Data Visualization

QBS 103

July 23, 2024

Lesson Objectives

At the end of this lecture you should be able to:

1. Produce simple plots using *ggplot2*
2. Define a plot theme in *ggplot2*
3. Assign colors and color palettes to plots in *ggplot2*
4. Manually annotate plots in *ggplot2*

Resources

ggplot2 cheat sheet: https://www.maths.usyd.edu.au/u/UG/SM/STAT3022/r/current/Misc/data-visualization-2.1.pdf Pre-defined themes for *ggplot2*: https://ggplot2.tidyverse.org/reference/ggtheme.html

Generating Data

We're going to use the same data set we generated for the data wrangling lectures:

```
# Load packages
library(tidyverse) # note: tidyverse includes ggplot2

# Set a random seed
set.seed(103)

# Define a data frame with our randomly generated data
randomData <- data.frame('SubjectID' = seq(1:1000),
                         'Systolic.BP' = rnorm(n = 1000,mean = 128,sd = 20),
                         'Diastolic.BP' = rnorm(n = 1000,mean = 71,sd = 10),
                         'Age' = trunc(runif(n = 1000,min = 18,max = 70)),
                         'Male' = rbinom(n = 1000,size = 1,prob = 0.5))
# Define binary variable for biological sex
randomData$BiologicalSex <- factor(ifelse(randomData$Male == 1,'Male','Female'))
# Define variable specifying age above 65 (medicare eligible)
randomData$MedicareAge <- ifelse(randomData$Age < 65,F,T)
# Define a variable for hypertension
randomData$Hypertension <- ifelse(randomData$Systolic.BP > 130 | randomData$Diastolic.BP > 80,
                                  'Hypertensive','Normotensive')

longData <- randomData %>%
  # Melt into a wide format data frame
```

```
  gather(key = BP.Type, value = BP,c('Systolic.BP','Diastolic.BP')) %>%
  # Split BP.Type into two columns: "Systolic"/"Diastolic" in BP.Type and "BP" in Bad.ID
  separate(col = BP.Type, into = c('BP.Type','Bad.ID')) %>%
  # Keep only required columns
  select(SubjectID,Age,BiologicalSex,Hypertension,BP.Type,BP)

# Print top entries
head(longData)
```

```
##   SubjectID Age BiologicalSex Hypertension  BP.Type        BP
## 1         1  52        Female Normotensive Systolic 112.28054
## 2         2  56        Female Normotensive Systolic 129.09478
## 3         3  25          Male Normotensive Systolic 104.54879
## 4         4  41        Female Normotensive Systolic 124.65374
## 5         5  41        Female Normotensive Systolic  90.69937
## 6         6  31        Female Normotensive Systolic 125.59120
```

```
# Print last entries
tail(longData)
```

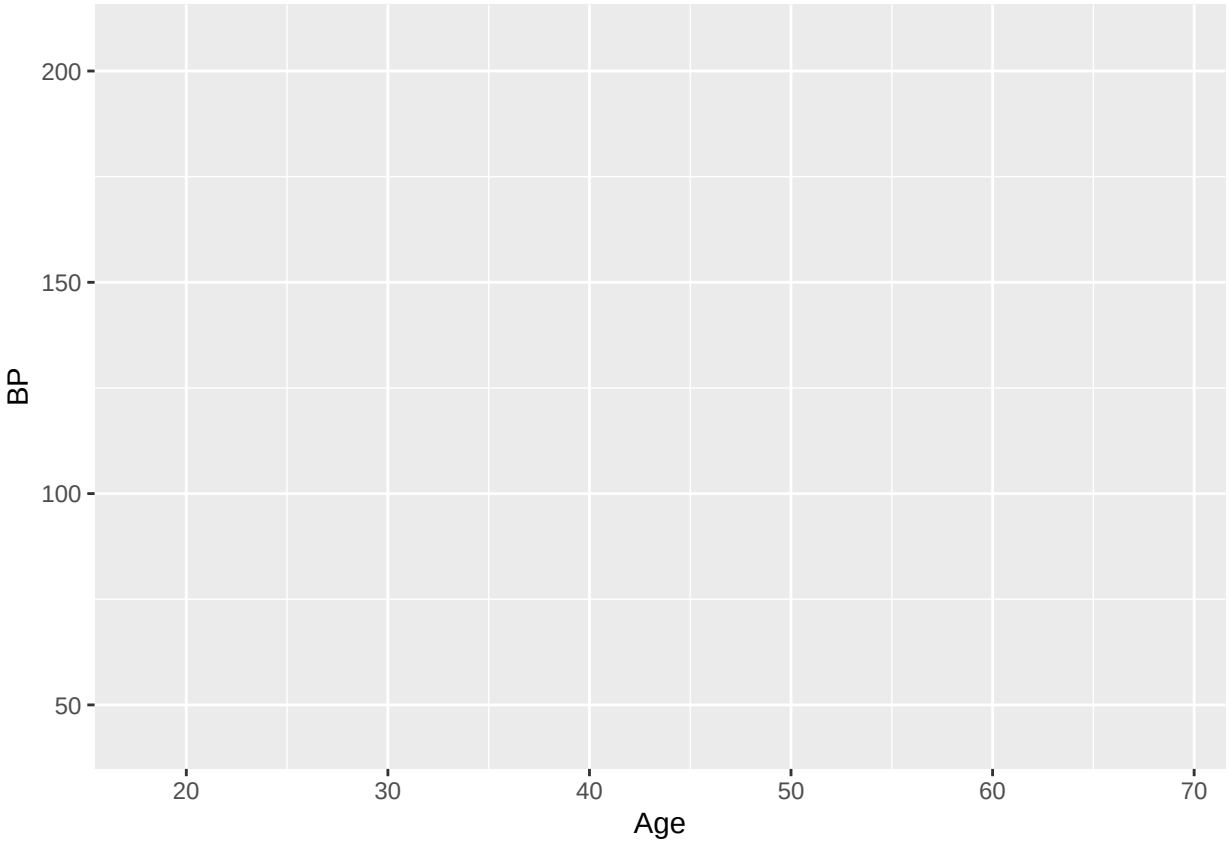
```
##      SubjectID Age BiologicalSex Hypertension   BP.Type       BP
## 1995       995  45          Male Hypertensive Diastolic 79.71709
## 1996       996  52        Female Hypertensive Diastolic 86.06482
## 1997       997  34        Female Normotensive Diastolic 70.35285
## 1998       998  56          Male Hypertensive Diastolic 69.03647
## 1999       999  57          Male Normotensive Diastolic 75.38034
## 2000      1000  39          Male Normotensive Diastolic 78.96079
```

Basic Plotting in *ggplot2*

In *ggplot2*, you refer first to your data, and then to the elements you want to add to your plot. This definitely takes more steps than the plots we made last class, but as you'll see, it allows for much more customization of our final plots. We'll start here with a simple scatter plot looking at age and blood pressure.

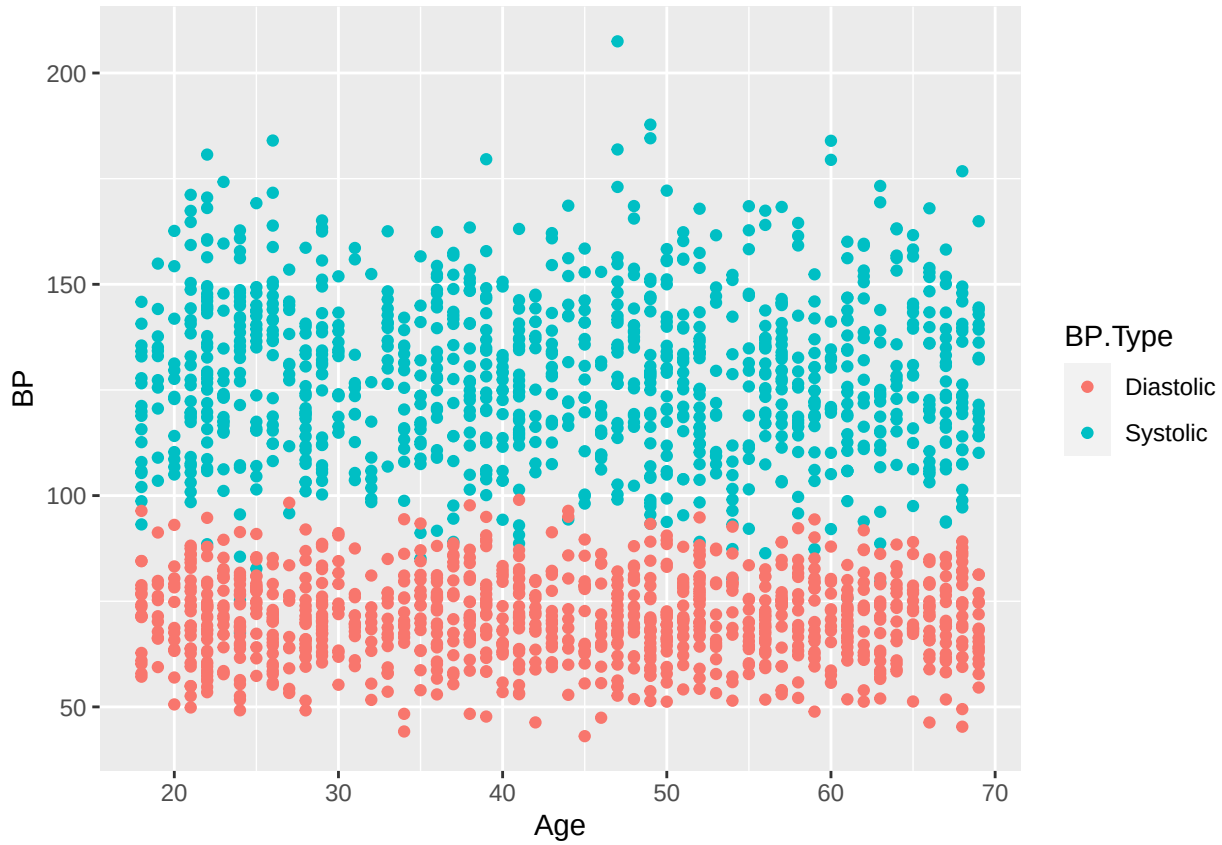
First, what happens if we only use the *ggplot* function?

```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Age,y = BP,color = BP.Type))
```



We need to use the *geom_point()* function to tell it we want to add the points to the plot.

