FHL\_code\_draft1

Jasmine Reighard

2023-09-20

#download data  
  
#all data  
cym\_dat <- read.csv("data/rawdata\_allcombinedSep28.csv")  
  
#NND data  
nnd\_df <- read.csv("data/rawdata\_combined\_NND\_9.28.csv")  
  
#spontaneous turns  
spon\_turn <- read.csv("data/sp\_turn\_10-9-23.csv")  
  
#categorize by s and w  
cym\_dat$s\_w\_reactor <- ifelse(substr(cym\_dat$responder, 1, 1) == "s", "s", "w")  
  
  
wave\_dat <- subset(cym\_dat, substr(responder, 1, 1) == "w")  
  
#make sure variables are as they are supposed to be  
  
cym\_dat$s\_w\_reactor <- as.factor(cym\_dat$s\_w\_reactor)  
cym\_dat$size <- as.factor(cym\_dat$size)  
cym\_dat$school <- as.factor(cym\_dat$school)  
cym\_dat$stimulus <- as.factor(cym\_dat$stimulus)

#Define thresholds for fast and slow responses for small and large  
  
#look at averages  
turnrate\_result <- aggregate(turning\_rate ~ size, data = spon\_turn, FUN = mean)  
print(turnrate\_result)

## size turning\_rate  
## 1 large 0.2243052  
## 2 small 0.2435468

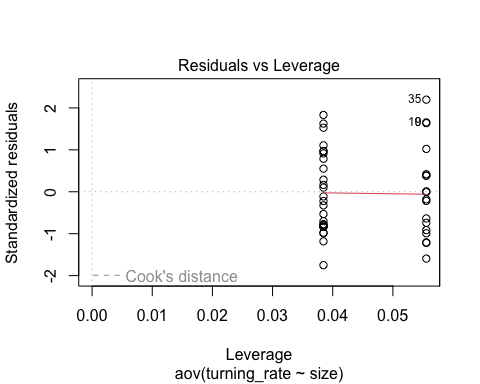
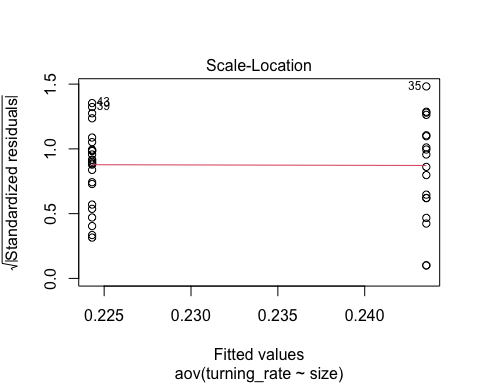
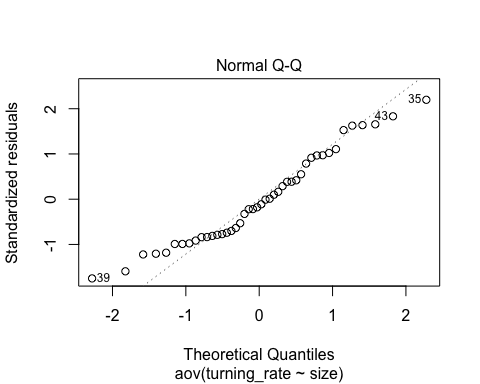
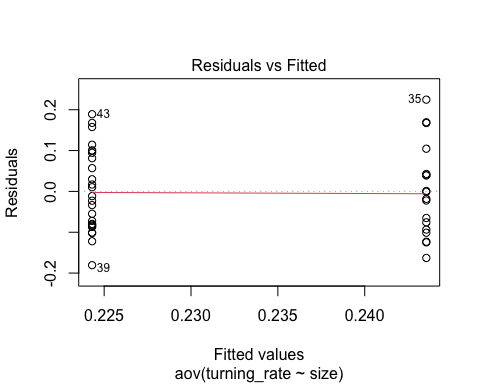
#large 0.2243052  
#small 0.243546  
  
#Mann-Whitney test  
manwhit\_spon <- wilcox.test(turning\_rate ~ size, data = spon\_turn)  
print(manwhit\_spon)

##   
## Wilcoxon rank sum exact test  
##   
## data: turning\_rate by size  
## W = 213, p-value = 0.6281  
## alternative hypothesis: true location shift is not equal to 0

# p-value is less not less than 0.05, conclusion: there is no statistically significant difference in the distributions of "turning\_rate" between the two size.  
  
  
spon\_anova <- aov(turning\_rate ~ size, data = spon\_turn)  
print(summary(spon\_anova))

