Introduction to ggplot2

Alice Kamau| Ken Mwai | Mark Otiende

5/5/2021

What is ggplot

- ggplot2 is an R package for producing statistical, or data, graphics.
- Under the tidyverse family of packages
- ggplot2 has an underlying grammar, based on the Grammar of Graphics
- compose graphs by combining independent components.

How ggplot works

- ggplot2 divides plot into three different fundamental parts:
  - *Plot = data + Aesthetics + Geometry*
- The principal components of every plot can be defined as follow:
  - ***data*** *is a data frame*
  - ***Aesthetics*** *is used to indicate x and y variables. It can also be used to control the color, the size or the shape of points, the height of bars, etc…..*
  - ***Geometry*** *defines the type of graphics (histogram, box plot, line plot, density plot, dot plot, ….)*

Load required package

- We begin by loading the required packages. ggplot2 is included in the tidyverse package.

```
library(tidyverse)
```

- Set the directory

```
setwd("~")
```

- Load the data

```
bw_df <- read.csv("Data/birthweight2.csv")
names(bw_df)
```

```
##  [1] "id"       "matage"   "ht"       "gestwks" "sex"      "bweight"
##  [7] "ethnic"   "lbw"      "agegrp"   "lbw2"    "agegrp1"
```

Why?

- To have an understanding of you data we normally conduct exploratory data analysis (EDA) which can be graphical or numerical
- Primarily EDA is for seeing what the data can tell us before the formal modelling or hypothesis testing task
- Typical graphical techniques used in EDA for one measure:
  - *Histogram (one variable continuous),*
  - *Density plot (one variable continuous),*
  - *Bar plots (one variable discrete)*

Elements of grammar of graphics

- Data: variables mapped to aesthetic (aes function) features of the graph.
- Geoms: objects/shapes on the graph.
- Stats: statistical transformations that summarize data,(e.g mean, confidence intervals)
- Scales: define which aesthetic values are mapped to data values. Legends and axes display these mappings.
- Coordinate systems: define the plane on which data are mapped on the graphic.
- Faceting: splits the data into subsets to create multiple variations of the same graph (paneling).

Aesthetic mappings and aes

- Aesthetics are the visually perceivable components of the graph.
- Map variables to aesthetics using the aes function, such as:
  - *which variables appear on the x-axis and y-axis.*
  - *a classification variable to colors*
  - *a numeric variable to the size of graphical objects*

ggplot() template

```
ggplot(data = DATA) +
  GEOM_FUNCTION(mapping = aes(AESTHETIC MAPPINGS))
```

```
ggplot(data = troops) +
  geom_path(mapping = aes(x = longitude,
                          y = latitude,
                          color = direction,
                          size = survivors))
```

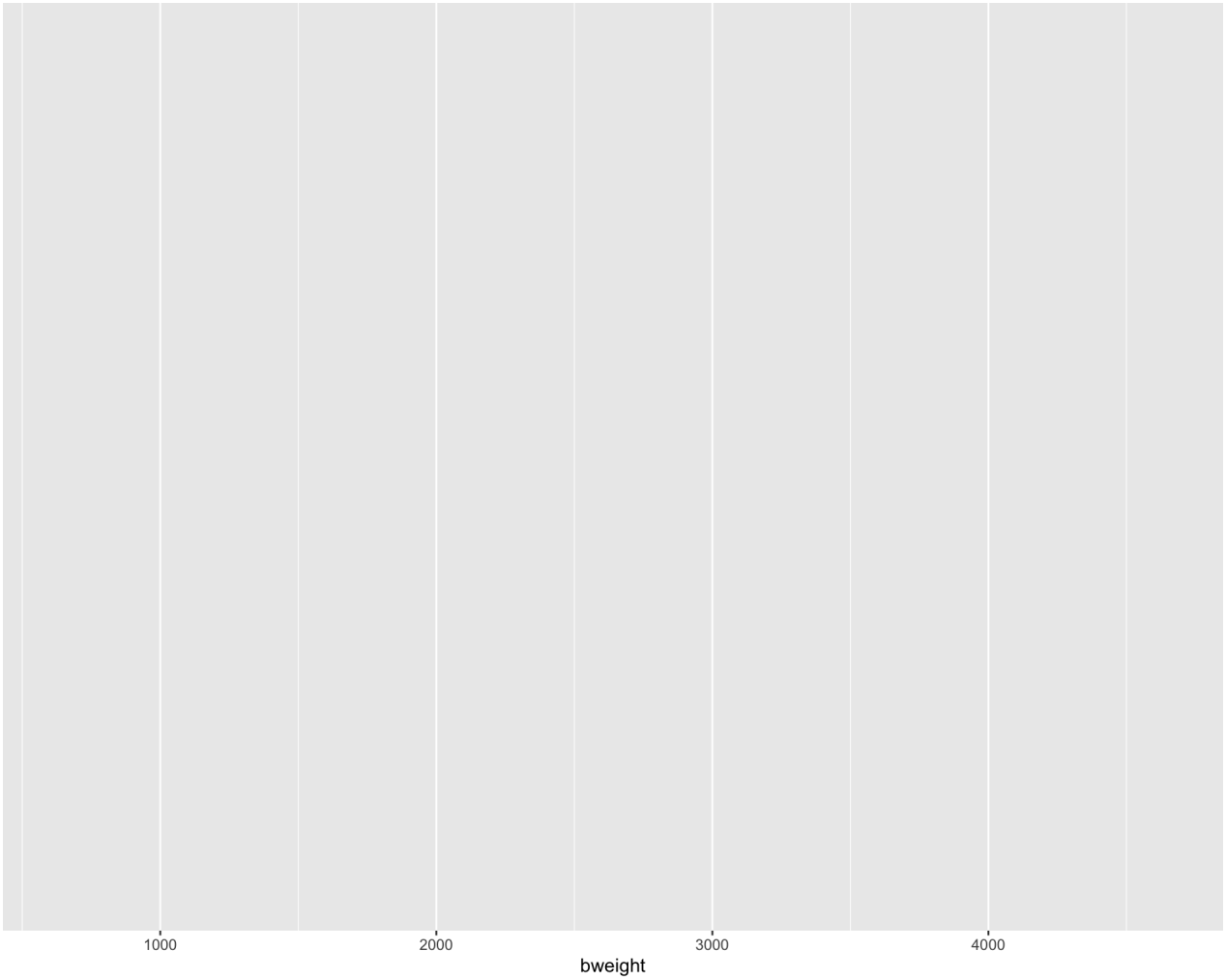
To build a ggplot

- Use the ggplot() function and bind the plot to a specific data frame using the data argument

```
ggplot(data = bw_df)
```

