

3_Bar_Graphs

Gino Tesei

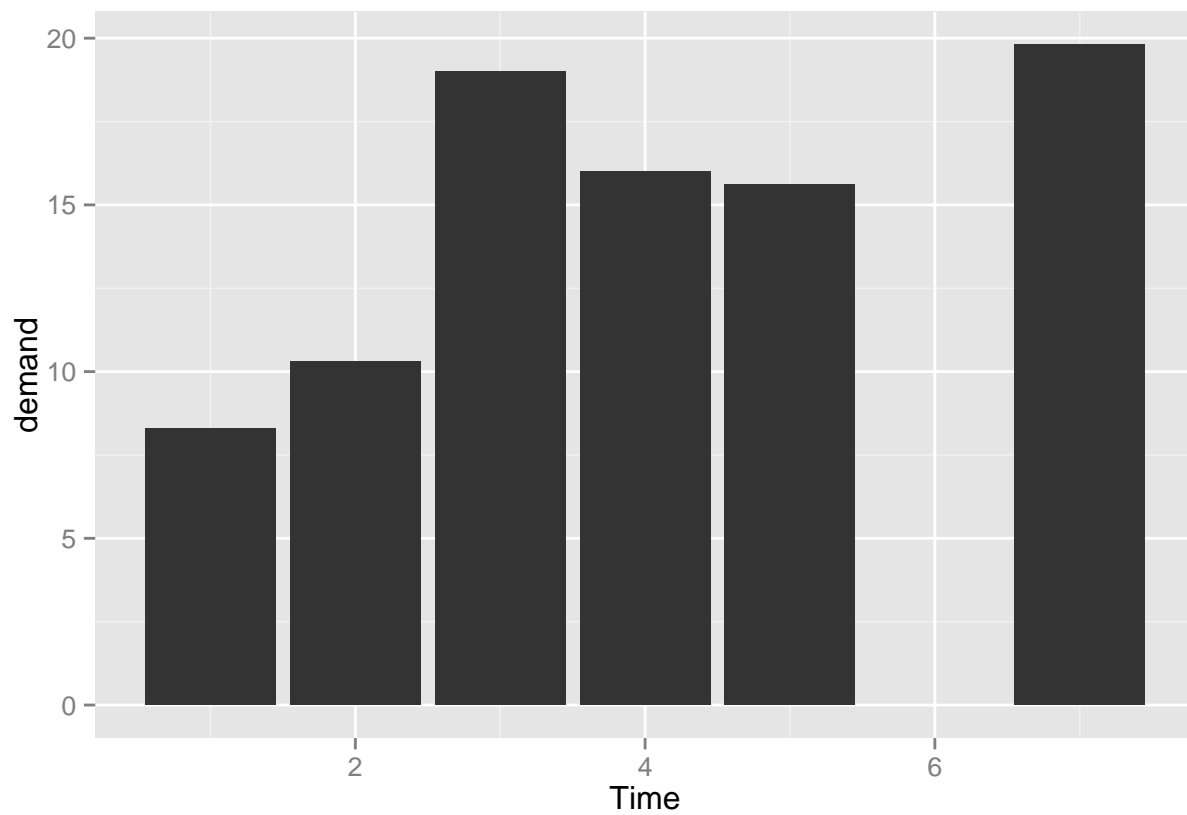
December 12, 2014

1. Making a Basic Bar Graph

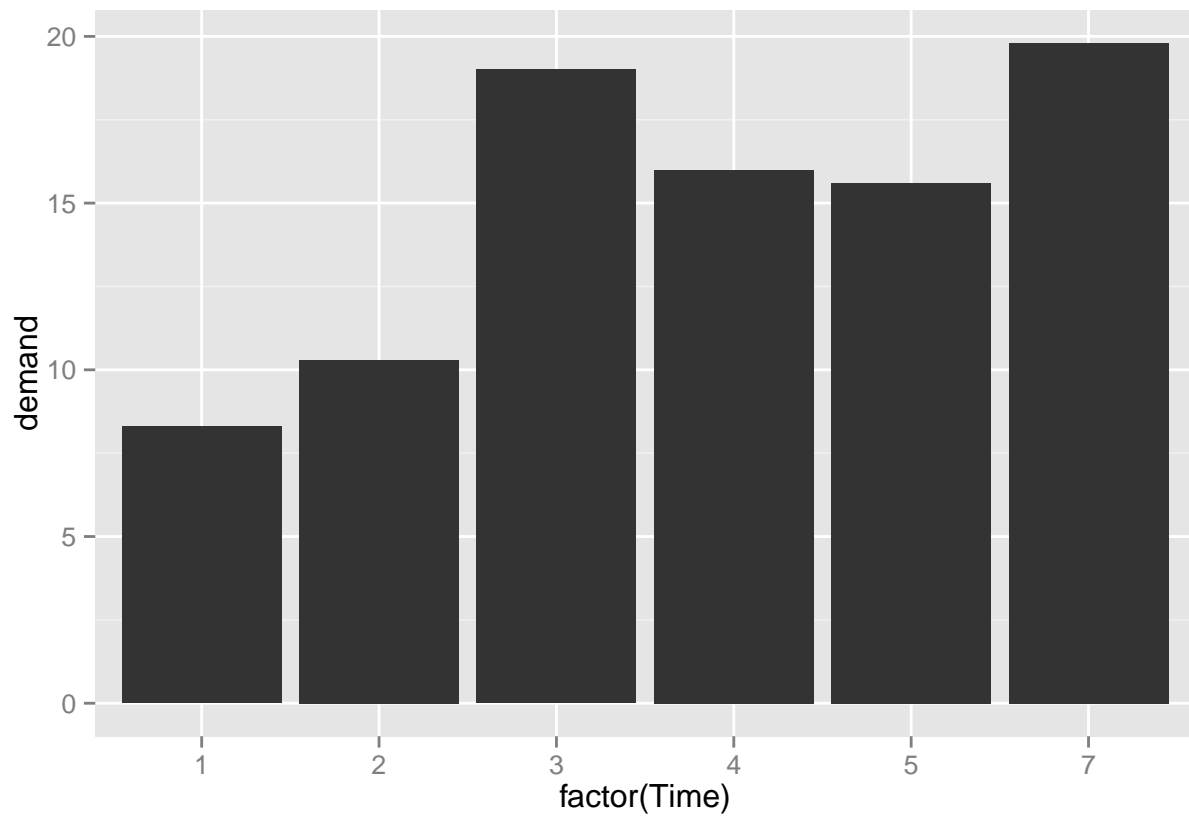
```
library(ggplot2)
library( gcookbook) # For the data set

library(plyr) ##

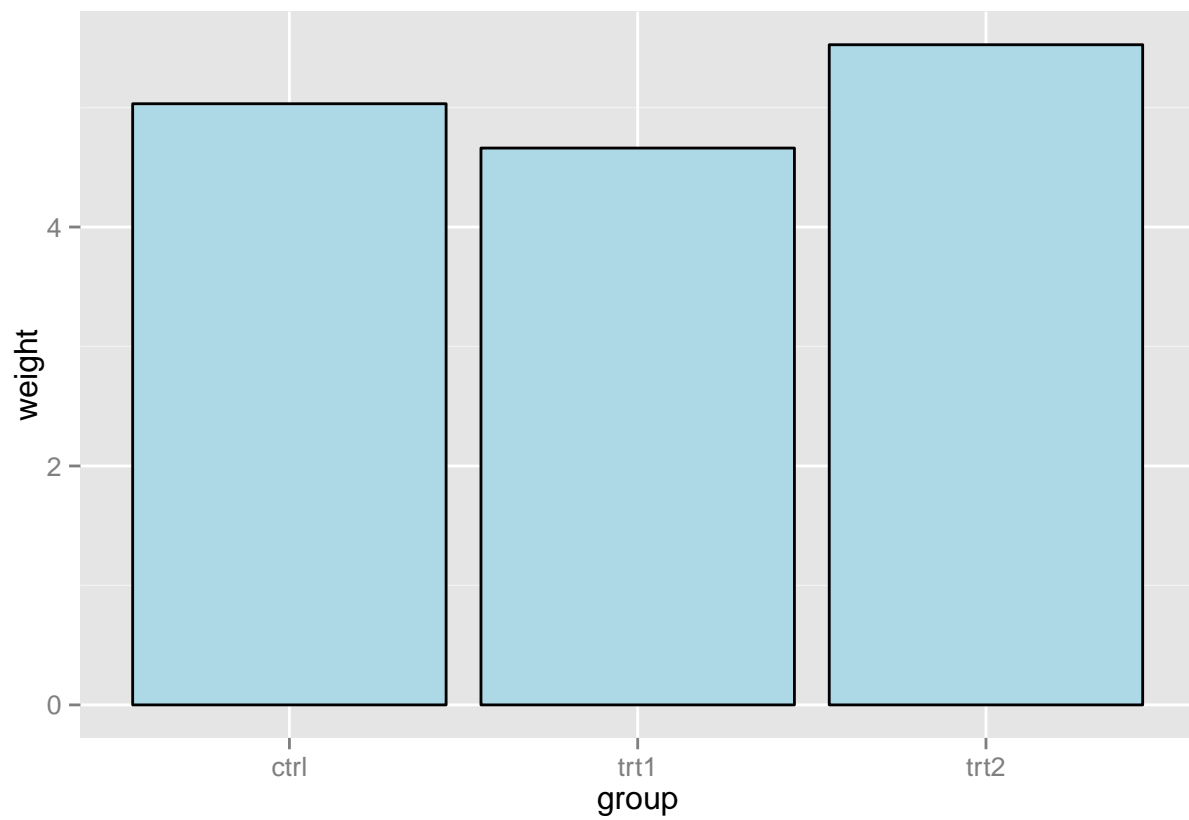
ggplot( BOD, aes( x = Time, y = demand)) + geom_bar( stat ="identity")
```



```
ggplot( BOD, aes( x = factor( Time), y = demand)) + geom_bar( stat ="identity")
```

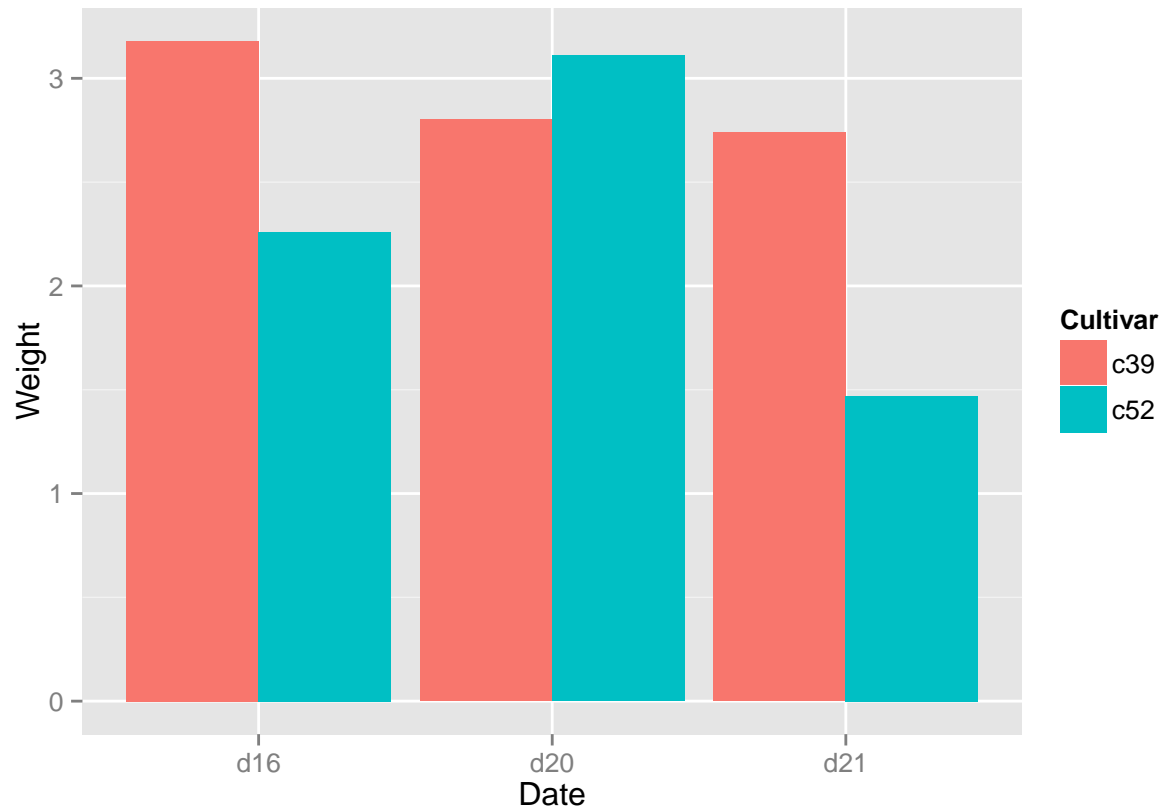


```
ggplot( pg_mean, aes( x = group, y = weight)) + geom_bar( stat ="identity", fill ="lightblue", colour =
```

