Follow Previous Case: Identify the layup

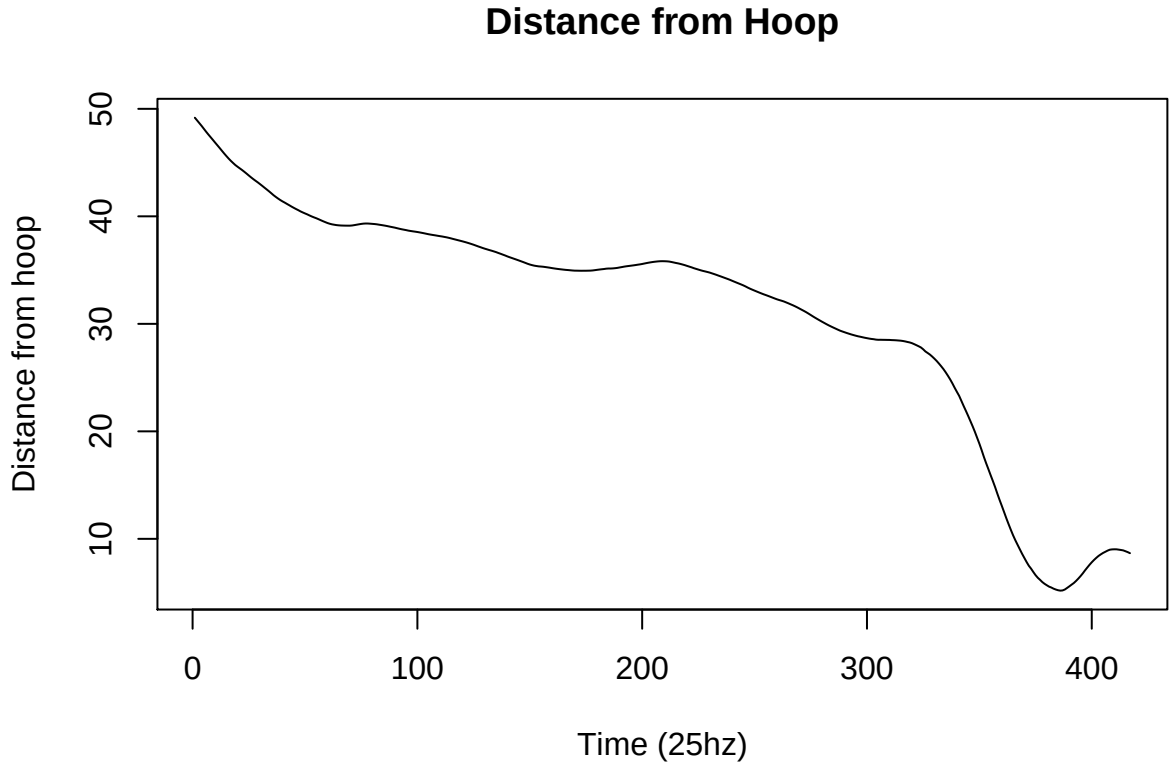
Tagging drive event

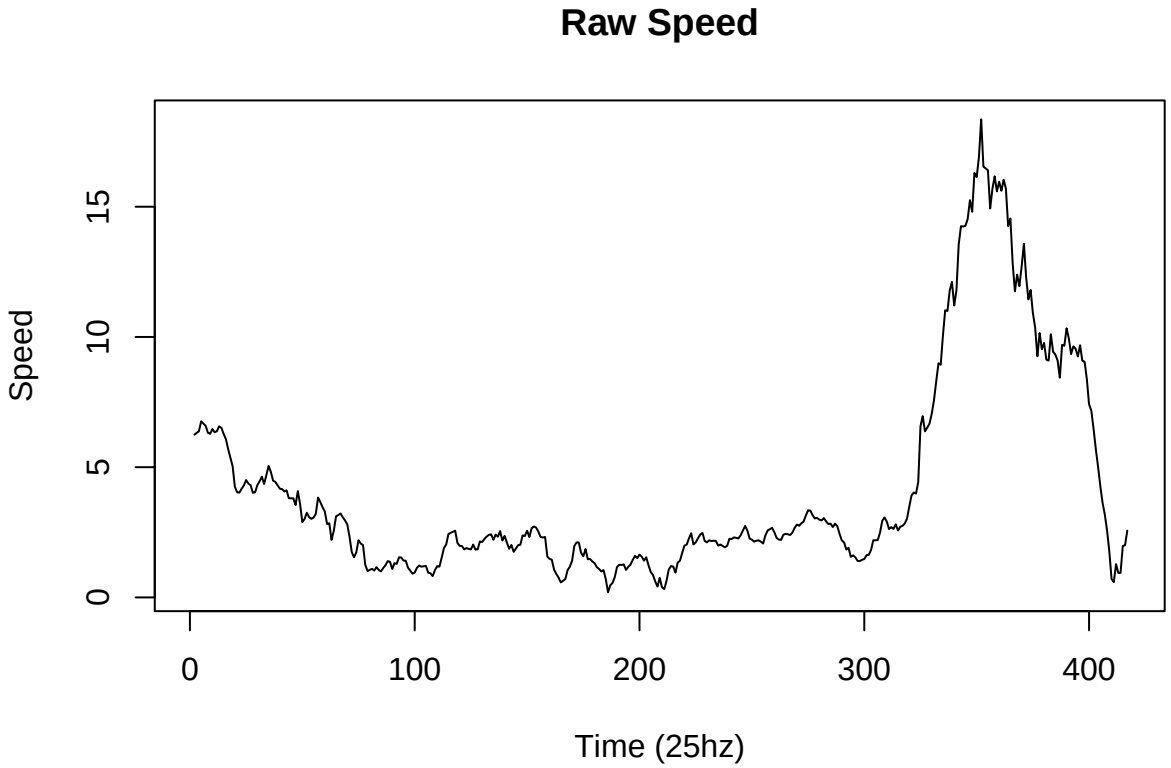
In the basketball, a drive occurs when an offensive player dribbles the ball towards