

# simulations

*Brian Masinde (brima748)*

*2/6/2020*

## Sensitivity Analysis Simulations

### Trzcinniczek *Acrocephalus scirpaceus* (Eurasian Reed Wabler)

From the snippet data provided it a sample of 2160 has mean mass of 0.01175 and fat mass of 0.00049. Assuming that these are uniformly distributed.

#### Uniformly distributed body mass and fat mass

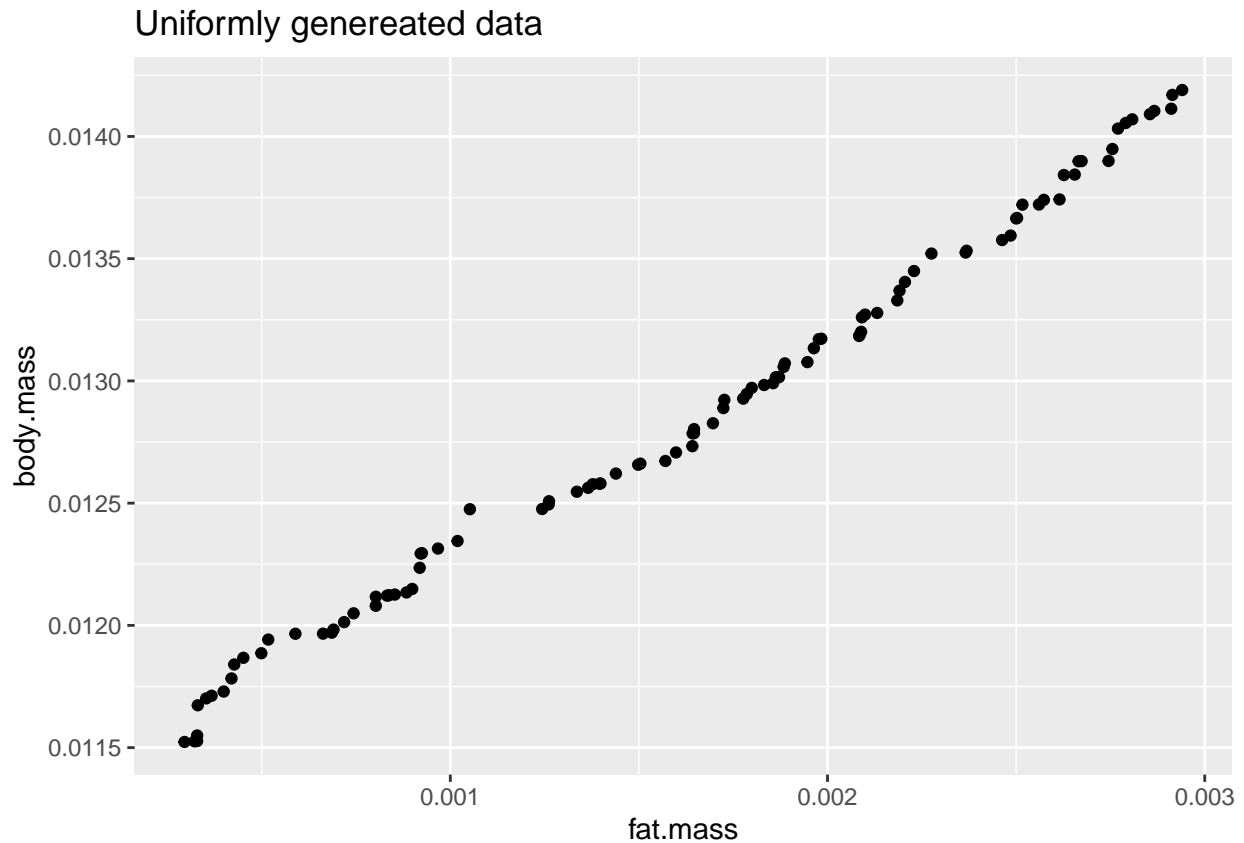
```
set.seed(2020)
acr_irp_body_mass <- sort(runif(100, min = 0.01152, max = 0.01421),
                          decreasing = FALSE)

acr_irp_fat_mass <- sort(runif(100, min = 0.00025138, max = 0.002942),
                        decreasing = FALSE)

acr_sim_data <- data.frame(cbind("body mass" = acr_irp_body_mass,
                                "fat mass" = acr_irp_fat_mass))

