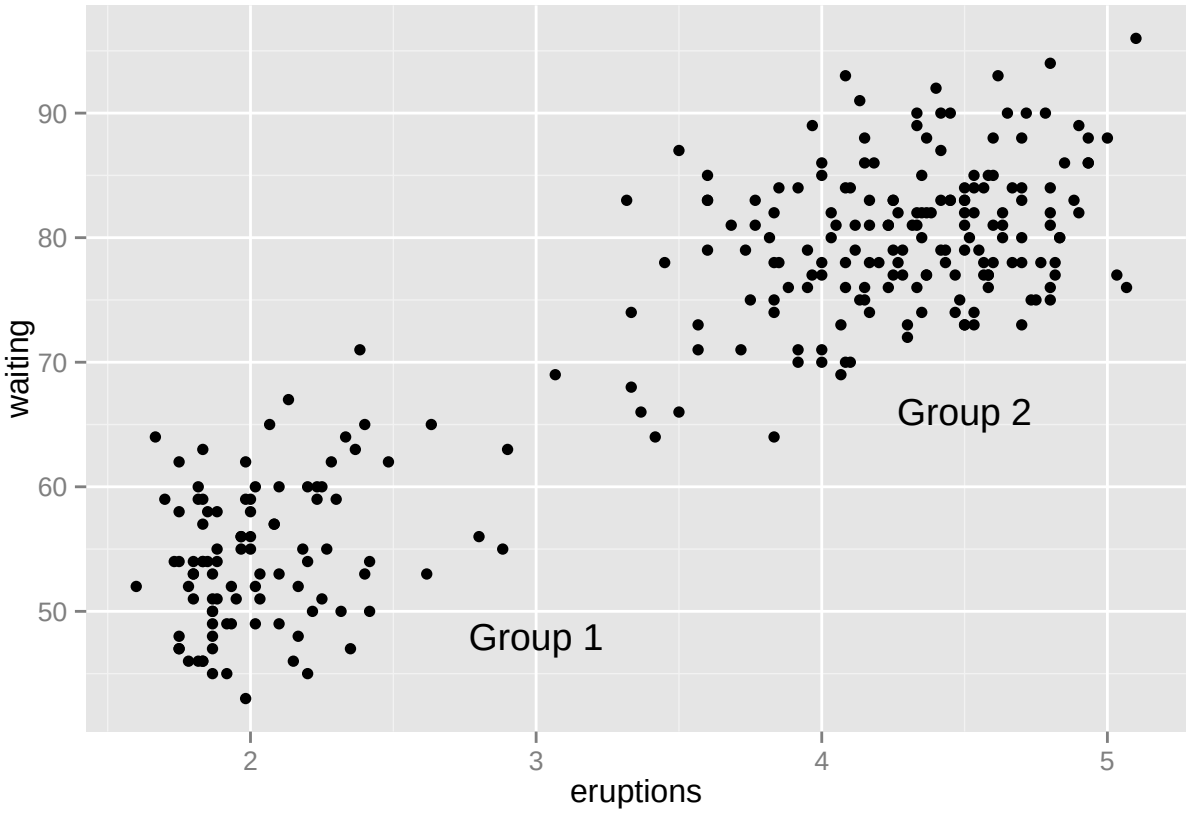
7_Annotations

Gino Tesei

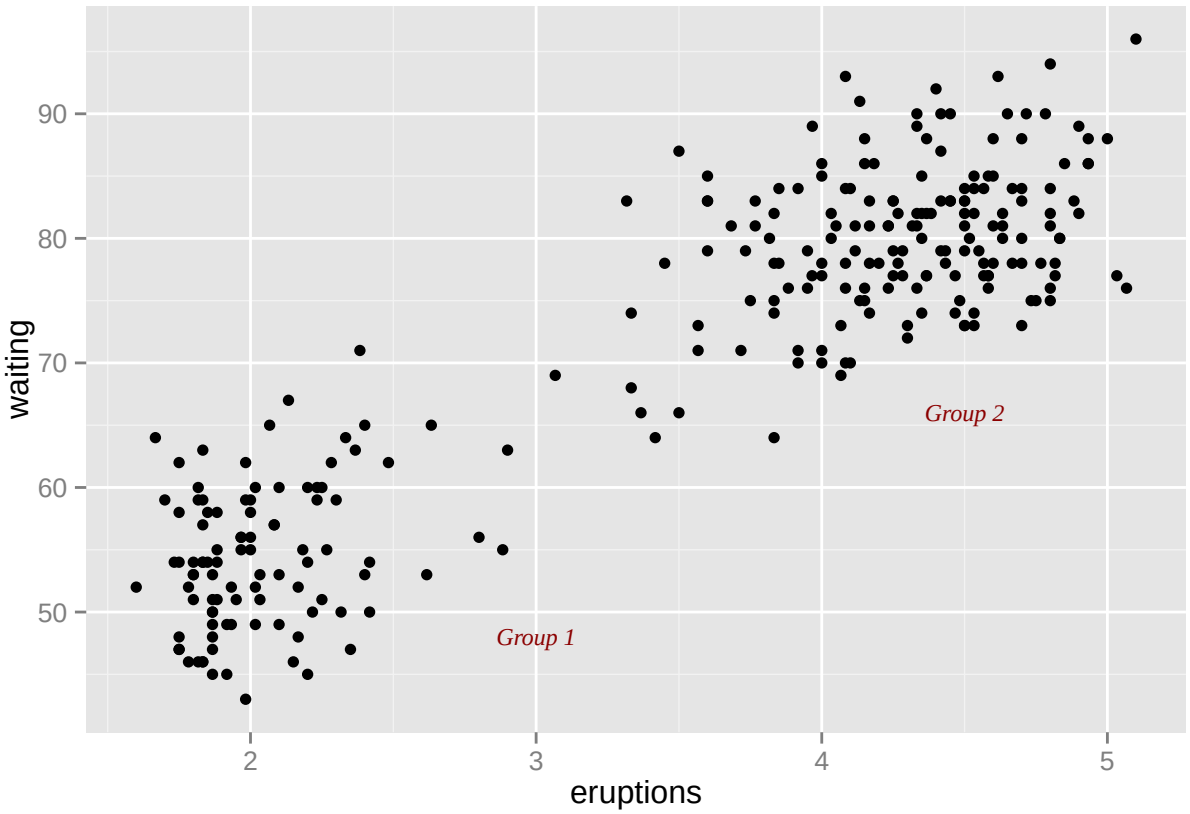
December 13, 2014

1. Adding Text Annotations

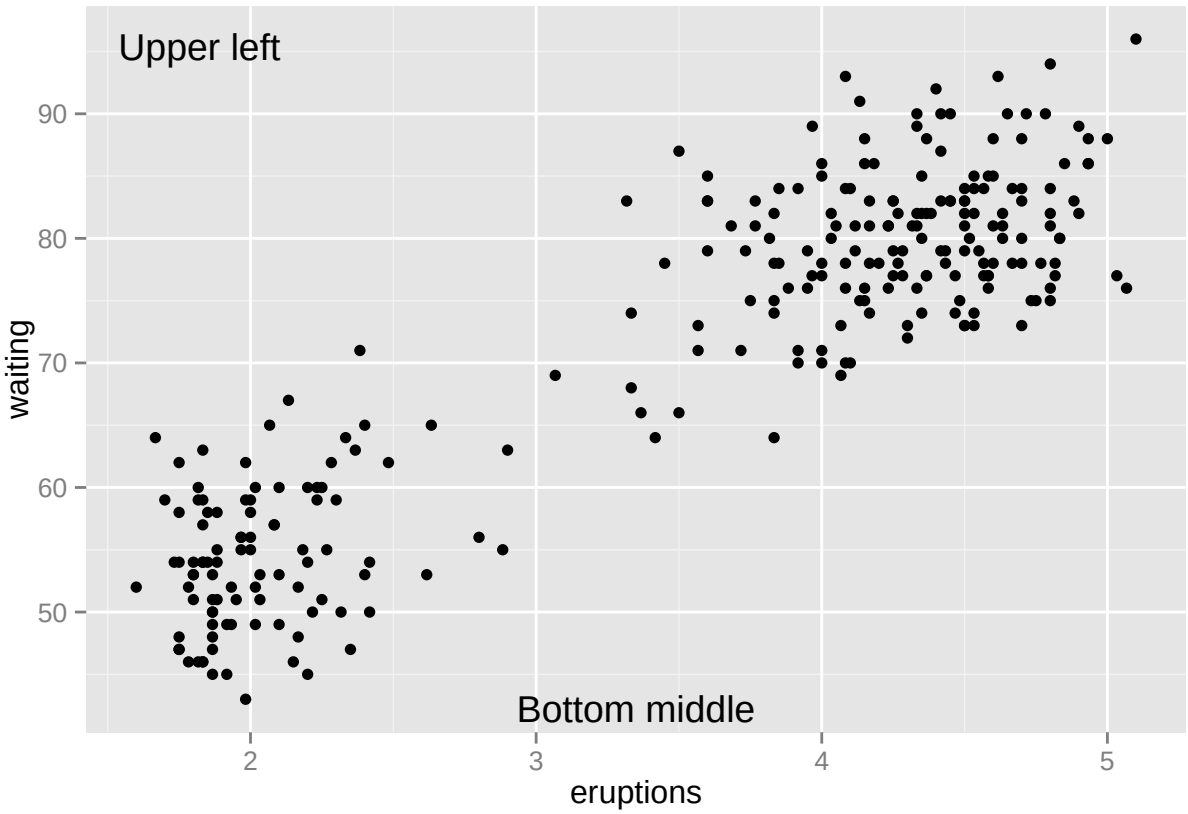
```
library(ggplot2)
library(gcookbook) # For the data set

library(plyr)

# Use annotate() and a text geom
p <- ggplot( faithful, aes( x = eruptions, y = waiting)) +
  geom_point()

p + annotate("text", x = 3, y = 48, label ="Group 1") +
  annotate("text", x = 4.5, y = 66, label ="Group 2")
```




```
# other text properties can be specified,
p +
  annotate("text", x = 3, y = 48, label ="Group 1", family ="serif", fontface ="italic", colour ="darkr
```

