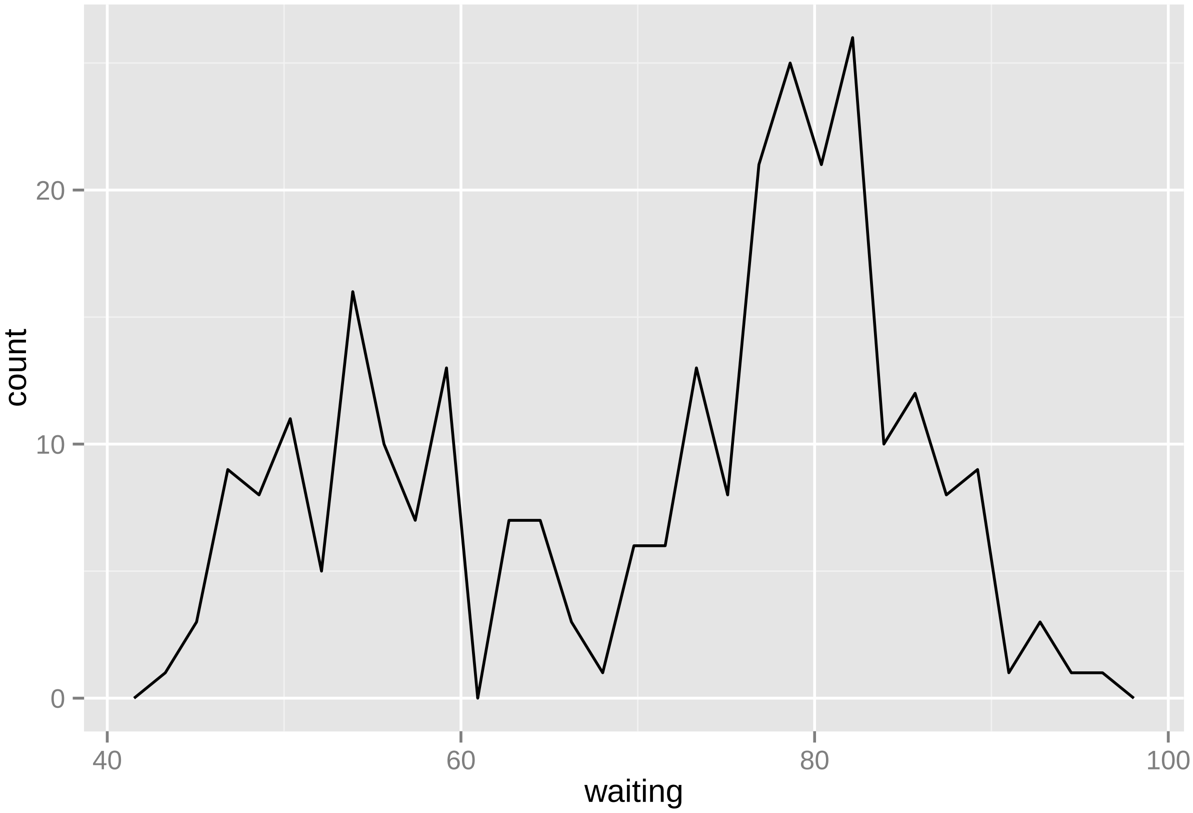
```
# If you want to see the histograms along with the density curves,
ggplot( birthwt1, aes( x = bwt, y =..density..)) +
  geom_histogram( binwidth = 200, fill ="cornsilk", colour ="grey60", size =.2) +
  geom_density() +
  facet_grid( smoke ~ .)
```



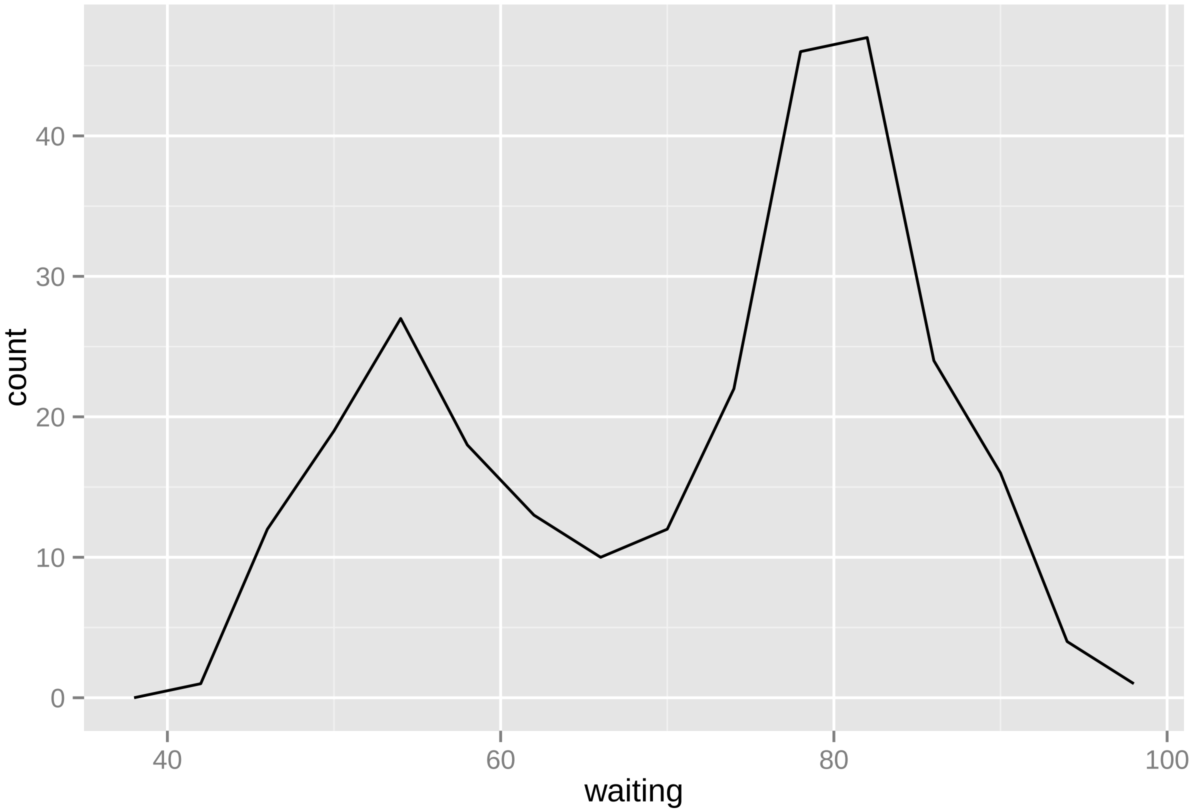
5. Making a Frequency Polygon

```
ggplot( faithful, aes( x = waiting)) +
  geom_freqpoly()
```

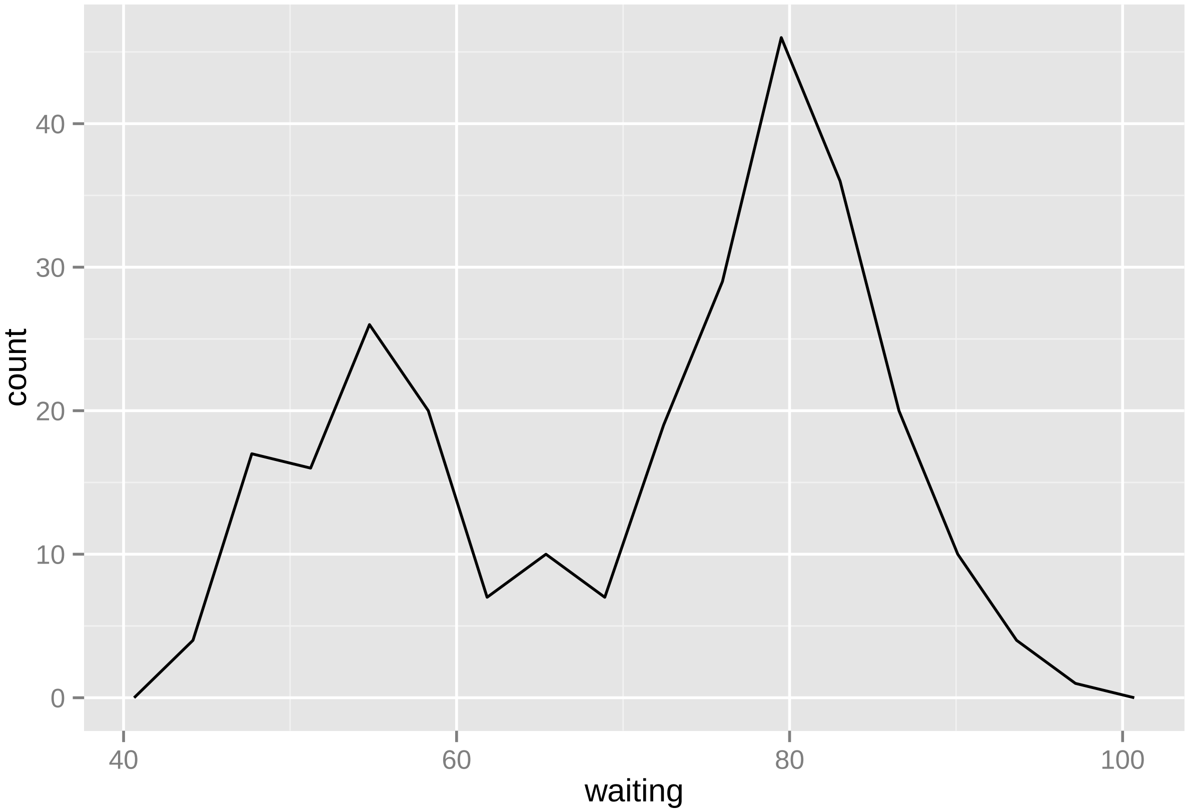
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



```
# you can control the bin width for the frequency polygon
ggplot( faithful, aes( x = waiting)) +
  geom_freqpoly( binwidth = 4)
```

