

Andropogonae Dispersal Trait Sensitivity Analysis

Binbin Wang, Elizabeth Kellogg, Kurt Petersen, Huijie Wu, Lauren Sullivan

Last updated: 2025-02-17

Contents

Simulation Sensitivity Analysis	1
Simulation	1
Graph Results	3
Simulation with our TV data	5
Graph Data Results	7

Simulation Sensitivity Analysis

This is to determine how many seeds we need to drop to reach the mean and variance calculated from dropping 20 seeds (3 reps per seed), assuming that dropping 20 seeds will give you the true mean.

```
# PROLOG #####

# PROJECT: Andropogonae_dispersaltraits
# PURPOSE: Determine the number of drops needed for each species.
# AUTHOR: Lauren Sullivan (llsull@msu.edu)
# COLLAB: Toby Kellogg, Kurt Petersen, Binbin Wang, Huijie Wu
# CREATED: 7 April 2021
# LATEST: 8 Oct 2024
# NOTES:
#
# FILES: "terminal_velocity.csv" - tv and mass data for all dropped seeds
#
#

### Set working directory to "source file location" to make sure code works!

# PROLOG #####
```

Simulation

First we looked to see that simulated values for the data would give us what we expected.

```
## Set up dataset
set.seed(4)

species <- rep("spp1", 60)
id <- c(seq(1,20,1), seq(1,20,1), seq(1,20,1))
rep <- c(rep(1,20), rep(2,20), rep(3,20))
tv <- c(rnorm(60, 2.5, 0.25))
```

```

dat <- data.frame(Species = species,
                  ID = id,
                  Rep = rep,
                  TV = tv)
dat <- dat %>%
  arrange(Species, ID, Rep)

## get mean TV based on 3 reps of each seed
dat_mean <- dat %>%
  group_by(Species, ID) %>%
  summarize(mean_tv = mean(TV))

## `summarise()` has grouped output by 'Species'. You can override using the
## `.groups` argument.

grand_mean <- mean(dat_mean$mean_tv)
grand_sd <- sd(dat_mean$mean_tv)

## draw n individuals 1000 times, store and see how close you get to the grand mean.
reps <- 1000
n_draws <- seq(2,20,1)

results_all <- tibble(
  no_draws = numeric(),
  no_sims = numeric(),
  mean_tv = numeric(),
  diff = numeric()
)

for(ii in 1:length(n_draws-1)){

  #make a blank tibble
  results_temp <- tibble(no_draws = rep(0, reps),
                        no_sim = rep(0, reps),
                        mean_tv = rep(0, reps),
                        mean_diff = rep(0, reps)
                        )

  for(jj in 1:reps){

    #get a mean from the sample
    sample_mean <- mean(sample(dat_mean$mean_tv, size = n_draws[ii]), replace = FALSE)

    #fill the tibble
    results_temp[jj,1] <- n_draws[ii]
    results_temp[jj,2] <- jj
    results_temp[jj,3] <- sample_mean
    results_temp[jj,4] <- abs(grand_mean - sample_mean)
  }
}

```

```

}
  #rbind to total results table
  results_all <- rbind(results_all, results_temp)
}

```

Graph Results

These are what the graphed outputs should look like if we're getting our data to do what we wanted it to do.

```

stats_sum <- results_all %>%
  group_by(no_draws) %>%
  summarize(tv_m = mean(mean_tv),
            tv_sd = sd(mean_tv),
            diff_m = mean(mean_diff),
            diff_sd = sd(mean_diff))

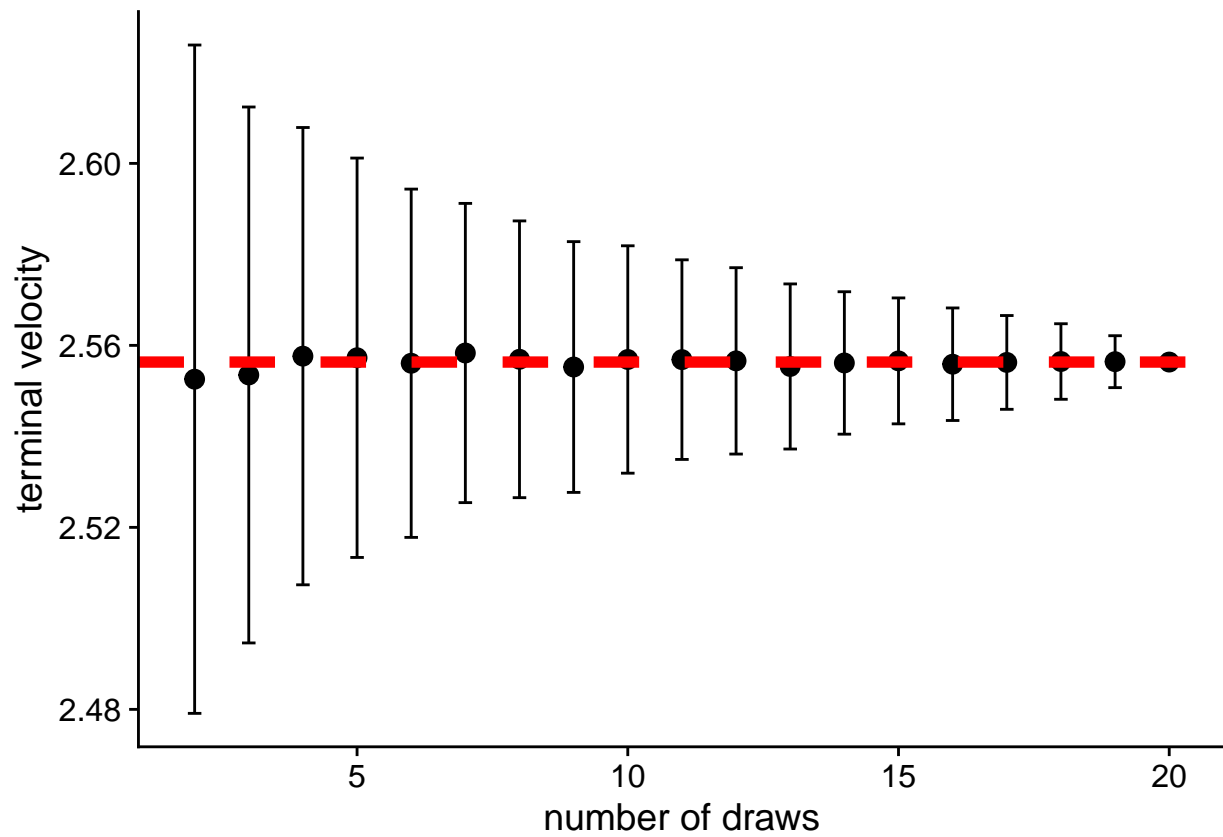
# plot of means and errors
ggplot(stats_sum, aes(x = no_draws, y = tv_m))+
  geom_point(cex = 3)+
  geom_errorbar(aes(x = no_draws, ymin = tv_m - tv_sd, ymax = tv_m + tv_sd), width = 0.25) +
  geom_hline(yintercept=grand_mean, linetype="dashed", color = "red", size=2)+
  #geom_hline(yintercept=grand_mean+grand_sd, linetype="solid", color = "red", size=1)+
  #geom_hline(yintercept=grand_mean-grand_sd, linetype="solid", color = "red", size=1)+
  labs(x = "number of draws", y = "terminal velocity", main = "Mean Terminal Velocity")