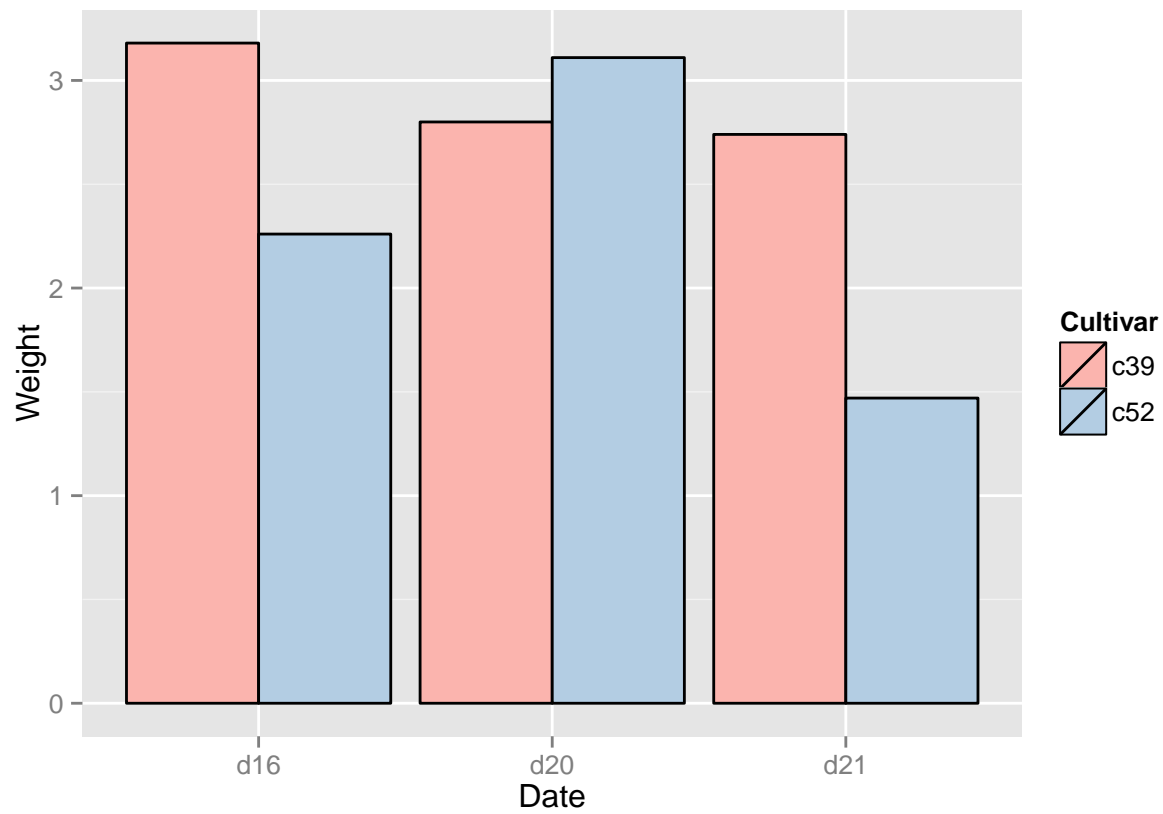


2. Grouping Bars Together

```
ggplot( cabbage_exp, aes( x = Date, y = Weight, fill = Cultivar)) +  
  geom_bar( position = "dodge", stat = "identity")
```

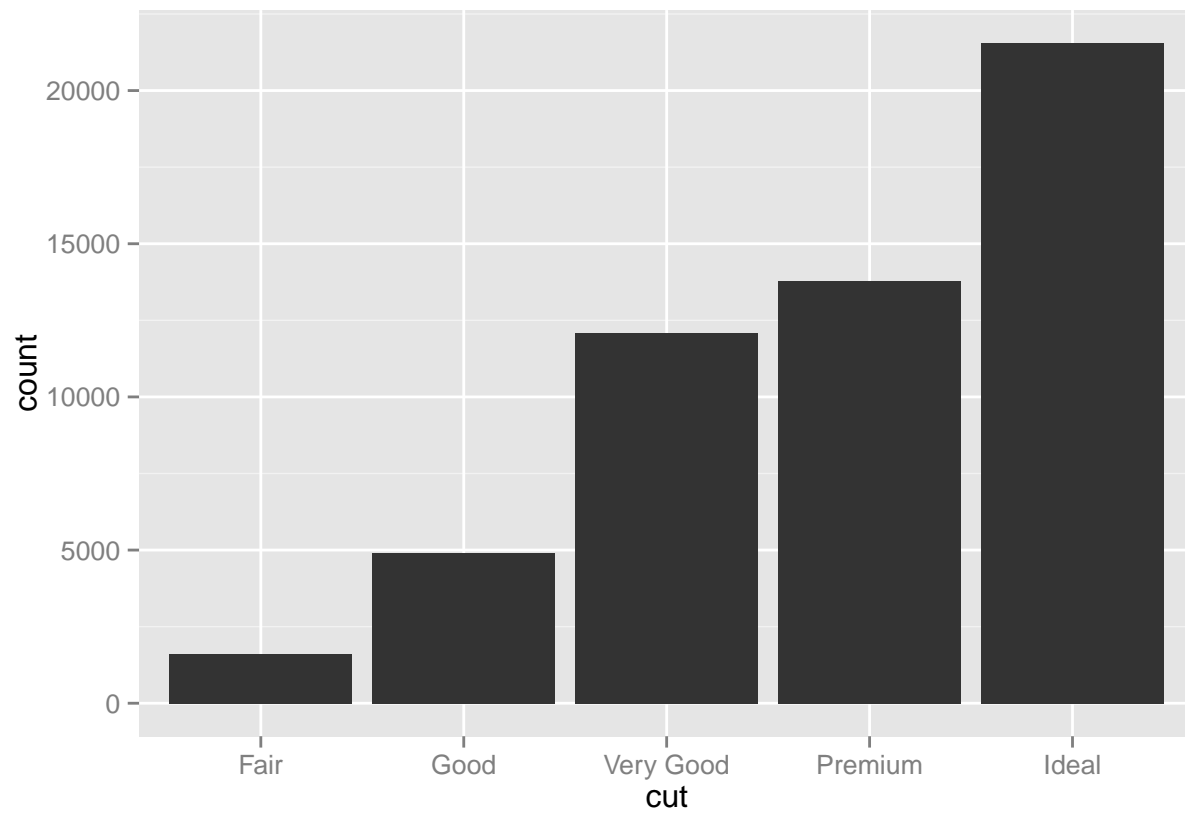


```
ggplot( cabbage_exp, aes( x = Date, y = Weight, fill = Cultivar)) +  
  geom_bar( position = "dodge", colour = "black", stat = "identity") +  
  scale_fill_brewer( palette = "Pastell1")
```



3. Making a Bar Graph of Counts

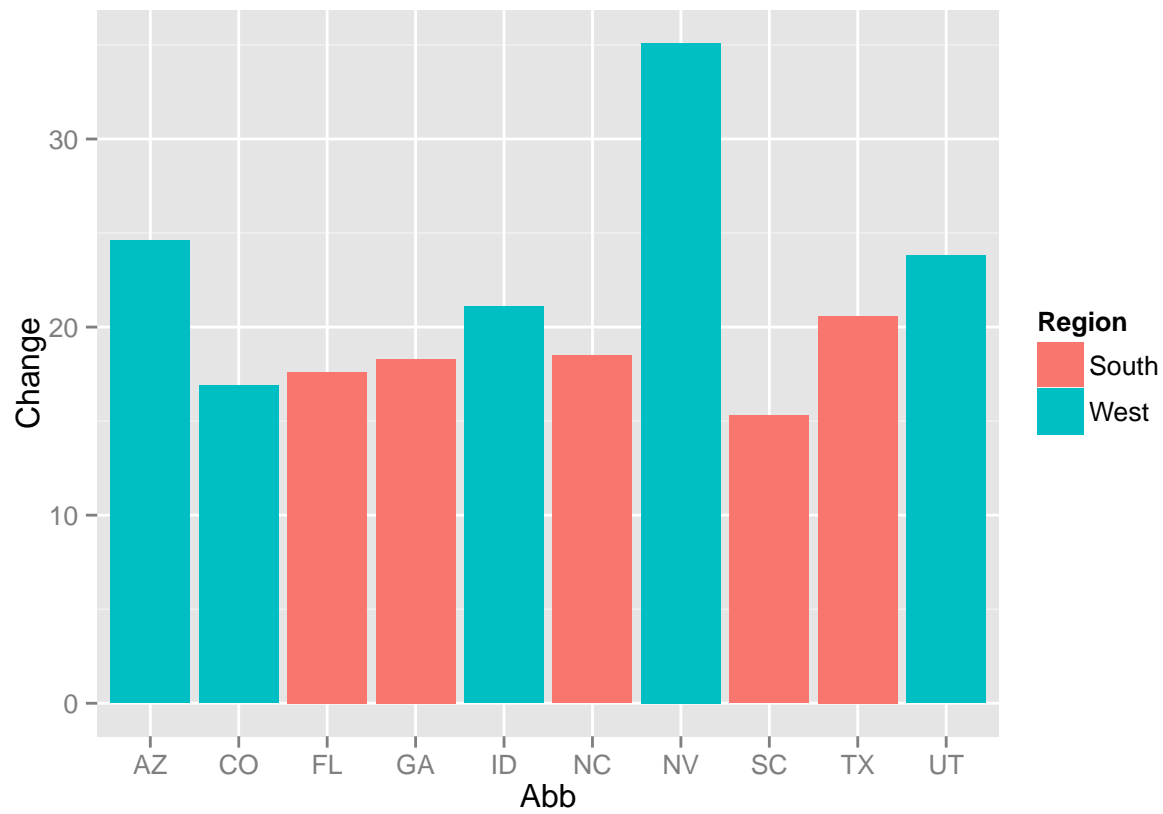
```
ggplot( diamonds, aes( x = cut)) + geom_bar()
```



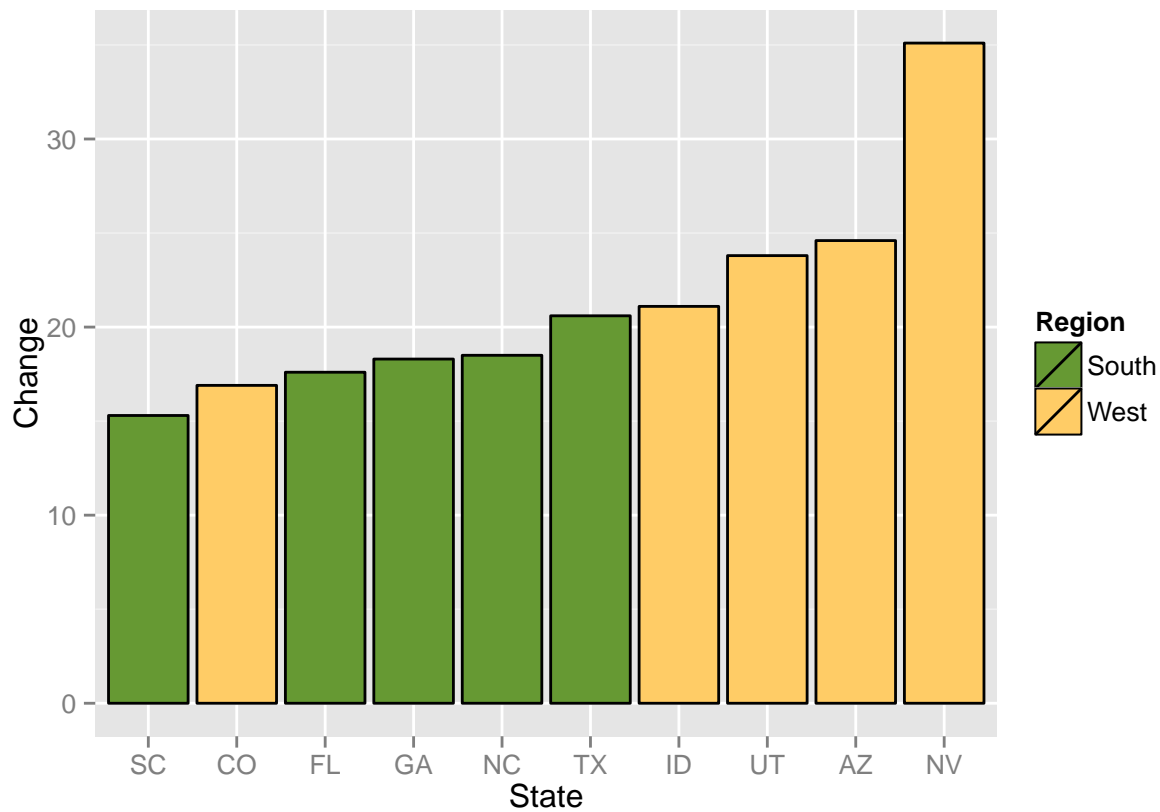
4. Using Colors in a Bar Graph

```
upc <- subset( uspopchange, rank( Change) > 40)

ggplot( upc, aes( x = Abb, y = Change, fill = Region)) + geom_bar( stat = "identity")
```



```
ggplot( upc, aes( x = reorder( Abb, Change), y = Change, fill = Region)) +  
  geom_bar( stat = "identity", colour = "black") +  
  scale_fill_manual( values = c("#669933", "#FFCC66")) + xlab("State")
```