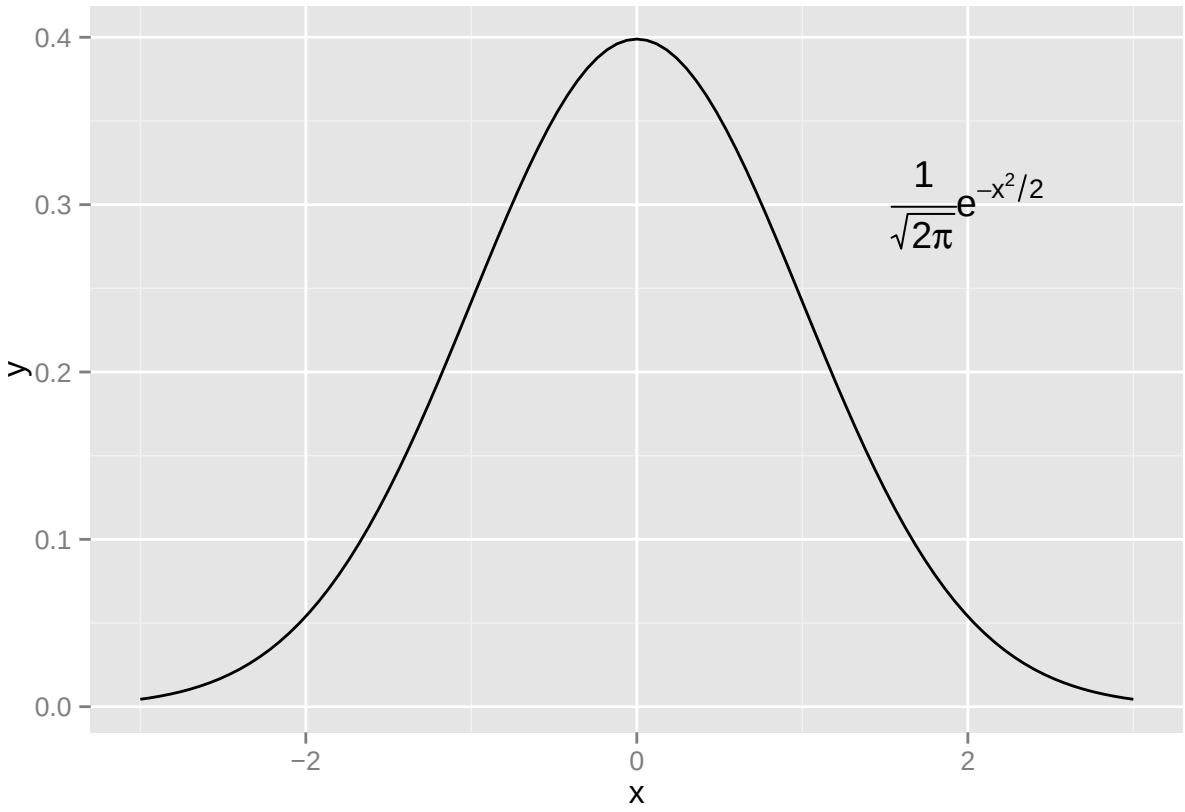



```
# If the axes are continuous, you can use the special values Inf and -Inf to place text annotations at
p +
  annotate("text", x =-Inf, y =Inf, label ="Upper left", hjust =-.2, vjust = 2) +
  annotate("text", x = mean( range( faithful$eruptions)), y =-Inf, vjust =-0.4, label ="Bottom middle")
```

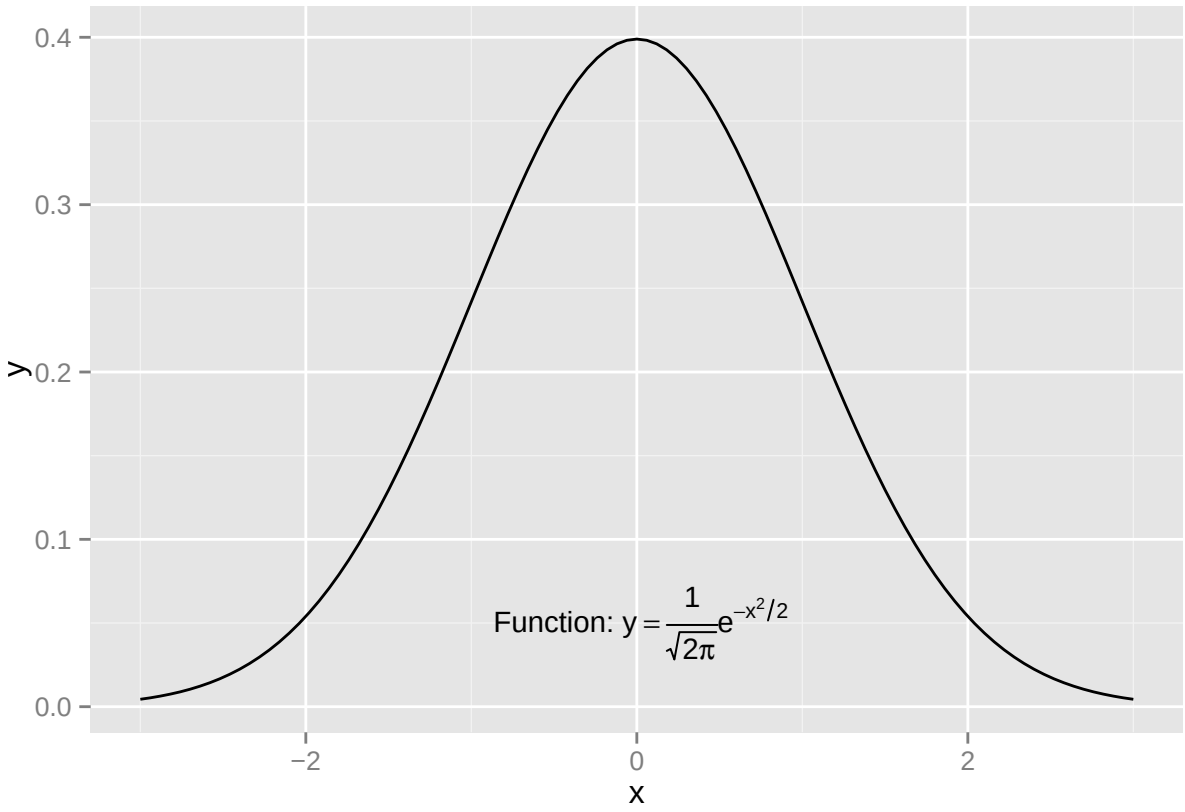


2. Using Mathematical Expressions in Annotations

```
# A normal curve
p <- ggplot( data.frame( x = c(-3,3)), aes( x = x)) +
  stat_function( fun = dnorm)

p + annotate("text", x = 2, y = 0.3, parse = TRUE,
             label ="frac( 1, sqrt( 2 * pi)) * e ^ {-x ^ 2 / 2}")
```