```

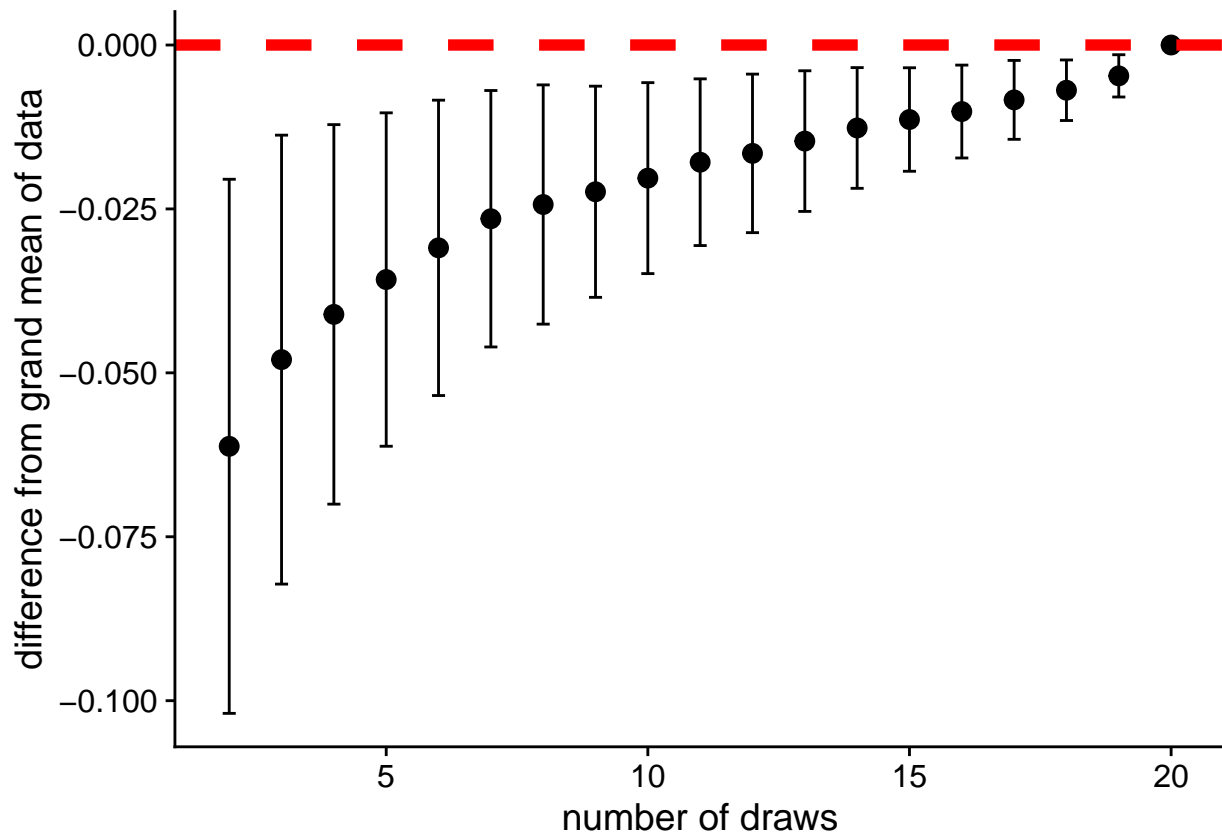
```

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```



```
# plot of differences and errors
ggplot(stats_sum, aes(x = no_draws, y = -diff_m)) +
  geom_point(cex = 3) +
  geom_errorbar(aes(x = no_draws, ymin = -diff_m - diff_sd, ymax = -diff_m + diff_sd), width = 0.25) +
  geom_hline(yintercept=0, linetype="dashed", color = "red", size=2) +
  labs(x = "number of draws", y = "difference from grand mean of data", main = "Difference from Grand M
```



Simulation with our TV data

Now we take the data we measured and sample it to determine the number of trials that is reasonable to estimate the actual terminal velocity given we assume that the mean of 20 drops gives the population mean.

```
## Set up dataset
set.seed(4)
seeddat <- read_csv("../data/terminal_velocity.csv")

#parameters
reps <- 100
n_individual <- seq(1, length(unique(seeddat$Individual)), 1)
n_individual <- length(unique(seeddat$Individual))
individ_name <- as.character(unique(seeddat$Individual))

results_all <- tibble(
  individual = character(),
  no_draws = numeric(),
  no_sims = numeric(),
  mean_tv = numeric(),
  diff = numeric(),
  gm = numeric()
)

for(jj in 1:n_individual){
  #jj = 1
```

```

#subset out the individuals
individ <- secdat %>%
  filter(Individual == individ_name[[jj]])
#glimpse(individ)

#define the number of draws because they all have different numbers of runs
n_draws <- seq(2,nrow(individ),1)

#calculate grand mean and sd per individual
gm <- mean(individ$Vmean)
#grand_sd <- sd(individ$Vmean)

for(ii in 1:max(n_draws-1)){
  #ii = 1

  #make a blank tibble
  results_temp <- tibble(individual = as.character(rep(0, reps)),
    no_draws = rep(0, reps),
    no_sim = rep(0, reps),
    mean_tv = rep(0, reps),
    mean_diff = rep(0, reps),
    gm = rep(0, reps)
  )

  for(kk in 1:reps){
    # kk = 1

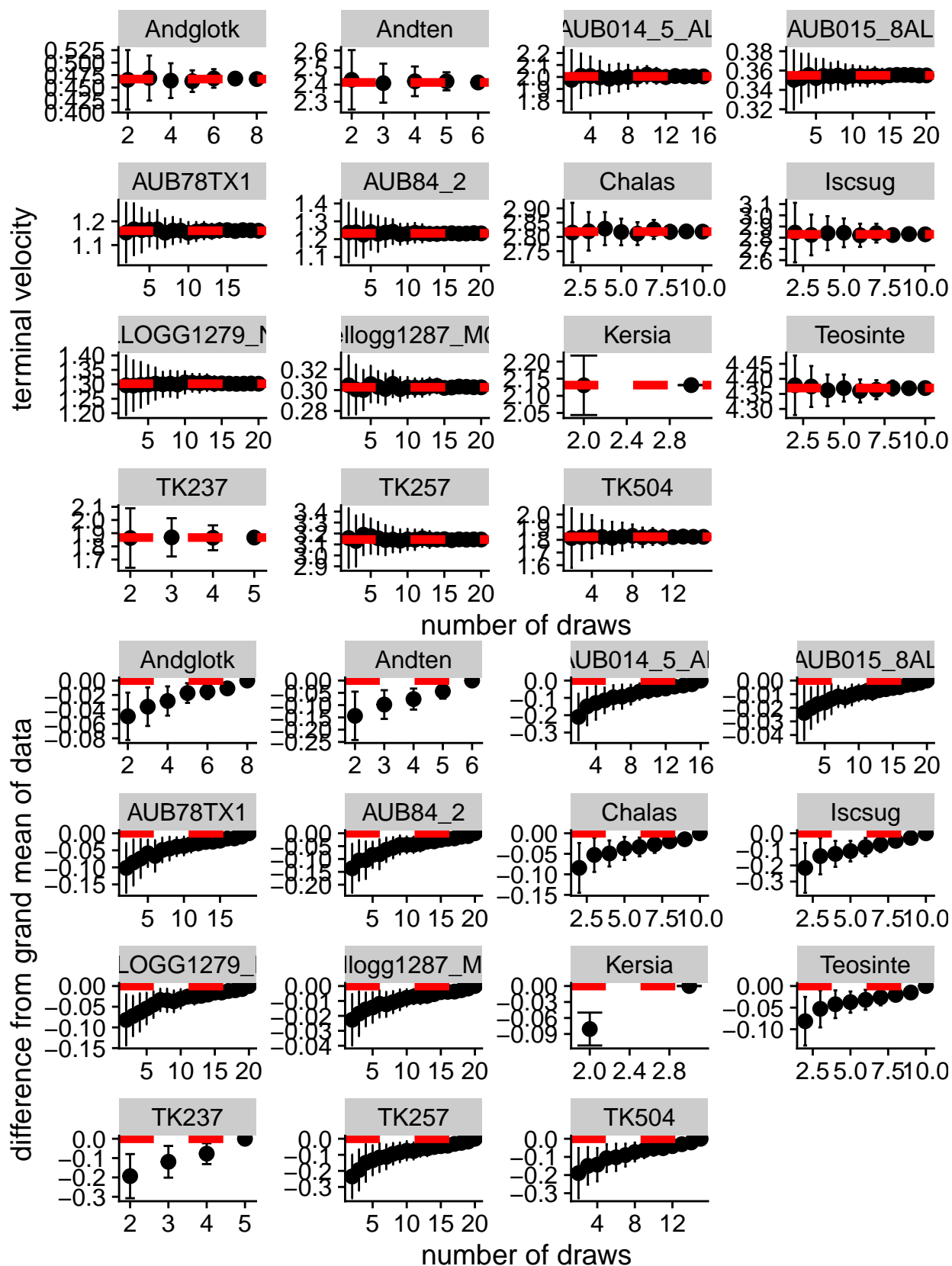
    #get a mean from the sample
    sample_mean <- mean(sample(individ$Vmean, size = n_draws[ii]), replace = FALSE)

    #fill the tibble
    results_temp[kk,1] <- unique(individ$Individual)
    results_temp[kk,2] <- n_draws[ii]
    results_temp[kk,3] <- kk
    results_temp[kk,4] <- sample_mean
    results_temp[kk,5] <- abs(gm - sample_mean)
    results_temp[kk,6] <- gm