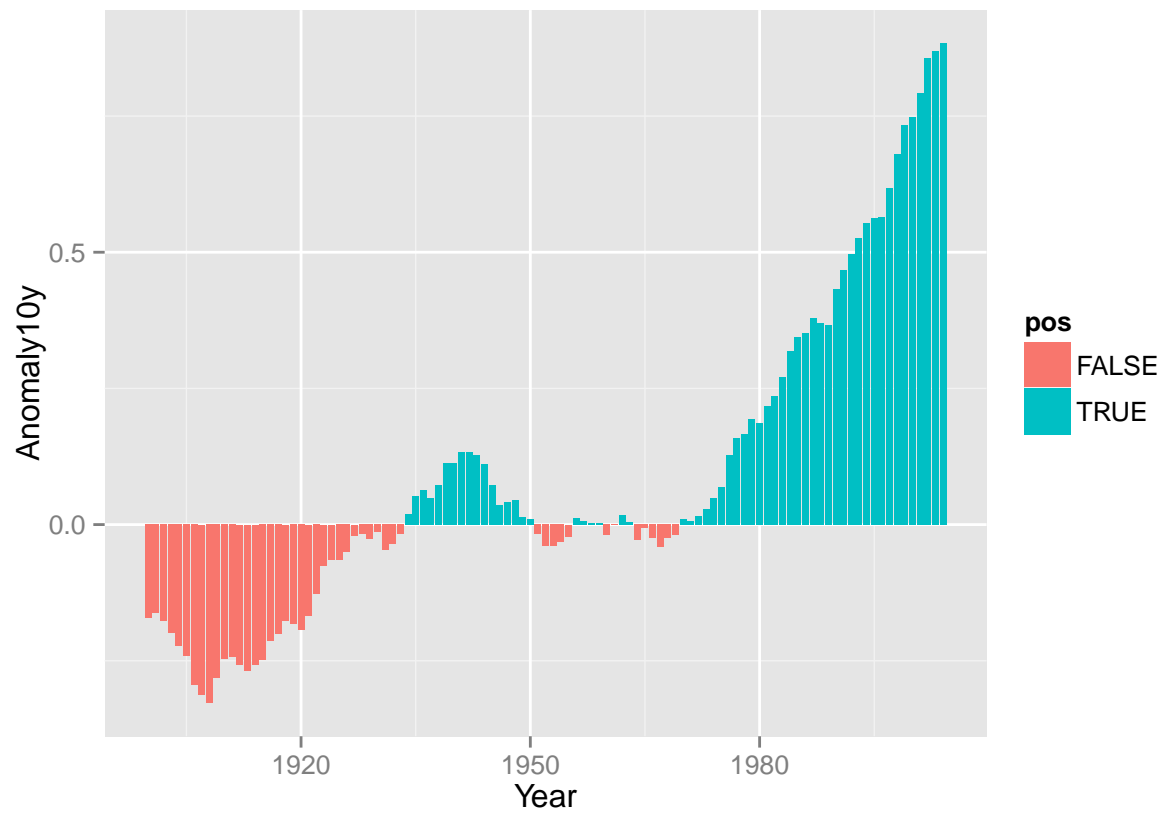


5. Coloring Negative and Positive Bars Differently

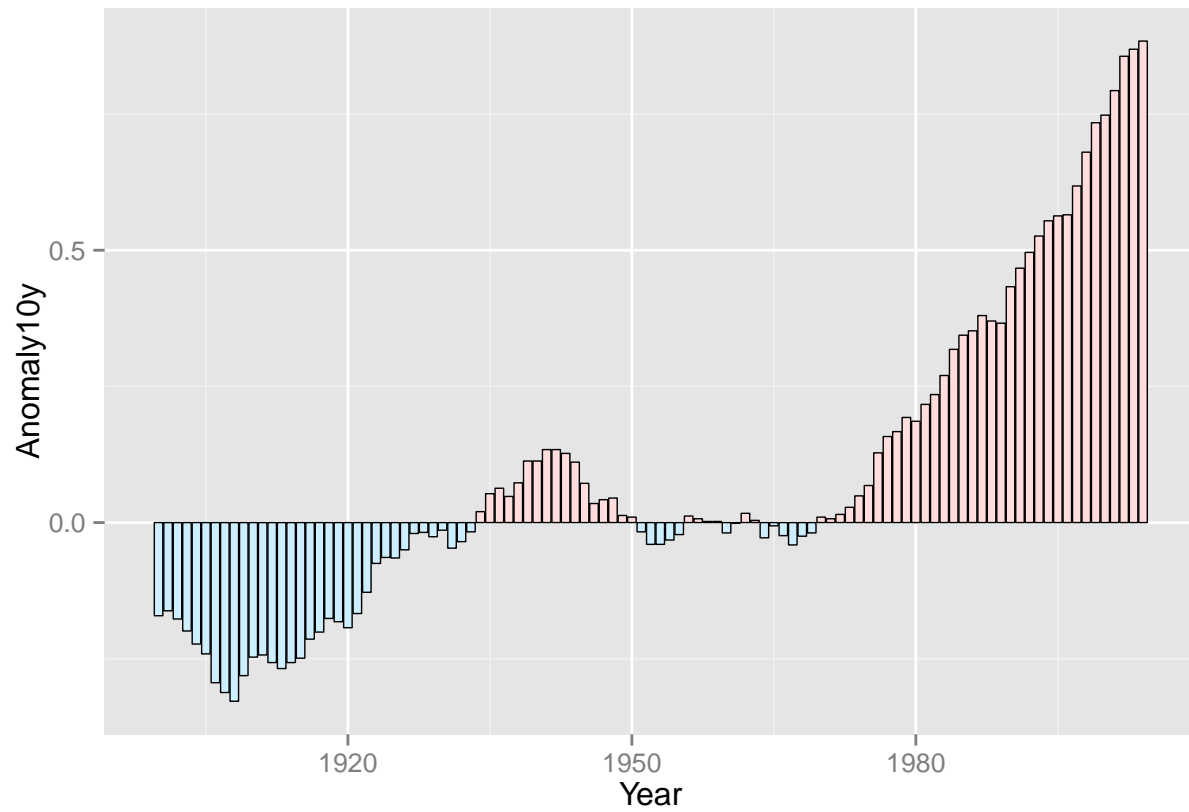
```
csub <- subset( climate, Source == "Berkeley" & Year >= 1900)
csub$pos <- csub$Anomaly10y >= 0
head(csub)
```

```
##      Source Year Anomaly1y Anomaly5y Anomaly10y Unc10y  pos
## 101 Berkeley 1900      NA      NA    -0.171  0.108 FALSE
## 102 Berkeley 1901      NA      NA    -0.162  0.109 FALSE
## 103 Berkeley 1902      NA      NA    -0.177  0.108 FALSE
## 104 Berkeley 1903      NA      NA    -0.199  0.104 FALSE
## 105 Berkeley 1904      NA      NA    -0.223  0.105 FALSE
## 106 Berkeley 1905      NA      NA    -0.241  0.107 FALSE
```

```
ggplot( csub, aes( x = Year, y = Anomaly10y, fill = pos)) +
  geom_bar( stat = "identity", position = "identity")
```



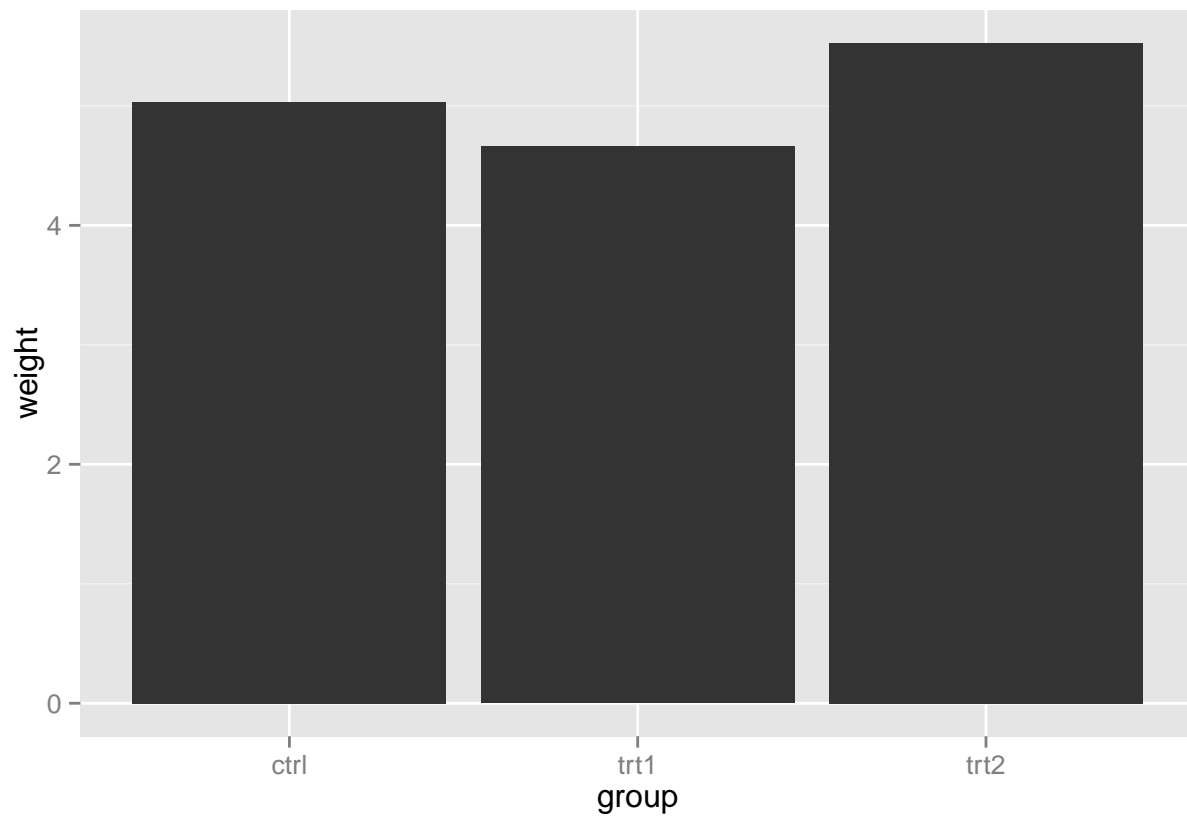
```
ggplot( csub, aes( x = Year, y = Anomaly10y, fill = pos)) +
  geom_bar( stat = "identity", position = "identity", colour = "black", size = 0.25) +
  scale_fill_manual( values = c("#CCEEFF", "#FFDDDD"), guide = FALSE)
```

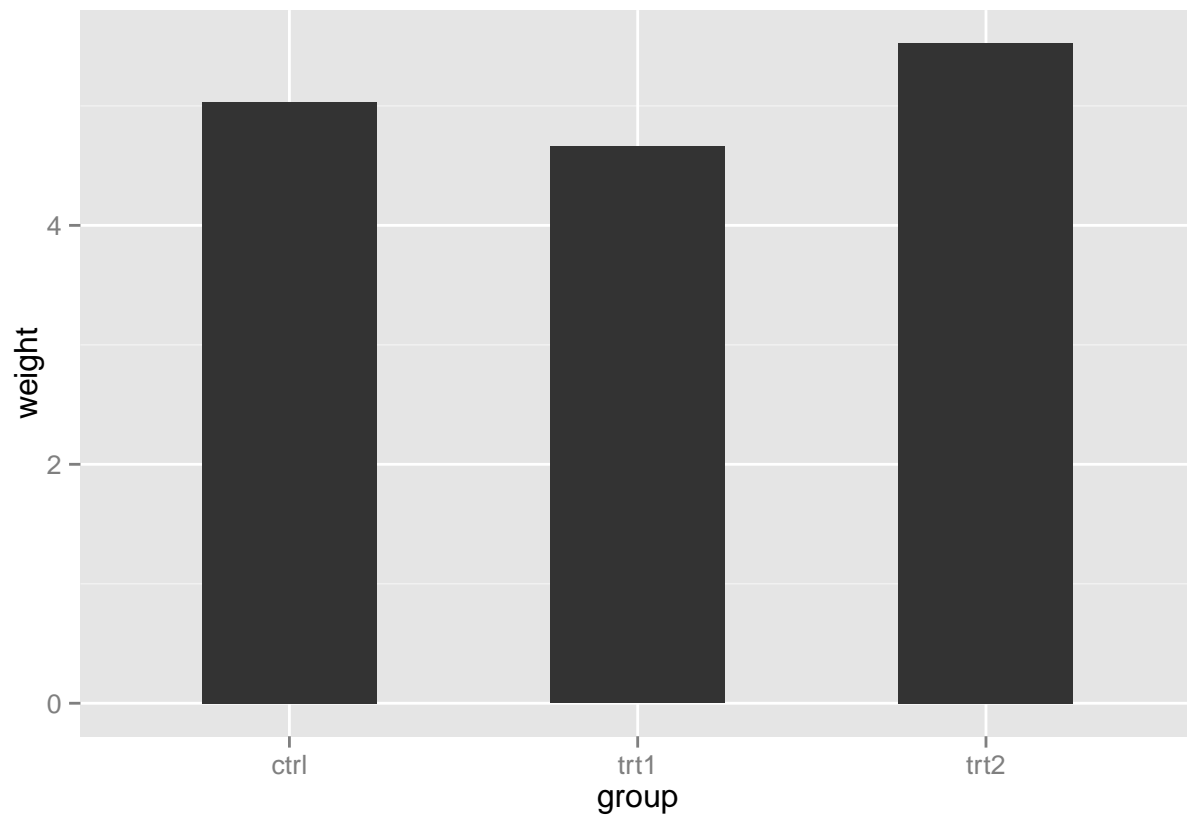
6. Adjusting Bar Width and Spacing

To make the bars narrower or wider, set **width** in `geom_bar()`. The default value is 0.9.

