

12__Using__Colors

Gino Tesei

December 14, 2015

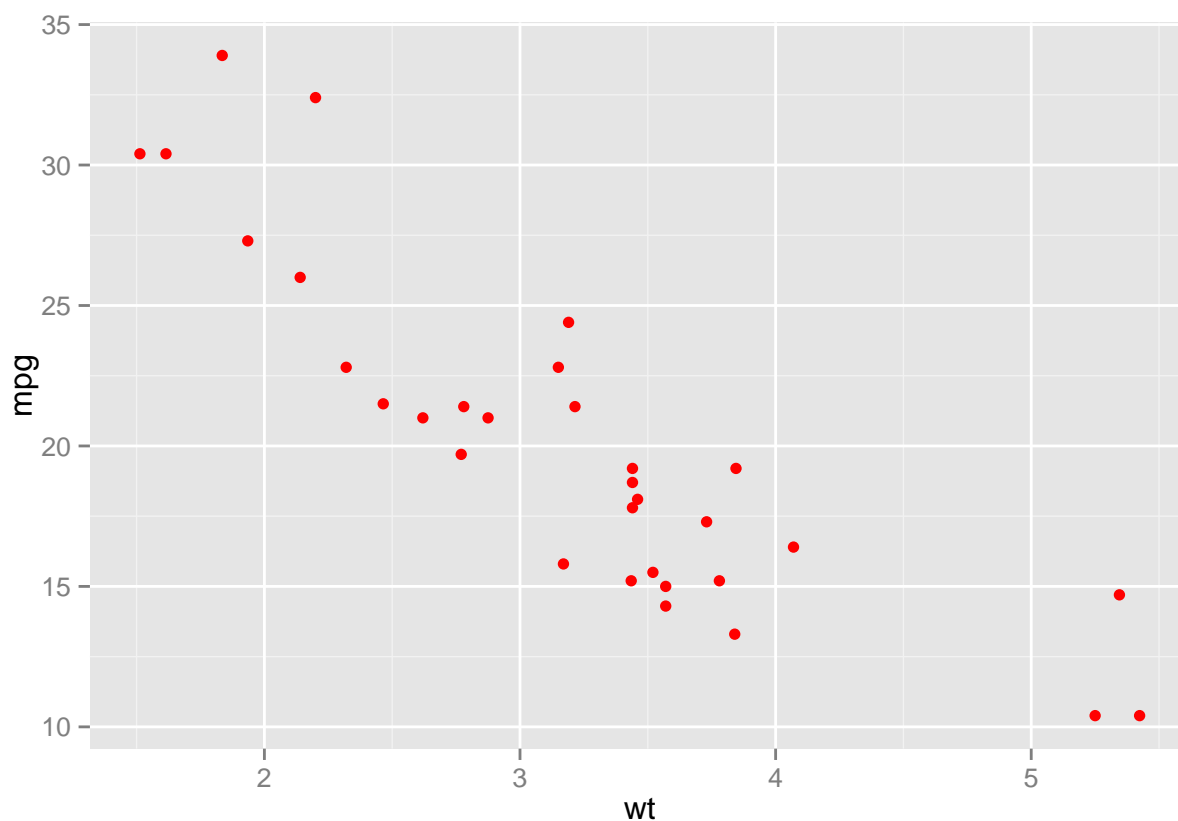
1. Setting the Colors of Objects

In the call to the geom, set the values of colour or fill

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

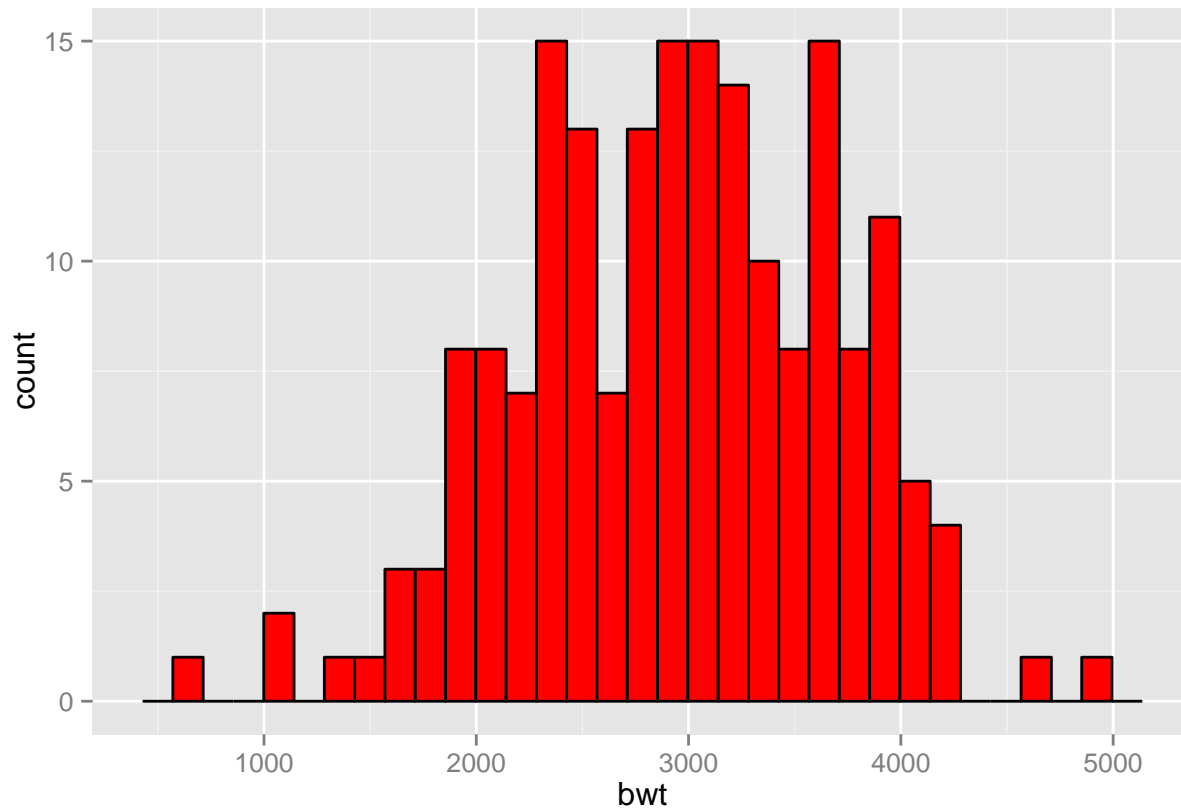
library(plyr)
library(scales)