  }
  #rbind to total results table
  results_all <- rbind(results_all, results_temp)
}
}

```

Graph Data Results



```

## Figures by Individual
for(jj in 1:n_individual){
  #jj = 5

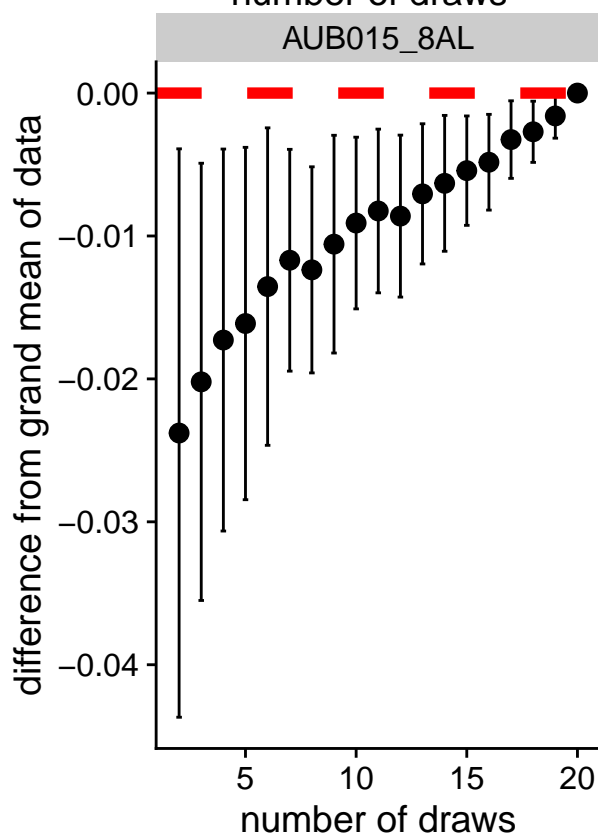
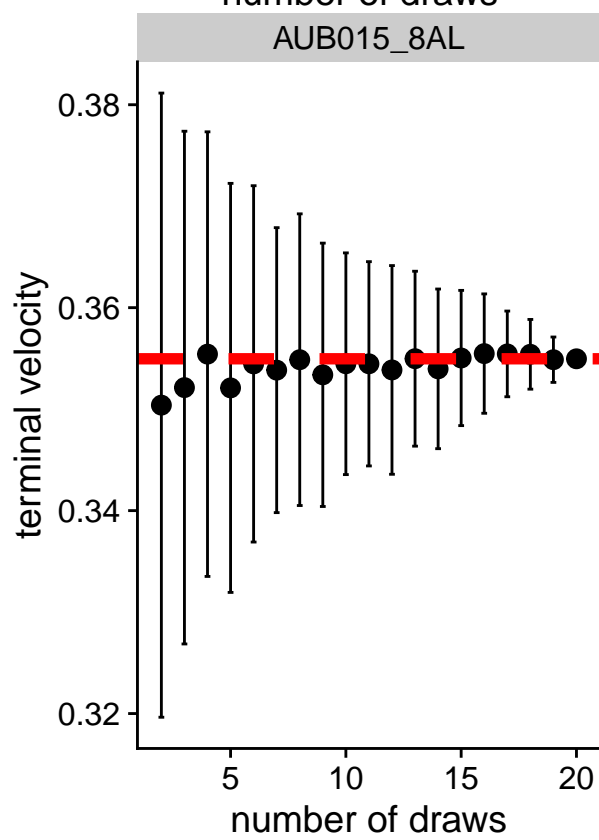
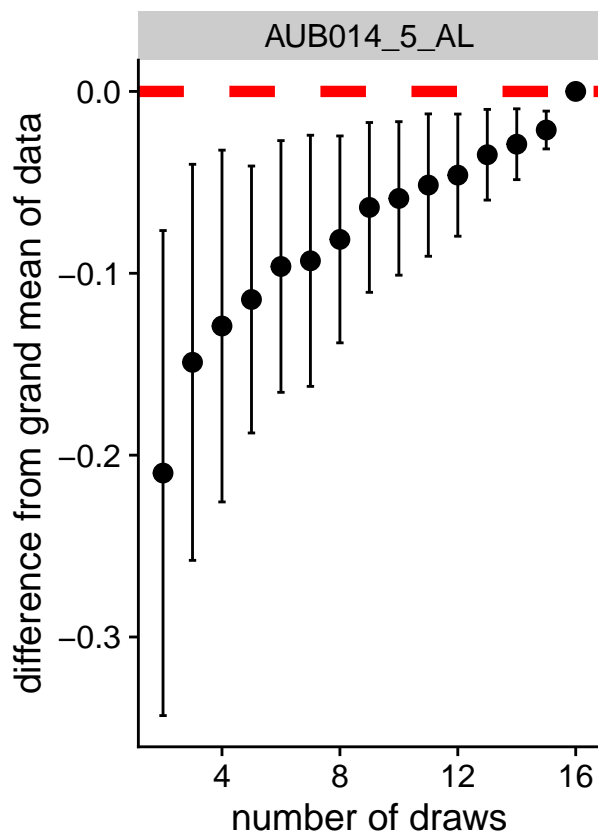
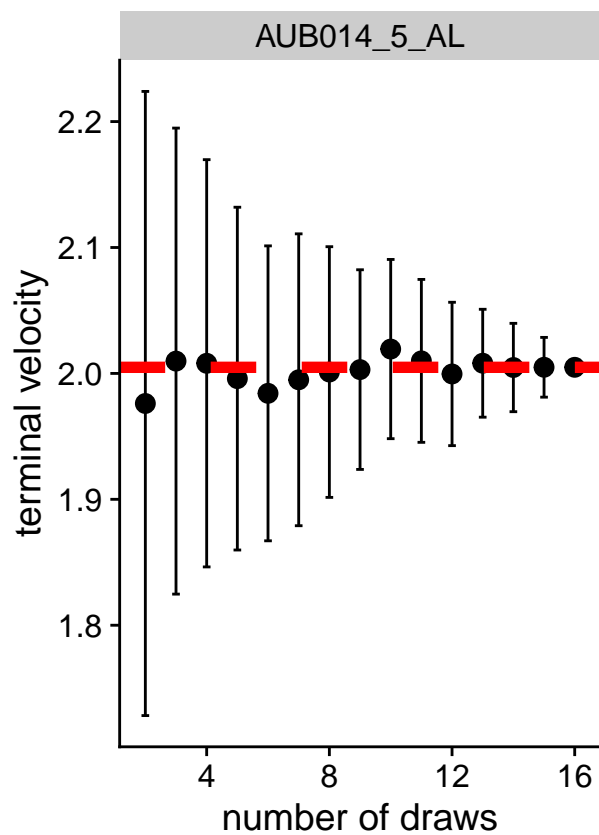
  #subset out the individuals
  individ <- stats_sum_data %>%
    filter(individual == individ_name[[jj]])

