Import packages and data

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)

data <- read_csv("2020heights.csv")
```

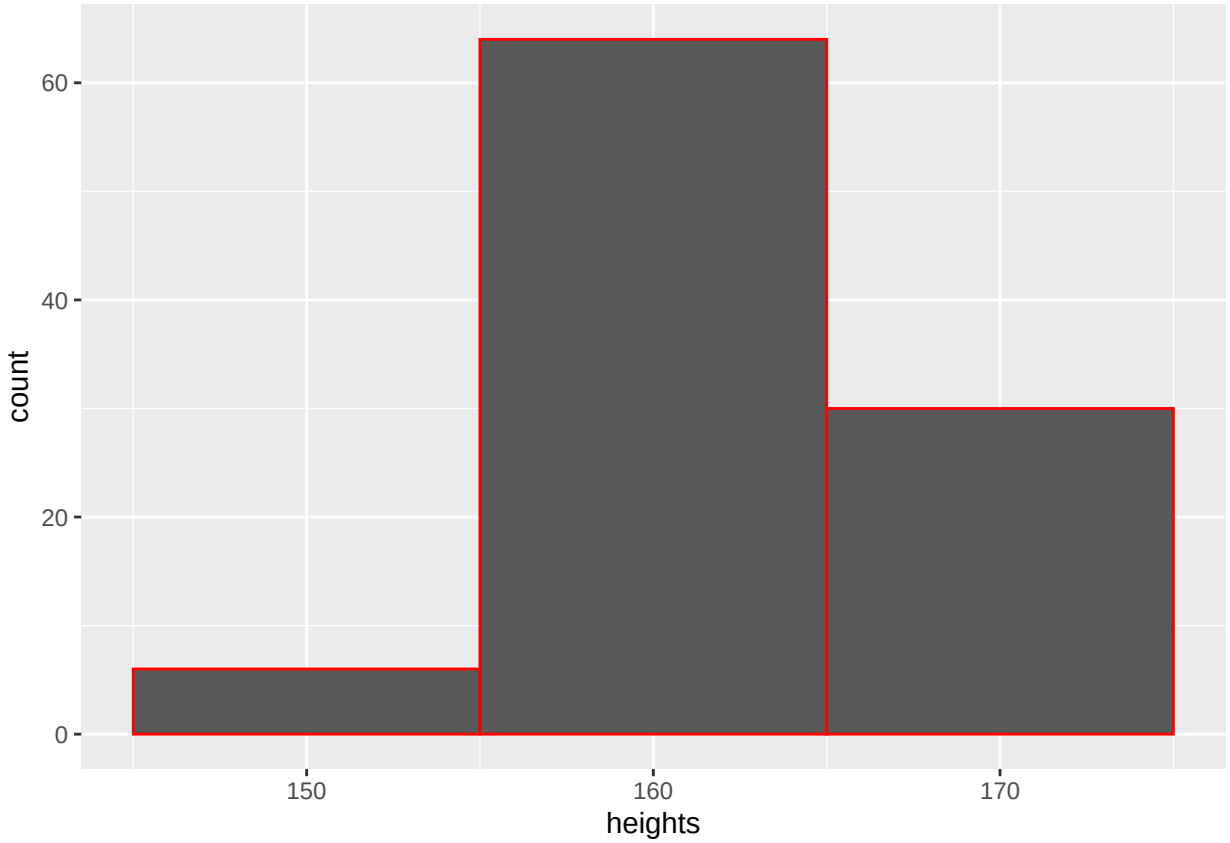
```
## Rows: 100 Columns: 1
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1): heights
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(data)
```

