Chapter 4: Data Viz using ggplot2

Dr Megan L. Wood

2022-12-01

Chapter Overview:

1. **Creating our first plot**
2. **APA formatting**
3. **Changing aesthetics**
4. **Plotting by groups**
5. **Faceting**
6. **Plotting counts**
7. **Plotting means**

> "The simple graph has brought more information to the data analyst's mind than any other device." — John Tukey

There are hundreds of ways we can visualise our data in R, but we will be focussing on ggplot2 which is part of the **tidyverse**. It is arguably the most versatile and widely-used.

There are so many ways we can build on the basic ggplot template to create beautiful plots with customisation and complexity.

The type of plot we produce should depend upon the data we have available and the message we are trying to convey.

Remember that oftentimes, simplicity is key. Therefore, the humble bar chart may actually be the most appropriate choice.

We are going to use our ckat data here so make sure you have that loaded into your environment.

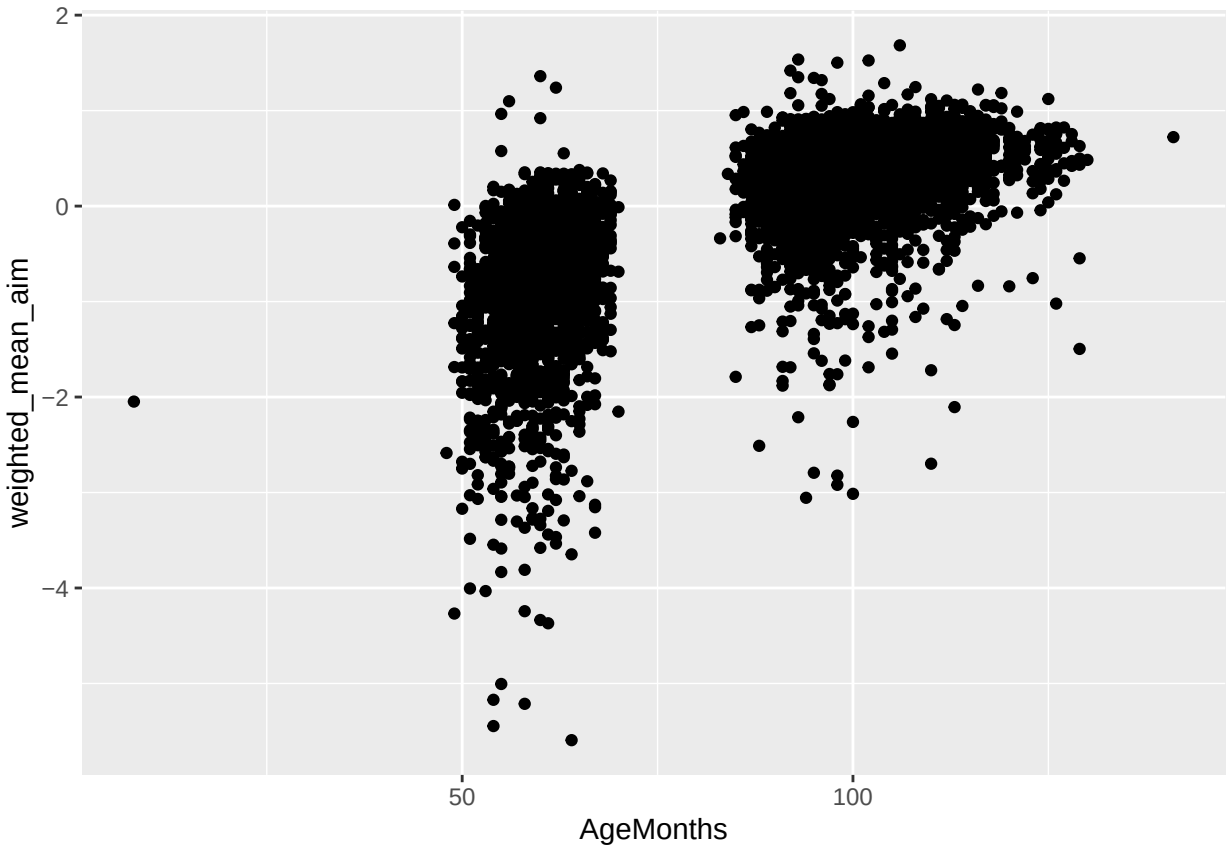
Creating our first plot

Let's start with one of the most simple plots - the scatter plot.

Scatter plots can help us look at the relationships between two continuous variables. Let's give this a go by plotting the relationship between age and aiming score.

```
ggplot(ckat.dat, aes(x = AgeMonths, y = weighted_mean_aim)) + # define the data and variables
  geom_point() # tell R in what format we want the data plotted
```

```
## Warning: Removed 1312 rows containing missing values (geom_point).
```



