MICE MEMORY

J.A.Hartigan Yale University
September 2013

1 Data Sources

Adam Kaufman from the Yale Medical School presented these data on 30 March 2012 for the Yale Statistics Clinic. Susan Wang of the Yale Statistics Department assisted in the analysis .

2 Mice with Altzheimer's disease

The mice used in the experiment are bred from a pure strain in which one of the parents has a genetic mutation known to be associated with Altzheimer's. The descendant mice are tested for the presence or absence of the mutation. In the experiment there were 6 mice without mutation(WT) and 15 mutated mice(APP), all about the same age. (The APP stands for Amyloid Precursor Protein.)

3 The memory experiment

Each mouse had its memory tested as follows. The mouse was placed in a tub full of water at a random place around the edge . A platform was placed just beneath the water surface, invisible to the mouse. The amount of time in seconds until the mouse found the platform was measured. However, the mouse was taken from the tub if the swimming time exceeds 60 seconds.

The experiment was repeated 24 times over a 3 day period, with 4 swims in the morning, and 4 swims in the afternoon on each day. The whole sequence of experiments is repeated after a few days, after the mouse is given an intervention drug which is expected to ameleoriate the alzheimer's in the APP mice. The platform location is changed.
A third sequence of experiments is repeated after another week, to test if the effectiveness of the drug is sustained.
All 21 mice go through the three experiment sequence.

The aim is to discover if receiving the drug reduces the search times after a few trials in the APP mice, for the after drug experiments compared to the before drug experiments.

5 First plots

In later runs, start here:

```
baw <-read.csv("data/MiceMemory.csv", as.is=T,
  header=T)
dim(baw)
```

```
[1] 21 74
```

```
head(baw, 2)
```

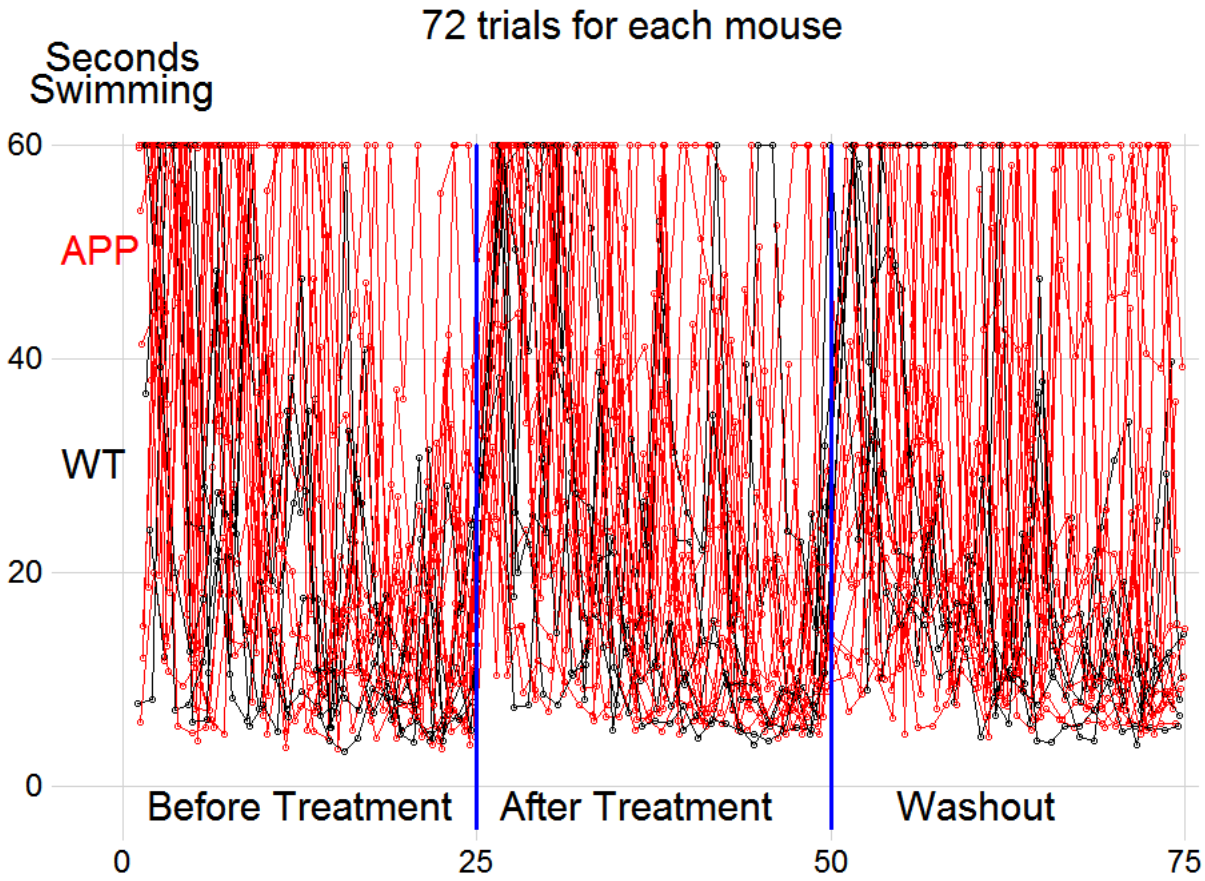
```
 mouseid type   b1   b2 b3 b4   b5   b6 b7 b8 b9 b10 b11
1       A   WT  7.8  8.2 60 60 60.0 10.7 27 24 60  20  18
2       B  APP 60.0 60.0 13 21  7.3  7.1 23 21 28  18  35
  b12 b13 b14  b15 b16 b17 b18 b19  b20  b21  b22 b23  b24
1  26  35 8.7  6.9  23  41 9.9 7.1  9.5 23.1 15.5 7.1 10.1
2  60  12 8.5 10.7  11  16 9.5 6.5 10.3  5.3  5.7 8.1  5.1
  a1 a2 a3 a4 a5 a6   a7 a8   a9  a10 a11  a12  a13  a14
1 39 58 60 60 60 40 60.0 52 37.7 27.7  20 13.1 45.9 23.1
2 60 15 15  9  7 20  9.8 16  6.5  6.5  10  6.3  5.1  8.3
   a15  a16 a17 a18  a19  a20  a21 a22 a23 a24 w1 w2 w3 w4
1 22.9 13.7  13 7.3 39.5 17.1 21.7 7.1 9.3 6.5 60 58 47 50
2  7.7  6.1  39 7.1  5.3  6.9  6.1 8.5 5.9 5.7 60 34 24 19
  w5 w6 w7 w8 w9 w10  w11  w12 w13  w14 w15 w16 w17 w18
1 46 12 25 15 15  22 17.3  9.3  12  5.9  20  12  25 9.1
2 14 19 19 20 14  15  9.5 60.0  15 31.5  20  11  24 7.3
   w19  w20  w21 w22  w23  w24
1 24.3 30.5 34.1 5.5 24.9 39.7
2  7.5  5.7  5.9 9.7  7.1  7.9
```