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.5.2
ggplot(acr_sim_data, aes(x = fat.mass, y = body.mass)) +
  geom_point() +
  ggtitle("Uniformly generated data")
```



```

wing.span <- rep(0.1946, 100)
wing.area <- rep(0.00773, 100)
taxon <- rep(1, 100)
muscle.mass <- 0.09 * acr_irp_body_mass

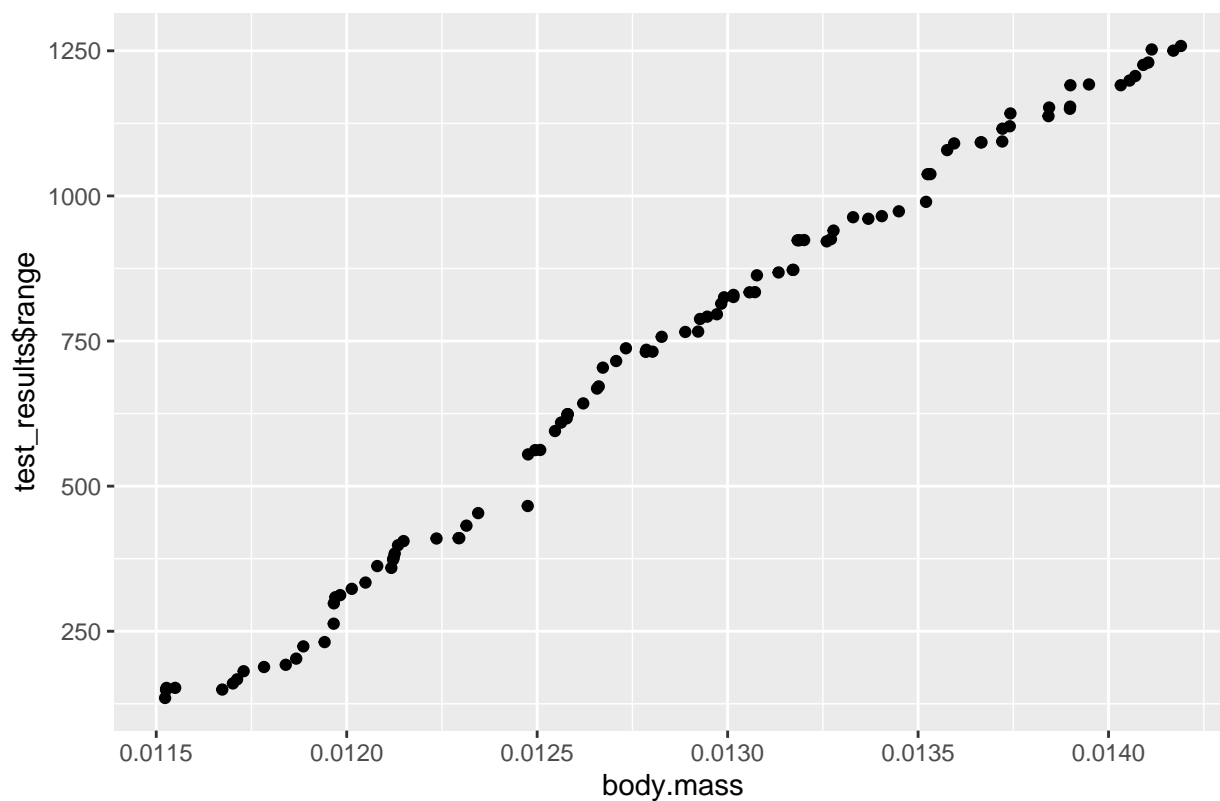
data <- data.frame(cbind(acr_irp_body_mass, acr_irp_fat_mass, wing.span,
                        wing.area, taxon, muscle.mass))

library(flying)
test_results <- migrate(data = data, method = "cmm", speed_control = "constant_speed")

ggplot(acr_sim_data, aes(x = body.mass, y = test_results$range)) +
  geom_point() +
  ggtitle("Uniformly generated data")

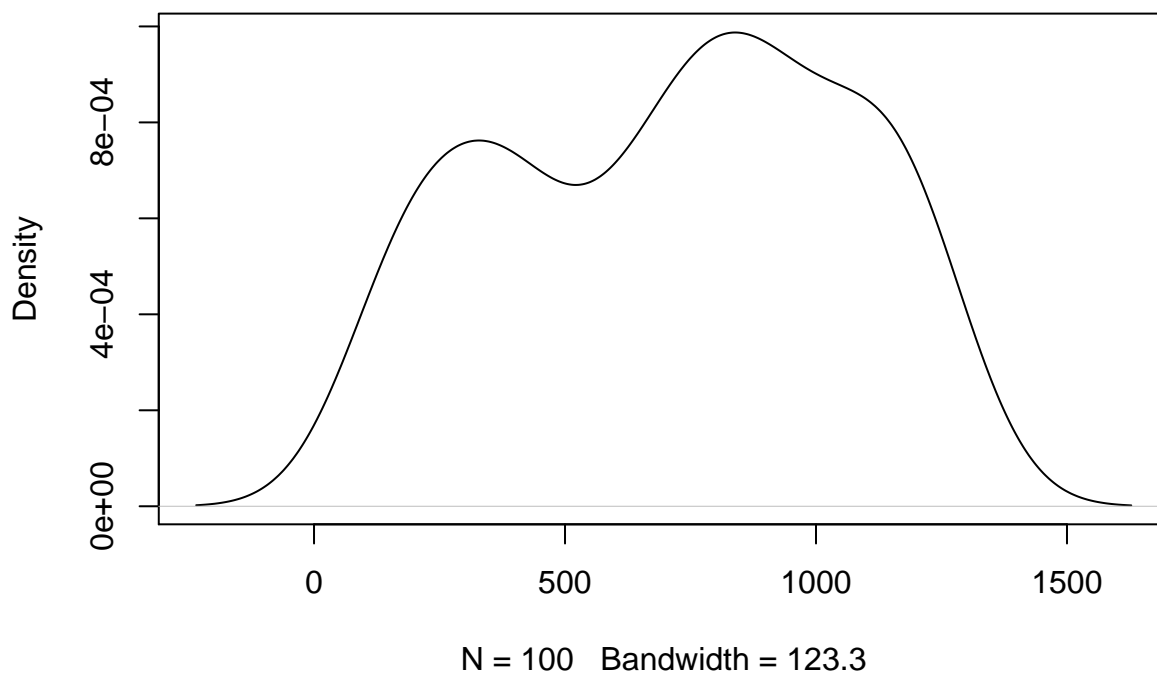
```