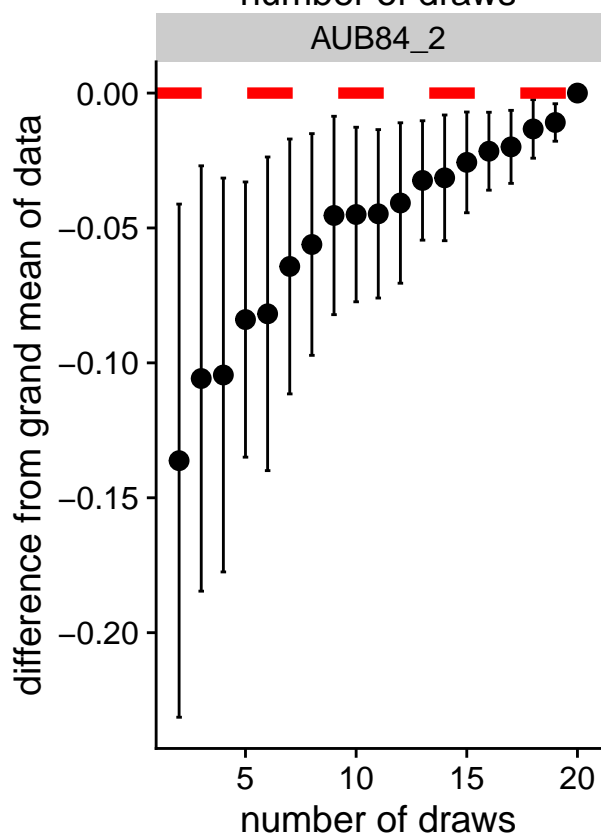
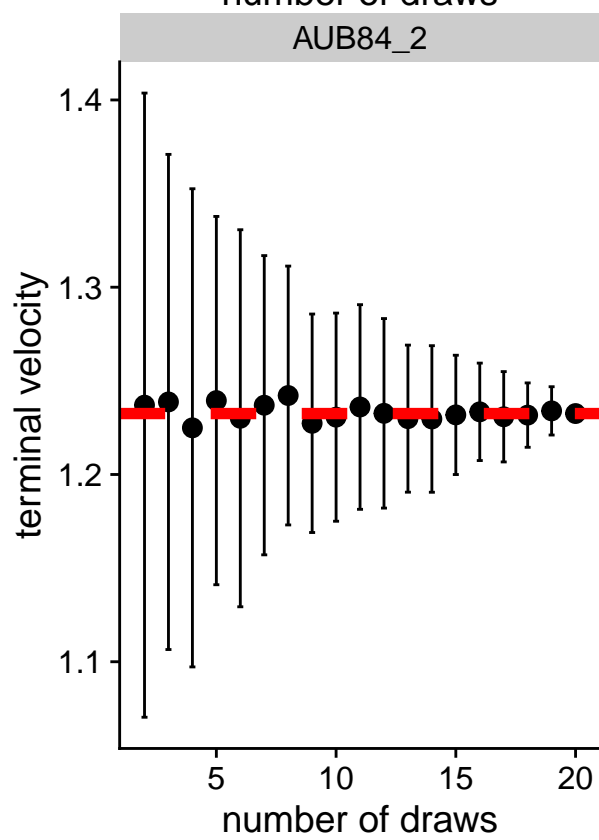
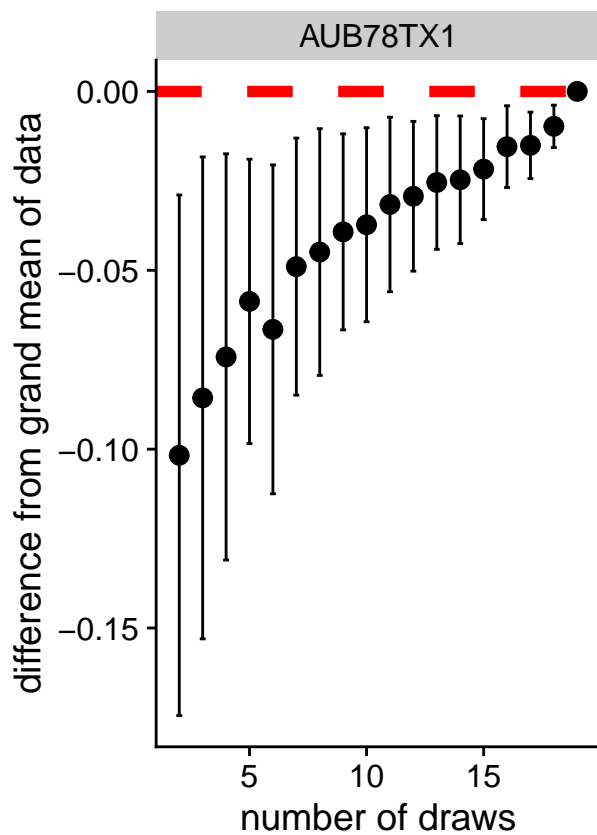
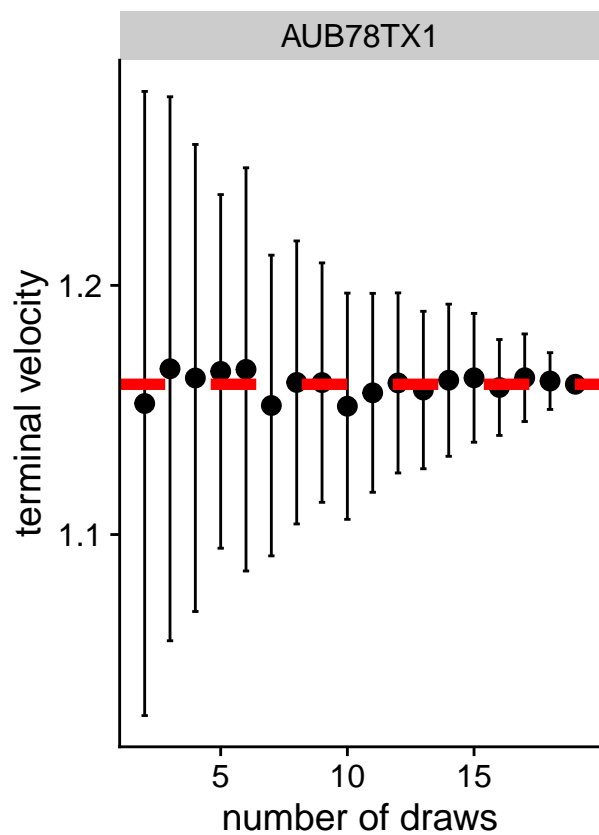
  a <- ggplot(individ, aes(x = no_draws, y = tv_m))+
    geom_point(cex = 3)+
    geom_errorbar(aes(x = no_draws, ymin = tv_m - tv_sd, ymax = tv_m + tv_sd), width = 0.25)+
    geom_hline(aes(yintercept = gm), linetype="dashed", color = "red", size=2)+
    #geom_hline(yintercept=grand_mean+grand_sd, linetype="solid", color = "red", size=1)+
    #geom_hline(yintercept=grand_mean-grand_sd, linetype="solid", color = "red", size=1)+
    labs(x = "number of draws", y = "terminal velocity", main = "Mean Terminal Velocity")+
    facet_wrap(~ individual, scale = "free")

