

Data_analysis

2022-08-04

```
#read in data
dat <- read.csv("data_108.csv")

library(dplyr)
```

```
##
## Attaching package: 'dplyr'

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

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(ggExtra)
```

NOTES

#This is how to compute a new variable counting occurrences of value "1" in V1-V4.
df\$count.1 <- apply(df, 1, function(x) length(which(x==1)))

#You can do the same to count how many time the value "2" occurs per row in V1-V4. Note that you need to select the columns (variables) in df to which the function is applied. df\$count.2 <- apply(df[1:4], 1, function(x) length(which(x==2)))

CALCULATE SUM LOOKS AT TARGET

#count number of occurrences of numbers 1 to 8 across all subjects. Saved as vector.

```
dat$dat_count.1 <- apply(dat, 1, function(x) length(which(x==1)))

dat$dat_count.2 <- apply(dat[2:32], 1, function(x) length(which(x==2)))
dat$dat_count.3 <- apply(dat[2:32], 1, function(x) length(which(x==3)))
dat$dat_count.4 <- apply(dat[2:32], 1, function(x) length(which(x==4)))
dat$dat_count.5 <- apply(dat[2:32], 1, function(x) length(which(x==5)))
dat$dat_count.6 <- apply(dat[2:32], 1, function(x) length(which(x==6)))
dat$dat_count.7 <- apply(dat[2:32], 1, function(x) length(which(x==7)))
dat$dat_count.8 <- apply(dat[2:32], 1, function(x) length(which(x==8)))
```

#count looks at target across all targets 1-8

```
dat$sum <- rowSums(dat[,c("dat_count.1", "dat_count.2", "dat_count.3", "dat_count.4", "dat_count.5", "dat_count.6", "dat_count.7", "dat_count.8")])
```

CALCULATE SUM OF LOOKS AT SCREEN

```
#count number of zeros
dat$count_zero <- apply(dat[2:32], 1, function(x) length(which(x==0)))

#count number of possible looks to target (aka total everything but zero)
dat <- mutate(dat, looks_at_screen=31-count_zero)
```

PERCENT LOOKS AT TARGET

```
dat <- mutate(dat, percent_looks=(sum/looks_at_screen)*100)
```

TAKE OUT COMULMNS WE WANT

```
#take out sum and subj.time columns
looks_over_time <- data.frame(dat$subj.time, dat$percent_looks)
```

GRAPHS

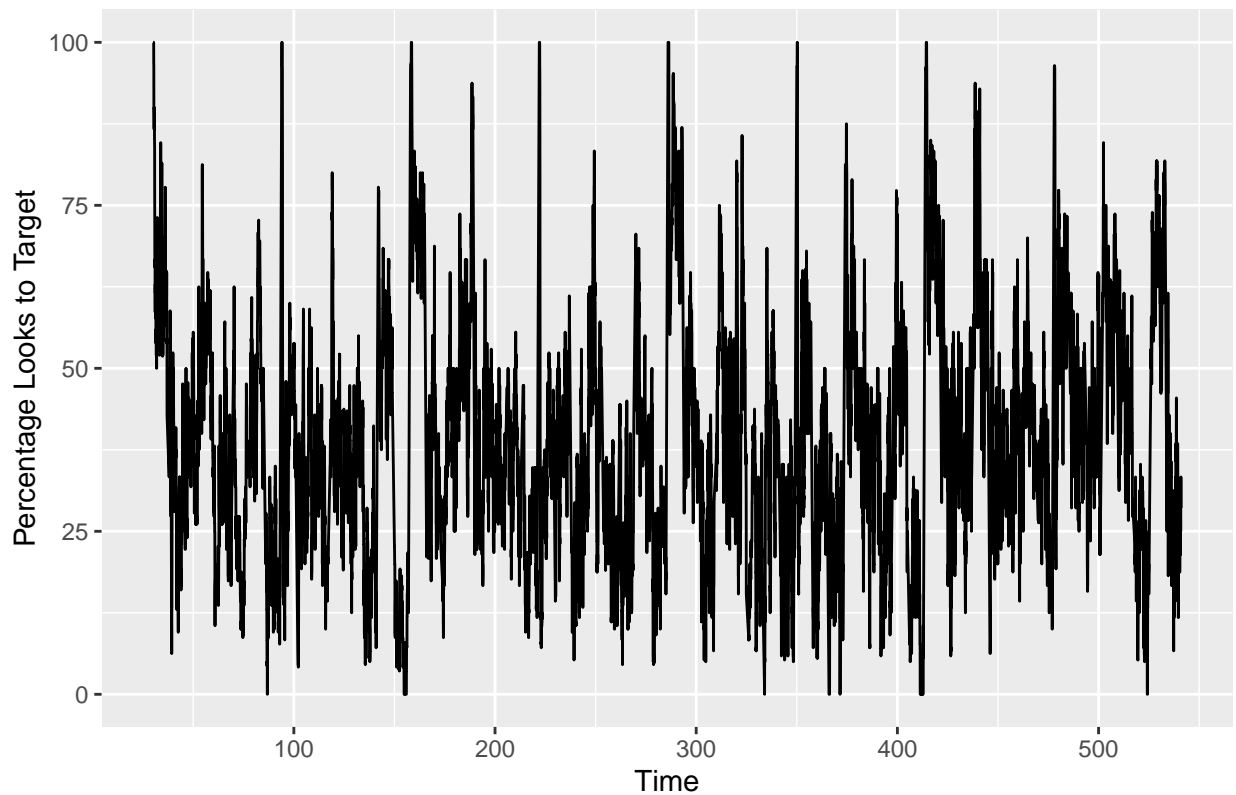
```
#graph for all children over whole experiment
#ONE TEMPORAL PROFILE

ggplot (looks_over_time, aes ( x =dat.subj.time,
                               y=dat.percent_looks )) +
  geom_line () +

  labs ( x= "Time",
         y= "Percentage Looks to Target",
         title= "Line Graph of % Looks to Target Over Time" )
```

```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```

Line Graph of % Looks to Target Over Time



```
#scatterplot with histogram showing distribution and regression line  
#https://okanbulut.github.io/bigdata/visualizing-big-data.html  
p1 <- ggplot(data=looks_over_time,  
             mapping = aes(x =dat.subj.time, y =dat.percent_looks)) +  
  geom_point() +  
  geom_smooth(method = "loess") +  
  labs(x = "Time", y = "Percent Looks at Target") +  
  theme_bw()  
  
# Replace "histogram" with "boxplot" or "density" for other types  
ggMarginal(p1, type = "histogram")
```

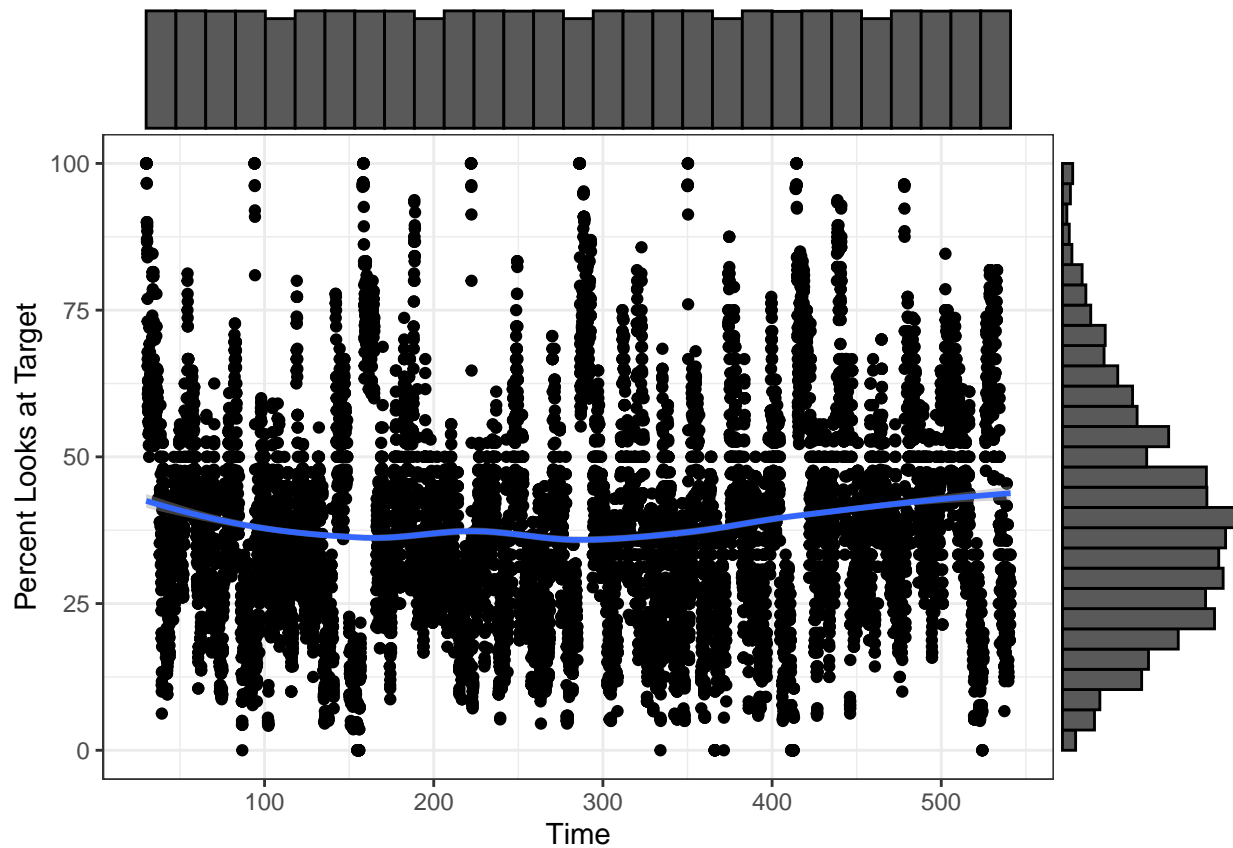
```
## 'geom_smooth()' using formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

```
## Warning: Removed 3 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



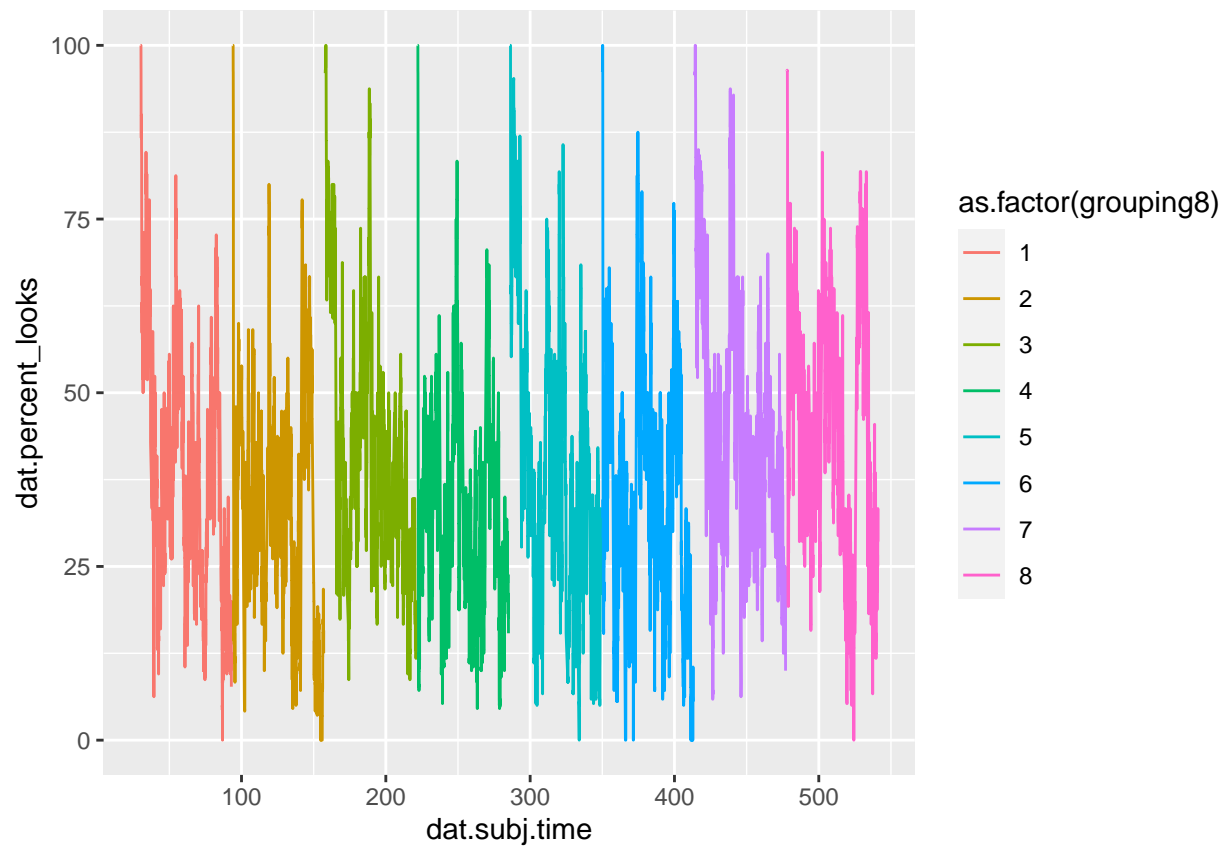
```
#make a new column with groups of
# divided by 210 = 64 groups (one per trial)
looks_over_time$grouping64 = rep(1:ceiling(nrow(looks_over_time)/210), each = 210)[1:nrow(looks_over_time)]
# divided by 1680 = 8 groups (one per block)
looks_over_time$grouping8 = rep(1:ceiling(nrow(looks_over_time)/1680), each = 1680)[1:nrow(looks_over_time)]

#groupings with time
# divided by 210 = 64 groups
looks_over_time$trialG<-rep(c(1:210),times=64)
# divided by 1680 = 8 groups
looks_over_time$blockG<-rep(c(1:1680),times=8)
```

```
#graphs grouped by 8 blocks and 64 trials

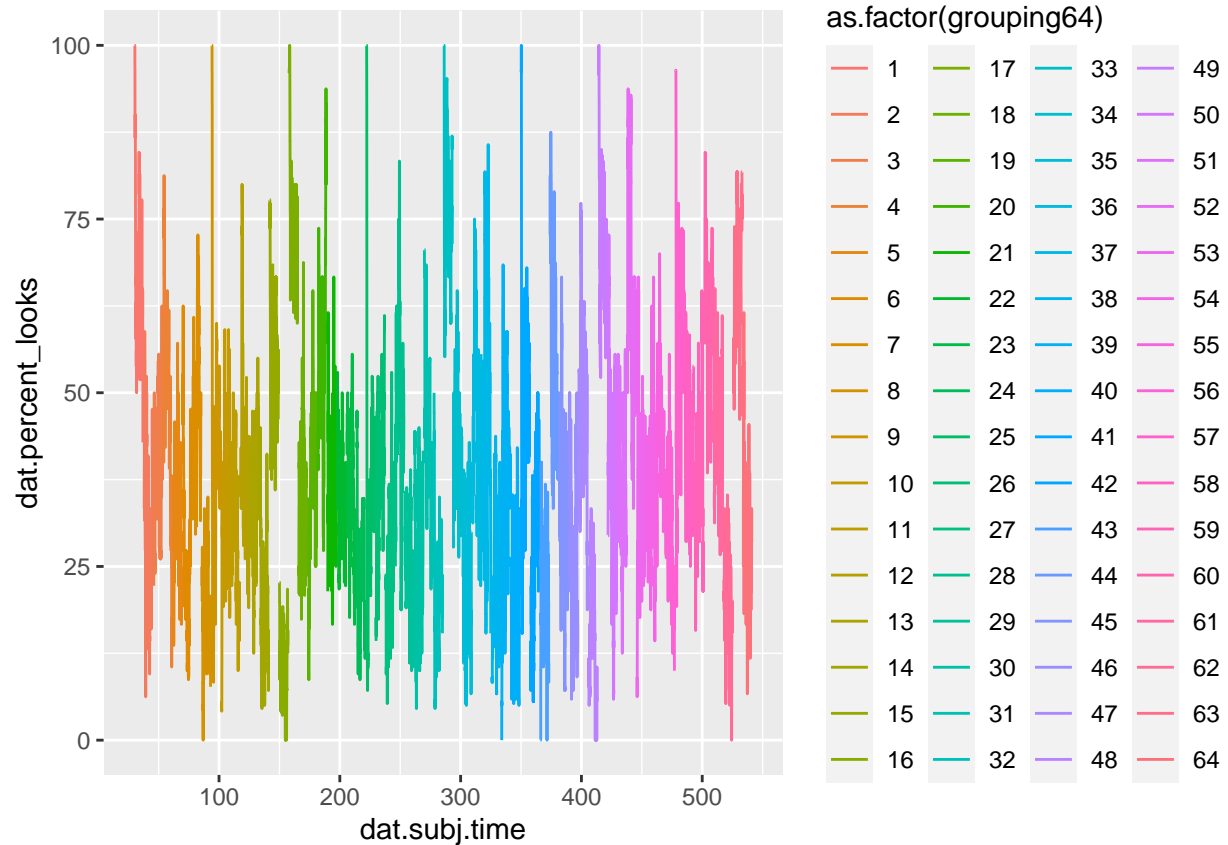
ggplot (looks_over_time, aes ( x =dat.subj.time,
                             y=dat.percent_looks, group=grouping8, colour=as.factor(grouping8))) +
  geom_line ()
```

