Normative Jump Data Malaysia

Samuel Montalvo, Ph.D.

11/20/2022

library(readr)  
Df <- read\_csv("Project-Session-09\_13\_22-Session-09\_13\_22-AveragesAndTests\_Countermovement\_Jump.csv")  
View(Df)  
  
library(dplyr)  
library(ggplot2)  
library(PupillometryR)  
library(ggprism)  
library(flextable)  
library(officer)  
library(table1)  
library(janitor) #convert spaces to \_  
library(interlimb)  
library(caret)  
library(ggpubr)  
library(plotly)  
## Reduce jump trials to average  
Df\_average <- Df %>% filter(TestId == "AVERAGE")

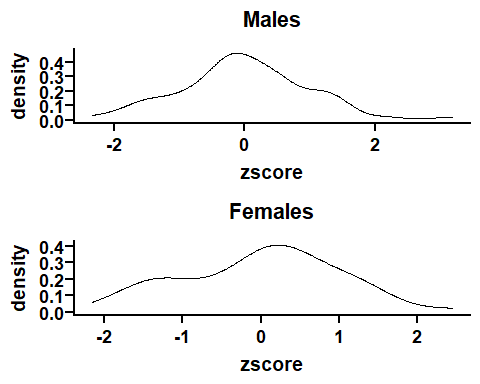
## Data Wrangling and Cleaning

Names of participants and sex needs to be join into the main jump dataset to have a complete dataset. New dataset is then saved into an csv document

## import participant details  
library(readxl)  
Df\_details <- read\_excel("Athletes details SUKMA XX.xlsx")  
  
# change capital name letters so that data sets match  
Df\_details <- Df\_details %>% rename(Name = NAME)  
  
# re name males and females  
Df\_details$Sex <- recode\_factor(Df\_details$Sex, F = "Female",  
 M = "Male")  
# merge data sets by Name  
Df <- inner\_join(x=Df\_details,y=Df\_average,  
 by = "Name")  
  
## Subset NA  
Df <- Df |>  
 subset(!is.na(`Jump Height`))  
  
# convert spaces to \_  
Df <- clean\_names(Df)  
  
#remove names  
Df <- Df %>% select(-c(name))  
  
## Save as a new csv file  
write.csv(Df,"C:\\Users\\Samuel\\OneDrive\\Documents\\R\\SUKMA\_Wushu\\SUKMA\_Wushu\_Data.csv", row.names = FALSE)

## Zscore

Creation and distribuition of the zscore. First need to create two data frames, one for males and one for females.



## Tscore

next step is to create a tscore based on the zscore \* 10 + 50 as recommended by the article. Then, a rank class is created based on the tscore. Lastly, factors are reorderd so that they appear in that order in the plots.

# rank based on T score  
Df\_male <- Df\_male%>% mutate(Tscore = zscore\*10+50)  
Df\_female <- Df\_female%>% mutate(Tscore = zscore\*10+50)  
  
Df\_male <- Df\_male %>% mutate(Description =   
 case\_when(Tscore > 80 ~ 'Excellent',  
 Tscore < 81 & Tscore > 70 ~ 'Very Good',  
 Tscore < 71 & Tscore > 60 ~ 'Good',  
 Tscore < 61 & Tscore > 55 ~ 'Above Average',  
 Tscore < 56 & Tscore > 45 ~ 'Average',  
 Tscore < 46 & Tscore > 40 ~ 'Below Average',  
 Tscore < 41 & Tscore > 30 ~ 'Poor',  
 Tscore < 31 & Tscore > 20 ~ 'Very Poor',  
 Tscore < 21 ~ 'Extremely Poor'))   
  
Df\_male$Description <- factor(Df\_male$Description,  
 levels = c("Excellent", "Very Good", "Good",   
 "Above Average", "Average",  
 "Below Average","Poor",   
 "Very Poor", "Extremely Poor"),  
 ordered = TRUE)  
  
  
Df\_female <- Df\_female %>% mutate(Description =   
 case\_when(Tscore > 80 ~ 'Excellent',  
 Tscore < 81 & Tscore > 70 ~ 'Very Good',  
 Tscore < 71 & Tscore > 60 ~ 'Good',  
 Tscore < 61 & Tscore > 55 ~ 'Above Average',  
 Tscore < 56 & Tscore > 45 ~ 'Average',  
 Tscore < 46 & Tscore > 40 ~ 'Below Average',  
 Tscore < 41 & Tscore > 30 ~ 'Poor',  
 Tscore < 31 & Tscore > 20 ~ 'Very Poor',  
 Tscore < 21 ~ 'Extremely Poor'))   
  
