DAY2 - plotting

This document contains the code that will be displayed during the second day of the course. It also includes extra code for better understanding of the concepts.

Dataset: superhero

We will use the hero dataset throughout the document.

There are two files related to superhero. first dataset contains various information about the superhero's appearance,their body measurements etc. There are 10 variables in this dataset. Variable descriptions are as follows: Name: Name of the superhero Gender: Gender of the superhero Eye color: Eye color of the superhero Race: Species of the superhero Height: Height of the superhero (in cms) Publisher: Comic category of the superhero Skin color: Skin color of the superhero Alignment: Superhero's nature. Weight: Weight of the superhero (in kgs)

The second dataset will be used on the third day of the course and records the superhero attributes such as their strength level,speed level etc.

Import data and make a few changes to better visualise the data later

```r
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   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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 error
```

```r
heroes = read.csv("../datasets/heroes_information.csv")
View(heroes)
str(heroes)
```

```
## 'data.frame':    734 obs. of  11 variables:
##  $ X         : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ name      : chr  "A-Bomb" "Abe Sapien" "Abin Sur" "Abomination" ...
##  $ Gender    : chr  "Male" "Male" "Male" "Male" ...
##  $ Eye.color : chr  "yellow" "blue" "blue" "green" ...
##  $ Race      : chr  "Human" "Icthyo Sapien" "Ungaran" "Human / Radiation" ...
##  $ Hair.color: chr  "No Hair" "No Hair" "No Hair" "No Hair" ...
##  $ Height    : num  203 191 185 203 -99 193 -99 185 173 178 ...
##  $ Publisher : chr  "Marvel Comics" "Dark Horse Comics" "DC Comics" "Marvel Comics" ...
##  $ Skin.color: chr  "-" "blue" "red" "-" ...
##  $ Alignment : chr  "good" "good" "good" "bad" ...
##  $ Weight    : num  441 65 90 441 -99 122 -99 88 61 81 ...
```

```r
heroes$Publisher = as.factor(heroes$Publisher) # change Publisher variable to factor
heroes$Gender = as.factor(heroes$Gender) # change Gender variable to factor
levels(heroes$Gender)= c("Not Specified", "Female", "Male")# Define a new level as "Not Specified"
```

Missing values

```r
# Let´s create a vector with multiple NAs
my_data = c(3, 4, NA, 4, 5, NA, 1, NA, NA)

# Test if there are NAs in our data
is.na(my_data) # Gives us a logical vector
```

```
## [1] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE  TRUE
```

```r
which(is.na(my_data)) # gives us the location (indexes) of these NAs (TRUE)
```

```
## [1] 3 6 8 9
```

```r
# However if we are testing a large vector
sum(is.na(my_data)) # identify number of NA
```

```
## [1] 4
```

```r
table(is.na(my_data)) # counts number of NA
```

```
##
## FALSE  TRUE
##     5     4
```

```r
# We can test a column of our hero dataset
which(is.na(heroes$Weight))
```

```
## [1] 287 390
```

```r
# Remove NA
remove_na = is.na(my_data) # logical vector which information about where NAs are
my_data = my_data[!remove_na] # subset my_data to not include the TRUE elements
my_data
```

```
## [1] 3 4 4 5 1
```

```r
is.na(my_data)
```

```
## [1] FALSE FALSE FALSE FALSE FALSE
```

```r
# Now let's work with a data frame
data.frame = data.frame (c1 = 1:8, c2 = factor (c("a", "b", "a", "c", "b", "c", "a", "b")))
View(data.frame)
data.frame[4,1] = data.frame[6,2] = NA

# Test if there are NAs
is.na(data.frame)
```

```
##         c1    c2
## [1,] FALSE FALSE
## [2,] FALSE FALSE
## [3,] FALSE FALSE
## [4,]  TRUE FALSE
## [5,] FALSE FALSE
## [6,] FALSE  TRUE
## [7,] FALSE FALSE
## [8,] FALSE FALSE
```

```
table(is.na(data.frame))
```

```
##
## FALSE  TRUE
##    14     2
```

```
# Filter missing data
na.omit(data.frame)
```

```
##   c1 c2
## 1  1  a
## 2  2  b
## 3  3  a
## 5  5  b
## 7  7  a
## 8  8  b
```

```
View(data.frame)
```

Recoding missing values

In our dataset heroes we can see that a lot of missing values are actually treated as -. We can recode this by using normal subsettings and assignment operations.

```
# Let's work on a similar data.frame
data.frame = data.frame (c1 = 1:8, c2 = factor (c("a", "-", "a", "c", "b", "c", "-", "b")))
View(data.frame)

# Check if there is any NA
is.na(data.frame)
```

```
##         c1    c2
## [1,] FALSE FALSE
## [2,] FALSE FALSE
## [3,] FALSE FALSE
## [4,] FALSE FALSE
## [5,] FALSE FALSE
## [6,] FALSE FALSE
## [7,] FALSE FALSE
## [8,] FALSE FALSE
```

```
# However we know that column 2 misses some values as -
sum(data.frame == "-")
```

```
## [1] 2
```

```
# Let´s recode them to NAs
data.frame[data.frame == "-"] = NA
is.na(data.frame)
```

```
##         c1    c2
## [1,] FALSE FALSE
## [2,] FALSE  TRUE
## [3,] FALSE FALSE
## [4,] FALSE FALSE
## [5,] FALSE FALSE
## [6,] FALSE FALSE
## [7,] FALSE  TRUE
## [8,] FALSE FALSE
```

```r
View(data.frame)

# The hero dataset has some missing values that are not strictly shown as NA. Can you identify any and
```

Merging data

When you have related information in two or more data.frames, you can use the `merge()` function to join them correctly. The two data.frames that you want to join must have one or more columns with the same name - these are the ID columns that R uses to reorder the rows to match in both data.frames. If there are equivalent columns but the names are different, the column name in one has to be renamed to match the name in the other data.frame (as in the following example).

```r
# heroes <- read.csv("../datasets/heroes_information.csv")
powers <- read.csv("../datasets/super_hero_powers.csv")

colnames(powers) <- sub("hero_names", "name", colnames(powers))

# see the difference between the following
heroes_with_powers <- merge(heroes, powers, by = "name")
heroes_with_powers <- merge(heroes, powers, by = "name", all = TRUE)
heroes_with_powers <- merge(heroes, powers, by = "name", all.x = TRUE)
heroes_with_powers <- merge(heroes, powers, by = "name", all.y = TRUE)
```

Note that there are duplicated names in the `heroes` dataset. Because of this, the merging may give unexpected results. Try doing the same as the above, after removing the duplicated superheroes. Hint: you can filter out the duplicated superheroes using the `duplicated()` function on the `name` column.

```r
# Merge after removing duplicates
```

Reshaping data

Sometimes the data is not in the right format for the plot or analysis you want to run. You may need to reshape the data into wide or long format. The `tidyr` package (part of tidyverse) has the `pivot_` functions for this. An alternative is the `data.table` package with the `melt` and `dcast` functions.

```r
powers %>%
  pivot_longer(cols = -name, names_to = "power", values_to = "value") -> powers_long

powers_long %>%
  pivot_wider(names_from = "power", values_from = "value") -> powers_wide
```

Basic statistics

Are the mean weights of male and female superheroes different? Is this difference statistically significant? Let's set our significance cutoff at 0.5

```r
male_heroes_weight <- heroes[heroes$Gender == "Male", "Weight"]
female_heroes_weight <- heroes[heroes$Gender == "Female", "Weight"]
```

```r
t.test(male_heroes_weight, female_heroes_weight)
```

```
##
##  Welch Two Sample t-test
##
## data:  male_heroes_weight and female_heroes_weight
## t = 2.5818, df = 487.13, p-value = 0.01012
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   5.923307 43.649867
## sample estimates:
## mean of x mean of y
##  52.05159  27.26500
```

Is the weight of a superhero correlated to their height?

```r
cor.test(heroes$Weight, heroes$Height)
```

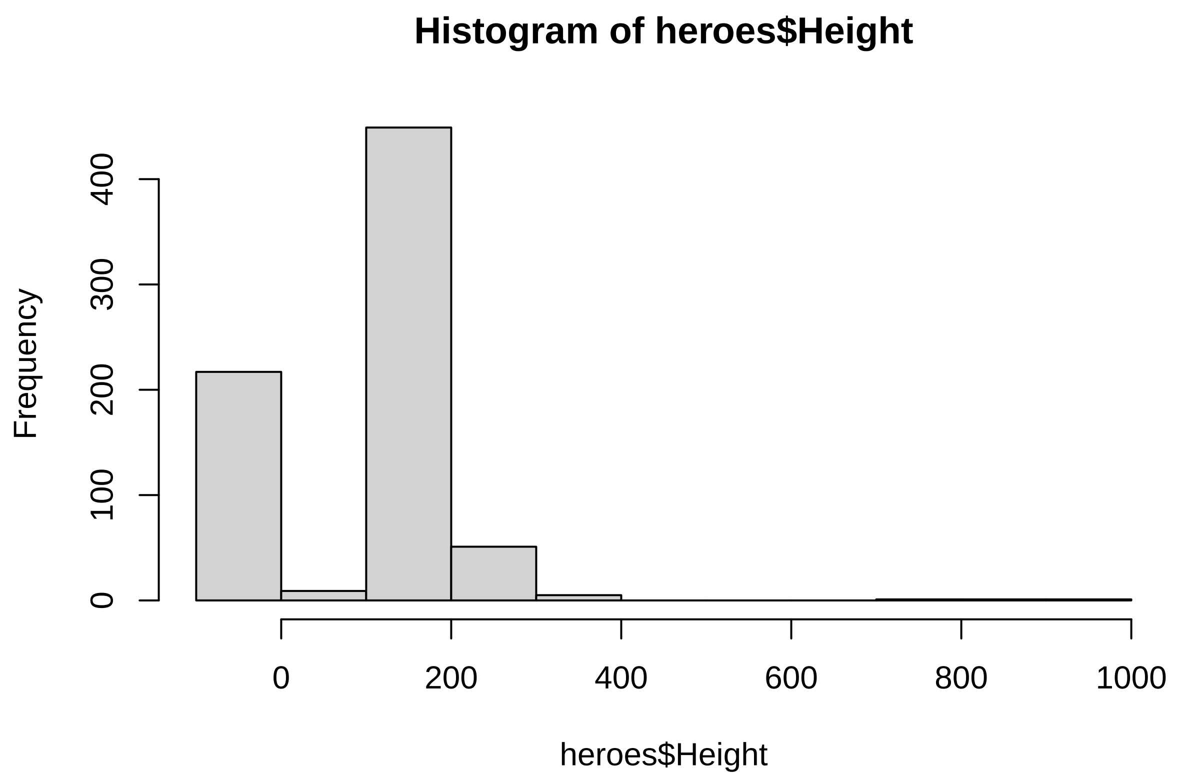
```
##
##  Pearson's product-moment correlation
##
## data:  heroes$Weight and heroes$Height
## t = 25.29, df = 730, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6427390 0.7201754
## sample estimates:
##       cor
## 0.6833745
```

DAY 2 AFTERNOON SESSIONS: PLOTTING WITH BASE R AND GGPLOT2

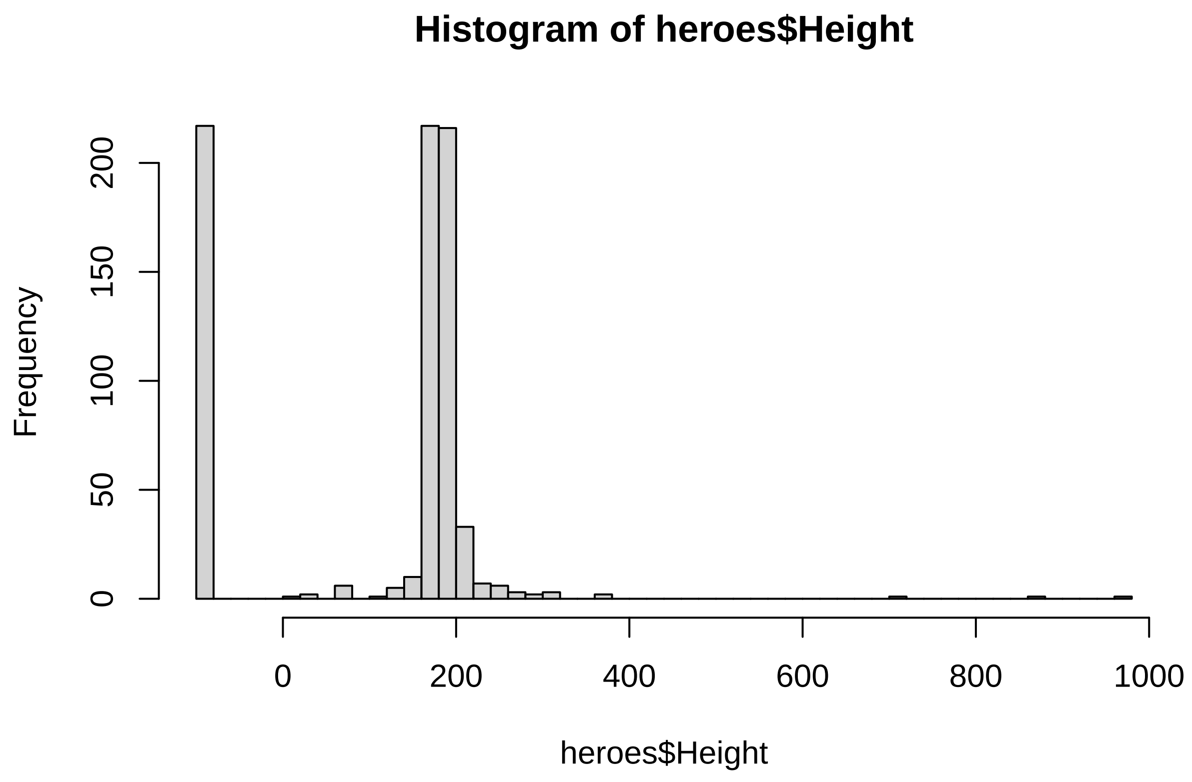
Base R

Let´s start by exploring basic R plotting. It is important to understand there is no need to remember all the arguments and optional things we can do with graphs. The important is to understand how plotting works and know the sources to get the information to reach your desired plot.

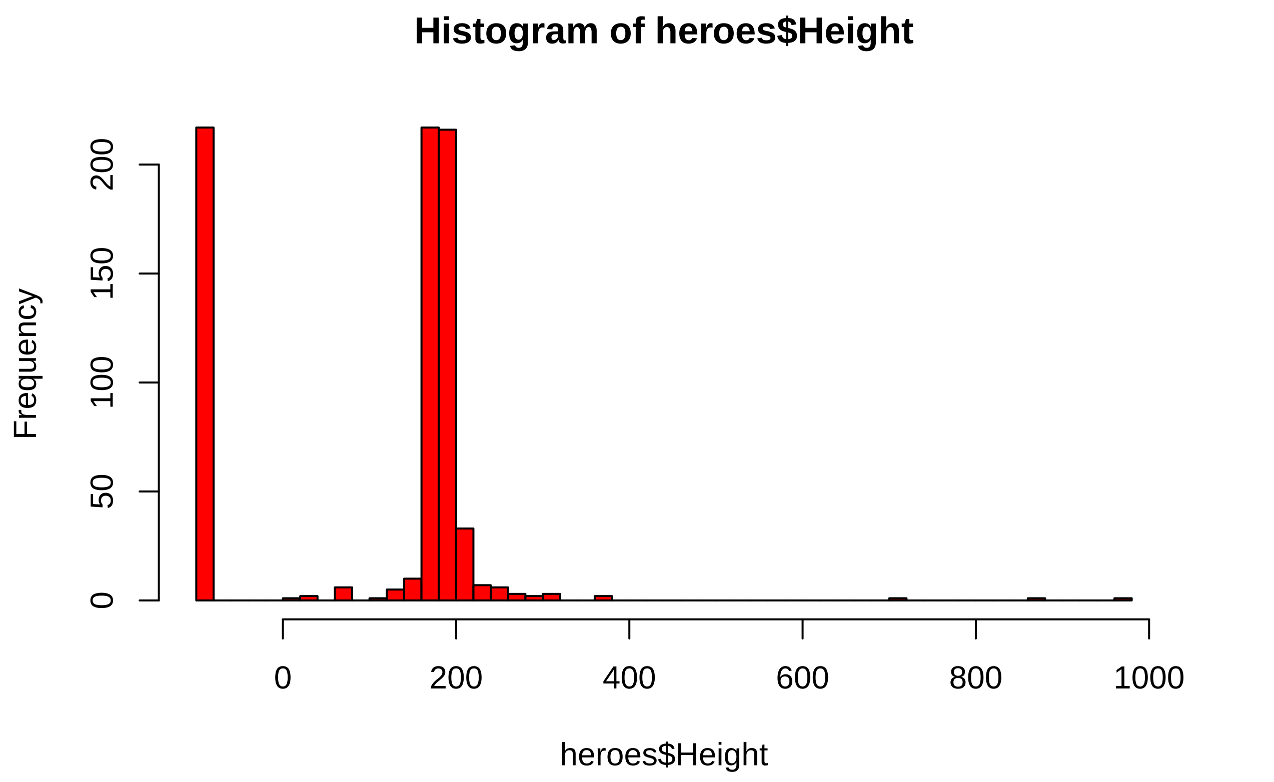
```r
# Histograms
hist(heroes$Height) # Histogram to see distribution
```



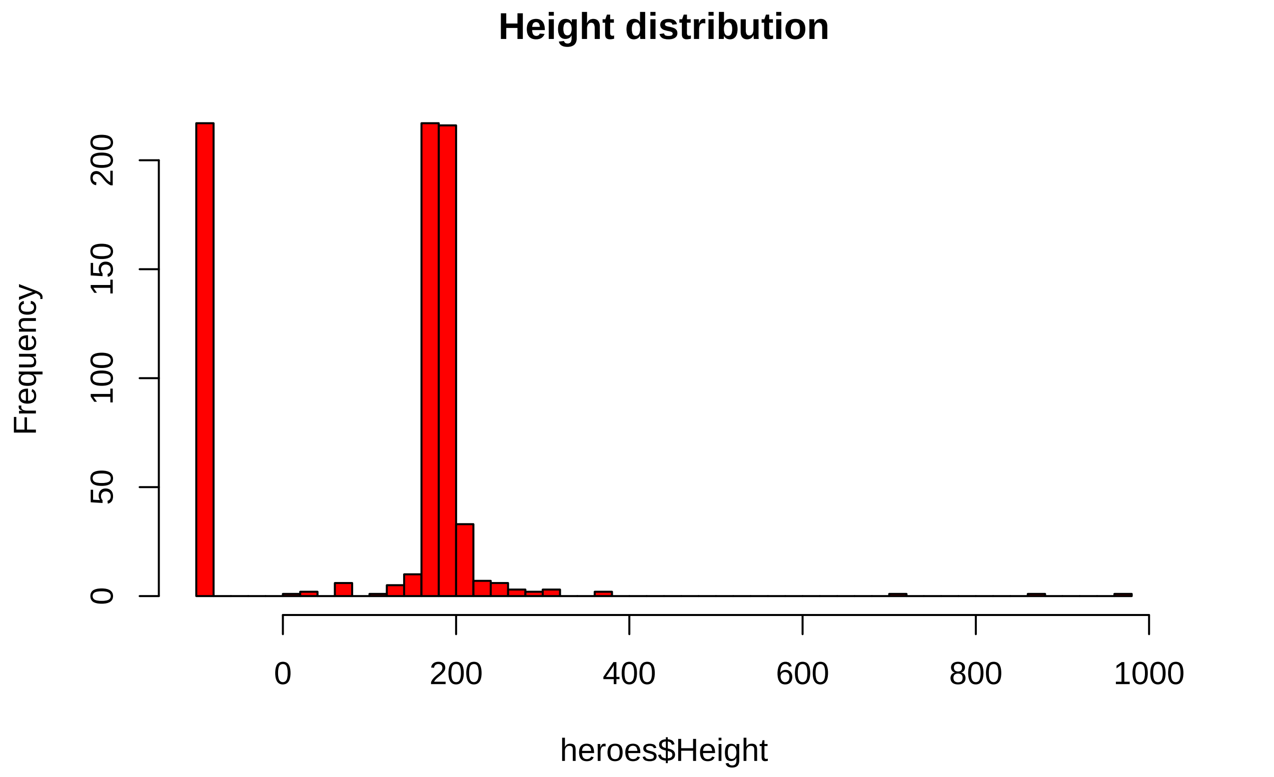
```r
hist(heroes$Height, breaks=50) # More detail
```



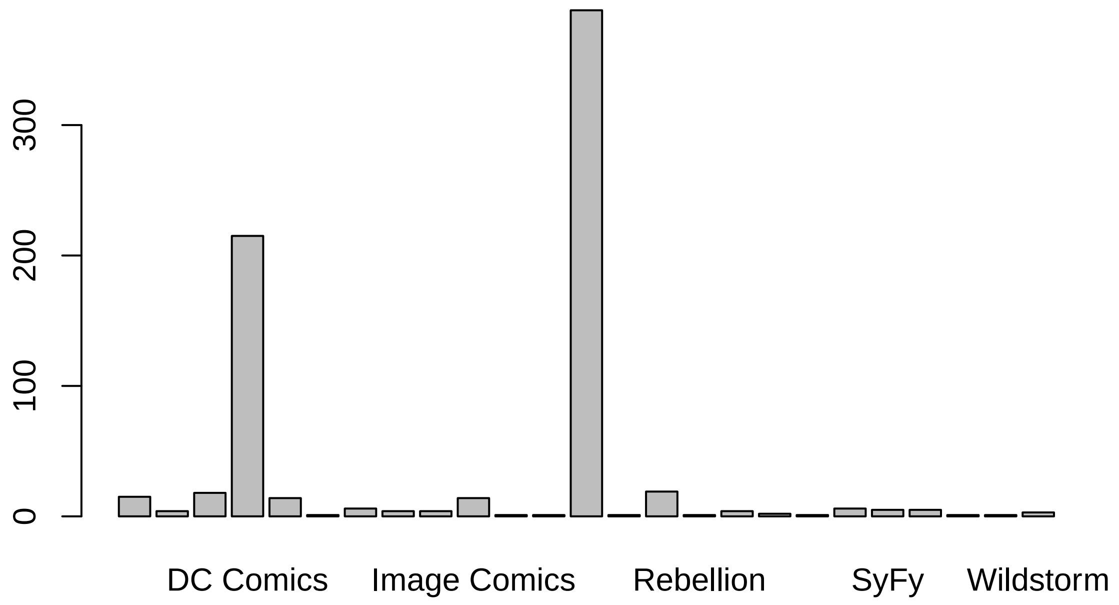
```r
hist(heroes$Height, breaks=50, col="red") # Add colour
```



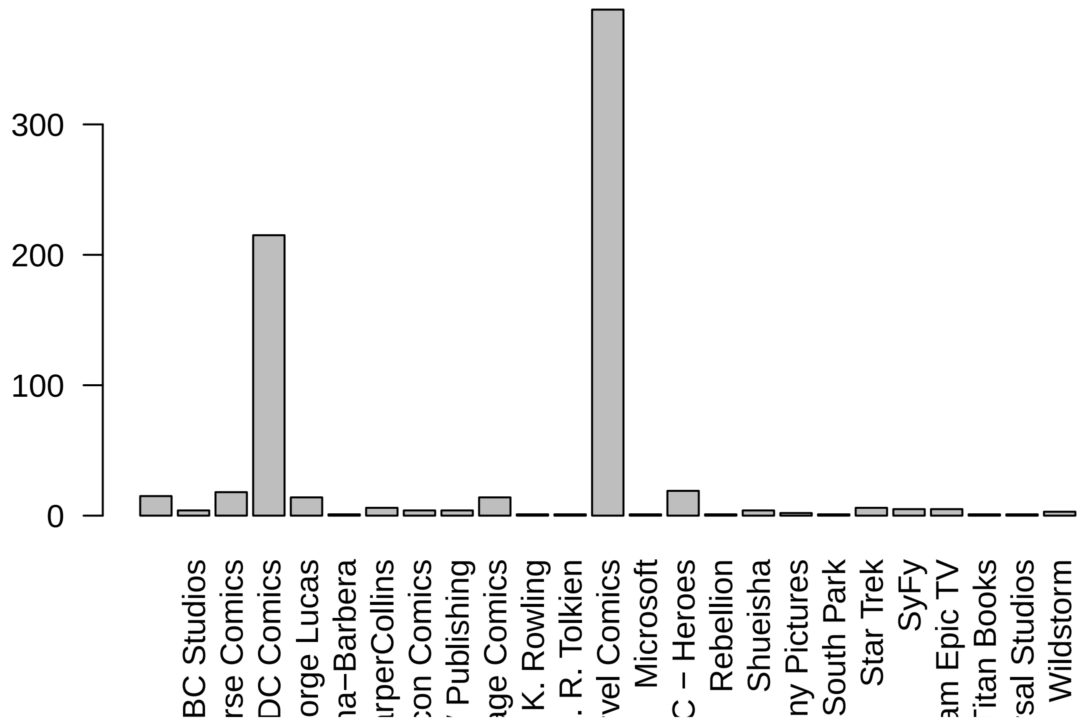
```
hist(heroes$Height, breaks=50, col="red", main = "Height distribution") # Add title
```



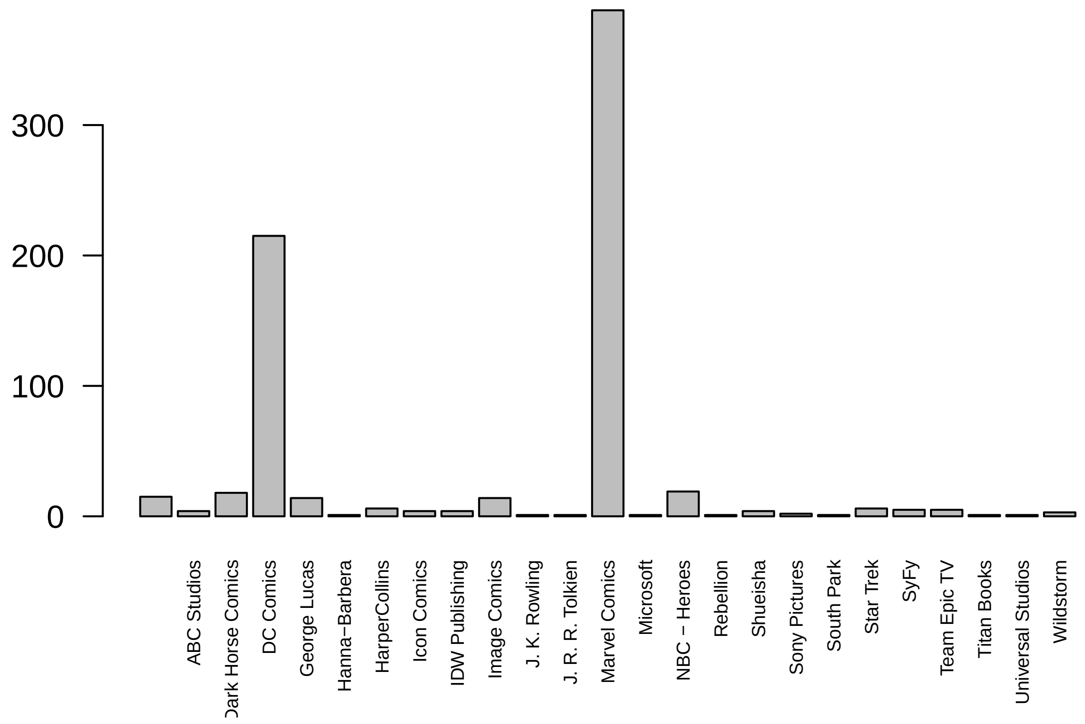
```
# Barplots
plot(heroes$Publisher) # Barplot
```



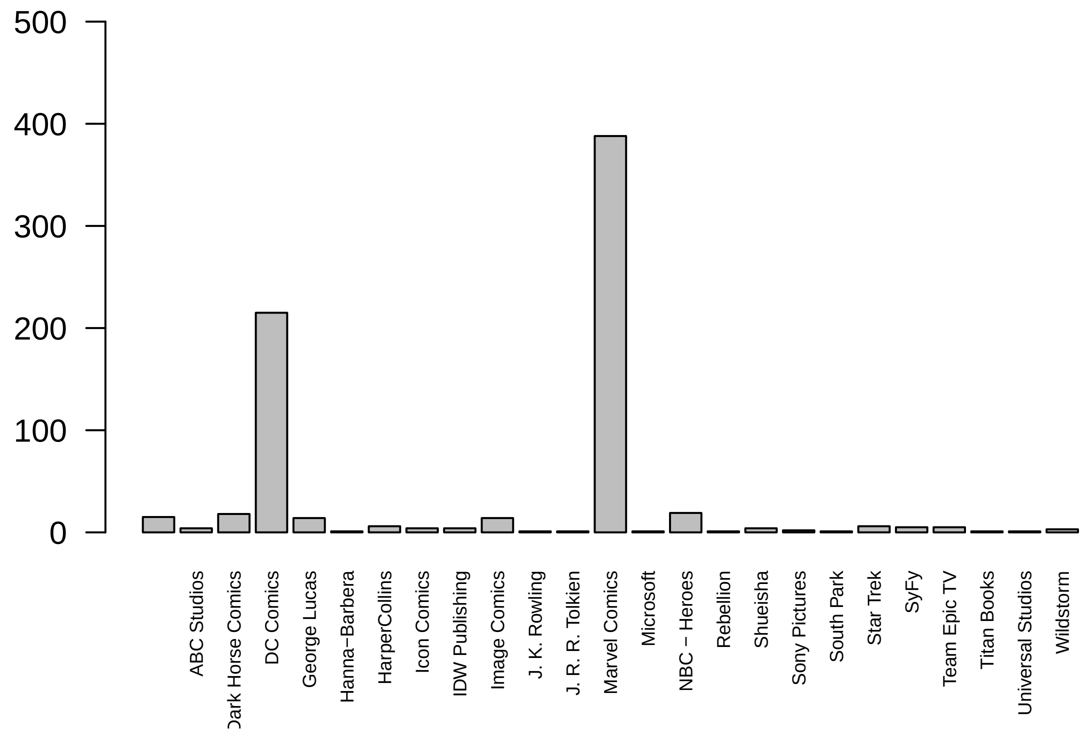
```r
plot(heroes$Publisher, las=2) # turn 90 degrees x labels
```



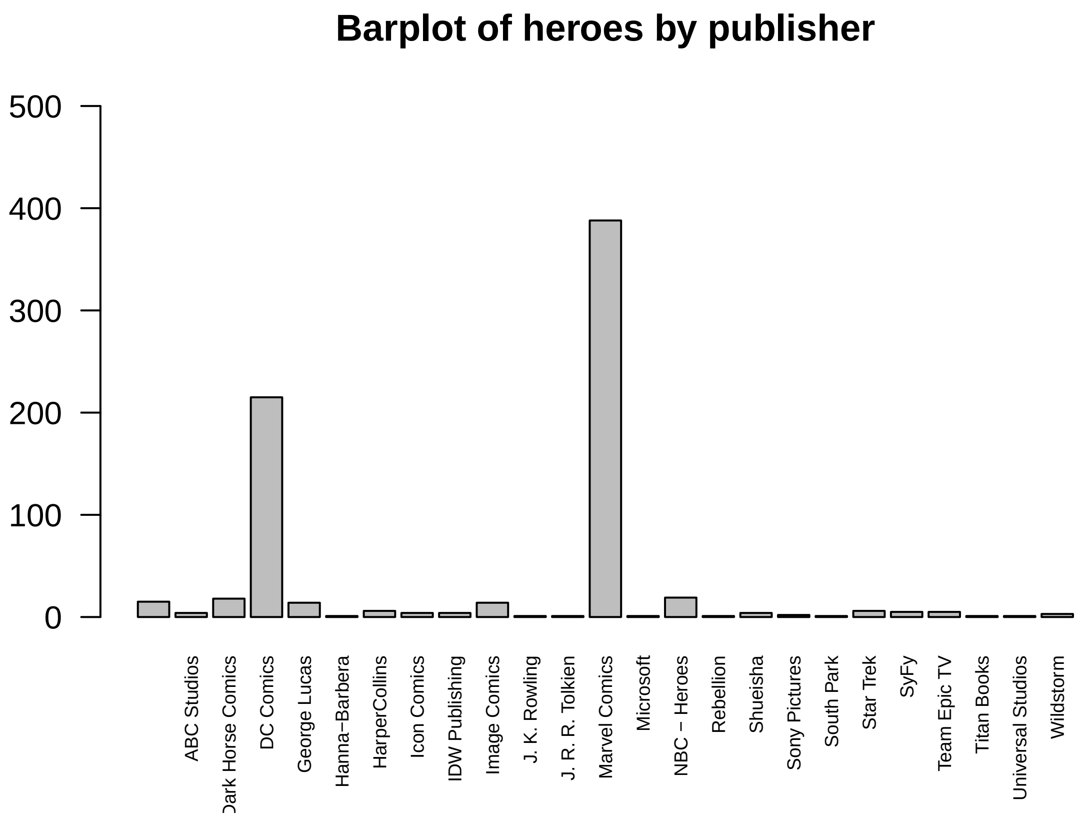
```r
plot(heroes$Publisher, las=2, cex.names=0.6) # change size labels
```