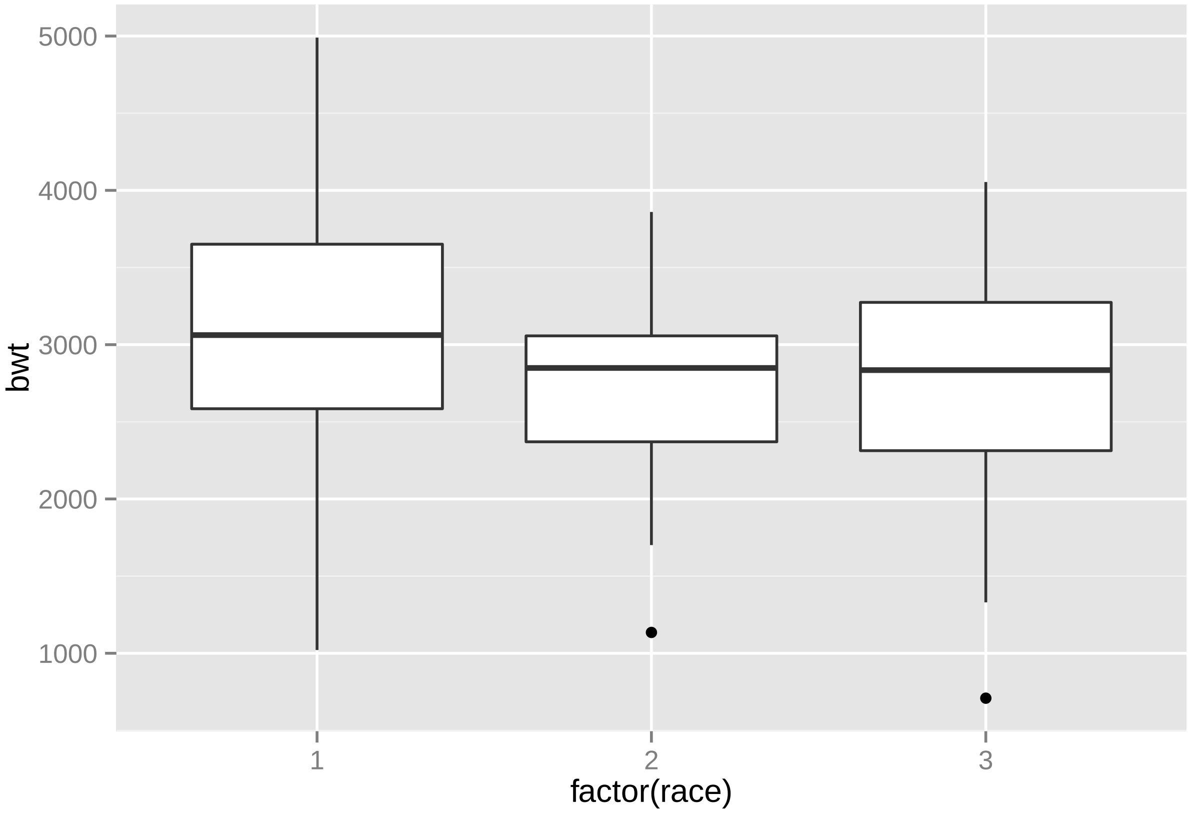


```
# Or, instead of setting the width of each bin directly,
# Use 15 bins
binsize <- diff( range( faithful$waiting))/ 15
ggplot( faithful, aes( x = waiting)) + geom_freqpoly( binwidth = binsize)
```

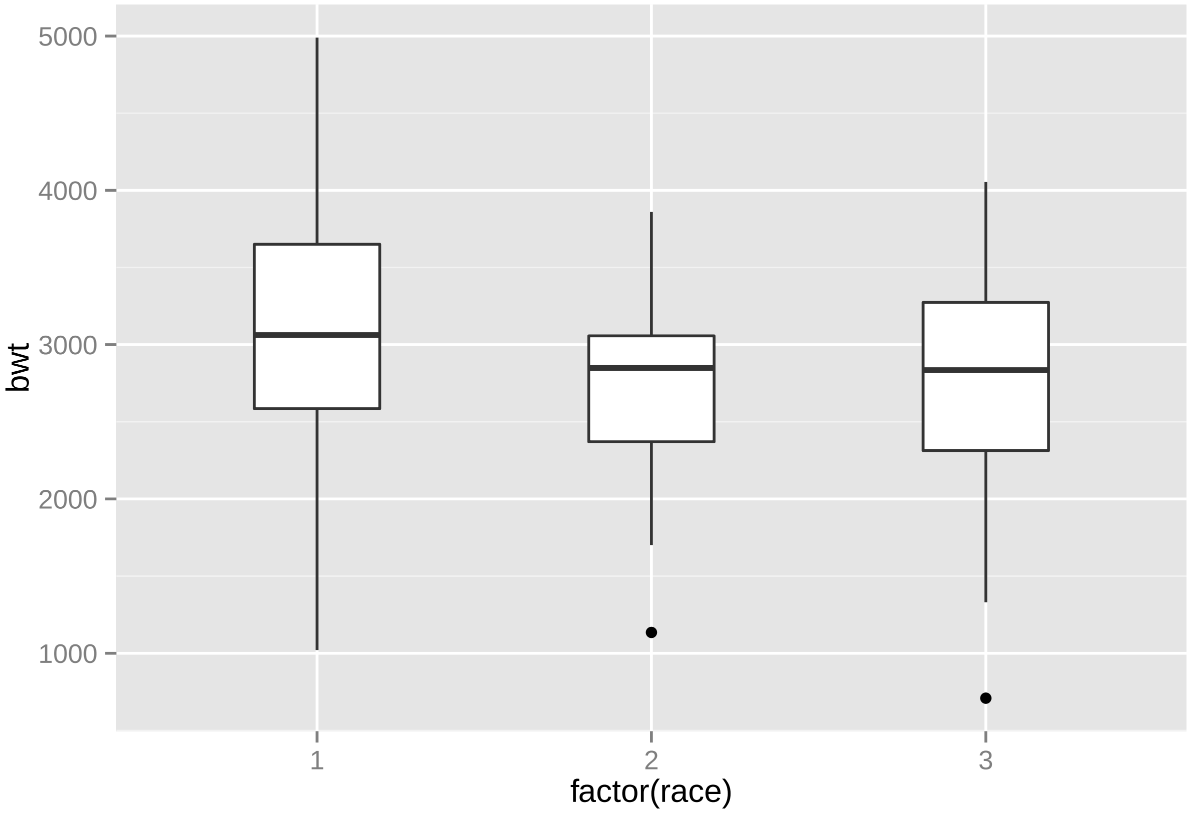


6. Making a Basic Box Plot

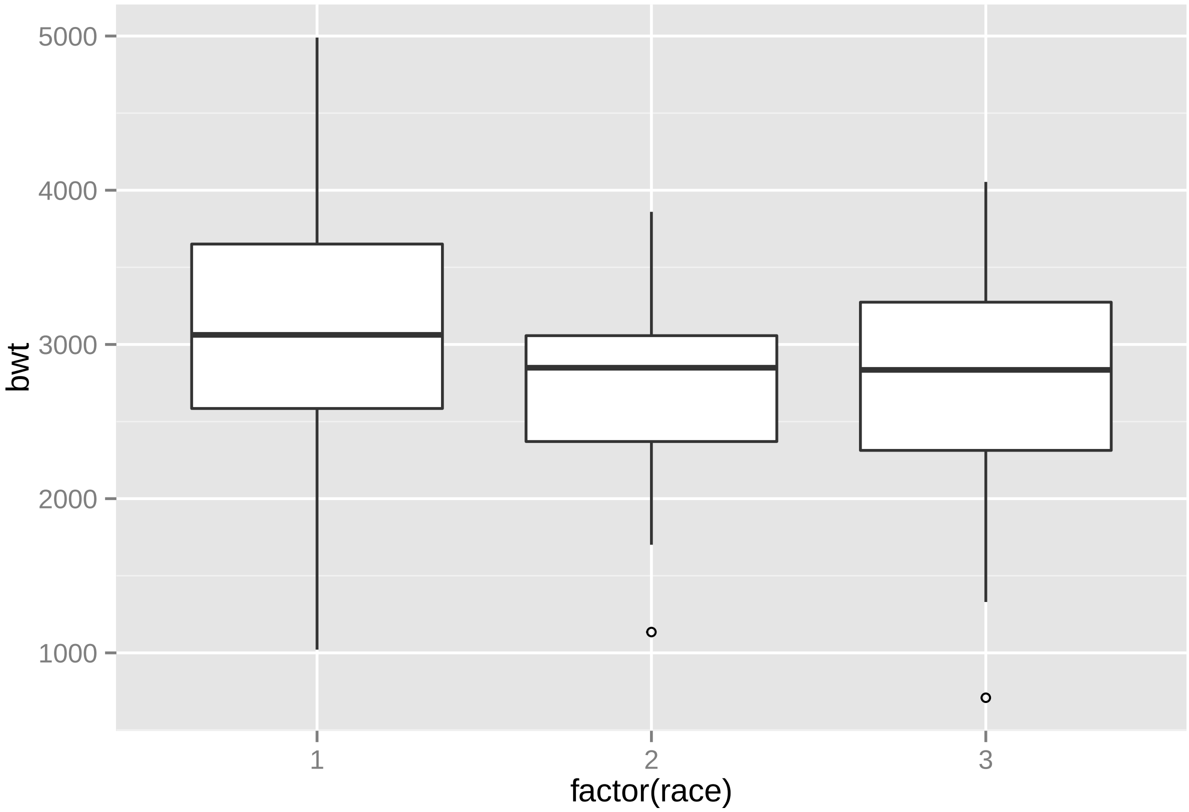
```
library( MASS) # For the data set
ggplot( birthwt, aes( x = factor( race), y = bwt)) +
  geom_boxplot()
```



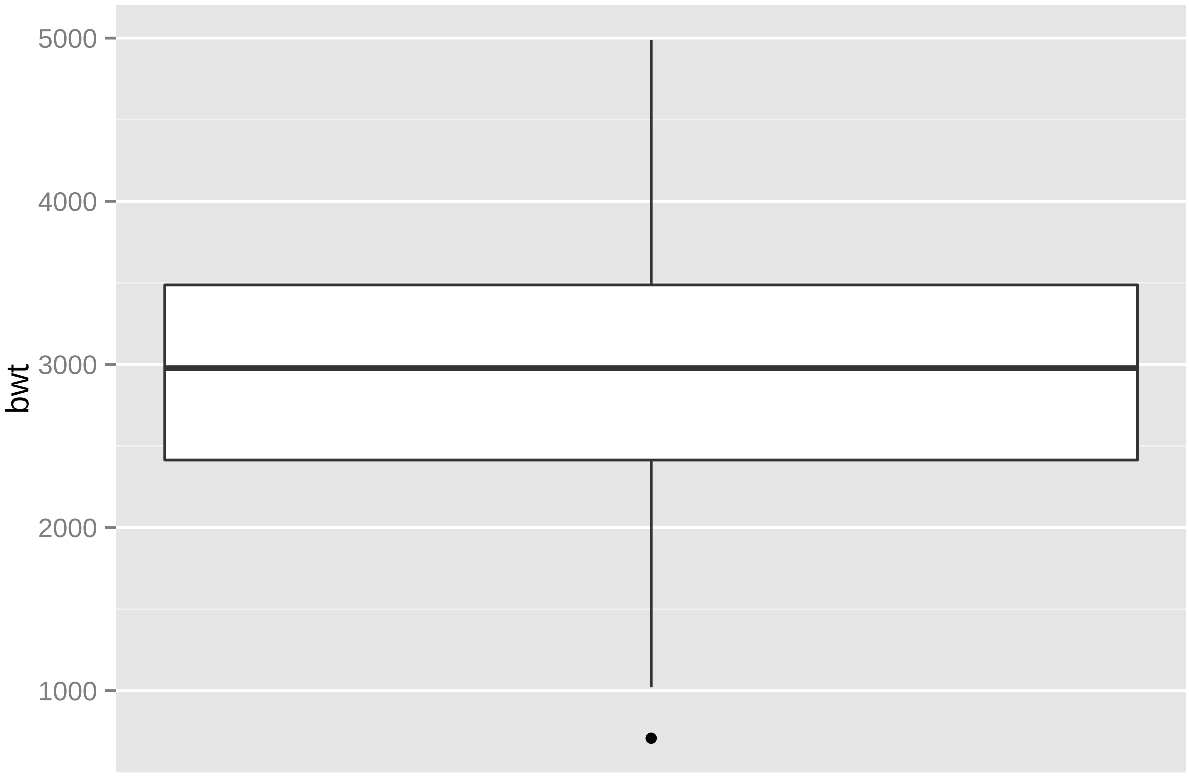
```
# To change the width of the boxes, you can set width
ggplot( birthwt, aes( x = factor(race), y = bwt)) +
  geom_boxplot( width =.5)
```



```
# If there are many outliers and there is overplotting, you can change the size and shape of the outlie
ggplot( birthwt, aes( x = factor( race), y = bwt)) +
  geom_boxplot( outlier.size = 1.5, outlier.shape = 21)
```



```
# To make a box plot of just a single group,
ggplot( birthwt, aes( x = 1, y = bwt)) +
  geom_boxplot() +
  scale_x_continuous( breaks = NULL) +
  theme( axis.title.x = element_blank())
```