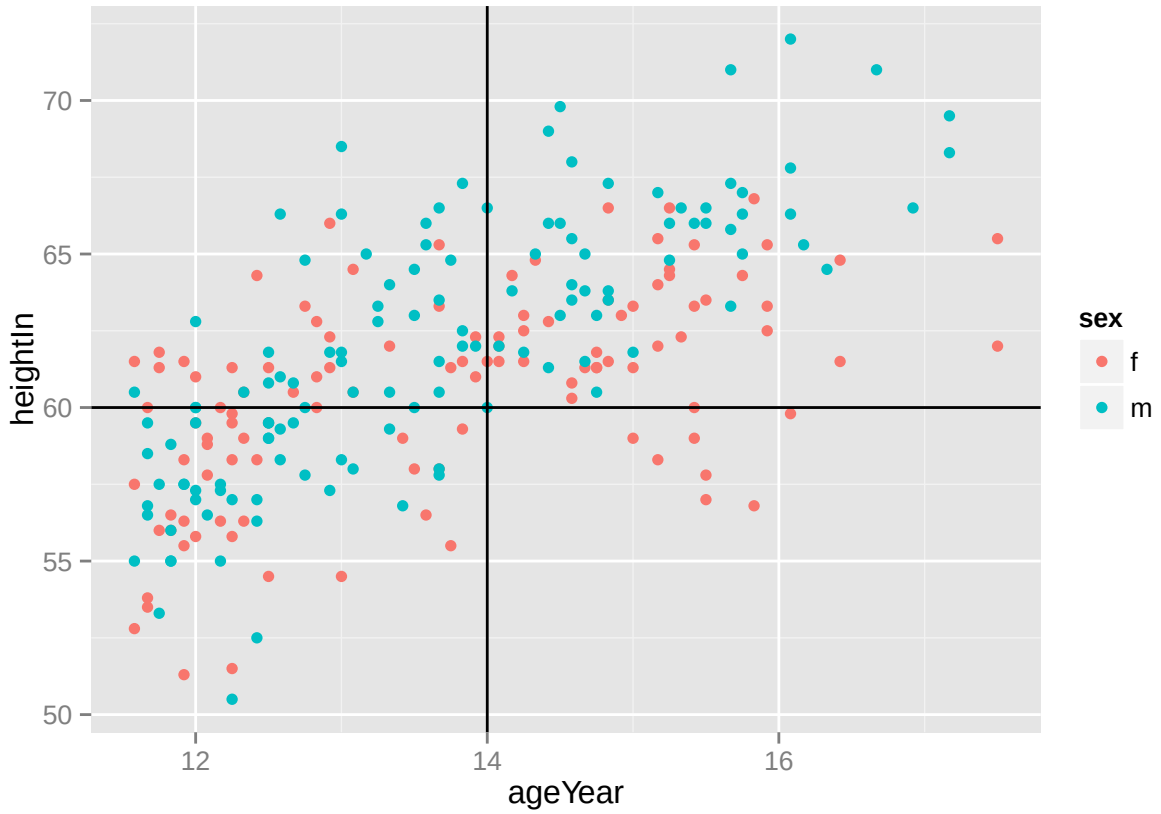



```
# To display two variables next to each other put a * operator between them
p +
  annotate("text", x = 0, y = 0.05, parse = TRUE, size = 4,
           label ="' Function: ' * y == frac( 1, sqrt( 2* pi)) * e ^{-x ^ 2/ 2}")
```

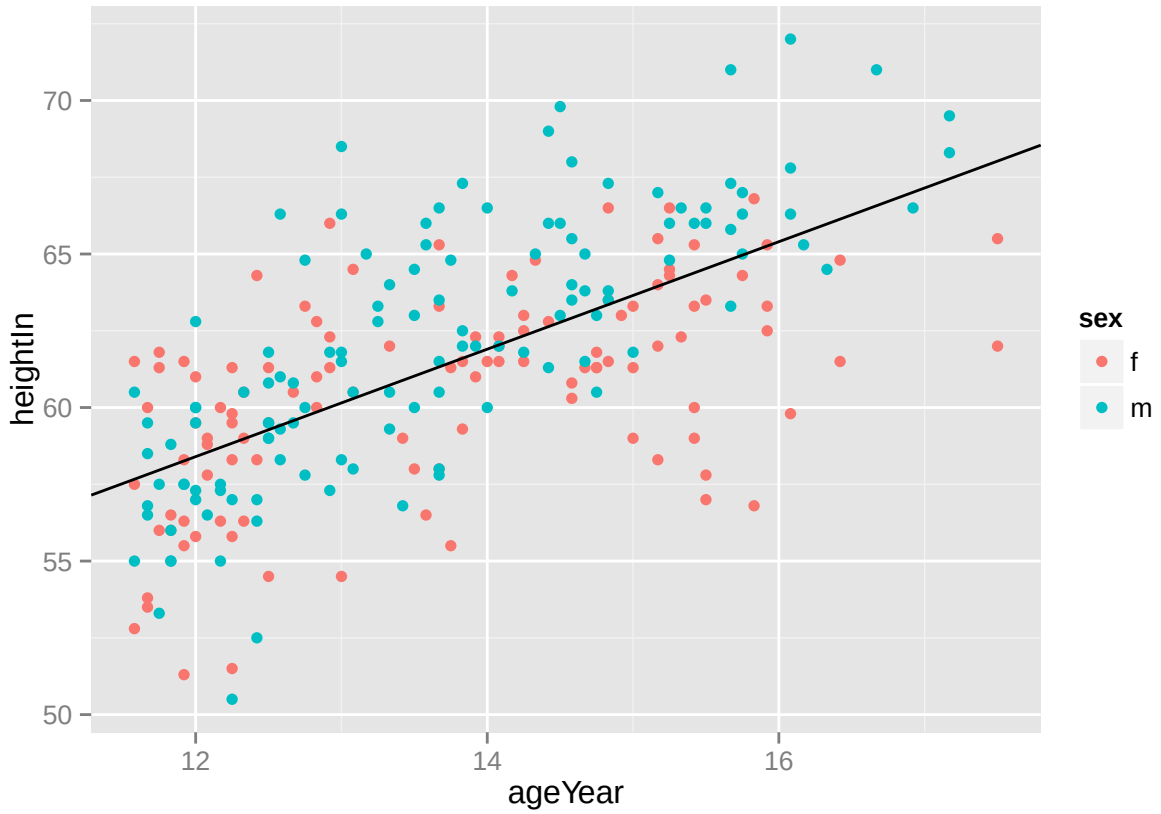


3. Adding Lines

```
p <- ggplot( heightweight, aes( x = ageYear, y = heightIn, colour = sex)) +
  geom_point() # Add horizontal and vertical lines

p +
  geom_hline( yintercept = 60) +
  geom_vline( xintercept = 14) # Add angled line
```



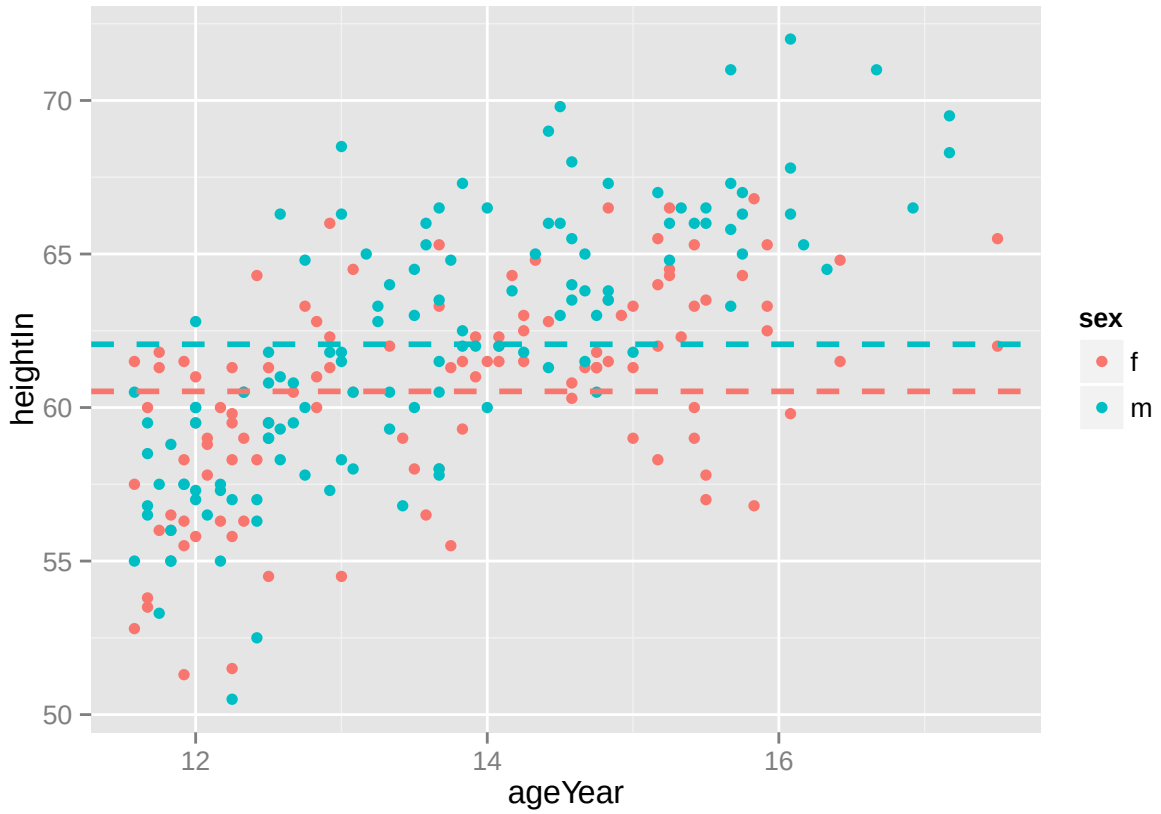

```
p +
  geom_abline( intercept = 37.4, slope = 1.75)
```