ggplot( mtcars, aes( x = wt, y = mpg)) +
  geom_point( colour = "red")
```



```
library( MASS) # For the data set
ggplot( birthwt, aes( x = bwt)) +
  geom_histogram( fill = "red", colour = "black")
```

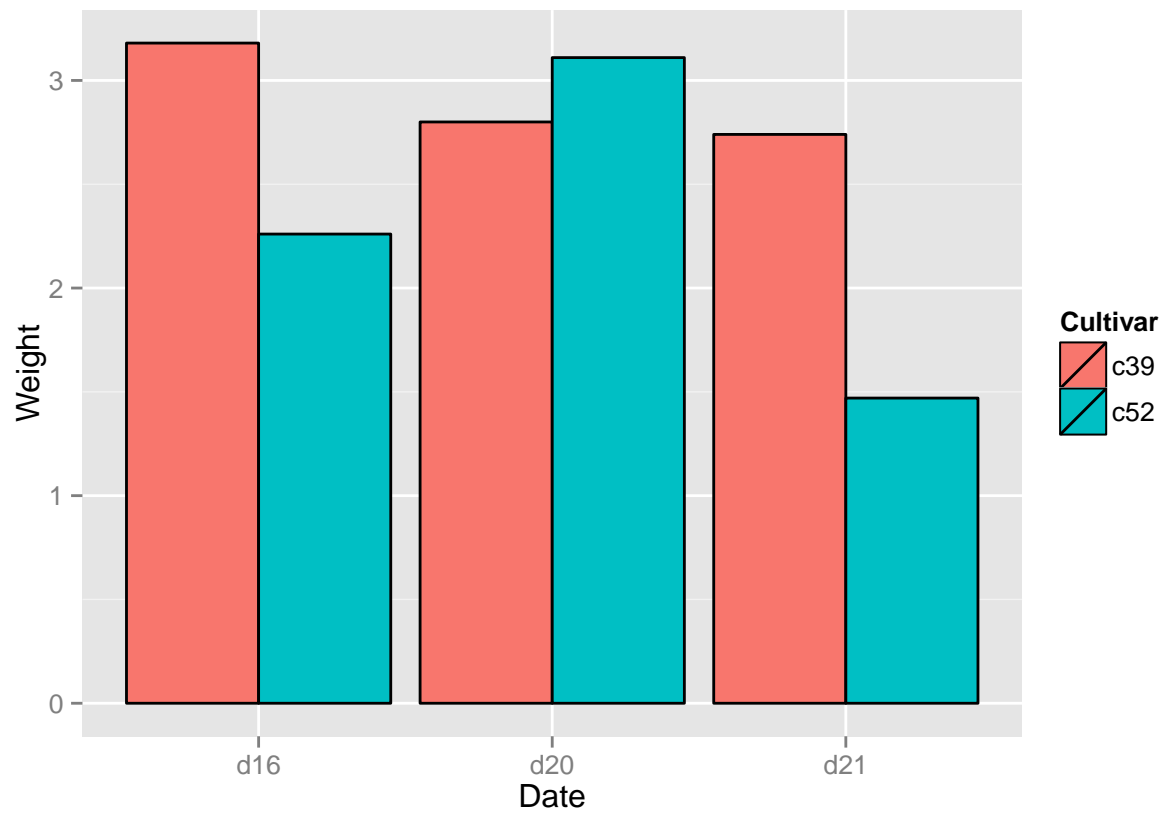
stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.



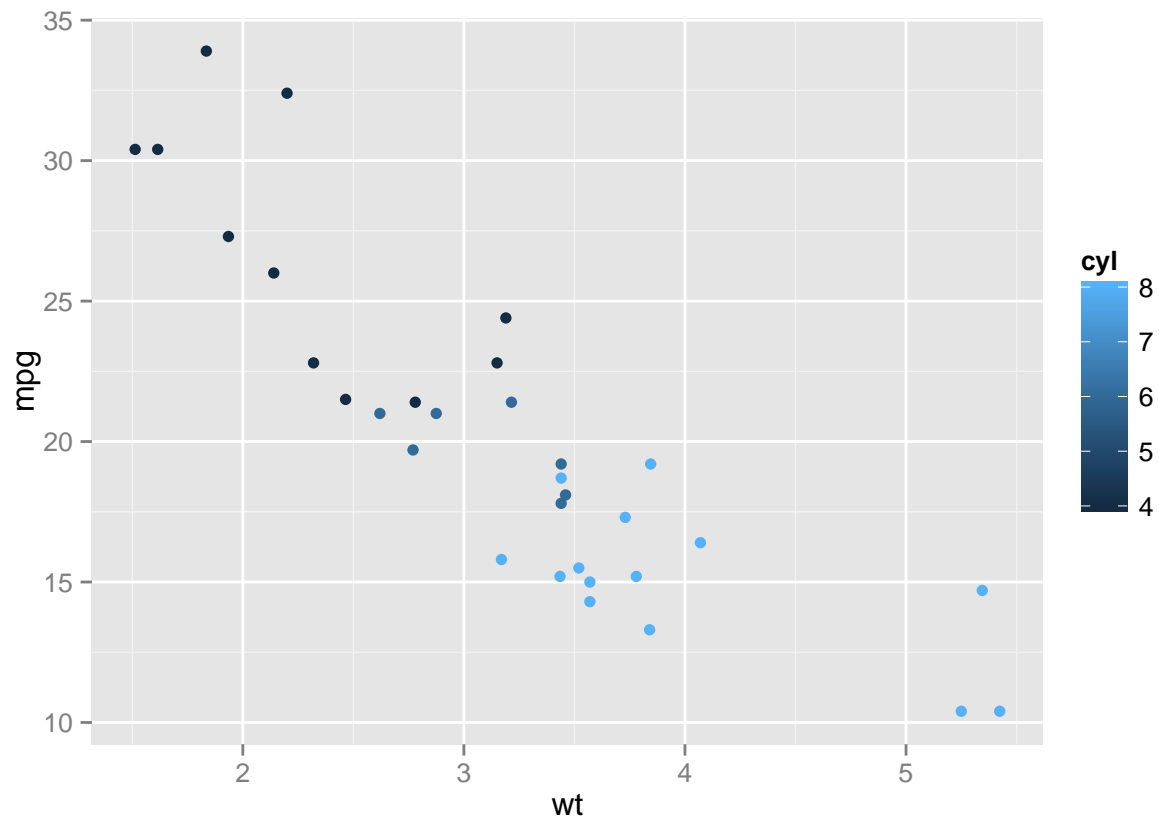
2. Mapping Variables to Colors

You want to use a variable (column from a data frame) to control the color of geoms.

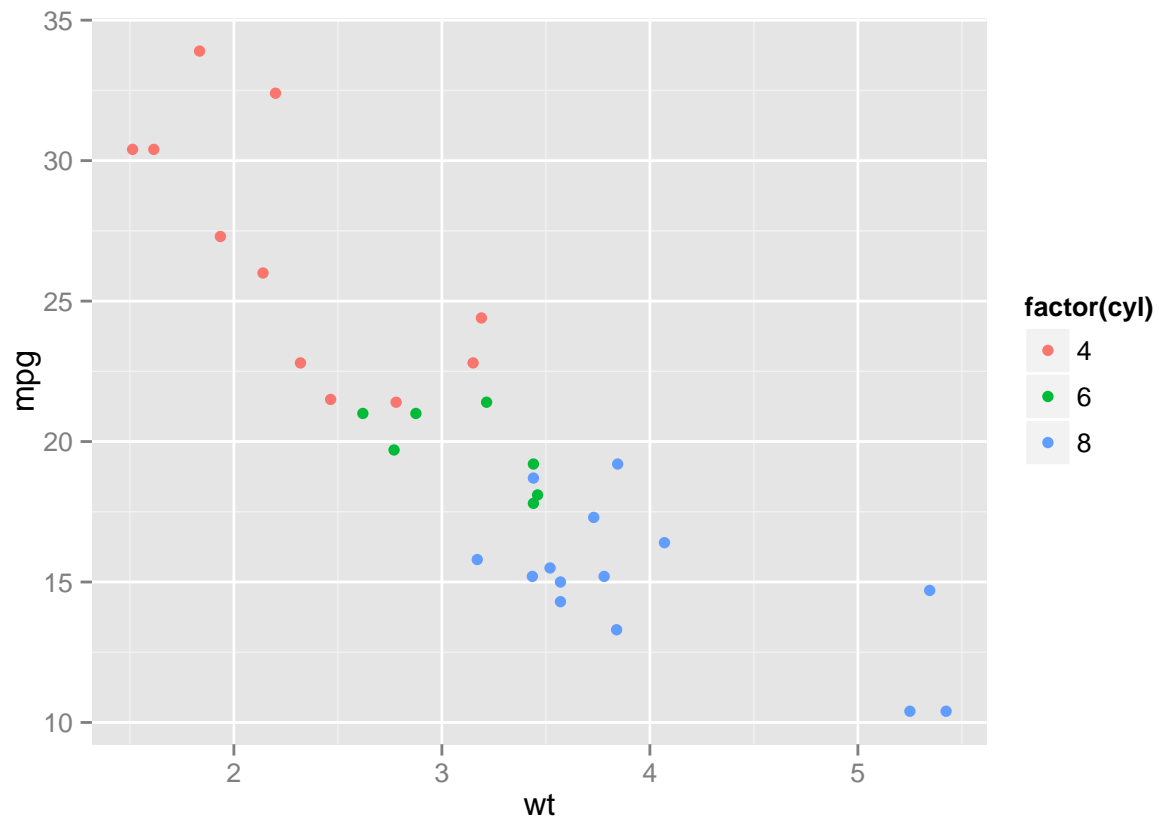
```
ggplot( cabbage_exp, aes( x = Date, y = Weight, fill = Cultivar)) +  
  geom_bar( colour ="black", position ="dodge" , stat = "identity")
```



