

Segmentation of Trip Data - Part2

Dave Hurst

Thursday, December 25, 2014

Notes for automating detection of straight trip segments

A note on the speed plot:

I started my notes using a lowess smoothed plot (original PlotSegment pair plots), and have migrated to plotting the moving average ($t=5$). Before moving forward I wanted to pick one or the other and based on the plot below, I think MA is better than Lowess, as it will help smooth out the variation better. In particular, I'm looking at the long deceleration between $t \sim 212$ and $t \sim 219$. In PlotSegment, I had picked a Lowess smoothing factor of .01 based on the plot segments I was looking at when I wrote it, but in this case the results are the same for $f=.01$ and $f=.1$, and both look identical to the raw data, which has a couple of stair steps that are probably just noise. The smoothing parameter for lowess is less intuitive than for MA (seconds), and is sensitive to the length of the segment, while MA is not.

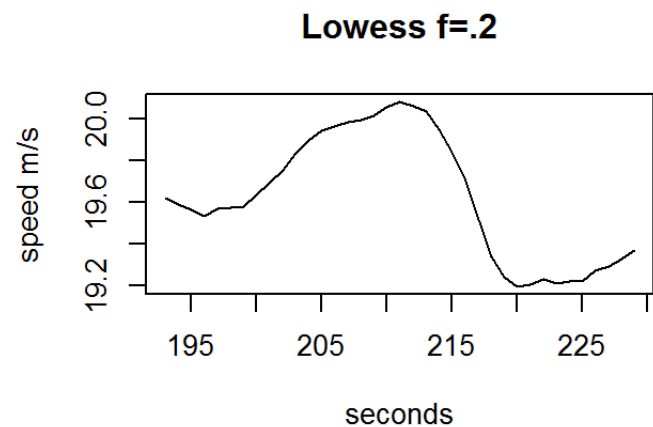
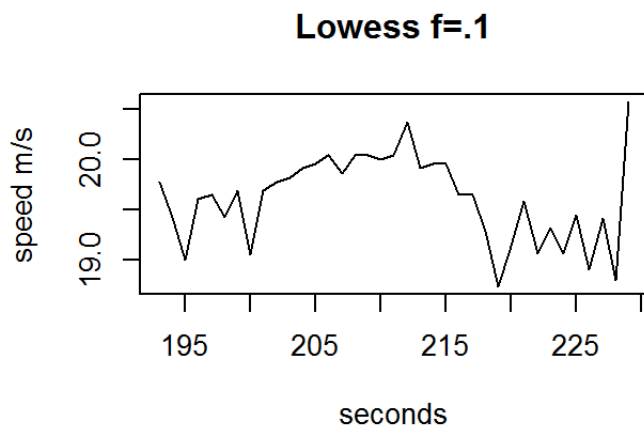
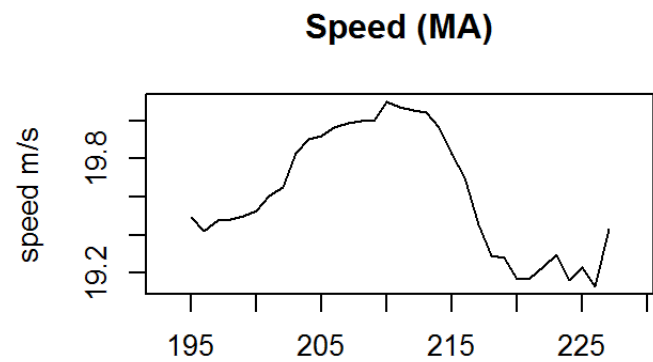
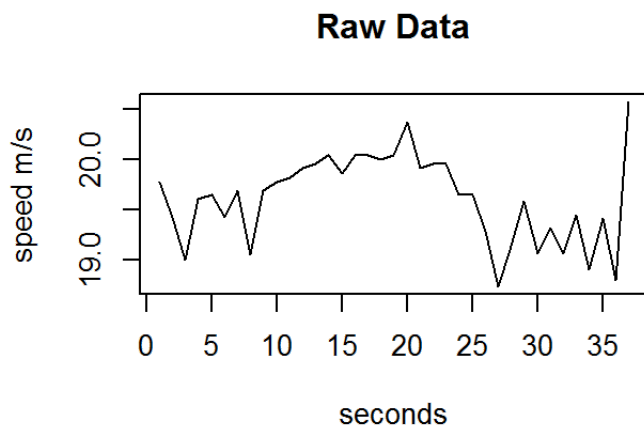
Here's the comparison plot:

```
par.orig <- par(mfrow=c(2,2))
trip <- getTrip( 2591, 199 )
t1 <- 193
t2 <- 229

plot(trip$v[t1:t2], xlab="seconds", ylab="speed m/s", main="Raw Data", type="l")
plotTripSegment.speed(trip, t1, t2)
```

```
## Warning in is.na(x): is.na() applied to non-(list or vector) of type
## 'NULL'
```

```
plot(lowess(t1:t2, trip$v[t1:t2], f=.1), type="l", xlab="seconds", ylab="speed m/s", main="Lowess f=.1")
plot(lowess(t1:t2, trip$v[t1:t2], f=.2), type="l", xlab="seconds", ylab="speed m/s", main="Lowess f=.2")
```



Automated segment detection (based on bearing lag $< \pm 3^\circ$)

Note: in Part1 I suggested the trip should start where the bearing switched direction and then headed out of the zone. While writing the segment detection code, it was easier just to mark the last point that was in the zone, and then shrink the zone a little. That may be better anyway. I still need to iterate on a good shrink factor. 10 seconds seems like a good first guess.

Also, I've shorted the minimum segment length to 30, since trip 200 otherwise had no straight segments. We can still filter on the length of the segment if necessary. Another variable we might want to play with is the min-speed threshold I've set at 5 m/s, but the lower we go the more the GPS accuracy will be a problem. I think what we'll need to do is have models that take into account the type of trip – i.e short/versus long, etc.

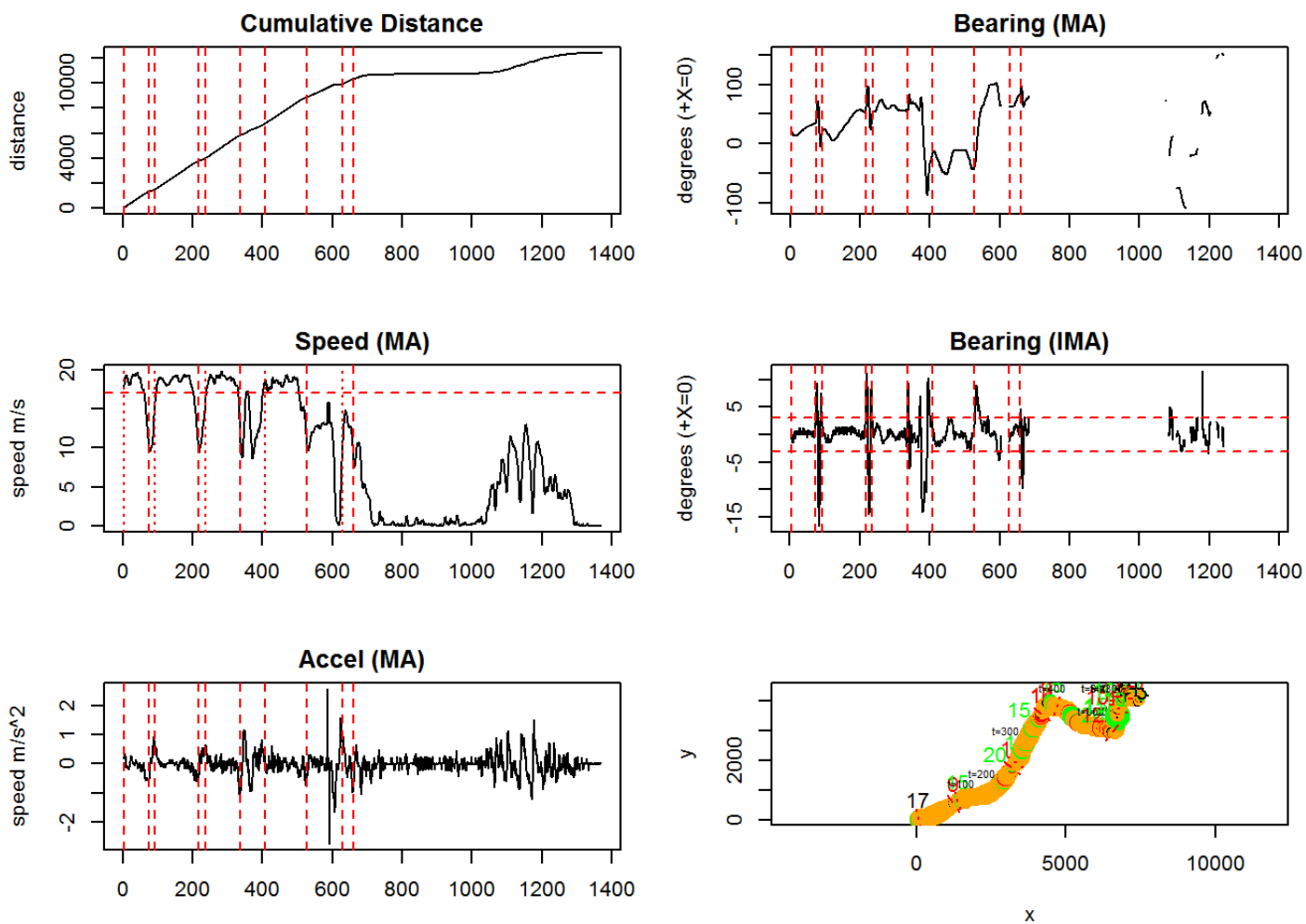
Generally, I'm really happy with these results. Here are some visual inspections of the segments that were auto detected for various routes

Observations from Driver 2591, Trip 99

```
ss <- segment.parse.bearing(trip)
print(nrow(ss))
```

```
## [1] 5
```

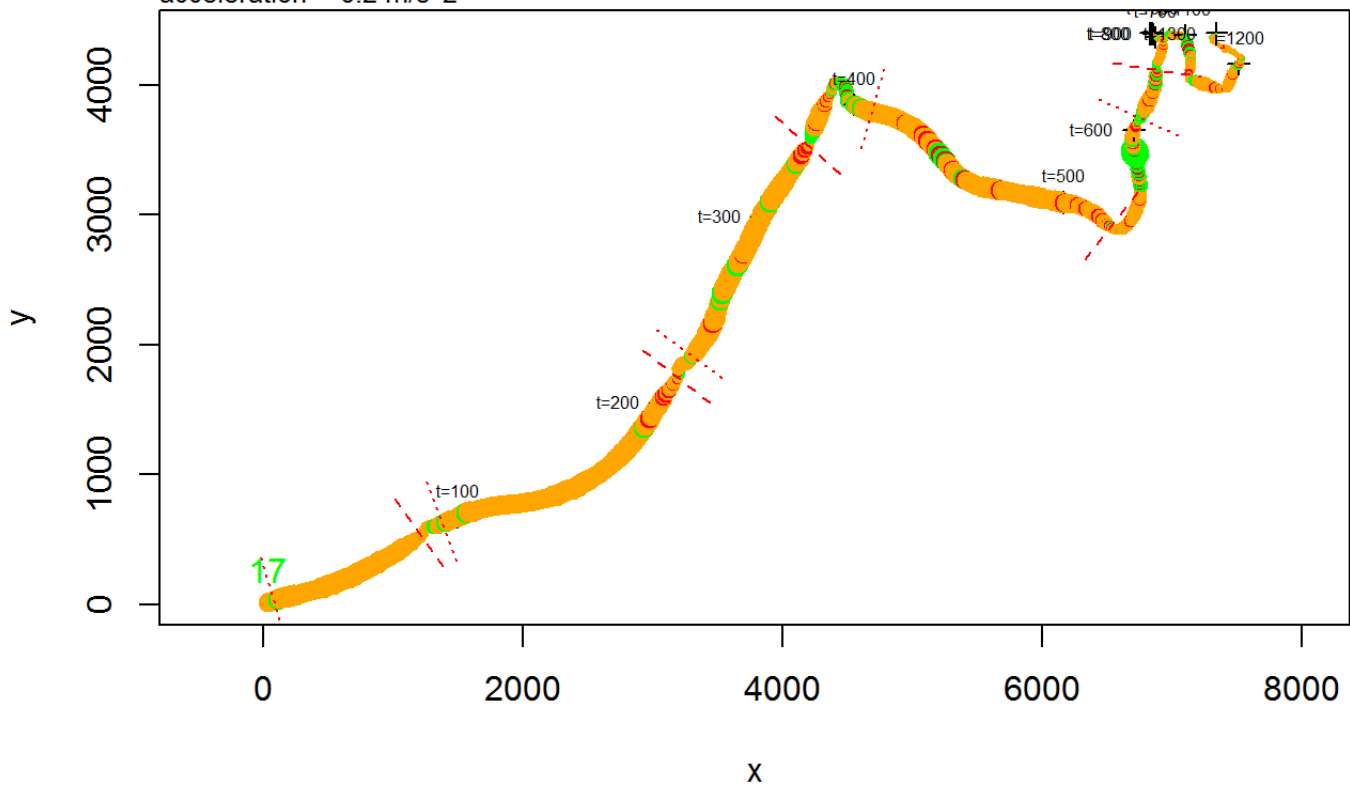
```
plotTripSegment6(trip, 1, 2000, b.marks=sort( c(ss$t0, ss$t0+ss$tlen)) )
```



```
plotTrip(trip, v.mark=50, b.marks=sort( c(ss$t0, ss$t0+ss$tlen)))
```