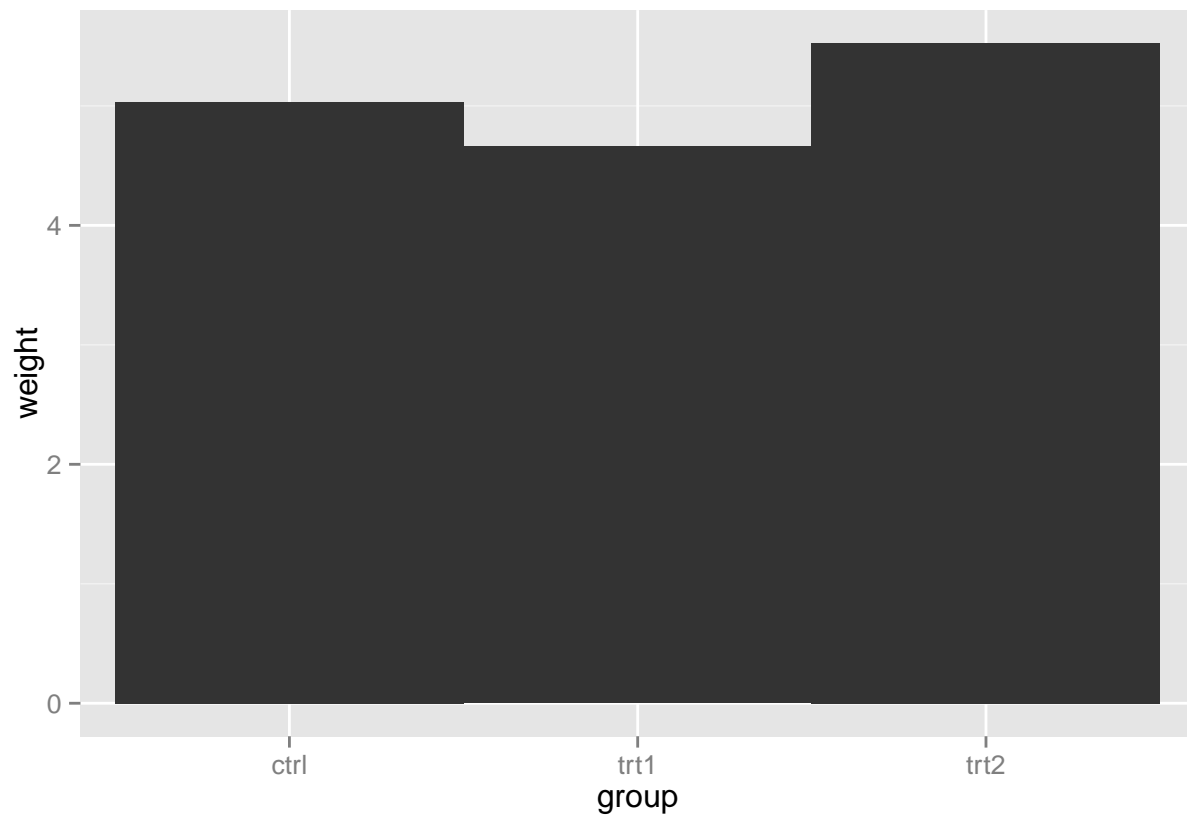
```
## default  
ggplot( pg_mean, aes( x = group, y = weight)) + geom_bar( stat ="identity")
```



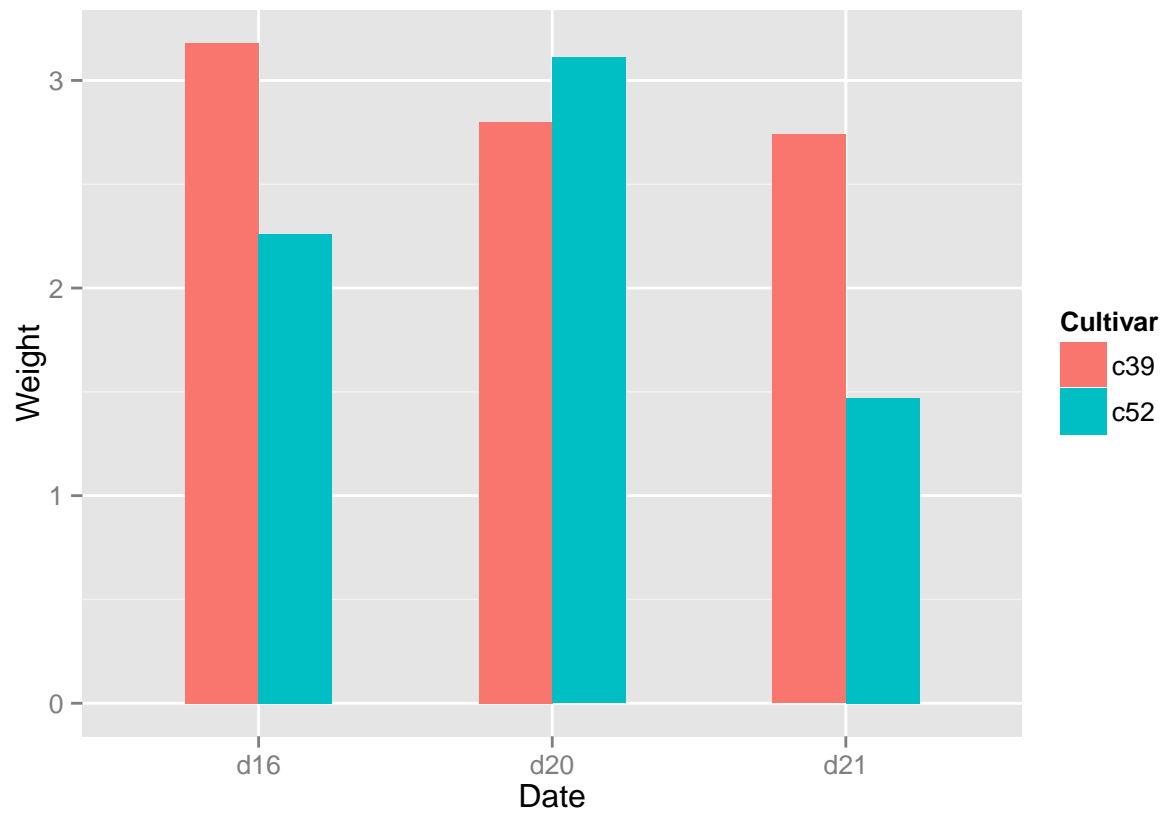
```
## for narrower bars  
ggplot( pg_mean, aes( x = group, y = weight)) + geom_bar( stat = "identity", width = 0.5)
```



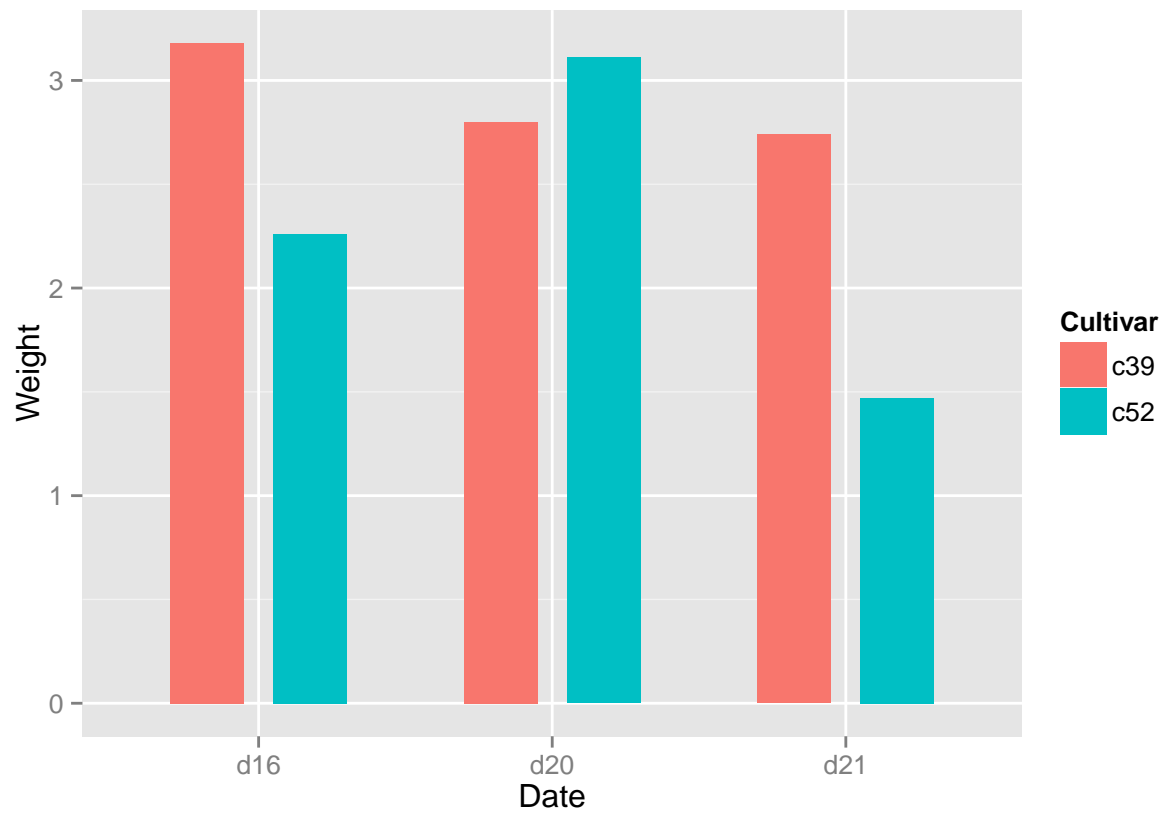
```
## for wider bars  
ggplot( pg_mean, aes( x = group, y = weight)) + geom_bar( stat ="identity", width = 1)
```



```
## To add space between bars within a group, make width smaller and set the value for position_dodge
ggplot( cabbage_exp, aes( x = Date, y = Weight, fill = Cultivar)) +
  geom_bar( stat = "identity", width = 0.5, position = "dodge")
```

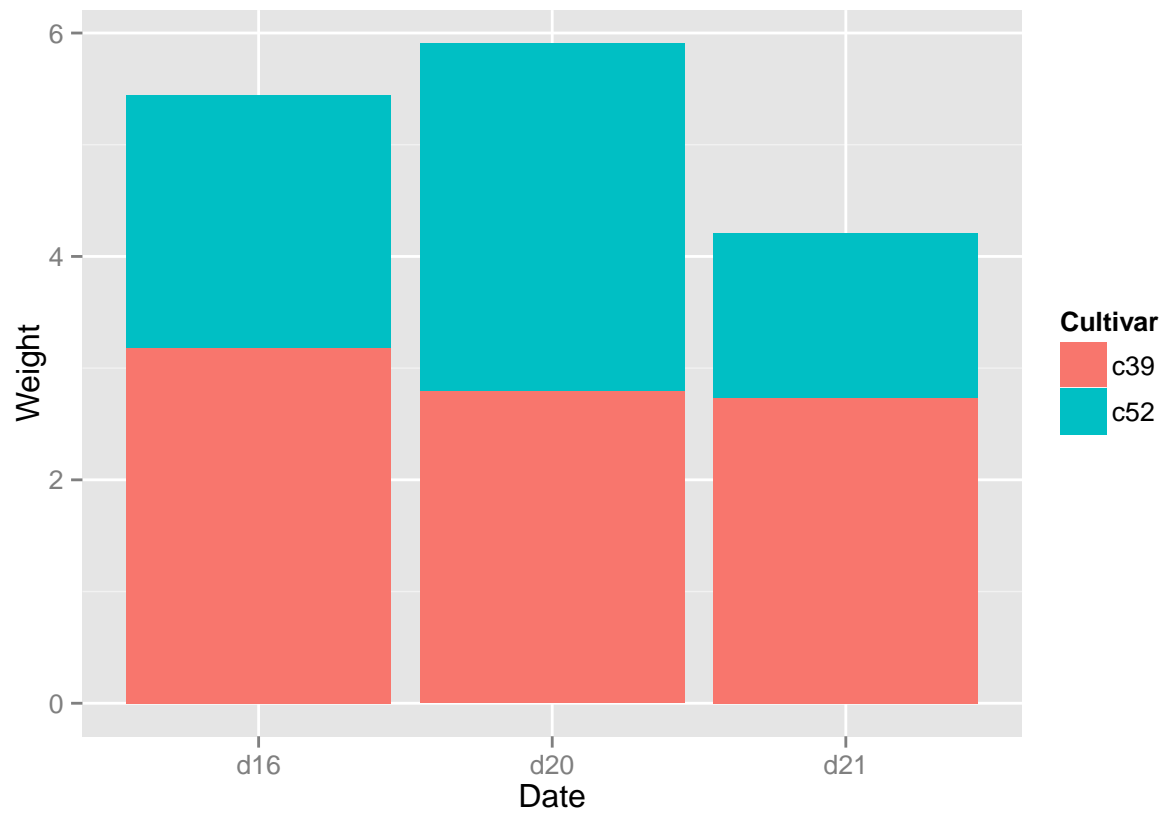


```
## And with some space between the bars:  
ggplot( cabbage_exp, aes( x = Date, y = Weight, fill = Cultivar)) +  
  geom_bar( stat = "identity", width = 0.5, position = position_dodge( 0.7))
```