```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```



```
ggplot (looks_over_time, aes ( x =dat.subj.time,
                               y=dat.percent_looks, group=grouping64, colour=as.factor(grouping64))) +
  geom_line ()
```

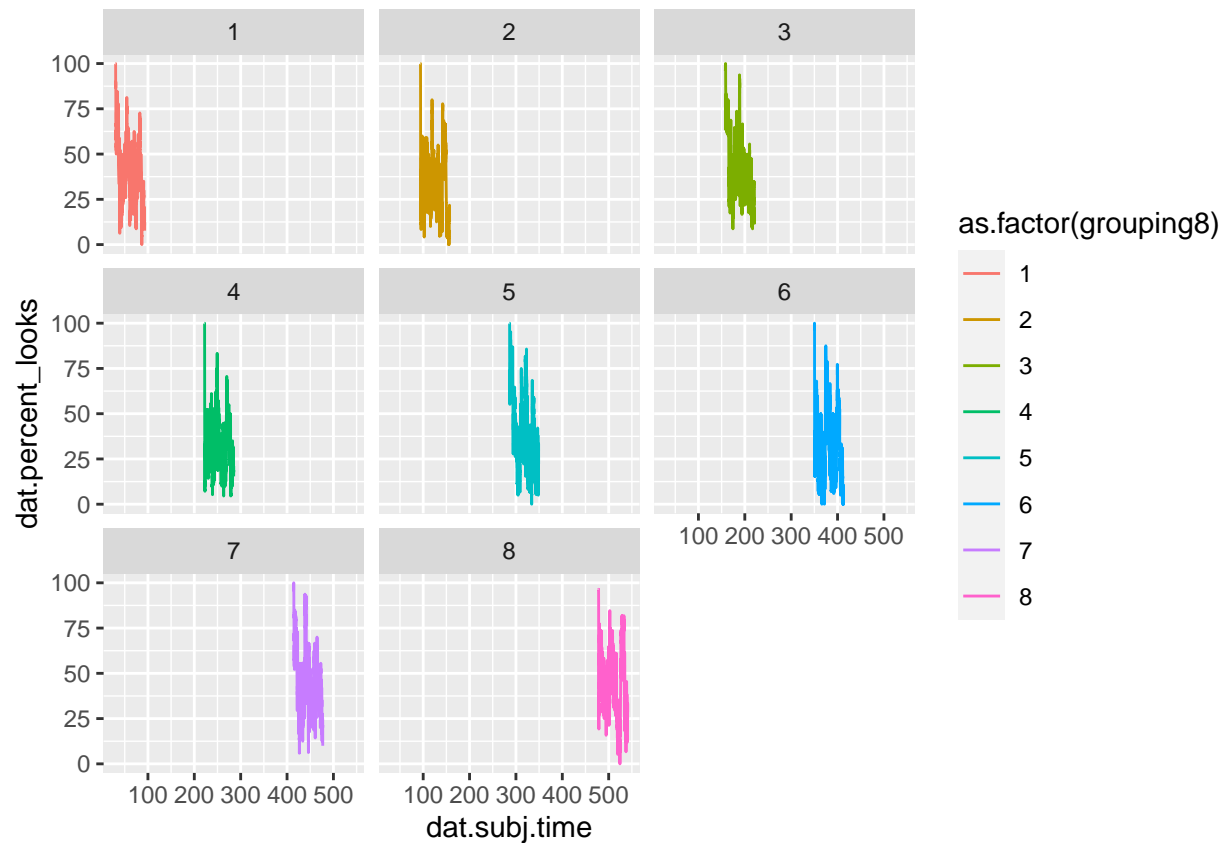
Warning: Removed 3 row(s) containing missing values (geom_path).



#graphs faceted by 8 and 64

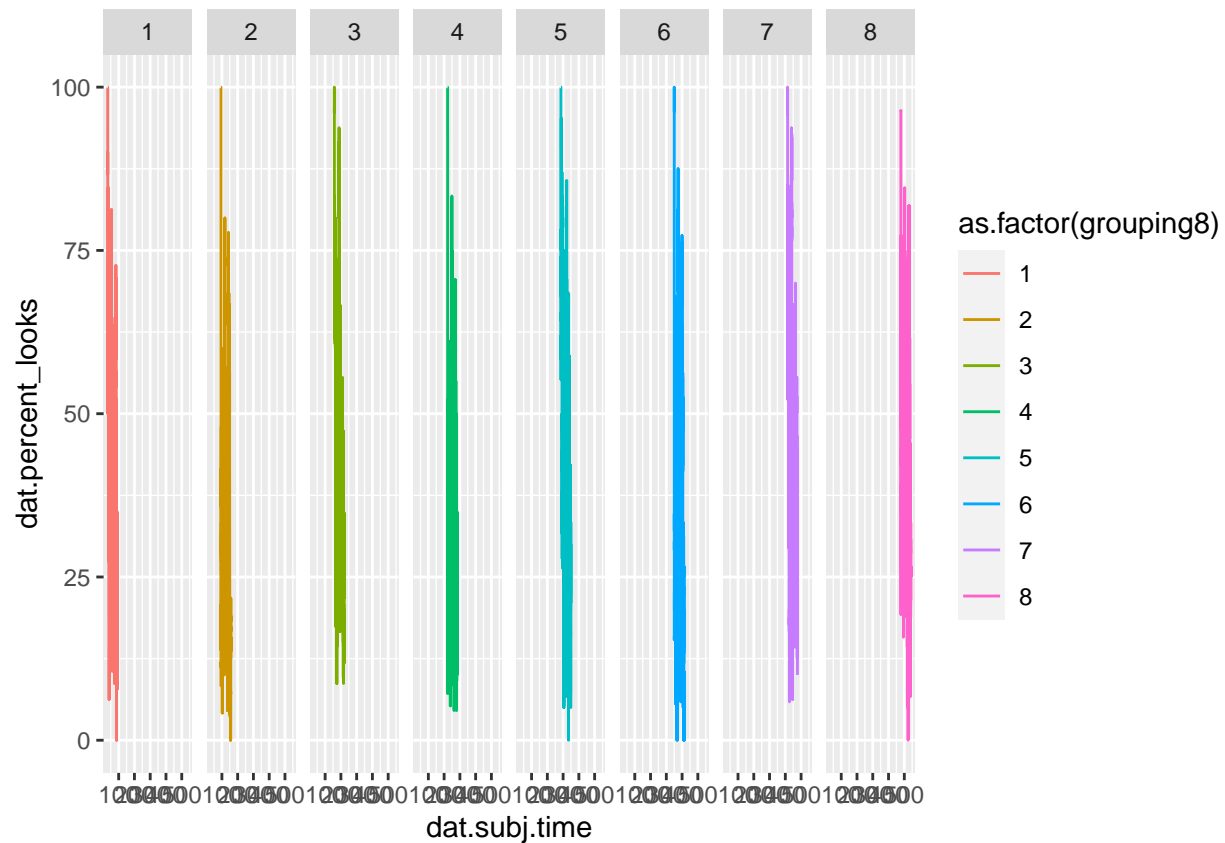
```
#1
ggplot (looks_over_time, aes ( x =dat.subj.time,
                               y=dat.percent_looks, group=grouping8, colour=as.factor(grouping8))) +
  geom_line () + facet_wrap(~grouping8)
```

Warning: Removed 3 row(s) containing missing values (geom_path).



```
#2
ggplot (looks_over_time, aes ( x =dat.subj.time,
                               y=dat.percent_looks, group=grouping8, colour=as.factor(grouping8))) +
  geom_line () + facet_grid(. ~grouping8)
```

```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```

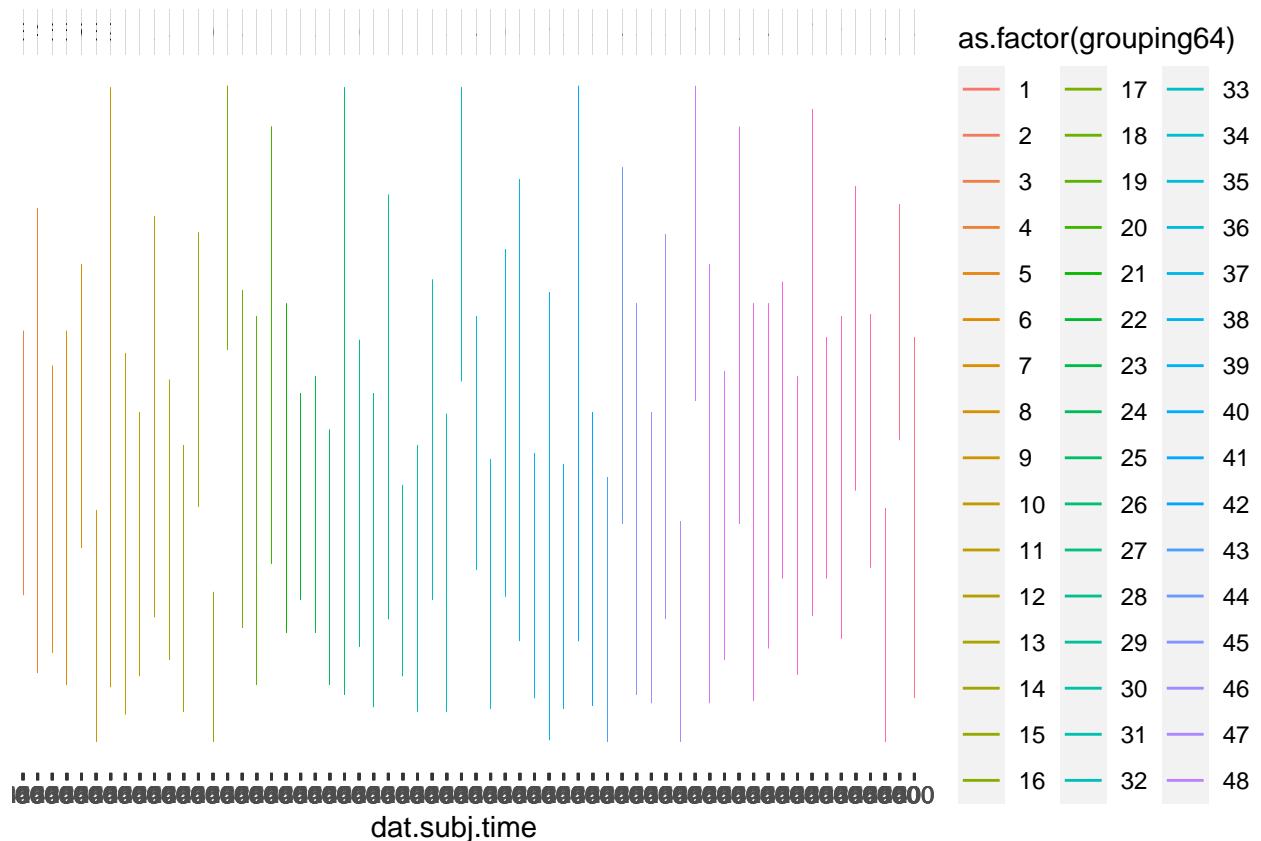


```
ggsave('Facet1.png',limitsize = FALSE,width =200, height = 100, units='cm')
```

```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```

```
#3
ggplot (looks_over_time, aes ( x =dat.subj.time,
                              y=dat.percent_looks, group=grouping64, colour=as.factor(grouping64))) +
  geom_line () + facet_grid(. ~grouping64)
```

```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```

```
ggsave('Facet2.png',limitsize = FALSE)
```

```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 3 row(s) containing missing values (geom_path).
```

```
#does not work
#facet_wrap(~groupig64)
```

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v tibble 3.1.8      v purrr 0.3.4
## v tidyr 1.2.0      v stringr 1.4.0
## v readr 2.1.2      v forcats 0.5.1
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

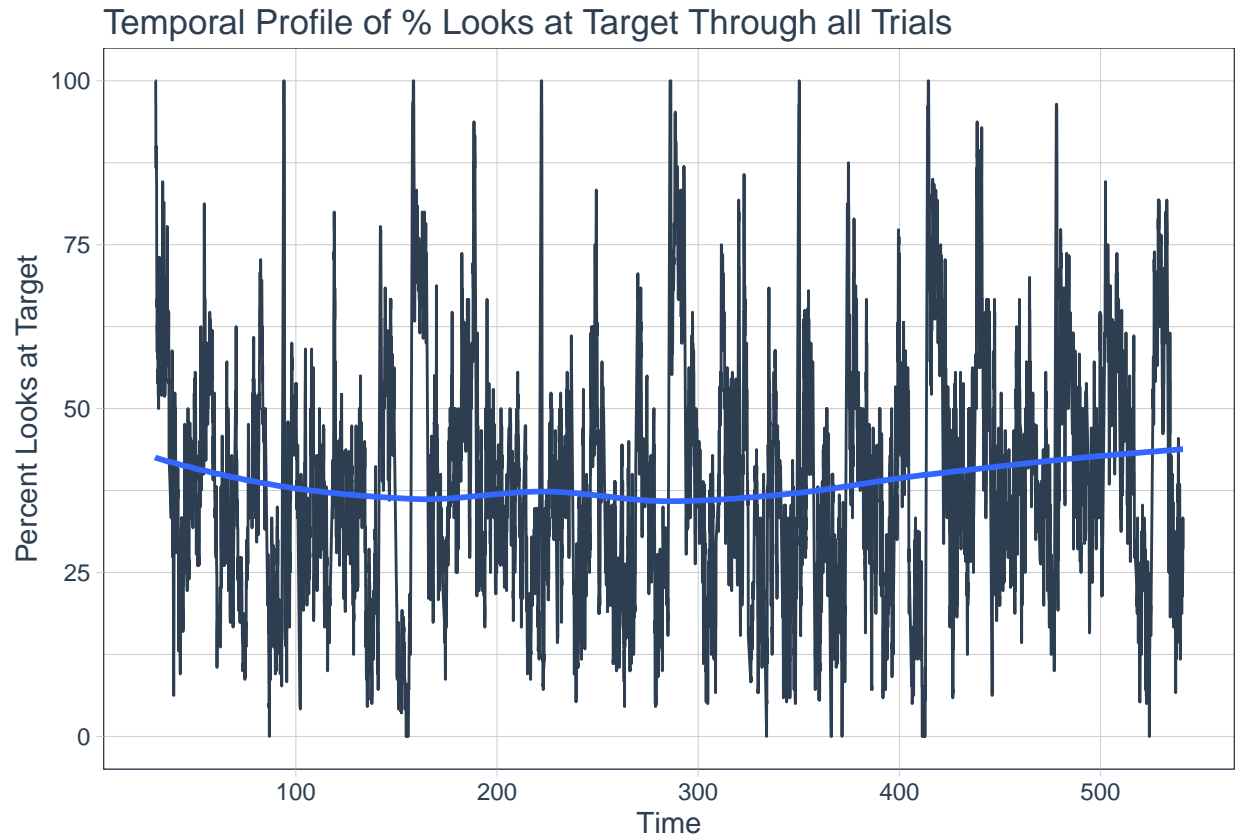
```
##
## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union

library(timetk)
# Setup for the plotly charts (# FALSE returns ggplots)
interactive <- FALSE
#https://cran.r-project.org/web/packages/timetk/vignettes/TK04_Plotting_Time_Series.html

