Project Analysis

Maia Czerwonka

2024-02-27

Data analysis and visualizations for Maia Czerwonka's University of Washington Psychology Department Cognition and Cortical Dynamics Laboratory research project exploring the individual differences between people's information processing and attentional bias when presented with conflicting information in the card sorting task.

Data Initialization

```
data <- read.csv("MaiaData_CardSort.csv")
head(data)
```

```
##    Subject Age Sex IncorRespCount SymRespCount TxtRespCount  BiasScore
## 1 0156ce12  51   2              0            9           39  0.6250000
## 2 054ba968  87   2              0           20           28  0.1666667
## 3 05546b7f  60   2              5           12           31  0.4418605
## 4 08f746fa  38   2              2           39            7 -0.6956522
## 5 0aef8687  63   2              8            8           32  0.6000000
## 6 0b2a2504  34   2              4           17           27  0.2272727
##      Con_RT  InCon_RT Incon.ConRT
## 1  878.3542 1014.1875   135.83333
## 2 1192.7917 1273.8125    81.02083
## 3 1122.9306 1194.4583    71.52778
## 4 1013.8264 1029.7708    15.94444
## 5  990.1875  988.1875    -2.00000
## 6 1066.3611 1098.3750    32.01389
```

Descriptives of Dataset

```
data %>%
  select(-c(Subject, Sex)) %>%
  psych::describe()
```

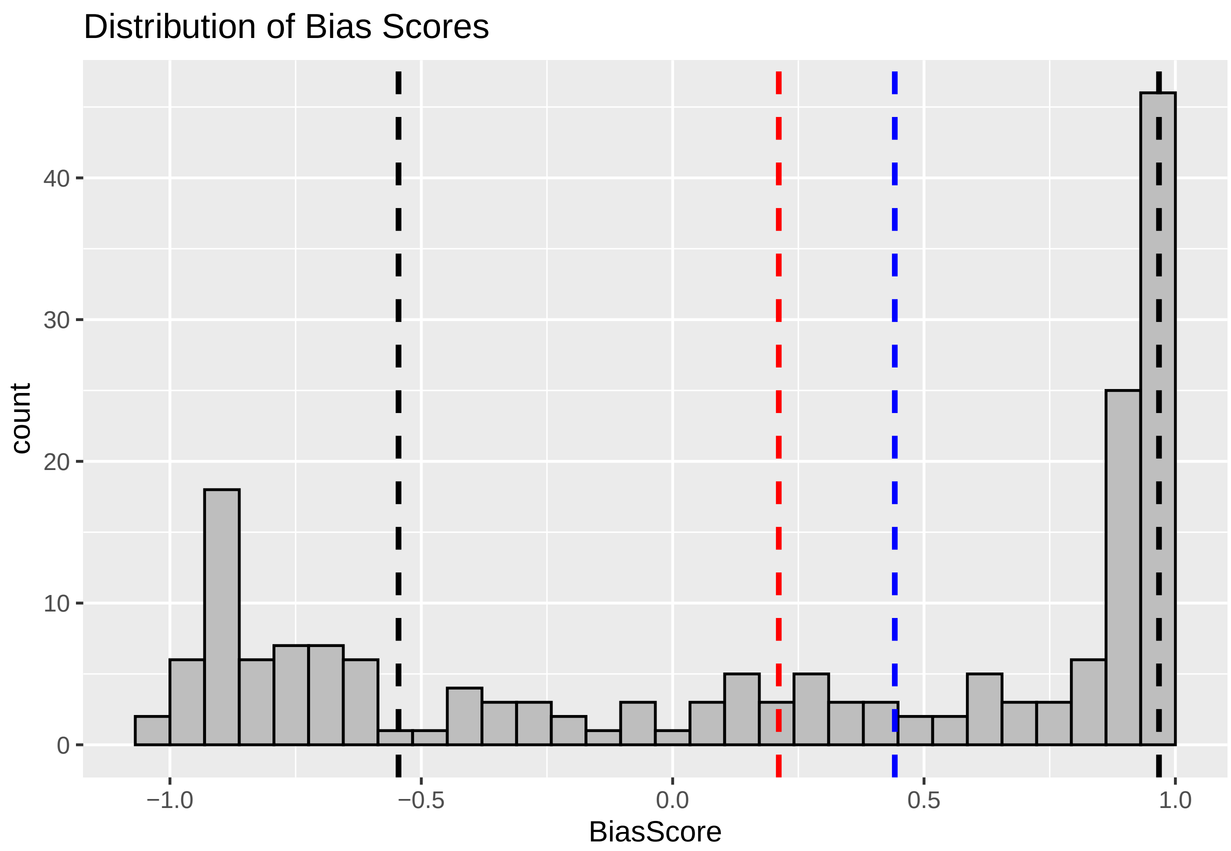
```
##                vars   n  mean    sd median trimmed   mad   min   max
## Age               1 183 40.05 14.52  38.00   39.22 13.34 10.00 87.00
## IncorRespCount    2 185  1.83  3.12   1.00    1.13  1.48  0.00 19.00
## SymRespCount      3 185 18.10 17.66  12.00   16.96 16.31  0.00 48.00
## TxtRespCount      4 185  28.07  17.95  33.00   28.91  20.76    0.00   48.00
## BiasScore         5 185   0.21   0.76   0.44    0.25   0.77   -1.00    1.00
## Con_RT            6 185 885.79 174.50 866.42  875.88 160.51  529.08 1579.72
## InCon_RT          7 185 987.53 325.00 925.81  946.21 209.20  528.29 2863.08
## Incon.ConRT       8 185 101.74 239.42  46.57   62.58  74.70 -116.09 1902.28
##                  range  skew kurtosis    se
## Age              77.00  0.55     0.20  1.07
## IncorRespCount   19.00  3.35    11.70  0.23
## SymRespCount     48.00  0.45    -1.49  1.30
## TxtRespCount     48.00 -0.31    -1.61  1.32
## BiasScore         2.00 -0.38    -1.55  0.06
## Con_RT         1050.64  0.63     0.59 12.83
## InCon_RT       2334.79  2.91    13.12 23.89
## Incon.ConRT    2018.37  5.34    33.94 17.60
```

Distributions of Bias Scores

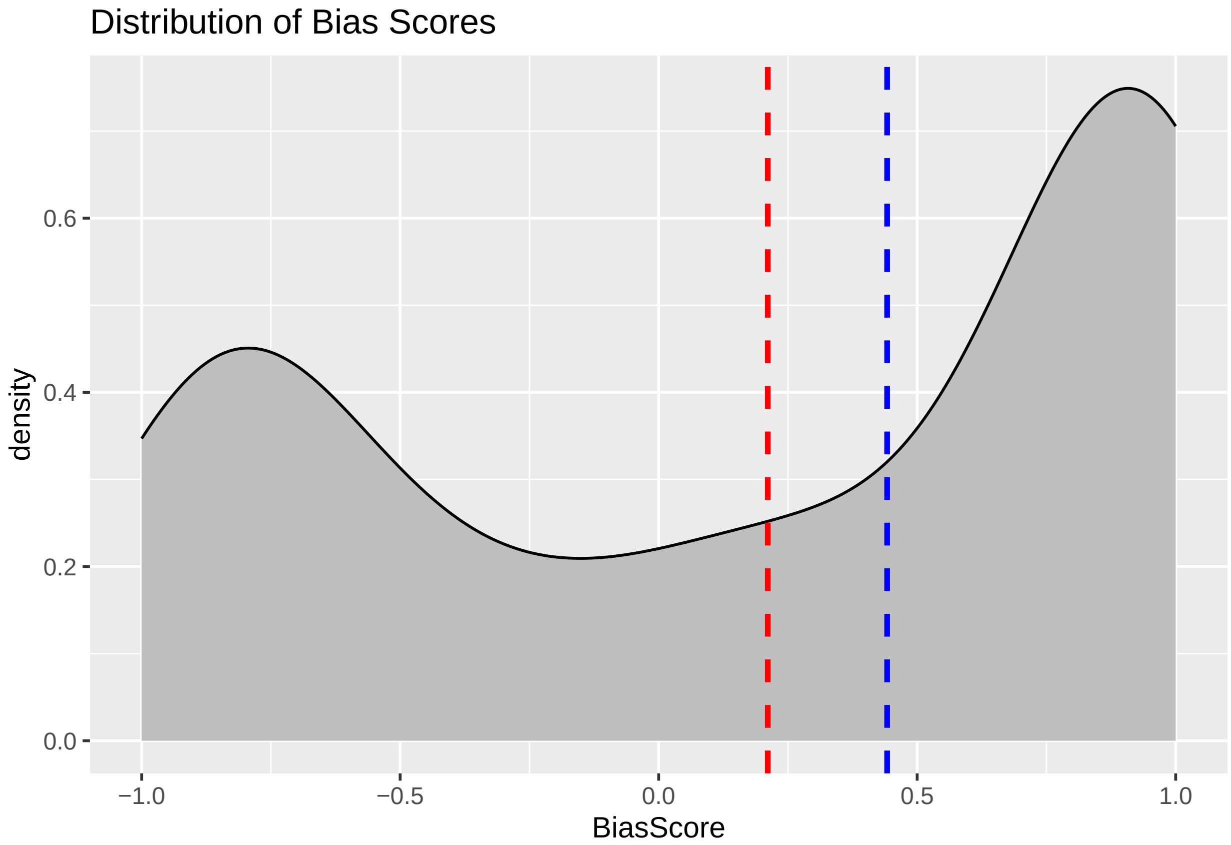
```
biasMean <- mean(data$BiasScore, na.rm = T)
biasMedian <- median(data$BiasScore, na.rm = T)

bias1sd <- biasMean + sd(data$BiasScore)
biasneg1sd <- biasMean - sd(data$BiasScore)


data %>%
  ggplot(aes(x=BiasScore)) +
  geom_histogram(fill = "grey", color = "black") +
  geom_vline(mapping = aes(xintercept = biasMean), color = "red", linetype = "dashed", size = 1) +
  geom_vline(mapping = aes(xintercept = biasMedian), color = "blue", linetype = "dashed", size = 1) +
   geom_vline(mapping = aes(xintercept = bias1sd), color = "black", linetype = "dashed", size = 1) +
   geom_vline(mapping = aes(xintercept = biasneg1sd), color = "black", linetype = "dashed", size = 1) +
  ggtitle("Distribution of Bias Scores")
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



```
data %>%
  ggplot(aes(x=BiasScore)) +
  geom_density(fill = "grey", color = "black") +
   geom_vline(mapping = aes(xintercept = biasMean), color = "red", linetype = "dashed", size = 1) +
  geom_vline(mapping = aes(xintercept = biasMedian), color = "blue", linetype = "dashed", size = 1) +
  ggtitle("Distribution of Bias Scores")
```

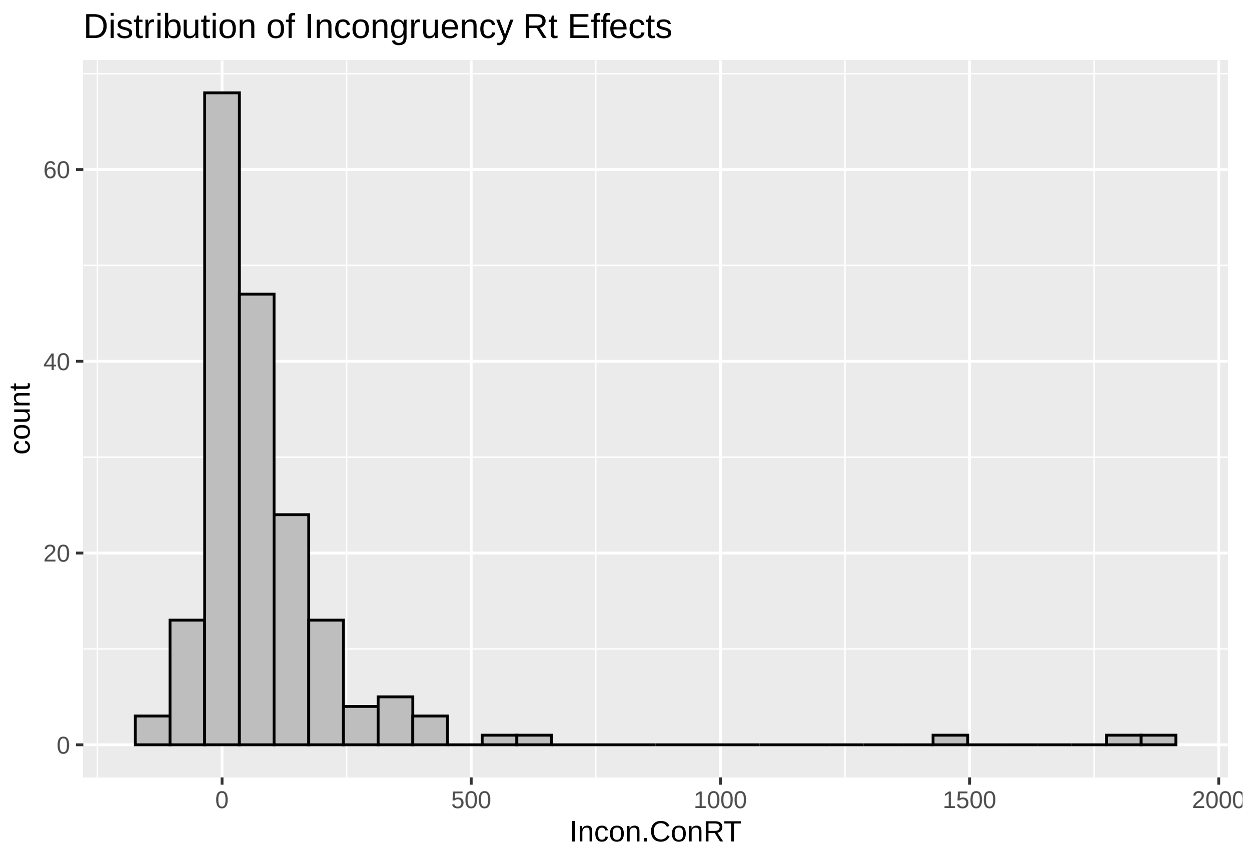


#A lot less neutral people than biased people

Distributions of Incongruency Effect - Whole Group & Bias Split

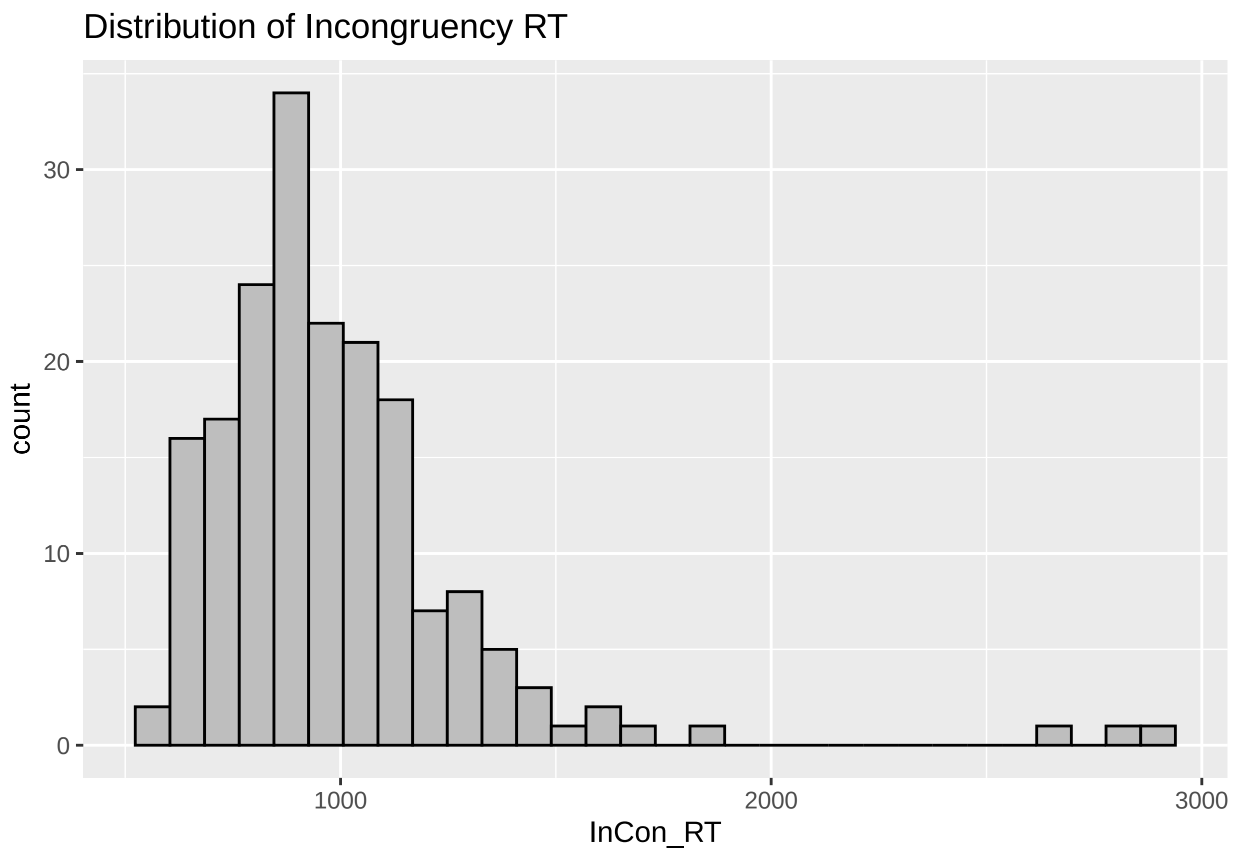
```
# Whole group distribution of incongruency effect

data %>%
  ggplot(aes(x=Incon.ConRT)) +
  geom_histogram(fill = "grey", color = "black") +
  ggtitle("Distribution of Incongruency Rt Effects")
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



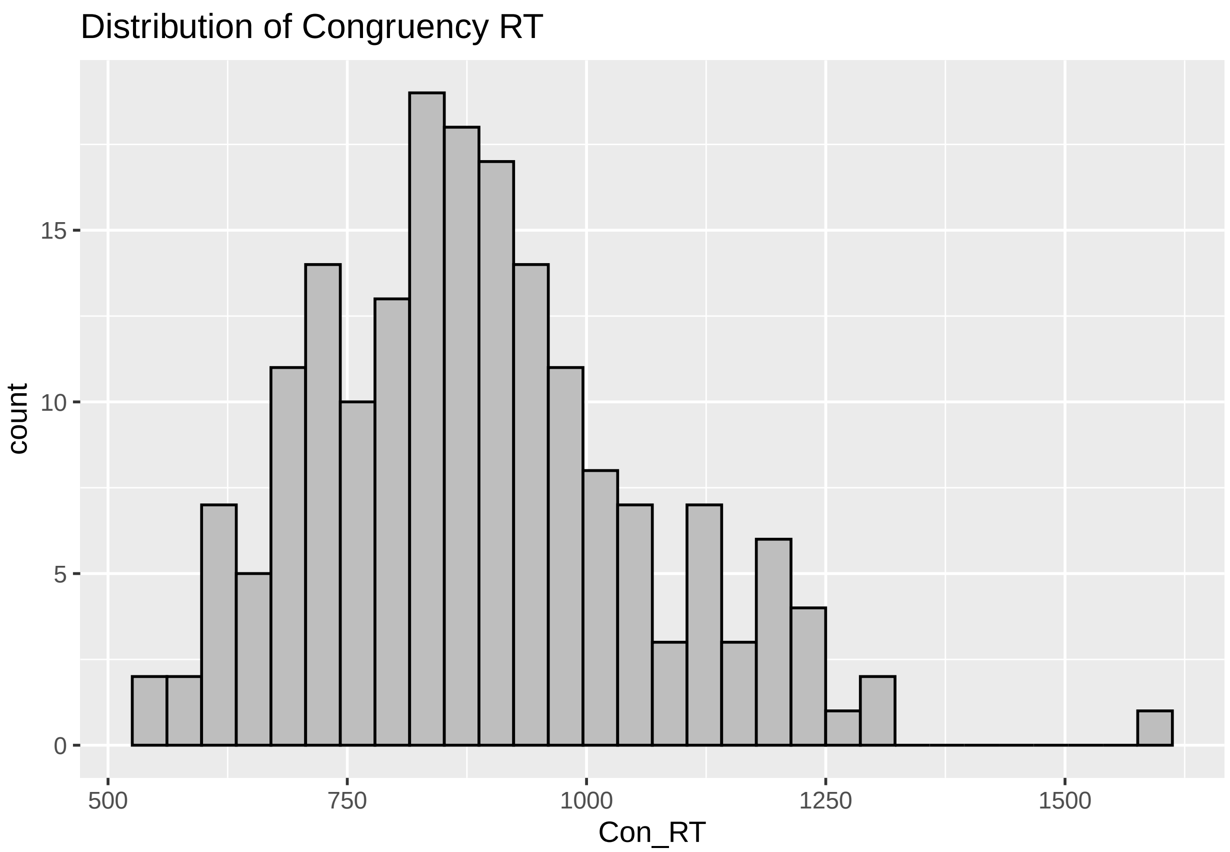
```
data %>%
  ggplot(aes(x=InCon_RT)) +
  geom_histogram(fill = "grey", color = "black") +
  ggtitle("Distribution of Incongruency RT")
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



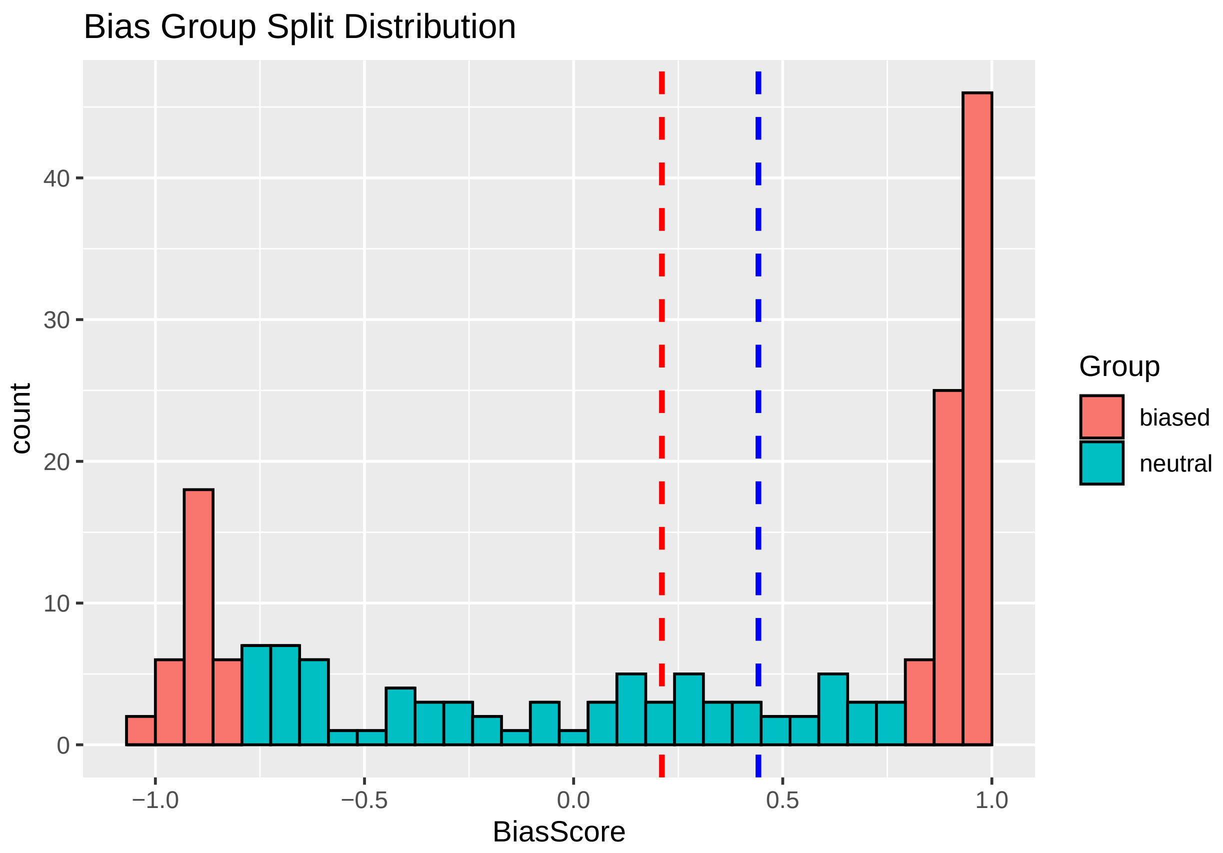
```
data %>%
  ggplot(aes(x=Con_RT)) +
  geom_histogram(fill = "grey", color = "black") +
  ggtitle("Distribution of Congruency RT")
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

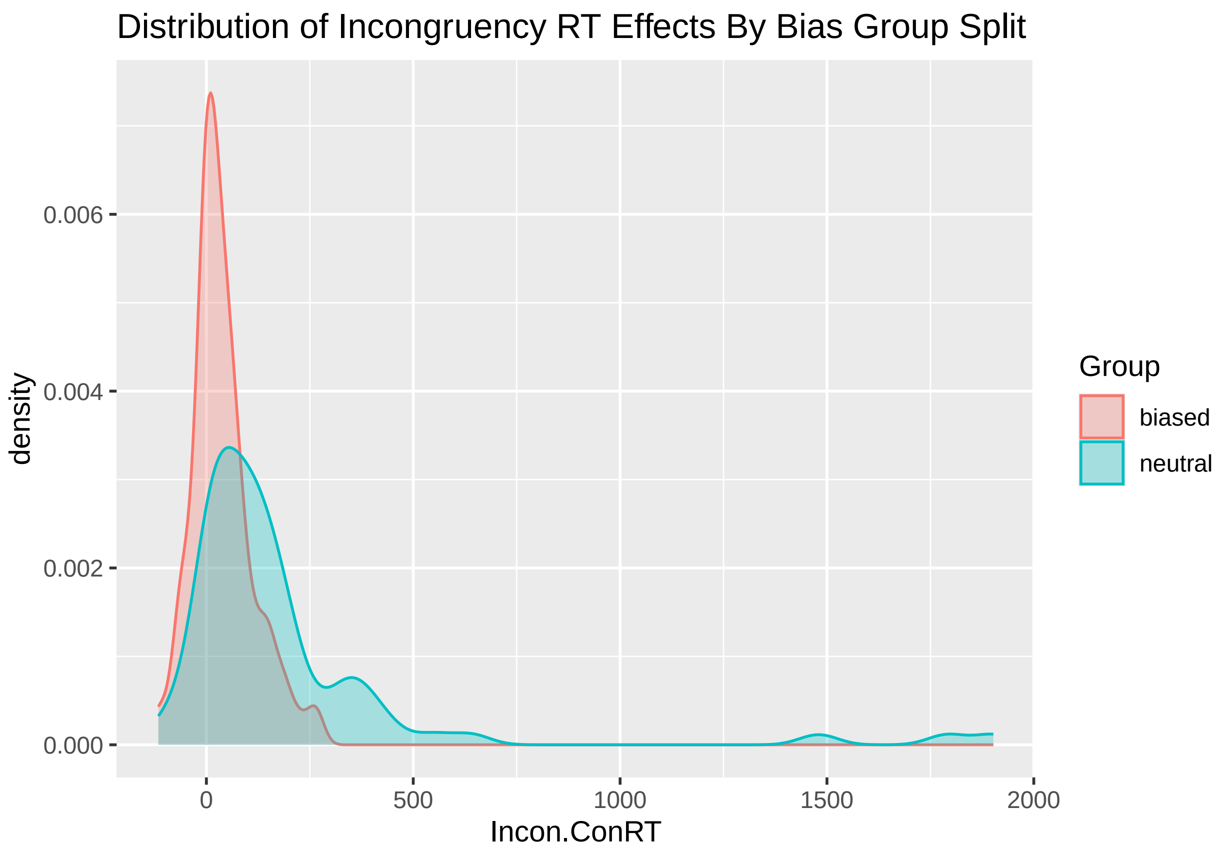


```
# Split groups based on bias score
biased <- subset(data, data$BiasScore > 0.8 | data$BiasScore < -0.8)
neutral <- subset(data, data$BiasScore <= 0.8 & data$BiasScore >= -0.8)

biased$Group = "biased"
neutral$Group = "neutral"

data_grouped <- rbind(biased, neutral)

data_grouped %>%
  ggplot(aes(x = BiasScore, fill = Group))+ geom_vline(mapping = aes(xintercept = biasMean), color = "re
  geom_vline(mapping = aes(xintercept = biasMedian), color = "blue", linetype = "dashed", size = 1) +
  geom_histogram(color="black") + ggtitle("Bias Group Split Distribution")
```