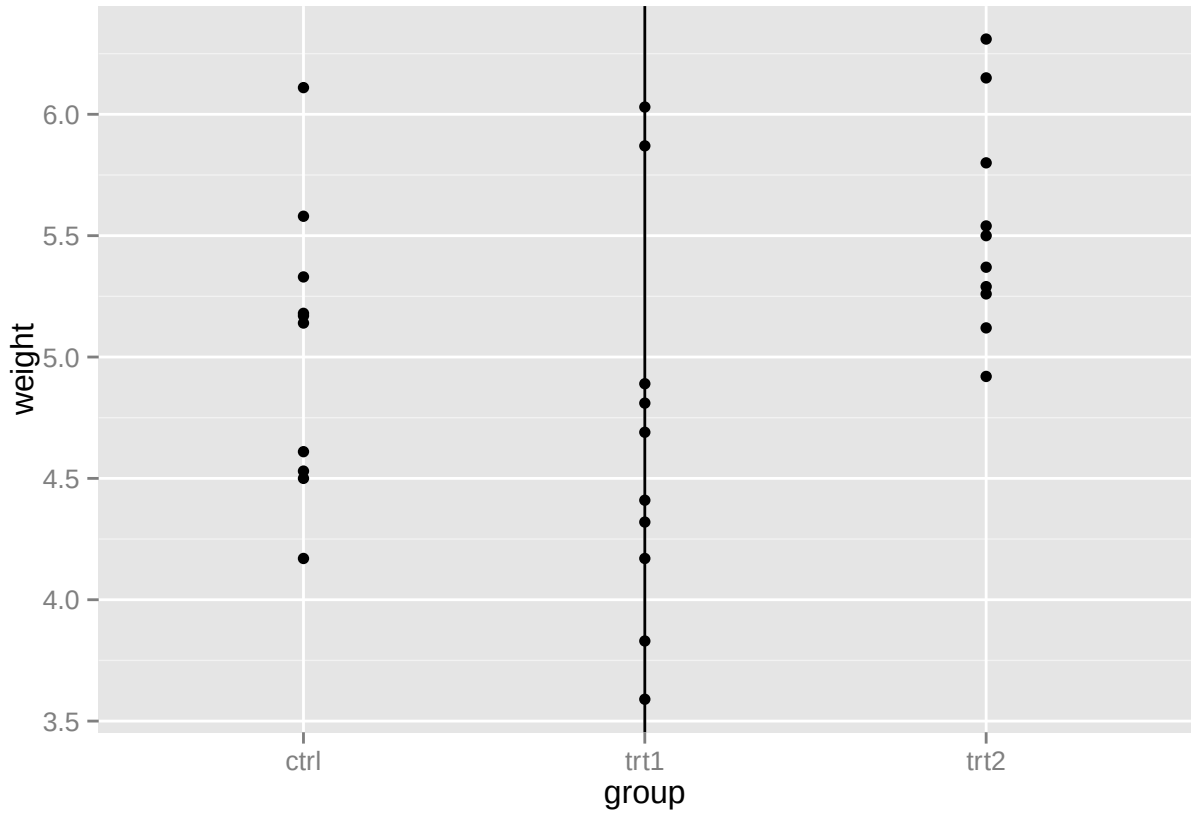

```
# Here we'll take the average height for males and females and store it in a data frame, hw_means
hw_means <- ddply( heightweight, "sex", summarise, heightIn = mean( heightIn))
hw_means
```

```
##   sex heightIn
## 1   f 60.52613
## 2   m 62.06000
```

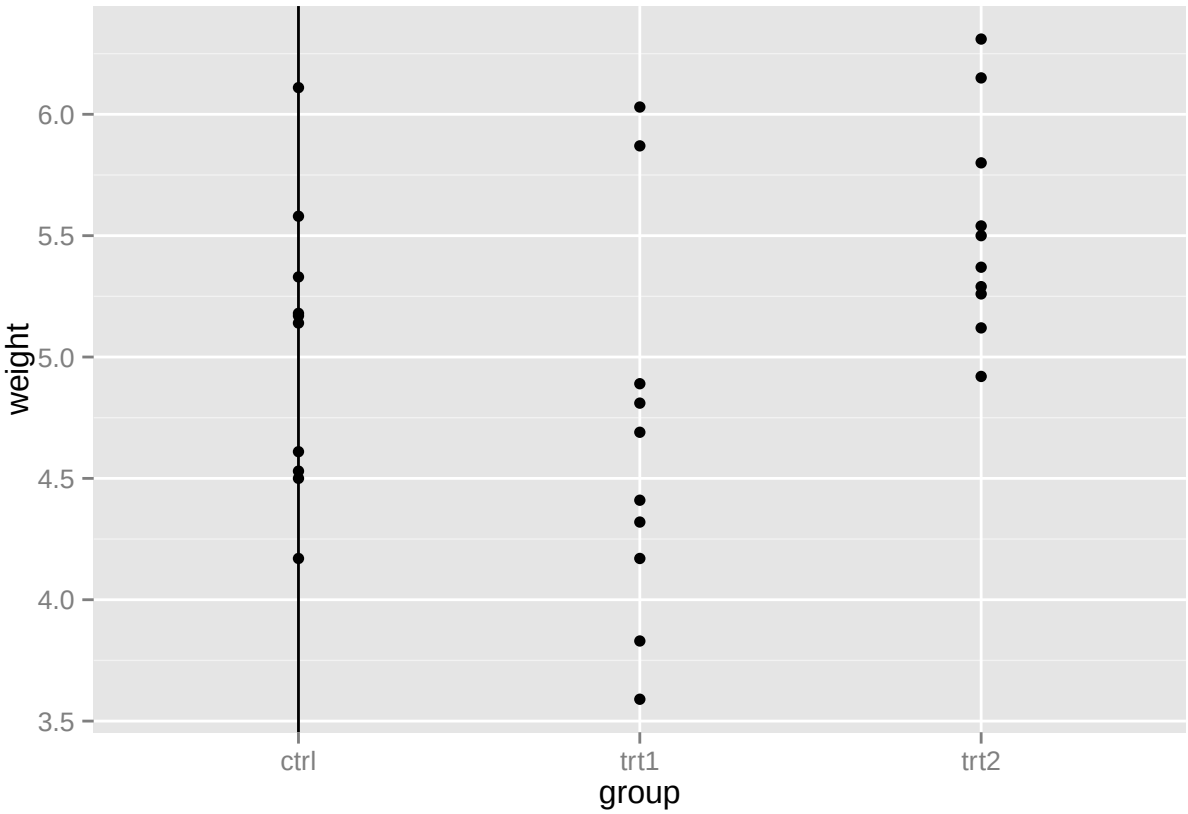
```
p +
  geom_hline( aes( yintercept = heightIn, colour = sex), data = hw_means, linetype ="dashed", size = 1)
```




```
# You can specify the numerical intercept manually, or calculate the numerical value using which( level
pg <- ggplot( PlantGrowth, aes( x = group, y = weight)) +
  geom_point()
pg + geom_vline( xintercept = 2)
```

