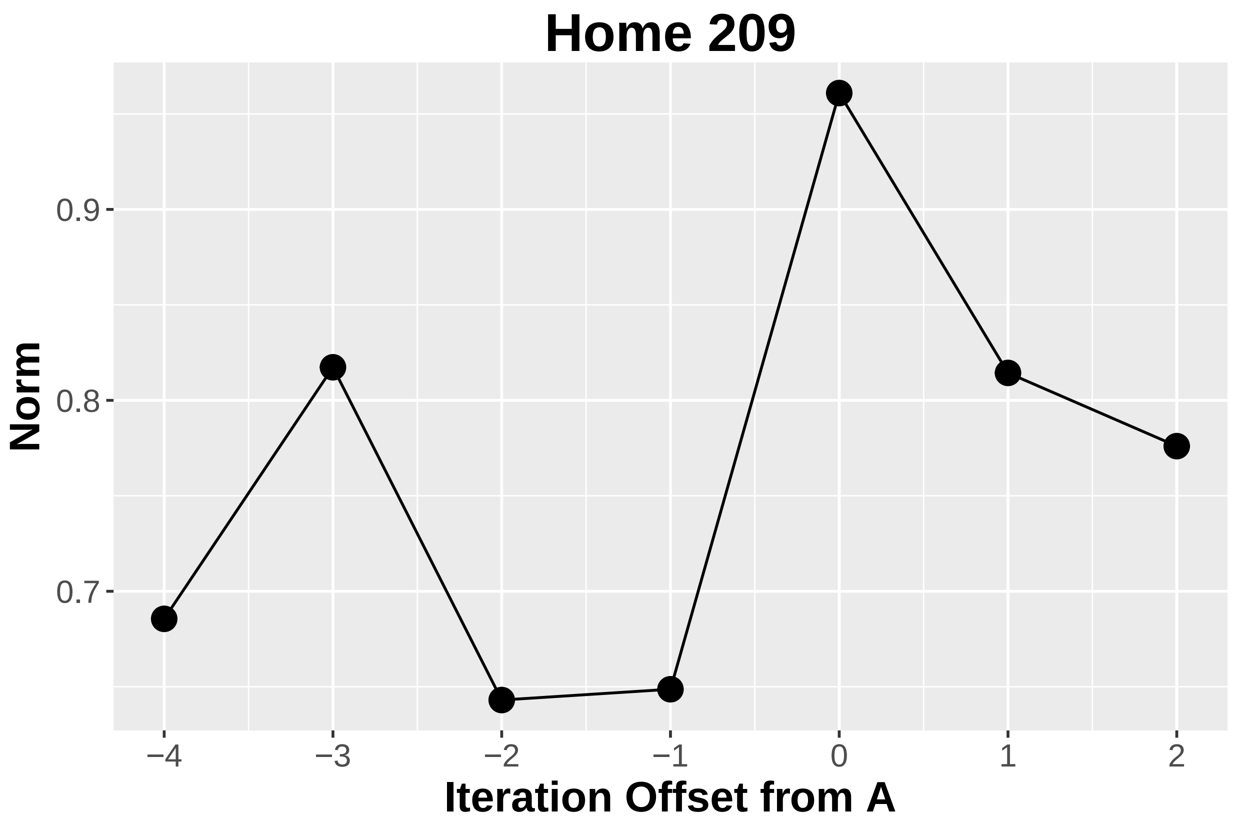
HW4 GGPLOT2 Recreation

Ryan Richardson

18/10/2020

Question 1

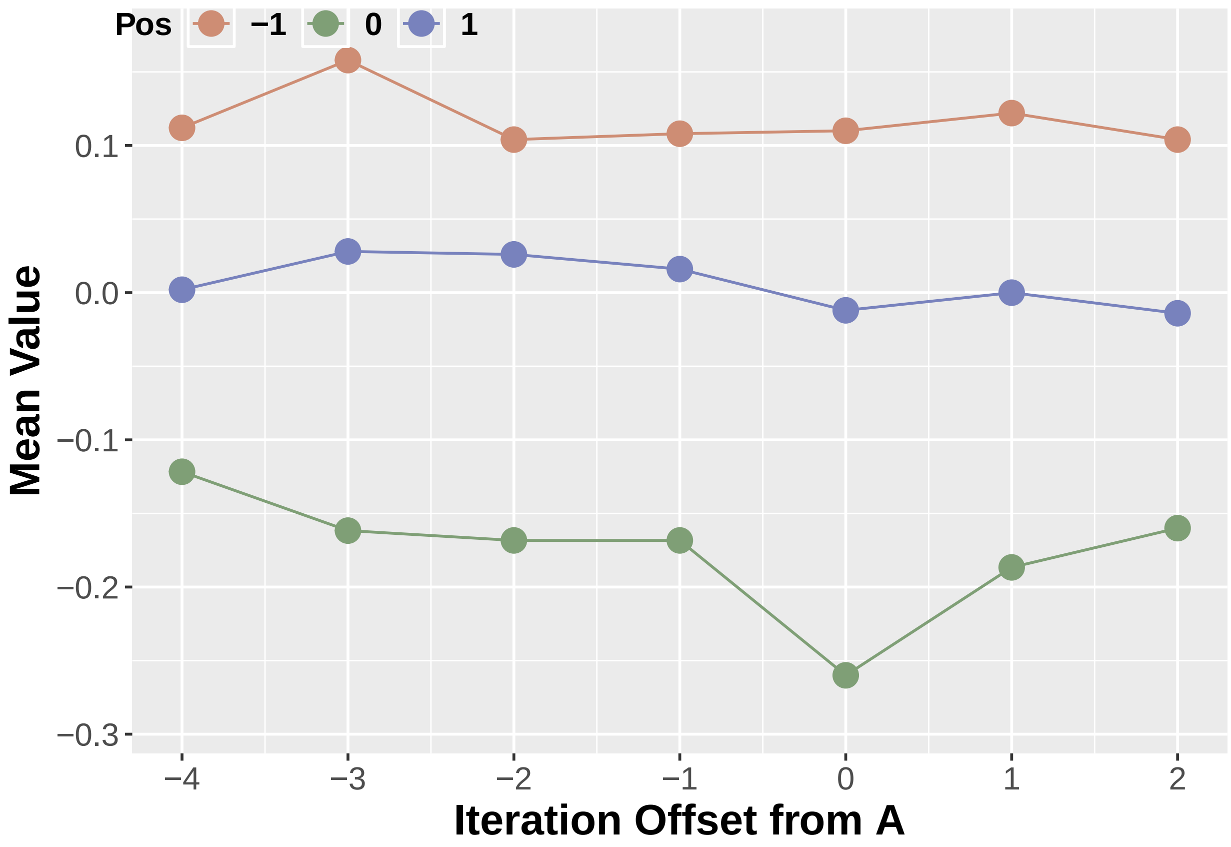
```r
homeX = c(-4,-3,-2,-1,0,1,2)

axisFormatting = element_text(size = 16, face = "bold")
titleFormatting = element_text(size = 20, face = "bold", hjust = 0.5, margin = margin(10,0,0,0))
tickFormatting = element_text(size = 12)

homeDf = data.frame(homeX, vv$norm)

homePlot = ggplot(aes(y = vv.norm, x = homeX), data = homeDf) +
  geom_point( size = 4) +
  geom_line() +
  scale_x_continuous(breaks = seq(min(homeDf$homeX), max(homeDf$homeX), by = 1))+
  ggtitle("Home 209") +
  labs(y = "Norm", x = "Iteration Offset from A") +
  theme(plot.title = titleFormatting,
        axis.title = axisFormatting,
        axis.text = tickFormatting)
homePlot
```