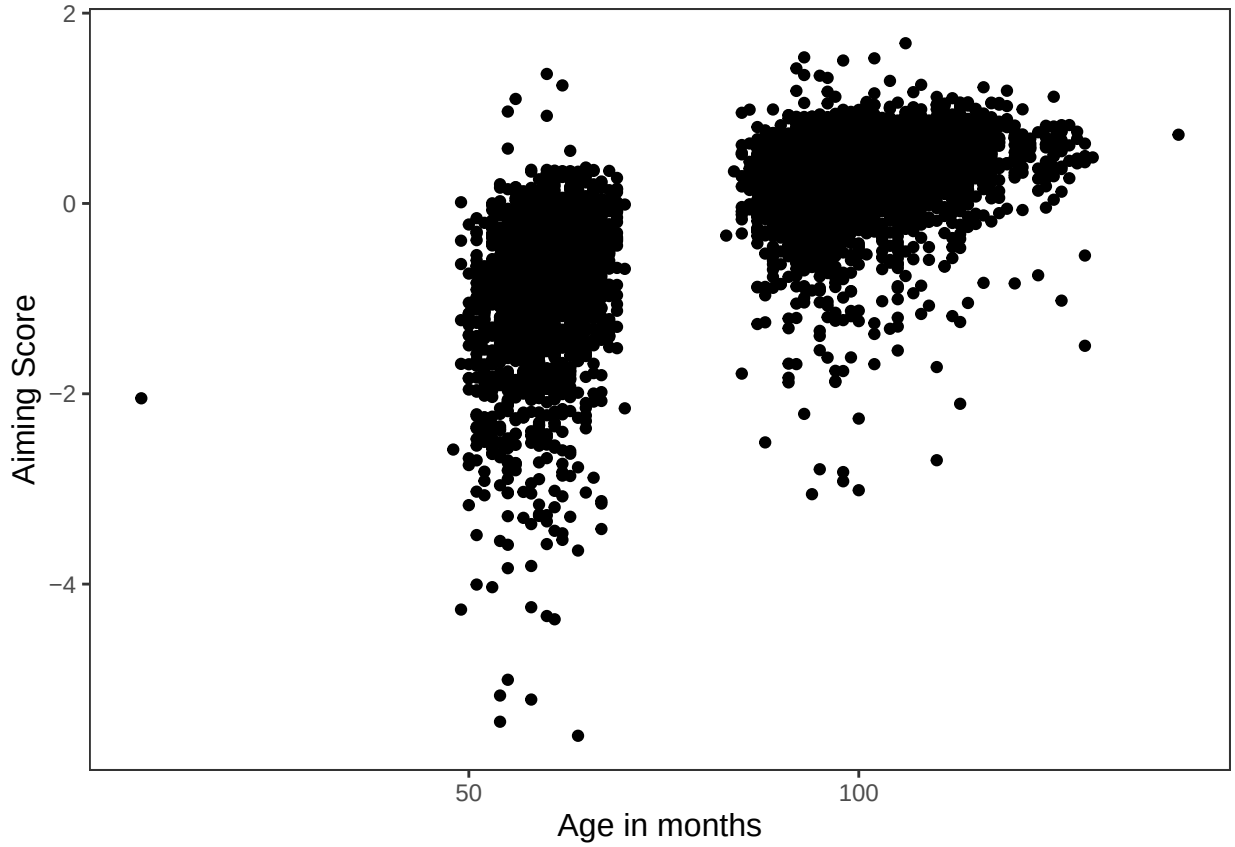
APA formatting with themes

If we are producing figures for publication, we will often need to ensure they align with APA formatting standards. This includes no background colour, formatted axes etc. However, instead of doing this by hand, there is a handy function within the package **jtools** which does all this formatting for us, easy!

Let's convert your first scatter plot to APA formatting and add in some appropriate labels for our axes.

```
ggplot(ckat.dat, aes(x = AgeMonths, y = weighted_mean_aim)) + # define the data and variables
  geom_point() +# tell R in what format we want the data plotted
  labs(x = "Age in months", y = "Aiming Score") +
  theme_apa()
```

```
## Warning: Removed 1312 rows containing missing values (geom_point).
```



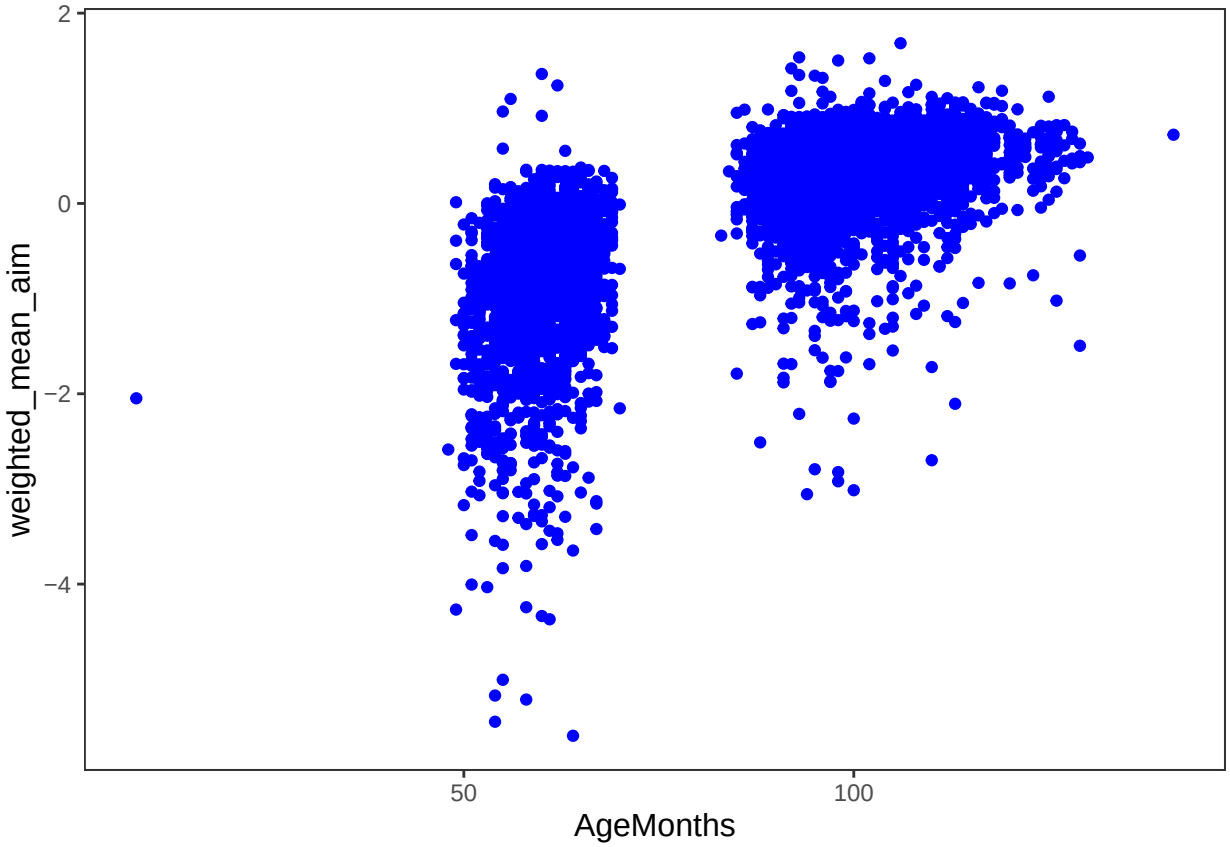
Can you tell the difference?

We can then add to this to customise or make it easier interpret by changing the...

... colour of the geoms

```
ggplot(ckat.dat, aes(x = AgeMonths, y = weighted_mean_aim)) + # define the data and variables
  geom_point(colour = "blue") + # tell R in what format we want the data plotted
  theme_apa()
```

