Making Figures with ggplot2

Dr. Sarah Hunter

Introduction to the ggplot2 package

The ggplot2 package was designed to create professional-looking figures in R. This package uses slightly different syntax than the base R plotting functions. This document is designed to introduce you to basicggPlot figures and syntax. ggPlot figures are much more elaborate and flexible than those created using base R. You have more control over the asthetics of your figure using ggplot2.

The Basic Syntax

ggPlot2 syntax is somewhat complicated and uses a different language than base R. However, all types of figures use the same syntax. Once you understand the basics, you can create any type of figure. The basic ggplot command is just ggplot(). Inside this command, you should indicate the dataframe and variables that you will be including in your figure. For this document, I will be using the world dataset from the poliscidata library in R (see Pollock and Edwards (2020)).

```
#load the poliscidata library
library(poliscidata)
```

```
## Registered S3 method overwritten by 'gdata':
##   method         from
##   reorder.factor gplots
```

```
#loading the world data
worlddata<-poliscidata::world

#make sure it loaded correctly
head(worlddata)
```

```
##       country gini10    dem_level4 dem_rank14 dem_score14 lifeex_f lifeex_m
## 1 Afghanistan   29.4 Authoritarian        151        2.77    45.25    44.79
## 2     Albania   33.0        Hybrid         88        5.67    80.30    74.82
## 3     Algeria   35.3 Authoritarian        117        3.83    76.31    72.78
## 4      Angola   58.6 Authoritarian        133        3.35    39.83    37.74
## 5   Argentina   48.8    Part Democ         52        6.84    80.36    73.71
## 6     Armenia   30.2        Hybrid        113        4.13    77.31    69.59
##   literacy     oil pop_0_14 pop_15_64 pop_65_older fertility govregrel
## 1     28.1       0     42.3      55.3          2.4      5.39    10.000
## 2       NA    5400     21.4      68.1         10.5      1.48     0.000
## 3     69.9 2125000     24.2      70.6          5.2      1.75     8.611
## 4     67.4 1948000     43.2      54.1          2.7      5.97     0.556
## 5     97.2  796300     25.4      63.6         11.0      2.31     0.000
## 6     99.4       0     17.6      72.4         10.1      1.37     6.944
##                  regionun          religoin spendeduc spendhealth spendmil
## 1                    Asia            Muslim        NA         1.8      1.9
## 2                  Europe            Muslim       2.9         2.9      2.0
```

```
## 3                 Africa             Muslim       4.3         3.6      3.0
## 4                 Africa           Catholic       2.6         2.0      3.0
## 5 Latin America/Caribbean           Catholic       4.9         5.1      0.8
## 6                   Asia Orthodox Christian       3.0         2.1      3.3
##     hdi pop_age sexratio pop_total pop_urban gender_unequal gender_unequal_rank
## 1 0.349    16.9    106.0      29.1      22.6          0.797                 134
## 2 0.719    30.0    107.0       3.2      51.9          0.545                  61
## 3 0.677    26.2    104.6      35.4      66.5          0.594                  70
## 4 0.403    17.4     99.9      19.0      58.5             NA                  NA
## 5 0.775    30.4    103.6      40.7      92.4          0.534                  60
## 6 0.695    32.0    116.5       3.1      64.2          0.570                  66
##   arda lifeex_total debt       colony confidence
## 1    1        45.02   NA           UK         NA
## 2    3        77.41 59.3 Soviet Union  49.335926
## 3    4        74.50 25.7       France  52.055735
## 4    7        38.76 20.3     Portugal         NA
## 5   11        76.95 50.3        Spain   7.299325
## 6   12        73.23   NA Soviet Union  27.132735
##                                       decent08 dem_other dem_other5 democ
## 1                             No local elections      10.5        10%    No
## 2    Legislature and executive are locally elected      63.0 Approx 60%   Yes
## 3 Legislature is elected but executive is appointed      40.8 Approx 40%    No
## 4                             No local elections      40.8 Approx 40%    No
## 5                                           <NA>      87.5 Approx 90%   Yes
## 6    Legislature and executive are locally elected      63.0 Approx 60%   Yes
##   democ11 democ_regime democ_regime08    district_size3 durable effectiveness
## 1      NA           No             No     single member       4      13.71158
## 2       9          Yes            Yes              <NA>       3      35.46099
## 3       3           No             No 6 or more members       5      32.62411
## 4       2           No             No              <NA>       3      19.14894
## 5       8          Yes            Yes 6 or more members      17      34.98818
## 6       5          Yes            Yes              <NA>       2      36.64303
##    enpp3_democ enpp3_democ08 dnpp_3         eu fhrate04_rev fhrate08_rev
## 1         <NA>          <NA>     NA Not member          2.5            3
## 2  1-3 parties   1-3 parties      1 Not member          5.0            8
## 3         <NA>          <NA>      3 Not member          2.5            3
## 4         <NA>          <NA>      1 Not member          2.5            3
## 5  1-3 parties   1-3 parties      1 Not member          6.0           10
## 6 6-11 parties  6-11 parties      3 Not member          3.5            4
##   frac_eth frac_eth2 frac_eth3 free_business free_corrupt free_finance
## 1   0.7693      High      High            NA           NA           NA
## 2   0.2204       Low       Low          68.0           34           70
## 3   0.3394       Low    Medium          71.2           32           30
## 4   0.7867      High      High          43.4           19           40
## 5   0.2550       Low       Low          62.1           29           30
## 6   0.1272       Low       Low          83.4           29           70
##   free_fiscal free_govspend free_invest free_labor free_monetary free_property
## 1          NA            NA          NA         NA            NA            NA
## 2        92.6          74.2          70       52.1          78.7            35
## 3        83.5          73.4          45       56.4          77.2            30
## 4        85.1          62.8          35       45.2          62.6            20
## 5        69.5          75.6          45       50.1          61.2            20
## 6        89.3          90.9          75       70.6          72.9            30
##   free_trade free_overall free_overall_4 gdp08 gdp_10_thou gdp_cap2 gdp_cap3
```

```
## 1         NA           NA           <NA>  30.6          NA     <NA>     <NA>
## 2       85.8         66.0          MidHi  24.3      0.1535      Low   Middle
## 3       70.7         56.9         MidLow 276.0      0.1785      Low   Middle
## 4       70.4         48.4            Low 106.3      0.0857      Low   Middle
## 5       69.5         51.2            Low 571.5      0.2797     High   Middle
## 6       80.5         69.2           High  18.7      0.0771      Low      Low
##   gdpcap2_08 gdpcap3_08 gdpcap08_2 gdppcap08 gdppcap08_3 gender_equal3 gini04
## 1        Low        Low        Low        NA          NA          <NA>     NA
## 2        Low        Mid        Low      7715           2          <NA>   28.2
## 3       High        Mid       High      8033           2          <NA>   35.3
## 4       High        Mid       High      5899           2          <NA>     NA
## 5       High       High       High     14333           3          High   52.2
## 6        Low        Mid        Low      6070           2          <NA>   37.9
##   gini08   hi_gdp indy muslim     natcode       oecd      pmat12_3 polity
## 1     NA     <NA> 1919    Yes afghanistan Not member          <NA>     NA
## 2   31.1  Low GDP 1991    Yes     albania Not member  Low post-mat      9
## 3   35.3  Low GDP 1962    Yes     algeria Not member          <NA>      2
## 4     NA  Low GDP 1975     No      angola Not member          <NA>     -2
## 5   51.3 High GDP 1816     No   argentina Not member High post-mat      8
## 6   33.8  Low GDP 1991     No     armenia Not member  Low post-mat      5
##   pr_sys protact3        regime_type3 rich_democ unions unnetgro unnetuse
## 1     No     <NA>        Dictatorship         NA     NA       NA      1.7
## 2     No Moderate Parliamentary democ          0     NA    21329     23.9
## 3    Yes     <NA>        Dictatorship          0     NA     2633     11.9
## 4    Yes     <NA>        Dictatorship          0     NA     3567      3.1
## 5    Yes Moderate  Presidential democ          1   25.4      331     28.1
## 6     No     High                <NA>          0     NA      378      6.2
##   unpovnpl unremitp unremitt vi_rel3 votevap00s votevap90s women05 women09
## 1     42.0       NA       NA    <NA>         NA         NA      NA      NA
## 2     18.5      476     12.2  20-50%      59.56   85.25755     6.4    16.4
## 3       NA       64      1.3    >50%         NA   71.43356      NA      NA
## 4       NA        5      0.1    <NA>         NA   88.28227      NA      NA
## 5       NA       17      0.2  20-50%      70.88   79.68567    33.7    41.6
## 6     50.9      345      8.9  20-50%         NA   53.32164     5.3     8.4
##   women13 ipu_wom13_all womyear       womyear2 dem_economist democ.yes
## 1      NA          27.7      NA           <NA>             0         0
## 2    15.7          15.7    1920 1944 or before             0       100
## 3      NA          31.6    1962     After 1944             0         0
## 4      NA          34.1    1975     After 1944             0         0
## 5    37.4          37.4    1947     After 1944             1       100
## 6    10.7          10.7    1921 1944 or before             0       100
##      country1
## 1 Afghanistan
## 2     Albania
## 3     Algeria
## 4      Angola
## 5   Argentina
## 6     Armenia
```

After you get the data loaded, you can begin with the basic `ggplot2` command structure. The code below demonstrates how to install, load, and begin a `ggplot2` figure.

```
#install ggplot2
#install.packages("ggplot2")
```

```
#load the package
library(ggplot2)

#The basic command:
#here, you use the ggplot() command, and include the data name (data=)
#and include the variable you want to plot (x=) inse the aes() option,
#each separated with a column
ggplot(data=worlddata, aes(x=women13))
```

Just using the basic `ggplot` command does not give you anything. To get R to give you the figure you want, you need to add the type of figure to the main command, connecting them with a `+` symbol. In this document, I will demonstrate how to make histograms, bar plots, scatterplots, line graphs, and box and whisker plots. These are just basic figures. You can make many, many more types of figures, based on your projects' needs. This document is intended to introduce you to the basics of `ggplot`. The table below shows which command is used for each type of figure:

Type of Figure	Command
Histogram	`geom_historgram()`
Bar plot	`geom_bar()`
Scatterplot	`geom_point()`
Line graph	`geom_line()`
Boxplot	`geom_boxplot()`

Each of the different types of figures comes with a unique set of options. For the rest of this document, I will demonstrate how to make a professional-looking figure for each type.

Histograms

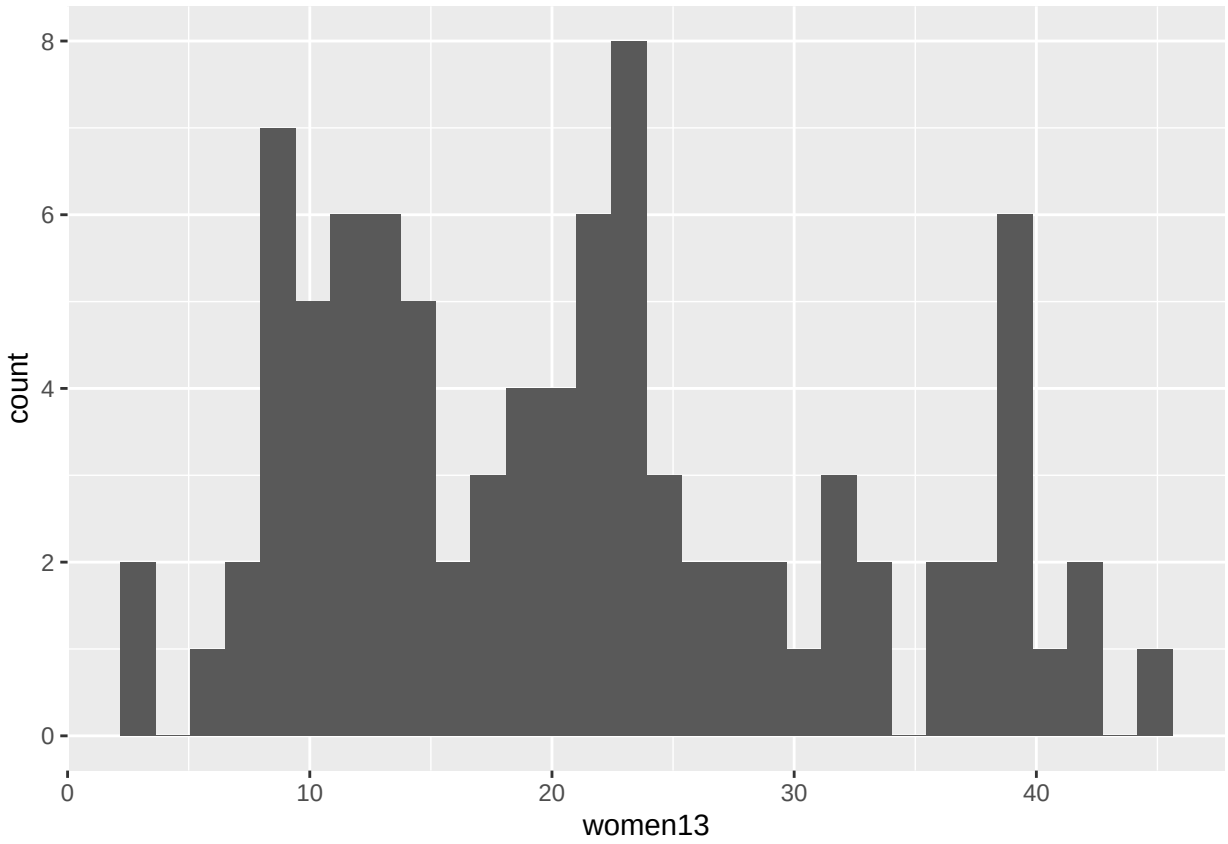
Histograms are great univariate displays for interval/continuous variables. They show the distribution of the variable (i.e. the values of the variable and the number of observations at each value). On the horizontal, or x, axis, there are ranges of values of the varible. On the vertical, or y, axis, there is the frequency, or the number of observations of that variable that fall into the x-axis range. Histograms can give you a sense of the mean, median, mode, and skewness of a variable.