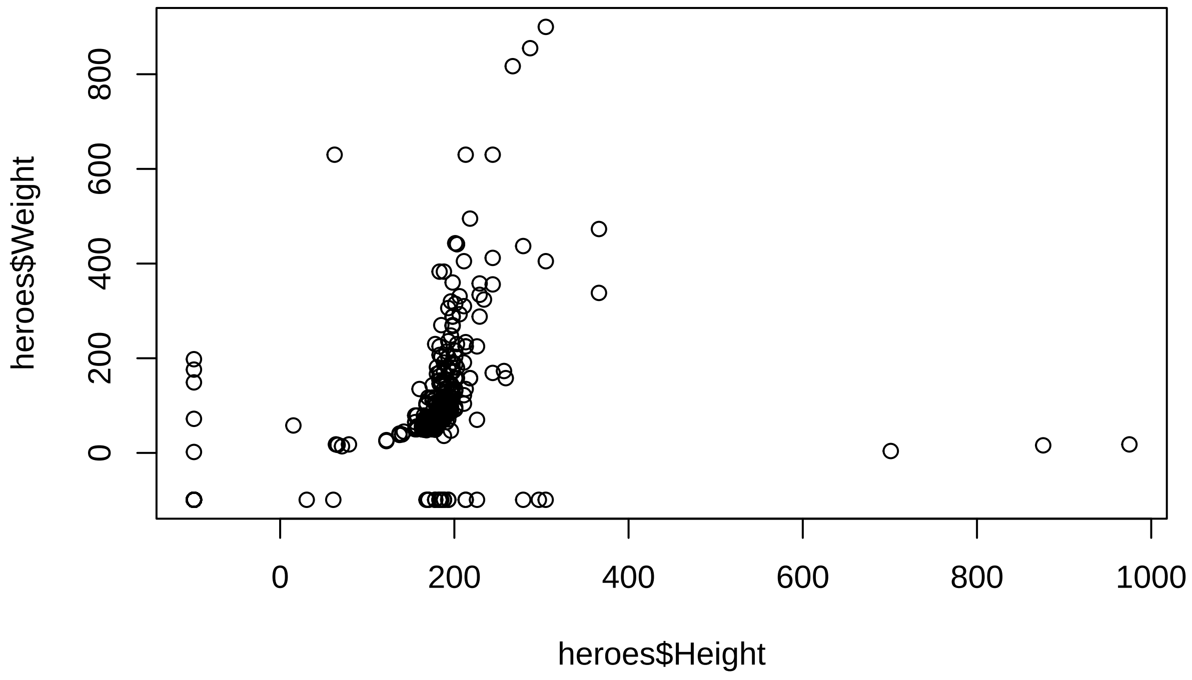
```r
plot(heroes$Publisher, las=2, cex.names=0.6, ylim= c(0,500)) # Increase y axis range
```



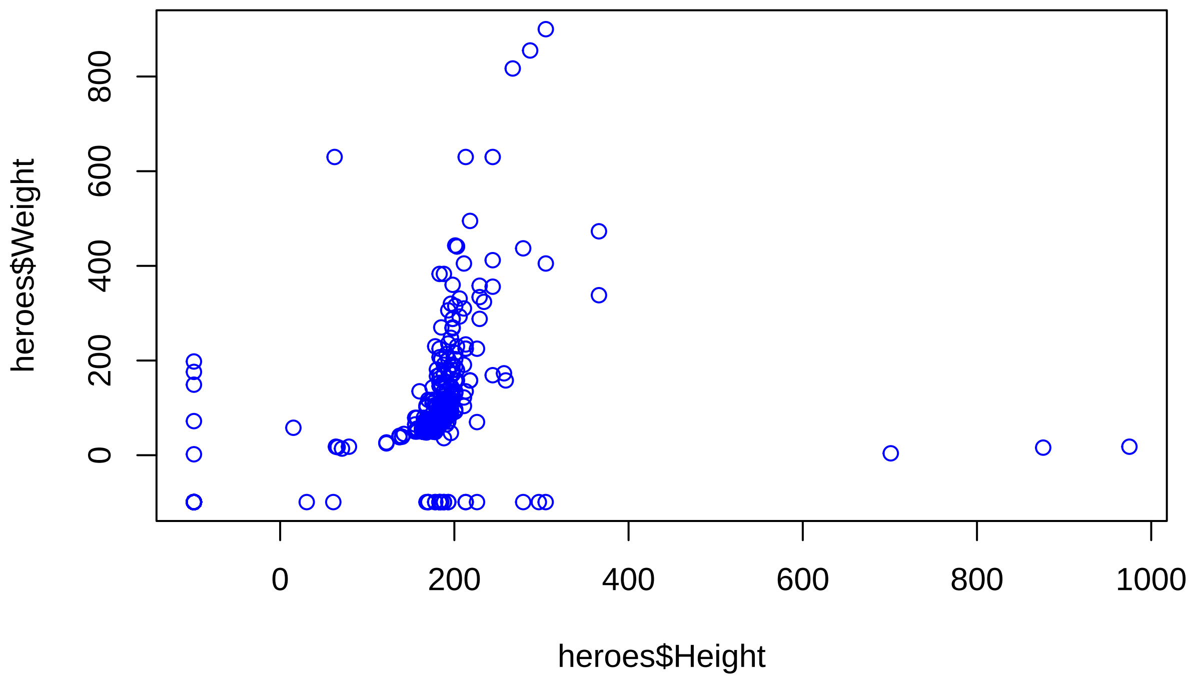
```r
plot(heroes$Publisher, las=2, cex.names=0.6, ylim= c(0,500), main = "Barplot of heroes by publisher") #
```



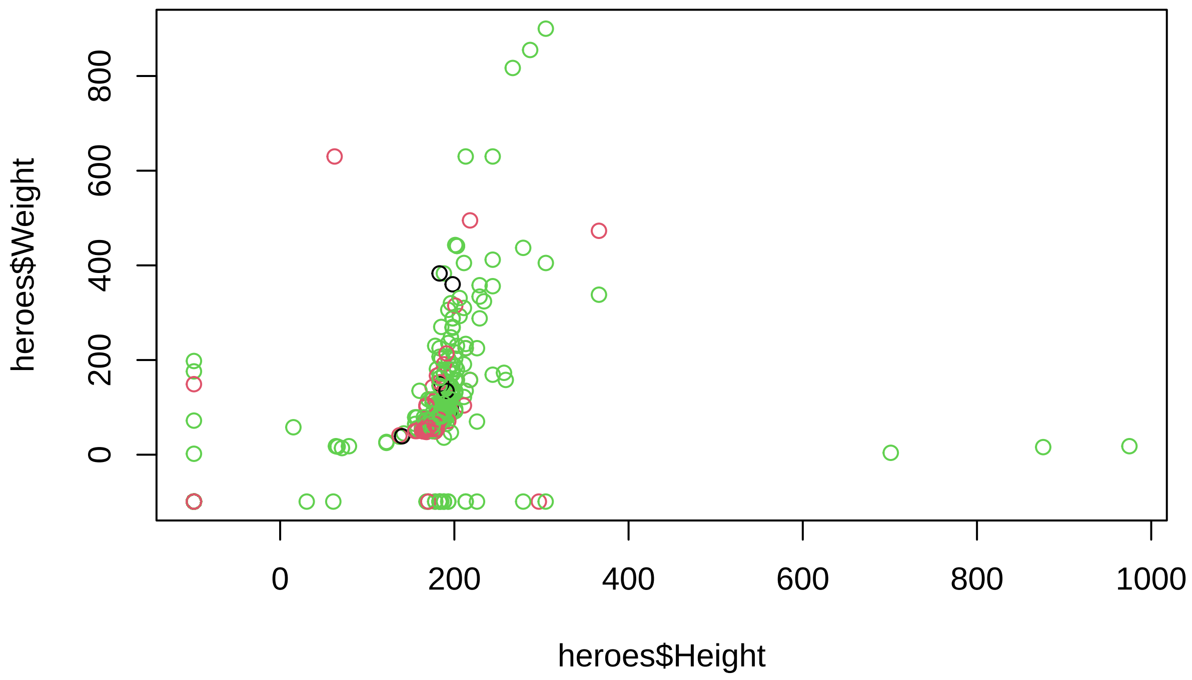
```r
# XY graphs / scatterplots / correlation
plot(heroes$Height, heroes$Weight) # Scatterplot of x and y values
```



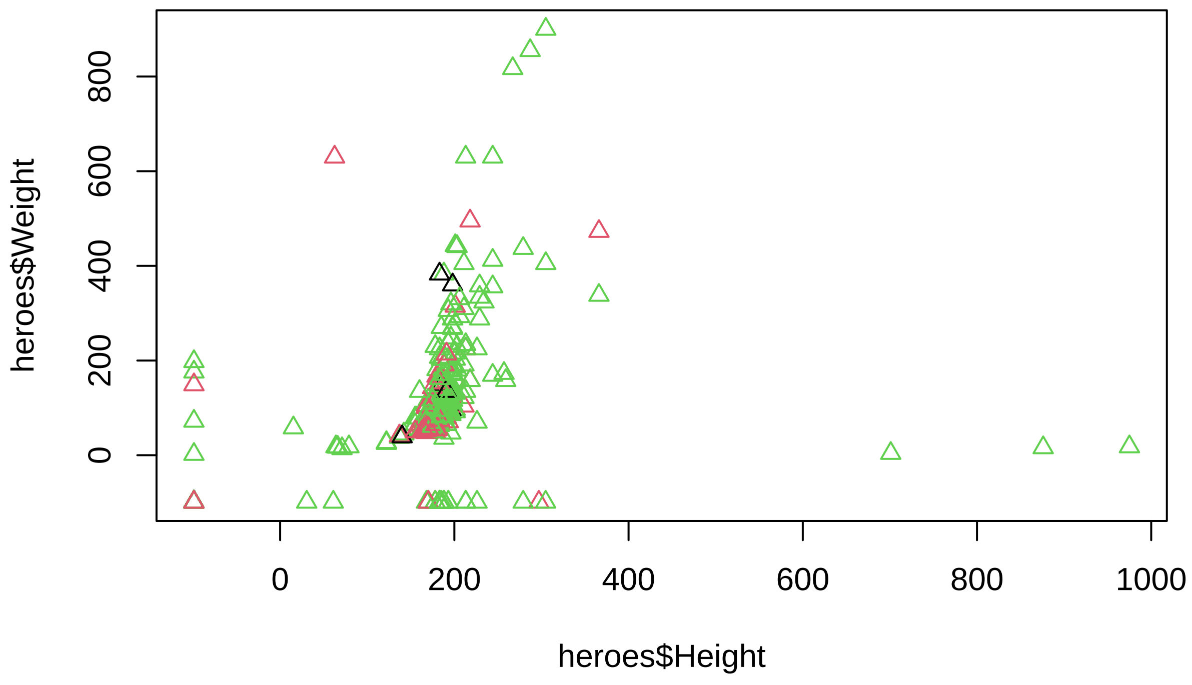
```r
plot(heroes$Height, heroes$Weight, col="blue") #colour for every gender
```



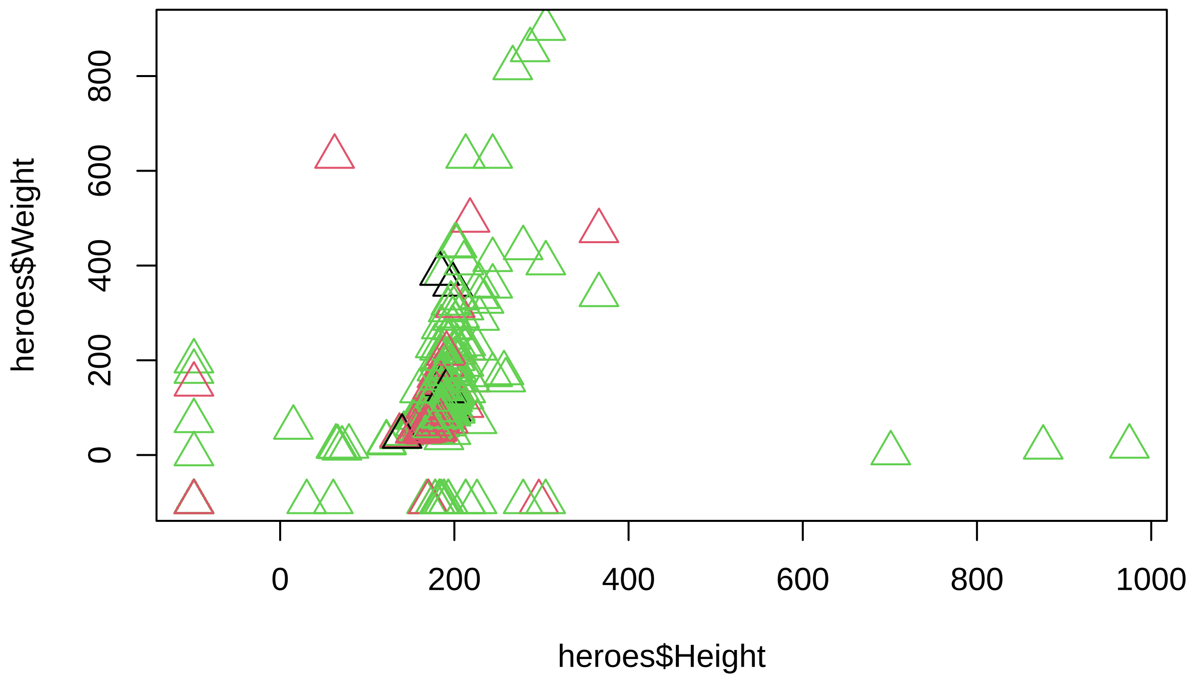
```
plot(heroes$Height, heroes$Weight, col=heroes$Gender) #colour for every gender
```



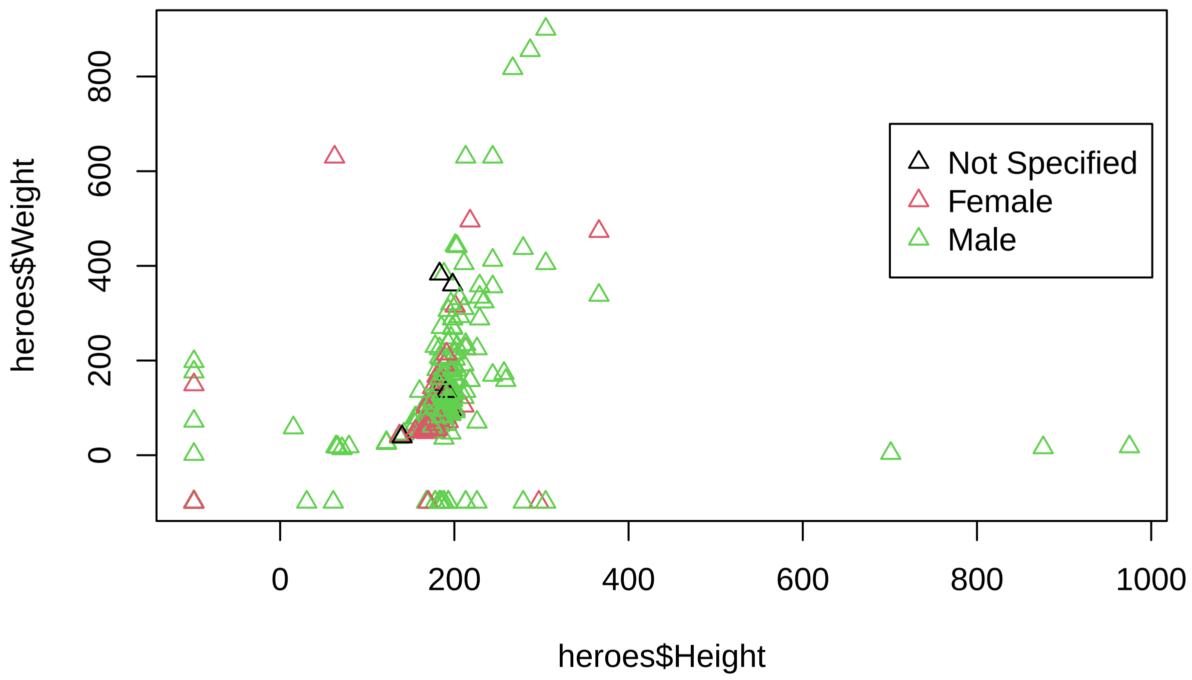
```
plot(heroes$Height, heroes$Weight, col=heroes$Gender, pch=24) #change symbol
```



```
plot(heroes$Height, heroes$Weight, col=heroes$Gender, pch=24, cex=2) #change size
```

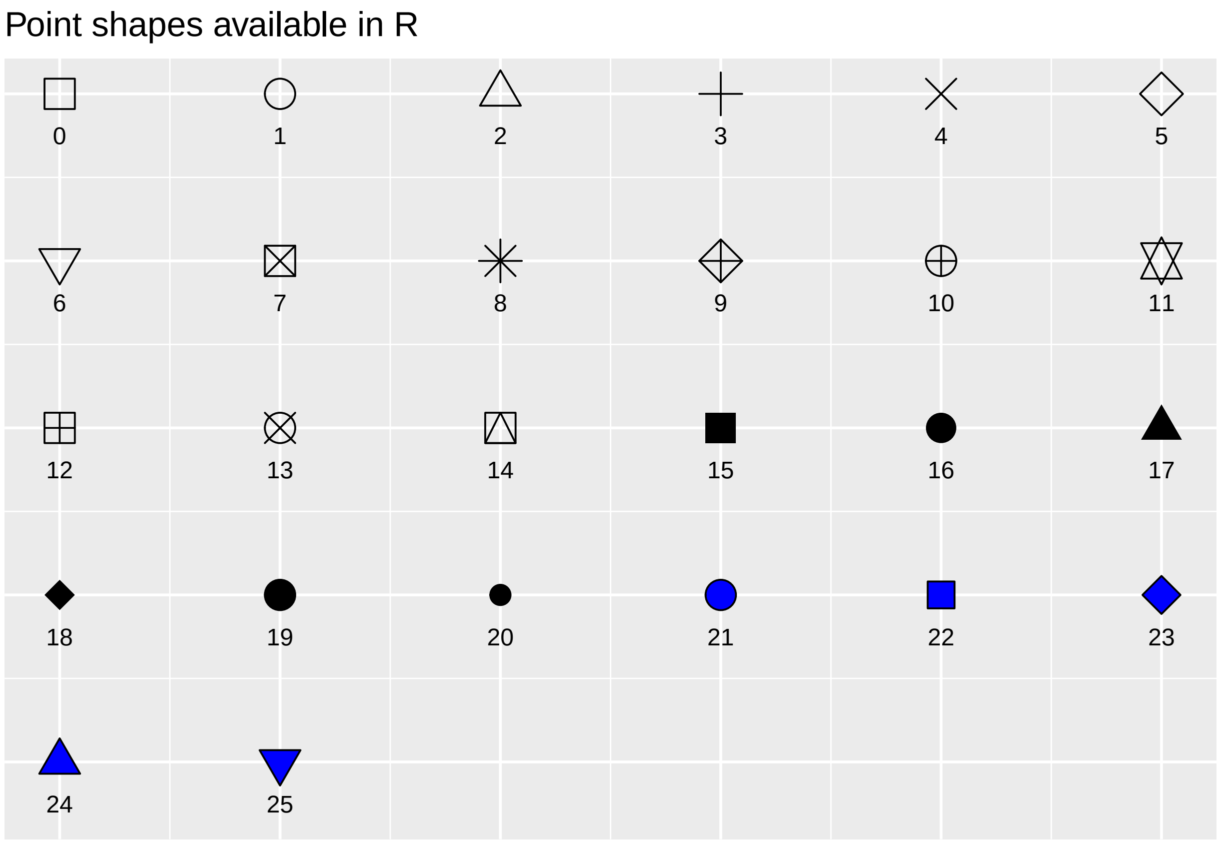


```
plot(heroes$Height, heroes$Weight, col=heroes$Gender, pch=24)
legend(x=700, y=700, legend = levels(heroes$Gender), col=c(1:3), pch= 24) #create legend
```

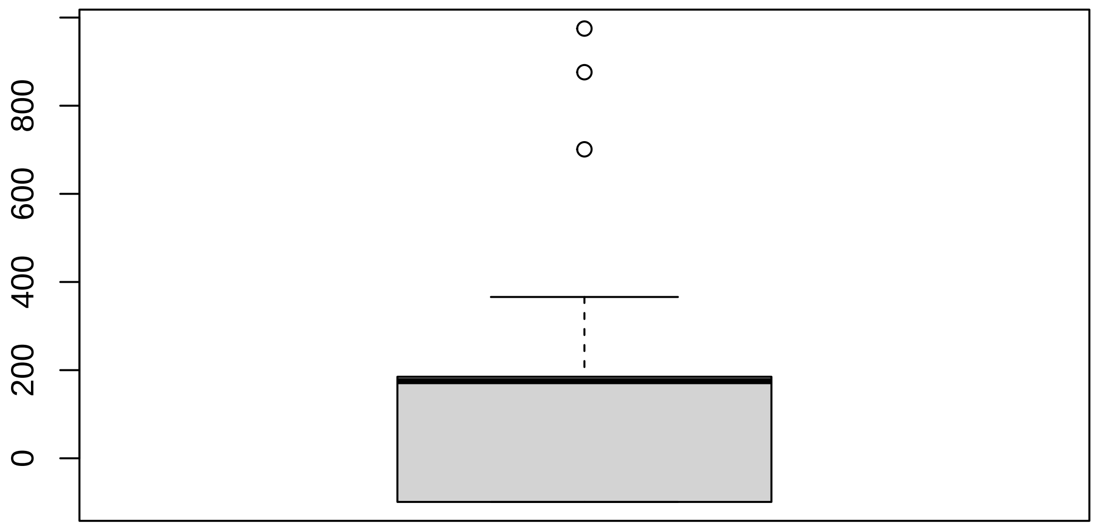


```r
# Remember we can check the different symbols we can use
ggpubr::show_point_shapes()
```

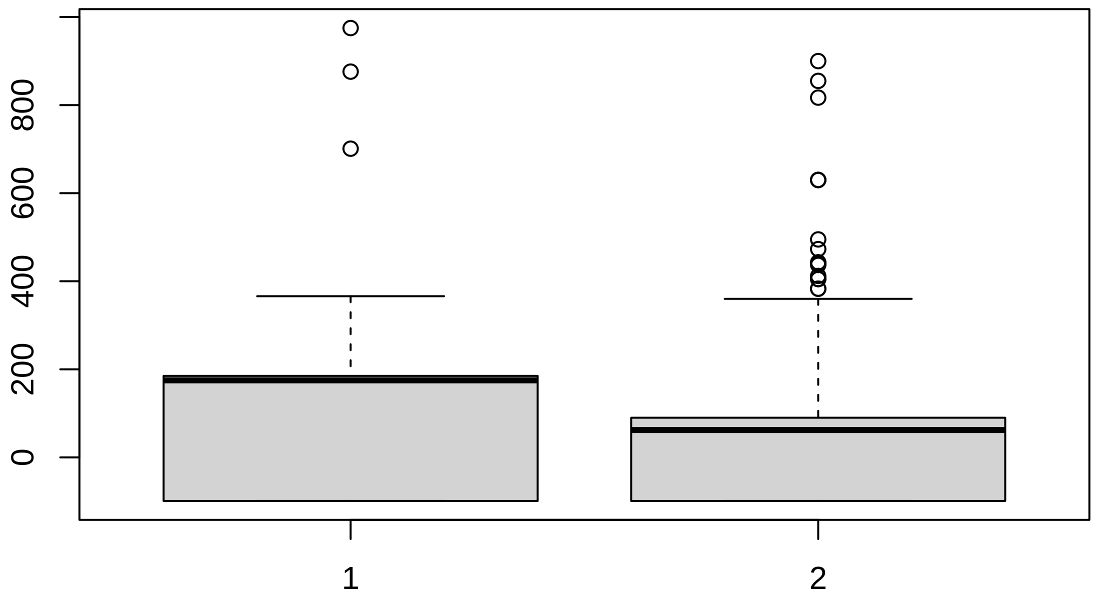
```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```



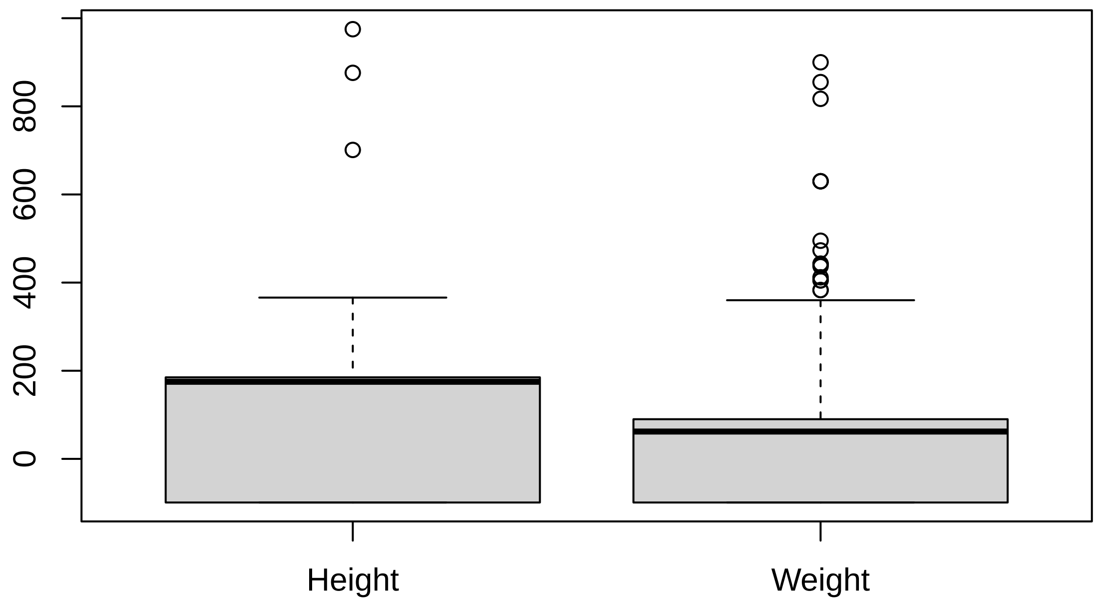
```r
# Boxplots
boxplot(heroes$Height) # We can see distribution of the variable Height
```



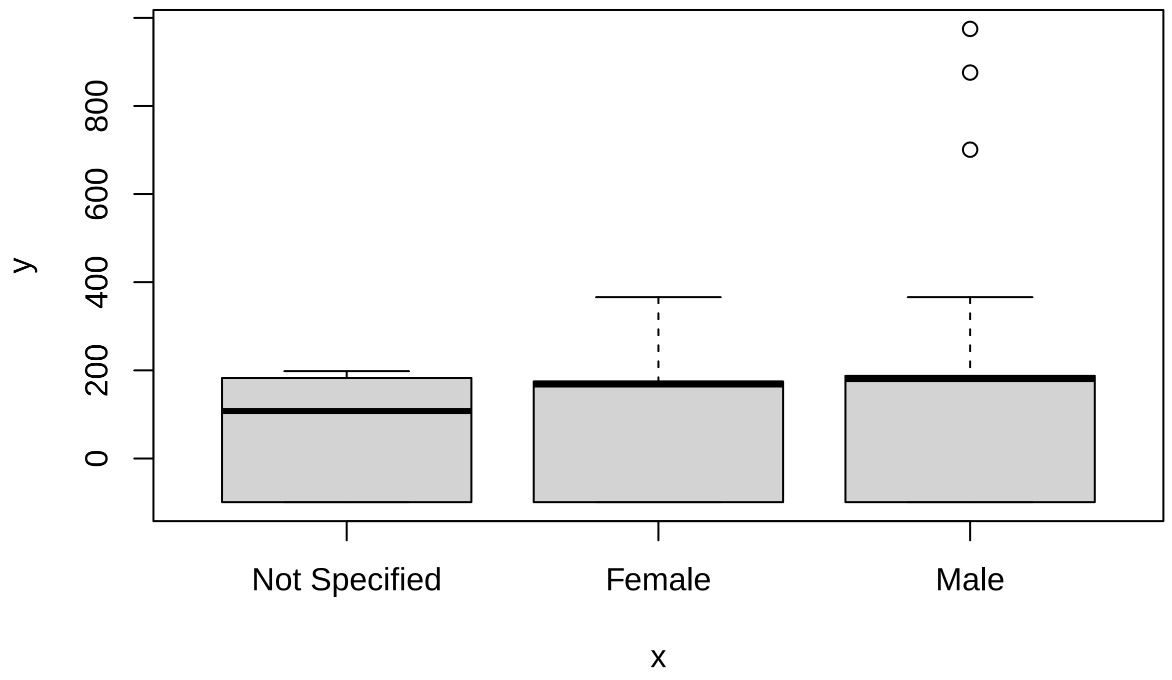
```r
boxplot(heroes$Height,heroes$Weight) # We can plot multiple boxplots (multiple variables)
```



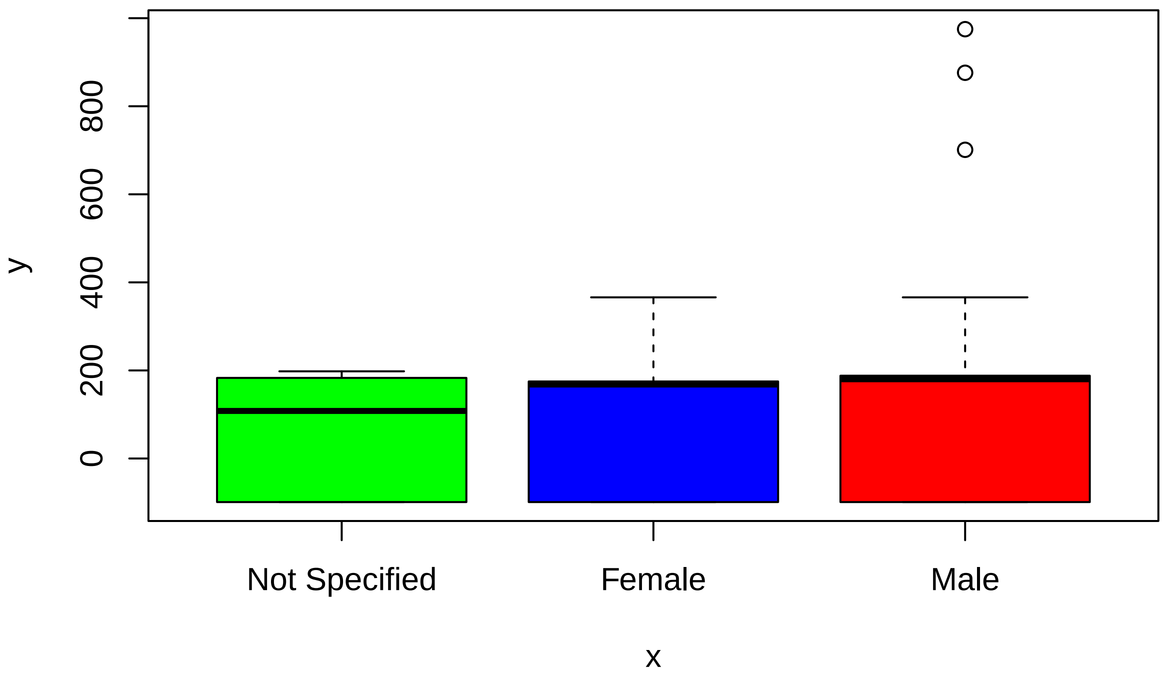
```r
boxplot(heroes$Height,heroes$Weight, names=c("Height", "Weight")) # Add name of variables plotted
```



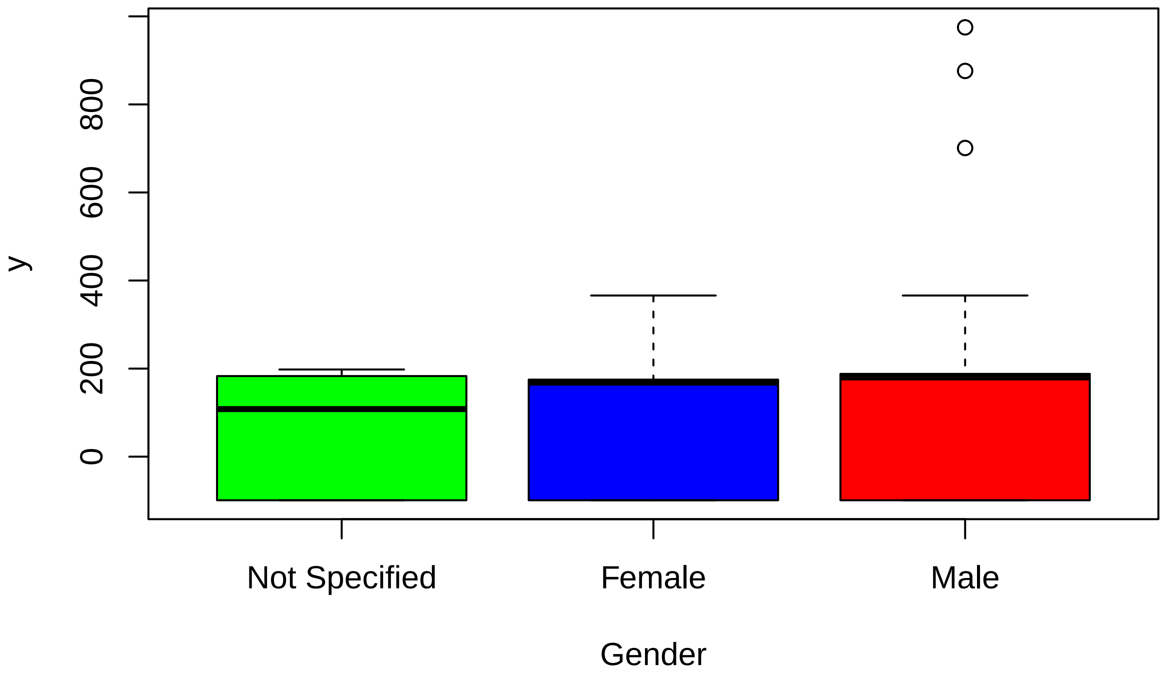
```r
# If we want to plot data based on a categorical data we can use plot()
plot(heroes$Gender, heroes$Height) # plot height data splited by Gender
```