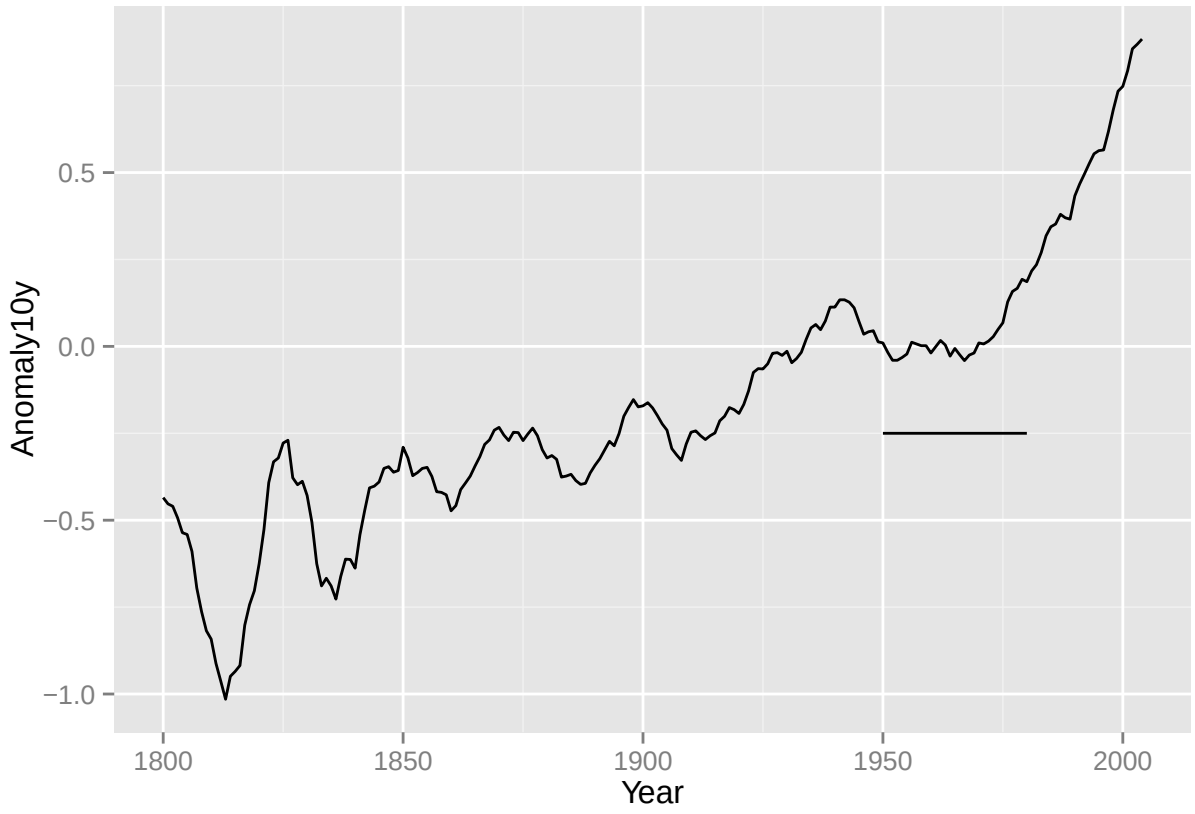



```
pg +
  geom_vline( xintercept = which( levels( PlantGrowth$group) =="ctrl"))
```

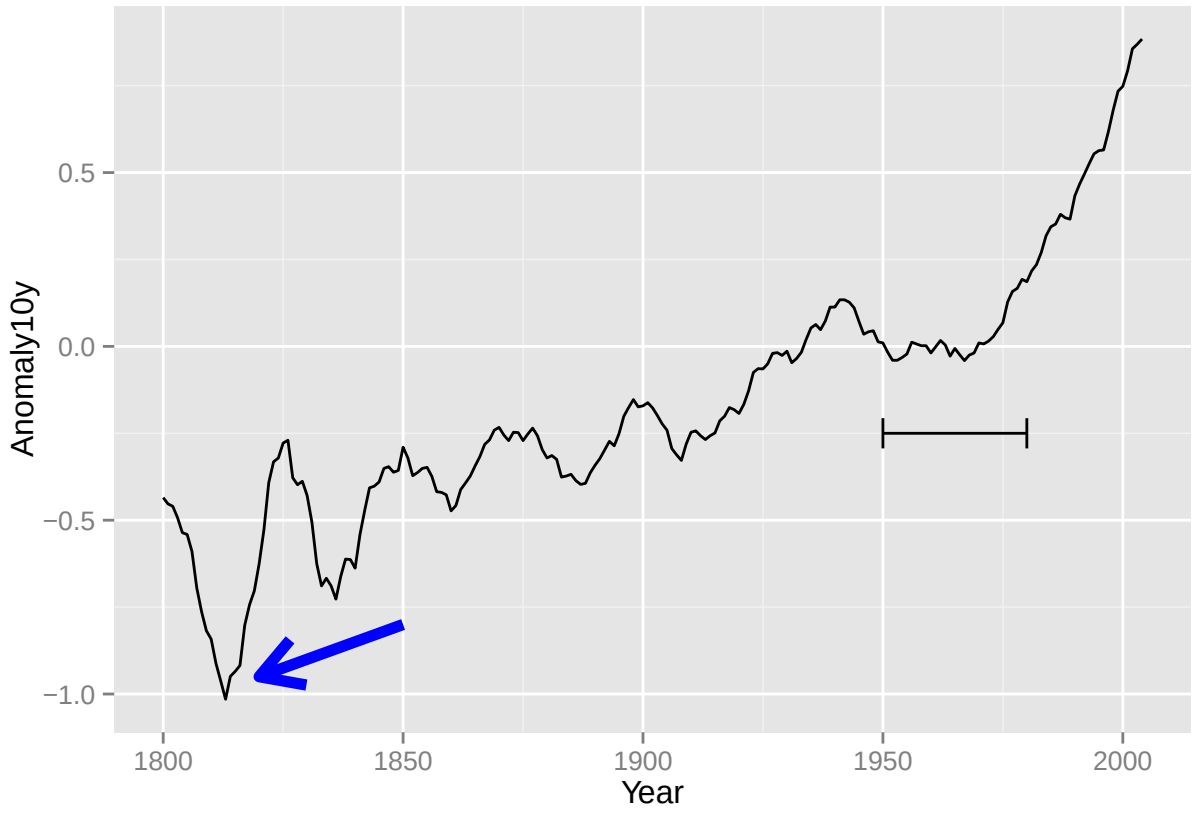


4. Adding Line Segments and Arrows

```
p <- ggplot( subset( climate, Source =="Berkeley"), aes( x = Year,y = Anomaly10y)) +
  geom_line()
p + annotate("segment", x = 1950, xend = 1980, y =-.25, yend =-.25)
```

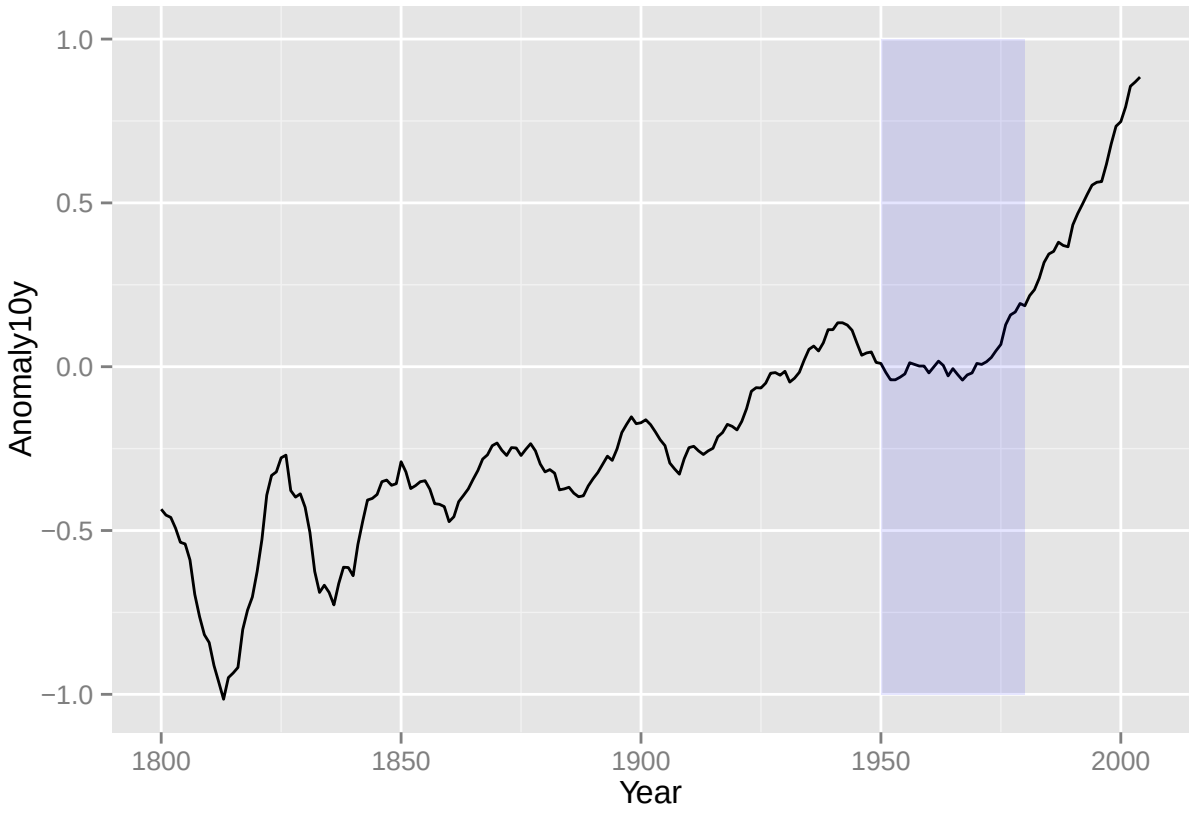



```
# It's possible to add arrowheads or flat ends to the line segments, using arrow() from the grid packag
library(grid)
p +
  annotate("segment", x = 1850, xend = 1820, y =-.8, yend =-.95, colour ="blue", size = 2, arrow = arrow
  annotate("segment", x = 1950, xend = 1980, y =-.25, yend =-.25,
           arrow = arrow( ends ="both", angle = 90, length = unit(.2,"cm")))
```