```
## # A tibble: 6 x 1
##   heights
##     <dbl>
## 1    154.
## 2    158.
## 3    160.
## 4    159.
## 5    163.
## 6    168.
```

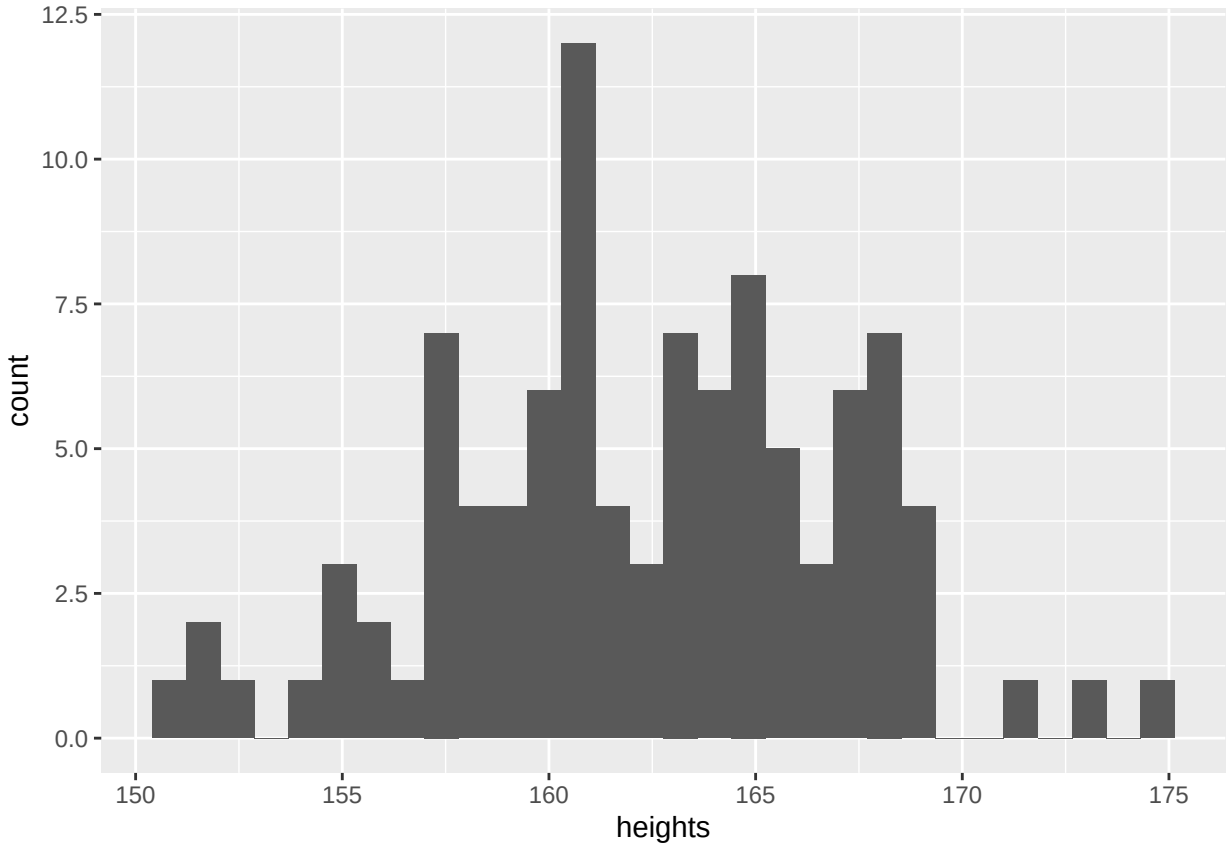
Plotting a Histogram

```
ggplot(data, aes(heights)) + geom_histogram(binwidth = 10, color="red")
```

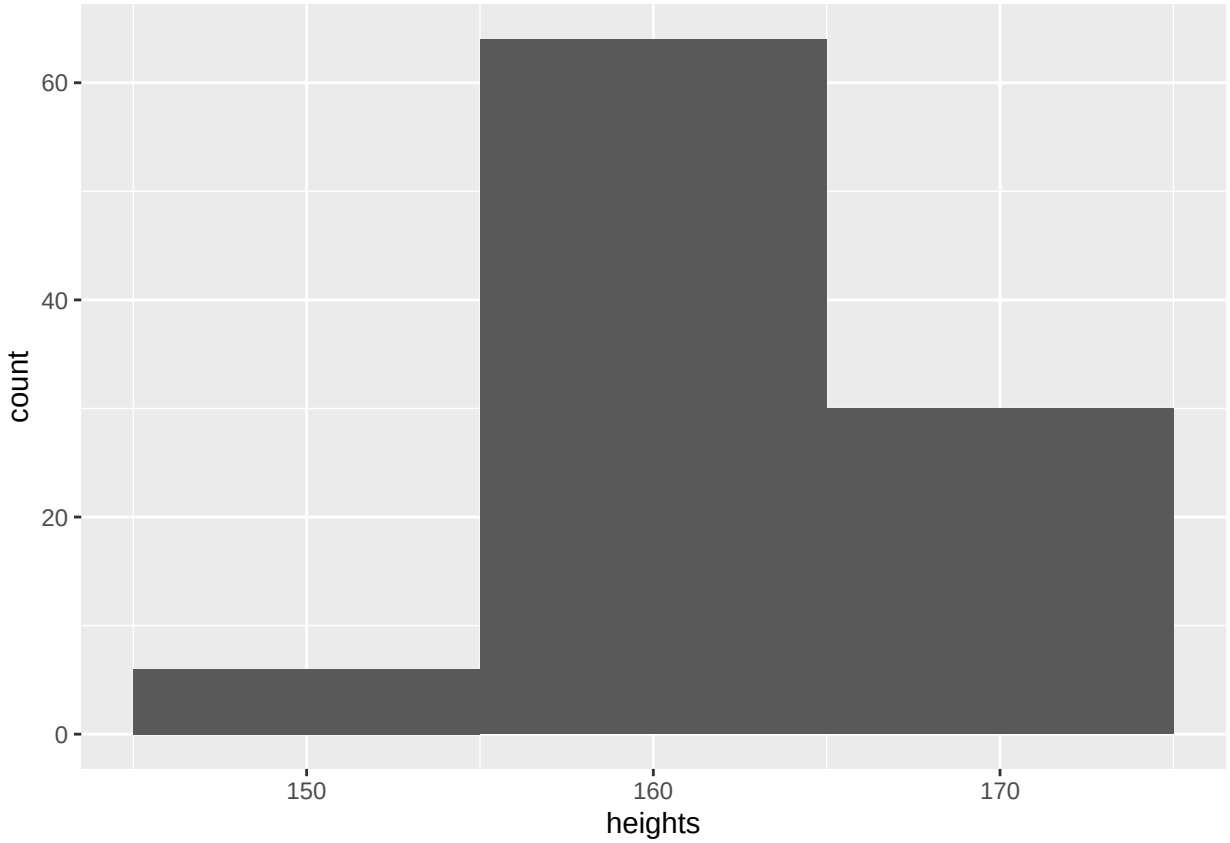