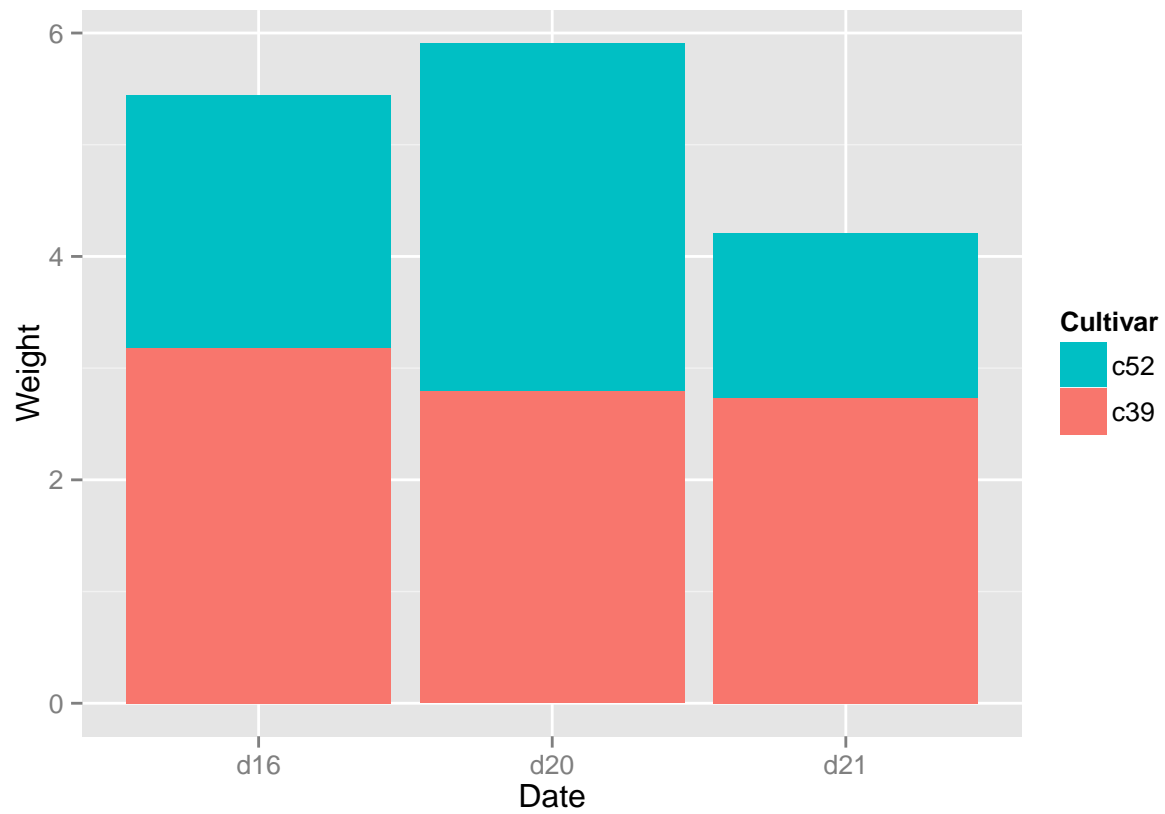


7. Making a Stacked Bar Graph

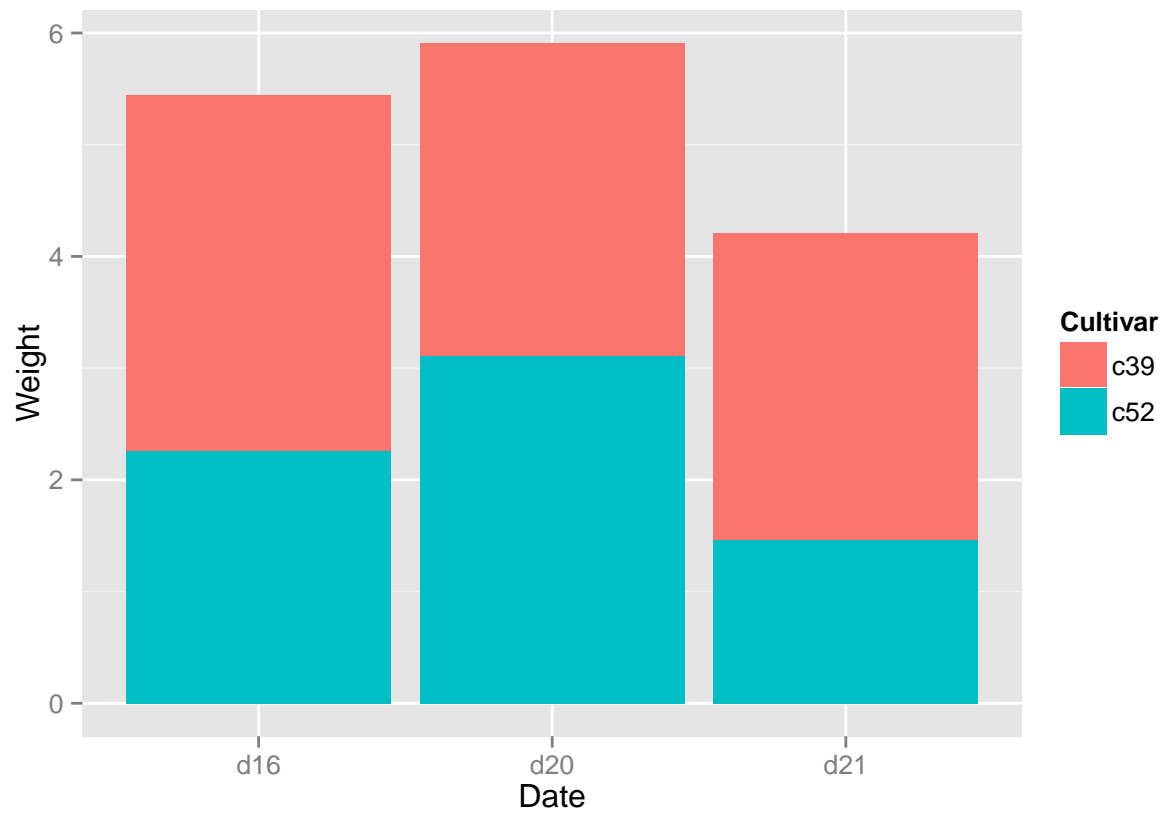
```
ggplot( cabbage_exp, aes( x = Date, y = Weight, fill = Cultivar)) +  
  geom_bar( stat ="identity")
```



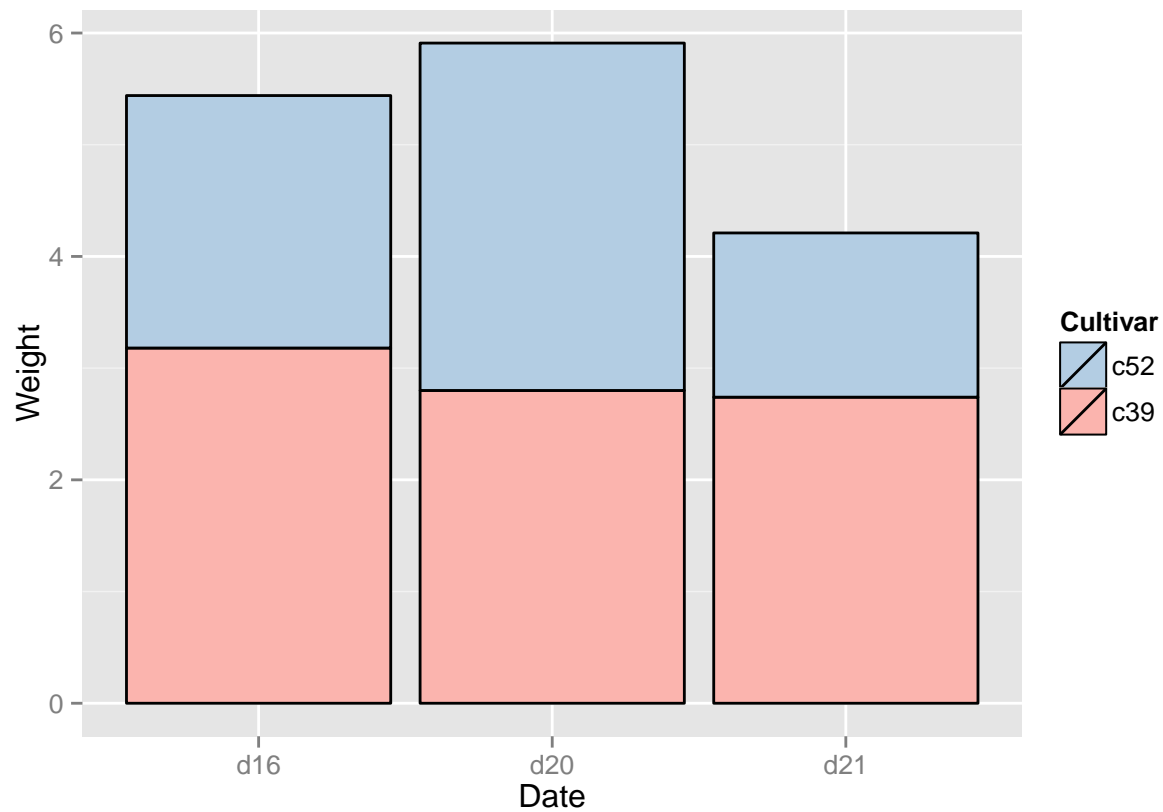
```
## reverse the order of items in the legend by using guides()
ggplot( cabbage_exp, aes( x = Date, y = Weight, fill = Cultivar)) +
  geom_bar( stat = "identity" ) +
  guides( fill = guide_legend( reverse = TRUE))
```



```
## reverse the stacking order
ggplot( cabbage_exp, aes( x = Date, y = Weight, fill = Cultivar, order = desc( Cultivar))) +
  geom_bar( stat = "identity")
```

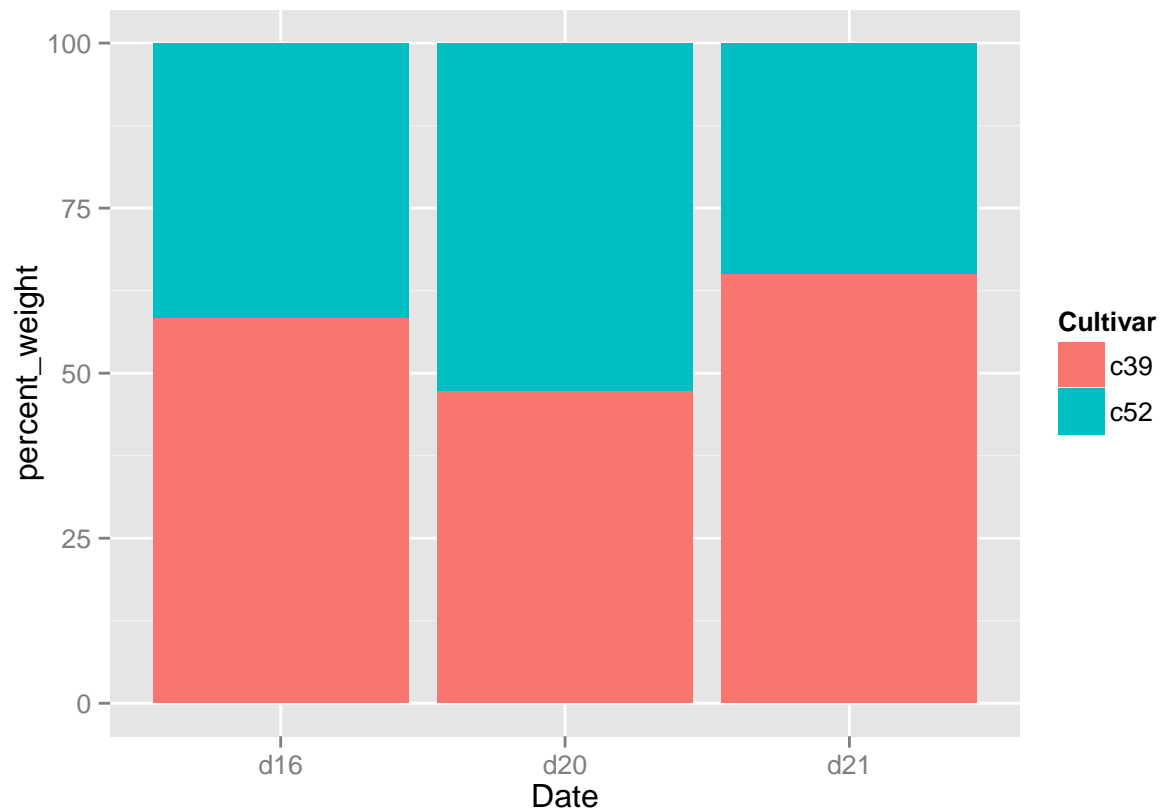
```
## scale_fill_brewer
ggplot( cabbage_exp, aes( x = Date, y = Weight, fill = Cultivar)) +
  geom_bar( stat = "identity", colour = "black") +
  guides( fill = guide_legend( reverse = TRUE)) +
  scale_fill_brewer( palette = "Pastel1")
```