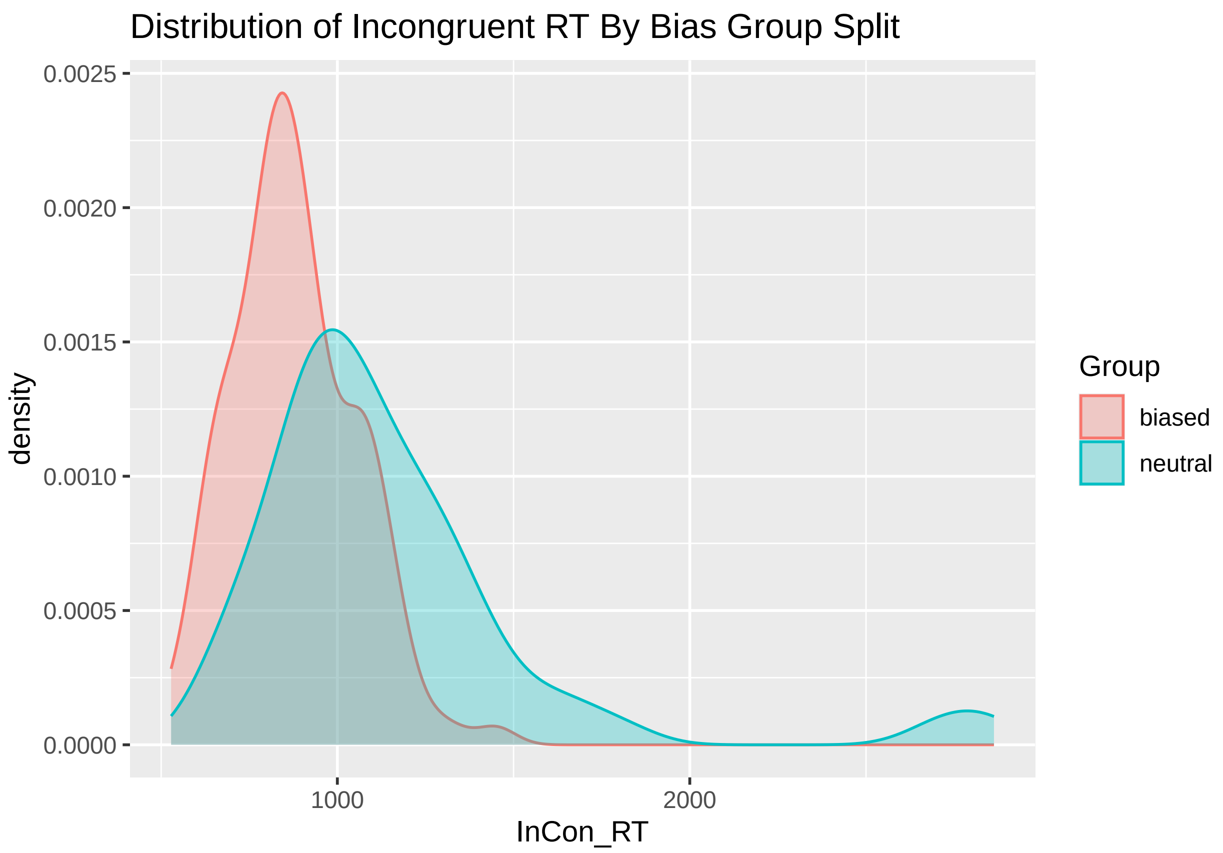
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



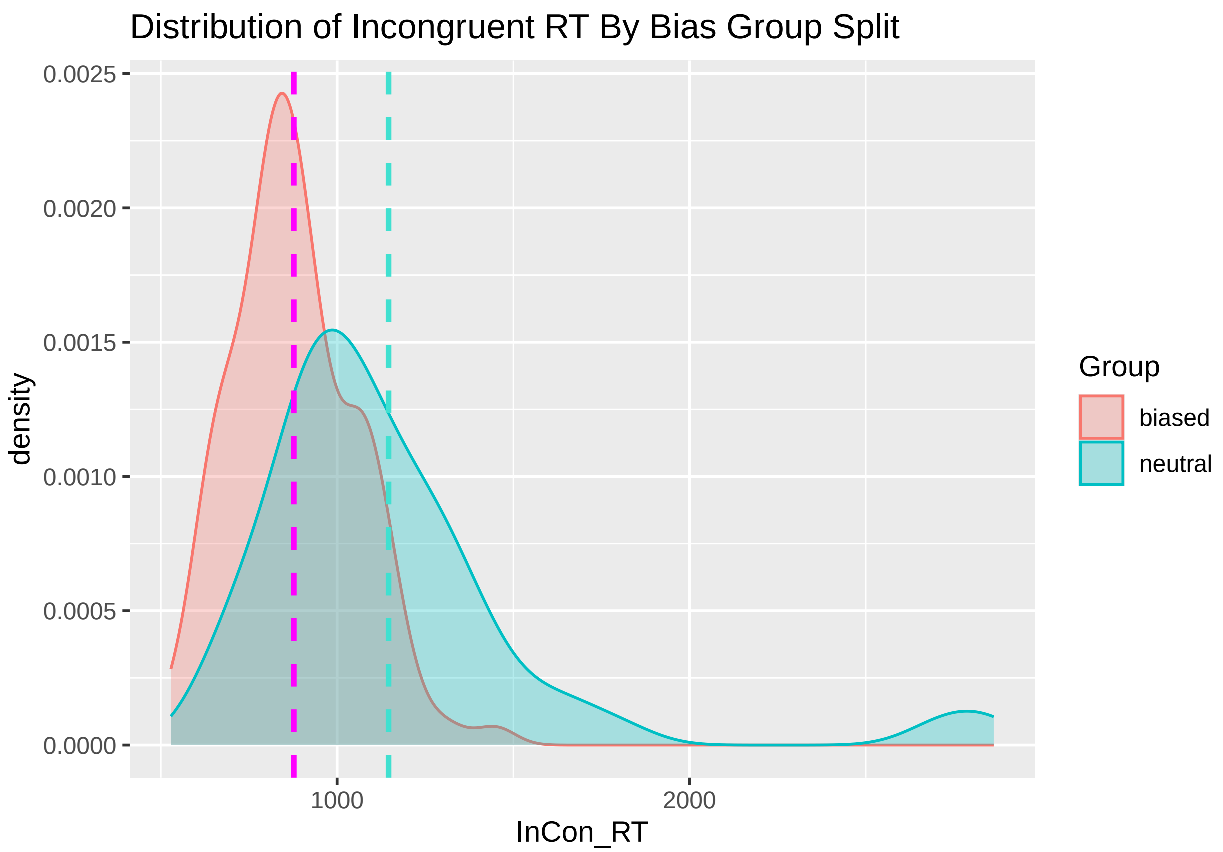
```
data_grouped %>%
  ggplot(aes(x = Incon.ConRT, fill = Group, color = Group))+
  geom_density(alpha = 0.3)+
  ggtitle("Distribution of Incongruency RT Effects By Bias Group Split")
```



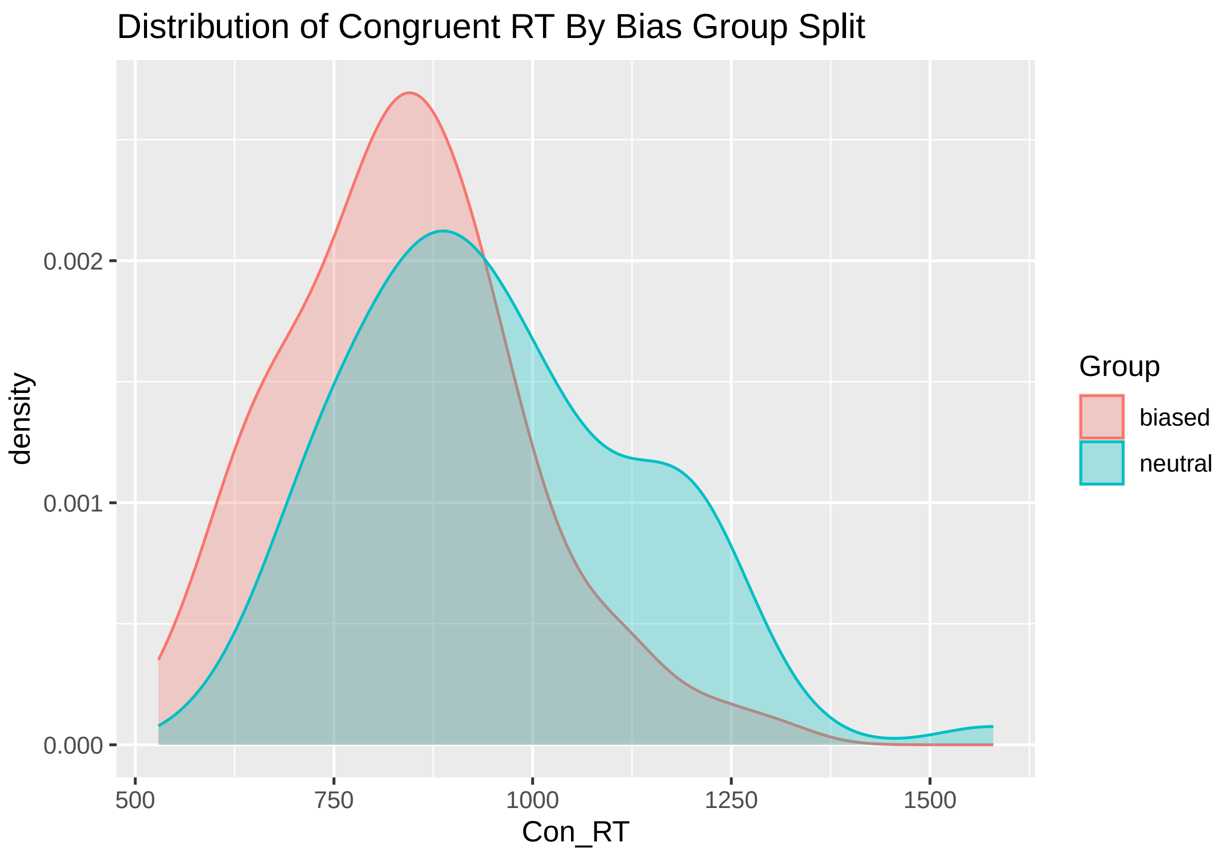
```
data_grouped %>%
  ggplot(aes(x = InCon_RT, fill = Group, color = Group))+
  geom_density(alpha = 0.3)+
  ggtitle("Distribution of Incongruent RT By Bias Group Split")
```



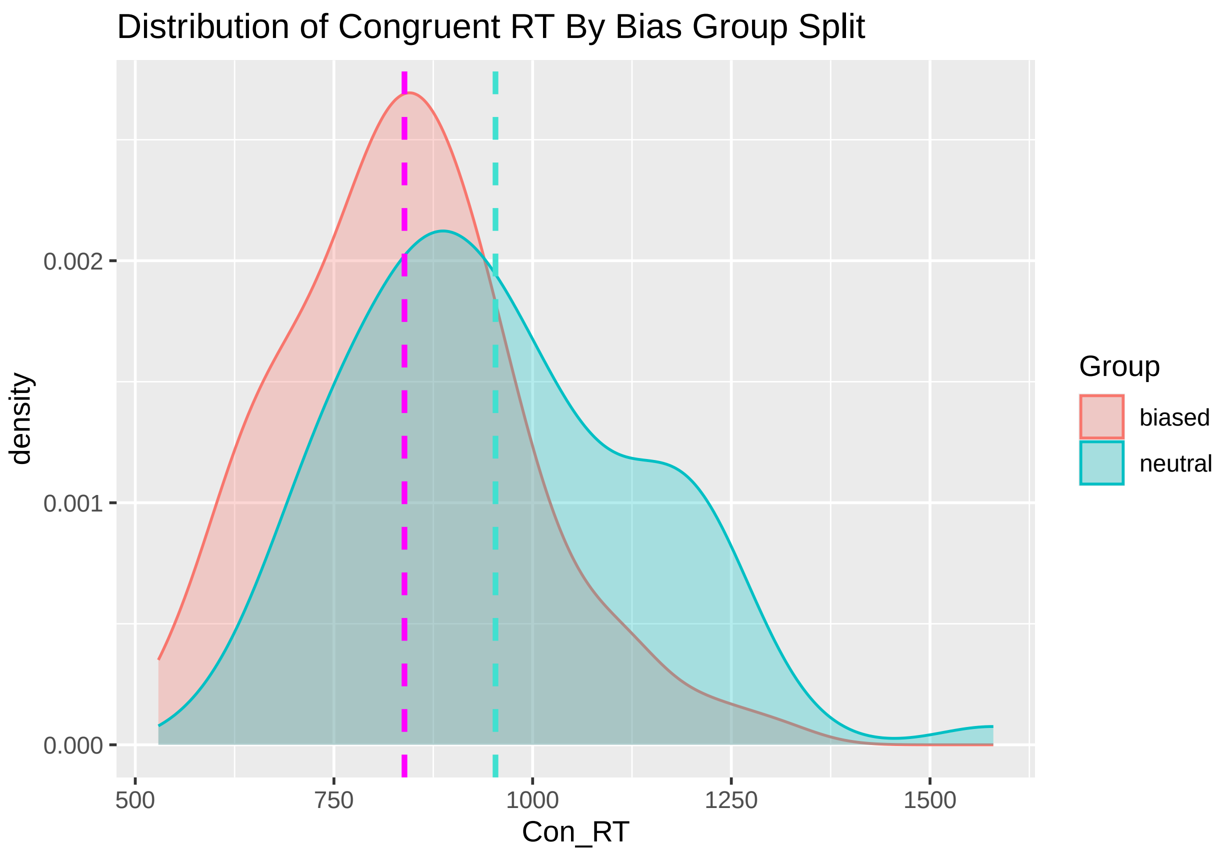
```
neutral_incon_mean = mean(neutral$InCon_RT)
neutral_con_mean = mean(neutral$Con_RT)
biased_incon_mean = mean(biased$InCon_RT)
biased_con_mean = mean(biased$Con_RT)


data_grouped %>%
  ggplot(aes(x = InCon_RT, fill = Group, color = Group))+
  geom_density(alpha = 0.3)+geom_vline(mapping = aes(xintercept = neutral_incon_mean), color = "turquoi
  ggtitle("Distribution of Incongruent RT By Bias Group Split")
```



```
data_grouped %>%
  ggplot(aes(x = Con_RT, fill = Group, color = Group))+
  geom_density(alpha = 0.3)+
  ggtitle("Distribution of Congruent RT By Bias Group Split")
```



```
data_grouped %>%
  ggplot(aes(x = Con_RT, fill = Group, color = Group))+
  geom_density(alpha = 0.3)+geom_vline(mapping = aes(xintercept = neutral_con_mean), color = "turquoise
  ggtitle("Distribution of Congruent RT By Bias Group Split")
```



#Incongruency RT Effect Dist: Most people have an incongruency effect of 0 or > so, as expected, incongruent trial reaction times are generally greater than congruent trial reaction times. Slight right skew that is made extreme by outliers- will reexamine with outliers removed. #Distribution of Incongruency Effect by Bias Group Split: There are more people who have a lower incongruency effect in the biased attender group than in the neutral attender group. Neutral attender group has a lot less people but its density is a lot more spread out while biased attender incongruency effect has a lot less range and is more concentrated near 0. #Congruent and Incongruent Distributions by Bias Group Split: Biased group seems to be taking less time than the neutral group in both trial types. Could be indication of biased attenders mainly paying attention to their preferred information processing style and ignoring the other stimulus so RTs are faster in both trial types.

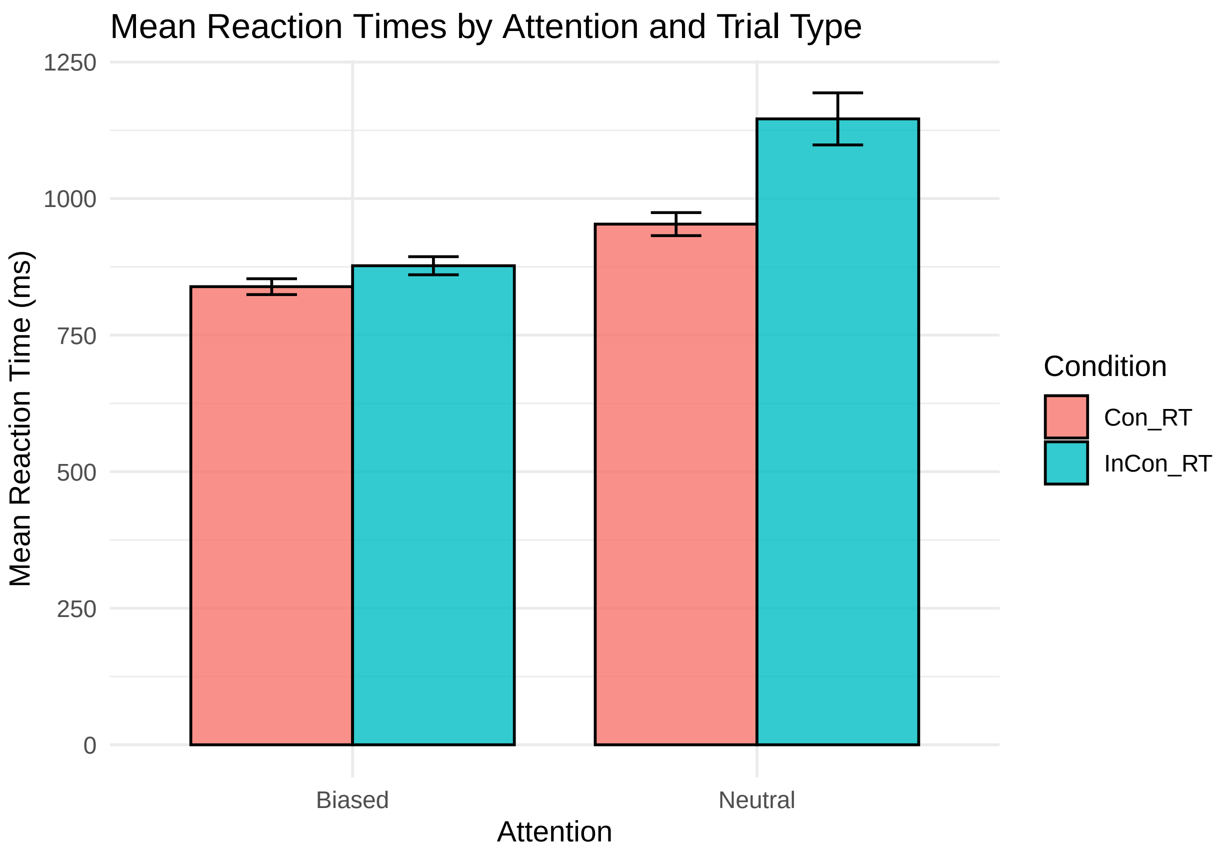
Reaction times by trial type and attention

```
# Load necessary libraries
library(tidyverse)
library(ggplot2)
library(reshape2)

data$Attention <- ifelse(data$BiasScore > 0.8 | data$BiasScore < -0.8, "Biased", "Neutral")
data$IPS <- ifelse(data$BiasScore > 0, "Verbal", "Visual")

#Reaction times
#Biased attender histograms and descriptive statistics
biased_data<- data[data$Attention == "Biased", ]
```

```
neutral_data<- data[data$Attention == "Neutral", ]

# Combine biased and neutral data
data$Attention <- ifelse(data$BiasScore > 0.8 | data$BiasScore < -0.8, "Biased", "Neutral")
biased_data<- data[data$Attention == "Biased", ]
combined_data <- rbind(biased_data, neutral_data)

# Calculate means by attention and trial type
means <- combined_data %>%
  group_by(Attention) %>%
  summarise(Con_RT = mean(Con_RT), InCon_RT = mean(InCon_RT)) %>%
  pivot_longer(cols = c(Con_RT, InCon_RT), names_to = "Trial_Type", values_to = "RT")

# Order factor levels for better plotting
means$Attention <- factor(means$Attention, levels = unique(means$Attention))

combined_long <- combined_data %>%
  select(Attention, InCon_RT, Con_RT) %>%
  pivot_longer(cols = c(InCon_RT, Con_RT), values_to = "RT", names_to = "Condition")

# Creat error bars
se_sum <- combined_long %>%
  group_by(Attention, Condition) %>%
  summarise(
    sd = sd(RT),
    n = n(),
    mean = mean(RT)
  ) %>%
  mutate(se = sd/sqrt(n))
```

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

```
# Create bar plot

ggplot(se_sum, aes(x = Attention, y = mean, fill = Condition)) +
  geom_bar(position = position_dodge(0.8), stat = "identity", color = "black", size = 0.5, width = 0.8,
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), position = position_dodge(0.8), width = 0.25,
  labs(title = "Mean Reaction Times by Attention and Trial Type",
       x = "Attention",
       y = "Mean Reaction Time (ms)") +
  theme_minimal()
```



```
combined_long_wid <- combined_data %>%
  select(Attention, InCon_RT, Con_RT, Subject) %>%
  pivot_longer(cols = c(InCon_RT, Con_RT), values_to = "RT", names_to = "Condition")
```

#incon rts are larger overall, but the difference between incon and con trial rts for the biased group is a lot smaller than the neutral group- perhaps because the neutral group is noticing both stimuli more? #figure out how to put error bars

Incongruency Effect Analysis

Incongruency Effect Descriptives

```
# Incongruency Effect Calculation
data$IncongruencyEffect <- data$InCon_RT - data$Con_RT
Incongruency_Effect_Data <- data$IncongruencyEffect

biased_data$Incongruency_Effect_Data <- biased_data$InCon_RT - biased_data$Con_RT
neutral_data$Incongruency_Effect_Data <- neutral_data$InCon_RT - neutral_data$Con_RT
```

Biased group descriptives

```r
psych::describe(biased_data$Incongruency_Effect_Data)
```

```
##    vars   n  mean   sd median trimmed   mad     min    max  range skew kurtosis
## X1    1 109 38.32 72.9  25.15   32.84 52.51 -114.36 265.55 379.91 0.85     0.97
##      se
## X1 6.98
```

Neutral group descriptives

```r
psych::describe(neutral_data$Incongruency_Effect_Data)
```

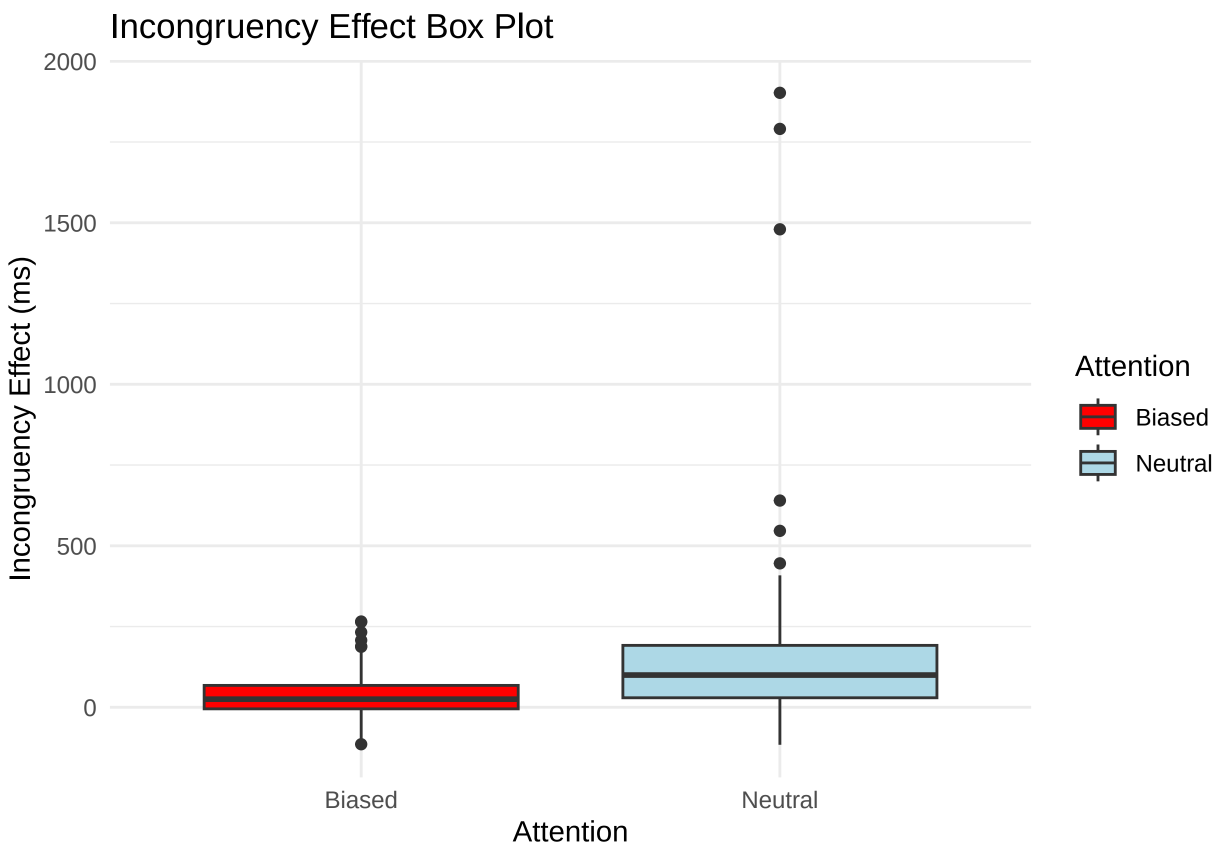
```
##    vars  n   mean    sd median trimmed    mad     min     max   range skew
## X1    1 76 192.68 344.6  99.85  124.84 121.86 -116.09 1902.28 2018.37  3.6
##    kurtosis    se
## X1    13.72 39.53
```

```r
# Biased Attender Incongruency Effect Descriptive Statistics
biased_IE_mean <- mean(biased_data$Incongruency_Effect_Data)
biased_IE_std <- sd(biased_data$Incongruency_Effect_Data)
biased_IE_min <- min(biased_data$Incongruency_Effect_Data)
biased_IE_max <- max(biased_data$Incongruency_Effect_Data)

# Data frame for Biased Attender Incongruency Effect Descriptive Statistics
biased_descriptive_IE <- data.frame(
  Attention = "Biased",
  Variable = "Incongruency Effect",
  Mean = biased_IE_mean,
  StdDev = biased_IE_std,
  Min = biased_IE_min,
  Max = biased_IE_max,
  stringsAsFactors = FALSE
)

# Neutral Attender Incongruency Effect Descriptive Statistics
neutral_IE_mean <- mean(neutral_data$Incongruency_Effect_Data)
neutral_IE_std <- sd(neutral_data$Incongruency_Effect_Data)
neutral_IE_min <- min(neutral_data$Incongruency_Effect_Data)
neutral_IE_max <- max(neutral_data$Incongruency_Effect_Data)

# Data frame for Neutral Attender Incongruency Effect Descriptive Statistics
neutral_descriptive_IE <- data.frame(
  Attention = "Neutral",
  Variable = "Incongruency Effect",
  Mean = neutral_IE_mean,
  StdDev = neutral_IE_std,
  Min = neutral_IE_min,
  Max = neutral_IE_max,
  stringsAsFactors = FALSE
)

descriptive_statistics_IE <- rbind(biased_descriptive_IE, neutral_descriptive_IE)
print(descriptive_statistics_IE)
```

```
##   Attention             Variable      Mean   StdDev       Min       Max
## 1    Biased Incongruency Effect  38.32225  72.8954 -114.3611  265.5486
## 2   Neutral Incongruency Effect 192.68412 344.5965 -116.0903 1902.2847
```

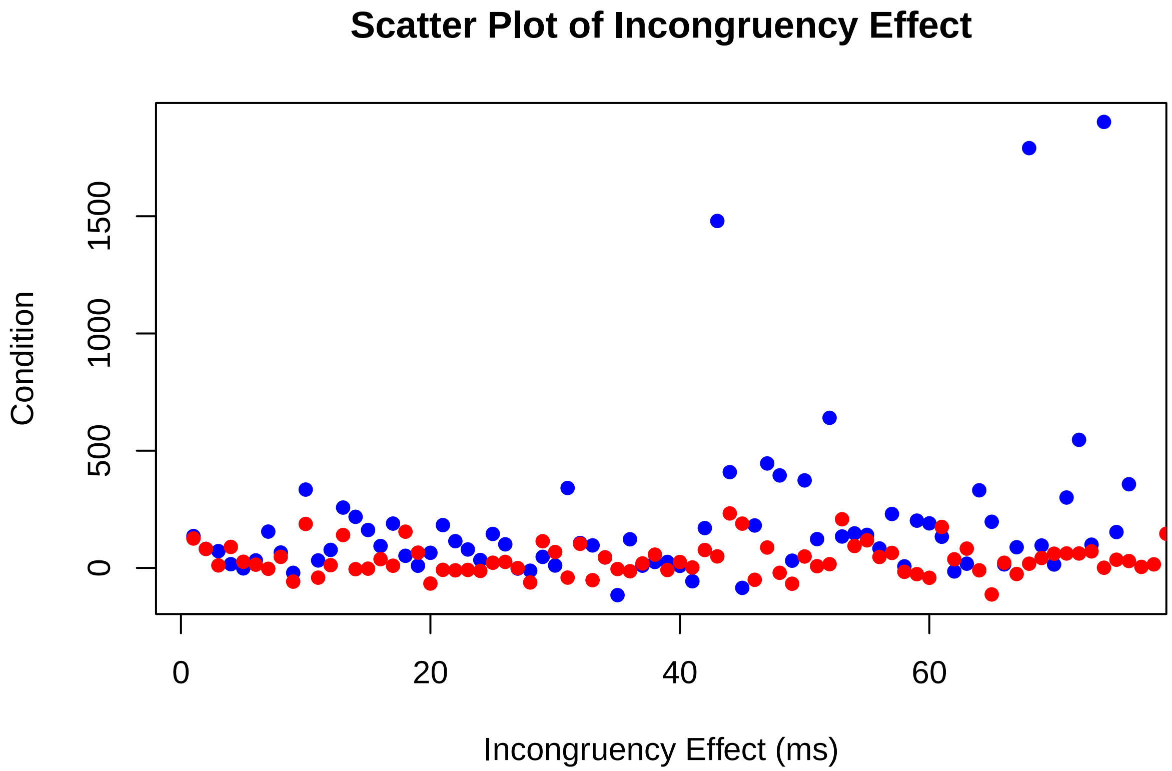
Incongruency Effect Box Plot

```
# Load necessary libraries
library(ggplot2)

# Combine biased and neutral data for the box plot
combined_data <- rbind(biased_data, neutral_data)

# Create box plot
ggplot(combined_data, aes(x = Attention, y = Incongruency_Effect_Data, fill = Attention)) +
  geom_boxplot() +
  labs(
    title = "Incongruency Effect Box Plot",
    x = "Attention",
    y = "Incongruency Effect (ms)"
  ) +
  scale_fill_manual(values = c("red", "lightblue")) +  # Color for biased and neutral data
  theme_minimal()
```