```r
plot(heroes$Gender, heroes$Height, col=c("green","blue", "red")) #add colour
```



```
plot(heroes$Gender, heroes$Height, col=c("green","blue", "red"), xlab = "Gender" ) #change x axis title
```



```
# Save graphs
pdf("Barplot_heroes.pdf")
plot(heroes$Publisher, las=2, cex.names = 0.6, ylim= c(0,500), main = "Barplot of heroes by publisher")
dev.off()
```

```
## pdf
##   2
```

```
pdf("Barplot_heroes.pdf", width=15, height = 15)
plot(heroes$Publisher, las=2, cex.names = 0.6, ylim= c(0,500), main = "Barplot of heroes by publisher")
```

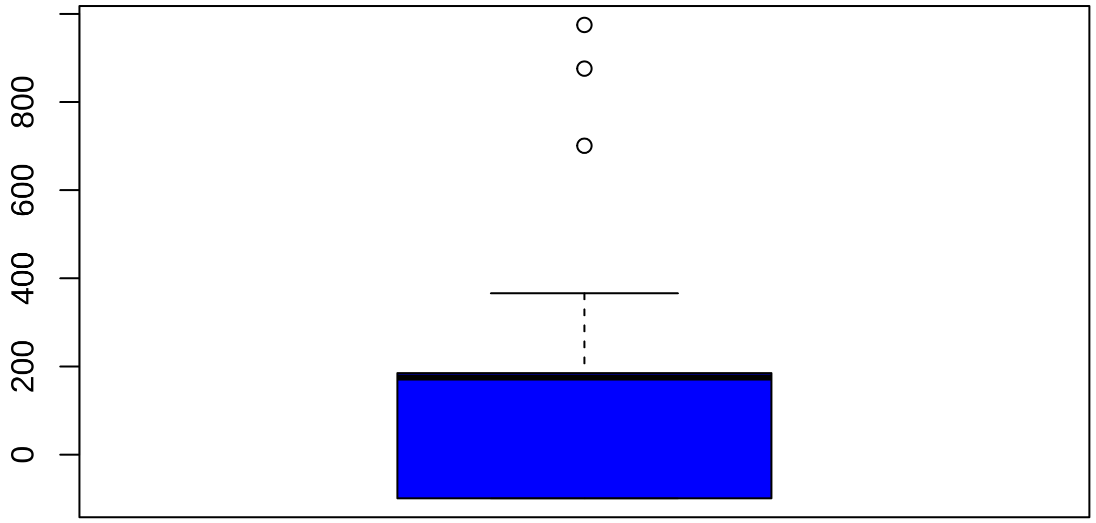
```
dev.off()
```

```
## pdf
##   2
```

```
png("Hero_scatterplot.png")
plot(heroes$Height, heroes$Weight, col=heroes$Gender, pch=24)
dev.off()
```

```
## pdf
##   2
```

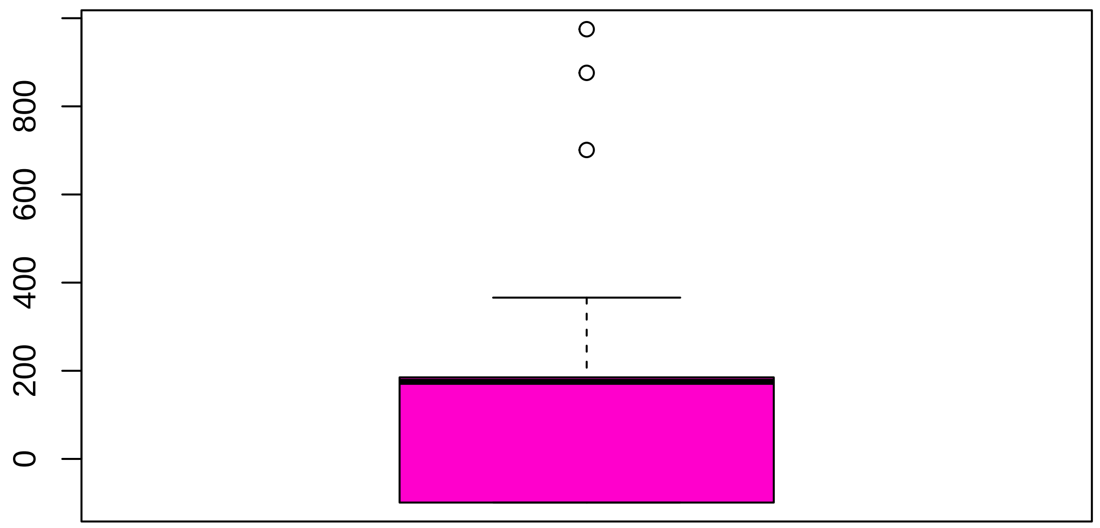
```
# More advanced information
# Colours can be defined in different ways

# specifing colour name (most common)
boxplot(heroes$Height, col="blue")
```



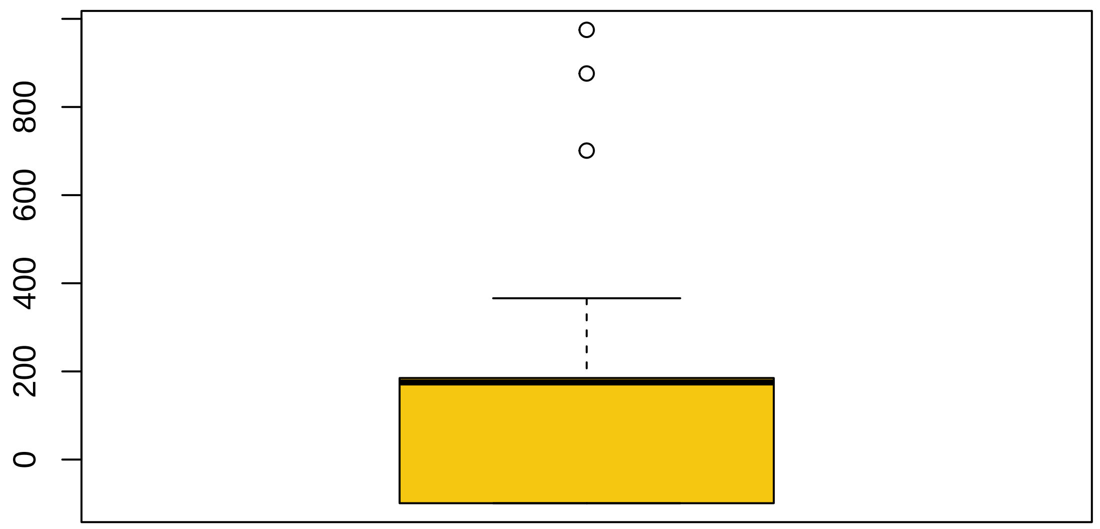
```
# you can find all the names for colours that R offers using
colors()
```

```
##   [1] "white"                "aliceblue"            "antiquewhite"
##   [4] "antiquewhite1"        "antiquewhite2"        "antiquewhite3"
##   [7] "antiquewhite4"        "aquamarine"           "aquamarine1"
##  [10] "aquamarine2"          "aquamarine3"          "aquamarine4"
##  [13] "azure"                "azure1"               "azure2"
##  [16] "azure3"               "azure4"               "beige"
##  [19] "bisque"               "bisque1"              "bisque2"
##  [22] "bisque3"              "bisque4"              "black"
##  [25] "blanchedalmond"       "blue"                 "blue1"
##  [28] "blue2"                "blue3"                "blue4"
##  [31] "blueviolet"           "brown"                "brown1"
##  [34] "brown2"               "brown3"               "brown4"
##  [37] "burlywood"            "burlywood1"           "burlywood2"
##  [40] "burlywood3"           "burlywood4"           "cadetblue"
##  [43] "cadetblue1"           "cadetblue2"           "cadetblue3"
##  [46] "cadetblue4"           "chartreuse"           "chartreuse1"
##  [49] "chartreuse2"          "chartreuse3"          "chartreuse4"
##  [52] "chocolate"            "chocolate1"           "chocolate2"
##  [55] "chocolate3"           "chocolate4"           "coral"
##  [58] "coral1"               "coral2"               "coral3"
##  [61] "coral4"               "cornflowerblue"       "cornsilk"
##  [64] "cornsilk1"            "cornsilk2"            "cornsilk3"
##  [67] "cornsilk4"            "cyan"                 "cyan1"
##  [70] "cyan2"                "cyan3"                "cyan4"
##  [73] "darkblue"             "darkcyan"             "darkgoldenrod"
##  [76] "darkgoldenrod1"       "darkgoldenrod2"       "darkgoldenrod3"
##  [79] "darkgoldenrod4"       "darkgray"             "darkgreen"
##  [82] "darkgrey"             "darkkhaki"            "darkmagenta"
##  [85] "darkolivegreen"       "darkolivegreen1"      "darkolivegreen2"
##  [88] "darkolivegreen3"      "darkolivegreen4"      "darkorange"
##  [91] "darkorange1"          "darkorange2"          "darkorange3"
##  [94] "darkorange4"          "darkorchid"           "darkorchid1"
##  [97] "darkorchid2"          "darkorchid3"          "darkorchid4"
## [100] "darkred"              "darksalmon"           "darkseagreen"
## [103] "darkseagreen1"        "darkseagreen2"        "darkseagreen3"
## [106] "darkseagreen4"        "darkslateblue"        "darkslategray"
## [109] "darkslategray1"       "darkslategray2"       "darkslategray3"
## [112] "darkslategray4"       "darkslategrey"        "darkturquoise"
## [115] "darkviolet"           "deeppink"             "deeppink1"
## [118] "deeppink2"            "deeppink3"            "deeppink4"
## [121] "deepskyblue"          "deepskyblue1"         "deepskyblue2"
## [124] "deepskyblue3"         "deepskyblue4"         "dimgray"
## [127] "dimgrey"              "dodgerblue"           "dodgerblue1"
## [130] "dodgerblue2"          "dodgerblue3"          "dodgerblue4"
## [133] "firebrick"            "firebrick1"           "firebrick2"
## [136] "firebrick3"           "firebrick4"           "floralwhite"
## [139] "forestgreen"          "gainsboro"            "ghostwhite"
## [142] "gold"                 "gold1"                "gold2"
## [145] "gold3"                "gold4"                "goldenrod"
## [148] "goldenrod1"           "goldenrod2"           "goldenrod3"
## [151] "goldenrod4"           "gray"                 "gray0"
## [154] "gray1"                "gray2"                "gray3"
## [157] "gray4"                "gray5"                "gray6"
## [160] "gray7"                "gray8"                "gray9"
## [163] "gray10"               "gray11"               "gray12"
## [166] "gray13"               "gray14"               "gray15"
## [169] "gray16"               "gray17"               "gray18"
## [172] "gray19"               "gray20"               "gray21"
## [175] "gray22"               "gray23"               "gray24"
## [178] "gray25"               "gray26"               "gray27"
## [181] "gray28"               "gray29"               "gray30"
## [184] "gray31"               "gray32"               "gray33"
## [187] "gray34"               "gray35"               "gray36"
## [190] "gray37"               "gray38"               "gray39"
## [193] "gray40"               "gray41"               "gray42"
## [196] "gray43"               "gray44"               "gray45"
## [199] "gray46"               "gray47"               "gray48"
## [202] "gray49"               "gray50"               "gray51"
## [205] "gray52"               "gray53"               "gray54"
## [208] "gray55"               "gray56"               "gray57"
## [211] "gray58"               "gray59"               "gray60"
## [214] "gray61"               "gray62"               "gray63"
## [217] "gray64"                "gray65"                "gray66"
## [220] "gray67"                "gray68"                "gray69"
## [223] "gray70"                "gray71"                "gray72"
## [226] "gray73"                "gray74"                "gray75"
## [229] "gray76"                "gray77"                "gray78"
## [232] "gray79"                "gray80"                "gray81"
## [235] "gray82"                "gray83"                "gray84"
## [238] "gray85"                "gray86"                "gray87"
## [241] "gray88"                "gray89"                "gray90"
## [244] "gray91"                "gray92"                "gray93"
## [247] "gray94"                "gray95"                "gray96"
## [250] "gray97"                "gray98"                "gray99"
## [253] "gray100"               "green"                 "green1"
## [256] "green2"                "green3"                "green4"
## [259] "greenyellow"           "grey"                  "grey0"
## [262] "grey1"                 "grey2"                 "grey3"
## [265] "grey4"                 "grey5"                 "grey6"
## [268] "grey7"                 "grey8"                 "grey9"
## [271] "grey10"                "grey11"                "grey12"
## [274] "grey13"                "grey14"                "grey15"
## [277] "grey16"                "grey17"                "grey18"
## [280] "grey19"                "grey20"                "grey21"
## [283] "grey22"                "grey23"                "grey24"
## [286] "grey25"                "grey26"                "grey27"
## [289] "grey28"                "grey29"                "grey30"
## [292] "grey31"                "grey32"                "grey33"
## [295] "grey34"                "grey35"                "grey36"
## [298] "grey37"                "grey38"                "grey39"
## [301] "grey40"                "grey41"                "grey42"
## [304] "grey43"                "grey44"                "grey45"
## [307] "grey46"                "grey47"                "grey48"
## [310] "grey49"                "grey50"                "grey51"
## [313] "grey52"                "grey53"                "grey54"
## [316] "grey55"                "grey56"                "grey57"
## [319] "grey58"                "grey59"                "grey60"
## [322] "grey61"                "grey62"                "grey63"
## [325] "grey64"                "grey65"                "grey66"
## [328] "grey67"                "grey68"                "grey69"
## [331] "grey70"                "grey71"                "grey72"
## [334] "grey73"                "grey74"                "grey75"
## [337] "grey76"                "grey77"                "grey78"
## [340] "grey79"                "grey80"                "grey81"
## [343] "grey82"                "grey83"                "grey84"
## [346] "grey85"                "grey86"                "grey87"
## [349] "grey88"                "grey89"                "grey90"
## [352] "grey91"                "grey92"                "grey93"
## [355] "grey94"                "grey95"                "grey96"
## [358] "grey97"                "grey98"                "grey99"
## [361] "grey100"               "honeydew"              "honeydew1"
## [364] "honeydew2"             "honeydew3"             "honeydew4"
## [367] "hotpink"               "hotpink1"              "hotpink2"
## [370] "hotpink3"              "hotpink4"              "indianred"
## [373] "indianred1"            "indianred2"            "indianred3"
## [376] "indianred4"            "ivory"                 "ivory1"
## [379] "ivory2"               "ivory3"               "ivory4"
## [382] "khaki"                "khaki1"               "khaki2"
## [385] "khaki3"               "khaki4"               "lavender"
## [388] "lavenderblush"        "lavenderblush1"       "lavenderblush2"
## [391] "lavenderblush3"       "lavenderblush4"       "lawngreen"
## [394] "lemonchiffon"         "lemonchiffon1"        "lemonchiffon2"
## [397] "lemonchiffon3"        "lemonchiffon4"        "lightblue"
## [400] "lightblue1"           "lightblue2"           "lightblue3"
## [403] "lightblue4"           "lightcoral"           "lightcyan"
## [406] "lightcyan1"           "lightcyan2"           "lightcyan3"
## [409] "lightcyan4"           "lightgoldenrod"       "lightgoldenrod1"
## [412] "lightgoldenrod2"      "lightgoldenrod3"      "lightgoldenrod4"
## [415] "lightgoldenrodyellow" "lightgray"            "lightgreen"
## [418] "lightgrey"            "lightpink"            "lightpink1"
## [421] "lightpink2"           "lightpink3"           "lightpink4"
## [424] "lightsalmon"          "lightsalmon1"         "lightsalmon2"
## [427] "lightsalmon3"         "lightsalmon4"         "lightseagreen"
## [430] "lightskyblue"         "lightskyblue1"        "lightskyblue2"
## [433] "lightskyblue3"        "lightskyblue4"        "lightslateblue"
## [436] "lightslategray"       "lightslategrey"       "lightsteelblue"
## [439] "lightsteelblue1"      "lightsteelblue2"      "lightsteelblue3"
## [442] "lightsteelblue4"      "lightyellow"          "lightyellow1"
## [445] "lightyellow2"         "lightyellow3"         "lightyellow4"
## [448] "limegreen"            "linen"                "magenta"
## [451] "magenta1"             "magenta2"             "magenta3"
## [454] "magenta4"             "maroon"               "maroon1"
## [457] "maroon2"              "maroon3"              "maroon4"
## [460] "mediumaquamarine"     "mediumblue"           "mediumorchid"
## [463] "mediumorchid1"        "mediumorchid2"        "mediumorchid3"
## [466] "mediumorchid4"        "mediumpurple"         "mediumpurple1"
## [469] "mediumpurple2"        "mediumpurple3"        "mediumpurple4"
## [472] "mediumseagreen"       "mediumslateblue"      "mediumspringgreen"
## [475] "mediumturquoise"      "mediumvioletred"      "midnightblue"
## [478] "mintcream"            "mistyrose"            "mistyrose1"
## [481] "mistyrose2"           "mistyrose3"           "mistyrose4"
## [484] "moccasin"             "navajowhite"          "navajowhite1"
## [487] "navajowhite2"         "navajowhite3"         "navajowhite4"
## [490] "navy"                 "navyblue"             "oldlace"
## [493] "olivedrab"            "olivedrab1"           "olivedrab2"
## [496] "olivedrab3"           "olivedrab4"           "orange"
## [499] "orange1"              "orange2"              "orange3"
## [502] "orange4"              "orangered"            "orangered1"
## [505] "orangered2"           "orangered3"           "orangered4"
## [508] "orchid"               "orchid1"              "orchid2"
## [511] "orchid3"              "orchid4"              "palegoldenrod"
## [514] "palegreen"            "palegreen1"           "palegreen2"
## [517] "palegreen3"           "palegreen4"           "paleturquoise"
## [520] "paleturquoise1"       "paleturquoise2"       "paleturquoise3"
## [523] "paleturquoise4"       "palevioletred"        "palevioletred1"
## [526] "palevioletred2"       "palevioletred3"       "palevioletred4"
## [529] "papayawhip"           "peachpuff"            "peachpuff1"
## [532] "peachpuff2"           "peachpuff3"           "peachpuff4"
## [535] "peru"                 "pink"                 "pink1"
## [538] "pink2"                "pink3"                "pink4"
## [541] "plum"                  "plum1"                 "plum2"
## [544] "plum3"                 "plum4"                 "powderblue"
## [547] "purple"                "purple1"               "purple2"
## [550] "purple3"               "purple4"               "red"
## [553] "red1"                  "red2"                  "red3"
## [556] "red4"                  "rosybrown"             "rosybrown1"
## [559] "rosybrown2"            "rosybrown3"            "rosybrown4"
## [562] "royalblue"             "royalblue1"            "royalblue2"
## [565] "royalblue3"            "royalblue4"            "saddlebrown"
## [568] "salmon"                "salmon1"               "salmon2"
## [571] "salmon3"               "salmon4"               "sandybrown"
## [574] "seagreen"              "seagreen1"             "seagreen2"
## [577] "seagreen3"             "seagreen4"             "seashell"
## [580] "seashell1"             "seashell2"             "seashell3"
## [583] "seashell4"             "sienna"                "sienna1"
## [586] "sienna2"               "sienna3"               "sienna4"
## [589] "skyblue"               "skyblue1"              "skyblue2"
## [592] "skyblue3"              "skyblue4"              "slateblue"
## [595] "slateblue1"            "slateblue2"            "slateblue3"
## [598] "slateblue4"            "slategray"             "slategray1"
## [601] "slategray2"            "slategray3"            "slategray4"
## [604] "slategrey"             "snow"                  "snow1"
## [607] "snow2"                 "snow3"                 "snow4"
## [610] "springgreen"           "springgreen1"          "springgreen2"
## [613] "springgreen3"          "springgreen4"          "steelblue"
## [616] "steelblue1"            "steelblue2"            "steelblue3"
## [619] "steelblue4"            "tan"                   "tan1"
## [622] "tan2"                  "tan3"                  "tan4"
## [625] "thistle"               "thistle1"              "thistle2"
## [628] "thistle3"              "thistle4"              "tomato"
## [631] "tomato1"               "tomato2"               "tomato3"
## [634] "tomato4"               "turquoise"             "turquoise1"
## [637] "turquoise2"            "turquoise3"            "turquoise4"
## [640] "violet"                "violetred"             "violetred1"
## [643] "violetred2"            "violetred3"            "violetred4"
## [646] "wheat"                 "wheat1"                "wheat2"
## [649] "wheat3"                "wheat4"                "whitesmoke"
## [652] "yellow"                "yellow1"               "yellow2"
## [655] "yellow3"               "yellow4"               "yellowgreen"
```

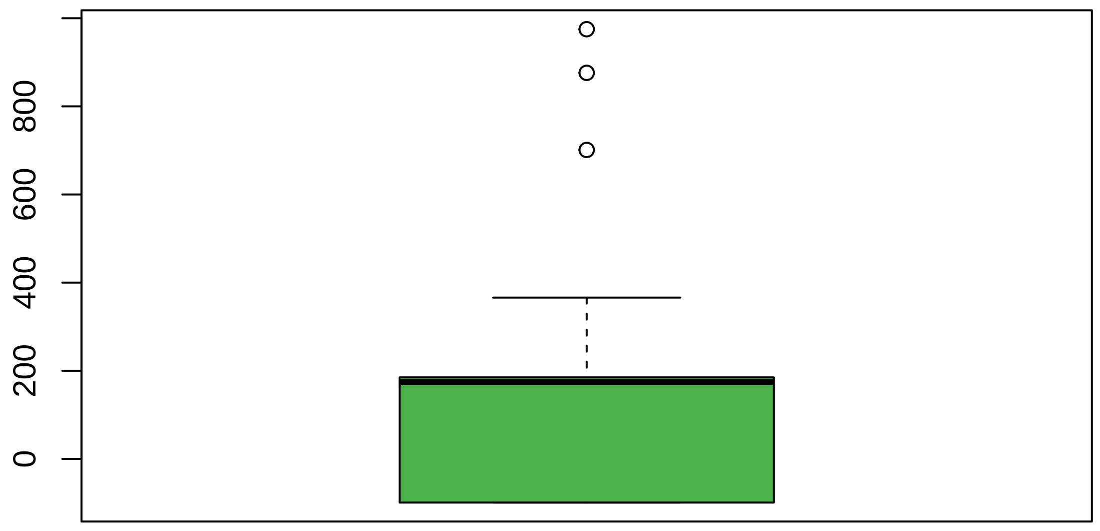
```r
# specifing hexadecimal color code
boxplot(heroes$Height, col="#FF00CC")
```



```r
# https://htmlcolorcodes.com will allow you to pick a colour and get the code

# using number
boxplot(heroes$Height, col=431)
```



```r
# using rgb() function to build a color (red, green, blue)
boxplot(heroes$Height, col=rgb(0.3, 0.7, 0.3))
```



# Exercises

```
# Plot a boxplot of the Weight by Alignment.
```

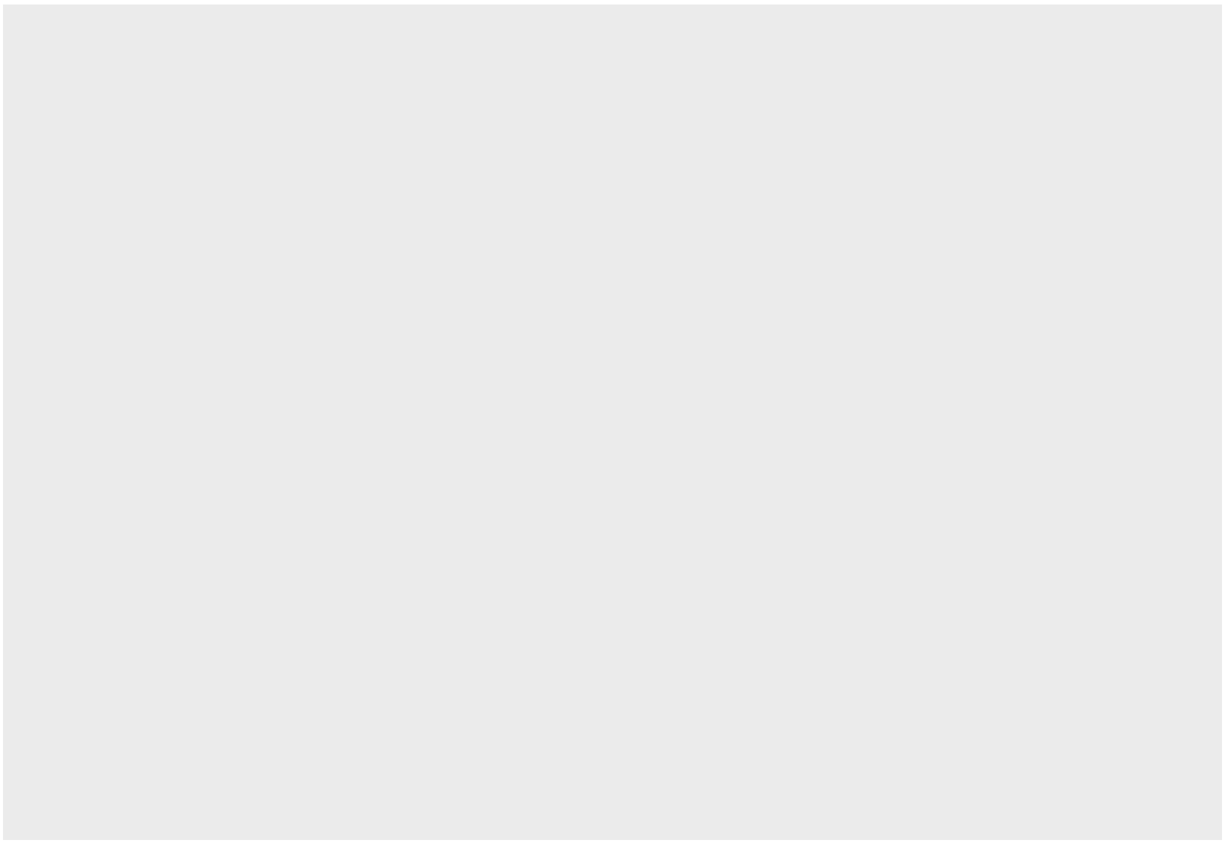
```
# Plot the frequency of every alignment in the dataset
```

```
# Colour the previous plot
```

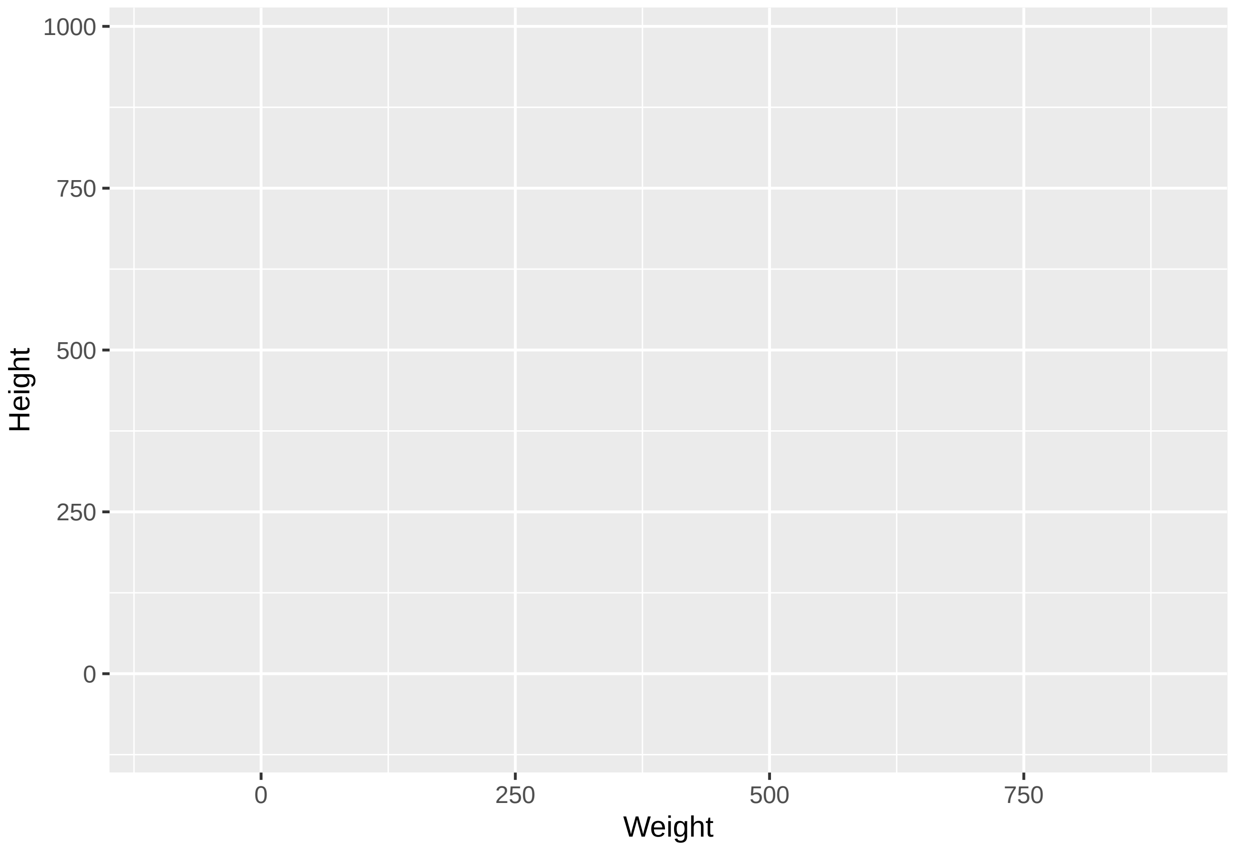
ggplot2

Plotting package that is part of tidyverse. We will work mostly on scatterplot

```
# Install/load ggplot2
#install.packages("ggplot2")
library(ggplot2)


# Let´s get rid of the not specified group for the sake of better visualisation
heroes = heroes[heroes$Gender != "Not Specified",]