The 21 mice were made to swim 24 times under each of 3 experimental conditions (a, b, w).

5.1 Each mouse a curve

Draw a curve through the 72 trial values for each mouse:

```
tiff("pictures/MiceLines.tif", w=1000, h=700)

Grid(c(-5, seq(0,75,25), 76), c(-5, seq(0,60,20), 61),
ylab="72 trials for each mouse/Seconds/Swimming",
at=c(36, 0, 0), cex=2.5)

# shift trial numbers a little to stop overlap
for (mouse in 1:21){
col <- (baw$type[mouse]=="APP") + 1
x <- c(1:24, 26:49, 51:74) + mouse/21
points(x, baw[mouse, 3:74], col=col)
lines(x, baw[mouse, 3:74], col=col)
}

# trial and mouse descriptions
text(pos=4,-5, 30, "WT", cex=2.5)
text(pos=4, -5, 50, "APP", col="red", cex=2.5)
text(pos=4, 1, -2, "Before Treatment",cex=2.5)
text(pos=4, 26, -2, "After Treatment", cex=2.5)
text(pos=4, 54, -2, "Washout", cex=2.5)

# vertical lines to separate experiments
lines(c(25,25), c(-5,60), lwd=3, col="blue")
lines(c(50, 50), c(-5,60), lwd=3, col="blue")

dev.off()
```



A single curve connects the swimming time for each mouse. The WT(wild type) mice curves are black, and the APP mice curves are red. It is quite a confusing mess, but you can see that the WT mice, on the whole, have lower time to escape than the APP mice. It is not evident whether or not there is a difference in the APP mice in their performances before and after being given the drug.

6.2 Each mouse a set of points

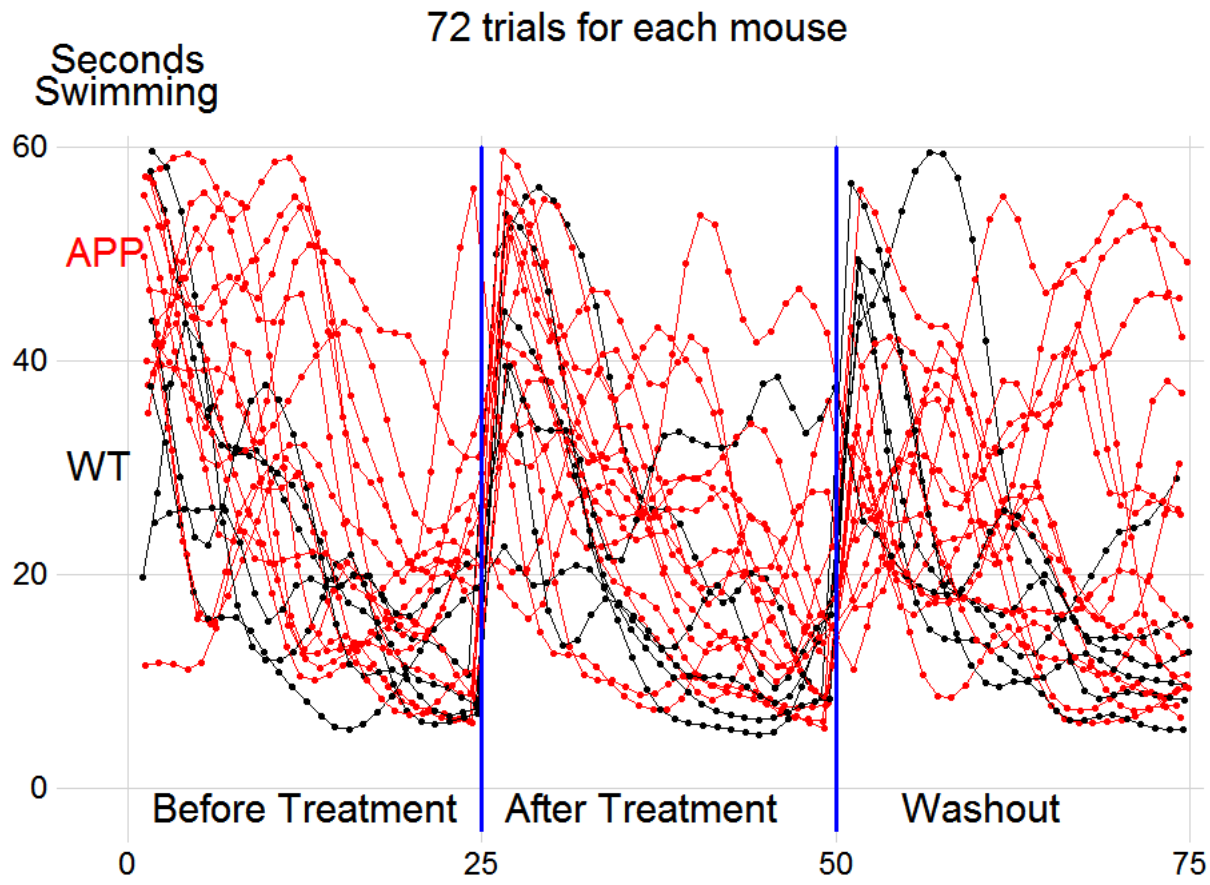
Make the same picture as before, with the curves for each mouse omitted:

```
tiff("pictures/MicePoints.tif", w=1000, h=700)

Grid(c(-5, seq(0,75,25), 76), c(-5, seq(0,60,20), 62),
ylab="72 trials for each mouse/Seconds/Swimming",
at=c(36, 0, 0), cex=2.5)

# shift trial numbers a little to stop overlap
for (mouse in 1:21){
col <- (baw$type[mouse]=="APP") + 1
x <- c(1:24, 26:49, 51:74) + mouse/21
points(x, baw[mouse, 3:74], col=col)
}

# trial and mouse descriptions
text(pos=4,-5, 30, "WT", cex=2.5)
text(pos=4, -5, 50, "APP", col="red", cex=2.5)
text(pos=4, 1, -2, "Before Treatment",cex=2.5)
text(pos=4, 26, -2, "After Treatment", cex=2.5)
text(pos=4, 54, -2, "Washout", cex=2.5)

# vertical lines to separate experiments
lines(c(25, 25), c(-5,60), lwd=3, col="blue")
lines(c(50, 50), c(-5,60), lwd=3, col="blue")

dev.off()
```