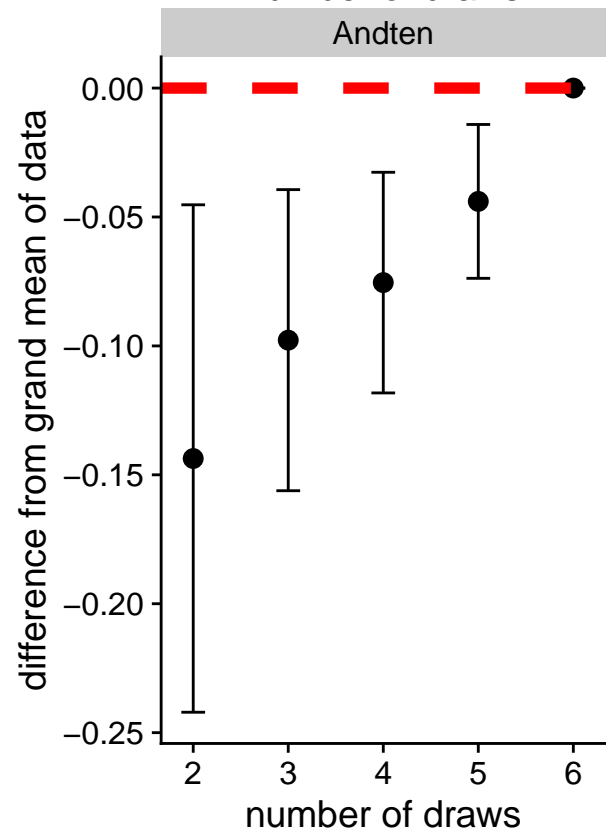
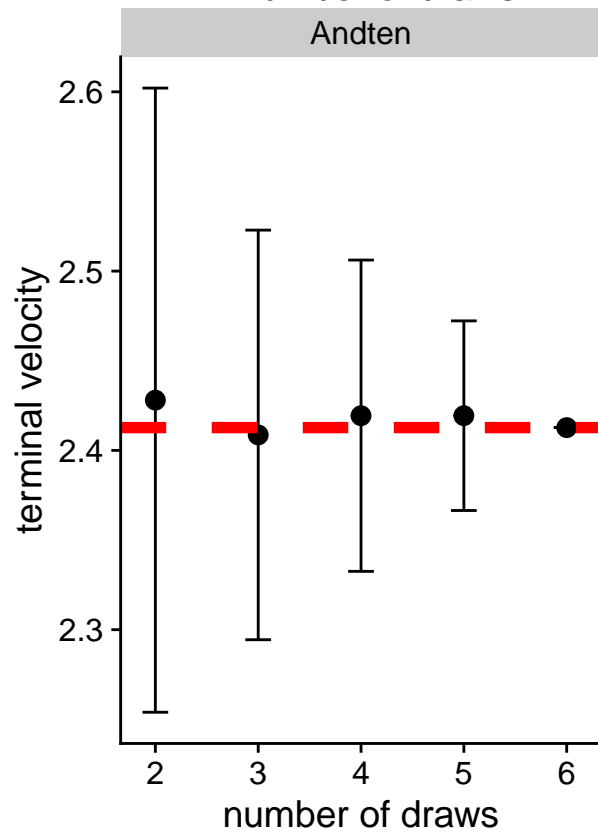
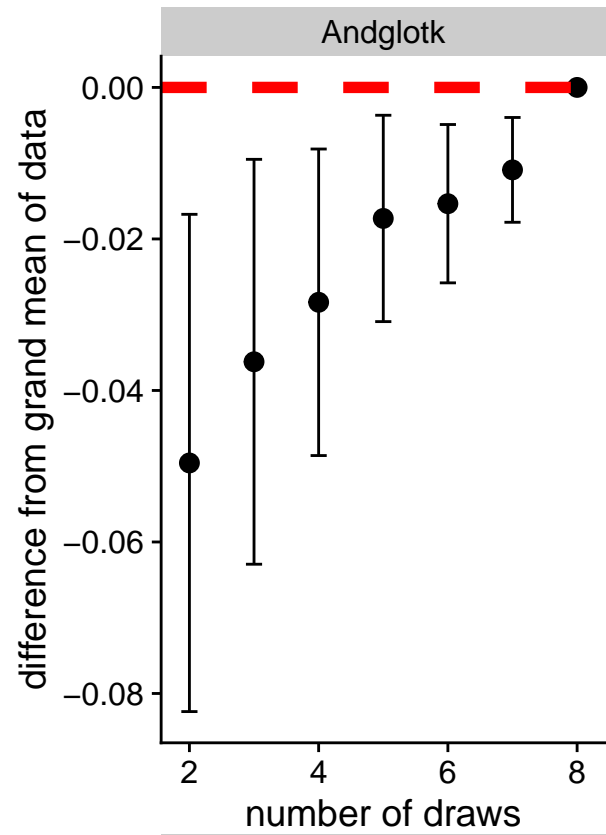
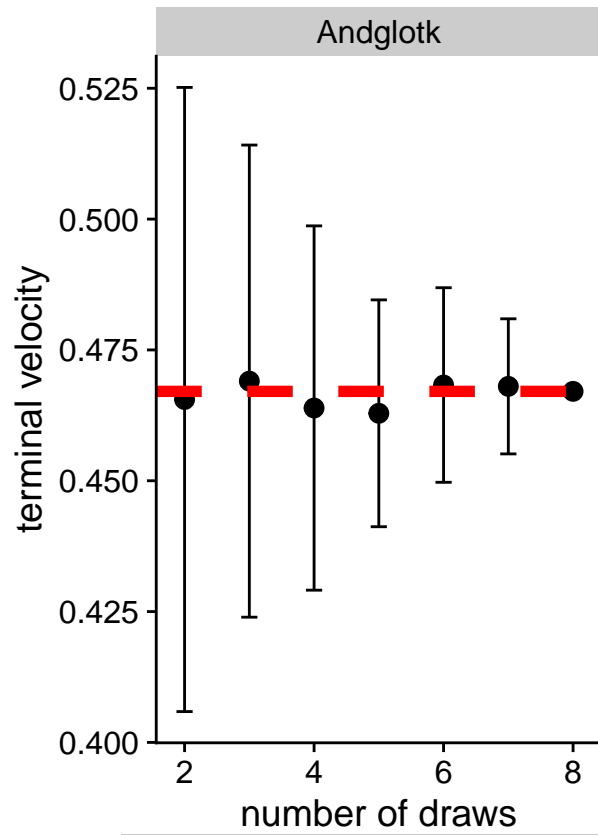
  # plot of differences and errors
  b <- ggplot(individ, aes(x = no_draws, y = -diff_m))+
    geom_point(cex = 3)+
    geom_errorbar(aes(x = no_draws, ymin = -diff_m - diff_sd, ymax = -diff_m + diff_sd), width = 0.25)+
    geom_hline(yintercept=0, linetype="dashed", color = "red", size=2)+
    labs(x = "number of draws", y = "difference from grand mean of data", main = "Difference from G...")+
    facet_wrap(~ individual, scale = "free")

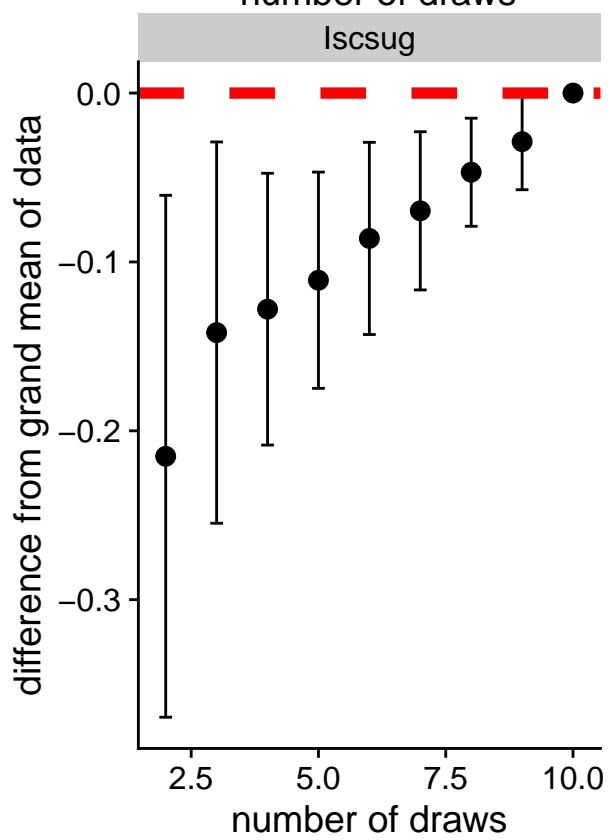
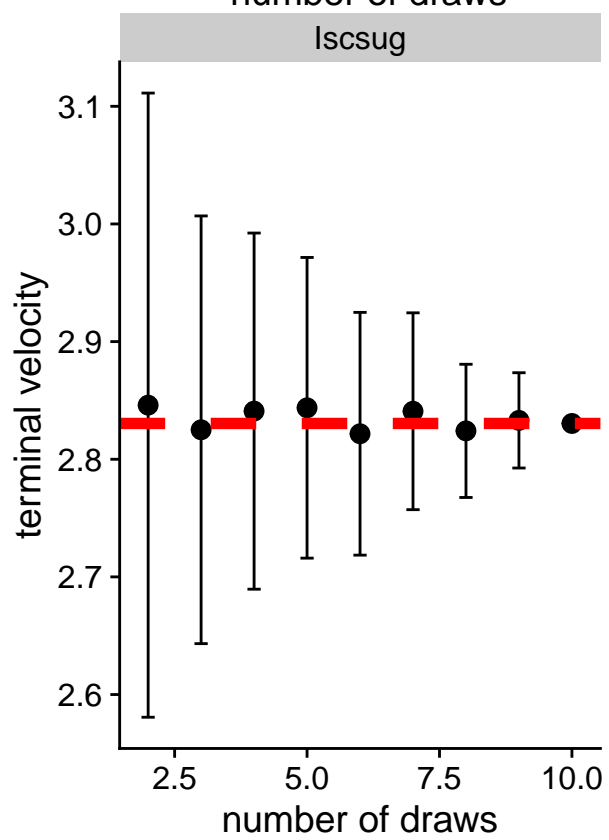
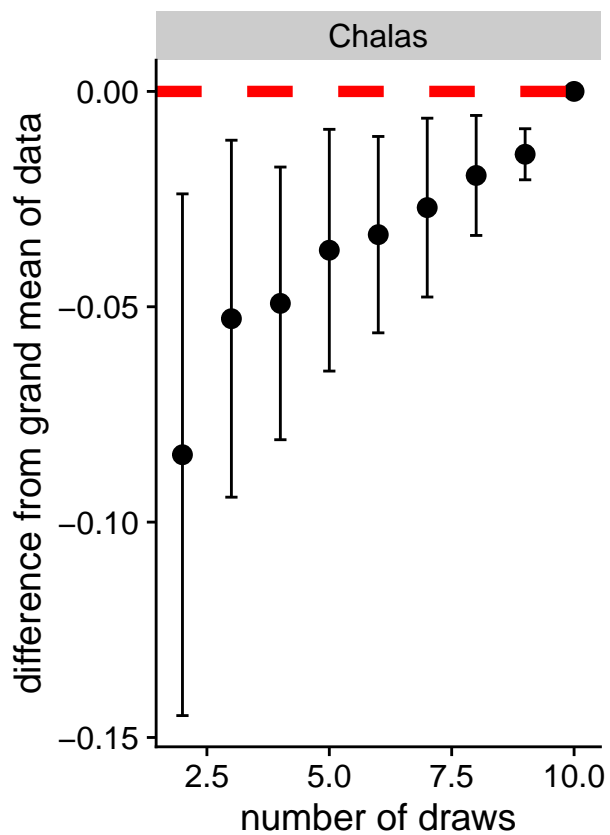
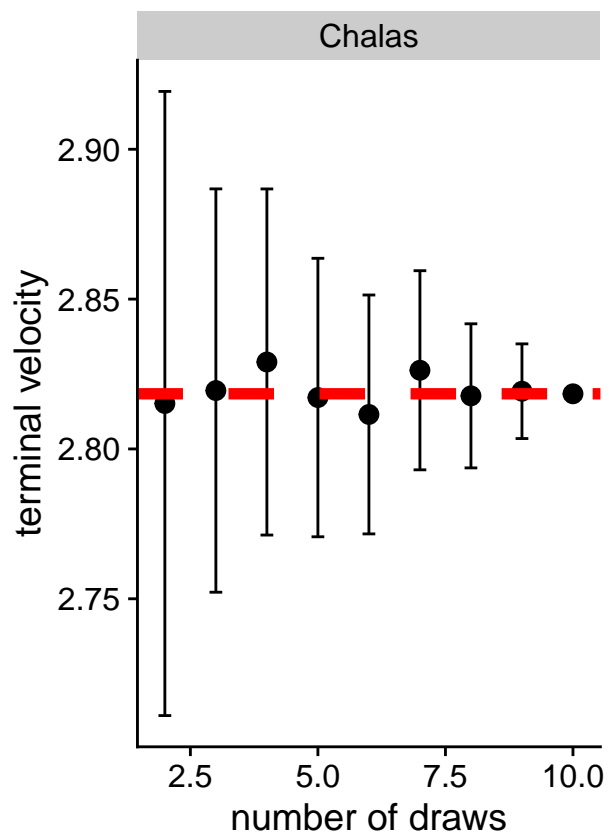
  print(plot_grid(a, b, nrow = 1))
}

