Advanced Machine Learning (732A96) Lab3

Anubhav Dikshit(anudi287)

11 October, 2019

Contents

Questions

The purpose of the lab is to put in practice some of the concepts covered in the lectures. To do so, you are asked to implement the particle filter for robot localization. For the particle filter algorithm, please check Section 13.3.4 of Bishop's book and/or the slides for the last lecture on state space models (SSMs). The robot moves along the horizontal axis according to the following SSM:

Trasition model:

$$p(z_t|z_{t-1}) = (N(z_t|z_{t-1}, 1) + N(z_t|z_{t-1} + 1, 1) + N(z_t|z_{t-1} + 2, 1))/3$$

Emission model:

$$p(x_t|z_t) = (N(x_t|z_t, 1) + N(x_t|z_t - 1), 1) + +N(x_t|z_t + 1), 1))/3$$

Initial model:

$$p(z_1) = Uniform(0, 100)$$

Questions 1

1) Implement the SSM above. Simulate it for T = 100 time steps to obtain $z_{1:100}$ (i.e., states) and $x_{1:100}$ (i.e., observations). Use the observations (i.e., sensor readings) to identify the state (i.e., robot location) via particle filtering. Use 100 particles. Show the particles, the expected location and the true location for the first and last time steps, as well as for two intermediate time steps of your choice.

```r
set.seed(12345)

gen_data <- function(N){
# states
z_t <- vector(length = N)
# observations
x_t <- vector(length = N)

# initial conditions
z_t[1] <- runif(n = 1, min = 0, max = N)

comp <- sample(1:3, prob = c(1 / 3, 1 / 3, 1 / 3), size = 1)
mu <- c(z_t[1], z_t[1] - 1, z_t[1] + 1)
x_t[1] <- rnorm(1, mean = mu[comp], 1)

for (i in 2:N) {
    comp <- sample(1:3, prob = c(1 / 3, 1 / 3, 1 / 3), size = 1)
    muz <- c(z_t[i - 1], z_t[i - 1] + 1, z_t[i - 1] + 2)
    z_t[i] <- rnorm(1, mean = muz[comp], 1)
    mu <- c(z_t[i], z_t[i] - 1, z_t[i] + 1)
    x_t[i] <- rnorm(1, mean = mu[comp], 1)
  }

sample_data <- data.frame(observation = x_t, states = z_t,
                          index=1:N)

# plot of samples
p1 <- ggplot(data=sample_data, aes(x=index)) +
  geom_line(aes(y=observation, color="Observations")) +
  geom_line(aes(y=states, color="True location")) +
```

```r
        scale_colour_manual("", breaks = c("Observations", "True location"),
                            values = c("#000000", "#E69F00", "#56B4E9", "#009E73")) +
  xlab("Index") +
  ylab("Location") +
  ggtitle("Observed and True location")

print(p1)

# density plots
p2 <- ggplot(data=sample_data) +
  geom_density(aes(x=observation, color="Observations")) +
  geom_density(aes(x=states, color="True location")) +
        scale_colour_manual("", breaks = c("Observations", "True location"),
                            values = c("#000000", "#E69F00", "#56B4E9", "#009E73")) +
    xlab("Index") +
  ylab("Density") +
  ggtitle("Desnity of Observed and True location")

print(p2)

return(list(z_t=z_t, x_t=x_t))
}
```

At time step 1, we know $p(z_1)$ and therefore sample $L = 100$ particles from it. And we also know the emission model $p(x_1|z_1^{(l)})$, which we use to compute the weights:

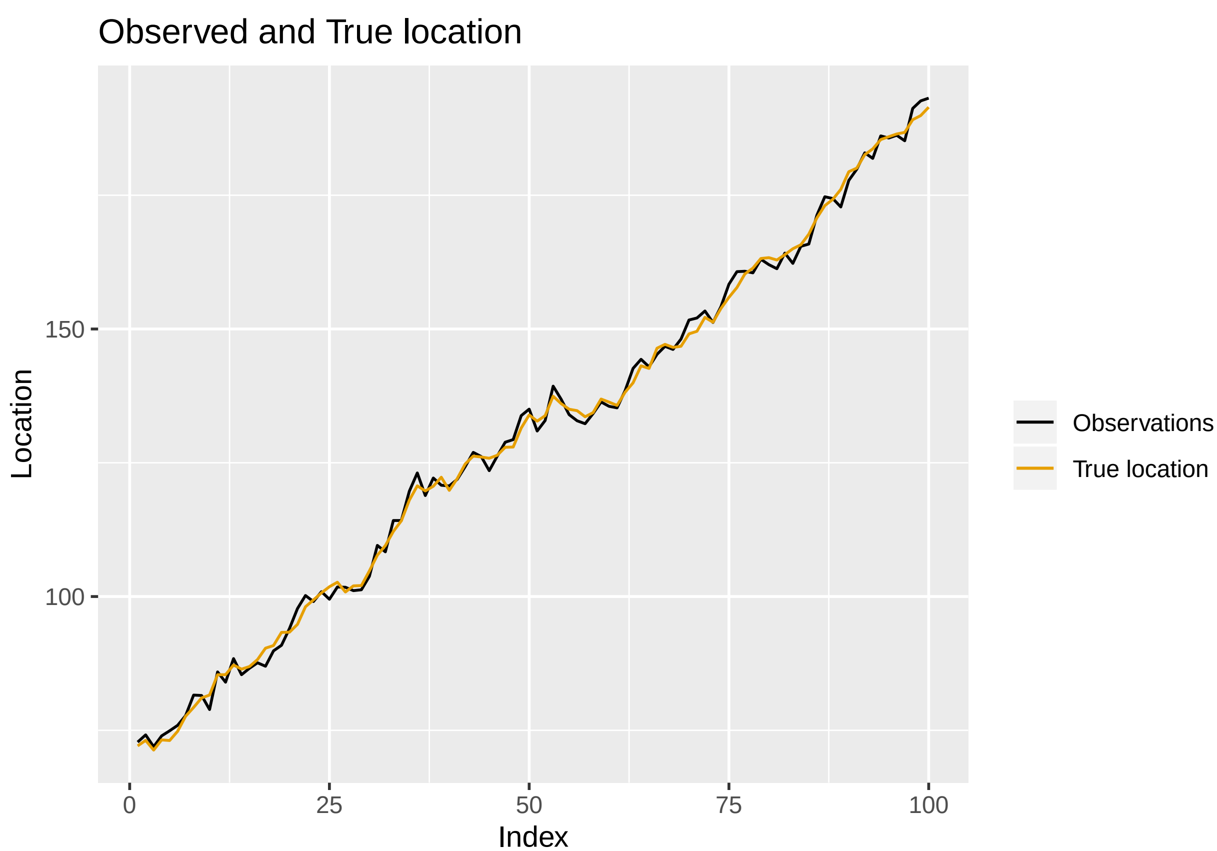
$$w_n^{(l)} = \frac{p(x_n|z_n^{(l)})}{\sum_{m=1}^{L} p(x_n|z_n^{(l)})}$$