We lose some information without the lines. For example, the high WT values in the Washout experiment are all from the same mouse, which we can see in the curves picture. The points picture makes it easier to judge the distribution, to see the better learning in the WT mice clearly, though it is still dubious about differences in the APP mice before and after drug.

6.3 Smoothed mouse curves

```
tiff("pictures/MiceSmooth.tif", w=1000, h=700)

Grid(c(-5, seq(0,75,25), 76), c(-5, seq(0,60,20), 61),
ylab="72 trials for each mouse/Seconds/Swimming",
at=c(36, 0, 0), cex=2.5)

# shift trial numbers a little to stop overlap
for (mouse in 1:21){
col <- (baw$type[mouse]=="APP") + 1
x <- c(1:24, 26:49, 51:74) + mouse/21

# separately smooth 3 experiments
s1 <- Smooth(baw[mouse,  3:26], 5)
s2 <- Smooth(baw[mouse, 27:50], 5)
s3 <- Smooth(baw[mouse, 51:74], 5)
points(x, c(s1, s2, s3), col=col, pch=16)
lines(x, c(s1, s2, s3), col=col)
}

# trial and mouse descriptions
text(pos=4,-5, 30, "WT", cex=2.5)
text(pos=4, -5, 50, "APP", col="red", cex=2.5)
text(pos=4, 1, -2, "Before Treatment",cex=2.5)
text(pos=4, 26, -2, "After Treatment", cex=2.5)
text(pos=4, 54, -2, "Washout", cex=2.5)

# vertical lines to separate experiments
lines(c(25,25), c(-5,60), lwd=3, col="blue")
lines(c(50, 50), c(-5,60), lwd=3, col="blue")

dev.off()
```



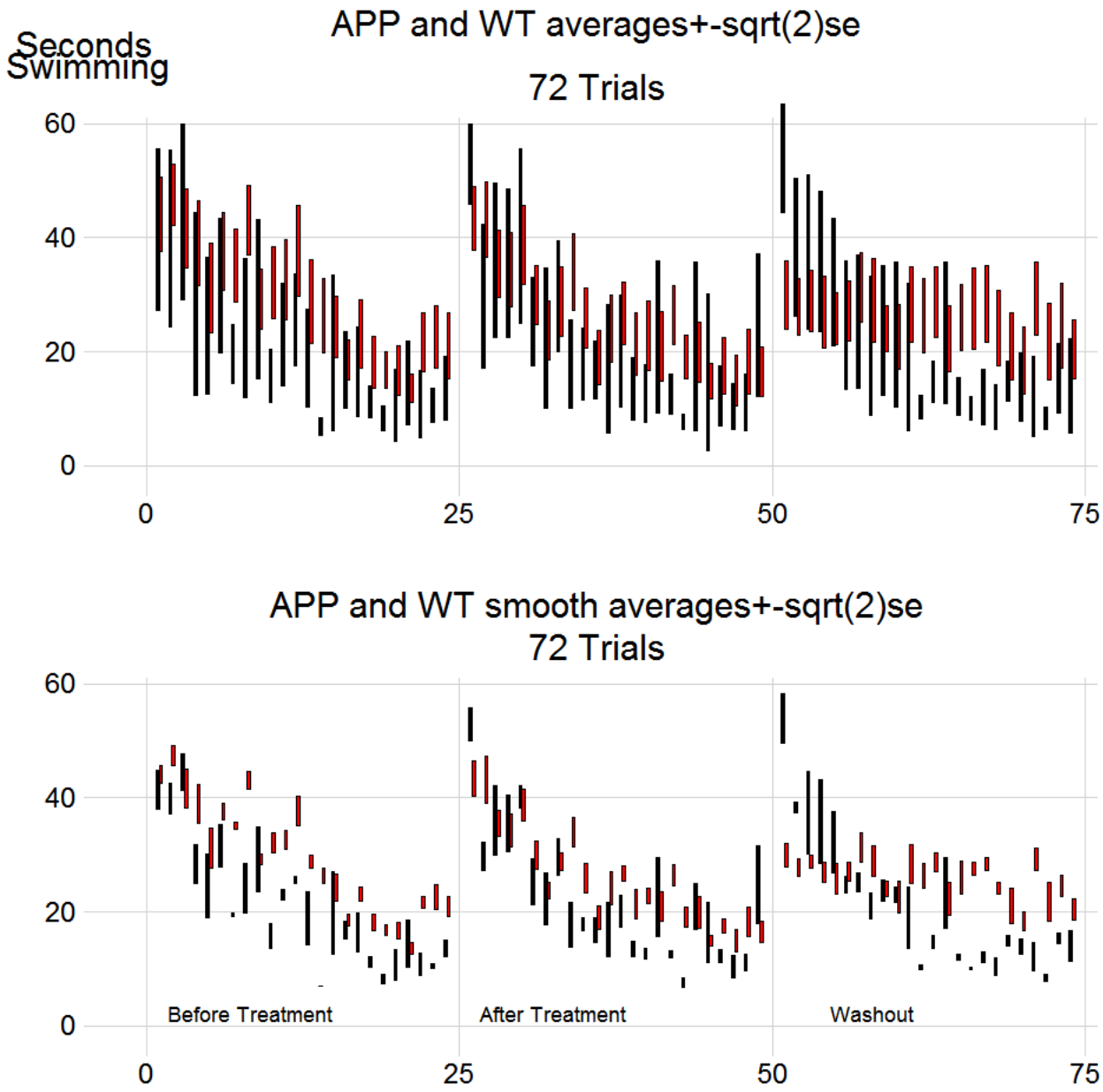
The faster learning of the WT compared to the APP is now more apparent. It seems that there may be a slight improvement for the APP in the after treatment group.