Question 2

```r
diagDf = data.frame(vv$diagconfig)
names(diagDf) = c(-4:2)
diagDf$position = c(1,0,-1)
diagDfMelt = melt(diagDf, id = "position")
diagDfMelt$variable = as.numeric(diagDfMelt$variable) - 5
diagDfMelt$position = as.factor(diagDfMelt$position)
```

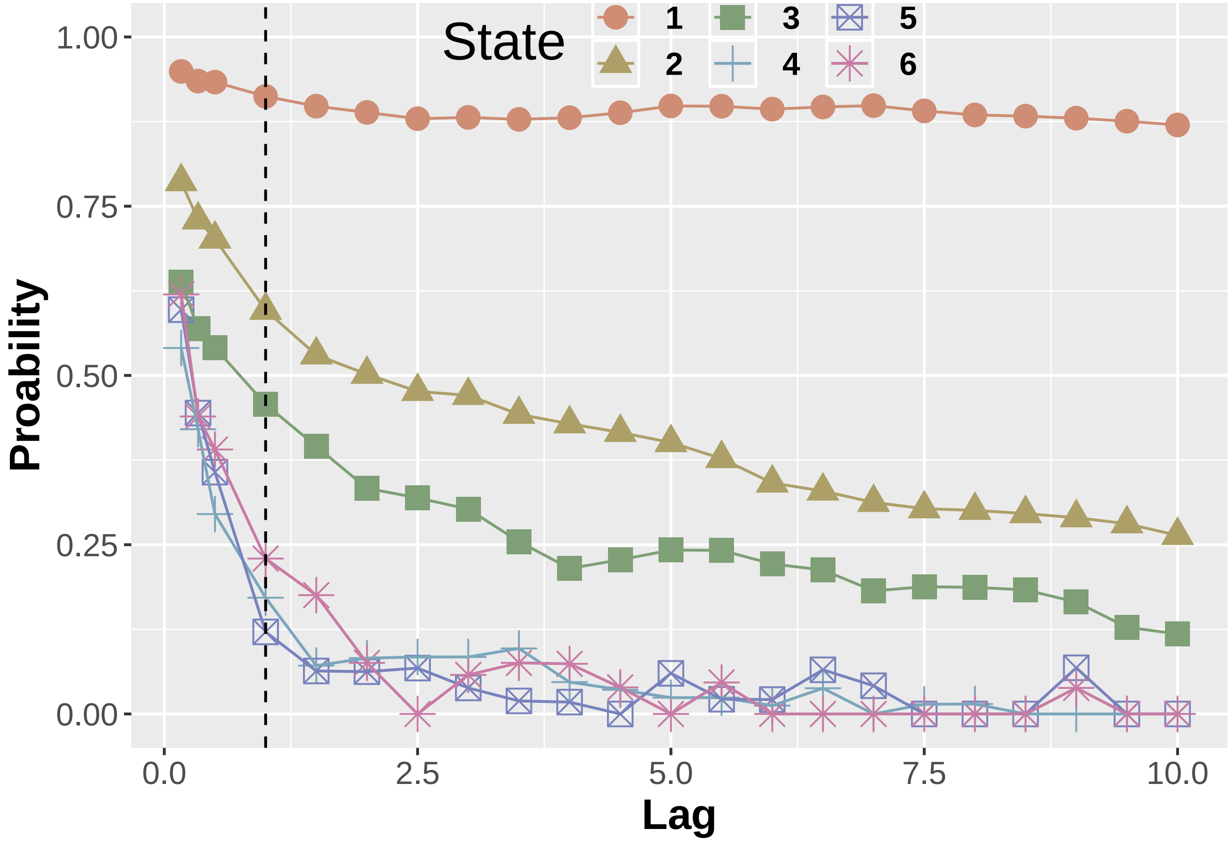
```r
offsetPlot = ggplot(aes(x = variable, y = value, group = position), data = diagDfMelt) +
  geom_point(aes(color = position), size = 4) +
  geom_line(aes(color = position)) +
  scale_colour_manual(values = c("#CE8D74", "#7F9F76", "#7882BD")) +
  scale_x_continuous(breaks = seq(min(diagDfMelt$variable), max(diagDfMelt$variable), by = 1)) +
  ylim(-0.29,0.17) +
  labs(y = "Mean Value", x = "Iteration Offset from A", color = "Pos") +
  theme(axis.title = axisFormatting,
        axis.text = tickFormatting,
        legend.title = element_text(face = "bold", size = 12),
        legend.text = element_text(face = "bold", size = 12),
        legend.direction = "horizontal",
        legend.position = c(0.15, 0.98),
        legend.key = element_rect(color = "white", fill = "transparent"),
        legend.background = element_rect(color = "transparent", fill = "transparent", inherit.blank = T
offsetPlot
```