## Incongruency Effect Scatter Plot

```
# Scatter Plot of Incongruency Effect
plot(neutral_data$Incongruency_Effect_Data,
     col = "blue",
     pch = 16,
     main = "Scatter Plot of Incongruency Effect",
     xlab = "Incongruency Effect (ms)",
     ylab = "Condition")

# Adding Biased Attender Data Points
points(biased_data$Incongruency_Effect_Data,
       col = "red",
       pch = 16)
```



Individual Differences Analysis

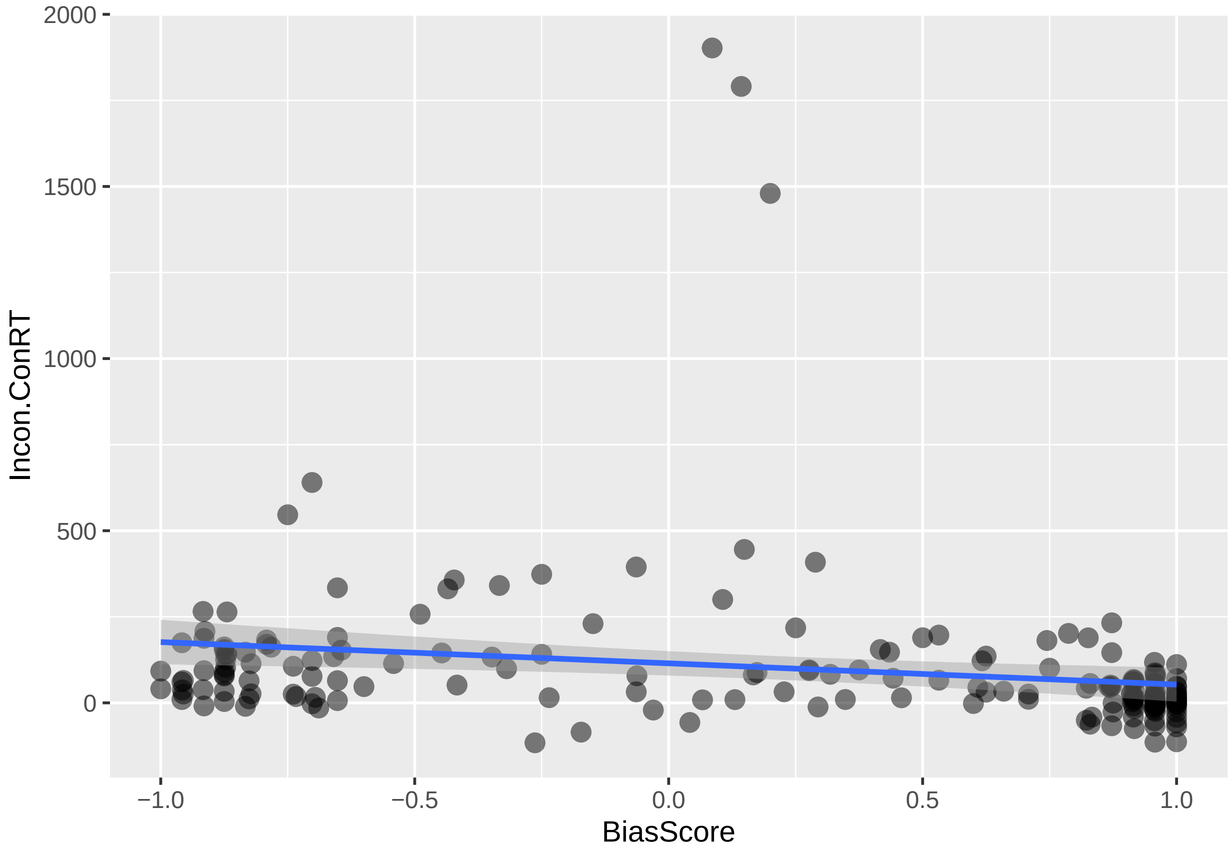
Correlate Bias score in whole group with incongruency RT effect

```r
cor.test(data$BiasScore, data$Incon.ConRT)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$BiasScore and data$Incon.ConRT
## t = -2.6908, df = 183, p-value = 0.007789
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.33006526 -0.05228936
## sample estimates:
##        cor
## -0.1950863
```

```r
data %>%
  ggplot(aes(x = BiasScore, y = Incon.ConRT)) +
  geom_point(size = 3, alpha = 0.5, fill = "grey") +
  geom_smooth(method = "lm")
```

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

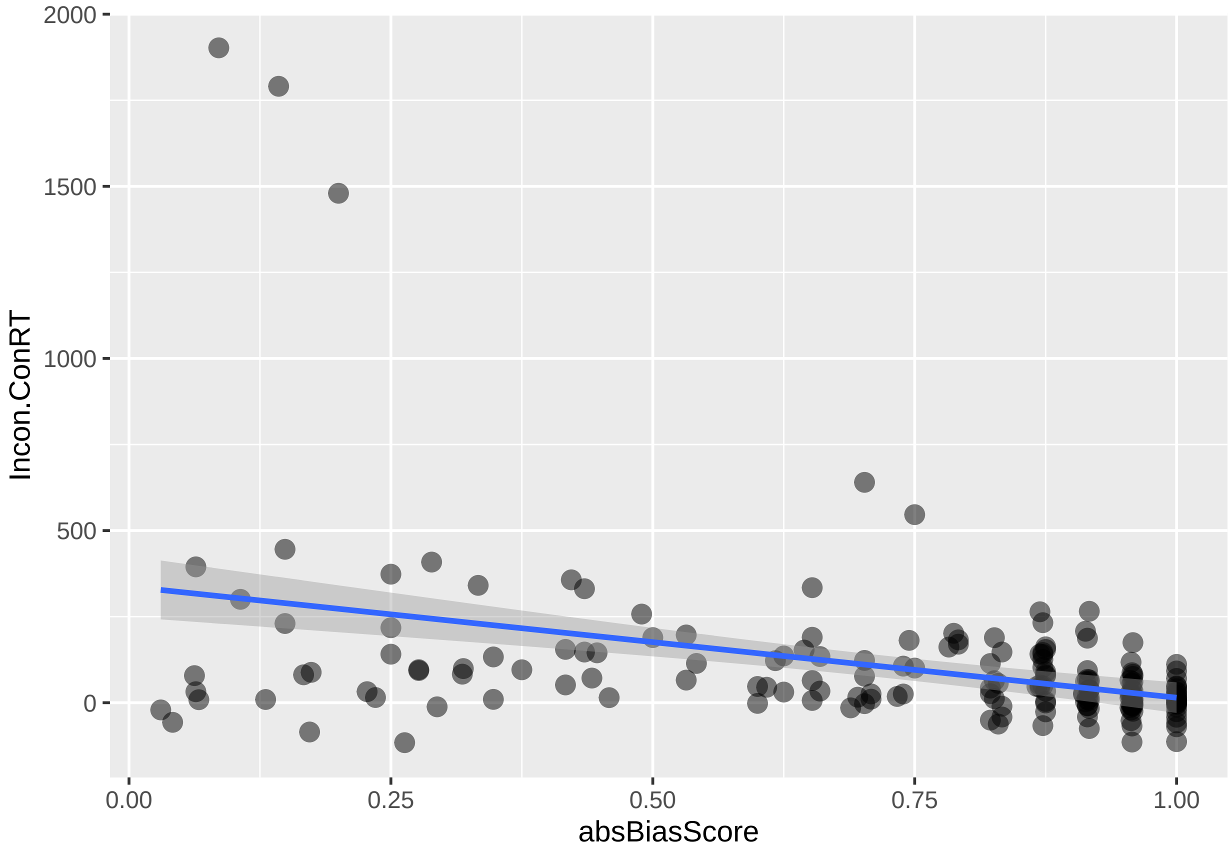


```
data$absBiasScore <- abs(data$BiasScore)

cor.test(data$absBiasScore, data$Incon.ConRT)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$absBiasScore and data$Incon.ConRT
## t = -5.6186, df = 183, p-value = 7.071e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5001629 -0.2533216
## sample estimates:
##       cor
## -0.383572
```

```
data %>%
  ggplot(aes(x = absBiasScore, y = Incon.ConRT)) +
  geom_point(size = 3, alpha = 0.5, fill = "grey") +
  geom_smooth(method = "lm")
```

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

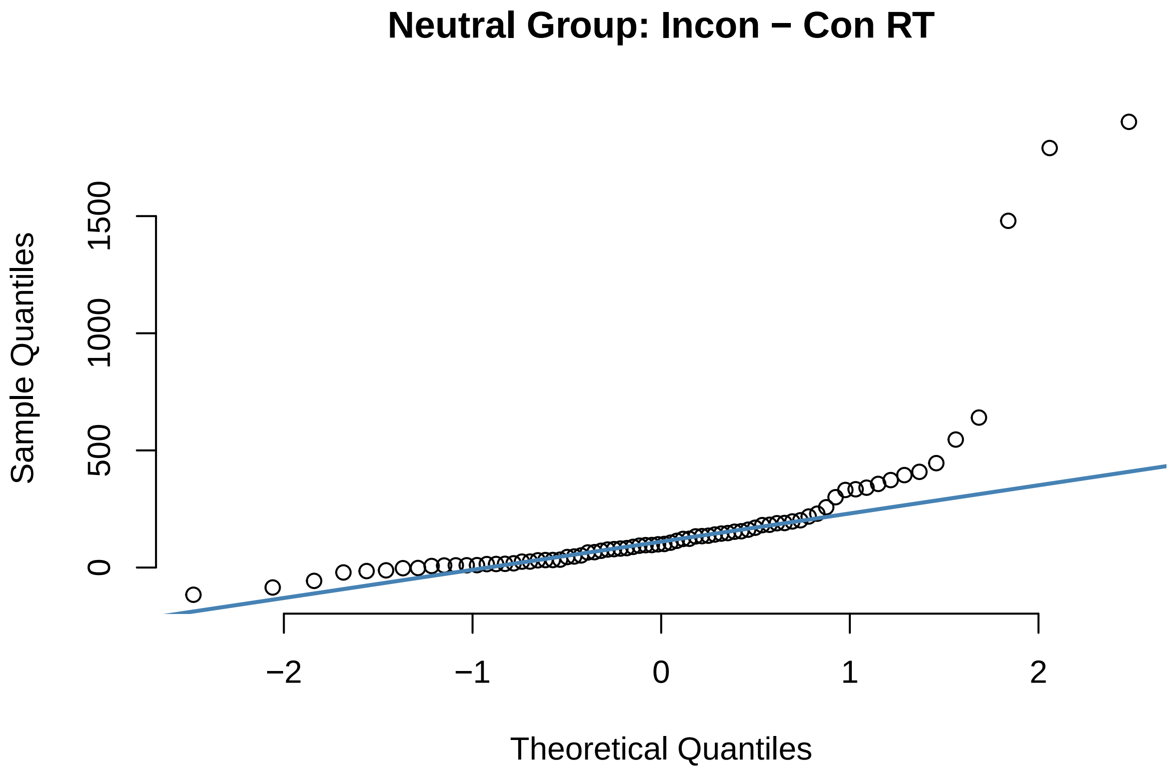


#doesn't look like much of a correlation, reexamine with outliers removed.

```
combined_long_wid <- combined_data %>%
  select(Attention, InCon_RT, Con_RT, Subject) %>%
  pivot_longer(cols = c(InCon_RT, Con_RT), values_to = "RT", names_to = "Condition")

#anova_test(data=combined_long_wid, dv=RT, wid=Subject, between=Attention, within = Condition)
```

T-test code - with full data (keeping outliers)

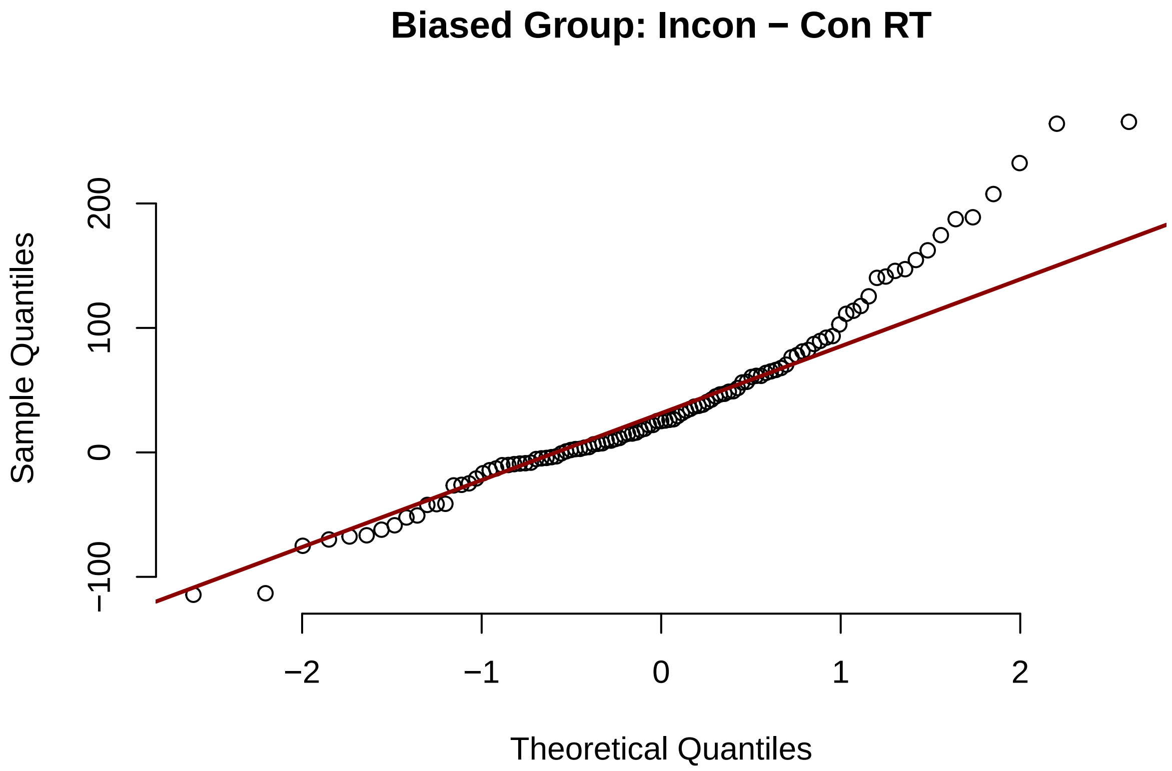
```
#   Difference in Incon-Con RT between Attention Groups

biased<- data[data$Attention == "Biased", ]
neutral<- data[data$Attention == "Neutral", ]

#First test normality assumption in Neutral Group
qqnorm(neutral$Incon.ConRT, pch = 1, frame = FALSE, main = "Neutral Group: Incon - Con RT")
qqline(neutral$Incon.ConRT, col = "steelblue", lwd = 2)
```



```
shapiro.test(neutral$Incon.ConRT) # Assumption of normality is violated; probably due to outliers
```

```
##
##  Shapiro-Wilk normality test
##
## data:  neutral$Incon.ConRT
## W = 0.5497, p-value = 6.594e-14
```

```
#Then test normality assumption in Biased Group
qqnorm(biased$Incon.ConRT, pch = 1, frame = FALSE, main = "Biased Group: Incon - Con RT")
qqline(biased$Incon.ConRT, col = "darkred", lwd = 2)
```



```
shapiro.test(biased$Incon.ConRT) # Assumption of normality is marginally violated
```

```
##
##  Shapiro-Wilk normality test
##
## data:  biased$Incon.ConRT
## W = 0.94661, p-value = 0.0002631
```

```
#Check that the variance does not differ between groups
# Perform Levene's AT_FormTest
print(leveneTest(Incon.ConRT ~ Attention, data = combined_data)) # Variances are marginally different;
```

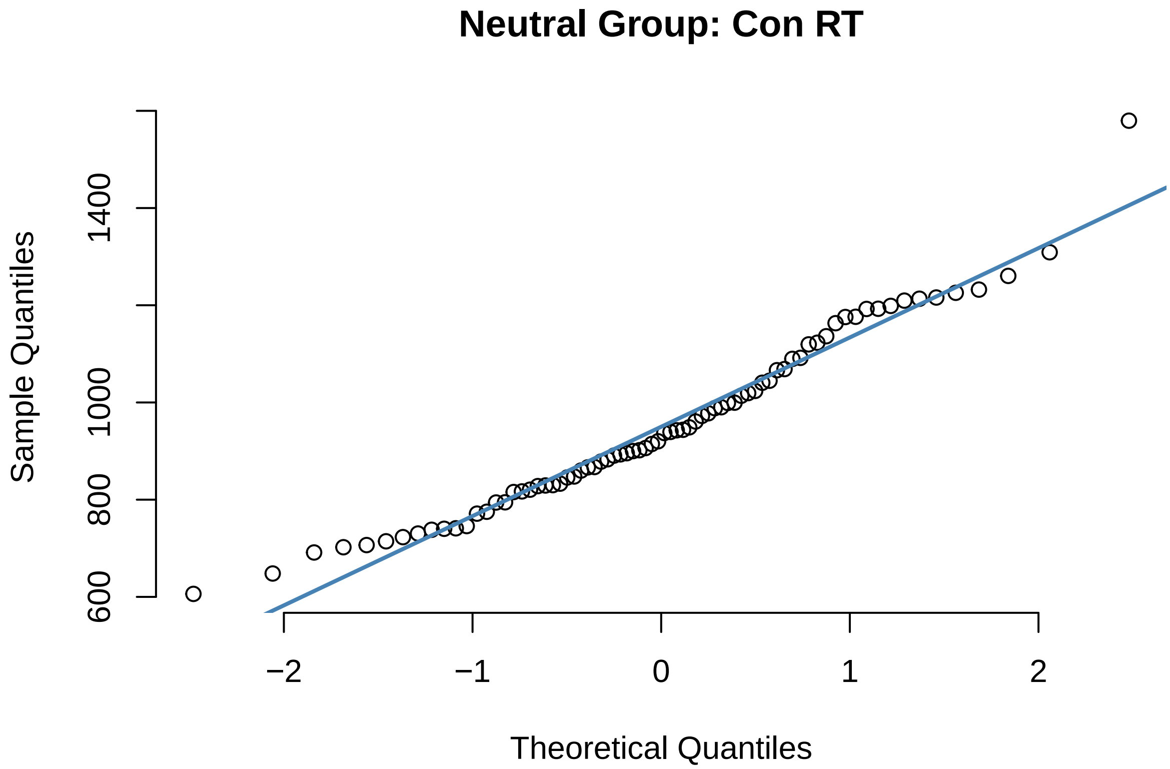
```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene’s Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   1  12.698 0.0004668 ***
##       183
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```
# Conduct t-test with equal variance assumption
print(t.test(neutral$Incon.ConRT, biased$Incon.ConRT, var.equal = F)) # T-Test is significant after cor
```

```
##
##  Welch Two Sample t-test
##
## data:  neutral$Incon.ConRT and biased$Incon.ConRT
## t = 3.8456, df = 79.699, p-value = 0.0002408
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   74.47641 234.24733
## sample estimates:
## mean of x mean of y
## 192.68412  38.32225
```

```
# test normality
qqnorm(neutral$Con_RT, pch = 1, frame = FALSE, main = "Neutral Group: Con RT")
qqline(neutral$Con_RT, col = "steelblue", lwd = 2)
```

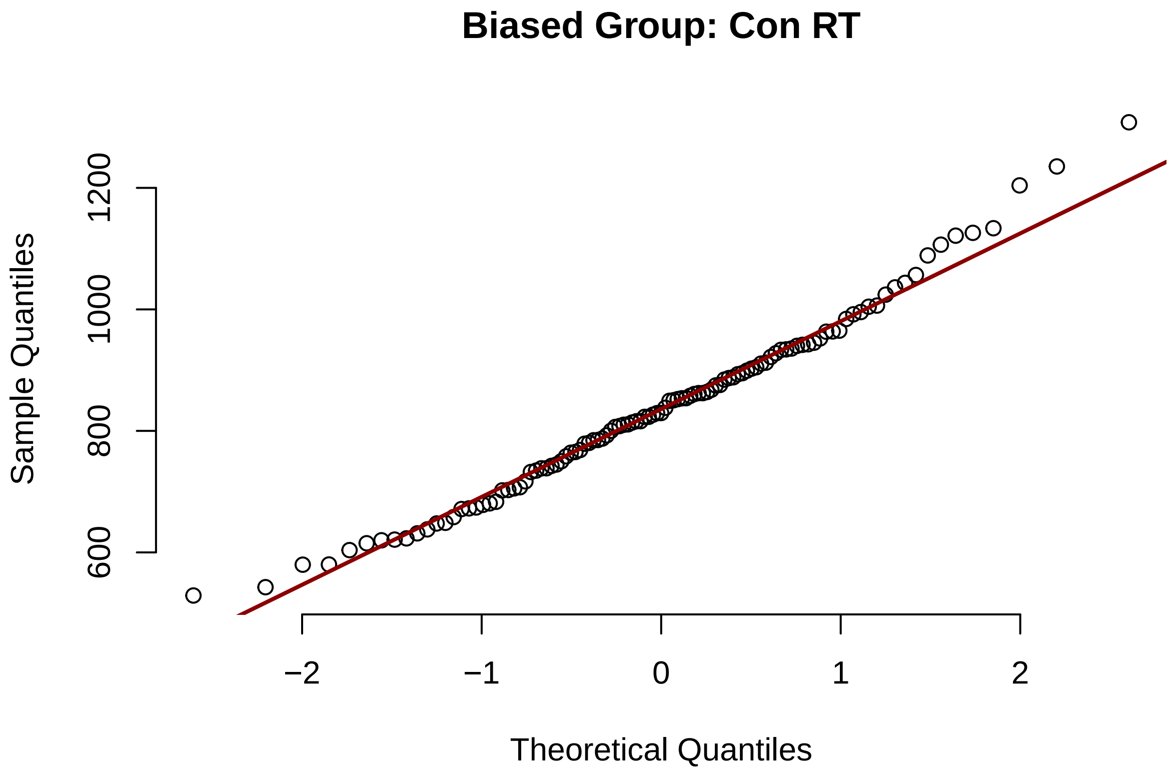


```
shapiro.test(neutral$Con_RT) #assumption of normality violated
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  neutral$Con_RT
## W = 0.96673, p-value = 0.04339
```

```
qqnorm(biased$Con_RT, pch = 1, frame = FALSE, main = "Biased Group: Con RT")
qqline(biased$Con_RT, col = "darkred", lwd = 2)
```



```
shapiro.test(biased$Con_RT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  biased$Con_RT
## W = 0.98426, p-value = 0.2281
```

```
# check if variance differs between groups
print(leveneTest(Con_RT ~ Attention, data = combined_data)) #assumption of homogeneity violated
```

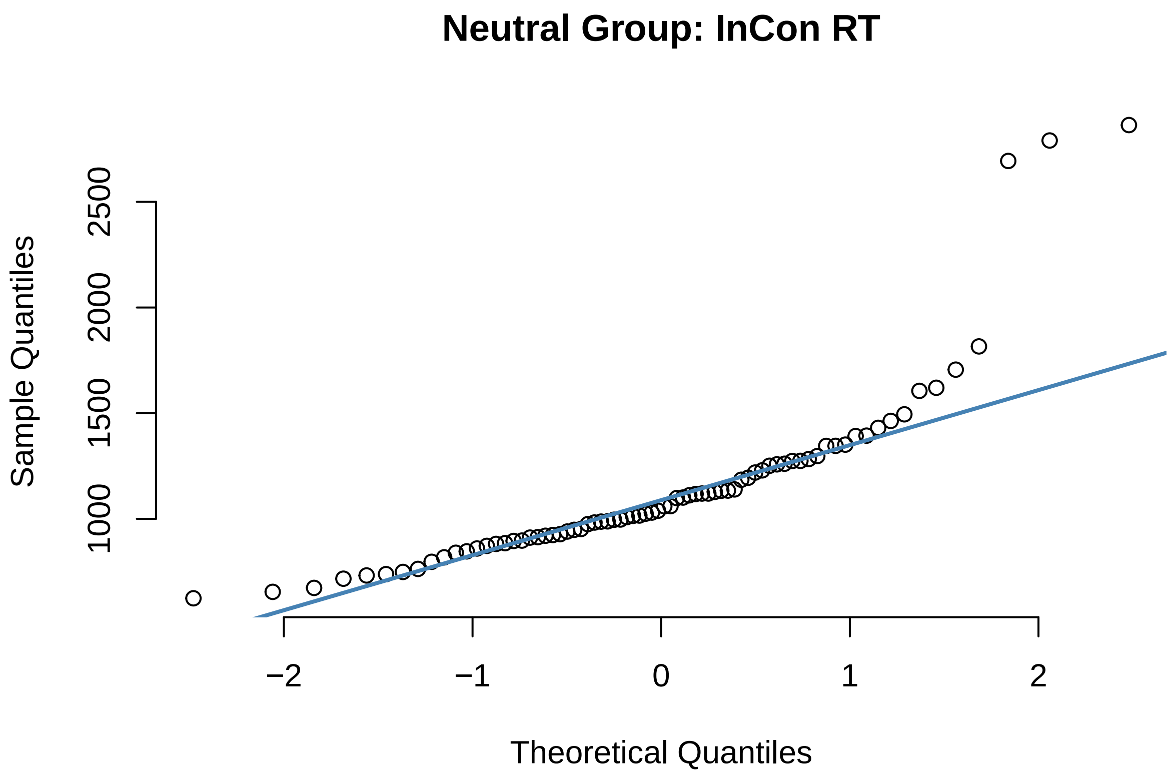
```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value  Pr(>F)
## group   1  3.3796 0.06763 .
##       183
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#do t-test
print(t.test(neutral$Con_RT, biased$Con_RT, var.equal = F))
```

```
##
##  Welch Two Sample t-test
##
## data:  neutral$Con_RT and biased$Con_RT
## t = 4.4737, df = 141.06, p-value = 1.57e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   63.90959 165.11498
## sample estimates:
## mean of x mean of y
##  953.2613  838.7490
```

```
# test normality
qqnorm(neutral$InCon_RT, pch = 1, frame = FALSE, main = "Neutral Group: InCon RT")
qqline(neutral$InCon_RT, col = "steelblue", lwd = 2)
```

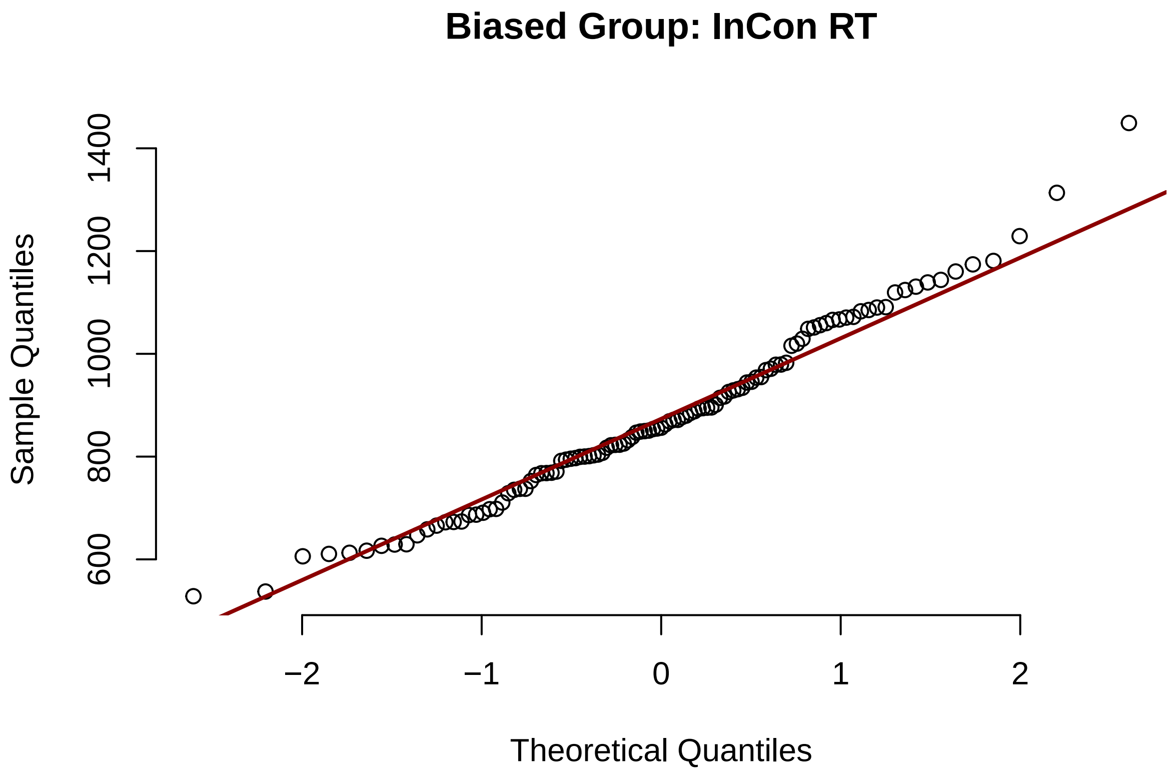


```
shapiro.test(neutral$InCon_RT)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  neutral$InCon_RT
## W = 0.75974, p-value = 8.108e-10
```

```
qqnorm(biased$InCon_RT, pch = 1, frame = FALSE, main = "Biased Group: InCon RT")
qqline(biased$InCon_RT, col = "darkred", lwd = 2)
```



```
shapiro.test(biased$InCon_RT) #assumption of normality violated
```

```
##
##  Shapiro-Wilk normality test
##
## data:  biased$InCon_RT
## W = 0.98125, p-value = 0.1282
```

```
# check if variance differs between groups
print(leveneTest(InCon_RT ~ Attention, data = combined_data))   #assumption of homogeneity not violated
```

```
## 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  12.393 0.0005437 ***
##       183
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#do t-test
print(t.test(neutral$InCon_RT, biased$InCon_RT, var.equal = T))
```

```
##
##  Two Sample t-test
##
## data:  neutral$InCon_RT and biased$InCon_RT
## t = 6.0477, df = 183, p-value = 8.089e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  181.1555 356.5928
## sample estimates:
## mean of x mean of y
## 1145.9454  877.0713
```