## Df Sum Sq Mean Sq F value Pr(>F)  
## size 1 0.0039 0.003938 0.356 0.554  
## Residuals 42 0.4649 0.011068

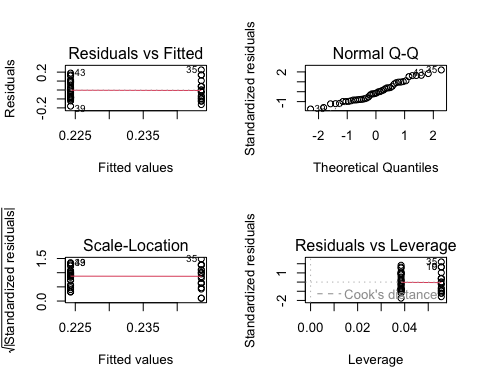
plot(spon\_anova)



shapiro.test(spon\_anova$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: spon\_anova$residuals  
## W = 0.95688, p-value = 0.09945

par(mfrow=c(2,2))  
plot(spon\_anova)



#normal and homogeneity pass  
  
  
#homogeneity of variances  
leveneTest(turning\_rate ~ size, data = spon\_turn)

## Warning in leveneTest.default(y = y, group = group, ...): group coerced to  
## factor.

## Levene's Test for Homogeneity of Variance (center = median)  
## Df F value Pr(>F)  
## group 1 0.0169 0.8972  
## 42

t.test(turning\_rate ~ size, data = spon\_turn)

##   
## Welch Two Sample t-test  
##   
## data: turning\_rate by size  
## t = -0.58589, df = 34.338, p-value = 0.5618  
## alternative hypothesis: true difference in means between group large and group small is not equal to 0  
## 95 percent confidence interval:  
## -0.08596025 0.04747703  
## sample estimates:  
## mean in group large mean in group small   
## 0.2243052 0.2435468

avg\_turnrate <- mean(spon\_turn$turning\_rate)  
print(avg\_turnrate)

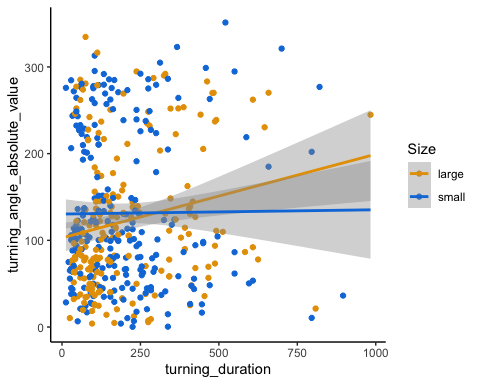
## [1] 0.2321768

#avg turn rate is .2321768

#calculate turning duration on all data  
# Calculate density   
  
cym\_dat\_frames <- read.csv("data/rawdata\_allcombinedSep28\_start\_end.csv")  
cym\_dat\_frames <- cym\_dat\_frames %>%   
 mutate(turning\_duration = (frame\_end - frame\_st)/240 \* 1000)  
  
#add turning duration to cym\_dat df  
cym\_dat$turning\_duration <- cym\_dat\_frames$turning\_duration  
  
#calculate turning rate  
cym\_dat <- cym\_dat %>%   
 mutate(turning\_rate = turning\_angle\_absolute\_value / turning\_duration)  
  
#categorize responses  
  
cym\_dat$response\_type <- ifelse(  
 (cym\_dat$size == "large" & cym\_dat$turning\_rate > 0.22) |  
 (cym\_dat$size == "small" & cym\_dat$turning\_rate > 0.24),  
 "fast",  
 "slow"  
)  
  
#make sure variables are as they are supposed to be  
  
cym\_dat$s\_w\_reactor <- as.factor(cym\_dat$s\_w\_reactor)  
cym\_dat$size <- as.factor(cym\_dat$size)  
cym\_dat$school <- as.factor(cym\_dat$school)  
cym\_dat$stimulus <- as.factor(cym\_dat$stimulus)  
cym\_dat$response\_type <- as.factor(cym\_dat$response\_type)

#visualize data fast and slow responses  
  
#all together  
ggplot(cym\_dat,aes(x=turning\_duration,y=turning\_angle\_absolute\_value,color=size))+  
 geom\_point()+  
 geom\_smooth(method=lm,fullrange=TRUE,  
 aes(color=size))+  
 scale\_color\_manual(name='Size',  
 breaks=c('large', 'small'),  
 values=c('large'='#E69F00', 'small'='#0C7BDC'))+  
 theme(legend.title=element\_text(size=14),  
 legend.text=element\_text(size=14))+  
 theme\_classic()