Uniformly generated data



```
plot(density(test_results$range), main = "Range density, uniform body mass and fat mass")
```

Range density, uniform body mass and fat mass



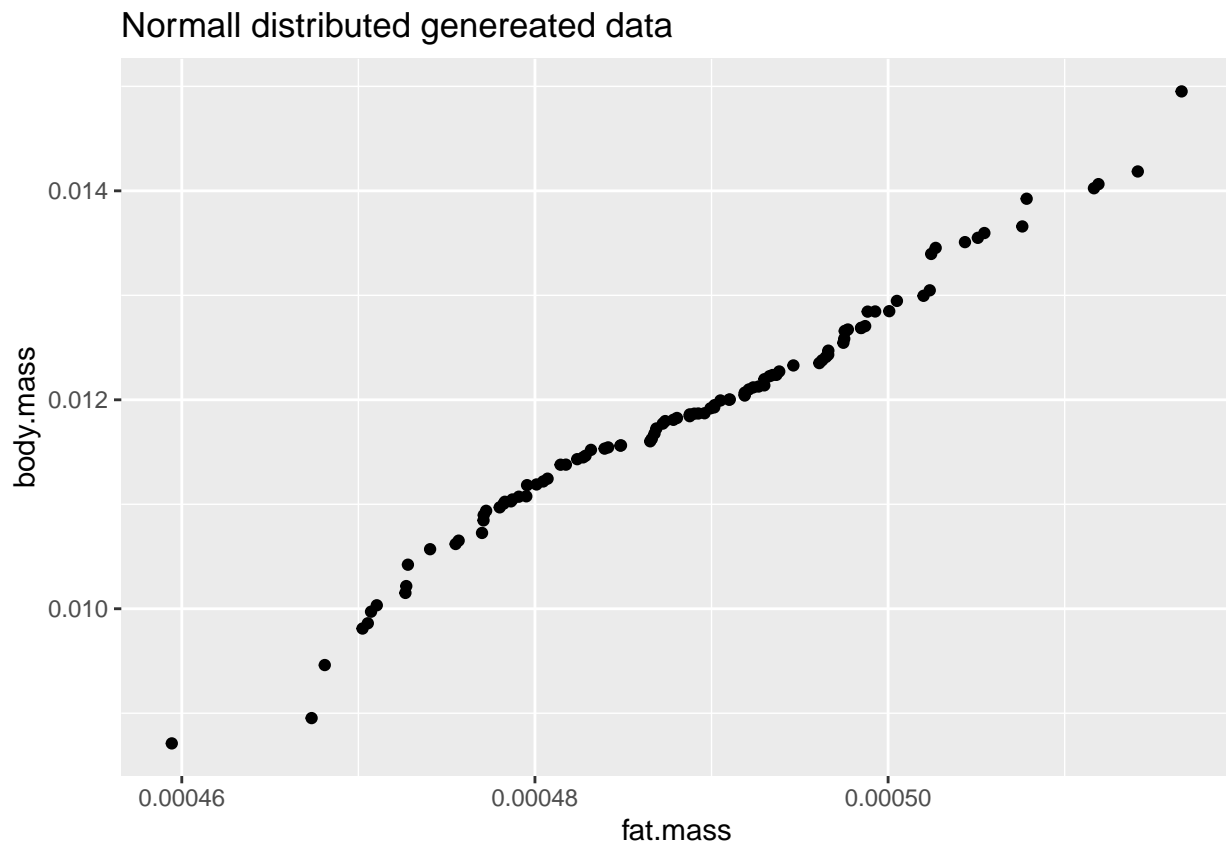
Gaussian body mass and fat mass.

```
# normally distributed data
set.seed(2020)
acr_irp_body_mass <- sort(rnorm(100, mean = 0.01175, sd = 0.001),
                          decreasing = FALSE)

acr_irp_fat_mass <- sort(rnorm(100, mean = 0.00049, sd = 0.00001),
                        decreasing = FALSE)

acr_sim_data <- data.frame(cbind("body mass" = acr_irp_body_mass,
                                "fat mass" = acr_irp_fat_mass))

ggplot(acr_sim_data, aes(x = fat.mass, y = body.mass)) +
  geom_point() +
  ggtitle("Normall distributed genereated data")
```



```
wing.span <- rep(0.1946, 100)
wing.area <- rep(0.00773, 100)
taxon <- rep(1, 100)
muscle.mass <- 0.09 * acr_irp_body_mass

data <- data.frame(cbind(acr_irp_body_mass, acr_irp_fat_mass, wing.span,
                        wing.area, taxon, muscle.mass))

library(flying)
test_results <- migrate(data = data, method = "cmm", speed_control = "constant_speed")
```