Df\_female$Description <- factor(Df\_female$Description,  
 levels = c("Excellent", "Very Good", "Good",   
 "Above Average", "Average",  
 "Below Average","Poor",   
 "Very Poor", "Extremely Poor"),  
 ordered = TRUE)

##Interactive Plots

p1 <- Df\_male %>% ggplot(y=jump\_height,x=zscore) +  
 geom\_point(aes(y=jump\_height,x=zscore,colour=Description),  
 position = position\_jitter(width = .25), size = 2, shape=20) +   
 scale\_color\_manual(values = c("#125016", "#1b7821", "#24a02c",  
 "#125016", "#000000", "#e7b416",  
 "#CC3232","#a32828","#7a1e1e")) +  
 theme\_bw() + labs(title="Males",y="Jump Height (m)")  
  
ggplotly(p1)

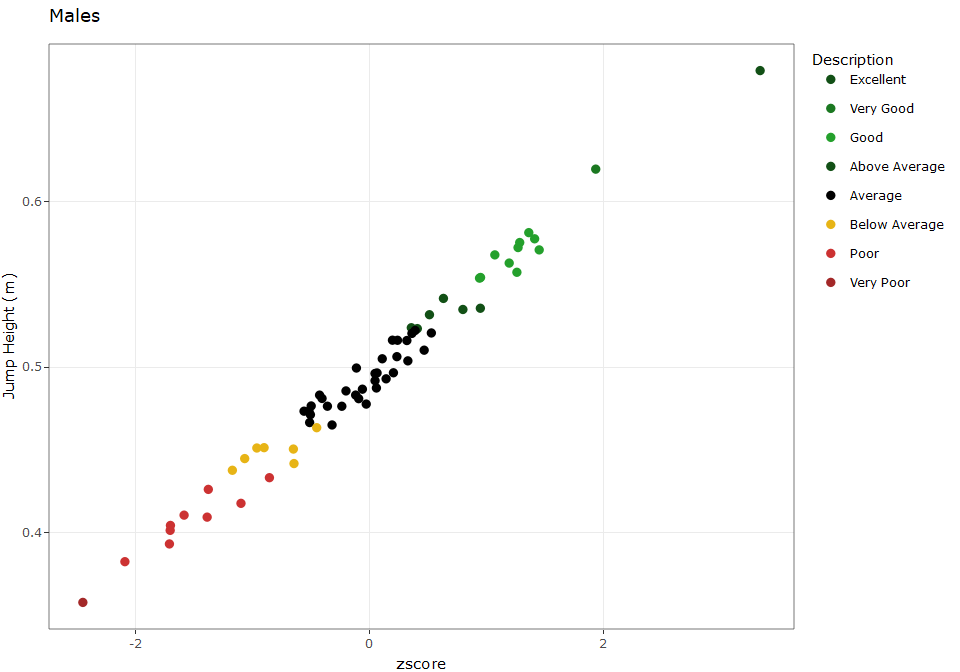


Figure 1. Interactive plot of normative jump data for male athletes.

p2 <- Df\_female %>% ggplot(y=jump\_height,x=zscore) +  
 geom\_point(aes(y=jump\_height,x=zscore,colour=Description),  
 position = position\_jitter(width = .25), size = 2, shape=20) +   
 scale\_color\_manual(values = c("#125016", "#1b7821", "#24a02c",  
 "#125016", "#000000", "#e7b416",  
 "#CC3232","#a32828","#7a1e1e")) +  
 theme\_bw() + labs(title="Female",y="Jump Height (m)")  
  
ggplotly(p2)