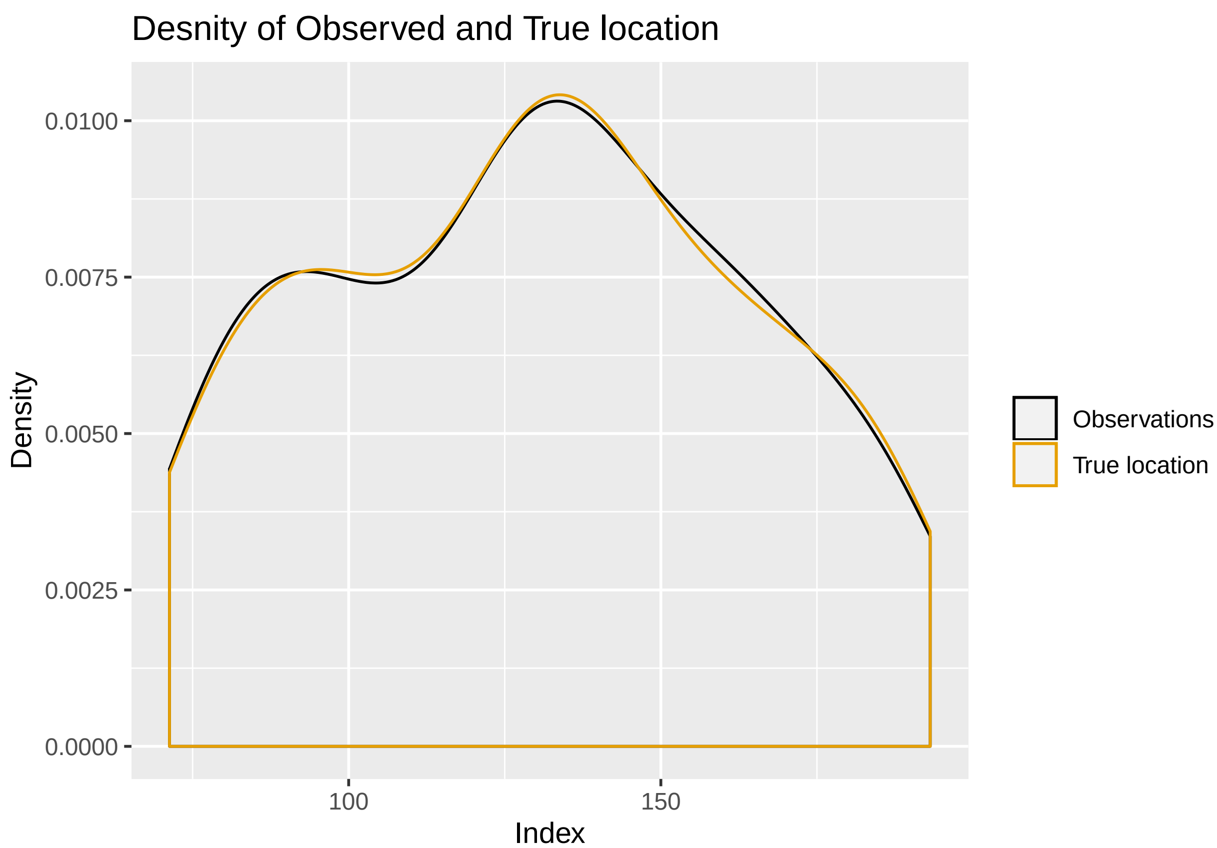
Note that the weights satisfy $0 \leq w_n^{(l)}$ and $\sum_l w_n^{(l)} = 1$

```r
my_particle_filter <- function(M, Times, obs, sd_tra, sd_emi, ignore_weight) {

  temp_particles <- NULL
  weights <- NULL
  # Create matrix that will contain all particles for each time step
  particles <- matrix(NA, nrow = Times, ncol = M)

  # Generate initial M particles, Uniform(0, 100)
  particles[1, ] <- runif(n = M, min = 0, max = 100)

  for (t in 2:Times) {
    for (m in 1:M) {

  # transiton matrix
    selection <- sample(1:3, prob = c(1 / 3, 1 / 3, 1 / 3), size = 1)
    mean_tra <- c(particles[t - 1, m], particles[t - 1, m] + 1,
                  particles[t - 1, m] + 2)
    temp_particles[m] <- rnorm(n=1, mean = mean_tra[selection], sd = sd_tra)


  # emission matrix
```

```r
    weights[m] <- mean(dnorm(x=obs[t], mean = temp_particles[m]-1, sd = sd_emi),
                       dnorm(x=obs[t], mean = temp_particles[m], sd = sd_emi),
                       dnorm(x=obs[t], mean = temp_particles[m]+1, sd = sd_emi))

    }
    weights = weights/sum(weights)

    if(ignore_weight == TRUE){
            particles[t, ] <- sample(x = temp_particles, size = M,
                                     replace = TRUE)
    }else{
      particles[t, ] <- sample(x = temp_particles, size = M, replace = TRUE,
                               prob = weights)}

  }

  return(particles)
}
```

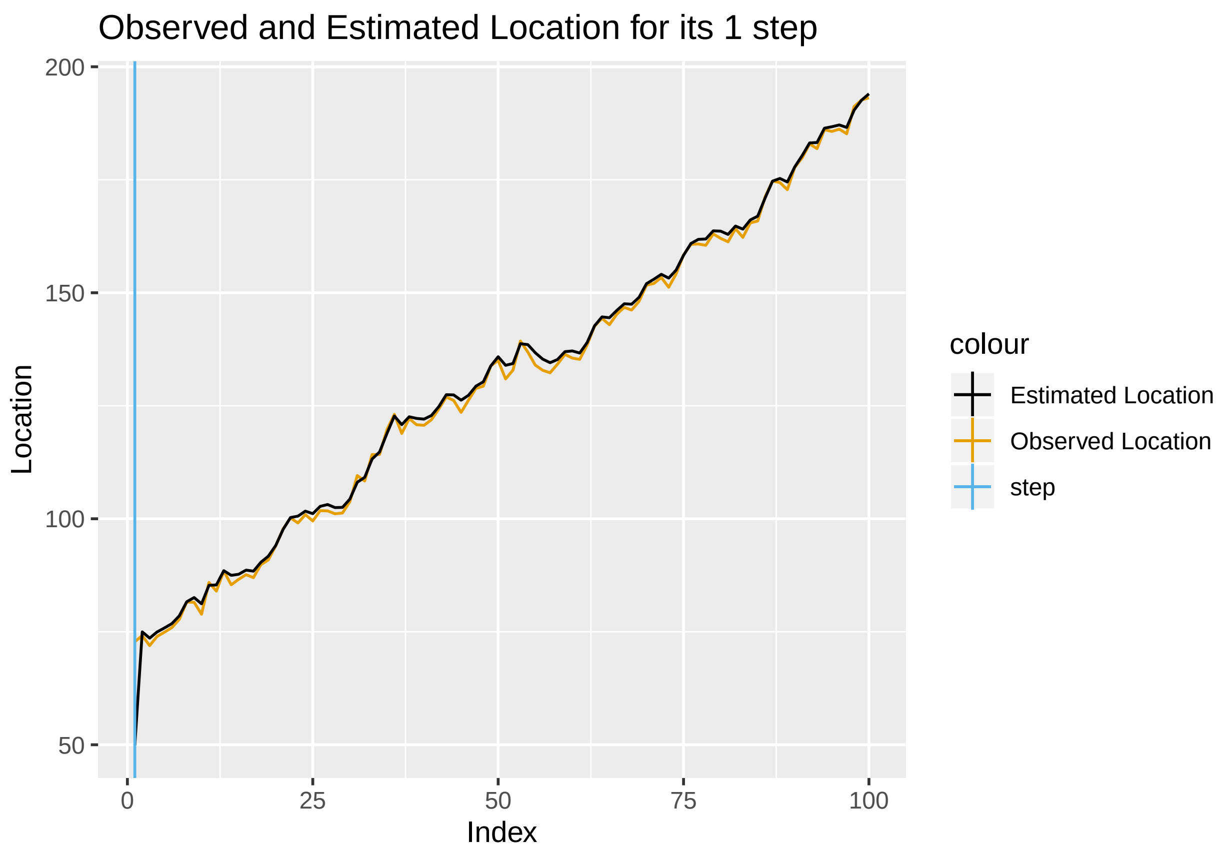
```r
set.seed(12345)
# generate data
sample_data <- gen_data(N=100)
```