To make a histogram using the `ggplot2` package, you will need to use the `geom_histogram()` option. The code below shows how to make a basic histogram with `ggplot2`.

```
#the first command is ggplot() which tells R to open ggplot,
#the aes part tells R what you want to plot
#Then you need to use the + and the type of ggplot you want to use.
ggplot(data=worlddata, aes(x=women13)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 77 rows containing non-finite values (stat_bin).
```



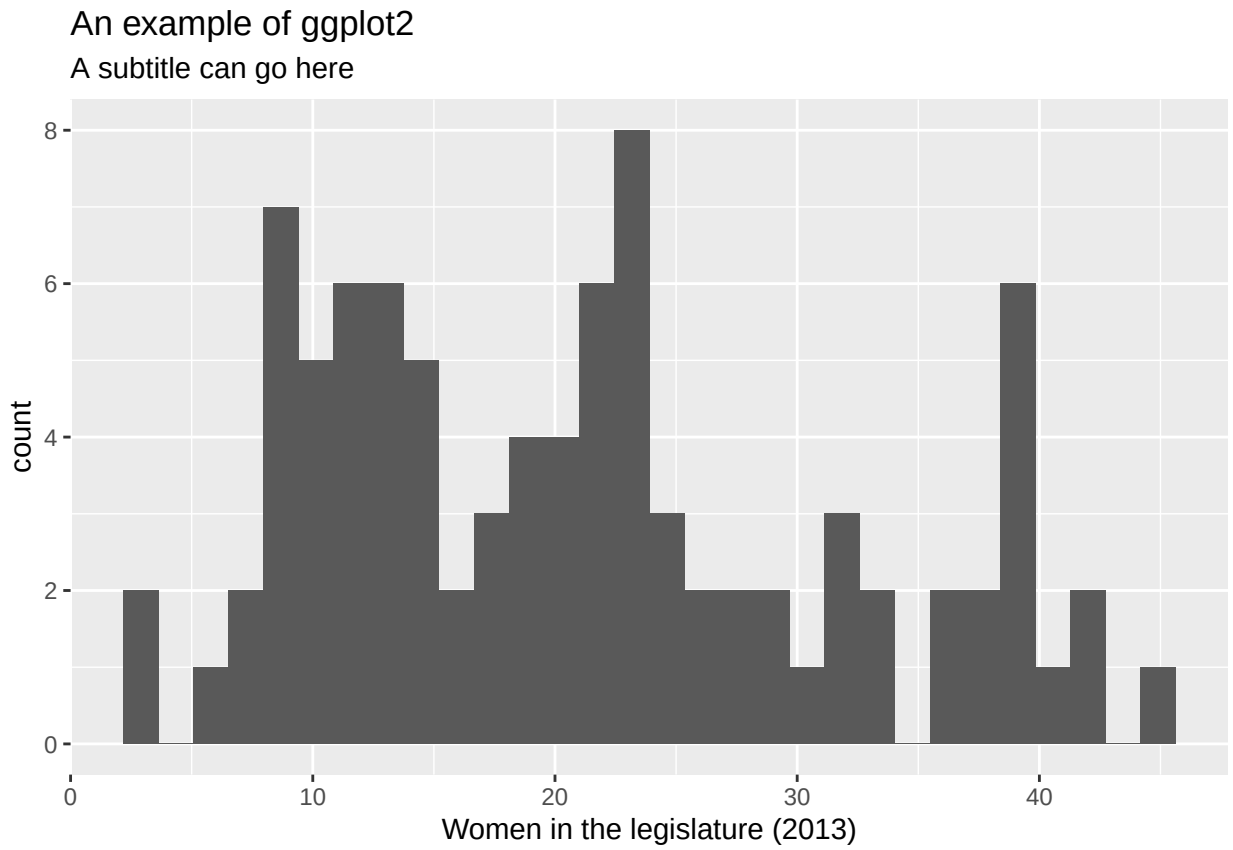
The above figure is just the basic histogram. There are many options that you can use to customize this figure. The most basic option for a ggplot is to create a title. To create the ggplot's title, you simply need to add another element to the plot using the + symbol and the command `ggtitle("Title here")`. To creat the label for the x-axis, you need to again use the + symbol followed by `xlab("Label here")`. The code below demonstrates both options.

```
#And, of course, add labels:

ggplot(data=worlddata, aes(x=women13)) + geom_histogram()+
  ggtitle("An example of ggplot2", subtitle= "A subtitle can go here") +
  xlab("Women in the legislature (2013)")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

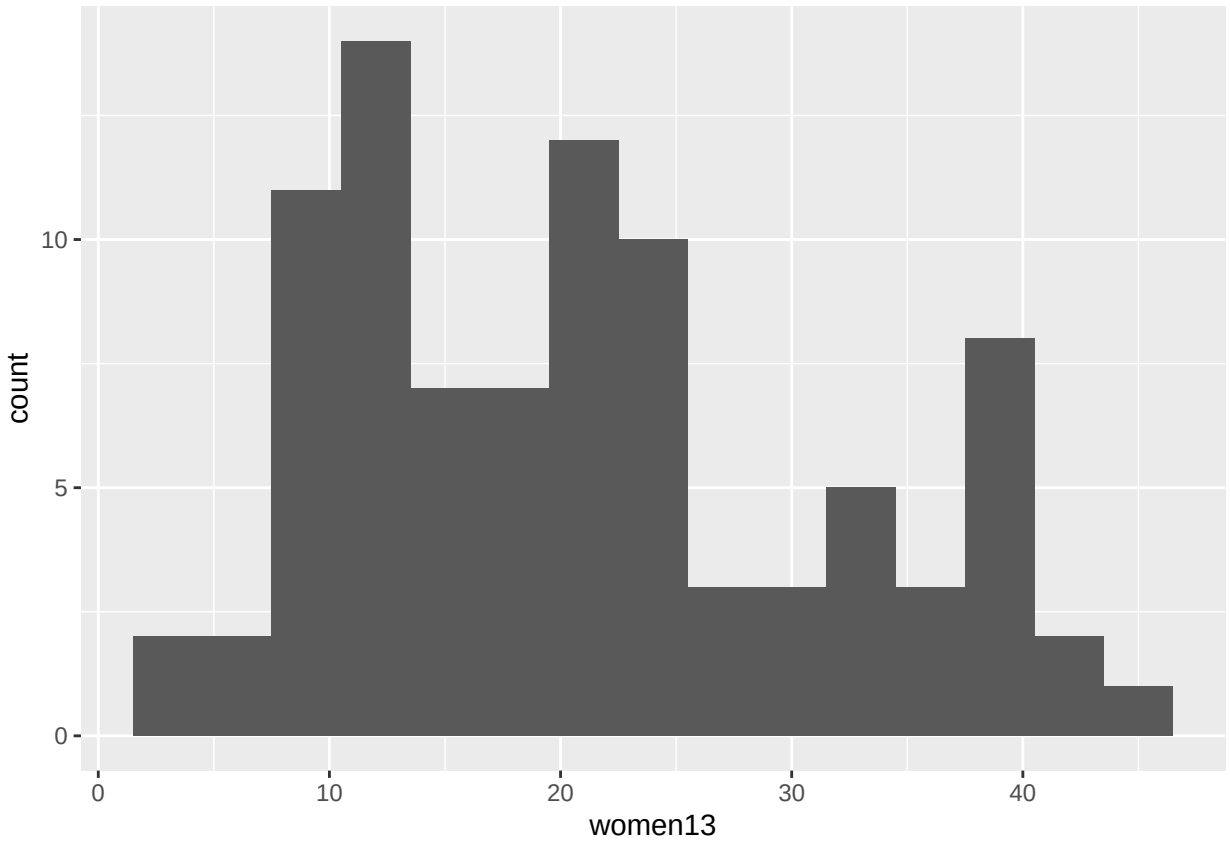
```
## Warning: Removed 77 rows containing non-finite values (stat_bin).
```



The next option you can change is the binwidth (`bins=`) which controls how wide or narrow each bar of the histogram is. The code below changes the binwidth from the default (`bins=30`) to `bins=15`.

```
#Change the size of the bars (bins=)
ggplot(data=worlddata, aes(x=women13)) + geom_histogram(bins=15)
```

```
## Warning: Removed 77 rows containing non-finite values (stat_bin).
```

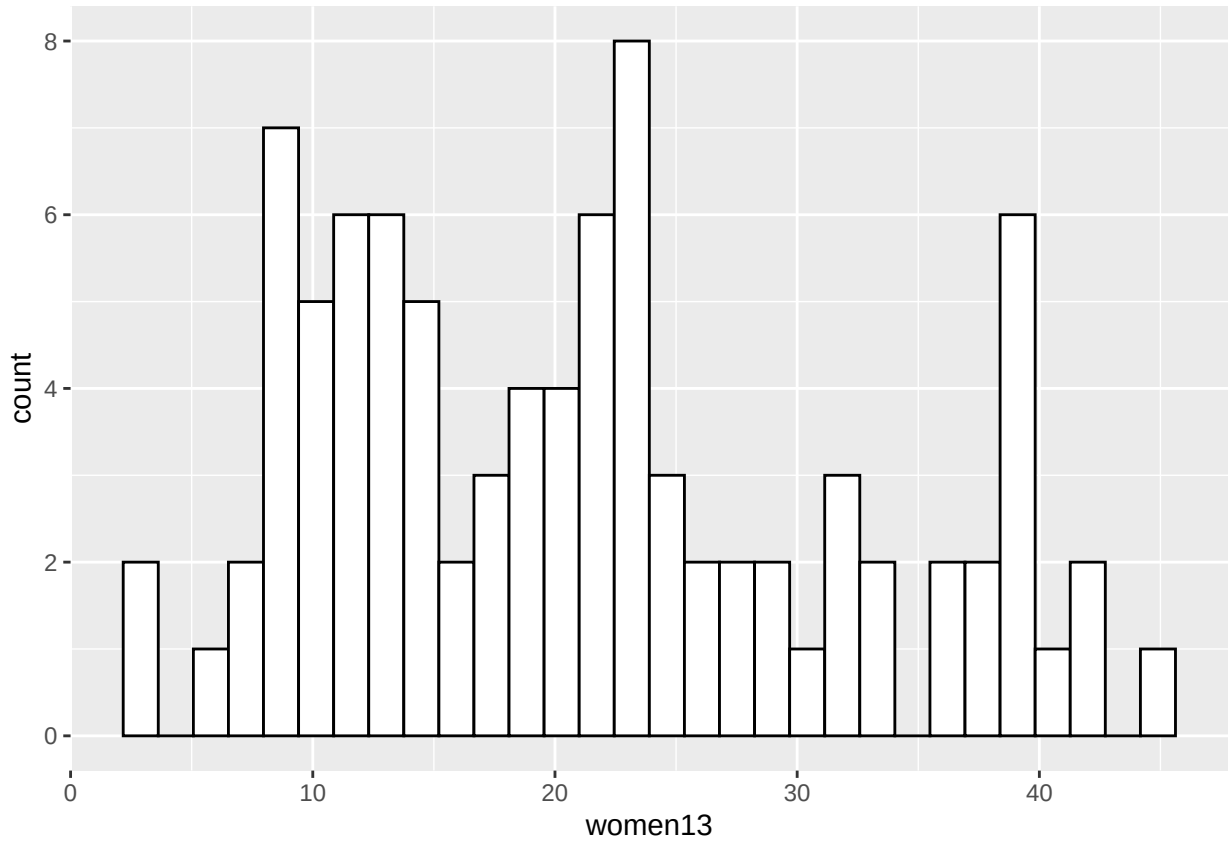


Another way to customize the figure is to change the colors of the bars and the background. Changing the color of the bars and the outline color of the bars adds code to the `geom_histogram()` command. To do this, you simply need to add the options `fill=` and `color=` for the bar color and outline color respectively. I demonstrate this in the code below:

```
ggplot(data=worlddata, aes(x=women13)) +
  geom_histogram(color="black", fill="white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 77 rows containing non-finite values (stat_bin).
```

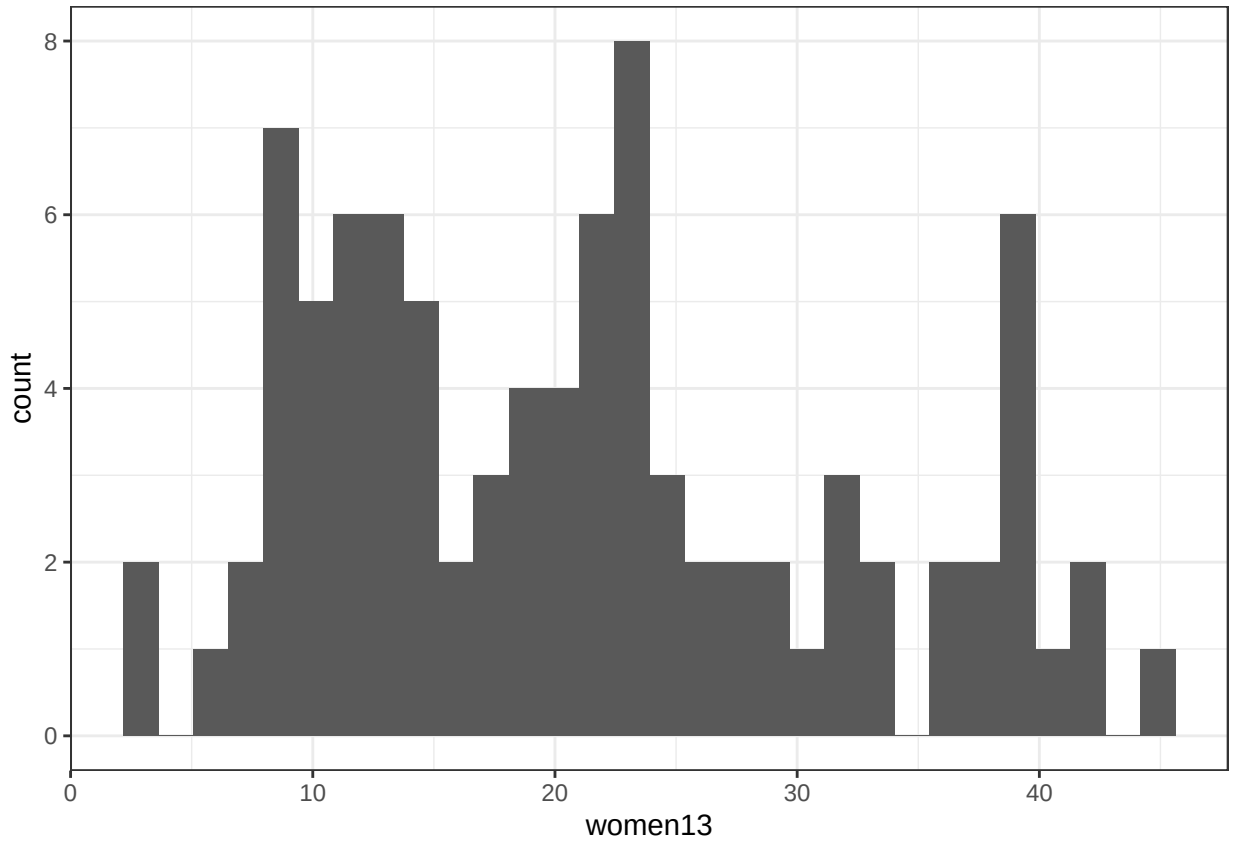


To change the background of the figure from grey to white, you can add the `theme_bw()` to the rest of the code, connected by the `+` symbol. An example is shown below:

```
ggplot(data=worlddata, aes(x=women13)) + geom_histogram() + theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

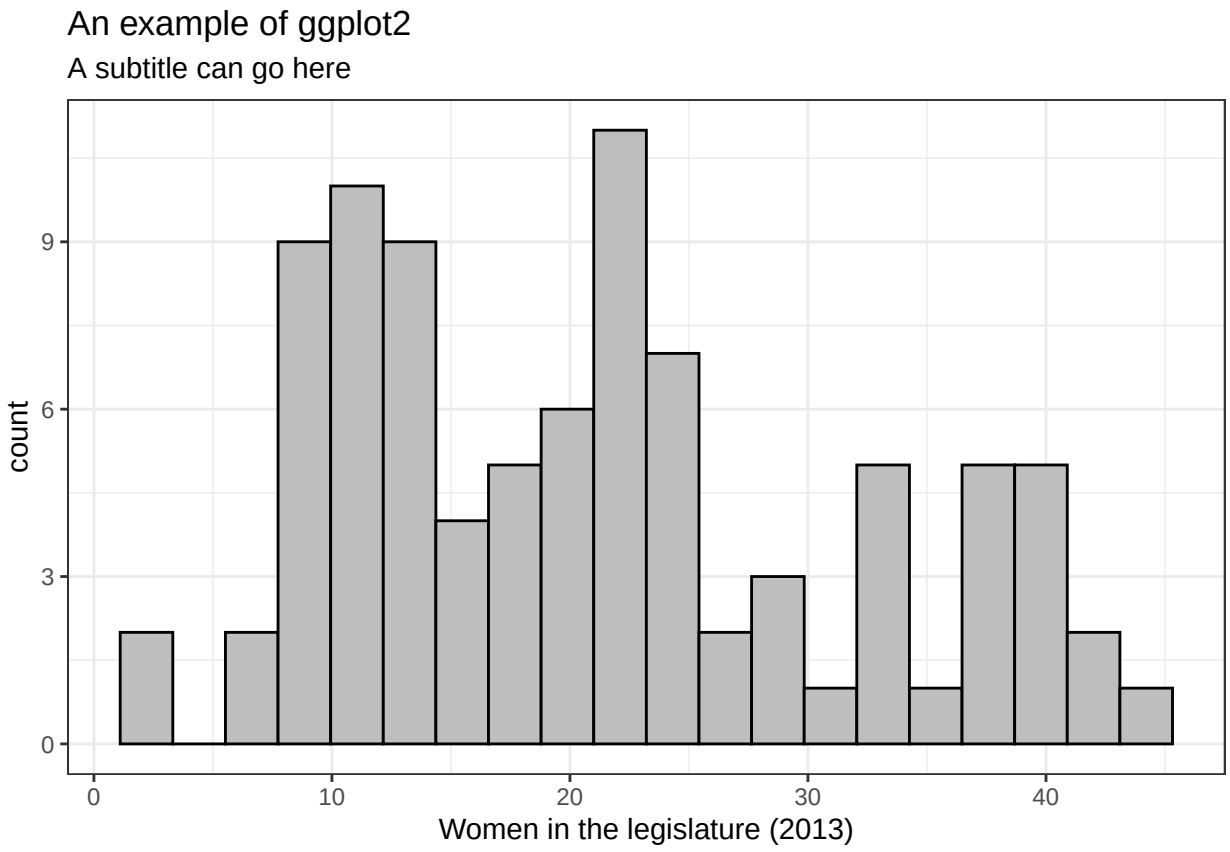
```
## Warning: Removed 77 rows containing non-finite values (stat_bin).
```



We can put all of these options together and create a much more professional looking figure:

```
ggplot(data=worlddata, aes(x=women13)) + geom_histogram(bins=20, color="black", fill="grey") +
  theme_bw() + ggtitle("An example of ggplot2", subtitle= "A subtitle can go here") +
  xlab("Women in the legislature (2013)")
```

```
## Warning: Removed 77 rows containing non-finite values (stat_bin).
```



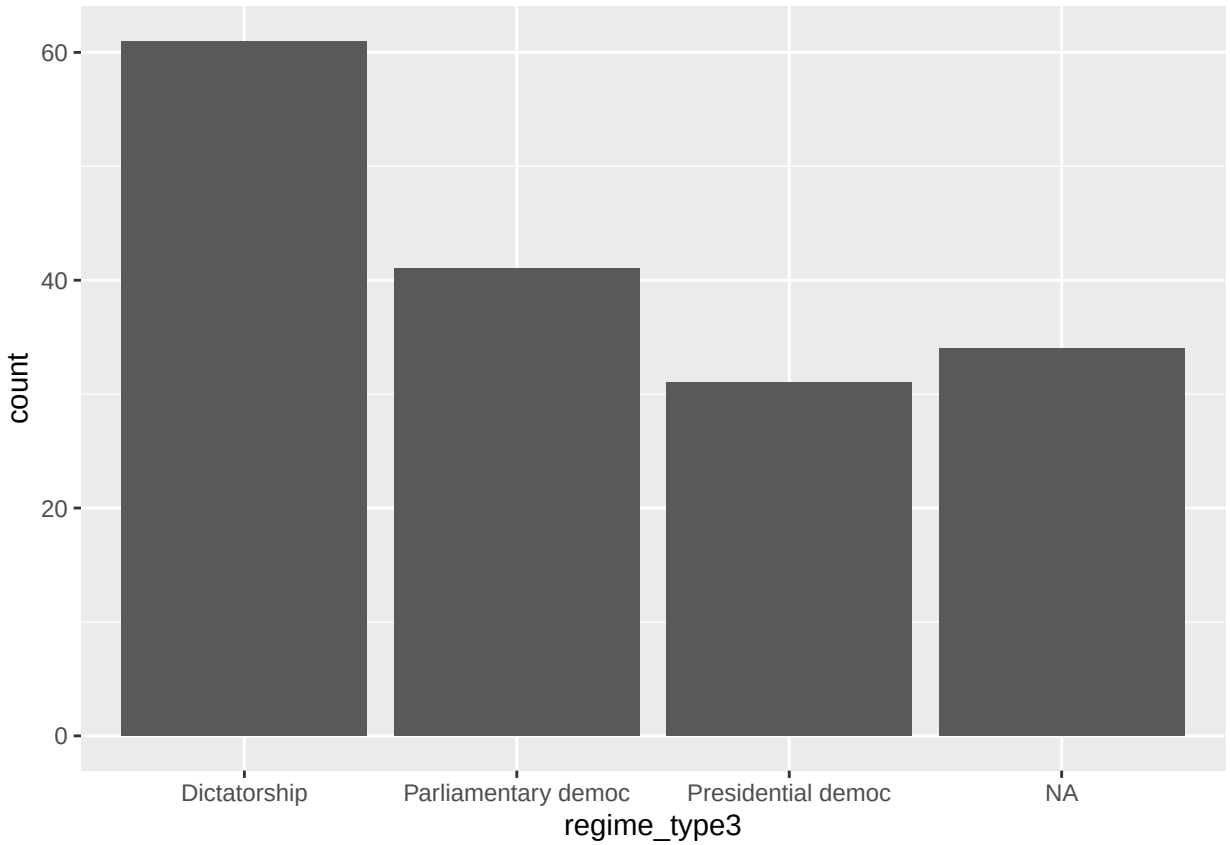
Bar charts

A great way to show the distribution of a nominal variable is to use a barplot. Barplots begin the exact same way as histograms with the `ggplot` core command. In this example, we will be using the variable *regime_type3*. This is a nominal variable that labels countires as Presidential Democracies, Parliamentary Democracies, or Dictatorships. Note, that there are many missing values for this variable. These are labelled as **NA**.

```
#The basic command:
#here, you use the ggplot() command, and include the data name (data=)
#and include the variable you want to plot (x=) inse the aes() option,
#each separated with a column
ggplot(data=worlddata, aes(x=regime_type3))
```

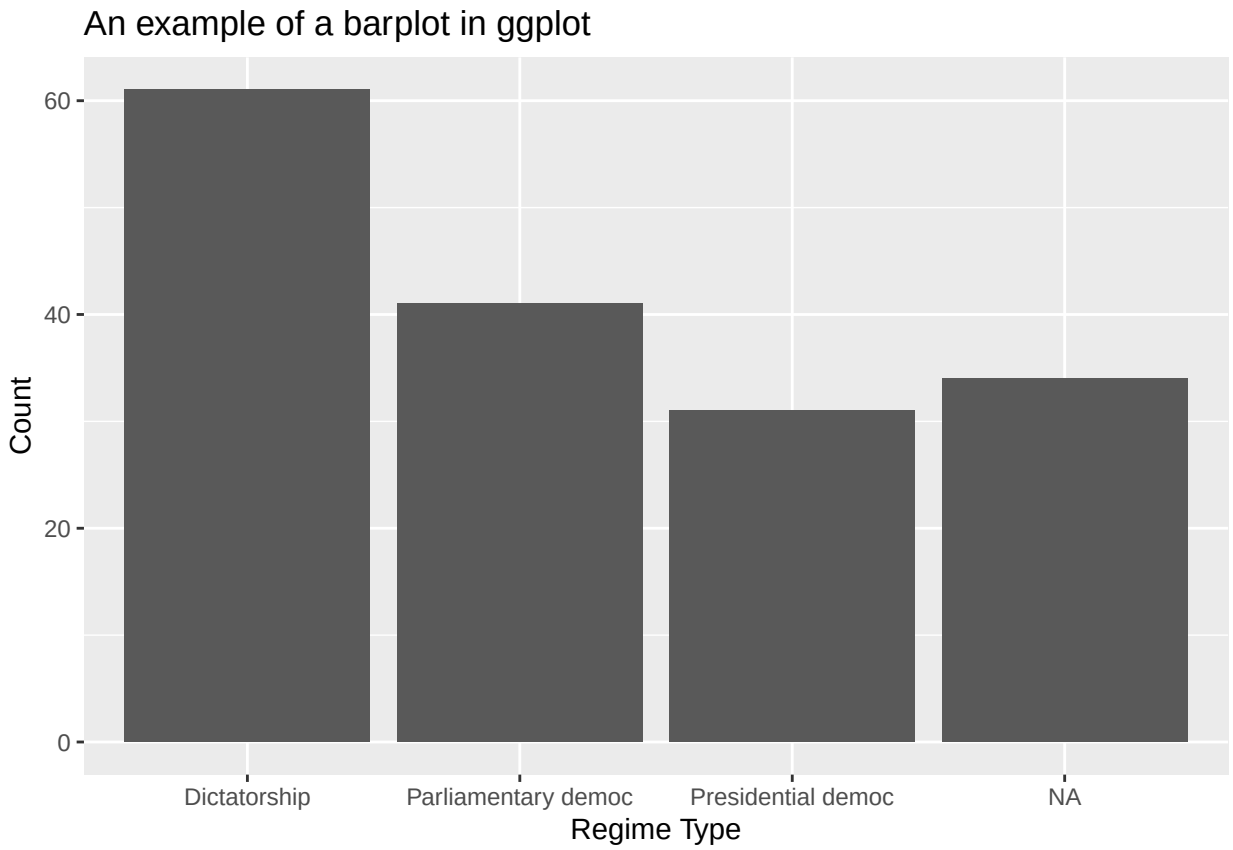
After the core `ggplot()` command, you need to add the `geom_bar()` part of the command.

```
ggplot(data=worlddata, aes(x=regime_type3)) + geom_bar()
```



After we have the basic plot, we can customize it, just like we did with the histogram. The first thing we can do is add labels. Adding labels for a Barplot in `ggplot2` is the exact same as adding labels to a `ggplot2` histogram. The example below uses the same `ggtitle()` and `xlab()` as before. Here, I have also added the `ylab()` command, which changes the label on the y axis.