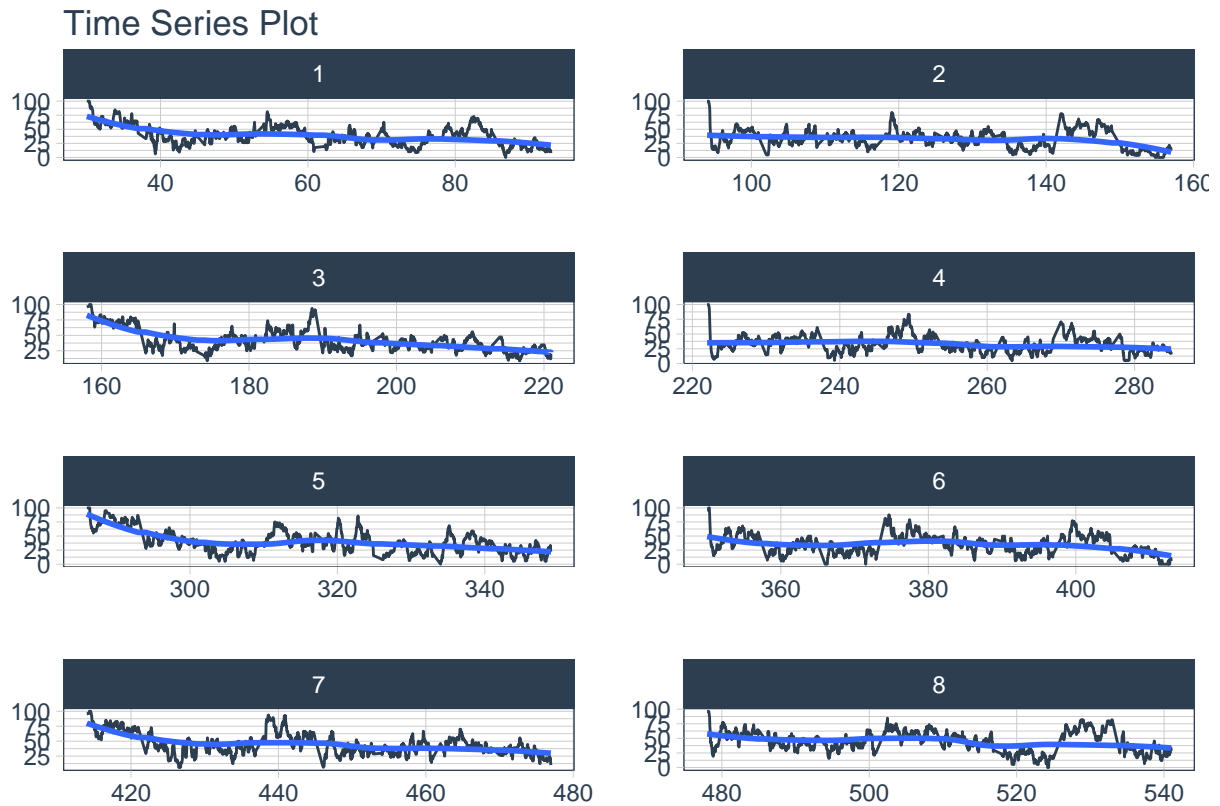
looks_over_time %>%
  plot_time_series(dat.subj.time, dat.percent_looks,
    .interactive = interactive,
    .plotly_slider = TRUE,

    # Customization
    .title = "Temporal Profile of % Looks at Target Through all Trials",
    .x_lab = "Time",
    .y_lab = "Percent Looks at Target",
    .color_lab = "Block") +
  scale_y_continuous(labels = scales::comma_format())

## Warning: Removed 3 row(s) containing missing values (geom_path).
## Removed 3 row(s) containing missing values (geom_path).
```

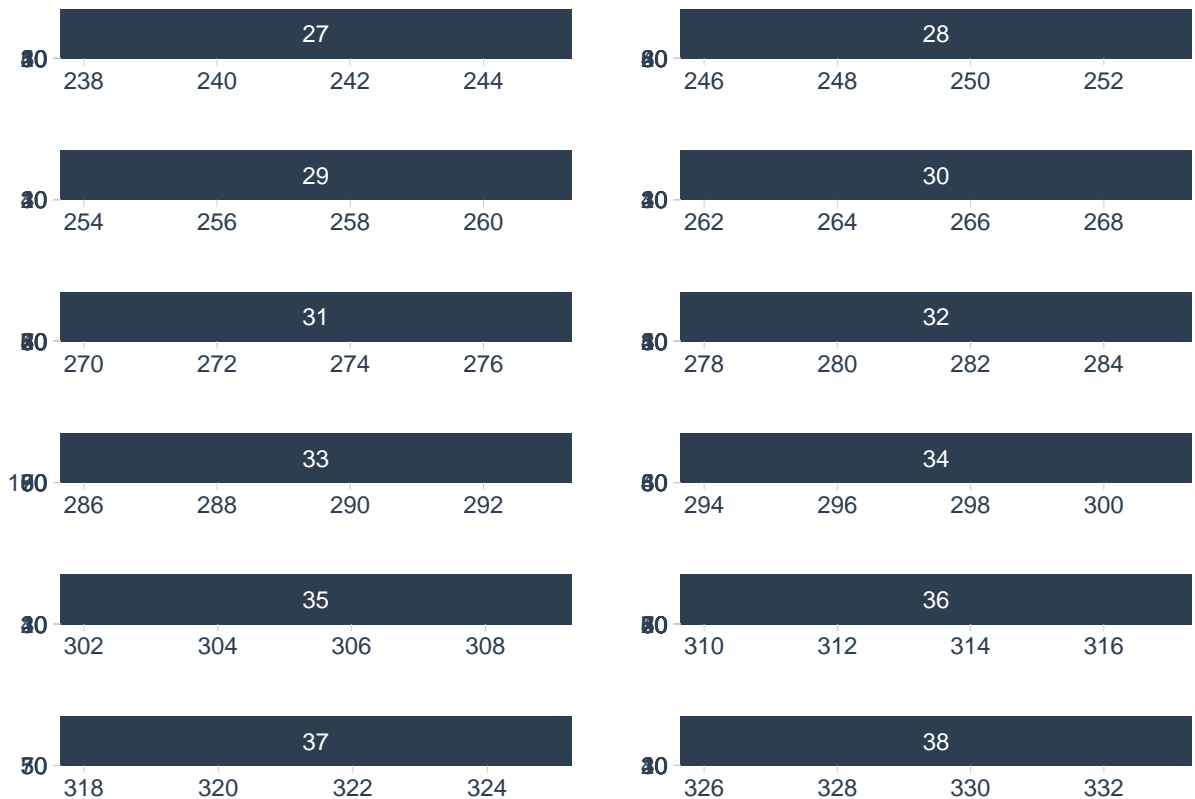


```
looks_over_time %>%
  group_by(grouping8) %>%
  plot_time_series(dat.subj.time, dat.percent_looks,
    .facet_ncol = 2,
    .facet_scales = "free",
    .interactive = interactive)
```



```
ggsave('TimeSeries1.png',limitsize = FALSE,width =200, height = 200, units='cm')

looks_over_time %>%
  group_by(grouping64) %>%
  plot_time_series(dat.subj.time, dat.percent_looks,
    .facet_ncol = 2,
    .facet_nrow = 32,
    .facet_scales = "free",
    .interactive = interactive)
```



```
ggsave('TimeSeries2.png',limitsize = FALSE,width=100, height = 200, units='cm')
```

```
a <- looks_over_time %>% mutate(group8=dplyr::recode(grouping8,'1'='A','2'='B','3'='C','4'='D','5'='E',
looks_over_time <- looks_over_time %>% mutate(group8=dplyr::recode(grouping8,'1'='A','2'='B','3'='C','4
```

#convert groupings into times and create graph where every block and every trial overlap (above)

#multiple line graphs

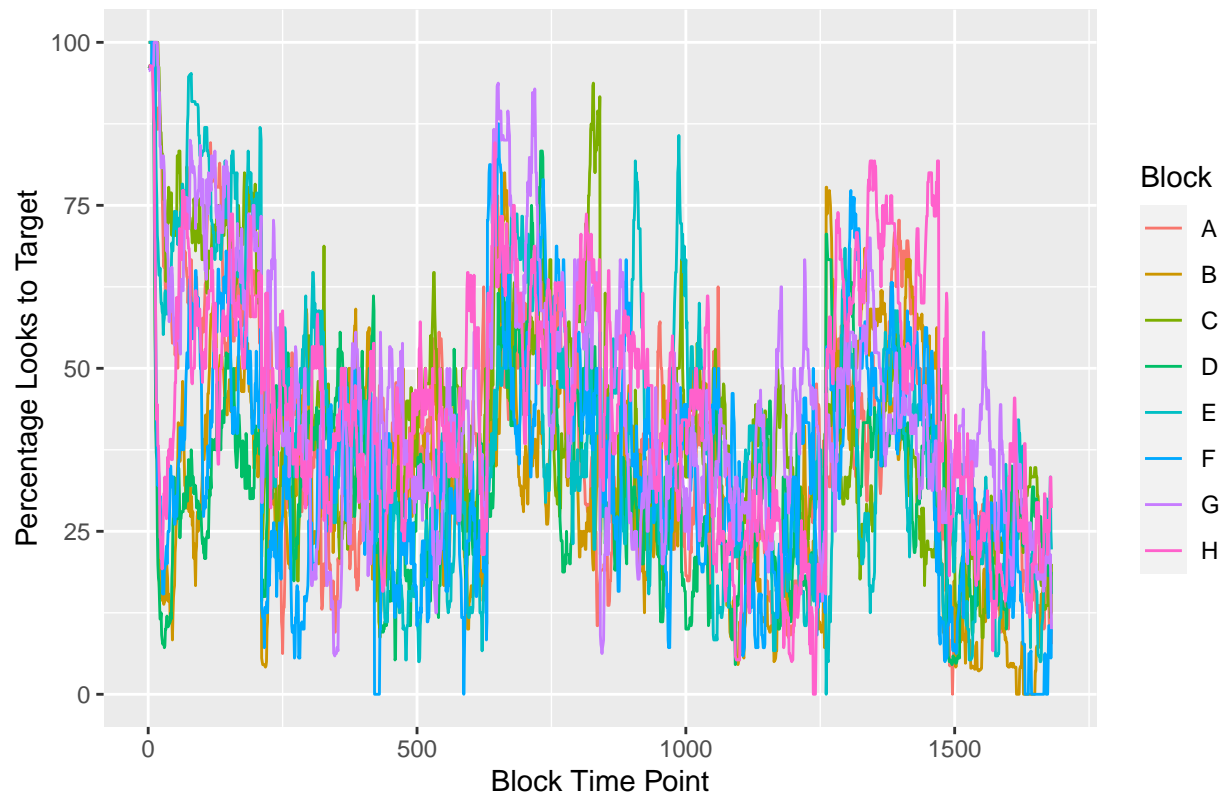
#time is in subjective variable (1 out of trial remainder, starting at 1)

#Block level

```
a %>%
ggplot( aes(x=blockG, y=dat.percent_looks, group=group8, color=group8)) +
geom_line() +
```

```
labs ( x= "Block Time Point",
y= "Percentage Looks to Target",
colour= "Block",
title= "Multiple Line Graph of 8 Trial Blocks" )
```

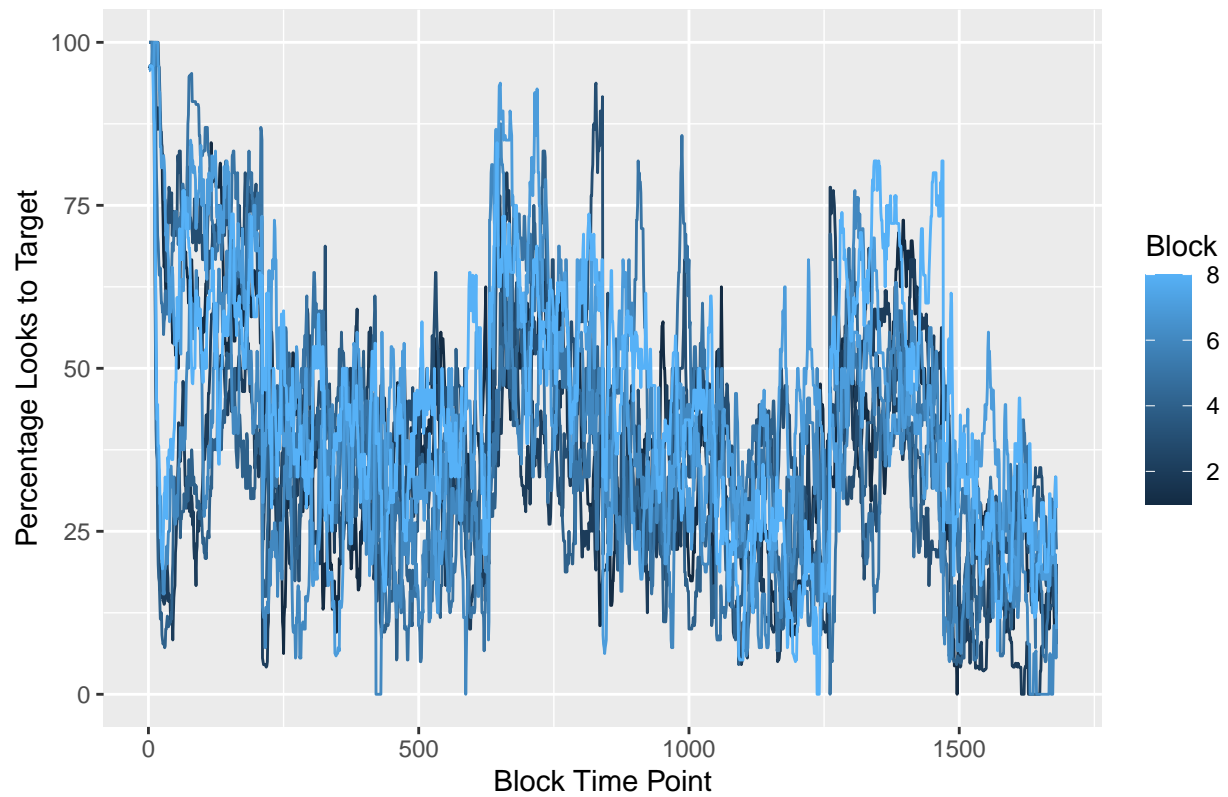
Multiple Line Graph of 8 Trial Blocks



