Lab3

Thijs Quast (thiqu264)

10/7/2019

Contents

Question 1

```
set.seed(12345)
```

```
transition <- function(z_t){
    action <- sample((0:2), 1)
    z_t <- rnorm(n = 1, mean = (z_t + action), sd = 1)

    return(z_t)
  }
```

```
emission <- function(z_t){
  action <- sample((-1:1), 1)
  x_t <- rnorm(n = 1, mean = (z_t + action), sd = 1)

  return(x_t)
}
```

```
weights <- function(observation, mean, sd){
  weight <- (dnorm(observation, mean = mean, sd = sd) +
              dnorm(observation, mean = mean-1, sd = sd) +
              dnorm(observation, mean = mean+1, sd = sd))/3

  return(weight)
}
```

```
z_t <- c()
x_t <- c()
z_t[1] <- runif(1, 0, 100)

for (i in 2:100){
  z_t[i] <- transition(z_t = z_t[i-1])
}

for (i in 1:100){
  x_t[i] <- emission(z_t = z_t[i])
}
```

```
particle_filter <- function(observations, M, Time, sd){

  particles <- matrix(0, nrow = M, ncol = Time)

  # Initialization, using initial model:
  initialization <- runif(n = M, min = 0, max = 100)
  particles[,1] <- initialization

  # Prediction:
  for (t in 2:Time){ # For every timestep
    for (m in 1:M){ # Number of particles
      particles[m,t] <- transition(z_t = particles[m, (t-1)])
    }
```

```r
  }

  # Compute weights
  W <- matrix(data = 0, nrow = M, ncol = Time)
  W[,1] <- weights(observation = observations[1], mean = particles[,1], sd = sd)

  for (t in 2:Time){
    for (m in 1:M){
      W[m,t] <- weights(observation = observations[m], mean = particles[m,t], sd = sd)
    }
  }

  corrected_particles <- matrix(0, nrow = M, ncol = Time)

  for (t in 1:Time){
    corrected_particles[,t] <- sample(x = particles[,t], size = M, replace = TRUE,
                                      prob = W[,t])
  }

  list <- list("particles" = particles, "weights" = W,
               "corrected_particles" = corrected_particles)
  return(list)
}
```

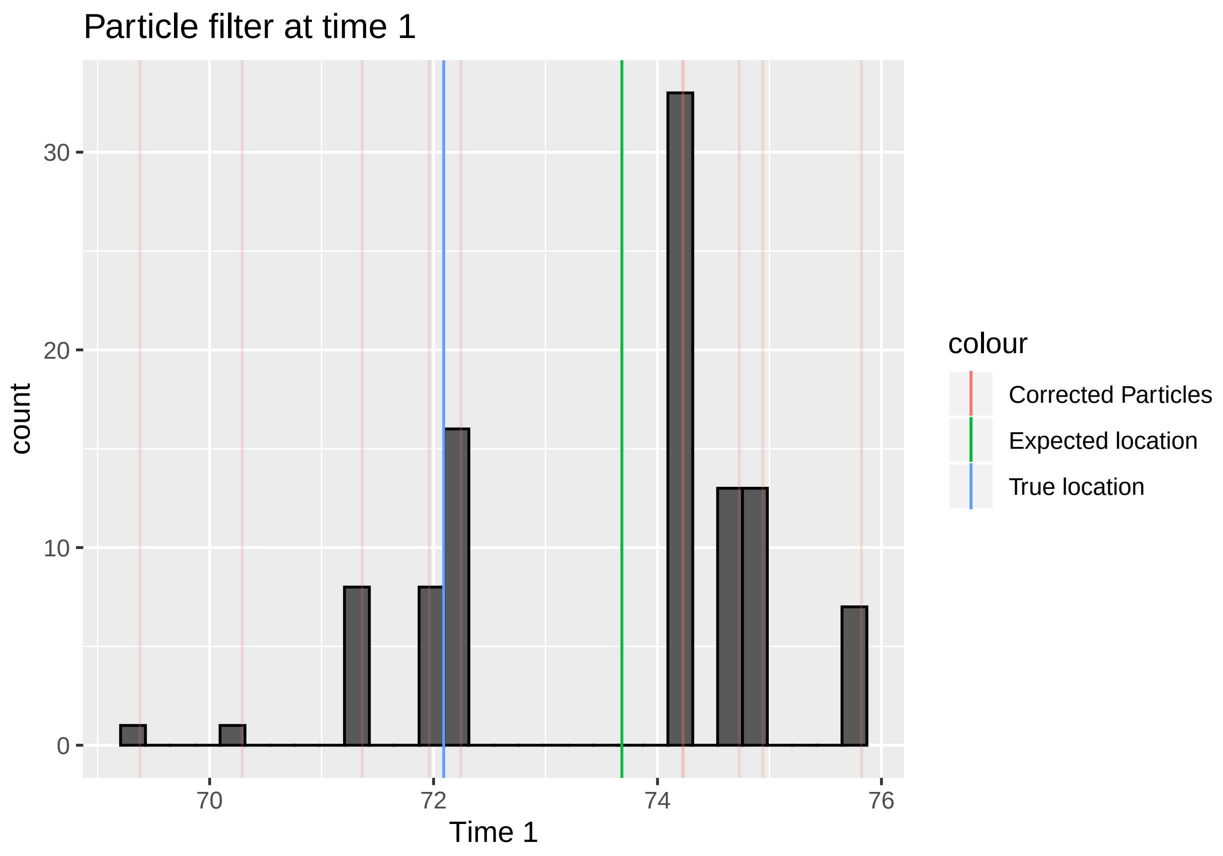
```r
# Creating the particle filter
filter <- particle_filter(observations = x_t, M = 100, Time = 100, sd = 1)
```

```r
library(ggplot2)
df <- as.data.frame(filter$corrected_particles)
colnames(df) <- as.character(c(1:100))
```

```r
# Function to plot particle filter at different timesteps
plot_particle_filter <- function(data, timestep){
  plot <- ggplot(data = data, aes(x = data[,timestep])) + geom_histogram(colour="black")
  plot <- plot + xlab(paste("Time", as.character(timestep)))
  plot <- plot + geom_vline(aes(xintercept = data[,timestep], col="Corrected Particles"), alpha=0.20)
  plot <- plot + geom_vline(aes(xintercept = mean(data[,timestep]), col="Expected location"))
  plot <- plot + geom_vline(aes(xintercept = z_t[timestep], col="True location"))
  plot <- plot + ggtitle(paste("Particle filter at time", timestep))
  return(plot)
}
```