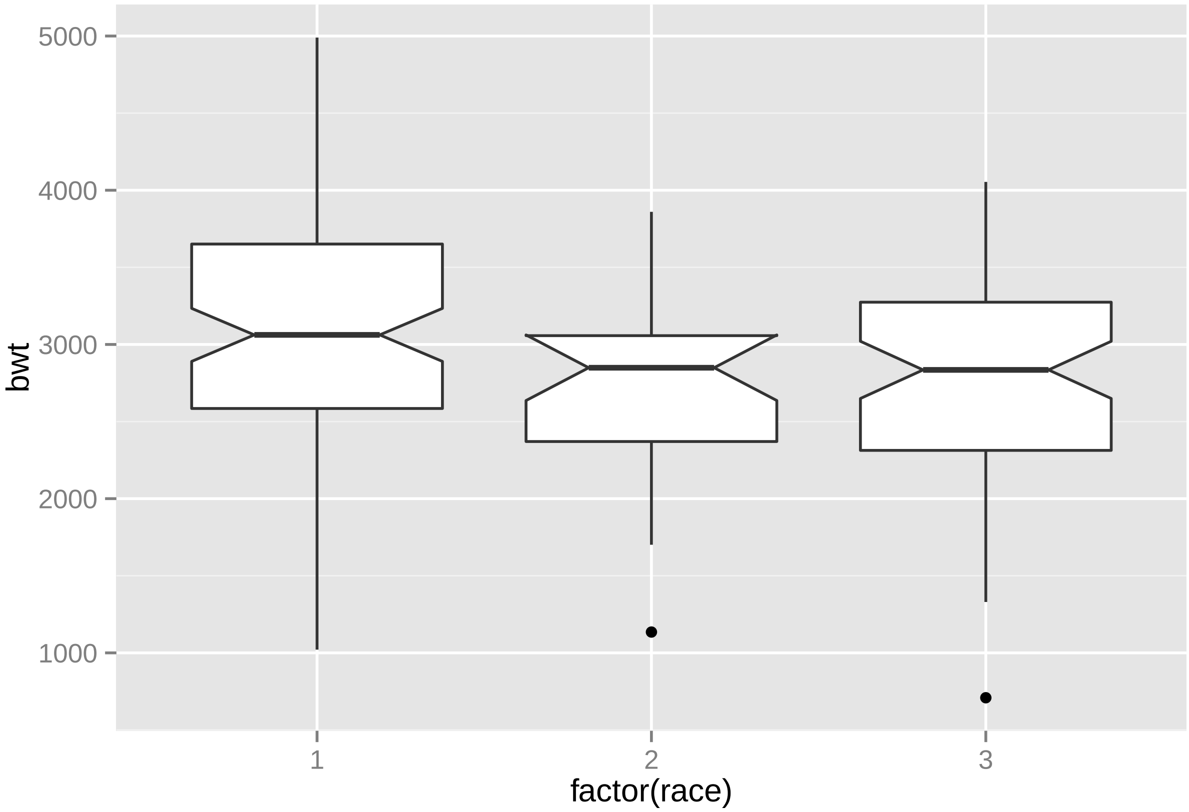


7. Adding Notches to a Box Plot

You want to add notches to a box plot to assess whether the medians are different.

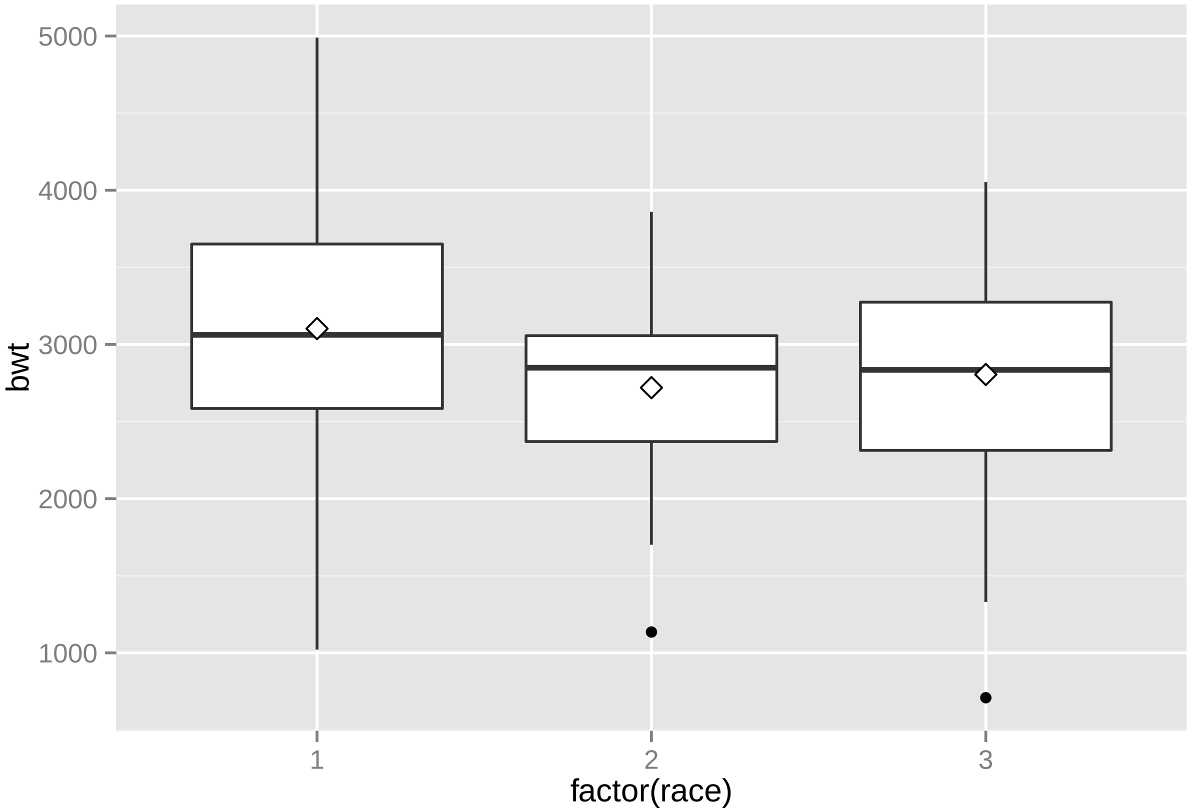
```
library( MASS) # For the data set

ggplot( birthwt, aes( x = factor( race), y = bwt)) +
  geom_boxplot( notch = TRUE)
```

```
## notch went outside hinges. Try setting notch=FALSE.
```



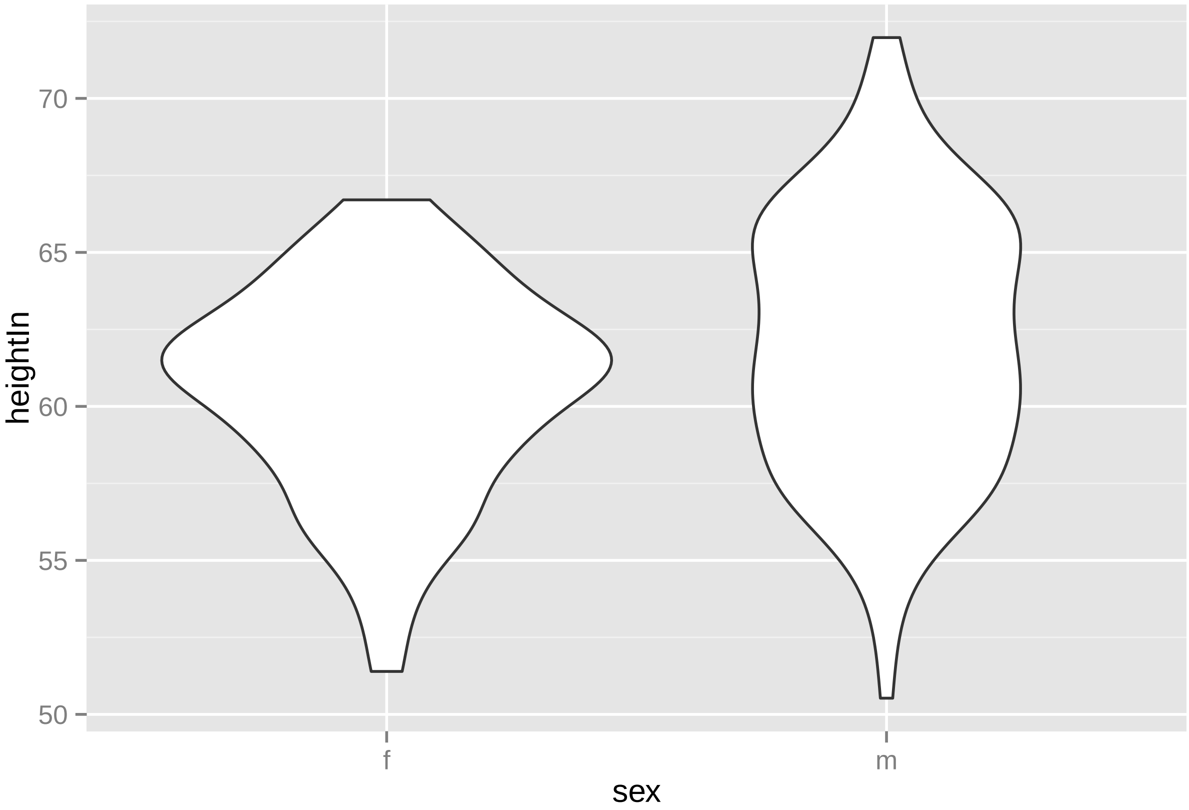
8. Adding Means to a Box Plot

```
library( MASS) # For the data set
ggplot( birthwt, aes( x = factor( race), y = bwt)) +
  geom_boxplot() +
  stat_summary( fun.y ="mean", geom ="point", shape = 23, size = 3, fill ="white")
```