Removing Outliers InCon RT

```r
Incon_minus3SD <- mean(data$InCon_RT) - (3* sd(data$InCon_RT))
Incon_plus3SD <- mean(data$InCon_RT) + (3* sd(data$InCon_RT))

data <- data %>%
  mutate(InconOutlier = InCon_RT >= Incon_plus3SD)

outliers_subset = subset(data, data$InconOutlier == TRUE)


data_outliersremoved <- subset(data, data$InconOutlier == FALSE)
```

InCon Outliers Removed Descriptives

```r
data_outliersremoved %>%
  select(-c(Subject, Sex)) %>%
  psych::describe()
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf
```

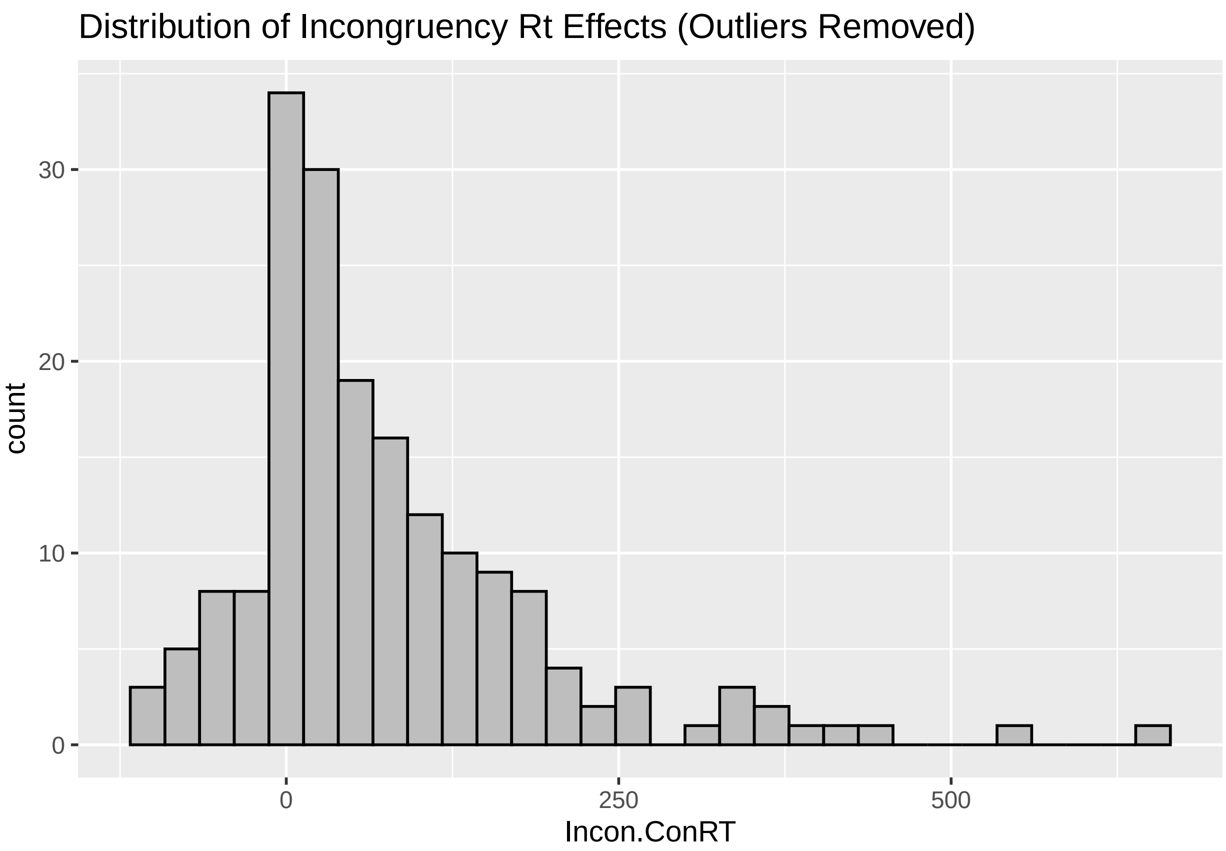
```
##                    vars   n  mean     sd median trimmed    mad     min     max
## Age                   1 180  39.89  14.54  38.00   38.99  13.34   10.00   87.00
## IncorRespCount        2 182   1.64   2.79   1.00    1.10   1.48    0.00   19.00
## SymRespCount          3 182  18.15  17.80  12.00   17.00  16.31    0.00   48.00
## TxtRespCount          4 182  28.20  18.07  34.50   29.10  18.53    0.00   48.00
## BiasScore             5 182   0.21   0.76   0.48    0.26   0.76   -1.00    1.00
## Con_RT                6 182 882.95 173.93 863.08  872.69 158.80  529.08 1579.72
## InCon_RT              7 182 957.95 230.28 921.78  939.41 209.53  528.29 1815.85
## Incon.ConRT           8 182  74.99 115.71  44.92   59.37  72.71 -116.09  640.06
## Attention*            9 182   1.40   0.49   1.00    1.38   0.00    1.00    2.00
## IPS*                 10 182   1.39   0.49   1.00    1.36   0.00    1.00    2.00
## IncongruencyEffect   11 182  74.99 115.71  44.92   59.37  72.71 -116.09  640.06
## absBiasScore         12 182   0.74   0.28   0.87    0.78   0.18    0.03    1.00
## InconOutlier         13 182    NaN     NA     NA     NaN     NA     Inf    -Inf
##                     range  skew kurtosis    se
## Age                 77.00  0.58     0.24  1.08
## IncorRespCount      19.00  3.90    17.10  0.21
## SymRespCount        48.00  0.43    -1.51  1.32
## TxtRespCount        48.00 -0.33    -1.62  1.34
## BiasScore            2.00 -0.38    -1.57  0.06
## Con_RT            1050.64  0.65     0.67 12.89
## InCon_RT          1287.56  0.87     1.00 17.07
## Incon.ConRT        756.15  1.75     4.40  8.58
## Attention*           1.00  0.40    -1.85  0.04
## IPS*                 1.00  0.45    -1.81  0.04
## IncongruencyEffect 756.15  1.75     4.40  8.58
## absBiasScore         0.97 -1.11    -0.04  0.02
## InconOutlier         -Inf    NA       NA    NA
```

#now that outliers are removed, incon effect mean went from 101.74 to 74.99.

Distributions of Incongruency Effect - Whole Group & Bias Split (InCon Outliers Removed)

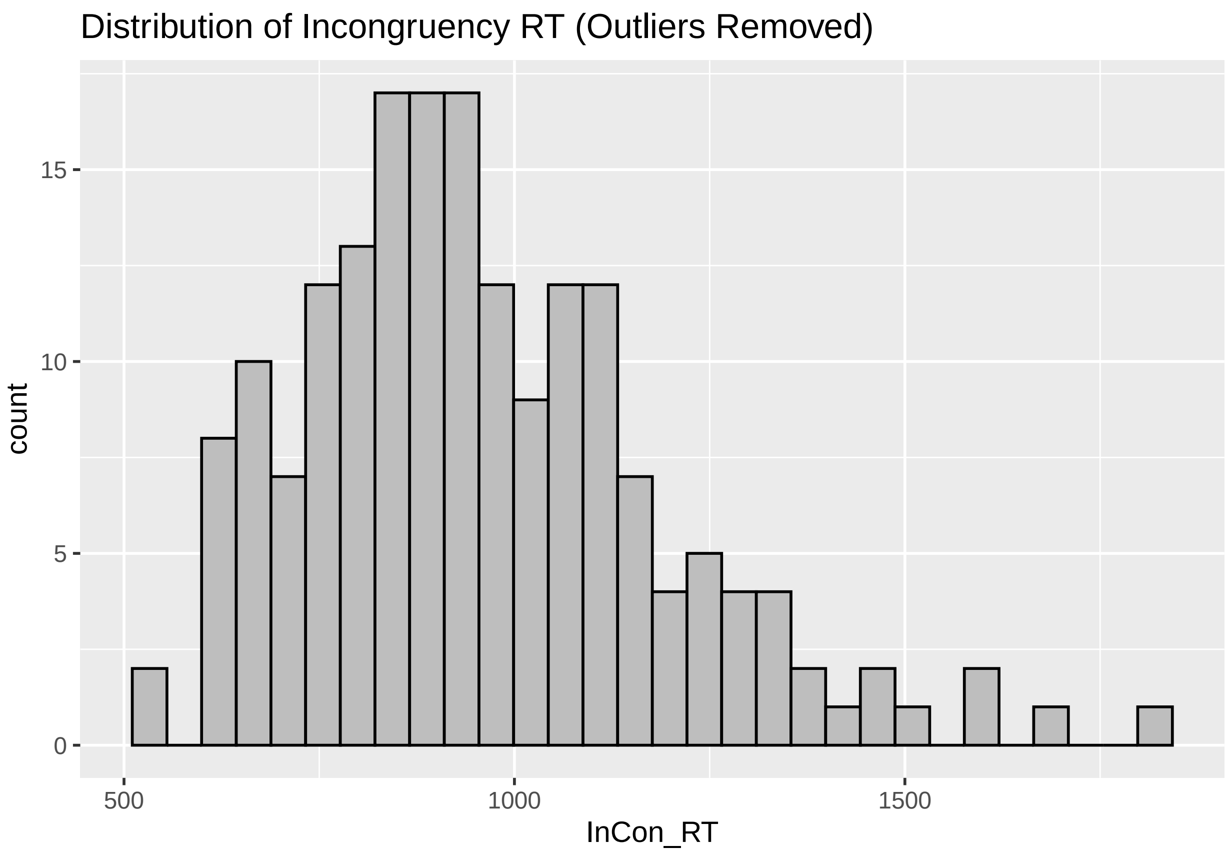
```
data_outliersremoved %>%
  ggplot(aes(x=Incon.ConRT)) +
  geom_histogram(fill = "grey", color = "black") +
  ggtitle("Distribution of Incongruency Rt Effects (Outliers Removed)")
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



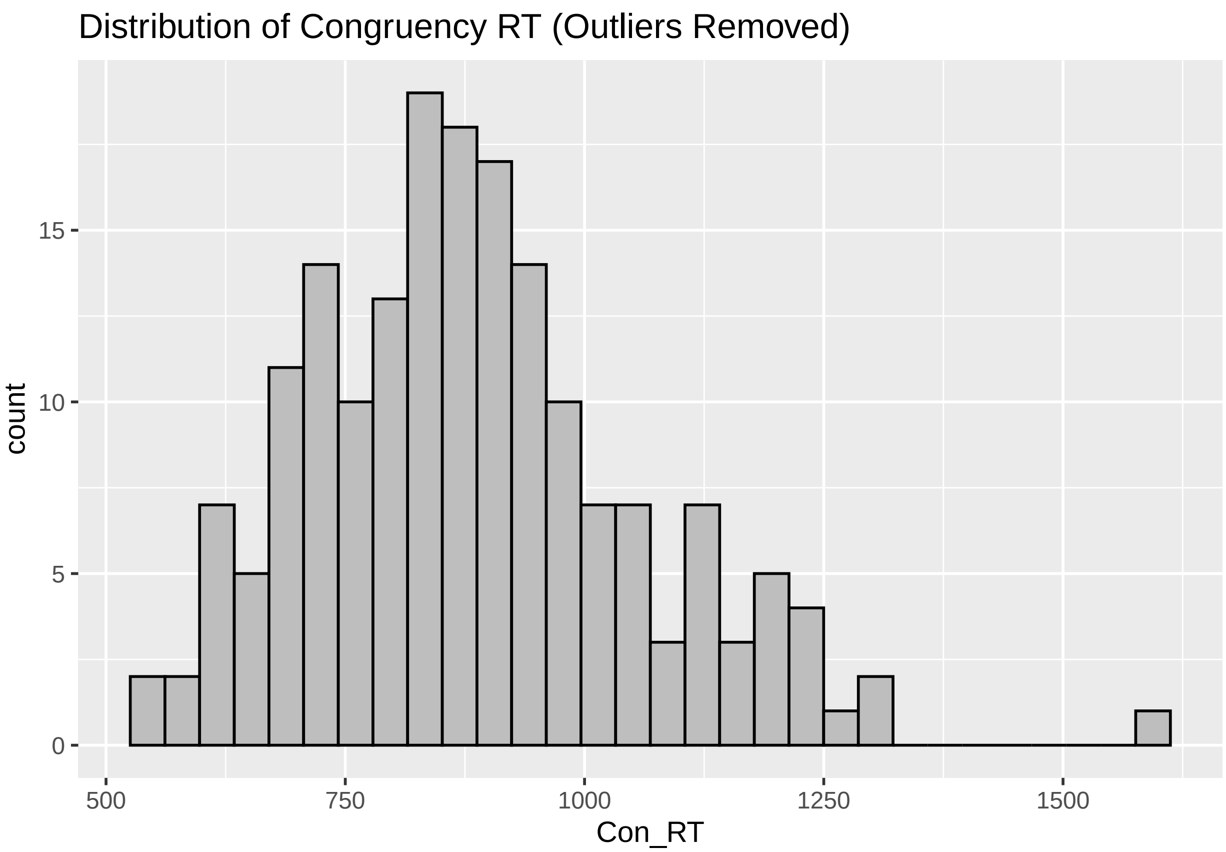
```
data_outliersremoved %>%
  ggplot(aes(x=InCon_RT)) +
  geom_histogram(fill = "grey", color = "black") +
  ggtitle("Distribution of Incongruency RT (Outliers Removed)")
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

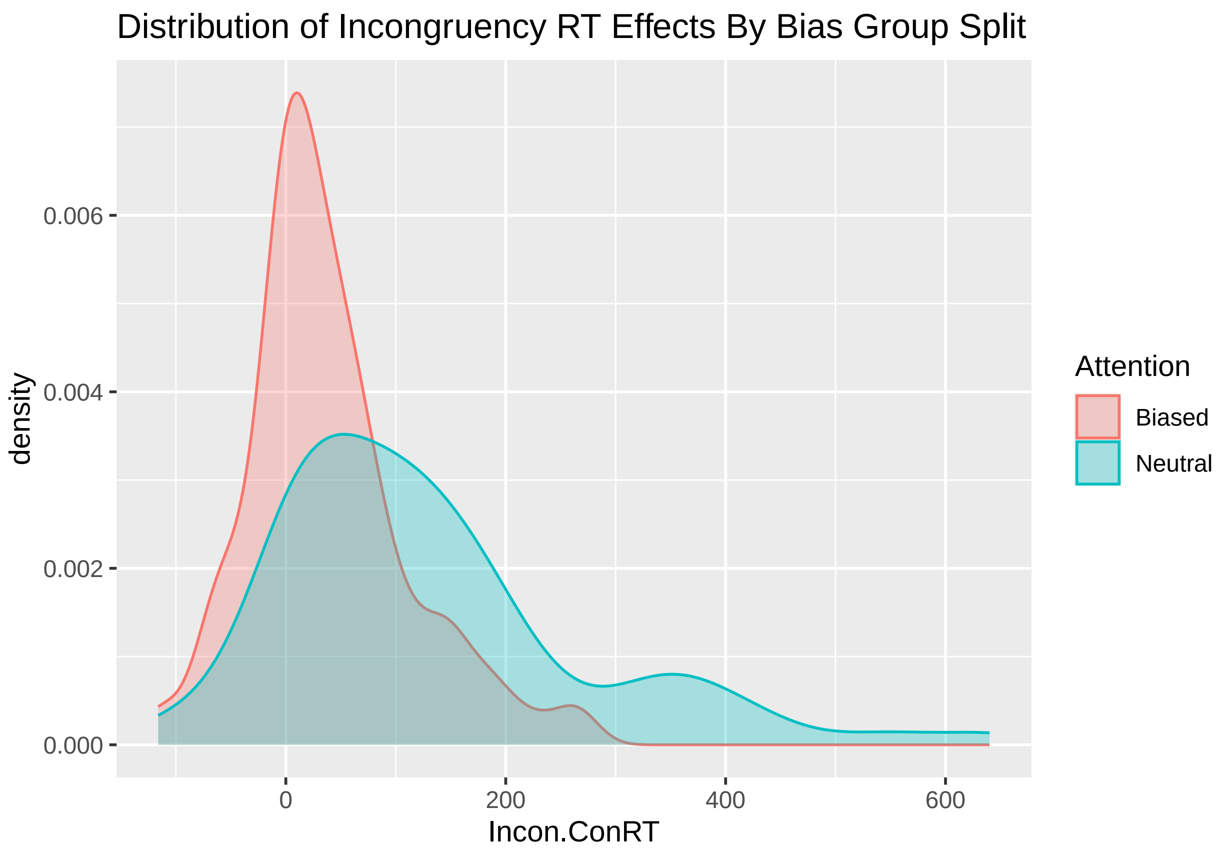


```
data_outliersremoved %>%
  ggplot(aes(x=Con_RT)) +
  geom_histogram(fill = "grey", color = "black") +
  ggtitle("Distribution of Congruency RT (Outliers Removed)")
```

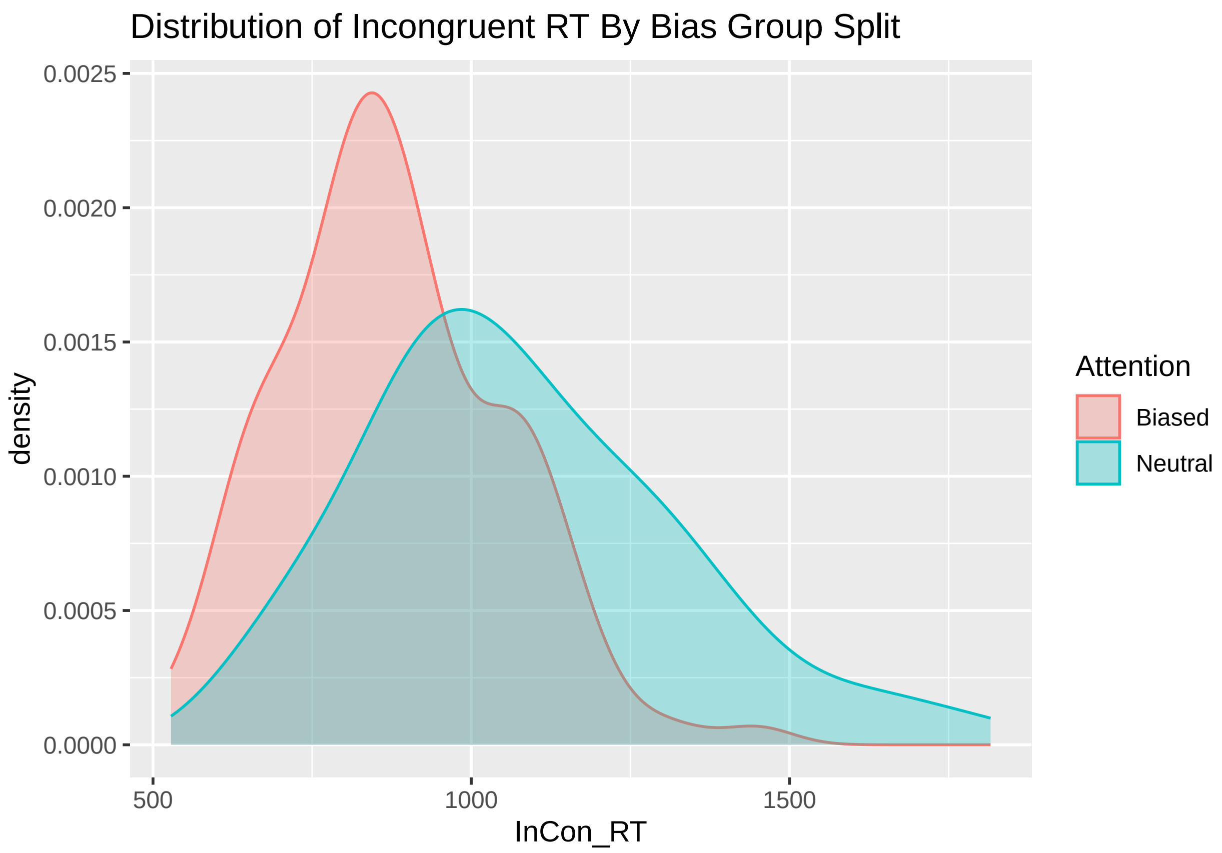
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



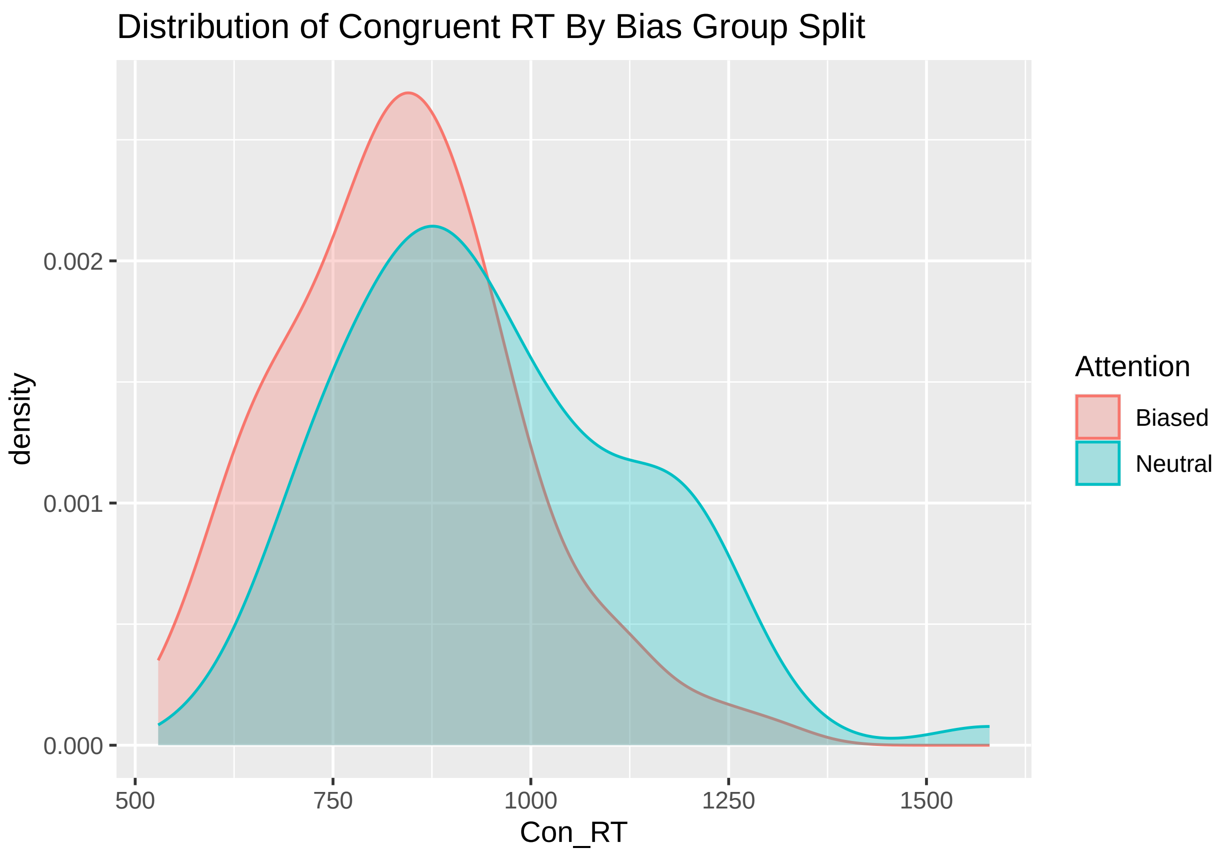
```
biased <- subset(data_outliersremoved, data_outliersremoved$BiasScore > 0.8 | data_outliersremoved$BiasS
neutral <- subset(data_outliersremoved, data_outliersremoved$BiasScore <= 0.8 & data_outliersremoved$Bia

data_grouped <- rbind(biased, neutral)

data_grouped %>%
  ggplot(aes(x = Incon.ConRT, fill = Attention, color = Attention))+
  geom_density(alpha = 0.3)+
  ggtitle("Distribution of Incongruency RT Effects By Bias Group Split")
```



```r
data_grouped %>%
  ggplot(aes(x = InCon_RT, fill = Attention, color = Attention))+
  geom_density(alpha = 0.3)+
  ggtitle("Distribution of Incongruent RT By Bias Group Split")
```



```
data_grouped %>%
  ggplot(aes(x = Con_RT, fill = Attention, color = Attention))+
  geom_density(alpha = 0.3)+
  ggtitle("Distribution of Congruent RT By Bias Group Split")
```



```
# Incon effect dist: Most people seem to fall between 0 and 250 ms incongruency effect- incongruent tri
```

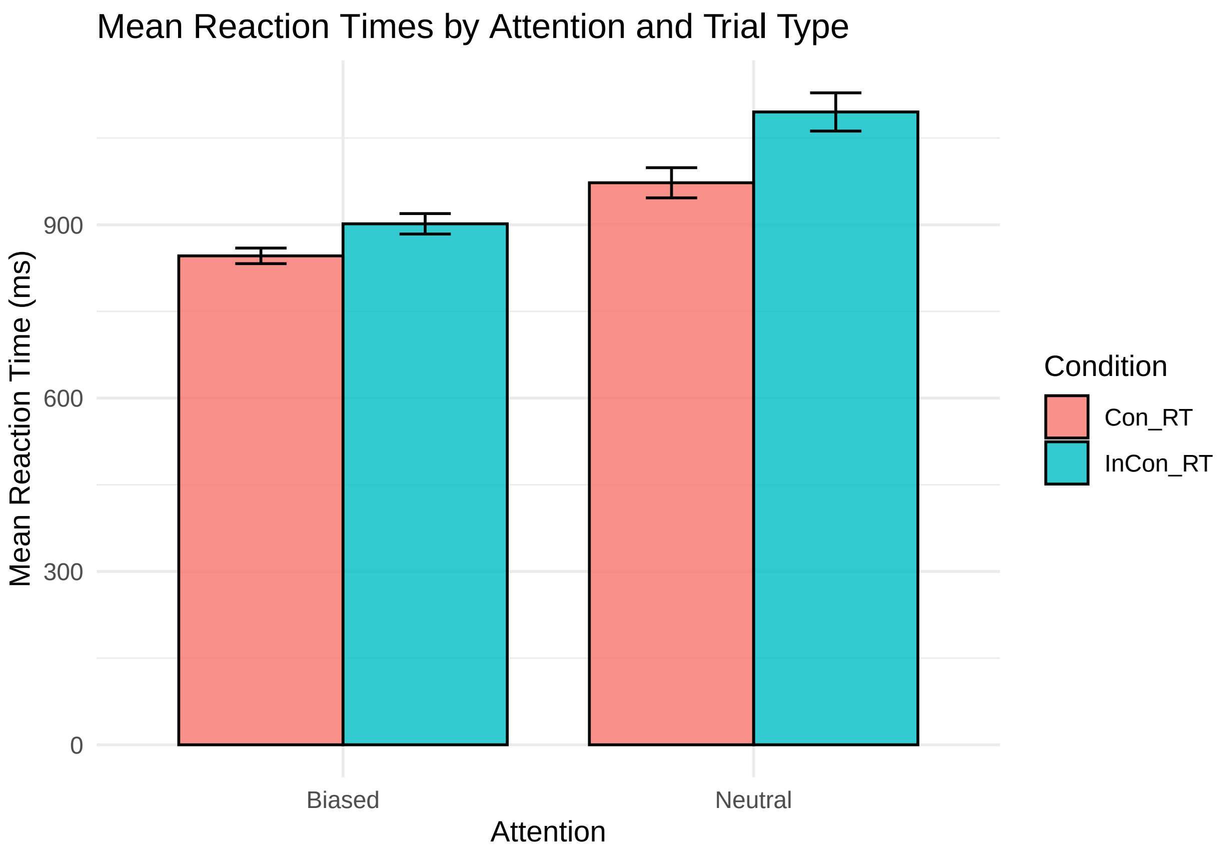
Reaction times by trial type and attention (InCon Outliers Removed)

```
data_outliersremoved$Attention <- ifelse(data_outliersremoved$BiasScore > 0.8 | data_outliersremoved$Bia
data_outliersremoved$IPS <- ifelse(data_outliersremoved$BiasScore > 0, "Verbal", "Visual")

# Reaction times
# Biased attender histograms and descriptive statistics
biased_data <- data_outliersremoved[data_outliersremoved$Attention == "Biased", ]
neutral_data <- data_outliersremoved[data_outliersremoved$Attention == "Neutral", ]

# Combine biased and neutral data
data_outliersremoved$Attention <- ifelse(data_outliersremoved$BiasScore > 0.8 | data_outliersremoved$Bia
biased_data <- data_outliersremoved[data_outliersremoved$Attention == "Biased", ]
combined_data <- rbind(biased_data, neutral_data)

# Calculate means by attention and trial type
library(dplyr)
library(tidyr)
library(ggplot2)

means <- combined_data %>%
  group_by(Attention) %>%
```

```r
  summarise(Con_RT = mean(Con_RT, na.rm = TRUE), InCon_RT = mean(InCon_RT, na.rm = TRUE)) %>%
  pivot_longer(cols = c(Con_RT, InCon_RT), names_to = "Trial_Type", values_to = "RT")

# Order factor levels for better plotting
means$Attention <- factor(means$Attention, levels = unique(means$Attention))

combined_long <- combined_data %>%
  select(Attention, InCon_RT, Con_RT) %>%
  pivot_longer(cols = c(InCon_RT, Con_RT), values_to = "RT", names_to = "Condition")

# Create error bars
se_sum <- combined_long %>%
  group_by(Attention, Condition) %>%
  summarise(
    sd = sd(RT, na.rm = TRUE),
    n = n(),
    mean = mean(RT, na.rm = TRUE)
  ) %>%
  mutate(se = sd/sqrt(n))
```