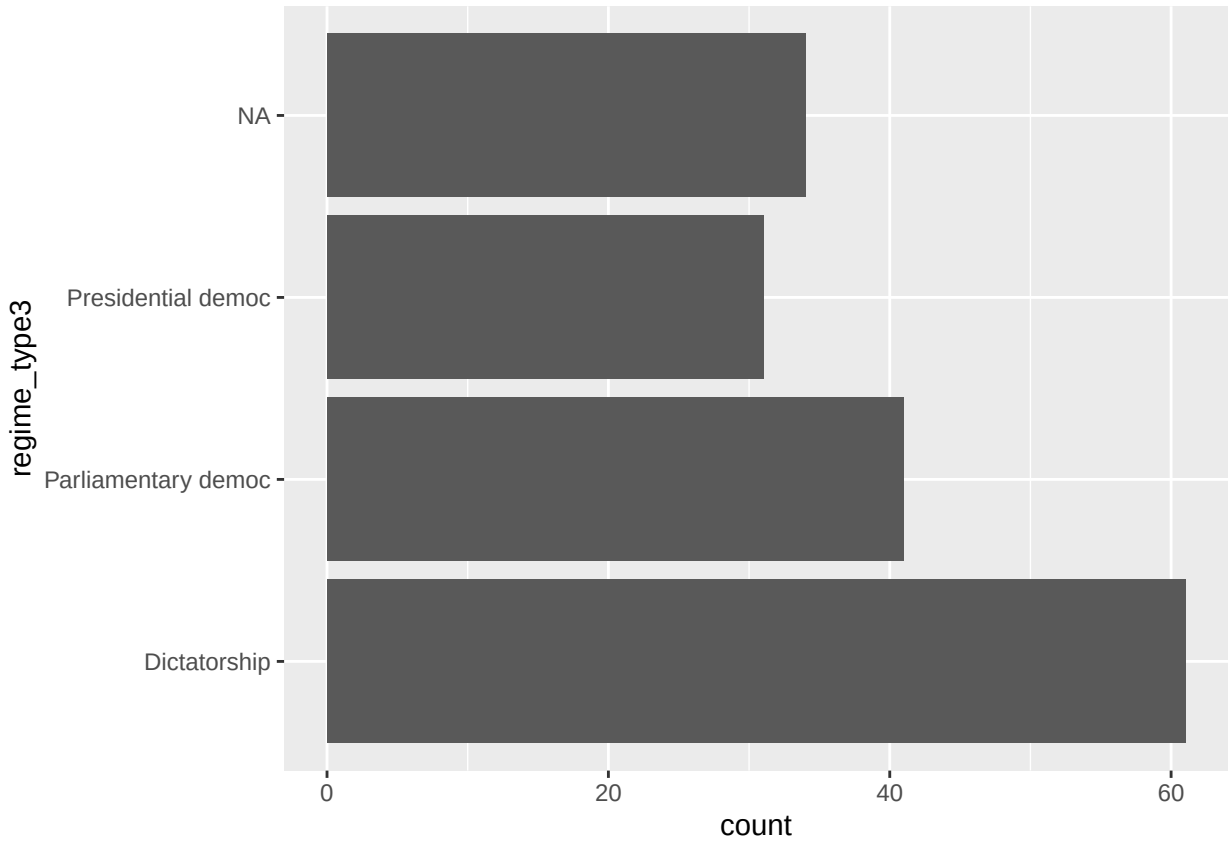
```
ggplot(data=worlddata, aes(x=regime_type3)) + geom_bar() +
  ggtitle("An example of a barplot in ggplot") +
  xlab("Regime Type") + ylab("Count")
```



Another thing you can do with a bar plot is make is horizontal rather than vertical. You can do this with the `coord_flip()` option. I demonstrate this bit of code int he next section:

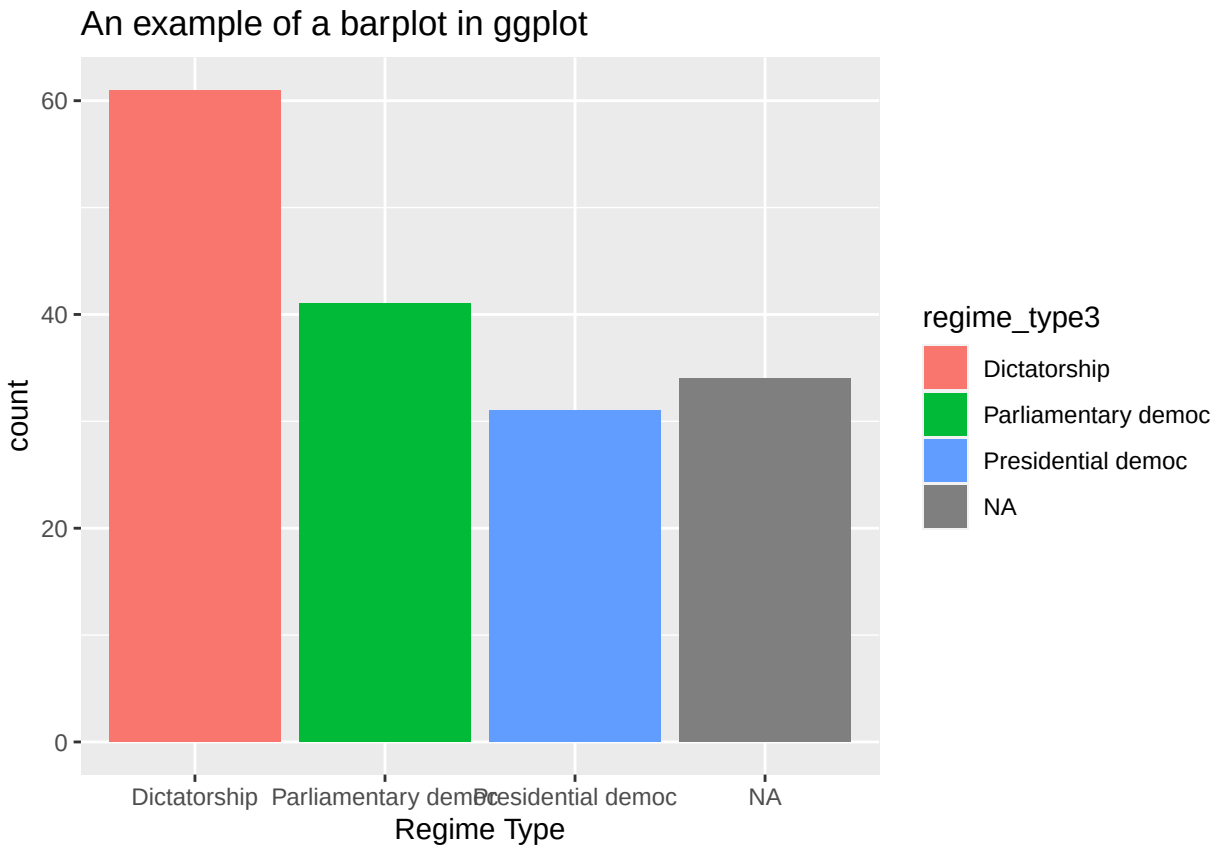
```
#flip the axis:

ggplot(data=worlddata, aes(x=regime_type3)) + geom_bar() + coord_flip()
```



With barplots, we can also change the colors of the bars, color coordinating them by the value of the nominal variable that we are plotting. This requires that we add another option to the `aes()` part of the code. The option is called `fill=`. You need to tell R which criteria to use in order to decide colors. You can do this using the values of the variable, or, when we get to bivariate displays, the values of another variable. The code below demonstrates this:

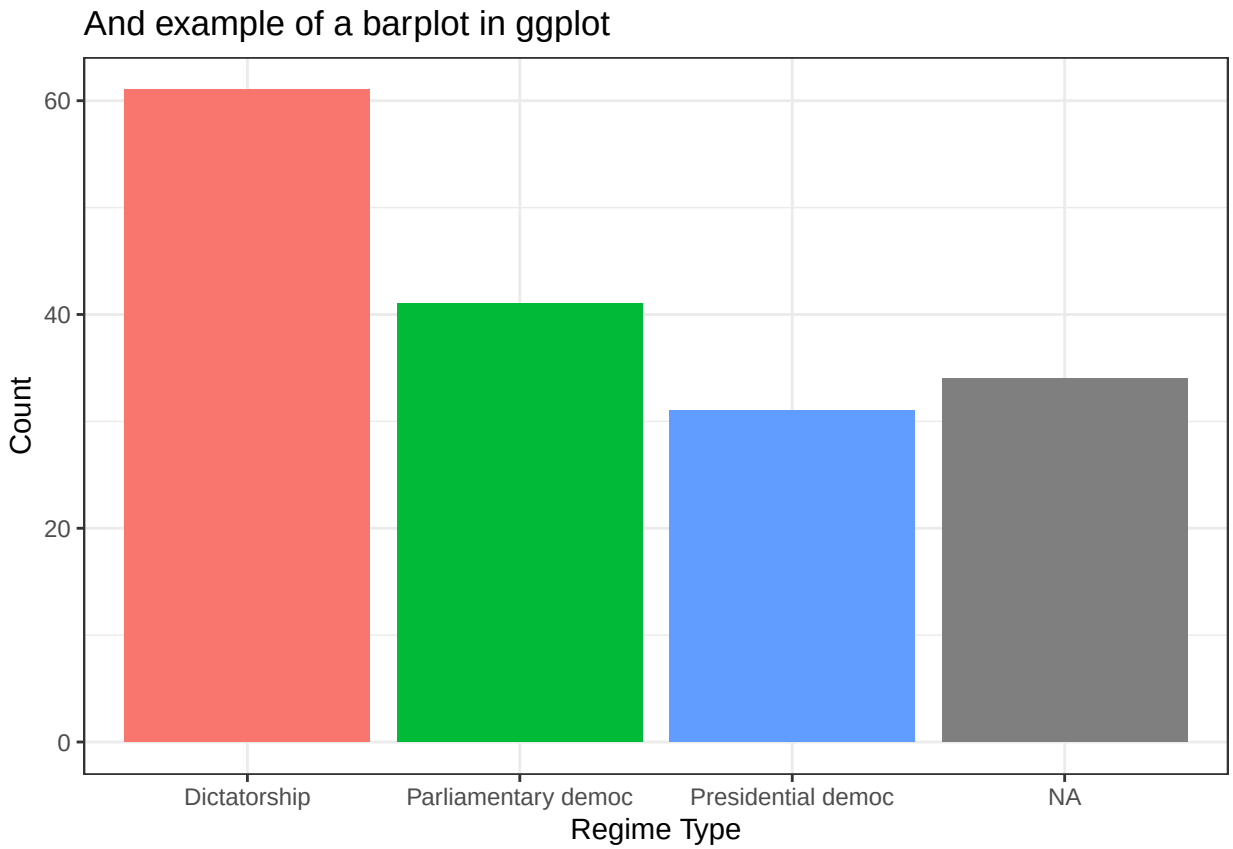
```
#TO change the fill color by regime type:
ggplot(data=worlddata, aes(x=regime_type3, fill=regime_type3)) +
  geom_bar() + ggtitle("An example of a barplot in ggplot") +
  xlab("Regime Type")
```



Another thing that you might want to do is to get rid of the legend. You can do this by adding another option to the figure with the `+` symbol. This option is in the `theme()` family of options. The code below shows how to get rid of the legend, but you can also move it to the `"top"` or `"bottom"`. This requires those words in place of the `"none"` part of the command.

```
#To get rid of the legend, use legend.position="none"

ggplot(data=worlddata, aes(x=regime_type3, fill=regime_type3)) +
  geom_bar() + ggtitle("And example of a barplot in ggplot") +
  xlab("Regime Type")+ ylab("Count") + theme_bw() + theme(legend.position="none")
```




```
#"top" and "bottom" are also options here, if you just want to move the legend
```