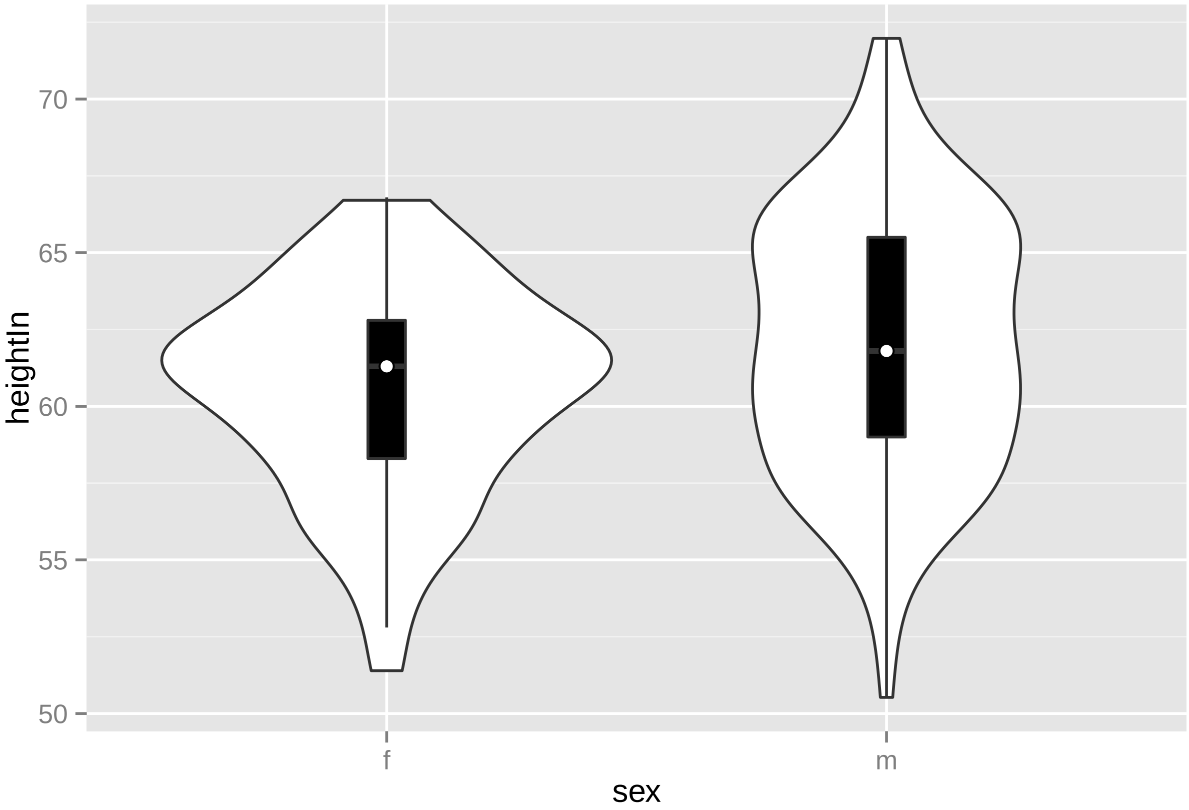


9. Making a Violin Plot

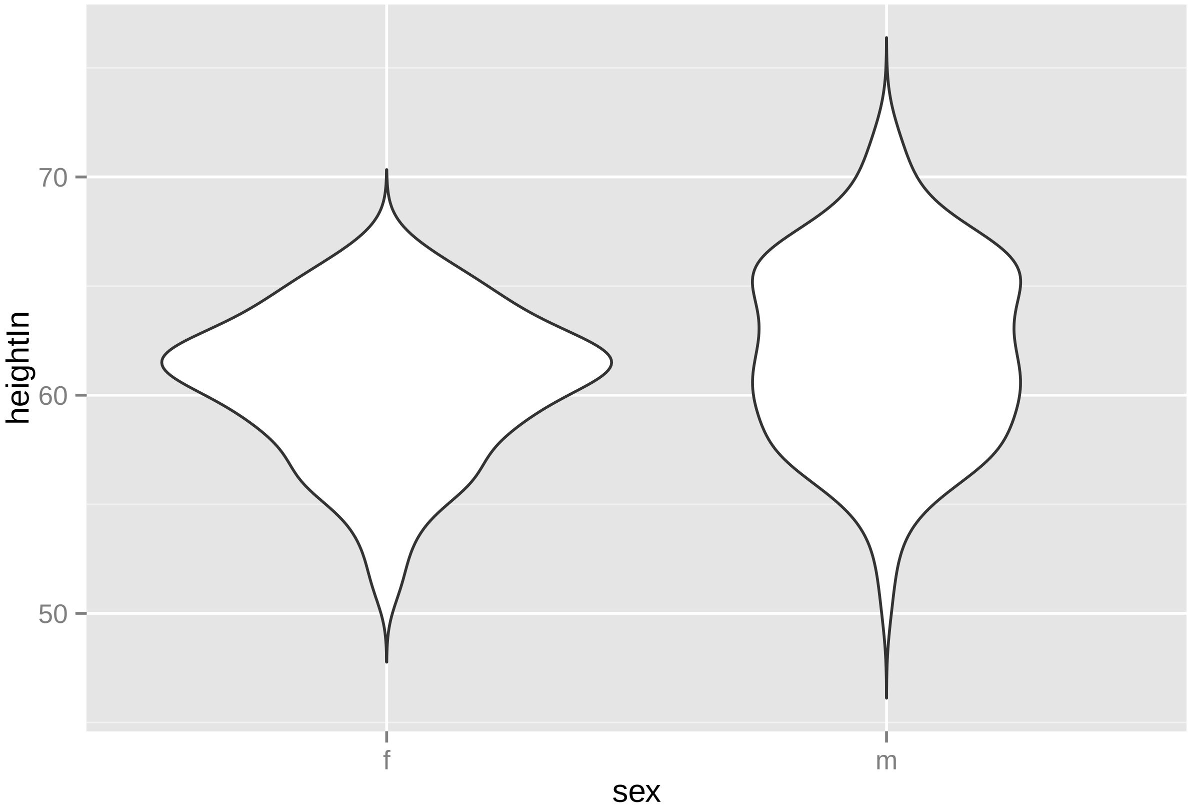
```
# Base plot
p <- ggplot( heightweight, aes( x = sex, y = heightIn))
p + geom_violin()
```



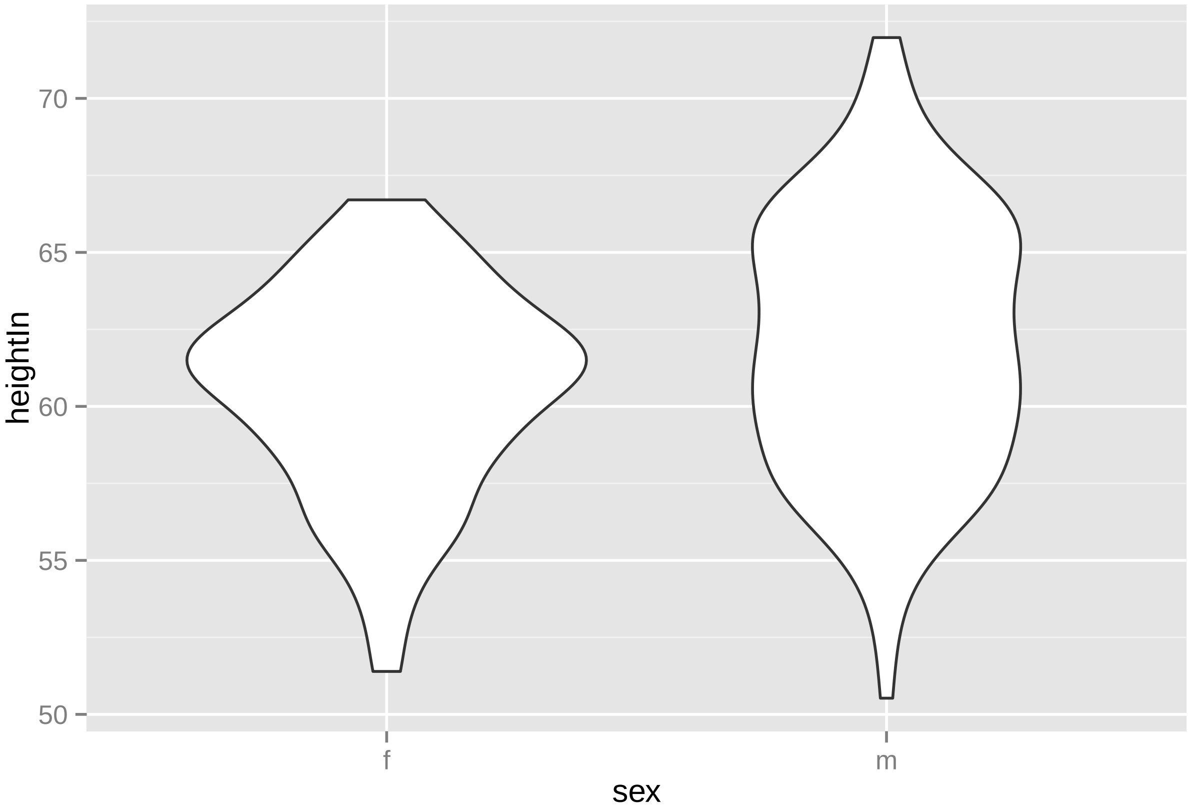
```
# Additionally, the box plot outliers are not displayed, which we do by setting outlier.colour = NA.
p +
  geom_violin() +
  geom_boxplot( width =.1, fill ="black", outlier.colour = NA) +
  stat_summary( fun.y = median, geom ="point", fill ="white", shape = 21, size = 2.5)
```



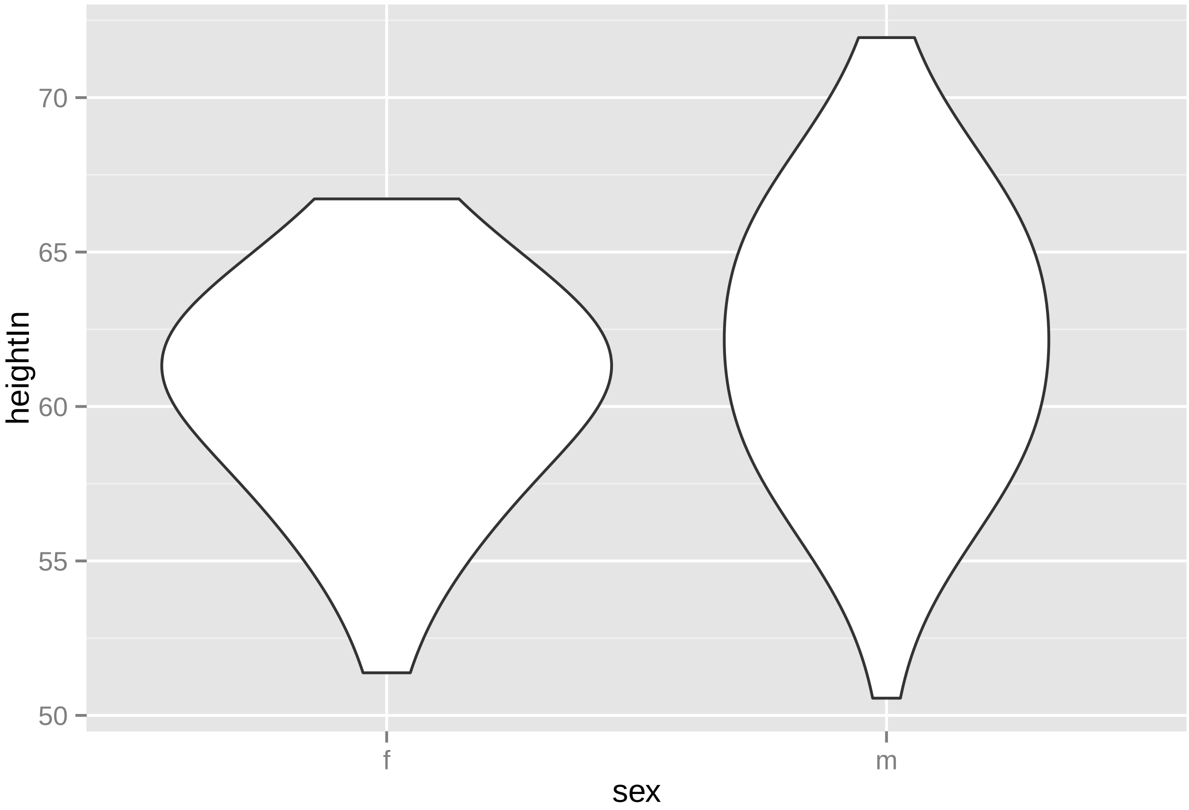
```
# It's possible to keep the tails, by setting trim = FALSE
p +
  geom_violin( trim = FALSE)
```