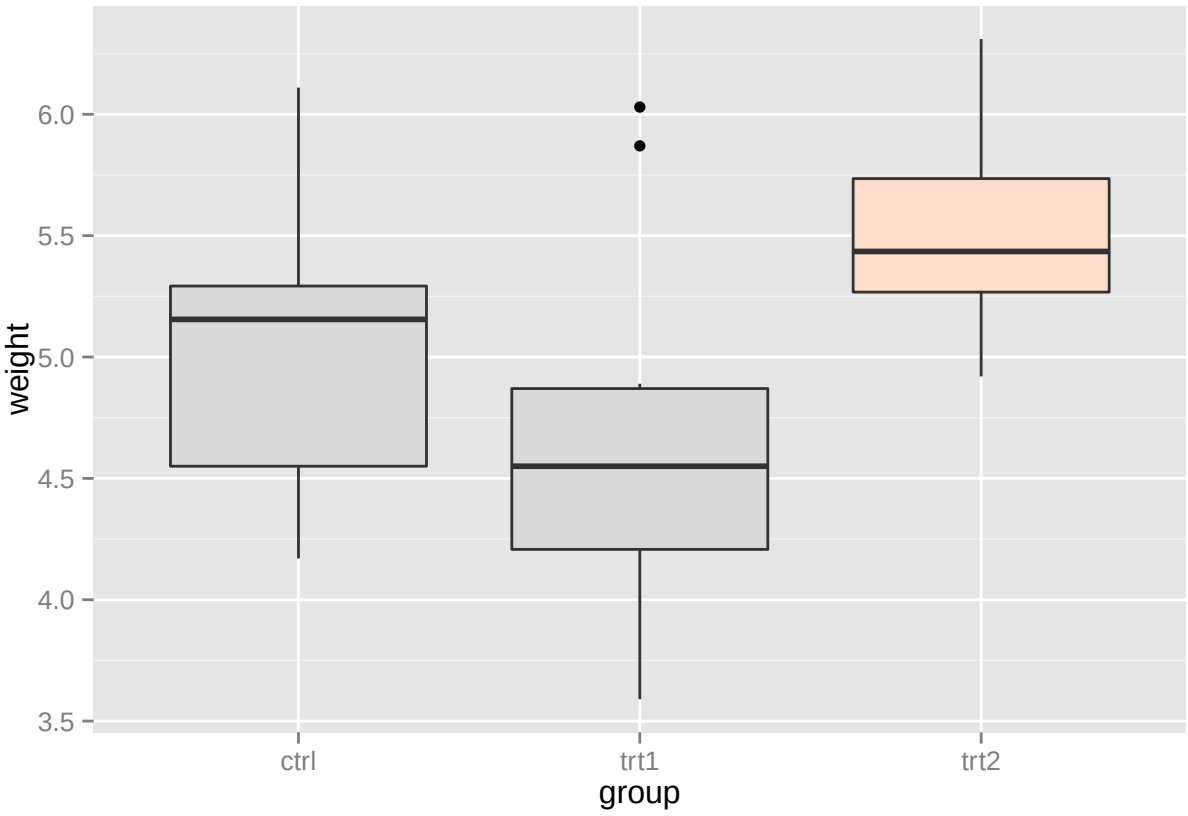
5. Adding a Shaded Rectangle

```
p <- ggplot( subset( climate, Source =="Berkeley"), aes( x = Year, y = Anomaly10y)) +
  geom_line()

p + annotate("rect", xmin = 1950, xmax = 1980, ymin =-1, ymax = 1, alpha =.1, fill ="blue")
```



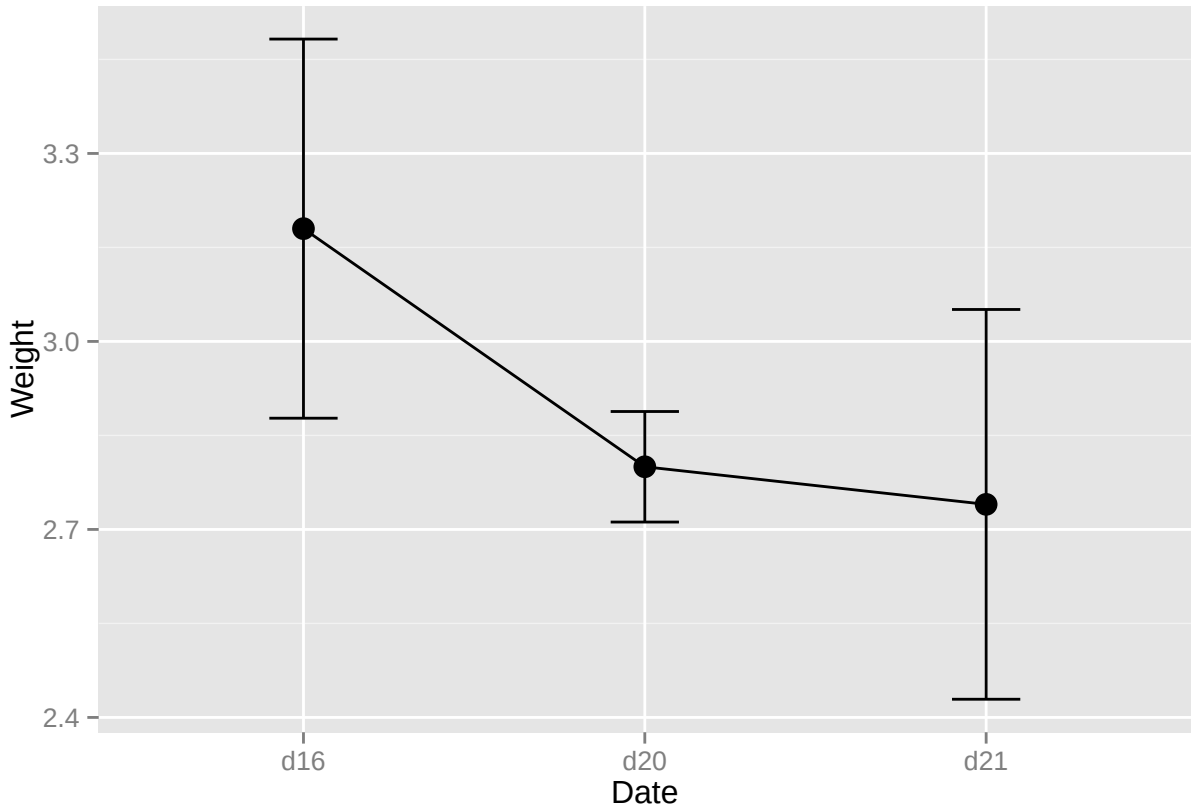
6. Highlighting an Item

```
pg <- PlantGrowth # Make a copy of the PlantGrowth data
pg$hl <- "no" # Set all to "no"
pg$hl[ pg$group =="trt2"] <- "yes" # If group is "trt2", set to "yes"

# Then we'll plot it with manually specified colors and with no legend
ggplot( pg, aes( x = group, y = weight, fill = hl)) +
  geom_boxplot() +
  scale_fill_manual( values = c("grey85", "#FFDDCC"), guide = FALSE)
```