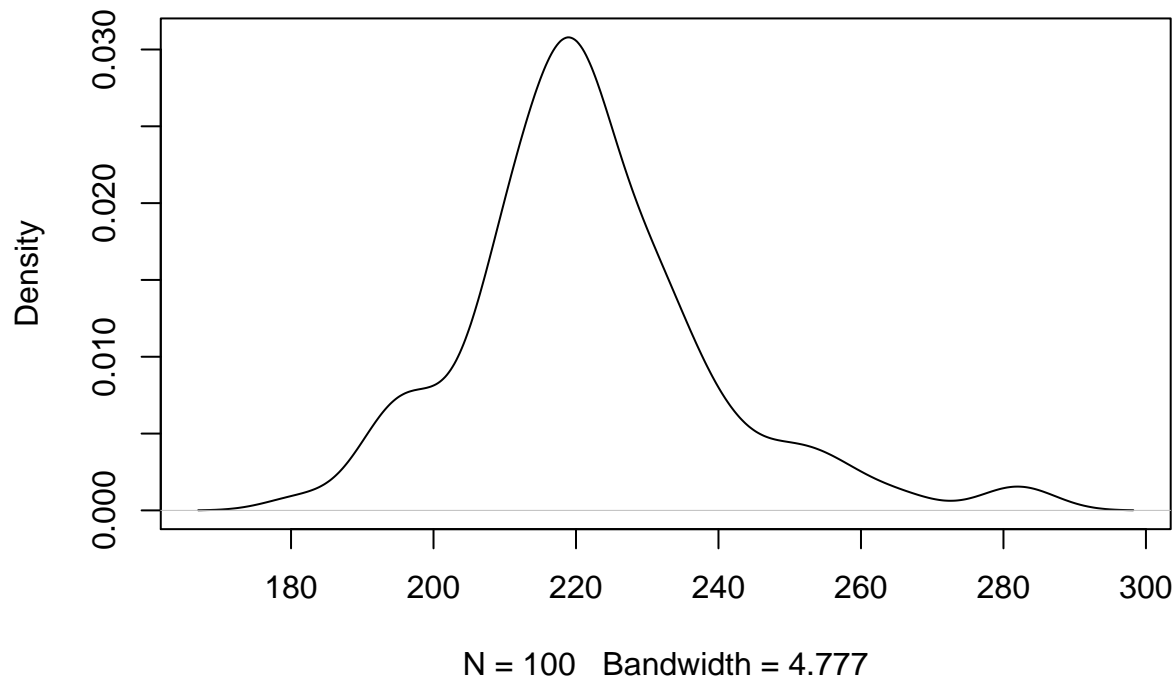
```
## Identifier column not found. Auto-gen
## ## settings not defined. Using default constants.
##
## Default airDensity = 1.00 kg m3
```

```
ggplot(acr_sim_data, aes(x = body.mass, y = test_results$range)) +
  geom_point() +
  ggtitle("Range vs body mass")
```



```
plot(density(test_results$range), main = "Range density, gaussian body mass and fat mass")
```

## Range density, gaussian body mass and fat mass



Gaussian body mass and fat mass (with independence, no sorting)

```
# normally distributed data
set.seed(2020)
acr_irp_body_mass <- rnorm(100, mean = 0.01175, sd = 0.001)

acr_irp_fat_frac <- rnorm(100, mean = 0.04170213, sd = 0.00001)

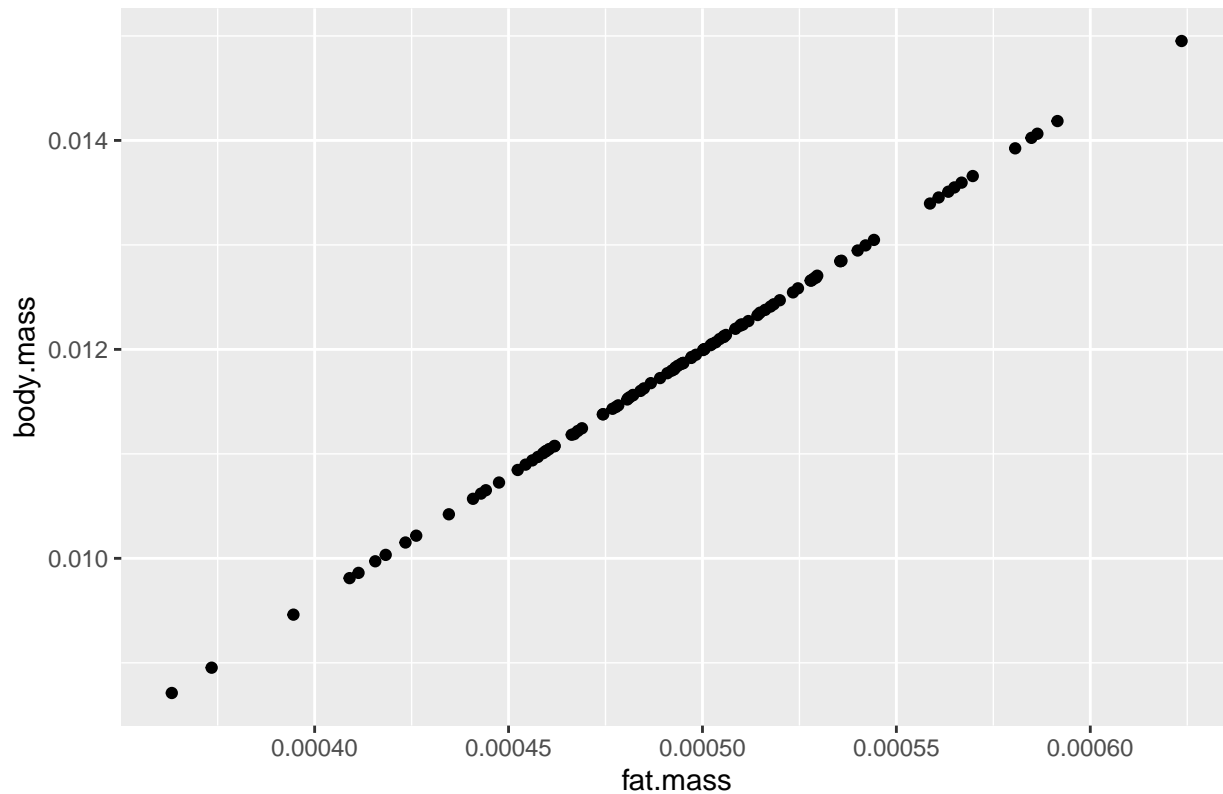
acr_irp_fat_mass <- acr_irp_body_mass * acr_irp_fat_frac

acr_sim_data <- data.frame(cbind("body mass" = acr_irp_body_mass,
                                "fat mass" = acr_irp_fat_mass))

data <- data.frame(cbind(acr_irp_body_mass, acr_irp_fat_mass, wing.span,
                        wing.area, taxon, muscle.mass))

ggplot(acr_sim_data, aes(x = fat.mass, y = body.mass)) +
  geom_point() +
  ggtitle("Normall distributed genereated data")
```