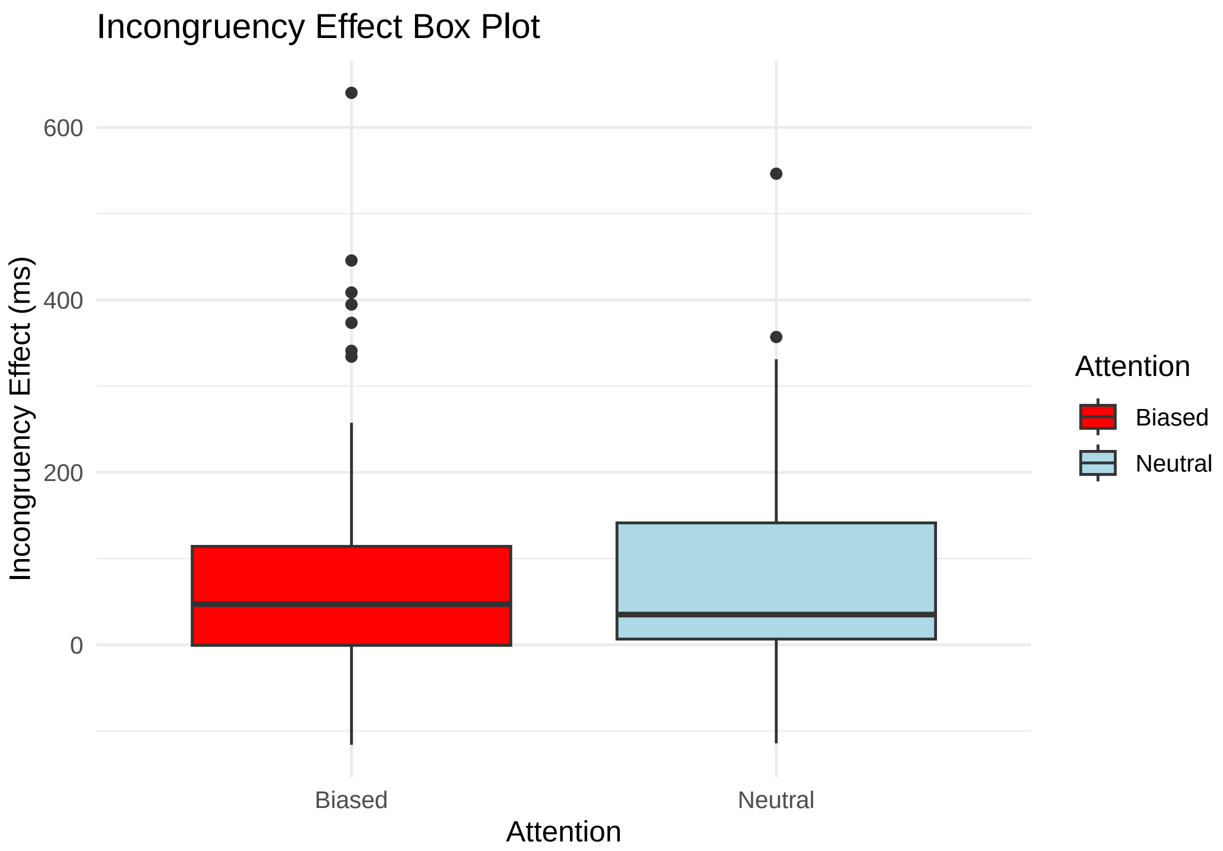
```
## 'summarise()' has grouped output by 'Attention'. You can override using the
## '.groups' argument.
```

```r
# Create bar plot
ggplot(se_sum, aes(x = Attention, y = mean, fill = Condition)) +
  geom_bar(position = position_dodge(0.8), stat = "identity", color = "black", size = 0.5, width = 0.8,
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), position = position_dodge(0.8), width = 0.25, c
  labs(title = "Mean Reaction Times by Attention and Trial Type",
       x = "Attention",
       y = "Mean Reaction Time (ms)") +
  theme_minimal()
```



## Incongruency Effect Box Plot (InCon Outliers Removed)

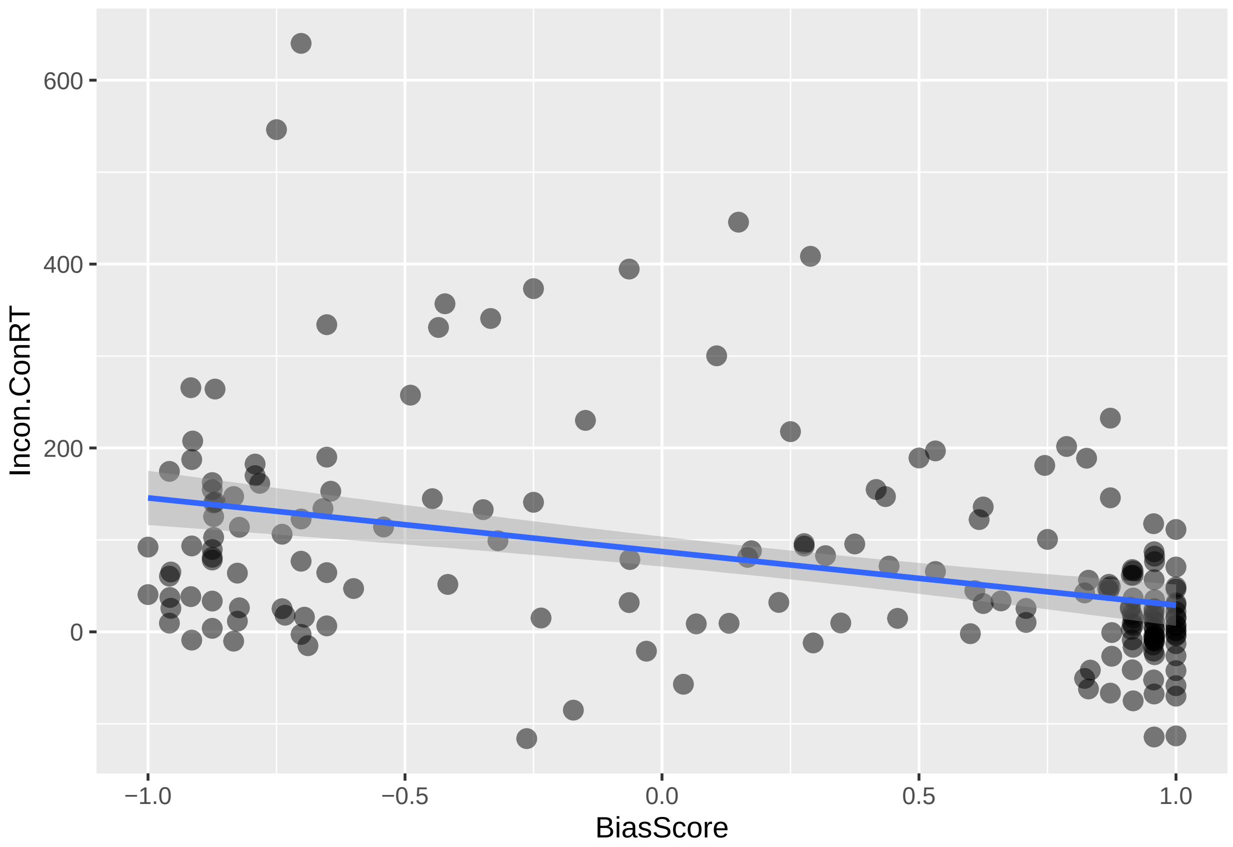
```
biased_data<- data_outliersremoved[data_outliersremoved$Attention == "Biased", ]
neutral_data<- data_outliersremoved[data_outliersremoved$Attention == "Neutral", ]

combined_data <- rbind(biased_data, neutral_data)
Incongruency_Effect_Data <- data_outliersremoved$IncongruencyEffect

# Create box plot
ggplot(combined_data, aes(x = Attention, y = Incongruency_Effect_Data, fill = Attention)) +
  geom_boxplot() +
  labs(
    title = "Incongruency Effect Box Plot",
    x = "Attention",
    y = "Incongruency Effect (ms)"
  ) +
  scale_fill_manual(values = c("red", "lightblue")) +  # Color for biased and neutral data
  theme_minimal()
```



## Bias Score and Incongruency Effect Correlation (InCon Outliers Removed)

```
data_outliersremoved %>%
  ggplot(aes(x = BiasScore, y = Incon.ConRT)) +
  geom_point(size = 3, alpha = 0.5, fill = "grey") +
  geom_smooth(method = "lm")
```

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



```
cor.test(data_outliersremoved$BiasScore, data_outliersremoved$Incon.ConRT)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data_outliersremoved$BiasScore and data_outliersremoved$Incon.ConRT
## t = -5.5861, df = 180, p-value = 8.46e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5017781 -0.2530704
## sample estimates:
##        cor
## -0.3843768
```

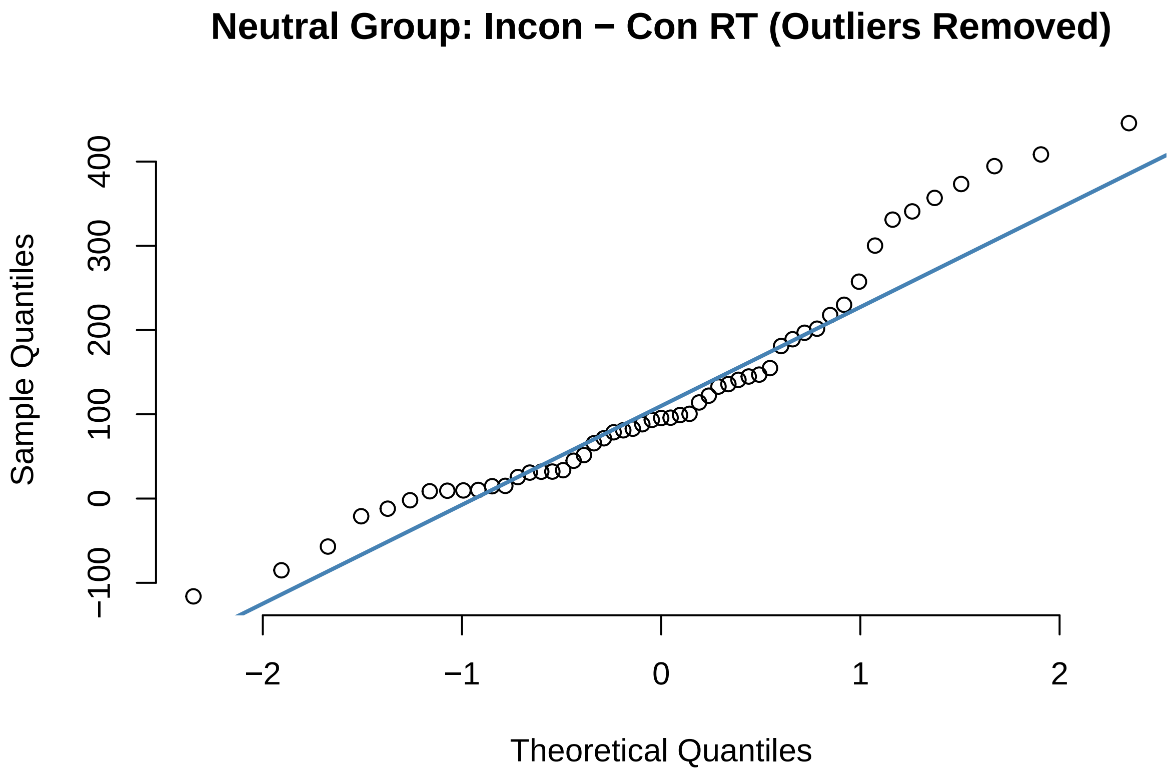
t-test (InCon Outliers Removed)

```
t.test(Incon.ConRT~Attention, data=data_outliersremoved)
```

```
##
##  Welch Two Sample t-test
##
## data:  Incon.ConRT by Attention
## t = -3.3291, df = 79.807, p-value = 0.00132
## alternative hypothesis: true difference in means between group Biased and group Neutral is not equal
## 95 percent confidence interval:
##  -107.18506  -26.98088
## sample estimates:
##  mean in group Biased mean in group Neutral
##              55.45634             122.53931
```

T-test Code - Removing Incon Outliers

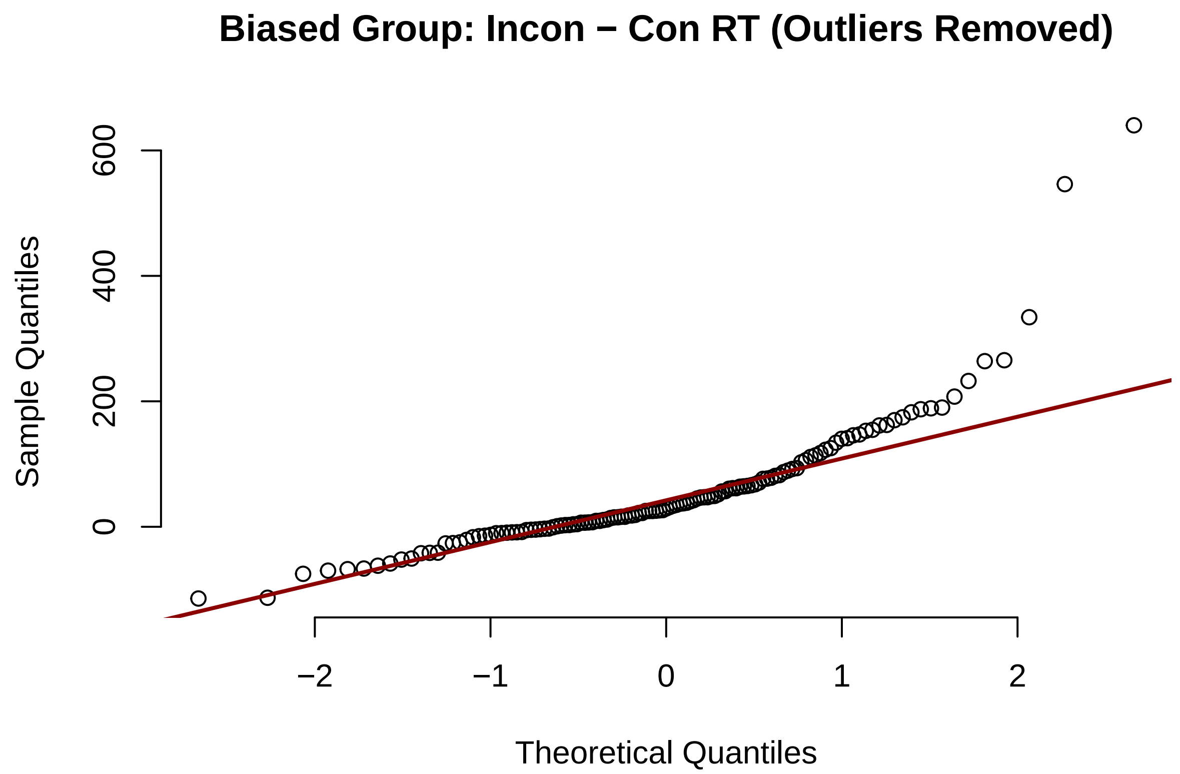
```
biased_data<- data_outliersremoved[data_outliersremoved$Attention == "Biased", ]
neutral_data<- data_outliersremoved[data_outliersremoved$Attention == "Neutral", ]

qqnorm(neutral_data$Incon.ConRT, pch = 1, frame = FALSE, main = "Neutral Group: Incon - Con RT (Outliers
qqline(neutral_data$Incon.ConRT, col = "steelblue", lwd = 2)
```



```
shapiro.test(neutral_data$Incon.ConRT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  neutral_data$Incon.ConRT
## W = 0.9365, p-value = 0.007351
```

```r
# Then test normality assumption in Biased Group
qqnorm(biased_data$Incon.ConRT, pch = 1, frame = FALSE, main = "Biased Group: Incon - Con RT (Outliers
qqline(biased_data$Incon.ConRT, col = "darkred", lwd = 2)
```



```r
shapiro.test(biased_data$Incon.ConRT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  biased_data$Incon.ConRT
## W = 0.80037, p-value = 5.813e-12
```

```r
# Check that the variance does not differ between groups
# Perform Levene's Test
print(leveneTest(Incon.ConRT ~ Attention, data = data_outliersremoved))
```

```
## 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  5.5793 0.01924 *
##       180
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

```r
# Conduct t-test with equal variance assumption
print(t.test(neutral_data$Incon.ConRT, biased_data$Incon.ConRT, var.equal = FALSE))
```

```
##
##  Welch Two Sample t-test
##
## data:  neutral_data$Incon.ConRT and biased_data$Incon.ConRT
## t = 3.3291, df = 79.807, p-value = 0.00132
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   26.98088 107.18506
## sample estimates:
## mean of x mean of y
## 122.53931  55.45634
```

Removing outliers- Congruent RT

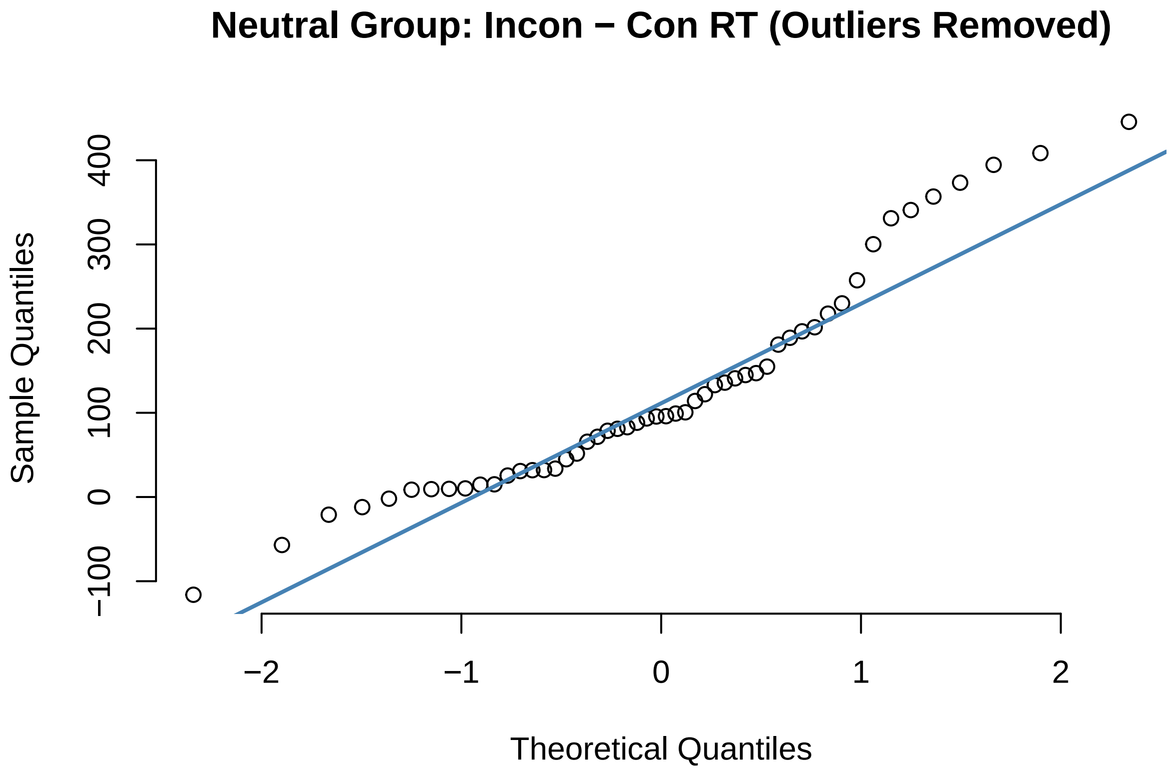
```r
Con_minus3SD <- mean(data_outliersremoved$Con_RT) - (3* sd(data_outliersremoved$Con_RT))
Con_plus3SD <- mean(data_outliersremoved$Con_RT) + (3* sd(data_outliersremoved$Con_RT))

data_outliersremoved <- data_outliersremoved %>%
  mutate(ConOutlier = Con_RT >= Con_plus3SD)

subset(data_outliersremoved, ConOutlier == TRUE)
```

```
##      Subject Age Sex IncorRespCount SymRespCount TxtRespCount  BiasScore
## 100 9449a552  57   2             19           17           12 -0.1724138
##       Con_RT InCon_RT Incon.ConRT Attention    IPS IncongruencyEffect
## 100 1579.715 1494.562   -85.15278   Neutral Visual          -85.15278
##     absBiasScore InconOutlier ConOutlier
## 100    0.1724138        FALSE       TRUE
```

```r
data_final <- subset(data_outliersremoved, ConOutlier == FALSE)
```

T-test with All Outliers Removed

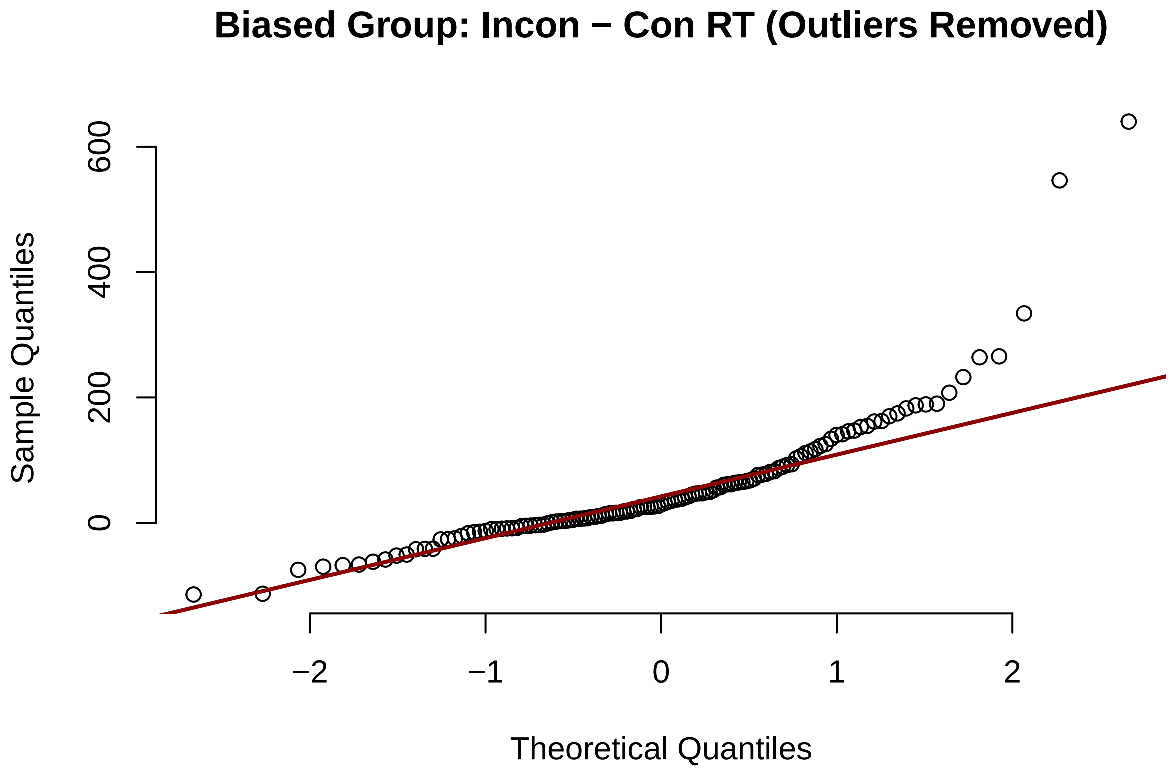
```r
biased_data <- data_final[data_final$Attention == "Biased", ]
neutral_data <- data_final[data_final$Attention == "Neutral", ]

qqnorm(neutral_data$Incon.ConRT, pch = 1, frame = FALSE, main = "Neutral Group: Incon - Con RT (Outlier
qqline(neutral_data$Incon.ConRT, col = "steelblue", lwd = 2)
```



```
shapiro.test(neutral_data$Incon.ConRT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  neutral_data$Incon.ConRT
## W = 0.92908, p-value = 0.004134
```

```
# Then test normality assumption in Biased Group
qqnorm(biased_data$Incon.ConRT, pch = 1, frame = FALSE, main = "Biased Group: Incon - Con RT (Outliers
qqline(biased_data$Incon.ConRT, col = "darkred", lwd = 2)
```



```
shapiro.test(biased_data$Incon.ConRT)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  biased_data$Incon.ConRT
## W = 0.80037, p-value = 5.813e-12
```

```
# Check that the variance does not differ between groups
# Perform Levene's Test
print(leveneTest(Incon.ConRT ~ Attention, data = data_final))
```

```
## 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  4.9887 0.02675 *
##       179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

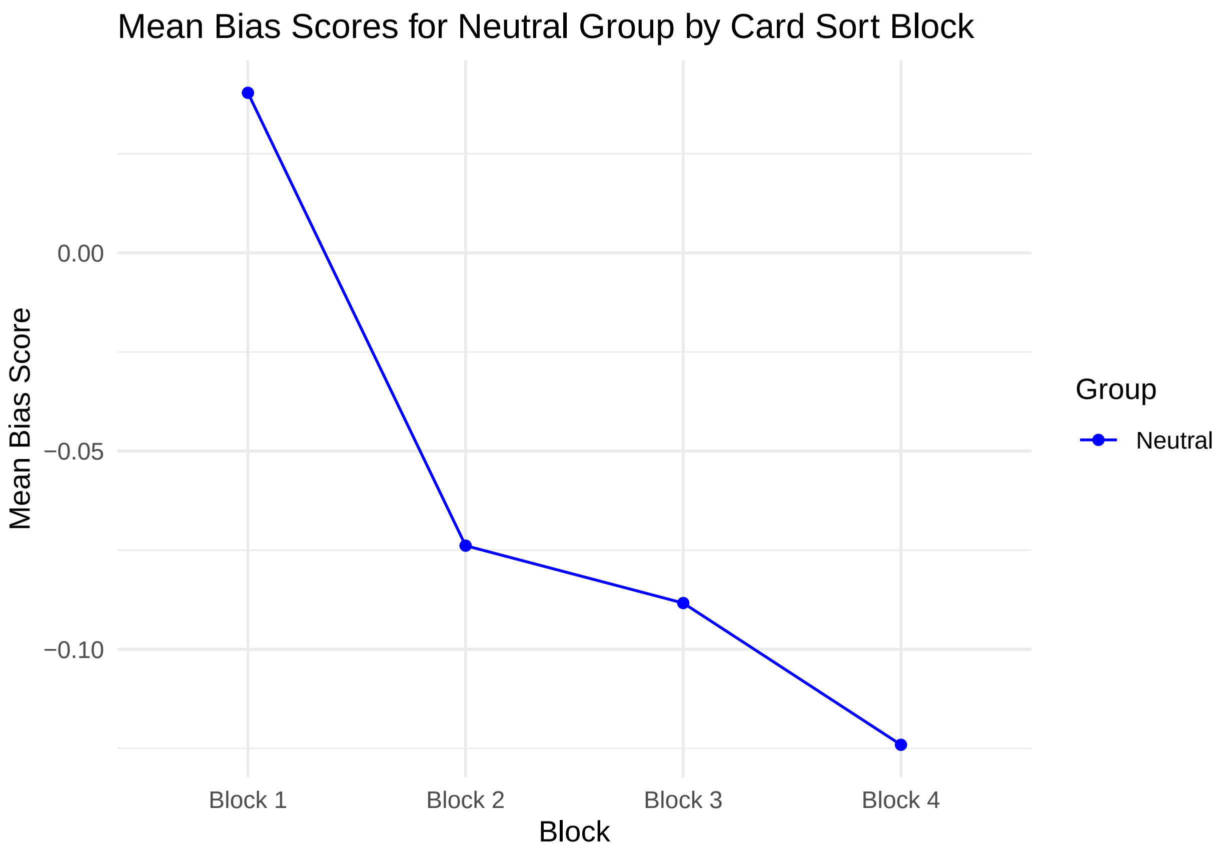
```r
# Conduct t-test with equal variance assumption
print(t.test(neutral_data$Incon.ConRT, biased_data$Incon.ConRT, var.equal = FALSE))
```

```
##
##  Welch Two Sample t-test
##
## data:  neutral_data$Incon.ConRT and biased_data$Incon.ConRT
## t = 3.5444, df = 78.651, p-value = 0.000666
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   31.15857 110.99552
## sample estimates:
## mean of x mean of y
## 126.53339  55.45634
```