Plot of Route

distance traveled: 12.3 km
 direction= 180 deg
 current speed= 0.0 km/h
 acceleration= -0.2 m/s²

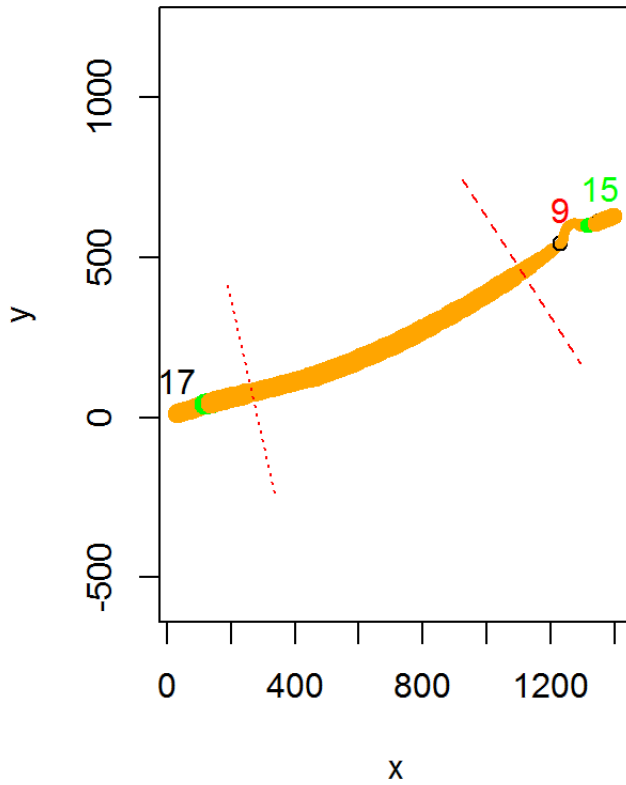


```
for(i in 1:nrow(ss)) {
  t1 <- ss$t0[i]
  t2 <- ss$t0[i] + ss$tlen[i]
  print (sprintf("segment %d: t1=%d t2=%d elapsed time=%d seconds\n", i, t1, t2, ss$tlen[i]))
  shrink <- 10
  plotTripSegment(trip, t1-20, t2+20, b.marks=c( t1+shrink, t2-shrink ))
}
```

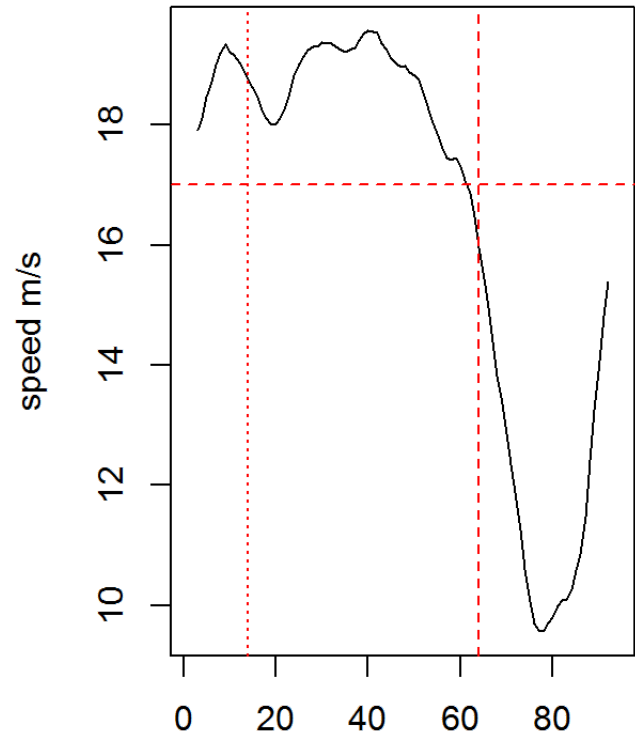
```
## [1] "segment 1: t1=4 t2=74 elapsed time=70 seconds\n"
```

Plot of Route

distance traveled: 1.5 km
direction= 24 deg
current speed= 58.3 km/h
acceleration= 0.8 m/s²



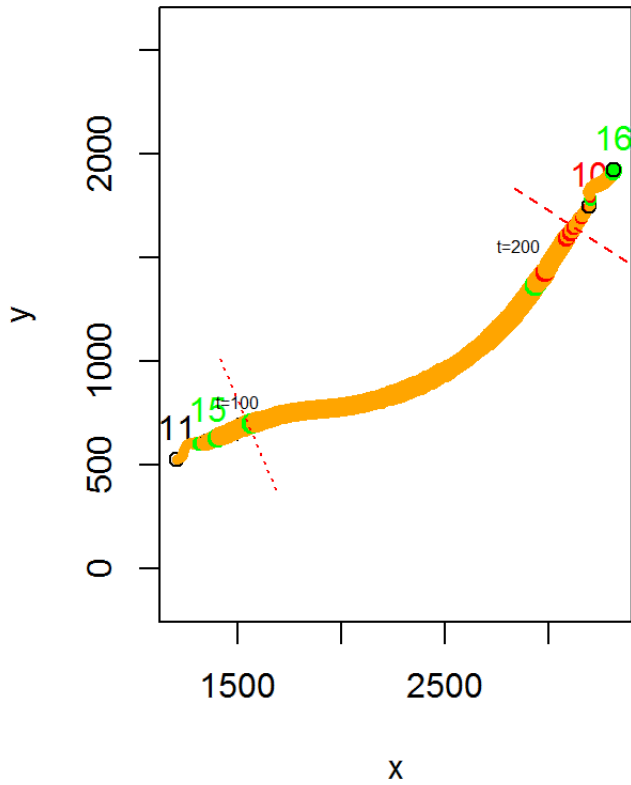
Speed (MA)



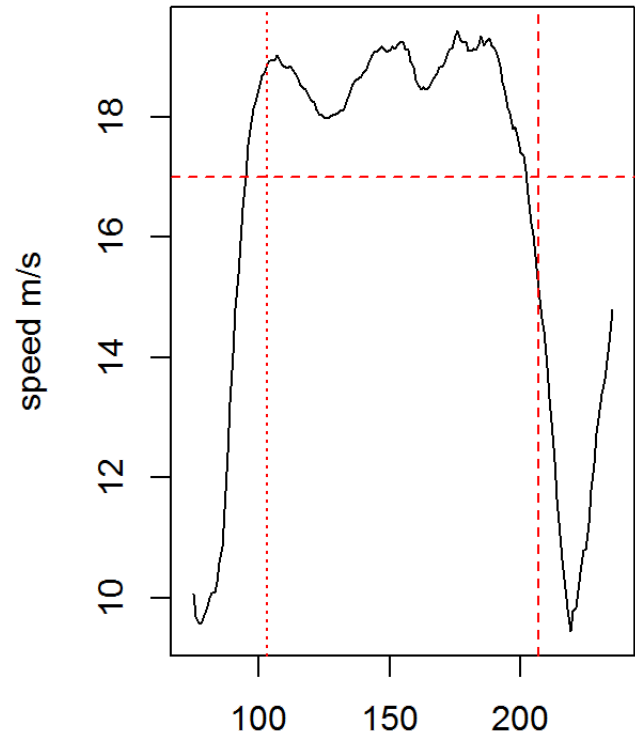
```
## [1] "segment 2: t1=93 t2=217 elapsed time=124 seconds\n"
```

Plot of Route

distance traveled: 2.7 km
direction= 55 deg
current speed= 57.7 km/h
acceleration= 1.1 m/s²



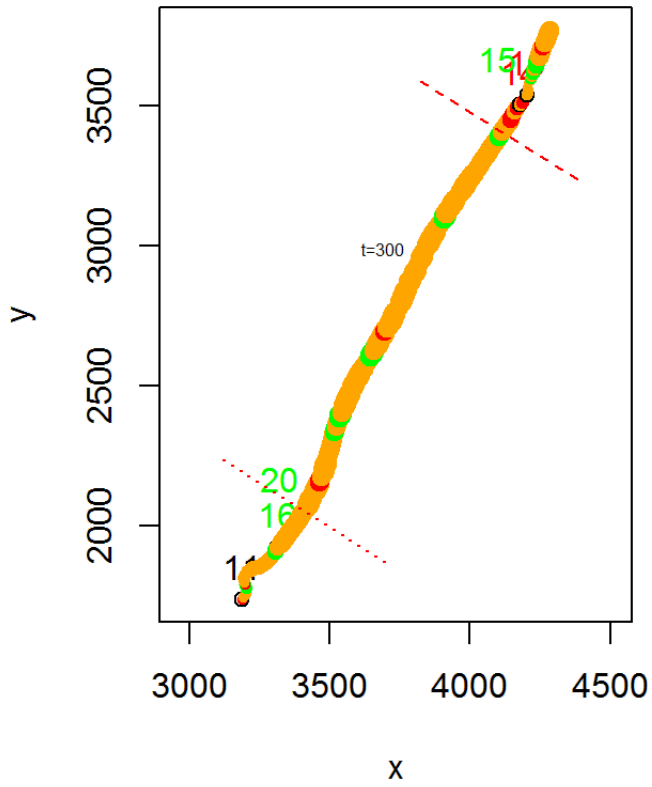
Speed (MA)



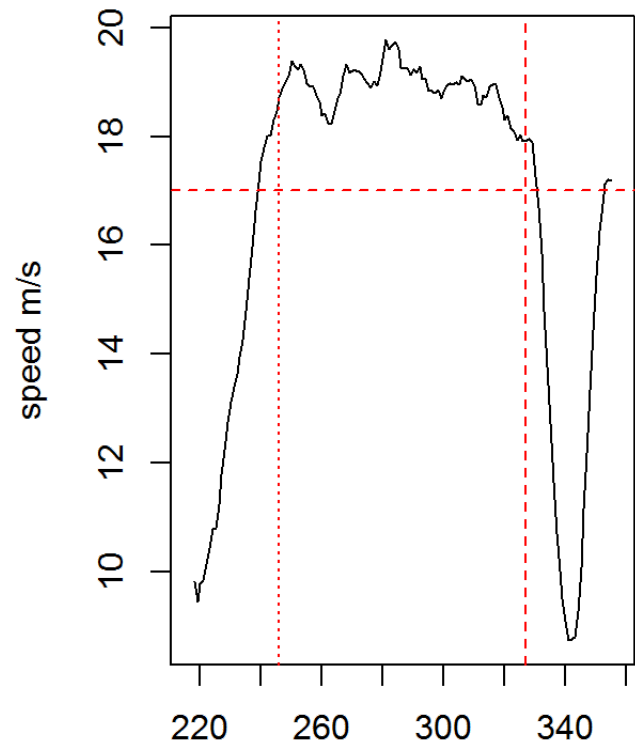
```
## [1] "segment 3: t1=236 t2=337 elapsed time=101 seconds\n"
```

Plot of Route

distance traveled: 2.4 km
direction= 69 deg
current speed= 61.9 km/h
acceleration= -0.1 m/s^2