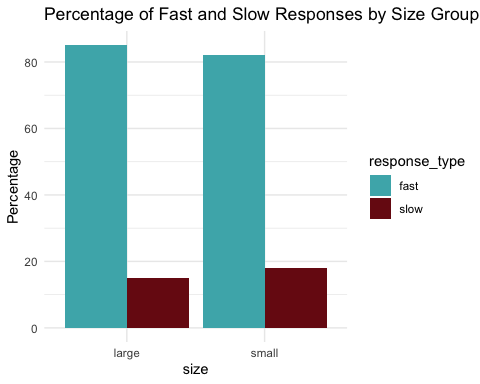
## `geom\_smooth()` using formula = 'y ~ x'



#visualize percent slow  
summary\_table <- cym\_dat %>%  
 group\_by(size, response\_type) %>%  
 summarise(count = n()) %>%  
 mutate(percentage = count / sum(count) \* 100)

## `summarise()` has grouped output by 'size'. You can override using the  
## `.groups` argument.

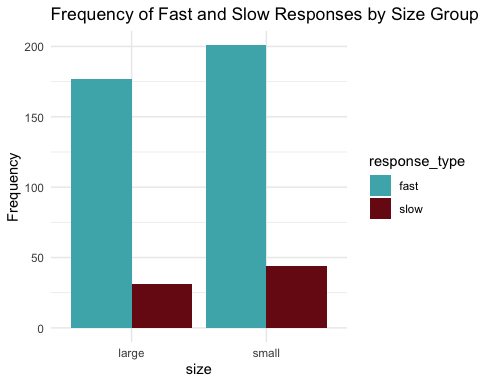
ggplot(summary\_table, aes(x = size, y = percentage, fill = response\_type)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(title = "Percentage of Fast and Slow Responses by Size Group", y = "Percentage") +  
 scale\_fill\_fish\_d(option = "Trimma\_lantana") +  
 theme\_minimal()



#frequency of fast and slow  
summary\_table <- cym\_dat %>%  
 group\_by(size, response\_type) %>%  
 summarise(count = n())

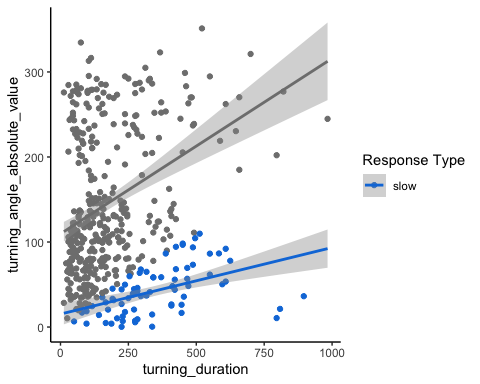
## `summarise()` has grouped output by 'size'. You can override using the  
## `.groups` argument.

ggplot(summary\_table, aes(x = size, y = count, fill = response\_type)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(title = "Frequency of Fast and Slow Responses by Size Group", y = "Frequency") +  
 scale\_fill\_fish\_d(option = "Trimma\_lantana") +  
 theme\_minimal()



#scatter plot grouped by fast and slow  
  
ggplot(cym\_dat,aes(x=turning\_duration,y=turning\_angle\_absolute\_value,color=response\_type))+  
 geom\_point()+  
 geom\_smooth(method=lm,fullrange=TRUE,  
 aes(color=response\_type))+  
 scale\_color\_manual(name='Response Type',  
 breaks=c('fast', 'slow'),  
 values=c('large'='#E69F00', 'slow'='#0C7BDC'))+  
 theme(legend.title=element\_text(size=14),  
 legend.text=element\_text(size=14))+  
 theme\_classic()

## `geom\_smooth()` using formula = 'y ~ x'



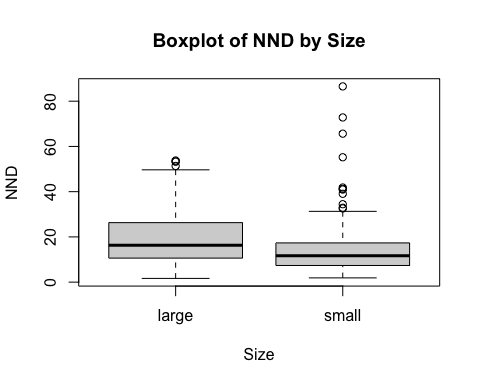
# Calculating NND for each responder within the same school and stimulus group  
#NAs come from two having the same nearest neighbor distance  
  
nnd\_df <- nnd\_df %>%  
 group\_by(school, stimulus) %>%  
 mutate(  
 NND = sqrt((s\_head\_x - lag(s\_head\_x))^2 + (s\_head\_y - lag(s\_head\_y))^2)  
 ) %>%  
 ungroup()  
  