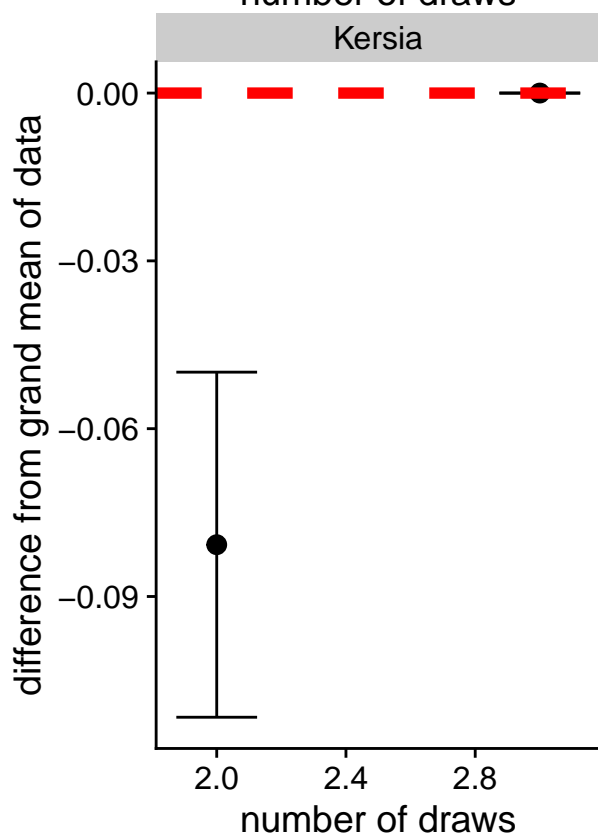
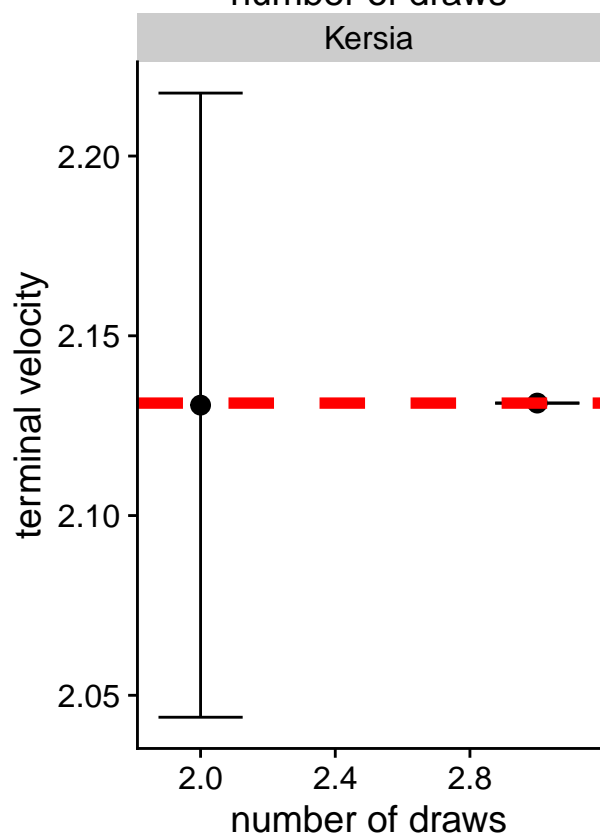
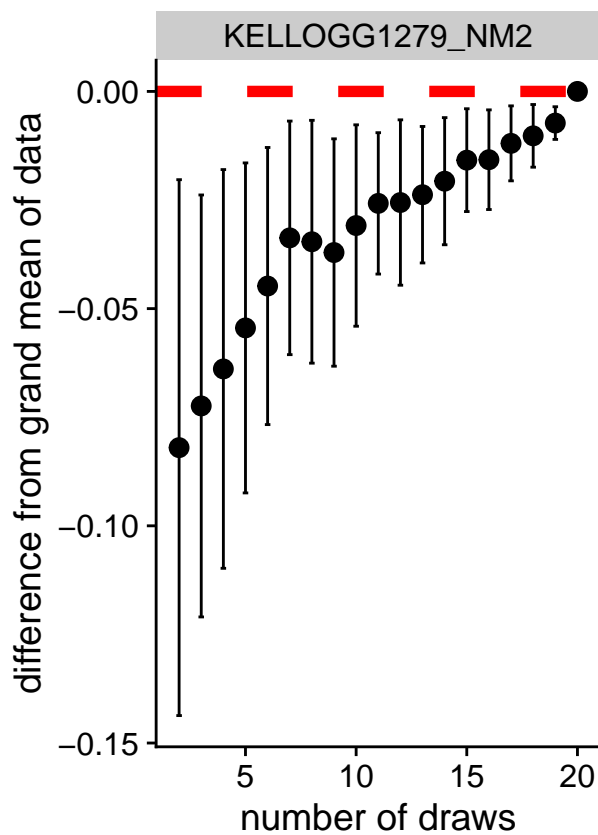
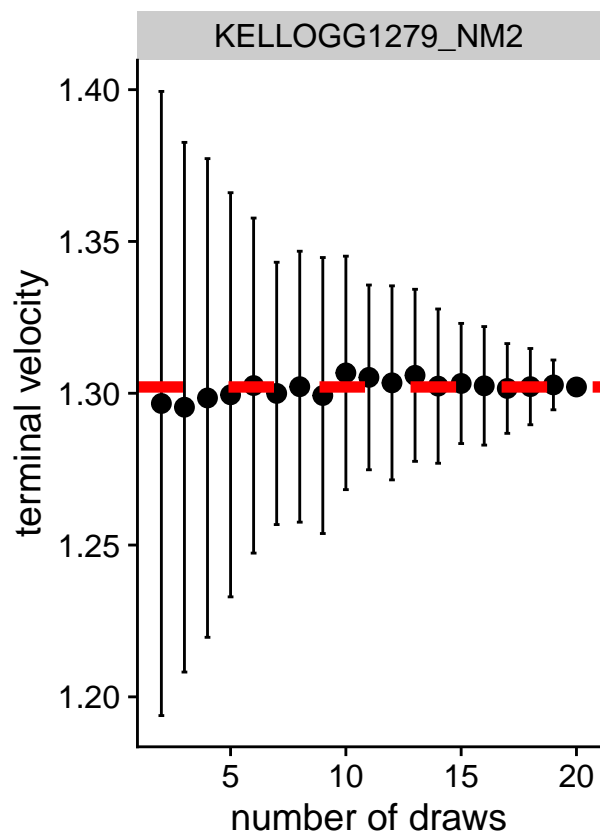
```

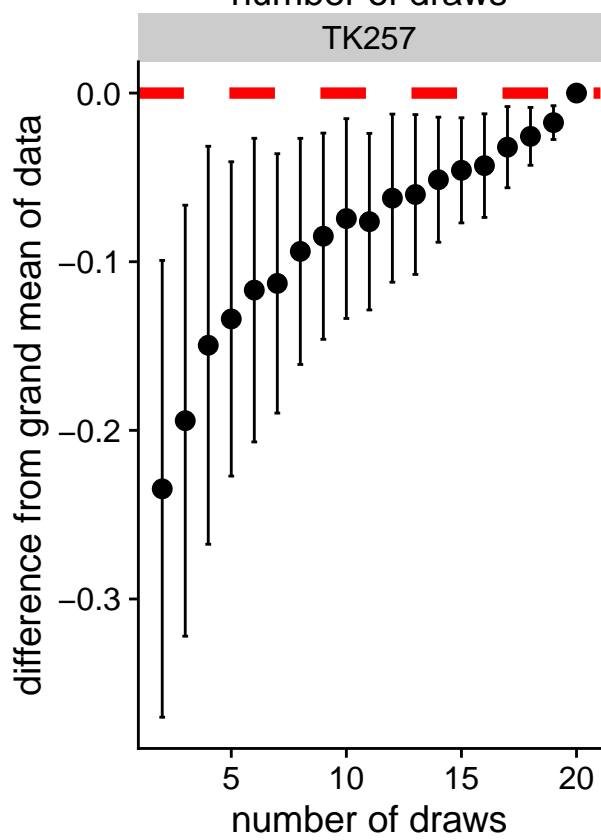
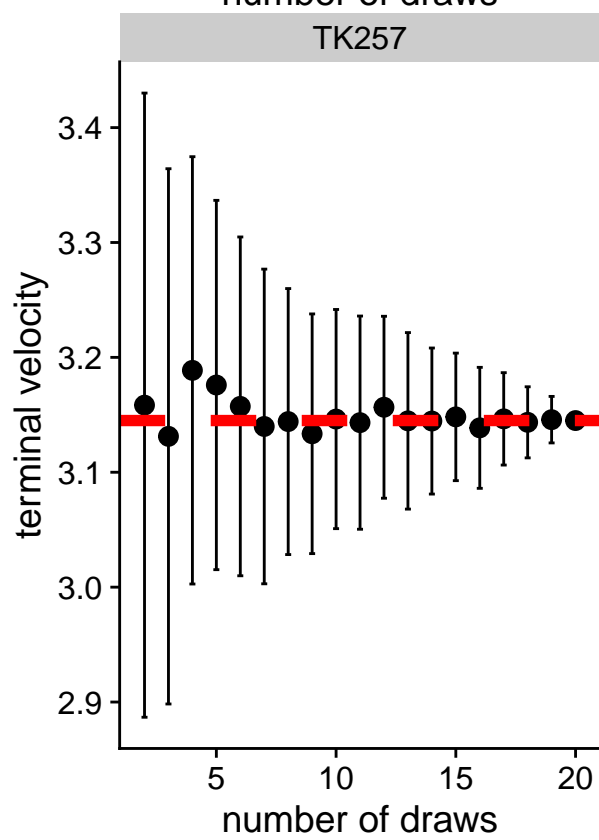
