

# Statistics & Probability Final Project

Carolina Montoya Bran

2023-10-19

## Establish the data

```
my_data <- data.frame(  
  age = c(20, 22, 23, 25, 26, 20, 26, 23, 20, 30, 28, 25, 27, 25, 28),  
  weight = c(80, 75, 90, 85, 80, 70, 60, 55, 60, 60, 70, 75, 85, 90, 85),  
  height = c(1.70, 1.60, 1.75, 1.65, 1.80, 1.90, 1.85, 1.95, 1.75, 1.80, 1.90, 1.85, 1.75, 1.90, 1.85),  
  size = c("M", "S", "L", "XL", "XL", "L", "L", "XL", "M", "XL", "M", "XL", "XL", "L", "M")  
)
```

## Calculate average weight per height using ‘tapply’

```
weight_average_persize <- tapply(my_data$weight, my_data$size, FUN = mean)
```

```
final_average <- weight_average_persize[my_data$size]
```

## Add the averages column to the dataframe

```
my_data$average <- final_average
```

## Check the data

```
print(my_data)
```

```
##      age weight height size  average  
## 1    20     80   1.70    M 73.75000  
## 2    22     75   1.60    S 75.00000  
## 3    23     90   1.75    L 77.50000  
## 4    25     85   1.65   XL 73.33333  
## 5    26     80   1.80   XL 73.33333  
## 6    20     70   1.90    L 77.50000  
## 7    26     60   1.85    L 77.50000  
## 8    23     55   1.95   XL 73.33333  
## 9    20     60   1.75    M 73.75000  
## 10   30     60   1.80   XL 73.33333  
## 11   28     70   1.90    M 73.75000  
## 12   25     75   1.85   XL 73.33333  
## 13   27     85   1.75   XL 73.33333  
## 14   25     90   1.90    L 77.50000
```

```
## 15 28 85 1.85 M 73.75000
```

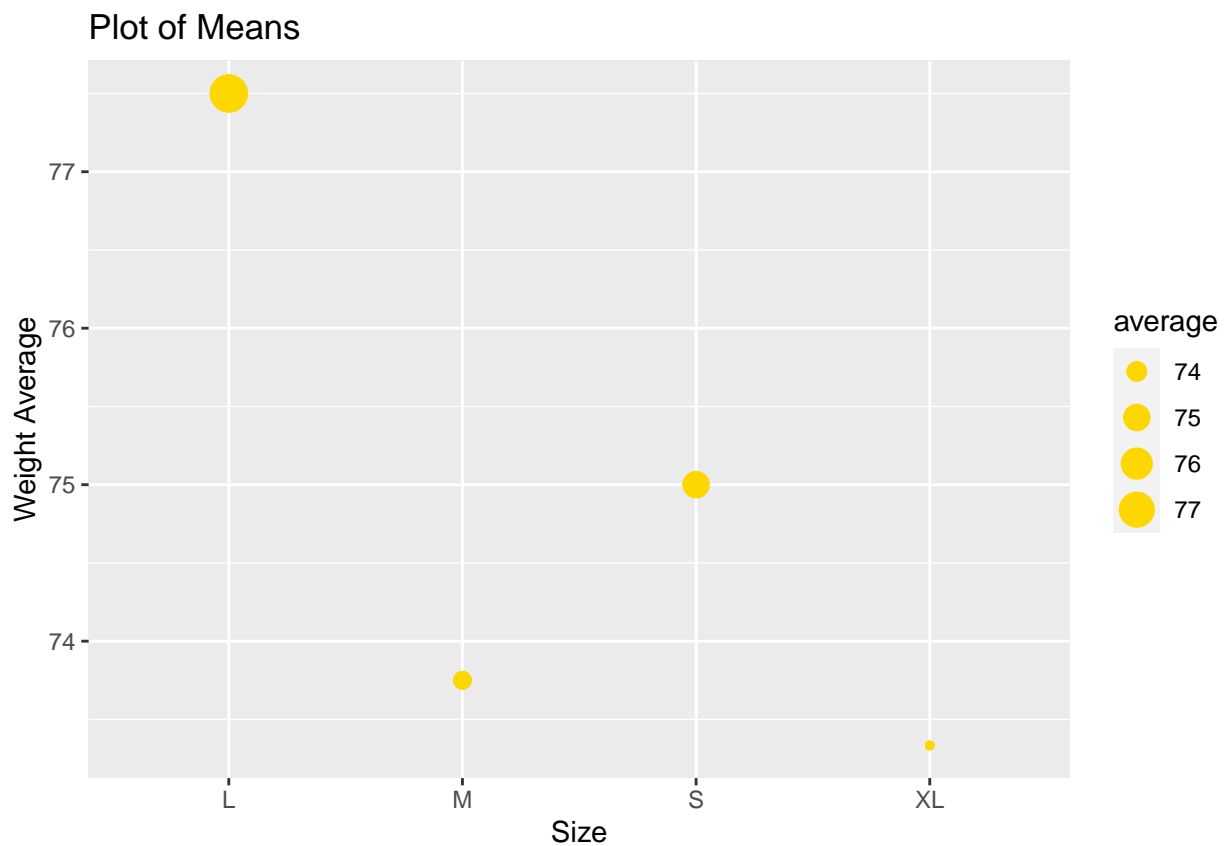
```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'  
## (as 'lib' is unspecified)
```

```
library(ggplot2)
```