# Plot the canvas
ggplot()
```

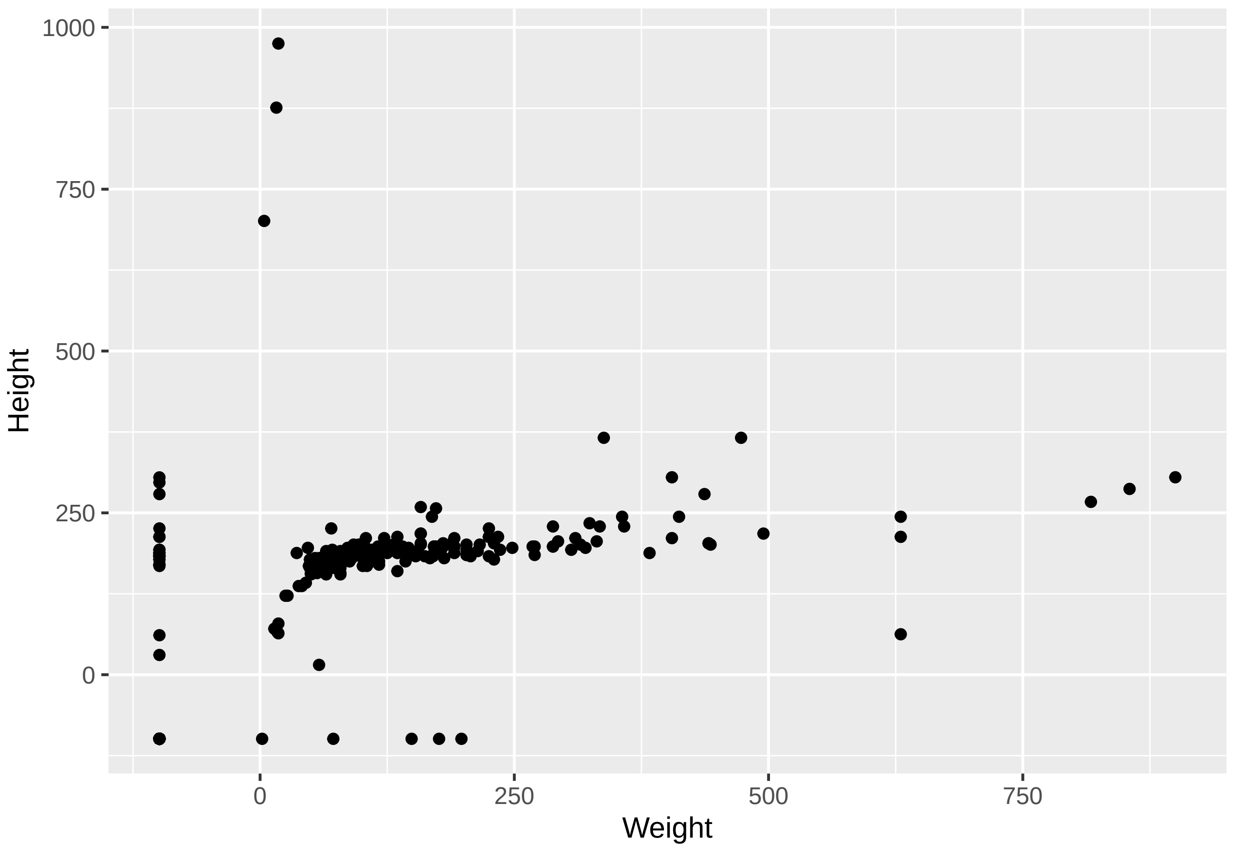


```r
# Add the data and mapping information
ggplot(data = heroes, mapping = aes(Weight,Height))
```



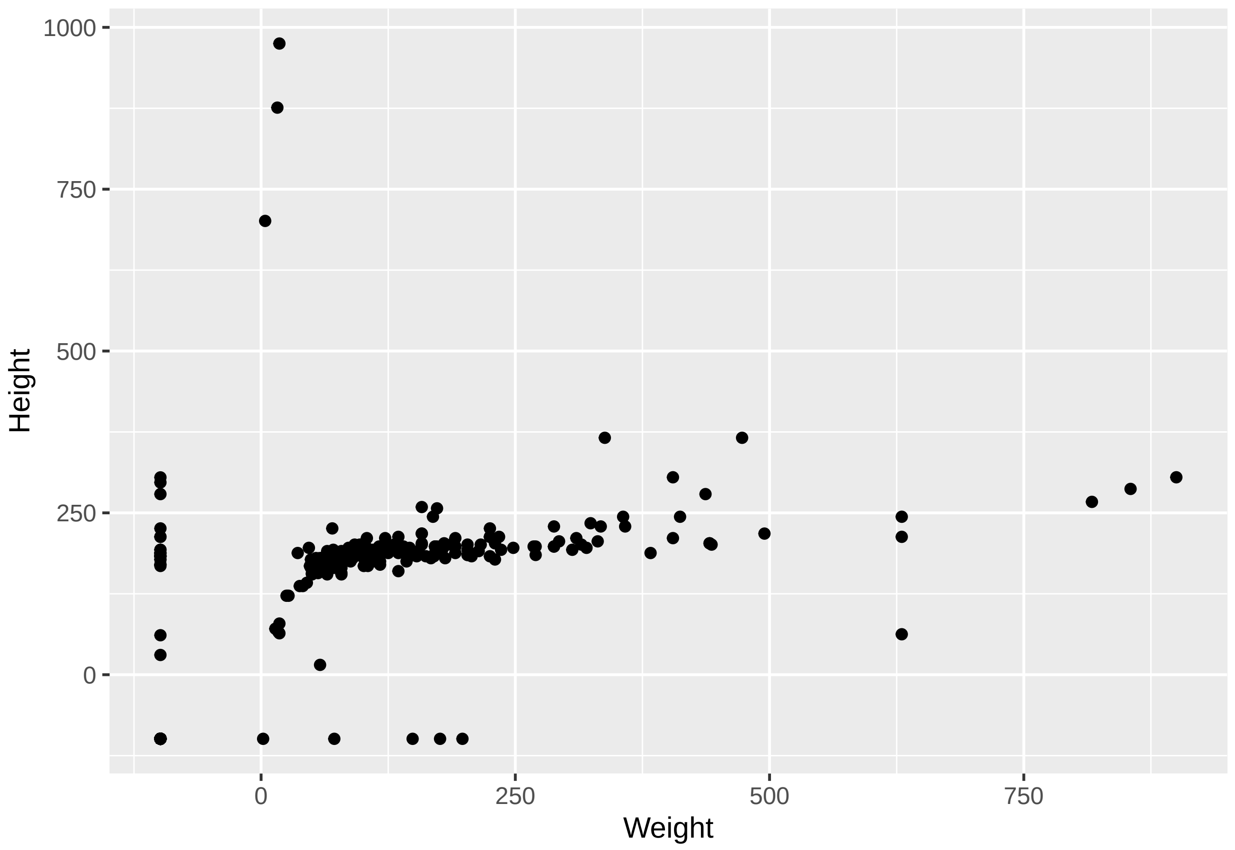
```r
# Add the geometry
ggplot(data = heroes, mapping = aes(Weight,Height))+
    geom_point()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



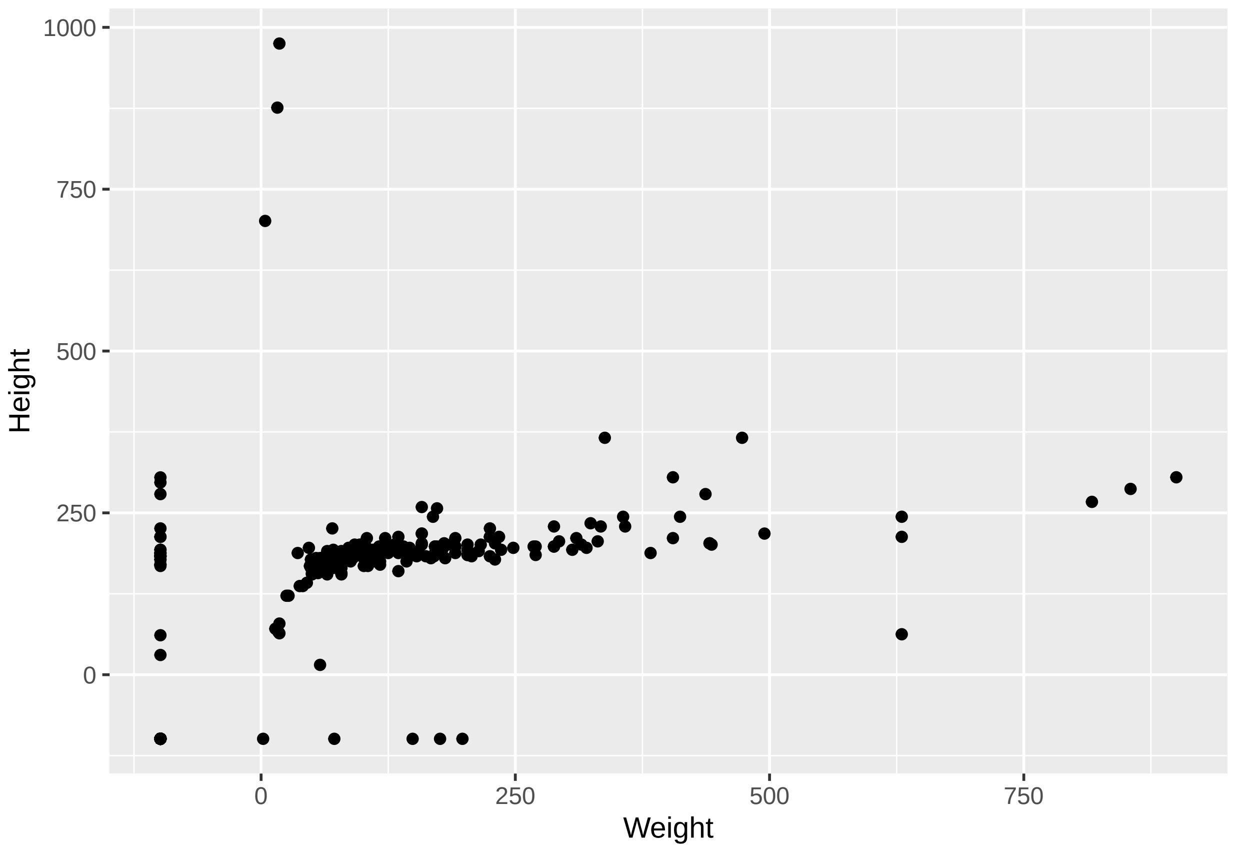
```r
# Data and mapping can be given as global or on the layer. Remember that will affect the inheritance an
ggplot(data = heroes)+
    geom_point(mapping = aes(Weight,Height))
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



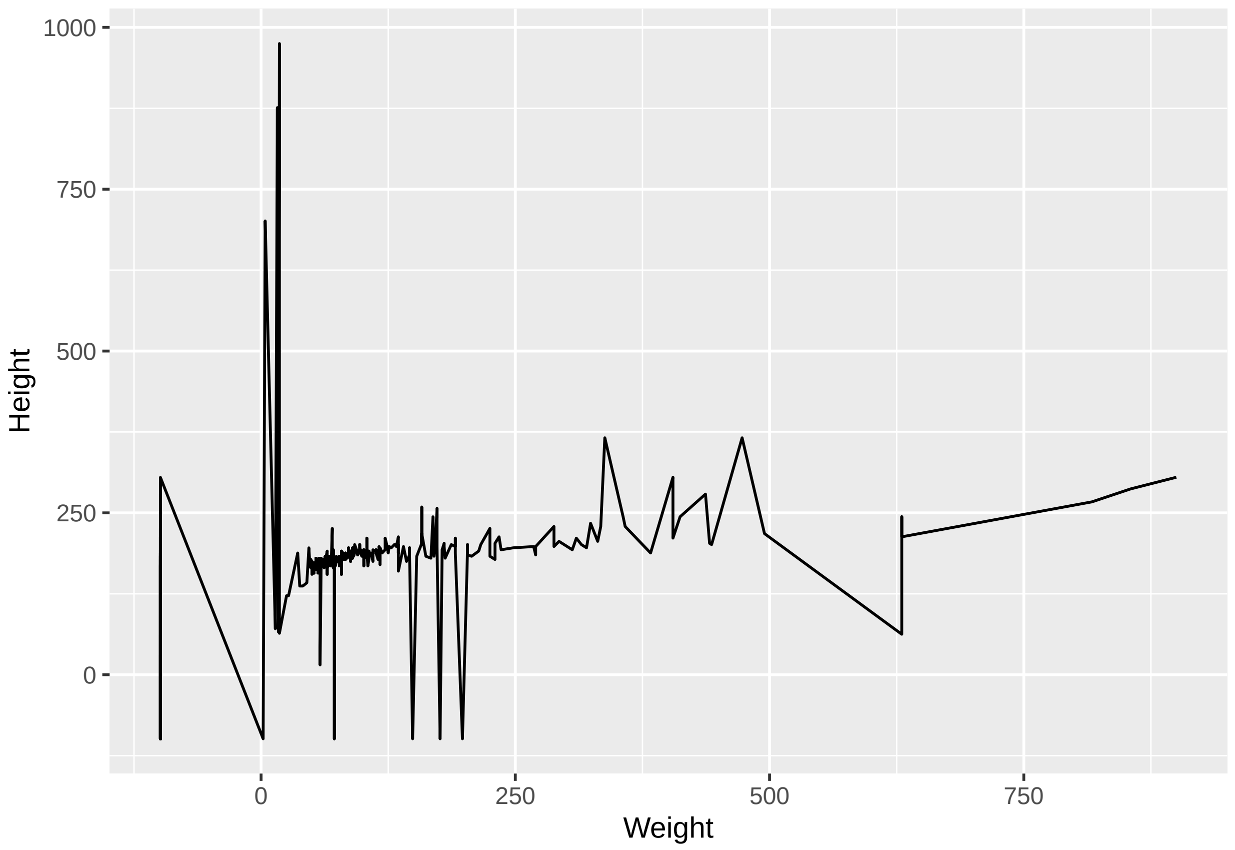
```r
ggplot()+
    geom_point(mapping = aes(Weight,Height), data= heroes)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



```r
# Use different type of gemoetry
ggplot(data = heroes, mapping = aes(Weight,Height)) +
    geom_line()
```

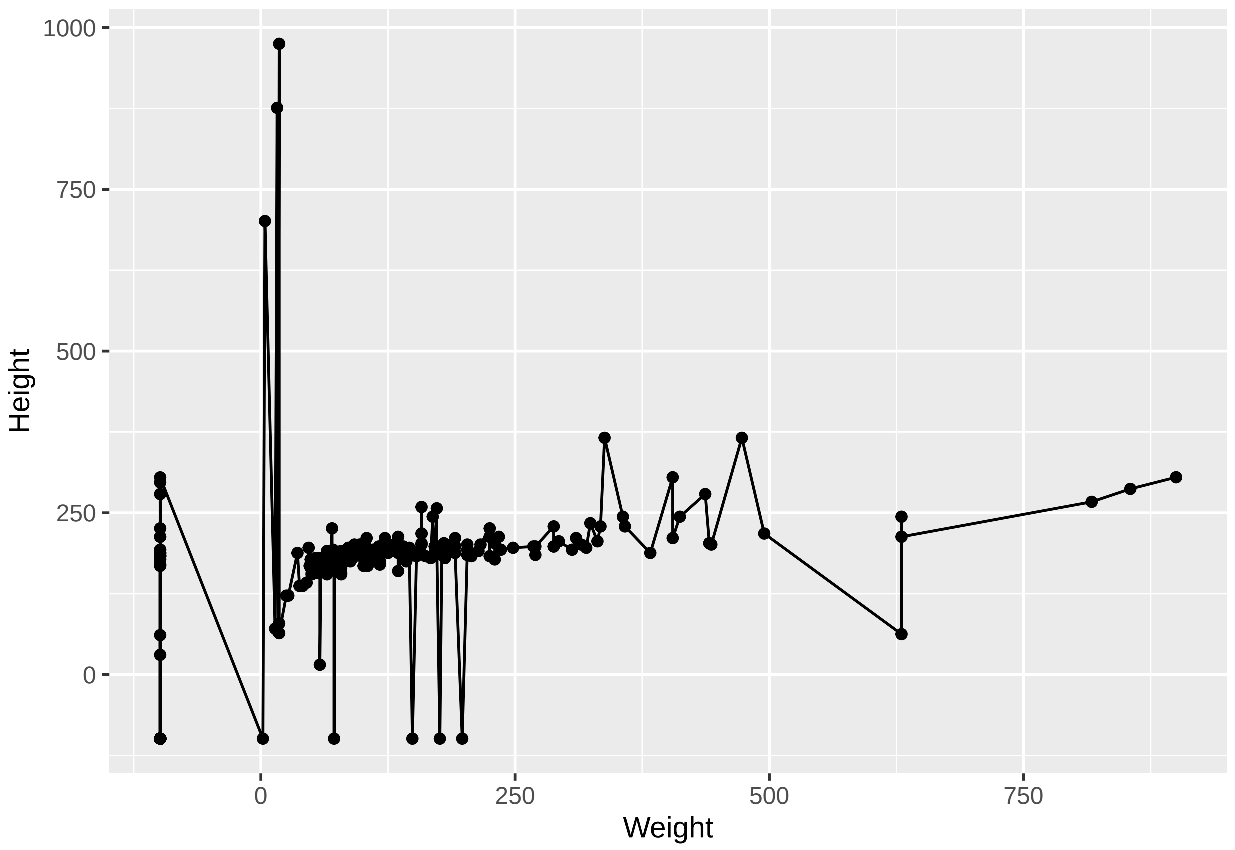
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



```
# We can add different geometries too
ggplot(data = heroes, mapping = aes(Weight,Height)) +
  geom_point() +
    geom_line()
```

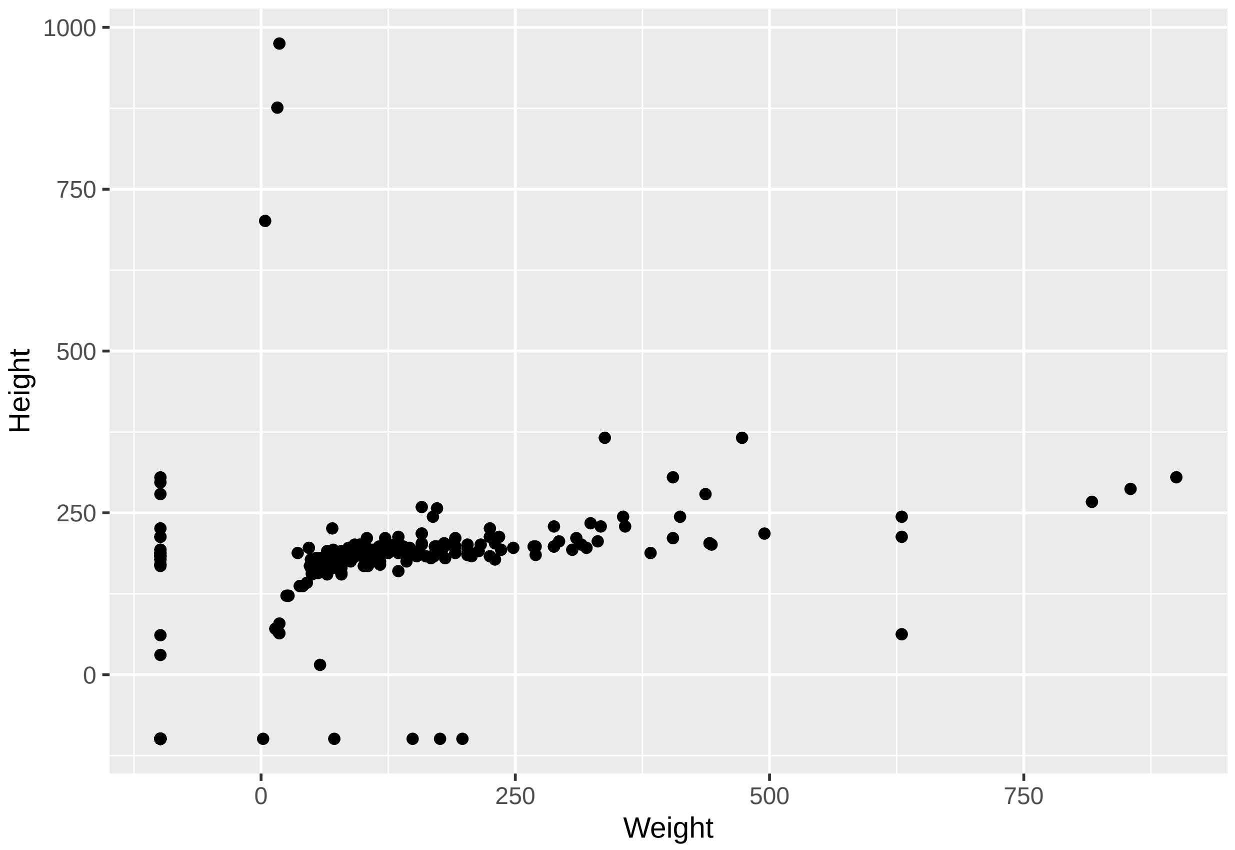
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).
```



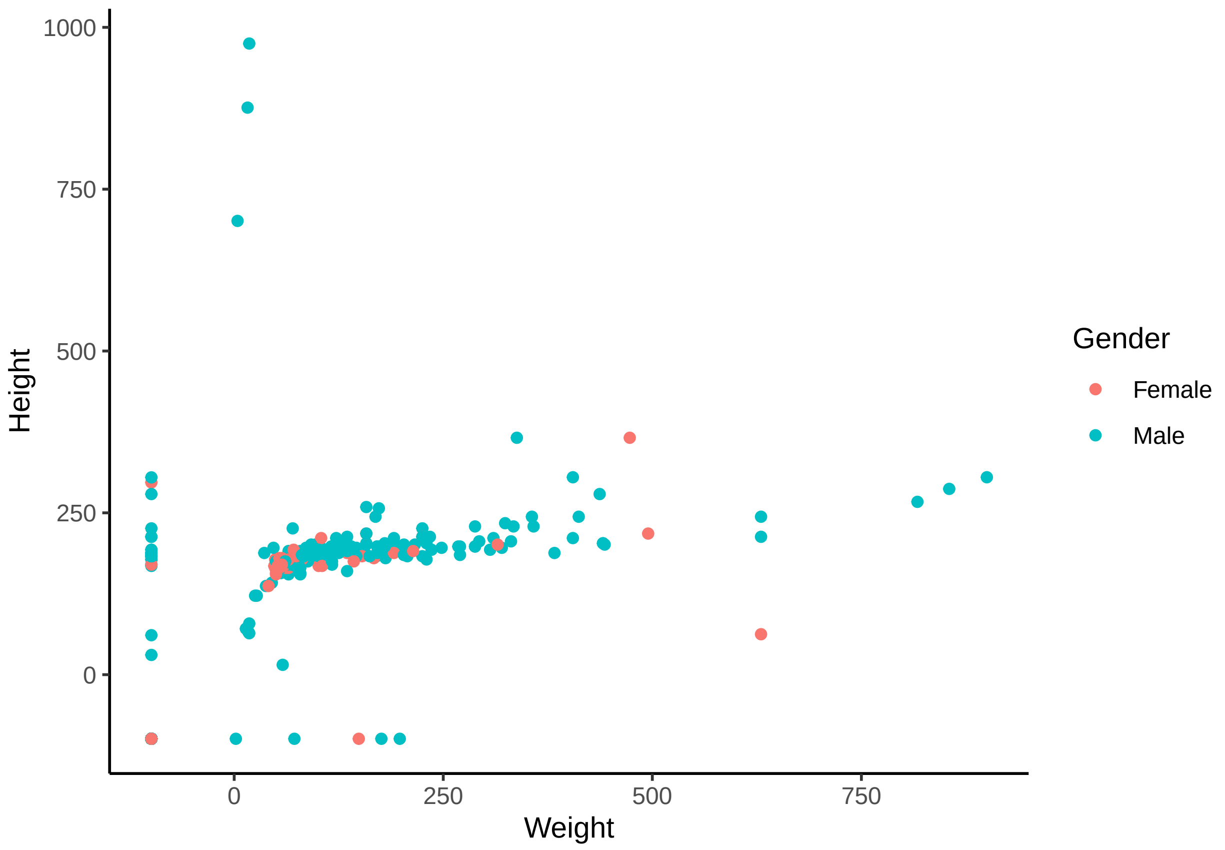
```r
# There is really no need to specify that the arguments are the data or mapping.
ggplot(heroes, aes(Weight,Height))+
    geom_point()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



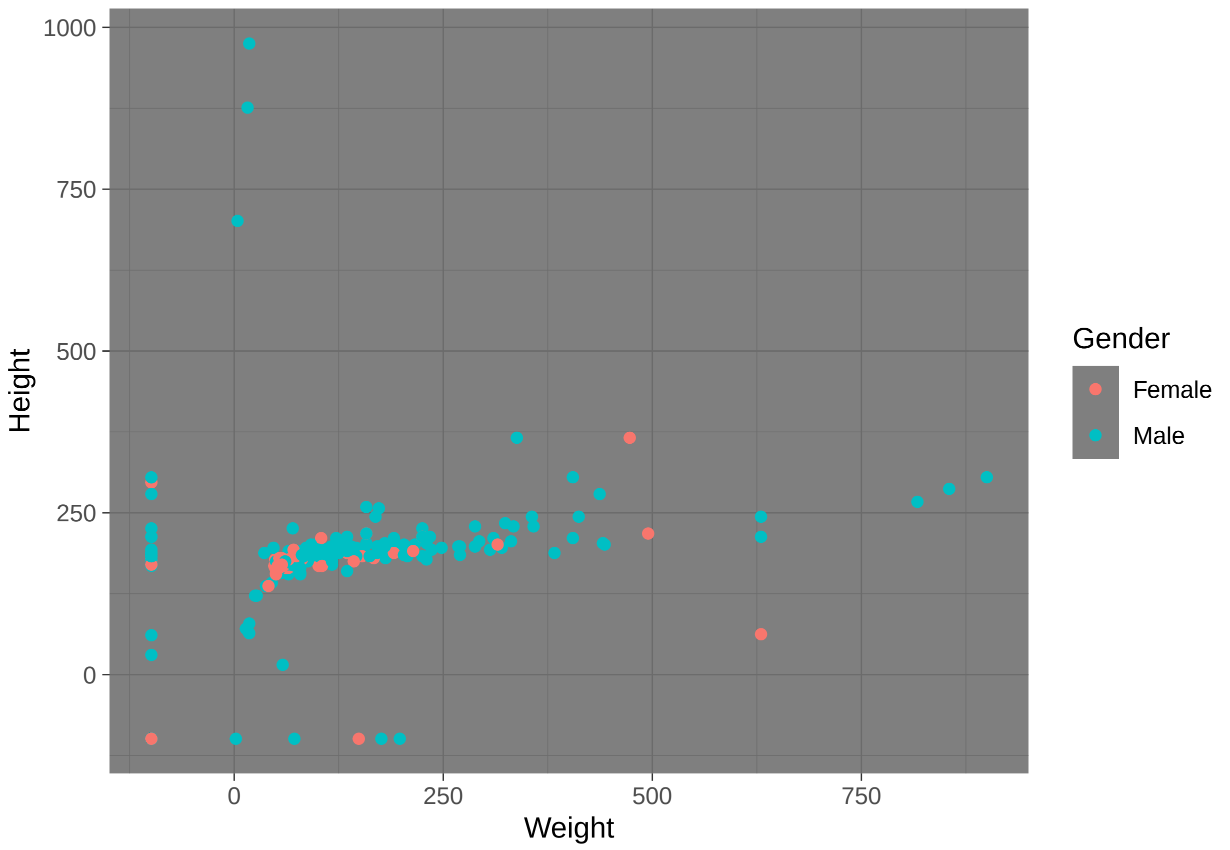
```r
# We can control the style with themes that are defined already
ggplot(data = heroes, mapping = aes(Weight,Height)) +
    geom_point(aes(colour = Gender)) +
    theme_classic() # classic style
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



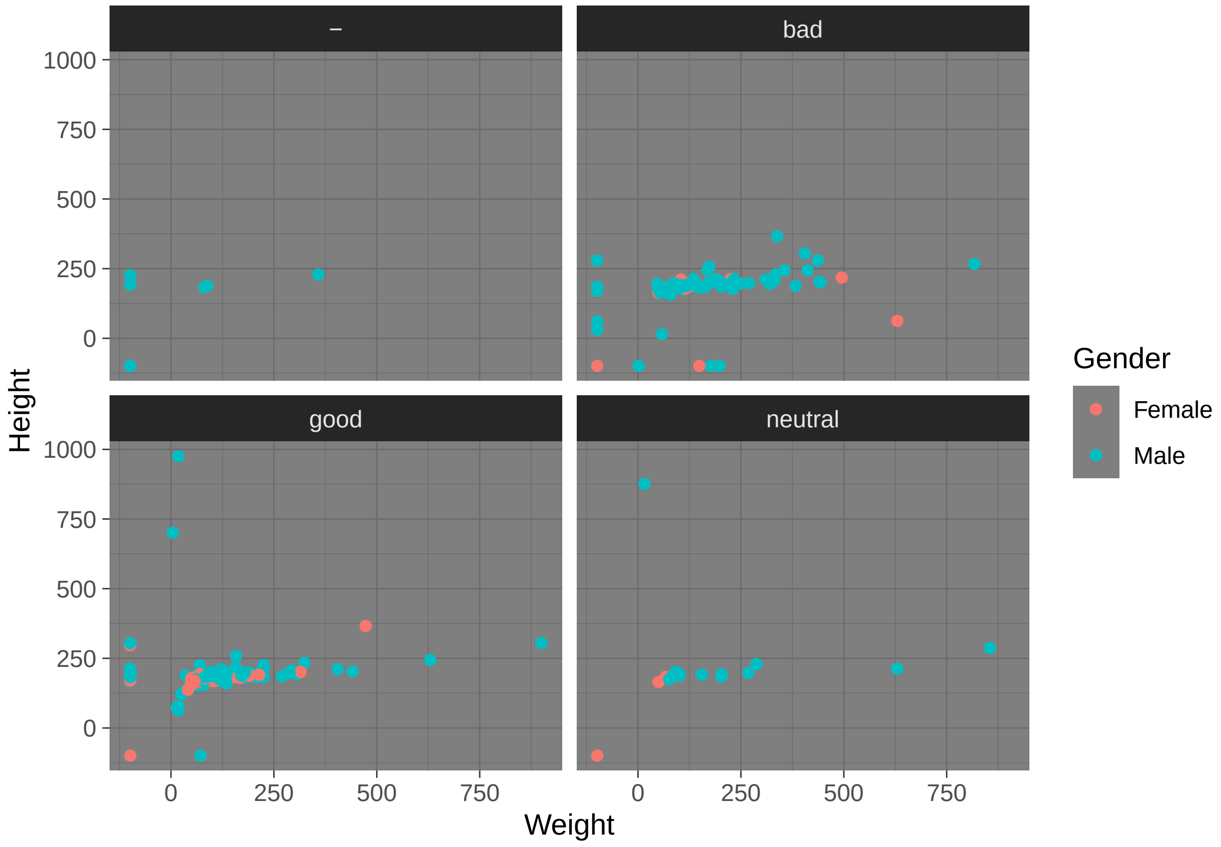
```r
ggplot(data = heroes, mapping = aes(Weight,Height))+
    geom_point(aes(colour = Gender))+
    theme_dark() # dark style
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