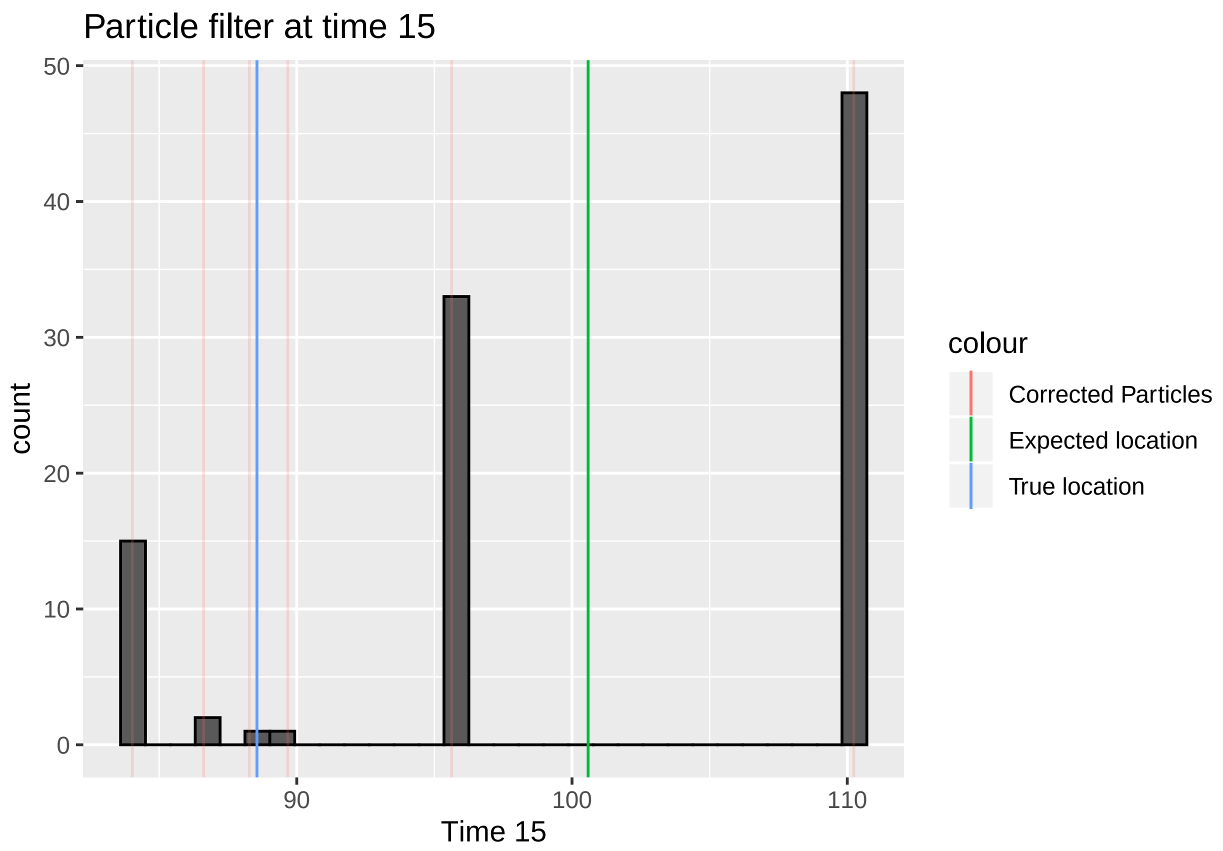
```r
plot_particle_filter(data = df, timestep = 1)
```

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



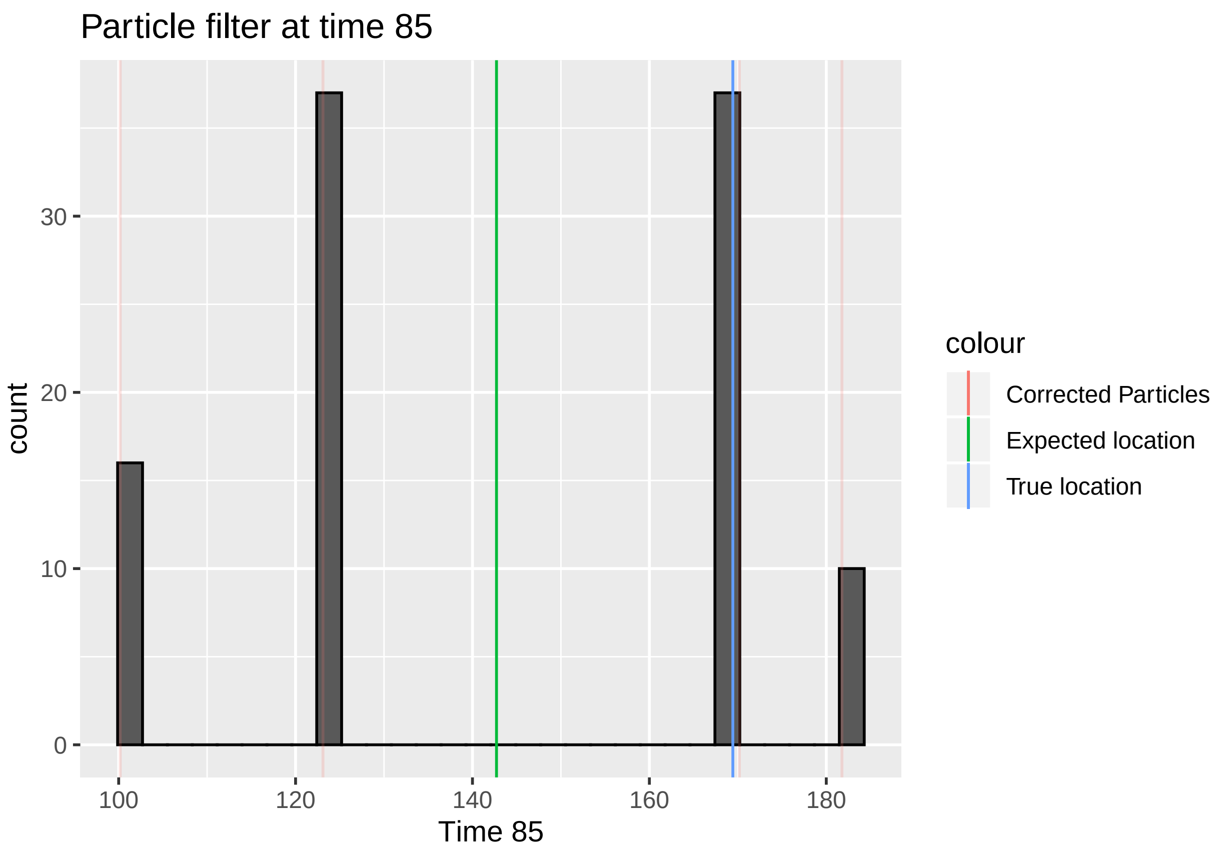
```
plot_particle_filter(data = df, timestep = 15)
```

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



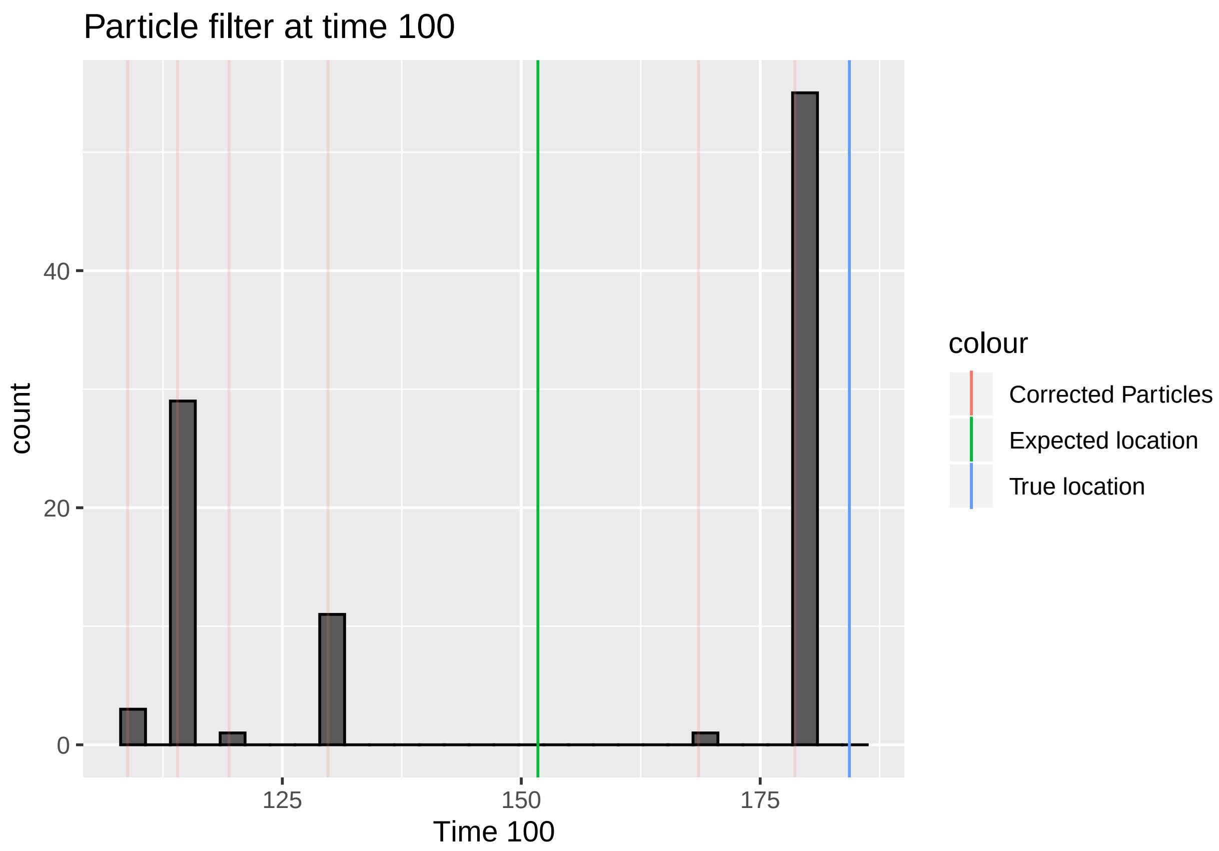
```r
plot_particle_filter(data = df, timestep = 85)
```