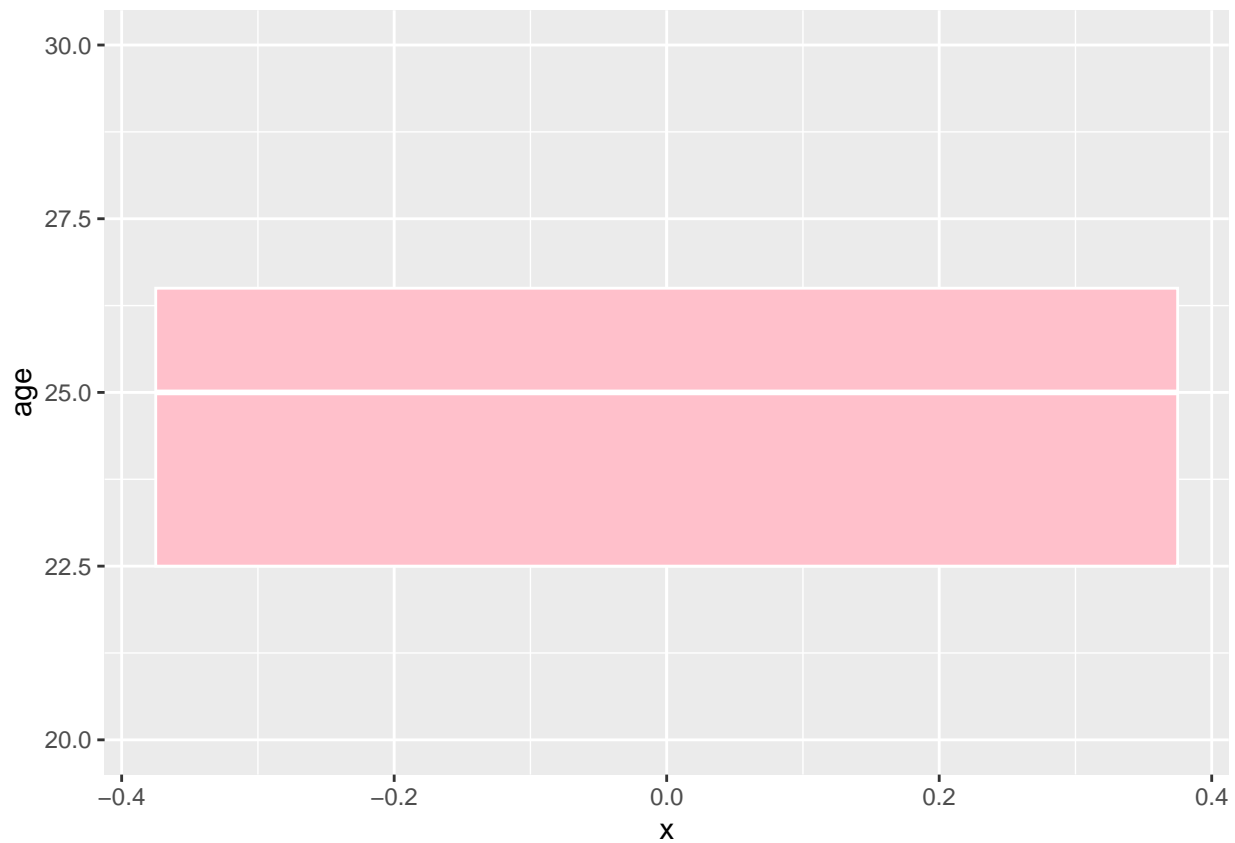
## Graphic 1

```
ggplot(data = my_data, aes(x= size, y= average, size = average)) +  
  geom_point(color= "gold") +  
  labs (x ="Size", y="Weight Average", title = "Plot of Means")
```