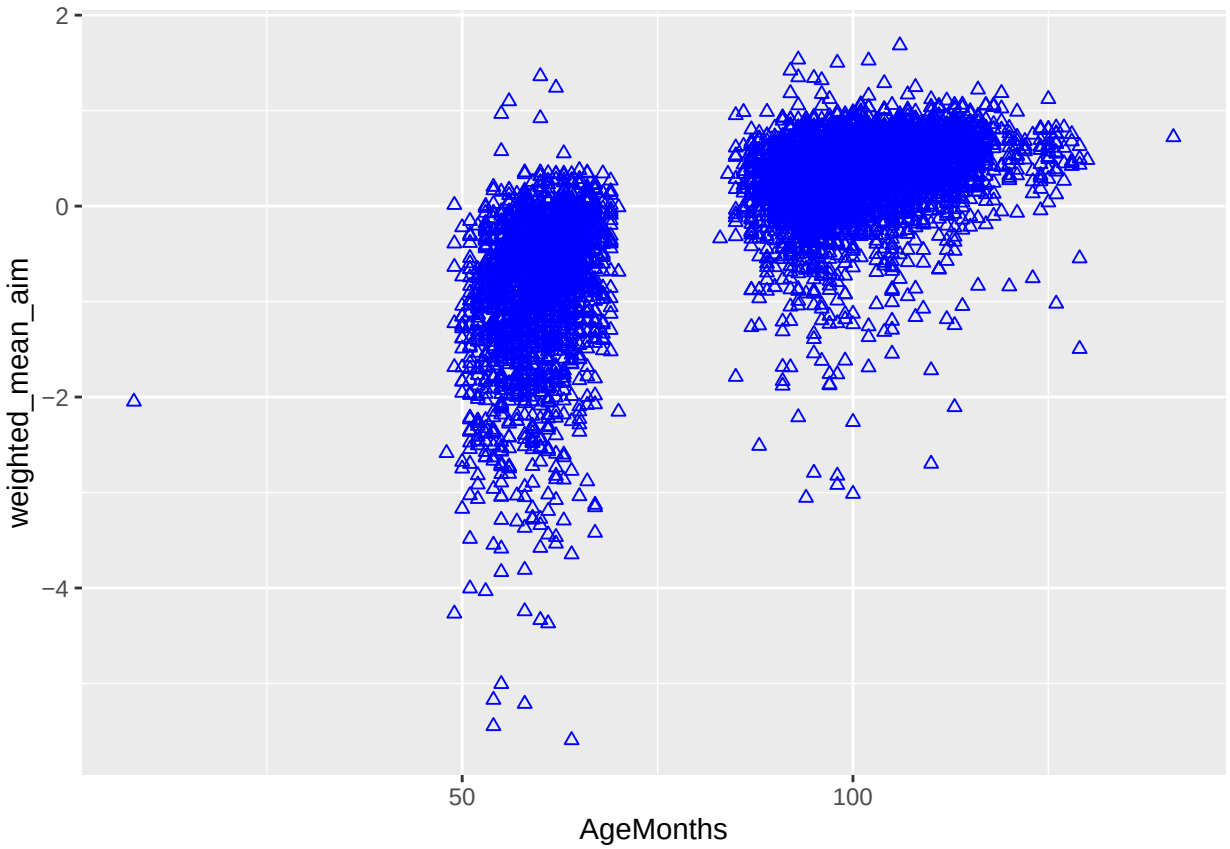
```
## Warning: Removed 1312 rows containing missing values (geom_point).
```



... shape of the geoms

```
ggplot(ckat.dat, aes(x = AgeMonths, y = weighted_mean_aim)) + # define the data and variables
  geom_point(colour = "blue", shape = 2) # tell R in what format we want the data plotted
```

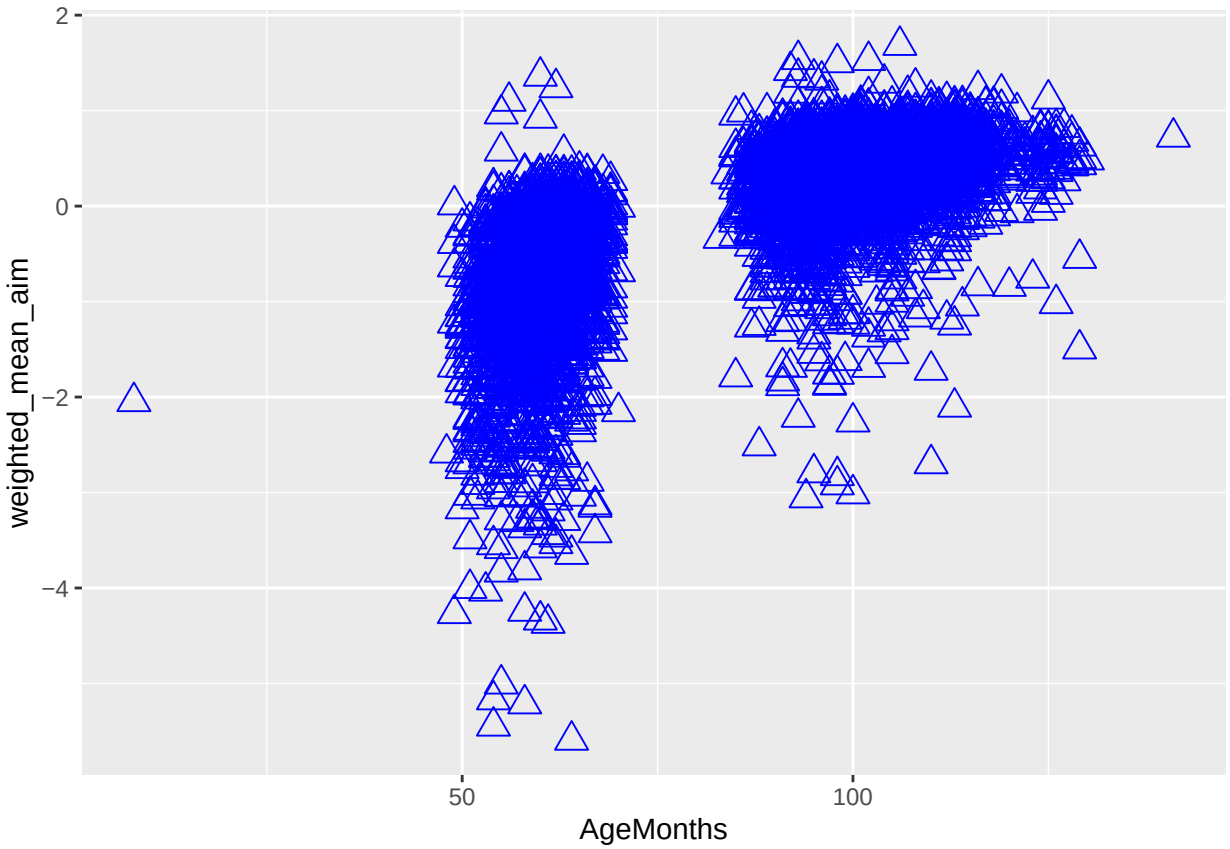
```
## Warning: Removed 1312 rows containing missing values (geom_point).
```



... size of the geoms

```
ggplot(ckat.dat, aes(x = AgeMonths, y = weighted_mean_aim)) + # define the data and variables
  geom_point(colour = "blue", shape = 2, size = 4) # tell R in what format we want the data plotted
```

```
## Warning: Removed 1312 rows containing missing values (geom_point).
```



Have a play with changing the aesthetic properties of the geoms.