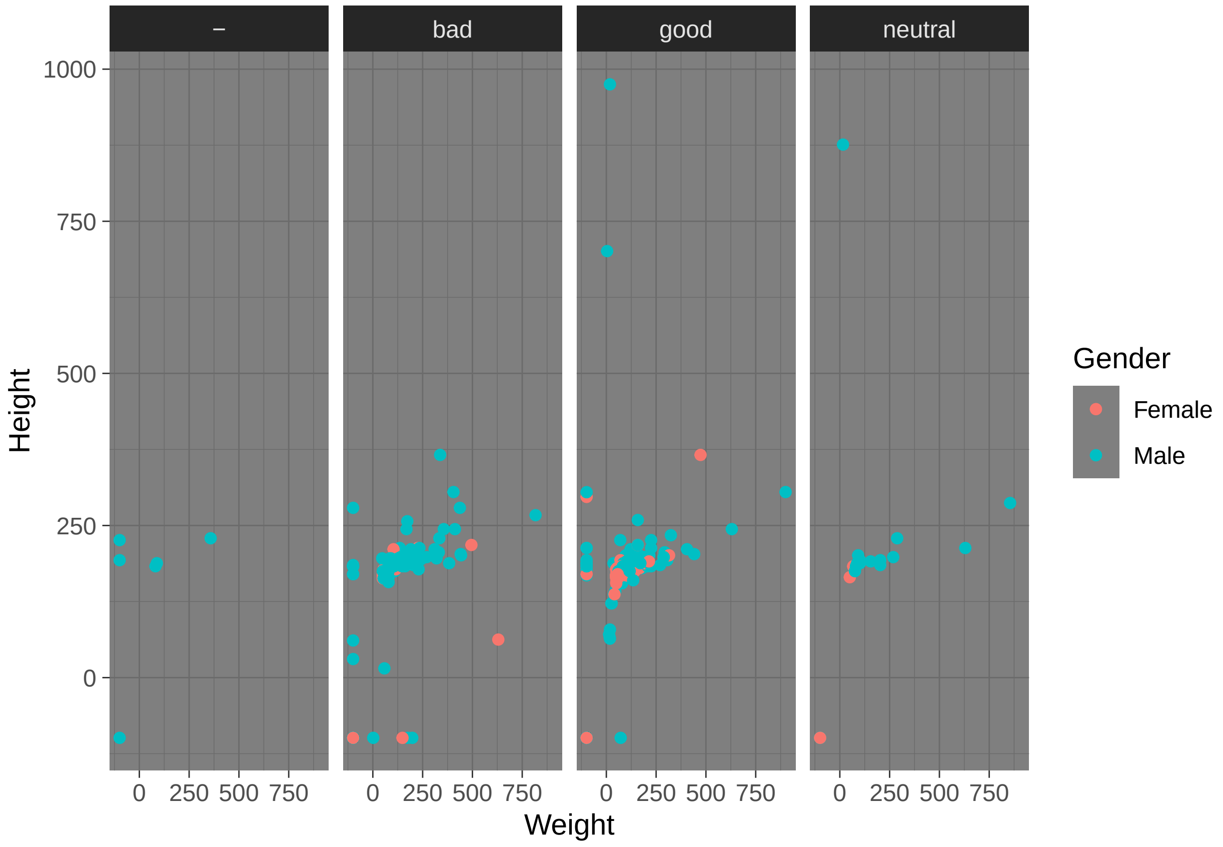
```r
# Split data into different graphs using facets
ggplot(data = heroes, mapping = aes(Weight,Height)) +
    geom_point(aes(colour = Gender)) +
    theme_dark() +
    facet_wrap(~Alignment)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



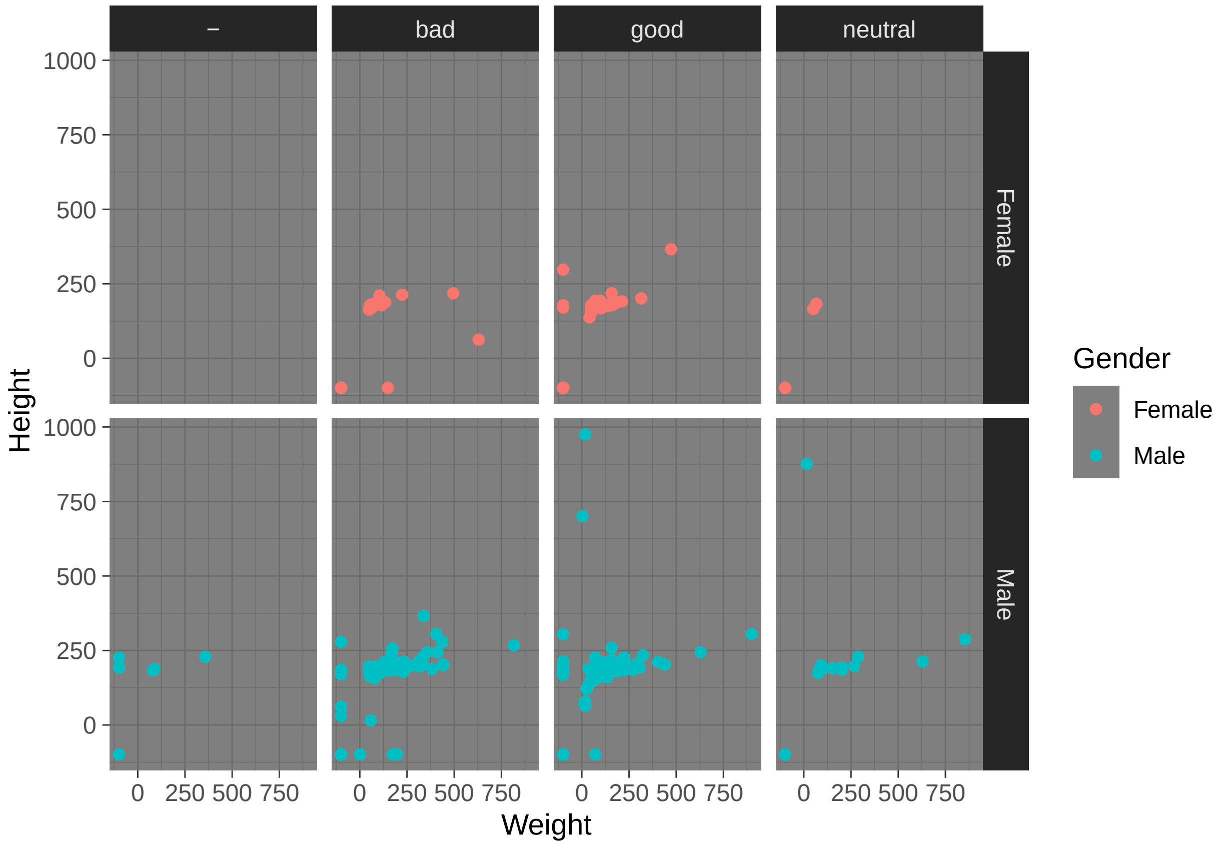
```r
ggplot(data = heroes, mapping = aes(Weight,Height)) +
    geom_point(aes(colour = Gender)) +
    theme_dark() +
    facet_grid(~Alignment)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



```r
ggplot(data = heroes, mapping = aes(Weight,Height))+
    geom_point(aes(colour = Gender))+
    theme_dark()+
    facet_grid(Gender~Alignment) # it can be split in two variable
```

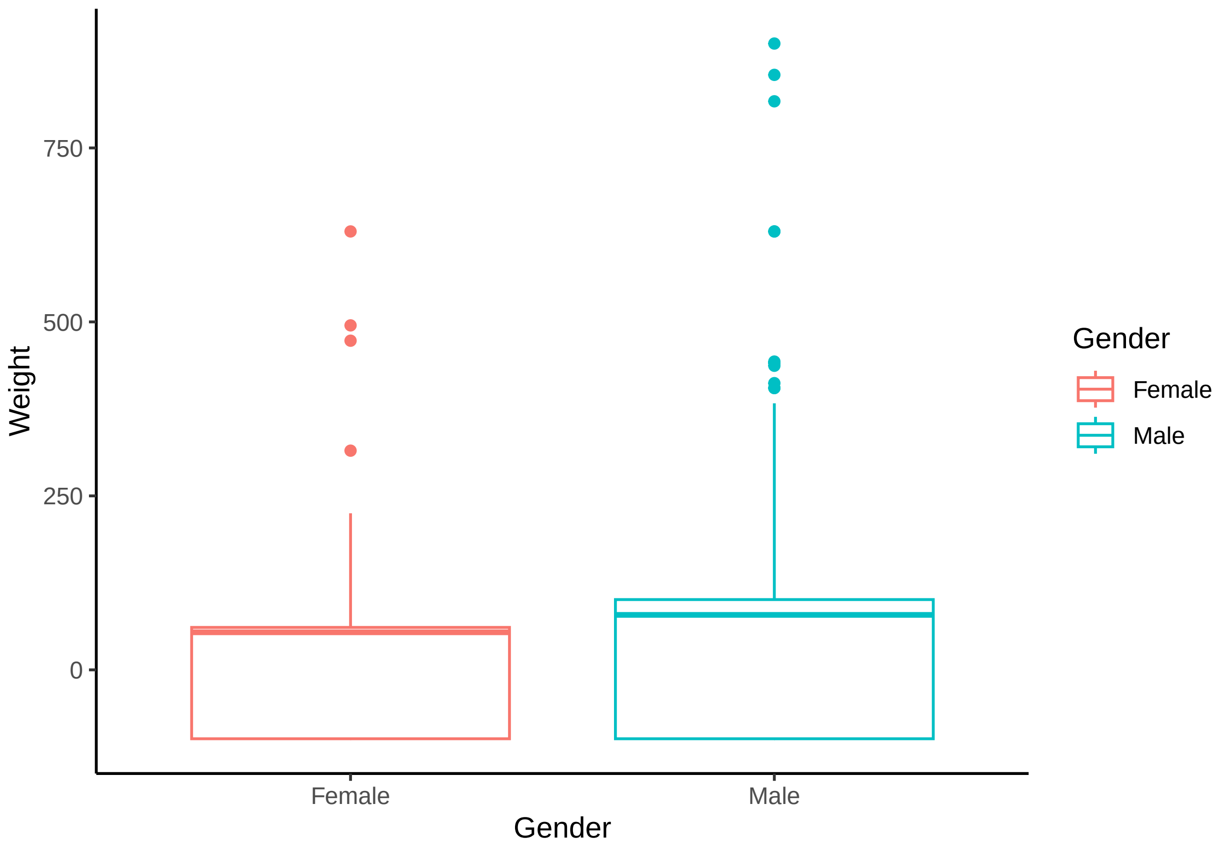
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



Let´s work now a bit on a different type of graph: boxplot

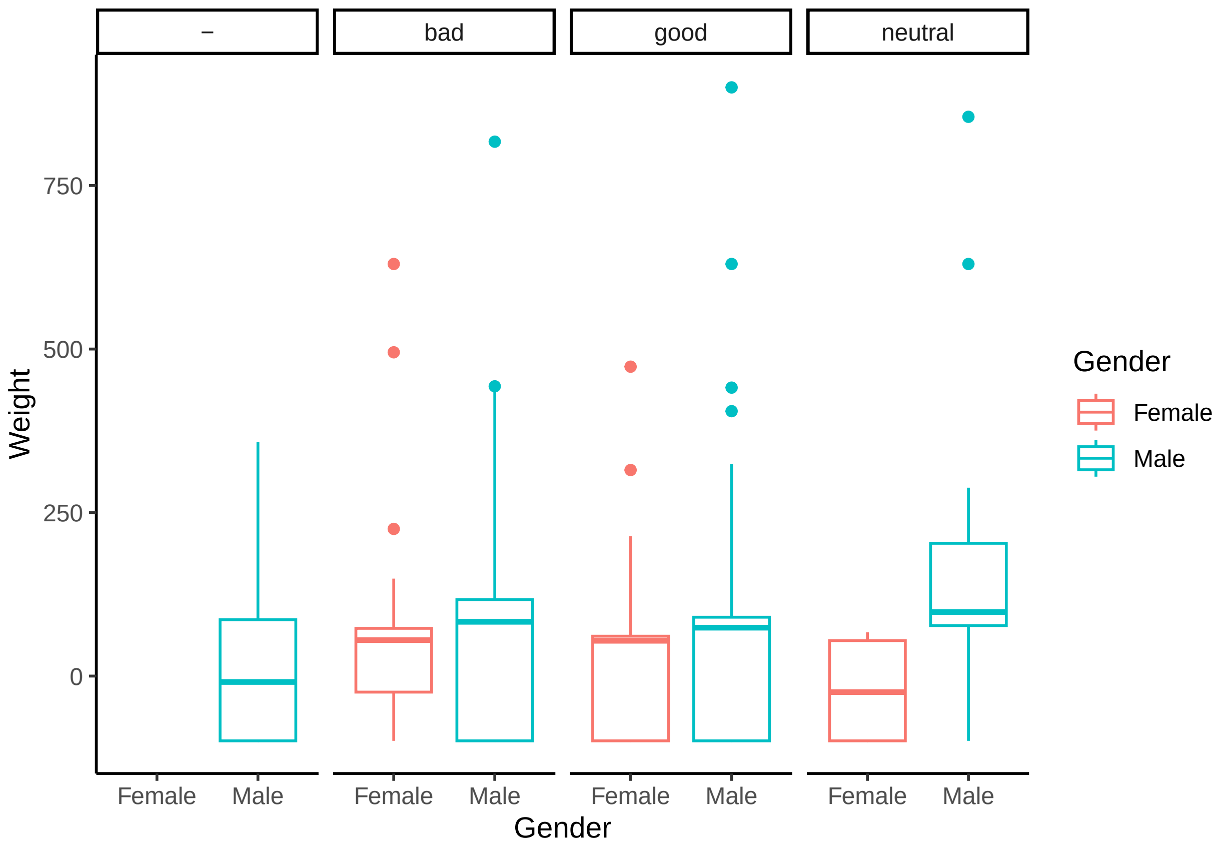
```
# Boxplot for Gender and Weight
ggplot(data = heroes, mapping = aes(Gender,Weight)) +
  geom_boxplot(aes(colour=Gender)) +
  theme_classic()
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



```
ggplot(data = heroes, mapping = aes(Gender,Weight)) +
  geom_boxplot(aes(colour=Gender))+
  theme_classic()+
  facet_grid(~Alignment)
```

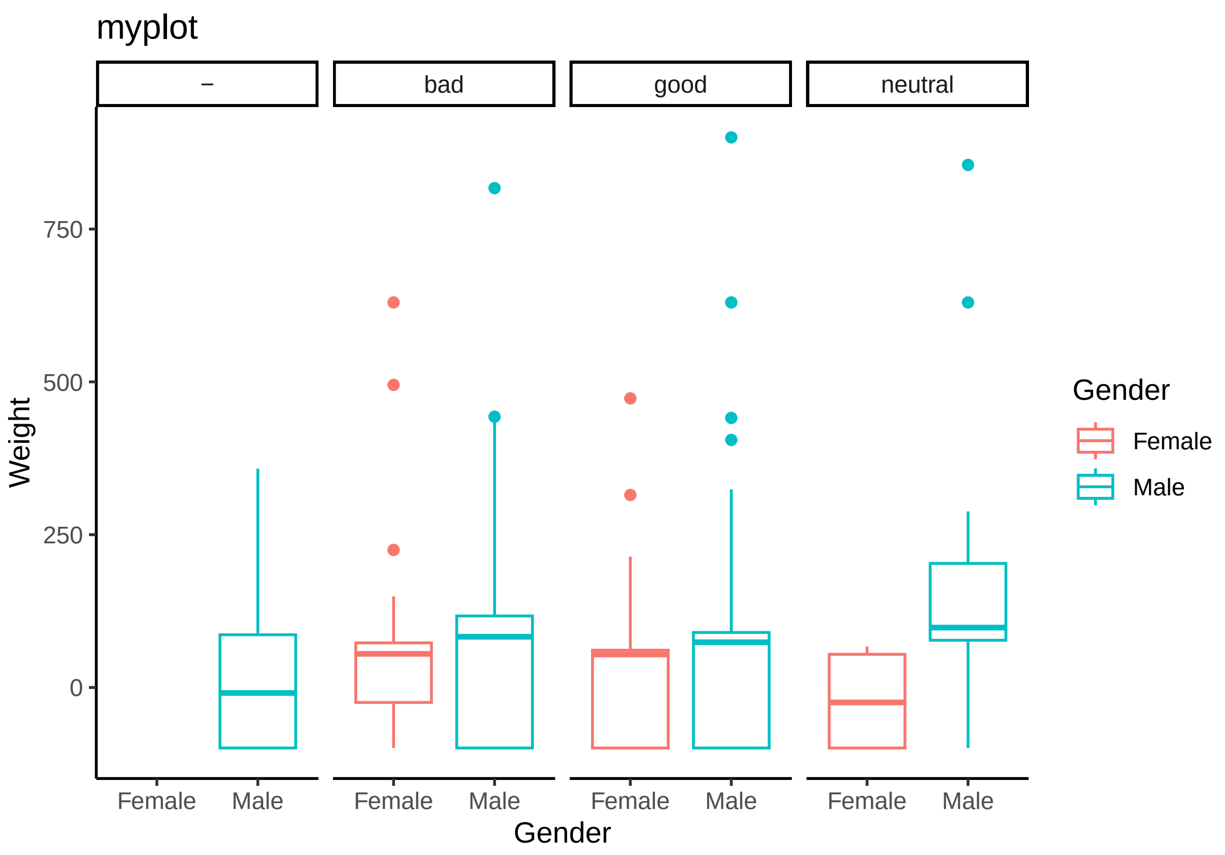
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Let´s learn how to save the plot as a variable and how to export it:

```
# Save as a an object that we can print
myplot = ggplot(data = heroes, mapping = aes(Gender, Weight)) +
  geom_boxplot(aes(colour=Gender)) +
  theme_classic() +
  facet_grid(~Alignment) +
  ggtitle ("myplot")

print(myplot)
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



```r
# Export the plot
ggsave("myplot.pdf",
       plot = myplot, width = 10, height = 5)
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```r
ggsave("myplot.png",
       plot = myplot)
```

```
## Saving 6.5 x 4.5 in image

## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

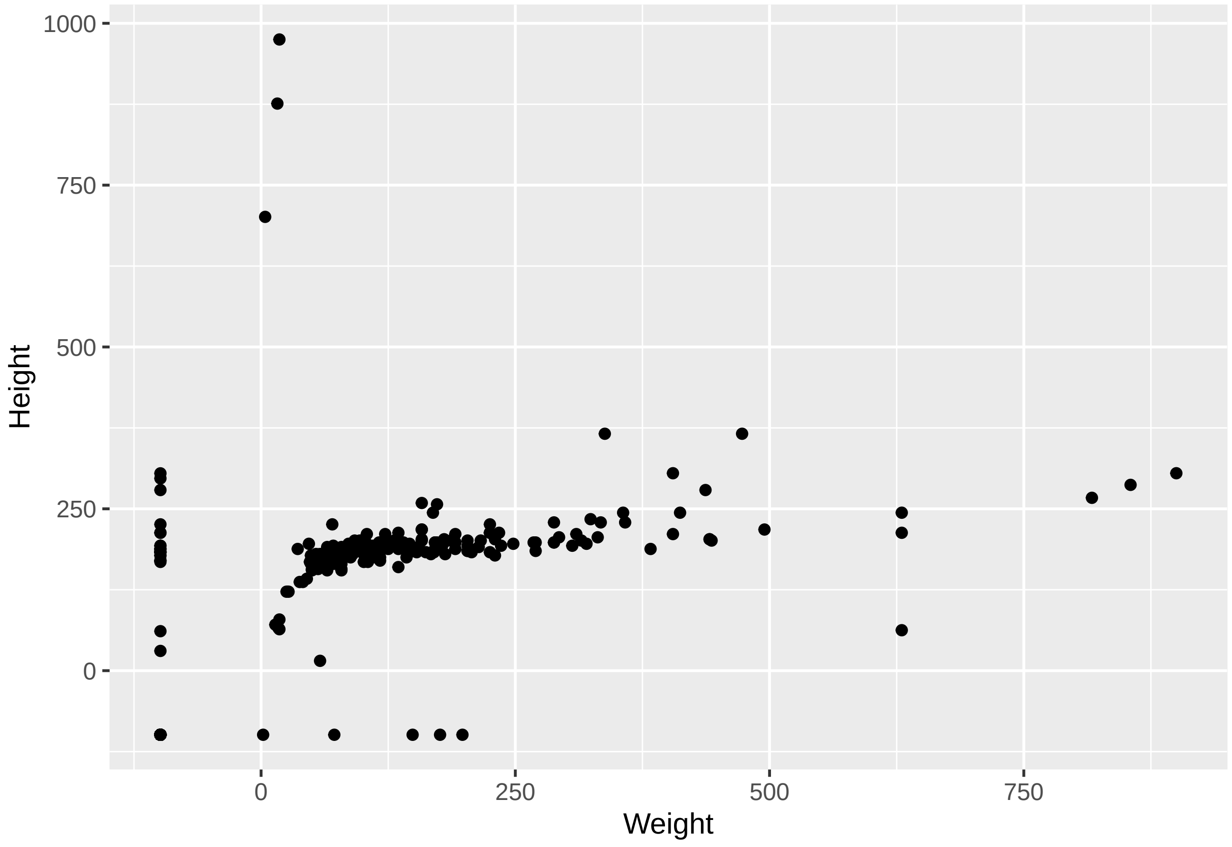
```r
ggsave("myplot.svg",
       plot = myplot)
```

```
## Saving 6.5 x 4.5 in image

## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

Some things to have in mind

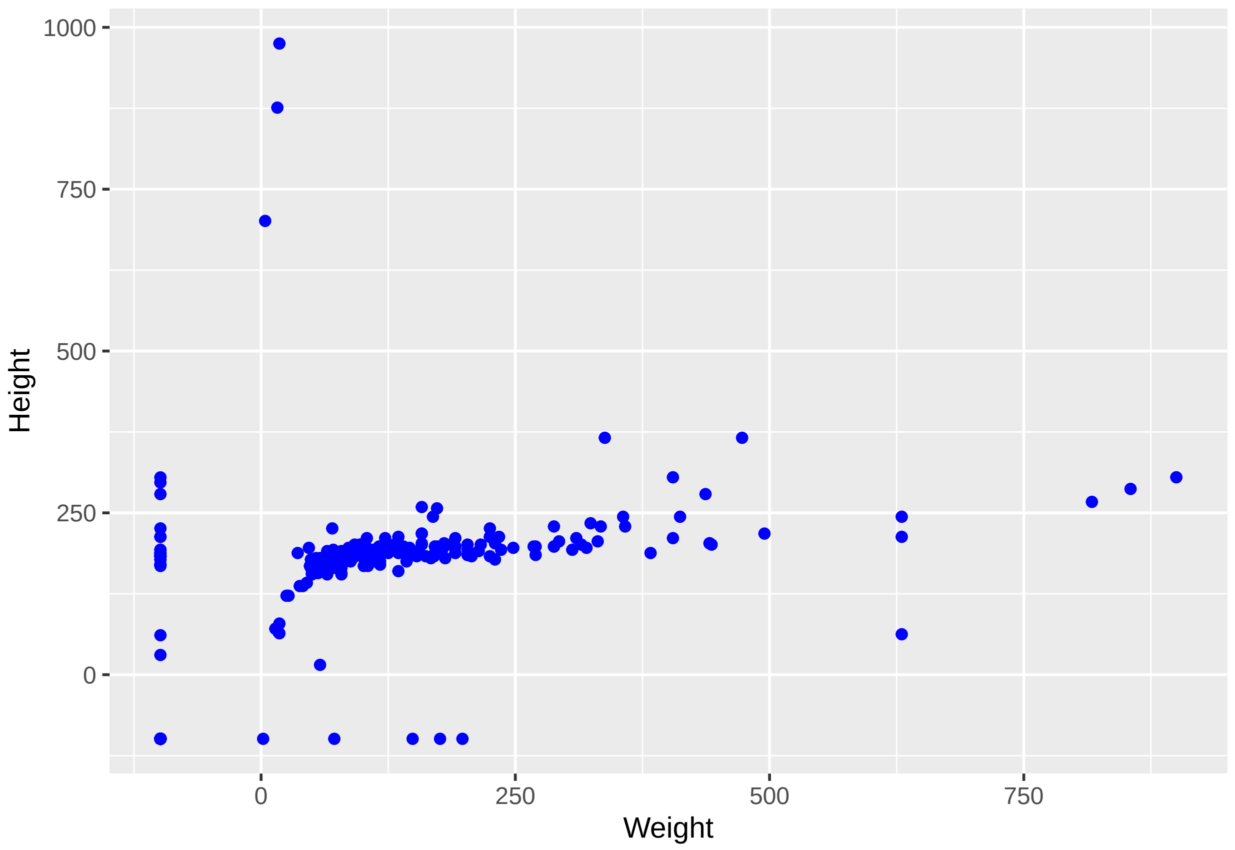
```r
# Let´s take this one graph we have worked with before
ggplot(data = heroes, mapping = aes(Weight,Height)) +
    geom_point()
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



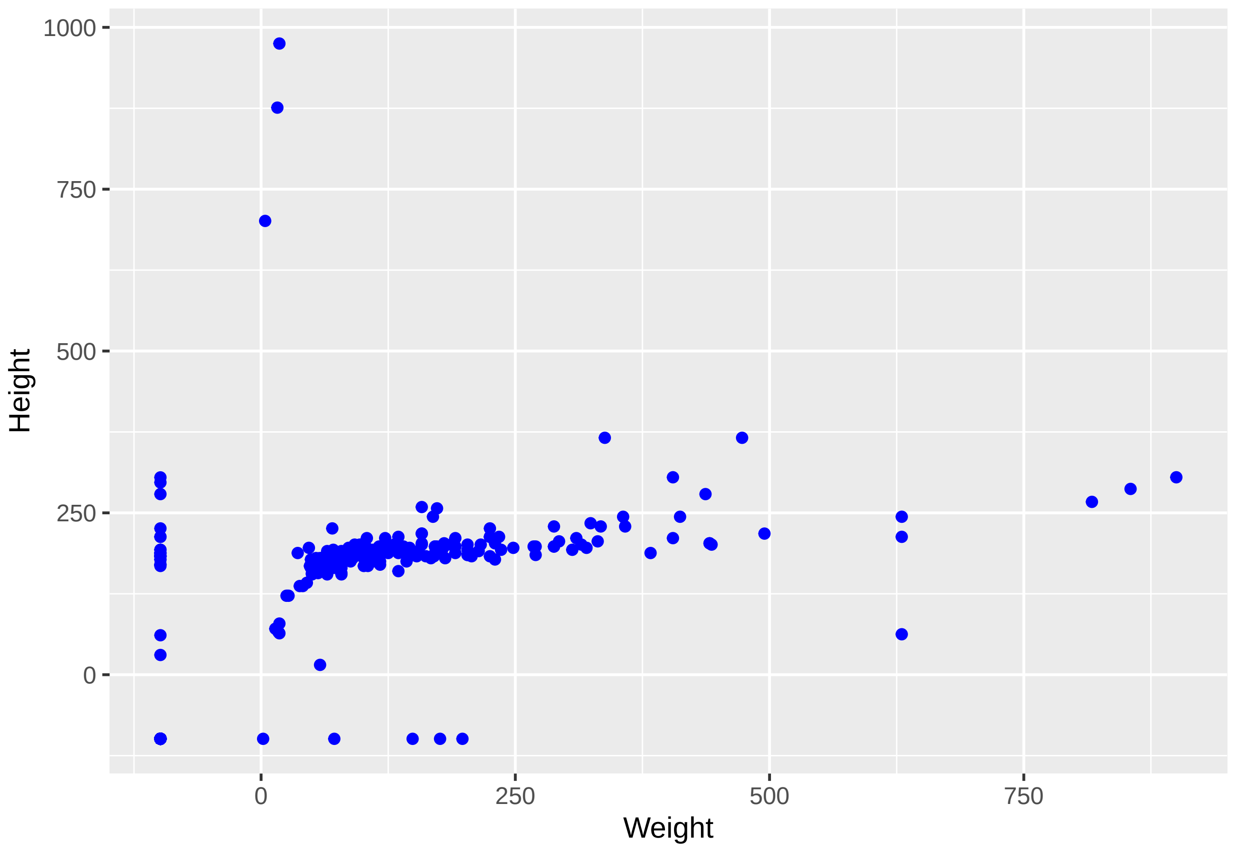
```r
# Let's add some colour
ggplot(data = heroes, mapping = aes(Weight,Height)) +
    geom_point(colour = "Blue")
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



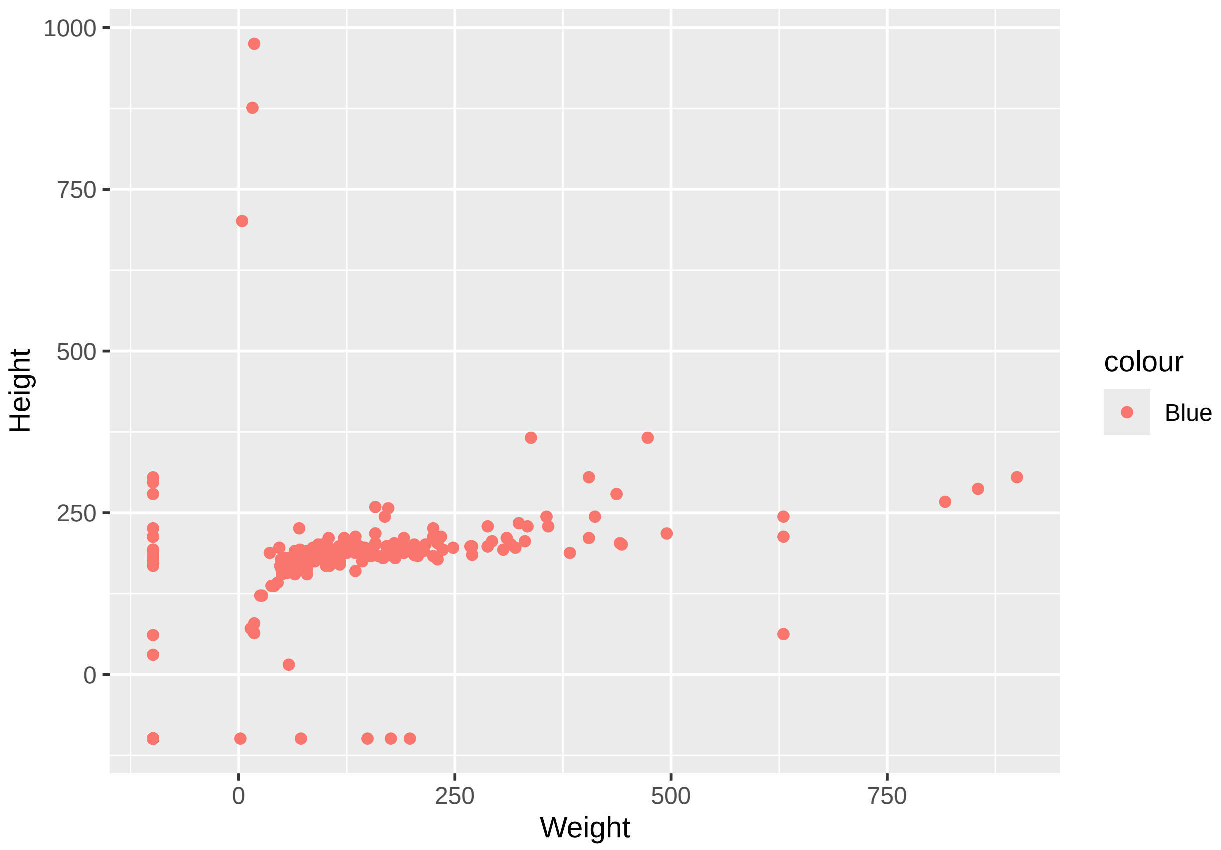
```
# Different way
ggplot(data = heroes) +
    geom_point( mapping = aes(Weight,Height),colour = "Blue")
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



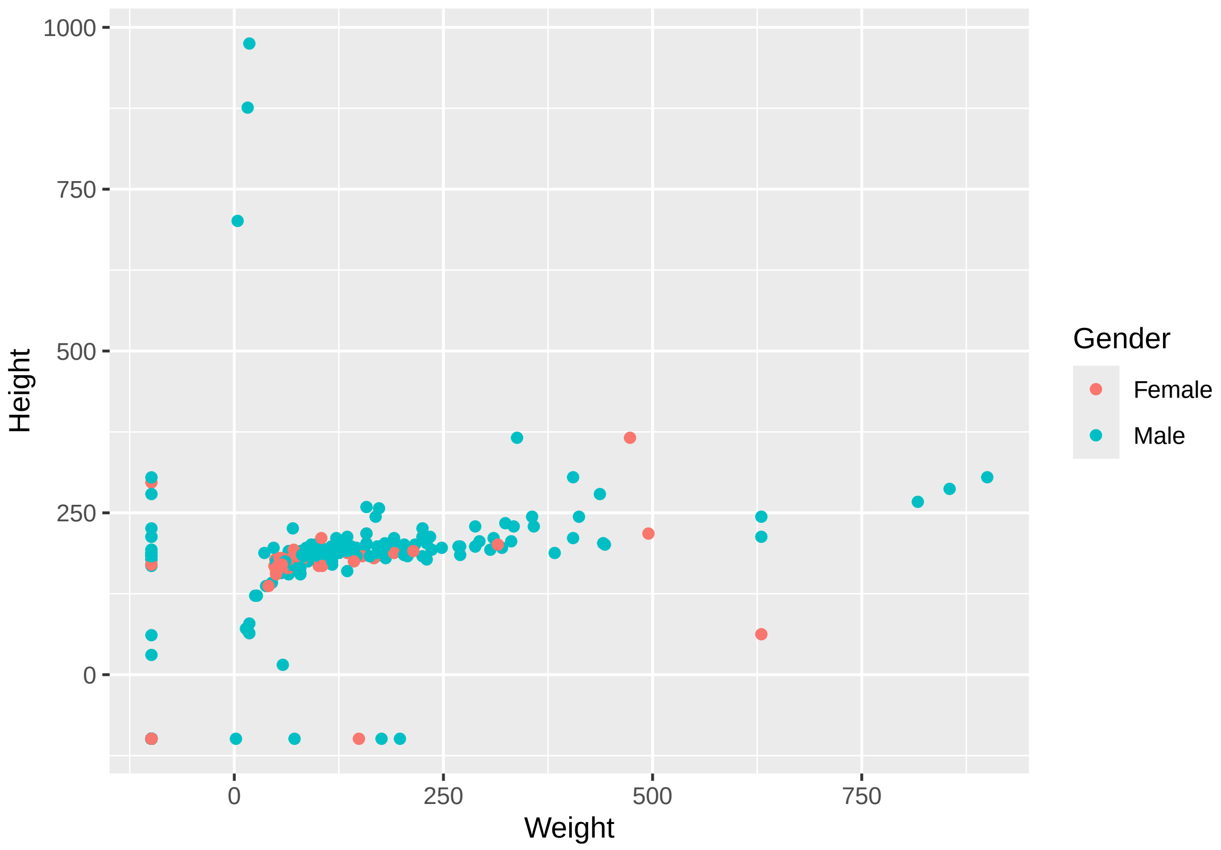
```r
# Be careful! Not the same as
ggplot(data = heroes)+
    geom_point( mapping = aes(Weight,Height,colour = "Blue")) #If we add the colour inside aesthetic th
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