```
#Block level version 2
looks_over_time %>%
  ggplot( aes(x=blockG, y=dat.percent_looks, group=grouping8, color=grouping8)) +
  geom_line() +

  labs ( x= "Block Time Point",
        y= "Percentage Looks to Target",
        colour= "Block",
        title= "Multiple Line Graph of 8 Trial Blocks" )
```

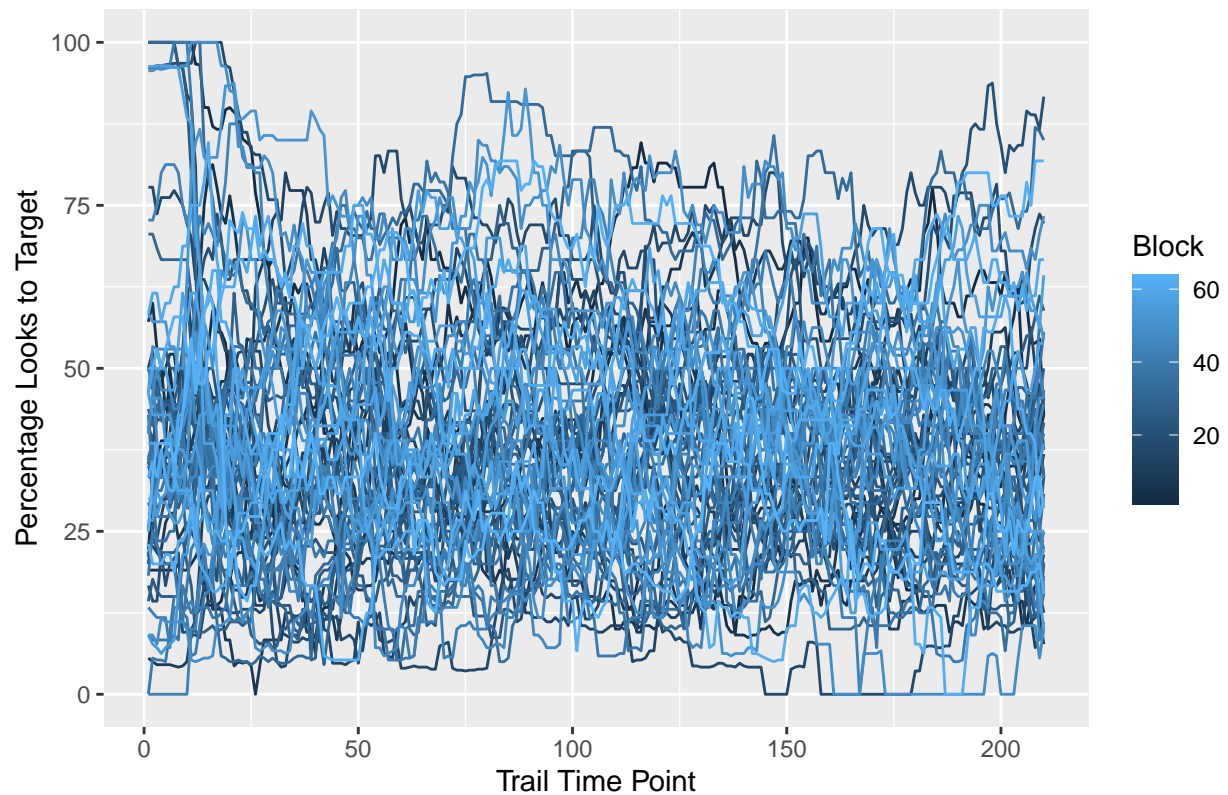
Multiple Line Graph of 8 Trial Blocks



```
#Trial level
looks_over_time %>%
  ggplot( aes(x=trialG, y=dat.percent_looks, group=grouping64, color=grouping64)) +
  geom_line() +

  labs ( x= "Trail Time Point",
        y= "Percentage Looks to Target",
        colour= "Block",
        title= "Multiple Line Graph of Each Trial" )
```

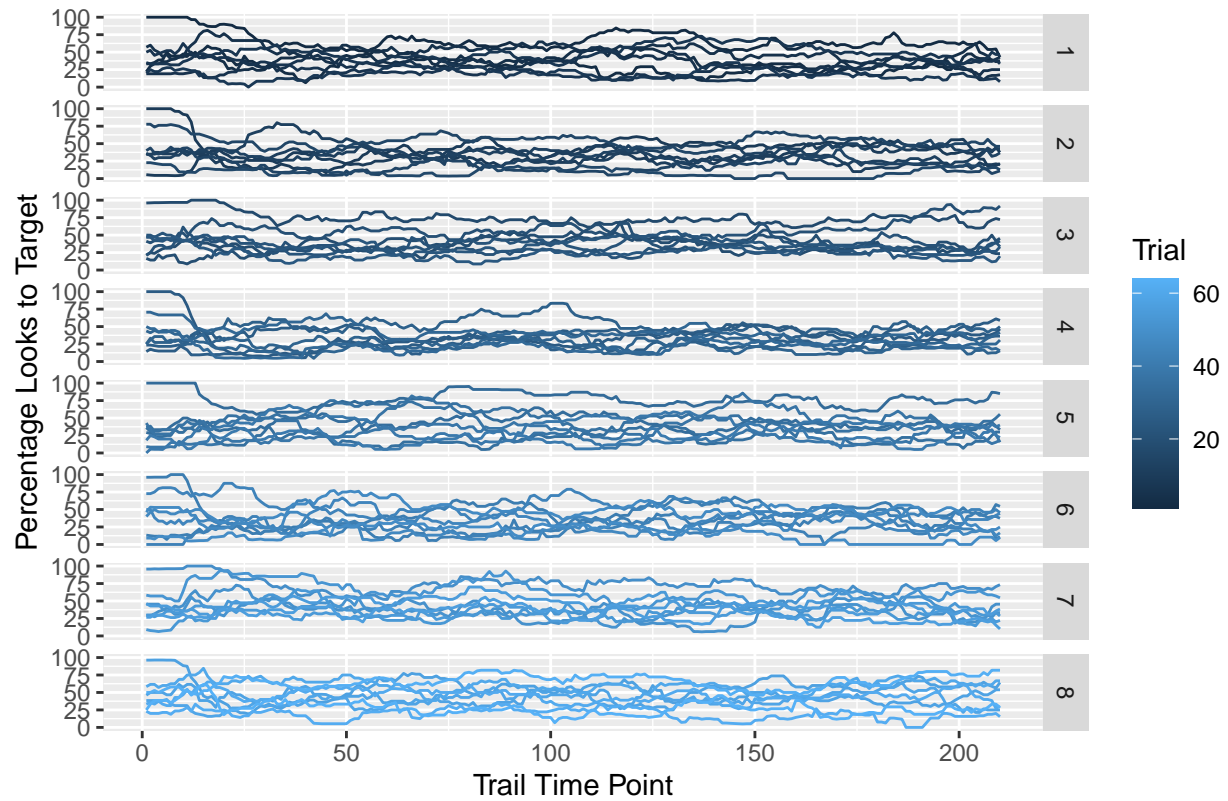
Multiple Line Graph of Each Trial



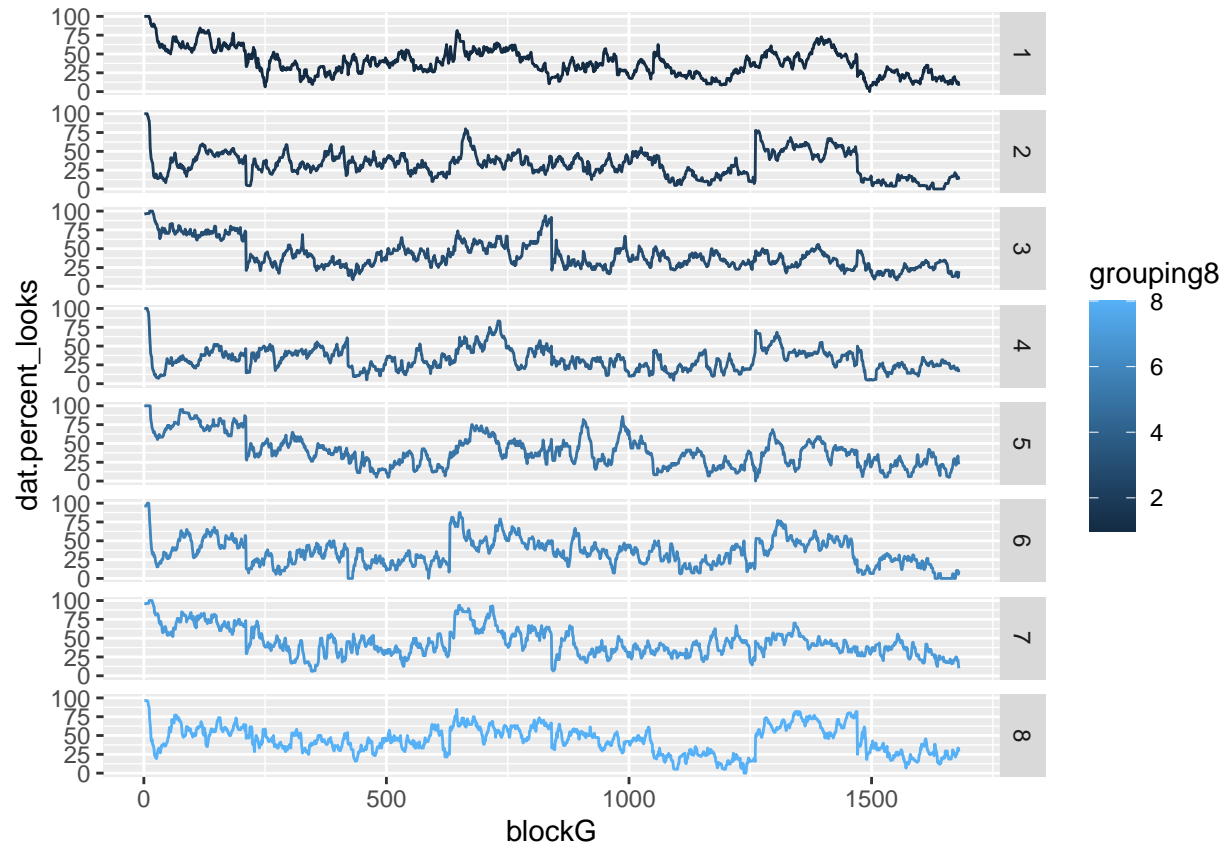
```
#Every Block individually, with individual trial lines
looks_over_time %>%
  ggplot( aes(x=trialG, y=dat.percent_looks, group=grouping64, color=grouping64)) +
  geom_line() + facet_grid(grouping8~ .) +

  labs ( x= "Trail Time Point",
        y= "Percentage Looks to Target",
        colour= "Trial",
        title= "Multiple Line Graph of Every Trial Grouped by Block" )
```

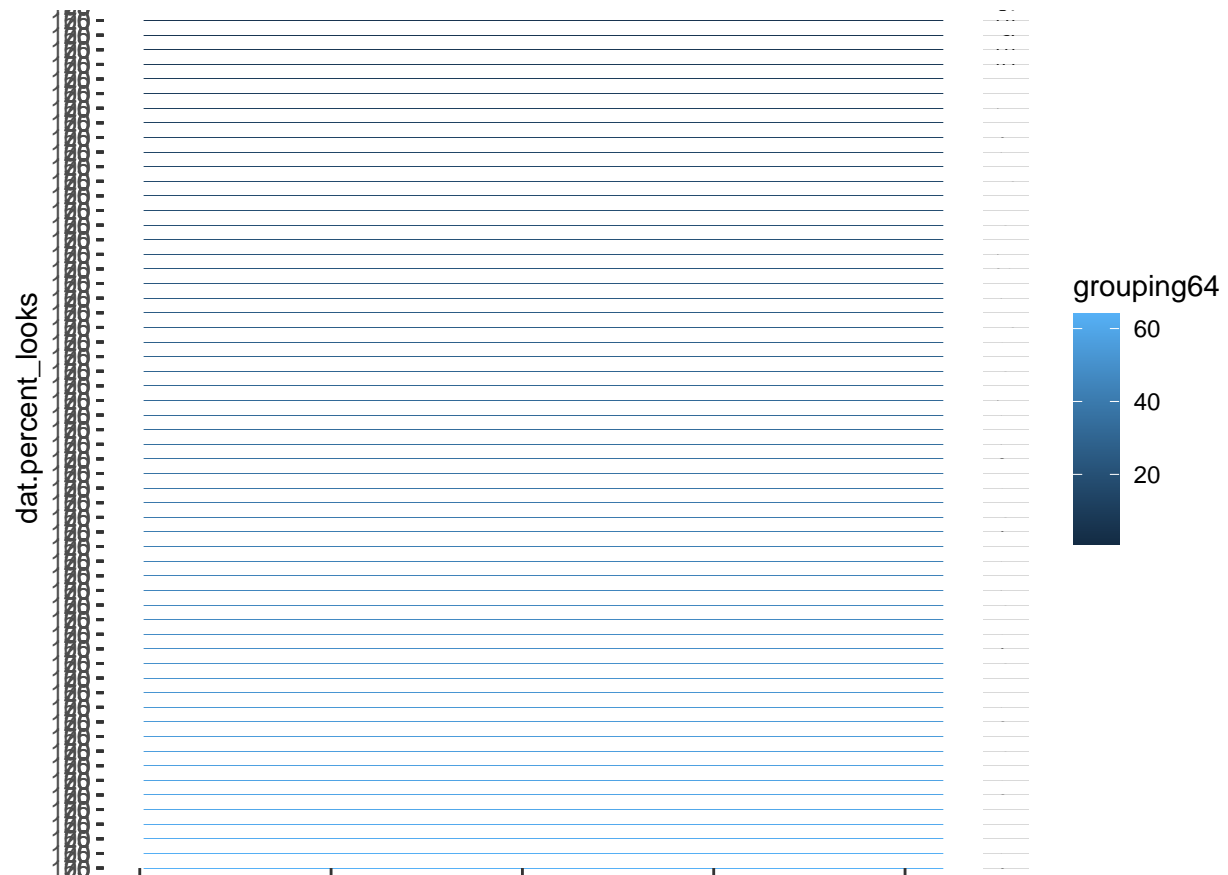
Multiple Line Graph of Every Trial Grouped by Block