7 Means and standard errors

7.1 Raw and smoothed data

```
# mean and se plots, raw data
mwt <- rep(0, 72)
sewt <- rep(0, 72)
mapp <- rep(0, 72)
seapp <- rep(0, 72)

for ( trial in 1:72){
mwt[trial] <- mean(baw[baw$type=="WT", trial + 2])
sewt[trial]<- sd(baw[baw$type=="WT",trial+2])/sqrt(5)
mapp[trial] <- mean(baw[baw$type=="APP", trial + 2])
seapp[trial]<-
sd(baw[baw$type=="APP",trial+2])/sqrt(20)
}

x <- c(1:24, 26:49, 51:74)

# Compare raw and smoothed means
tiff("pictures/Means.tif", w=1000, h=950)
par(mfrow=c(2,1))
par(mar=c(5,5,4,2))

# set up background grid
Grid(c(-5, seq(0,75,25), 76), c(-5, seq(0,60,20), 61),
ylab="APP and WT averages+-sqrt(2)se/Seconds/
Swimming/72 Trials",  at=c(36, -5, -5, 36),cex=2.5)

rect(x-0.25, mwt-sqrt(2)*sewt, x, mwt+sqrt(2)*sewt,
col="black")
rect(x, mapp-sqrt(2)*seapp, x+0.25,mapp+sqrt(2)*seapp,
col="red")

# mean and se plots, smoothed data
mwt <- rep(0, 72)
sewt <- rep(0, 72)
mapp <- rep(0, 72)
```

```
seapp <- rep(0, 72)

for ( trial in 1:72){
mwt[trial]<-
mean(Smooth(baw[baw$type=="WT",trial+ 2],5))
sewt[trial] <-
sd(Smooth(baw[baw$type=="WT", trial+2],5))/sqrt(5)
mapp[trial]<-
mean(Smooth(baw[baw$type=="APP", trial + 2],5))
seapp[trial] <-
sd(Smooth(baw[baw$type=="APP", trial+2],5))/sqrt(20)
}

# Set up background grid
Grid(c(-5, seq(0,75,25), 76), c(-5, seq(0,60,20), 61),
ylab="APP and WT smooth averages+-sqrt(2)se/
72 Trials",  at=c(36, 36),cex=2.5)
rect(x-0.25, mwt-sqrt(2)*sewt, x, mwt+sqrt(2)*sewt,
col="black")
rect(x,mapp-sqrt(2)*seapp, x+0.25, mapp+sqrt(2)*seapp,
col="red")

# identify treatment regimes
text(pos=4, 1, 2, "Before Treatment",cex=1.5)
text(pos=4, 26, 2, "After Treatment", cex=1.5)
text(pos=4, 54, 2, "Washout", cex=1.5)

dev.off()
```