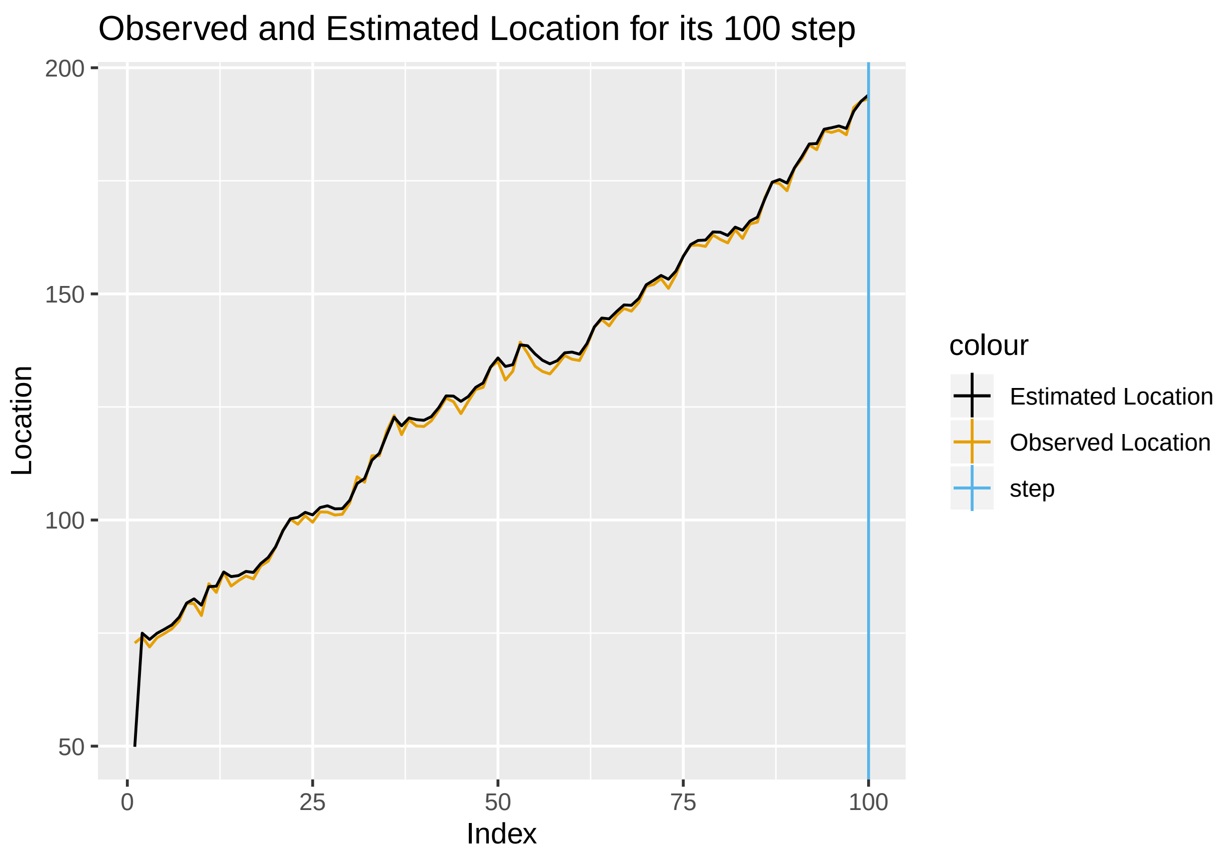

```
# particle filter
particles <- my_particle_filter(M=100, Times = 100, obs=sample_data$x_t,
                                sd_tra=1, sd_emi=1, ignore_weight = FALSE)


location <- data.frame(true_location=sample_data$z_t,
                  observed_location=sample_data$x_t,
                  estimated_location = rowMeans(particles, na.rm = TRUE),
                  index = 1:NROW(particles)) %>% as.data.frame()


# plot function
plot_location <- function(step, df){
plot1 <- ggplot(data=df, aes(x=index)) +
geom_line(aes(y=observed_location, color="Observed Location")) +
geom_line(aes(y=estimated_location, color="Estimated Location")) +
geom_vline(aes(xintercept=step, color="step")) +
  xlab("Index") +
  ylab("Location") +
      scale_colour_manual(values = c("#000000", "#E69F00", "#56B4E9", "#009E73")) +
ggtitle(paste0("Observed and Estimated Location for its ", step, " step"))

df <- df[step,]
result <- data.frame(step = step,
  true_location = df$true_location,
  observed_location = df$observed_location,
```

```r
  estimated_location = df$estimated_location)

print(plot1)
kable(result, "latex", booktabs = T) %>% kable_styling(latex_options = "striped")
}

plot_location(step=1, df=location)
```