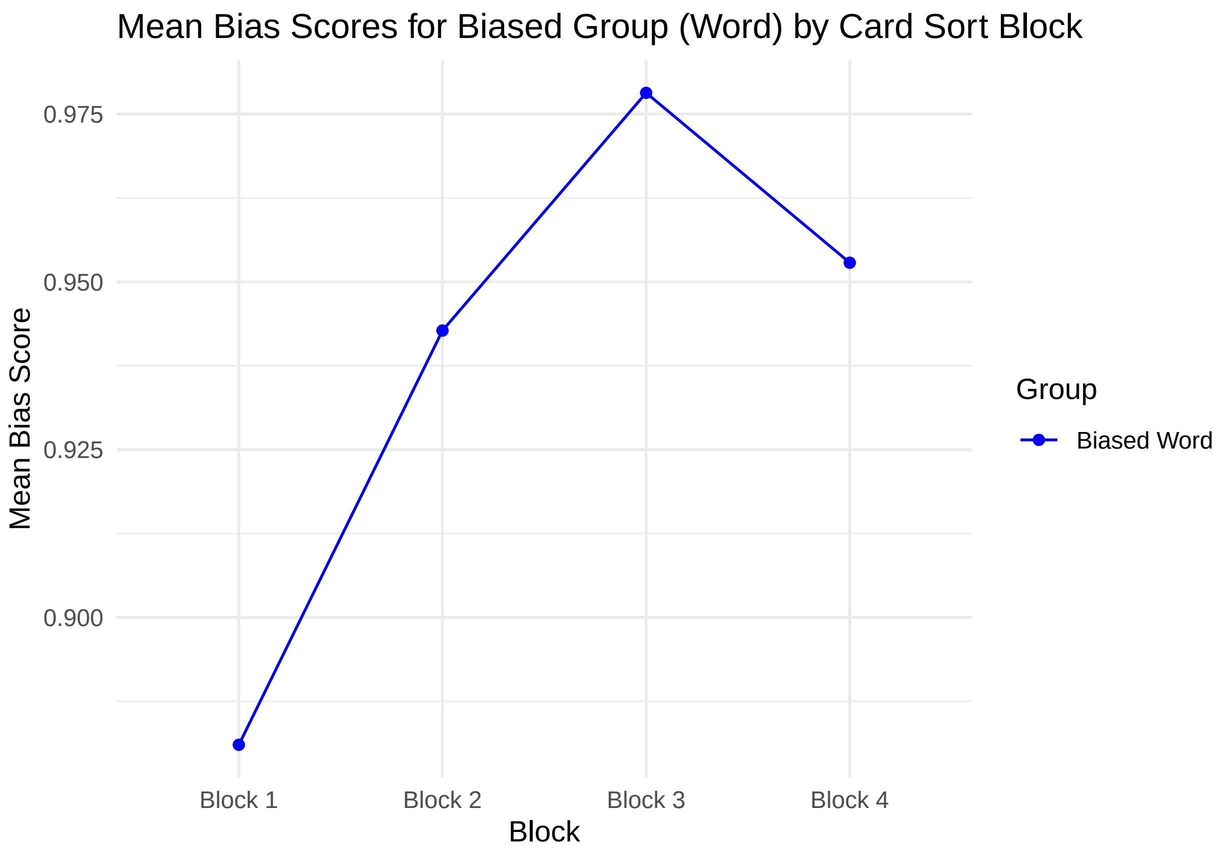
splitting blocks

block data

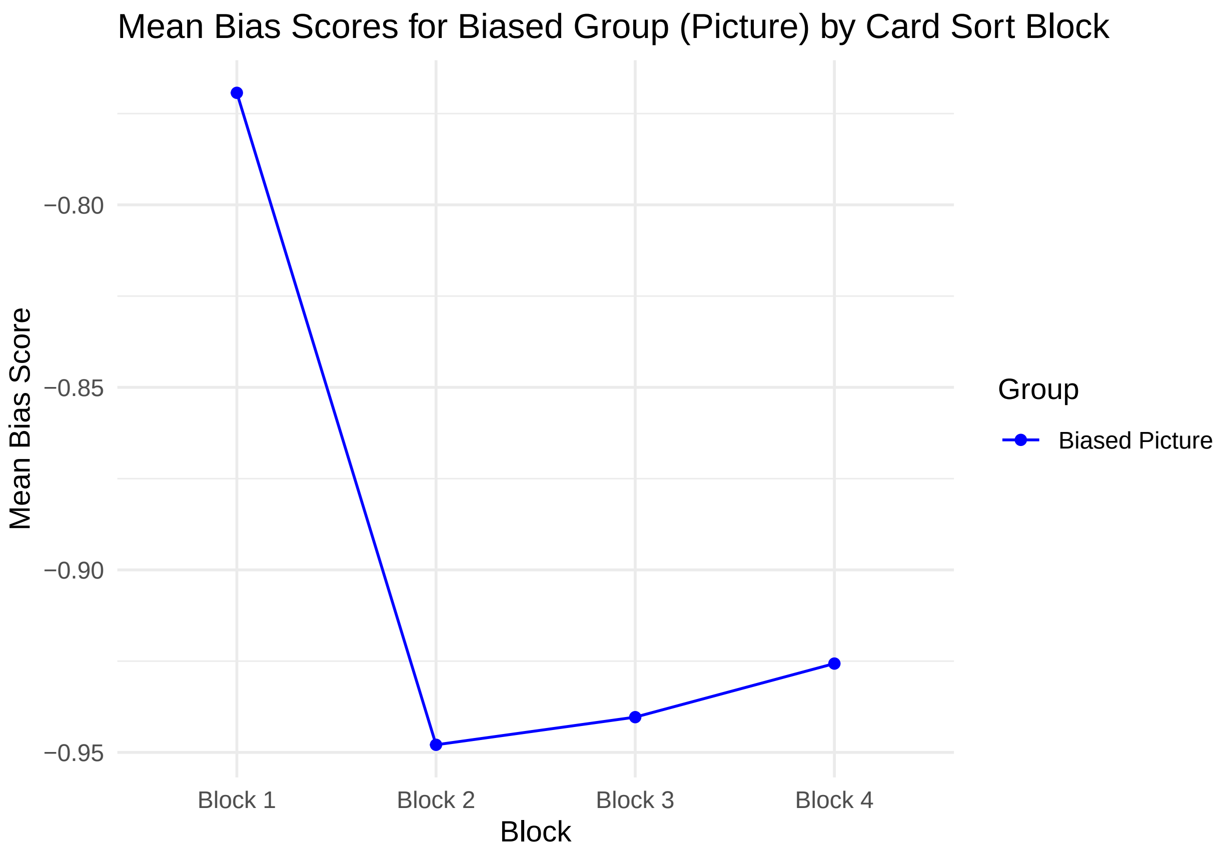
```r
block_data <- read.csv("CardSort Data.csv")

for_calc = data.frame(Subject = character(), block1_mean = numeric(), block2_mean = numeric(), block3_me

for (subject in unique(data$Subject)){
  subject_cols = block_data[block_data$SubjectNumber == subject, ]
  subject_block1 <- subject_cols[subject_cols$"Block" == "CardSort_Block1",]
  subject_block2 <- subject_cols[subject_cols$"Block" == "CardSort_Block2",]
  subject_block3 <- subject_cols[subject_cols$"Block" == "CardSort_Block3",]
  subject_block4 <- subject_cols[subject_cols$"Block" == "CardSort_Block4",]

  block1_mean <- mean(subject_block1$RT)
  block2_mean <- mean(subject_block2$RT)
  block3_mean <- mean(subject_block3$RT)
  block4_mean <- mean(subject_block4$RT)

  block1_incon_rows <- subset(subject_block1, Status == 2)
  block1_word <- max(subject_block1$TxtRespCount)
  block1_pic <- max(subject_block1$SymRespCount)
  block1_correct <- length(block1_incon_rows) - max(subject_block1$IncorrRespCount)
  block1_bias <- (block1_word - block1_pic) / block1_correct

  block2_incon_rows <- subset(subject_block2, Status == 2)
  block2_word <- max(subject_block2$TxtRespCount)-max(subject_block1$TxtRespCount)
  block2_pic <- max(subject_block2$SymRespCount)- max(subject_block1$SymRespCount)
  block2_correct <- length(block2_incon_rows) - (max(subject_block2$IncorrRespCount)-max(subject_block1$
  block2_bias <- (block2_word - block2_pic) / block2_correct

  block3_incon_rows <- subset(subject_block3, Status == 2)
  block3_word <- max(subject_block3$TxtRespCount)-max(subject_block2$TxtRespCount)
  block3_pic <- max(subject_block3$SymRespCount)-max(subject_block2$SymRespCount)
  block3_correct <- length(block3_incon_rows) - (max(subject_block3$IncorrRespCount)-max(subject_block2
  block3_bias <- (block3_word - block3_pic) / block3_correct

  block4_incon_rows <- subset(subject_block4, Status == 2)
  block4_word <- max(subject_block4$TxtRespCount)-max(subject_block3$TxtRespCount)
  block4_pic <- max(subject_block4$SymRespCount)-max(subject_block3$SymRespCount)
  block4_correct <- length(block4_incon_rows) - (max(subject_block4$IncorrRespCount)-max(subject_block3
  block4_bias <- (block4_word - block4_pic) / block4_correct

  new_row1 <- data.frame(Subject = subject, block1_mean = block1_mean, block2_mean =  block2_mean, bloc

  for_calc<- rbind(for_calc, new_row1)
}

data <- cbind(data, for_calc)


# mean rts for each block biased v neutral
biased_word <- subset(data, data$BiasScore > 0.8)
biased_picture <- subset(data, data$BiasScore < -0.8)
biased <- subset(data, data$BiasScore > 0.8 | data$BiasScore < -0.8)
neutral <- subset(data, data$BiasScore <= 0.8 & data$BiasScore >= -0.8)

biased_means <- colMeans(biased[, c("block1_bias", "block2_bias", "block3_bias", "block4_bias")], na.rm
# Calculate means for neutral group
neutral_means <- colMeans(neutral[, c("block1_bias", "block2_bias", "block3_bias", "block4_bias")], na.
biased_word_means <- colMeans(biased_word[, c("block1_bias", "block2_bias", "block3_bias", "block4_bias
biased_picture_means <- colMeans(biased_picture[, c("block1_bias", "block2_bias", "block3_bias", "block

means_df <- data.frame(
  Block = rep(c("Block 1", "Block 2", "Block 3", "Block 4"), 2),
  MeanBias = c(biased_means, neutral_means),
  Group = rep(c("Biased", "Neutral"), each = 4)
)

# subset of neutral
neutral_means_df <- subset(means_df, Group == "Neutral")

# Plotting neutral blocks from means_df
ggplot(neutral_means_df, aes(x = Block, y = MeanBias, group = Group, color = Group)) +
  geom_line() +
  geom_point() +
  labs(title = "Mean Bias Scores for Neutral Group by Card Sort Block", x = "Block", y = "Mean Bias Sco
  scale_color_manual(values = c("Neutral" = "blue")) +
  theme_minimal()
```



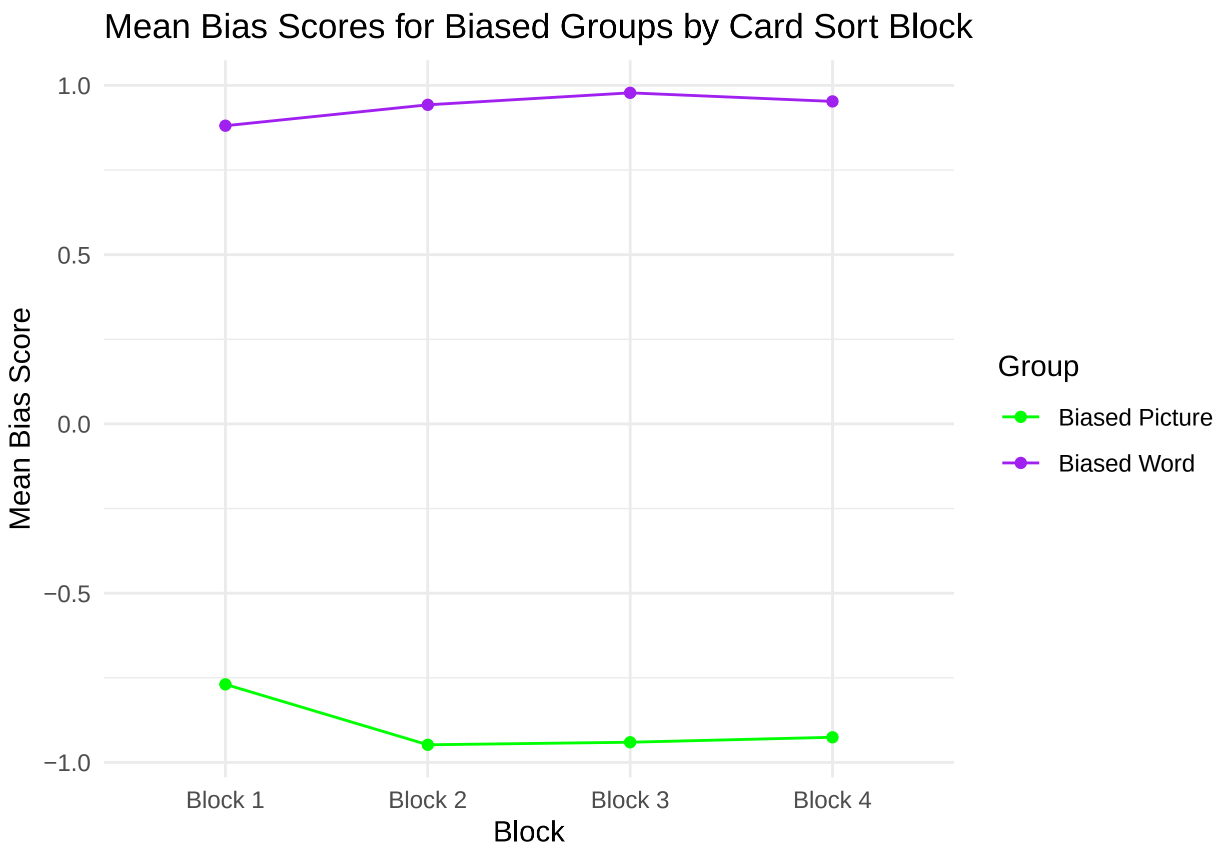
```r
# Creating means_df for two biased groups and one neutral group
biased_means_df <- data.frame(
  Block = rep(c("Block 1", "Block 2", "Block 3", "Block 4"), 3),
  MeanBias = c(biased_word_means, biased_picture_means, neutral_means),
  Group = rep(c("Biased Word", "Biased Picture", "Neutral"), each = 4)
)

#subset of word biased (positive)
biased_word_means_df <- subset(biased_means_df, Group == "Biased Word")
biased_picture_means_df <- subset(biased_means_df, Group == "Biased Picture")

ggplot(biased_word_means_df, aes(x = Block, y = MeanBias, group = Group, color = Group)) +
  geom_line() +
  geom_point() +
  labs(title = "Mean Bias Scores for Biased Group (Word) by Card Sort Block", x = "Block", y = "Mean Bia
  scale_color_manual(values = c("Biased Word" = "blue")) +
  theme_minimal()
```



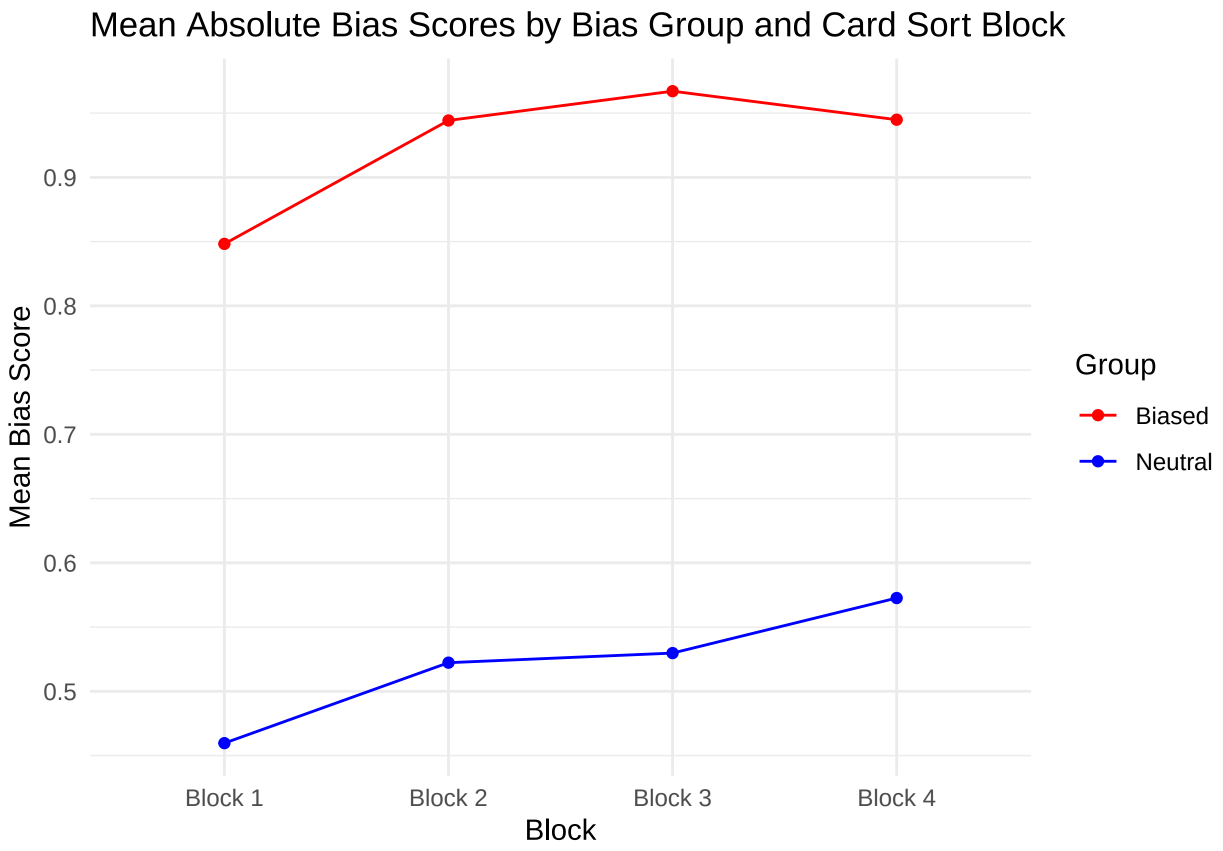
```
#subset of picture biased (negative)
ggplot(biased_picture_means_df, aes(x = Block, y = MeanBias, group = Group, color = Group)) +
  geom_line() +
  geom_point() +
  labs(title = "Mean Bias Scores for Biased Group (Picture) by Card Sort Block", x = "Block", y = "Mean
  scale_color_manual(values = c("Biased Picture" = "blue")) +
  theme_minimal()
```



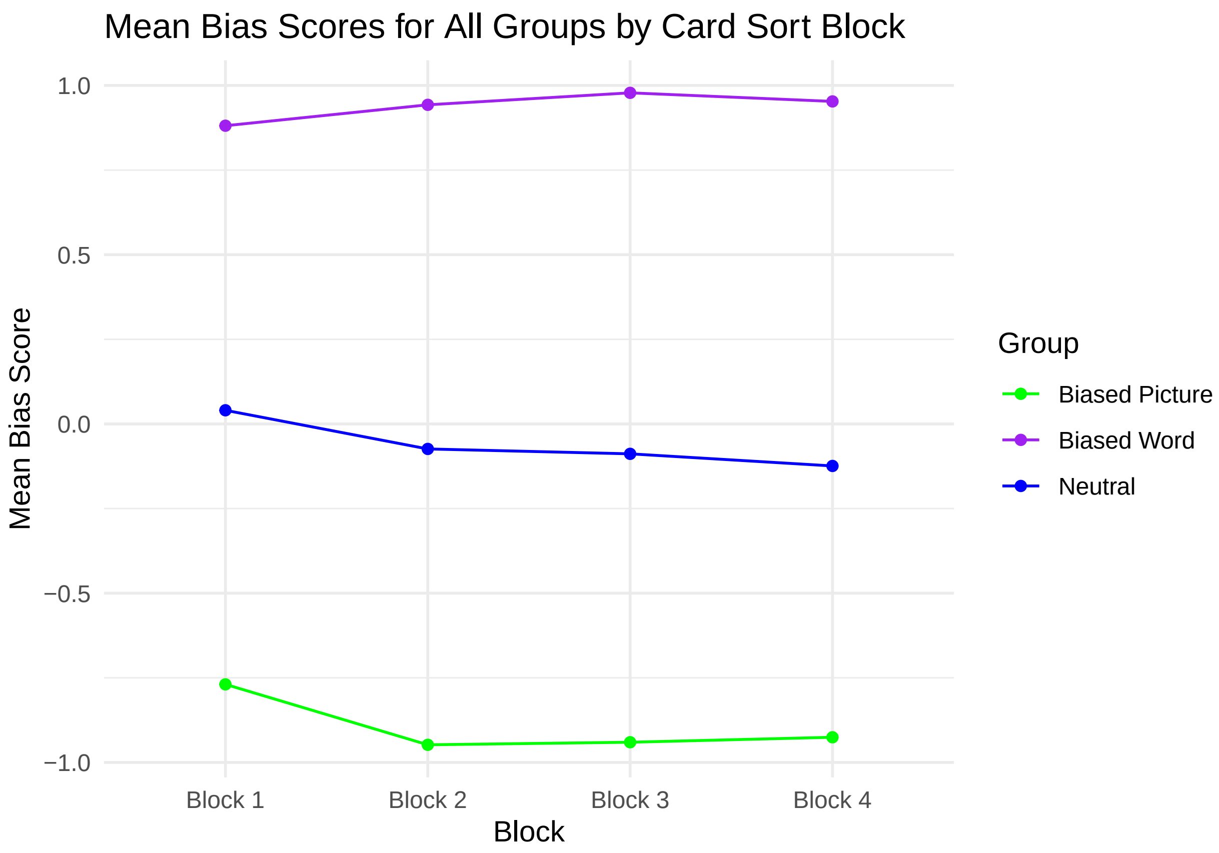
```
ggplot() +
  geom_line(data = biased_word_means_df, aes(x = Block, y = MeanBias, group = Group, color = "Biased Wo
  geom_point(data = biased_word_means_df, aes(x = Block, y = MeanBias, color = "Biased Word")) +
  geom_line(data = biased_picture_means_df, aes(x = Block, y = MeanBias, group = Group, color = "Biased
  geom_point(data = biased_picture_means_df, aes(x = Block, y = MeanBias, color = "Biased Picture")) +
  labs(title = "Mean Bias Scores for Biased Groups by Card Sort Block", x = "Block", y = "Mean Bias Sco
  scale_color_manual(name = "Group", values = c("Biased Word" = "purple", "Biased Picture" = "green"))
  theme_minimal()
```



```
biased$block1_bias <- abs(biased$block1_bias)
biased$block2_bias <- abs(biased$block2_bias)
biased$block3_bias <- abs(biased$block3_bias)
biased$block4_bias <- abs(biased$block4_bias)

neutral$block1_bias <- abs(neutral$block1_bias)
neutral$block2_bias <- abs(neutral$block2_bias)
neutral$block3_bias <- abs(neutral$block3_bias)
neutral$block4_bias <- abs(neutral$block4_bias)

biased_means <- colMeans(biased[, c("block1_bias", "block2_bias", "block3_bias", "block4_bias")], na.rm

# Calculate means for neutral group
neutral_means <- colMeans(neutral[, c("block1_bias", "block2_bias", "block3_bias", "block4_bias")], na.

# Combine the means into a new data frame for plotting
means_df <- data.frame(
  Block = rep(c("Block 1", "Block 2", "Block 3", "Block 4"), 2),
  MeanBias = c(biased_means, neutral_means),
  Group = rep(c("Biased", "Neutral"), each = 4)
)


ggplot(means_df, aes(x = Block, y = MeanBias, color = Group, group = Group)) +
  geom_line() +  # Add lines
  geom_point() + # Add points
  labs(title = "Mean Absolute Bias Scores by Bias Group and Card Sort Block", x = "Block", y = "Mean Bia
  scale_color_manual(values = c("Biased" = "red", "Neutral" = "blue")) +
  theme_minimal()
```