Question 3

```
le = lageval(tseries = TX,statebounds = sb, lagrange = c(1,2,seq(3,60,3)))
diagLag = data.frame(le$diagbylag)
diagLag$range = c(le$lagrange)
diagLag$range = diagLag$range * 1/6 # multiply by 10s then divide by 60 to convert to minutes
diagLagMelt = melt(diagLag, id = "range")
mapvalues(diagLagMelt$variable, from = (levels(diagLagMelt$variable)), to = c("1","2","3","4","5","6"))
levels(diagLagMelt$variable) <- c("1","2","3","4","5","6")
```

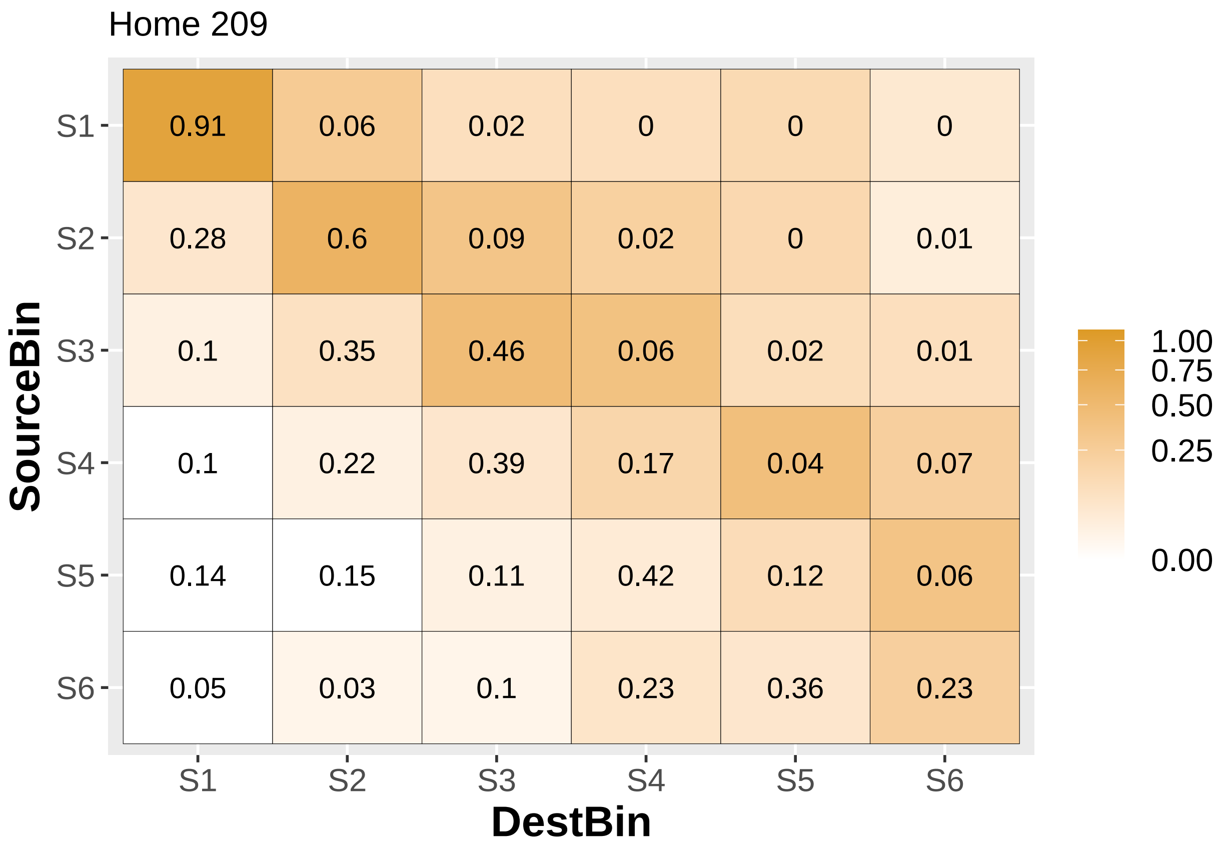
```
lagPlot = ggplot(aes(x = range, y = value, group = variable), data = diagLagMelt) +
  geom_point(aes(color = variable, shape = variable), size = 4) +
  geom_line(aes(color = variable)) +
  geom_vline(xintercept = c(1), linetype = "dashed") +
  labs(y = "Proability", x = "Lag", color = "State", shape = "State") +
  ylim(0.00, 1.00) +
  scale_colour_manual(values = c("#CE8D74", "#ACA068", "#7F9F76","#7BA6BB", "#7882BD", "#C87BA4")) +
  theme(axis.title = axisFormatting,
        axis.text = tickFormatting,
        legend.title = element_text(size = 20),
        legend.text = element_text(face = "bold", size = 12),
        legend.direction = "horizontal",
        legend.position = c(0.5, 0.95),
        legend.key = element_rect(color = "white", fill = "transparent"),
        legend.background = element_rect(color = "transparent", fill = "transparent", inherit.blank = T
lagPlot
```