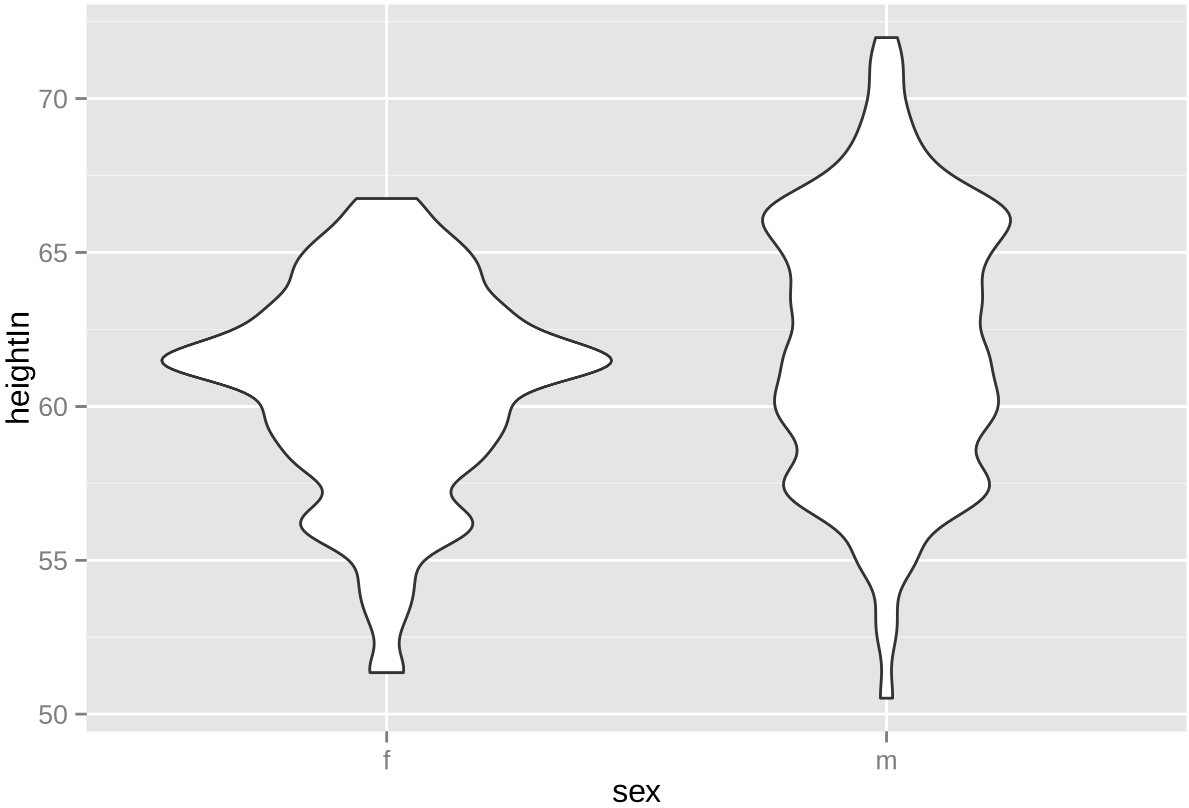
```
# Instead of equal areas, you can use scale =" count" to scale the areas proportionally to the number o
p +
  geom_violin( scale ="count")
```



```
# More smoothing
p +
  geom_violin( adjust = 2)
```



```
# Less smoothing
p +
  geom_violin( adjust =.5)
```

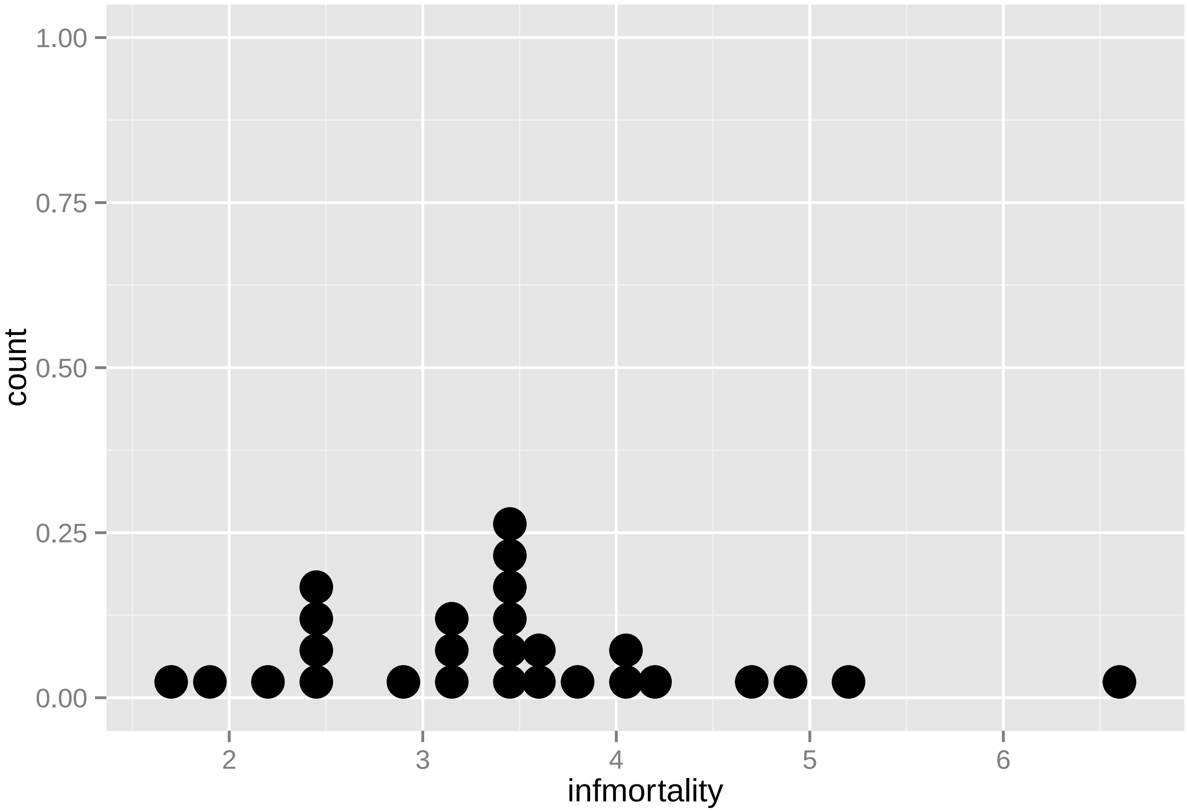


10. Making a Dot Plot

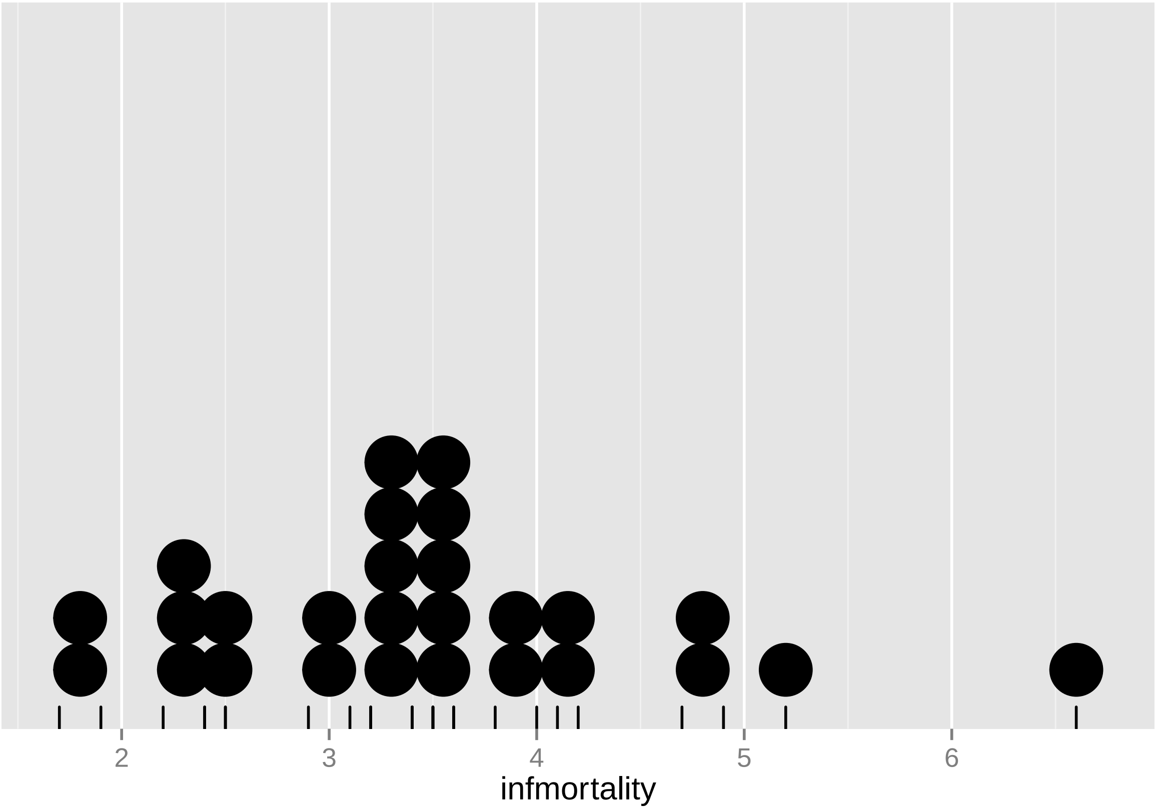
You want to make a Wilkinson dot plot, which shows each data point.

```
countries2009 <- subset( countries, Year == 2009 & healthexp > 2000)
p <- ggplot( countries2009, aes( x = infmortality))

# default
p +
  geom_dotplot()
```