- Define an aesthetic mapping (using the aesthetic (aes) function), by selecting the variables to be plotted and specifying how to present them in the graph, e.g., as x

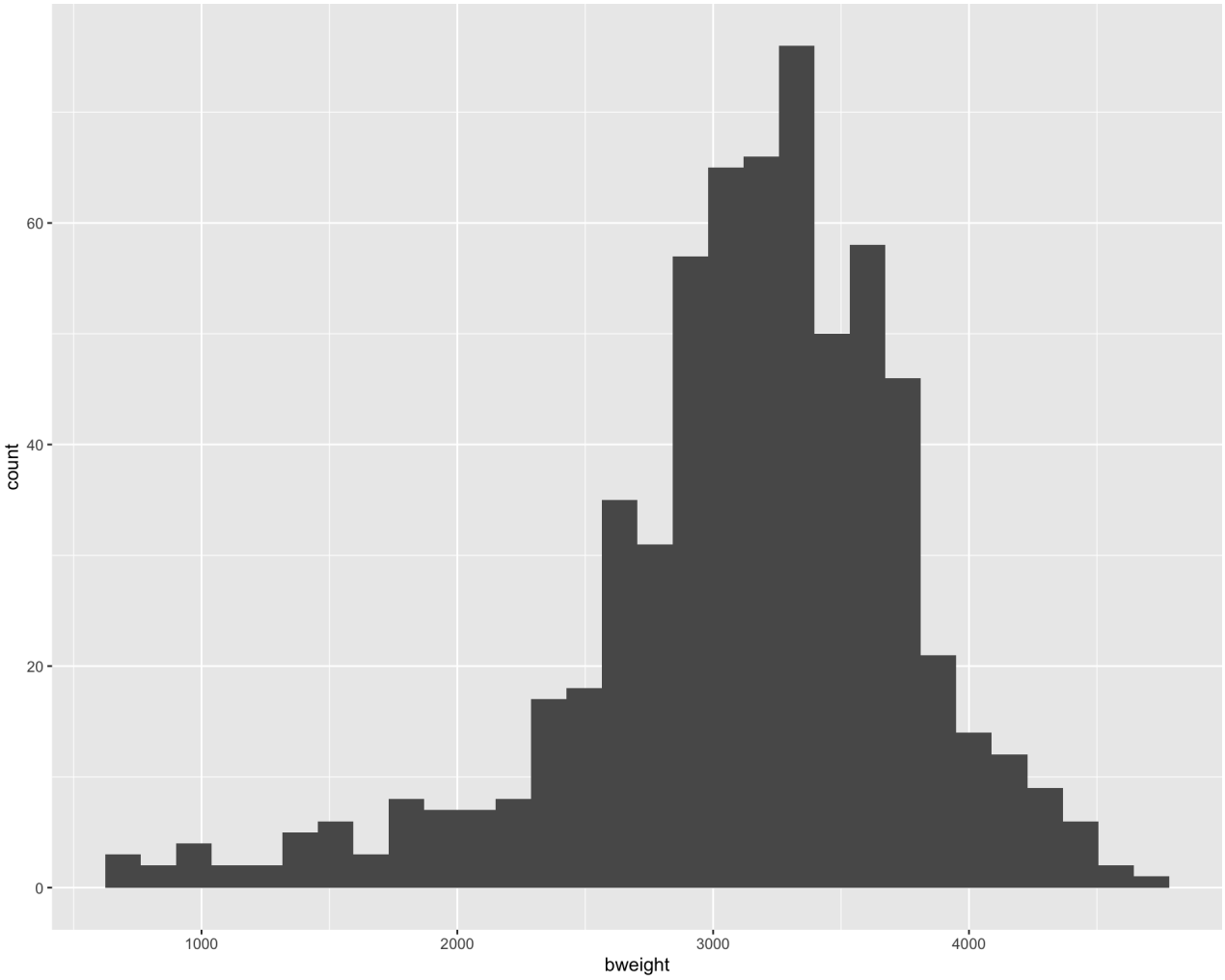
```
# declare data and x aesthetics, but no aesthetics
ggplot(data = bw_df, aes(x = bweight))
```



- Add ‘geoms’ – graphical representations of the data in the plot (histogram, density, bars). ggplot2 offers many different geoms; we will use some common ones today, including:
  - *geom_histogram() for histograms*
  - *geom_density() for density plots*
  - *geom_area() for area plots*
  - *geom_bar() for bar plots*
- To add a geom to the plot use + operator. Because we have one continuous variables, let’s use geom_histogram() first:
- You can easily set up plot “templates” and conveniently explore different types of plots, so the above plot can also be generated with code like this:

```r
# declare data and x aesthetics,
ggplot(data = bw_df, aes(x = bweight)) + geom_histogram()
```