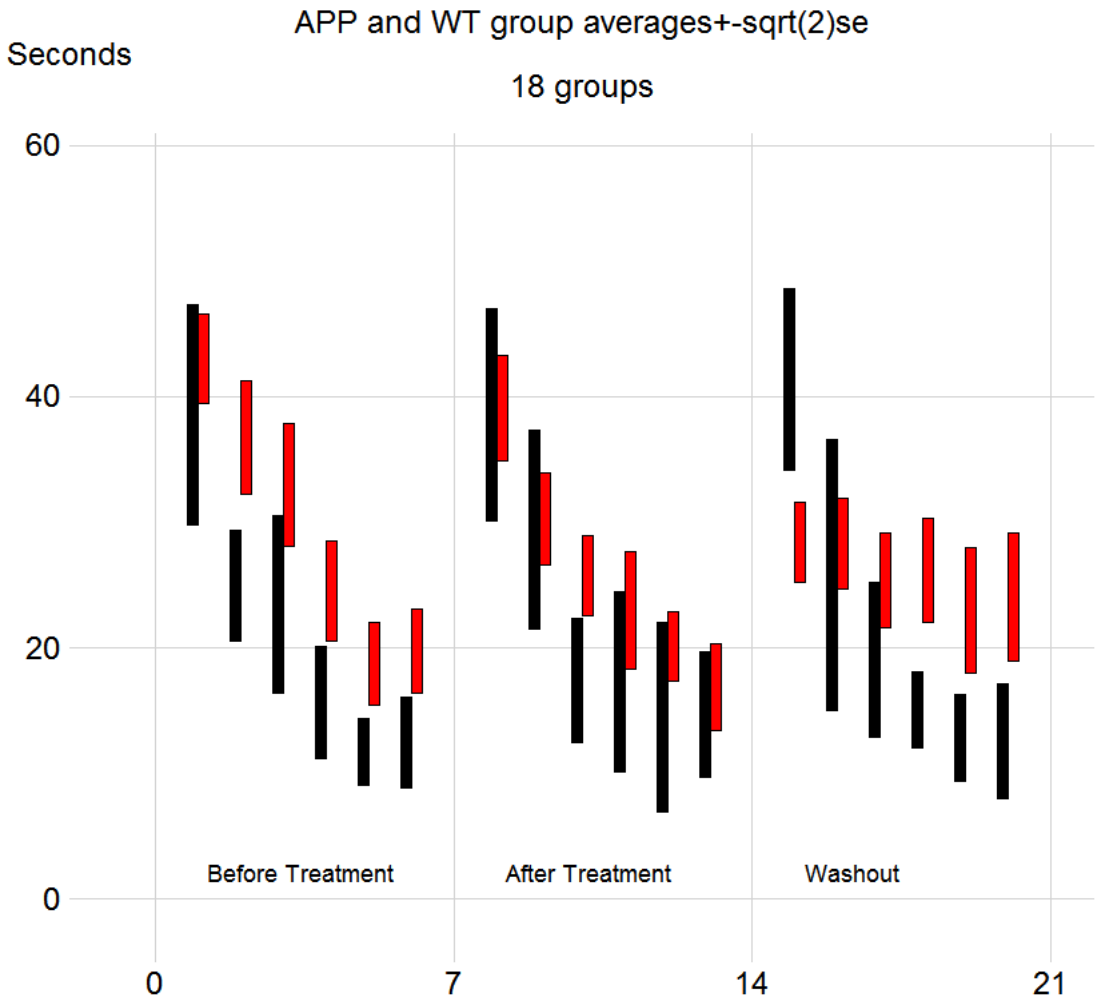
The Wt and APP values for each trial are significantly different at the 95% level when the red and black blocks do not overlap. The smoothed data tend show more significant differences because variability at each trial time is reduced.

We see distinct declines in search times within each treatment in both WT and APP groups, with the WT groups consistently lower. The question is whether or not the treatment group, the middle group, has relatively lower search times towards the end of the experiment, showing that the drug improves memory. Maybe, maybe not.

7.2 Grouped data

We compute mean and se plots for the original groups of sets of 4 trials. This is a better way of smoothing because we are smoothing over natural groups.

```
tiff("pictures/mean and se grouped.tif", w=900, h=800)
group <- baw[, 1:20]
for ( mouse in 1:21){
for( g in 1:18)
group[mouse,g+2] <-
mean(unlist(baw[mouse,(3:6)+4*(g-1)]))
}
# Compute means and standard errors in each group
mwt <- rep(0, 18)
sewt <- rep(0, 18)
mapp <- rep(0, 18)
seapp <- rep(0, 18)
for (g in 1:18){
mwt[g] <- mean(group[group$type=="WT", g + 2])
sewt[g]<-sd(group[group$type=="WT", g + 2])/sqrt(5)
mapp[g] <- mean(group[group$type=="APP", g + 2])
seapp[g] <-
sd(group[group$type=="APP", g + 2])/sqrt(20)
}
x <- c(1:6, 8:13, 15:20)
Grid(c(-2, seq(0,21,7), 22), c(-5, seq(0,60,20), 61),
ylab="APP and WT group averages+-sqrt(2)se/Seconds/
18 groups", at=c(10,-2,10), cex=2)

rect(x-0.25, mwt-sqrt(2)*sewt, x, mwt+sqrt(2)*sewt,
col="black")
rect(x, mapp-sqrt(2)*seapp, x+0.25,mapp+sqrt(2)*seapp,
col="red")

# identify treatment regimes
text(pos=4, 1, 2, "Before Treatment",cex=1.5)
text(pos=4, 8, 2, "After Treatment", cex=1.5)
text(pos=4, 15, 2, "Washout", cex=1.5)
dev.off()
```

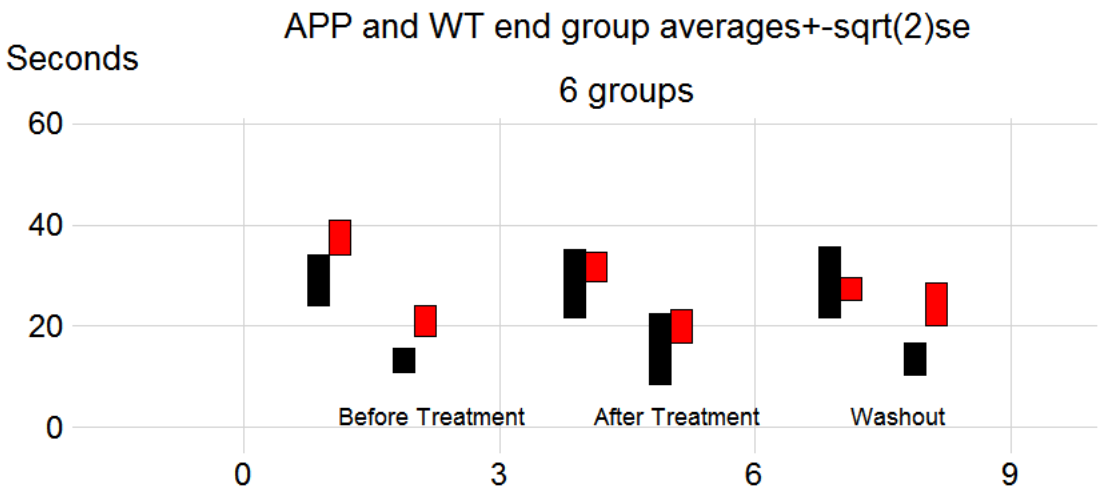


We have seen this before; a more rapid decline of learning times for wild type compared to mutated; suggested but insignificant differences in the middle experiment where both groups are treated.