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



```r
plot_particle_filter(data = df, timestep = 100)
```

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



Question 2

```
# Emmision model with sd = 5
emission_5 <- function(z_t){
  action <- sample((-1:1), 1)
  x_t <- rnorm(n = 1, mean = (z_t + action), sd = 5)

  return(x_t)
}
```

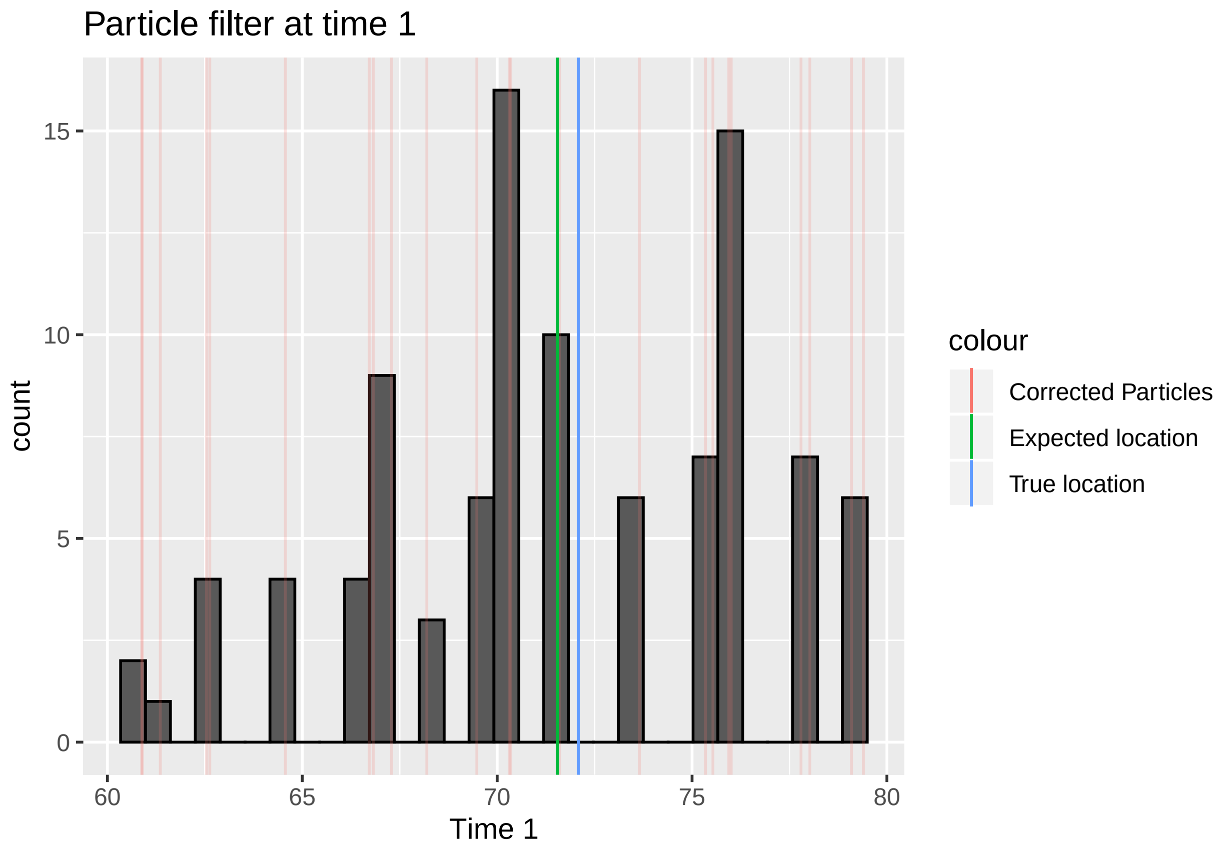
```
x_t_5 <- c()

for (i in 1:100){
  x_t_5[i] <- emission_5(z_t = z_t[i])
}
```

```
filter_5 <- particle_filter(observations = x_t_5, M = 100, Time = 100, sd = 5)
```

```
library(ggplot2)
df_5 <- as.data.frame(filter_5$corrected_particles)
colnames(df_5) <- as.character(c(1:100))
```

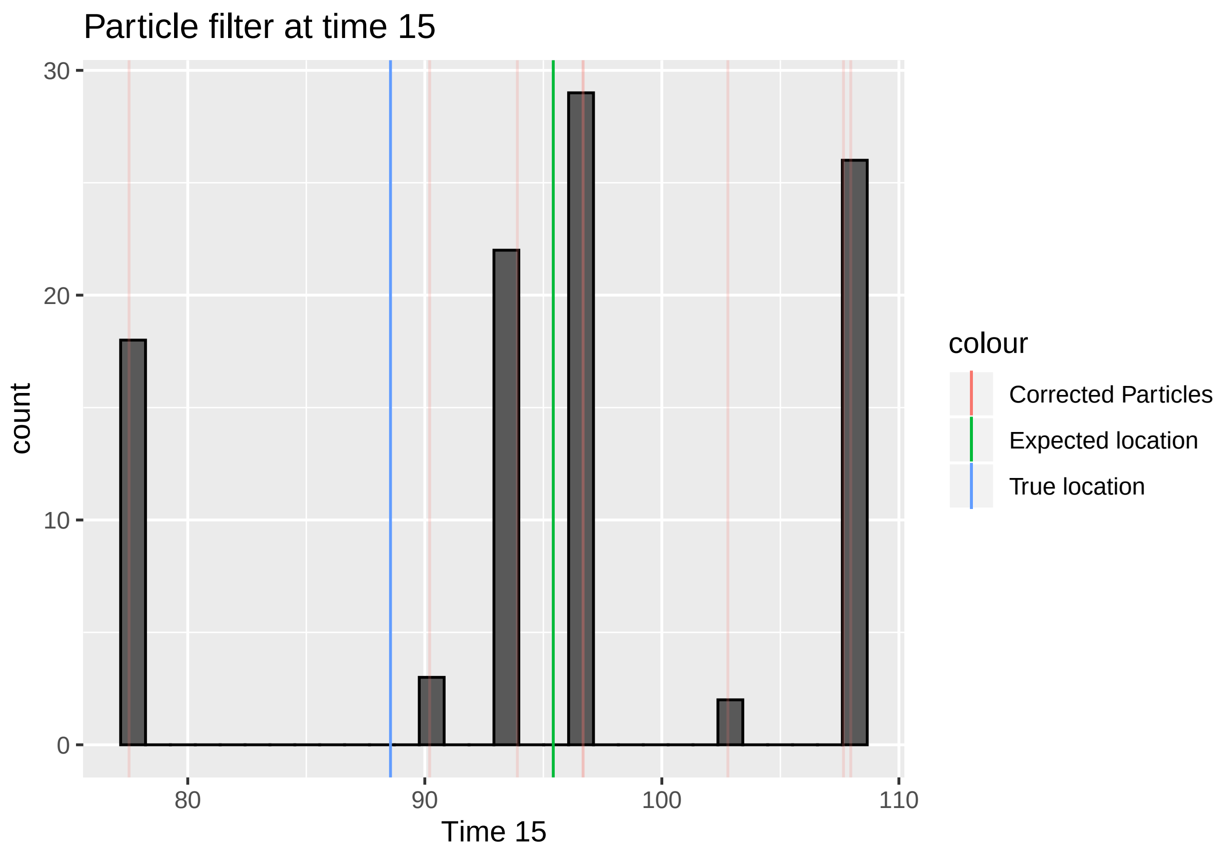
```
plot_particle_filter(data = df_5, timestep = 1)
```