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

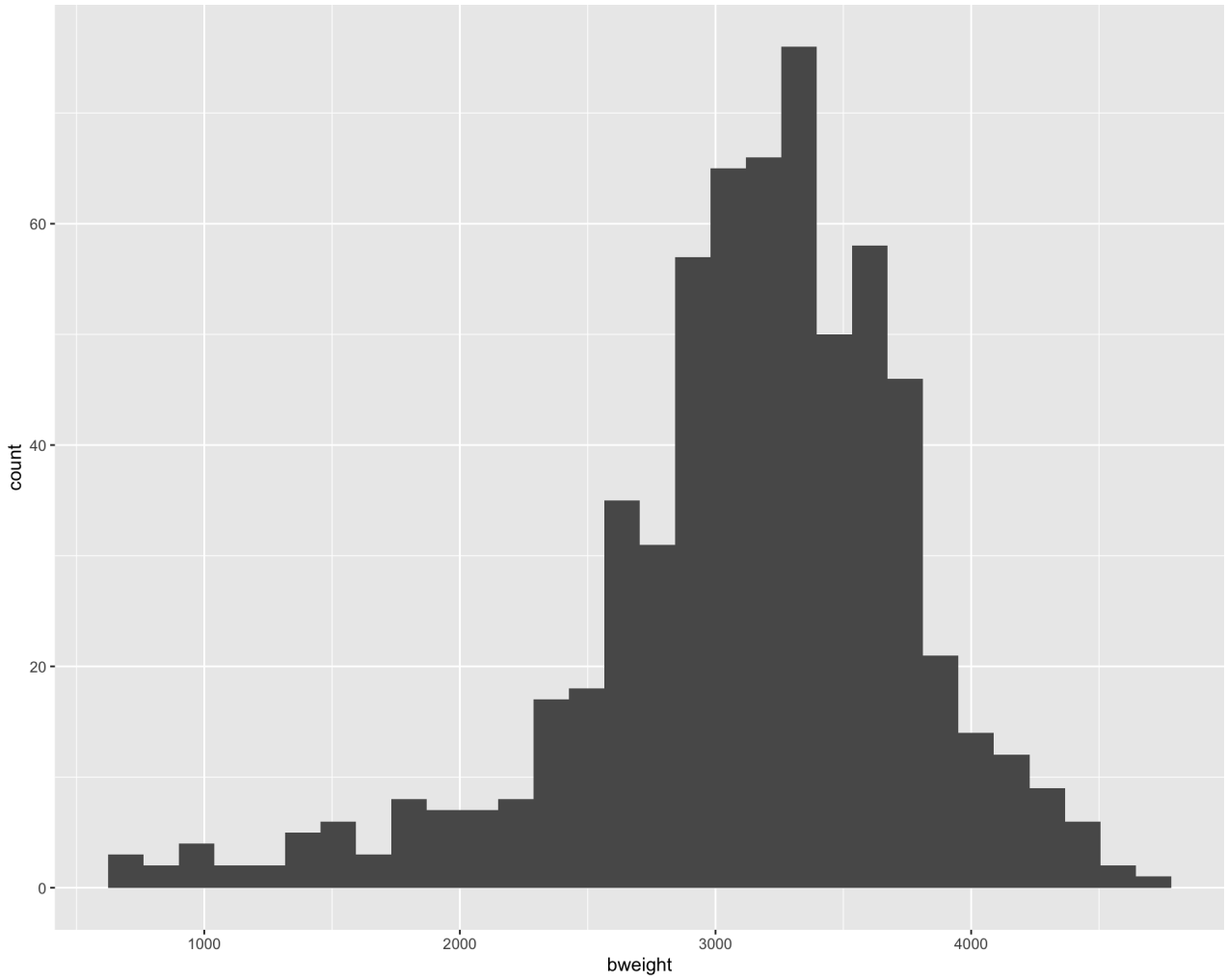


- The + in the ggplot2 package is particularly useful because it allows you to modify existing ggplot objects.
- The + sign used to add layers must be placed at the end of each line containing a layer.
- If, instead, the + sign is added in the line before the other layer, ggplot2 will not add the new layer and will return an error message.

```
# This will not add the new layer and will return an error
# message
ggplot(data = bw_df, aes(x = bweight)) + geom_histogram()
```

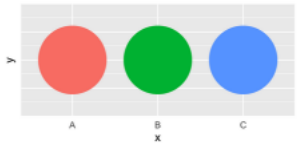
```
# This is the correct syntax for adding layers
ggplot(data = bw_df, aes(x = bweight)) + geom_histogram()
```

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

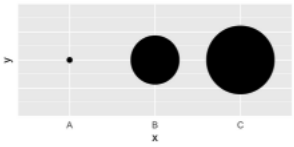


Aesthetics

`color` (discrete)

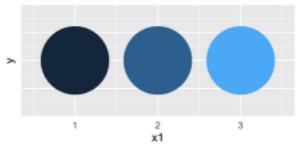


`size`

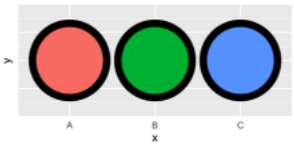


`shape`

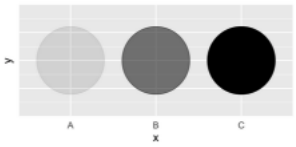
`color` (continuous)