```
ggplot( mtcars, aes( x = wt, y = mpg, colour = cyl)) +  
  geom_point()
```

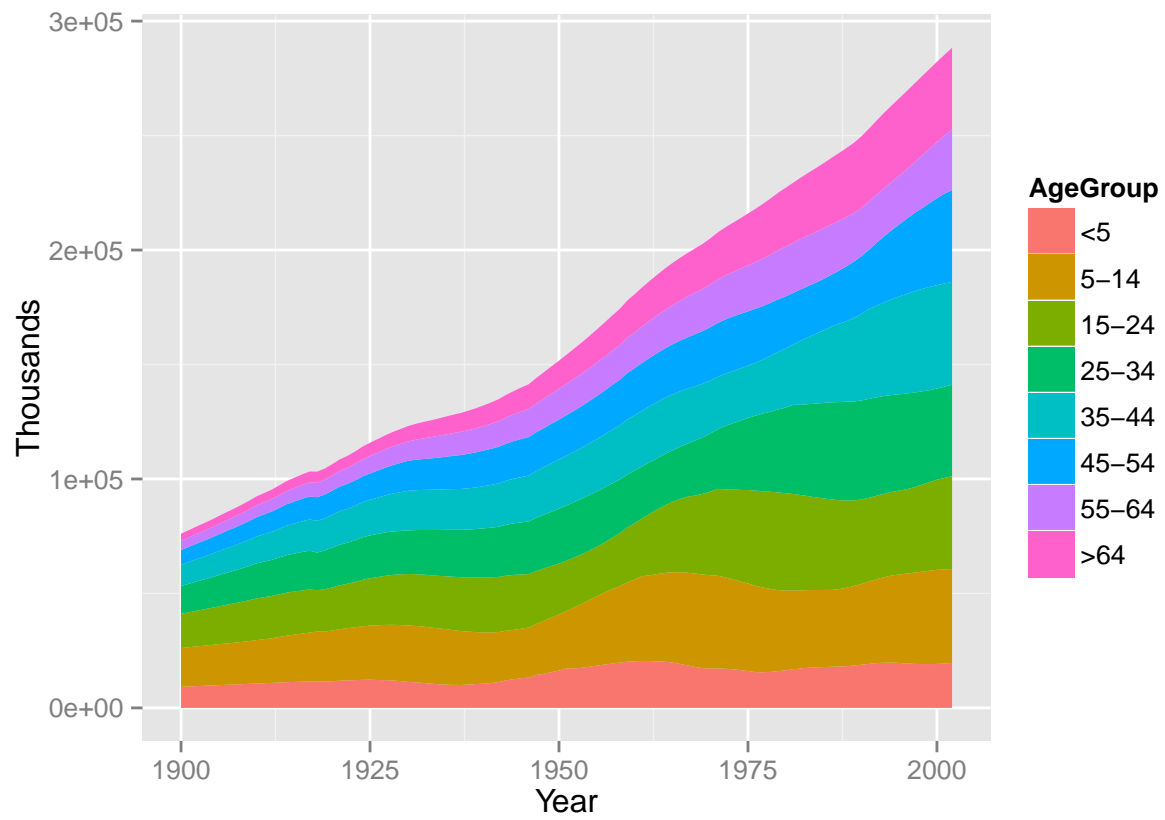