the hoop for a shot attempt (layup).

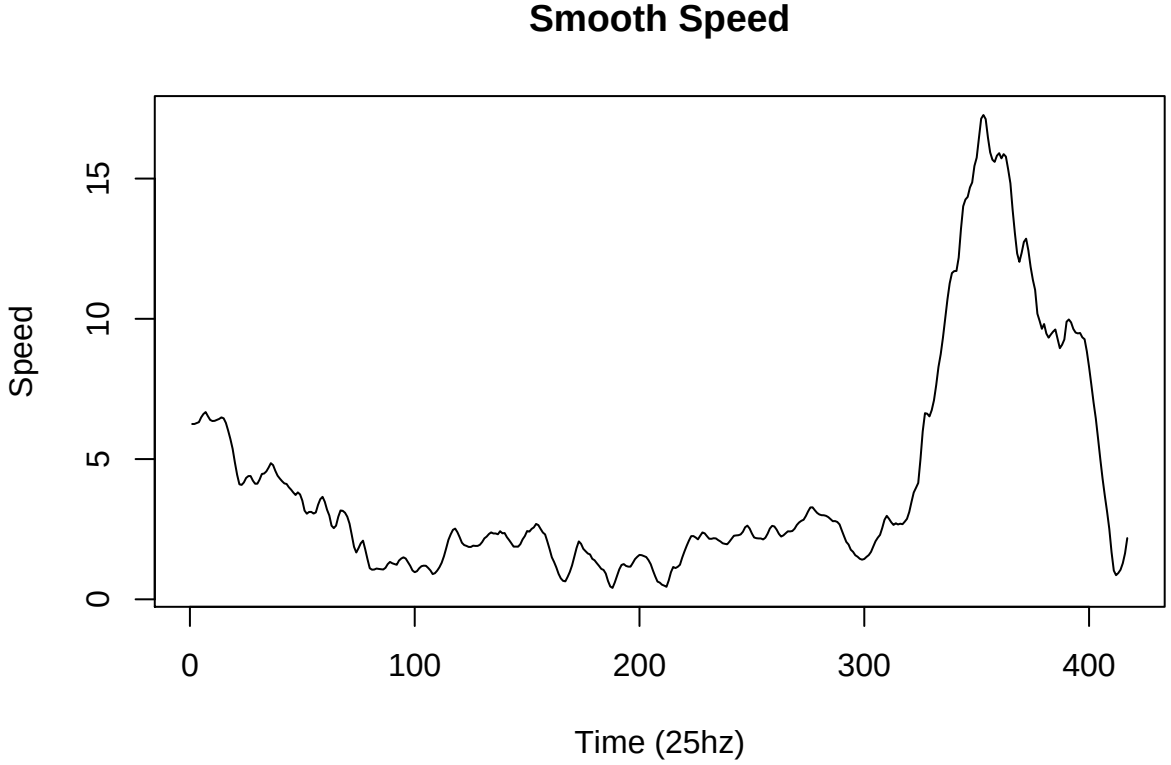
We define the indications of drive as follows:
1. The player increases their speed.
2. The player reduces the distances between himself and the basket.