`fill`

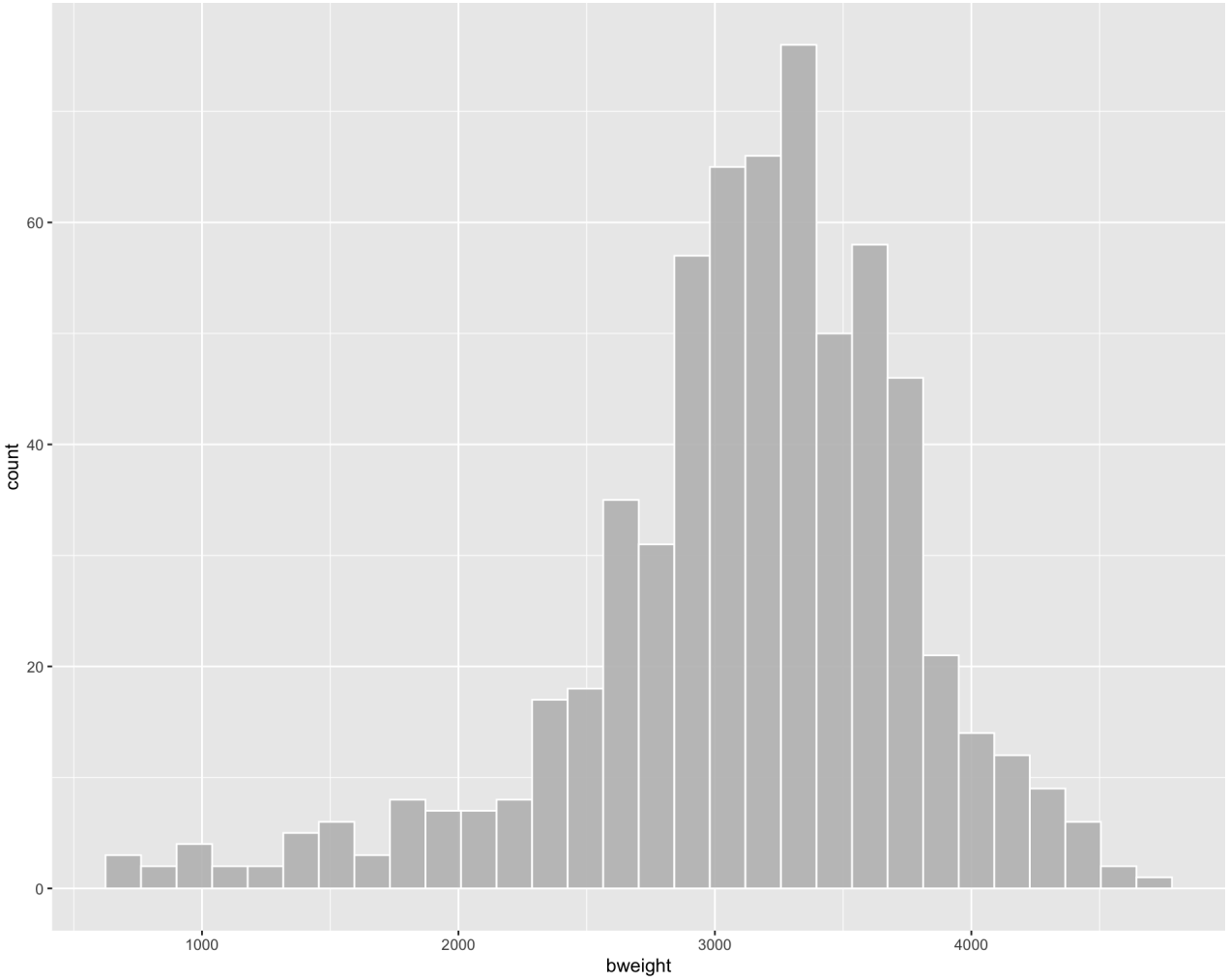


`alpha`



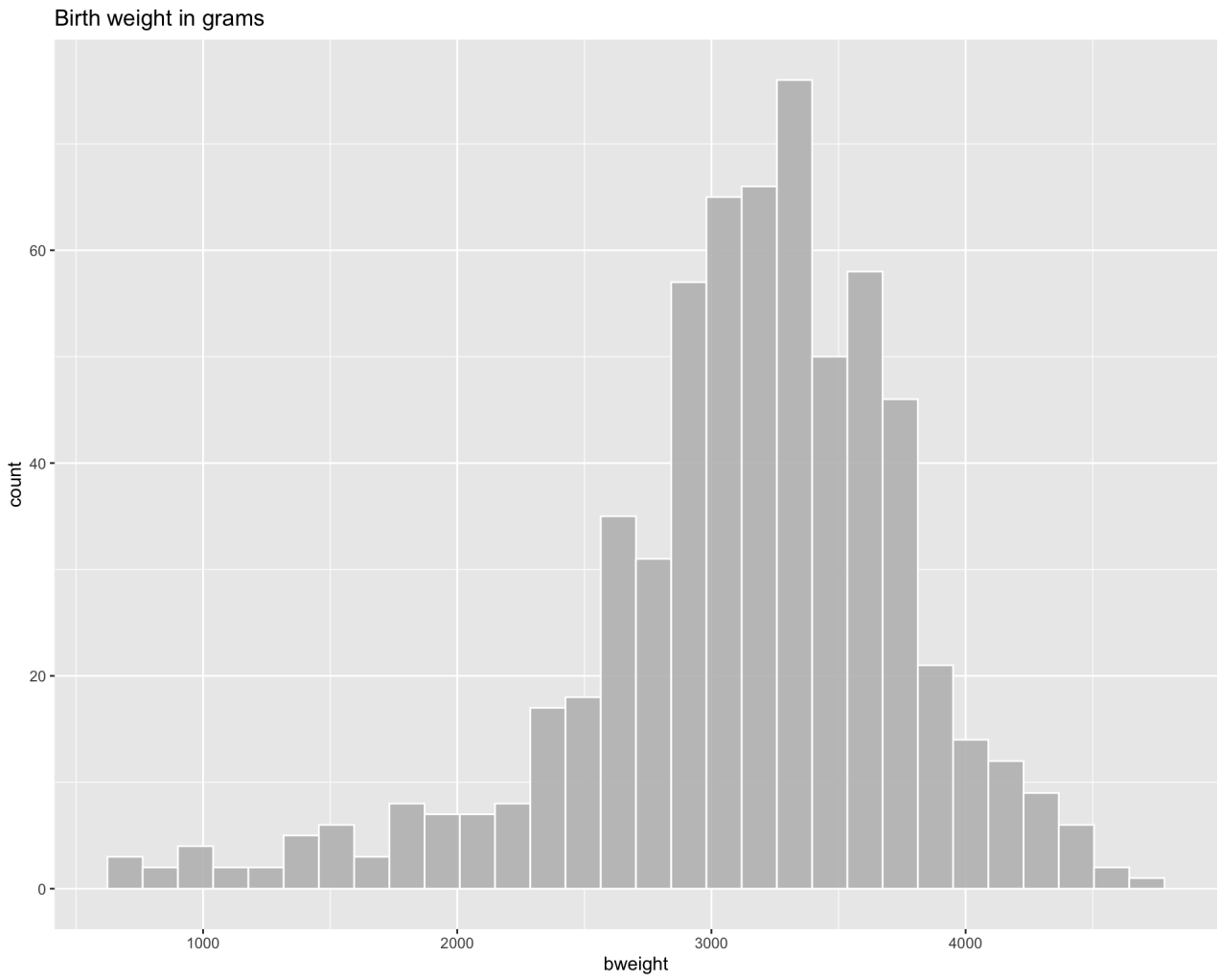
```r
# adding aesthetics to the histogram
ggplot(data = bw_df, aes(x = bweight)) + geom_histogram(fill = "gray",
    color = "white", alpha = 0.9)
```