8. Making a Proportional Stacked Bar Graph

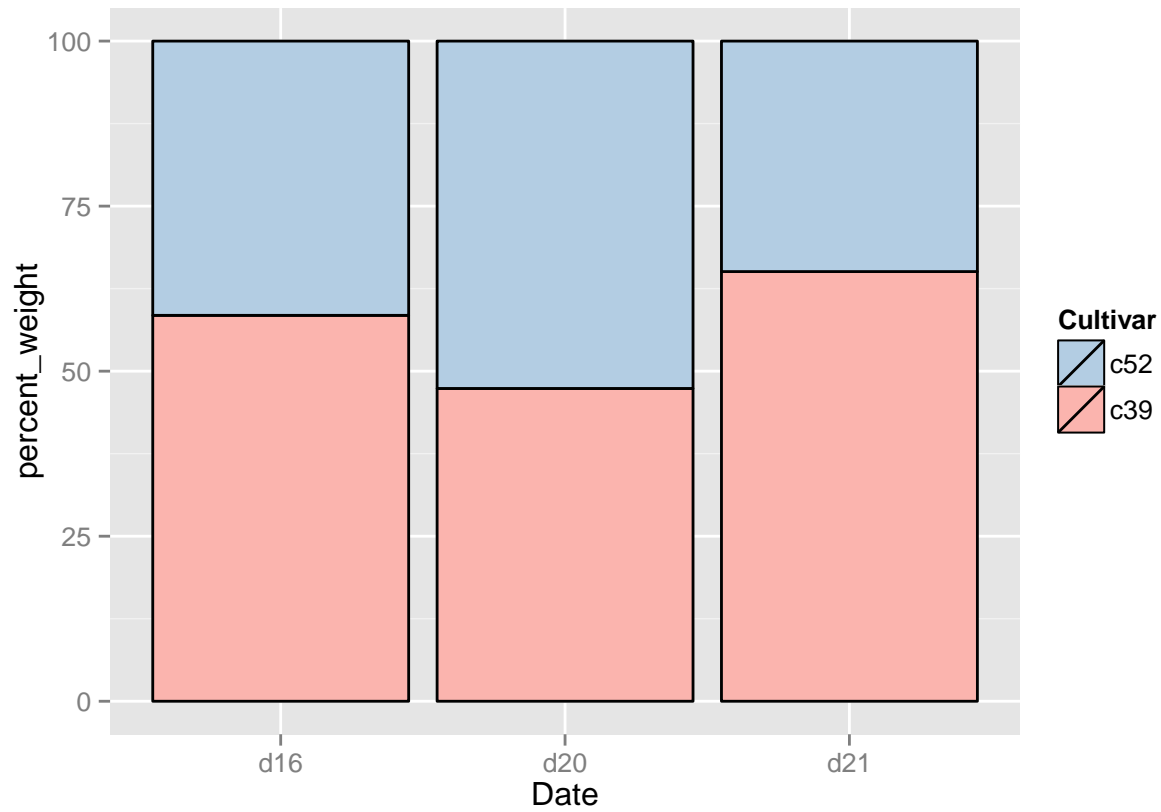
```
ce <- ddply( cabbage_exp, "Date", transform, percent_weight = Weight / sum( Weight) * 100)

ggplot( ce, aes( x = Date, y = percent_weight, fill = Cultivar)) +
  geom_bar( stat ="identity")
```



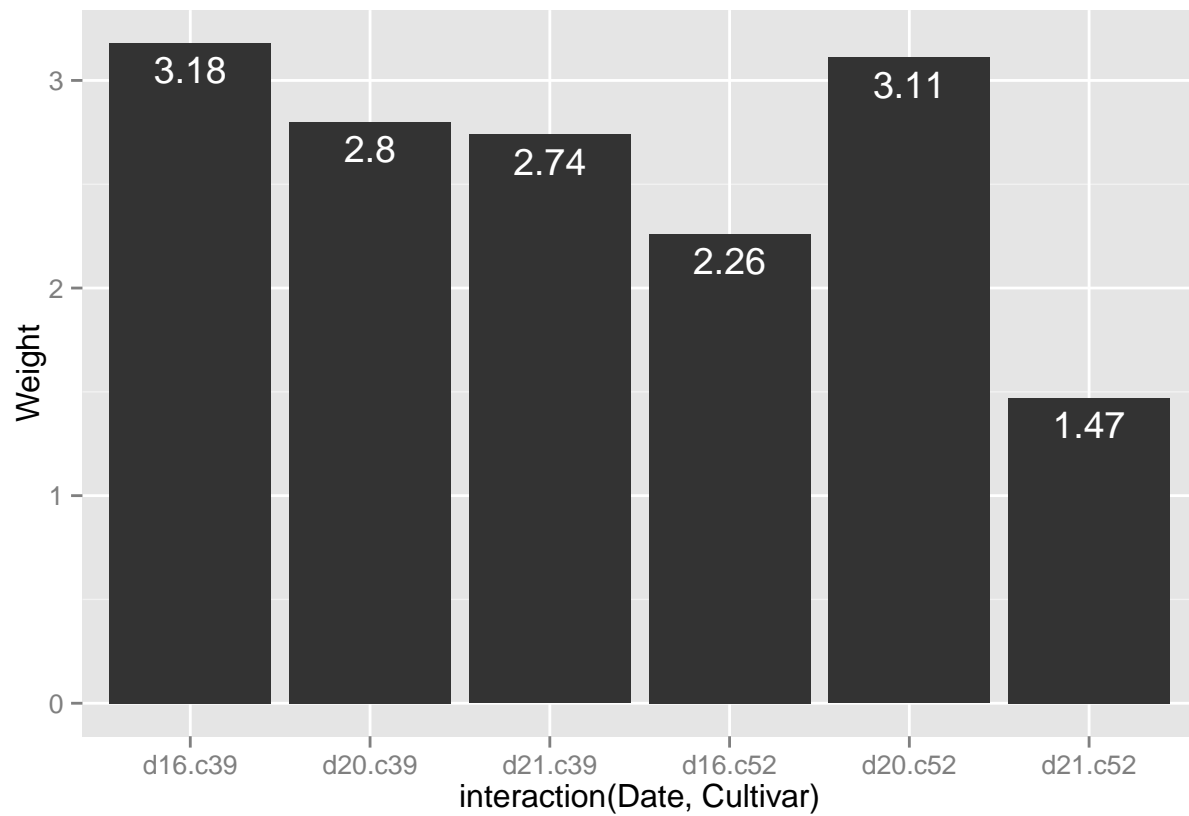
```
ce <- ddply( cabbage_exp, "Date", transform, percent_weight = Weight / sum( Weight) * 100)

ggplot( ce, aes( x = Date, y = percent_weight, fill = Cultivar)) +
  geom_bar( stat = "identity", colour = "black") +
  guides( fill = guide_legend( reverse = TRUE)) +
  scale_fill_brewer( palette = "Pastell1")
```

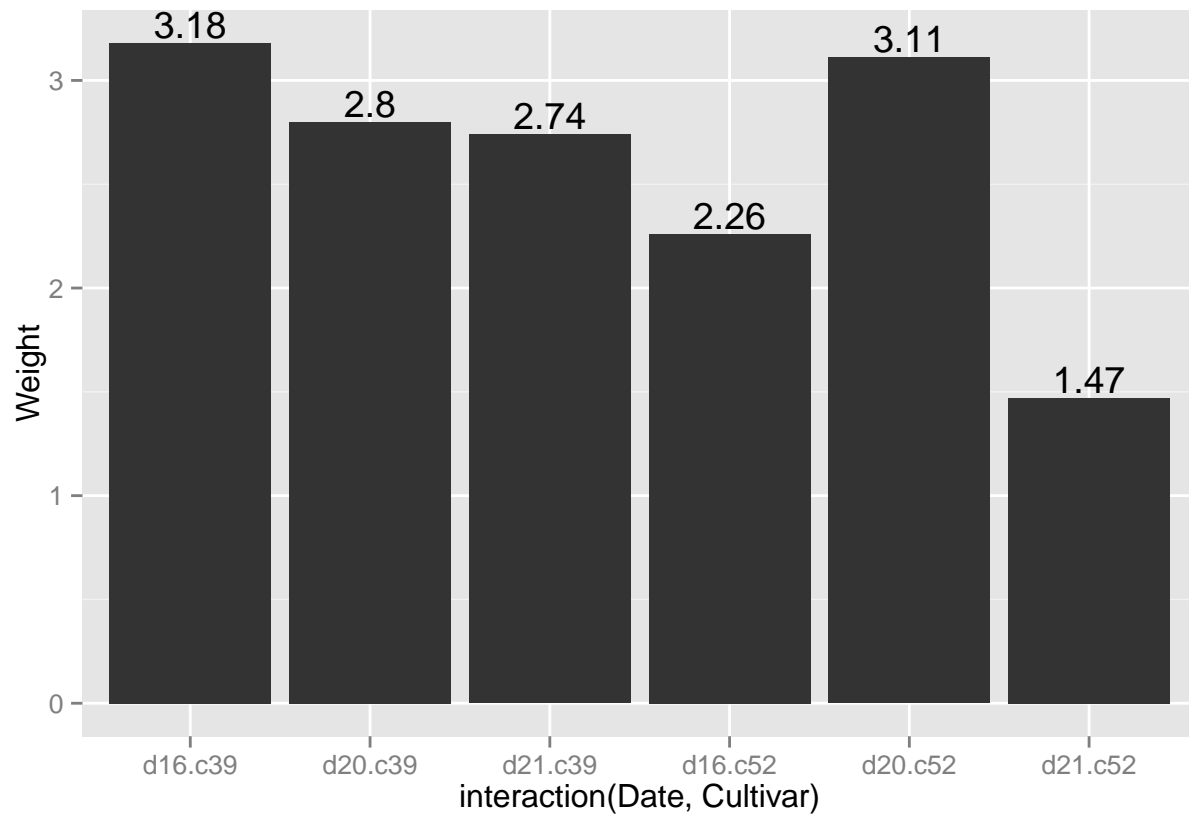


9. MAdding Labels to a Bar Graph

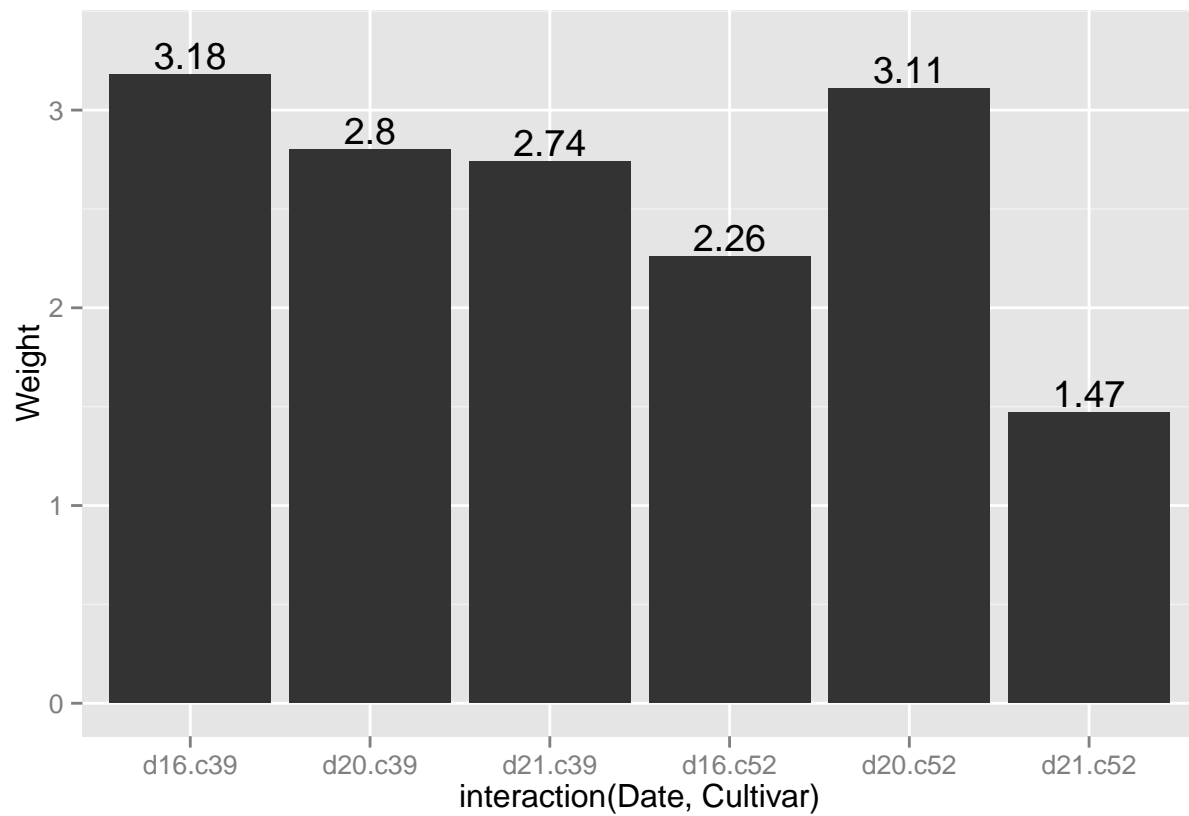
```
# Below the top  
ggplot( cabbage_exp, aes( x = interaction( Date, Cultivar), y = Weight)) +  
  geom_bar( stat ="identity") +  
  geom_text( aes( label = Weight), vjust = 1.5, colour ="white")
```