Here are some ways your geoms might be customised:

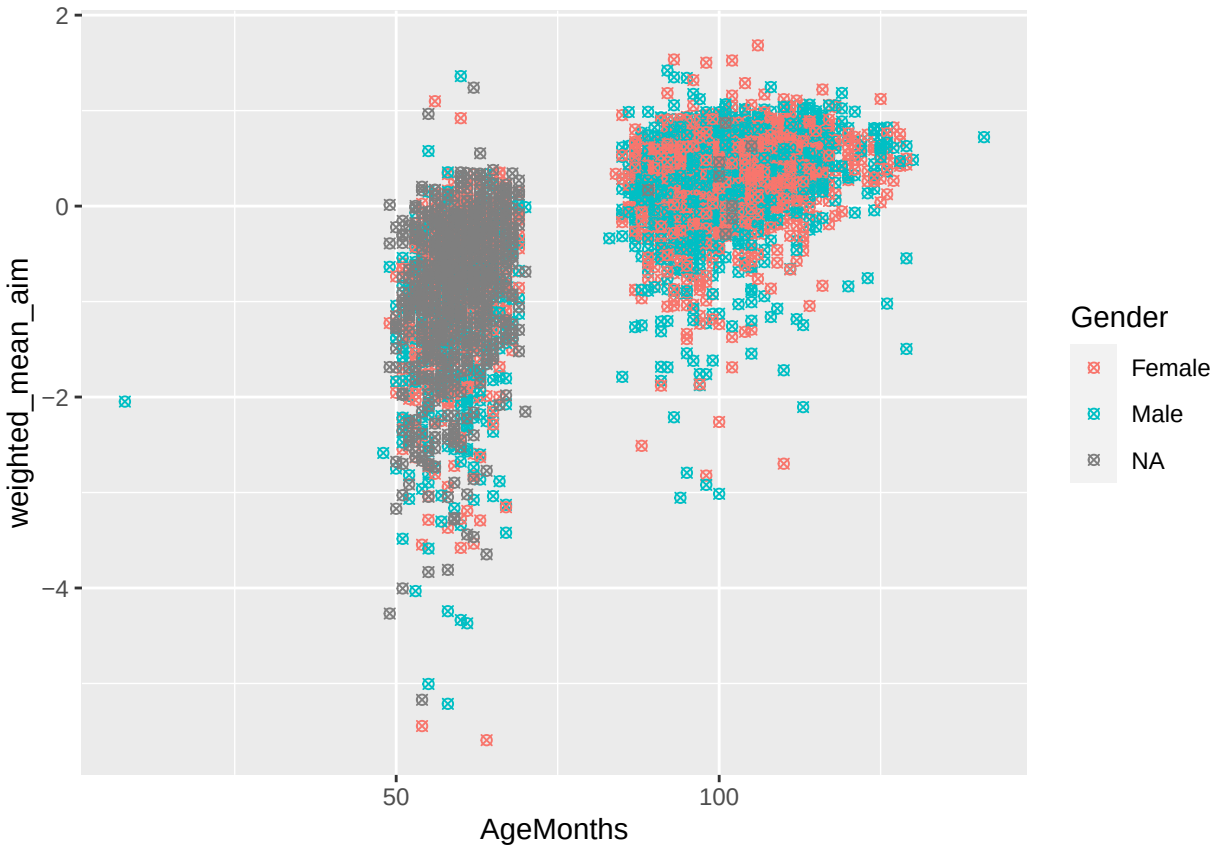
- size
- shape
- colour (outline / fill)
- alpha (transparency)

Comparing sub-groups

We can use these to compare different sub-groups within our data too.

```
ggplot(ckat.dat, aes(x = AgeMonths, y = weighted_mean_aim, colour = Gender)) + # define the data and va
  geom_point(shape = 13) # tell R in what format we want the data plotted
```

```
## Warning: Removed 1312 rows containing missing values (geom_point).
```

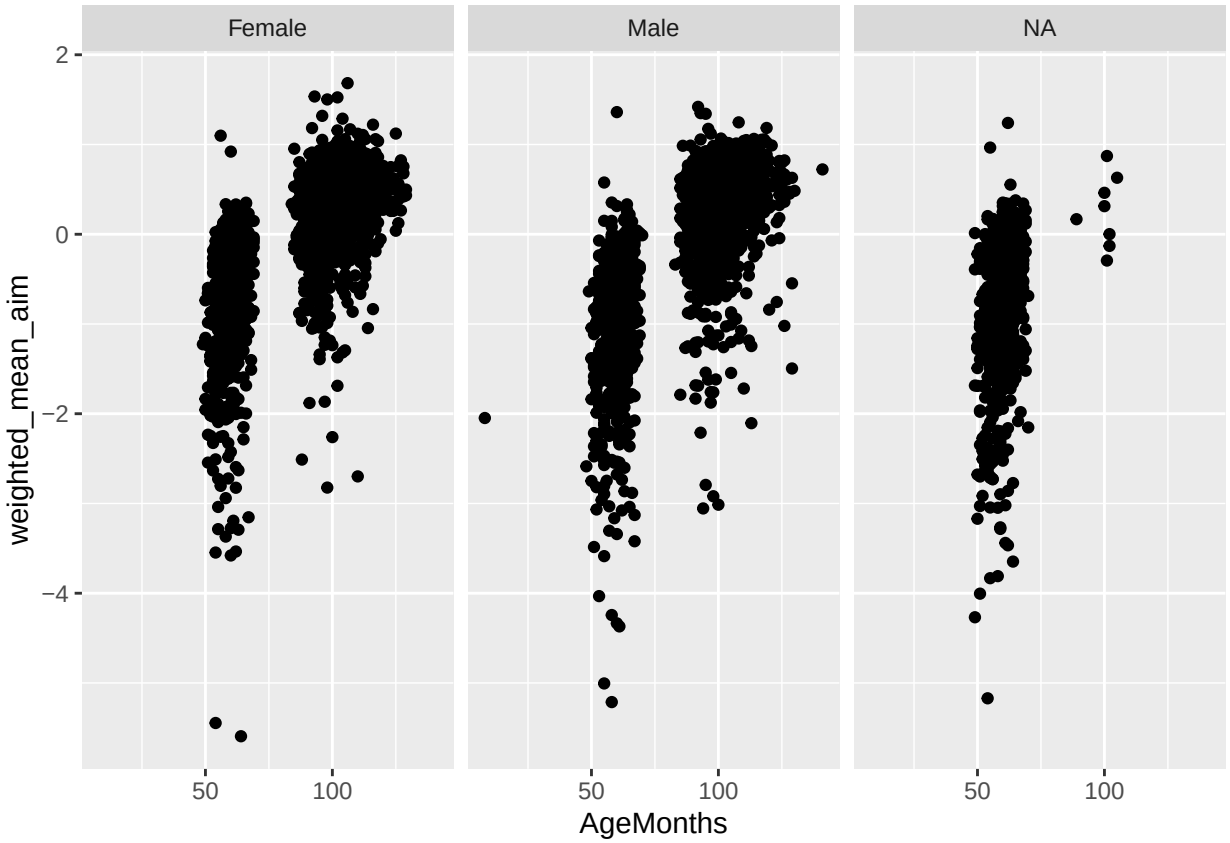


Facets

We can also add facets to produce subplots that each display one subset of the data. This can be particularly useful when we want to compare the same data across different groups.

```
ggplot(ckat.dat, aes(x = AgeMonths, y = weighted_mean_aim)) + # define the data and variables
  geom_point() +  # tell R in what format we want the data plotted
  facet_wrap(~Gender)
```

```
## Warning: Removed 1312 rows containing missing values (geom_point).
```