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



```r
# adding title to the histogram
ggplot(data = bw_df, aes(x = bweight)) + geom_histogram(fill = "gray",
    color = "white", alpha = 0.9) + ggtitle("Birth weight in grams")
```

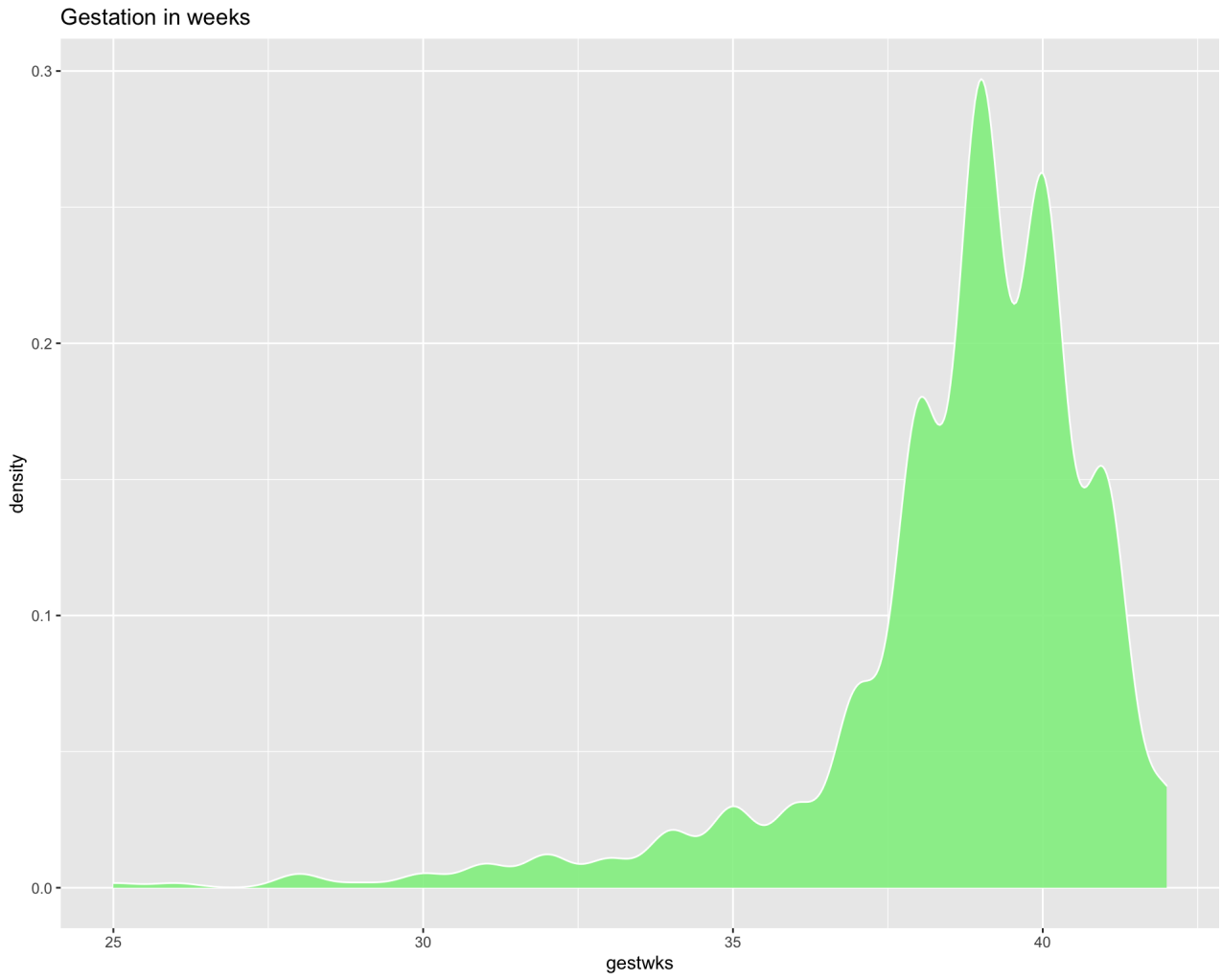
```
## `stat_bin()` using `bins = 30`. Pick better value with
## `binwidth`.
```



Density plot

- A density plot is a representation of the distribution of a numeric variable.
- It uses a kernel density estimate to show the probability density function of the variable.