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



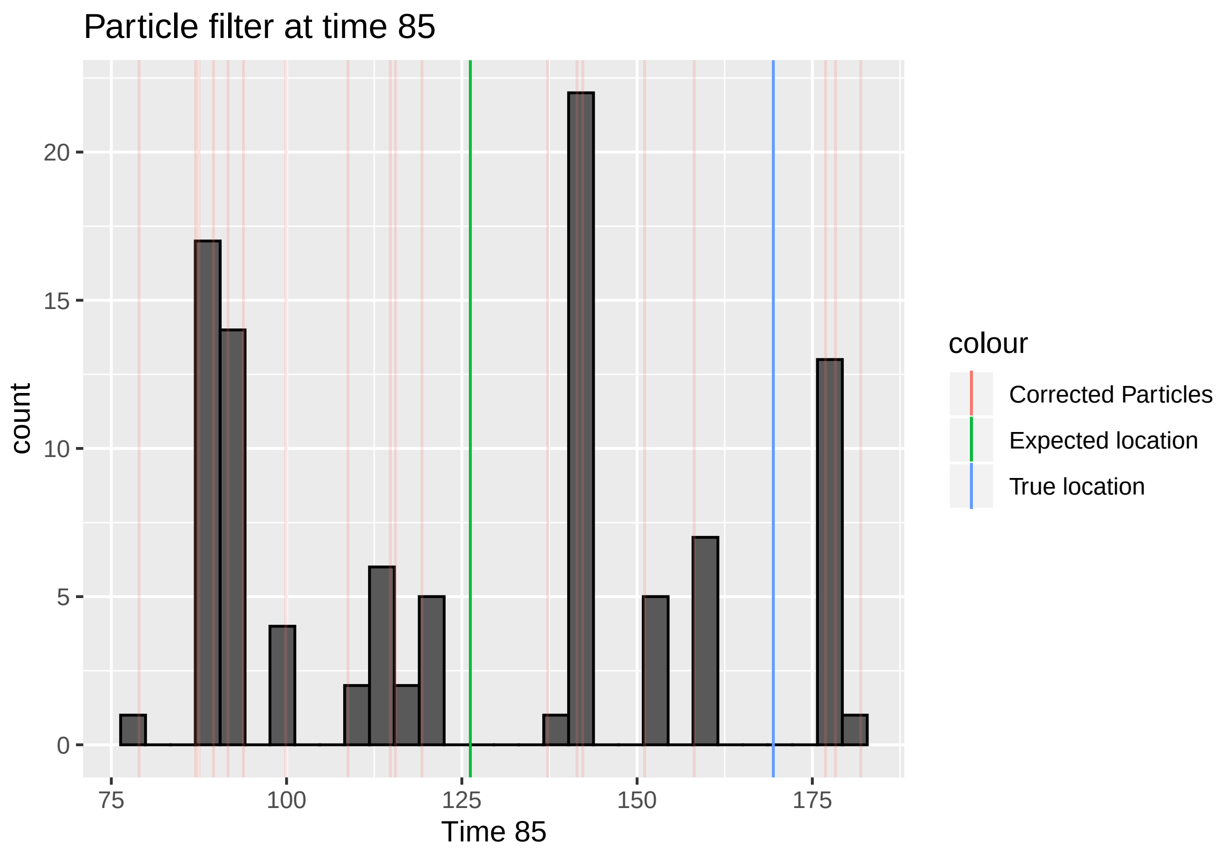
```
plot_particle_filter(data = df_5, timestep = 15)
```

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



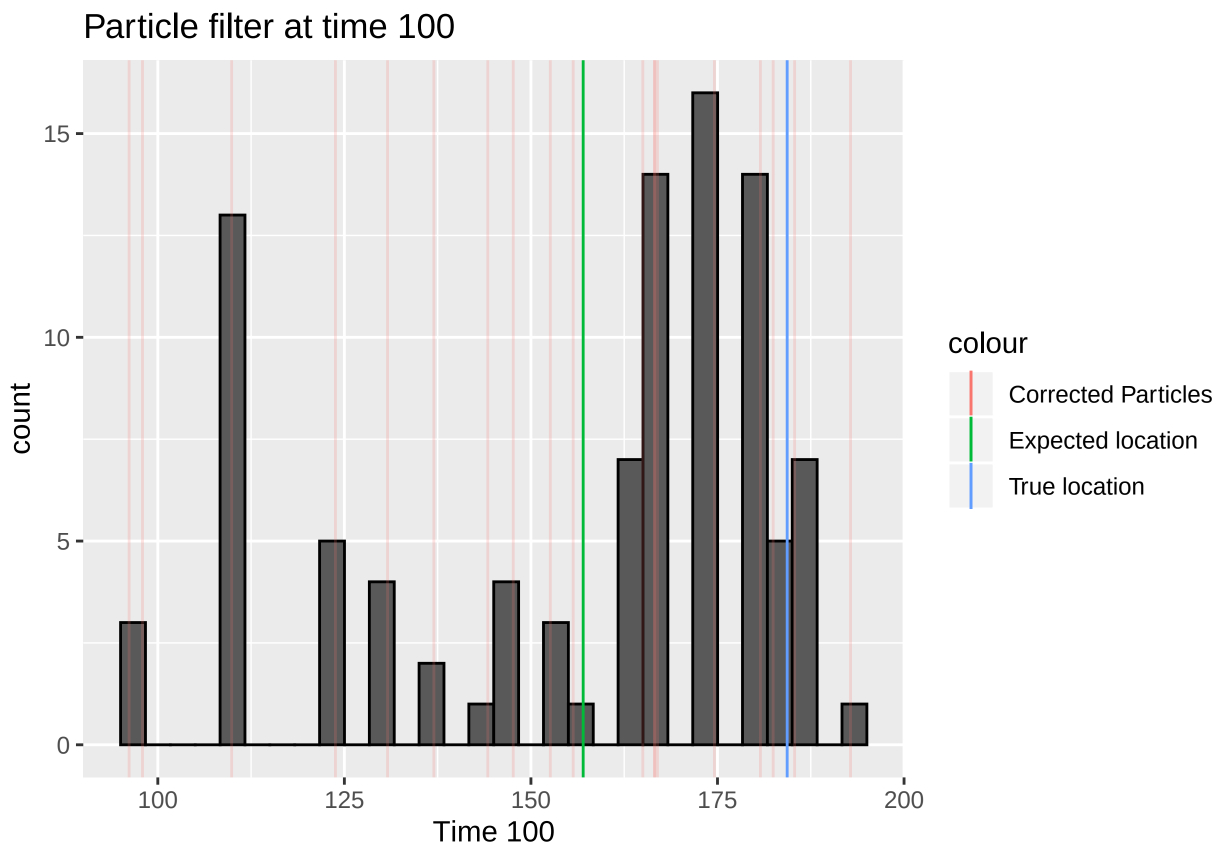
```
plot_particle_filter(data = df_5, timestep = 85)
```