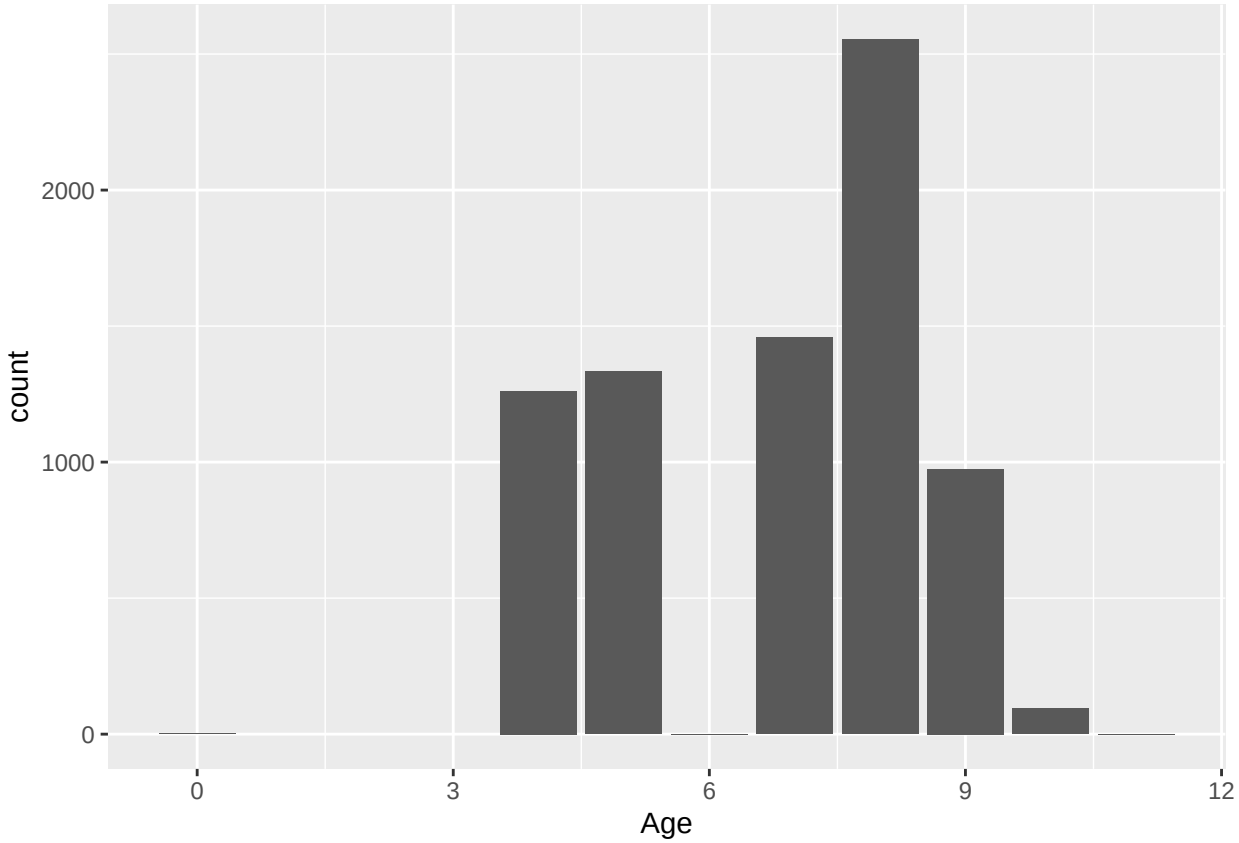


Plotting counts

We can use bar charts to visualise counts of individuals across different groups.

```
ggplot(ckat.dat, aes(x = Age)) +
  geom_bar()
```

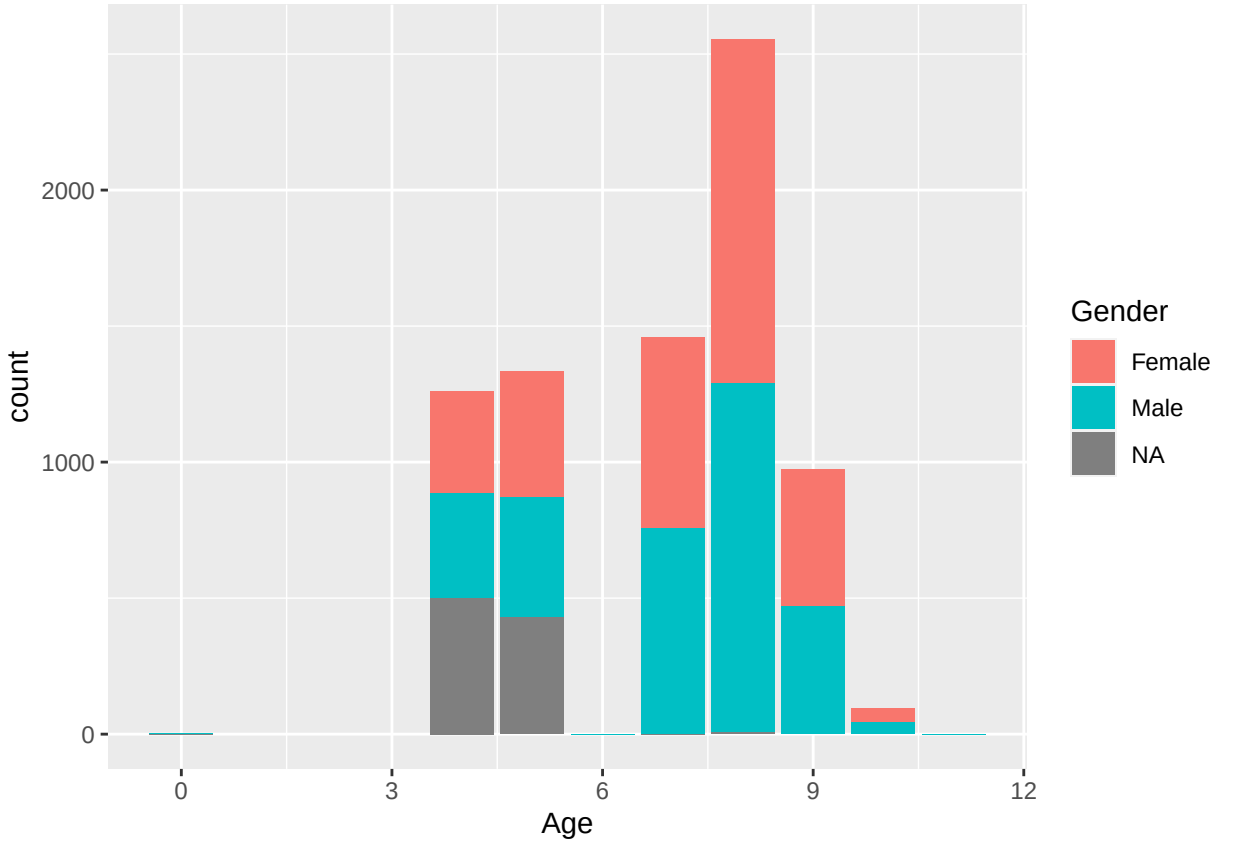
```
## Warning: Removed 20 rows containing non-finite values (stat_count).
```