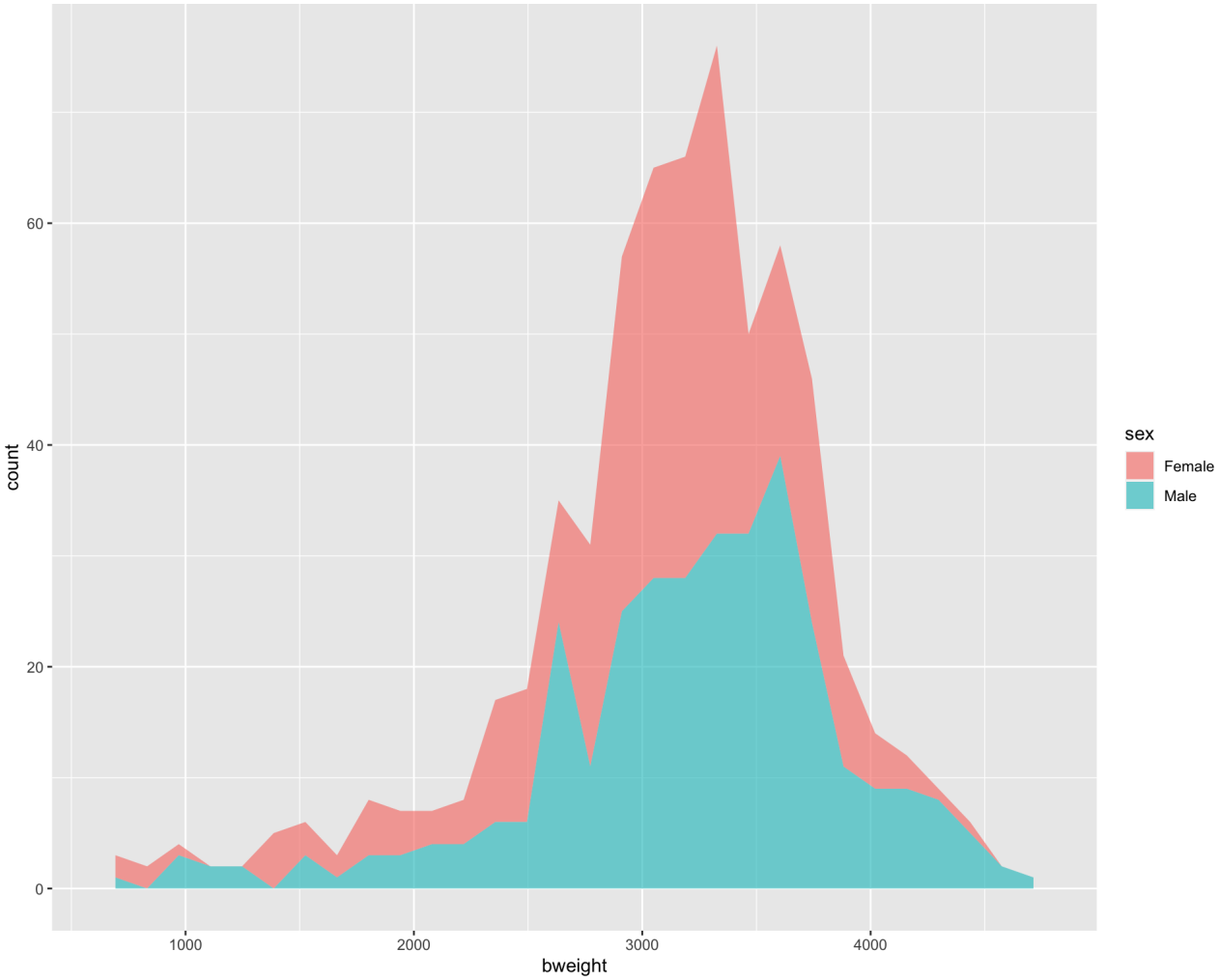
```
#
ggplot(data = bw_df, aes(x = gestwks)) + geom_density(fill = "light green",
    color = "white", alpha = 0.9) + ggtitle("Gestation in weeks")
```



Area plot with ggplot2

```
#
ggplot(data = bw_df, aes(x = bweight)) + geom_area(aes(fill = sex),
    stat = "bin", alpha = 0.6)
```

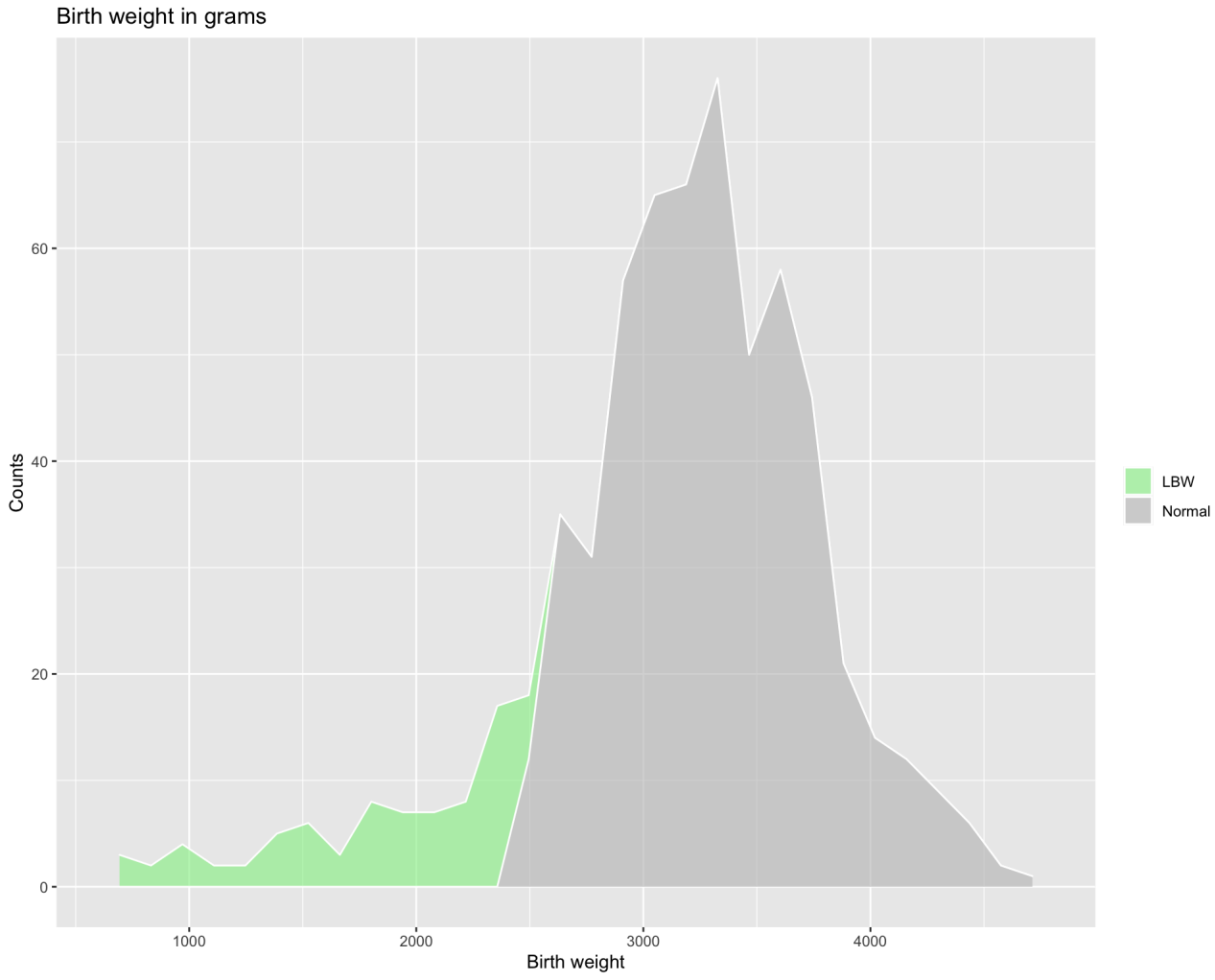
```
## `stat_bin()` using `bins = 30`. Pick better value with
## `binwidth`.
```