Normal distributed generated data



```
test_results <- migrate(data = data, method = "cmm", speed_control = "constant_speed")
```

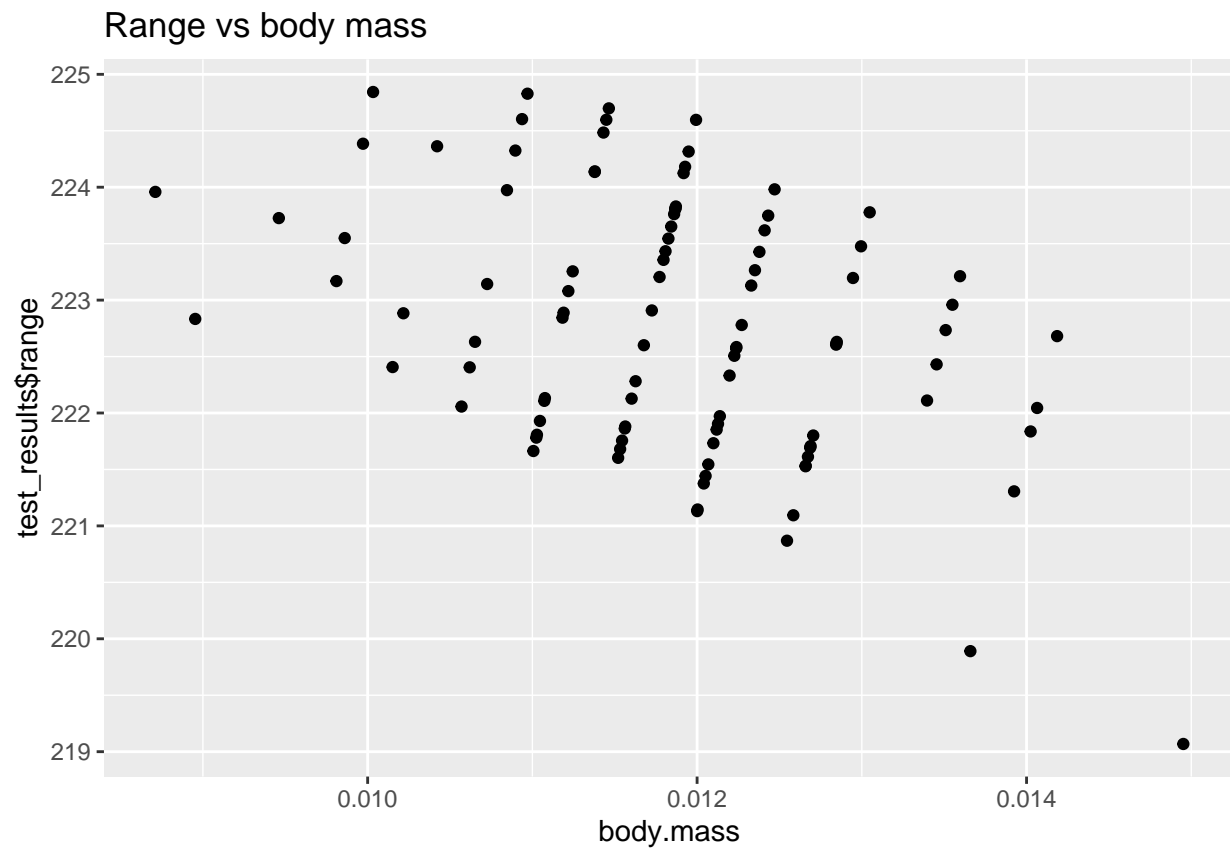
```
## Identifier column not found. Auto-gen
```

```
## ## settings not defined. Using default constants.
```

```
##
```

```
## Default airDensity = 1.00 kg m3
```