Counts by groups

```
ggplot(ckat.dat, aes(x = Age, fill = Gender)) + geom_bar()
```

```
## Warning: Removed 20 rows containing non-finite values (stat_count).
```

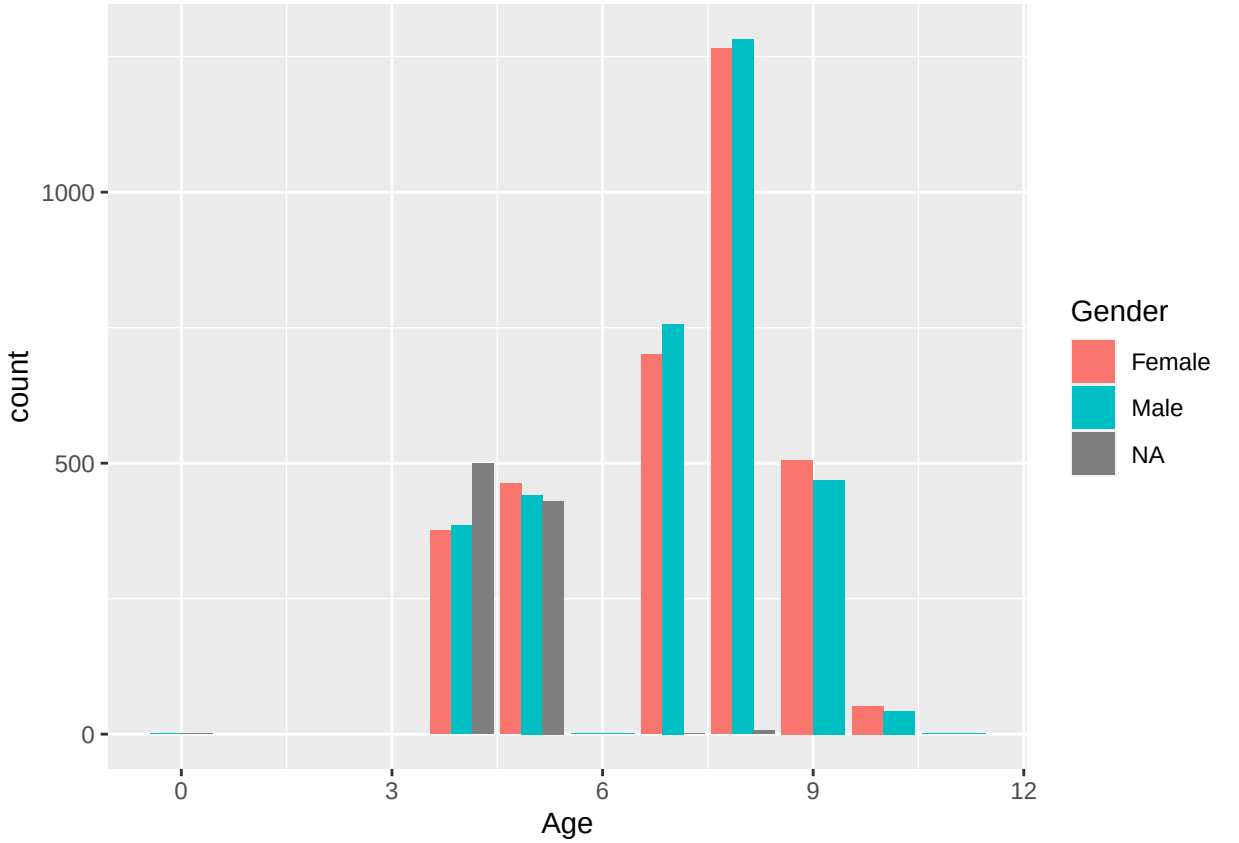


Stacked bar charts aren't always very useful for looking at exactly how many people there are. For example, this plot makes it difficult to compare the number of 8 year old females to 4 year old males.

Introducing the dodge

```
ggplot(ckat.dat, aes(x = Age, fill = Gender)) + geom_bar(position = position_dodge())
```

```
## Warning: Removed 20 rows containing non-finite values (stat_count).
```