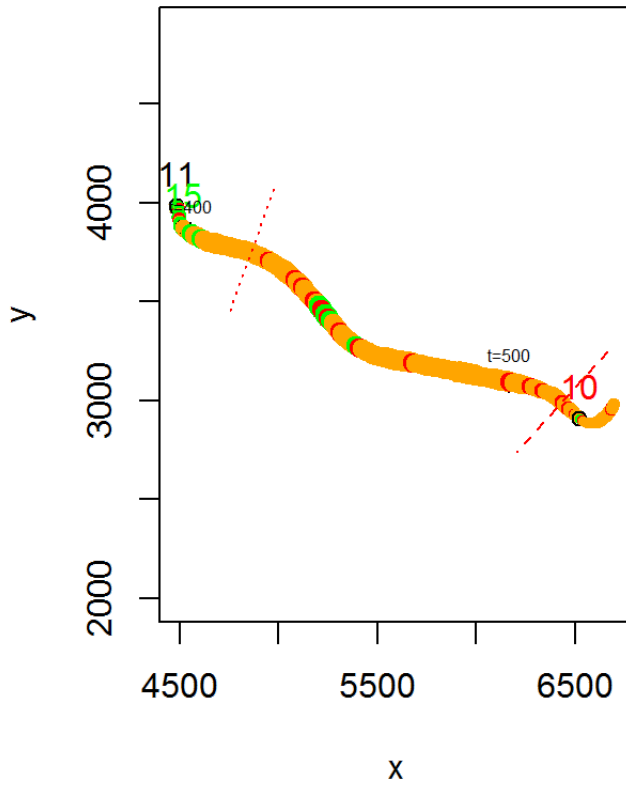
Speed (MA)



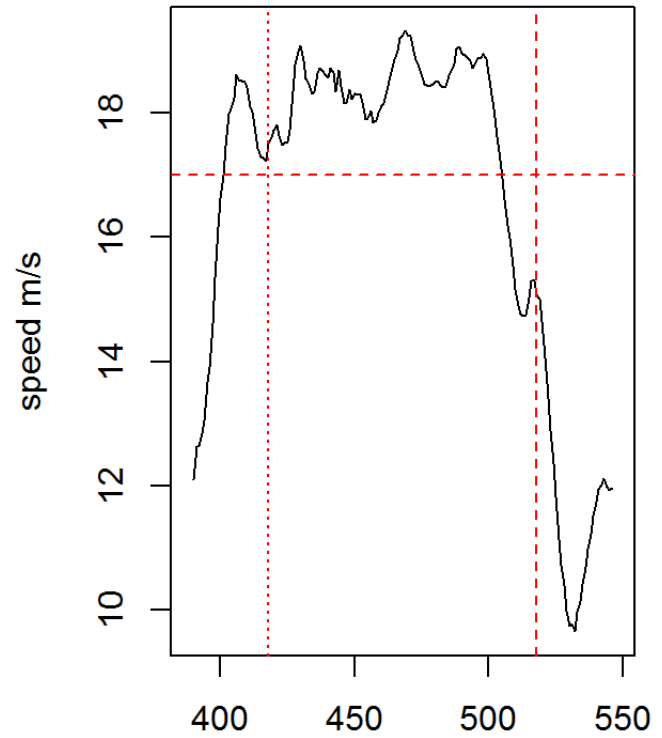
```
## [1] "segment 4: t1=408 t2=528 elapsed time=120 seconds\n"
```

Plot of Route

distance traveled: 2.6 km
direction= 63 deg
current speed= 44.4 km/h
acceleration= 0.6 m/s²

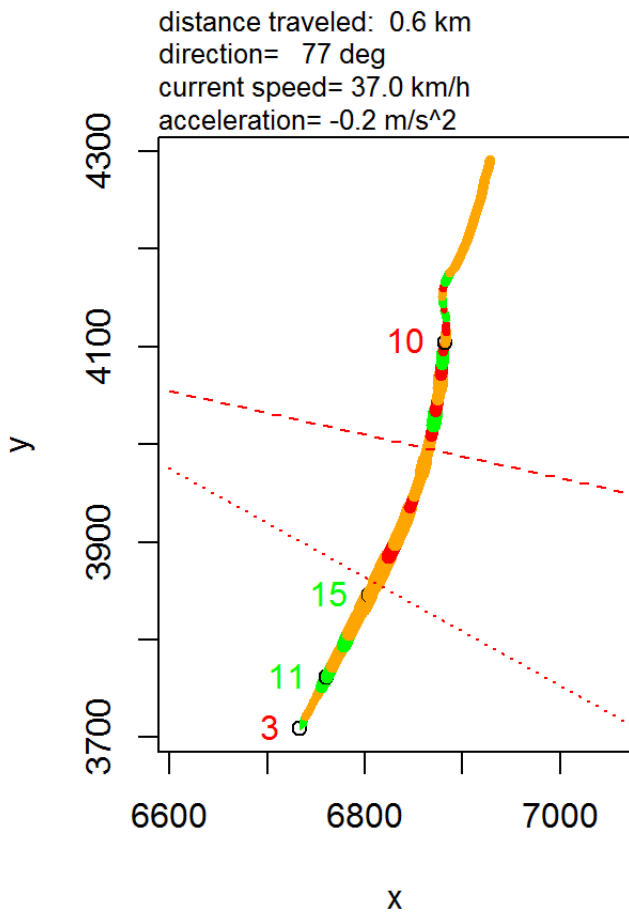


Speed (MA)

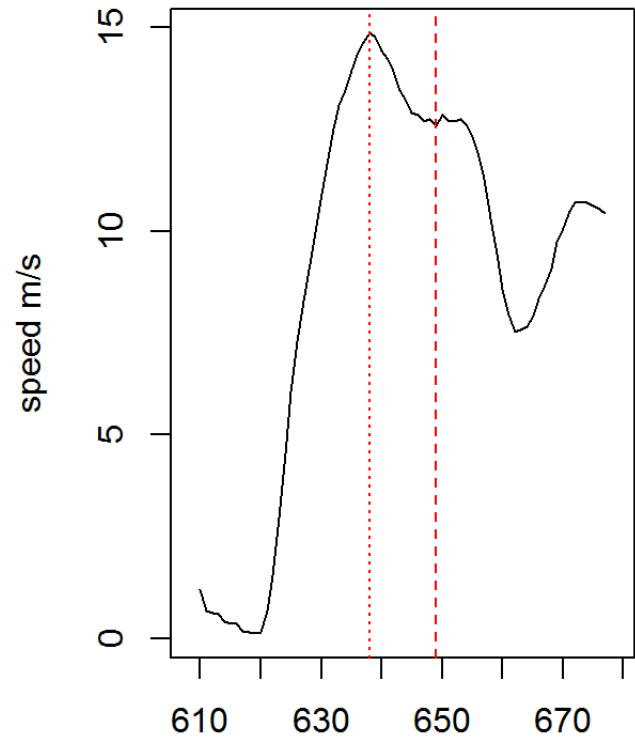


```
## [1] "segment 5: t1=628 t2=659 elapsed time=31 seconds\n"
```


Plot of Route



Speed (MA)



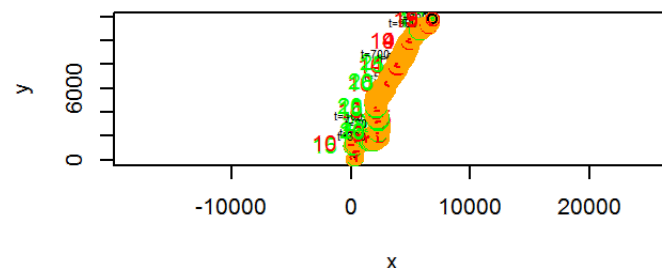
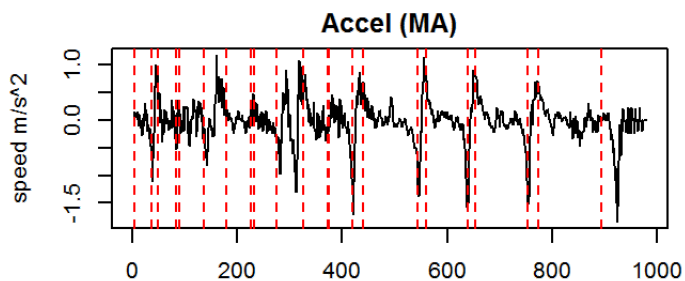
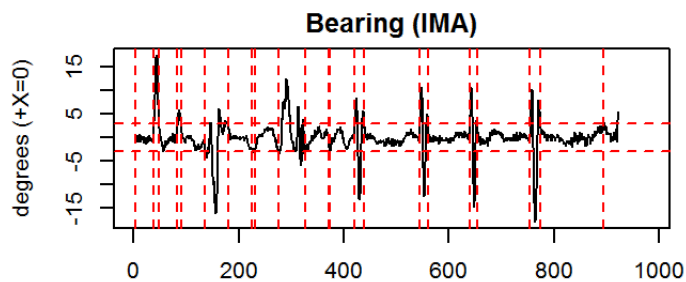
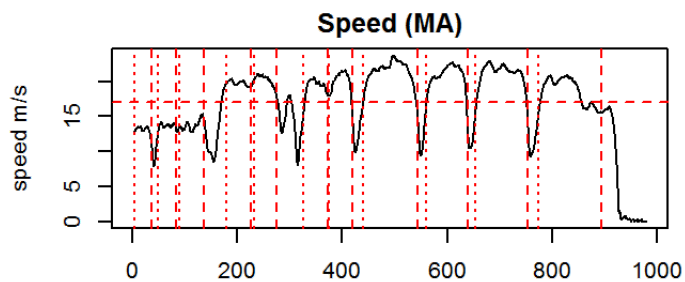
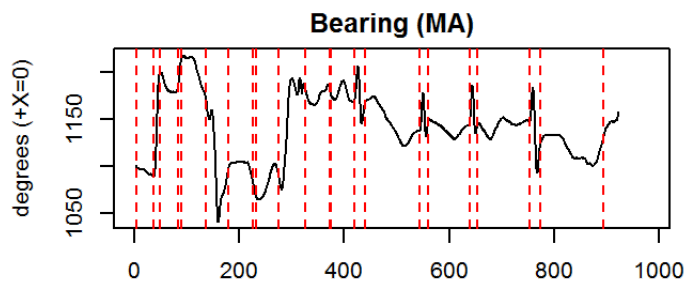
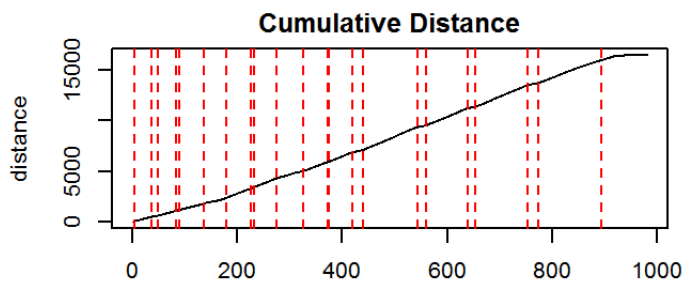
- Segments 1 & 2 (S1:2) look similar with the segments ending during the deceleration
- S3 ends before the deceleration ... I can't see any variable that accounts for the difference
- S4 looks like S1:2, but it has a slight acceleration midway through the braking
- S5 is only 31 seconds, and has a very different profile than the others. I think it will be important to treat short segments as a different animal
- S1:4 ≥ 70 secs