## Graphic 2

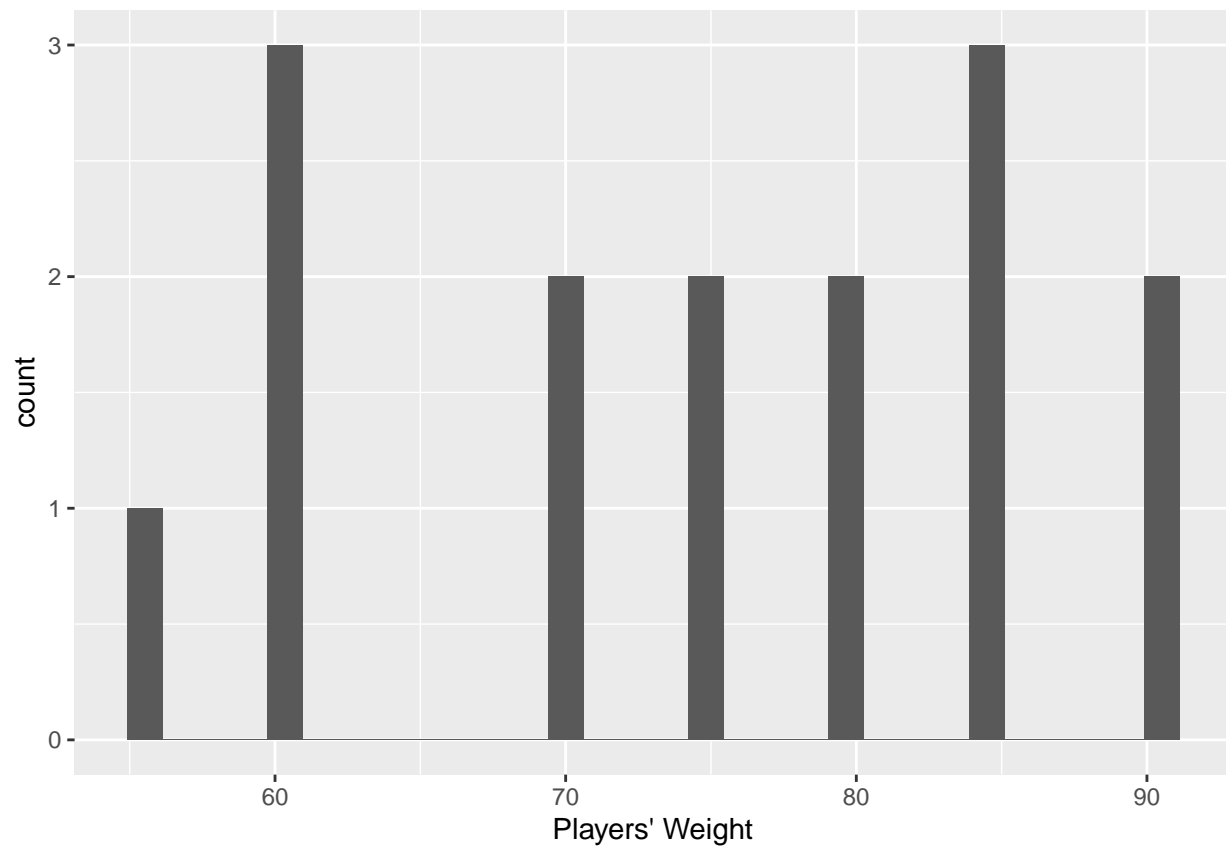
```
ggplot(data = my_data) + geom_boxplot(mapping = aes(y= age, x= NULL), color = "white", fill = "pink")
```