```
# Convert to factor in call to ggplot()
ggplot( mtcars, aes( x = wt, y = mpg, colour = factor( cyl))) +
  geom_point()
```

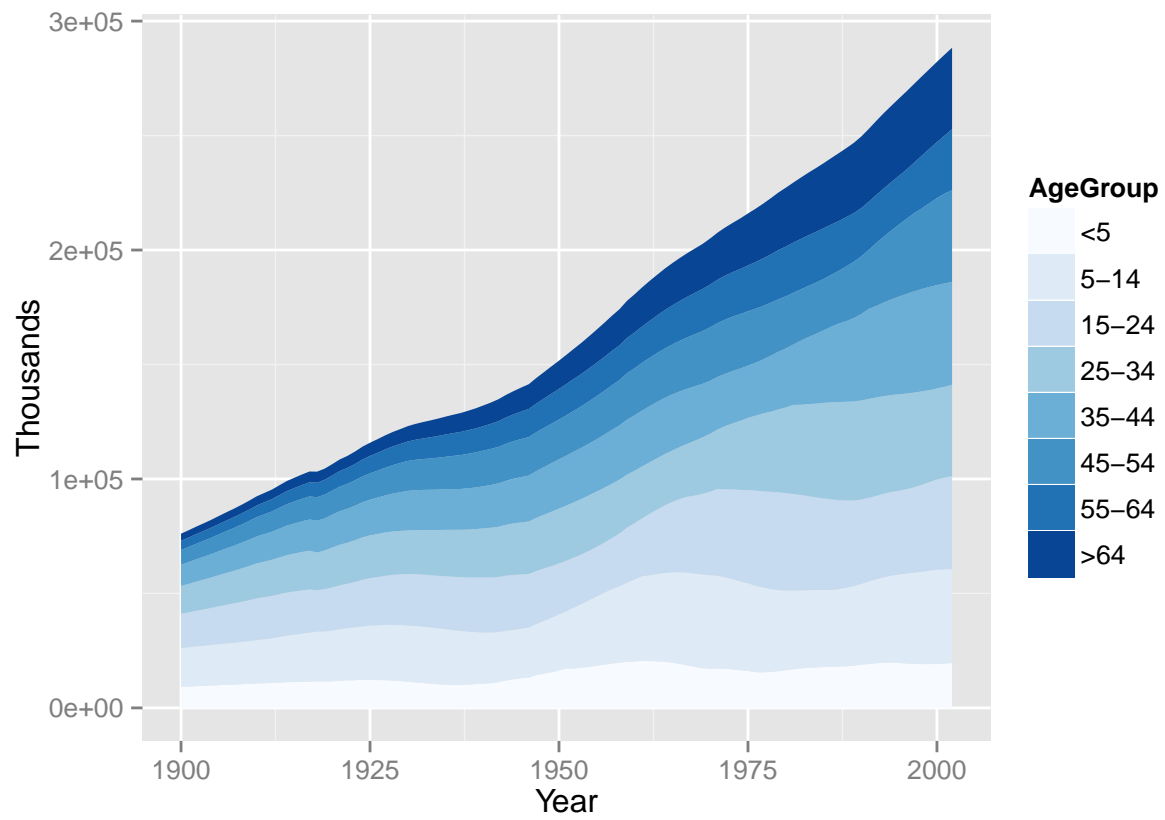


3. Using a Different Palette for a Discrete Variable

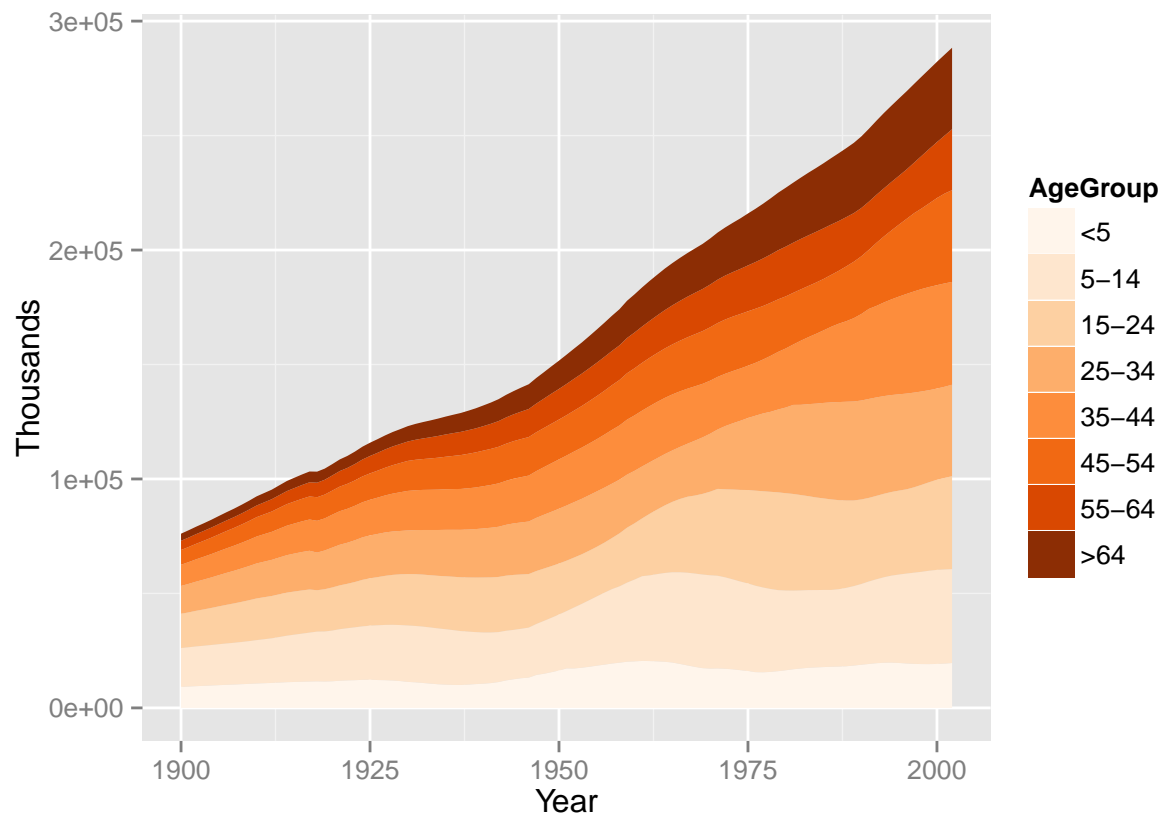
```
# Base plot
p <- ggplot( uspopage, aes( x = Year, y = Thousands, fill = AgeGroup)) +
  geom_area()
p
```



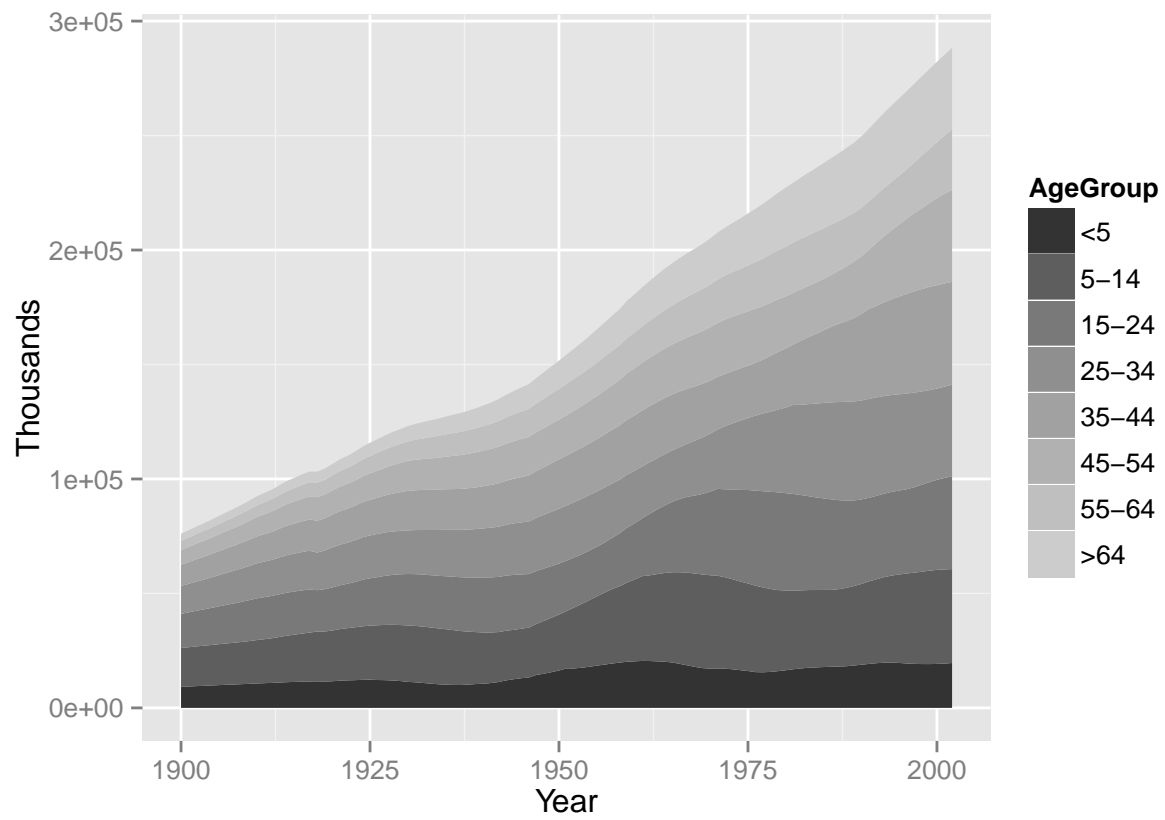
```
# ColorBrewer palette
p + scale_fill_brewer()
```



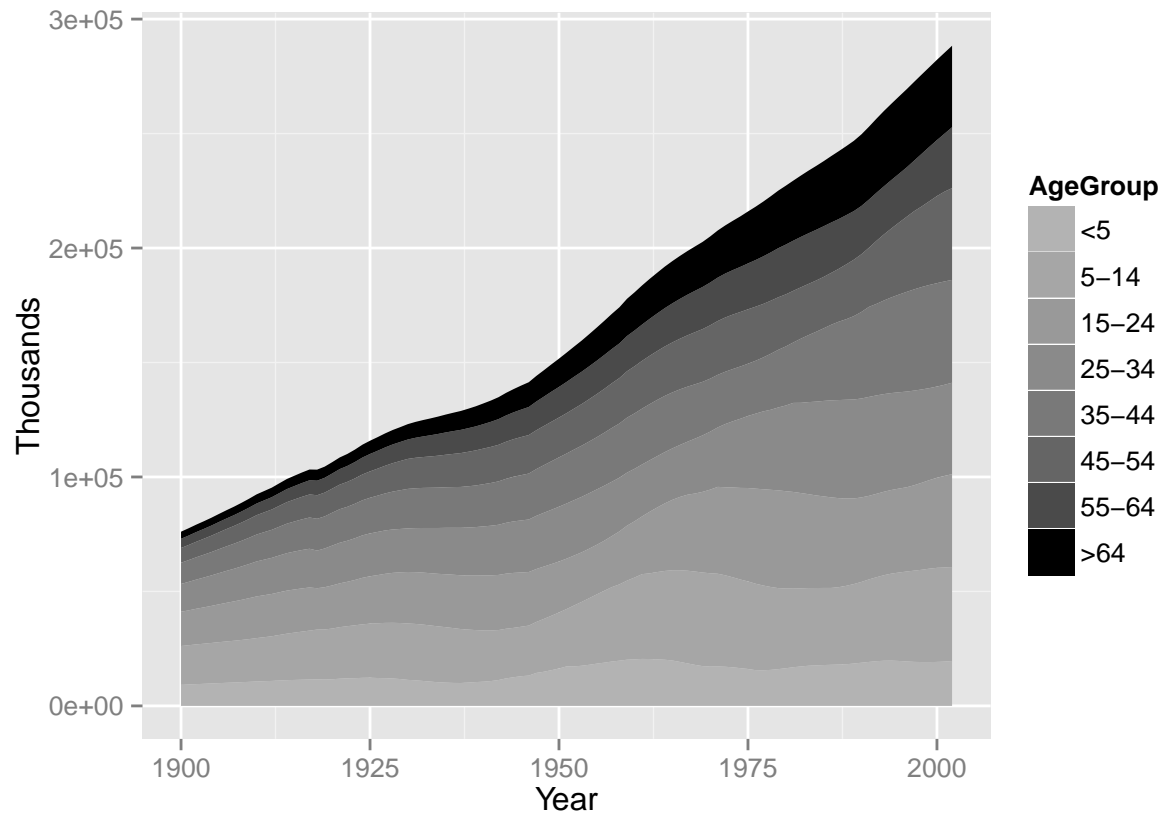
```
# ColorBrewer palette Oranges
p + scale_fill_brewer( palette = "Oranges")
```