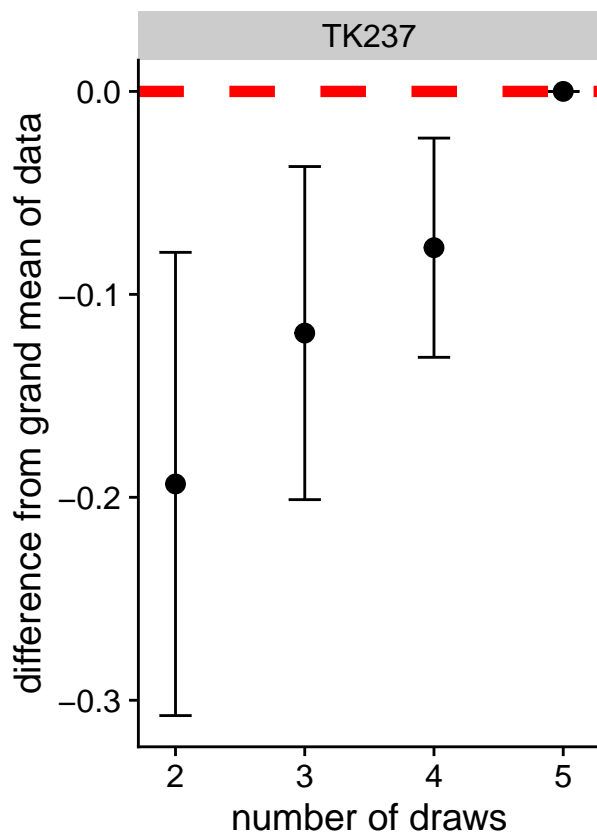
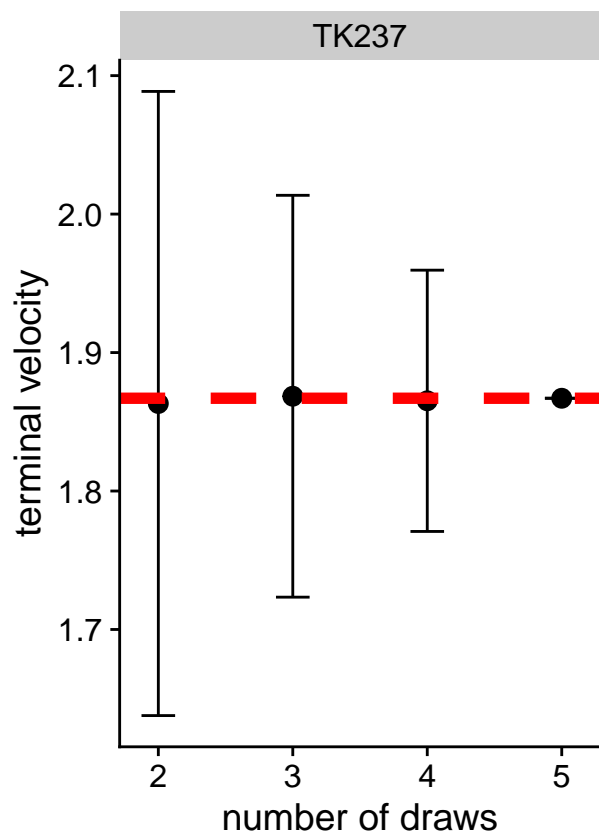



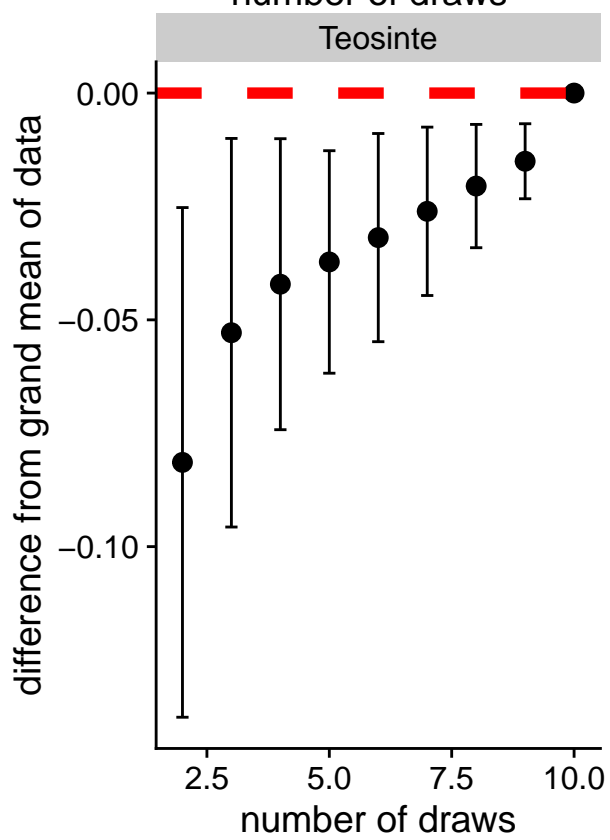
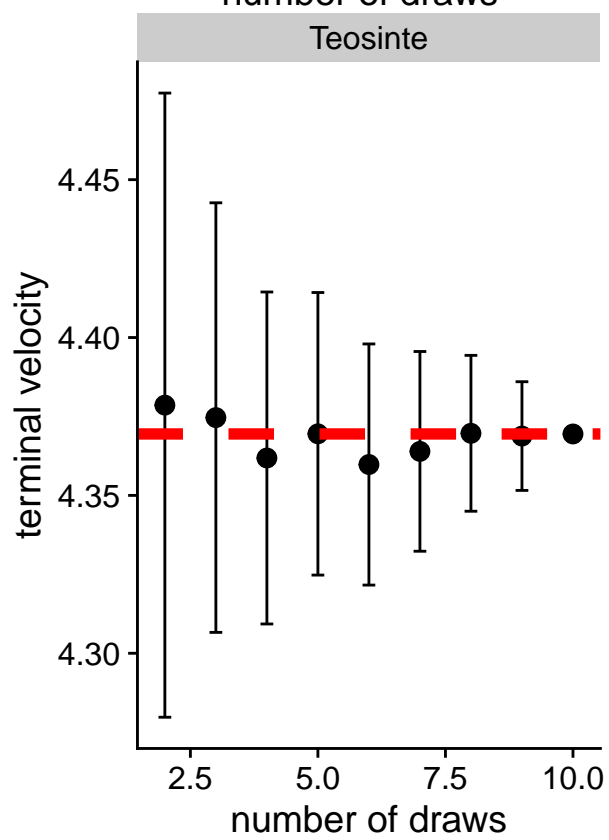
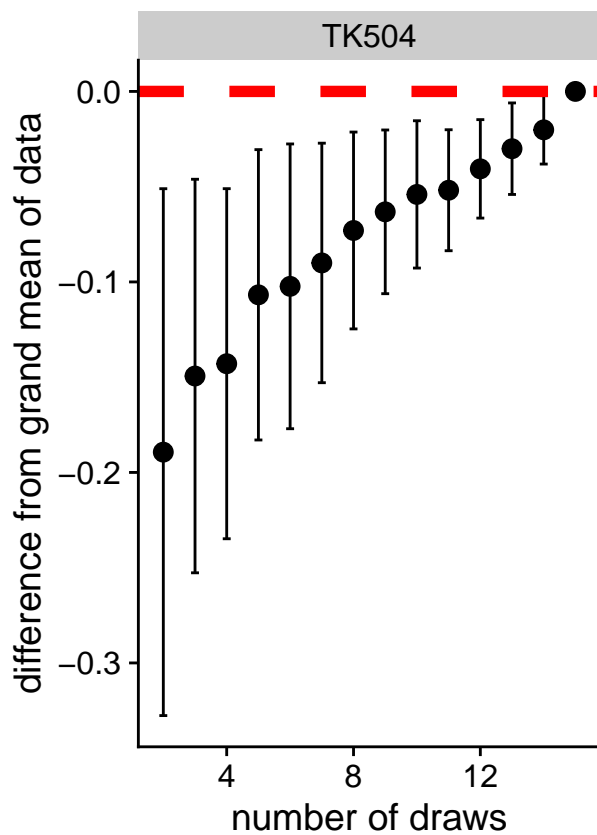
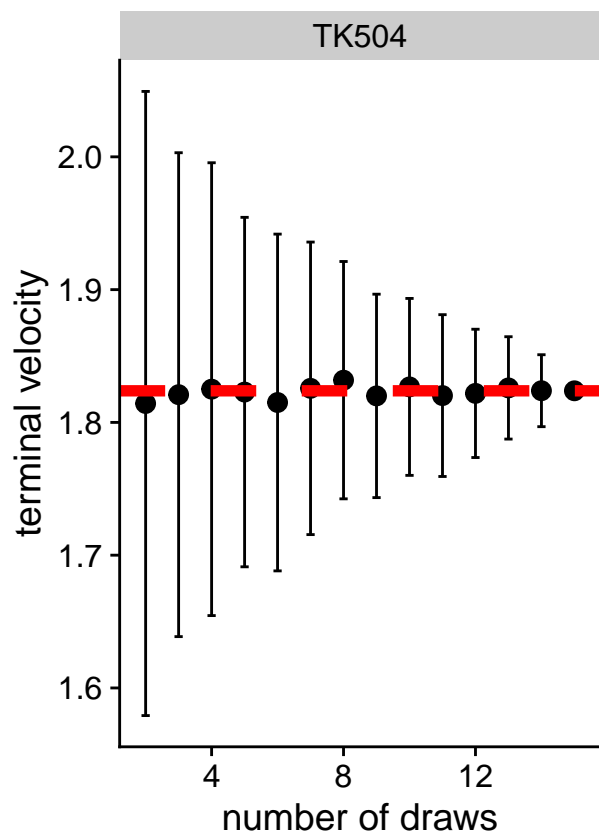