## Graphic 3

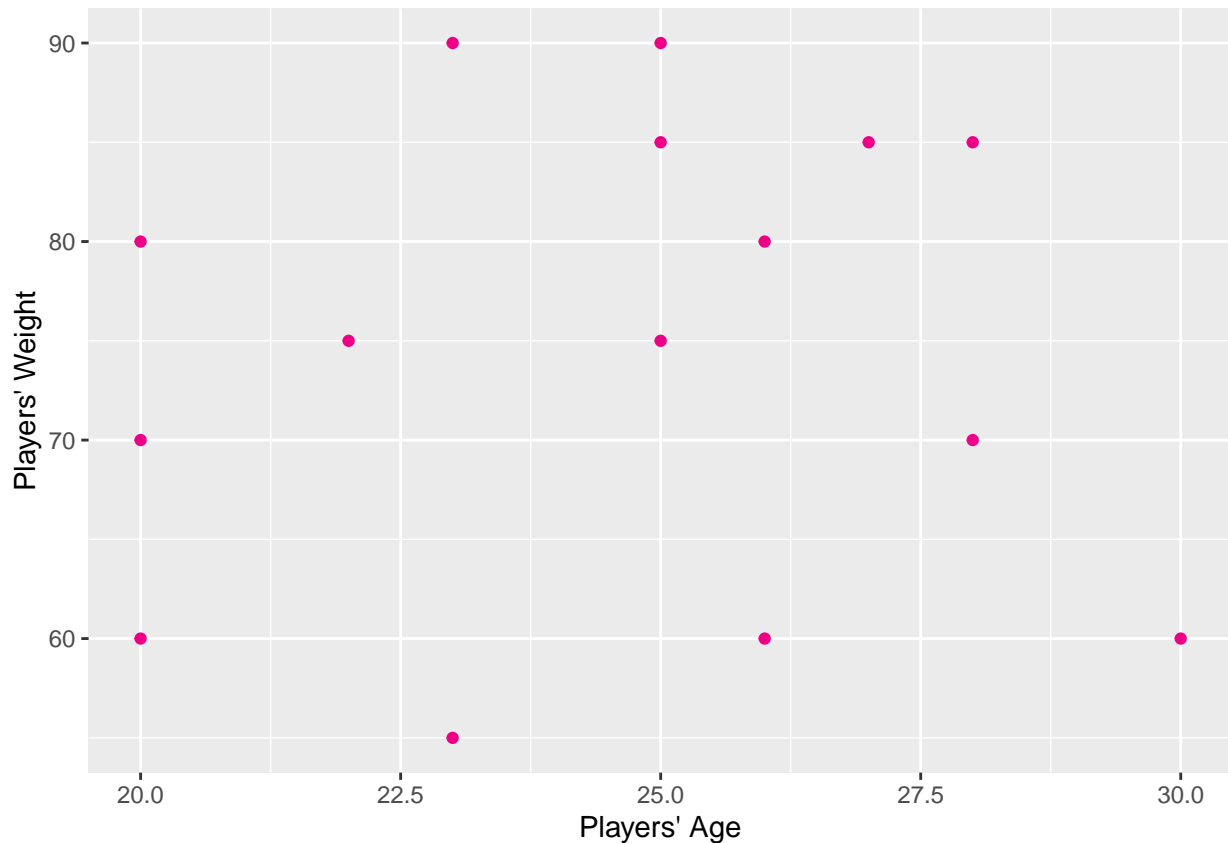
```
ggplot(data = my_data, aes(x = weight)) +  
  geom_histogram(mapping = aes(y = ..count..)) +  
  labs(x = "Players' Weight")
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.  
## i Please use `after_stat(count)` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was  
## generated.  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Graphic 4

```
ggplot(data = my_data, aes(x = age, y = weight)) +  
  geom_point(color = "#ed0086") +  
  labs(x = "Players' Age", y = "Players' Weight")
```



## Graphic 5

Define the value of variables using the 'rnorm' function

```
the_average <- 300
standard_deviation <- 15
data1 <- rnorm(n = 1000, mean = the_average, sd = standard_deviation)
```

Create a dataframe

```
df <- data.frame(values = data1)
```

```
ggplot(df, aes(x = values)) +
  geom_density(color = "blue", fill = "lightblue")+
  labs(x = "x", y = "Density")+
  ggtitle("Normal Distribution: Mean = 300, Standard deviation = 15")
```

Normal Distribution: Mean = 300, Standard deviation = 15