Integrating the pipe operator with ggplot2

```
# Make the with fill based on
bw_df <- bw_df %>% mutate(bw_cat = ifelse(bweight < 2500, "LBW",
    "Normal"))
ggplot(data = bw_df, aes(x = bweight, fill = bw_cat)) + geom_area(stat = "bin",
    color = "white", alpha = 0.6) + scale_fill_manual(values = c("light green",
    "gray")) + labs(fill = "") + ylab("Counts") + xlab("Birth weight") +
    ggtitle("Birth weight in grams")
```

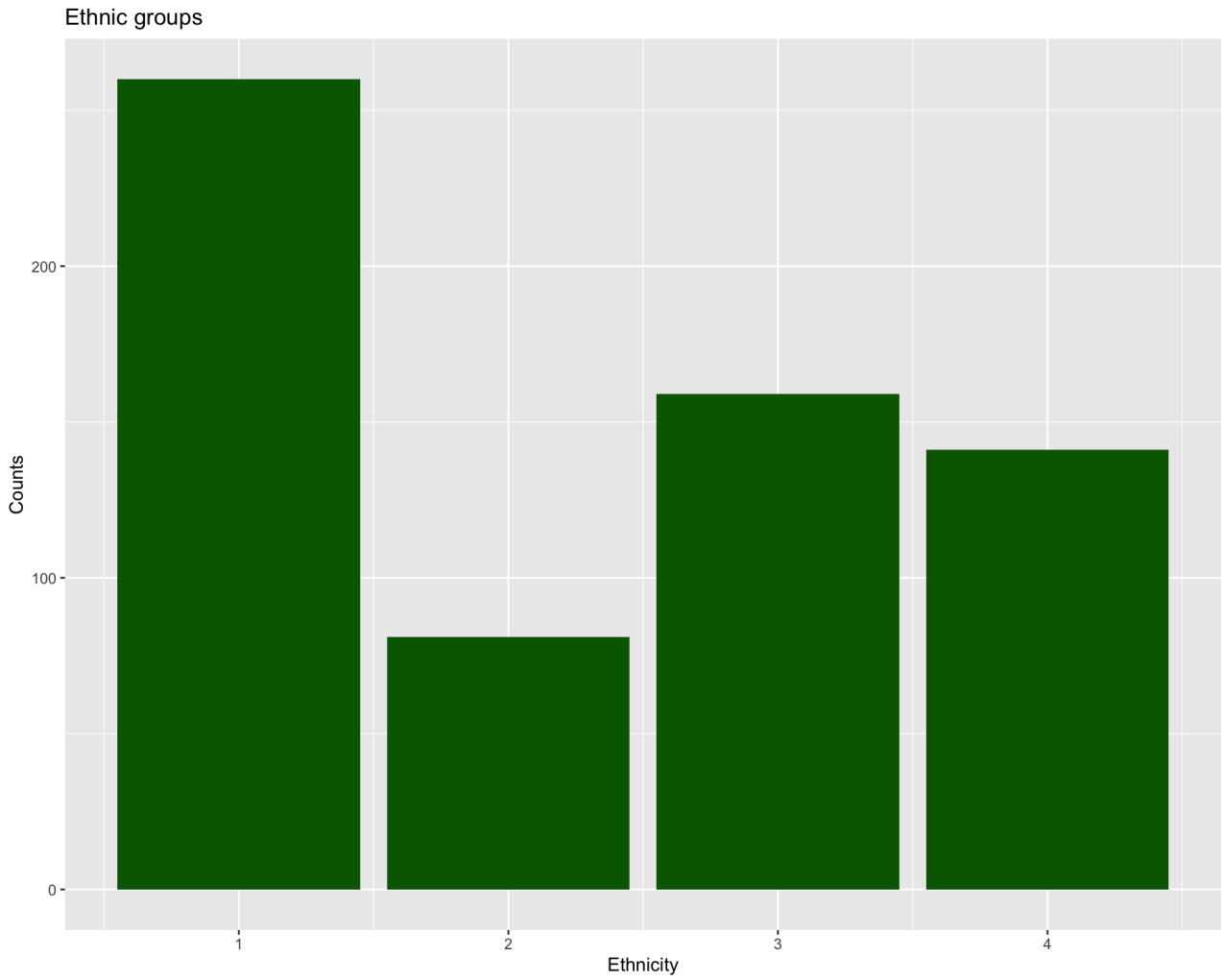
```
## `stat_bin()` using `bins = 30`. Pick better value with
## `binwidth`.
```



Plot for discrete variables

- A bar plot is one of the most common types of graphic for discrete/categorical variables
- Each entity of the categorical variable is represented as a bar and the size of the bar represents its numeric value.