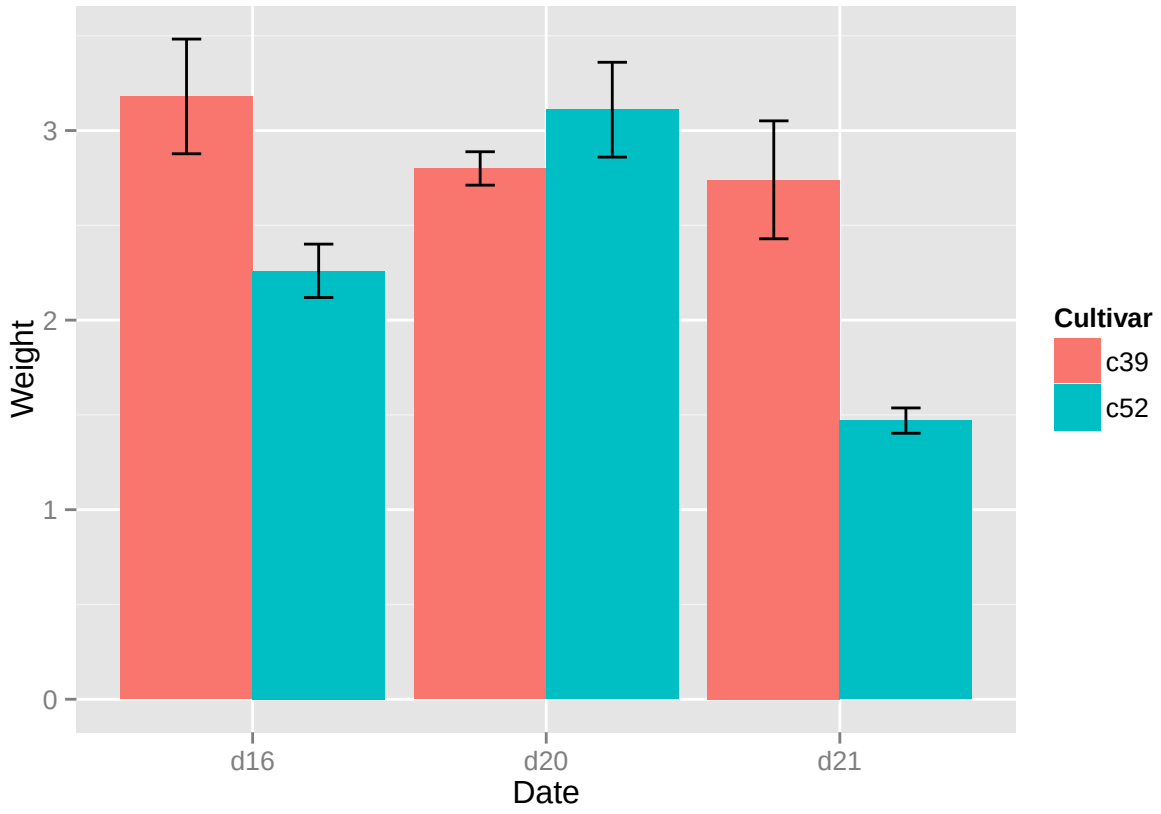


7. Adding Error Bars

```
# Take a subset of the cabbage_exp data for this example
ce <- subset( cabbage_exp, Cultivar == "c39")

# With a line graph
ggplot( ce, aes( x = Date, y = Weight)) +
  geom_line( aes( group = 1)) +
  geom_point( size = 4) + geom_errorbar( aes( ymin = Weight-se, ymax = Weight + se), width =.2)
```

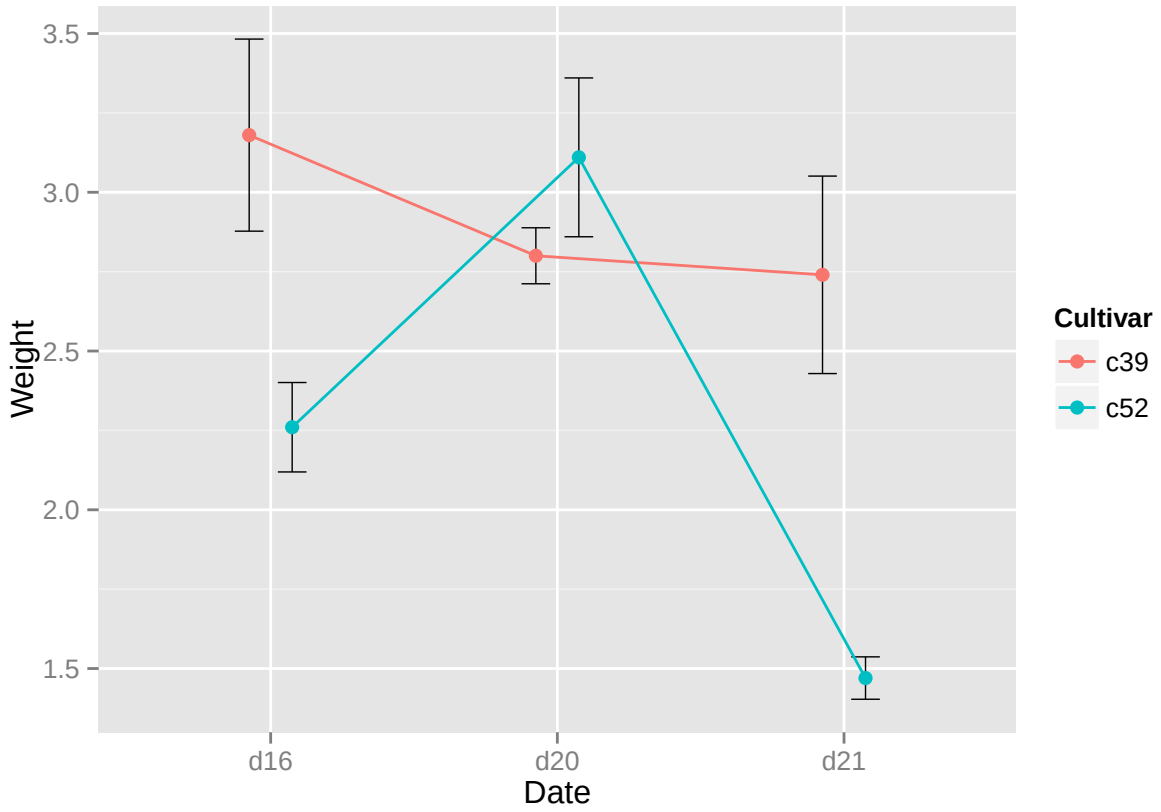



```
# Good: dodge width set to same as bar width (0.9)
ggplot( cabbage_exp, aes( x = Date, y = Weight, fill = Cultivar)) +
  geom_bar( position ="dodge" , stat ="identity") +
  geom_errorbar( aes( ymin = Weight-se, ymax = Weight + se), position = position_dodge( 0.9), width =.2
```




```
# For line graphs, if the error bars are a different color than the lines and points, you should draw t
pd <- position_dodge(.3)

# Save the dodge spec because we use it repeatedly
ggplot( cabbage_exp, aes( x = Date, y = Weight, colour = Cultivar, group = Cultivar)) +
  geom_errorbar( aes( ymin = Weight-se, ymax = Weight + se), width =.2, size = 0.25, colour ="black", p
  geom_line( position = pd) + geom_point( position = pd, size = 2.5)
```