```
#facet grid
looks_over_time %>%
  ggplot( aes(x=blockG, y=dat.percent_looks, group=grouping8, color=grouping8)) +
    geom_line() + facet_grid(grouping8~ .)
```

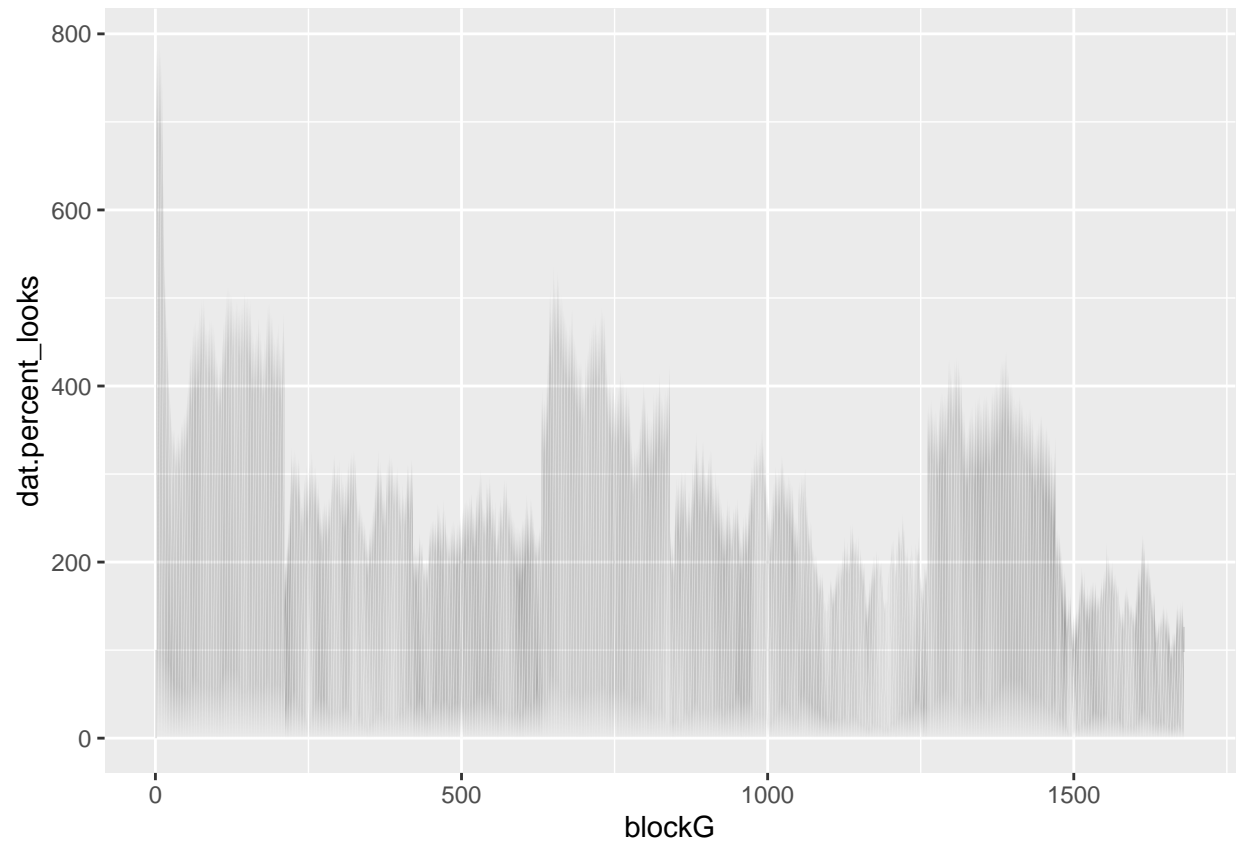



```
looks_over_time %>%
  ggplot( aes(x=trialG, y=dat.percent_looks, group=grouping64, color=grouping64)) +
  geom_line() + facet_grid(grouping64~ .)
```

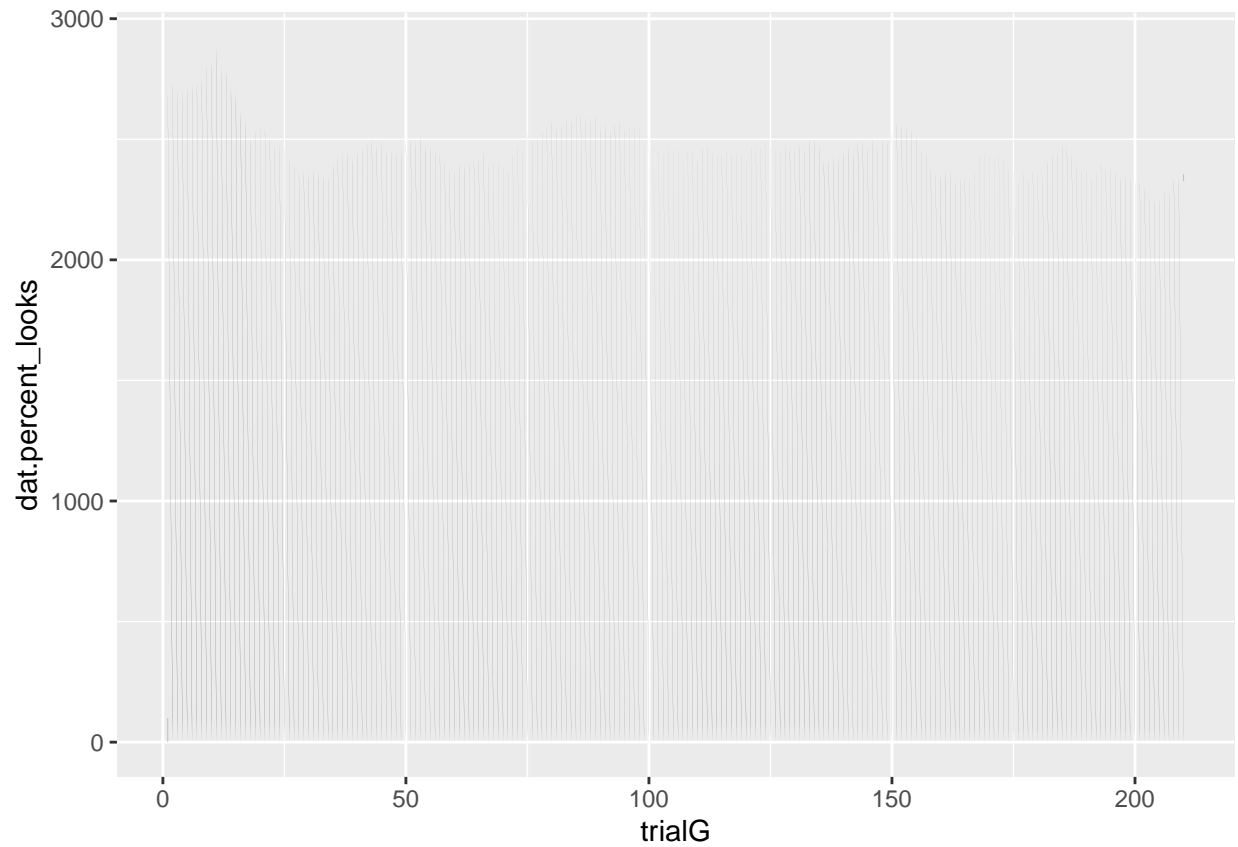


```
#ggsave('Mydata.png',limitsize = FALSE,width =100, height = 200, units='cm')
```

```
#area chart  
ggplot(looks_over_time, aes(x=blockG, y=dat.percent_looks)) + geom_area()
```

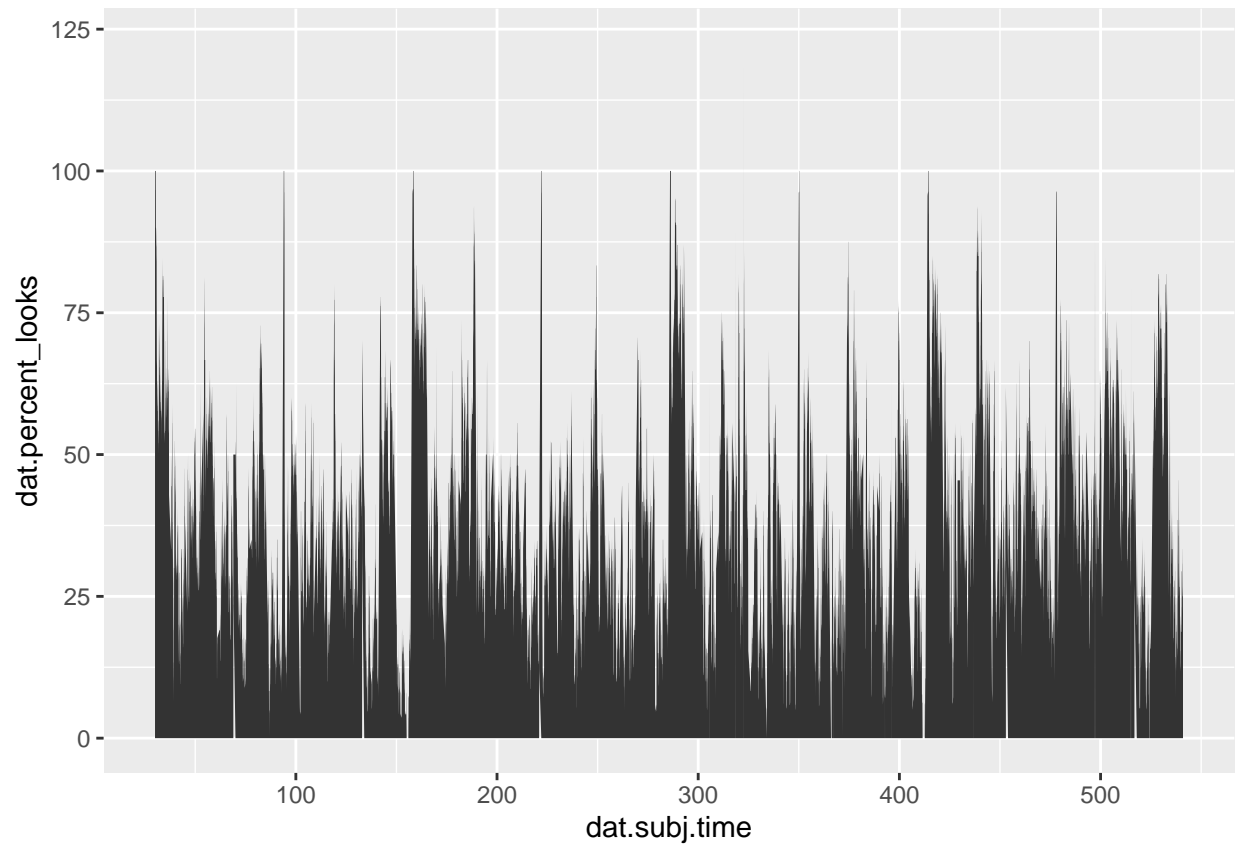


```
ggplot(looks_over_time, aes(x=trialG, y=dat.percent_looks)) + geom_area()
```



```
ggplot (looks_over_time, aes ( x =dat.subj.time,  
                                y=dat.percent_looks )) + geom_area()
```

```
## Warning: Removed 3 rows containing missing values (position_stack).
```



```
library(viridis)
```

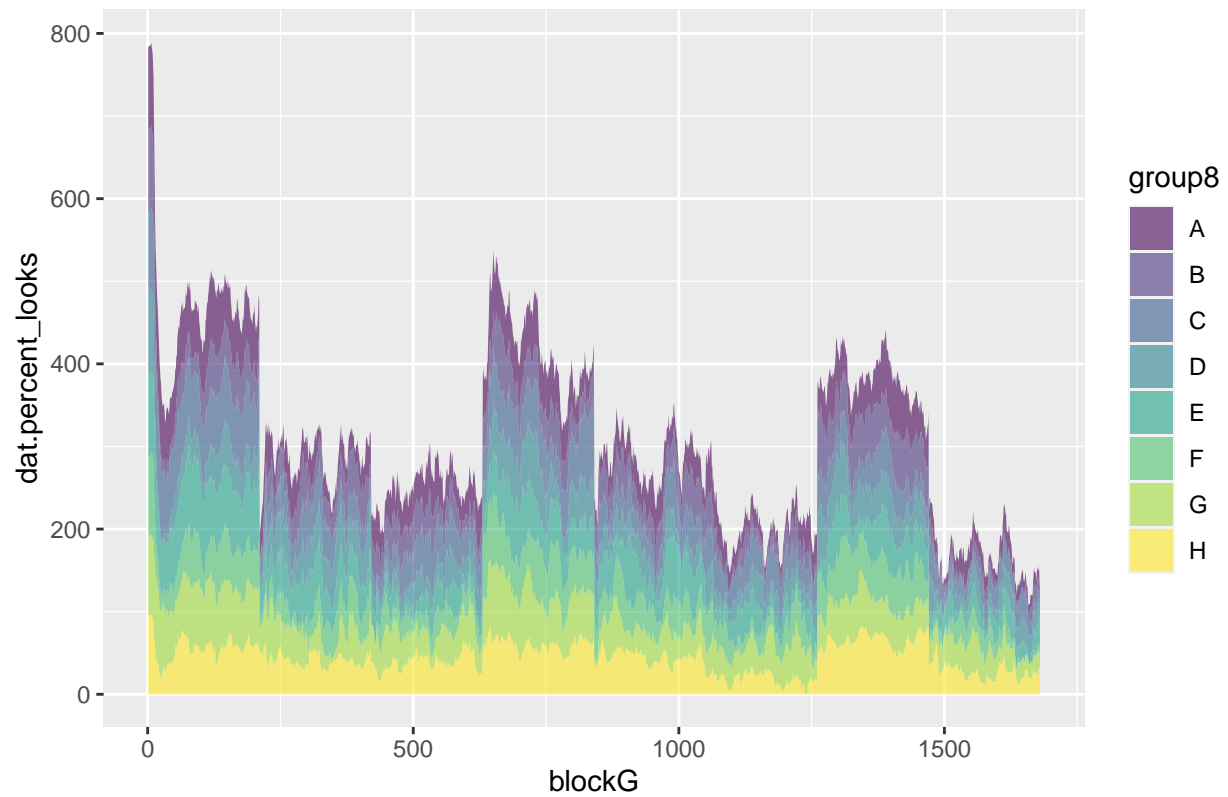
```
## Loading required package: viridisLite
```

```
#stacked area chart
```

```
cols <- hcl.colors(8, palette = "ag_Sunset")
```

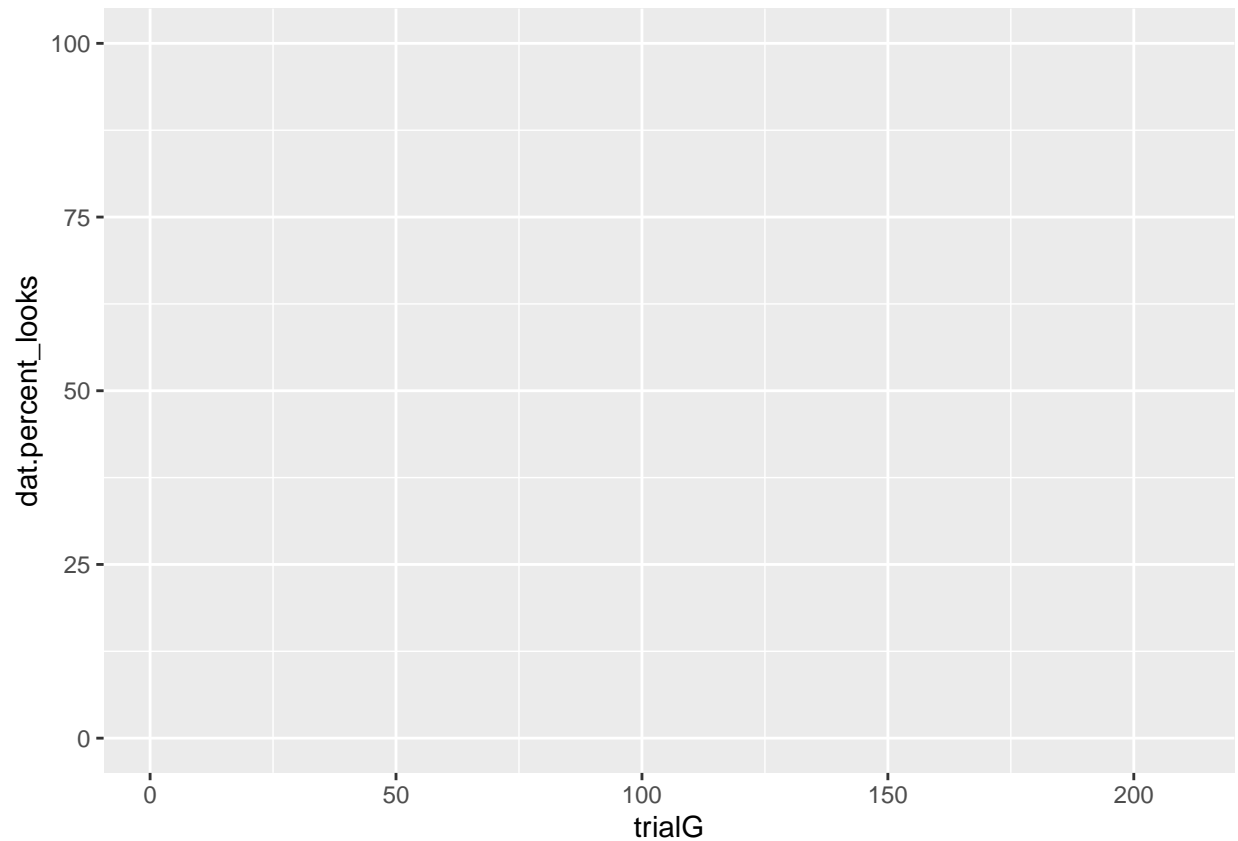
```
ggplot(a, aes(x=blockG, y=dat.percent_looks, fill=group8)) +  
  geom_area(alpha=0.6) +  
  scale_fill_viridis(discrete = T) +  
  ggtitle("Stacked Area Chart for All Blocks")
```

Stacked Area Chart for All Blocks



#