#If we want:  
# Replace NA values in the NND column with 0   
#nnd\_df$NND[is.na(nnd\_df$NND)] <- 0  
  
cym\_dat$NND <- nnd\_df$NND  
  
#add nnd to wave df  
nnd\_wave\_dat <- merge(wave\_dat, nnd\_df[, c('school', 'stimulus', 'responder', 'NND')],   
 by = c('school', 'stimulus', 'responder'),   
 all.x = TRUE)  
  
#average NND   
avgNND\_result <- aggregate(NND ~ size, data = cym\_dat, FUN = function(x) mean(x, na.rm = TRUE))  
print(avgNND\_result)

## size NND  
## 1 large 19.66457  
## 2 small 13.98682

manwhit\_NND <- wilcox.test(NND ~ size, data = cym\_dat)  
print(manwhit\_NND)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: NND by size  
## W = 31680, p-value = 1.444e-09  
## alternative hypothesis: true location shift is not equal to 0

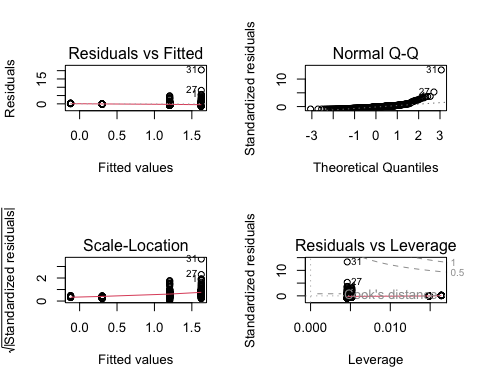
#they are significantly different from each other and further apart!   
  
boxplot(NND ~ size, data = cym\_dat, main = "Boxplot of NND by Size",  
 xlab = "Size", ylab = "NND")



#ANCOVA  
ancova\_mod <- aov(turning\_rate ~ size + response\_type, data = cym\_dat)  
summary(ancova\_mod)

## Df Sum Sq Mean Sq F value Pr(>F)   
## size 1 16.4 16.42 6.93 0.00877 \*\*   
## response\_type 1 109.7 109.68 46.28 3.27e-11 \*\*\*  
## Residuals 450 1066.4 2.37   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

par(mfrow=c(2,2))  
plot(ancova\_mod)



wilcox.test(turning\_rate ~ response\_type, data = cym\_dat)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: turning\_rate by response\_type  
## W = 28341, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

# p-value <2.2e-16, conclusion: there is a statistically significant difference in the distributions of "turning\_rate" between the two response types.

#Only interested in fast responses, making new dataset excluding slow responses  
  
fast\_dat <- cym\_dat %>%  
 filter(response\_type == "fast")  
  
#glm for fast  
mod\_fast <- glm(latency\_ms ~ s\_w\_reactor + distance\_from\_stimulus + size+ angle\_between\_fish\_and\_stimulus + distance\_from\_first\_responder +NND, data = fast\_dat, family=poisson, na.action = na.exclude)

## Warning in dpois(y, mu, log = TRUE): non-integer x = 4.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 12.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 20.800000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 20.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 41.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 16.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 54.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 41.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 112.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 104.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 108.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 141.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 187.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 179.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 216.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 195.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 220.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 8.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 20.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 33.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 29.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 41.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 83.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 70.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 104.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 112.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 91.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 195.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 191.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 241.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 308.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 237.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 483.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 987.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 379.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 233.300000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 233.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 291.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 308.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 433.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 504.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 295.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 133.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 287.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 420.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 458.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 470.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 508.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 658.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 495.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 8.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 37.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 41.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 37.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 45.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 66.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 154.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 62.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 112.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 154.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 183.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 195.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 266.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 33.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 58.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 133.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 154.200000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 154.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 170.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 241.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 170.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 233.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 170.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 212.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 229.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 233.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 383.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 12.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 16.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 45.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 66.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 87.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 95.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 133.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 141.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 137.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 141.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 158.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 204.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 320.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 95.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 270.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 233.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 241.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 420.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 416.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 516.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 29.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 33.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 91.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 341.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 395.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 166.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 212.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 245.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 262.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 283.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 41.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 45.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 116.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 145.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 233.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 583.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 170.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 204.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 233.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 237.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 241.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 337.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 58.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 87.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 116.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 458.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 112.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 187.500000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 187.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 258.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 262.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 12.500000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 12.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 45.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 62.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 91.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 162.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 120.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 133.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 154.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 162.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 233.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 287.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 366.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 370.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 379.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 391.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 404.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 29.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 37.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 66.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 91.700000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 91.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 166.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 179.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 216.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 229.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 258.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 329.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 441.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 483.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 8.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 20.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 170.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 179.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 216.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 304.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 312.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 341.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 512.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 591.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1087.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 91.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 104.200000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 104.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 162.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 516.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 283.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 362.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 383.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 445.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 454.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 458.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 41.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 79.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 141.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1391.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 91.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 216.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 233.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 308.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 612.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 429.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 262.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 8.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 20.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 137.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 145.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 204.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 220.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 329.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 404.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1366.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 283.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 316.700000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 316.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 345.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 570.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 695.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 720.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 29.200000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 29.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 308.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 220.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 391.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 512.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 566.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1070.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 12.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 104.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 329.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1679.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 12.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 33.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 37.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 41.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 137.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 162.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 187.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 220.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 395.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 441.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 504.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 537.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 558.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1295.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 62.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 91.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 116.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 195.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 329.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 158.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 212.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 262.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 304.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 479.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 33.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 41.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 54.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 116.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 637.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 383.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 487.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1241.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 1383.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 37.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 45.800000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 45.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 62.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 70.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 87.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 120.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 129.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 141.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 162.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 204.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 212.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 283.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 391.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 395.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 404.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 566.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 29.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 83.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 245.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 383.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 395.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 262.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 287.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 308.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 358.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 366.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 429.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 54.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 70.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 95.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 116.700000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 179.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 212.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 179.200000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 183.300000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 287.500000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 345.800000  
  