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



```r
plot_particle_filter(data = df_5, timestep = 100)
```

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



```
# Emmision model with sd = 50
emission_50 <- function(z_t){
  action <- sample((-1:1), 1)
  x_t <- rnorm(n = 1, mean = (z_t + action), sd = 50)

  return(x_t)
}
```

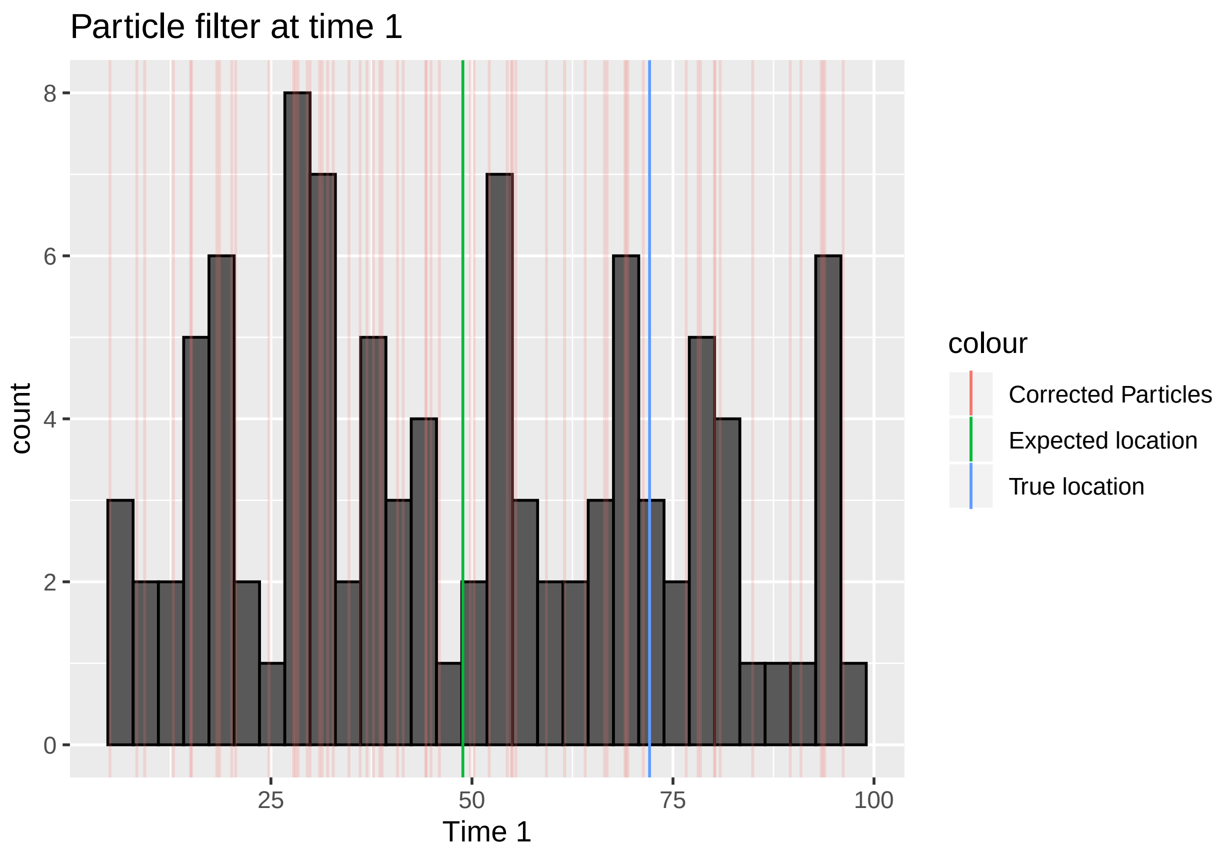
```
x_t_50 <- c()

for (i in 1:100){
  x_t_50[i] <- emission_50(z_t = z_t[i])
}
```

```
filter_50 <- particle_filter(observations = x_t_50, M = 100, Time = 100, sd = 50)
```

```
library(ggplot2)
df_50 <- as.data.frame(filter_50$corrected_particles)
colnames(df_50) <- as.character(c(1:100))
```

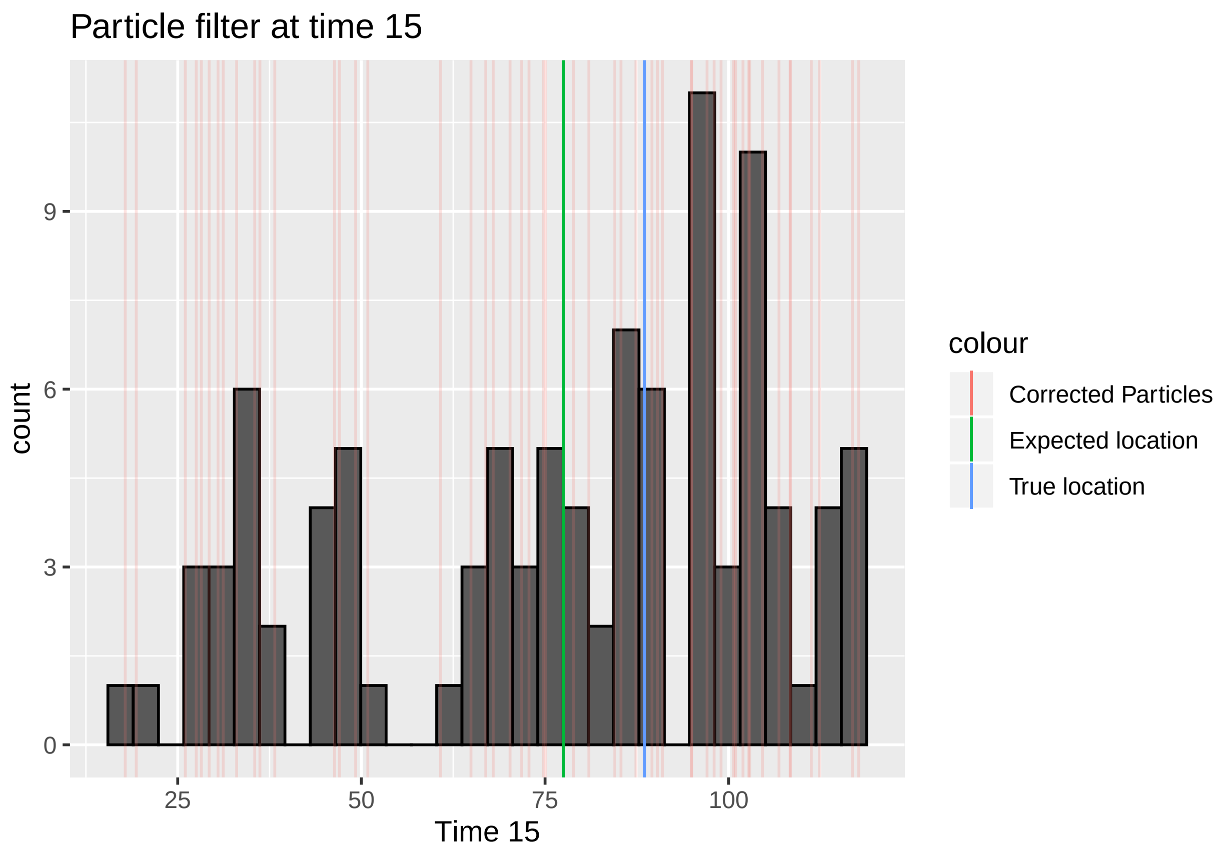
```
plot_particle_filter(data = df_50, timestep = 1)
```