Repeat for two more trips:

```
trip.id = 199
```

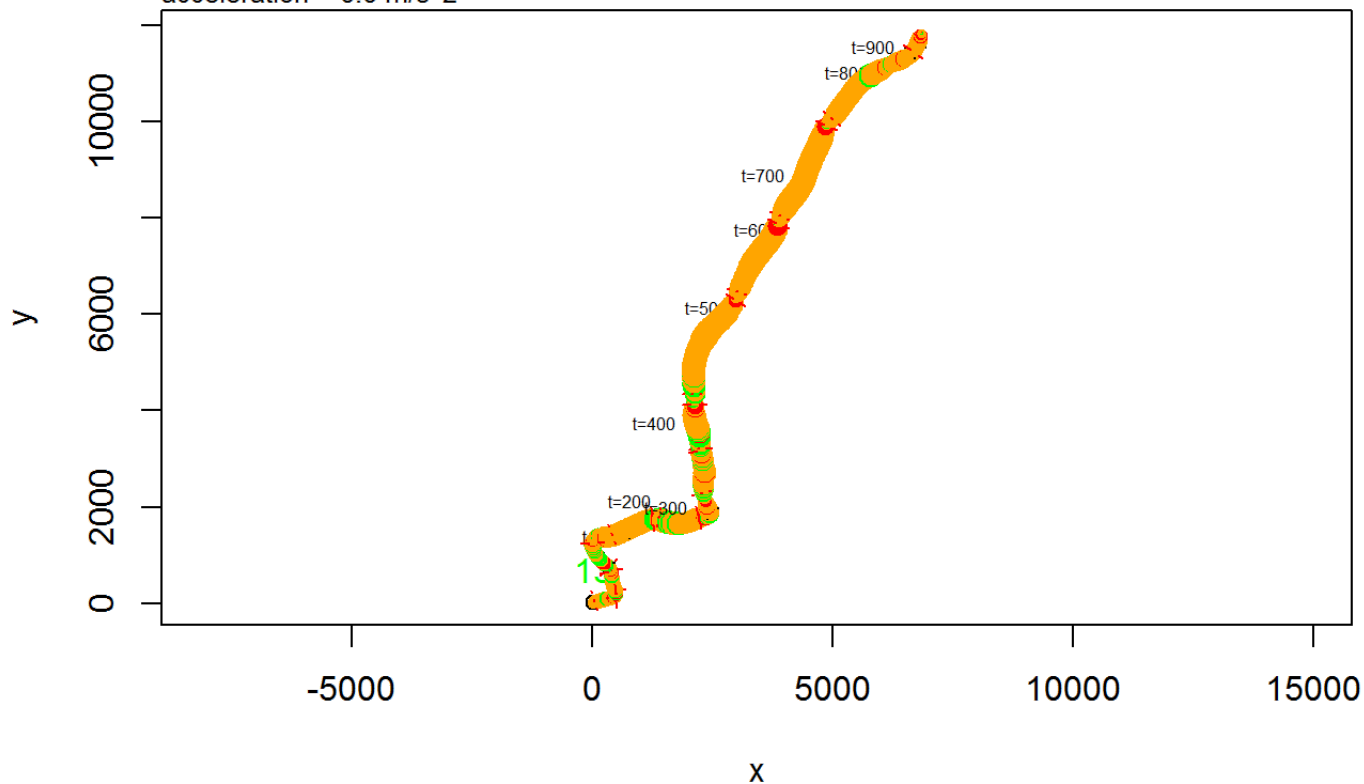
```
tripDetails <- function (driver.id, trip.id) {  
  trip <- getTrip( driver.id, trip.id )  
  ss <- segment.parse.bearing(trip)  
  cat (nrow(ss), "segments were detected for trip", trip.id, ".\n")  
  plotTripSegment6(trip, 1, 20000, b.marks=c(ss$t0, ss$t0+ss$tlen))  
  plotTrip(trip, v.mark=50, b.marks=c(ss$t0, ss$t0+ss$tlen))  
  
  for(i in 1:nrow(ss)) {  
    t1 <- ss$t0[i]  
    t2 <- ss$t0[i] + ss$tlen[i]  
    print (sprintf("segment %d: t1=%d t2=%d elapsed time=%d seconds\n", i, t1, t2, ss$tlen[i]))  
    shrink <- 10  
    plotTripSegment(trip, t1-20, t2+20, b.marks=c( t1+shrink, t2-shrink  ) )  
  }  
  return(ss)  
}  
ss <- tripDetails( 2591, trip.id)
```

```
## 11 segments were detected for trip 199 .
```



Plot of Route

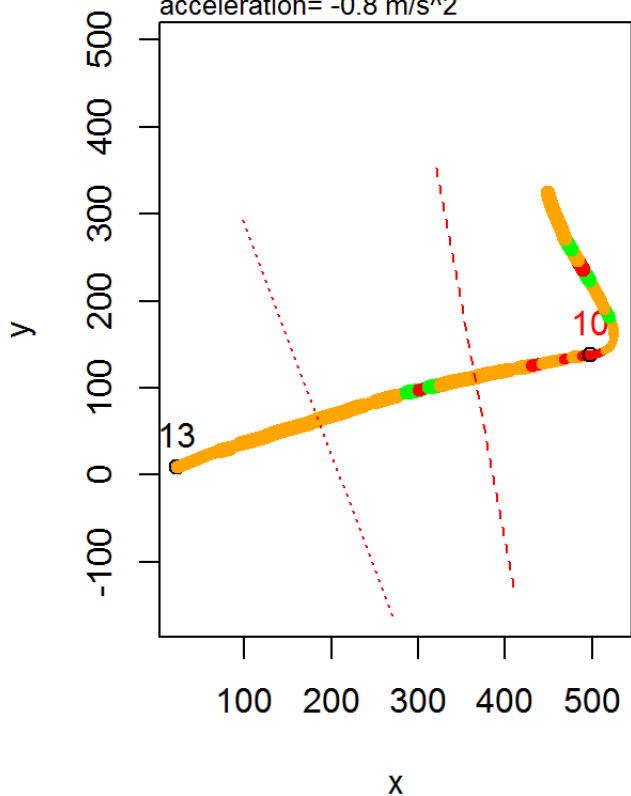
distance traveled: 16.4 km
direction= 270 deg
current speed= 0.0 km/h
acceleration= 0.0 m/s^2



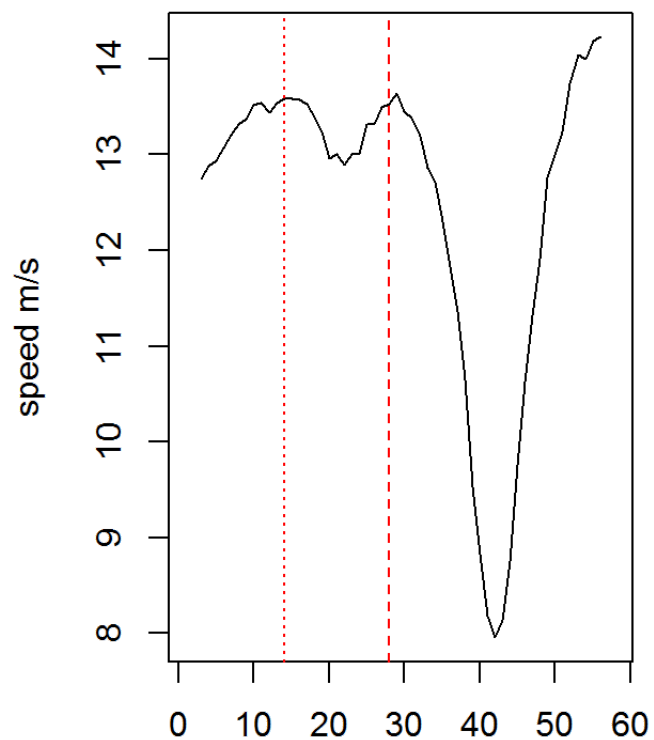
```
## [1] "segment 1: t1=4 t2=38 elapsed time=34 seconds\n"
```

Plot of Route

distance traveled: 0.7 km
direction= 108 deg
current speed= 49.1 km/h
acceleration= -0.8 m/s^2



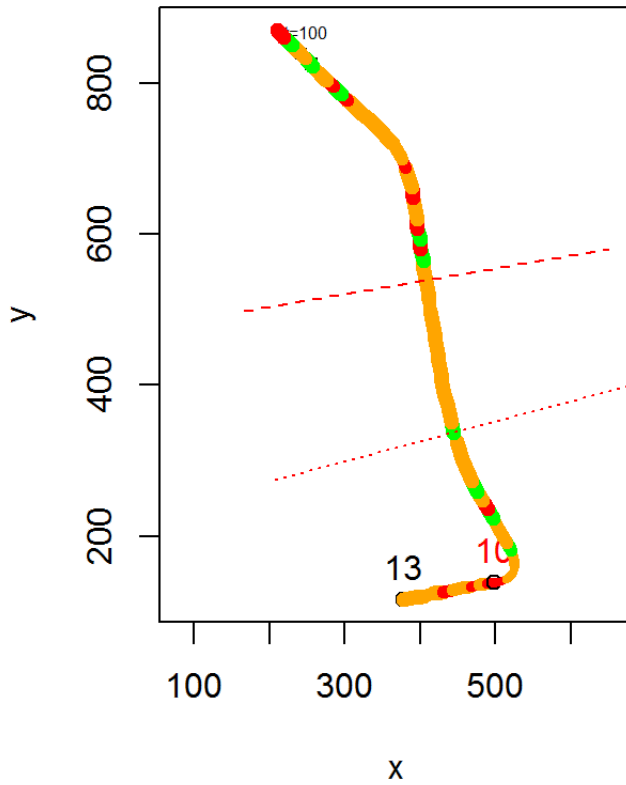
Speed (MA)



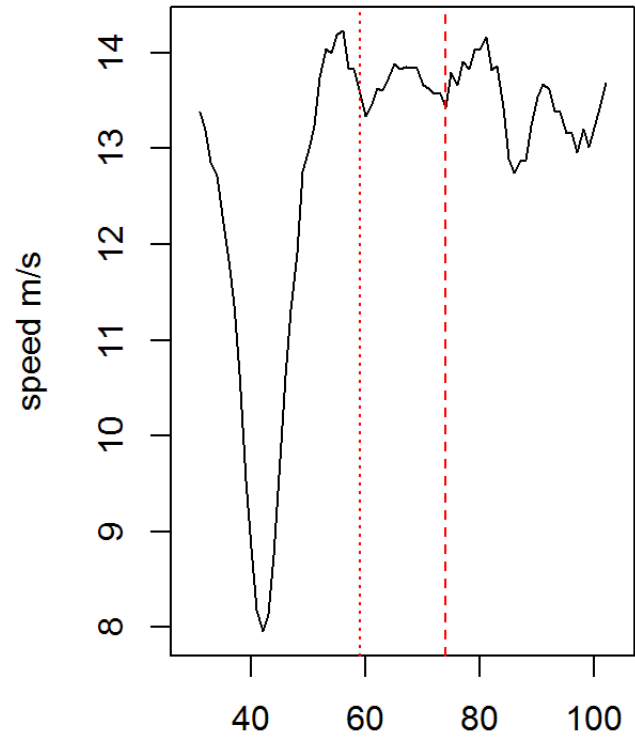
```
## [1] "segment 2: t1=49 t2=84 elapsed time=35 seconds\n"
```

Plot of Route

distance traveled: 1.0 km
direction= 134 deg
current speed= 49.4 km/h
acceleration= -1.1 m/s²



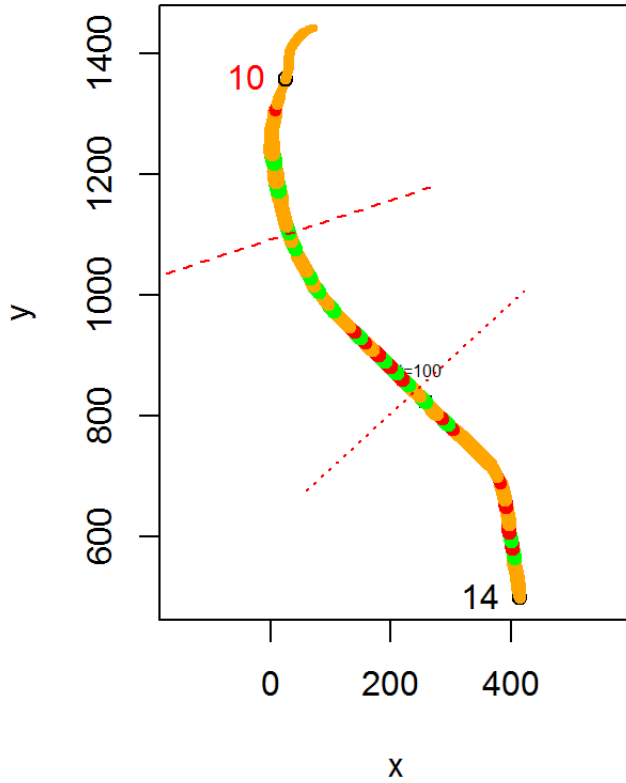
Speed (MA)



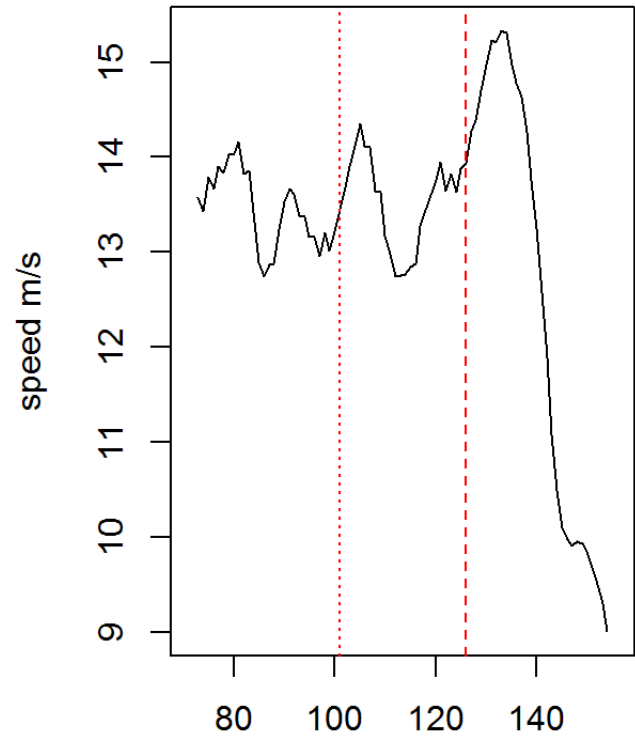
```
## [1] "segment 3: t1=91 t2=136 elapsed time=45 seconds\n"
```