6.3 Comparing end groups

Now do a similar calculation comparing the beginning and end groups of 12 trials in each experiment, 6 comparisons in all:

```
roup <- baw[, 1:8]
for ( mouse in 1:21){
for( g in 1:6)
group[mouse,g+2]<-
mean(unlist(baw[mouse,(3:14)+12*(g-1)]))
}

# means and se for groups
mwt <- rep(0, 6)
sewt <- rep(0, 6)
mapp <- rep(0, 6)
seapp <- rep(0, 6)
for ( g in 1:6){
mwt[g] <- mean(group[group$type=="WT", g + 2])
sewt[g]<-sd(group[group$type=="WT", g + 2])/sqrt(4)
mapp[g] <- mean(group[group$type=="APP", g + 2])
seapp[g] <-
sd(group[group$type=="APP", g + 2])/sqrt(20)
}
x <- c(1, 2, 4, 5, 7, 8)
tiff("pictures/group means.tif", w=900, h=400)
Grid(c(-2, seq(0,9,3), 10), c(-5, seq(0,60,20), 61),
ylab="APP and WT end group averages+-sqrt(2)se/
Seconds/6 groups",
at=c(4.5,-2, 4.5),cex=2.2)
rect(x - 0.25, mwt - sqrt(2) * sewt, x, mwt + sqrt(2)
* sewt, col="black")
rect(x, mapp - sqrt(2) * seapp, x + 0.25, mapp +
sqrt(2) * seapp, col="red")

text(pos=4, 1, 2, "Before Treatment",cex=1.5)
text(pos=4, 4, 2, "After Treatment", cex=1.5)
text(pos=4, 7, 2, "Washout", cex=1.5)
dev.off()
```



Again we see that the treated middle group shows no significant differences in swimming times between mutant and wild type, whereas the two untreated groups show such a difference. It suggests that the treatment is improving the performance of the mutants.

7 The end-trials learning statistic

The end learning statistic is the difference for each mouse between the average of the last 12 trials under treatment and the average of the last 12 trials before treatment and during washout. Our hypothesis is that the treated mice will do relatively better for the APP mice than for the wild type.:

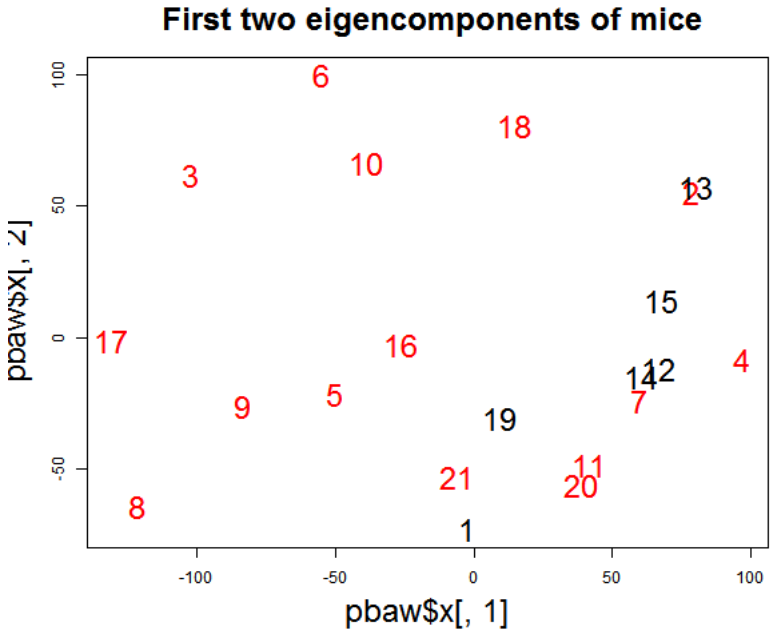
```
learn <- rep(0,21)
for (mouse in 1:21)
learn[mouse] <- mean(unlist(baw[mouse, c(13:24,
61:72)])) - mean(unlist(baw[mouse, c(37:48)]))

t.test(learn ~baw$type)
```

```
     Welch Two Sample t-test
data:  learn by baw$type
t = 1, df = 15, p-value = 0.3121
alternative hypothesis: true difference in means is not equal to
0
95 percent confidence interval:
 -4.6 13.5
sample estimates:
mean in group APP  mean in group WT
             3.54              -0.91
```

8 Eigenvector analysis to identify different trajectories

8.1 Display projected curves in first two eigenvectors

```
tiff("pictures/First two eigenvectors for mice.tif",
w=600, h=500)
pbaw <- prcomp(baw[,-(1:2)])

plot(pbaw$x[,1], pbaw$x[,2], pch="",cex.lab=2,
main=" First two eigencomponents of mice", cex.main=2)

text(pbaw$x[,1], pbaw$x[,2],
    1:21,col=1+(baw$type=="APP"), cex=2)

dev.off()
```