```
ggplot(a, aes(x=trialG, y=dat.percent_looks, fill=grouping64))
```



```
ggsave('DatStacked.png',limitsize = FALSE,width =100, height = 200, units='cm')
```

```
#http://cran.nexr.com/web/packages/tabplot/vignettes/tabplot-vignette.html
```

```
#grouping across blocks by trial number out of 8.
```

```
x <- rep(letters[1:8], 8)
looks_over_time$newCol <- rep(x,each=210)
```

```
#graph
```

```
#label blocks a HLLHLLTT
```

```
block_names <- c(
  "1"="H",
  "2"="L",
  "3"="L",
  "4"="H",
  "5"="L",
  "6"="L",
  "7"="T",
  "8"="T")
```

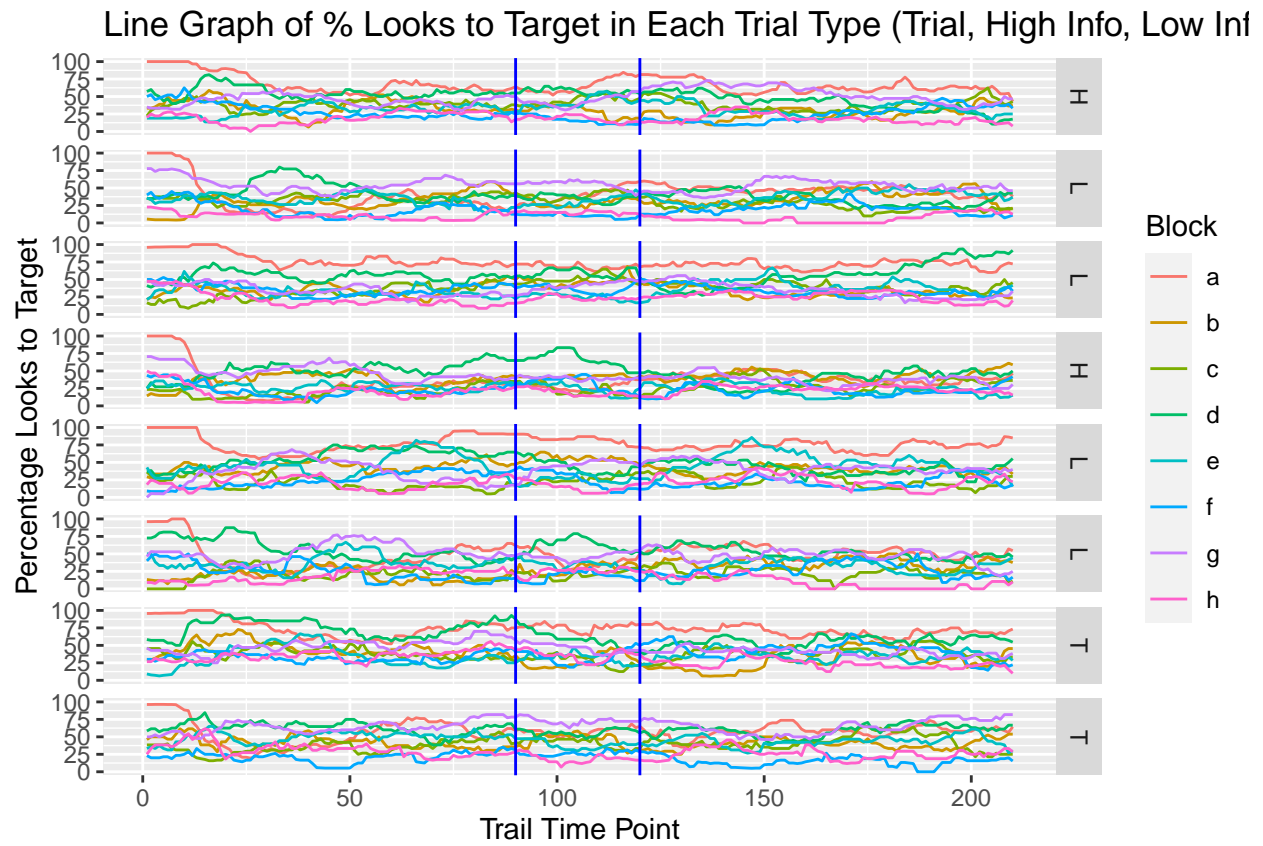
```
looks_over_time %>%
  ggplot( aes(x=trialG, y=dat.percent_looks, group=newCol, colour=newCol)) +
  geom_line() + facet_grid(grouping8~ ., labeller = as_labeller(block_names)) +
```

```

geom_vline(xintercept = 90, colour="blue") +
geom_vline(xintercept = 120, colour="blue") +

labs ( x= "Trail Time Point",
       y= "Percentage Looks to Target",
       title= "Line Graph of % Looks to Target in Each Trial Type (Trial, High Info, Low Info)",
       colour = "Block")

```



#as previous, but plot mean line

```

block_names <- c(
  "1"="H",
  "2"="L",
  "3"="L",
  "4"="H",
  "5"="L",
  "6"="L",
  "7"="T",
  "8"="T")

looks_over_time %>%
  ggplot( aes(x=trialG, y=dat.percent_looks, group=newCol, colour=newCol)) +
  geom_line() + facet_grid(grouping8~ ., labeller = as_labeller(block_names)) +

```



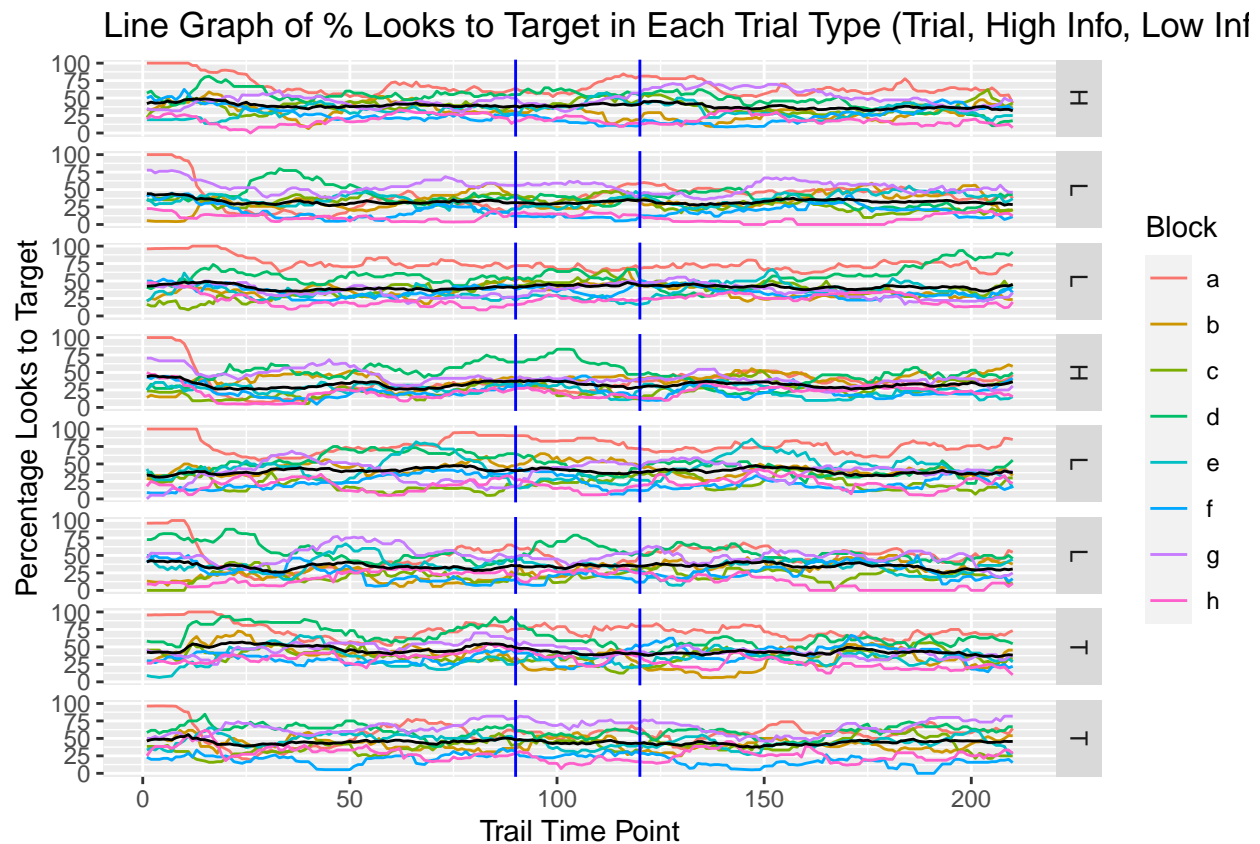
```

geom_vline(xintercept = 90, colour="blue") +
geom_vline(xintercept = 120, colour="blue") +
stat_summary(aes(y =dat.percent_looks,group=1), fun.y=mean, colour="black", geom="line",group=1)

labs ( x= "Trail Time Point",
       y= "Percentage Looks to Target",
       title= "Line Graph of % Looks to Target in Each Trial Type (Trial, High Info, Low Info)",
       colour = "Block")

```

Warning: 'fun.y' is deprecated. Use 'fun' instead.



#three plots, seperated by T, H, L

```

looks_over_time <- looks_over_time %>% mutate(conditionGroups=dplyr::recode(newCol,
  "a"="H",
  "b"="L",
  "c"="L",
  "d"="H",
  "e"="L",
  "f"="L",
  "g"="T",
  "h"="T"))

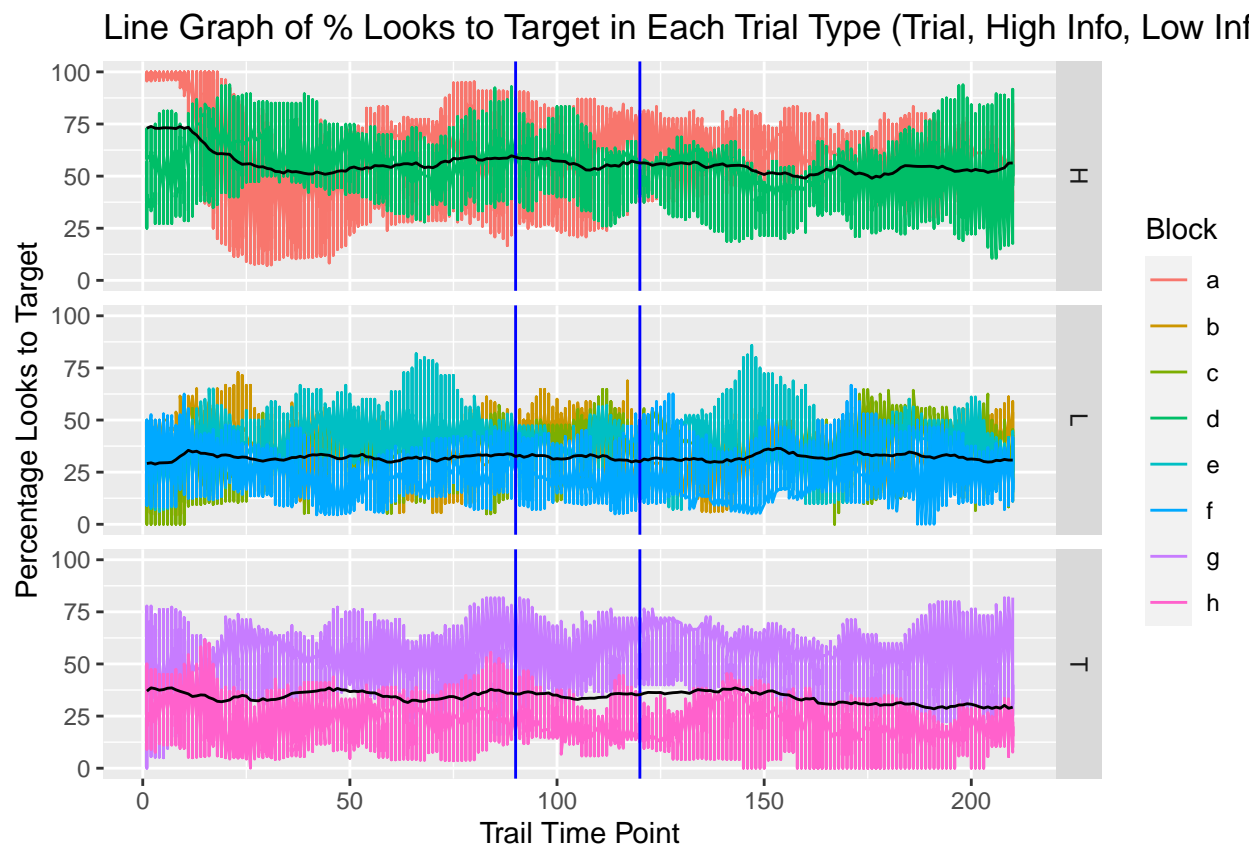
looks_over_time %>%
  ggplot( aes(x=trialG, y=dat.percent_looks, group=newCol, colour=newCol)) +

```