```
# gray  
p + scale_fill_grey()
```

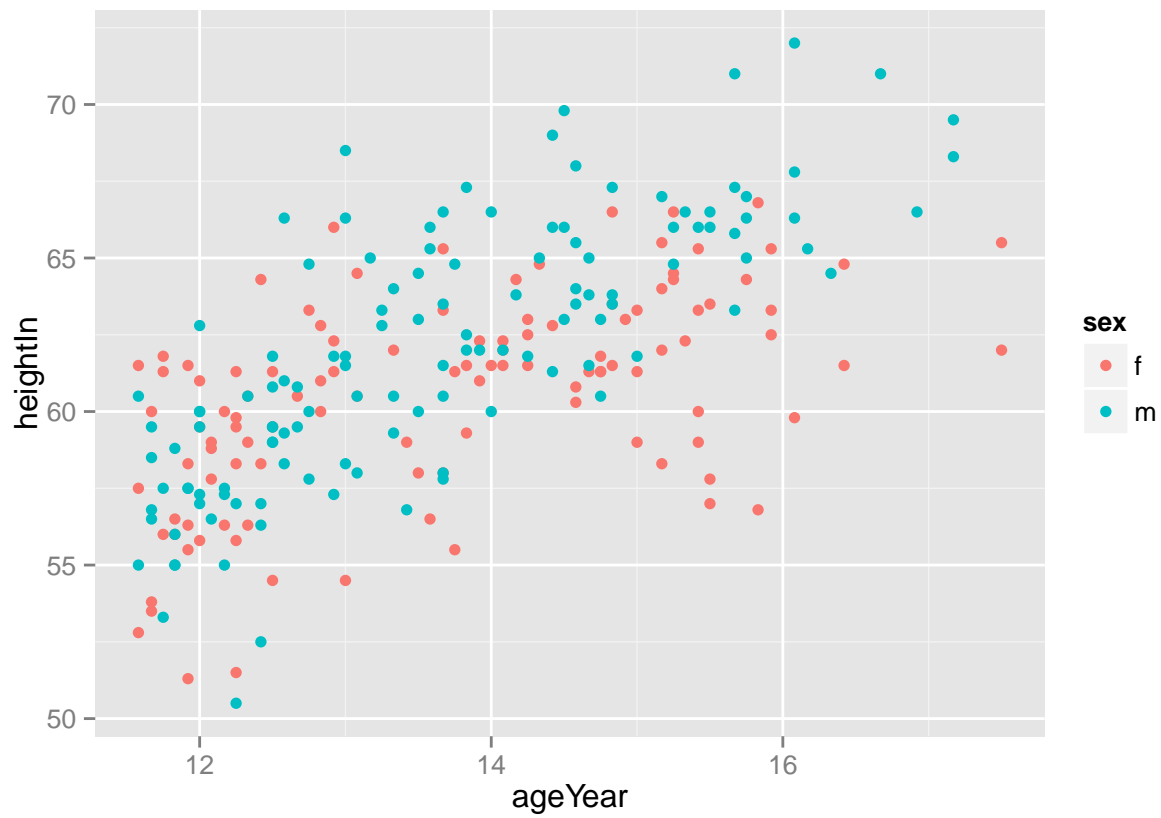
```
# Reverse the direction and use a different range of greys
p + scale_fill_grey( start = 0.7, end = 0)
```



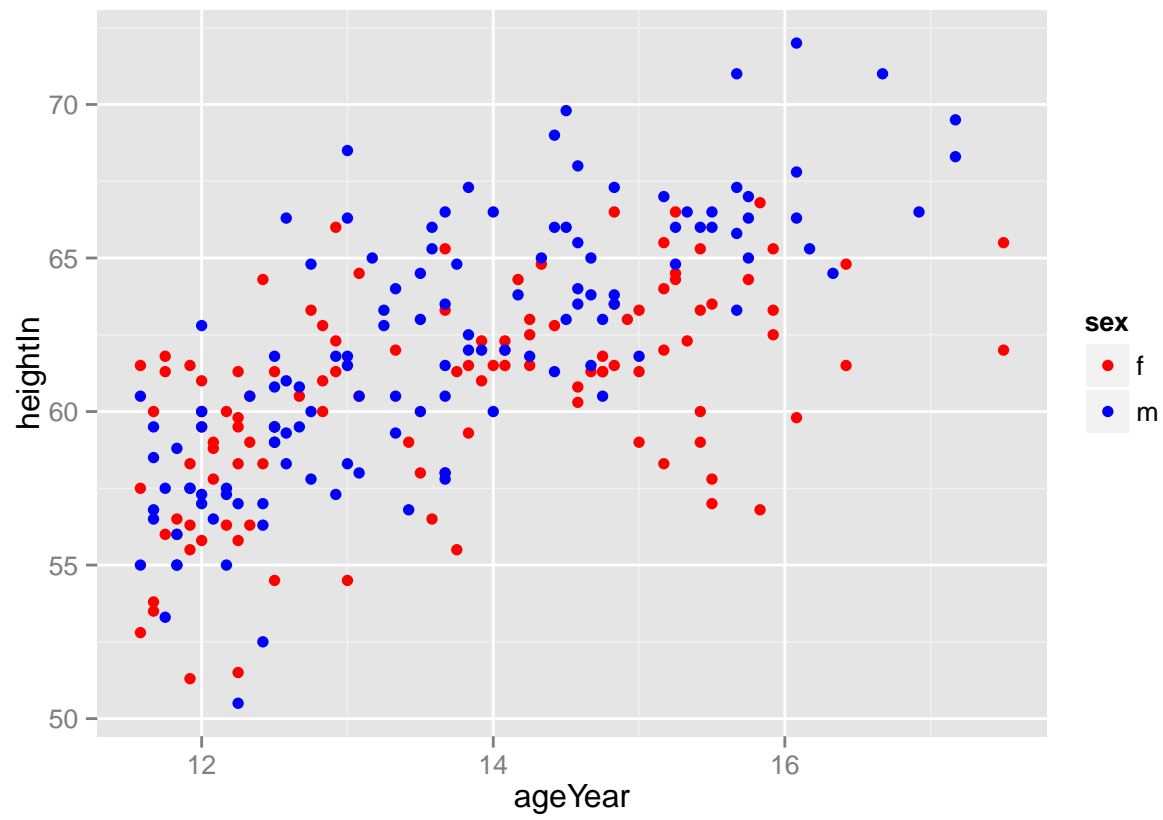
4. Using a Manually Defined Palette for a Discrete Variable

```
# Base plot
h <- ggplot( heightweight, aes( x = ageYear, y = heightIn, colour = sex)) +
  geom_point()