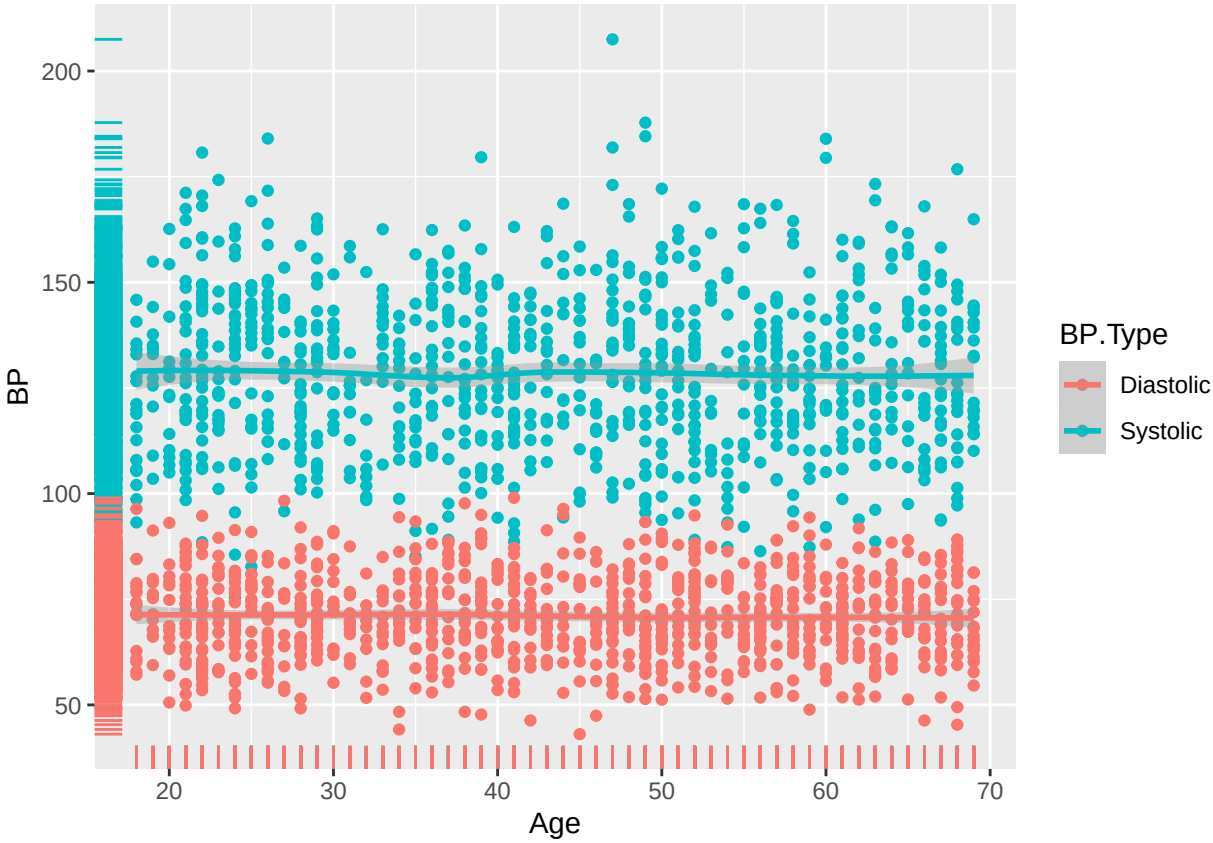
```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add points to our scatter plot
  geom_point()
```



We can also add other elements on top of this such as a smoothed loess curve and a rug

```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add points to our scatter plot
  geom_point() +
  # Add loess curve with shaded error
  geom_smooth(method = loess) +
  # Add rug
  geom_rug()
```

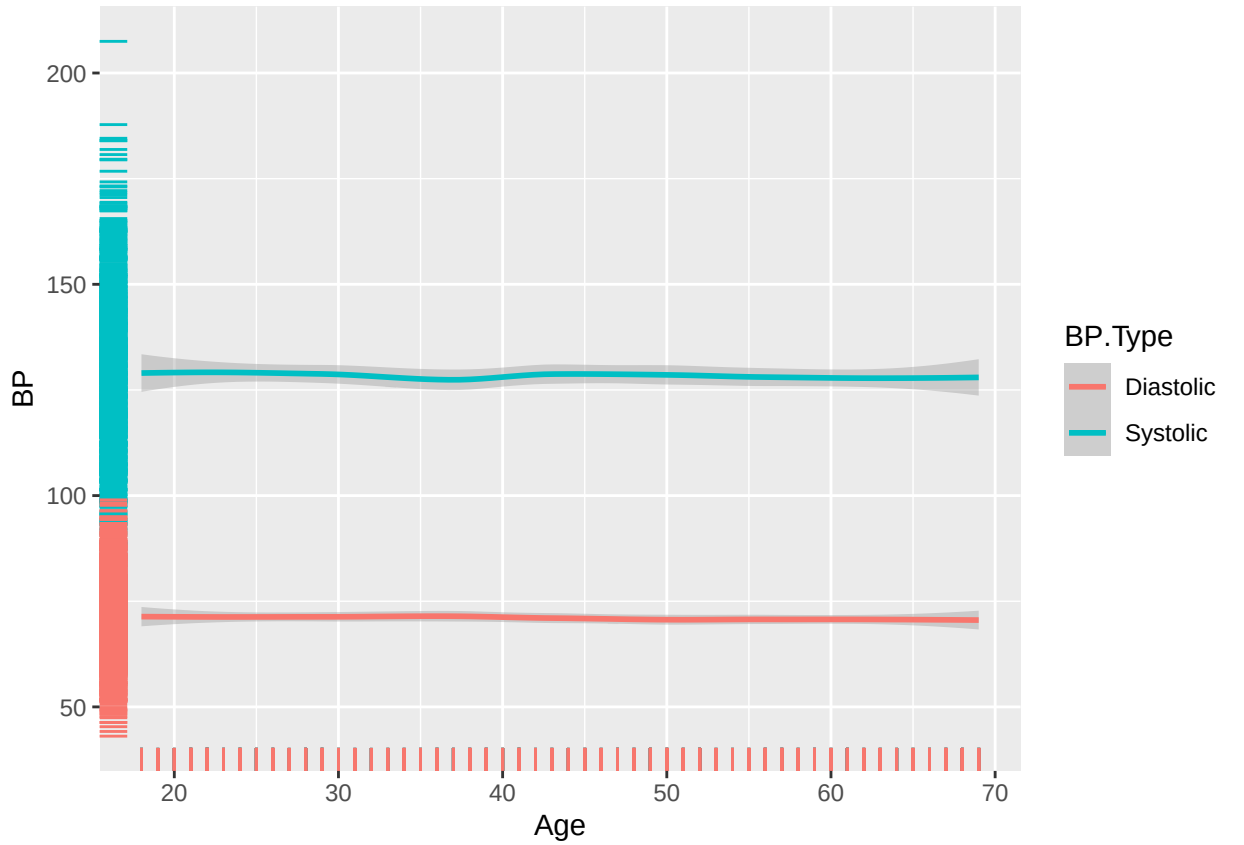
```
## `geom_smooth()` using formula = 'y ~ x'
```



We can also plot only the loess curve and rug.

```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add loess curve with shaded error
  geom_smooth(method = loess) +
  # Add rug
  geom_rug()
```

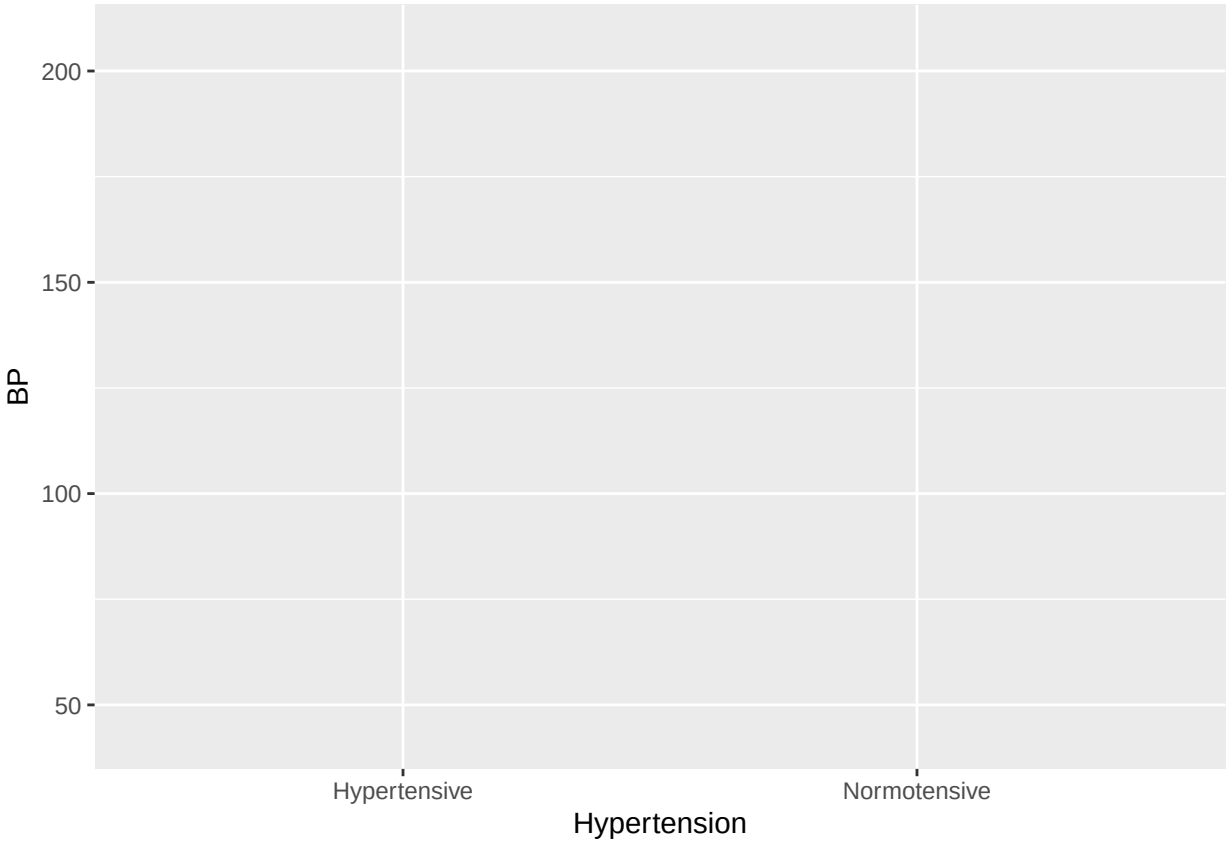
```
## `geom_smooth()` using formula = 'y ~ x'
```



Essentially, the *ggplot()* function forms as our base where we feed it the parameters of the data we want plotted and then we stack upon it all the plot elements we want to include.