Plot of Route

distance traveled: 1.1 km
direction= 19 deg
current speed= 29.3 km/h
acceleration= -0.6 m/s²



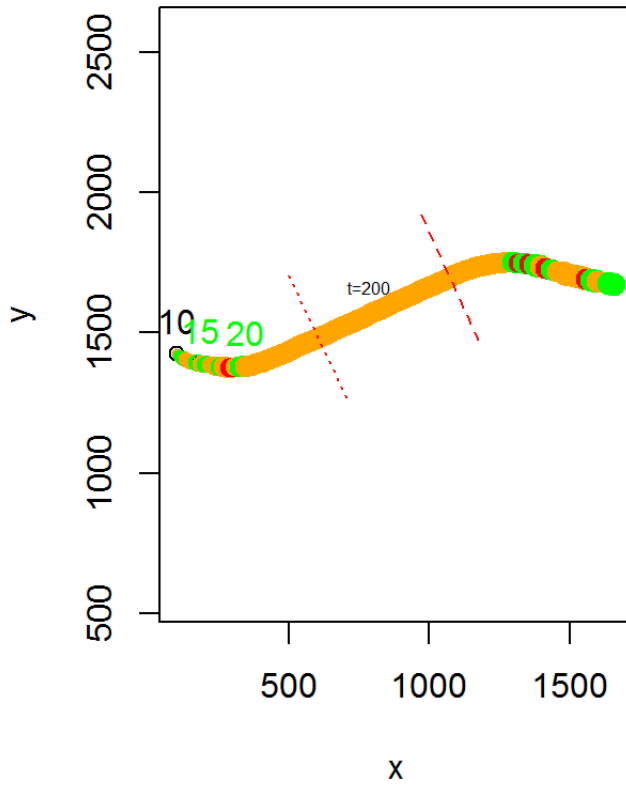
Speed (MA)



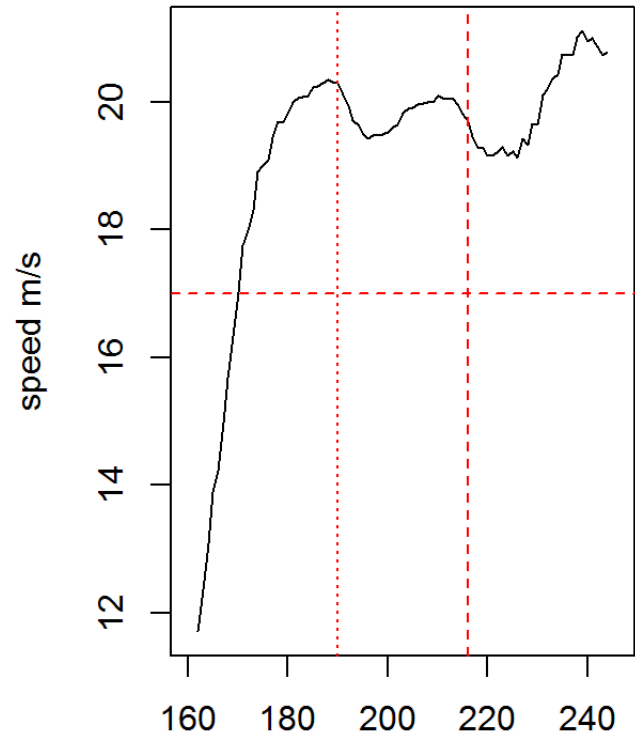
```
## [1] "segment 4: t1=180 t2=226 elapsed time=46 seconds\n"
```

Plot of Route

distance traveled: 1.6 km
direction= 349 deg
current speed= 77.9 km/h
acceleration= 1.1 m/s²



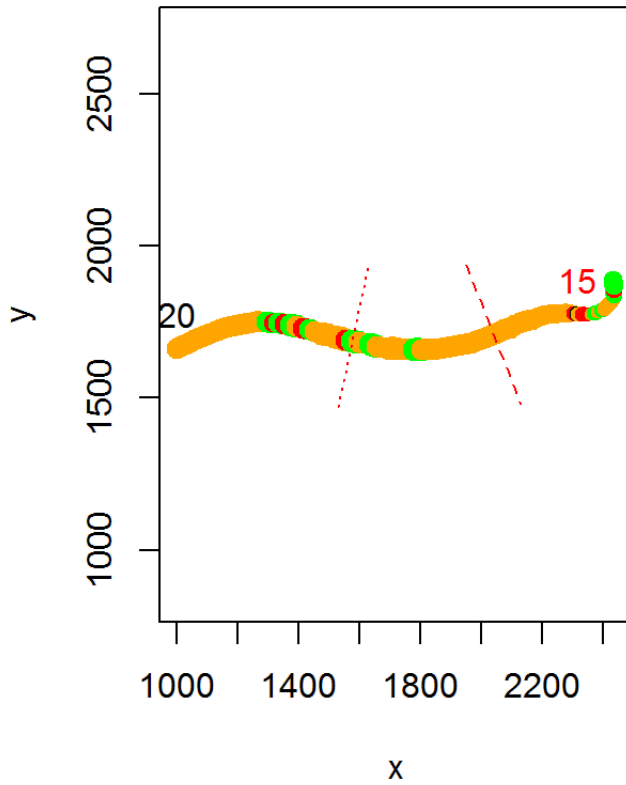
Speed (MA)



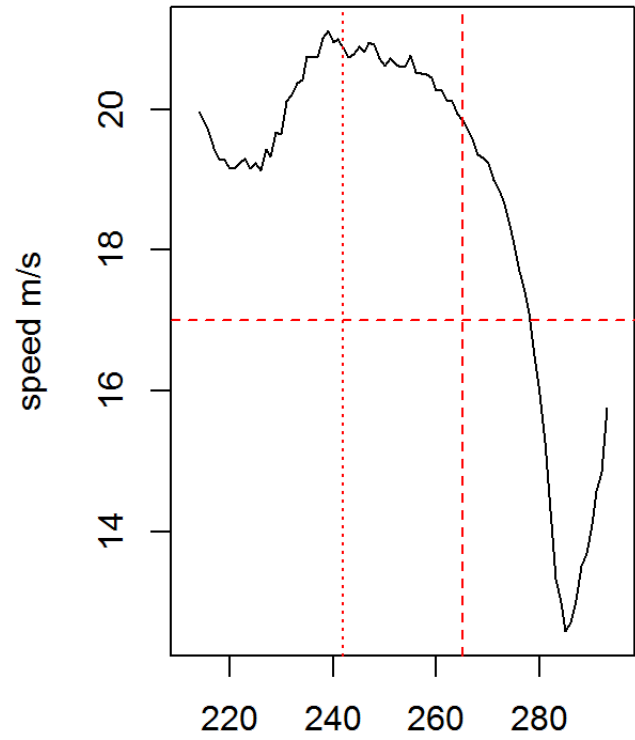
```
## [1] "segment 5: t1=232 t2=275 elapsed time=43 seconds\n"
```

Plot of Route

distance traveled: 1.6 km
direction= 104 deg
current speed= 67.1 km/h
acceleration= 3.4 m/s²



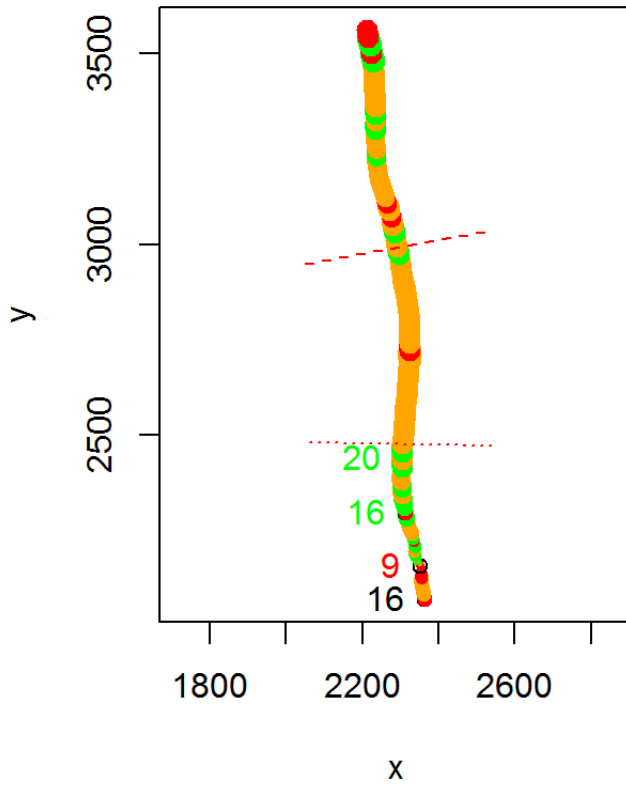
Speed (MA)



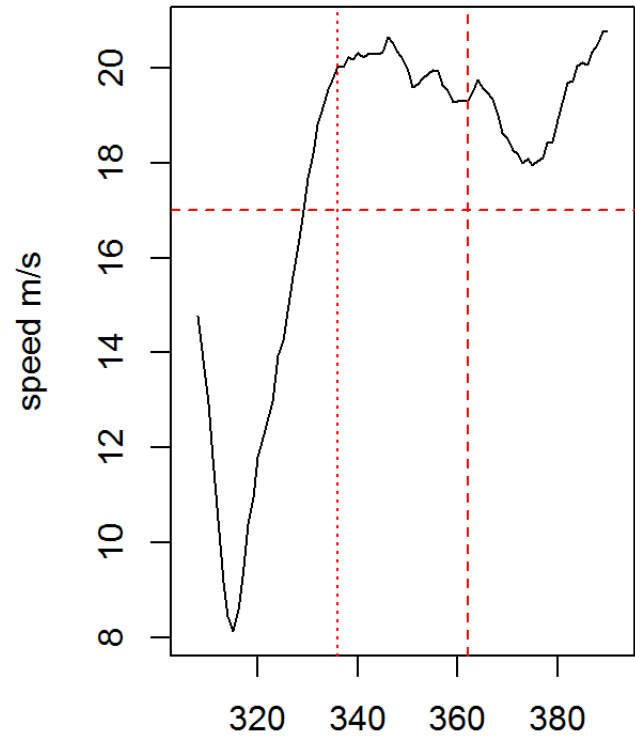
```
## [1] "segment 6: t1=326 t2=372 elapsed time=46 seconds\n"
```


Plot of Route

distance traveled: 1.5 km
direction= 103 deg
current speed= 73.5 km/h
acceleration= -1.3 m/s²



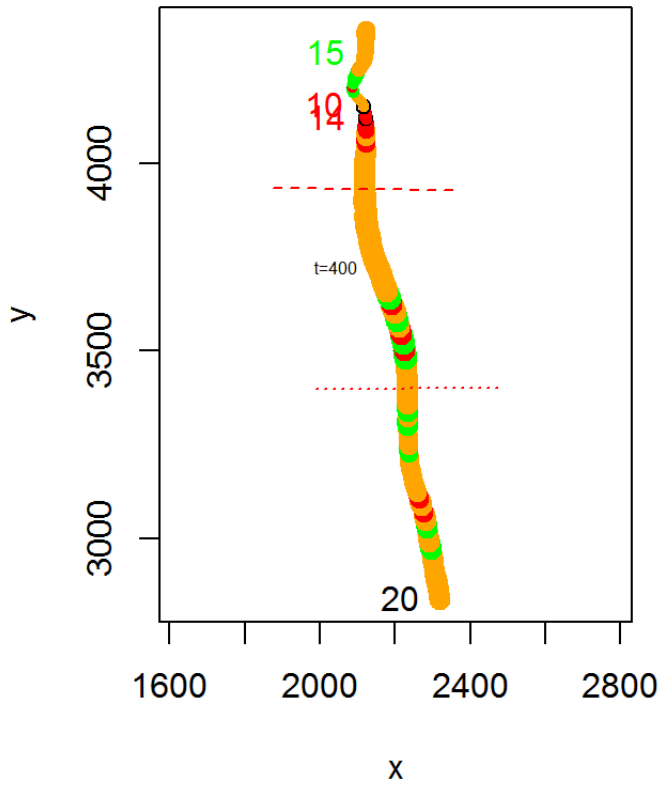
Speed (MA)