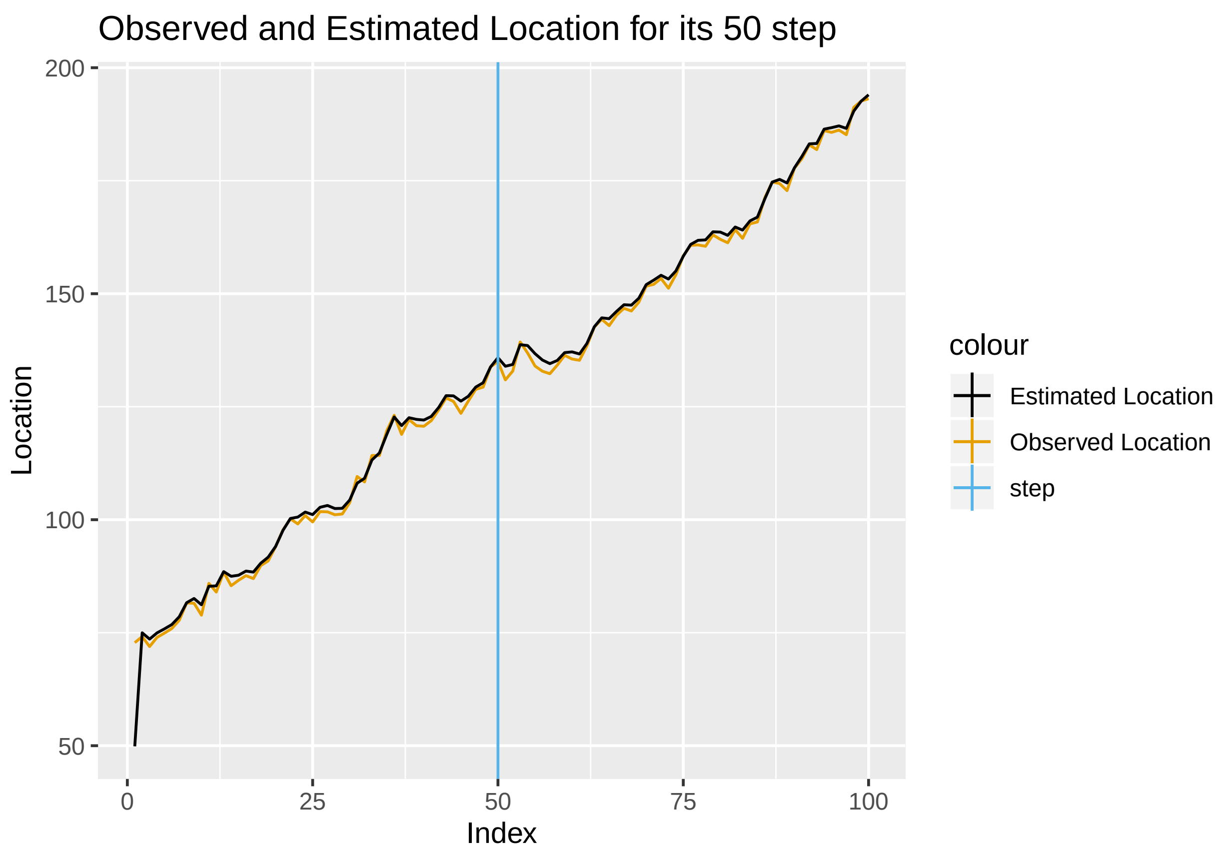
step	true_location	observed_location	estimated_location
1	72.09039	72.79986	49.85403

```r
plot_location(step=100, df=location)
```



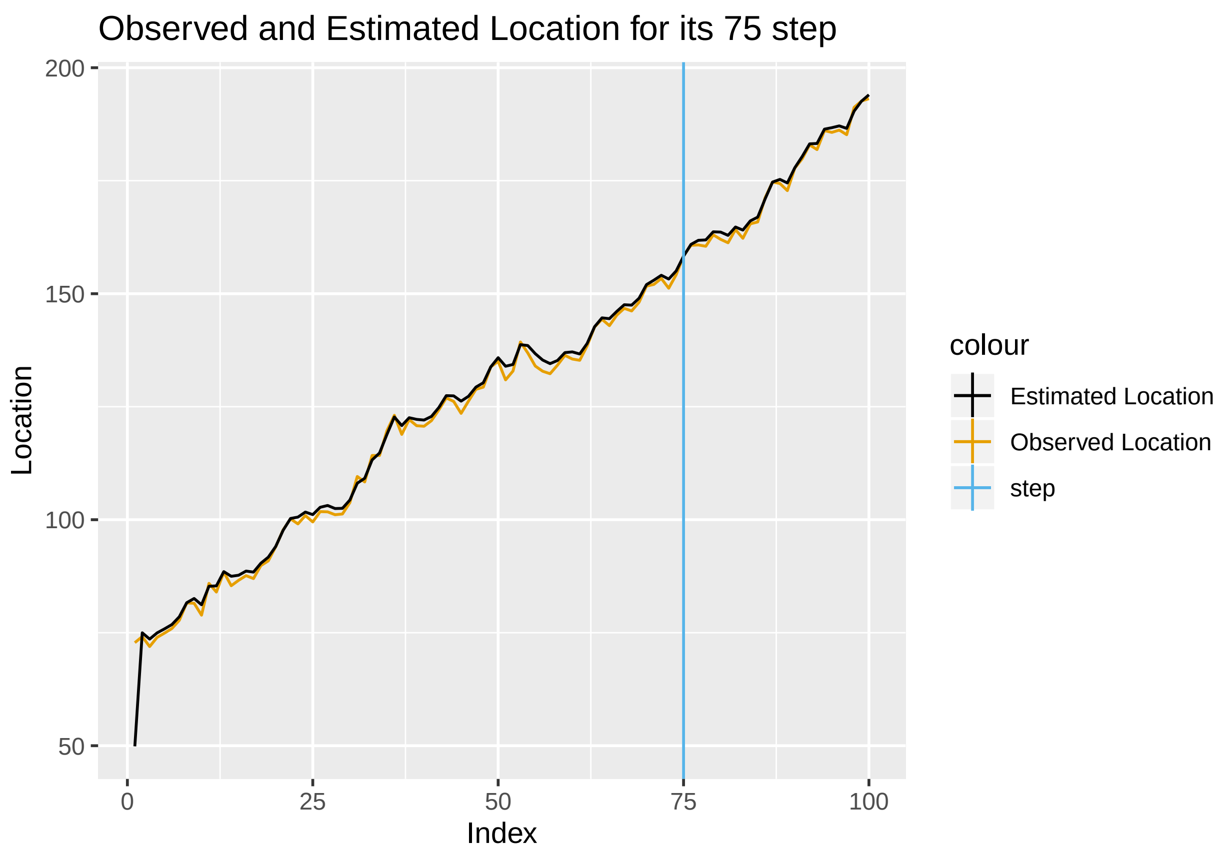
step	true_location	observed_location	estimated_location
100	191.4443	193.1422	194.0084

```
plot_location(step=50, df=location)
```



step	true_location	observed_location	estimated_location
50	133.9368	135.0235	135.847

```
plot_location(step=75, df=location)
```