You will also notice that we still have a bar of "NA's", or missing values. These are the observations for which there is no information on regime type. We generally don't want to include them in our figures, so we can remove them from our figure using the `filter()` command from the `dplyr` library. We can use the filter command to tell R to only keep observations that are not missing for the regime type variable. We will need to include this code in the first part of the main `ggplot` command, directly after the data. Also, we will need to connect the `filter` command to the data name with the `%>%` operator (called a "pipe"). The code is demonstrated below:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

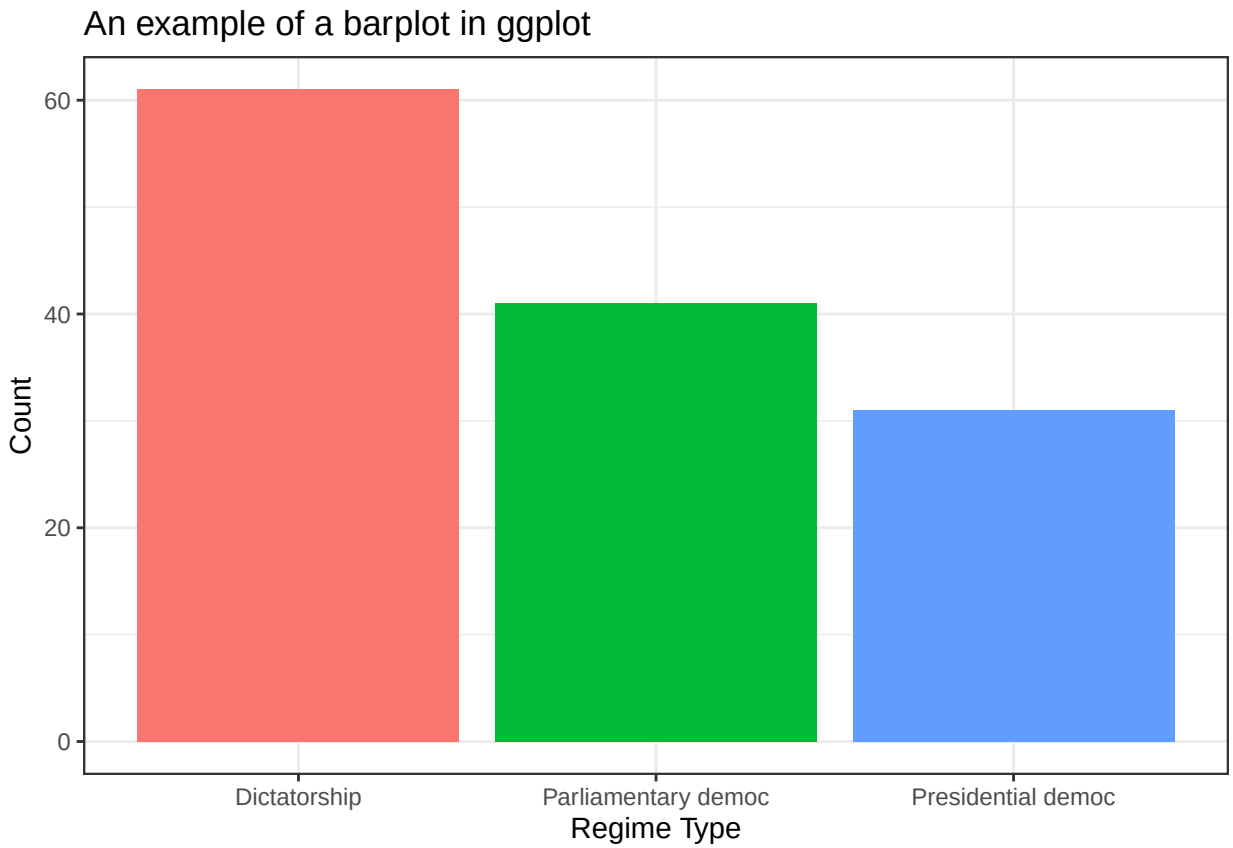
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)

#If you wanted to get rid of the NAs:
ggplot(data=worlddata %>% filter(!is.na(regime_type3)),
       aes(x=regime_type3, fill=regime_type3)) + geom_bar() +
  ggtitle("An example of a barplot in ggplot") + xlab("Regime Type")+
```

```
  ylab("Count") + theme_bw() + theme(legend.position="none")
```




```
#The filter command comes form dplyr
#and will sort through the dataset to only keep certain values.
#!is.na() means to keep all the observations that are not NAs for that variable.
#the ! is a NOT symbol.
```

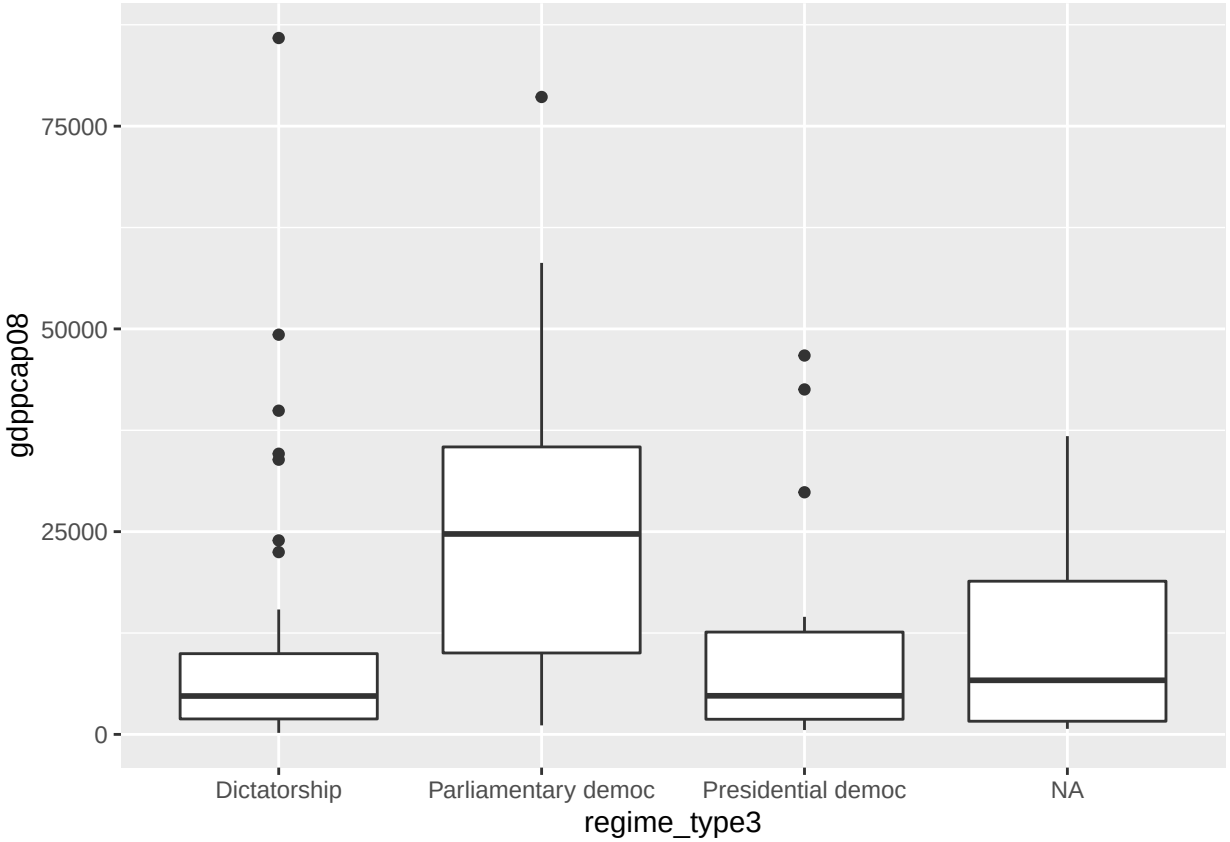
Boxplots

Box plots are great presentations fo bivariate (meaning two variables). However, one of the variables must be either nominal or ordinal (categorical) while the other must be interval (or continuous) level data. With the box plot, you can show relationships between these types of variables in a clean and interesting way. You can make these look very interesting (and, in my opinion, better) in `ggplot2`. This section shows you how to do this.

A boxplot with `ggplot2` starts just like any other figure with `ggplot2`: with the basic `ggplot()` command that specifies the data and variables. However, what does need to be added here in the `aes()` section of the command is the second variable. You should have an x and a y variable inside the `aes()`. In the case of the box plot, the x should be your nominal (or categorical) variable and the y should be the continuous (or interval) variable. After that is established in the base `ggplot()` command, you can then use the `geom_boxplot()` command, connected to the base `ggplot()` command with a `+`, just as we did with the other plots above. I have demonstrated this below. In this example, the regime type variable (`regime_type3`) we used for the bar plot example is our x. The y for this plot is GDP per capita (`gdppcap08`)'.

```
ggplot(data=worlddata, aes(x=regime_type3, y=gdppcap08)) + geom_boxplot()
```

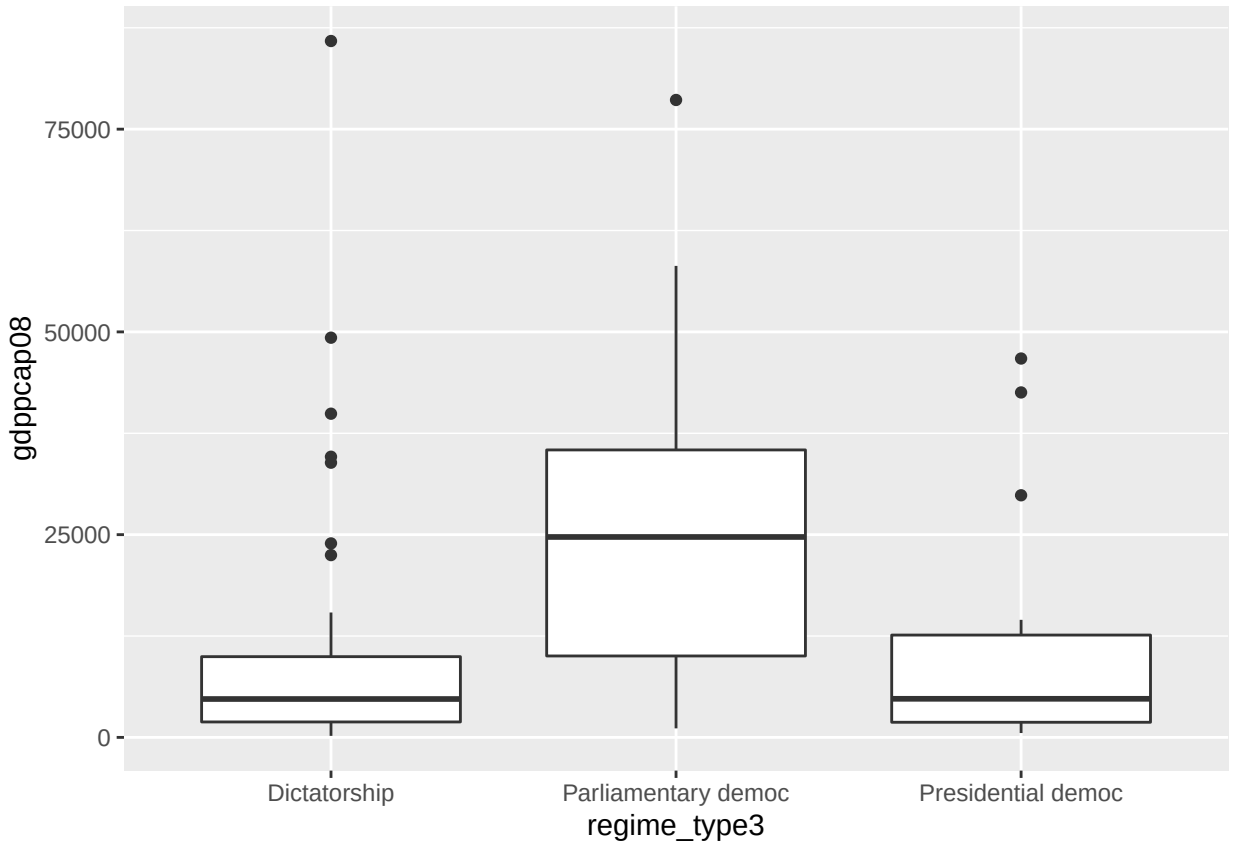
```
## Warning: Removed 15 rows containing non-finite values (stat_boxplot).
```



However, this boxplot has the same issue as the bar chart above: the NA category. We can use the same series of commands here to take those out:

```
ggplot(data=worlddata %>% filter(!is.na(regime_type3)), aes(x=regime_type3, y=gdppcap08)) + geom_boxplo
```

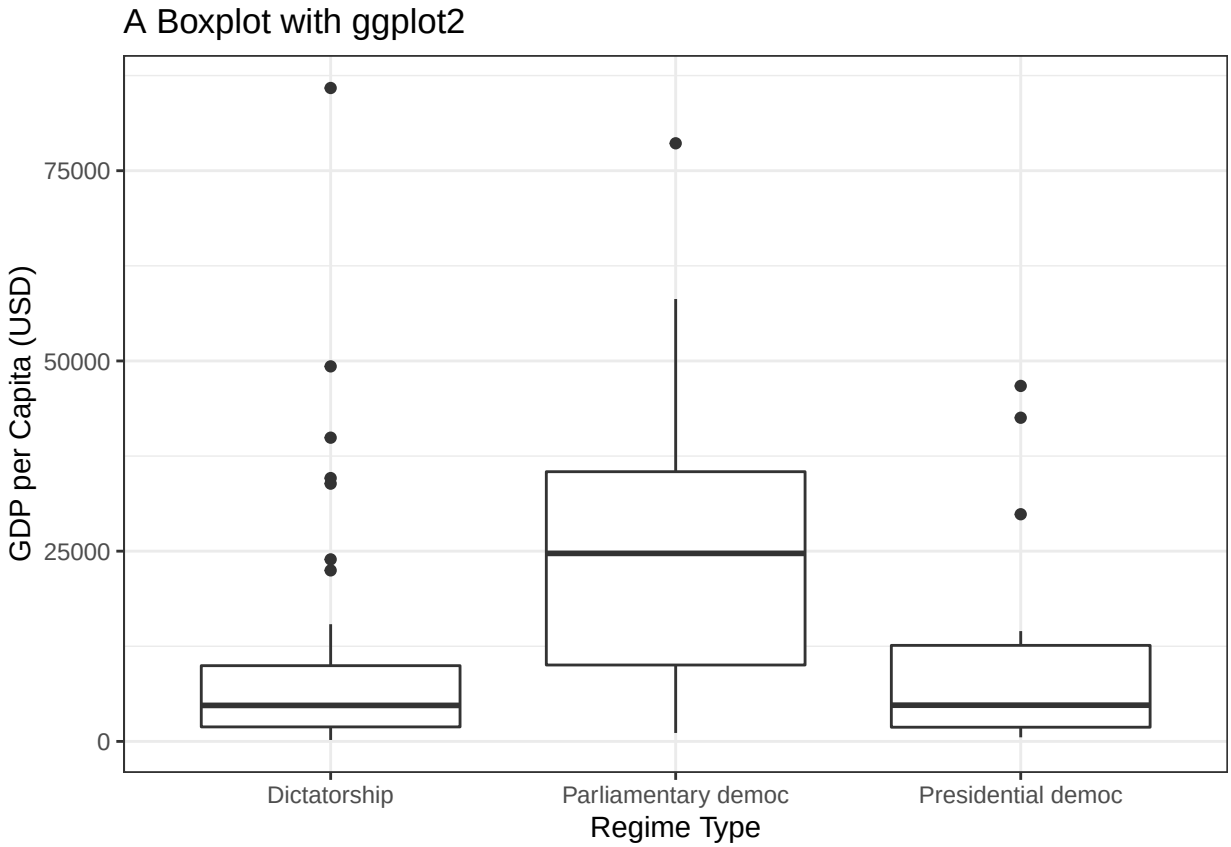
```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```



From here, you can relabel the axes and give the figure a title with the exact same process as all the other figures shown here.

```
ggplot(data=worlddata %>% filter(!is.na(regime_type3)),
       aes(x=regime_type3, y=gdppcap08)) + geom_boxplot()+
       ggtitle("A Boxplot with ggplot2") + xlab("Regime Type")+
       ylab("GDP per Capita (USD)") + theme_bw()
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```



Scatterplots

Another type of bivariate figures is a scatterplot. Scatterplots show relationships between two continuous variables. In base `R`, scatterplots are made using the `plot()` command, but `ggplot2` requires a bit more work. However, scatterplots with `ggplot2` look much better. This section demonstrates a `ggplot2` style scatterplot.

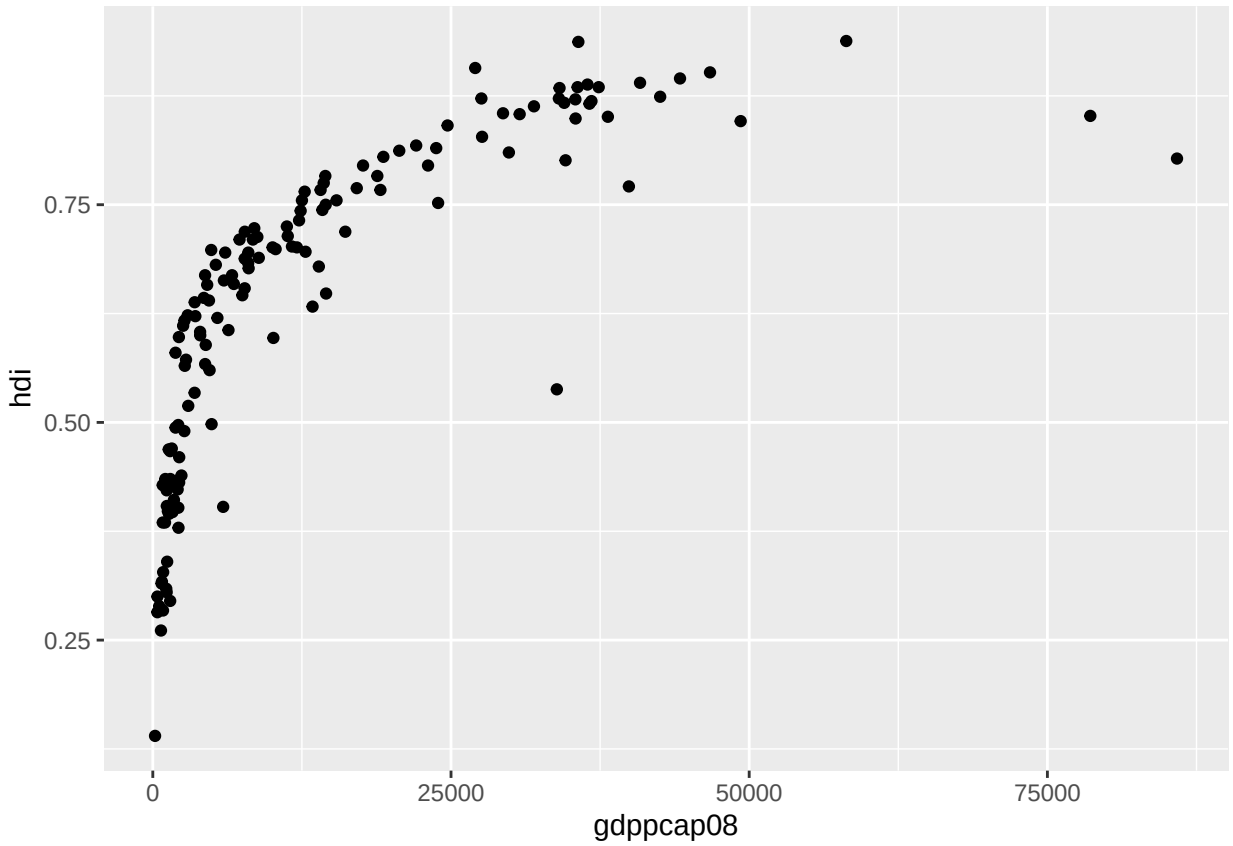
Making a scatterplot in `ggplot2` is much like making a boxplot. You first start with the main `ggplot()` command. Inside that command, you need to us the `aes()` option to specify x and y variables. Both of these variables need to be continuous for the scatterplot to work. After the main `ggplot()` command where the variables are established, you need to add the `geom_point()`, which is the command to add a scatterplot to the main plot. the code below shows the basic scatterplot code.

```
#Here, the x variable (horizontal axis) is the GDP per capita
#Y variable (vertical axis) is the human development index (HDI)

#Unit of analysis here is country

ggplot(data=worlddata, aes(x=gdppcap08 , y=hdi))+geom_point()
```

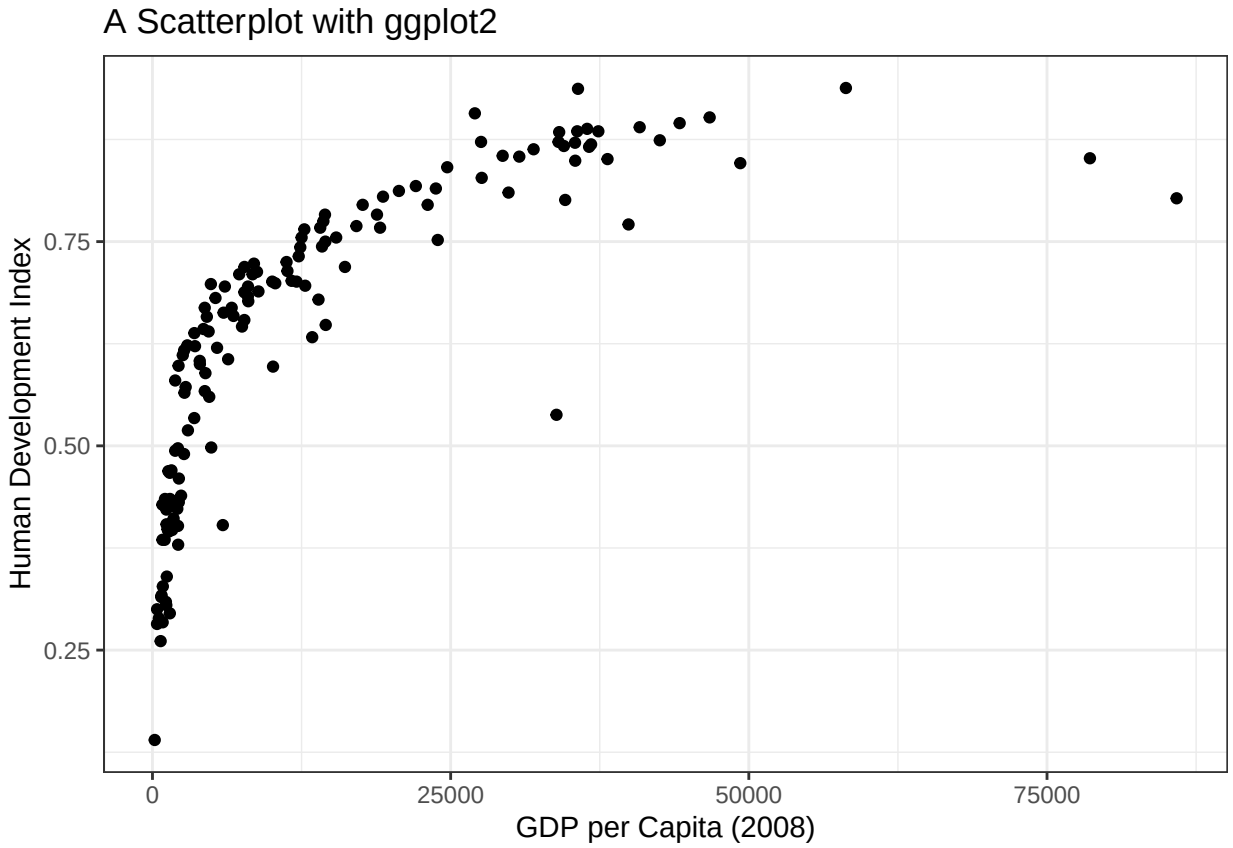
```
## Warning: Removed 22 rows containing missing values (geom_point).
```



Of course, this plot misses much of the nicer things that we have added in the past such as axis labels and a title. We can add these with the same commands as above.

```
ggplot(data=worlddata, aes(x=gdppcap08 , y=hdi))+geom_point()+
  xlab("GDP per Capita (2008)") + ylab("Human Development Index")+
  ggtitle("A Scatterplot with ggplot2") +theme_bw()
```

```
## Warning: Removed 22 rows containing missing values (geom_point).
```

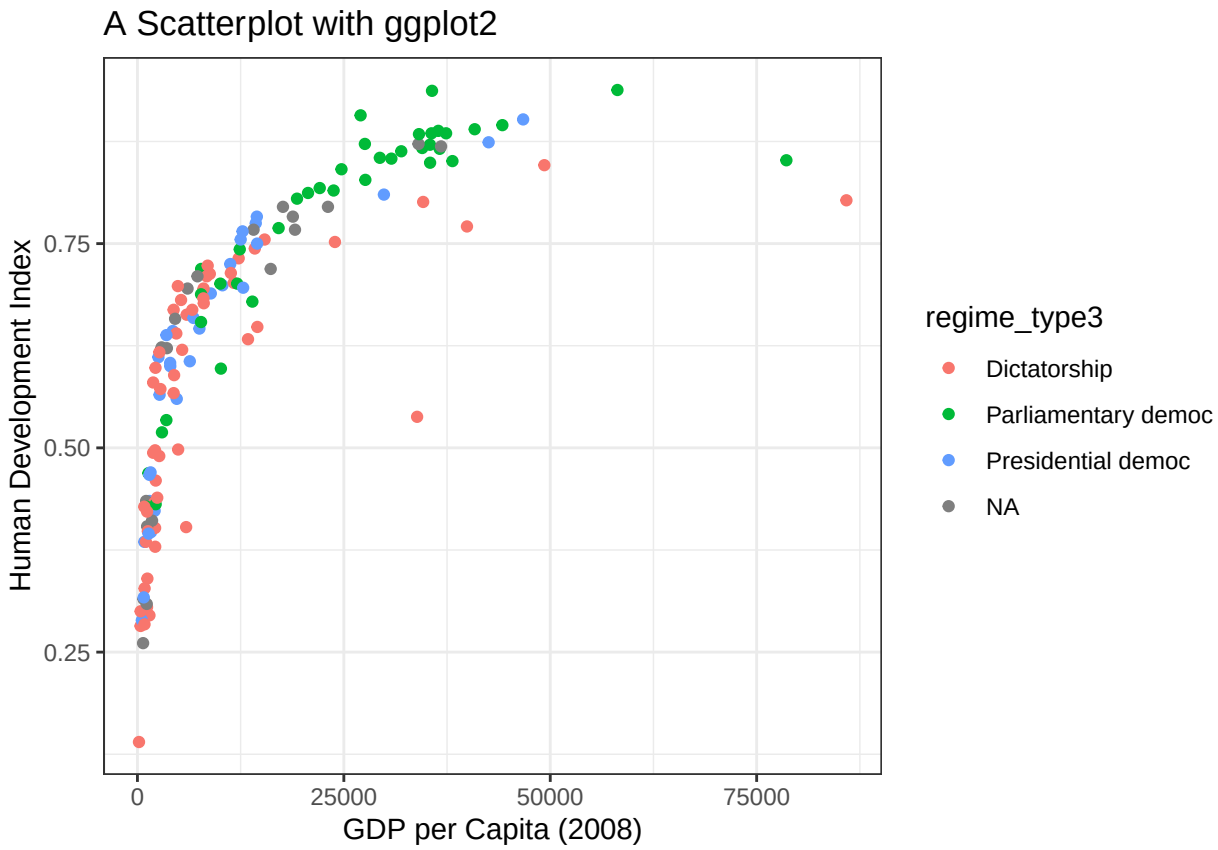


Another nice thing about scatterplots with `ggplot2` is that you can add other variables. By this, I mean that you can change the color of the points by another variable (for example, regime type). To do this, you simply need to add another criteria to the `aes()` option. Inside this option, you just need to add `color=""` with the name of some variable. For best results, this color variable should be categorical.

In the example below, I change the colors of the points to correspond with the regime type variable we used in the above example.