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



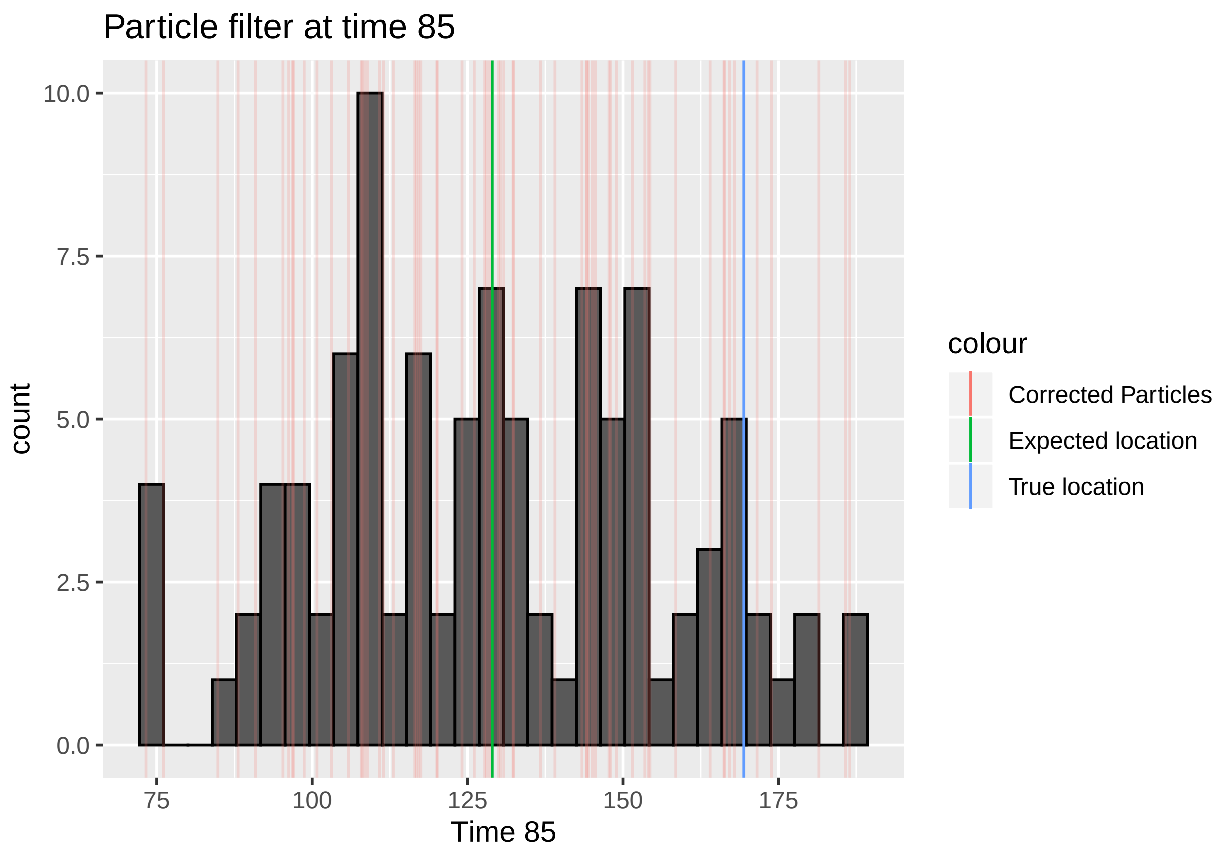
```
plot_particle_filter(data = df_50, timestep = 15)
```

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



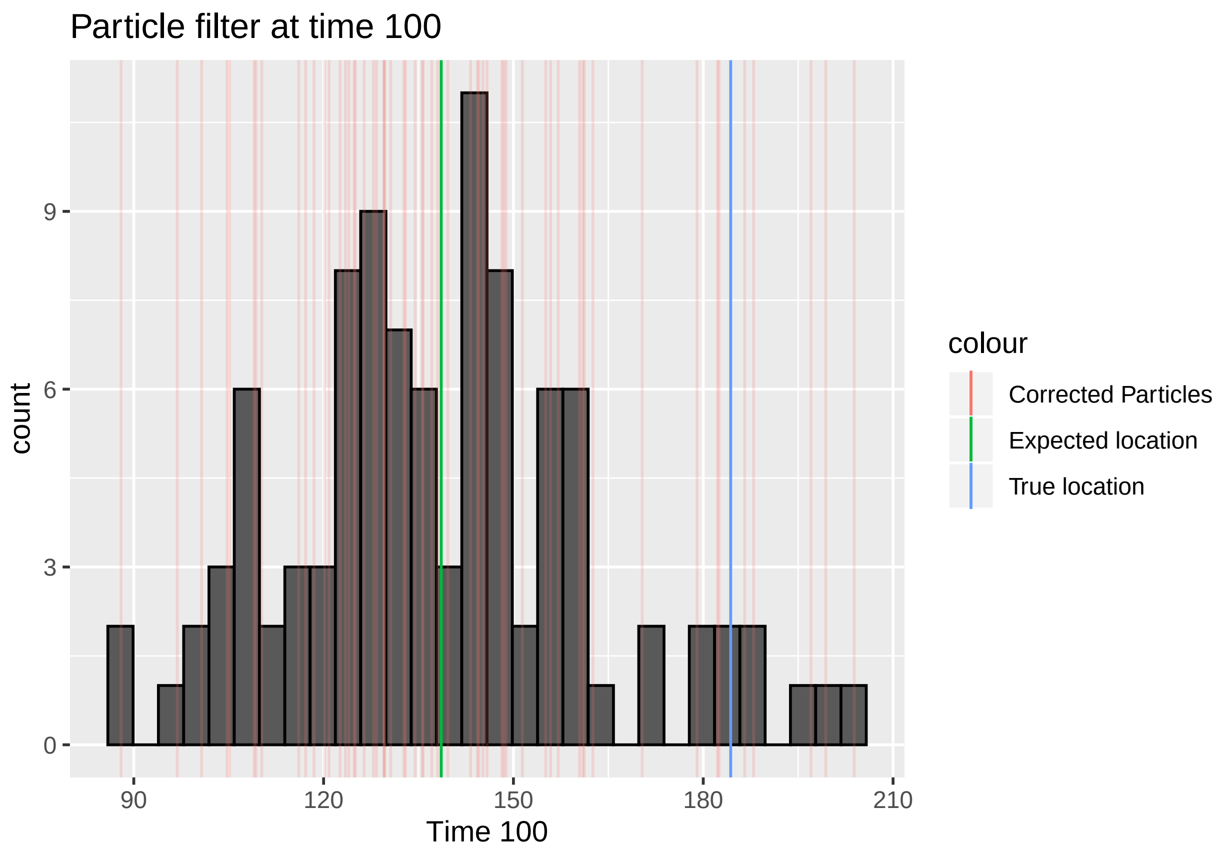
```
plot_particle_filter(data = df_50, timestep = 85)
```