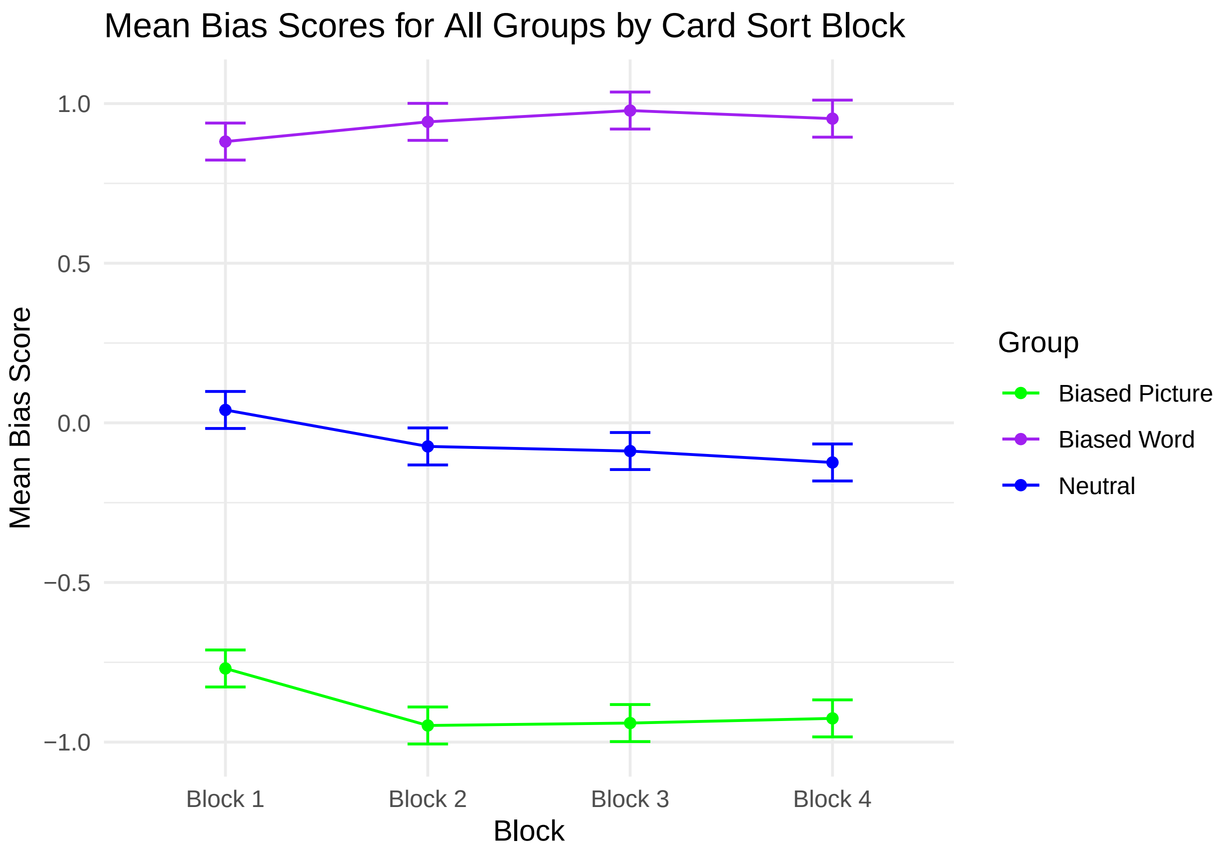
```
# Create the combined plot
combined_plot <- ggplot() +
  geom_line(data = neutral_means_df, aes(x = Block, y = MeanBias, group = Group, color = "Neutral")) +
  geom_point(data = neutral_means_df, aes(x = Block, y = MeanBias, color = "Neutral")) +
  geom_line(data = biased_word_means_df, aes(x = Block, y = MeanBias, group = Group, color = "Biased Wo
  geom_point(data = biased_word_means_df, aes(x = Block, y = MeanBias, color = "Biased Word")) +
  geom_line(data = biased_picture_means_df, aes(x = Block, y = MeanBias, group = Group, color = "Biased
  geom_point(data = biased_picture_means_df, aes(x = Block, y = MeanBias, color = "Biased Picture")) +
  labs(title = "Mean Bias Scores for All Groups by Card Sort Block", x = "Block", y = "Mean Bias Score"
  scale_color_manual(name = "Group", values = c("Neutral" = "blue", "Biased Word" = "purple", "Biased P
  theme_minimal()

# Display the combined plot
print(combined_plot)
```



```
# trying to make it with error bars
# Step 1: Calculate descriptive statistics and store in a variable
biased_word <- subset(data, data$BiasScore > 0.8)
biased_picture <- subset(data, data$BiasScore < -0.8)
neutral <- subset(data, data$BiasScore <= 0.8 & data$BiasScore >= -0.8)

#biased word cleaning
cleaned_biased_word <- biased_word %>%
  select(-c(Subject.1))

descriptives_biased_word <- cleaned_biased_word %>%
  select(-c(Subject, Sex, InconOutlier,Attention)) %>%
  psych::describe()

# Convert the 'desc_df' to a regular dataframe
bw_descdf <- as.data.frame(descriptives_biased_word)

#biased picture cleaning

cleaned_biased_pic <- biased_picture %>%
  select(-c(Subject.1))

descriptives_biased_pic <- cleaned_biased_pic %>%
  select(-c(Subject, Sex, InconOutlier,Attention)) %>%
  psych::describe()
```

```r
# Convert the 'desc_df' to a regular dataframe
bp_descdf <- as.data.frame(descriptives_biased_pic)

#cleaning neutral
cleaned_neutral <- neutral %>%
  select(-c(block1_mean, block2_mean, block3_mean, block4_mean,
            block1_bias, block2_bias, block3_bias, block4_bias, Subject))

descriptives_neutral <- cleaned_neutral %>%
  select(-c (Sex, InconOutlier,Attention)) %>%
  psych::describe()

# Convert the 'desc_df' to a regular dataframe
n_descdf <- as.data.frame(descriptives_neutral)

# Calculate SE for each block for biased word data
bw_descdf$se <- bw_descdf$sd / sqrt(bw_descdf$n)

# Calculate SE for each block for biased picture data
bp_descdf$se <- bp_descdf$sd / sqrt(bp_descdf$n)

# Calculate SE for each block for neutral data
n_descdf$se <- n_descdf$sd / sqrt(n_descdf$n)

neutral_means_df$se <- NA  # Create a new column for SE

# Assuming order of SEs in 'n_descdf' corresponds to the blocks in 'neutral_means_df'
neutral_means_df$se[neutral_means_df$Block == "Block 1"] <- n_descdf$se[5]  # BiasScore SE for Block 1
neutral_means_df$se[neutral_means_df$Block == "Block 2"] <- n_descdf$se[5]  # BiasScore SE for Block 2
neutral_means_df$se[neutral_means_df$Block == "Block 3"] <- n_descdf$se[5]  # BiasScore SE for Block 3
neutral_means_df$se[neutral_means_df$Block == "Block 4"] <- n_descdf$se[5]  # BiasScore SE for Block 4


# Adding error bars to the plot
se<- n_descdf$se[5]
combined_plot <- ggplot() +
  geom_line(data = neutral_means_df, aes(x = Block, y = MeanBias, group = Group, color = "Neutral")) +
  geom_point(data = neutral_means_df, aes(x = Block, y = MeanBias, color = "Neutral")) +
  geom_errorbar(data = neutral_means_df, aes(x = Block, ymin = MeanBias - se, ymax = MeanBias + se, colo

  geom_line(data = biased_word_means_df, aes(x = Block, y = MeanBias, group = Group, color = "Biased Wor
  geom_point(data = biased_word_means_df, aes(x = Block, y = MeanBias, color = "Biased Word")) +
  geom_errorbar(data = biased_word_means_df, aes(x = Block, ymin = MeanBias - se, ymax = MeanBias + se,

  geom_line(data = biased_picture_means_df, aes(x = Block, y = MeanBias, group = Group, color = "Biased
  geom_point(data = biased_picture_means_df, aes(x = Block, y = MeanBias, color = "Biased Picture")) +
  geom_errorbar(data = biased_picture_means_df, aes(x = Block, ymin = MeanBias - se, ymax = MeanBias +

  labs(title = "Mean Bias Scores for All Groups by Card Sort Block", x = "Block", y = "Mean Bias Score"
  scale_color_manual(name = "Group", values = c("Neutral" = "blue", "Biased Word" = "purple", "Biased Pi
  theme_minimal()

# Display the combined plot
```

```r
print(combined_plot)
```



```r
# Calculating changes between blocks for both groups
change1_bias = biased_means["block2_bias"] - biased_means["block1_bias"]
change2_bias = biased_means["block4_bias"] - biased_means["block3_bias"]

change1_neutral = neutral_means["block2_bias"] - neutral_means["block1_bias"]
change2_neutral = neutral_means["block4_bias"] - neutral_means["block3_bias"]

changes_bias = c(change1_bias, change2_bias)
changes_neutral = c(change1_neutral, change2_neutral)

t_test_all_changes = t.test(changes_bias, changes_neutral, alternative = "two.sided", var.equal = TRUE)

# Print the results
print(t_test_all_changes)
```

```
##
##  Two Sample t-test
##
## data:  changes_bias and changes_neutral
## t = -0.26396, df = 2, p-value = 0.8165
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2736995  0.2420583
## sample estimates:
##  mean of x  mean of y
## 0.03691966 0.05274028
```

```
biased$block_diff <- abs(biased$block4_bias) - abs(biased$block1_bias)
neutral$block_diff <- abs(neutral$block4_bias) - abs(neutral$block1_bias)

# Perform t-test
t_test_result <- t.test(biased$block_diff, neutral$block_diff, var.equal = FALSE)

# Display the result
t_test_result
```

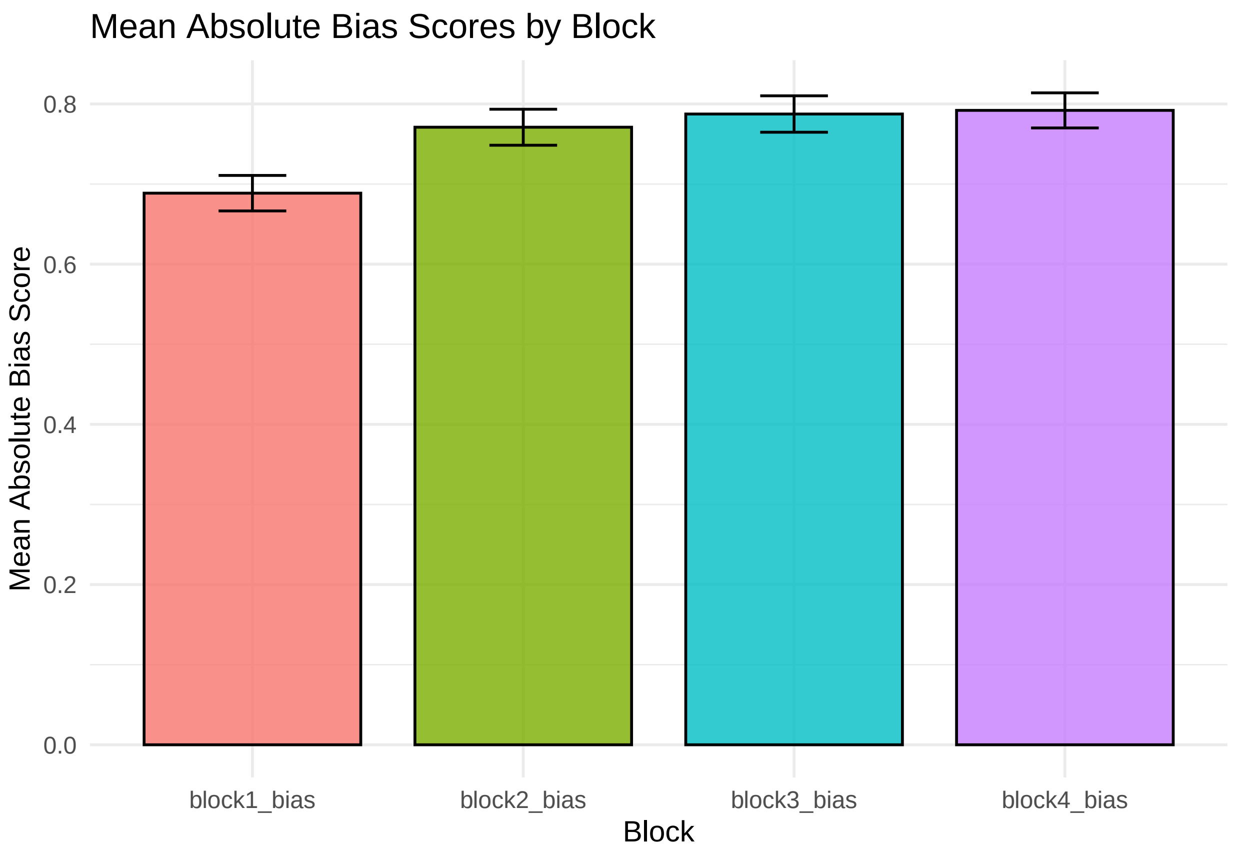
```
##
##  Welch Two Sample t-test
##
## data:  biased$block_diff and neutral$block_diff
## t = -0.33269, df = 102.15, p-value = 0.7401
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1135304  0.0809159
## sample estimates:
##  mean of x  mean of y
## 0.09663609 0.11294334
```

```
biased_ttest_b1 <- abs(biased$block1_bias)
neutral_ttest_b2 <- abs(neutral$block1_bias)

# Perform t-test
t_test_result <- t.test(biased_ttest_b1, neutral_ttest_b2, var.equal = FALSE)

# Display the result
t_test_result
```

```
##
##  Welch Two Sample t-test
##
## data:  biased_ttest_b1 and neutral_ttest_b2
## t = 10.214, df = 110.47, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3131712 0.4639433
## sample estimates:
## mean of x mean of y
## 0.8482254 0.4596681
```

```
#convert data frame to long format
data <- data[, !duplicated(colnames(data))]
data <- data %>%
  mutate(Attention = ifelse(BiasScore > 0.8 | BiasScore < -0.8, "biased", "neutral"))

   long_block_data <- data %>%
    gather(key = "Condition", value = "BiasScore", block1_bias:block4_bias) %>%
    mutate(Condition = gsub("_bias", "", Condition)) %>%
    mutate(BiasScore = abs(BiasScore)) %>%
    select(Attention, Subject, Condition, BiasScore)


#block_aov = anova_test(data=long_block_data, dv=BiasScore, wid=Subject, between=Attention, within = Co
#print(block_aov)

data <- data[, !duplicated(colnames(data))]
data <- data %>%
  mutate(Attention = ifelse(BiasScore > 0.8 | BiasScore < -0.8, "biased", "neutral"))

   long_block_data <- data %>%
    gather(key = "Condition", value = "BiasScore", block1_bias:block2_bias) %>%
    mutate(Condition = gsub("_bias", "", Condition)) %>%
    mutate(BiasScore = abs(BiasScore)) %>%
    select(Attention, Subject, Condition, BiasScore)



#block_aov = anova_test(data=long_block_data, dv=BiasScore, wid=Subject, between=Attention, within = Co
#print(block_aov)
```

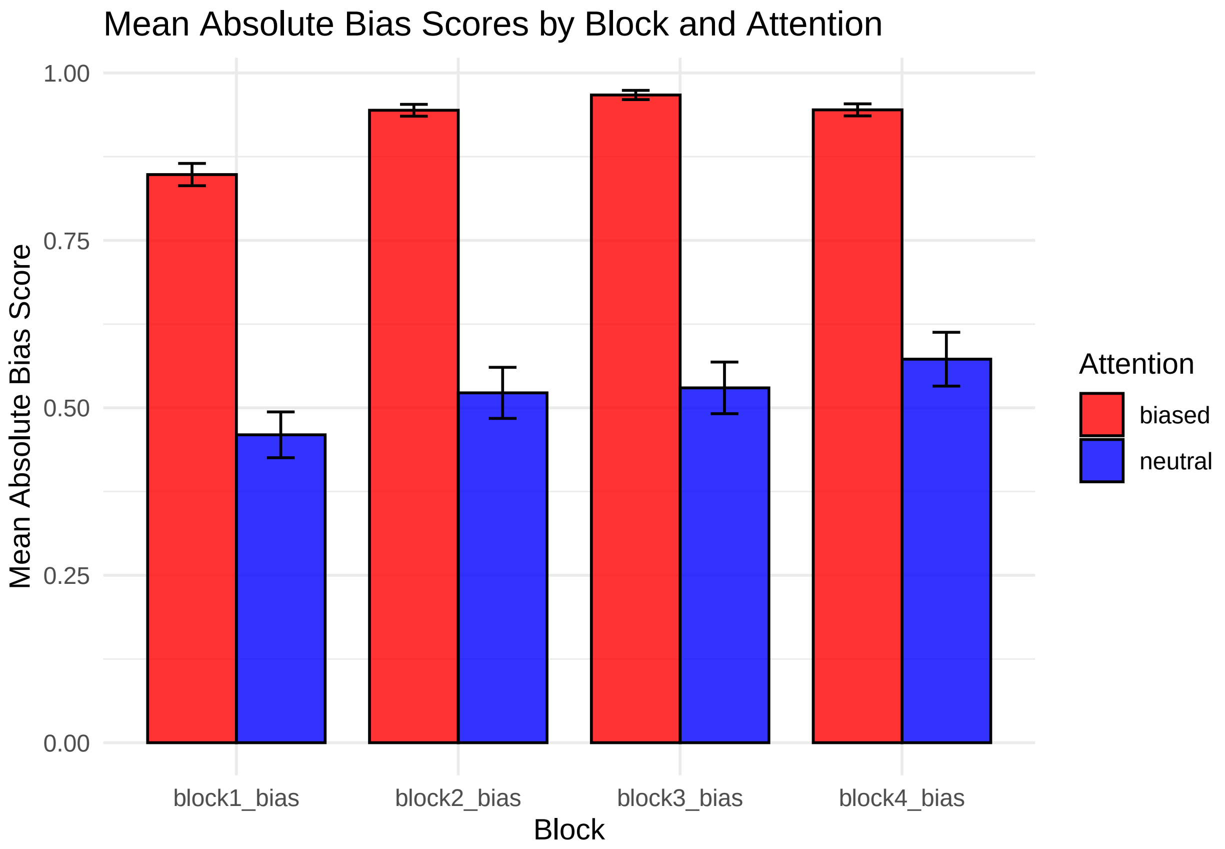
```
df_unique<-data
data_long <- df_unique %>%
  select(Subject, block1_bias, block2_bias, block3_bias, block4_bias) %>%
  pivot_longer(cols = starts_with("block"), names_to = "Block", values_to = "BiasScore") %>%
  mutate(absBiasScore = abs(BiasScore))

# Calculate mean and standard error for each block
se_sum <- data_long %>%
  group_by(Block) %>%
  summarise(
    mean = mean(absBiasScore, na.rm = TRUE),
    sd = sd(absBiasScore, na.rm = TRUE),
    n = n()
  ) %>%
  mutate(se = sd/sqrt(n))

# Plotting
ggplot(se_sum, aes(x = Block, y = mean, fill = Block)) +
  geom_bar(position = position_dodge(0.8), stat = "identity", color = "black", size = 0.5, width = 0.8,
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), position = position_dodge(0.8), width = 0.25, c
  labs(title = "Mean Absolute Bias Scores by Block",
       x = "Block",
       y = "Mean Absolute Bias Score") +
  theme_minimal() +
  theme(legend.position = "none")
```



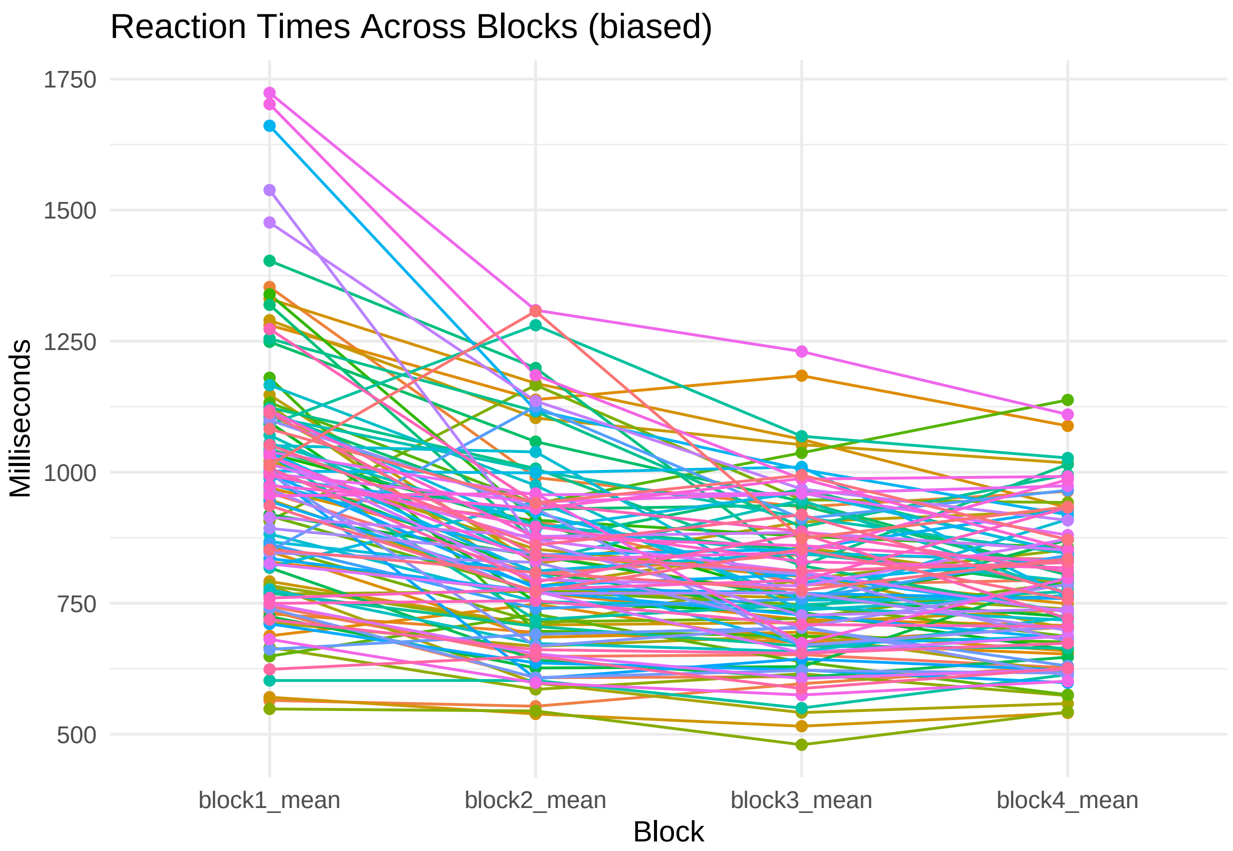
```
data_long <- df_unique %>%
  select(Subject, Attention, block1_bias, block2_bias, block3_bias, block4_bias) %>%
  pivot_longer(cols = starts_with("block"), names_to = "Block", values_to = "BiasScore") %>%
  mutate(absBiasScore = abs(BiasScore))

# Calculate mean and standard error for each block and attention group
se_sum <- data_long %>%
  group_by(Attention, Block) %>%
  summarise(
    mean = mean(absBiasScore, na.rm = TRUE),
    sd = sd(absBiasScore, na.rm = TRUE),
    n = n()
  ) %>%
  mutate(se = sd/sqrt(n))
```

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

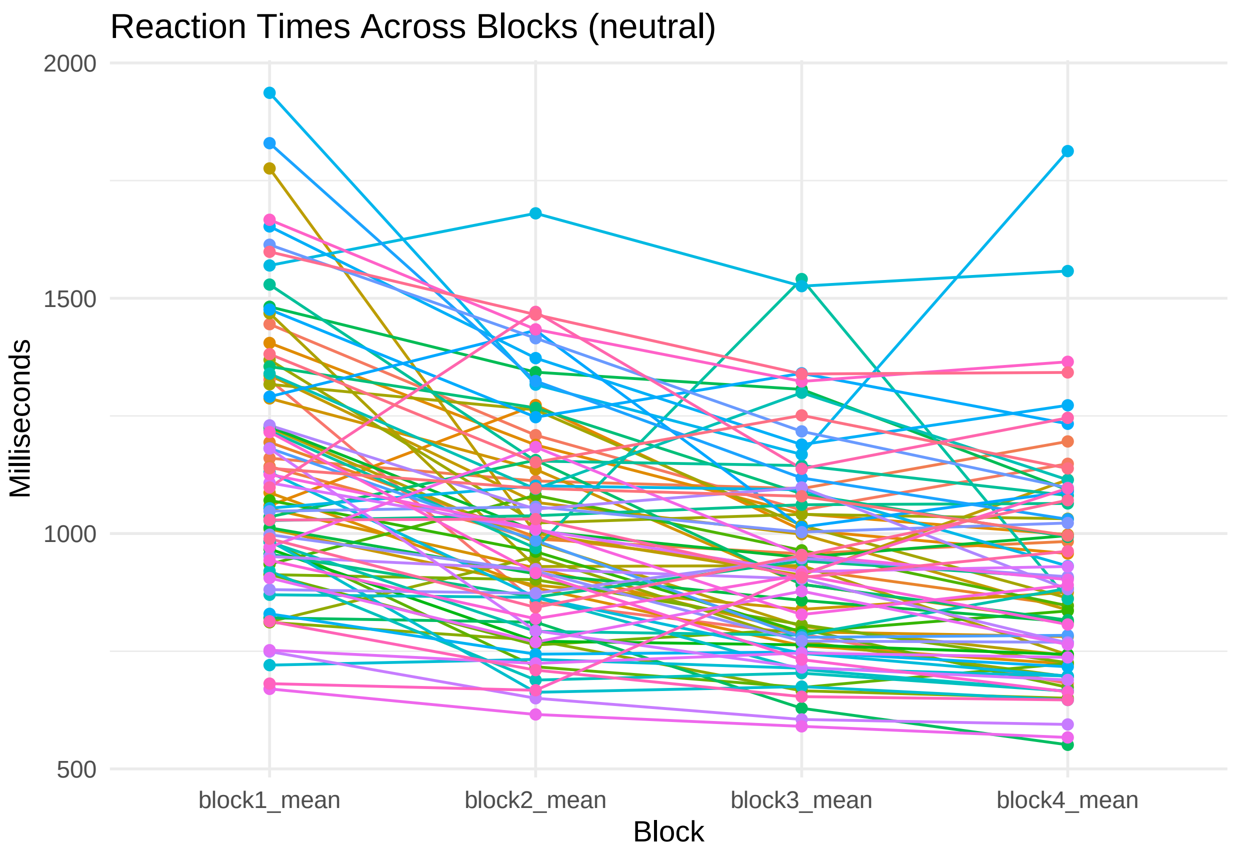
```
# Plotting
ggplot(se_sum, aes(x = Block, y = mean, fill = Attention)) +
  geom_bar(position = position_dodge(0.8), stat = "identity", color = "black", size = 0.5, width = 0.8,
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), position = position_dodge(0.8), width = 0.25, c
  scale_fill_manual(values = c("neutral" = "blue", "biased" = "red")) +
  labs(title = "Mean Absolute Bias Scores by Block and Attention",
       x = "Block",
      y = "Mean Absolute Bias Score") +
  theme_minimal()
```



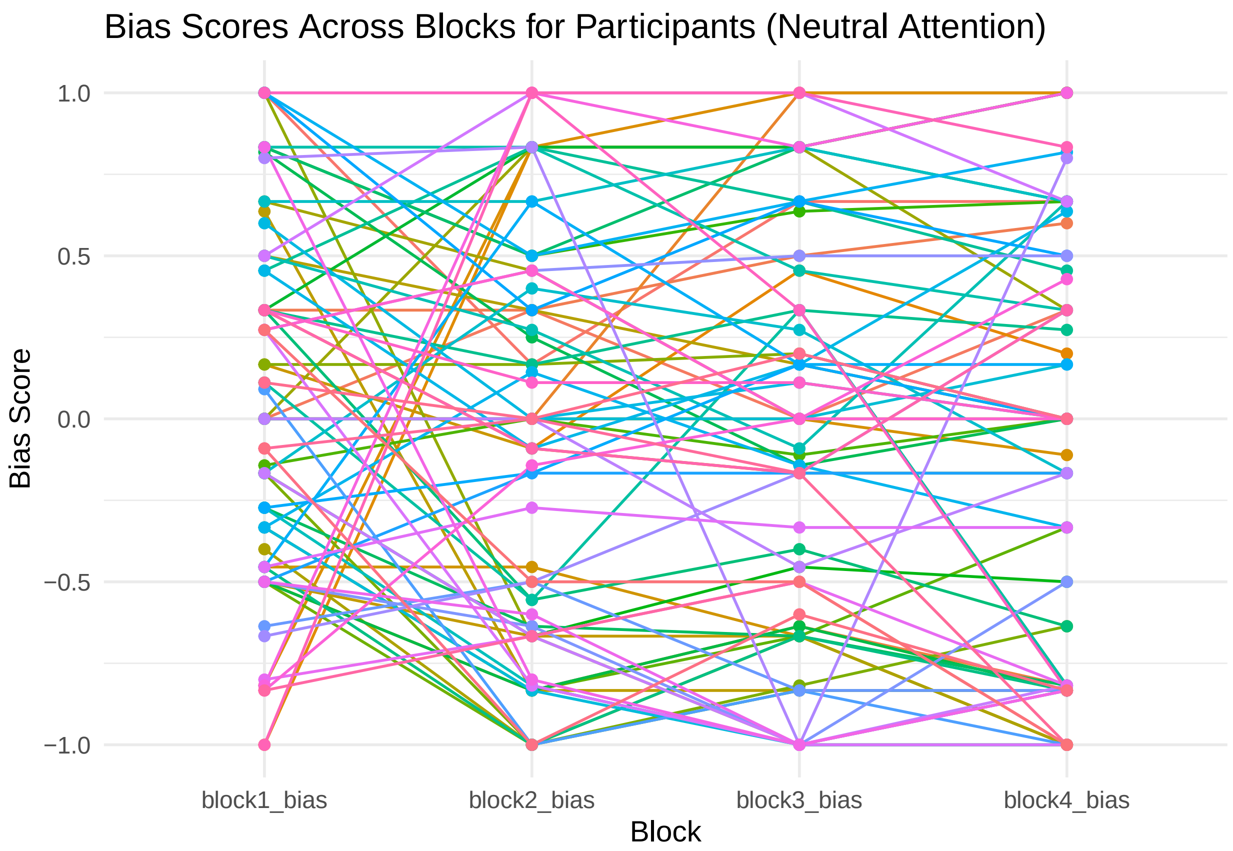
```
long_df <- data %>%
  gather(key = "Block", value = "RT", block1_mean:block4_mean)

# Convert block names to a factor to ensure proper ordering
long_df$Block <- factor(long_df$Block, levels = c("block1_mean", "block2_mean", "block3_mean", "block4_

biased <- subset(long_df, long_df$BiasScore > 0.8 | long_df$BiasScore < -0.8)
neutral <- subset(long_df, long_df$BiasScore <= 0.8 & long_df$BiasScore >= -0.8)


# Plot using ggplot2 with a subset of subjects
ggplot(biased, aes(x = Block, y = RT, group = Subject, color = Subject)) +
  geom_line() +
  geom_point() +
  labs(
    title = "Reaction Times Across Blocks (biased)",
    x = "Block",
    y = "Milliseconds"
  ) +
  theme_minimal()+
  theme(legend.position = "none")
```