Question 4

```
B = transmat(tseries = TX,statebounds = sb,lag = 6)
binDf = data.frame(B$prob)
binDf$target = c("S1","S2","S3","S4","S5","S6")
binMelt = melt(binDf, id = "target")
binMelt$value = round(binMelt$value, digits = 2)
binMelt$target = as.factor(binMelt$target)
binMelt$square = binMelt$value ** 0.5
```

```
binPlot = ggplot(aes(target, ordered(variable, levels = rev(binDf$target))), data = binMelt) +
  geom_tile(aes(fill = value), colour = "black") +
  geom_text(aes(x = variable, y = target, label = value)) +
  scale_fill_gradient(low="white", high="#DD9A25", limits= c(0.0, 1.1), trans = "sqrt") +
  # scale_fill_gradient2(low = "white", mid = "#EBCB86",  high = "#DD9A25", aesthetics = "fill", guide
  # scale_fill_gradientn(breaks = c(0.0,0.25,0.5,0.75,1.0),
  #                      colours = c("white", "#EBCB86", "#E6B65B", "#E1A439", "#DD9A25"),
  #                      limits = c(0.0,1.05), aesthetics = "fill") +
  ggtitle("Home 209") +
  labs(y = "SourceBin", x = "DestBin", fill = "") +
  theme(axis.title = axisFormatting,
        axis.text = tickFormatting,
        legend.title = element_text(size = 20),
        legend.text = element_text(size = 12),
        legend.key = element_rect(color = "white", fill = "transparent"),
        legend.background = element_rect(color = "transparent", fill = "transparent", inherit.blank = T

binPlot
```



Question 4 Opine

We use the square root of the probability value for our fill in order to provide a visual distinction between levels. Without that, the levels are too close together and are mostly white and indistinguishable. Unfortunately,I've been unable to get my gradient to behave properly with 0.0 values, which creates a similar problem with some values blending, and not being easily distinguishable. This issue existed across each scale_fill_gradient function offfered by ggplot2.