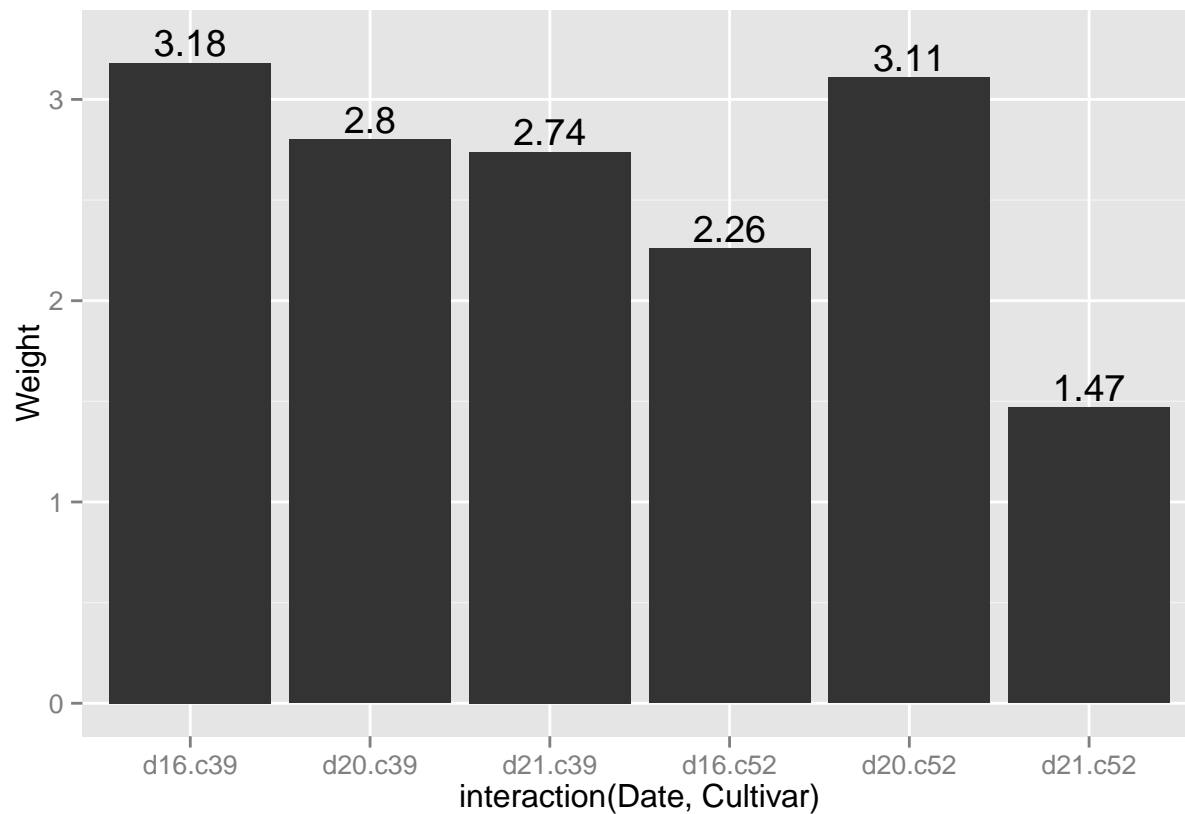
```
# Above the top  
ggplot( cabbage_exp, aes( x = interaction( Date, Cultivar), y = Weight)) +  
  geom_bar( stat = "identity") +  
  geom_text( aes( label = Weight), vjust = -0.2)
```



```
# Adjust y limits to be a little higher  
ggplot( cabbage_exp, aes( x = interaction( Date, Cultivar), y = Weight)) +  
  geom_bar( stat = "identity") + geom_text( aes( label = Weight), vjust = -0.2) +  
  ylim( 0, max( cabbage_exp$Weight) * 1.05)
```



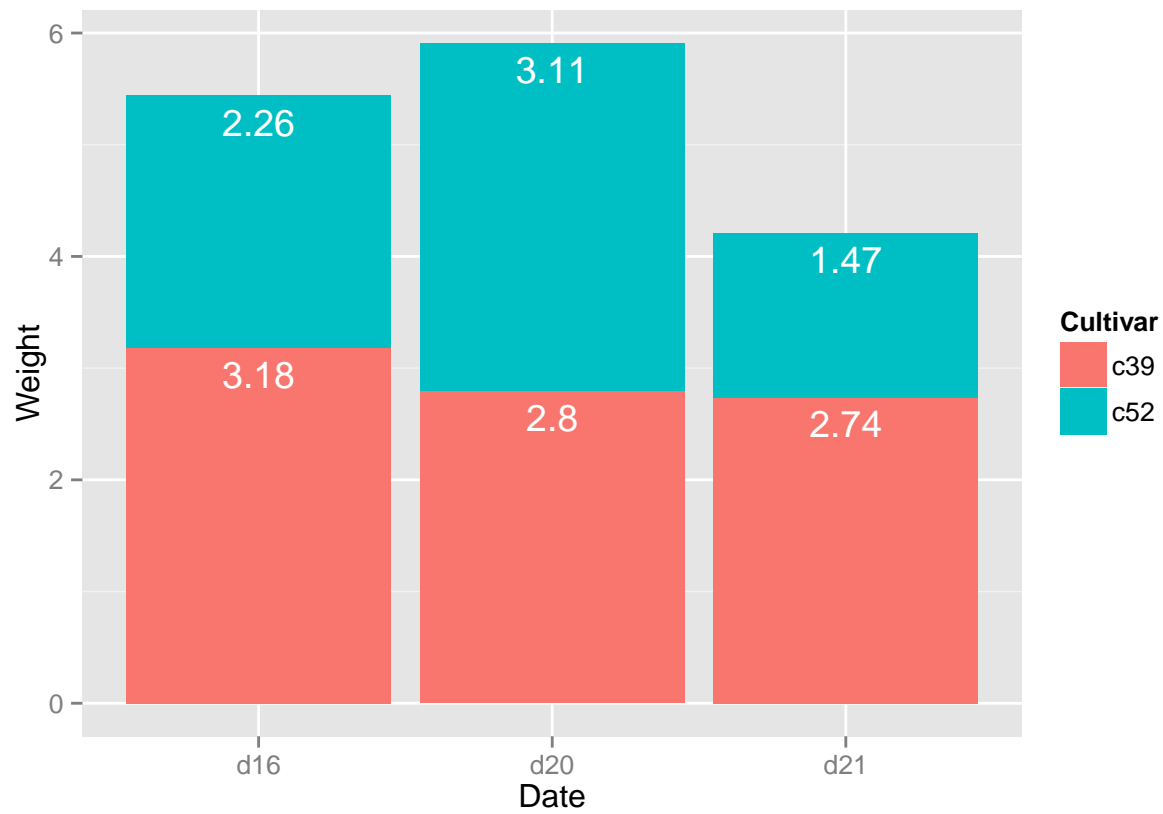
```
# Map y positions slightly above bar top - y range of plot will auto-adjust  
ggplot( cabbage_exp, aes( x = interaction( Date, Cultivar), y = Weight)) +  
  geom_bar( stat = "identity") + geom_text( aes( y = Weight + 0.1, label = Weight))
```



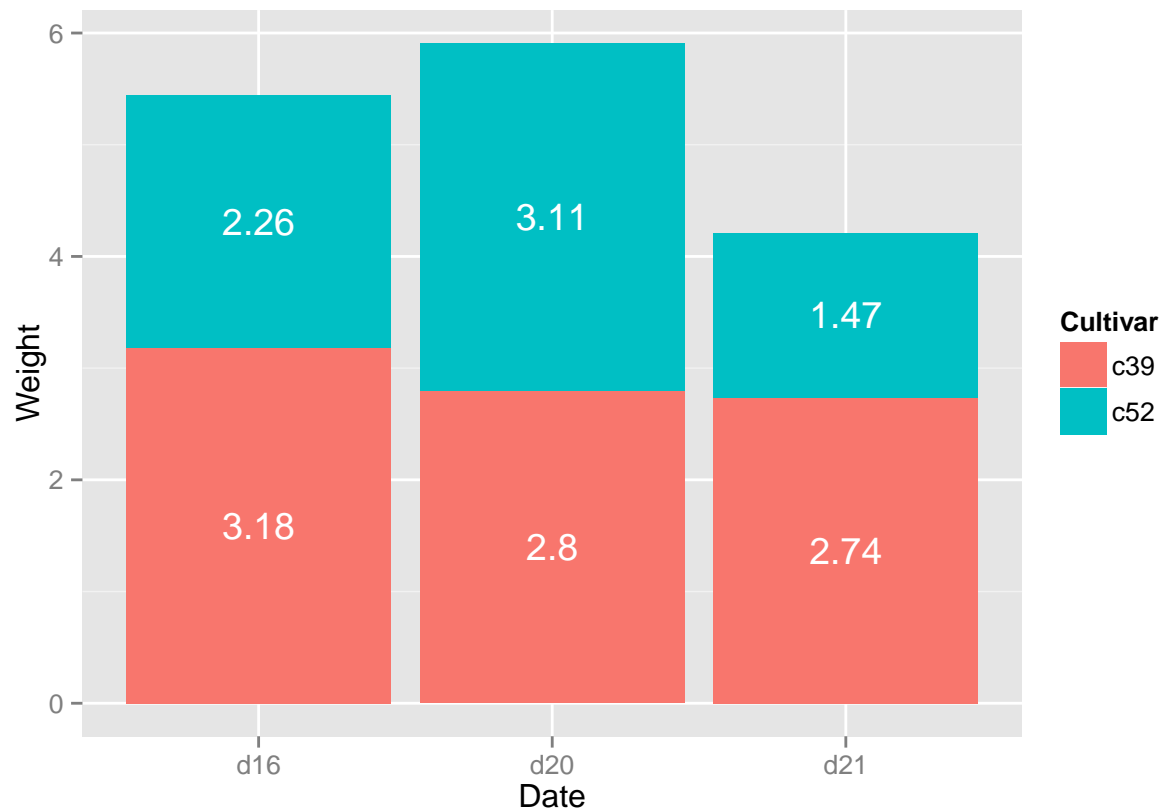
```
# Sort by the day and sex columns
ce <- arrange( cabbage_exp, Date, Cultivar)

## Get the cumulative sum
ce <- ddply( ce, "Date", transform, label_y = cumsum( Weight))

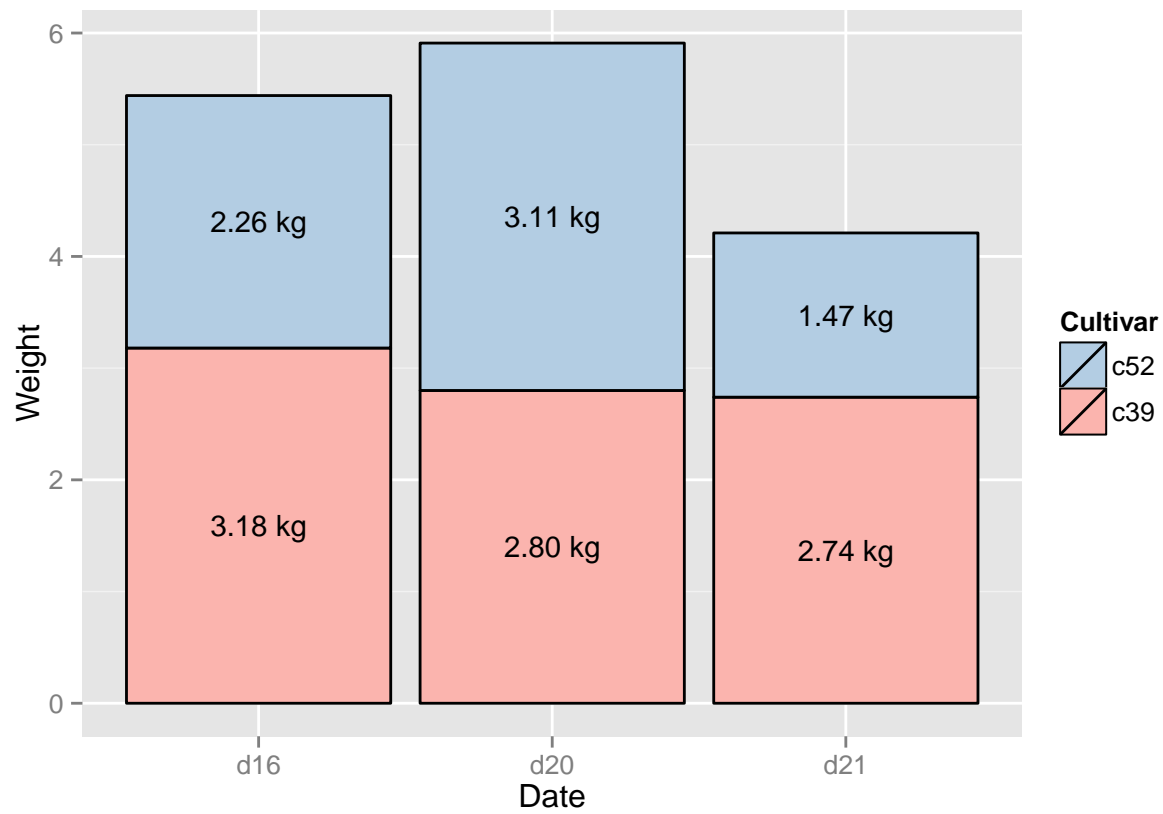
ggplot( ce, aes( x = Date, y = Weight, fill = Cultivar)) +
  geom_bar( stat = "identity") +
  geom_text( aes( y = label_y, label = Weight), vjust = 1.5, colour = "white")
```

```
# Calculate y position, placing it in the middle
ce <- ddply( ce, "Date", transform, label_y = cumsum( Weight)-0.5* Weight)
ggplot( ce, aes( x = Date, y = Weight, fill = Cultivar)) +
  geom_bar( stat ="identity") +
  geom_text( aes( y = label_y, label = Weight), colour ="white")
```