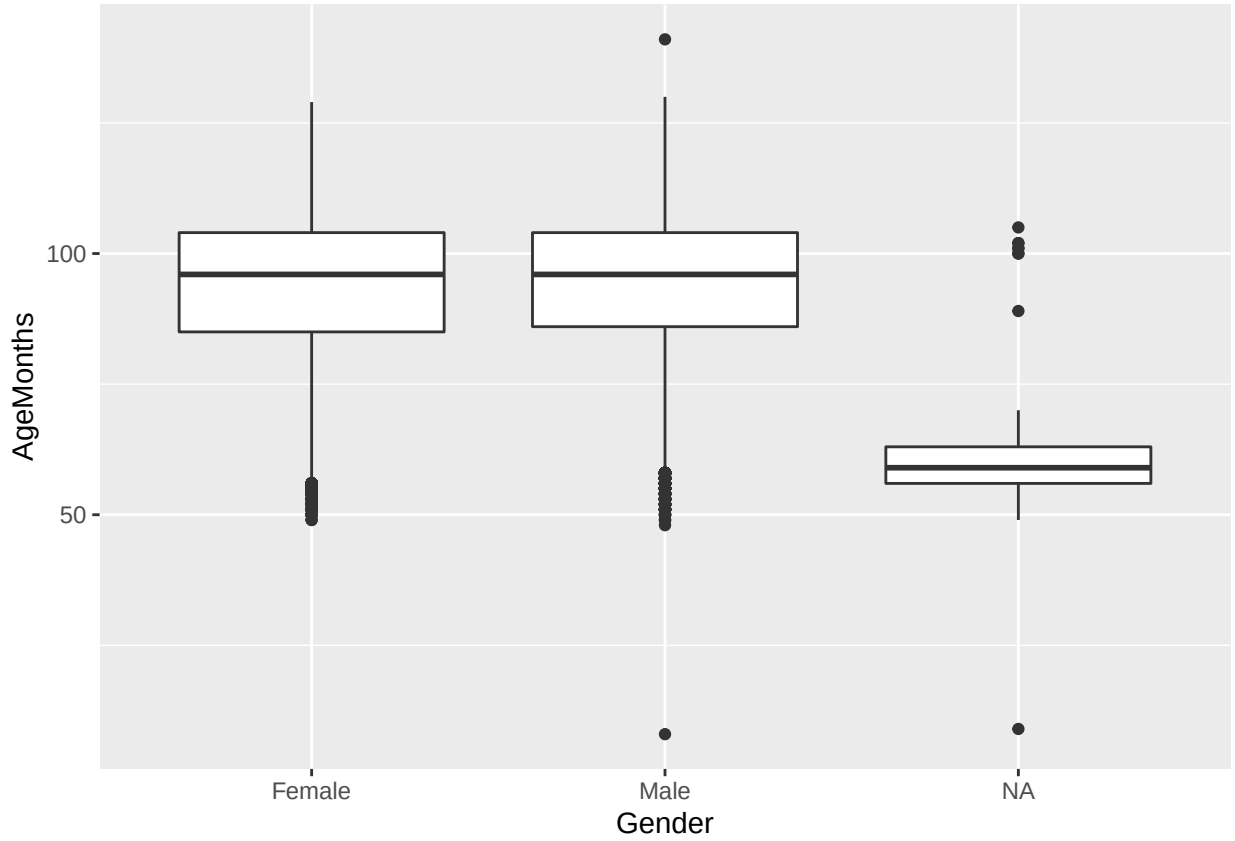


Plotting means

A bar plot is not the most appropriate to visualise a mean, however. We cannot tell how dispersed the data may be, for example.

```
ggplot(ckat.dat, aes(x = Gender, y = AgeMonths)) +
  geom_boxplot()
```

```
## Warning: Removed 20 rows containing non-finite values (stat_boxplot).
```

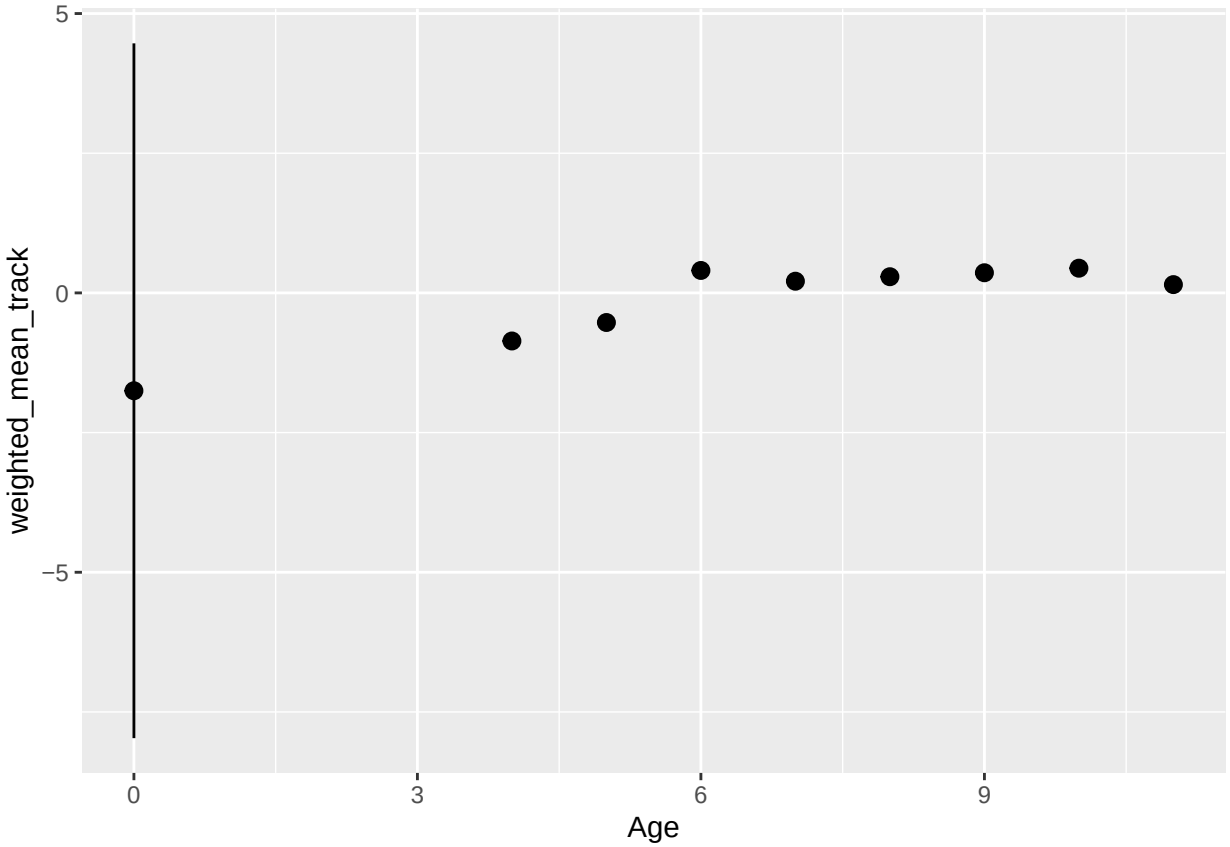


Now, let's try another way. Here, we can also ask R to include some confidence intervals too.

```
ggplot(ckat.dat, aes(x = Age, y = weighted_mean_track)) +
  stat_summary(fun.data = "mean_cl_normal")
```

```
## Warning: Removed 65 rows containing non-finite values (stat_summary).
```