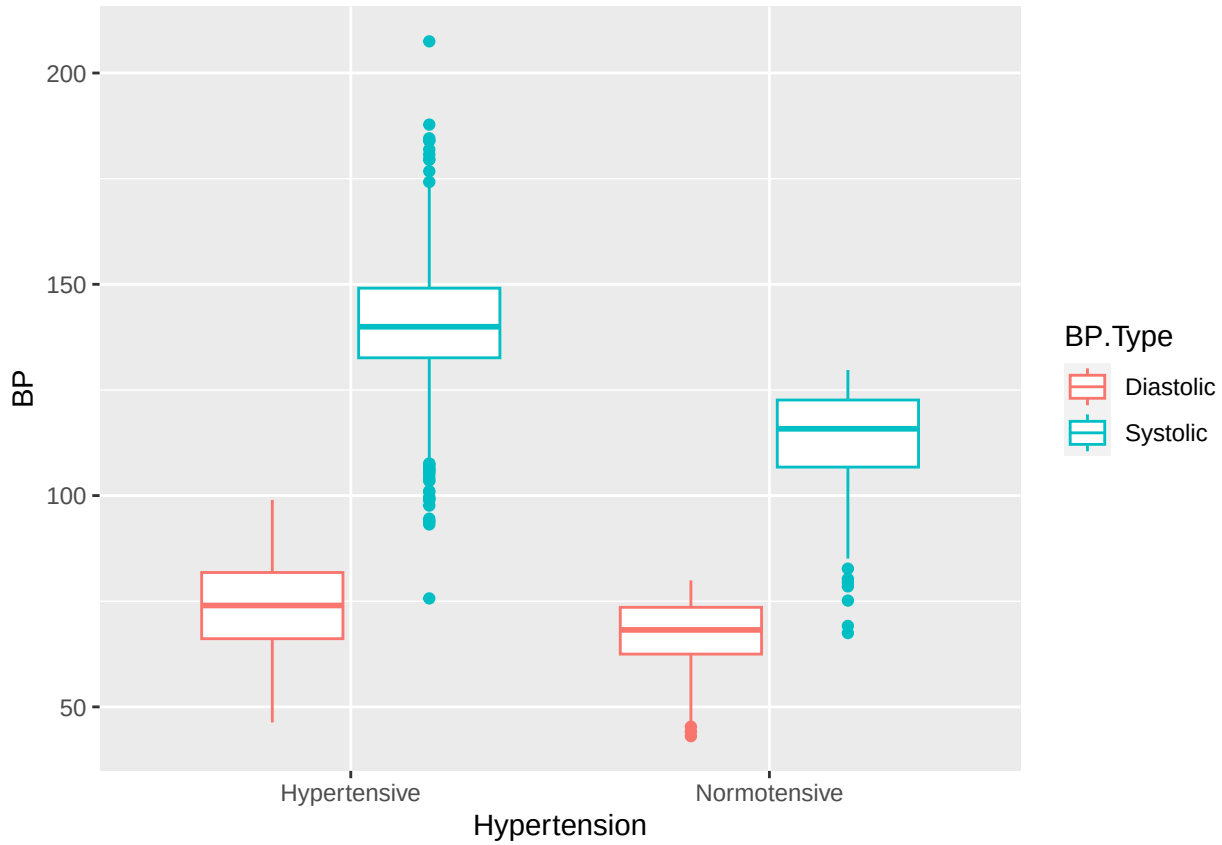
We can also do this using one continuous and one categorical variable, such as looking at the distribution of blood pressure by hypertension status. If we again start only with the *ggplot* function, our base plot will look like this:

```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Hypertension,y = BP,color = BP.Type))
```



Now we can add elements like a box plot or a violin plot on top of that base:

```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Hypertension,y = BP,color = BP.Type)) +
  # Add box plot
  geom_boxplot()
```

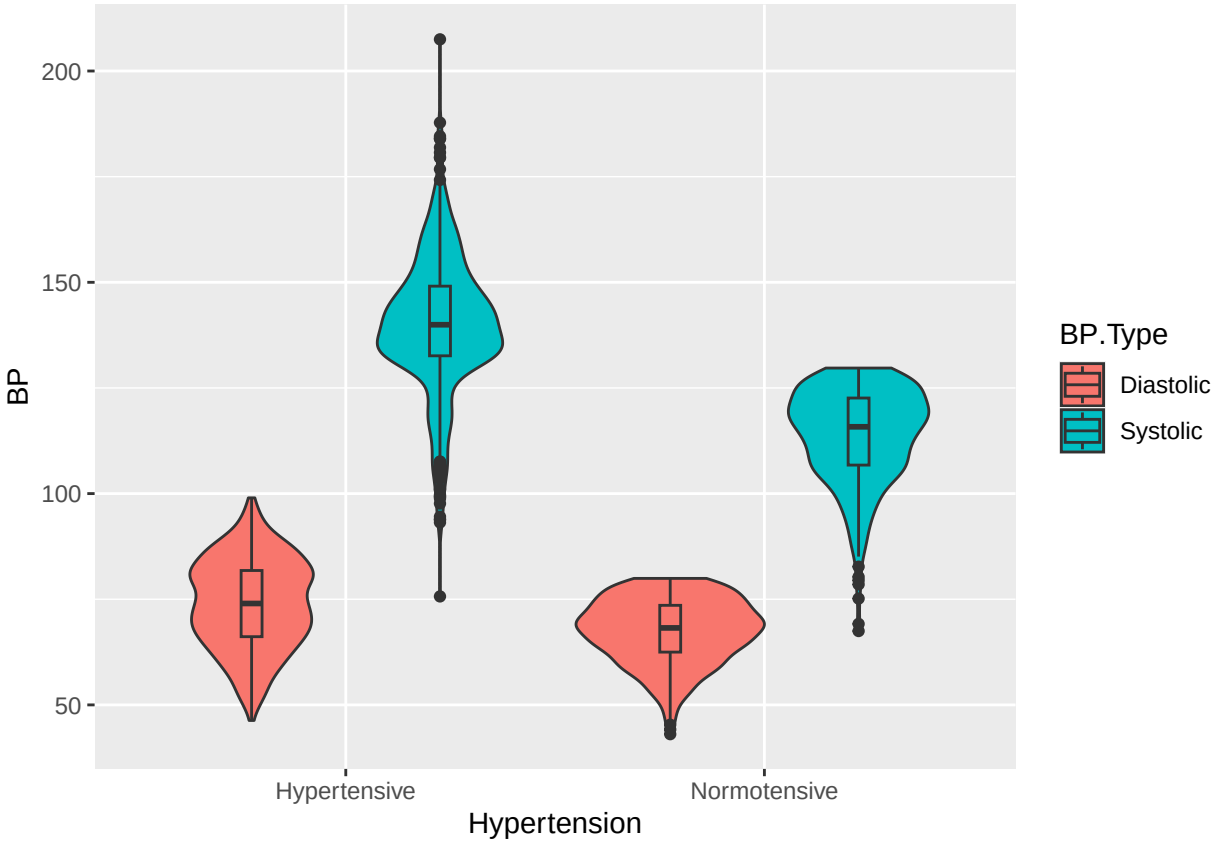



```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Hypertension,y = BP,color = BP.Type)) +
  # Add violin plot
  geom_violin()
```

We haven't looked at violin plots before, how do you think they are summarizing the data differently than the boxplot?

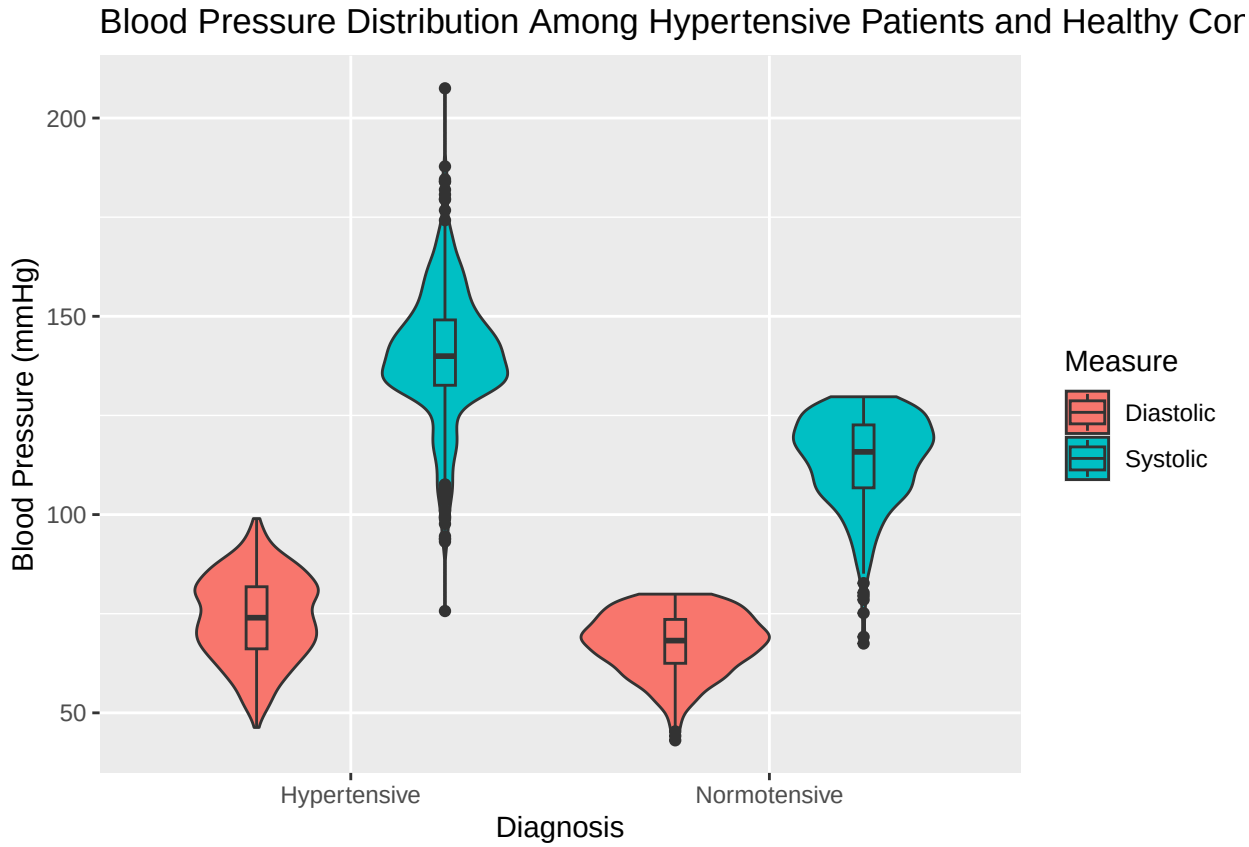
Again, these plots are stackable so we can begin stacking them on top to make a more customized and complex plot.

```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Hypertension,y = BP,fill = BP.Type)) +
  # Add violin plot
  geom_violin() +
  # Add boxplot
  geom_boxplot(width = 0.1,position = position_dodge(0.9))
```



We can also further specify labels for our plot by using the *labs()* function like this:

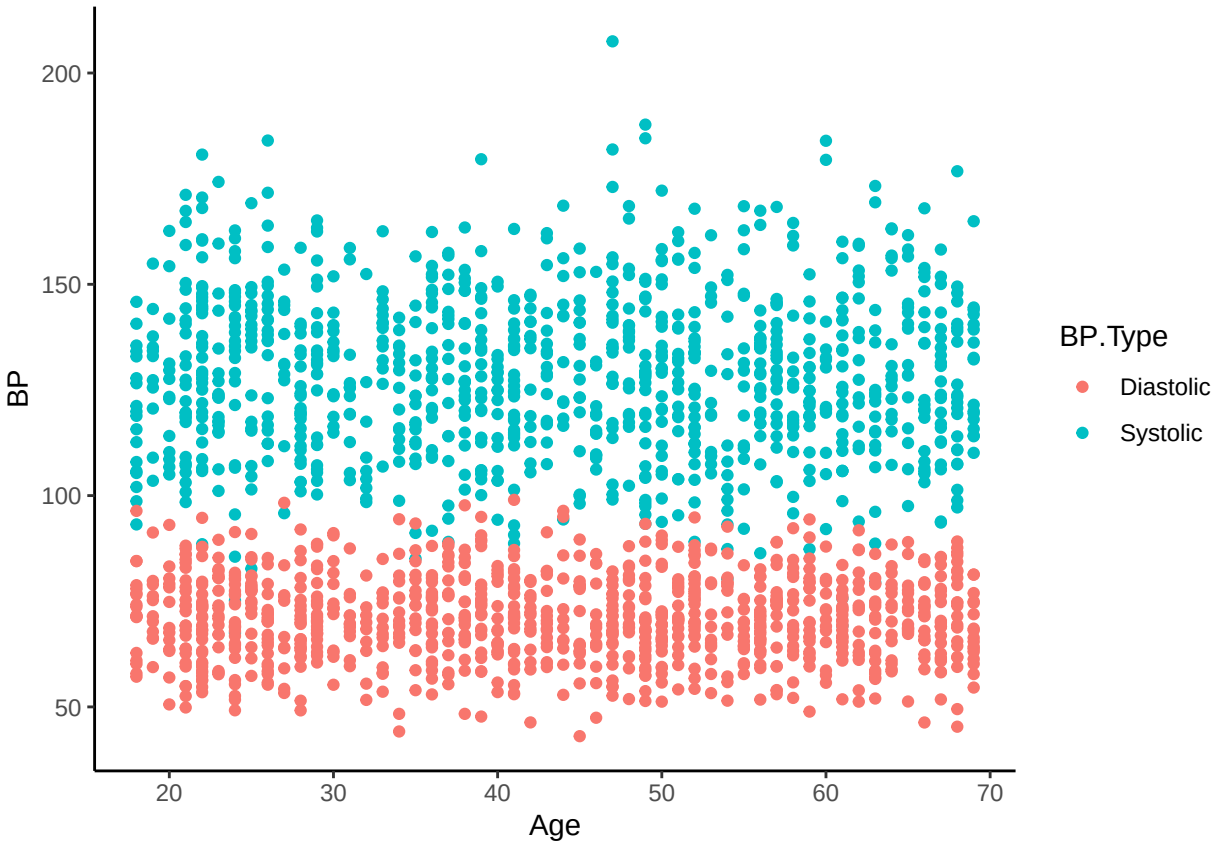
```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Hypertension,y = BP,fill = BP.Type)) +
  # Add violin plot
  geom_violin() +
  geom_boxplot(width = 0.1,position = position_dodge(0.9)) +
  labs(title = "Blood Pressure Distribution Among Hypertensive Patients and Healthy Controls",
       x = "Diagnosis",y = "Blood Pressure (mmHg)",fill = "Measure")
```



Using Themes in *ggplot2*

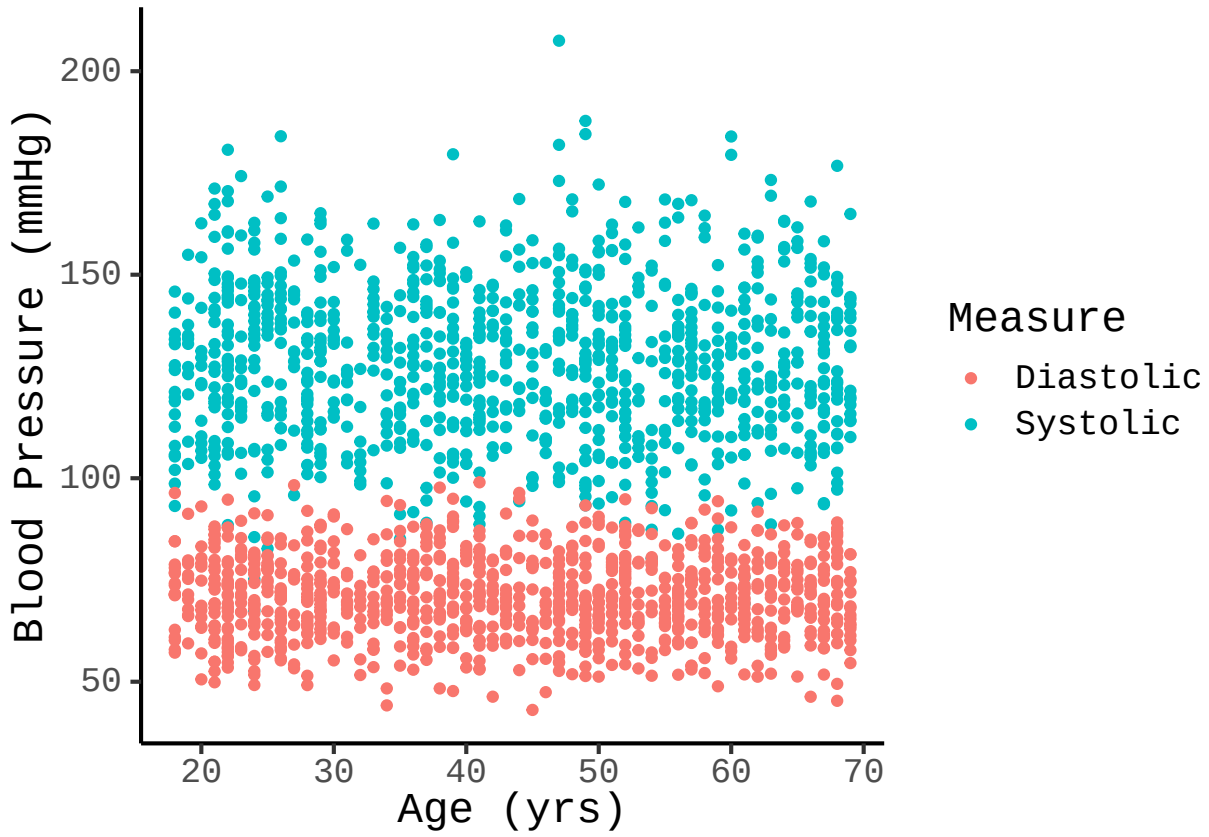
The default format for *ggplot2* is to have a grey background with grid lines. We can completely customize this, however using themes. There are many pre-existing themes that we can simply add to our plot as another layer like this:

```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add points to our scatter plot
  geom_point() +
  # Define theme as basic
  theme_classic()
```



You can find a list of themes and descriptions linked in the resources at the beginning of this lecture. You can also further customize pre-made themes like this:

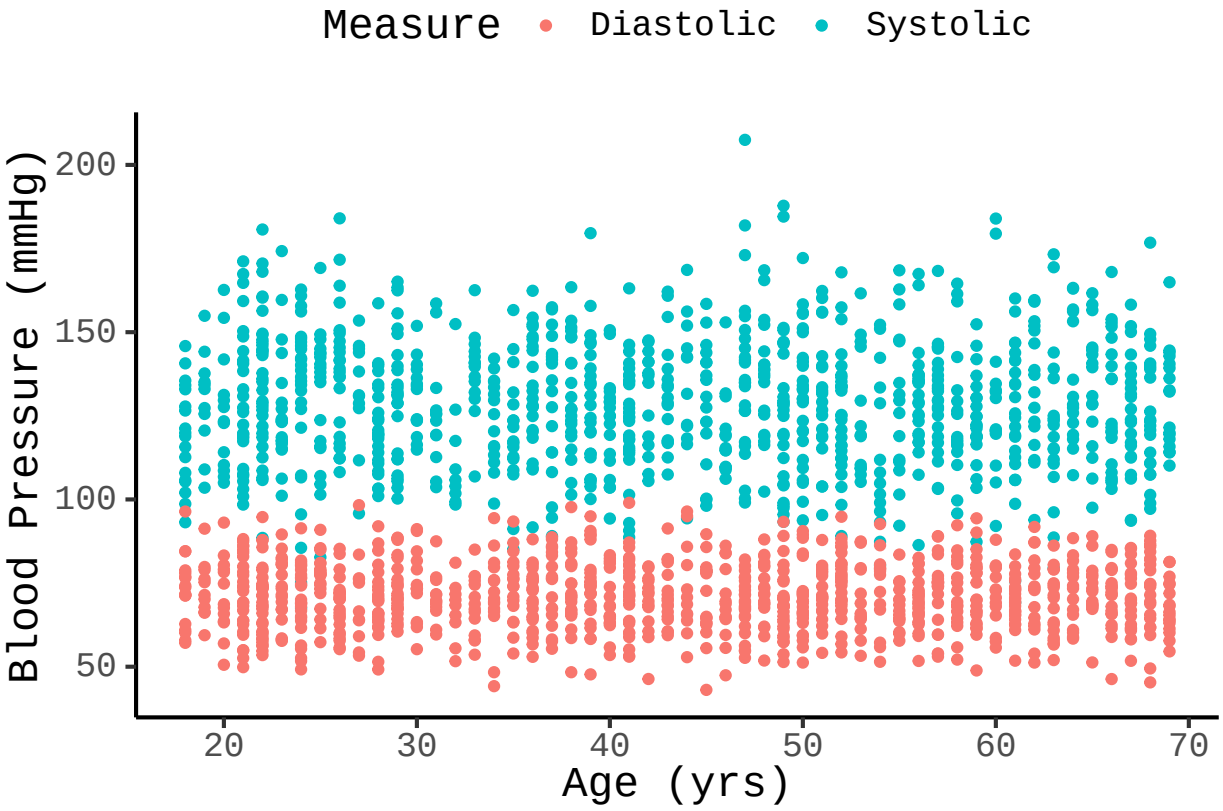
```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add points to our scatter plot
  geom_point() +
  # Change labels
  labs(x = 'Age (yrs)',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Define theme as basic but with different font
  theme_classic(base_family = 'Courier',base_size = 16)
```



Note: changing the font size like this didn't make all text the same size, it simply set a new minimum text size for the plot and scaled everything else up accordingly.

We can also change the legend using themes.

```
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add points to our scatter plot
  geom_point() +
    # Change labels
  labs(x = 'Age (yrs)',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Define theme as basic but with different font
  theme_classic(base_family = 'Courier',base_size = 16) +
  # Change legend position
  theme(legend.position = 'top')
```



We can also use the *theme()* function to define our theme entirely from scratch. You can customize essentially ever single aspect of your plot from here but we will just start with a basic example.