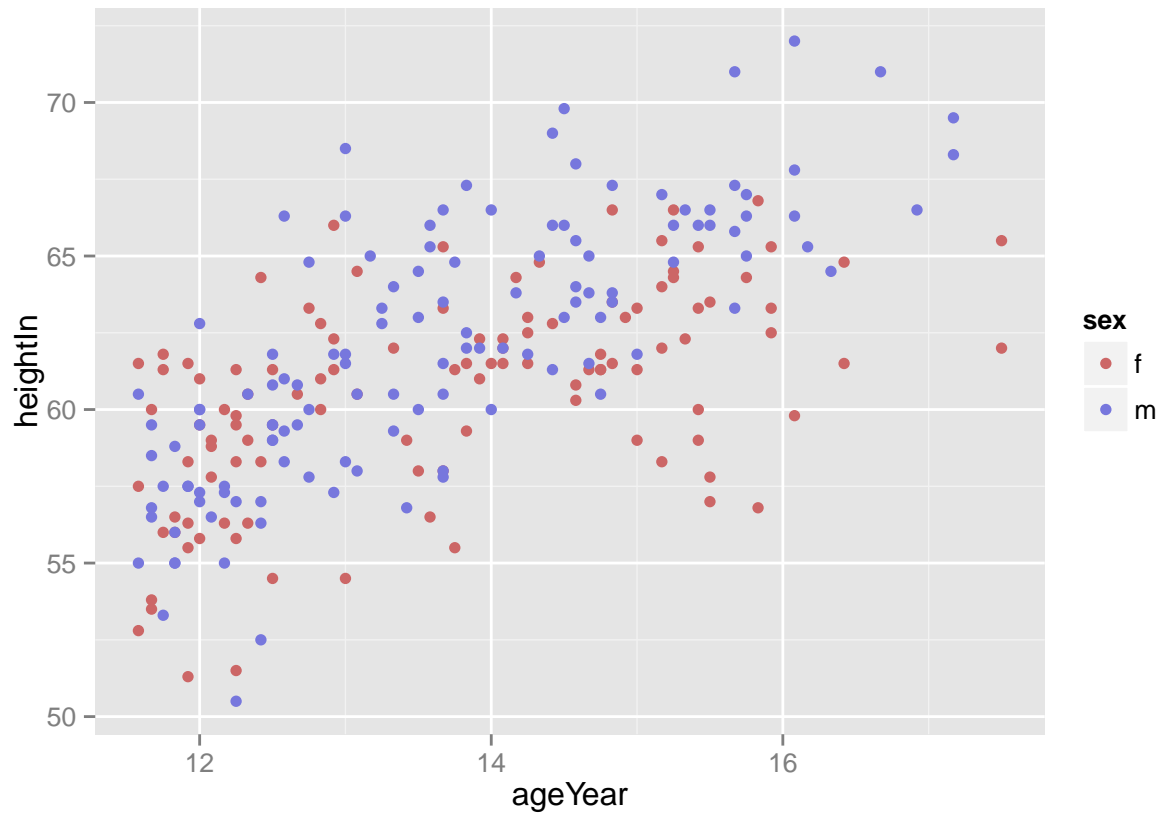
h
```



```
# Using color names  
h + scale_colour_manual( values = c("red", "blue"))
```



```
# Using RGB values  
h + scale_colour_manual( values = c("#CC6666", "#7777DD"))
```

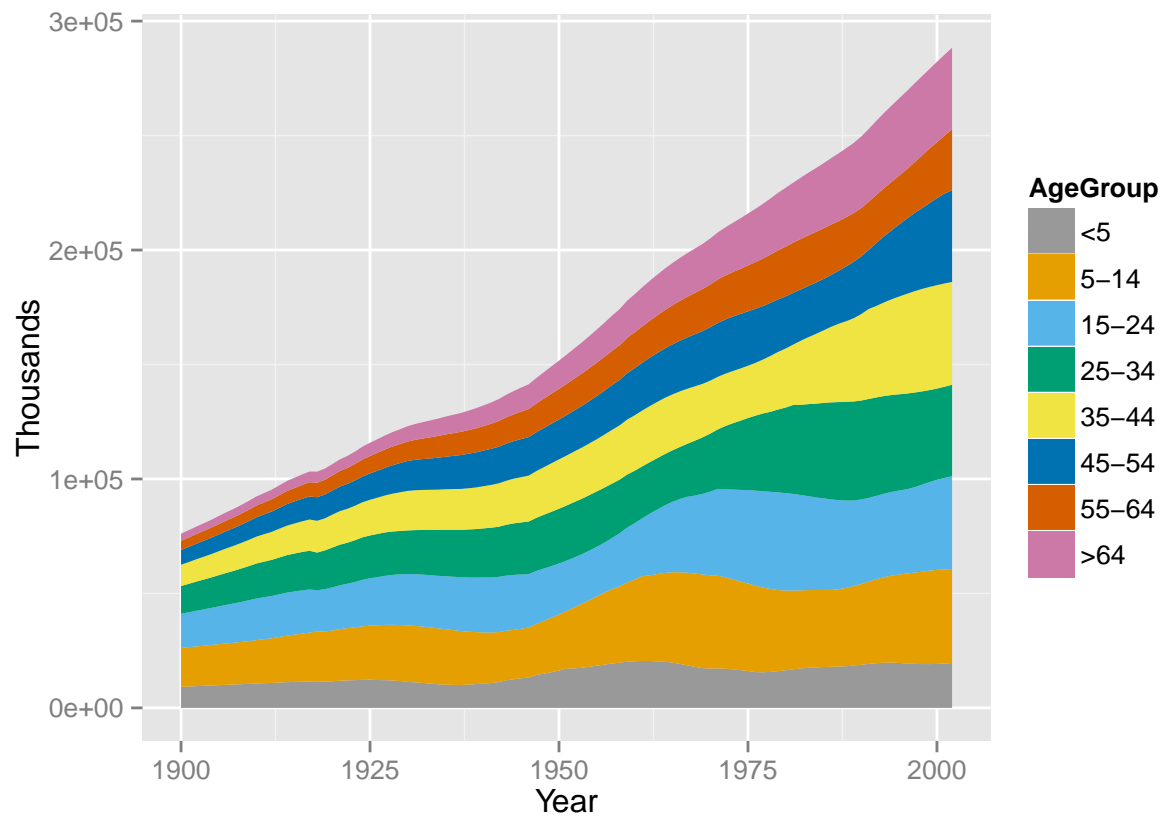


5. Using a Colorblind-Friendly Palette

```
# Base plot
p <- ggplot( uspopage, aes( x = Year, y = Thousands, fill = AgeGroup)) +
  geom_area()

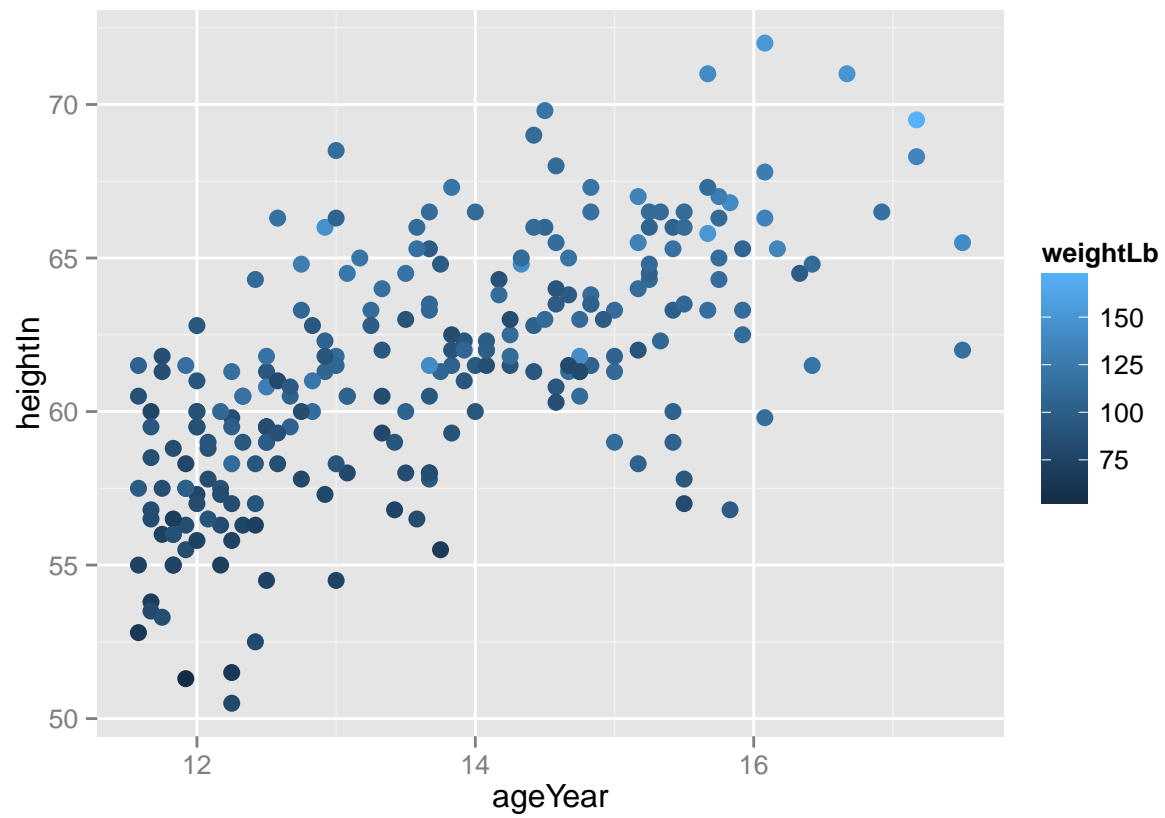
# The palette with grey:
cb_palette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")