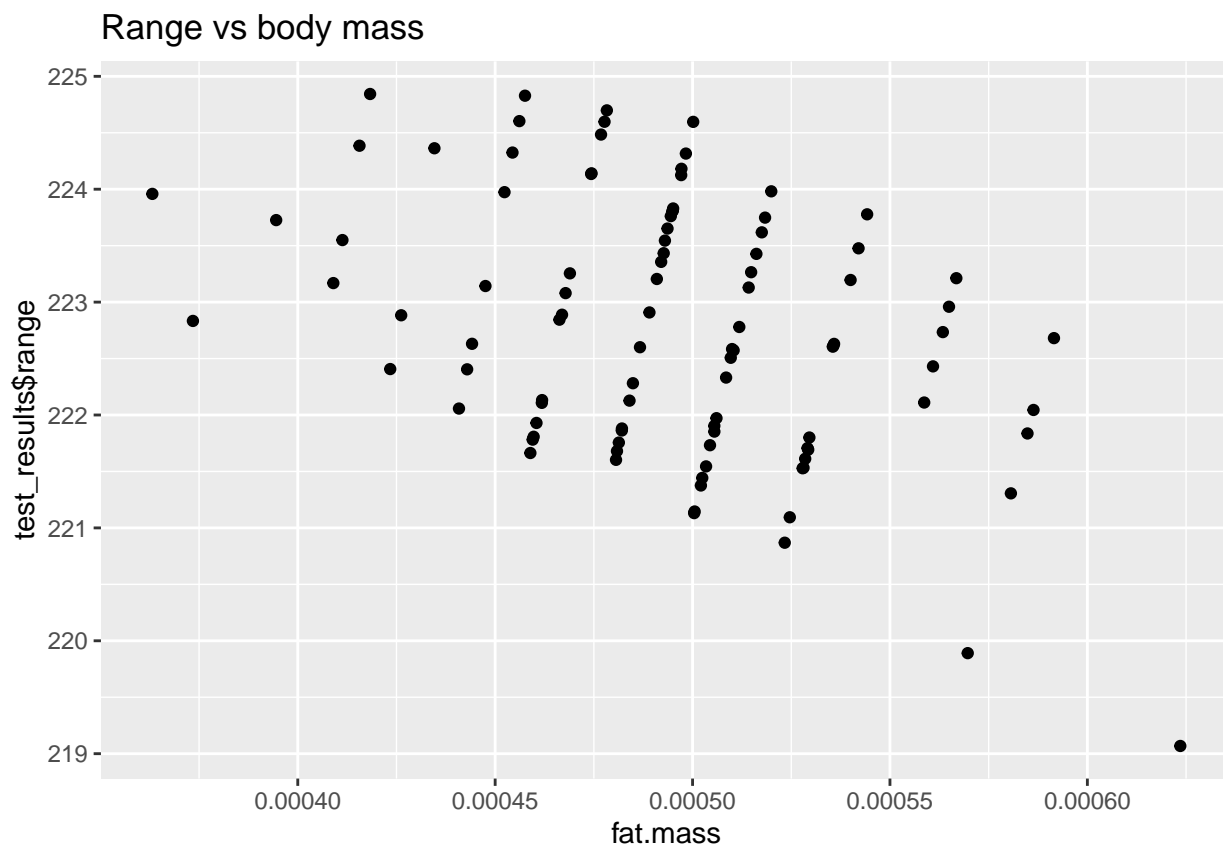
```
ggplot(acr_sim_data, aes(x = body.mass, y = test_results$range)) +  
  geom_point() +  
  ggtitle("Range vs body mass")
```



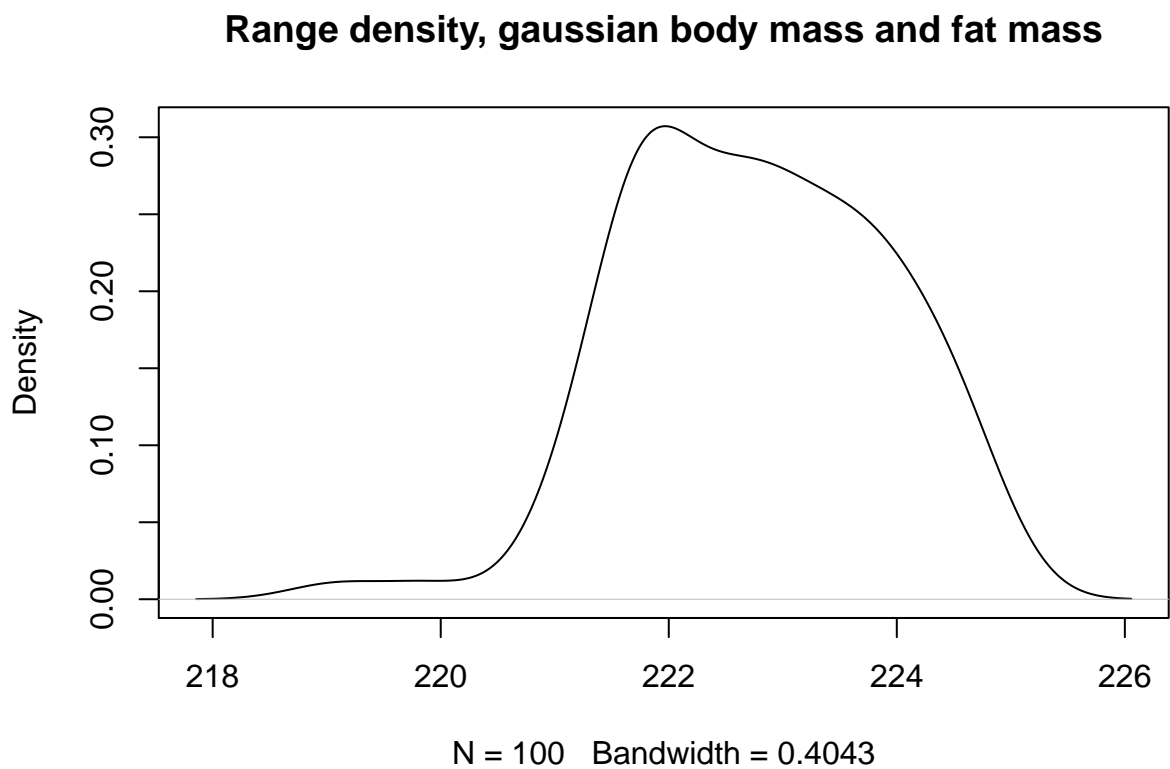
```
test_results <- migrate(data = data, method = "cmm", speed_control = "constant_speed")

ggplot(acr_sim_data, aes(x = fat.mass, y = test_results$range)) +
  geom_point() +
  ggtitle("Range vs body mass")
```





```
plot(density(test_results$range), main = "Range density, gaussian body mass and fat mass")
```