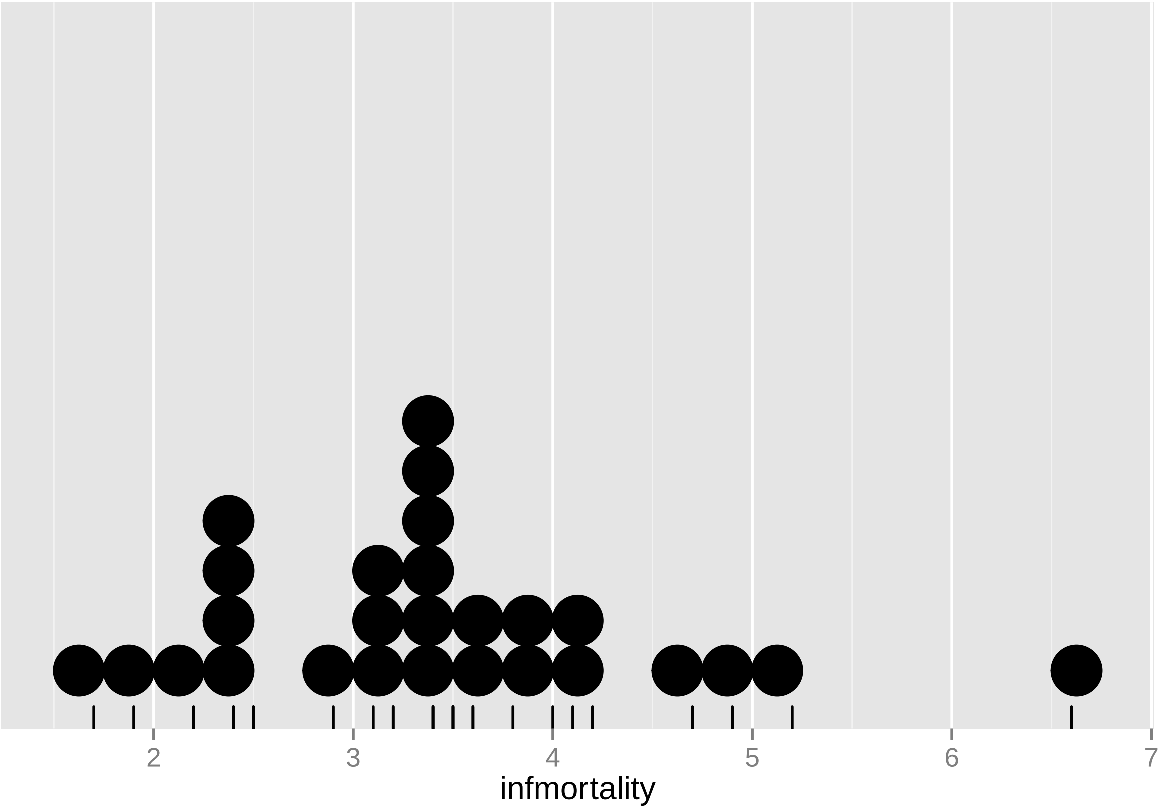
```
## stat_bindot: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



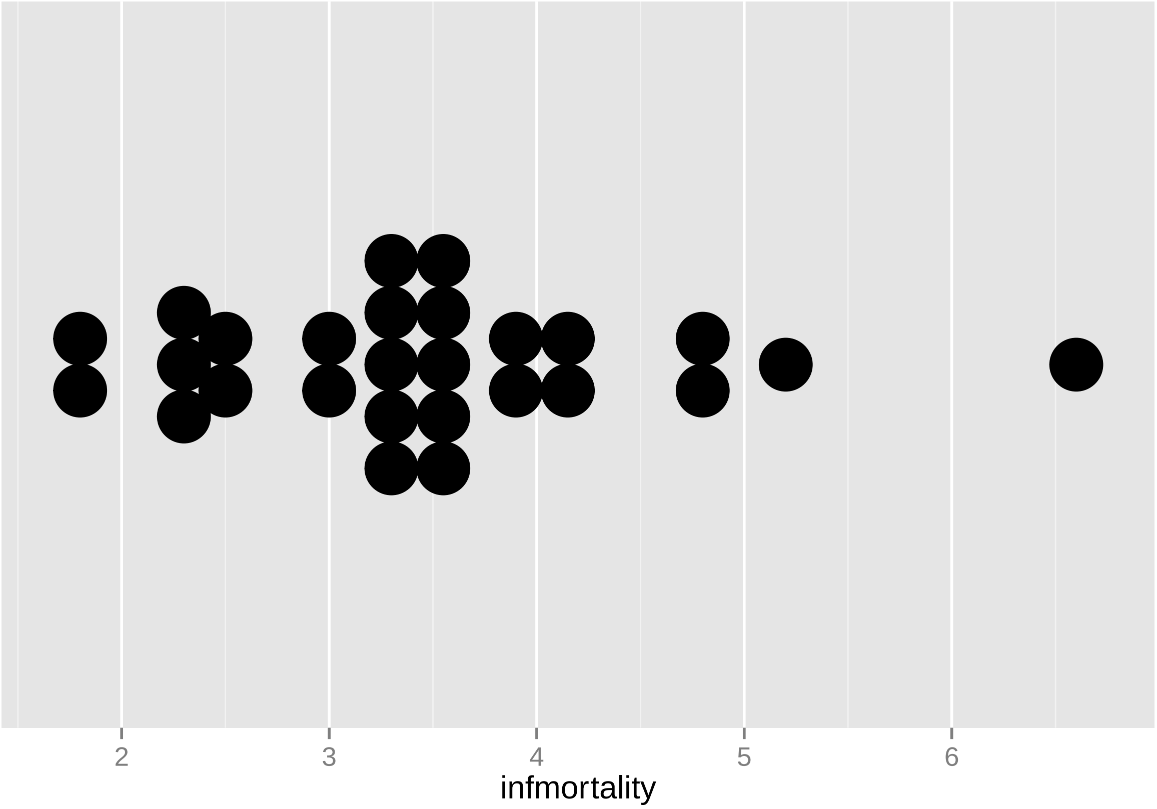
```
# The y-axis labels can be removed by using scale_y_continuous(). We'll also use geom_rug() to show exa
p +
  geom_dotplot( binwidth =.25) +
  geom_rug() +
  scale_y_continuous( breaks = NULL) + # Remove tick markers
  theme( axis.title.y = element_blank()) # Remove axis label
```



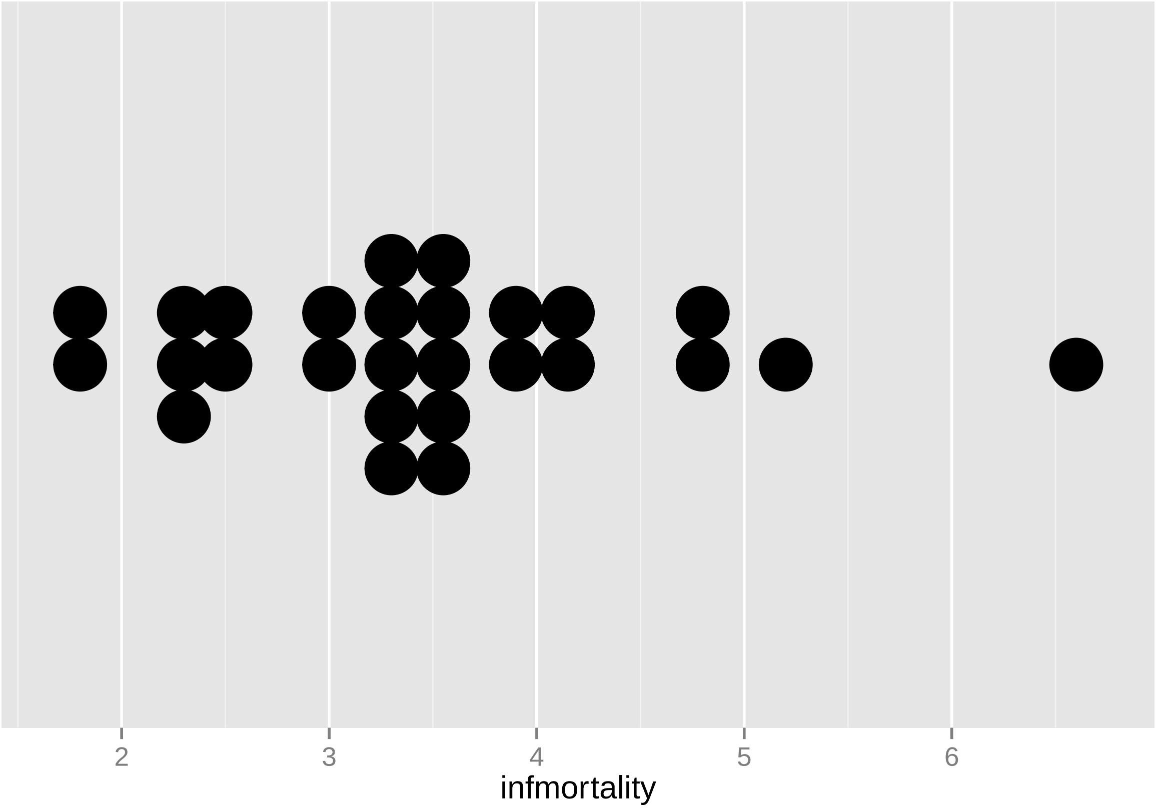
```
# To use bins that are arranged with a fixed, regular spacing, like a histogram, use method =" histodot
p +
  geom_dotplot( method ="histodot", binwidth =.25) +
  geom_rug() +
  scale_y_continuous( breaks = NULL) +
  theme( axis.title.y = element_blank())
```



```
# The dots can also be stacked centered, or centered in such a way that stacks with even and odd quanti