Preprocess the data

Following the case study, we use the player tracking data we can construct the speed and distance metrics associated with LaVine's drive event.
Since the speed metric is pretty noisy, we smooth it by rolling mean with three time steps. In our example the data is not so noisy that it would affect.







Fit the data with 2-state Normal HMM

By observing the data, there is only one sharp decreasing trend for distance and one increasing trend for speed from around 300.

Thus, the $d_1 = 36.87$ and $d_2 = 16.37$, $v_1 = 2.44$ and $v_2 = 8.48$ are picked by the mean values of two time

intervals (0-300 and 300-400).

```
d1; d2; sd11; sd12
```

```
## [1] 36.87068
```

```
## [1] 16.36524
```

```
## [1] 4.322093
```

```
## [1] 9.192305
```

```
v1; v2; sd21; sd22
```

```
## [1] 2.443514
```

```
## [1] 8.480311
```

```
## [1] 1.400358
```

```
## [1] 4.950963
```

To specify the transition probability, we assume probability distribution between change from not drive to drive and change from drive to not drive can both be defined by exponential distribution. $\lambda_1 = \frac{1}{300} \approx 0.01$ and $\lambda_2 = \frac{1}{80} \approx 0.02$.

```
1/300 * exp(-1/300 * 300)
```

```
## [1] 0.001226265
```

```
1/80 * exp(-1/80 * 80)
```

```
## [1] 0.004598493
```

As for initial distribution, we just simply given that it would definitely start by not drive, i.e. $\delta = (1, 0)$.

In conclusion, the parameters are defined as below:

```
m <- 2
mu0 <- c(d1, d2)
sigma0 <- c(sd11, sd12)
mu1 <- c(v1, v2)
sigma1 <- c(sd21, sd22)
gamma <- matrix(c(0.99, 0.01, 0.98, 0.02), m, m, byrow = TRUE)
delta <- c(1, 0)
```

Fit 2-state HMM

```
defaultW <- getOption("warn")
options(warn = -1)
est_dist <- norm.HMM.fit(drive_data$game$lavine_dist, m, mu0, sigma0, gamma, stationary = TRUE)
est_speed <- norm.HMM.fit(lavine_speed_smooth, m, mu1, sigma1, gamma, stationary = TRUE)
SEdist <- norm.HMM.params_SE(drive_data$game$lavine_dist, 20, est_dist, stationary = TRUE)
SEspeed <- norm.HMM.params_SE(lavine_speed_smooth, 20, est_dist, stationary = TRUE)
```