```
ggplot(data, aes(x=heights)) + geom_histogram()
```

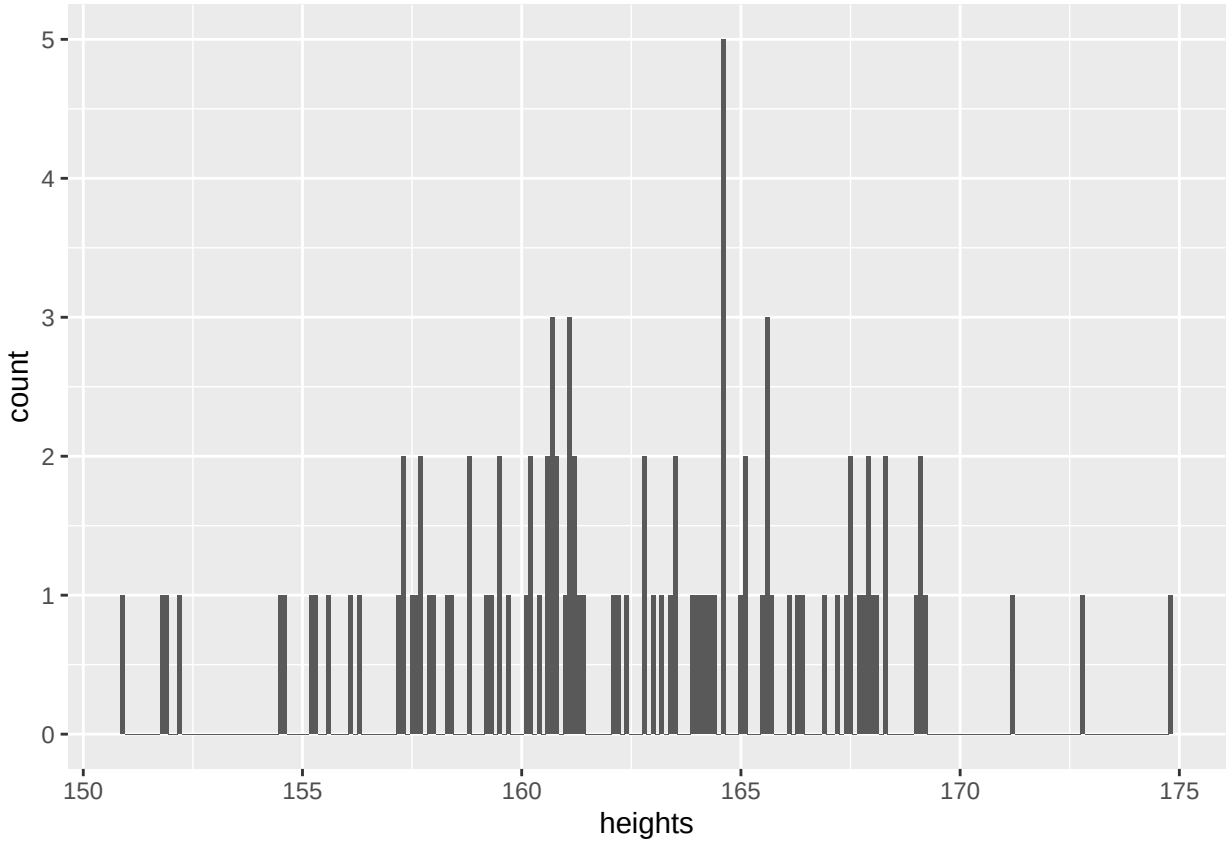
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



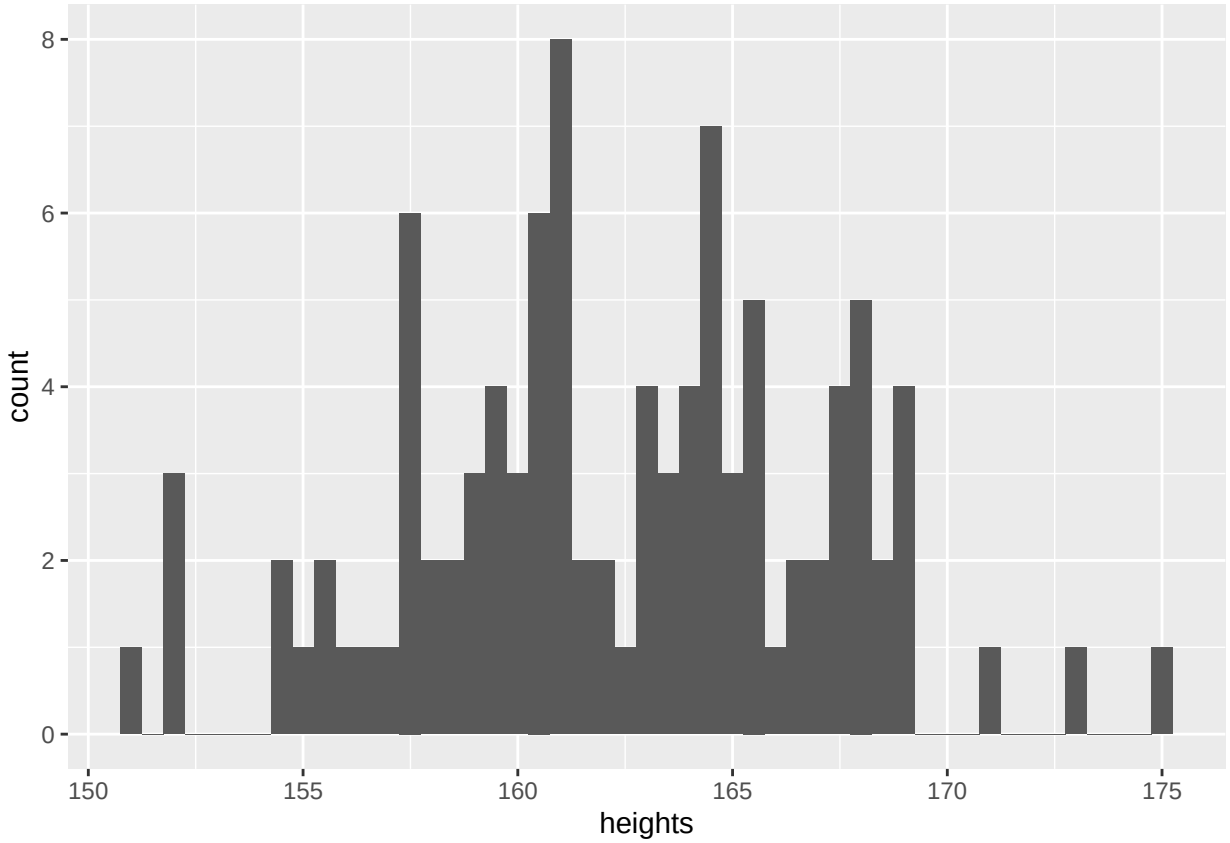