# Add it to the plot
p + scale_fill_manual( values = cb_palette)
```

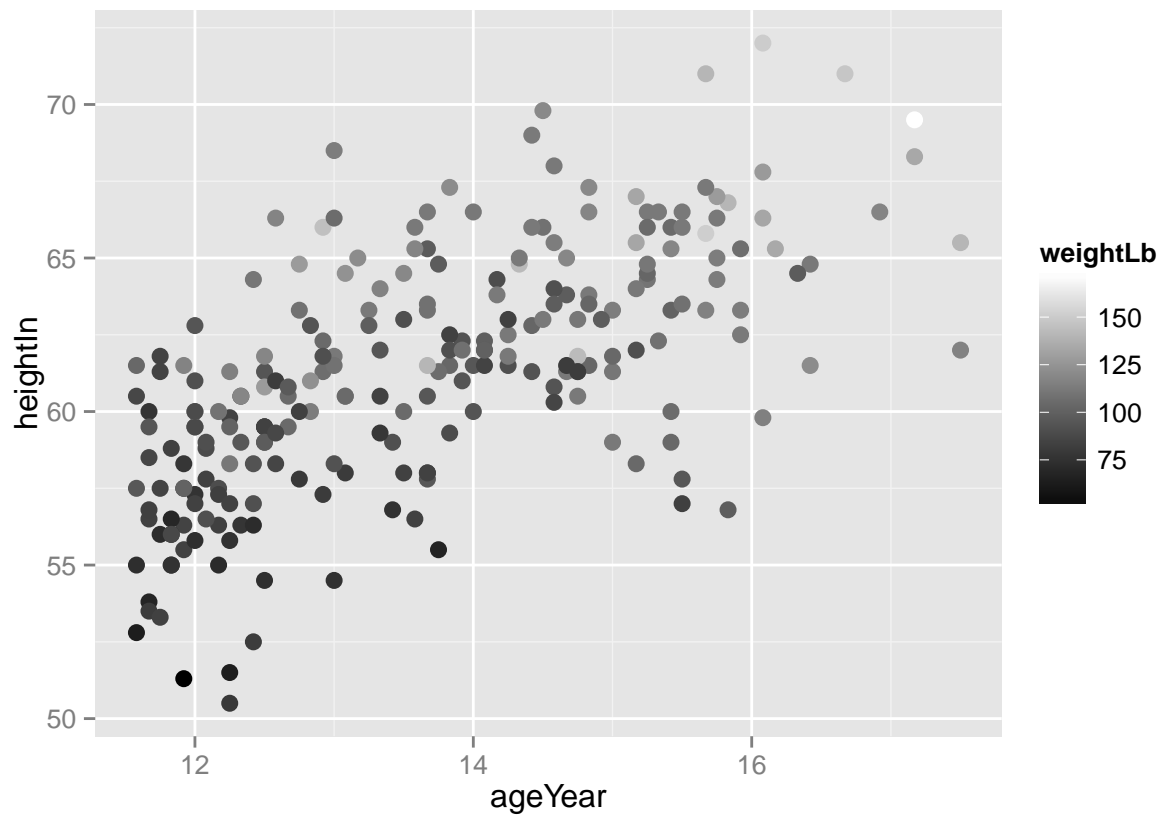


6. Using a Colorblind-Friendly Palette

```
# Base plot
p <- ggplot( heightweight, aes( x = ageYear, y = heightIn, colour = weightLb)) +
  geom_point( size = 3)
p
```

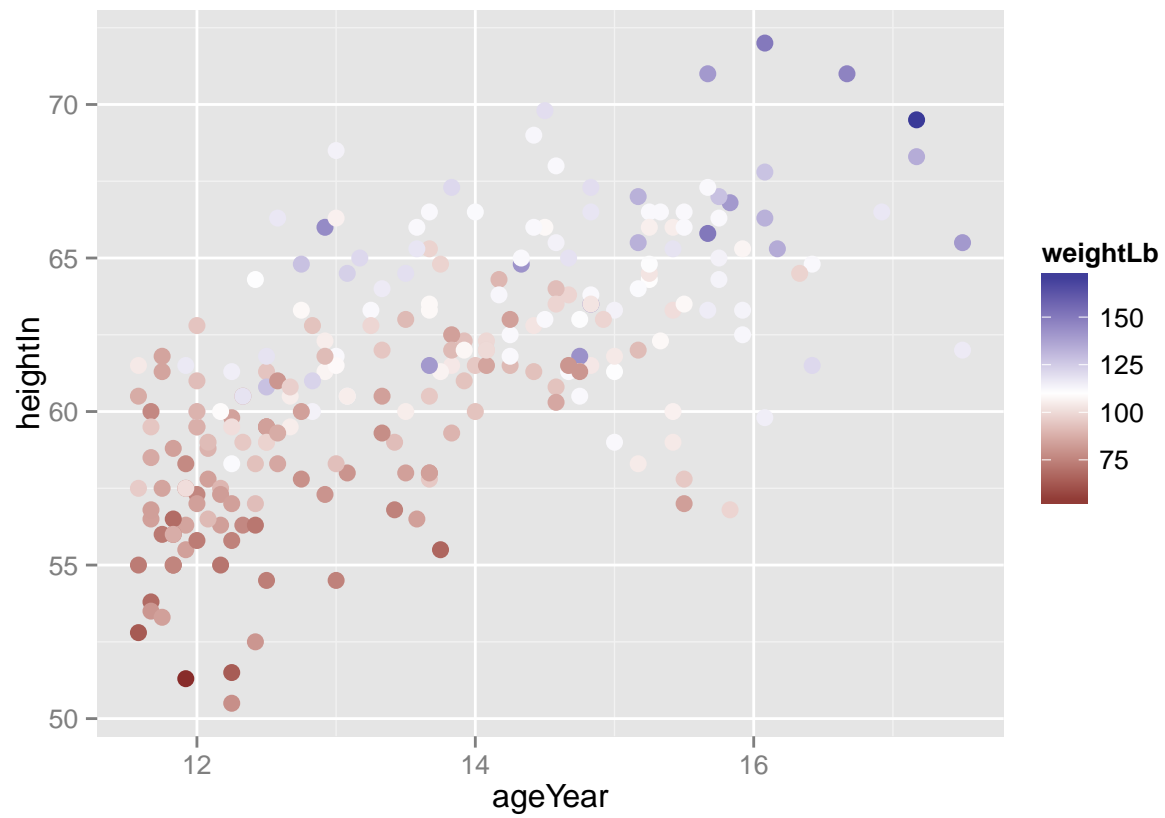