```
geom_line() + facet_grid(conditionGroups~ .) +
  geom_vline(xintercept = 90, colour="blue") +
  geom_vline(xintercept = 120, colour="blue") +
  stat_summary(aes(y =dat.percent_looks,group=1), fun.y=mean, colour="black", geom="line",group=1)

labs ( x= "Trail Time Point",
  y= "Percentage Looks to Target",
  title= "Line Graph of % Looks to Target in Each Trial Type (Trial, High Info, Low Info)",
  colour = "Block")
```

Warning: 'fun.y' is deprecated. Use 'fun' instead.



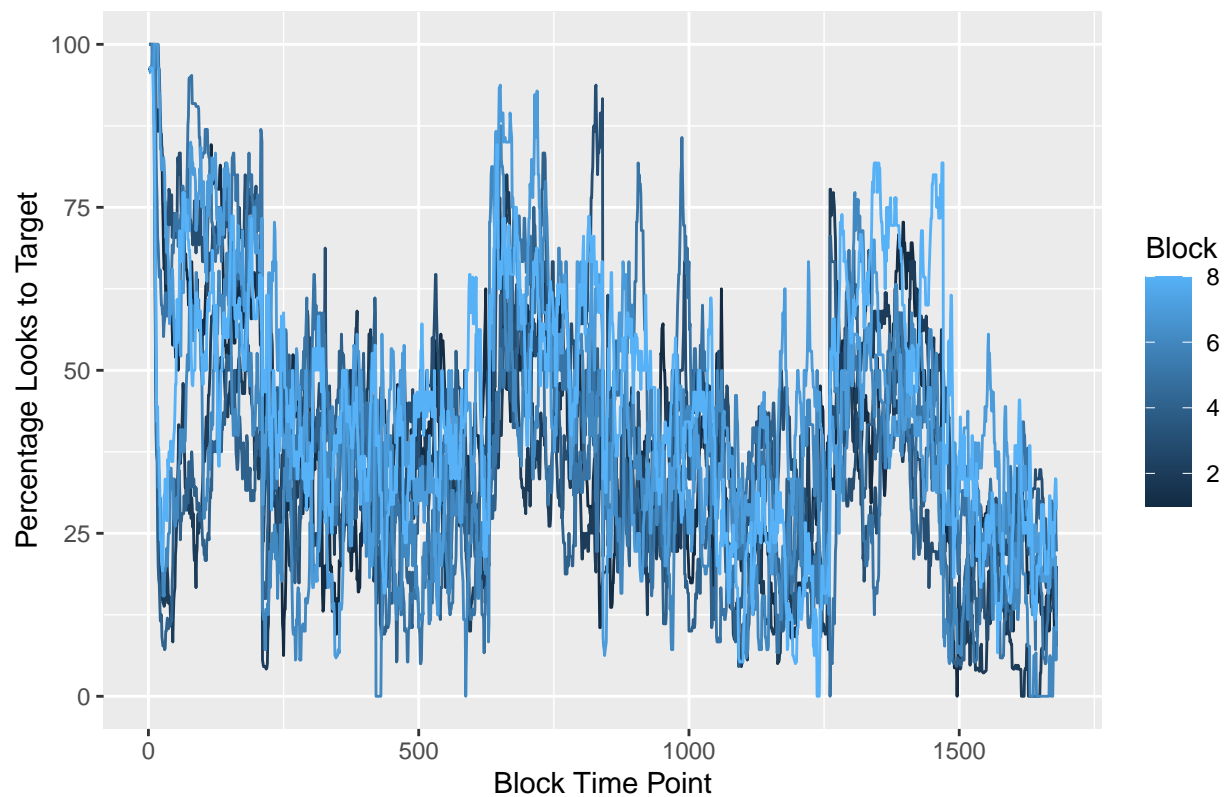
####plot subset of graph

####EXEMPLAR - Block level

```
a %>%
ggplot( aes(x=blockG, y=dat.percent_looks, group=grouping8, color=grouping8)) +
geom_line() +

labs ( x= "Block Time Point",
  y= "Percentage Looks to Target",
  colour= "Block",
  title= "Multiple Line Graph of 8 Trial Blocks" )
```

Multiple Line Graph of 8 Trial Blocks



```
#### organise data
```

```
#order block G from smallest to largest number
arranged <- arrange(looks_over_time, blockG)
```

```
#divide dataset
```

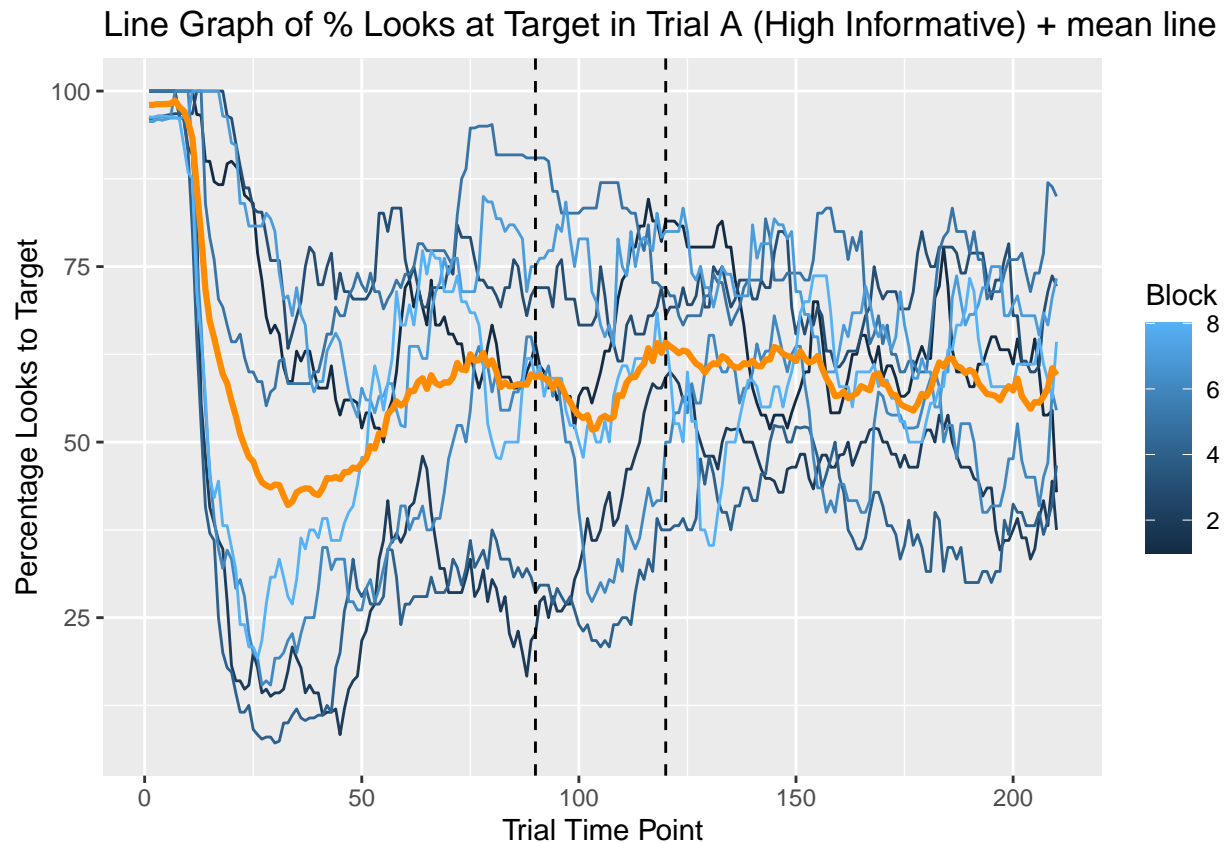
```
data_1 <- arranged[1:1680, ]
data_2 <- arranged[1681:3360, ]
data_3 <- arranged[3361:5040, ]
data_4 <- arranged[5041:6720, ]
data_5 <- arranged[6721:8400, ]
data_6 <- arranged[8401:10080, ]
data_7 <- arranged[10081:11760, ]
data_8 <- arranged[11761:13440, ]
```

```
####create graphs
```

```
data_1 %>%
  ggplot( aes(x=blockG, y=dat.percent_looks,group=grouping8, color=grouping8)) +
  geom_line() +
  geom_vline(xintercept = 90, colour="black", linetype = "dashed") +
  geom_vline(xintercept = 120, colour="black", linetype = "dashed") +
  stat_summary(aes(y =dat.percent_looks,group=1), fun.y=mean, size=1.1, colour="darkorange", geom="point",
    position="dodge",
    aes(x=blockG, y=dat.percent_looks,group=grouping8, color=grouping8)) +
  labs ( x= "Trial Time Point",
    y= "Percentage Looks to Target",
    title="Multiple Line Graph of 8 Trial Blocks")
```

```
colour= "Block",
title= "Line Graph of % Looks at Target in Trial A (High Informative) + mean line" )
```

Warning: 'fun.y' is deprecated. Use 'fun' instead.

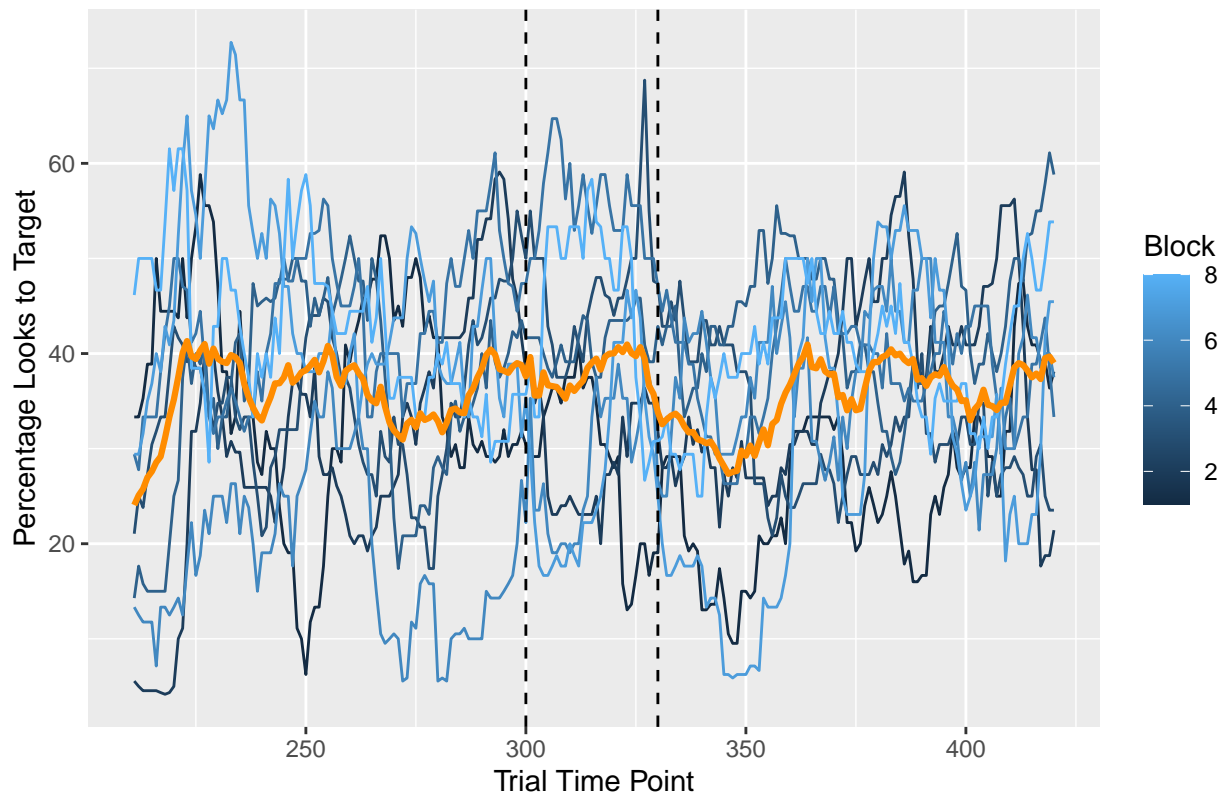


```
data_2 %>%
  ggplot( aes(x=blockG, y=dat.percent_looks,group=grouping8, color=grouping8)) +
  geom_line() +
  geom_vline(xintercept = 300, colour="black", linetype = "dashed") +
  geom_vline(xintercept = 330, colour="black", linetype = "dashed") +
  stat_summary(aes(y =dat.percent_looks,group=1), fun.y=mean, size=1.1, colour="darkorange", geom="line")

  labs ( x= "Trial Time Point",
        y= "Percentage Looks to Target",
        colour= "Block",
        title= "Line Graph of % Looks at Target in Trial B (Low Informative) + mean line" )
```

Warning: 'fun.y' is deprecated. Use 'fun' instead.

Line Graph of % Looks at Target in Trial B (Low Informative) + mean line

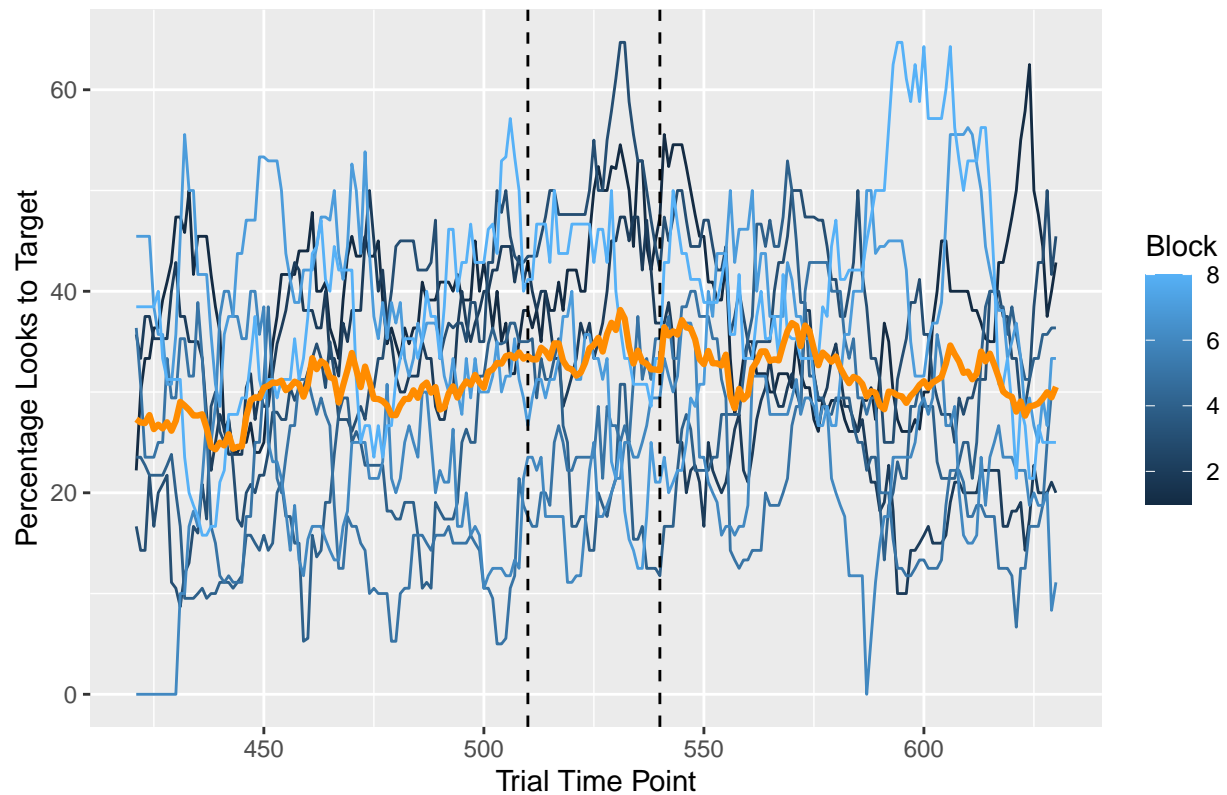