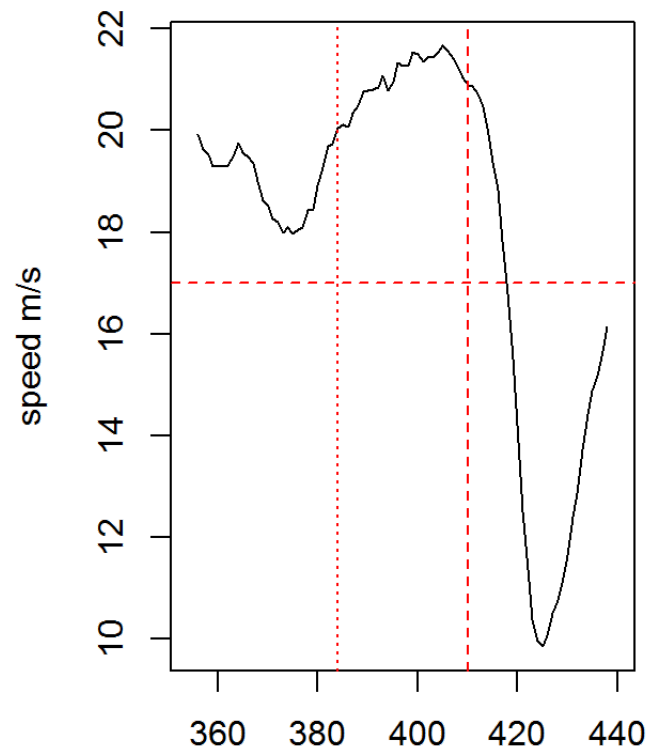
```
## [1] "segment 7: t1=374 t2=420 elapsed time=46 seconds\n"
```

Plot of Route

distance traveled: 1.6 km
direction= 88 deg
current speed= 62.3 km/h
acceleration= 0.6 m/s²



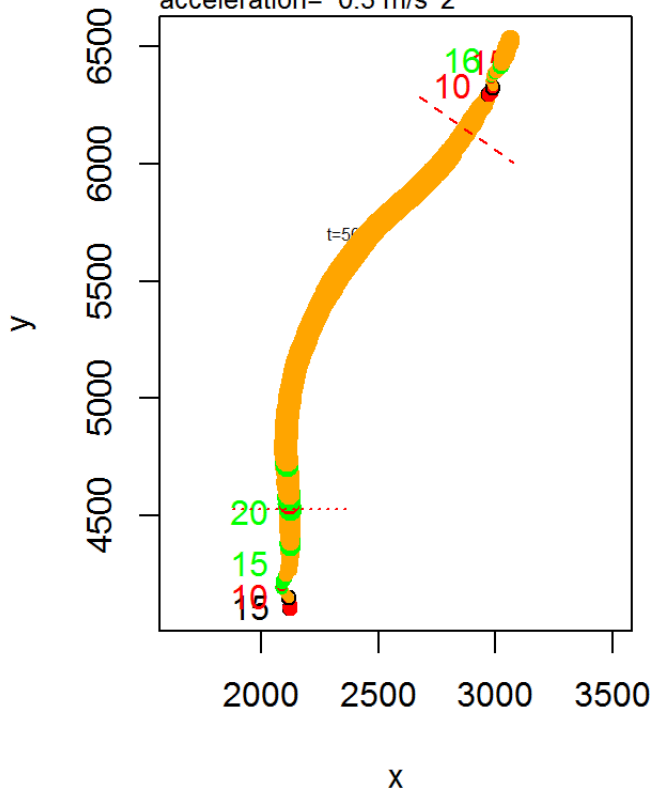
Speed (MA)



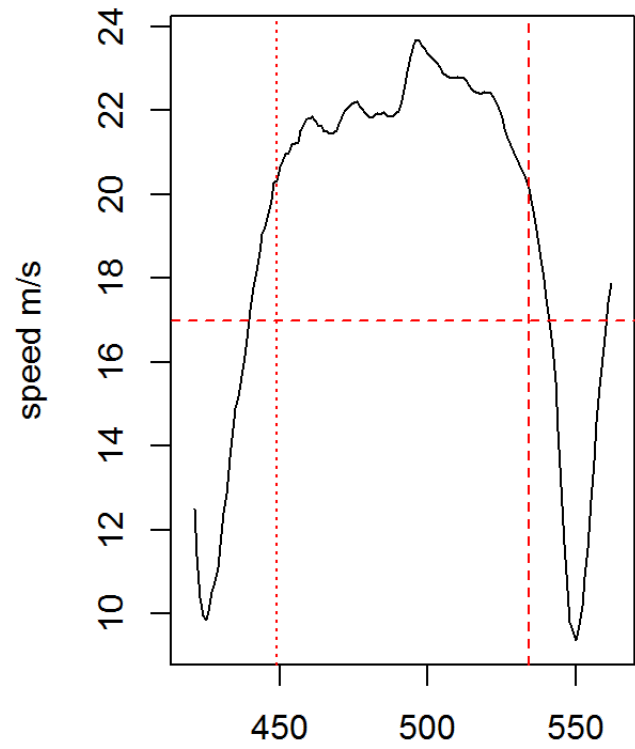
```
## [1] "segment 8: t1=439 t2=544 elapsed time=105 seconds\n"
```

Plot of Route

distance traveled: 2.8 km
direction= 70 deg
current speed= 66.7 km/h
acceleration= 0.3 m/s²



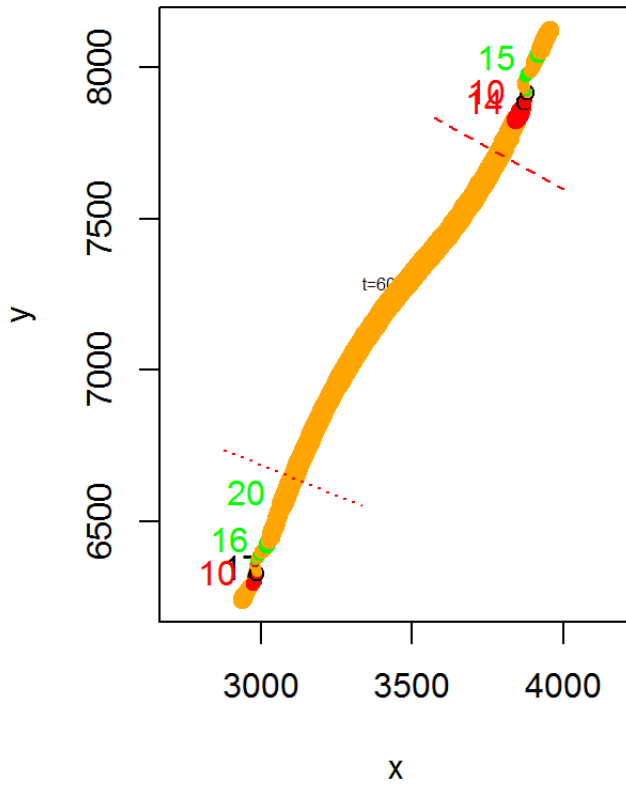
Speed (MA)



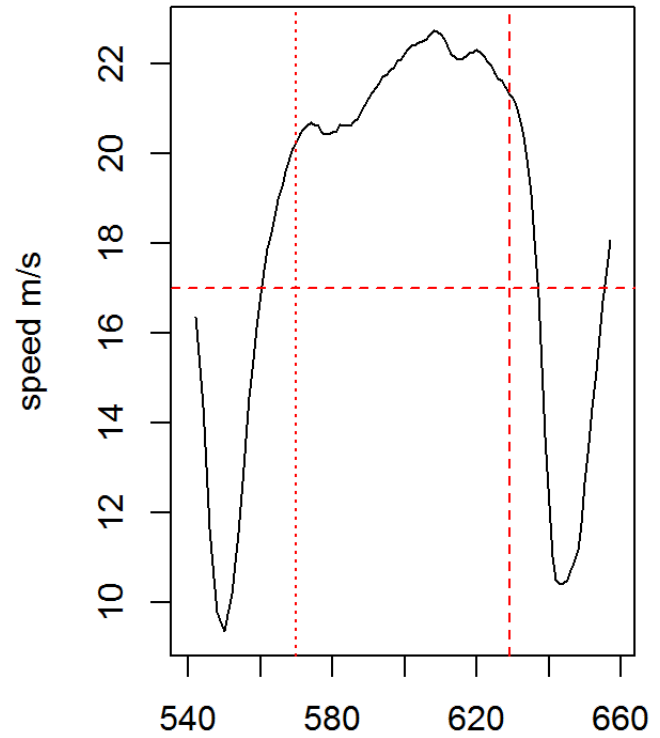
```
## [1] "segment 9: t1=560 t2=639 elapsed time=79 seconds\n"
```

Plot of Route

distance traveled: 2.2 km
direction= 62 deg
current speed= 67.5 km/h
acceleration= -0.1 m/s²



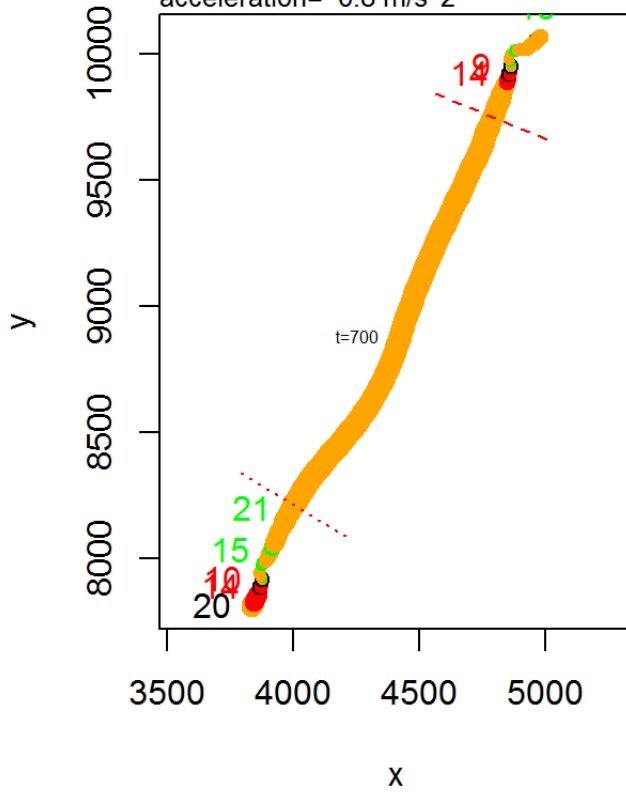
Speed (MA)



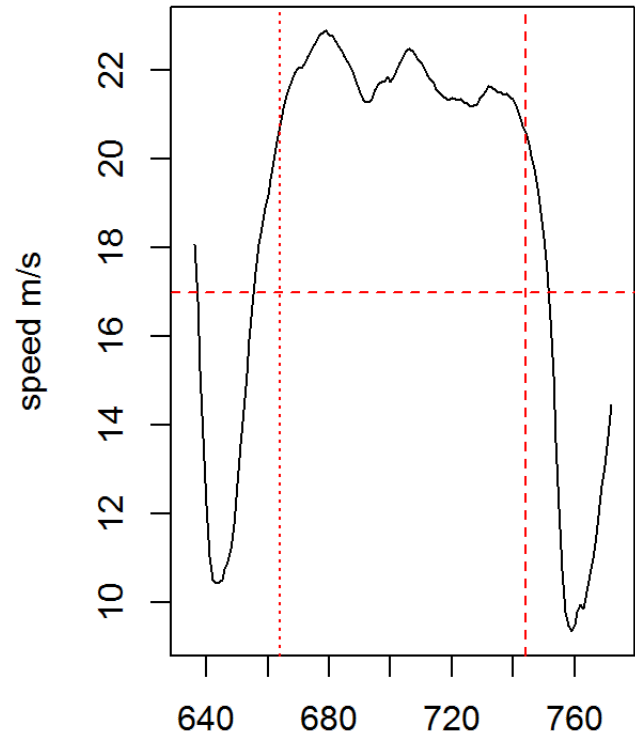
```
## [1] "segment 10: t1=654 t2=754 elapsed time=100 seconds\n"
```

Plot of Route

distance traveled: 2.6 km
direction= 48 deg
current speed= 57.4 km/h
acceleration= 0.8 m/s²



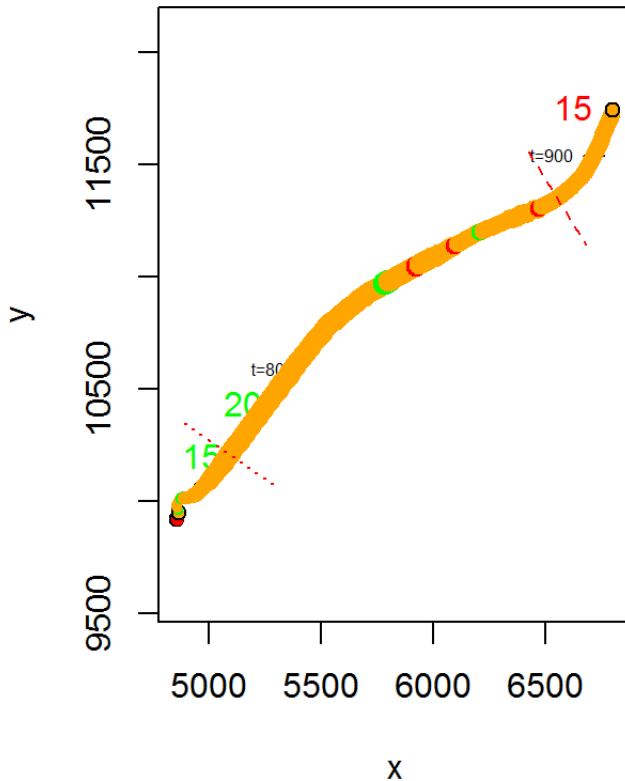
Speed (MA)