```
ggplot(data, aes(x=heights)) + geom_histogram(binwidth = 10)
```



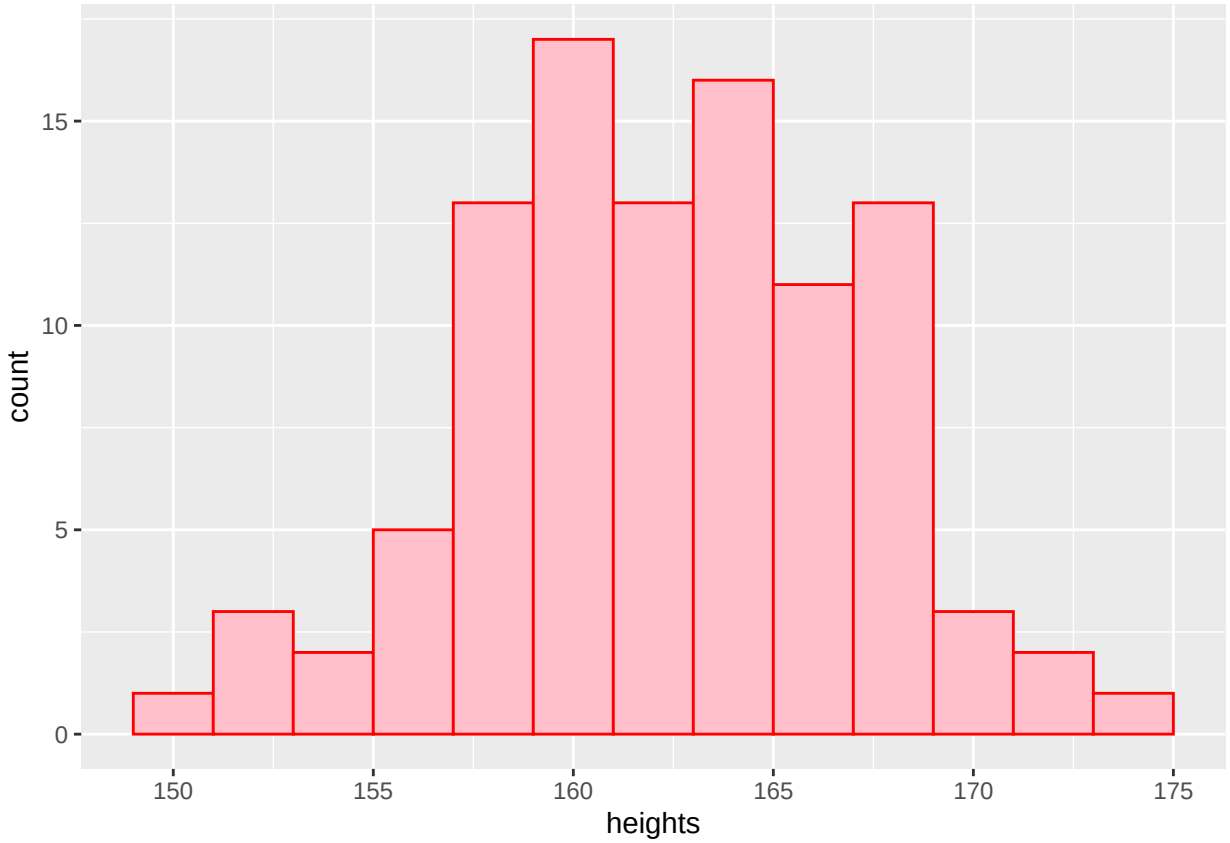