## Warning in dpois(y, mu, log = TRUE): non-integer x = 345.800000

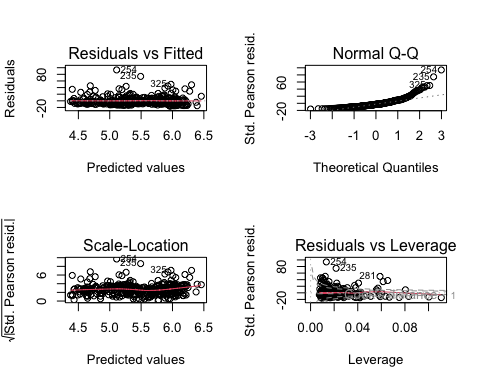
## Warning in dpois(y, mu, log = TRUE): non-integer x = 370.800000

## Warning in dpois(y, mu, log = TRUE): non-integer x = 645.800000

summary(mod\_fast)

##   
## Call:  
## glm(formula = latency\_ms ~ s\_w\_reactor + distance\_from\_stimulus +   
## size + angle\_between\_fish\_and\_stimulus + distance\_from\_first\_responder +   
## NND, family = poisson, data = fast\_dat, na.action = na.exclude)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -19.775 -9.307 -3.078 3.434 58.017   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 4.159e+00 1.507e-02 275.930 < 2e-16 \*\*\*  
## s\_w\_reactorw 5.642e-01 8.873e-03 63.583 < 2e-16 \*\*\*  
## distance\_from\_stimulus 1.732e-02 3.357e-04 51.610 < 2e-16 \*\*\*  
## sizesmall 5.673e-01 7.711e-03 73.568 < 2e-16 \*\*\*  
## angle\_between\_fish\_and\_stimulus -3.794e-04 4.777e-05 -7.942 1.98e-15 \*\*\*  
## distance\_from\_first\_responder 3.501e-03 2.616e-04 13.380 < 2e-16 \*\*\*  
## NND -5.084e-04 3.109e-04 -1.635 0.102   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for poisson family taken to be 1)  
##   
## Null deviance: 68771 on 361 degrees of freedom  
## Residual deviance: 48263 on 355 degrees of freedom  
## (16 observations deleted due to missingness)  
## AIC: Inf  
##   
## Number of Fisher Scoring iterations: 5

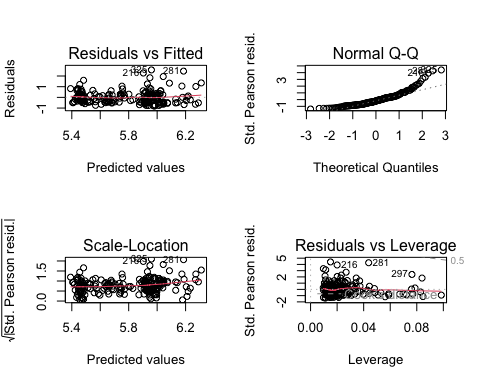
par(mfrow=c(2,2))  
plot(mod\_fast)



#subsetting just the wave  
wave\_dat\_fast <- subset(fast\_dat, substr(responder, 1, 1) == "w")  
  
#glm for wave  
glm\_wave\_fast <- glm(latency\_ms ~ size+ angle\_between\_fish\_and\_stimulus+distance\_from\_first\_responder +NND, data = wave\_dat\_fast, family=Gamma(link = "log"), na.action = na.exclude)  
summary(glm\_wave\_fast)