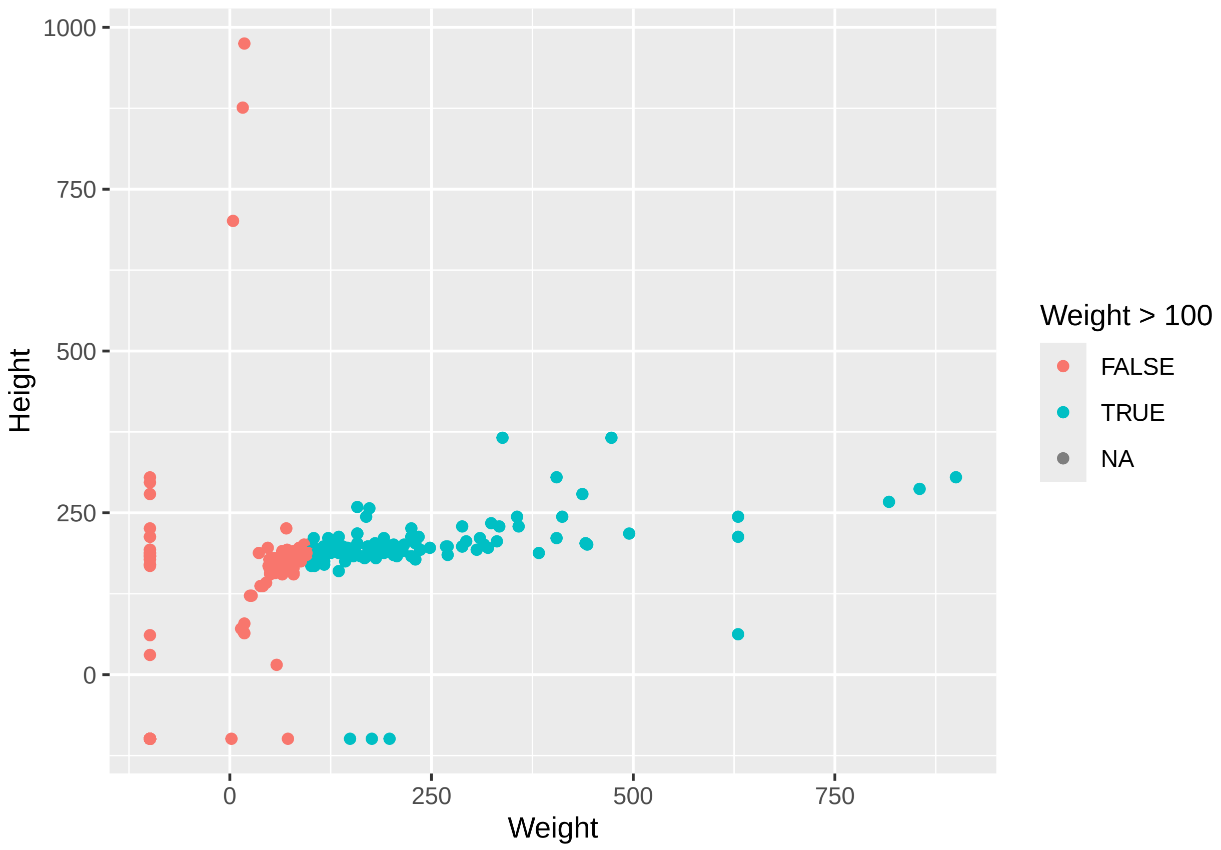
```
# Colour based on a variable
ggplot(data = heroes, mapping = aes(Weight,Height))+
    geom_point(aes(colour = Gender)) # Here it makes sense as colour is linked now to data
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



```r
# Specifiy colours
ggplot(data = heroes, mapping = aes(Weight,Height))+
    geom_point(aes(colour = Weight > 100)) # we can even decide a condition to based our colour in (we
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

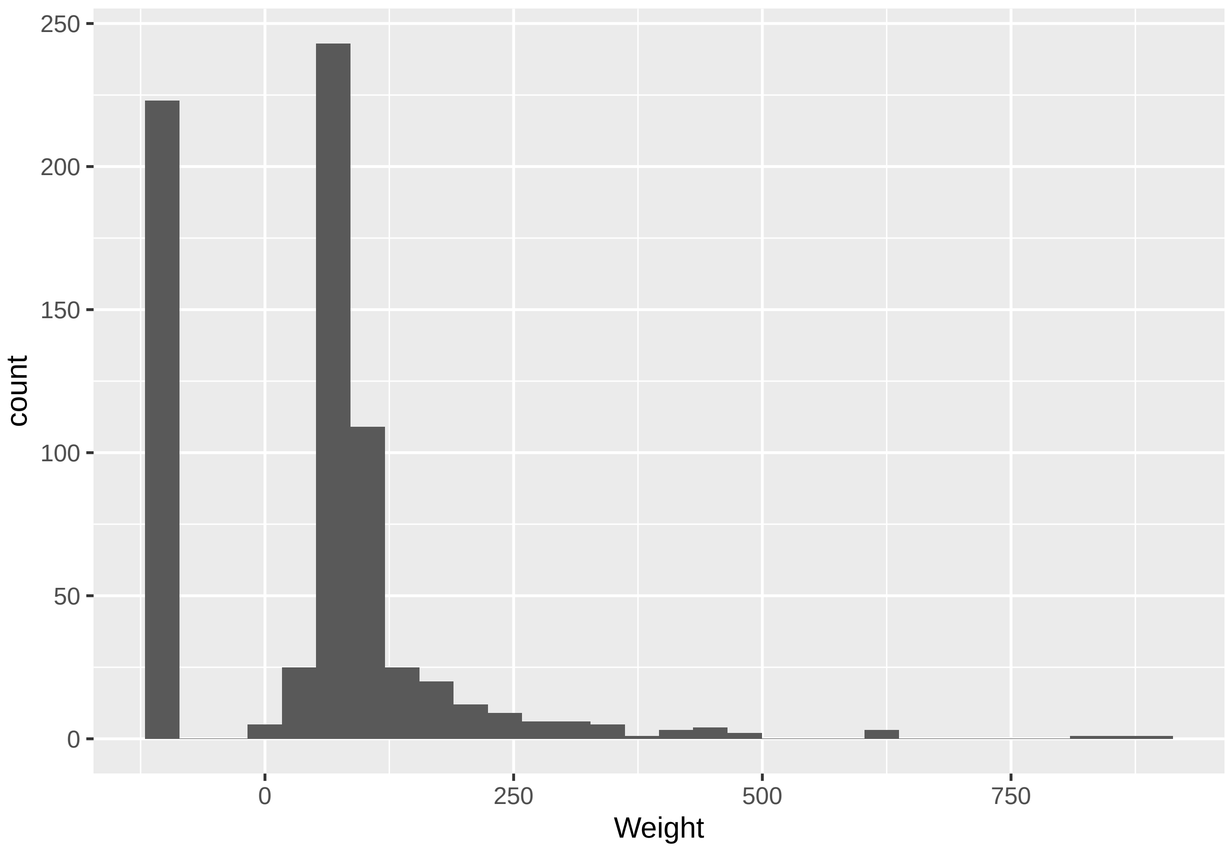


Some geometries only need a single mapping

```
ggplot(heroes) +
  geom_histogram(aes(x = Weight))
```

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

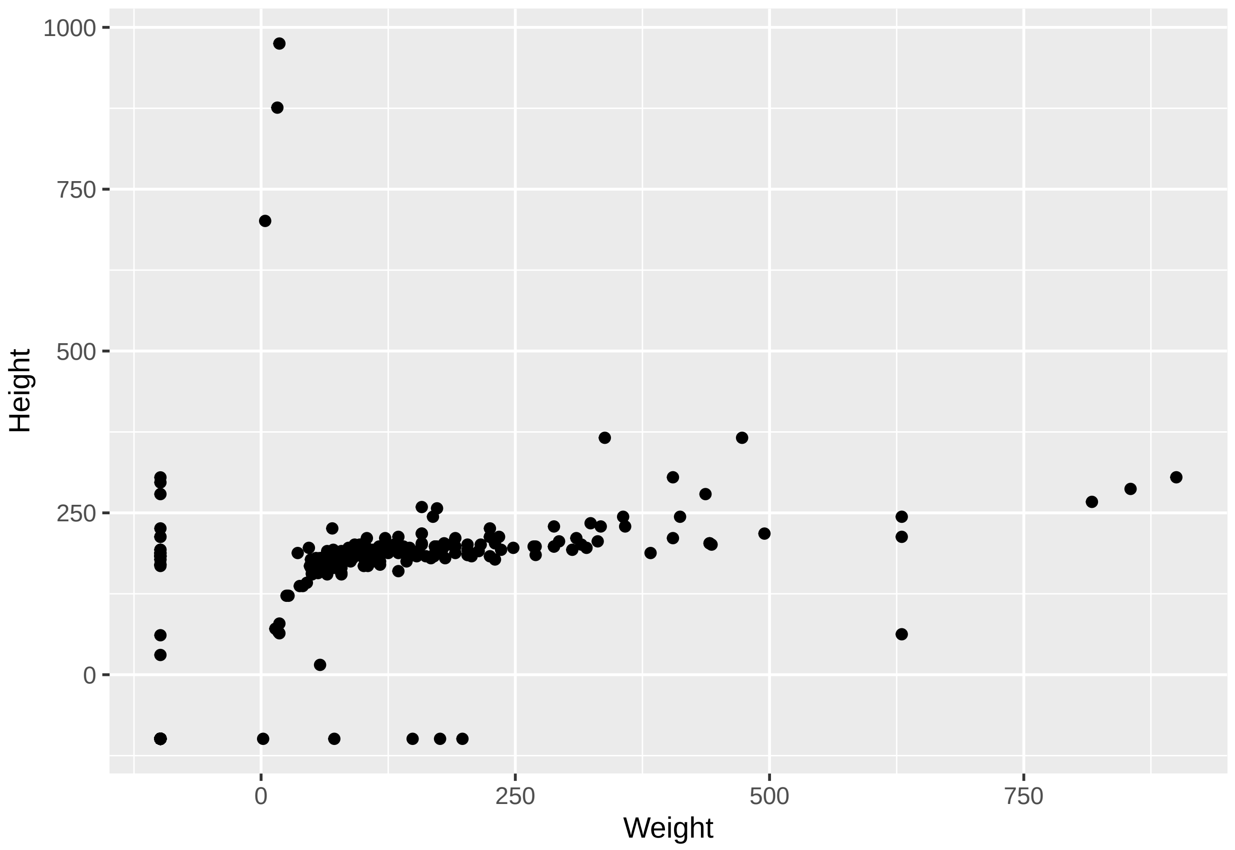
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```



Some practice Modify the code below to make points as squares and more transparent.

```r
ggplot(heroes) +
  geom_point(aes(Weight,  Height))
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

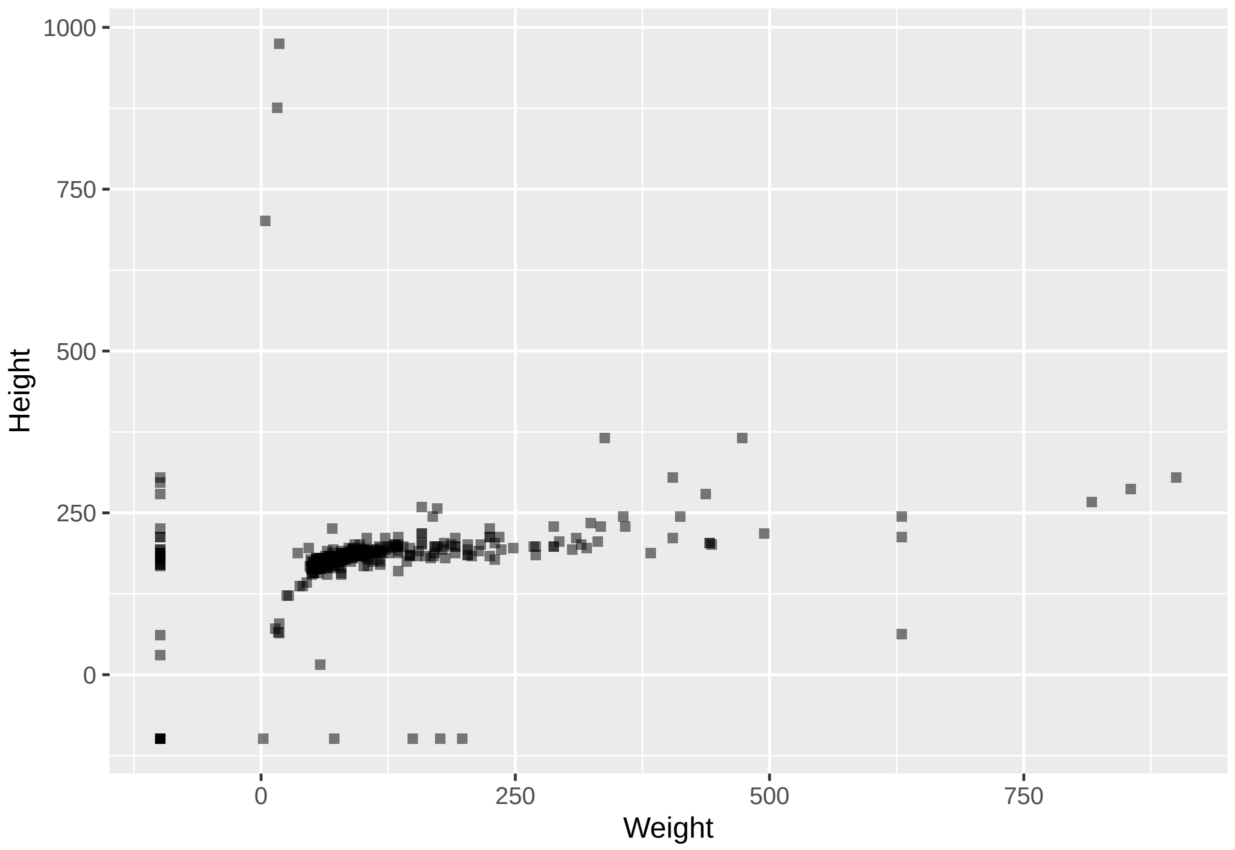


```r
# Symbols as squares


# Symbols transparent
```

Hint 1: transparency is controlled with alpha, and shape with shape Hint 2: remember the difference between mapping and setting aesthetics

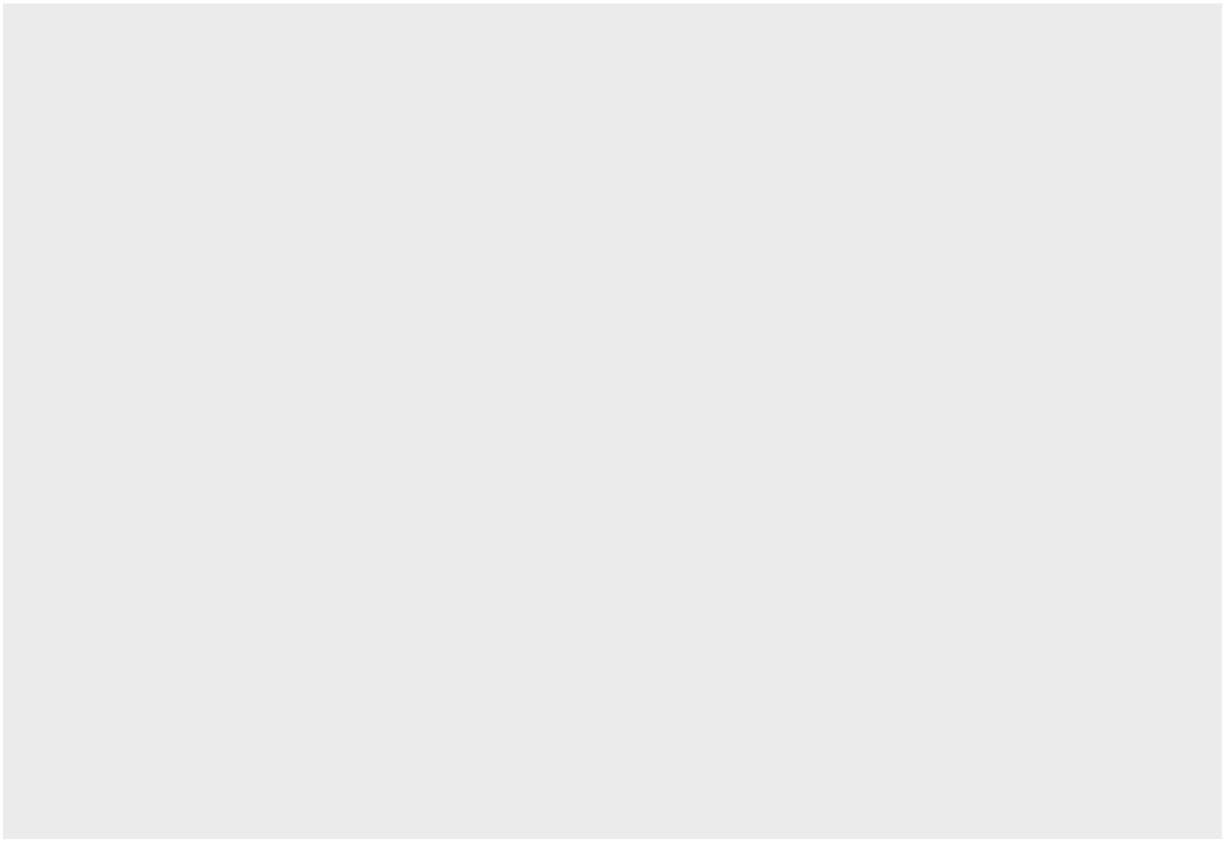
```r
# Solution
ggplot(heroes) +
  geom_point(aes(Weight,  Height), shape = "square", alpha=0.5)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



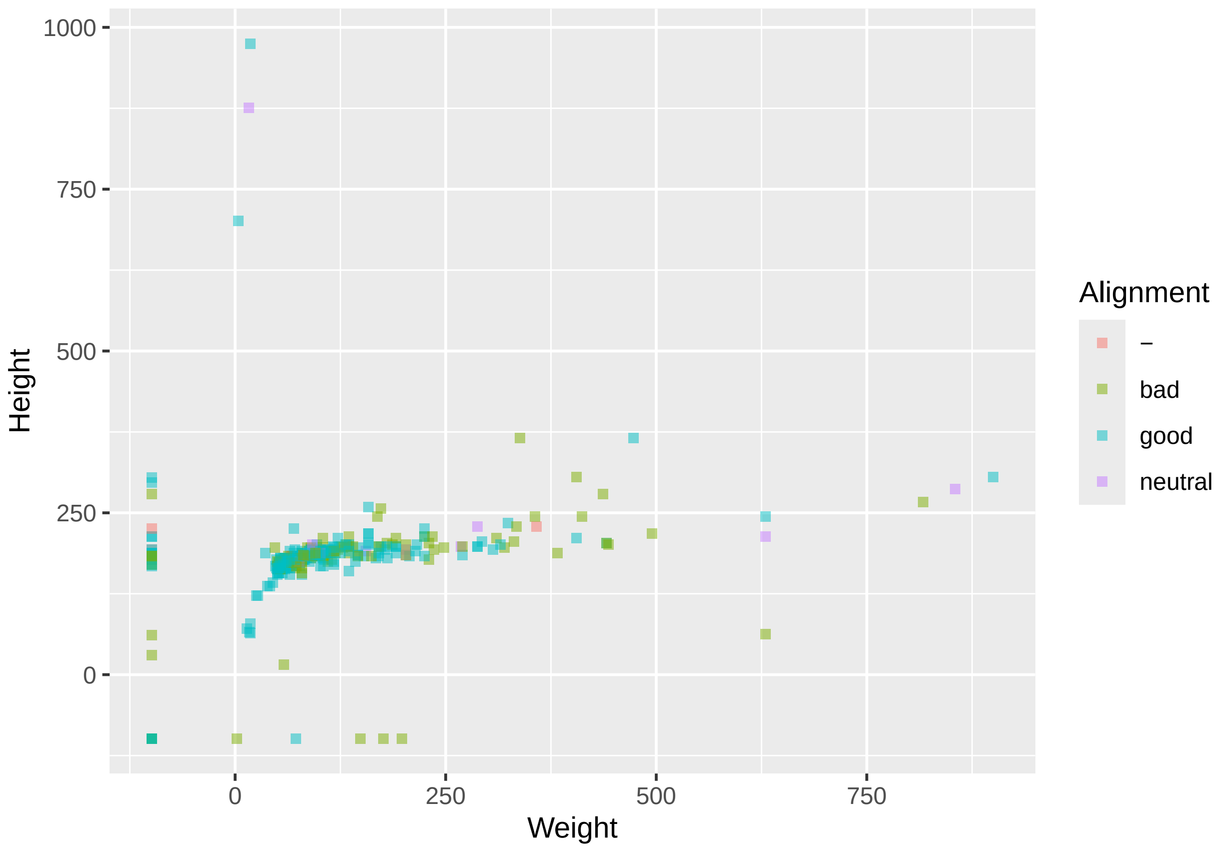
Colour the data on the graph depending on alignment

```
ggplot()
```



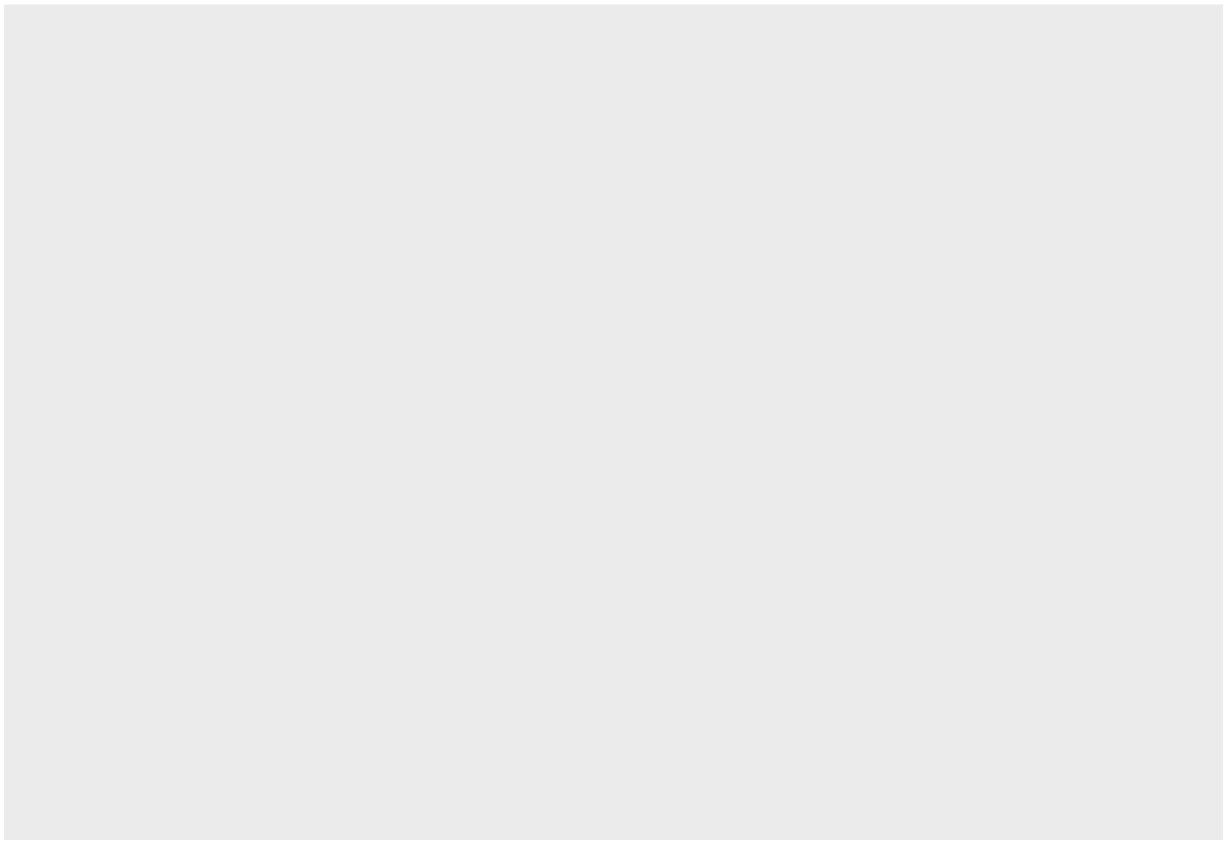
```r
# Solution
ggplot(heroes) +
  geom_point(aes(Weight,  Height, colour= Alignment), shape = "square", alpha=0.5)
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



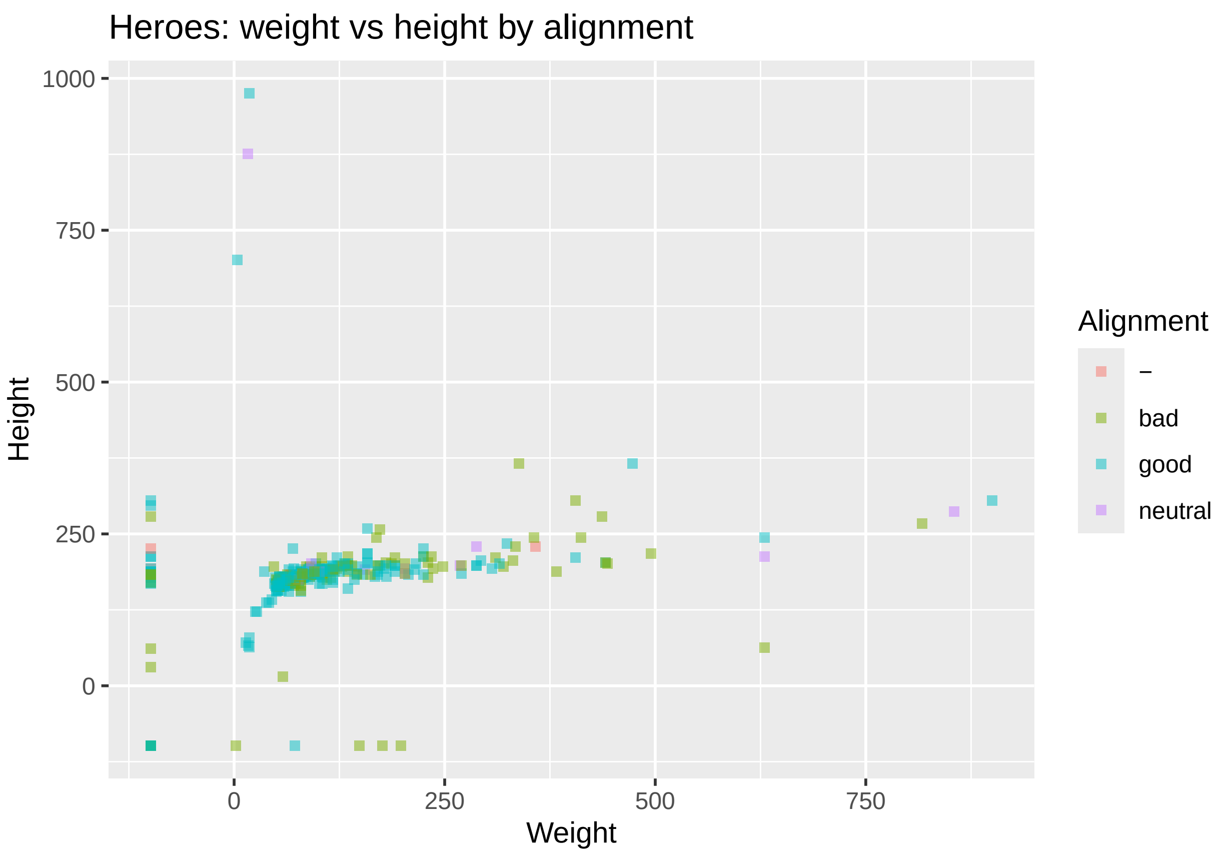
Now add a main title

```r
ggplot()
```



```r
# Solution
ggplot(heroes) +
  geom_point(aes(Weight,  Height, colour= Alignment), shape = "square", alpha=0.5) +
  ggtitle ("Heroes: weight vs height by alignment")
```

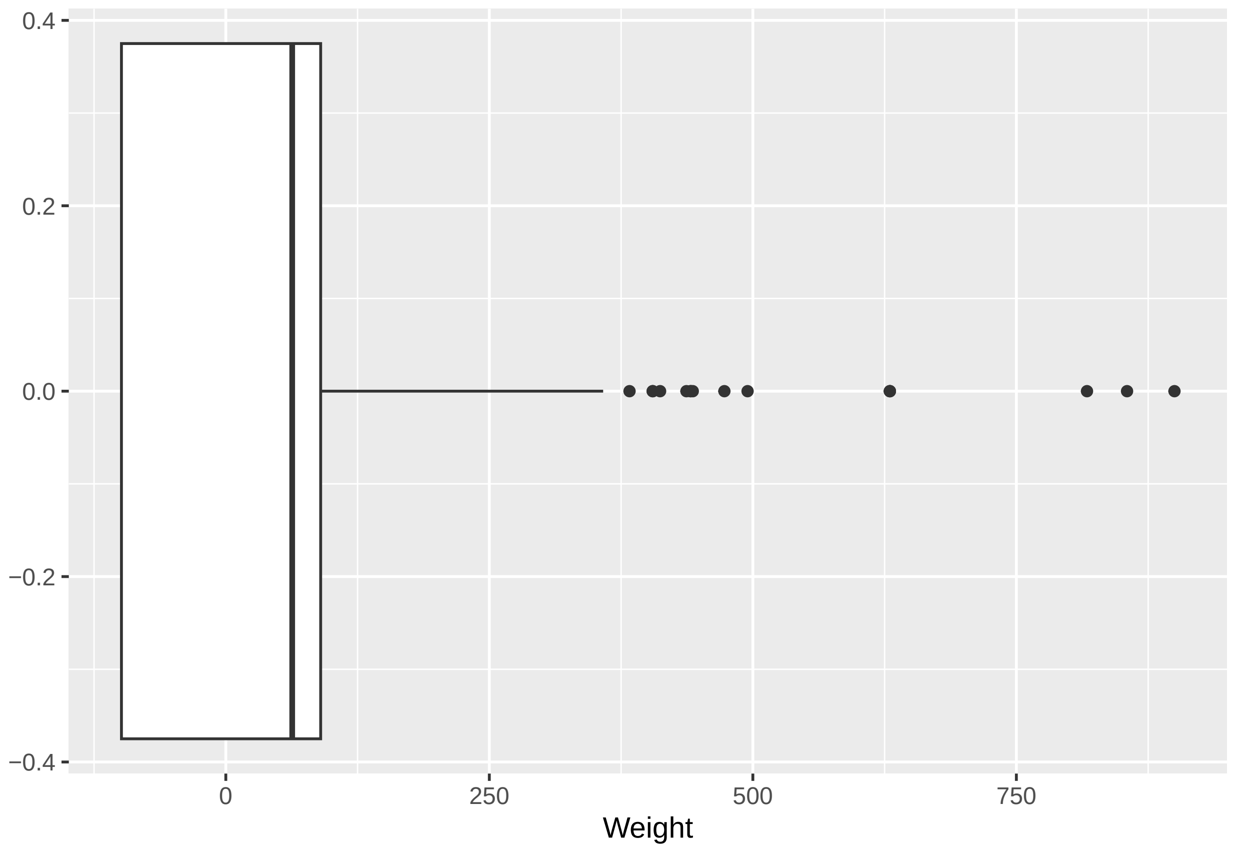
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



Now let´s work with categorical data and boxplot. First plot just the continuos data Weigth

```
ggplot(data = heroes, mapping = aes(Weight)) +
  geom_boxplot()
```

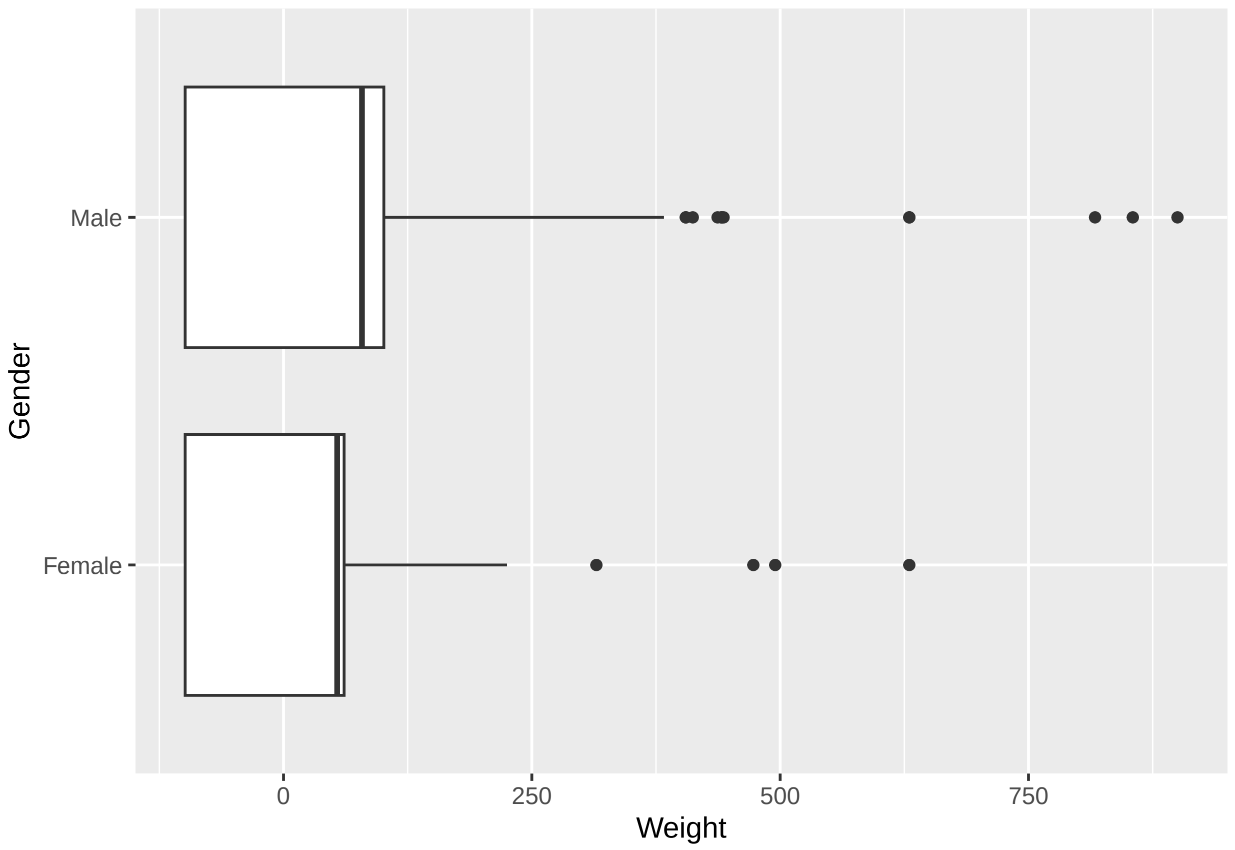
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



We can see the boxplot representing the weights of the whole dataset. With a line there is the median represented. Let´s add now the categorical data we would like to plot too.