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



```
plot_particle_filter(data = df_50, timestep = 100)
```

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



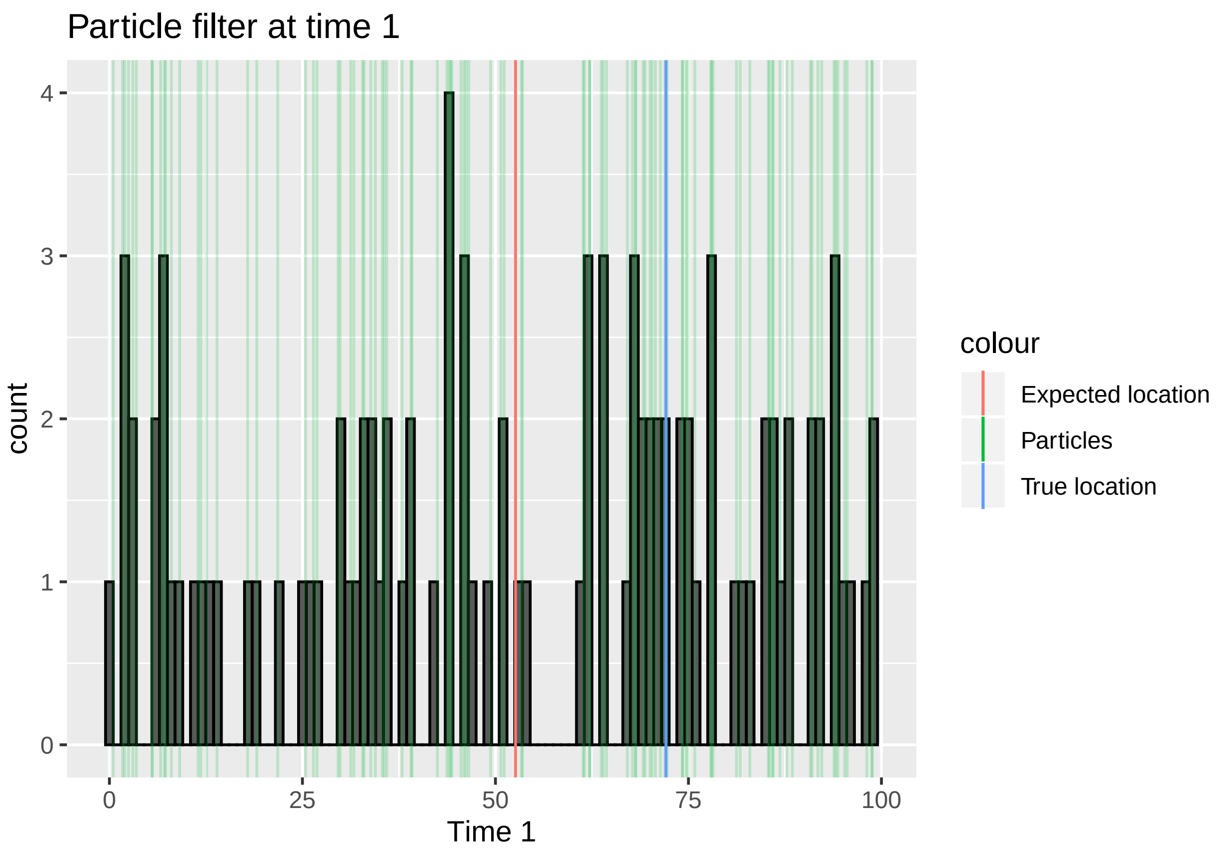
Generally, as one increases the the standard deviation of the emission model the histogram of the particles is distributed over a wider deviation.

Question 3

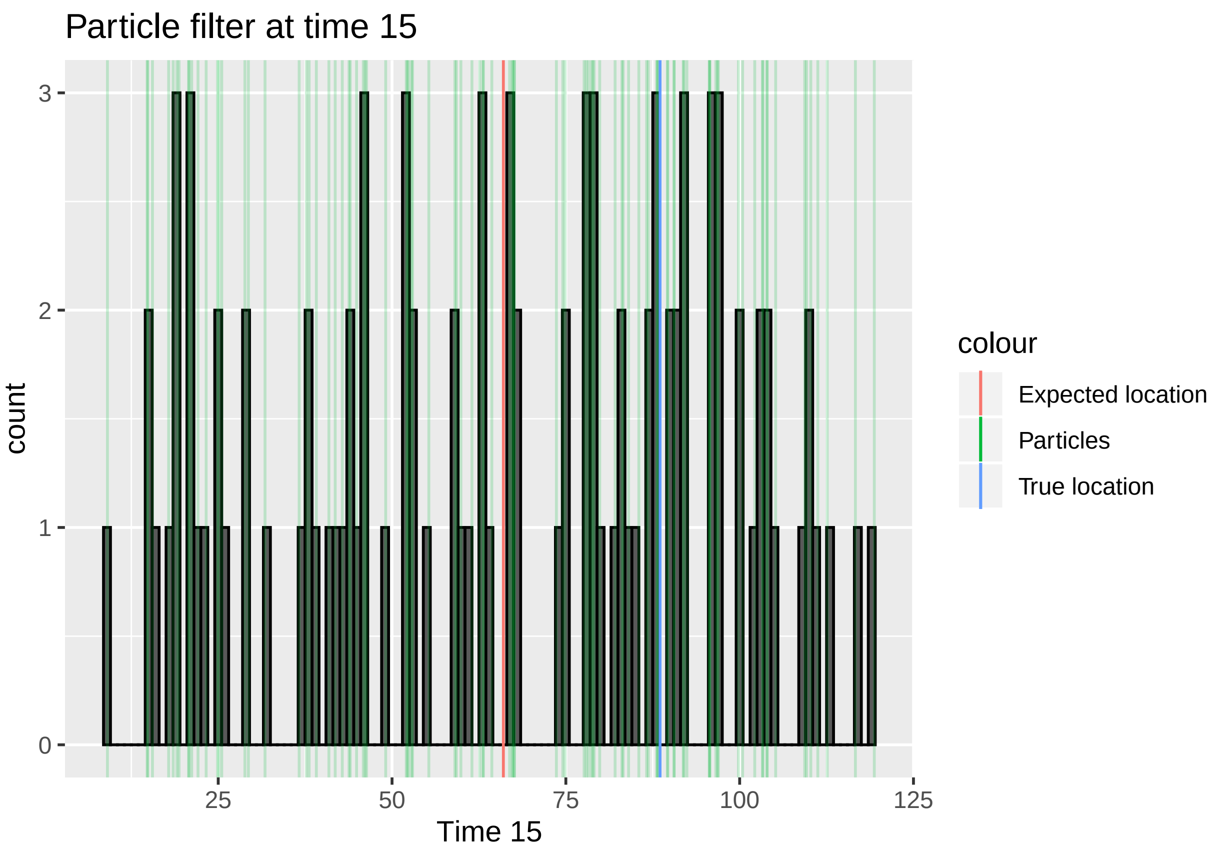
```
# Function to plot particle filter at different timesteps
plot_particle_filter2 <- function(data, timestep){
  plot <- ggplot(data = data, aes(x = data[,timestep])) + geom_histogram(binwidth = 1, colour="black")
  plot <- plot + xlab(paste("Time", as.character(timestep)))
  plot <- plot + geom_vline(aes(xintercept = data[,timestep], col="Particles"), alpha=0.20)
  plot <- plot + geom_vline(aes(xintercept = mean(data[,timestep]), col="Expected location"))
  plot <- plot + geom_vline(aes(xintercept = z_t[timestep], col="True location"))
  plot <- plot + ggtitle(paste("Particle filter at time", timestep))
  return(plot)
}
```

```
df <- as.data.frame(filter$particles)
colnames(df) <- as.character(c(1:100))
```