##   
## Call:  
## glm(formula = latency\_ms ~ size + angle\_between\_fish\_and\_stimulus +   
## distance\_from\_first\_responder + NND, family = Gamma(link = "log"),   
## data = wave\_dat\_fast, na.action = na.exclude)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3409 -0.4967 -0.1336 0.2082 1.6107   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.4347868 0.1305232 41.638 < 2e-16 \*\*\*  
## sizesmall 0.4656562 0.0901161 5.167 5.64e-07 \*\*\*  
## angle\_between\_fish\_and\_stimulus -0.0008496 0.0006864 -1.238 0.21723   
## distance\_from\_first\_responder 0.0085481 0.0031592 2.706 0.00739 \*\*   
## NND -0.0016675 0.0041097 -0.406 0.68536   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Gamma family taken to be 0.3461998)  
##   
## Null deviance: 75.546 on 208 degrees of freedom  
## Residual deviance: 62.990 on 204 degrees of freedom  
## AIC: 2726.9  
##   
## Number of Fisher Scoring iterations: 6

par(mfrow=c(2,2))  
plot(glm\_wave\_fast)



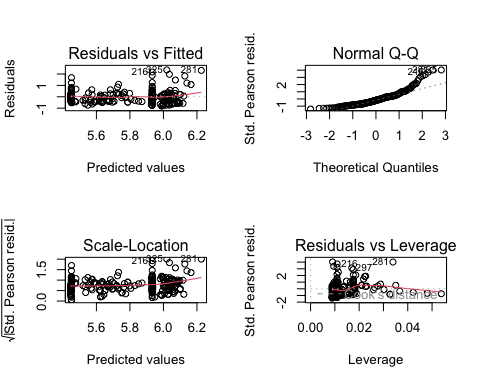
with(summary(glm\_wave\_fast), 1 - deviance/null.deviance)

## [1] 0.1661971

#taking out all insignificant variables  
glm\_size\_distfr <- glm(latency\_ms ~ size + distance\_from\_first\_responder , data = wave\_dat\_fast, family=Gamma(link = "log"))  
summary(glm\_size\_distfr)

##   
## Call:  
## glm(formula = latency\_ms ~ size + distance\_from\_first\_responder,   
## family = Gamma(link = "log"), data = wave\_dat\_fast)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3469 -0.5065 -0.1143 0.2011 1.5153   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.371731 0.101721 52.809 < 2e-16 \*\*\*  
## sizesmall 0.482675 0.084874 5.687 4.39e-08 \*\*\*  
## distance\_from\_first\_responder 0.007346 0.003004 2.446 0.0153 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Gamma family taken to be 0.3458747)  
##   
## Null deviance: 75.546 on 208 degrees of freedom  
## Residual deviance: 63.585 on 206 degrees of freedom  
## AIC: 2725  
##   
## Number of Fisher Scoring iterations: 6

par(mfrow=c(2,2))  
plot(glm\_size\_distfr)



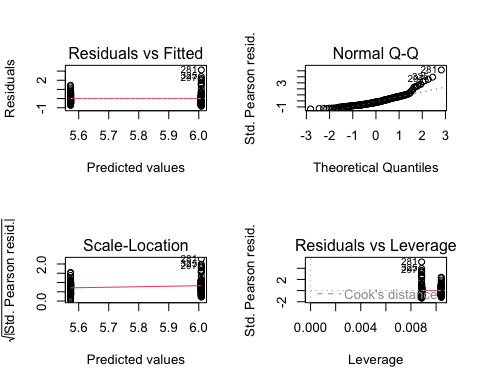
#size is more significant  
  
#calculate McFadden's R-squared for model for size and distance from first responder glm  
with(summary(glm\_size\_distfr), 1 - deviance/null.deviance)

## [1] 0.1583209

glm\_size <- glm(latency\_ms ~ size, data = wave\_dat\_fast, family=Gamma(link = "log"))  
summary(glm\_size)

##   
## Call:  
## glm(formula = latency\_ms ~ size, family = Gamma(link = "log"),   
## data = wave\_dat\_fast)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.3583 -0.4910 -0.1352 0.2313 1.8493   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.57289 0.06235 89.384 < 2e-16 \*\*\*  
## sizesmall 0.43549 0.08479 5.136 6.46e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for Gamma family taken to be 0.3731722)  
##   
## Null deviance: 75.546 on 208 degrees of freedom  
## Residual deviance: 65.891 on 207 degrees of freedom  
## AIC: 2730.8  
##   
## Number of Fisher Scoring iterations: 5

par(mfrow=c(2,2))  
plot(glm\_size)



with(summary(glm\_size), 1 - deviance/null.deviance)