```
ggplot(data = heroes, mapping = aes(Weight,Gender)) +
  geom_boxplot()
```

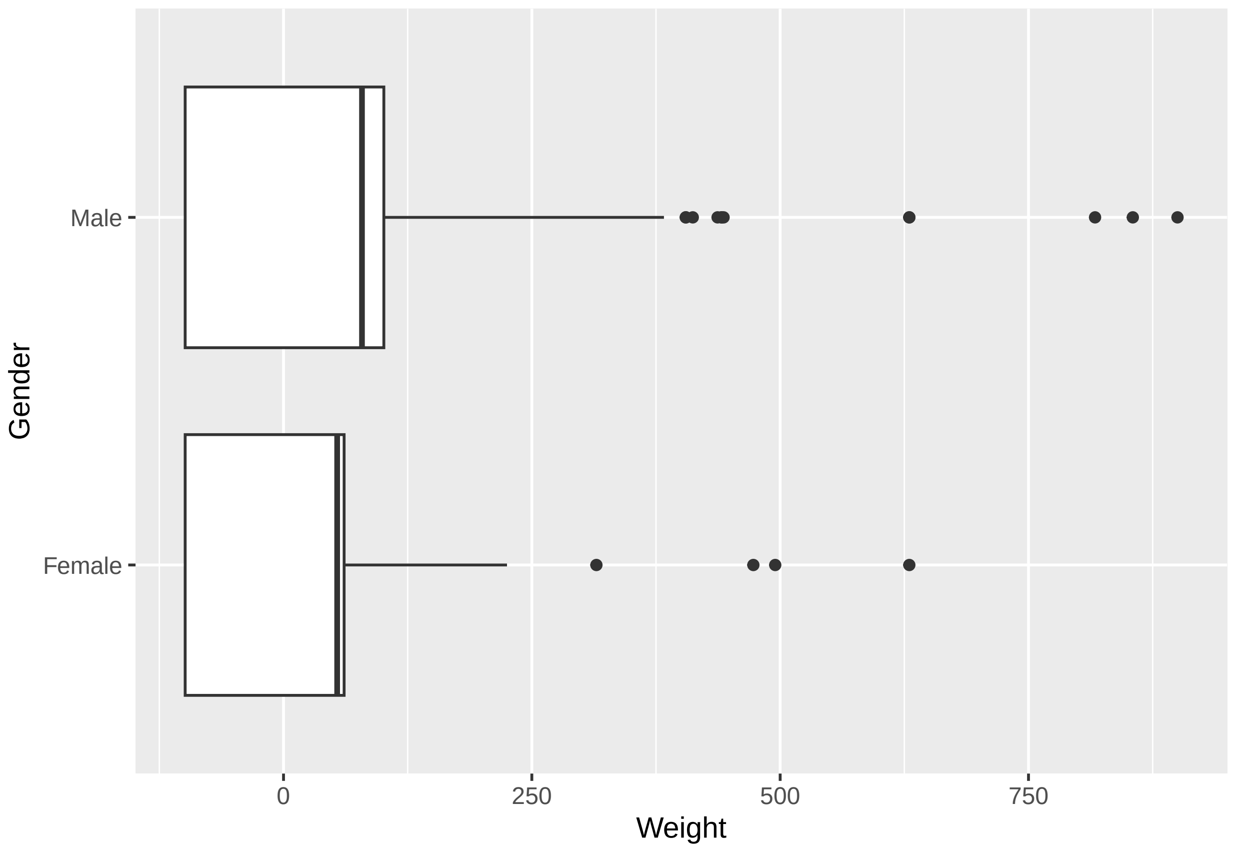
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Let´s consider just Male and Female data

```
ggplot(data = heroes, mapping = aes(Weight,Gender)) +
  geom_boxplot()
```

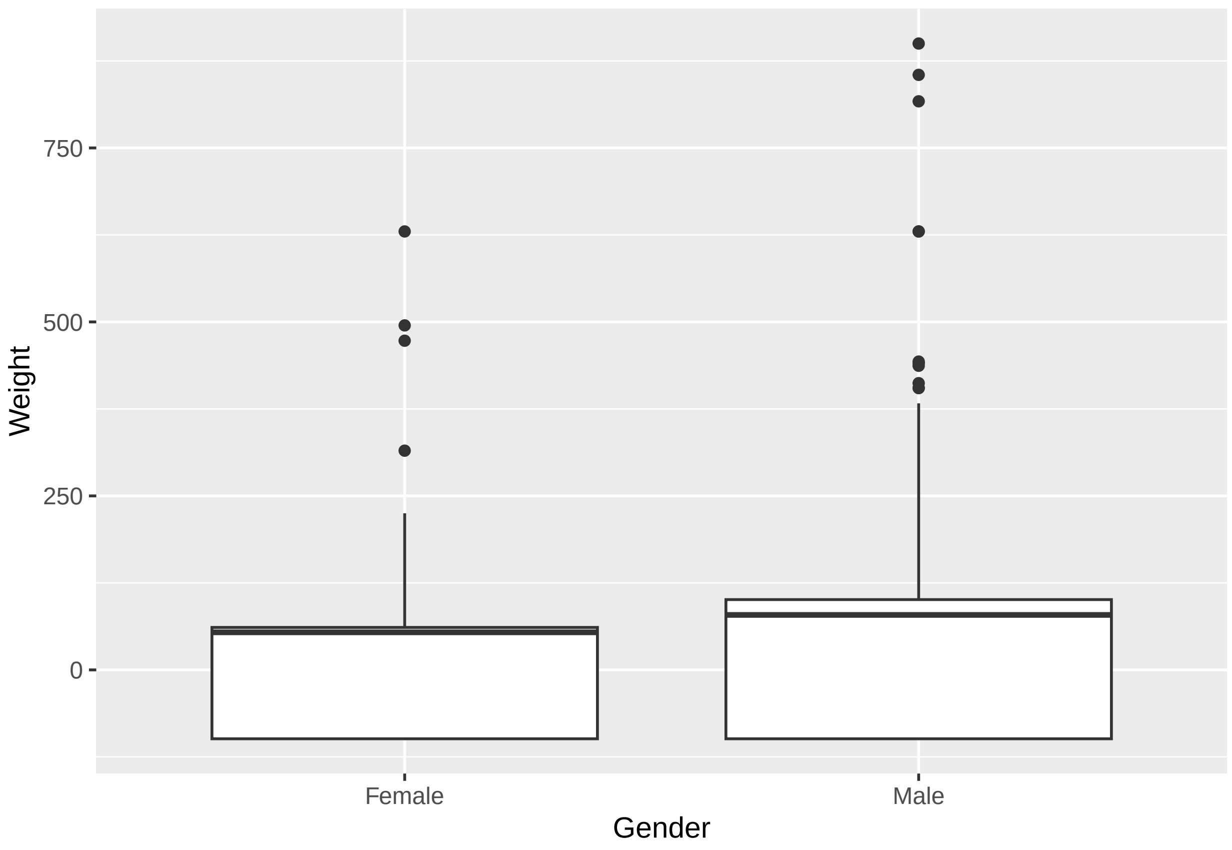
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Try to change the axis orientation so that Gender shows in x axis

```r
ggplot(data = heroes, mapping = aes(Gender, Weight)) +
  geom_boxplot()
```

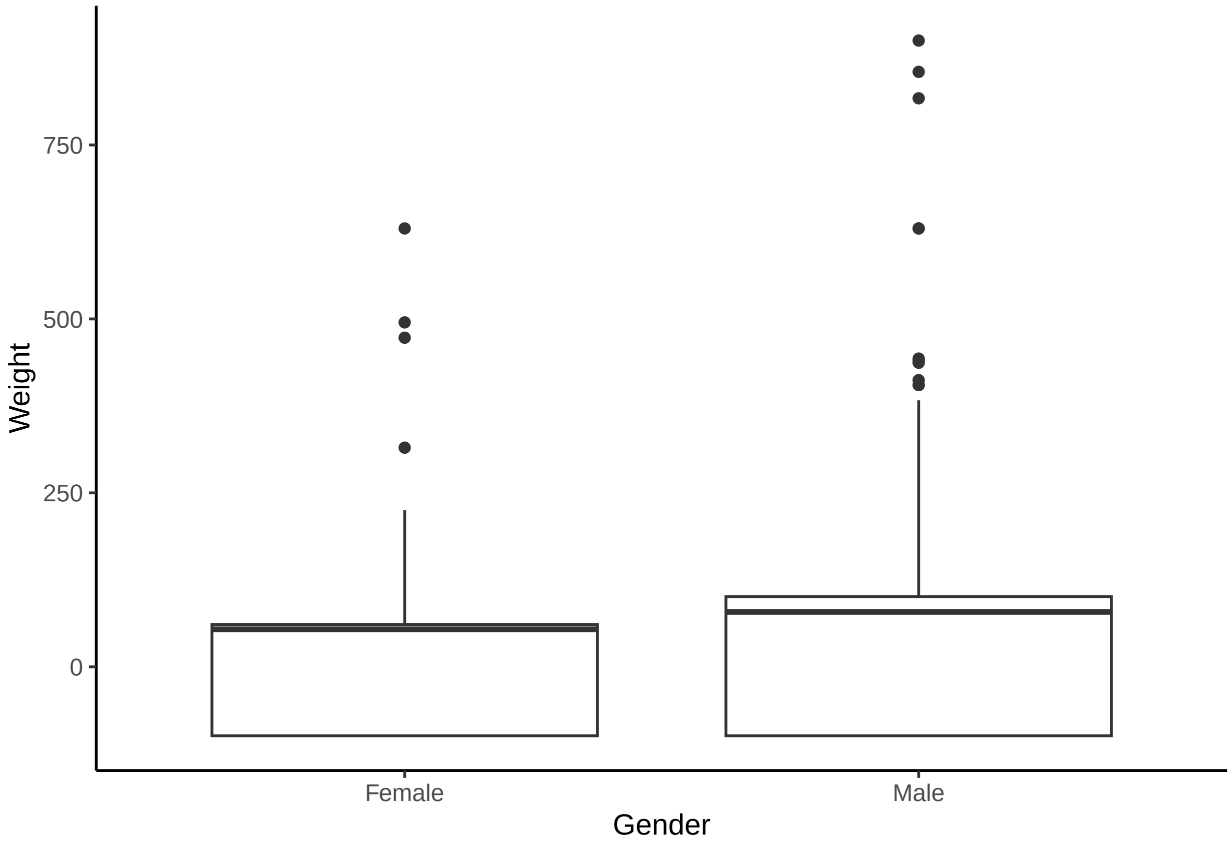
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Let´s add the classic theme

```r
ggplot(data = heroes, mapping = aes(Gender, Weight)) +
  geom_boxplot() +
  theme_classic()
```

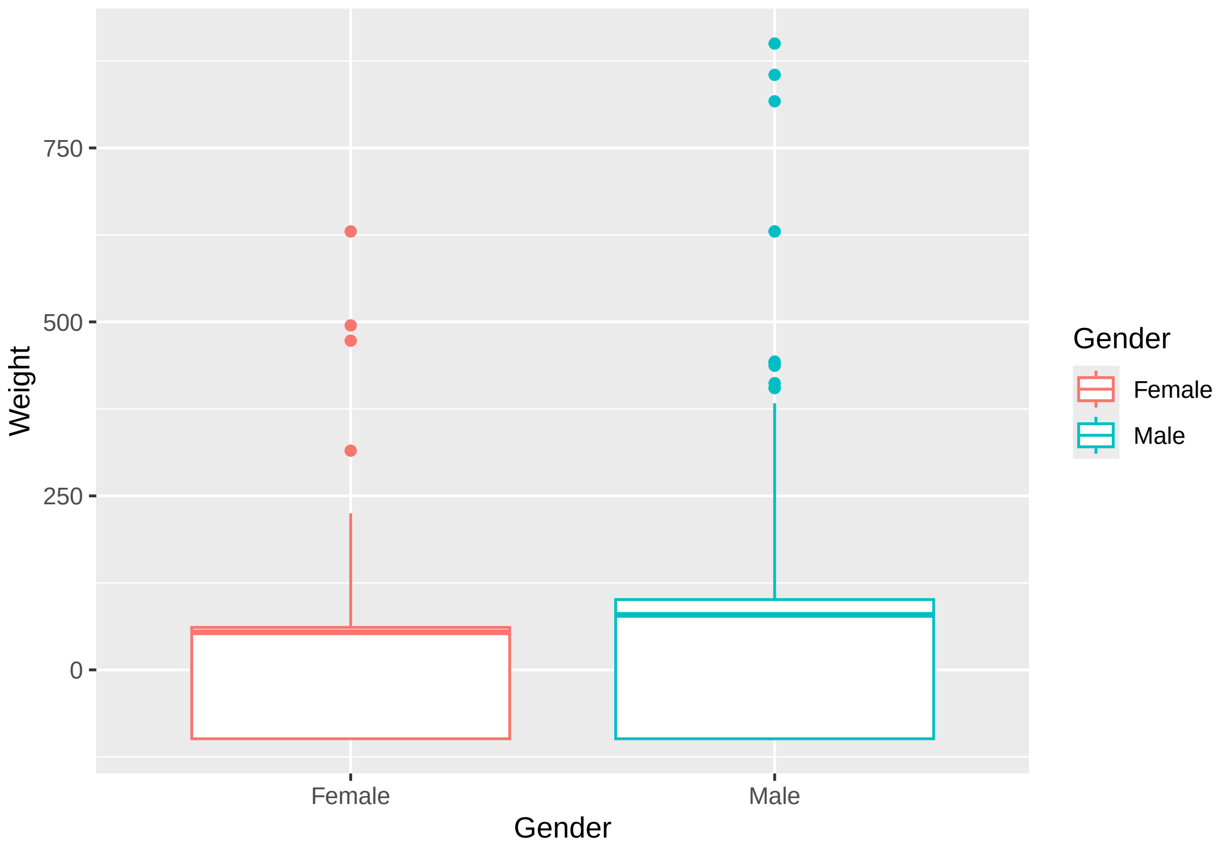
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Add some colour

```r
ggplot(data = heroes, mapping = aes(Gender, Weight, colour= Gender)) +
  geom_boxplot()
```

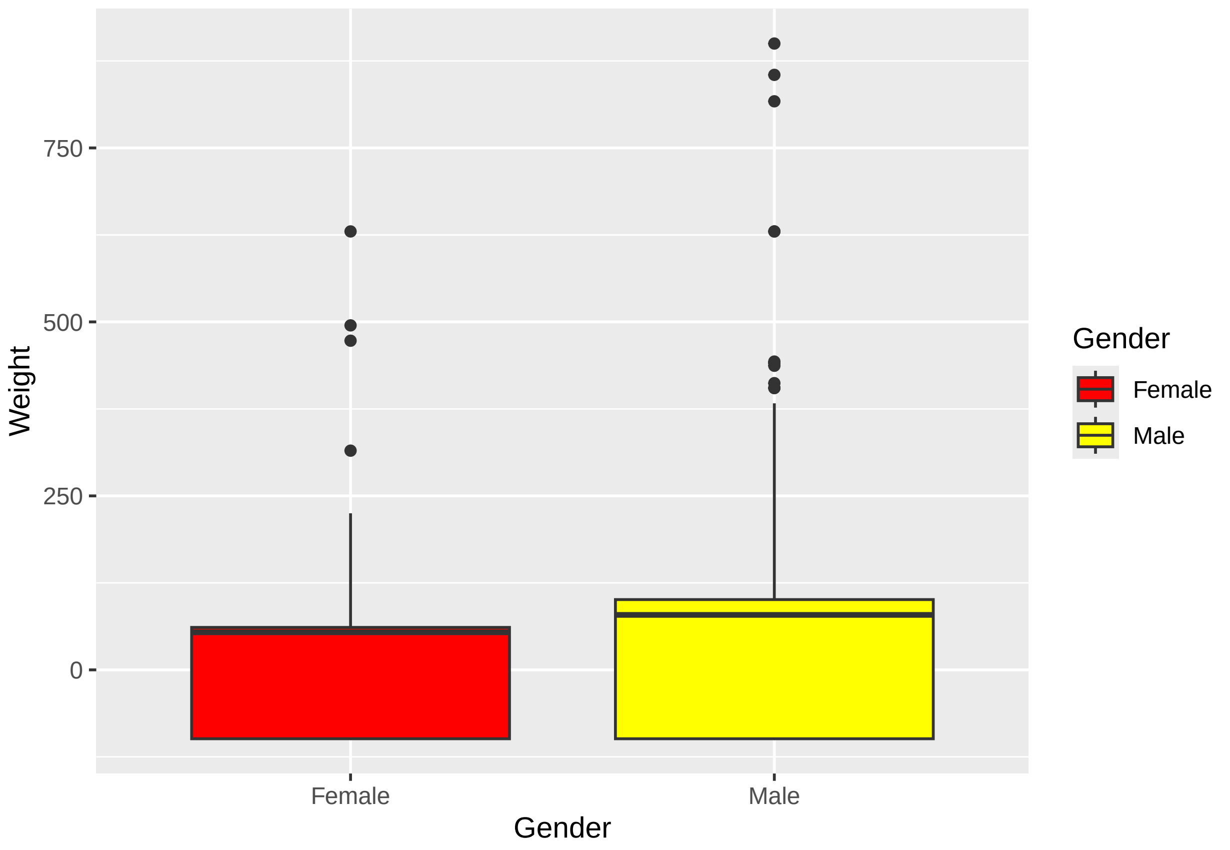
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Choose your own colours

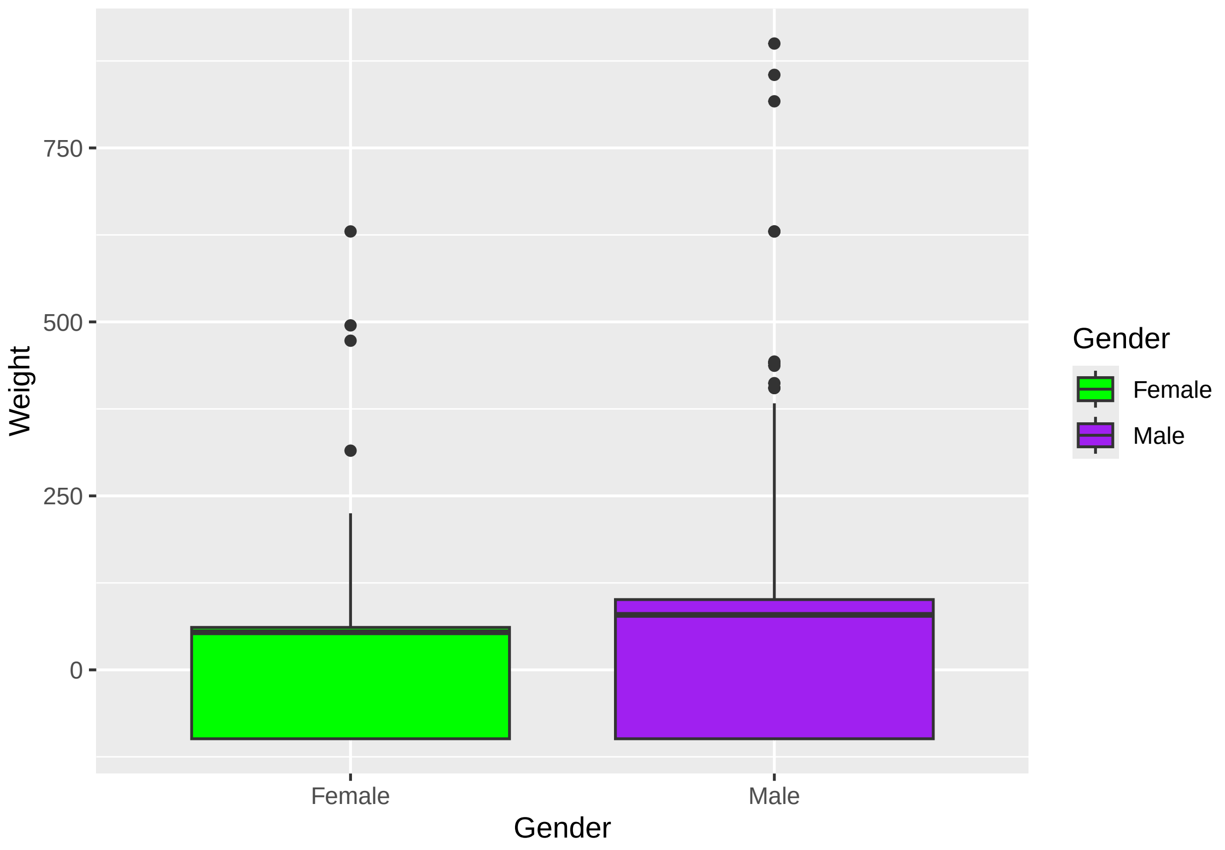
```r
ggplot(data = heroes, mapping = aes(Gender, Weight, fill= Gender)) +
  geom_boxplot() +
  scale_fill_manual(values=c("red", "yellow"))
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



```r
ggplot(data = heroes, mapping = aes(Gender, Weight)) +
  geom_boxplot(aes(fill= Gender)) +
  scale_fill_manual(values=c("green", "purple"))
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

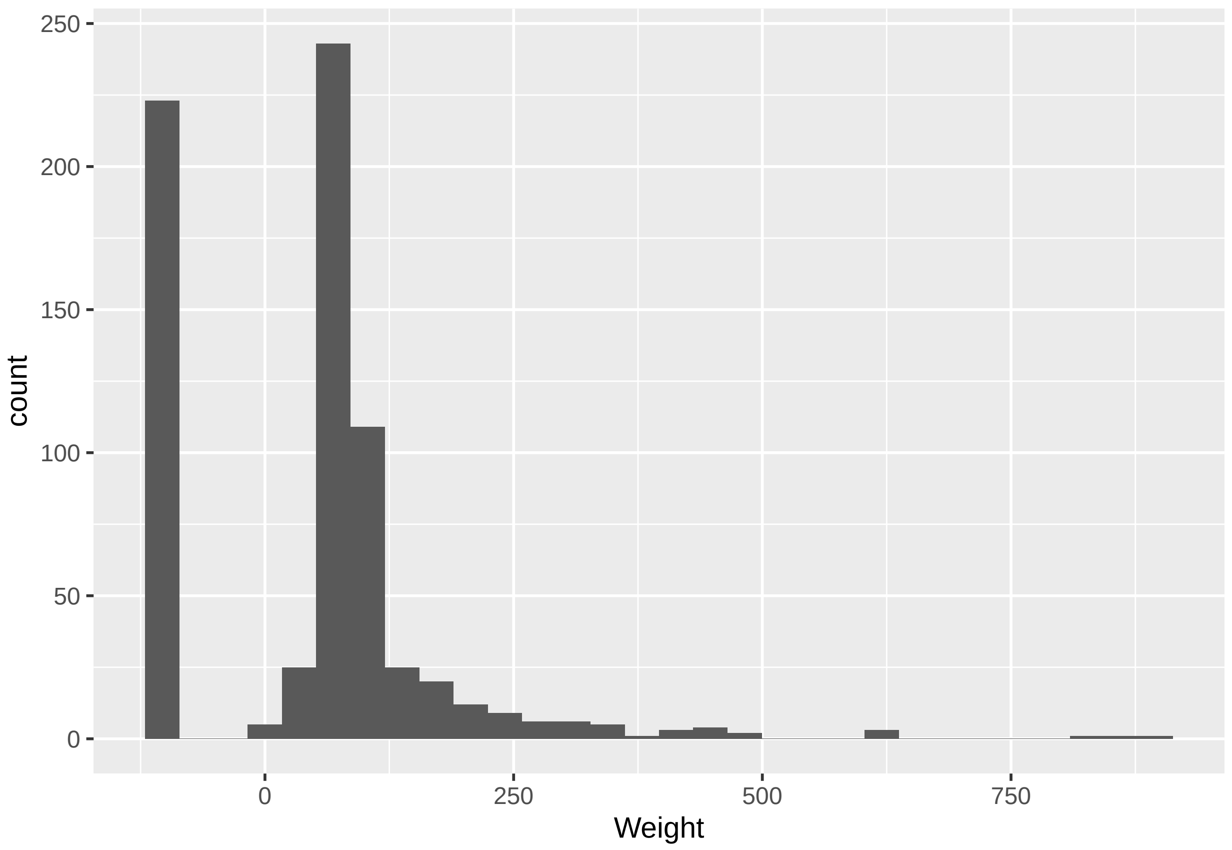


Let us have a look to the following histogram

```
ggplot(heroes, aes(Weight)) +
  geom_histogram()
```

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

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```

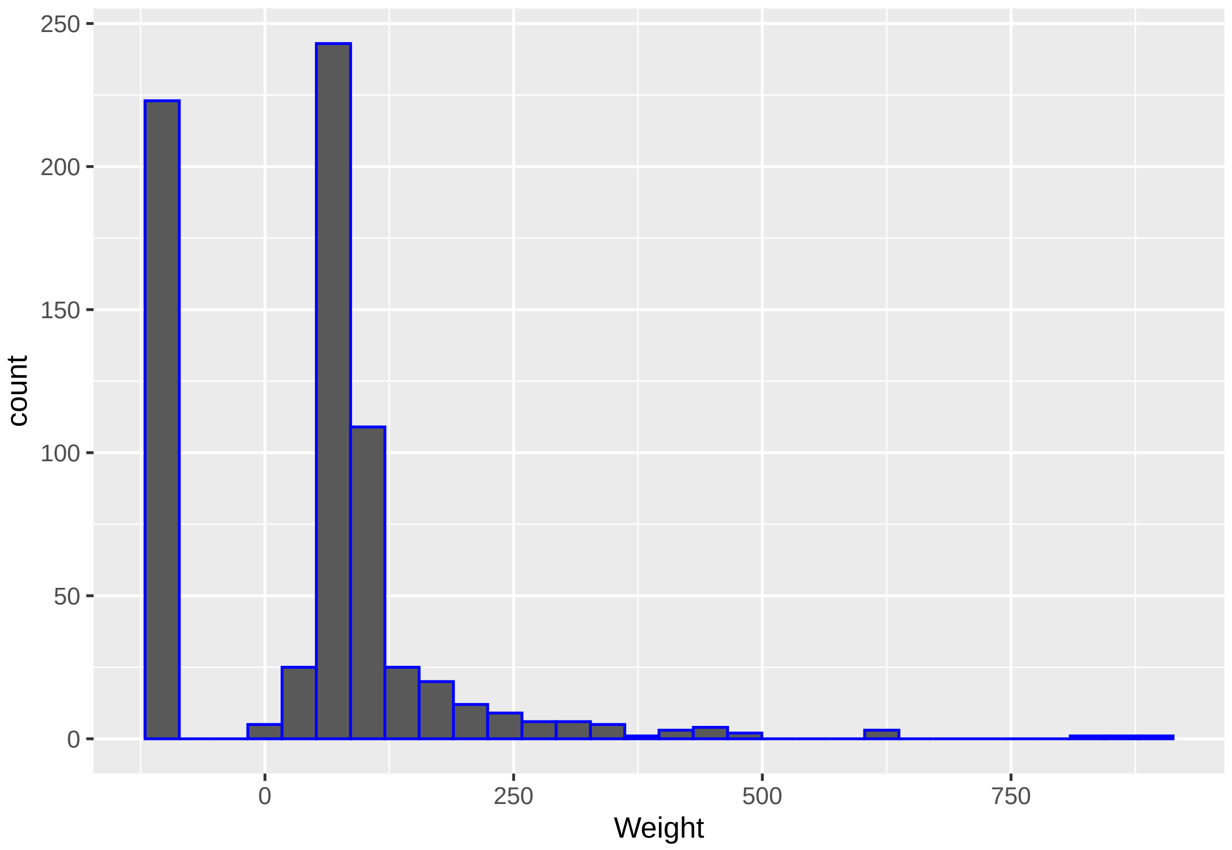


Colour the bars in blue:

```r
ggplot(heroes, aes(Weight)) +
  geom_histogram(colour="Blue")
```

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

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```

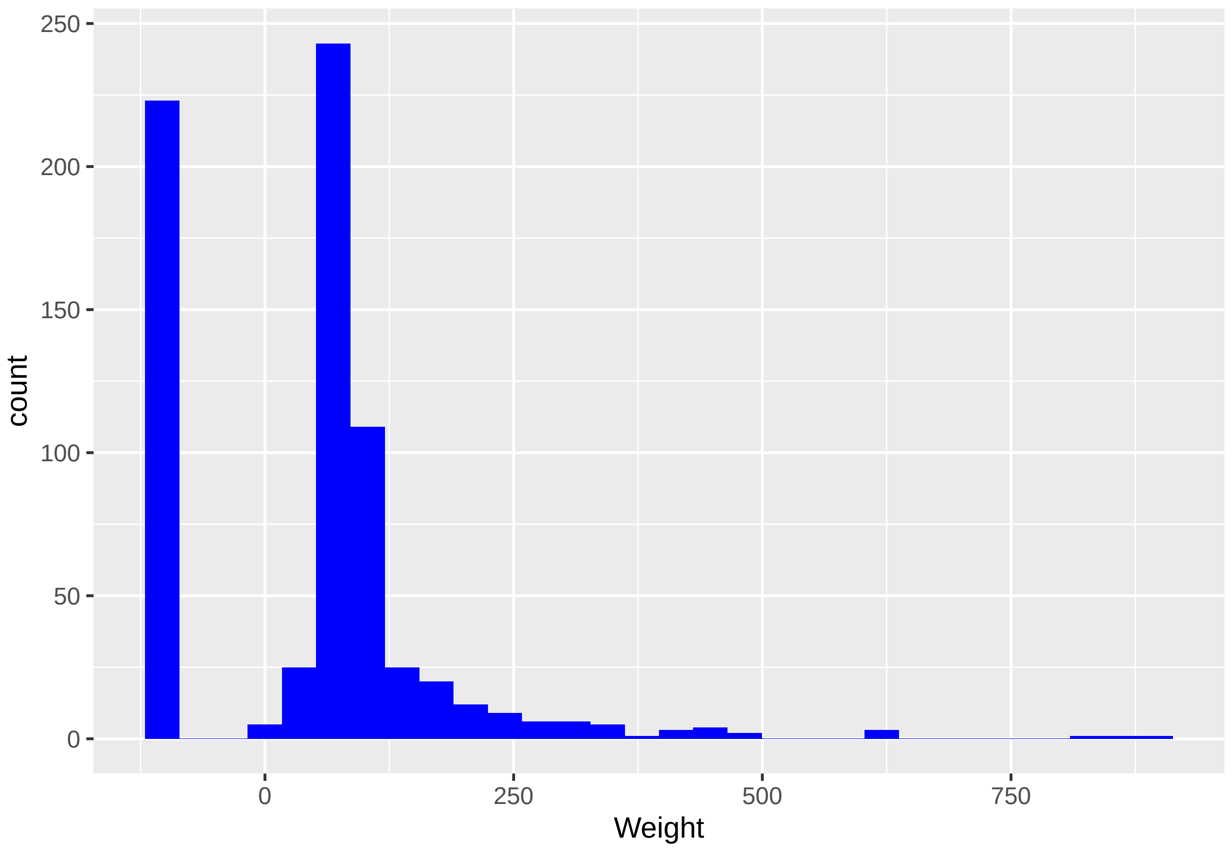


Try fill instead of colour to define the blue bars:

```
ggplot(heroes, aes(Weight)) +
  geom_histogram(fill="Blue")
```

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

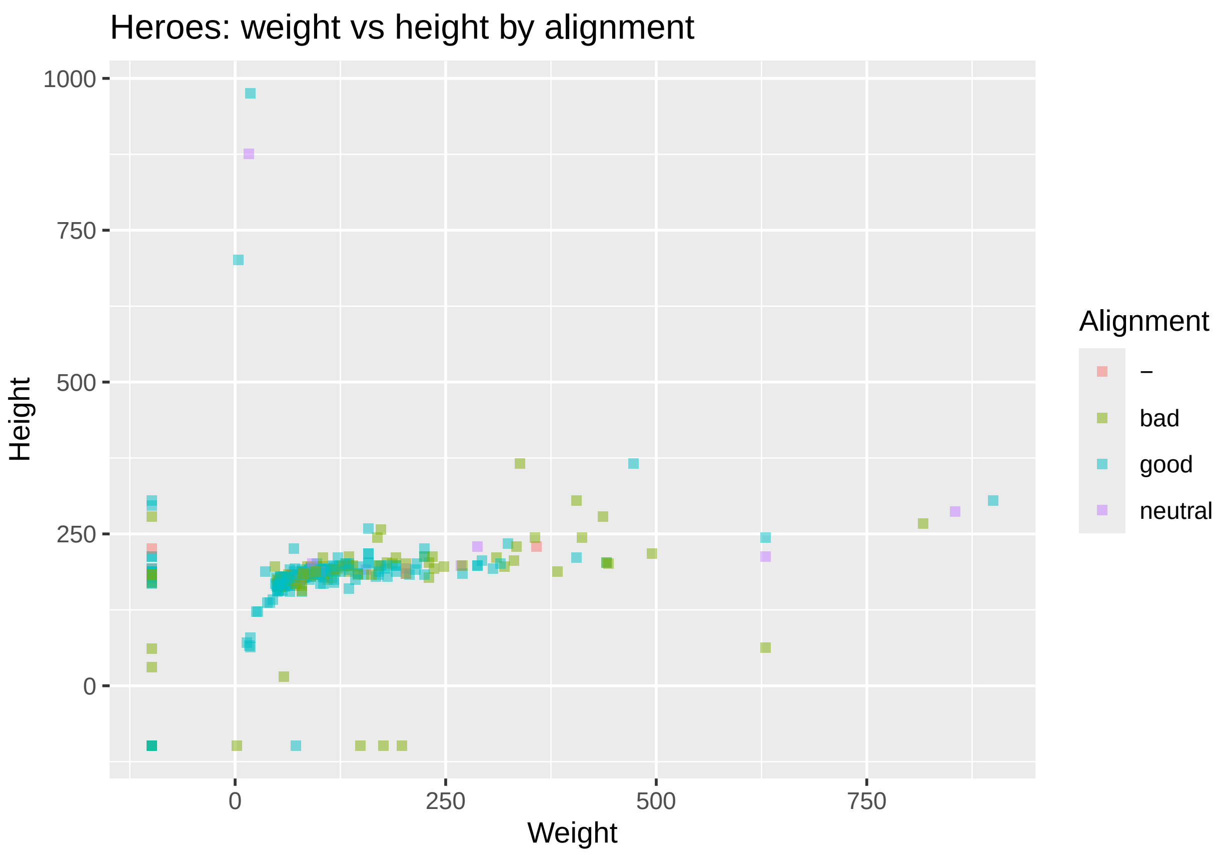
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_bin()`).
```



Going back to our x y plot:

```r
ggplot(heroes) +
  geom_point(aes(Weight,  Height, colour= Alignment), shape = "square", alpha=0.5) +
  ggtitle ("Heroes: weight vs height by alignment")
```

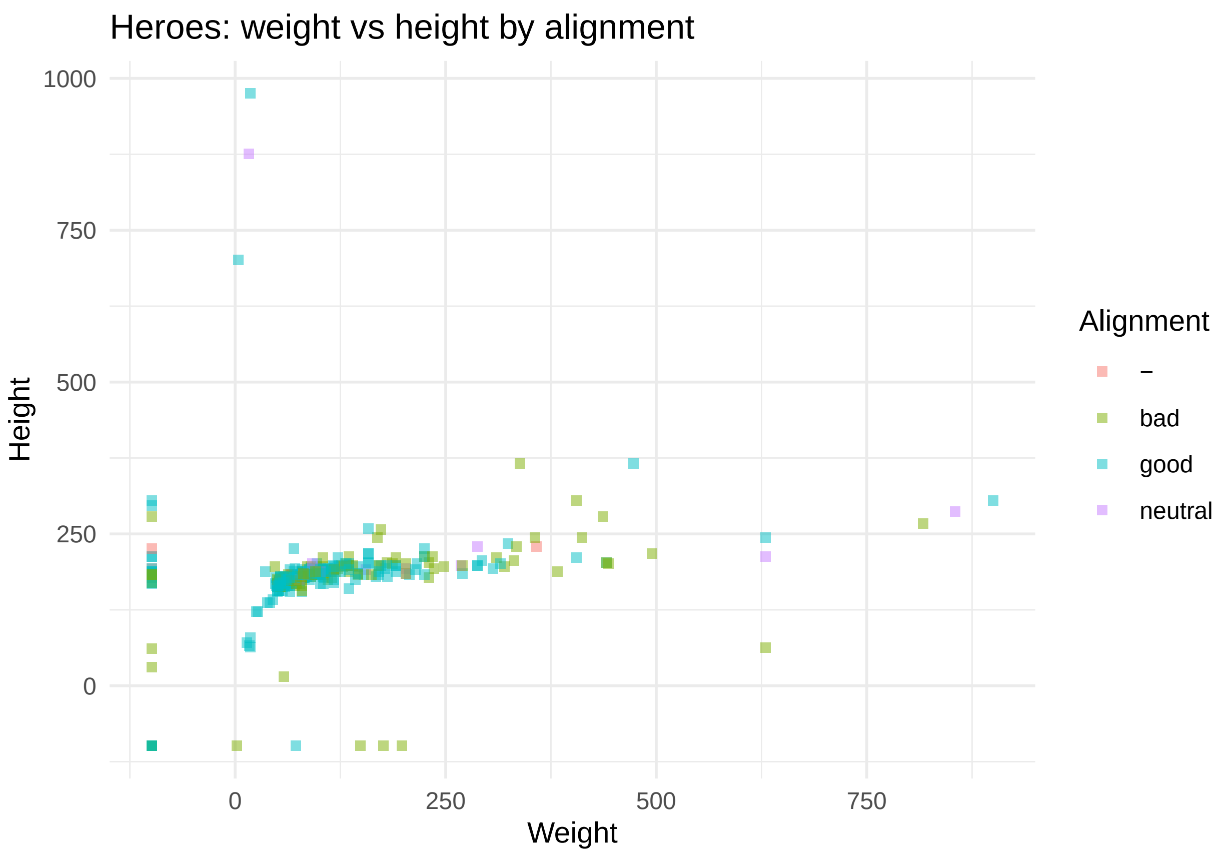
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



Let´s add a theme:

```r
ggplot(heroes) +
  geom_point(aes(Weight,  Height, colour= Alignment), shape = "square", alpha=0.5) +
  ggtitle ("Heroes: weight vs height by alignment") +
  theme_minimal()
```

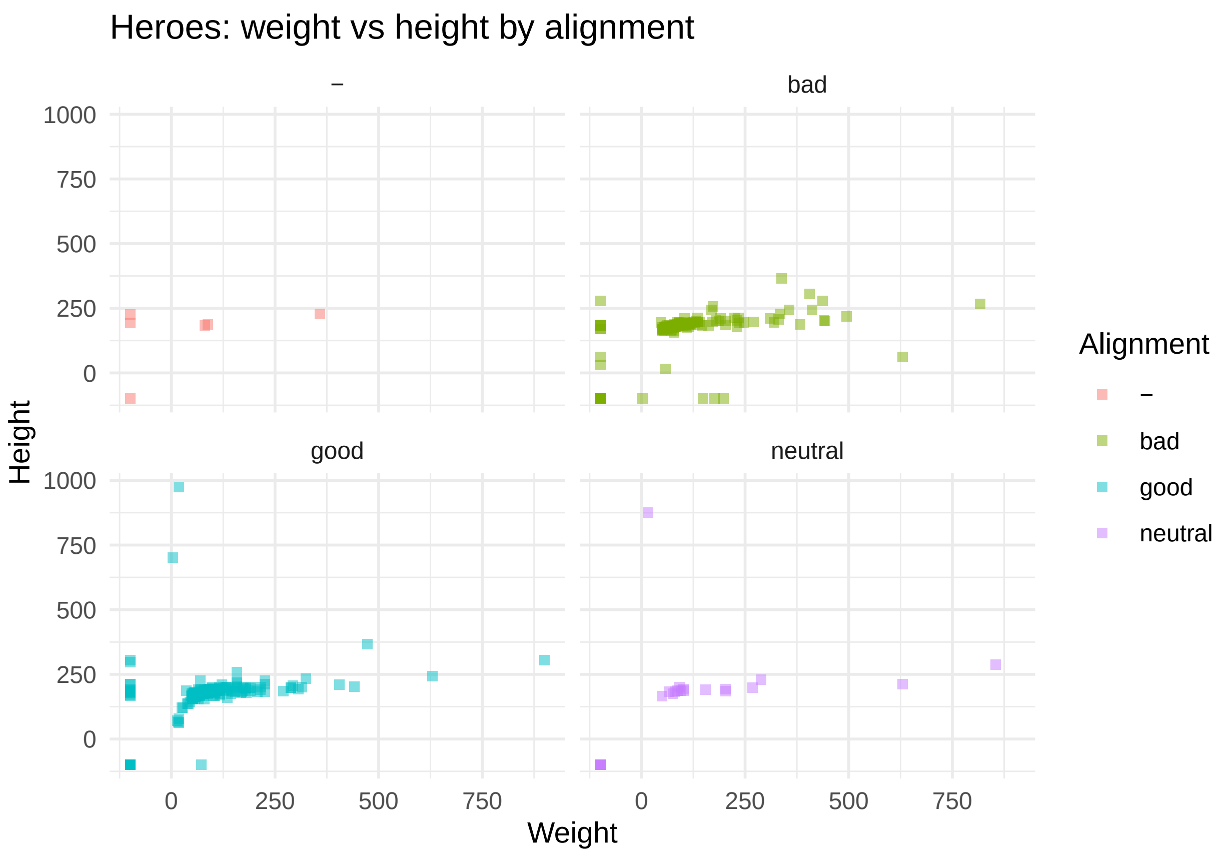
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



Now we will split the data using facets. First let´s try facet_wrap and split the data by alignment.

```
ggplot(heroes) +
  geom_point(aes(Weight,  Height, colour= Alignment), shape = "square", alpha=0.5) +
  ggtitle ("Heroes: weight vs height by alignment") +
  theme_minimal() +
  facet_wrap(~Alignment)
```

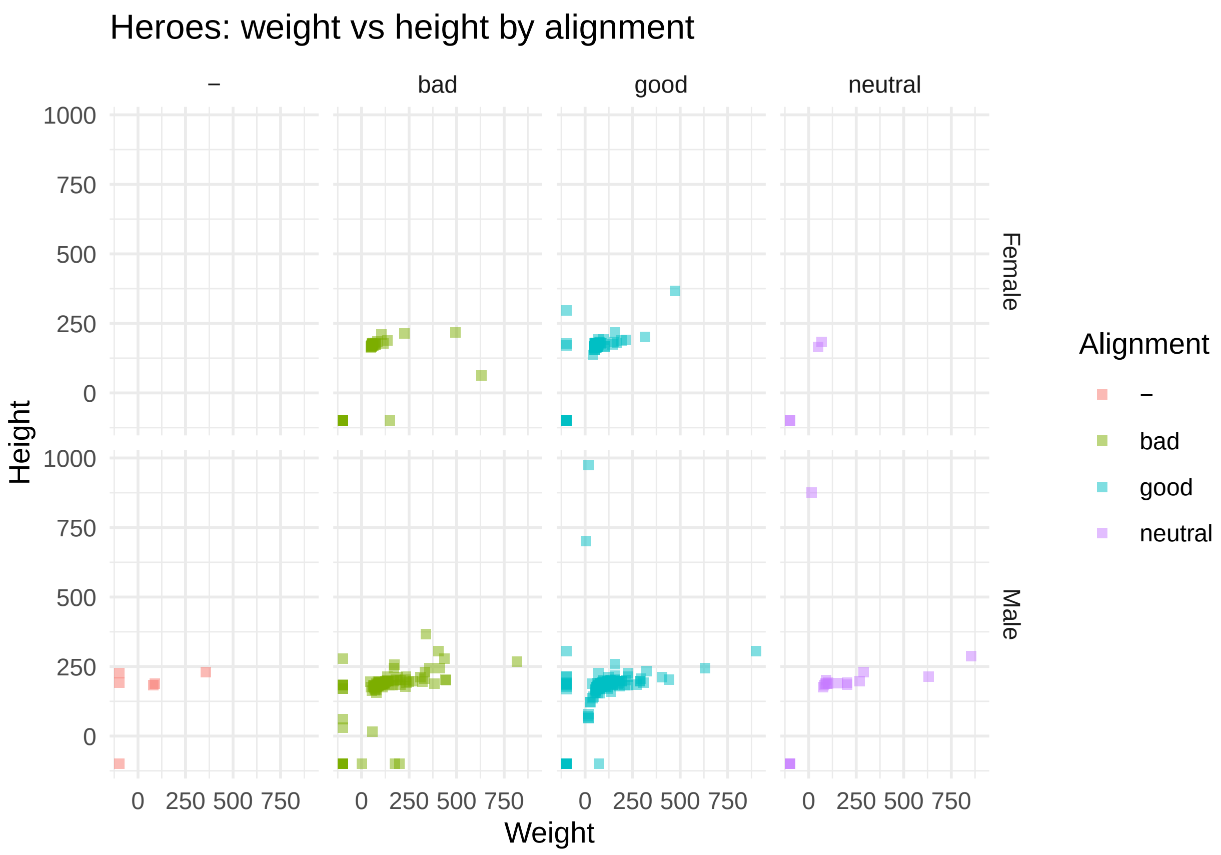
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



Let´s try facet_grid using Gender and Alignment to split the data.

```
ggplot(heroes) +
  geom_point(aes(Weight,  Height, colour= Alignment), shape = "square", alpha=0.5) +
  ggtitle ("Heroes: weight vs height by alignment") +
  theme_minimal() +
  facet_grid(Gender~Alignment)
```

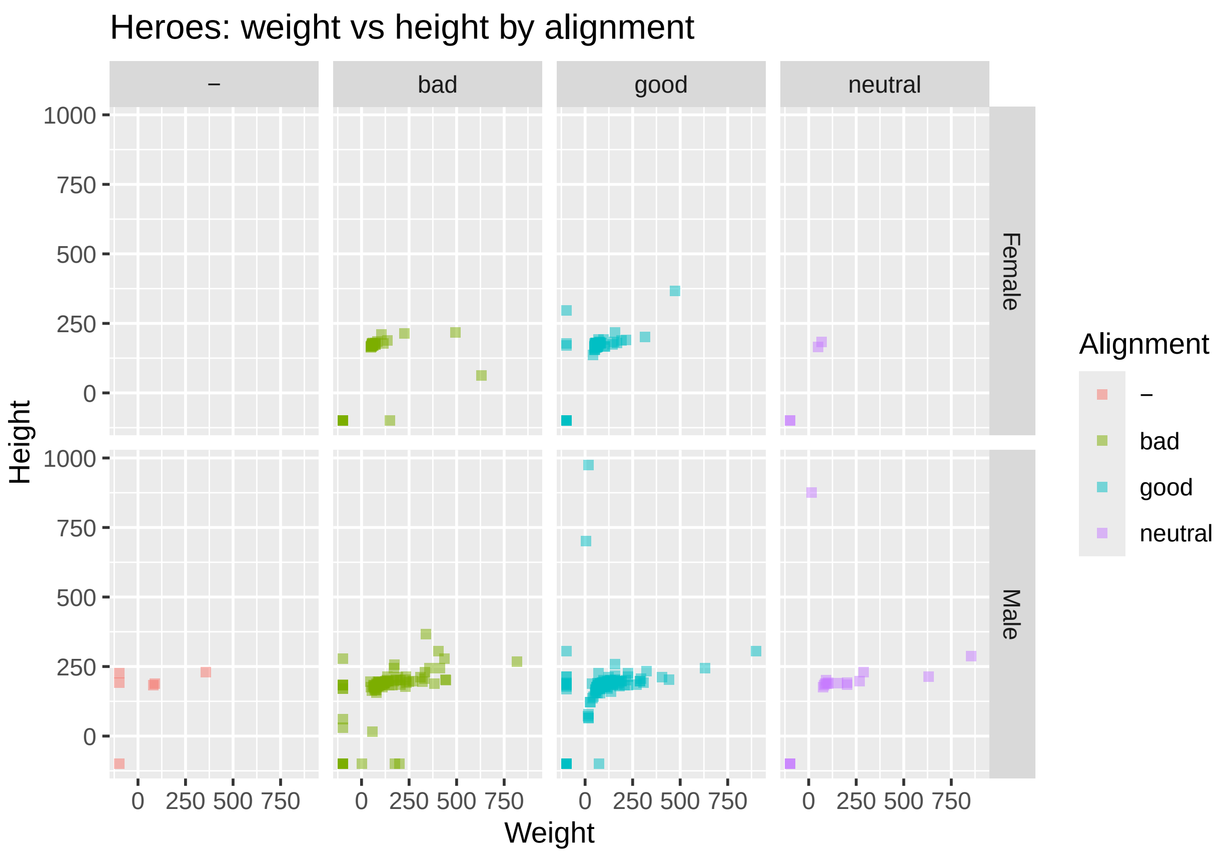
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



We would probably visualise the data better with a darker canvas

```
ggplot(heroes) +
  geom_point(aes(Weight,  Height, colour= Alignment), shape = "square", alpha=0.5) +
  ggtitle ("Heroes: weight vs height by alignment") +
  facet_grid(Gender~Alignment)
```

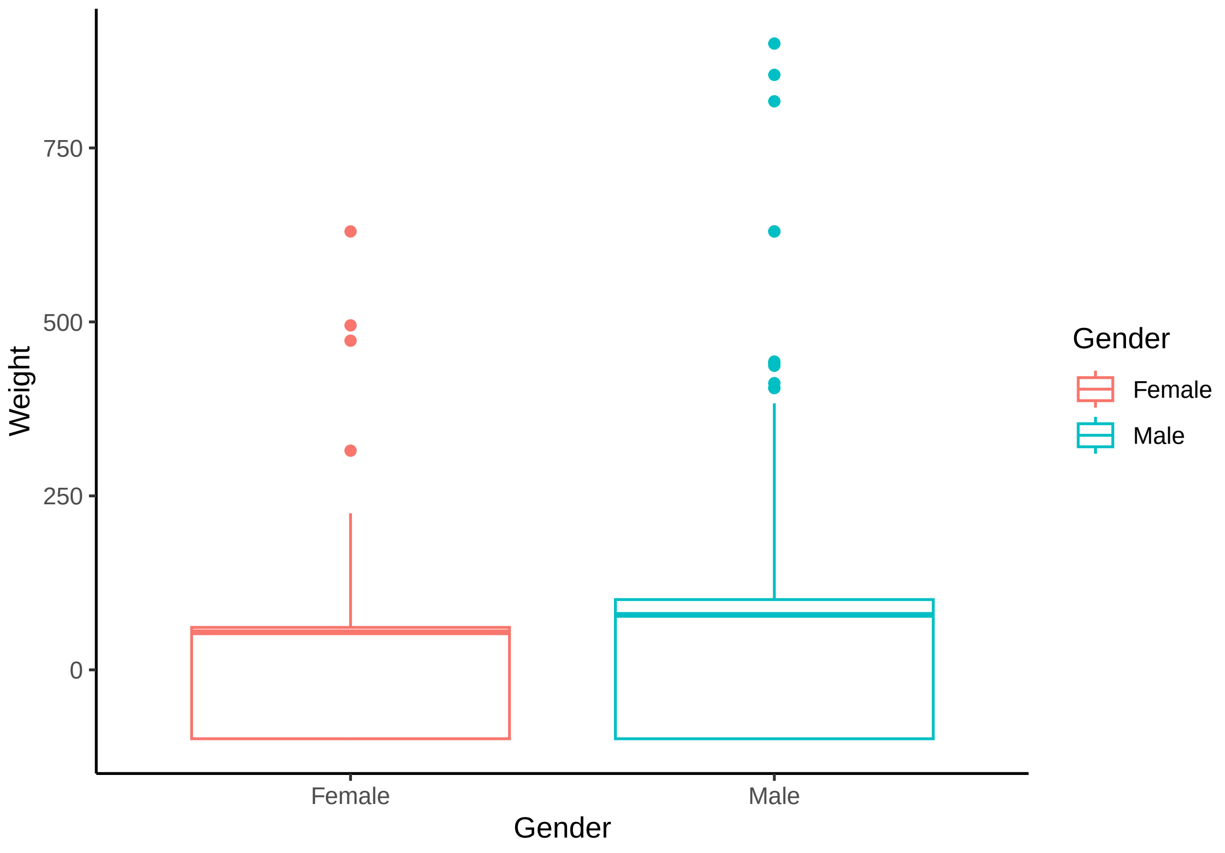
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



Now we try facets on the previous boxplot graph

```r
ggplot(data = heroes, mapping = aes(Gender, Weight)) +
  geom_boxplot(aes(colour=Gender)) +
  theme_classic()
```

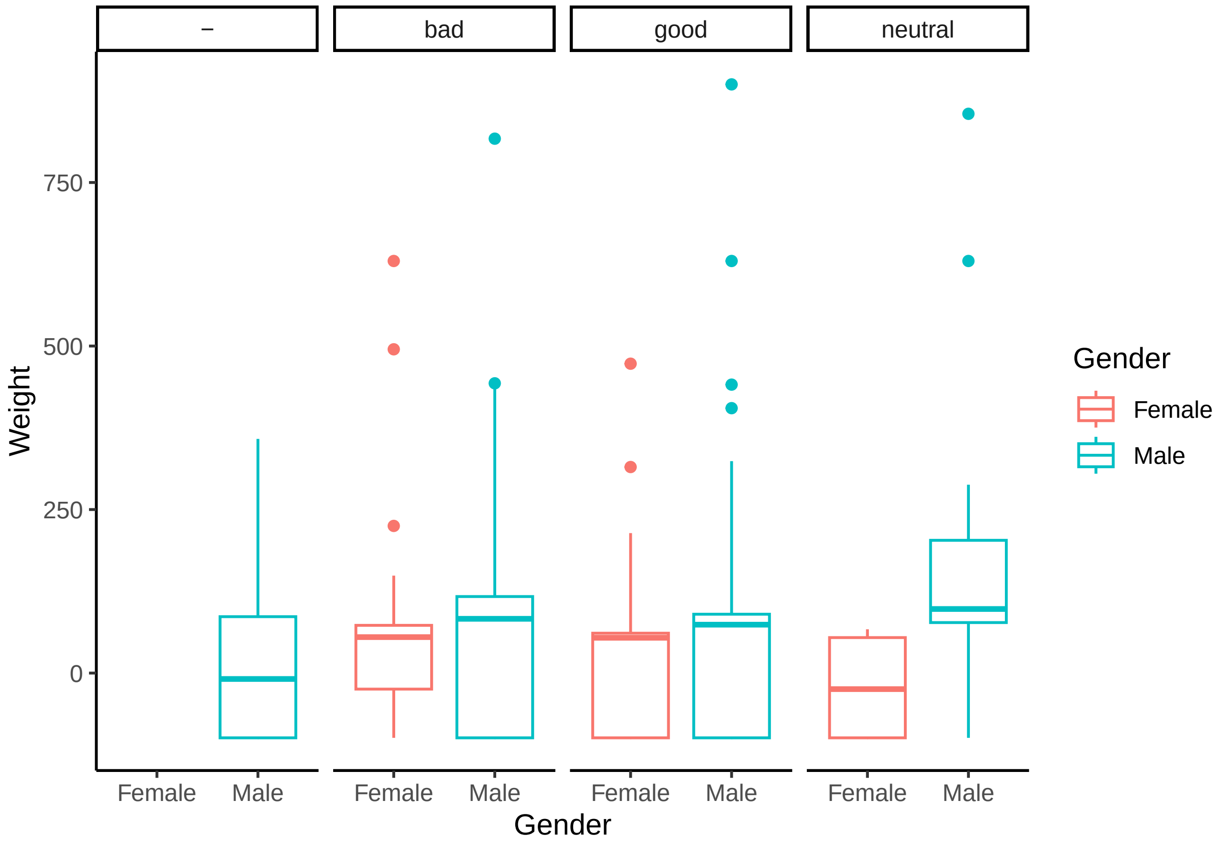
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Again let´s split data by alignment

```
ggplot(data = heroes, mapping = aes(Gender, Weight)) +
  geom_boxplot(aes(colour=Gender)) +
  theme_classic() +
  facet_grid(~Alignment)
```

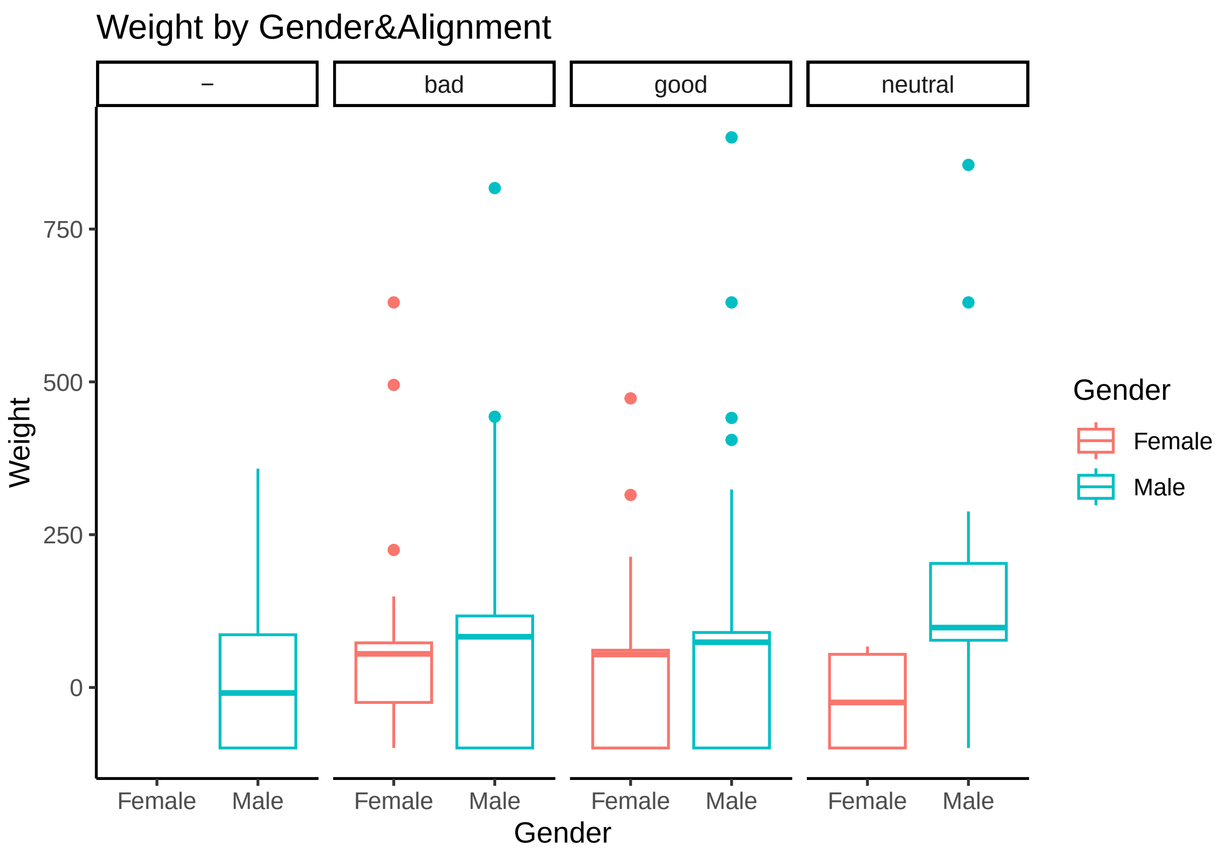
```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



How to export graphs

```
# Store graph in a variable
myplot = ggplot(data = heroes, mapping = aes(Gender, Weight)) +
  geom_boxplot(aes(colour=Gender)) +
  theme_classic() +
  facet_grid(~Alignment) +
  ggtitle ("Weight by Gender&Alignment")

print(myplot) # This way we can print the variable
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```



```r
# Save graph to file (saves last file )
ggsave("myplot.pdf")
```

```
## Saving 6.5 x 4.5 in image

## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```r
ggsave("myplot.png")
```

```
## Saving 6.5 x 4.5 in image

## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```r
# If we want to specify width and height
ggsave("myplot.pdf", width = 10, height = 5)
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```r
# If we want to specify a different file name
ggsave("hero_plot.pdf",
       plot = myplot, width = 10, height = 5)
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```r
# Saving graph without ggsave()
pdf("myplot.pdf")
print(myplot)
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_boxplot()`).
```

```r
dev.off()
```

```
## pdf
##   2
```

```r
# From RScript file we can export directly from the outcome window (go to RScript)
```