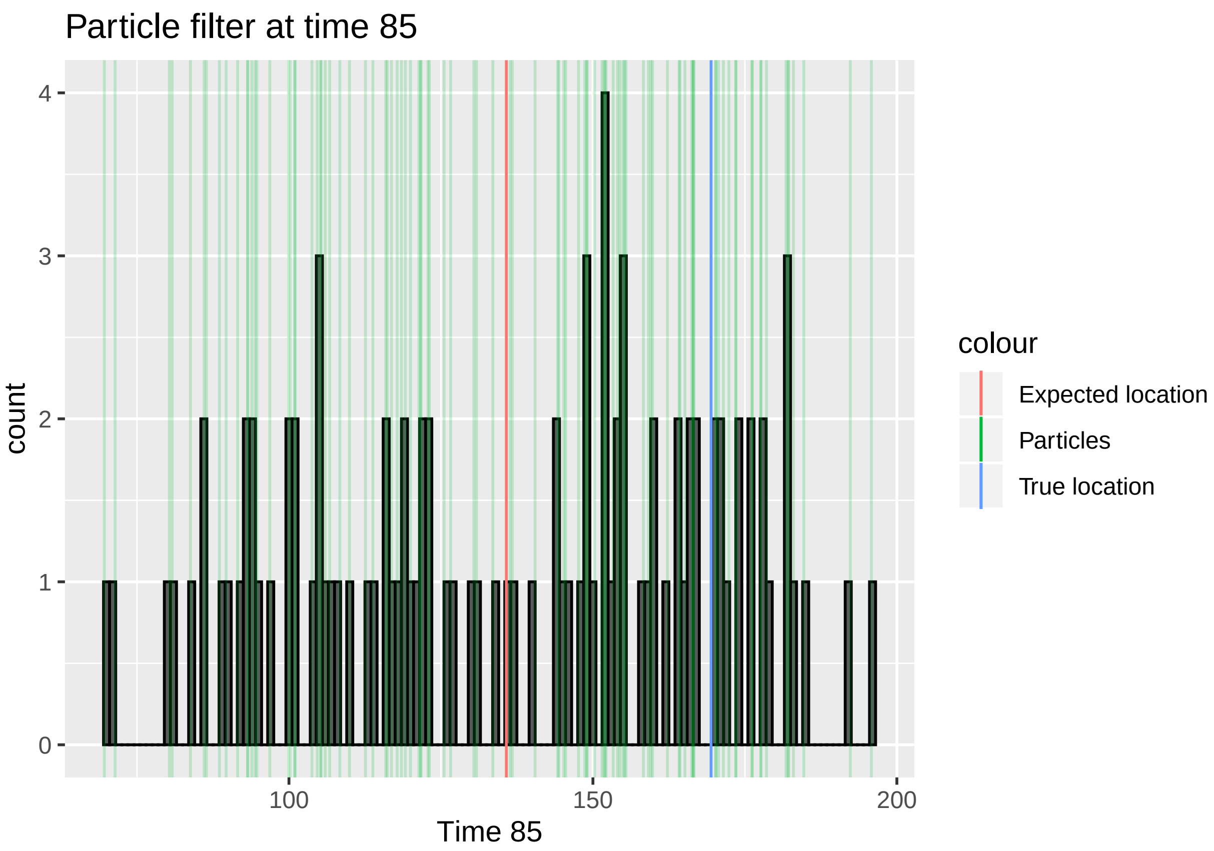
```
plot_particle_filter2(data = df, timestep = 1)
```



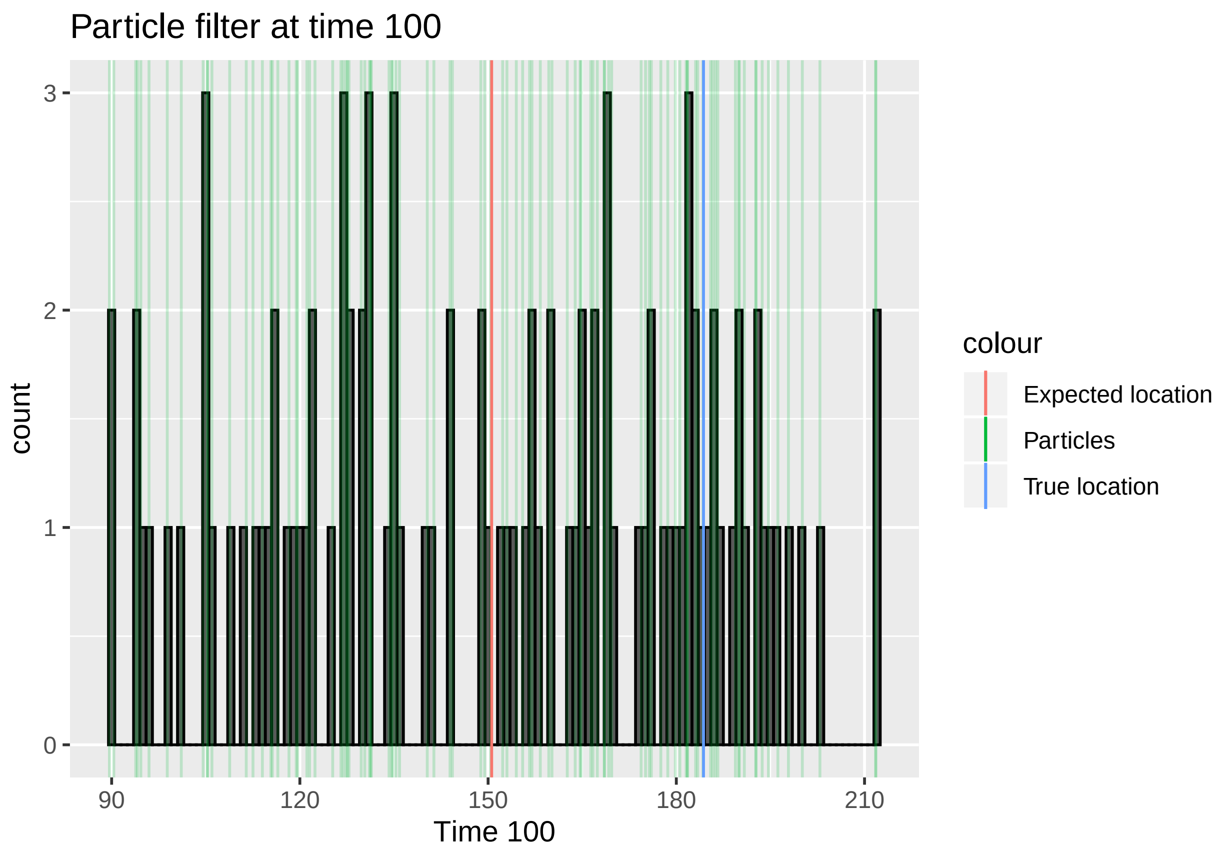
```r
plot_particle_filter2(data = df, timestep = 15)
```



```r
plot_particle_filter2(data = df, timestep = 85)
```



```r
plot_particle_filter2(data = df, timestep = 100)
```



The difference with the correction in the particles filter is that without a correction the plot shows way more particles, distributed over a wider range. This is because in the correction the filter will select the ones with higher weights multiple times.