p + geom_dotplot( binwidth =.25, stackdir ="center") +
  scale_y_continuous( breaks = NULL) +
  theme( axis.title.y = element_blank())
```



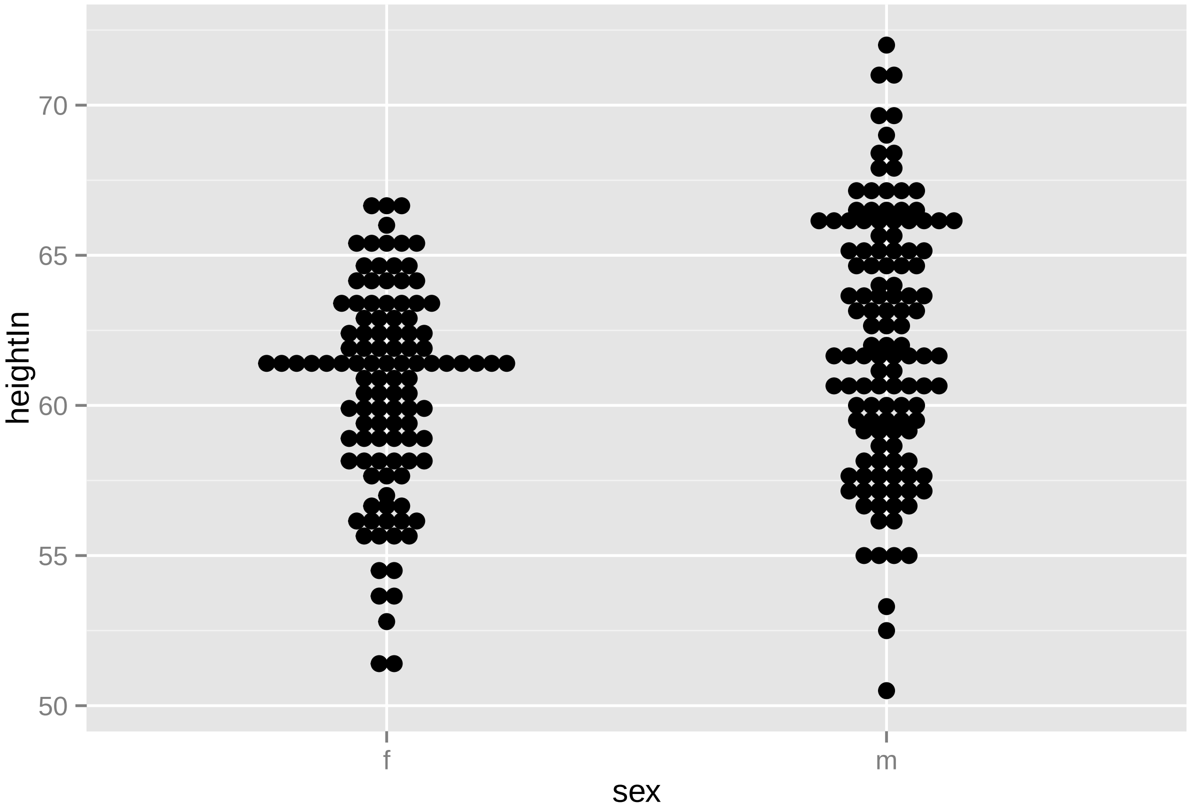
```
p + geom_dotplot( binwidth =.25, stackdir ="centerwhole") +
  scale_y_continuous( breaks = NULL) +
  theme( axis.title.y = element_blank())
```



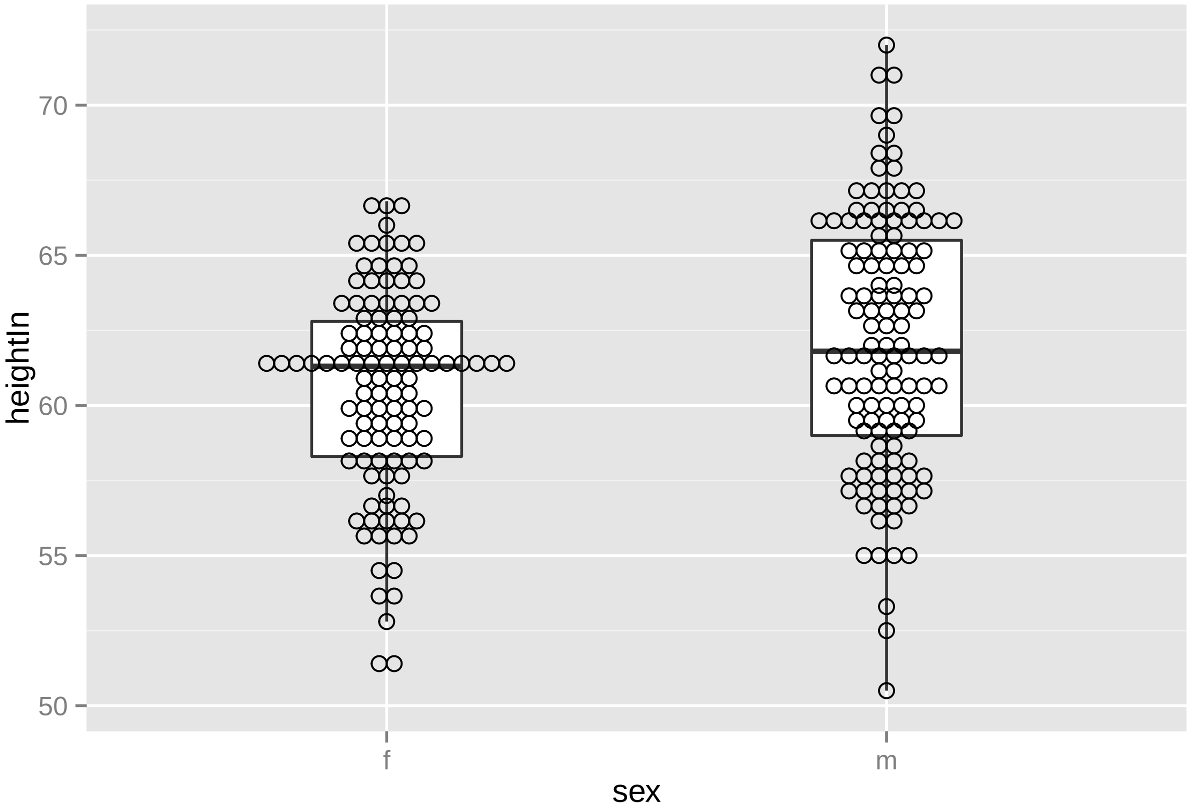
11. Making Multiple Dot Plots for Grouped Data

You want to make multiple dot plots from grouped data.

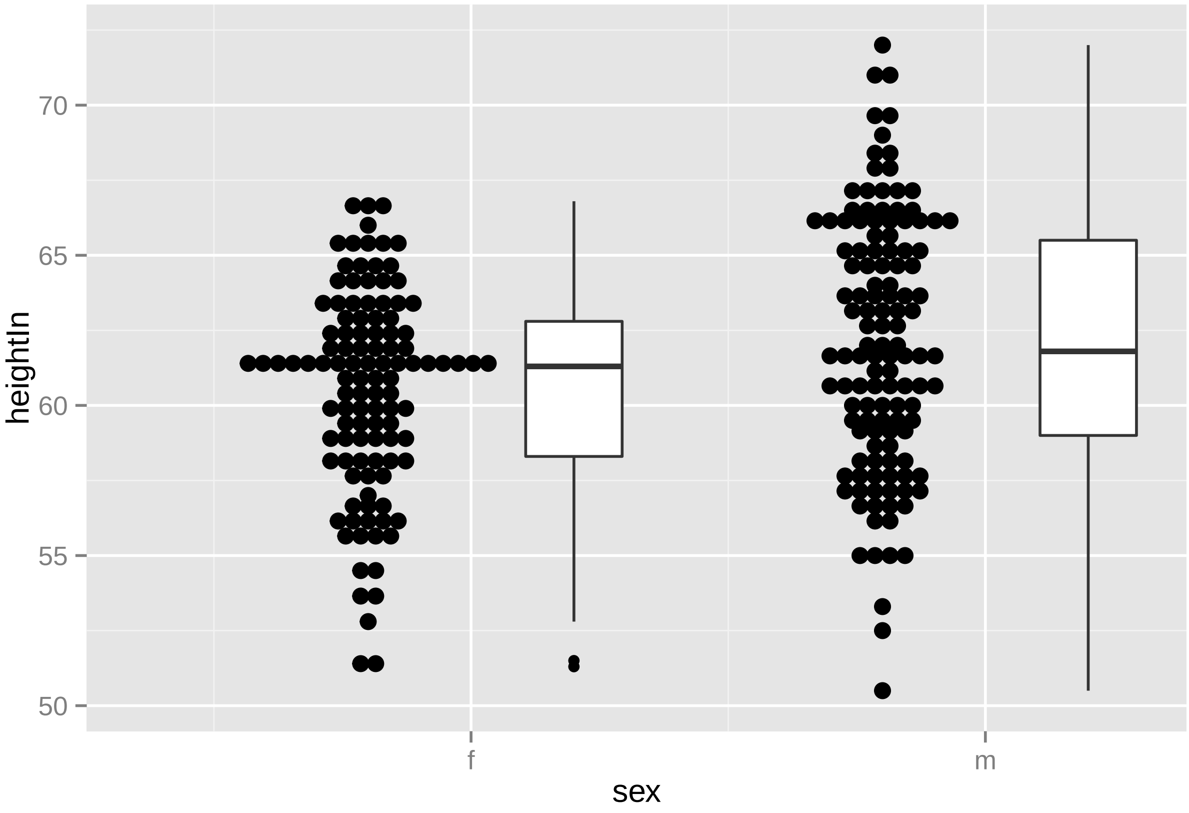
```
ggplot( heightweight, aes( x = sex, y = heightIn)) +
  geom_dotplot( binaxis ="y", binwidth =.5, stackdir ="center")
```



```
ggplot( heightweight, aes( x = sex, y = heightIn)) +
  geom_boxplot( outlier.colour = NA, width =.4) +
  geom_dotplot( binaxis ="y", binwidth =.5, stackdir ="center", fill = NA)
```

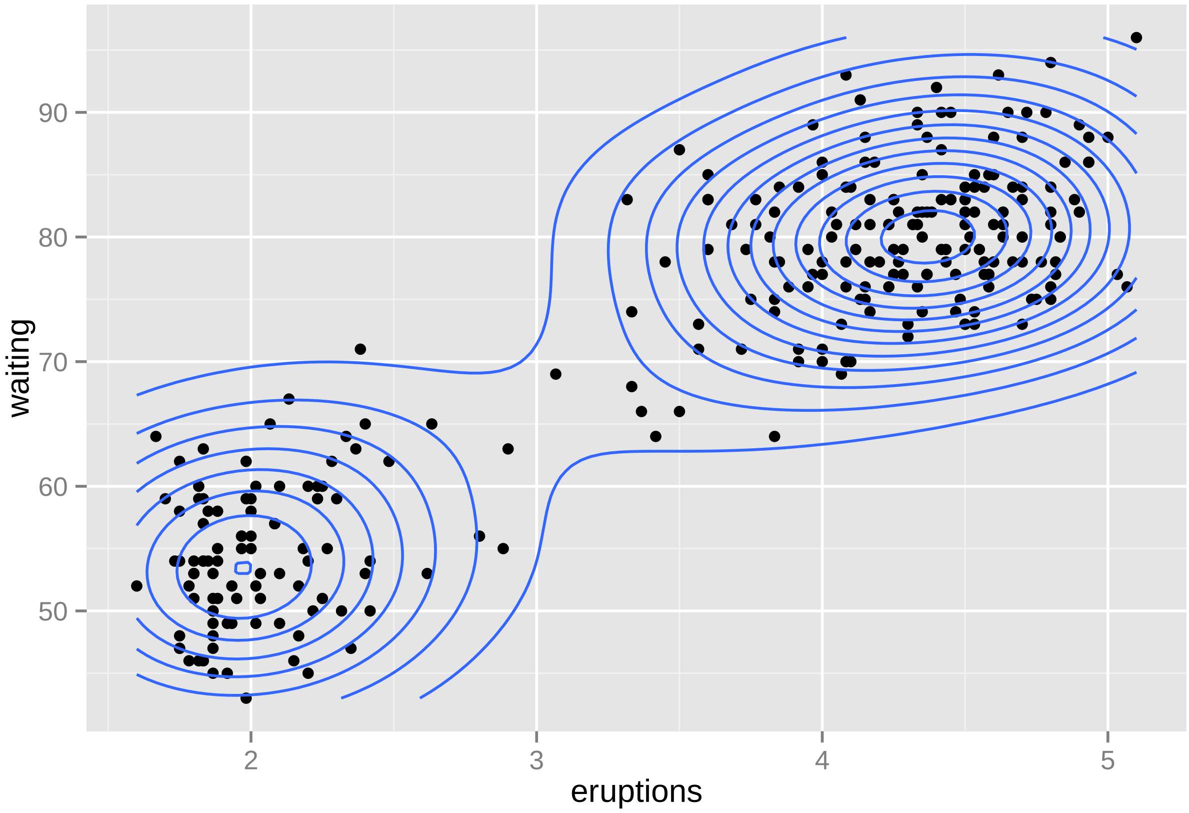