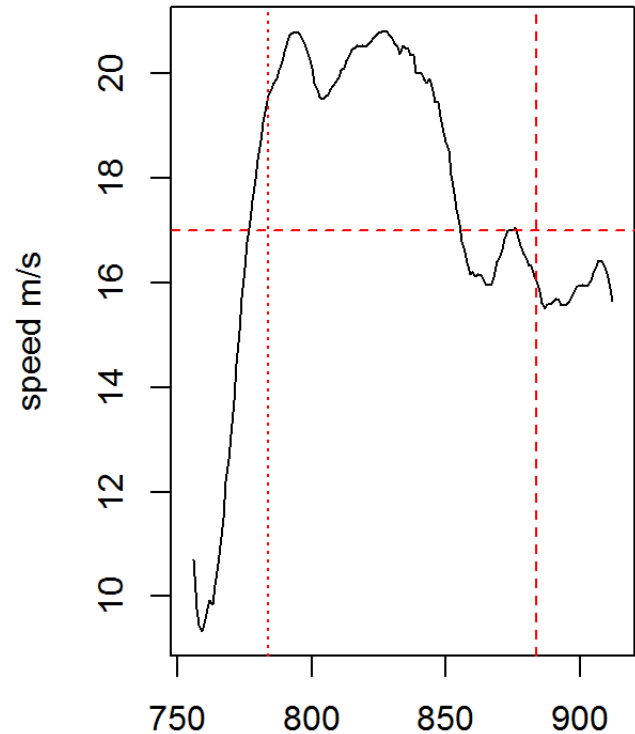
```
## [1] "segment 11: t1=774 t2=894 elapsed time=120 seconds\n"
```

Plot of Route

distance traveled: 2.8 km
 direction= 65 deg
 current speed= 53.2 km/h
 acceleration= -0.8 m/s²



Speed (MA)

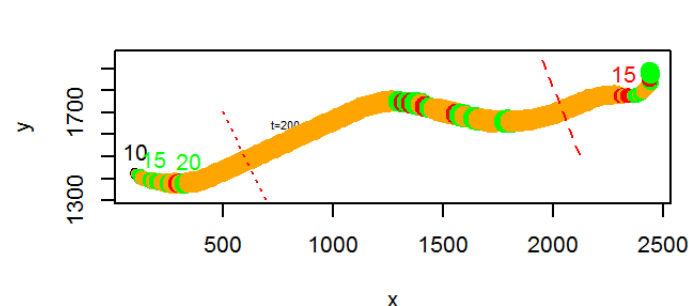
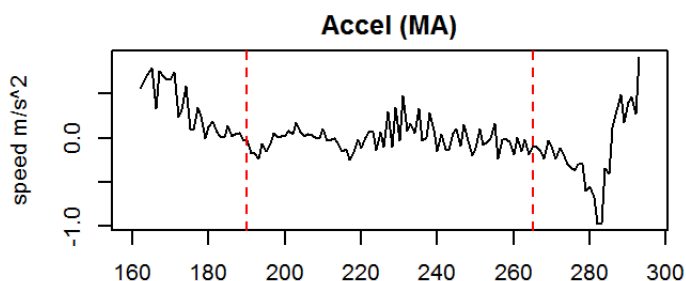
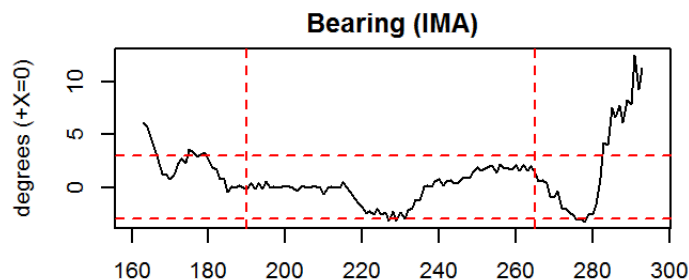
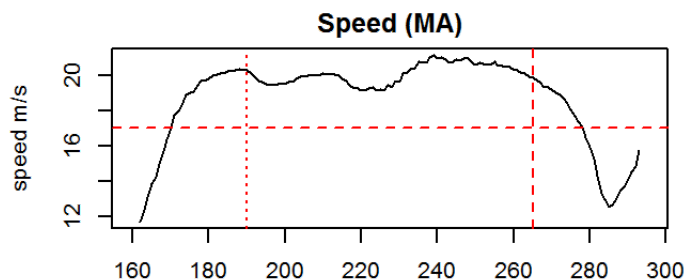
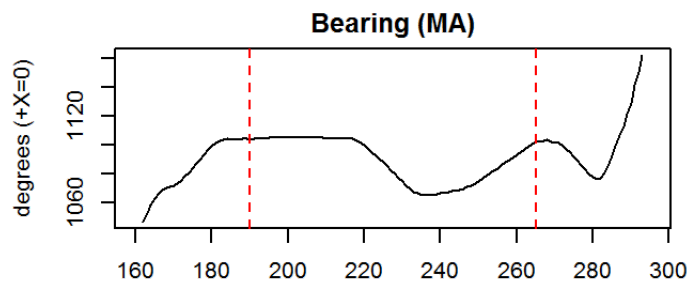
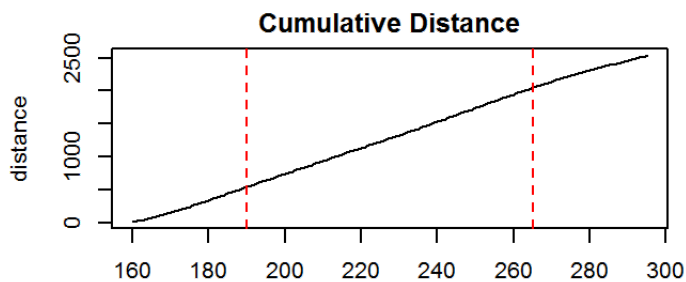


Trip 199

- 11 segments found, but subjectively (from the speed plot) it looks to be between 7-9
- only 4 segments are > 70 sec (the last 4)
- S2:3, S4:5 and S6:7 look like they should be combined
- ignore S2:3 for now since they fall below the 17 m/s threshold
- S4:5 looks like a legit division, since there is bend in the road (plot below). Looks like the driver kept his speed, but I think we want to capture this as a curve between 2 segments, rather than one long segment (so the algorithm got it right)
- S6:7 is still suspect

Here's S4:5 plotted as if it had been combined

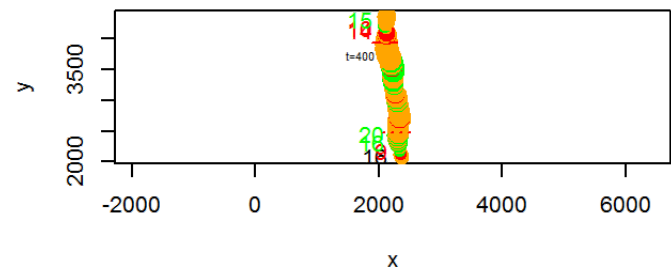
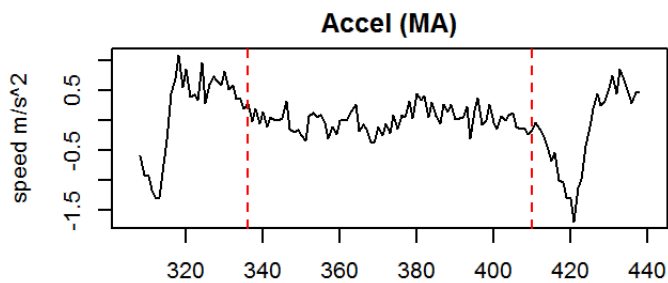
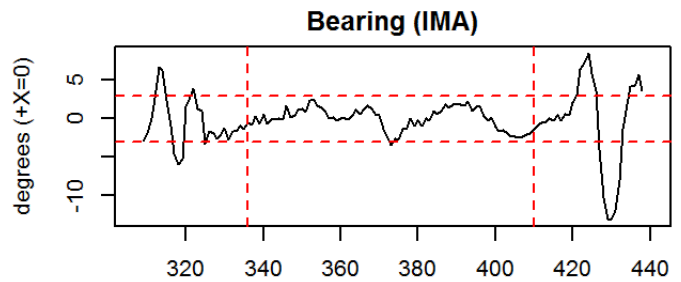
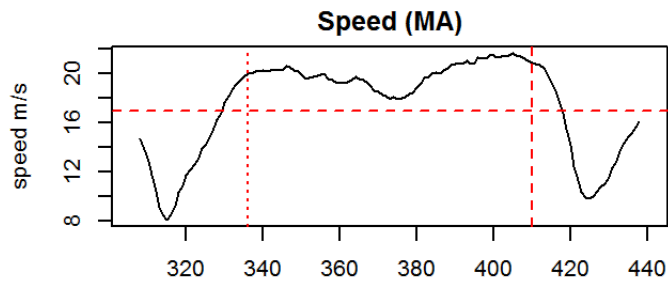
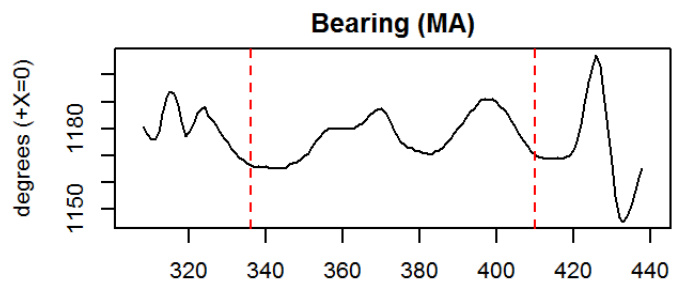
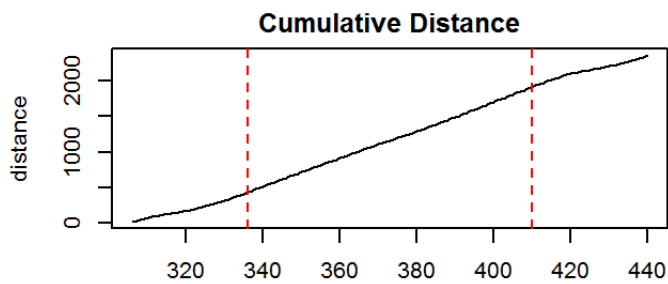
```
t1 <- ss$t0[4]
t2 <- ss$t0[5] + ss$tlen[5]
shrink <- 10
trip <- getTrip( driver.id, trip.id )
plotTripSegment6(trip, t1-20, t2+20, b.marks=c( t1+shrink, t2-shrink ) )
```



Looking at this, I'm torn between whether this should be one or two segments (contrary to what I wrote above). It looks like this just *barely* nicked out of the 3 degree zone, so that arbitrary choice will have a big effect on our results. For now, let's just note it and move on. It might not matter at all since I think the main use of this will be to find accel/deceleration segments, and we've still done that with the two segments (except we'll drop the end of S4 and the beginning of S5).