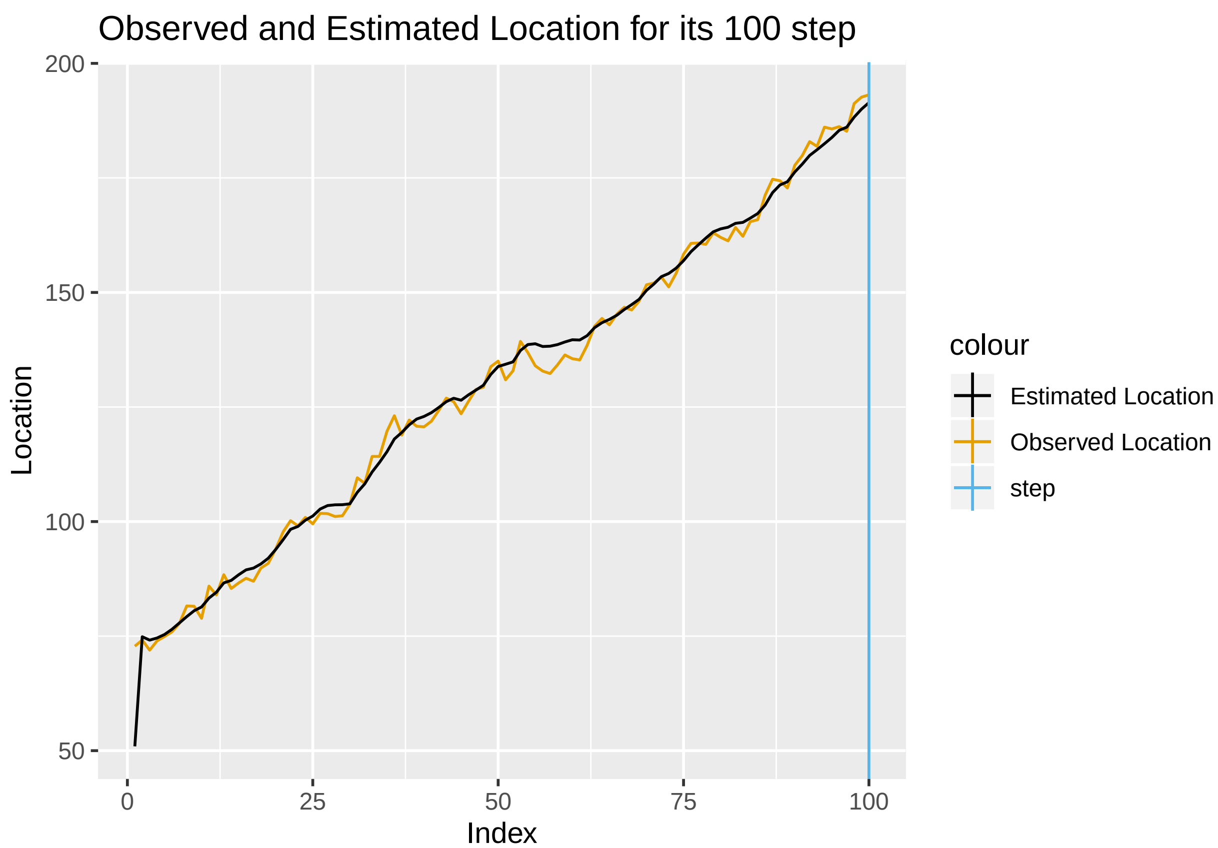


step	true_location	observed_location	estimated_location
75	155.9155	158.3693	158.2851

Questions 2

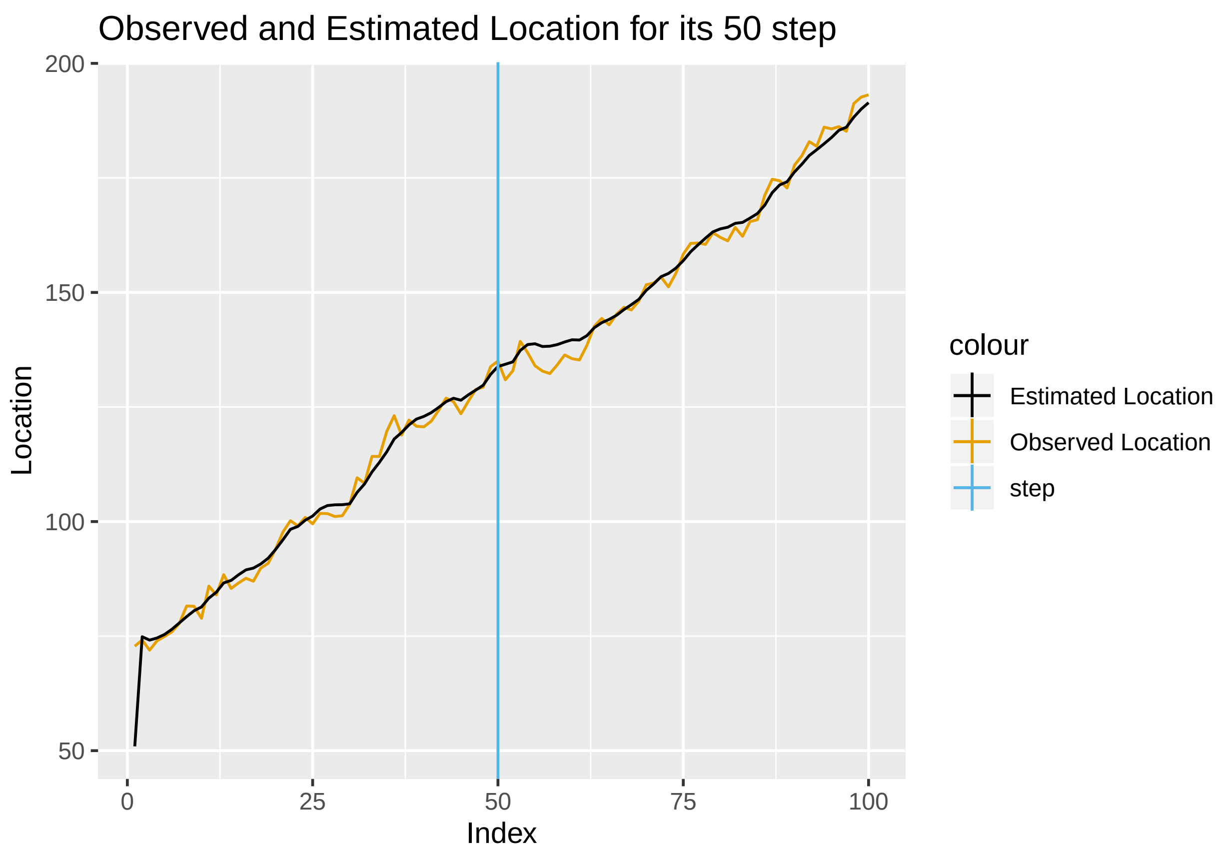
2) Repeat the exercise above replacing the standard deviation of the emission model with 5 and then with 50. Comment on how this affects the results.

```
set.seed(12345)
# with standard deviation of the emission model of 5
particles_sd_emis_5 <- my_particle_filter(M=100, Times = 100, obs=sample_data$x_t,
                                          sd_tra=1, sd_emi=5, ignore_weight = FALSE)

location_sd_5 <- data.frame(true_location=sample_data$z_t,
                            observed_location=sample_data$x_t,
                            estimated_location = rowMeans(particles_sd_emis_5, na.rm = TRUE),
                            index = 1:NROW(particles_sd_emis_5)) %>% as.data.frame()

plot_location(step=100, df=location_sd_5)
```