```
ggplot(data, aes(x=heights)) + geom_histogram(binwidth = 0.1)
```




```
ggplot(data, aes(x=heights)) + geom_histogram(binwidth = 0.5) # choose your own binwidth
```




```
ggplot(data, aes(x=heights)) + geom_histogram(binwidth = 2, fill = "pink", color="red")
```



Describe the general shape of the histogram here.

Pipes

```
data <- data %>% mutate(a=mean(heights))
head(data)
```

```
## # A tibble: 6 x 2
##   heights     a
##     <dbl> <dbl>
## 1    154.  162.
## 2    158.  162.
## 3    160.  162.
## 4    159.  162.
## 5    163.  162.
## 6    168.  162.
```

Chaining pipes

```
data <- data %>% mutate(b=sum(heights)) %>% arrange(heights)
head(data)
```

```
## # A tibble: 6 x 3
##   heights     a      b
##     <dbl> <dbl>  <dbl>
```

```
## 1    151.  162. 16234.
## 2    152.  162. 16234.
## 3    152.  162. 16234.
## 4    152.  162. 16234.
## 5    154.  162. 16234.
## 6    155.  162. 16234.
```

Calculating the Z score

```
data <- data %>% mutate(z_score =(heights-mean(heights))/sd(heights)) %>% arrange(abs(z_score))
```

What are the 5 observations with the most extreme (largest absolute values) z score?