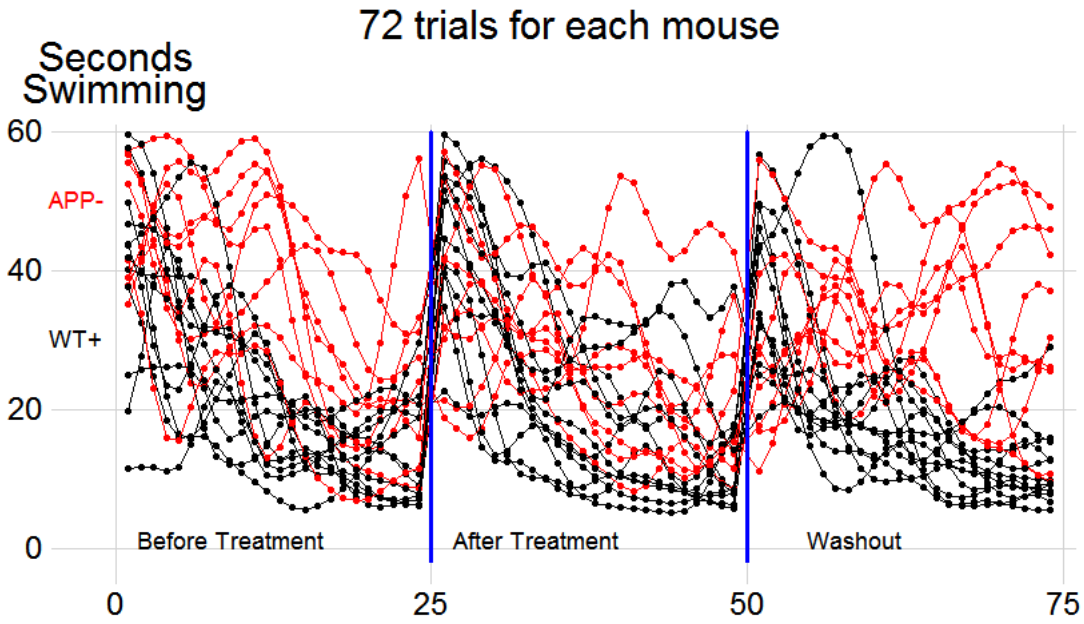


So, six of the mutated APP(red) mice are similar to wild type WT(black) mice in their memory behaviour, and the others are different.

8.2 Eigenvector groups displayed in smooth curves

Expand the wild type group to include also the APP mice that appear to behave similarly to the original WT:

```
black<- c(1,21,20,11,19,7,12,14,4,2,13,15)
tiff("pictures/Eigenvectors.tif", w=900, h=500)

Grid(c(-5, seq(0,75,25), 76), c(-5, seq(0,60,20), 61),
ylab="72 trials for each mouse/Seconds/Swimming",
at=c(36, 0, 0), cex=2.5)

# do all the mice, shift trial numbers a little to
stop overlap
for ( mouse in 1:21){
 col <- -(mouse %in% black) + 2
 x <- c(1:24, 26:49, 51:74)

# separately smooth 3 experiments
 s1 <- Smooth(baw[mouse,  3:26], 5)
 s2 <- Smooth(baw[mouse, 27:50], 5)
 s3 <- Smooth(baw[mouse, 51:74], 5)
 points(x, c(s1, s2, s3), col=col, pch=16)
 lines(x, c(s1, s2, s3), col=col)
}

# trial and mouse descriptions
text(pos=4,-6, 30, "WT+", cex=1.5)
text(pos=4, -6, 50, "APP-", col="red", cex=1.5)
text(pos=4, 1, 1, "Before Treatment",cex=1.5)
text(pos=4, 26, 1, "After Treatment", cex=1.5)
text(pos=4, 54, 1, "Washout", cex=1.5)

# vertical lines to separate experiments
lines(c(25,25), c(-5,60), lwd=3, col="blue")
lines(c(50, 50), c(-5,60), lwd=3, col="blue")

dev.off()
```



There is still clear learning speed advantages for the expanded wild type group compared to the residual APP group. It is even suggested that the residual APP group does relatively better after treatment, with the the two groups closer in their behaviour there. We can't claim too much after moving the group memberships around; if we are given free reign to change the basic classification, we can get pretty well any result we like.

8 Conclusions

The APP mice clearly learn more slowly than the WT mice in all circumstances. There is a slight improved learning in the APP mice after treatment (estimated to be 3 seconds faster than before treatment), but it is not statistically significant. Six of the APP mice behave similarly to the WT mice in their learning behavior; others are quite different. It may be that the six anomalous APP mice are not mentally effected by the mutation.

9 Data Preparation

Shape data to get 72 observations per mouse:

```
list.files("data/")
```

```
[1] "MiceMemory.csv"
[2] "Total Latency MTEP.csv"
[3] "Total Latency Untreated.csv"
[4] "Total Latency Washout.csv"
```

```
u <- read.csv("data/Total Latency Untreated.csv",
header=T, as.is=T)
head(u)
```

```
 Sbj.Code Type Lat..Target Lat..Target.1 Lat..Target.2
1        A   WT         7.8          60.0            60
2        A   WT         8.2          10.7            20
3        A   WT        60.0          26.7            18
4        A   WT        60.0          23.7            26
5        B  APP        60.0           7.3            28
6        B  APP        60.0           7.1            18
  Lat..Target.3 Lat..Target.4 Lat..Target.5
1          35.1          40.9          23.1
2           8.7           9.9          15.5
3           6.9           7.1           7.1
4          23.1           9.5          10.1
5          11.5          15.9           5.3
6           8.5           9.5           5.7
```

We do the same extraction for the three data files with swimming times before treatment, after treatment, and after a washout period. Altogether we need a data frame with the first two variables the mouse name and type, and the last 72 variables the swimming times under the different treatment conditions:

```
baw <- data.frame(matrix(0, nrow=21, ncol=74))
baw[,1:2] <- u[seq(1,84,4),1:2]
```

Specify names of variables, including three sets of 24 swim times:

```
names(baw) <- c("mouseid", "type",
                paste("b", 1:24, sep=""),
                paste("a", 1:24, sep=""),
                 paste("w", 1:24, sep=""))
```

Construct a little extract function to avoid repeating code: this function reads each file with mouse swimming times in cols, returning a matrix with one row per mouse.The columns containing swimming times are located in different places in the different data sets. The data for each mouse appears in a 4 by 6 block in the csv file, with the first trial the first column of 4.

```
Extract<- function(file, cols){
u <- read.csv(file, header=T, as.is=T)
ud <- matrix(0, nrow=21, ncol=24)
for( row in 1:21)
ud[row, ] <- unlist(u[(1:4) + 4*(row-1), cols])
return(ud)
}
```

9.1 Read in and fix "Untreated" data:

```
b <- Extract("data/Total Latency Untreated.csv",3:8)
baw[, 3:26] <- b
head(b, 2)
```

```
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  7.8  8.2   60   60 60.0 10.7   27   24   60    20
[2,] 60.0 60.0   13   21  7.3  7.1   23   21   28    18
     [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19]
[1,]    18    26    35   8.7   6.9    23    41   9.9   7.1
[2,]    35    60    12   8.5  10.7    11    16   9.5   6.5
     [,20] [,21] [,22] [,23] [,24]
[1,]   9.5  23.1  15.5   7.1  10.1
[2,]  10.3   5.3   5.7   8.1   5.1
```

Note that the first four values in the first row correspond to the first four values in the third column of the u array.