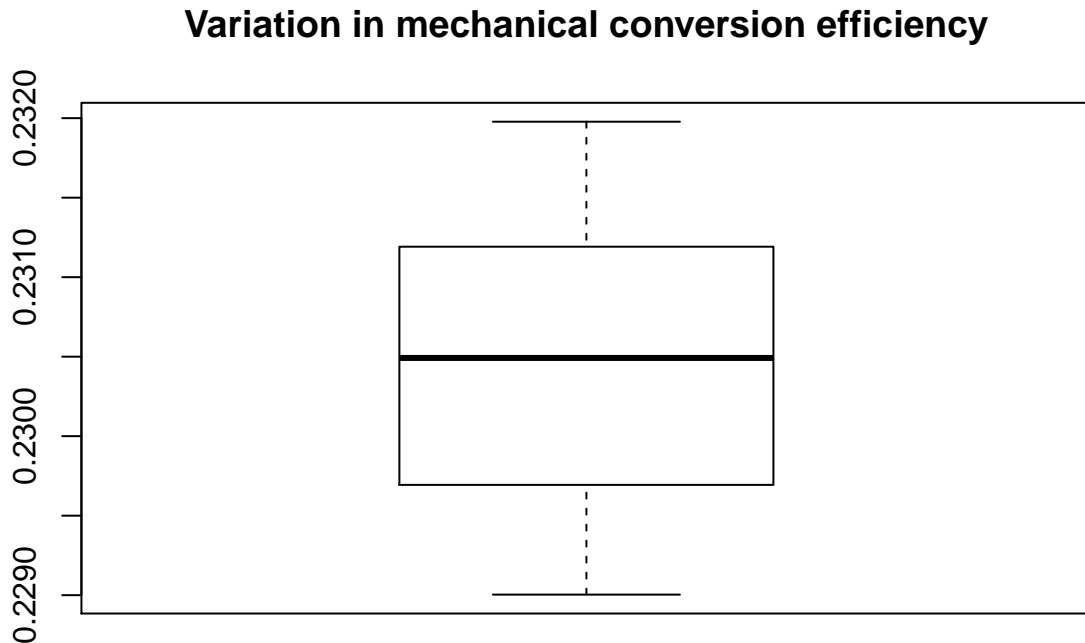


Uncertainty in mechanical conversion efficiency.

Default value in flight is 0.23

```
# uniform prior
set.seed(2020)
mce <- runif(100, min = 0.229, max = 0.232)

boxplot(mce, main = "Variation in mechanical conversion efficiency")
```



Fat mass and muscle mass as constants

```
body.mass <- 0.01175
fat.mass <- 0.00049
muscle.mass <- body.mass * 0.17
wing.span <- 0.1946
wing.area <- 0.00773
taxon <- 1
```

```
data <- data.frame(cbind(body.mass, fat.mass, muscle.mass, wing.span, wing.area, taxon))
```