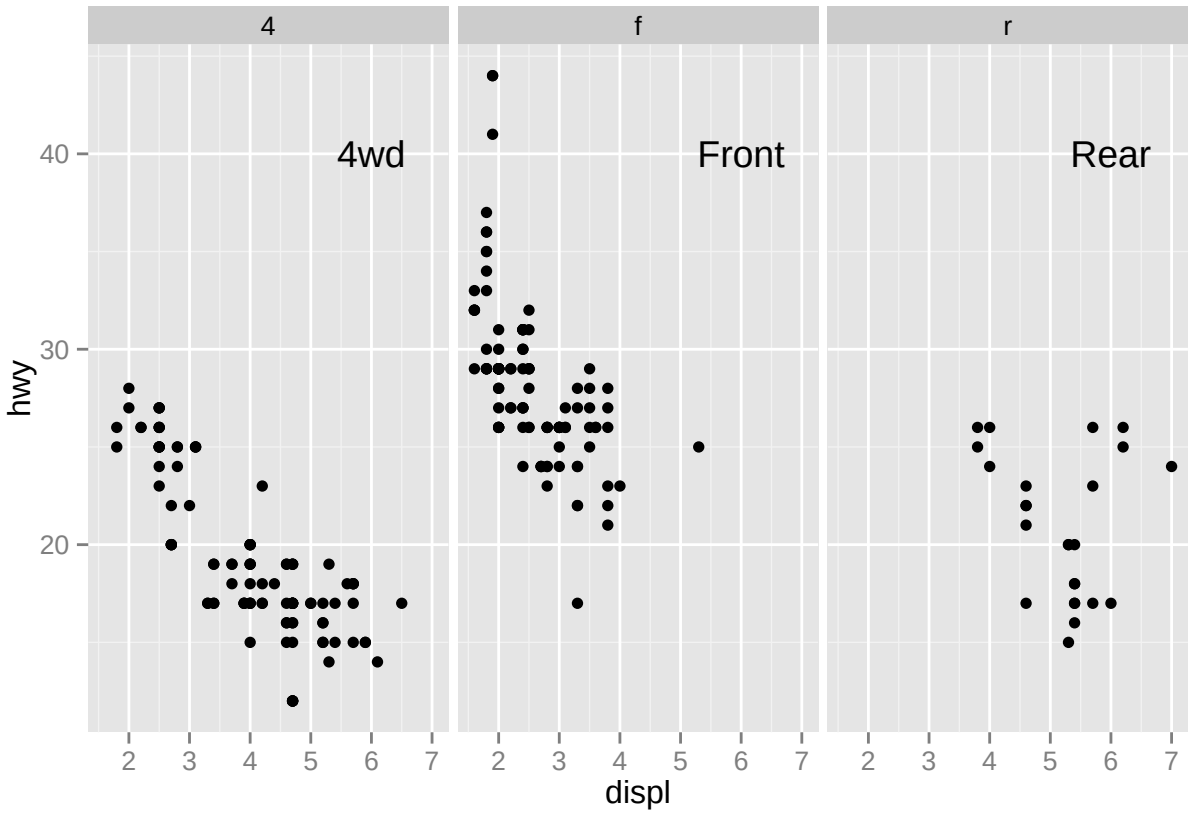
```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```



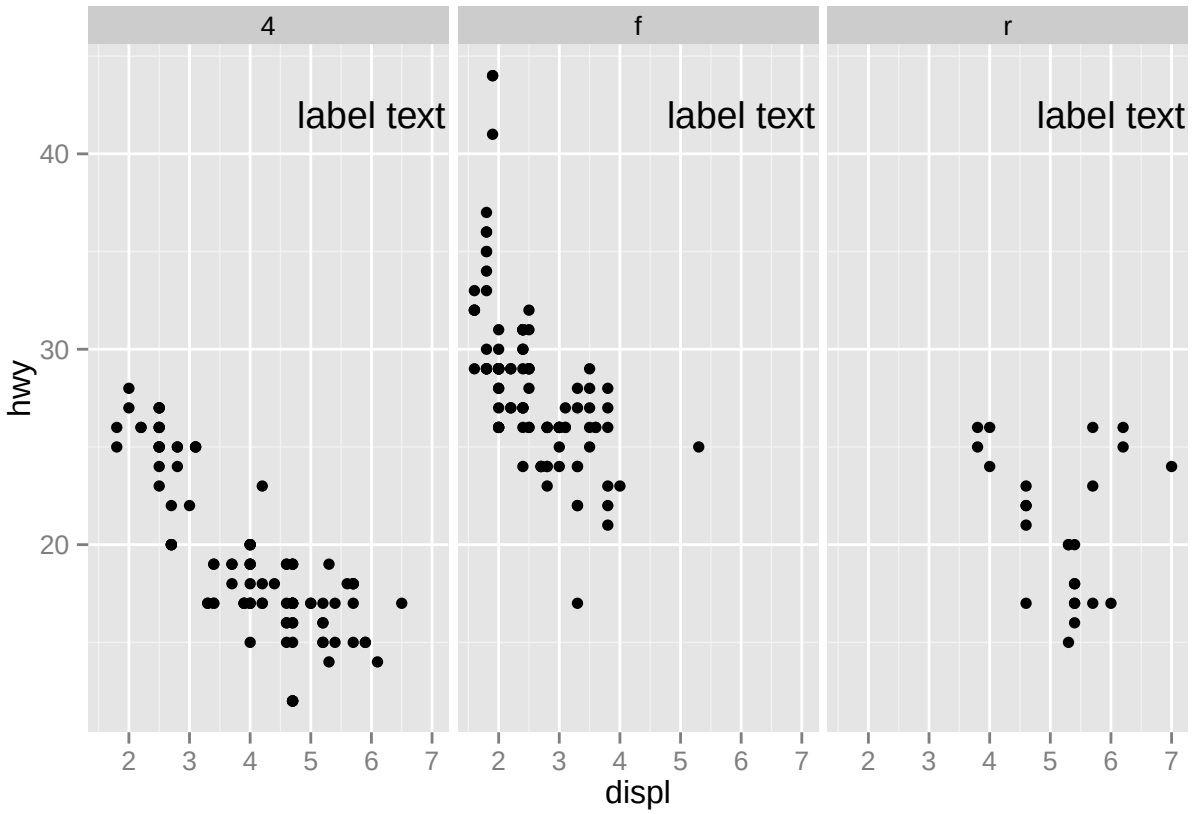

```
# Thinner error bar lines with size = 0.25, and larger points with size = 2.5
```

8. Adding Annotations to Individual Facets

```
# The base plot
p <- ggplot( mpg, aes( x = displ, y = hwy)) + geom_point() + facet_grid(. ~ drv)

# A data frame with labels for each facet
f_labels <- data.frame( drv = c("4", "f", "r"), label = c("4wd", "Front", "Rear"))
p + geom_text( x = 6, y = 40, aes( label = label), data = f_labels)
```



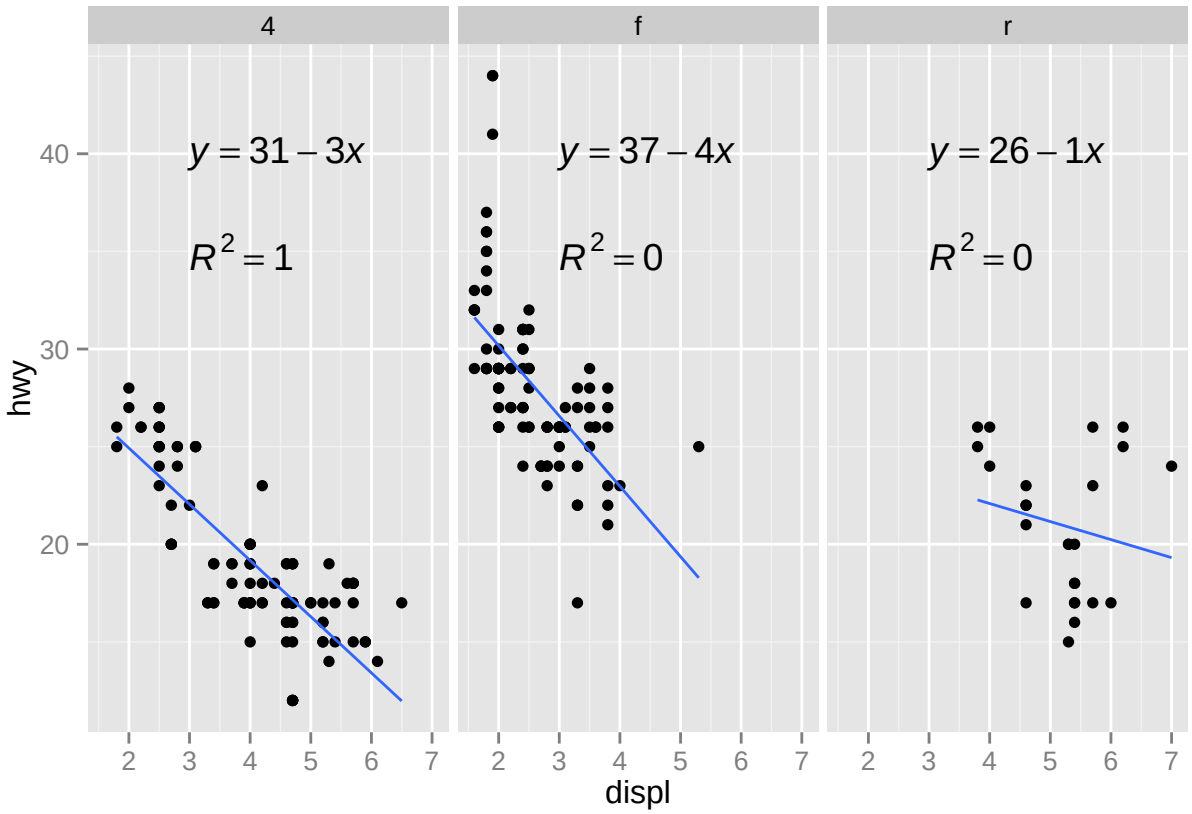

```
# If you use annotate(), the label will appear in all facets
p +
  annotate("text", x = 6, y = 42, label ="label text")
```




```
# This function returns a data frame with strings representing the regression
# equation, and the r ^ 2 value # These strings will be treated as R math expressions
lm_labels <- function( dat) {
  mod <- lm( hwy ~ displ, data = dat)
  formula <- sprintf(" italic(y) == %. 2f %+. 2f * italic(x)",
                     round( coef( mod)[1], 2),
                     round( coef( mod)[2], 2))
  r <- cor( dat$displ, dat$hwy)
  r2 <- sprintf(" italic( R ^ 2) == %. 2f", r ^ 2)
  data.frame( formula = formula, r2 = r2, stringsAsFactors = FALSE)
  }
# For the ddply() function
labels <- ddply( mpg, "drv", lm_labels)
labels
```

```
##   drv                          formula                    r2
## 1   4  italic(y) ==  31 -3 * italic(x)  italic( R ^ 2) ==  1
## 2   f  italic(y) ==  37 -4 * italic(x)  italic( R ^ 2) ==  0
## 3   r  italic(y) ==  26 -1 * italic(x)  italic( R ^ 2) ==  0
```

```
# Plot with formula and R ^ 2 values
p +
  geom_smooth( method = lm, se = FALSE) +
  geom_text( x = 3, y = 40, aes( label = formula), data = labels, parse = TRUE, hjust = 0) +
  geom_text( x = 3, y = 35, aes( label = r2), data = labels, parse = TRUE, hjust = 0)
```


4
f
r
$y = 31 - 3x$
$R^2 = 1$
$y = 37 - 4x$
$R^2 = 0$
$y = 26 - 1x$
$R^2 = 0$
hwy
40
30
20
2
3
4
5
6
7
displ