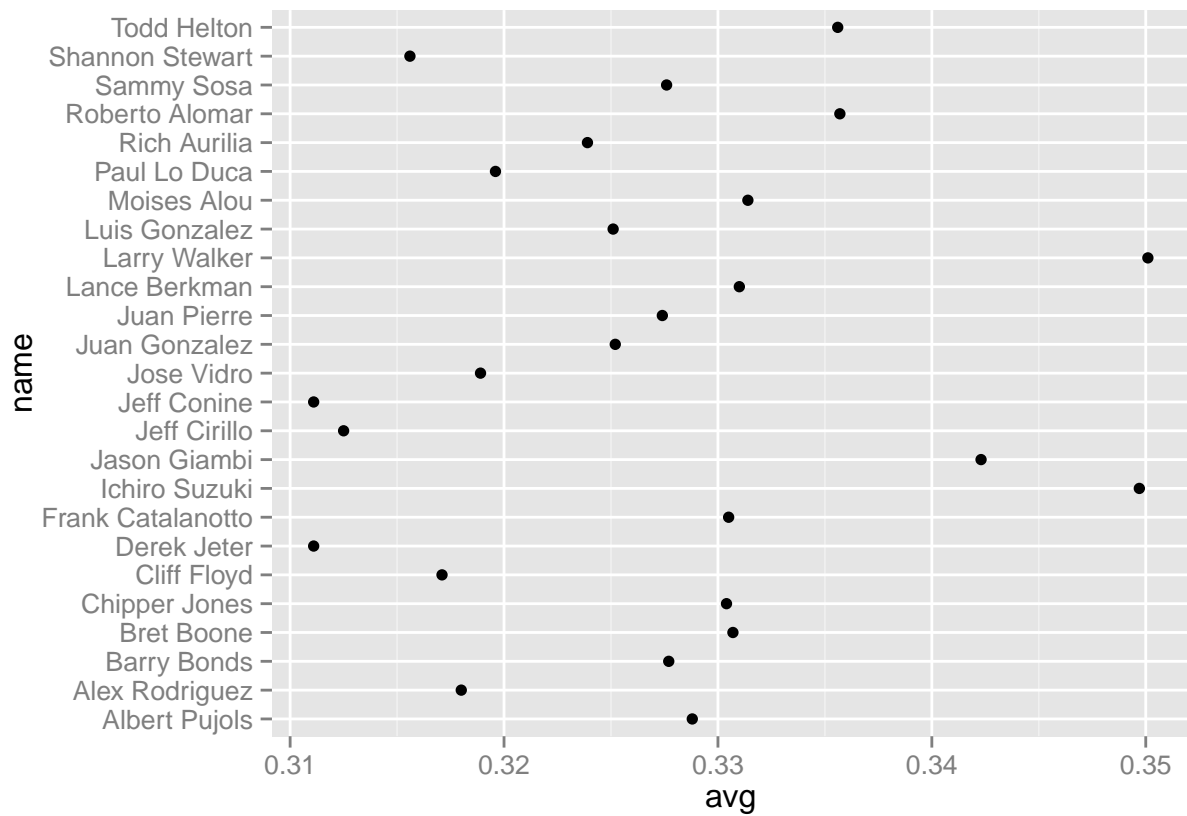


```
## add labels in the middle with a smaller font using size, add a "kg"
ggplot( ce, aes( x = Date, y = Weight, fill = Cultivar)) +
  geom_bar( stat = "identity", colour = "black") +
  geom_text( aes( y = label_y, label = paste( format( Weight, nsmall = 2), "kg")), size = 4) +
  guides( fill = guide_legend( reverse = TRUE)) + scale_fill_brewer( palette = "Pastel1")
```

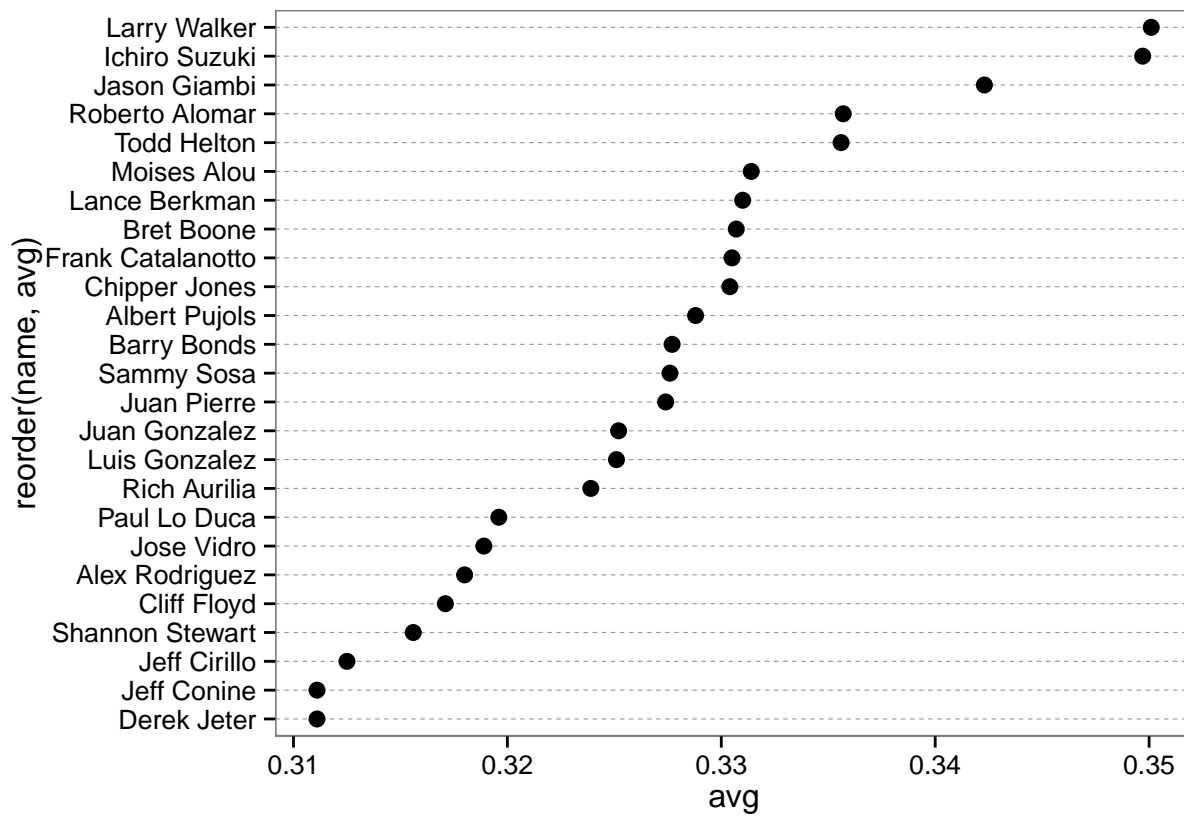


Making a Cleveland Dot Plot

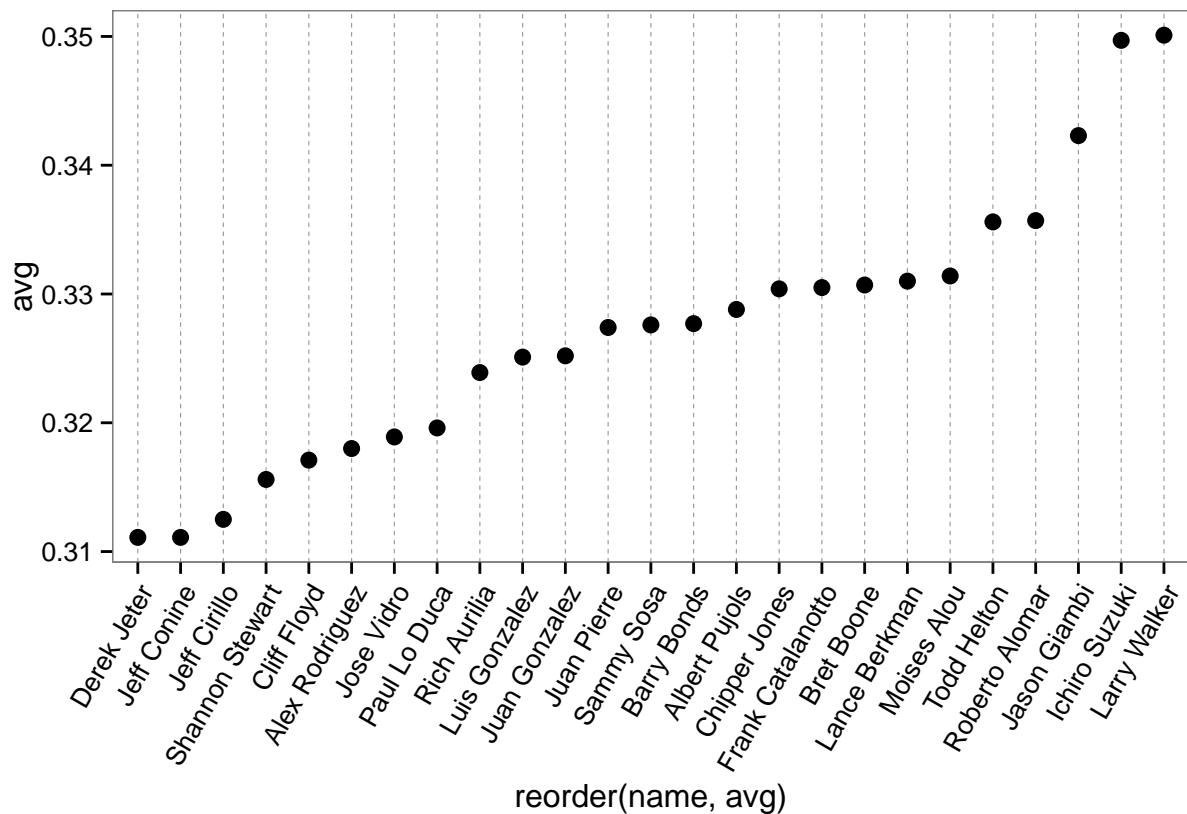
```
# Take the top 25 from the tophitters data set  
tophit <- tophitters2001[ 1: 25, ]  
ggplot( tophit, aes( x = avg, y = name)) + geom_point()
```



```
## reorder( name, avg),
ggplot( tophit, aes( x = avg, y = reorder( name, avg))) +
  geom_point( size = 3) +
  theme_bw() +
  theme( panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_line( colour = "grey60", linetype = "dashed"))
```



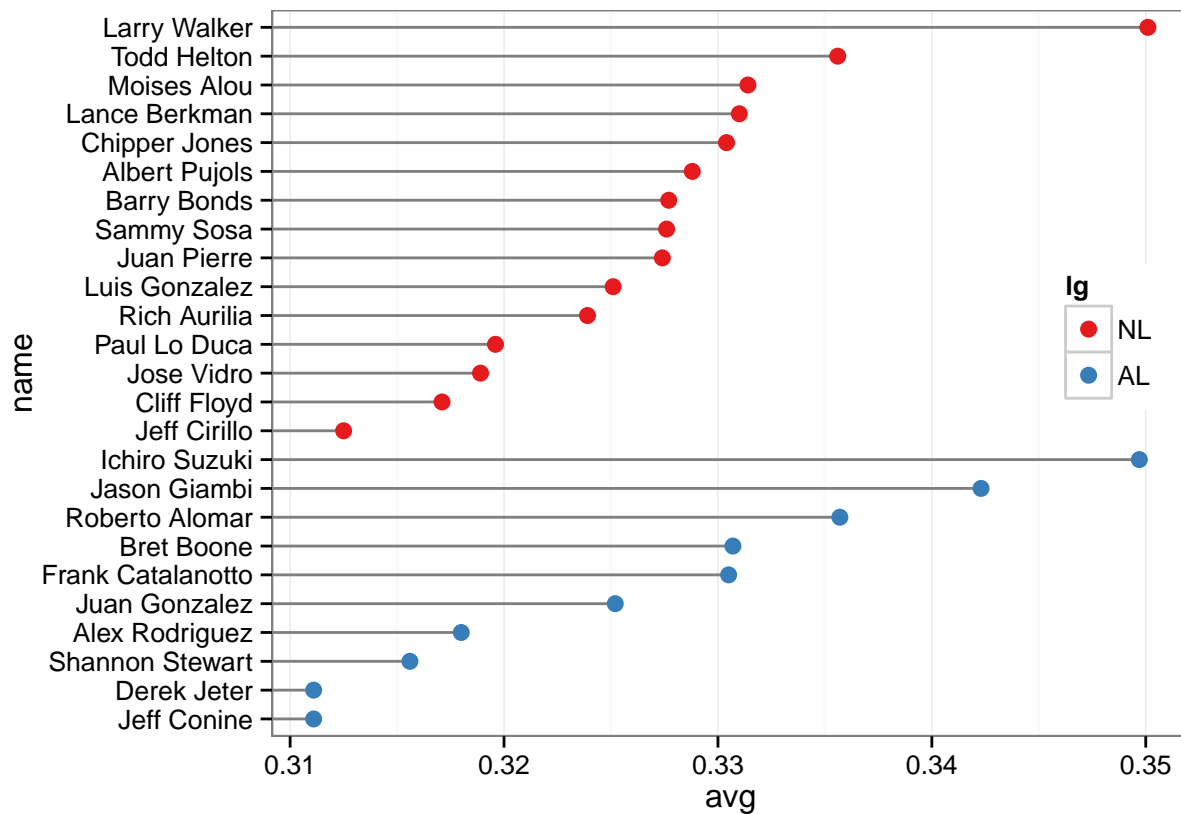
```
## swap the axes
ggplot( tophit, aes( x = reorder( name, avg), y = avg)) +
  geom_point( size = 3) +
  theme_bw() +
  theme( axis.text.x = element_text( angle = 60, hjust = 1),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank(),
        panel.grid.major.x = element_line( colour ="grey60", linetype ="dashed"))
```



```
# Get the names, sorted first by lg, then by avg
nameorder <- tophit$name[ order( tophit$lg, tophit$avg)]

# Turn name into a factor, with levels in the order of nameorder
tophit$name <- factor( tophit$name, levels = nameorder)

ggplot( tophit, aes( x = avg, y = name)) +
  geom_segment( aes( yend = name), xend = 0, colour ="grey50") +
  geom_point( size = 3, aes( colour = lg)) + scale_colour_brewer( palette ="Set1", limits = c("NL","AL"))
  theme_bw() +
  theme( panel.grid.major.y = element_blank(), # No horizontal grid lines
        legend.position = c( 1, 0.55), # Put legend inside plot area
        legend.justification = c( 1, 0.5))
```



```
## Another way to separate the two groups is to use facets,
ggplot( tophit, aes( x = avg, y = name)) +
  geom_segment( aes( yend = name), xend = 0, colour = "grey50") +
  geom_point( size = 3, aes( colour = lg)) +
  scale_colour_brewer( palette = "Set1", limits = c("NL","AL"), guide = FALSE) +
  theme_bw() +
  theme( panel.grid.major.y = element_blank()) +
  facet_grid( lg ~ ., scales = "free_y", space = "free_y")
```