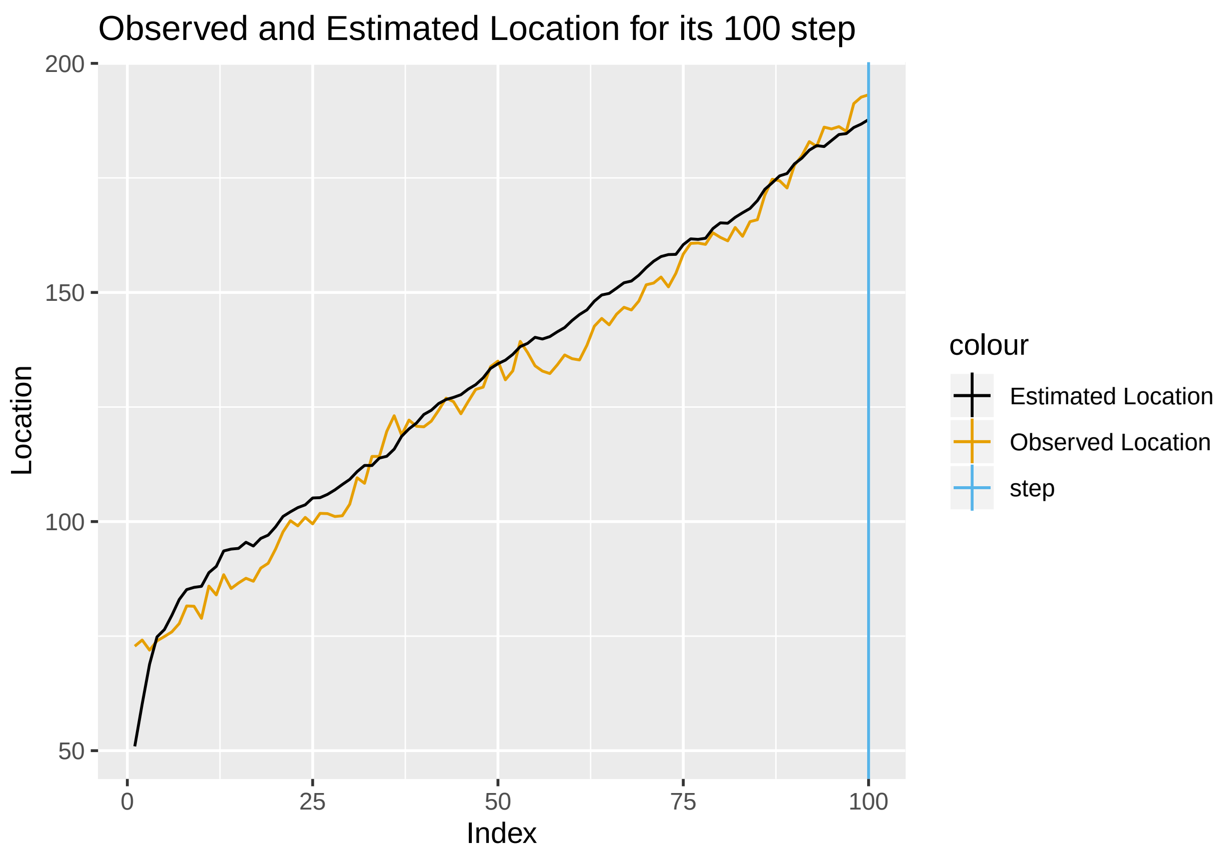
step	true_location	observed_location	estimated_location
100	191.4443	193.1422	191.4268

```
plot_location(step=50, df=location_sd_5)
```



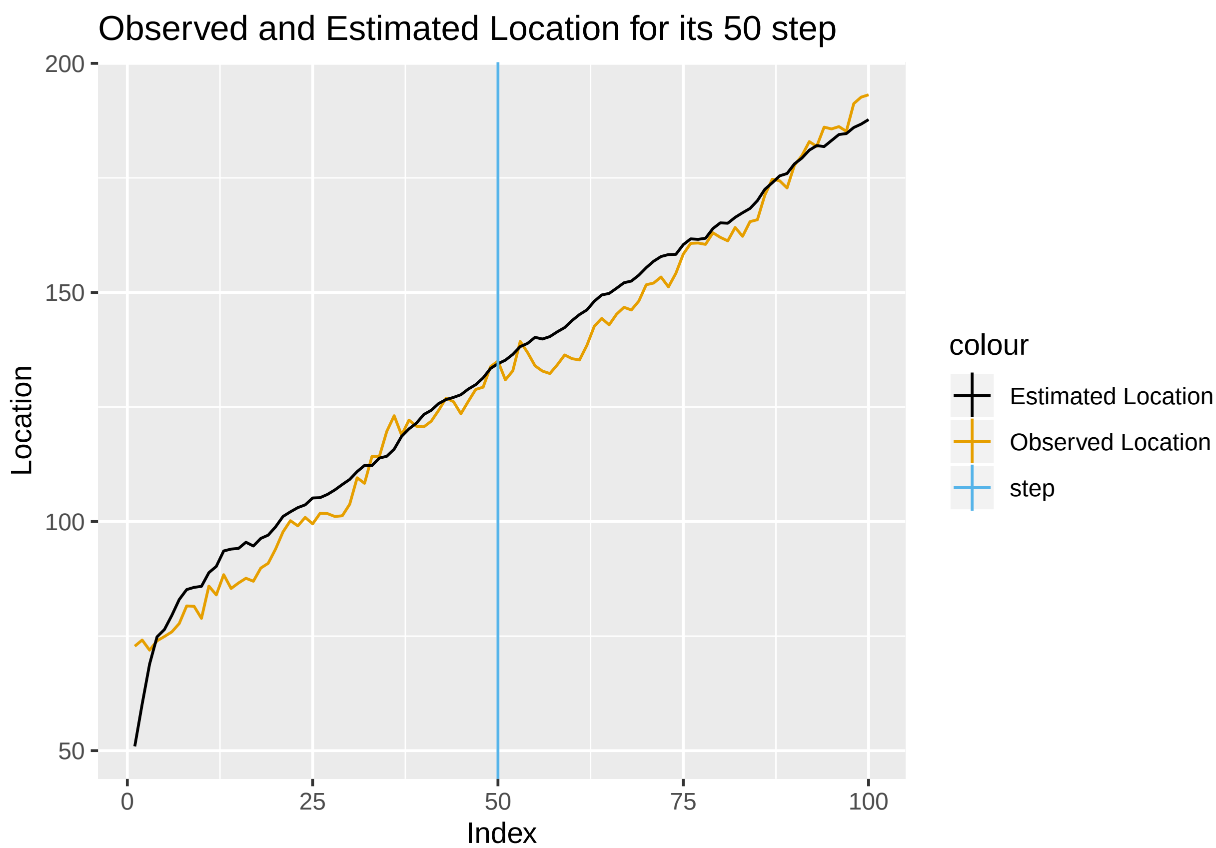
step	true_location	observed_location	estimated_location
50	133.9368	135.0235	133.8438

```
set.seed(12345)
# with standard deviation of the emission model of 50
particles_sd_emis_50 <- my_particle_filter(M=100, Times = 100, obs=sample_data$x_t,
                                sd_tra=1, sd_emi=50, ignore_weight = FALSE)

location_sd_50 <- data.frame(true_location=sample_data$z_t,
                 observed_location=sample_data$x_t,
                 estimated_location = rowMeans(particles_sd_emis_50, na.rm = TRUE),
                 index = 1:NROW(particles_sd_emis_50)) %>% as.data.frame()

plot_location(step=100, df=location_sd_50)
```



step	true_location	observed_location	estimated_location
100	191.4443	193.1422	187.7288

```
plot_location(step=50, df=location_sd_50)
```