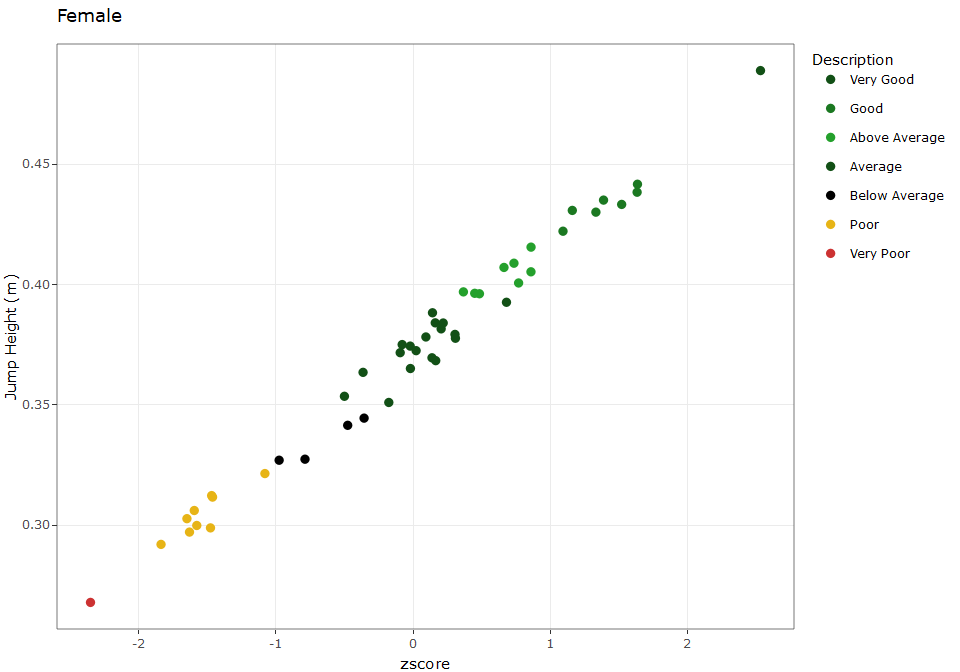


Figure 2. Interactive plot of normative jump data for female athletes.

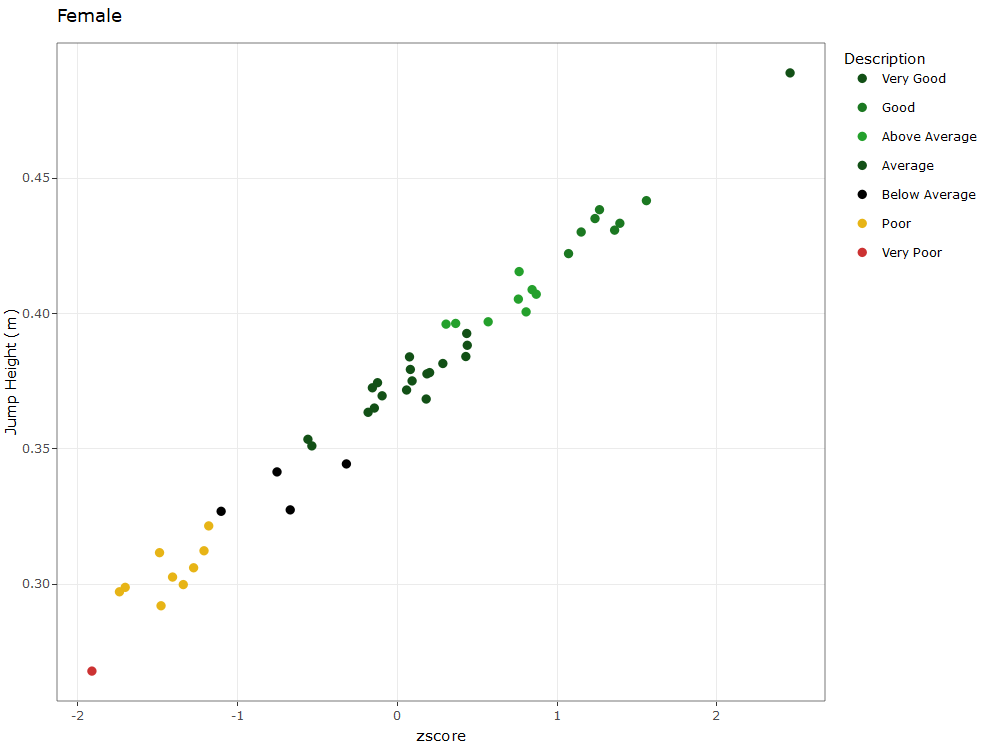
library(htmlwidgets)