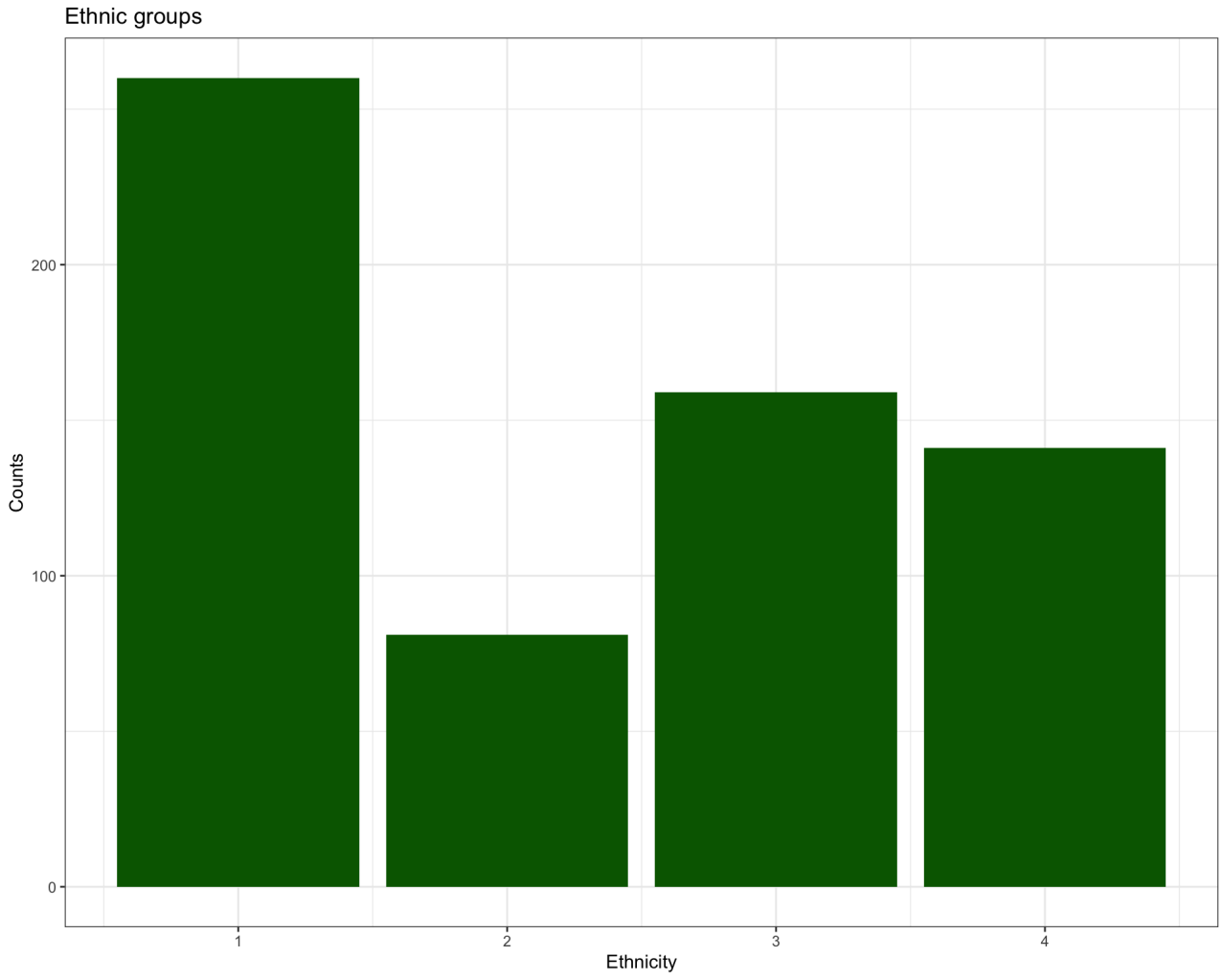
```
# Bar graphs for categorical variables
ggplot(bw_df, aes(x = ethnic)) + geom_bar(fill = "dark green") +
    ylab("Counts") + xlab("Ethnicity") + ggtitle("Ethnic groups")
```



ggplot2 themes

- Usually plots with white background look more readable when printed.
- Every single component of a ggplot graph can be customized using the generic theme() function, as we will see below.
- However, there are pre-loaded themes available that change the overall appearance of the graph without much effort.
- For example, we can change our previous graph to have a simpler white background using the theme_bw() function:

```
# Bar graphs for categorical variables
ggplot(bw_df, aes(x = ethnic)) + geom_bar(fill = "dark green") +
    ylab("Counts") + xlab("Ethnicity") + ggtitle("Ethnic groups") +
    theme_bw()
```



Themes

- The complete list of themes is available at https://ggplot2.tidyverse.org/reference/ggtheme.html

Other types of graphs

http://www.sthda.com/english/wiki/be-awesome-in-ggplot2-a-practical-guide-to-be-highly-effective-r-software-and-data-visualization

Exporting plots

```
my_plot <- ggplot(data = bw_df, aes(x = bweight)) + geom_area(aes(fill = sex),
    stat = "bin", alpha = 0.6) + theme_bw()
ggsave("Output/Ethnicity.png", my_plot, width = 15, height = 10)
```

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

```
my_plot
```

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