## [1] 0.1277931

#linear model for just the wave  
lm\_wavedat\_fast <- lm(latency\_ms ~ size, data=wave\_dat\_fast)  
summary(lm\_wavedat\_fast)

##   
## Call:  
## lm(formula = latency\_ms ~ size, data = wave\_dat\_fast)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -336.02 -125.69 -38.19 82.61 1272.38   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 263.19 23.46 11.220 < 2e-16 \*\*\*  
## sizesmall 143.63 31.90 4.502 1.12e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 229.8 on 207 degrees of freedom  
## Multiple R-squared: 0.08918, Adjusted R-squared: 0.08478   
## F-statistic: 20.27 on 1 and 207 DF, p-value: 1.123e-05

#normality  
shapiro.test(lm\_wavedat\_fast$residuals)

##   
## Shapiro-Wilk normality test  
##   
## data: lm\_wavedat\_fast$residuals  
## W = 0.81504, p-value = 5.023e-15

#homogeneity of variances  
bartlett.test(latency\_ms ~ size, data=wave\_dat\_fast)

##   
## Bartlett test of homogeneity of variances  
##   
## data: latency\_ms by size  
## Bartlett's K-squared = 71.483, df = 1, p-value < 2.2e-16

wilcox.test(latency\_ms ~ size, data = wave\_dat\_fast)

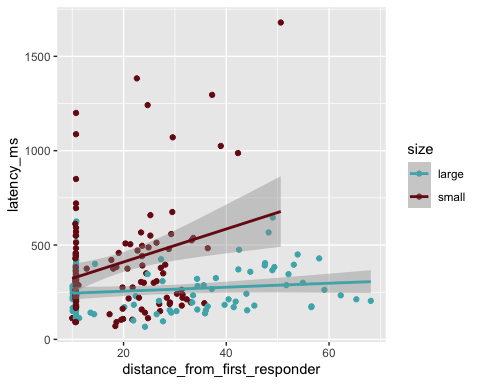
##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: latency\_ms by size  
## W = 3683.5, p-value = 6.5e-05  
## alternative hypothesis: true location shift is not equal to 0

#Correlation and Principal Component Analysis, not using this  
  
  
#new df with specified columns  
wave\_fast\_forPCA <- wave\_dat\_fast[, c("latency\_ms", "s\_w\_reactor", "distance\_from\_stimulus", "size", "angle\_between\_fish\_and\_stimulus", "distance\_from\_first\_responder","NND")]  
  
  
#make small 0 and large 1  
wave\_fast\_forPCA <- wave\_fast\_forPCA %>%  
 mutate(size\_ID = ifelse(size == "small", 0, 1))  
  
  
  
#use sapply to make wave\_dat\_fast numeric  
  
wave\_fast\_new <- wave\_fast\_forPCA[sapply(wave\_fast\_forPCA, is.numeric)]  
  
#correlation  
wave\_corr <- sapply(wave\_fast\_new[, setdiff(names(wave\_fast\_new), "latency\_ms")], function(x) {  
 cor(wave\_fast\_new$latency\_ms, x, method = "pearson")  
})  
print(wave\_corr)

## distance\_from\_stimulus angle\_between\_fish\_and\_stimulus   
## 0.05041053 -0.07227380   
## distance\_from\_first\_responder NND   
## 0.07761140 -0.11248255   
## size\_ID   
## -0.29863354

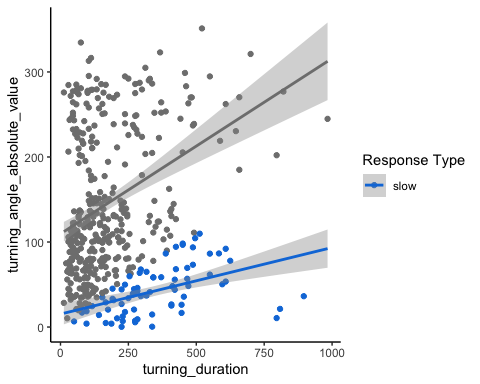
#quick visualization   
ggplot(wave\_dat\_fast) +  
 geom\_point(aes(x = distance\_from\_first\_responder, y = latency\_ms, color = size)) +  
 geom\_smooth(aes(x = distance\_from\_first\_responder, y = latency\_ms, color = size), method = "lm") +  
 scale\_color\_fish\_d(option = "Trimma\_lantana")