```
ggplot(data=worlddata, aes(x=gdppcap08 , y=hdi, color=regime_type3))+geom_point()+
  xlab("GDP per Capita (2008)") + ylab("Human Development Index")+
  ggtitle("A Scatterplot with ggplot2") +theme_bw()
```

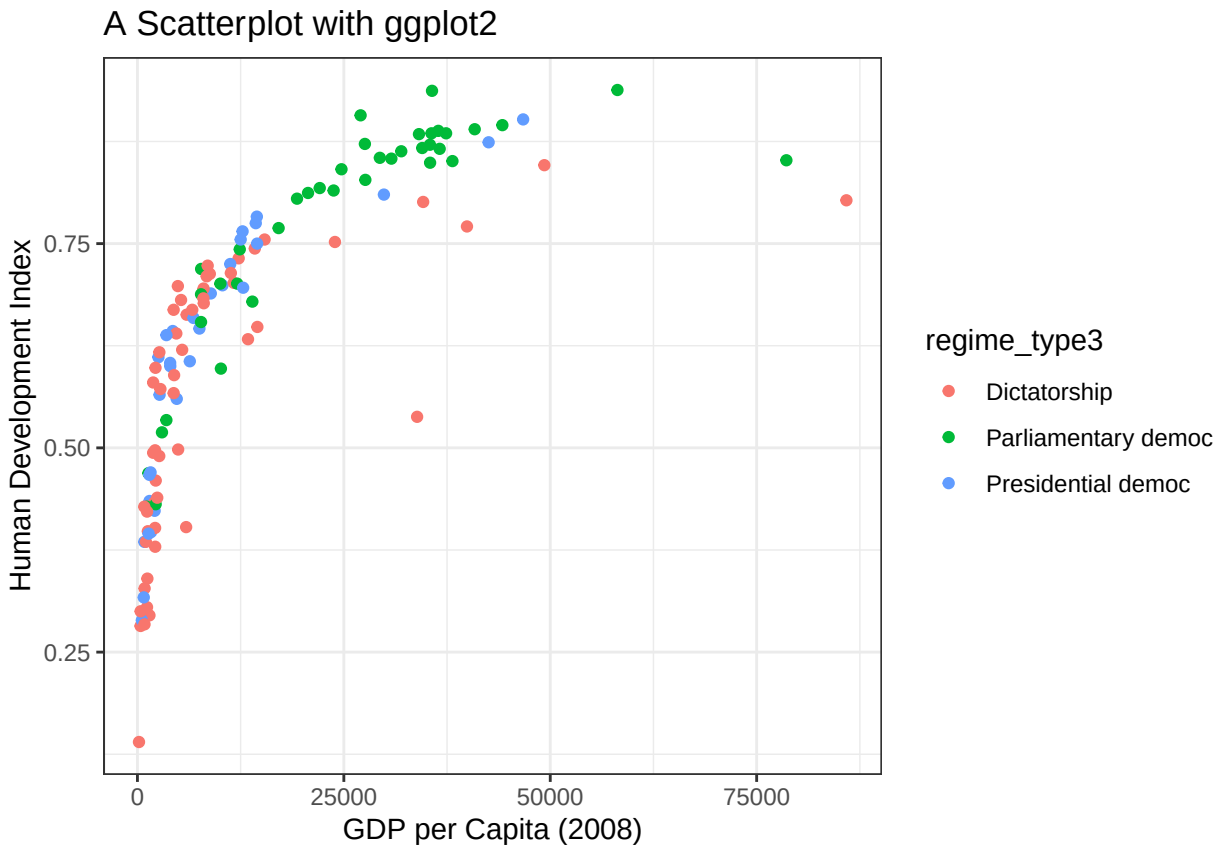
```
## Warning: Removed 22 rows containing missing values (geom_point).
```



However, we have the same problem with missing regime type variables. So we can use the same `filter()` command to get ride of those missing observations

```
ggplot(data=worlddata %>% filter(!is.na(regime_type3)),
       aes(x=gdppcap08 , y=hdi, color=regime_type3))+geom_point()+
  xlab("GDP per Capita (2008)") + ylab("Human Development Index")+
  ggtitle("A Scatterplot with ggplot2") +theme_bw()
```

```
## Warning: Removed 7 rows containing missing values (geom_point).
```



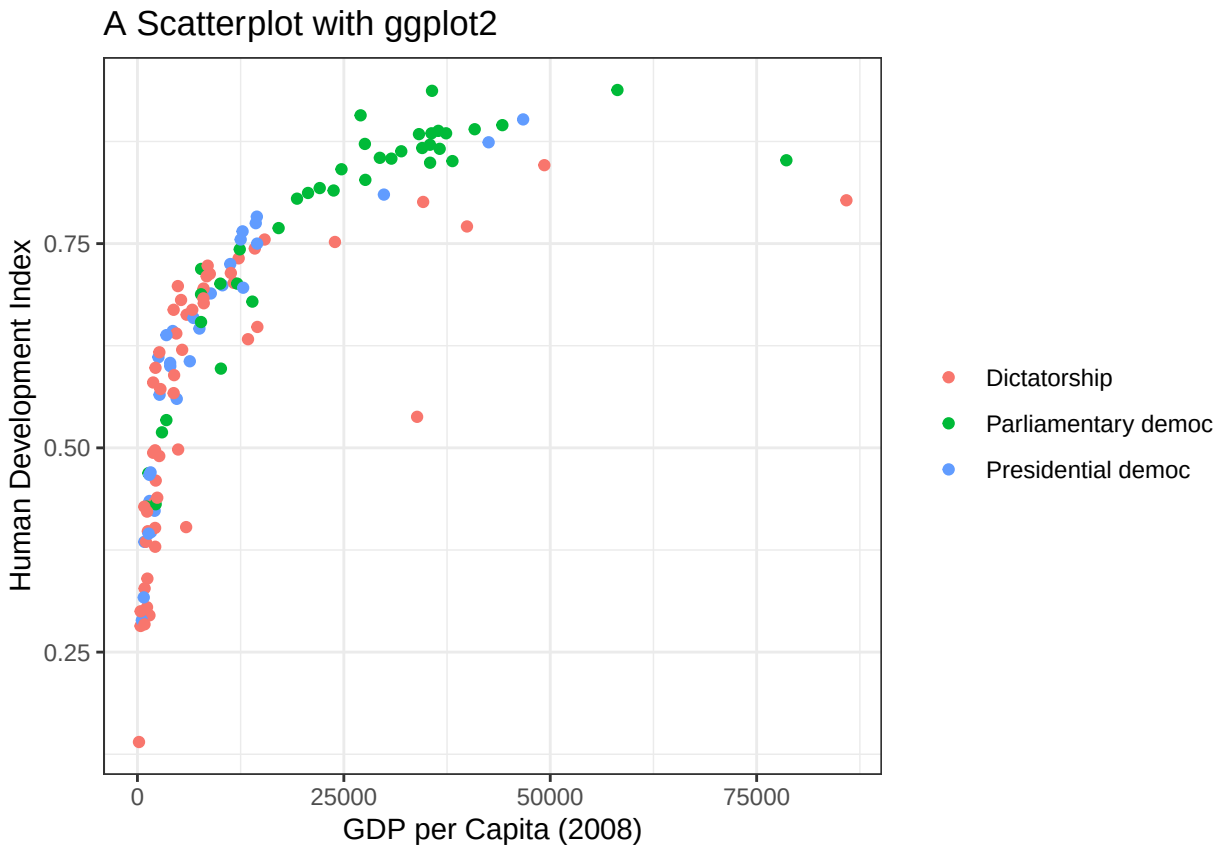
This looks much better! But we still have an issue with the legend, or the part of the figure that shows what the colors mean. As you can see, the legend title is still the variable's `R` label, not a more descriptive, better looking label. You can customize the legend as needed. You can change the title, remove the title altogether, and change the color labels. In the following sections I will show you some of the commands needed to customize the legend.

The first thing to know is the use of the command `theme()`. This is a more generic command inside which you can change the settings for the legend. Each part of the legend is assigned its own option. To change the title of the legend, you should use the `legend.title=` command.

```
#How to remove the legend title

ggplot(data=worlddata %>% filter(!is.na(regime_type3)),
       aes(x=gdppcap08 , y=hdi, color=regime_type3))+geom_point()+
  xlab("GDP per Capita (2008)") + ylab("Human Development Index")+
  ggtitle("A Scatterplot with ggplot2")+theme_bw()+
  theme(legend.title = element_blank())
```

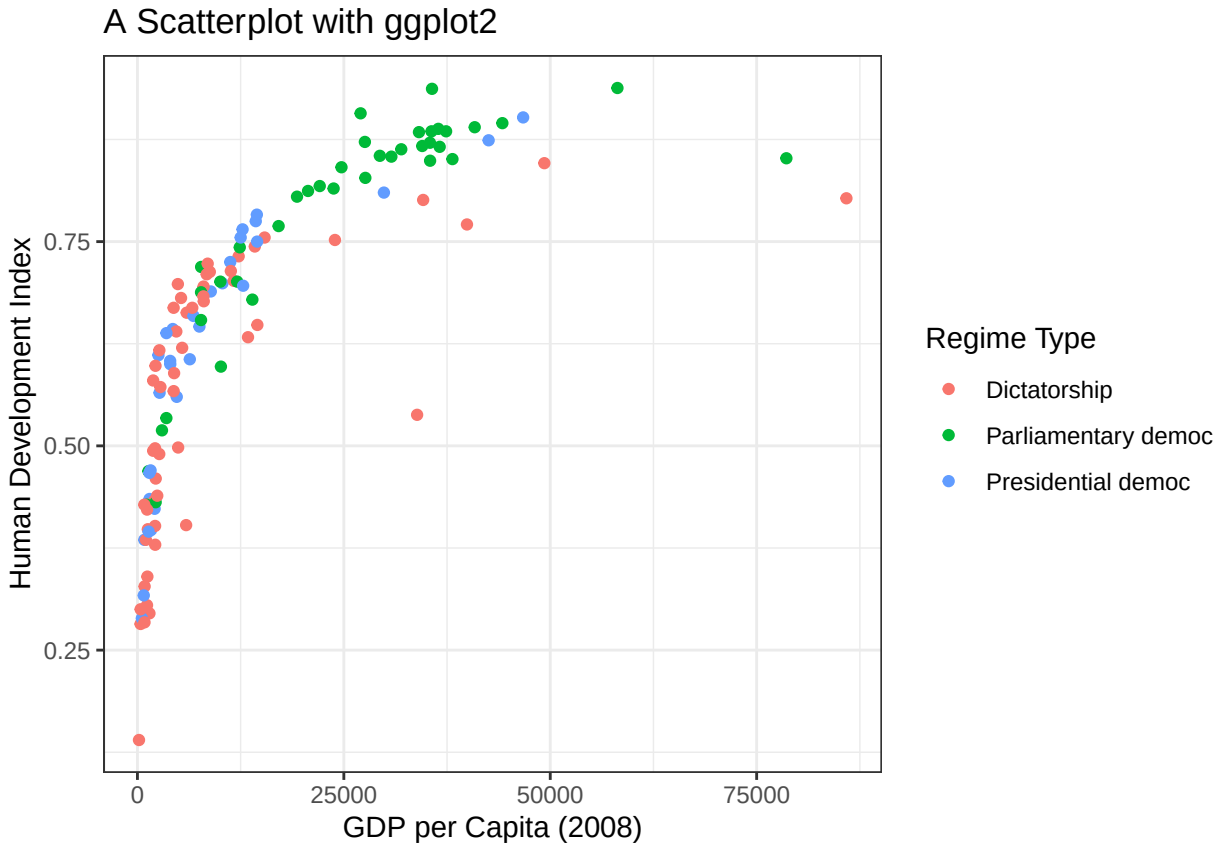
```
## Warning: Removed 7 rows containing missing values (geom_point).
```