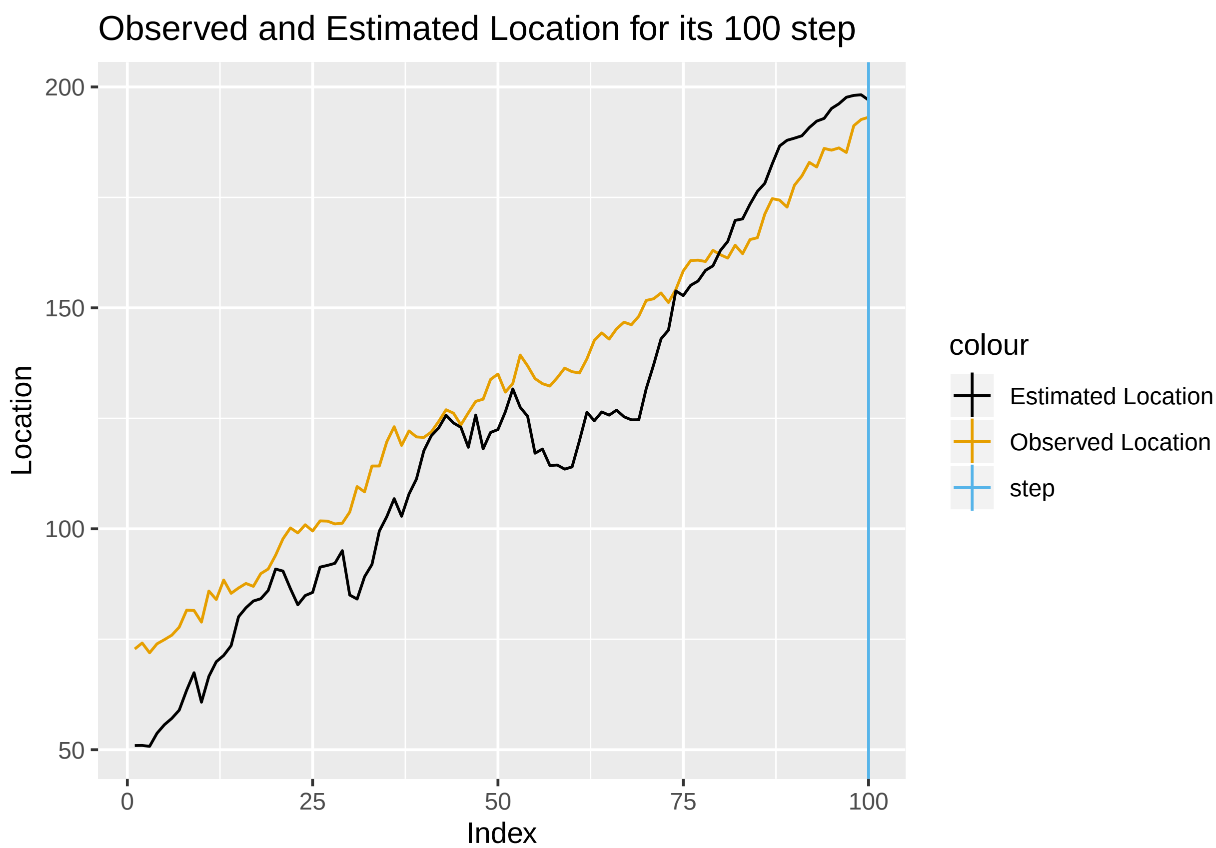
step	true_location	observed_location	estimated_location
50	133.9368	135.0235	134.446

Analysis: With higher standard deviation in emission model, there is greater spread of the particles thus higher uncertainity in for the detection of the robots location.

Questions 3

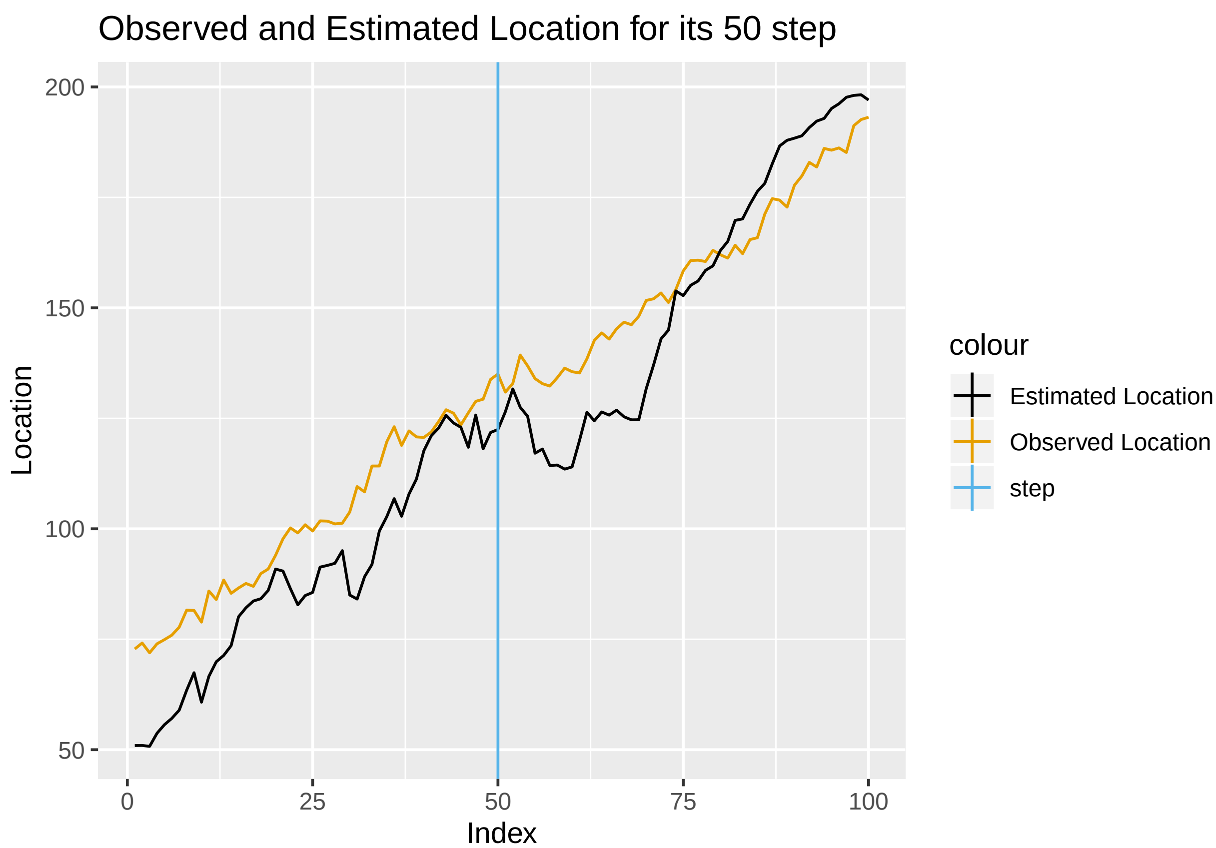
3) Finally, show and explain what happens when the weights in the particle filter are always equal to 1, i.e. there is no correction.

```
set.seed(12345)
# with standard deviation of the emission model of 50
particles_no_weight <- my_particle_filter(M=100, Times = 100, obs=sample_data$x_t,
                                          sd_tra=1, sd_emi=1, ignore_weight = TRUE)

df_no_weight <- data.frame(true_location=sample_data$z_t,
                 observed_location=sample_data$x_t,
                 estimated_location = rowMeans(particles_no_weight, na.rm = TRUE),
                 index = 1:NROW(particles_no_weight)) %>% as.data.frame()


plot_location(step=100, df=df_no_weight)
```



step	true_location	observed_location	estimated_location
100	191.4443	193.1422	197.0318

```
plot_location(step=50, df=df_no_weight)
```



step	true_location	observed_location	estimated_location
50	133.9368	135.0235	122.4591

Analysis: With no weights for the probability the whole resampling stage of the particle filter is defeated, thus there is no way for us to weed out the unlikely location and we are left with the as much as uncertainity with robots location as before.