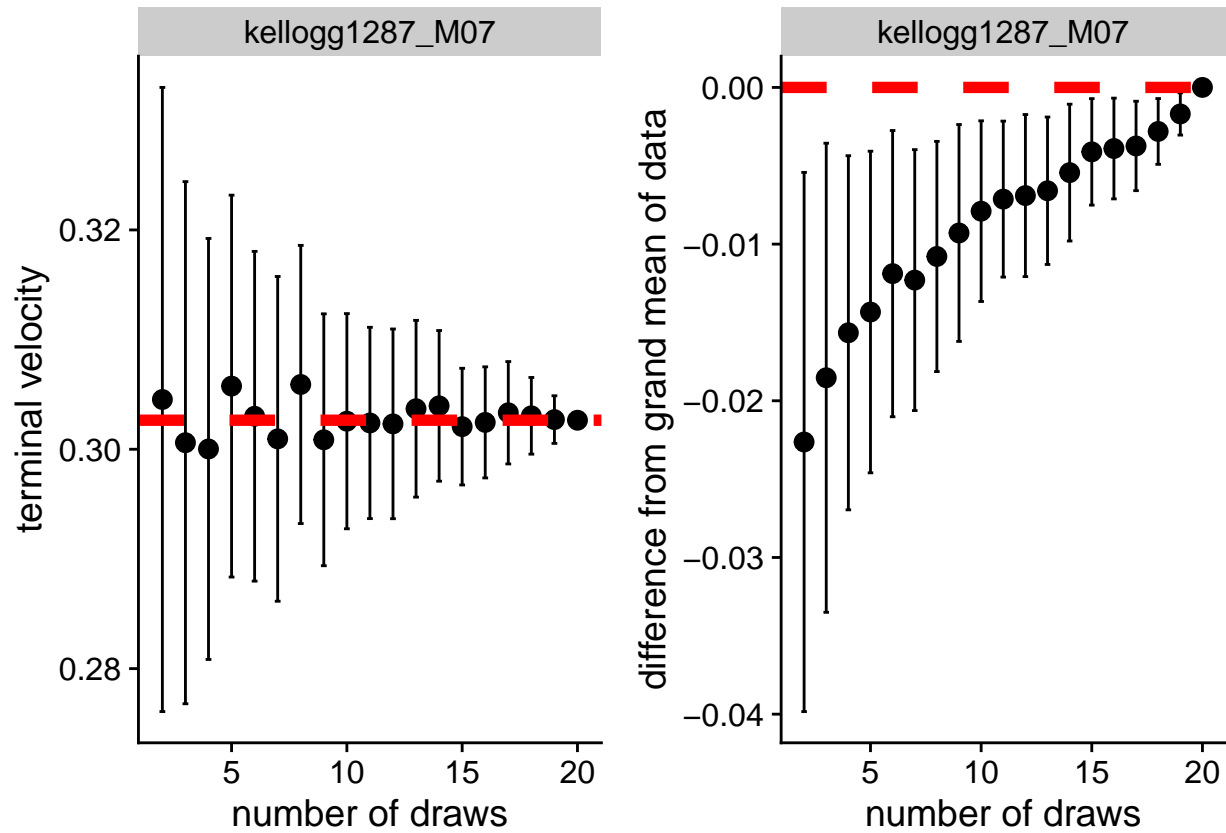












In the end, we went with 10 replicates because it represented a close enough match to the mean for 20 seeds, and also not all species had more than 10 seeds to drop, so it was a good balance value for number of drops.