## Warning: package 'htmlwidgets' was built under R version 4.1.3

library(webshot)

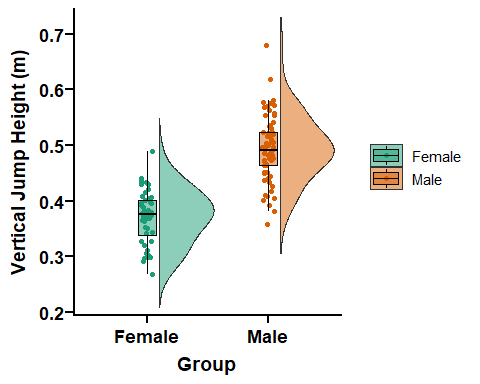
## Warning: package 'webshot' was built under R version 4.1.3

x<- ggplotly(p2)  
saveWidget(x, "temp.html")  
webshot("temp.html", "temp.png")



## Distribuition of Jump Height by groups

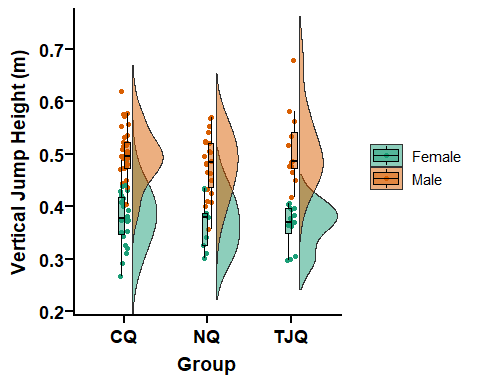
Df %>% ggplot(aes(x = sex, y = jump\_height, fill = sex)) +  
 geom\_flat\_violin(aes(fill = sex),  
 position = position\_nudge(x =.1, y=0), adjust = 1.5,trim = F, alpha =.5) +  
 geom\_point(aes(x=sex, y=jump\_height,colour=sex),  
 position = position\_jitter(width = .05), size = 2.5, shape = 20)+  
 geom\_boxplot(aes(x = sex, y = jump\_height, fill = sex),outlier.shape  
 = NA, alpha = .5, width = .15, colour = "black")+  
 coord\_cartesian(ylim=c(0.22,0.72)) +  
 scale\_colour\_brewer(palette = "Dark2")+  
 scale\_fill\_brewer(palette = "Dark2")+ theme\_bw() +  
 ylab('Vertical Jump Height (m)')+  
 xlab('Group') + theme\_prism()



ggsave("Vertical\_jump\_height.png")

## Saving 5 x 4 in image

Df %>% ggplot(aes(x = events, y = jump\_height, fill = sex)) +  
 geom\_flat\_violin(aes(fill = sex),  
 position = position\_nudge(x =.1, y=0), adjust = 1.5,trim = F, alpha =.5) +  
 geom\_point(aes(x=events, y=jump\_height,colour=sex),  
 position = position\_jitter(width = .05), size = 2.5, shape = 20)+  
 geom\_boxplot(aes(x = events, y = jump\_height, fill = sex),outlier.shape  
 = NA, alpha = .5, width = .15, colour = "black")+  
 coord\_cartesian(ylim=c(0.22,0.75)) +  
 scale\_colour\_brewer(palette = "Dark2")+  
 scale\_fill\_brewer(palette = "Dark2")+ theme\_bw() +  
 ylab('Vertical Jump Height (m)')+  
 xlab('Group') + theme\_prism()



Df %>% ggplot(aes(x = events, y = jump\_height, fill = events)) +  
 geom\_flat\_violin(aes(fill = events),  
 position = position\_nudge(x =.1, y=0), adjust = 1.5,trim = F, alpha =.5) +  
 geom\_point(aes(x=events, y=jump\_height,colour=events),  
 position = position\_jitter(width = .05), size = 2.5, shape = 20)+  
 geom\_boxplot(aes(x = events, y = jump\_height, fill = events),outlier.shape  
 = NA, alpha = .5, width = .15, colour = "black")+  
 coord\_cartesian(ylim=c(0.22,0.8)) +  
 scale\_colour\_brewer(palette = "Dark2")+  
 scale\_fill\_brewer(palette = "Dark2")+ theme\_bw() +  
 ylab('Vertical Jump Height (m)')+  
 xlab('Group') + theme\_prism()