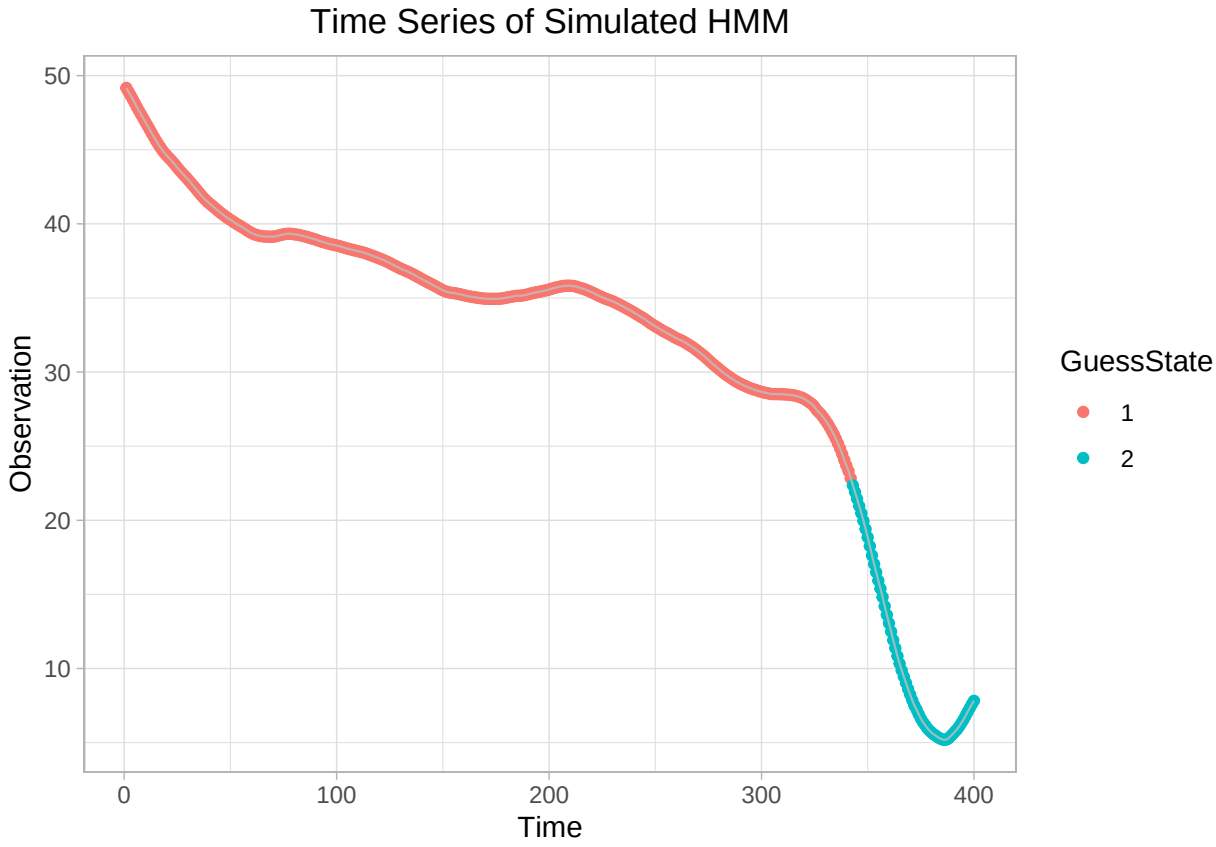
```
defaultW <- getOption("warn")
options(warn = -1)
range1 <- round(min(drive_data$game$lavine_dist)):round(max(drive_data$game$lavine_dist))
dist_ci_plot <- norm.HMM.CI_MonteCarlo(range1, m = 2, n = length(drive_data$game$lavine_dist), SEdist)
```

```
range2 <- round(min(lavine_speed_smooth)):round(max(lavine_speed_smooth))
speed_ci_plot <- norm.HMM.CI_MonteCarlo(range2, m = 2, n = length(lavine_speed_smooth), SEspeed)
```

Find the state sequence using global decoding by the Viterbi algorithm.

```
state_seq1 <- norm.HMM.viterbi(drive_data$game$lavine_dist, est_dist)
state_seq2 <- norm.HMM.viterbi(lavine_speed_smooth, est_speed)
est_dist$GuessState <- as.factor(state_seq1)
est_speed$GuessState <- as.factor(state_seq2)
```

```
data_dist <- data.frame(Time = c(1:length(drive_data$game$lavine_dist)),
                        Observation = drive_data$game$lavine_dist)
data_speed <- data.frame(Time = c(1:length(lavine_speed_smooth)),
                         Observation = lavine_speed_smooth)
data_dist$GuessState <- as.factor(state_seq1)
data_speed$GuessState <- as.factor(state_seq2)
# norm_hist_dist_CI(m, data_dist, est_dist, dist_ci_plot, width = 1)
# norm_hist_dist_CI(m, data_speed, est_speed, speed_ci_plot, width = 1)
timeseriesfit(head(data_dist, 400))
```




```
timeseriesfit(head(data_speed, 400))
```


Time Series of Simulated HMM
Observation
Time
0
5
10
15
0
100
200
300
400
GuessState
1
2