```
# It's also possible to show the dot plots next to the box plots,
ggplot( heightweight, aes( x = sex, y = heightIn)) +
  geom_boxplot( aes( x = as.numeric( sex) + .2, group = sex), width =.25) +
  geom_dotplot( aes( x = as.numeric( sex) - .2, group = sex), binaxis ="y", binwidth =.5, stackdir ="cen
  scale_x_continuous( breaks = 1: nlevels( heightweight$sex), labels = levels( heightweight$sex))
```

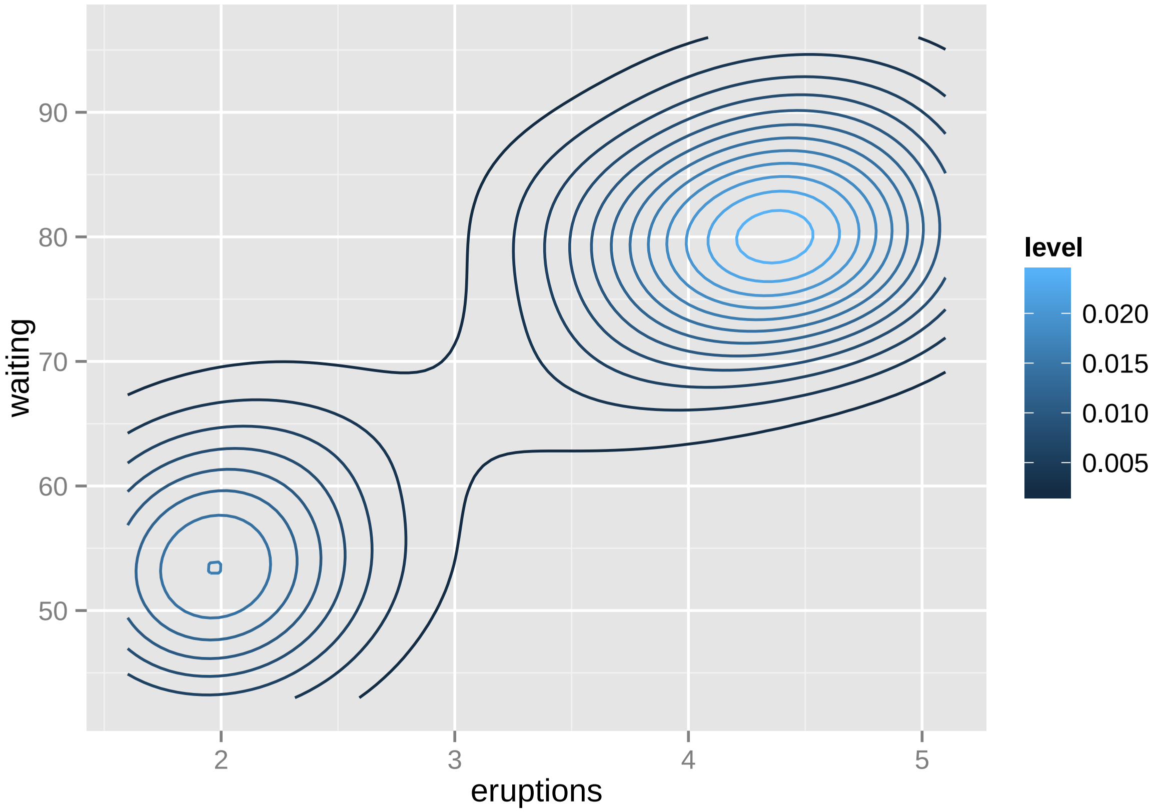


12. Making a Density Plot of Two-Dimensional Data

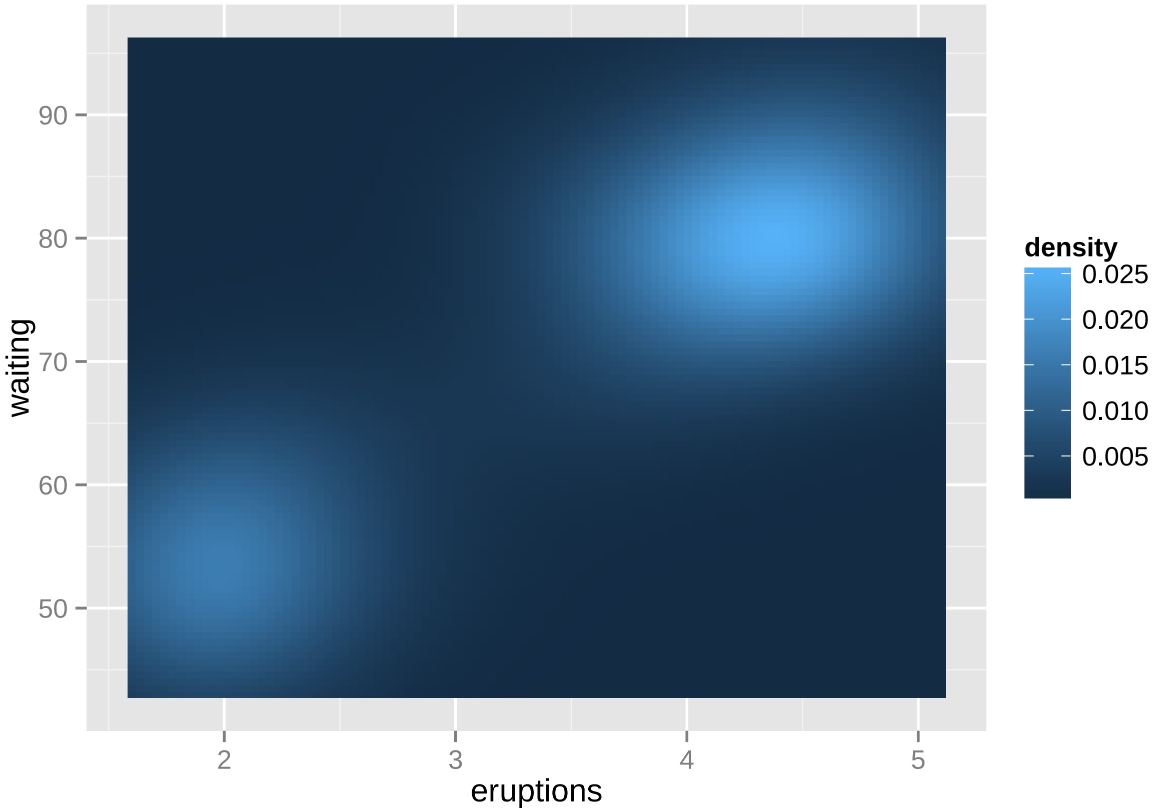
```
# The base plot
p <- ggplot( faithful, aes( x = eruptions, y = waiting))
p +
  geom_point() +
  stat_density2d()
```



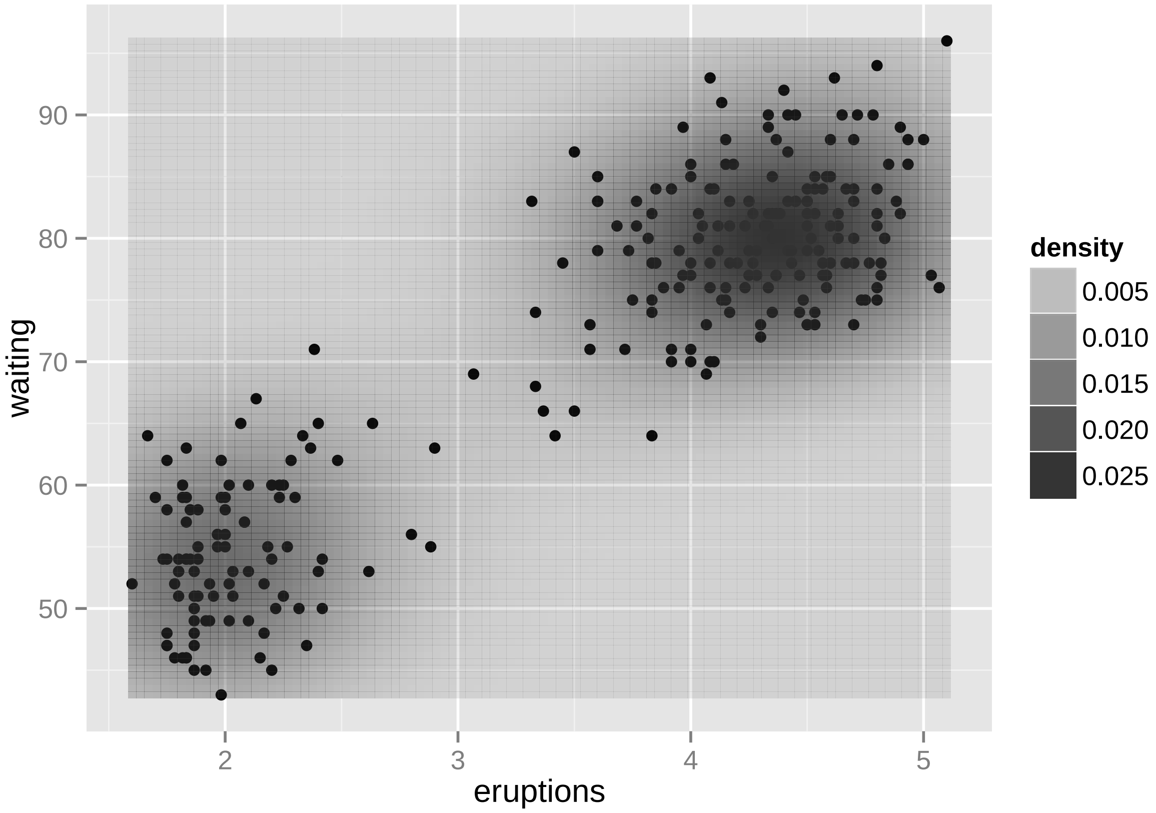
```
# Contour lines, with "height" mapped to color
p +
  stat_density2d( aes( colour =..level..))
```



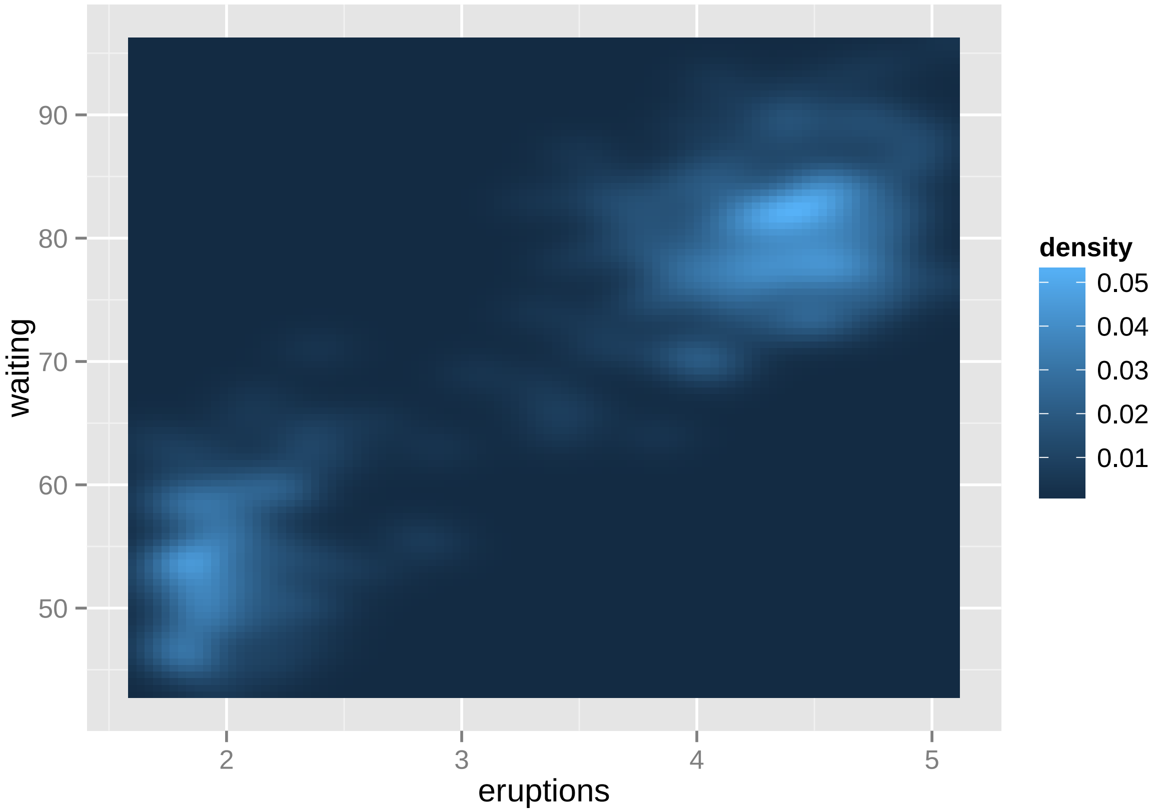
```
# Map density estimate to fill color
p +
  stat_density2d( aes( fill =..density..), geom ="raster", contour = FALSE)
```



```
# With points, and map density estimate to alpha
p +
  geom_point() +
  stat_density2d( aes( alpha =..density..), geom ="tile", contour = FALSE)
```



```
# we'll use a smaller bandwidth in the x and y directions, so that the density estimate is more closely
p +
  stat_density2d( aes( fill =..density..), geom ="raster", contour = FALSE, h = c(.5,5))
```



90
80
70
60
50
waiting
2
3
4
5
eruptions
density
0.05
0.04
0.03
0.02
0.01