```
data_3 %>%
  ggplot( aes(x=blockG, y=dat.percent_looks,group=grouping8, color=grouping8)) +
  geom_line() +
  geom_vline(xintercept = 510, colour="black", linetype = "dashed") +
  geom_vline(xintercept = 540, colour="black", linetype = "dashed") +
  stat_summary(aes(y =dat.percent_looks,group=1), fun.y=mean, size=1.1, colour="darkorange", geom="line")

  labs ( x= "Trial Time Point",
        y= "Percentage Looks to Target",
        colour= "Block",
        title= "Line Graph of % Looks at Target in Trial C (Low Informative) + mean line" )
```

Warning: 'fun.y' is deprecated. Use 'fun' instead.

Line Graph of % Looks at Target in Trial C (Low Informative) + mean line

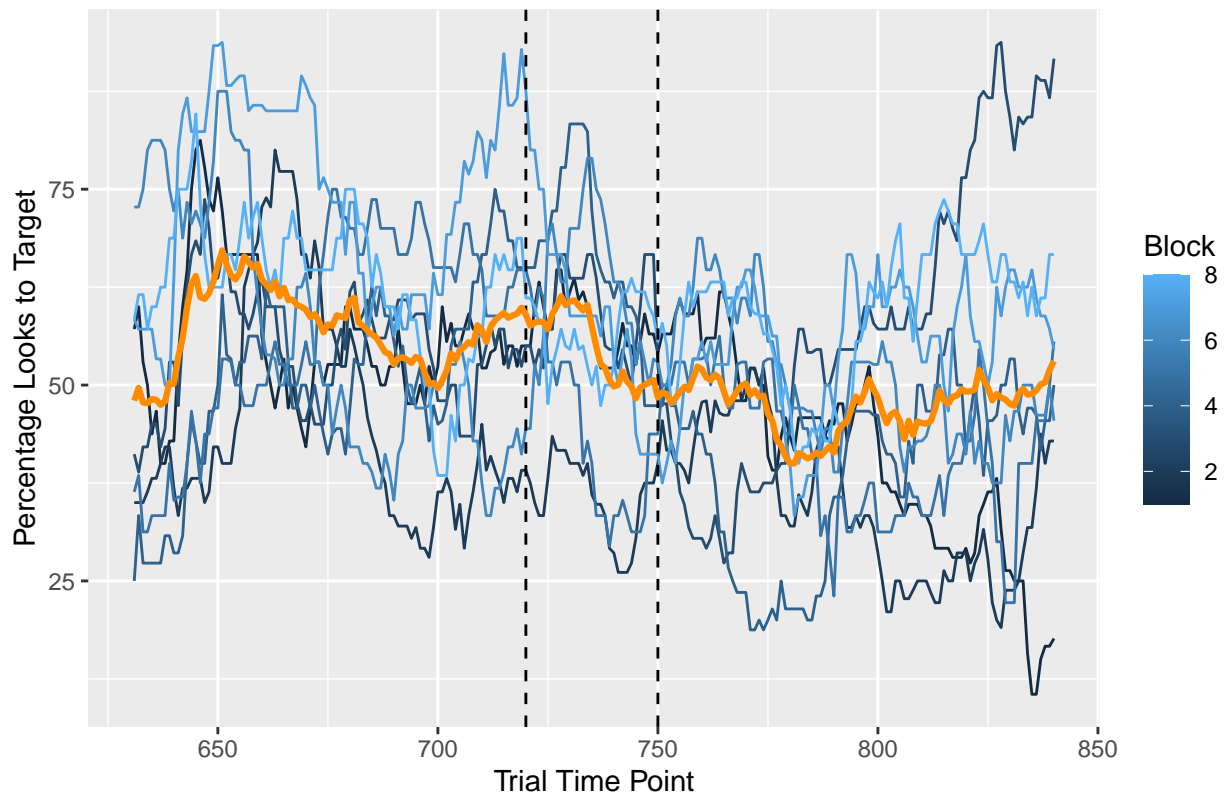


```
data_4 %>%
  ggplot( aes(x=blockG, y=dat.percent_looks,group=grouping8, color=grouping8)) +
  geom_line() +
  geom_vline(xintercept = 720, colour="black", linetype = "dashed") +
  geom_vline(xintercept = 750, colour="black", linetype = "dashed") +
  stat_summary(aes(y =dat.percent_looks,group=1), fun.y=mean, size=1.1, colour="darkorange", geom="line")

  labs ( x= "Trial Time Point",
        y= "Percentage Looks to Target",
        colour= "Block",
        title= "Line Graph of % Looks at Target in Trial D (High Informative) + mean line" )
```

Warning: 'fun.y' is deprecated. Use 'fun' instead.

Line Graph of % Looks at Target in Trial D (High Informative) + mean line

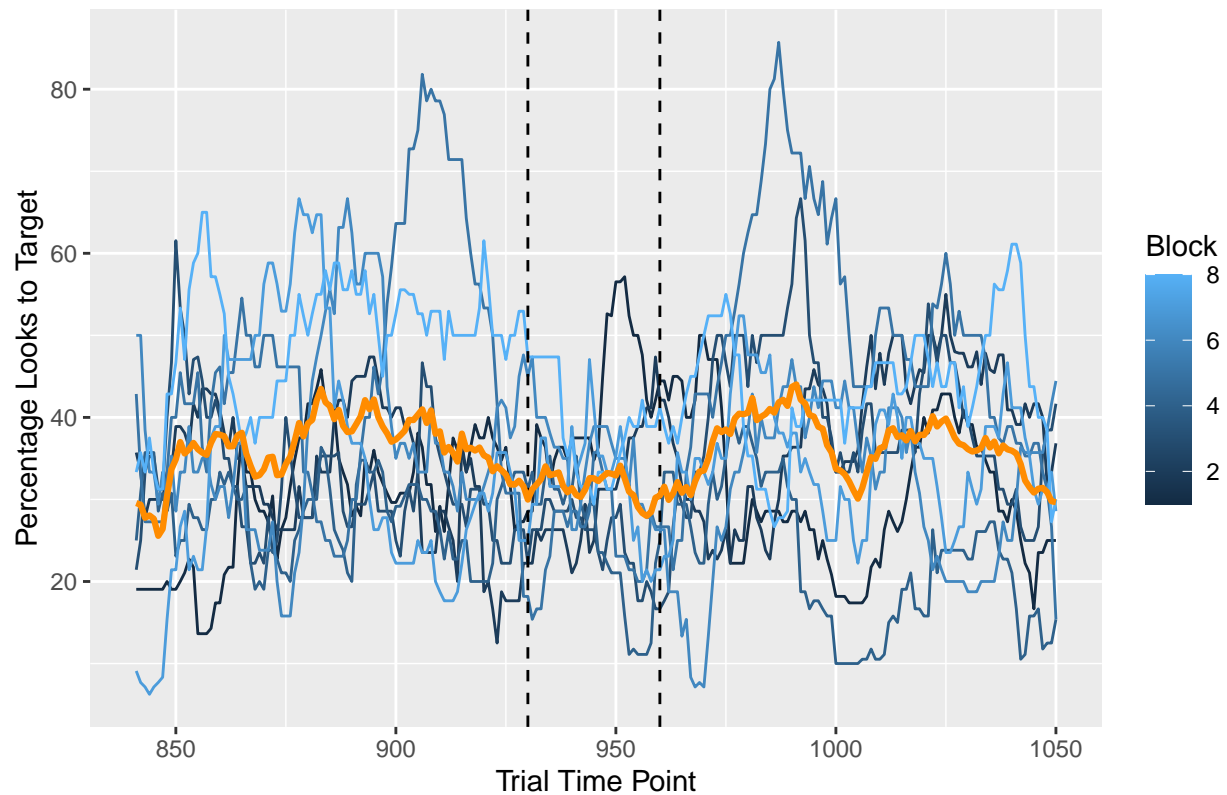


```
data_5 %>%
  ggplot( aes(x=blockG, y=dat.percent_looks,group=grouping8, color=grouping8)) +
  geom_line() +
  geom_vline(xintercept = 930, colour="black", linetype = "dashed") +
  geom_vline(xintercept = 960, colour="black", linetype = "dashed") +
  stat_summary(aes(y =dat.percent_looks,group=1), fun.y=mean, size=1.1, colour="darkorange", geom="line")

  labs ( x= "Trial Time Point",
        y= "Percentage Looks to Target",
        colour= "Block",
        title= "Line Graph of % Looks at Target in Trial E (Low Informative) + mean line" )
```

Warning: 'fun.y' is deprecated. Use 'fun' instead.

Line Graph of % Looks at Target in Trial E (Low Informative) + mean line

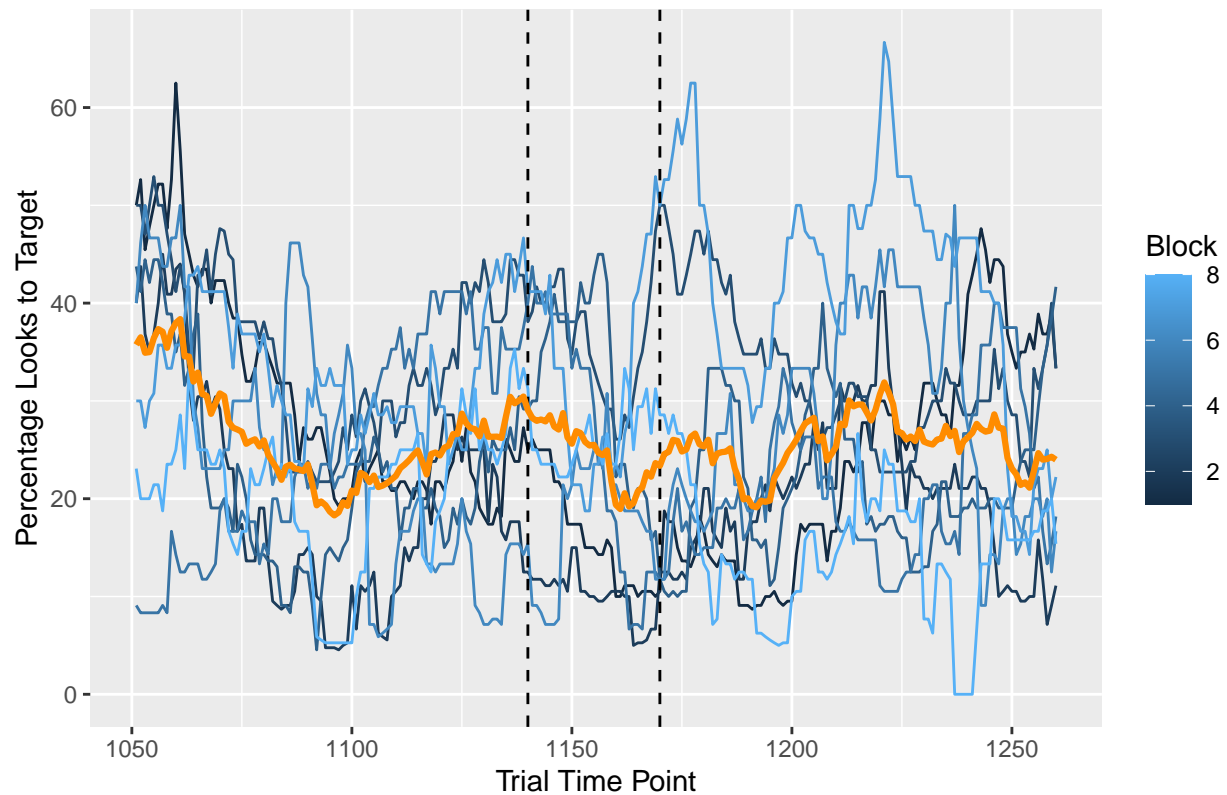


```
data_6 %>%
  ggplot( aes(x=blockG, y=dat.percent_looks,group=grouping8, color=grouping8)) +
  geom_line() +
  geom_vline(xintercept = 1140, colour="black", linetype = "dashed") +
  geom_vline(xintercept = 1170, colour="black", linetype = "dashed") +
  stat_summary(aes(y =dat.percent_looks,group=1), fun.y=mean, size=1.1, colour="darkorange", geom="line")

  labs ( x= "Trial Time Point",
        y= "Percentage Looks to Target",
        colour= "Block",
        title= "Line Graph of % Looks at Target in Trial F (Low Informative) + mean line" )
```

```
## Warning: 'fun.y' is deprecated. Use 'fun' instead.
```


Line Graph of % Looks at Target in Trial F (Low Informative) + mean line

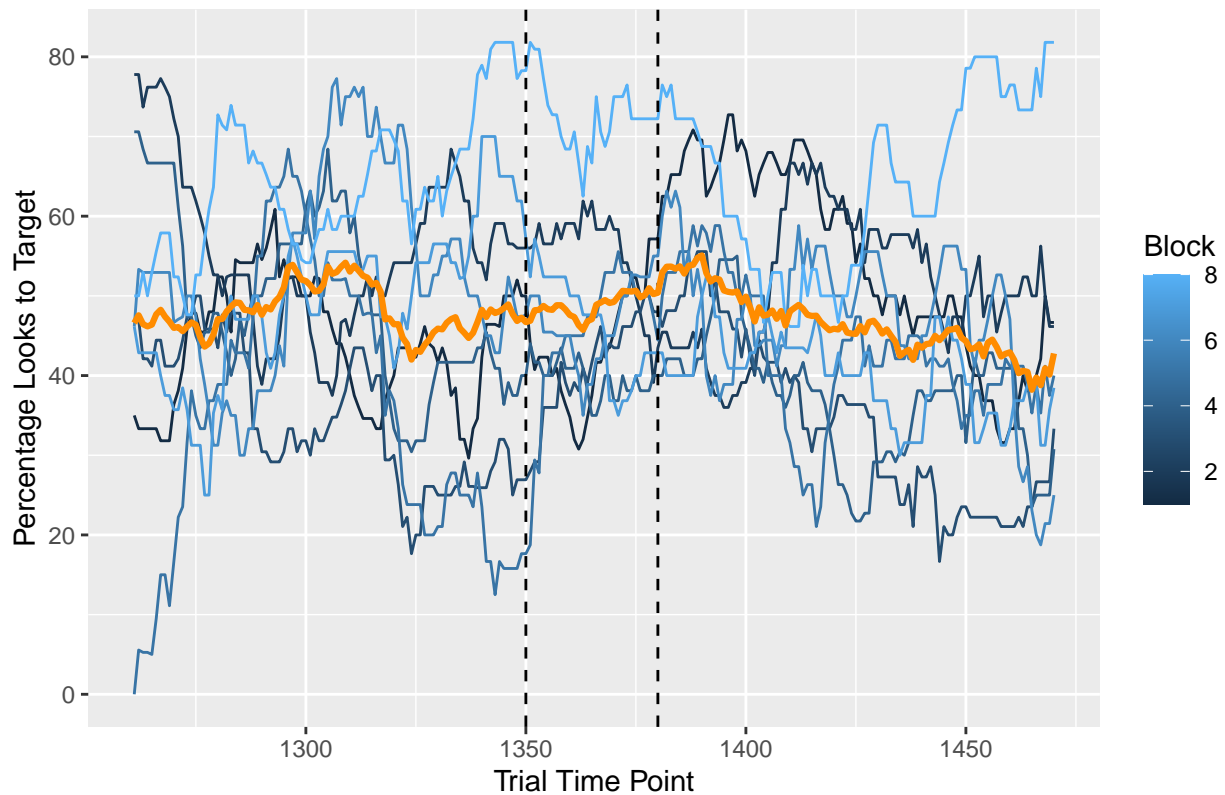


```
data_7 %>%
  ggplot( aes(x=blockG, y=dat.percent_looks,group=grouping8, color=grouping8)) +
  geom_line() +
  geom_vline(xintercept = 1130, colour="black", linetype = "dashed") +
  geom_vline(xintercept = 1170, colour="black", linetype = "dashed") +
  stat_summary(aes(y =dat.percent_looks,group=1), fun.y=mean, size=1.1, colour="darkorange", geom="line")

  labs ( x= "Trial Time Point",
        y= "Percentage Looks to Target",
        colour= "Block",
        title= "Line Graph of % Looks at Target in Trial G (Testing 1x2 Objects) + mean line" )
```

```
## Warning: 'fun.y' is deprecated. Use 'fun' instead.
```

Line Graph of % Looks at Target in Trial G (Testing 1x2 Objects) + mean line



```
data_8 %>%
  ggplot( aes(x=blockG, y=dat.percent_looks,group=grouping8, color=grouping8)) +
  geom_line() +
  geom_vline(xintercept =1560, colour="black", linetype = "dashed") +
  geom_vline(xintercept = 1590, colour="black", linetype = "dashed") +
  stat_summary(aes(y =dat.percent_looks,group=1), fun.y=mean, size=1.1, colour="darkorange", geom="line")

  labs ( x= "Trial Time Point",
        y= "Percentage Looks to Target",
        colour= "Block",
        title= "Line Graph of % Looks at Target in Trial H (Testing 2x2 Objects) + mean line" )
```

```
## Warning: 'fun.y' is deprecated. Use 'fun' instead.
```

Line Graph of % Looks at Target in Trial H (Testing 2x2 Objects) + mean line