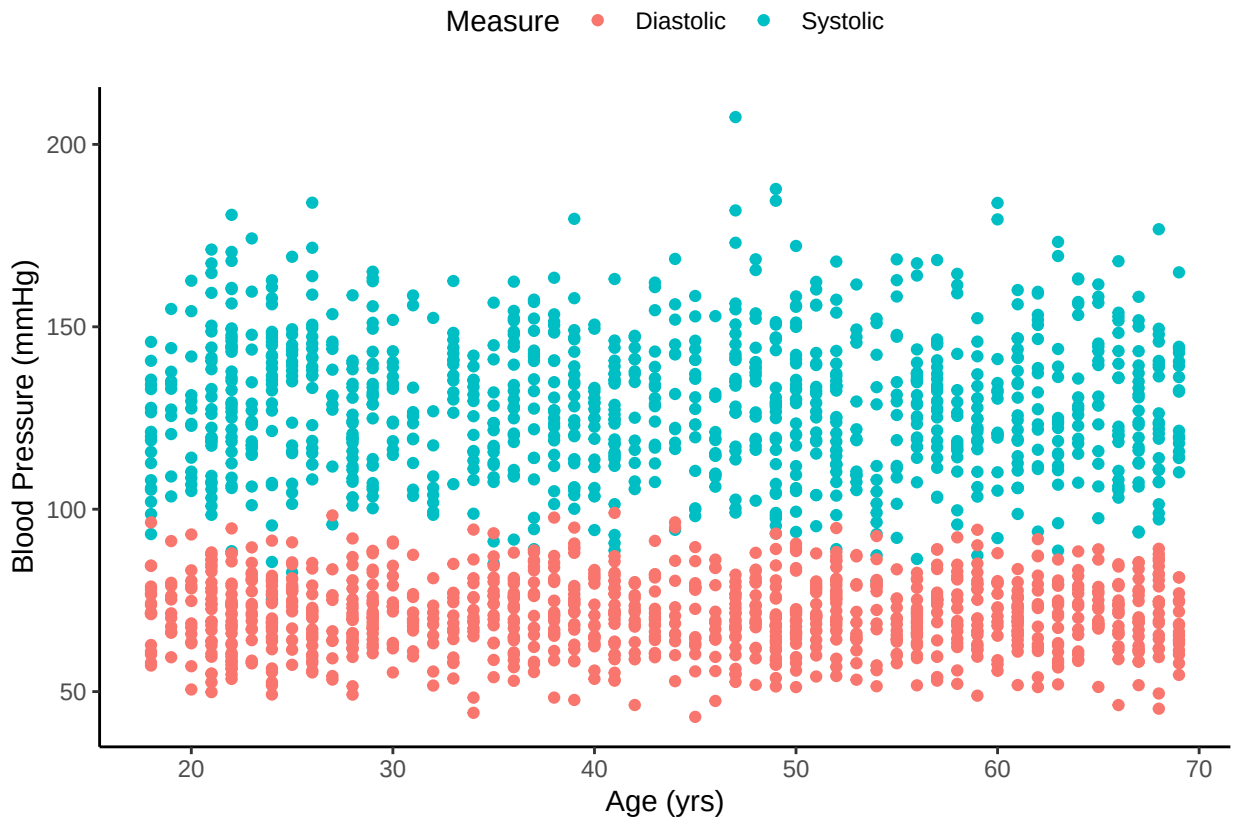
For starters, we can look at how they defined their "classic" theme like this to pull inspiration.

```
theme_classic
```

```
## function (base_size = 11, base_family = "", base_line_size = base_size/22,
##     base_rect_size = base_size/22)
## {
##     theme_bw(base_size = base_size, base_family = base_family,
##         base_line_size = base_line_size, base_rect_size = base_rect_size) %+replace%
##         theme(panel.border = element_blank(), panel.grid.major = element_blank(),
##             panel.grid.minor = element_blank(), axis.line = element_line(colour = "black",
##                 linewidth = rel(1)), legend.key = element_blank(),
##             strip.background = element_rect(fill = "white", colour = "black",
##                 linewidth = rel(2)), complete = TRUE)
## }
## <bytecode: 0x7f7bb1966588>
## <environment: namespace:ggplot2>
```

Note: Any time you call a function without () at the end, it will show you the code that function runs. This can be really helpful when you're trying to understand how some functions work.

```r
# Define overall plot including the data set name and the variables you want to look at within that dat
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add points to our scatter plot
  geom_point() +
    # Change labels
  labs(x = 'Age (yrs)',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Define theme - I'm directly pulling some of the elements that I want to keep from their classic the
  theme(# Remove all the extra borders and grid lines
        panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        # Define my axis
        axis.line = element_line(colour = "black", linewidth = rel(1)),
        # Set plot background
        plot.background = element_rect(fill = "white"),
        panel.background = element_blank(),
        legend.key = element_rect(fill = 'white'),
        # Move legend
        legend.position = 'top')
```

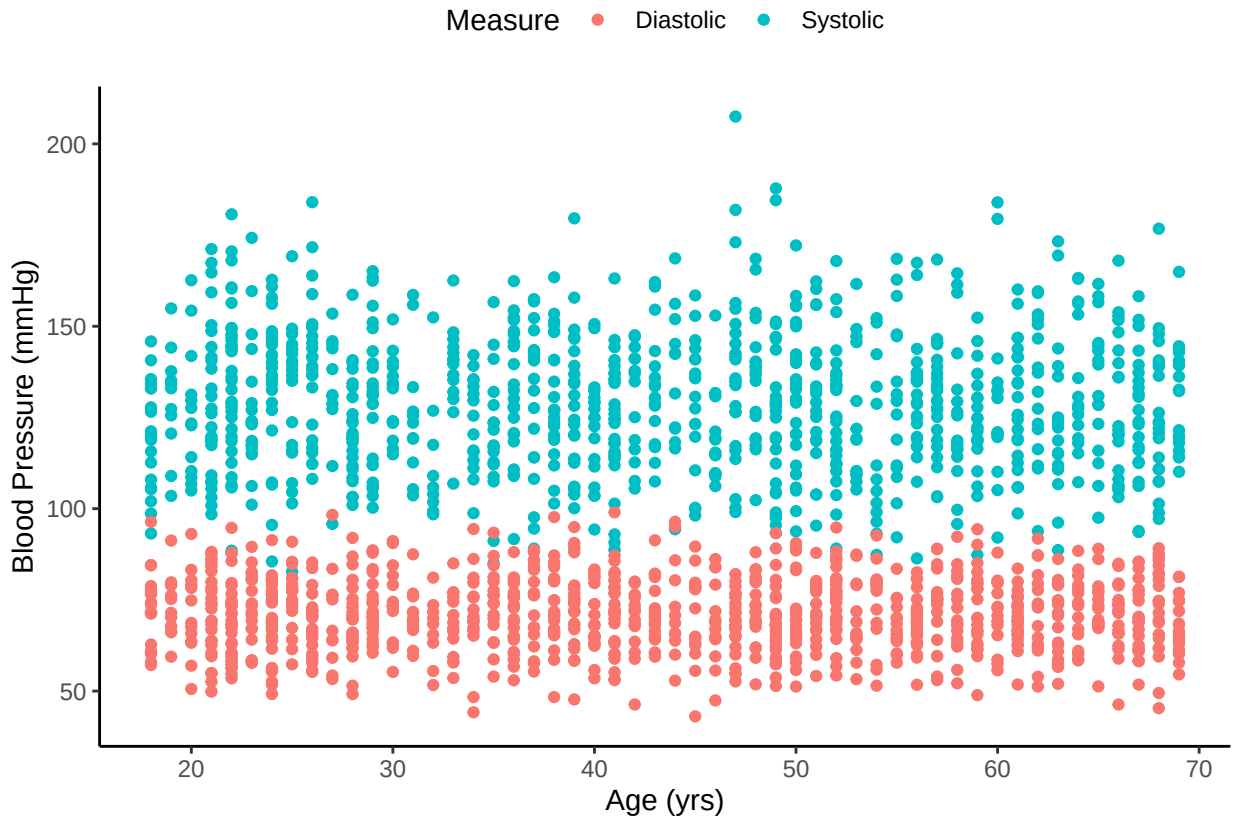


If we don't want to have to type our theme out for every single plot and change it every single time, we can also just define it once and apply it to all plots.

```r
# Define our theme
newBlankTheme <- theme(# Remove all the extra borders and grid lines
        panel.border = element_blank(), panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
```

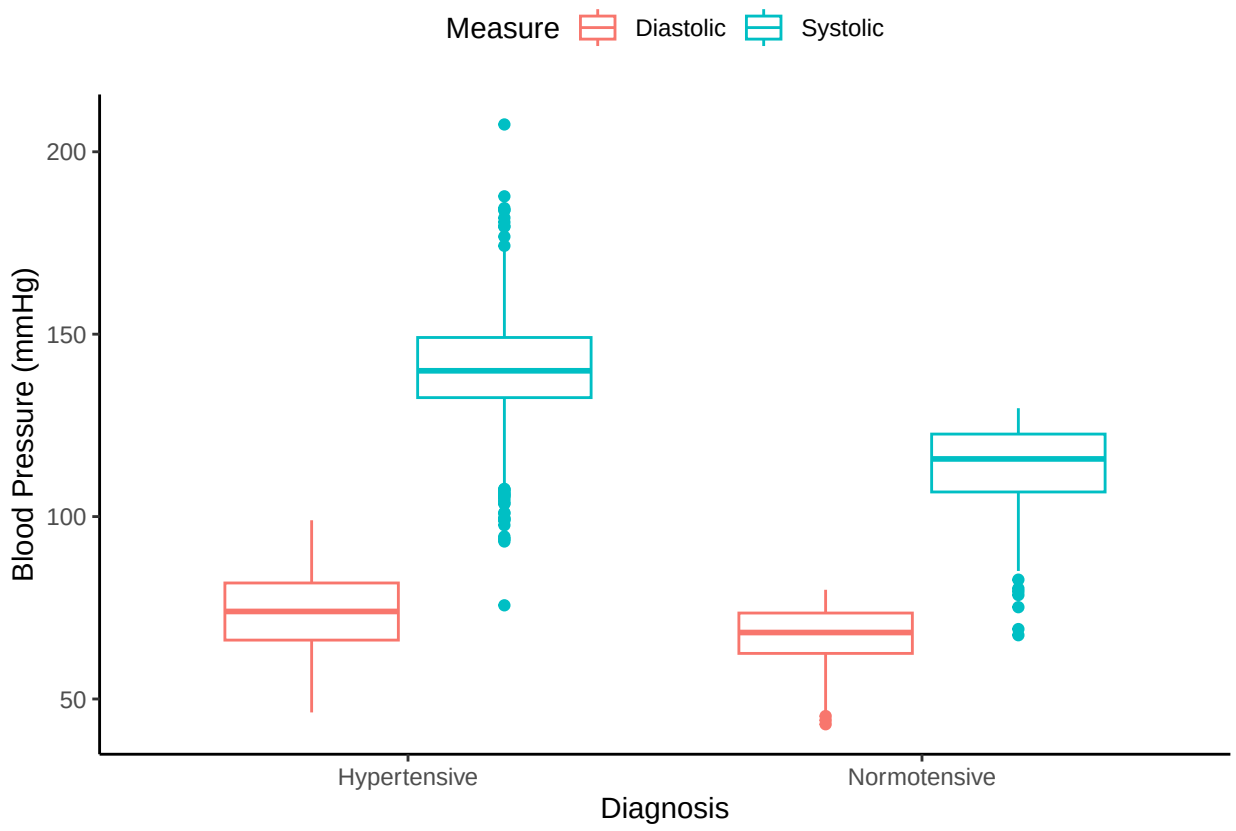
```
        # Define my axis
        axis.line = element_line(colour = "black", linewidth = rel(1)),
        # Set plot background
        plot.background = element_rect(fill = "white"),
        panel.background = element_blank(),
        legend.key = element_rect(fill = 'white'),
        # Move legend
        legend.position = 'top')

# Generate scatter plot using theme
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add points to our scatter plot
  geom_point() +
  # Change labels
  labs(x = 'Age (yrs)',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Set theme
  newBlankTheme
```




```
# Generate boxplot using theme
ggplot(longData,aes(x = Hypertension,y = BP,color = BP.Type)) +
  # Add box plot
  geom_boxplot() +
  # Change labels
  labs(x = 'Diagnosis',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Set theme
```