## `geom\_smooth()` using formula = 'y ~ x'

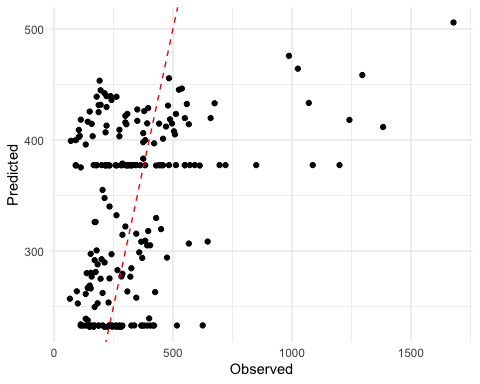


ggplot(cym\_dat,aes(x=turning\_duration,y=turning\_angle\_absolute\_value,color=response\_type))+  
 geom\_point()+  
 geom\_smooth(method=lm,fullrange=TRUE,  
 aes(color=response\_type))+  
 scale\_color\_manual(name='Response Type',  
 breaks=c('fast', 'slow'),  
 values=c('large'='#E69F00', 'slow'='#0C7BDC'))+  
 theme(legend.title=element\_text(size=14),  
 legend.text=element\_text(size=14))+  
 theme\_classic()

## `geom\_smooth()` using formula = 'y ~ x'

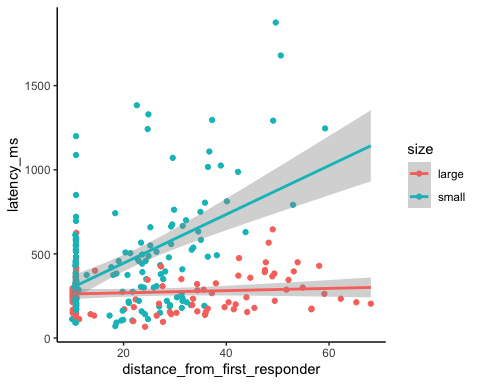


# Predict the values using the fitted model  
predicted\_values <- predict(glm\_size\_distfr, type = "response")  
  
# Create a data frame for plotting  
plot\_data <- data.frame(Observed = wave\_dat\_fast$latency\_ms, Predicted = predicted\_values)  
  
#scatterplot  
ggplot(plot\_data, aes(x = Observed, y = Predicted)) +  
 geom\_point() +  
 geom\_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") + # Add a reference line  
 labs(x = "Observed", y = "Predicted") +  
 theme\_minimal()

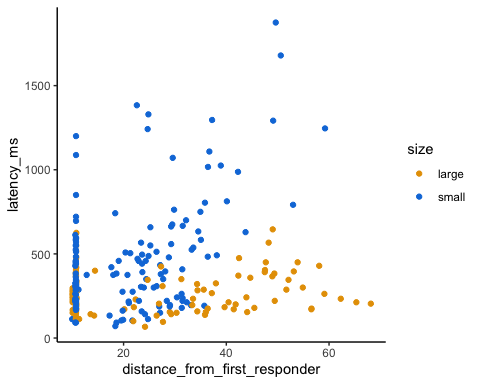


ggplt <- ggplot(wave\_dat,aes(x=distance\_from\_first\_responder,y=latency\_ms,color=size))+  
 geom\_point()+  
 theme\_classic()  
  
ggplt+geom\_smooth(method=lm,fullrange=TRUE,  
 aes(color=size))

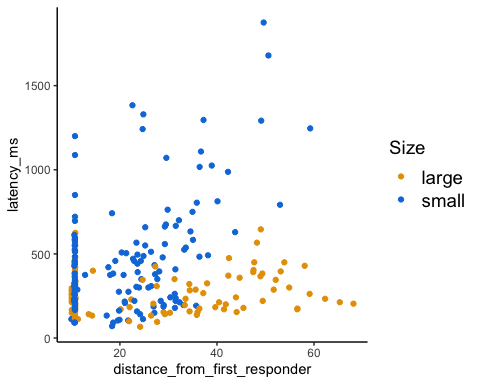
## `geom\_smooth()` using formula = 'y ~ x'



ggplt+scale\_color\_manual(values=c("#E69F00","#0C7BDC"))



ggplt+scale\_color\_manual(name='Size',  
 breaks=c('large', 'small'),  
 values=c('large'='#E69F00', 'small'='#0C7BDC'))+  
 theme(legend.title=element\_text(size=14),  
 legend.text=element\_text(size=14))



#plots   
  
ggplot(wave\_dat\_fast,aes(x=distance\_from\_first\_responder,y=latency\_ms,color=size))+  
 geom\_point()+  
 geom\_smooth(method=lm,fullrange=TRUE,  
 aes(color=size))+  
 scale\_color\_manual(name='Size',  
 breaks=c('large', 'small'),  
 values=c('large'='#E69F00', 'small'='#0C7BDC'))+  
 theme(legend.title=element\_text(size=14),  
 legend.text=element\_text(size=14))+  
 theme\_classic()

## `geom\_smooth()` using formula = 'y ~ x'