```
# With a gradient between two colors  
p + scale_colour_gradient( low ="black", high ="white")
```

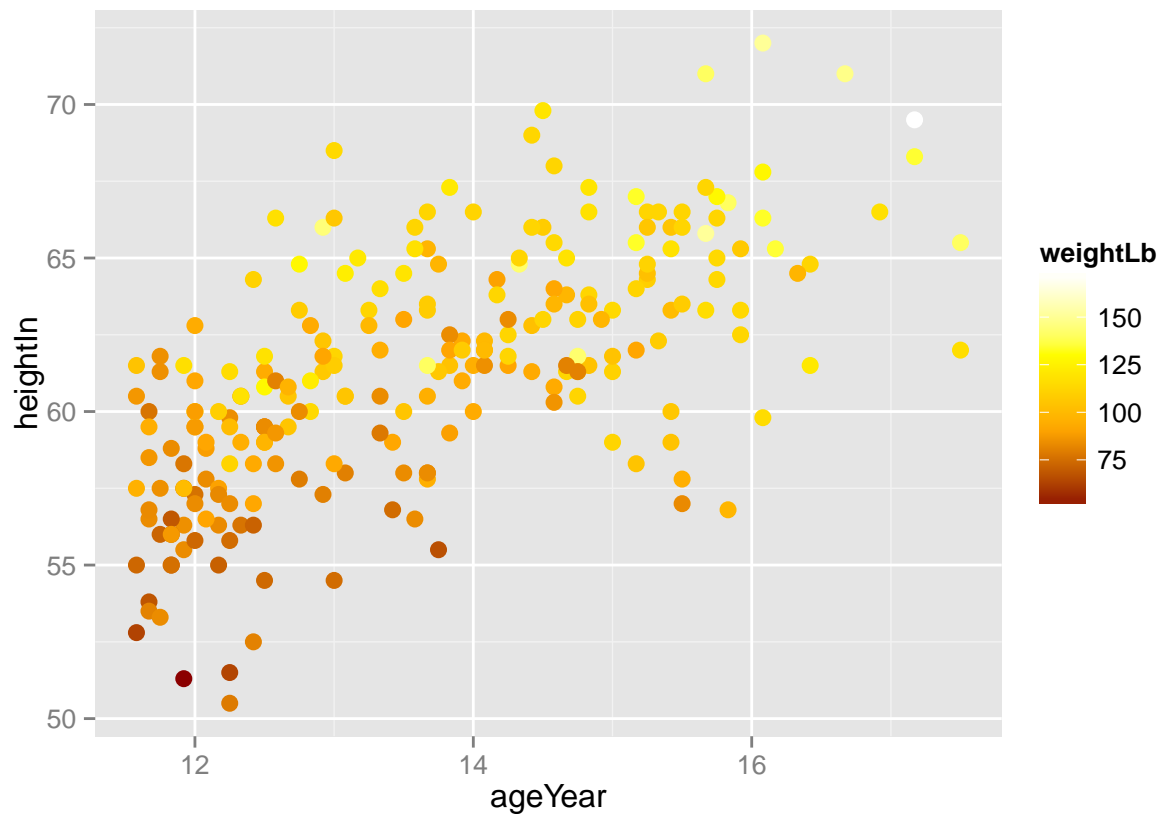


```
# A gradient with a white midpoint library( scales)
p + scale_colour_gradient2( low = muted("red"), mid = "white", high = muted("blue"), midpoint = 110)
```

```
## Warning: Non Lab interpolation is deprecated
```

```
# A gradient of n colors
p + scale_colour_gradientn( colours = c("darkred", "orange", "yellow", "white"))
```

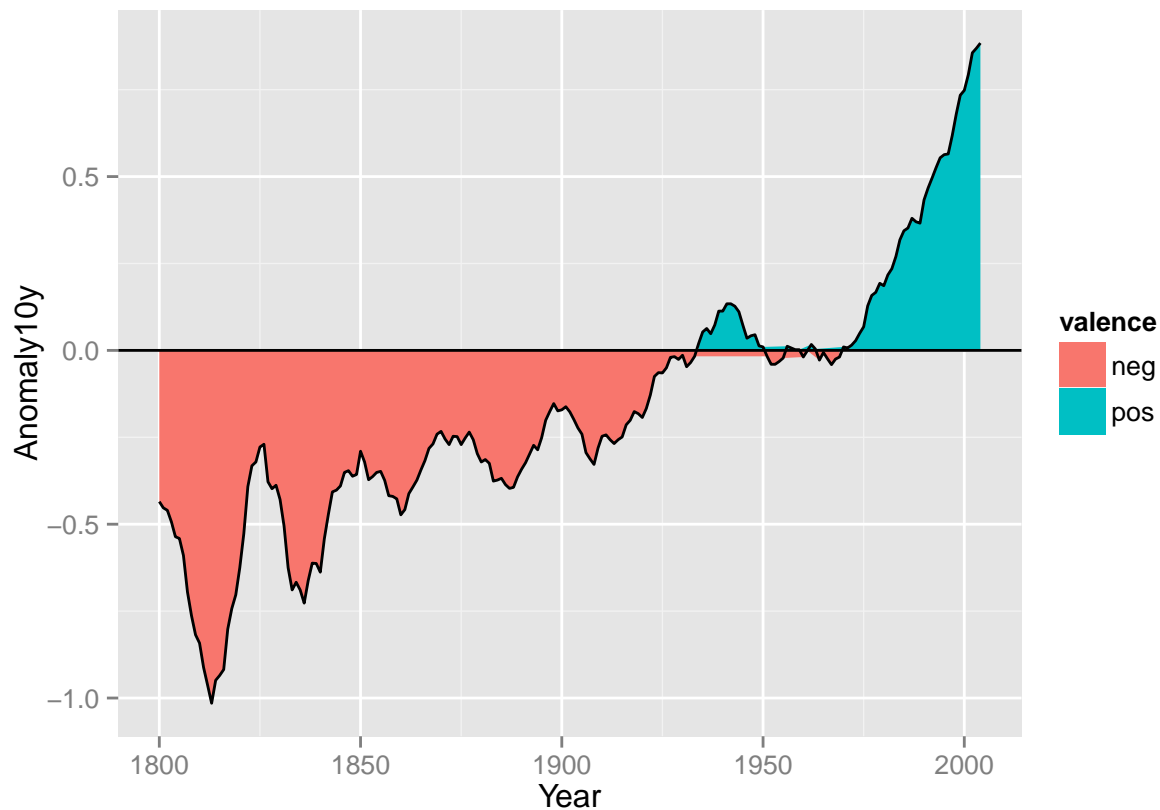


7. Coloring a Shaded Region Based on Value

```
cb <- subset( climate, Source == "Berkeley")
cb$valence[ cb$Anomaly10y >= 0] <- "pos"
cb$valence[ cb$Anomaly10y < 0] <- "neg"
str(cb)
```

```
## 'data.frame': 205 obs. of 7 variables:
## $ Source : chr "Berkeley" "Berkeley" "Berkeley" "Berkeley" ...
## $ Year : num 1800 1801 1802 1803 1804 ...
## $ Anomaly1y : num NA NA NA NA NA NA NA NA NA NA ...
## $ Anomaly5y : num NA NA NA NA NA NA NA NA NA NA ...
## $ Anomaly10y: num -0.435 -0.453 -0.46 -0.493 -0.536 -0.541 -0.59 -0.695 -0.763 -0.818 ...
## $ Unc10y : num 0.505 0.493 0.486 0.489 0.483 0.475 0.468 0.461 0.453 0.451 ...
## $ valence : chr "neg" "neg" "neg" "neg" ...
```

```
ggplot( cb, aes( x = Year, y = Anomaly10y)) +
  geom_area( aes( fill = valence)) +
  geom_line() +
  geom_hline( yintercept = 0)
```



```
# approx() returns a list with x and y vectors
interp <- approx( cb$Year, cb$Anomaly10y, n = 1000)

# Put in a data frame and recalculate valence
cbi <- data.frame( Year = interp$x, Anomaly10y = interp$y)
cbi$valence[ cbi$Anomaly10y >= 0] <- "pos"
cbi$valence[ cbi $ Anomaly10y < 0] <- "neg"

## plot with intrapolations cbi
ggplot( cbi, aes( x = Year, y = Anomaly10y)) +
  geom_area( aes( fill = valence), alpha = .4) +
  geom_line() + geom_hline( yintercept = 0) +
  scale_fill_manual( values = c("#CCEEFF", "#FFDDDD"), guide = FALSE) +
  scale_x_continuous( expand = c( 0, 0))
```