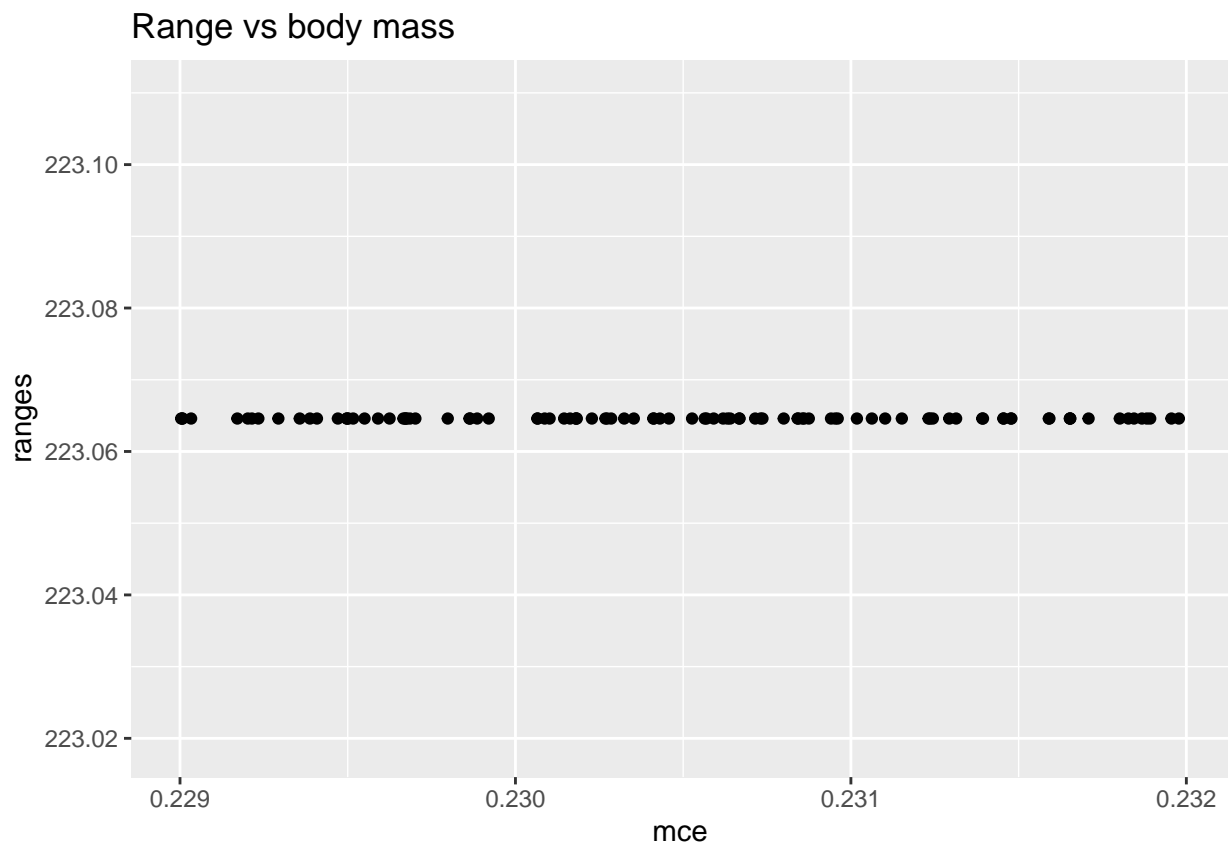
```
res <- list()
for (i in 1:length(mce)) {
  res[[i]] <- migrate(data = data, method = "cmm", speed_control = "constant_speed",
    settings = list(mce = mce[i]) )
}
```

```
ranges <- c()
```

```
for (i in 1:length(mce)) {
  ranges[i] <- res[[i]][[1]]
}
```

```
ggplot() +
  geom_point(aes(x = mce, y = ranges)) +
```

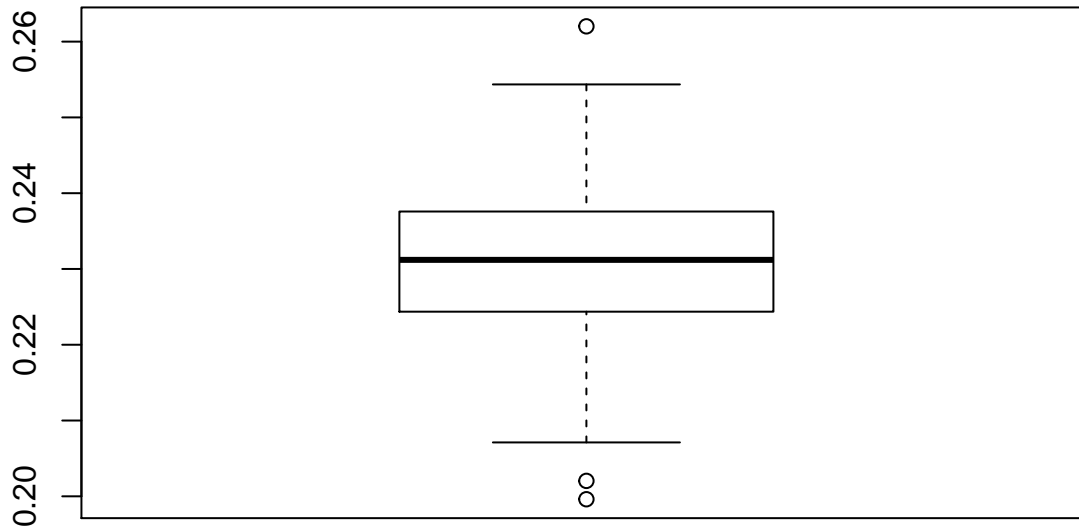
```
ggtitle("Range vs body mass")
```



#### Larger variation in mce

```
# uniform prior  
set.seed(2020)  
mce <- rnorm(100, mean = 0.23, sd = 0.01)  
  
boxplot(mce, main = "Variation in mechanical conversion efficiency")
```

## Variation in mechanical conversion efficiency



```
body.mass <- 0.01175
fat.mass <- 0.00049
muscle.mass <- body.mass * 0.17
wing.span <- 0.1946
wing.area <- 0.00773
taxon <- 1

data <- data.frame(cbind(body.mass, fat.mass, muscle.mass, wing.span, wing.area, taxon))

res <- list()
for (i in 1:length(mce)) {
  res[[i]] <- migrate(data = data, method = "cmm", speed_control = "constant_speed",
                      settings = list(mce = mce[i]) )
}

ranges <- c()

for (i in 1:length(mce)) {
  ranges[i] <- res[[i]][[1]]
}

ggplot() +
  geom_point(aes(x = mce, y = ranges)) +
  ggtitle("Range vs mechanical conversion efficiency")
```