```
#To change legend title, use the labs() command

ggplot(data=worlddata %>% filter(!is.na(regime_type3)),
       aes(x=gdppcap08 , y=hdi, color=regime_type3))+geom_point()+
  xlab("GDP per Capita (2008)") + ylab("Human Development Index")+
  ggtitle("A Scatterplot with ggplot2")+theme_bw()+
  labs(color="Regime Type")
```

```
## Warning: Removed 7 rows containing missing values (geom_point).
```

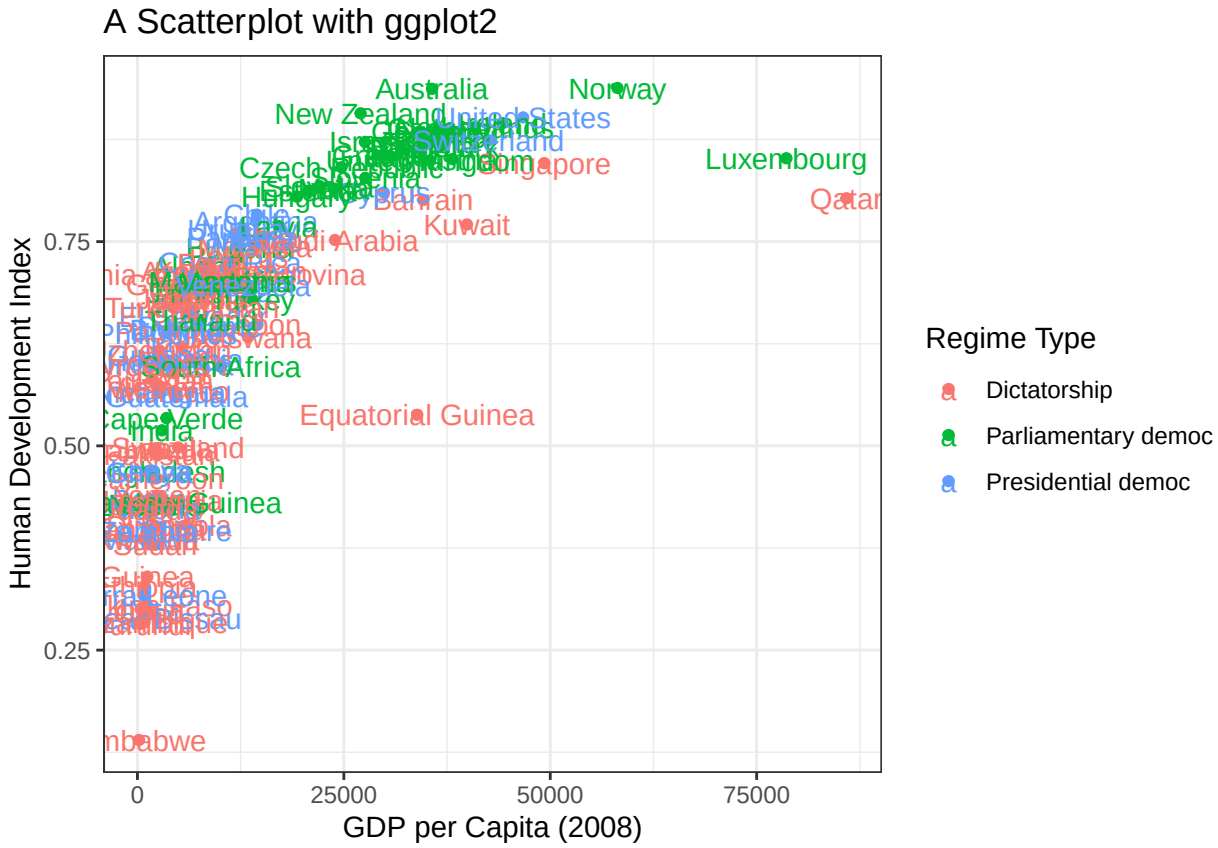


Another interesting thing you can do with `ggplot2` in a scatterplot is label the points. You can do this by adding the `geom_text()` command. In this case, the text you are adding is the name or abbreviations for the countries. I show this below:

```
ggplot(data=worlddata %>% filter(!is.na(regime_type3)),
       aes(x=gdppcap08 , y=hdi, color=regime_type3))+geom_point()+
  geom_text(aes(label=country))+
  xlab("GDP per Capita (2008)") + ylab("Human Development Index")+
  ggtitle("A Scatterplot with ggplot2")+theme_bw()+
  labs(color="Regime Type")
```

```
## Warning: Removed 7 rows containing missing values (geom_point).
```

```
## Warning: Removed 7 rows containing missing values (geom_text).
```

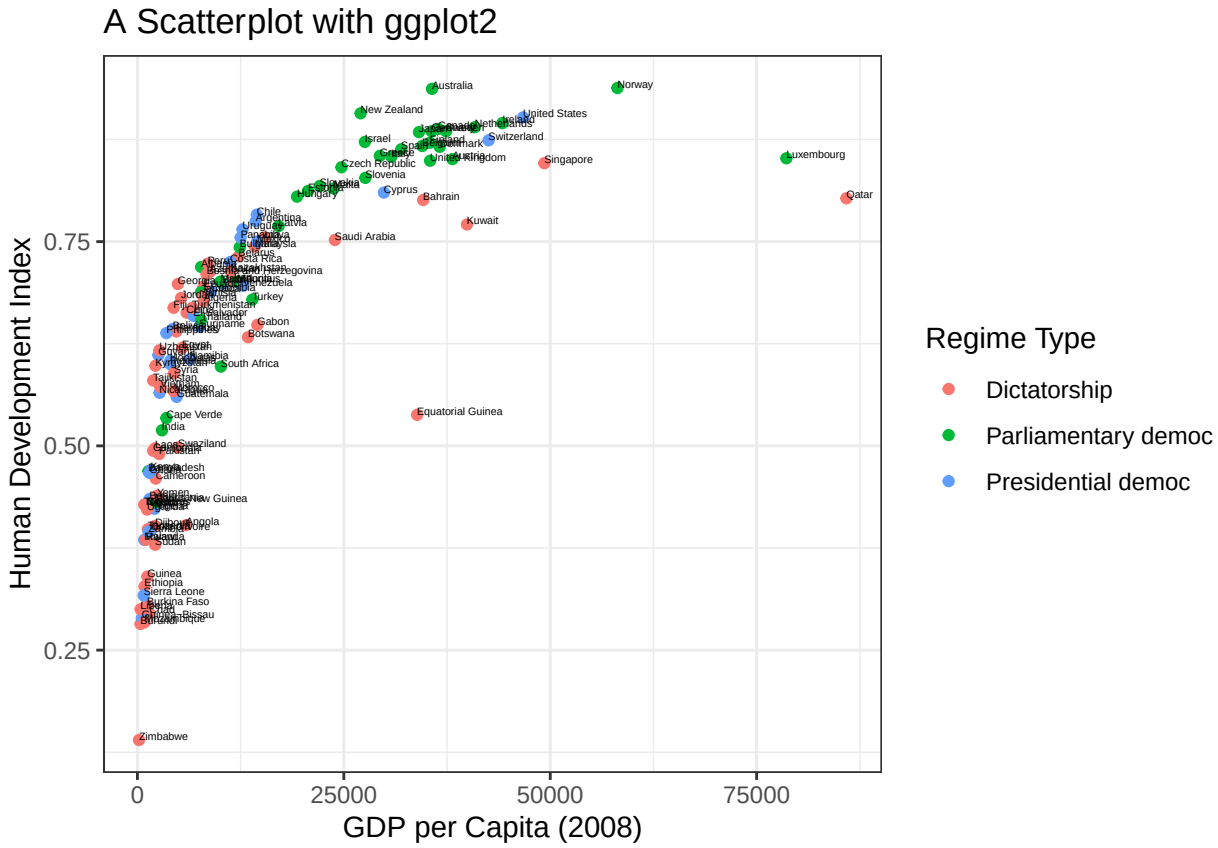


There are, of course, many things wrong with this figure. We can change this all through the options within the `geom_text()` command. You can play with the options to make the text larger or a different color. The `vjust` and `hjust` options change where the text is relative to the points. The default is for the text to be centered on the point. In the following figure, I changed it so that the labels would be to the right of the points.

```
ggplot(data=worlddata %>% filter(!is.na(regime_type3)),
       aes(x=gdppcap08 , y=hdi, color=regime_type3))+geom_point()+
  geom_text(aes(label=country), size=1.25, color="black", hjust=0, vjust=0)+
  xlab("GDP per Capita (2008)") + ylab("Human Development Index")+
  ggtitle("A Scatterplot with ggplot2")+theme_bw()+
  labs(color="Regime Type")
```

```
## Warning: Removed 7 rows containing missing values (geom_point).
```

```
## Warning: Removed 7 rows containing missing values (geom_text).
```



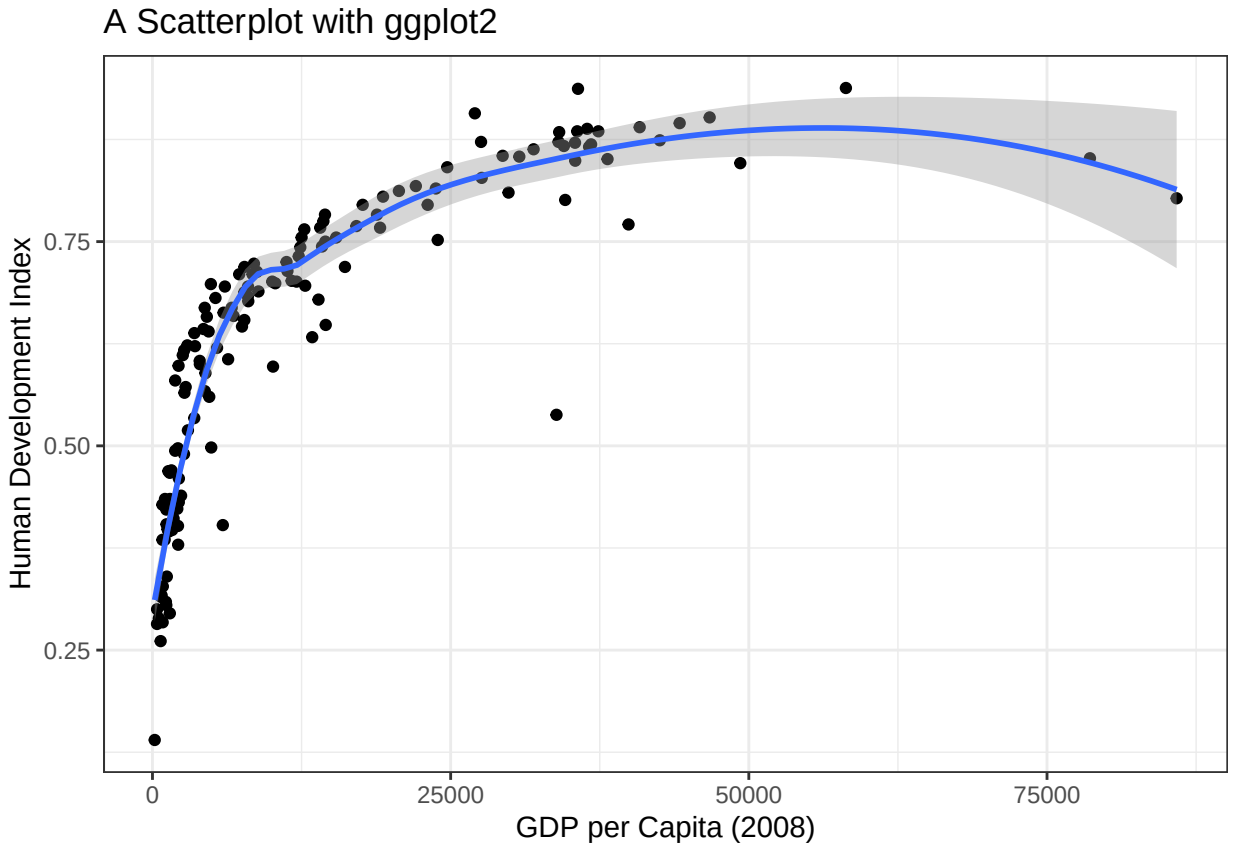
You can even add a line of best fit to a scatterplot using the `geom_smooth()` command. This command creates a smooth line that fits the best relationship between the two variables. This command also gives you the confidence interval of this line.

```
ggplot(data=worlddata, aes(x=gdppcap08 , y=hdi))+geom_point()+
  xlab("GDP per Capita (2008)") + ylab("Human Development Index")+
  ggtitle("A Scatterplot with ggplot2") +theme_bw() +
  geom_smooth()
```

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

```
## Warning: Removed 22 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 22 rows containing missing values (geom_point).
```



For more tips and tricks to customize `ggplot2` figures, see http://r-statistics.co/Complete-Ggplot2-Tutorial-Part2-Customizing-Theme-With-R-Code.html.