9.2 Read in and fix "after treatment" data

```
a<- Extract("data/Total Latency MTEP.csv", 3:8)
baw[, 27:50] <- a
head(a, 2)
```

```
    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]   39   58   60   60   60   40 60.0   52 37.7  27.7
[2,]   60   15   15    9    7   20  9.8   16  6.5   6.5
     [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19]
[1,]    20  13.1  45.9  23.1  22.9  13.7    13   7.3  39.5
[2,]    10   6.3   5.1   8.3   7.7   6.1    39   7.1   5.3
     [,20] [,21] [,22] [,23] [,24]
[1,]  17.1  21.7   7.1   9.3   6.5
[2,]   6.9   6.1   8.5   5.9   5.7
```

9.3 Read in and fix "washout" data

```
w<- Extract("data/Total Latency Washout.csv ",  2:7)
baw[, 51:74] <- w
head(w, 2)
```

```
    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]   60   58   47   50   46   12   25   15   15    22
[2,]   60   34   24   19   14   19   19   20   14    15
     [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19]
[1,]  17.3   9.3    12   5.9    20    12    25   9.1  24.3
[2,]   9.5  60.0    15  31.5    20    11    24   7.3   7.5
     [,20] [,21] [,22] [,23] [,24]
[1,]  30.5  34.1   5.5  24.9  39.7
[2,]   5.7   5.9   9.7   7.1   7.9
```

8.4 All data combined

Each line gives a mouse name, its type, and the 72 swim times in 3 series of 24 under the the 3 treatments, "before", "after" and "washout":

```
head(baw, 2)
```

```
 mouseid type   b1   b2 b3 b4   b5   b6 b7 b8 b9 b10 b11
1       A   WT  7.8  8.2 60 60 60.0 10.7 27 24 60  20  18
2       B  APP 60.0 60.0 13 21  7.3  7.1 23 21 28  18  35
  b12 b13 b14  b15 b16 b17 b18 b19  b20  b21  b22 b23  b24
1  26  35 8.7  6.9  23  41 9.9 7.1  9.5 23.1 15.5 7.1 10.1
2  60  12 8.5 10.7  11  16 9.5 6.5 10.3  5.3  5.7 8.1  5.1
  a1 a2 a3 a4 a5 a6   a7 a8   a9  a10 a11  a12  a13  a14
1 39 58 60 60 60 40 60.0 52 37.7 27.7  20 13.1 45.9 23.1
2 60 15 15  9  7 20  9.8 16  6.5  6.5  10  6.3  5.1  8.3
   a15  a16 a17 a18  a19  a20  a21 a22 a23 a24 w1 w2 w3 w4
1 22.9 13.7  13 7.3 39.5 17.1 21.7 7.1 9.3 6.5 60 58 47 50
2  7.7  6.1  39 7.1  5.3  6.9  6.1 8.5 5.9 5.7 60 34 24 19
  w5 w6 w7 w8 w9 w10  w11  w12 w13  w14 w15 w16 w17 w18
1 46 12 25 15 15  22 17.3  9.3  12  5.9  20  12  25 9.1
2 14 19 19 20 14  15  9.5 60.0  15 31.5  20  11  24 7.3
   w19  w20  w21 w22  w23  w24
1 24.3 30.5 34.1 5.5 24.9 39.7
2  7.5  5.7  5.9 9.7  7.1  7.9
```

Save the reframed data:

```
write.csv(baw, "data/MiceMemory.csv", row.names=F)
```

7 Functions

```
Grid <- function(xticks, yticks, ylab="",
at=(min(xticks)+ mean(xticks))/2, cex=2.5){

# background for plot using grid of light grey lines
par(mar=c(3,3,6,2))

plot(1, 1,  xlim=range(xticks),  ylim = range(yticks),
    xlab="", ylab="", axes=F, pch="")

# use only interior values of tick ranges in plots
usey <- rep( T, length(yticks) )
usey[c( 1, length(yticks) )] <- F
usex <- rep( T, length(xticks) )
usex[c( 1, length(xticks) )] <- F

# grey lines in both directions
for ( row in yticks[usey] )
 lines(range(xticks), c(row, row), col="light grey")
for ( col in xticks[usex] )
 lines(c(col, col), range(yticks), col="light grey")

# put ylab on left top, using / to split long
expressions
ylabs <- unlist(strsplit(ylab,"/"))

# identify tick marks on both axes
if (length(yticks) > 2)
 text(pos=2, rep(min(xticks), length(yticks)-2 ),
      yticks[usey], yticks[usey], cex=2, xpd=T)
if (length(xticks)>2)
 text(pos=1, xticks[usex],  rep(min(yticks),
            length(xticks)-2), xticks[usex], cex=2,
xpd=T)

# insert top labels
lylabs <- min(5, length(ylabs))
if(lylabs > 0)
 mtext(ylabs, side=3,line = (5/lylabs)*(lylabs-1):0,
```

```
       at = at, cex=cex)
par(mar=c(5, 4, 4, 2))
invisible()
}

Smooth <- function(x,w=5){
# smooth x at width w
# unlist to accept rows of data frame
x <- unlist(x)
lx <- length(x)
for (iter in 1:w){
 y <- c(x[1], x, x[lx])
 x <- 0.25*y[1:lx]+0.5*y[2:(lx+1)]+0.25*y[3:(lx+2)]
}
return(x)
}
```