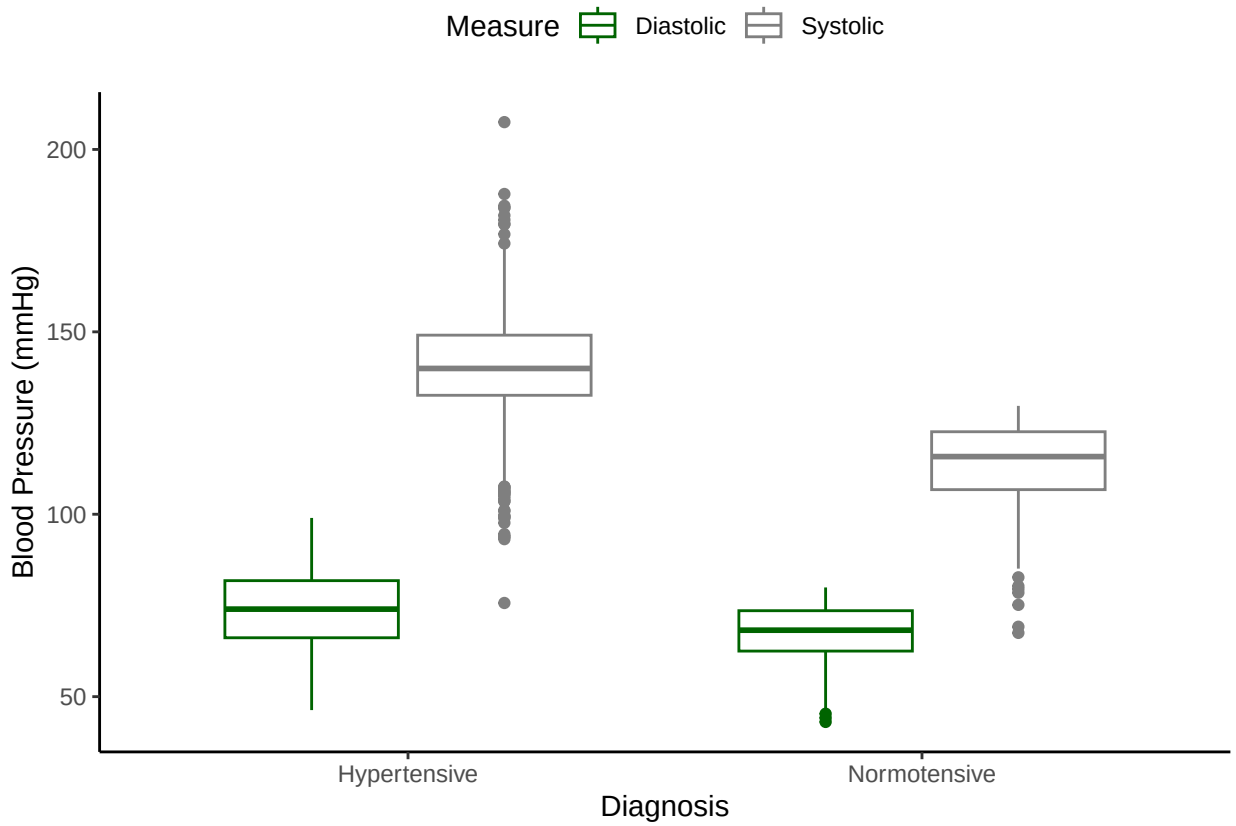
```
  newBlankTheme
```



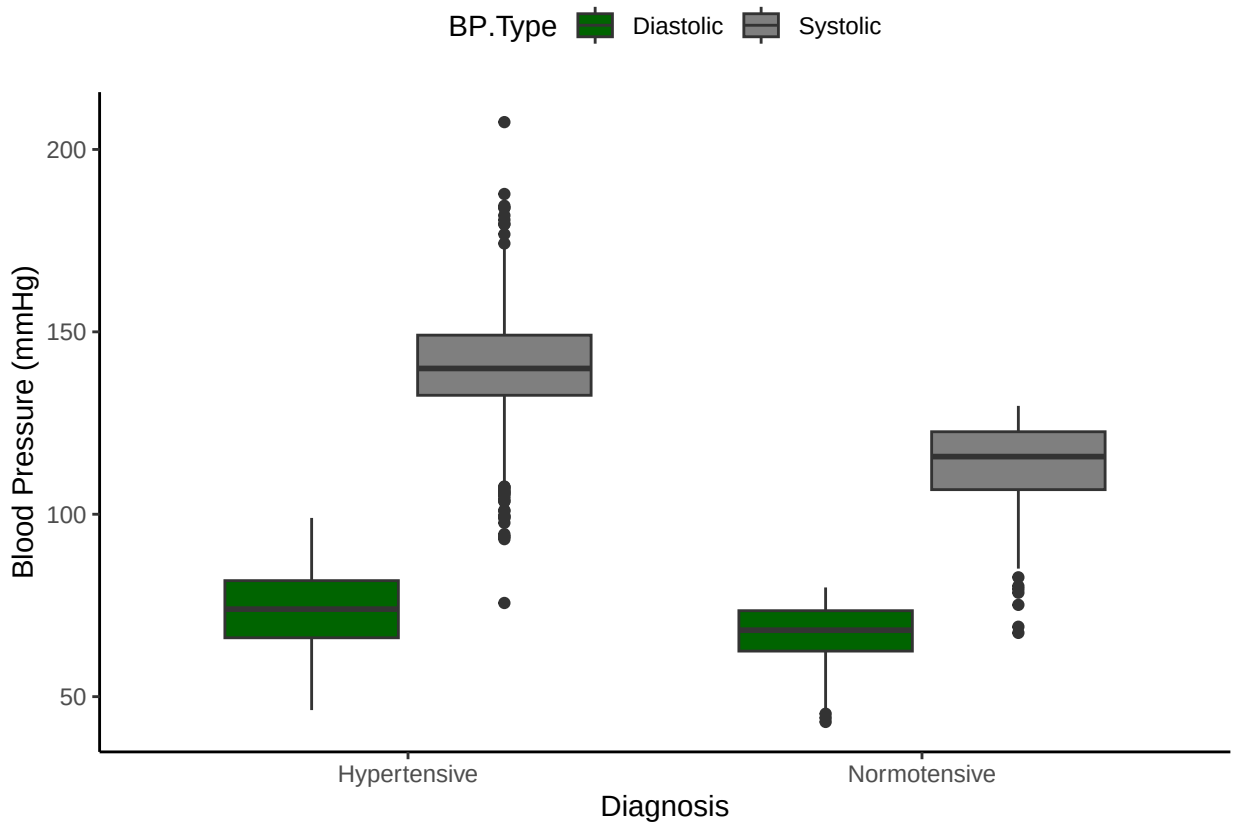
Define Custom Color Palettes

Often, you will want to specify the colors used for your plot. You can do this using the scale_color_manual() and scale_fill_manual() functions.

```
# Generate boxplot
ggplot(longData,aes(x = Hypertension,y = BP,color = BP.Type)) +
  # Add box plot
  geom_boxplot() +
  # Define colors
  scale_color_manual(values = c('darkgreen','grey50')) +
  # Change labels
  labs(x = 'Diagnosis',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Set theme
  newBlankTheme
```

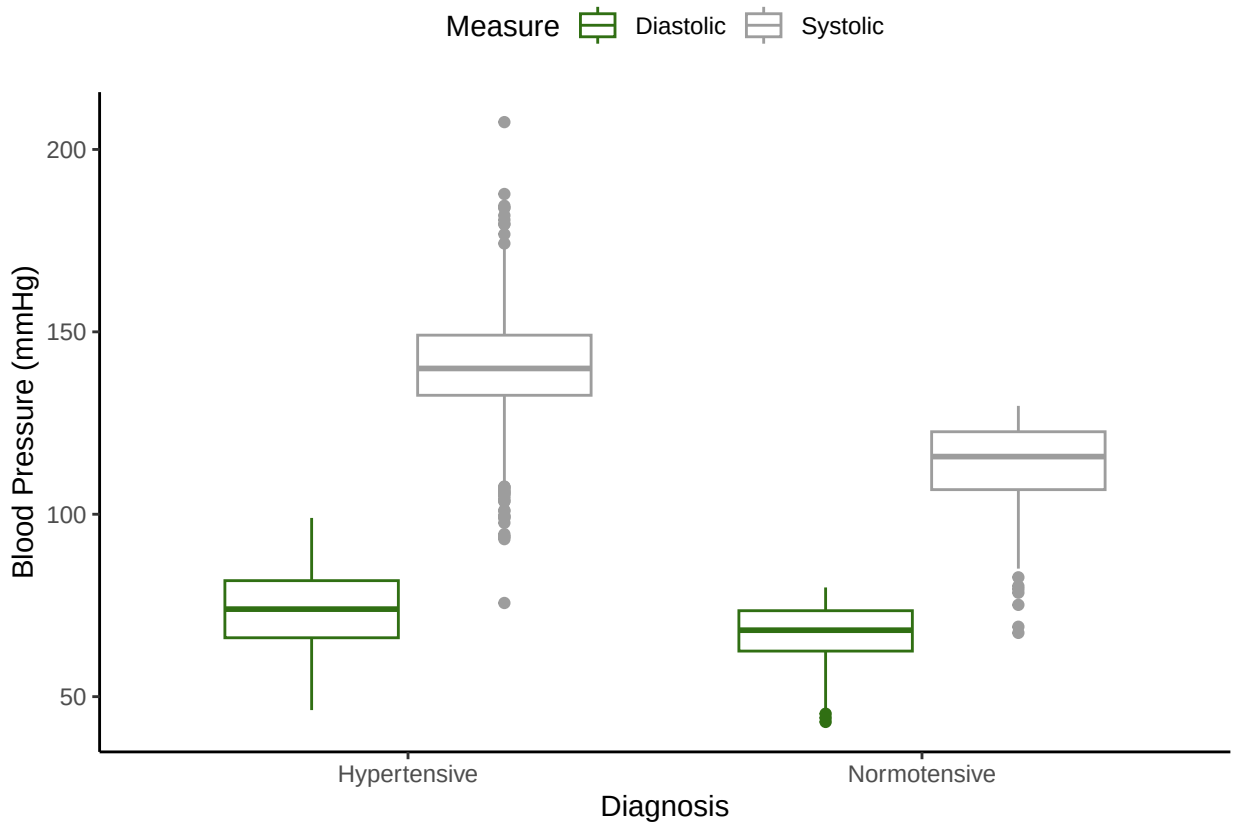



```
# Generate boxplot
ggplot(longData,aes(x = Hypertension,y = BP,fill = BP.Type)) +
  # Add box plot
  geom_boxplot() +
  # Define colors
  scale_fill_manual(values = c('darkgreen','grey50')) +
  # Change labels
  labs(x = 'Diagnosis',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Set theme
  newBlankTheme
```



You can also define colors using hex codes like this:

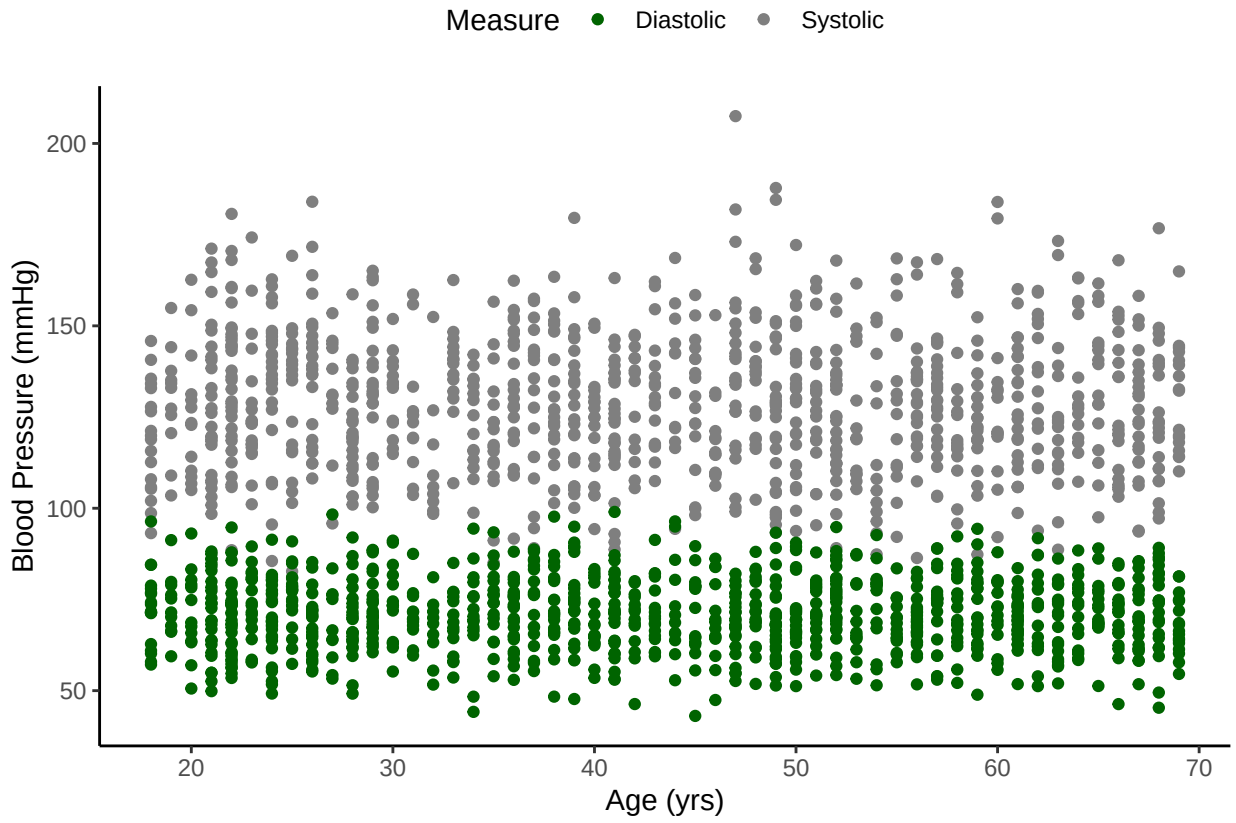
```
# Generate boxplot
ggplot(longData,aes(x = Hypertension,y = BP,color = BP.Type)) +
  # Add box plot
  geom_boxplot() +
  # Define colors
  scale_color_manual(values = c('#306F13','#9D9D9D')) +
  # Change labels
  labs(x = 'Diagnosis',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Set theme
  newBlankTheme
```



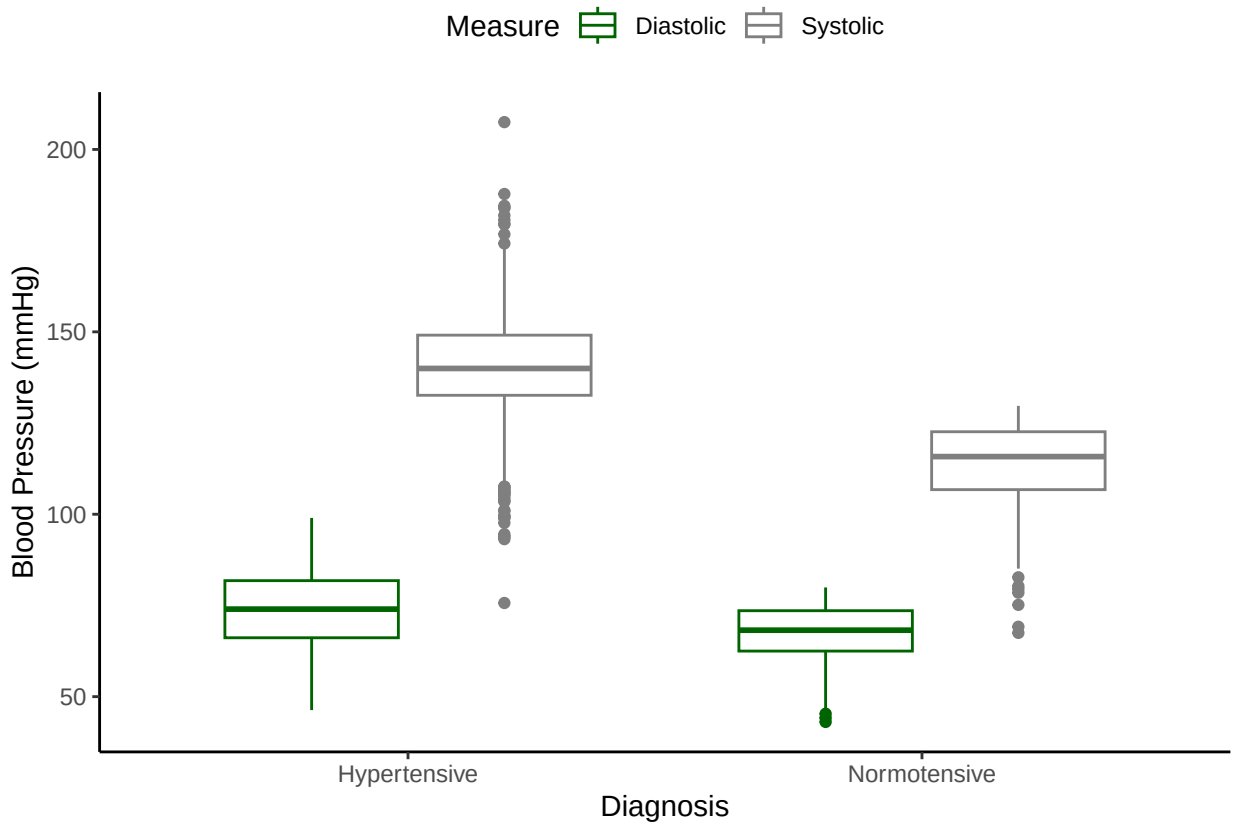
You can also define the palette you want to use if you are going to use it for multiple plots:

```
# Define color palette
colorPalette <- c('darkgreen','grey50')

# Generate scatter plot using theme
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add points to our scatter plot
  geom_point() +
  # Define colors
  scale_color_manual(values = colorPalette) +
  # Change labels
  labs(x = 'Age (yrs)',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Set theme
  newBlankTheme
```




```
# Generate boxplot using theme
ggplot(longData,aes(x = Hypertension,y = BP,color = BP.Type)) +
  # Add box plot
  geom_boxplot() +
  # Define colors
  scale_color_manual(values = colorPalette) +
  # Change labels
  labs(x = 'Diagnosis',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Set theme
  newBlankTheme
```

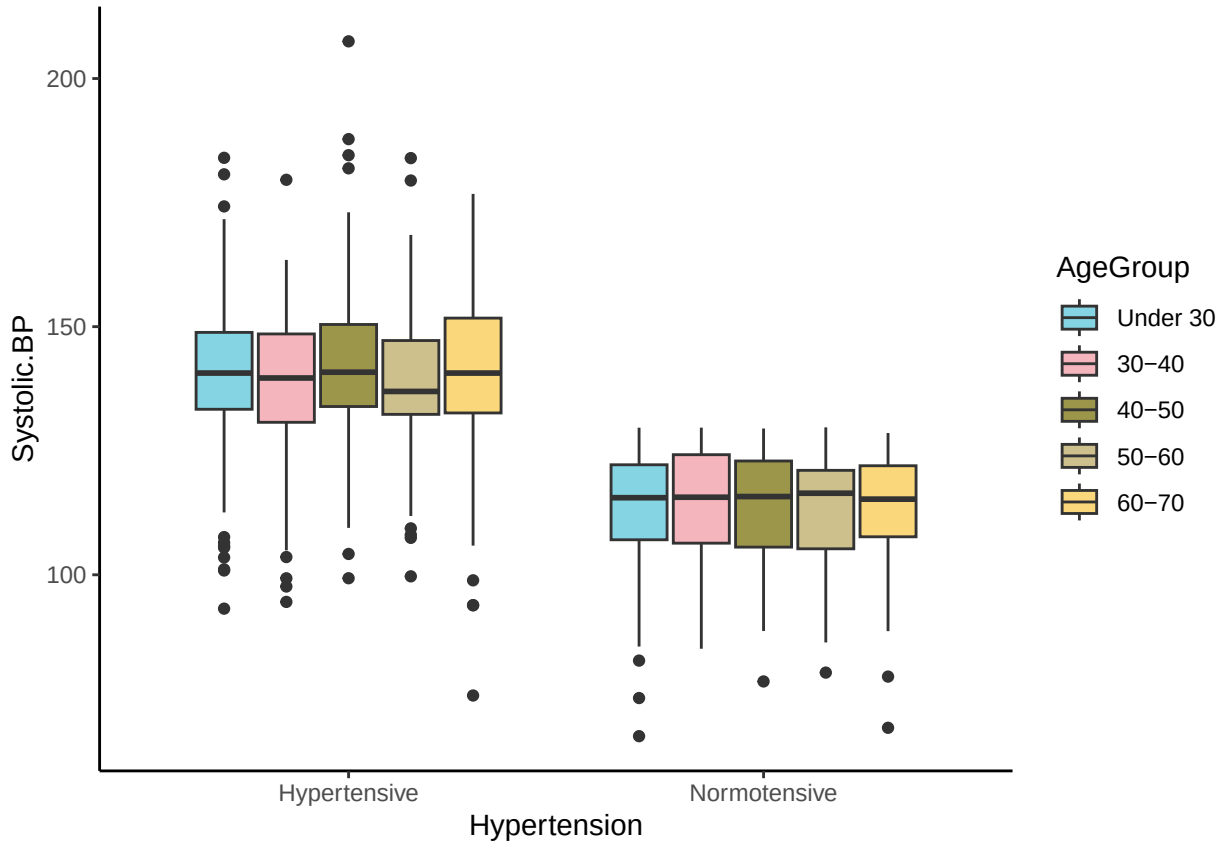


There are also lots of packages with curated color palettes that you can look up online which are particularly great when you have variables with more than 2 levels and don't want to manually define your own. Some fun examples:

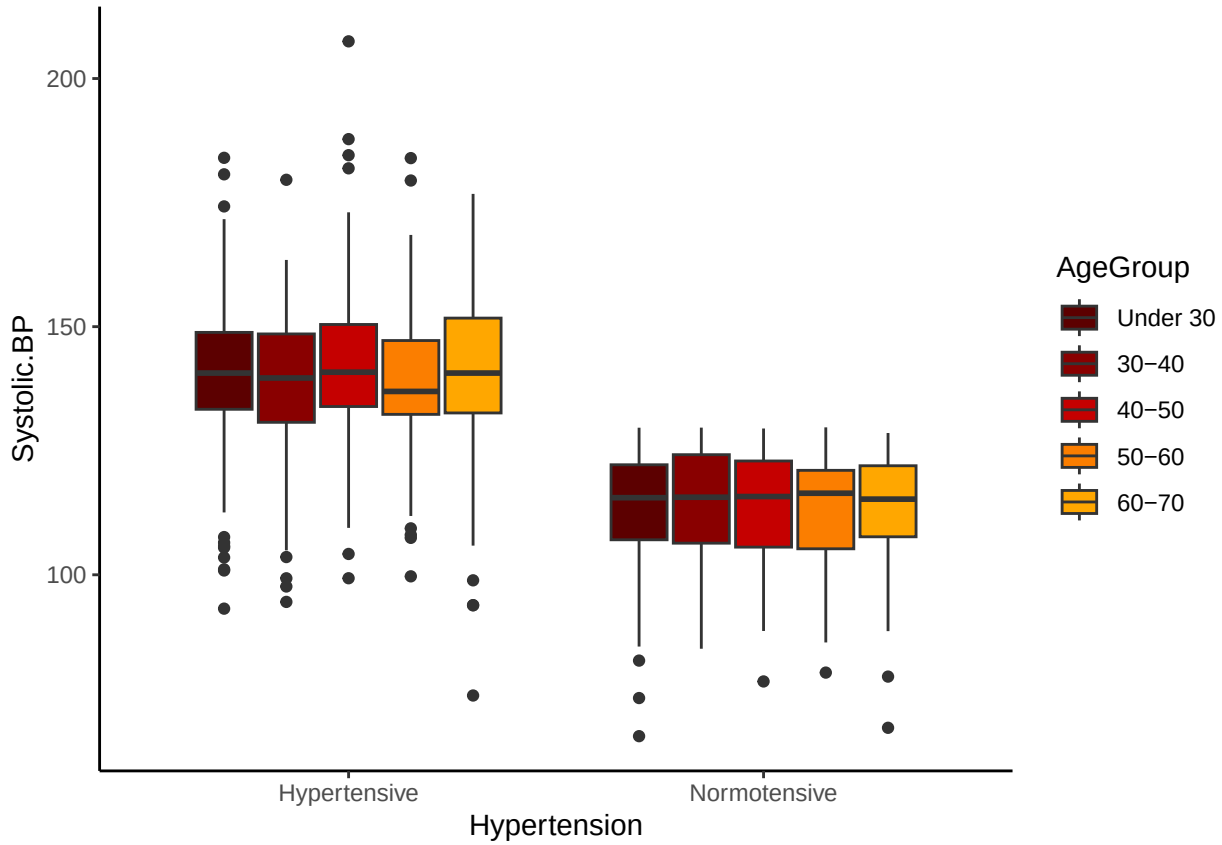
```
# Make more categories in our data
randomData$AgeGroup <- cut(randomData$Age,breaks = c(0,30,40,50,60,70),
                           labels = c('Under 30','30-40','40-50','50-60','60-70'))

# Wes Anderson Palettes
#install.packages('wesanderson')

colorPalette <- wesanderson::wes_palette('Moonrise3',n = 5)
# Generate boxplot
ggplot(randomData,aes(x = Hypertension,y = Systolic.BP,fill = AgeGroup)) +
  # Add boxplot
  geom_boxplot() +
  # Add theme
  theme_classic() +
  # Add colors
  scale_fill_manual(values = colorPalette)
```




```
# Harry Potter Palettes
#install.packages('harrypotter')
colorPalette <- harrypotter::harrypotter(n = 5,house = 'Gryffindor')
# Generate boxplot
ggplot(randomData,aes(x = Hypertension,y = Systolic.BP,fill = AgeGroup)) +
  # Add boxplot
  geom_boxplot() +
  # Add theme
  theme_classic() +
  # Add colors
  scale_fill_manual(values = colorPalette)
```

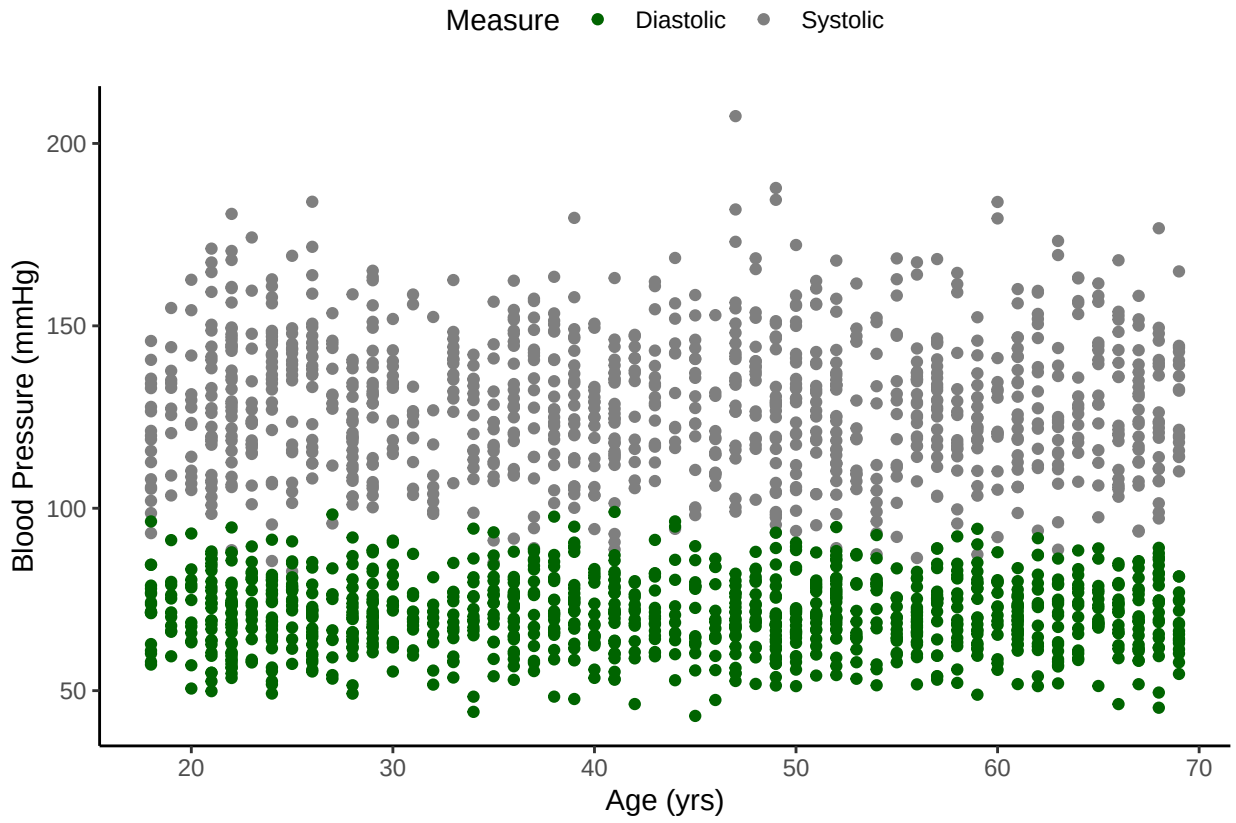


Add Annotations to Plots

We can use a variety of annotation functions to further customize our plots. These typically require specifying where in the plot you want them placed so it's good to have a sense of what your plot looks like before hand. Say we wanted to add annotation on the plot to signify systolic and diastolic BPs.

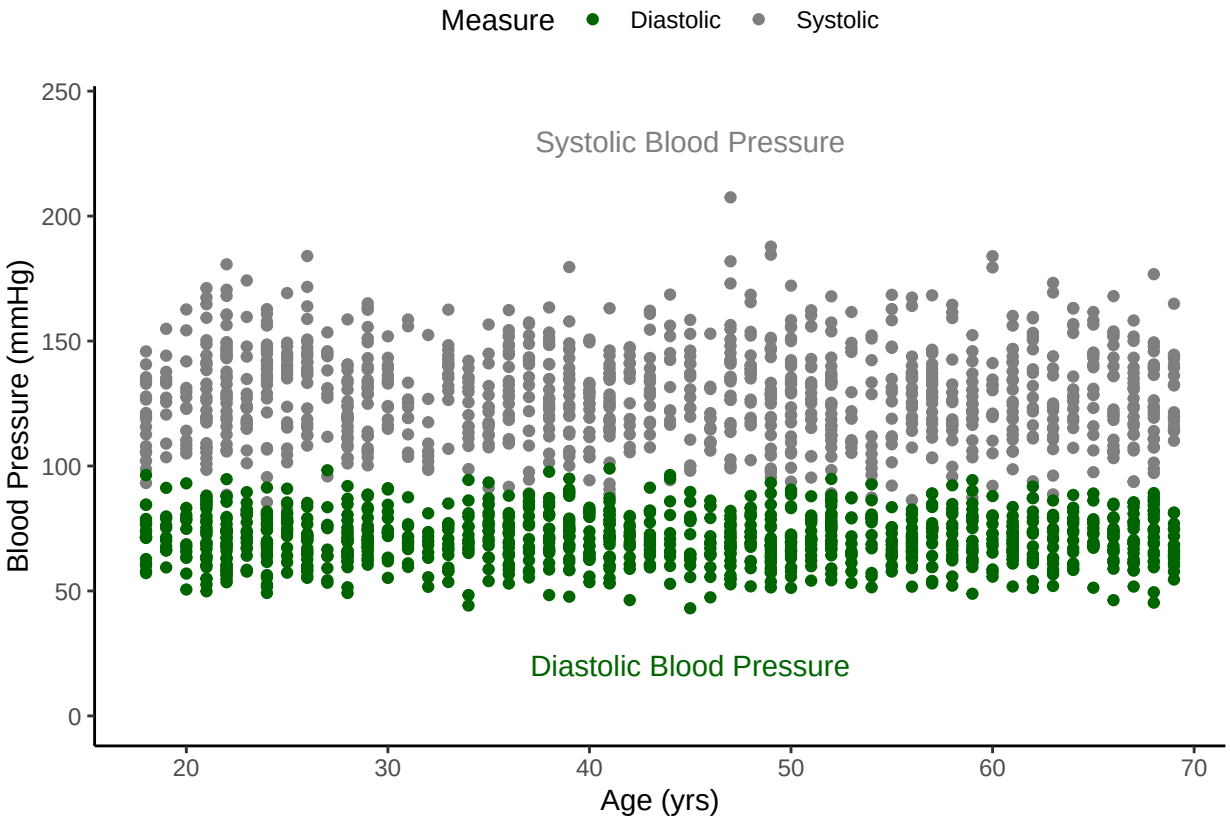
```
# Define color palette
colorPalette <- c('darkgreen','grey50')

# Generate scatter plot using theme
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add points to our scatter plot
  geom_point() +
  # Define colors
  scale_color_manual(values = colorPalette) +
  # Change labels
  labs(x = 'Age (yrs)',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Set theme
  newBlankTheme
```




```
# Generate scatter plot using theme
ggplot(longData,aes(x = Age,y = BP,color = BP.Type)) + # The + is how we communicate to ggplot2 that we
  # Add points to our scatter plot
  geom_point() +
  # Define colors
  scale_color_manual(values = colorPalette) +
  # Change labels
  labs(x = 'Age (yrs)',y = 'Blood Pressure (mmHg)',color = 'Measure') +
  # Set theme
  newBlankTheme +

  # Change scale of y axis to give us more space
  ylim(0,240) +
  # Add text labeling systolic bp
  annotate(geom = 'text',x = 45,y = 230,label = 'Systolic Blood Pressure',color = 'grey50') +
  # Add text labeling diastolic bp
  annotate(geom = 'text',x = 45,y = 20,label = 'Diastolic Blood Pressure',color = 'darkgreen')
```


Measure
Diastolic
Systolic
Systolic Blood Pressure
Diastolic Blood Pressure
Blood Pressure (mmHg)
250
200
150
100
50
0
20
30
40
50
60
70
Age (yrs)