PercAct exam

.libPaths( c("C:/Users/biank/Documents/Skole/CogSci/R\_packages", .libPaths() ) )  
.libPaths()

## [1] "C:/Users/biank/Documents/Skole/CogSci/R\_packages"   
## [2] "C:/Users/biank/OneDrive/Dokumenter/R/win-library/3.6"  
## [3] "C:/Program Files/R/R-3.6.2/library"

pacman::p\_load(tidyverse)  
pacman::p\_load(mFilter)

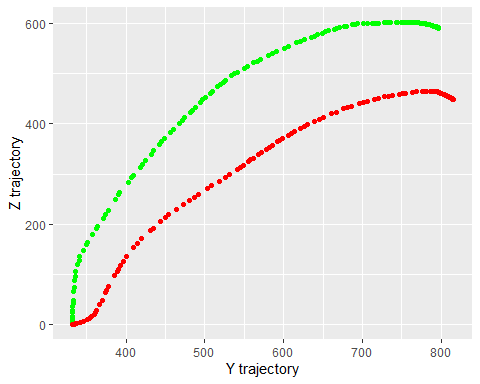
## R Markdown

3.2. The following data represent the right-hand movements of the person in 3.1 while moving towards the bottle to grasp it. We are viewing the person from the side so only y (back-front axis) and z (up-down axis) are given. (Note: The measurement units are meaningless in this example.) Download the two demo trajectories: 105\_1.txt and 105\_2.txt. Then do the following steps in R. Include your commented code and the resulting figures in your overall exam document, i.e. do not just send a link to your GitHub etc. [10%] o Load the two trajectories and rename the columns in each to ‘time’, ‘y’, ‘z’.

#we read the data  
  
t1<-read.delim("105\_1.txt",header=F,sep=",")  
  
t2<-read.delim("105\_2.txt",header=F,sep=",")  
  
names(t1)[names(t1) == "V1"] <- "time"  
names(t1)[names(t1) == "V2"] <- "y"  
names(t1)[names(t1) == "V3"] <- "z"  
  
  
names(t2)[names(t2) == "V1"] <- "time"  
names(t2)[names(t2) == "V2"] <- "y"  
names(t2)[names(t2) == "V3"] <- "z"

o Plot y versus z for the two curves on top of each other within the same graph. Add meaningful labels.

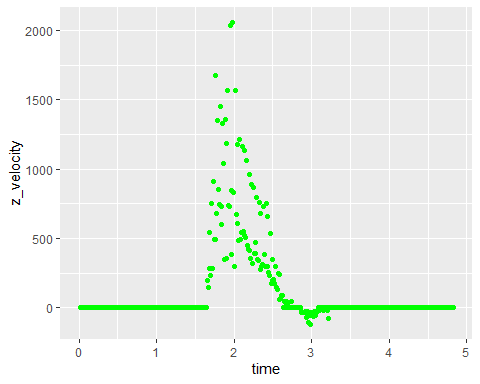
ggplot()+geom\_point(data=t1, aes(x=y,y=z, color="Trajectory 1"), color='green')+geom\_point(data=t2, aes(x=y,y=z, color="Trajectory 2"), color='red')+labs(x="Y trajectory", y="Z trajectory")



o For the first trajectory only: o Calculate z velocity and plot it against time in a new graph.

#Calculating velocity  
  
# Calculating z velocity. Velocity = delta y / delta t  
t1$z\_velocity<-(t1$z-lag(t1$z))/(t1$time-lag(t1$time))  
  
#Plotting the z velocity against time  
ggplot()+geom\_point(data=t1, aes(x=time,y=z\_velocity), color='green')

## Warning: Removed 1 rows containing missing values (geom\_point).



o Apply a Butterworth filter to the z velocity. Choose reasonable parameters yourself.

# Load library with butterworth filter  
library(signal)

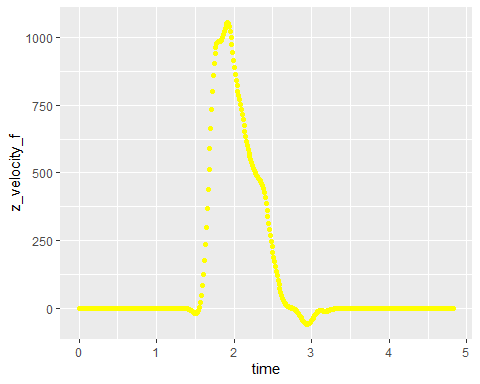
## Warning: package 'signal' was built under R version 3.6.3

##   
## Attaching package: 'signal'

## The following object is masked from 'package:dplyr':  
##   
## filter

## The following objects are masked from 'package:stats':  
##   
## filter, poly

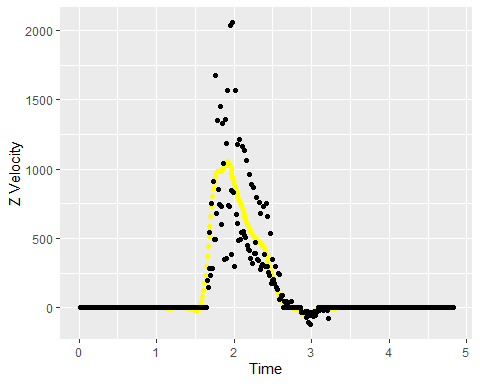
# Filter settings  
filter\_cutoff <- .1  
filter\_order <- 2  
bf <- butter(filter\_order, filter\_cutoff, type ='low')   
  
# Applying filter to z velocity  
t1$z\_velocity\_f<-"filtfilt"(bf, t1$z\_velocity)  
  
#we try to plot the butterworth-filtered z\_velocity  
ggplot()+geom\_point(data=t1, aes(x=time,y=z\_velocity\_f), color='yellow')



o Plot the filtered curve against time on top of the unfiltered one in the previous graph.

#Yellow is filtered, black is not filtered  
ggplot()+geom\_point(data=t1, aes(x=time,y=z\_velocity\_f), color='yellow')+geom\_point(data=t1, aes(x=time,y=z\_velocity), color='black')+labs(x="Time",y="Z Velocity", color="Filtered or not")

## Warning: Removed 1 rows containing missing values (geom\_point).



Policeman

Policeman

Policeman

Policeman

Policeman



Nurse

Policeman