Appendix

```
knitr::opts_chunk$set(echo = TRUE)
options(scipen=999)

library("tidyverse") #ggplot and dplyr
library("gridExtra") # combine plots
library("knitr") # for pdf
library("kableExtra") # For table formating

# The palette with black:
cbbPalette <- c("#000000", "#E69F00", "#56B4E9", "#009E73",
                "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
set.seed(12345)
set.seed(12345)
```

```r
gen_data <- function(N){
# states
z_t <- vector(length = N)
# observations
x_t <- vector(length = N)

# initial conditions
z_t[1] <- runif(n = 1, min = 0, max = N)

comp <- sample(1:3, prob = c(1 / 3, 1 / 3, 1 / 3), size = 1)
mu <- c(z_t[1], z_t[1] - 1, z_t[1] + 1)
x_t[1] <- rnorm(1, mean = mu[comp], 1)

for (i in 2:N) {
    comp <- sample(1:3, prob = c(1 / 3, 1 / 3, 1 / 3), size = 1)
    muz <- c(z_t[i - 1], z_t[i - 1] + 1, z_t[i - 1] + 2)
    z_t[i] <- rnorm(1, mean = muz[comp], 1)
    mu <- c(z_t[i], z_t[i] - 1, z_t[i] + 1)
    x_t[i] <- rnorm(1, mean = mu[comp], 1)
  }

sample_data <- data.frame(observation = x_t, states = z_t,
                          index=1:N)

# plot of samples
p1 <- ggplot(data=sample_data, aes(x=index)) +
  geom_line(aes(y=observation, color="Observations")) +
  geom_line(aes(y=states, color="True location")) +
        scale_colour_manual("", breaks = c("Observations", "True location"),
                        values = c("#000000", "#E69F00", "#56B4E9", "#009E73")) +
  xlab("Index") +
  ylab("Location") +
  ggtitle("Observed and True location")

print(p1)

# density plots
p2 <- ggplot(data=sample_data) +
  geom_density(aes(x=observation, color="Observations")) +
  geom_density(aes(x=states, color="True location")) +
        scale_colour_manual("", breaks = c("Observations", "True location"),
                        values = c("#000000", "#E69F00", "#56B4E9", "#009E73")) +
    xlab("Index") +
  ylab("Density") +
  ggtitle("Desnity of Observed and True location")

print(p2)

return(list(z_t=z_t, x_t=x_t))
}

my_particle_filter <- function(M, Times, obs, sd_tra, sd_emi, ignore_weight) {
```

```r
  temp_particles <- NULL
  weights <- NULL
  # Create matrix that will contain all particles for each time step
  particles <- matrix(NA, nrow = Times, ncol = M)

  # Generate initial M particles, Uniform(0, 100)
  particles[1, ] <- runif(n = M, min = 0, max = 100)

  for (t in 2:Times) {
    for (m in 1:M) {

  # transiton matrix
    selection <- sample(1:3, prob = c(1 / 3, 1 / 3, 1 / 3), size = 1)
    mean_tra <- c(particles[t - 1, m], particles[t - 1, m] + 1,
                  particles[t - 1, m] + 2)
    temp_particles[m] <- rnorm(n=1, mean = mean_tra[selection], sd = sd_tra)


  # emission matrix
    weights[m] <- mean(dnorm(x=obs[t], mean = temp_particles[m]-1, sd = sd_emi),
                       dnorm(x=obs[t], mean = temp_particles[m], sd = sd_emi),
                       dnorm(x=obs[t], mean = temp_particles[m]+1, sd = sd_emi))

    }
    weights = weights/sum(weights)

    if(ignore_weight == TRUE){
            particles[t, ] <- sample(x = temp_particles, size = M,
                                     replace = TRUE)
    }else{
      particles[t, ] <- sample(x = temp_particles, size = M, replace = TRUE,
                               prob = weights)}

  }

  return(particles)
}

set.seed(12345)
# generate data
sample_data <- gen_data(N=100)

# particle filter
particles <- my_particle_filter(M=100, Times = 100, obs=sample_data$x_t,
                                sd_tra=1, sd_emi=1, ignore_weight = FALSE)


location <- data.frame(true_location=sample_data$z_t,
                 observed_location=sample_data$x_t,
                 estimated_location = rowMeans(particles, na.rm = TRUE),
                 index = 1:NROW(particles)) %>% as.data.frame()
```

```r
# plot function
plot_location <- function(step, df){
plot1 <- ggplot(data=df, aes(x=index)) +
geom_line(aes(y=observed_location, color="Observed Location")) +
geom_line(aes(y=estimated_location, color="Estimated Location")) +
geom_vline(aes(xintercept=step, color="step")) +
  xlab("Index") +
  ylab("Location") +
      scale_colour_manual(values = c("#000000", "#E69F00", "#56B4E9", "#009E73")) +
ggtitle(paste0("Observed and Estimated Location for its ", step, " step"))

df <- df[step,]
result <- data.frame(step = step,
  true_location = df$true_location,
  observed_location = df$observed_location,
  estimated_location = df$estimated_location)

print(plot1)
kable(result, "latex", booktabs = T) %>% kable_styling(latex_options = "striped")
}

plot_location(step=1, df=location)
plot_location(step=100, df=location)
plot_location(step=50, df=location)
plot_location(step=75, df=location)


set.seed(12345)
# with standard deviation of the emission model of 5
particles_sd_emis_5 <- my_particle_filter(M=100, Times = 100, obs=sample_data$x_t,
                              sd_tra=1, sd_emi=5, ignore_weight = FALSE)

location_sd_5 <- data.frame(true_location=sample_data$z_t,
                observed_location=sample_data$x_t,
                estimated_location = rowMeans(particles_sd_emis_5, na.rm = TRUE),
                index = 1:NROW(particles_sd_emis_5)) %>% as.data.frame()

plot_location(step=100, df=location_sd_5)
plot_location(step=50, df=location_sd_5)
set.seed(12345)
# with standard deviation of the emission model of 50
particles_sd_emis_50 <- my_particle_filter(M=100, Times = 100, obs=sample_data$x_t,
                              sd_tra=1, sd_emi=50, ignore_weight = FALSE)

location_sd_50 <- data.frame(true_location=sample_data$z_t,
                observed_location=sample_data$x_t,
                estimated_location = rowMeans(particles_sd_emis_50, na.rm = TRUE),
                index = 1:NROW(particles_sd_emis_50)) %>% as.data.frame()

plot_location(step=100, df=location_sd_50)
plot_location(step=50, df=location_sd_50)
```

```r
set.seed(12345)
# with standard deviation of the emission model of 50
particles_no_weight <- my_particle_filter(M=100, Times = 100, obs=sample_data$x_t,
                                 sd_tra=1, sd_emi=1, ignore_weight = TRUE)

df_no_weight <- data.frame(true_location=sample_data$z_t,
                 observed_location=sample_data$x_t,
                 estimated_location = rowMeans(particles_no_weight, na.rm = TRUE),
                 index = 1:NROW(particles_no_weight)) %>% as.data.frame()


plot_location(step=100, df=df_no_weight)
plot_location(step=50, df=df_no_weight)
```