It looks like the same issue for S6:7

```
t1 <- ss$t0[6]
t2 <- ss$t0[7] + ss$tlen[7]
shrink <- 10
plotTripSegment6(trip, t1-20, t2+20, b.marks=c( t1+shrink, t2-shrink ))
```

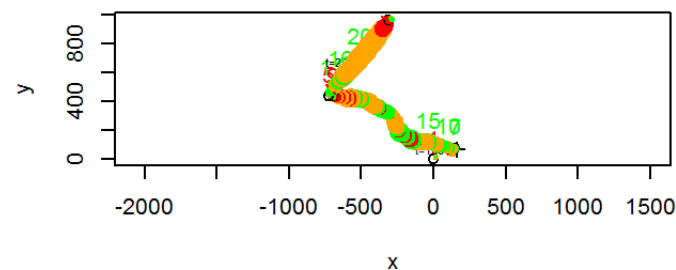
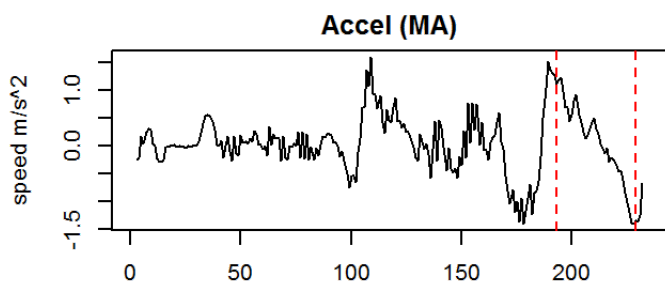
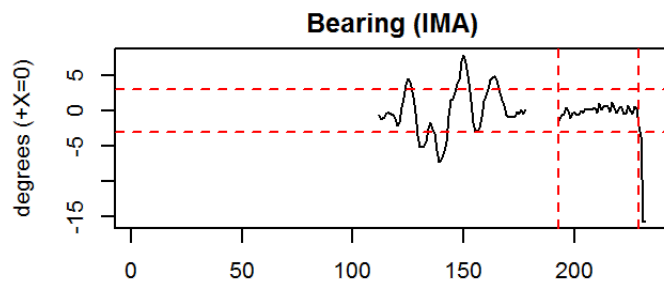
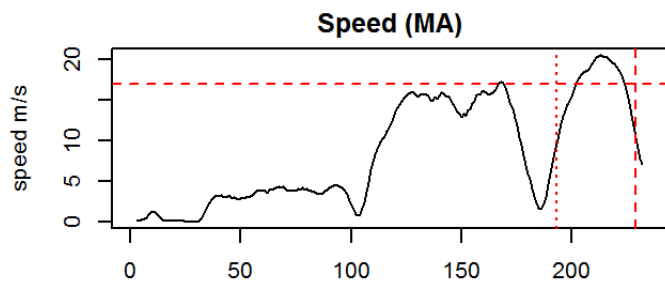
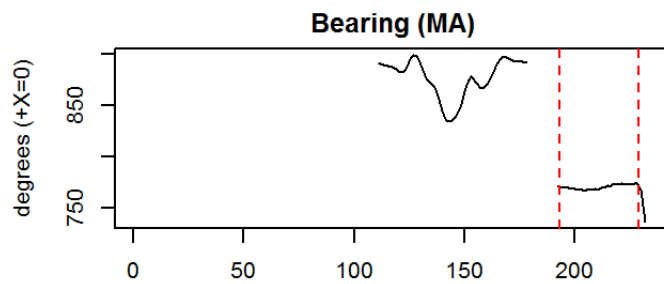
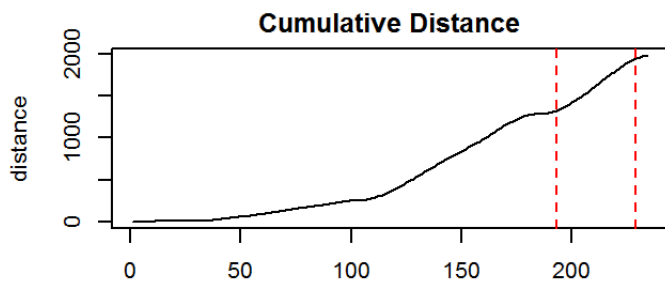


```
trip.id <- 200
trip <- getTrip( driver.id, trip.id )
```

Trip 200

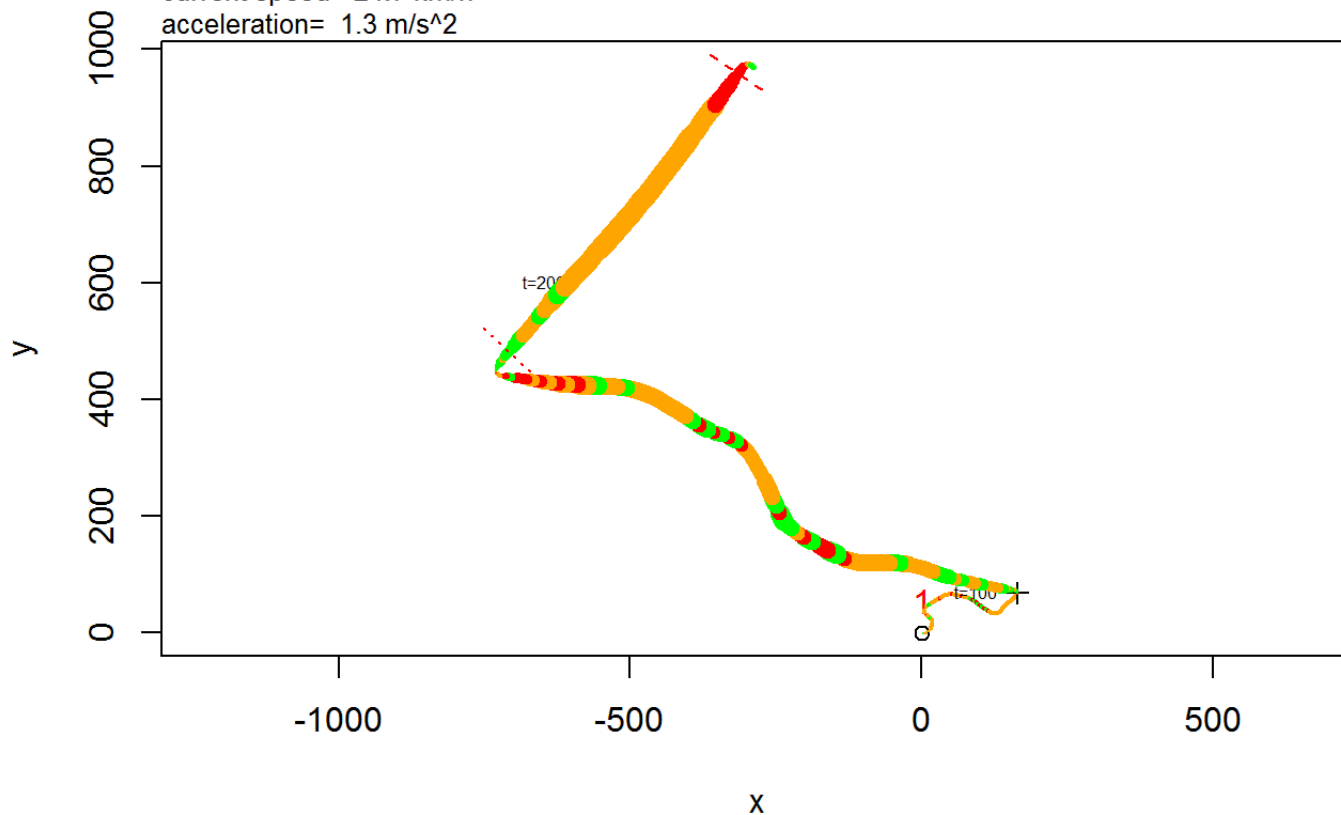
```
ss <- tripDetails( 2591, trip.id)
```

```
## 1 segments were detected for trip 200 .
```

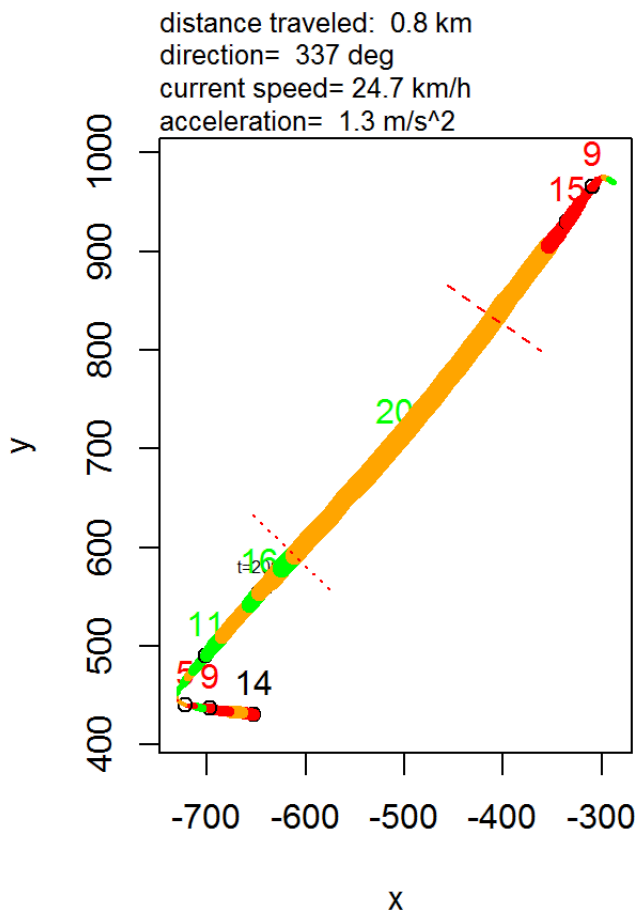
Plot of Route

distance traveled: 2.0 km
 direction= 337 deg
 current speed= 24.7 km/h
 acceleration= 1.3 m/s²

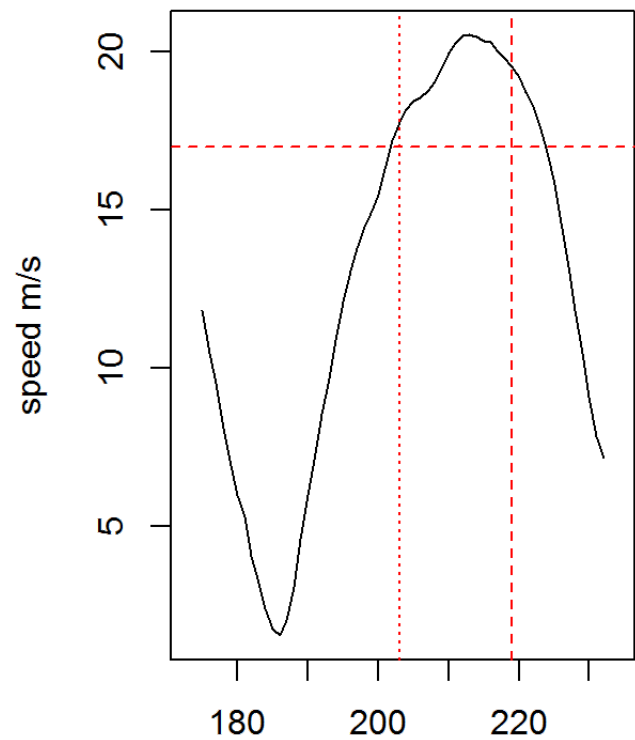


```
## [1] "segment 1: t1=193 t2=229 elapsed time=36 seconds\n"
```

Plot of Route



Speed (MA)



- Only one (short) segment found
- This the reason I changed the MIN_SEGMENT_LENGTH to 30

Summary of all trip segments for this driver

```

seg.summary <- data.frame( n.seg=integer(),
                           seg.min=numeric(), seg.max=numeric(), seg.avg=numeric(), seg.sd=numeric())
for(i.trip in 1:200) {
  trip <- getTrip( driver.id, i.trip )
  ss <- segment.parse.bearing(trip)
  seg.summary[i.trip, ]$n.seg <- nrow(ss)
  if ( nrow(ss) > 0) {
    seg.summary[i.trip, ]$seg.min <- min(ss$tlen)
    seg.summary[i.trip, ]$seg.max <- max(ss$tlen)
    seg.summary[i.trip, ]$seg.avg <- mean(ss$tlen)
    seg.summary[i.trip, ]$seg.sd <- sd(ss$tlen)
  }
}
print(summary(seg.summary))

```

```

##      n.seg      seg.min      seg.max      seg.avg
## Min.   : 0.000   Min.   : 30.00   Min.    : 30.0   Min.    : 30.00
## 1st Qu.: 1.000   1st Qu.: 34.00   1st Qu.: 63.0   1st Qu.: 56.21
## Median : 2.000   Median : 41.00   Median :108.0   Median : 72.50
## Mean   : 3.725   Mean   : 72.61   Mean    :128.3   Mean    : 95.59
## 3rd Qu.: 7.000   3rd Qu.: 68.00   3rd Qu.:202.5   3rd Qu.:111.43
## Max.   :14.000   Max.   :276.00   Max.    :324.0   Max.    :276.00
##
##      NA's      :25      NA's      :25      NA's      :25
##
##      seg.sd
## Min.    : 0.7071
## 1st Qu.: 18.4210
## Median : 26.8779
## Mean    : 38.6382
## 3rd Qu.: 53.0330
## Max.    :164.0488
## NA's    :91

```

```

seg0 <- which(seg.summary$n.seg == 0)
seg0

```

```

## [1]  6 16 40 45 46 60 72 79 82 104 115 122 123 131 133 147 152
## [18] 153 154 155 157 163 187 196 198

```

The bad news is that we've got 25 trips with no segments, so we'll need to look for something other than straight segments, unless there is a tweak that will cause segments to be found. I'll examine that in the next part.