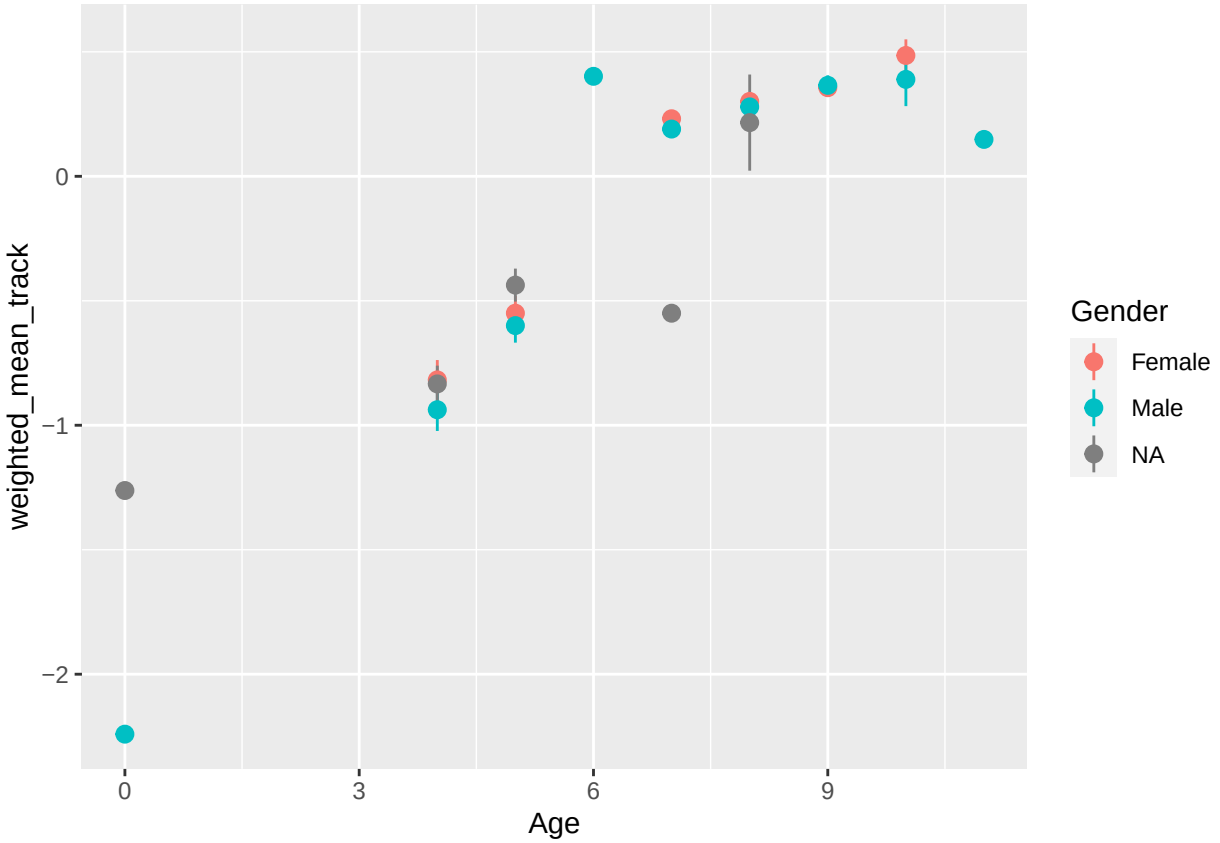
```
## Warning: Removed 2 rows containing missing values (geom_segment).
```




```
ggplot(ckat.dat, aes(x = Age, y = weighted_mean_track, colour = Gender)) +
  stat_summary(fun.data = "mean_cl_normal")
```

```
## Warning: Removed 65 rows containing non-finite values (stat_summary).
```

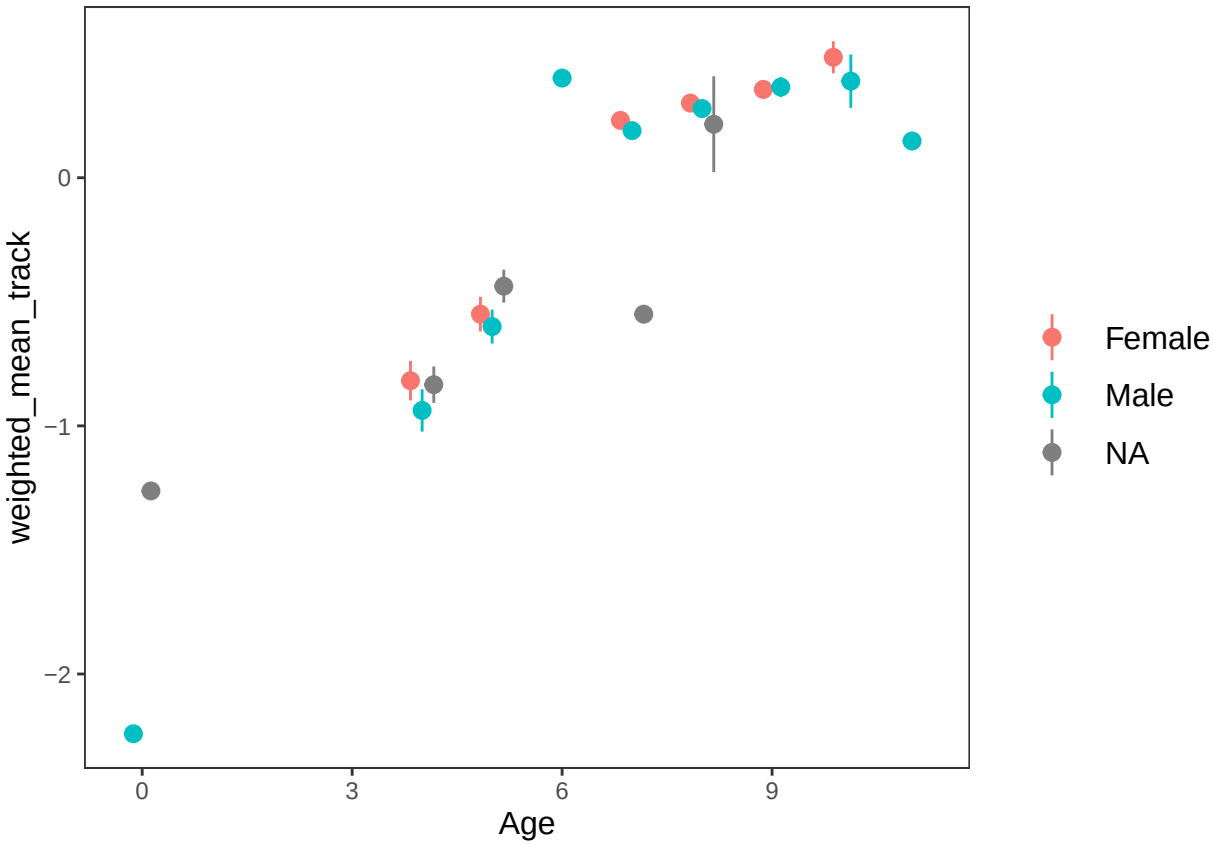
```
## Warning: Removed 5 rows containing missing values (geom_segment).
```




```
ggplot(ckat.dat, aes(x = Age, y = weighted_mean_track, colour = Gender)) +
  stat_summary(fun.data = "mean_cl_normal", position = position_dodge(width = 0.5)) +
  theme_apa()
```

```
## Warning: Removed 65 rows containing non-finite values (stat_summary).
```

```
## Warning: Removed 5 rows containing missing values (geom_segment).
```


weighted_mean_track
0
−1
−2
0
3
6
9
Age
Female
Male
NA