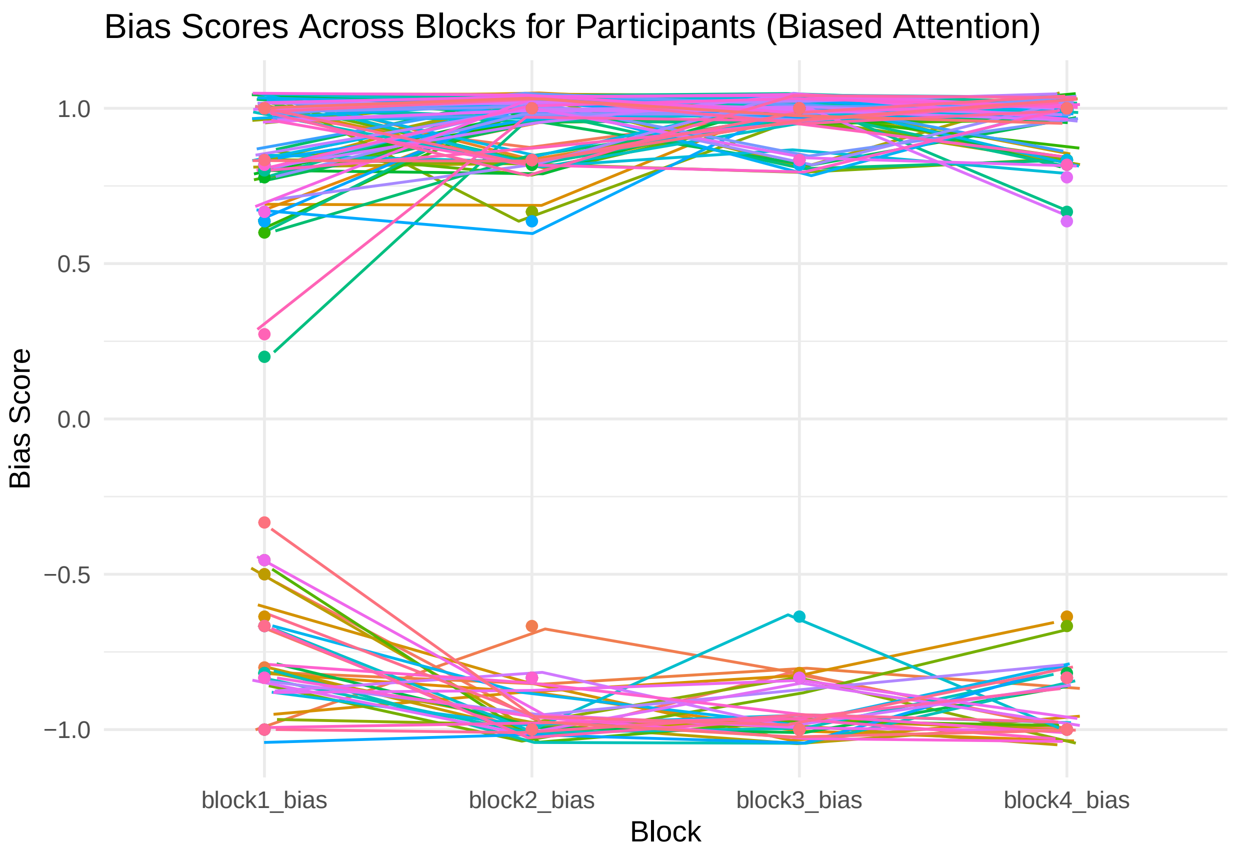
```r
ggplot(neutral, aes(x = Block, y = RT, group = Subject, color = Subject)) +
  geom_line() +
  geom_point() +
  labs(
    title = "Reaction Times Across Blocks (neutral)",
    x = "Block",
    y = "Milliseconds"
  ) +
  theme_minimal()+
  theme(legend.position = "none")
```



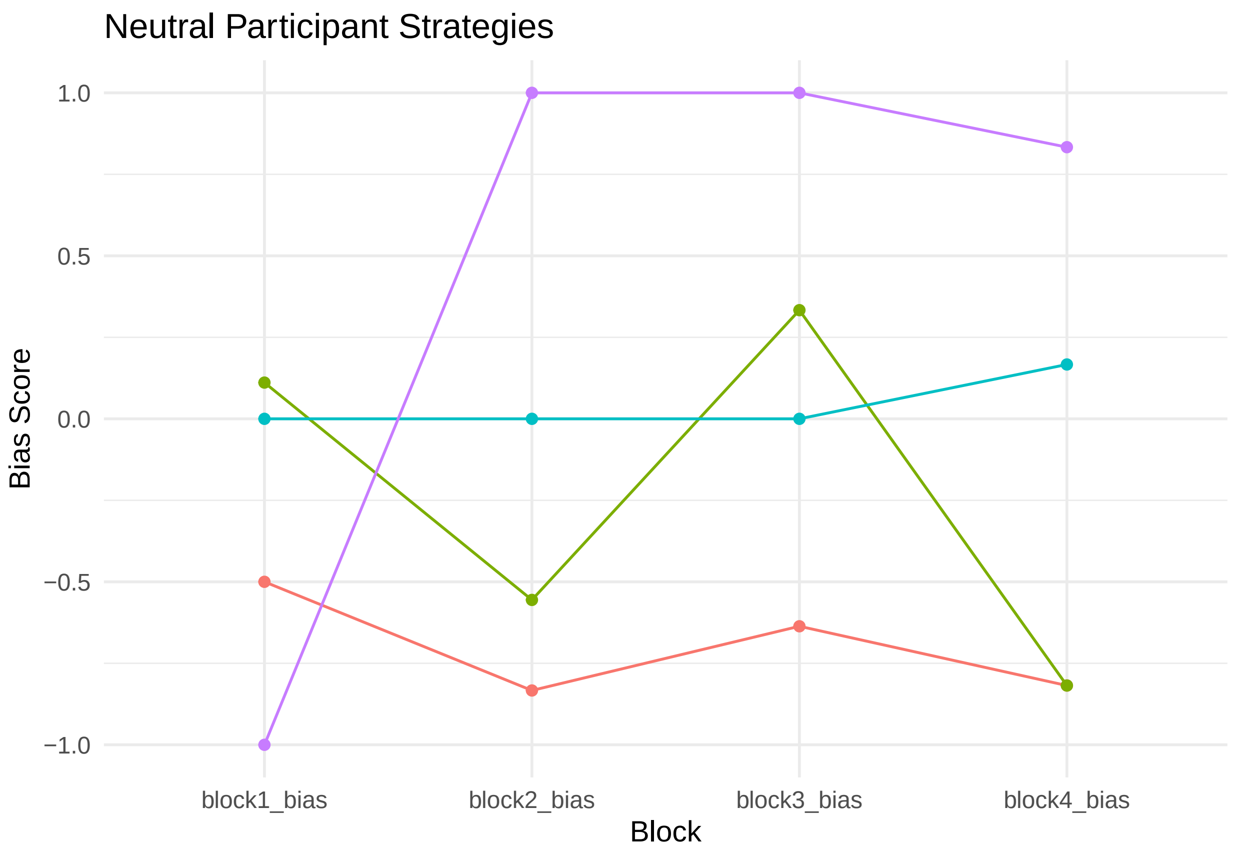
```
long_df2 <- data %>%
  gather(key = "Block", value = "BiasScore", block1_bias:block4_bias)

#Bias Score across blocks

df_unique <- data[, !duplicated(as.list(data))]

data_neutral <- df_unique %>%
  filter(Attention == 'neutral')

# Reshape the data for ggplot2
data_neutral_long <- data_neutral %>%
  select(Subject, block1_bias, block2_bias, block3_bias, block4_bias) %>%
  pivot_longer(cols = starts_with("block"), names_to = "Block", values_to = "BiasScore")

# Plotting
ggplot(data_neutral_long, aes(x = Block, y = BiasScore, group = Subject, color = Subject)) +
  geom_line() +
  geom_point() +
  labs(title = "Bias Scores Across Blocks for Participants (Neutral Attention)",
       x = "Block",
       y = "Bias Score") +
  theme_minimal() +
  theme(legend.position = "none")
```



```r
#biased
data_biased <- df_unique %>%
  filter(Attention == 'biased')

# Reshape the data for ggplot2
data_biased_long <- data_biased %>%
  select(Subject, block1_bias, block2_bias, block3_bias, block4_bias) %>%
  pivot_longer(cols = starts_with("block"), names_to = "Block", values_to = "BiasScore")

# Plotting
ggplot(data_biased_long, aes(x = Block, y = BiasScore, group = Subject, color = Subject)) +
  geom_line(position=position_jitter(w=0.05,h=0.05)) +
  geom_point() +
  labs(title = "Bias Scores Across Blocks for Participants (Biased Attention)",
       x = "Block",
       y = "Bias Score") +
  theme_minimal() +
  theme(legend.position = "none")
```



```
data_neutral_selected <- data_neutral_long %>%
  filter(Subject %in% c("e8b26ab1", "89601069", "08f746fa", "6409a8b2"))  # Replace 1, 2, 3 with the su

# Plot the selected participants
ggplot(data_neutral_selected, aes(x = Block, y = BiasScore, group = Subject, color = Subject)) +
  geom_line() +
  geom_point() +
  labs(
    title = "Neutral Participant Strategies",
    x = "Block",
    y = "Bias Score"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```



```
print(data_neutral_long)
```

```
## # A tibble: 304 x 3
##    Subject  Block       BiasScore
##    <chr>    <chr>           <dbl>
##  1 0156ce12 block1_bias     1
##  2 0156ce12 block2_bias     0.167
##  3 0156ce12 block3_bias     0.667
##  4 0156ce12 block4_bias     0.667
##  5 054ba968 block1_bias     0
##  6 054ba968 block2_bias     0.333
##  7 054ba968 block3_bias     0
##  8 054ba968 block4_bias     0.333
##  9 05546b7f block1_bias     0.333
## 10 05546b7f block2_bias     0.333
## # i 294 more rows
```

```
demographics_data <-read.csv("Copy of CardSortingTask_Analysis_09.28.2023.csv")
names(demographics_data)[names(demographics_data) == "SubjectNumber"] <- "Subject"
merged_df <- merge(demographics_data, df_unique, by = "Subject")
cols_toremove <- c("SymRespCount.x", "TxtRespCount.x", "IncorrRespCount")
merged_df <- merged_df %>% select(-one_of(cols_toremove))

merged_df$Block <- as.factor(sub("CardSort_Block", "", merged_df$Block))
```

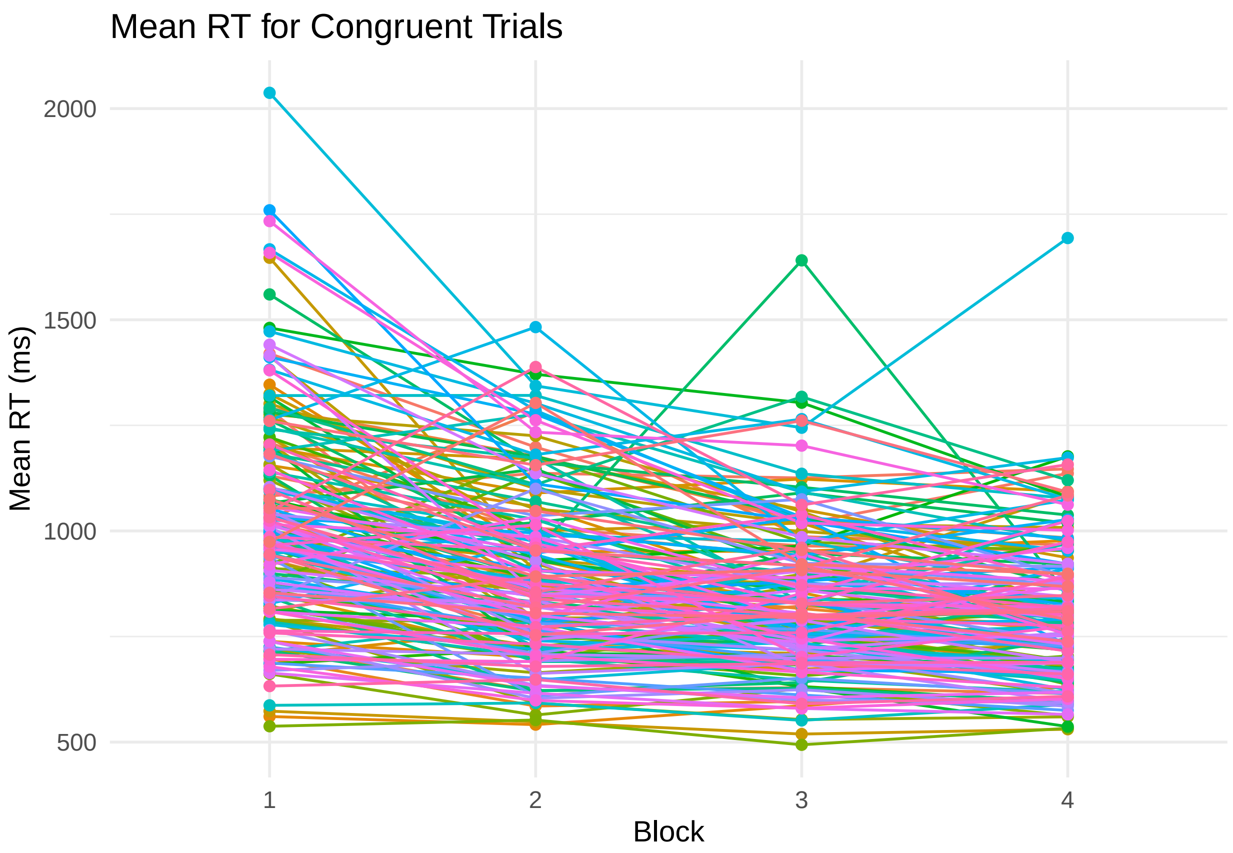
```r
# Split the data into congruent and incongruent trials based on some condition criteria
# Note: You need to adjust 'Condition' based on what defines congruent (1) and incongruent (2) trials i
congruent_data <- merged_df %>%
  filter(Condition == 1) %>%
  group_by(Subject, Block) %>%
  summarise(Mean_RT = mean(RT, na.rm = TRUE))
```

```
## ‘summarise()‘ has grouped output by ’Subject’. You can override using the
## ‘.groups‘ argument.
```

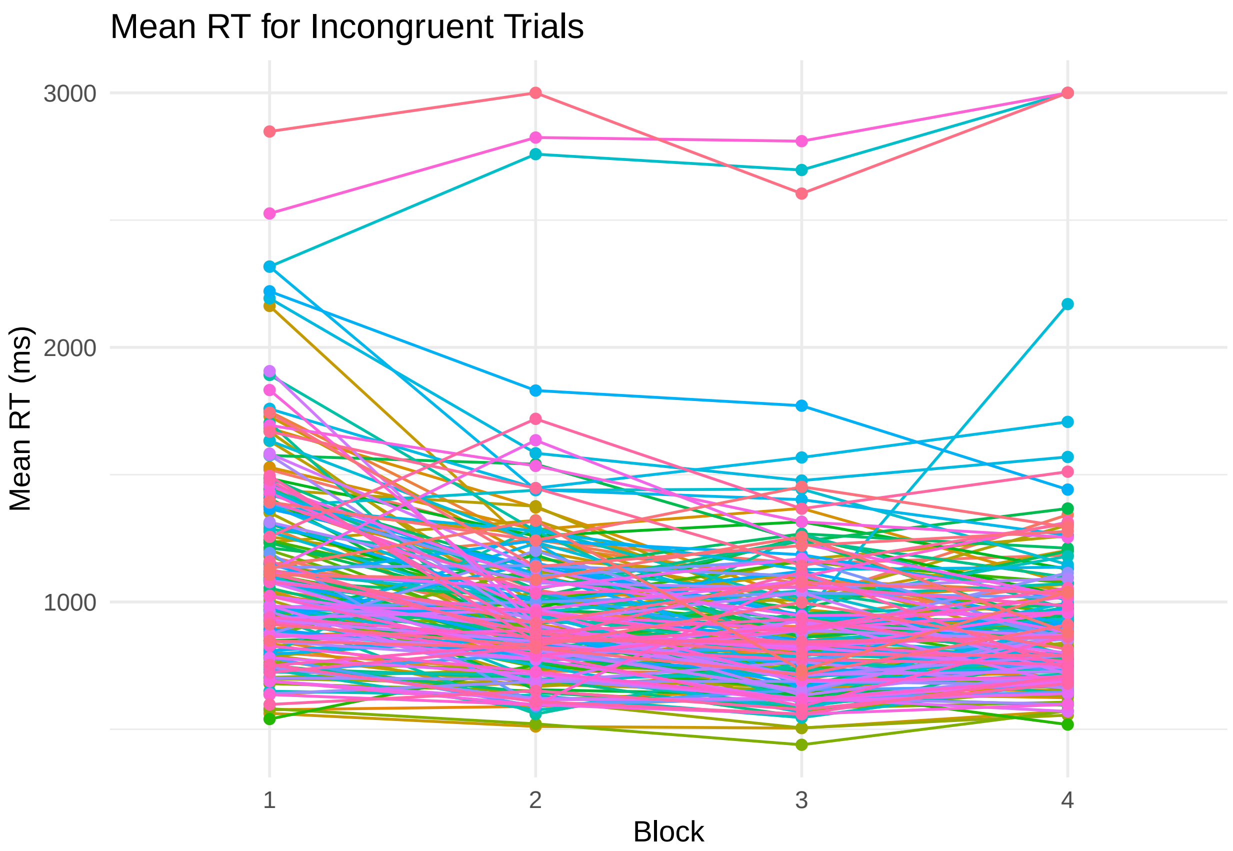
```r
incongruent_data <- merged_df %>%
  filter(Condition == 2) %>%
  group_by(Subject, Block) %>%
  summarise(Mean_RT = mean(RT, na.rm = TRUE))
```

```
## ‘summarise()‘ has grouped output by ’Subject’. You can override using the
## ‘.groups‘ argument.
```

```r
# Create a plot for Congruent Trials
p1 <- ggplot(congruent_data, aes(x = Block, y = Mean_RT, group = Subject, color = Subject)) +
  geom_line() +
  geom_point() +
  labs(title = "Mean RT for Congruent Trials", x = "Block", y = "Mean RT (ms)") +
  theme_minimal()+
  theme(legend.position = "none")

# Create a plot for Incongruent Trials
p2 <- ggplot(incongruent_data, aes(x = Block, y = Mean_RT, group = Subject, color = Subject)) +
  geom_line() +
  geom_point() +
  labs(title = "Mean RT for Incongruent Trials", x = "Block", y = "Mean RT (ms)") +
  theme_minimal()+
  theme(legend.position = "none")

# Print the plots
print(p1)
```


Mean RT for Congruent Trials
Mean RT (ms)
2000
1500
1000
500
1
2
3
4
Block

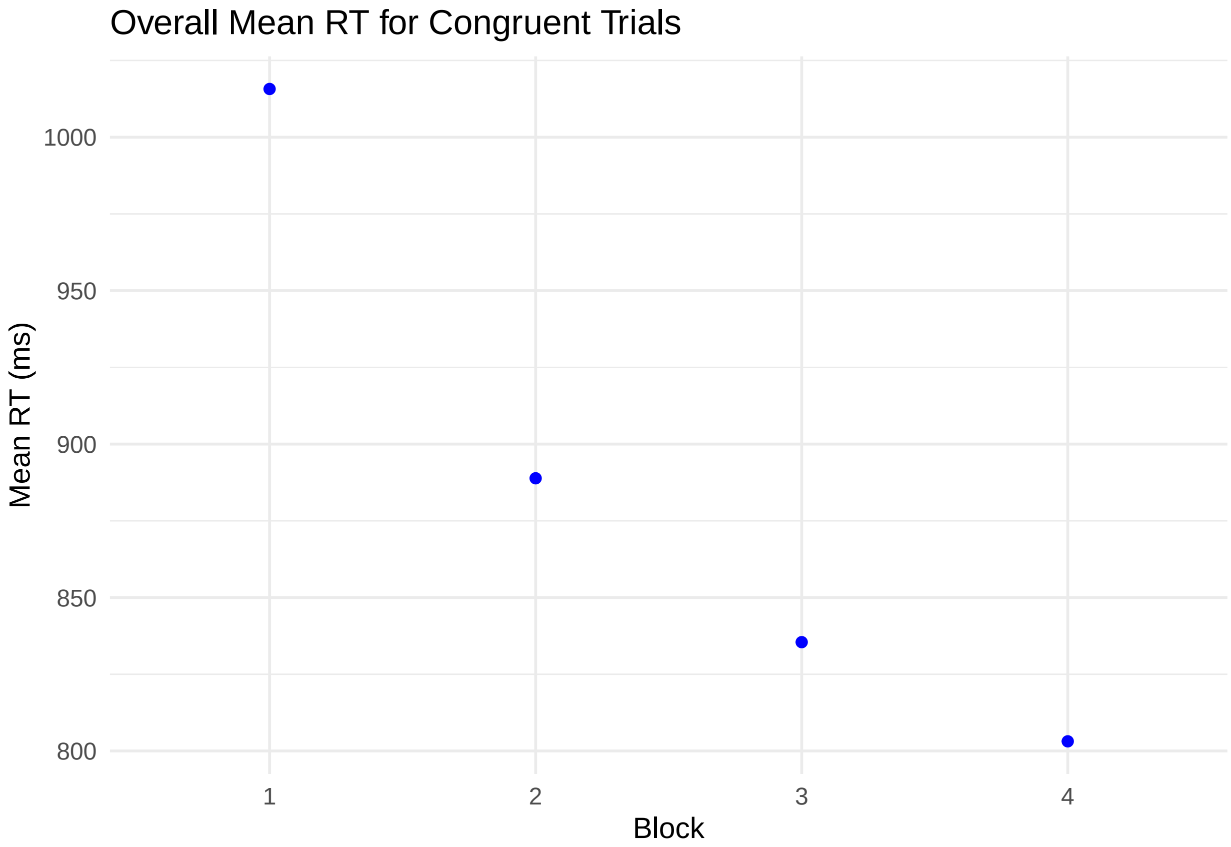
```r
print(p2)
```



```r
overall_congruent <- merged_df %>%
  filter(Condition == 1) %>%
  group_by(Block) %>%
  summarise(Mean_RT = mean(RT, na.rm = TRUE))

overall_incongruent <- merged_df %>%
  filter(Condition == 2) %>%
  group_by(Block) %>%
  summarise(Mean_RT = mean(RT, na.rm = TRUE))

# Plotting overall mean RT for Congruent Trials
p1 <- ggplot(overall_congruent, aes(x = Block, y = Mean_RT)) +
  geom_line(color = "blue") +
  geom_point(color = "blue") +
  labs(title = "Overall Mean RT for Congruent Trials", x = "Block", y = "Mean RT (ms)") +
  theme_minimal() +
  theme(legend.position = "none")

# Plotting overall mean RT for Incongruent Trials
p2 <- ggplot(overall_incongruent, aes(x = Block, y = Mean_RT)) +
  geom_line(color = "red") +
  geom_point(color = "red") +
  labs(title = "Overall Mean RT for Incongruent Trials", x = "Block", y = "Mean RT (ms)") +
  theme_minimal() +
  theme(legend.position = "none")
```

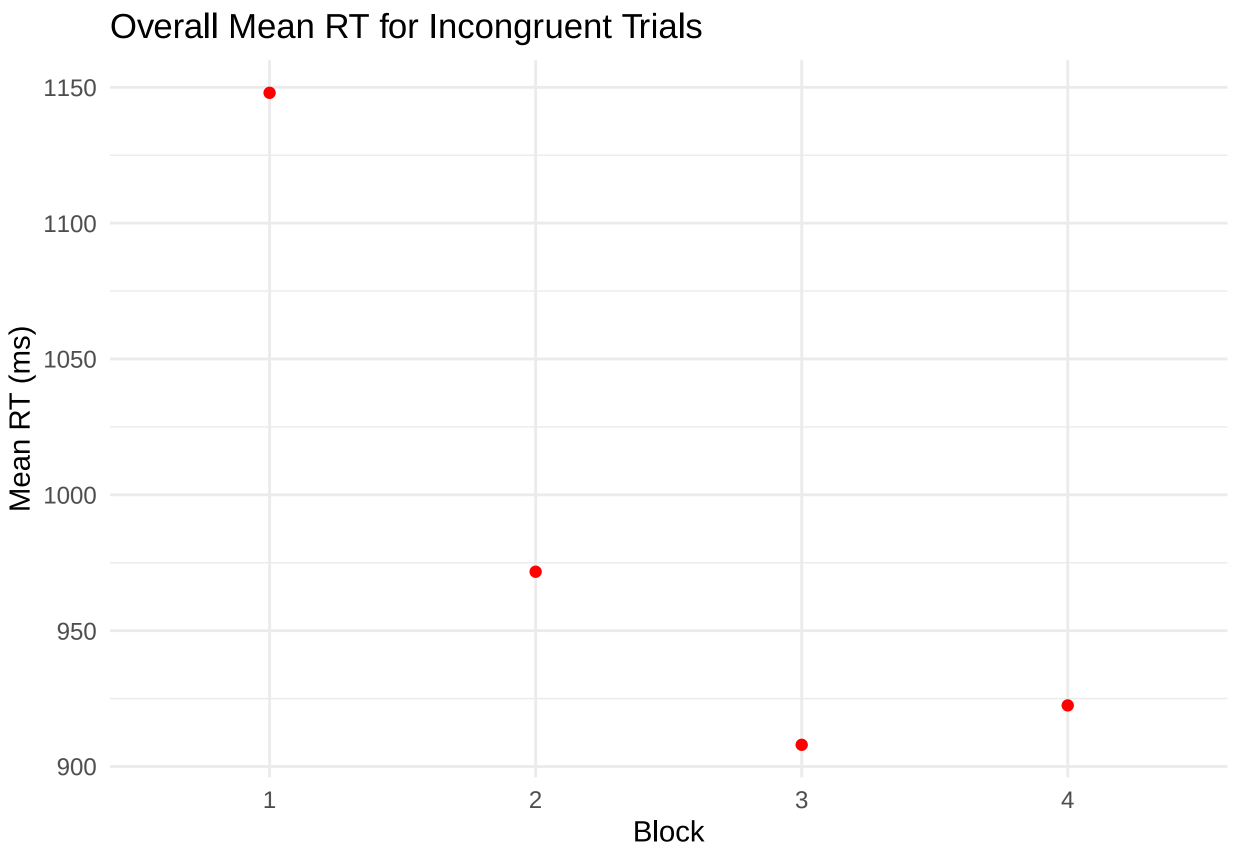
```r
# Print the plots
print(p1)
```

```
## ‘geom_line()‘: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```



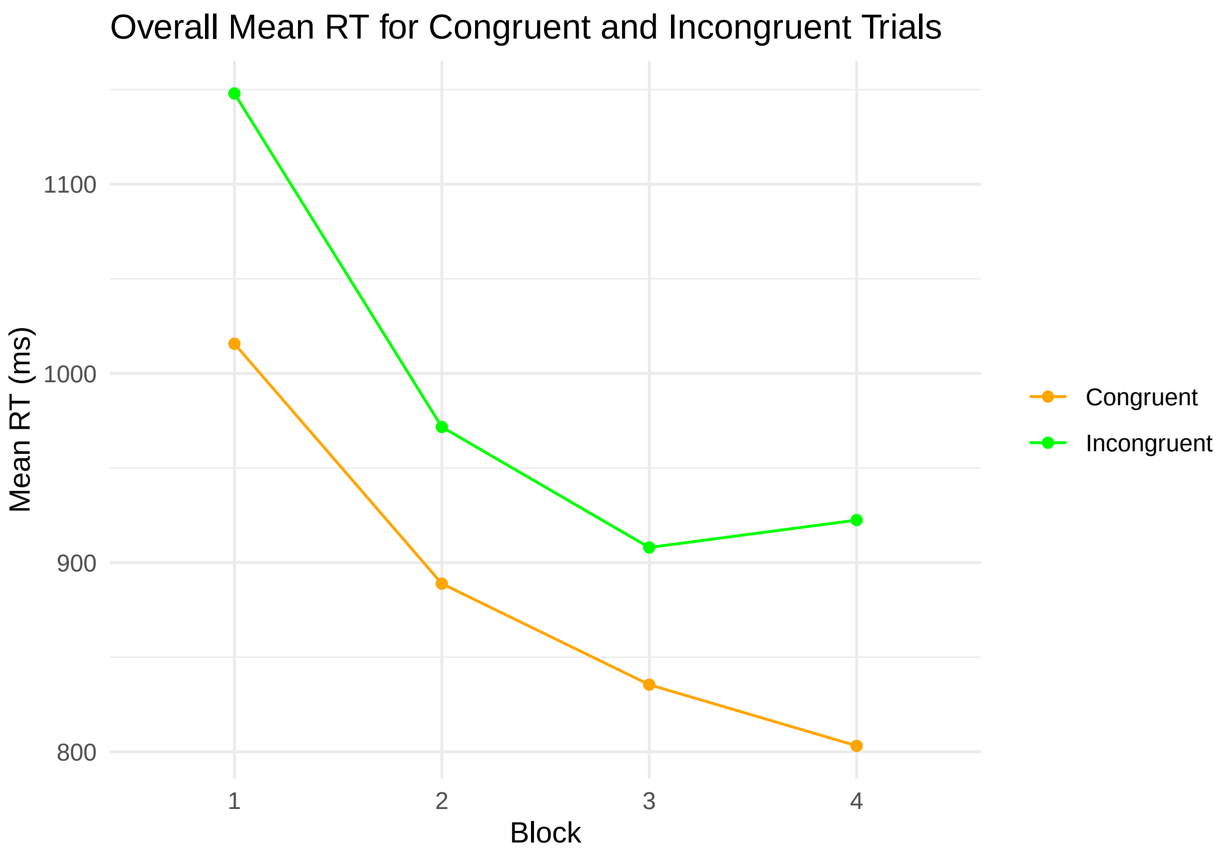
```r
print(p2)
```

```
## ‘geom_line()‘: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```



```
merged_df$Block <- as.factor(sub("CardSort_Block", "", merged_df$Block))

# Calculate overall mean RT for Congruent and Incongruent Trials for each block and create a new column
congruent_data <- merged_df %>%
  filter(Condition == 1) %>%
  group_by(Block) %>%
  summarise(Mean_RT = mean(RT, na.rm = TRUE)) %>%
  mutate(Type = "Congruent")

incongruent_data <- merged_df %>%
  filter(Condition == 2) %>%
  group_by(Block) %>%
  summarise(Mean_RT = mean(RT, na.rm = TRUE)) %>%
  mutate(Type = "Incongruent")

# Combine the datasets
combined_data <- rbind(congruent_data, incongruent_data)

# Plotting overall mean RT for both Congruent and Incongruent Trials
combined_plot <- ggplot(combined_data, aes(x = Block, y = Mean_RT, color = Type, group = Type)) +
  geom_line() +
  geom_point() +
  labs(title = "Overall Mean RT for Congruent and Incongruent Trials", x = "Block", y = "Mean RT (ms)")
  scale_color_manual(values = c("orange", "green")) +
  theme_minimal() +
  theme(legend.title = element_blank())  # Optionally remove legend title
```

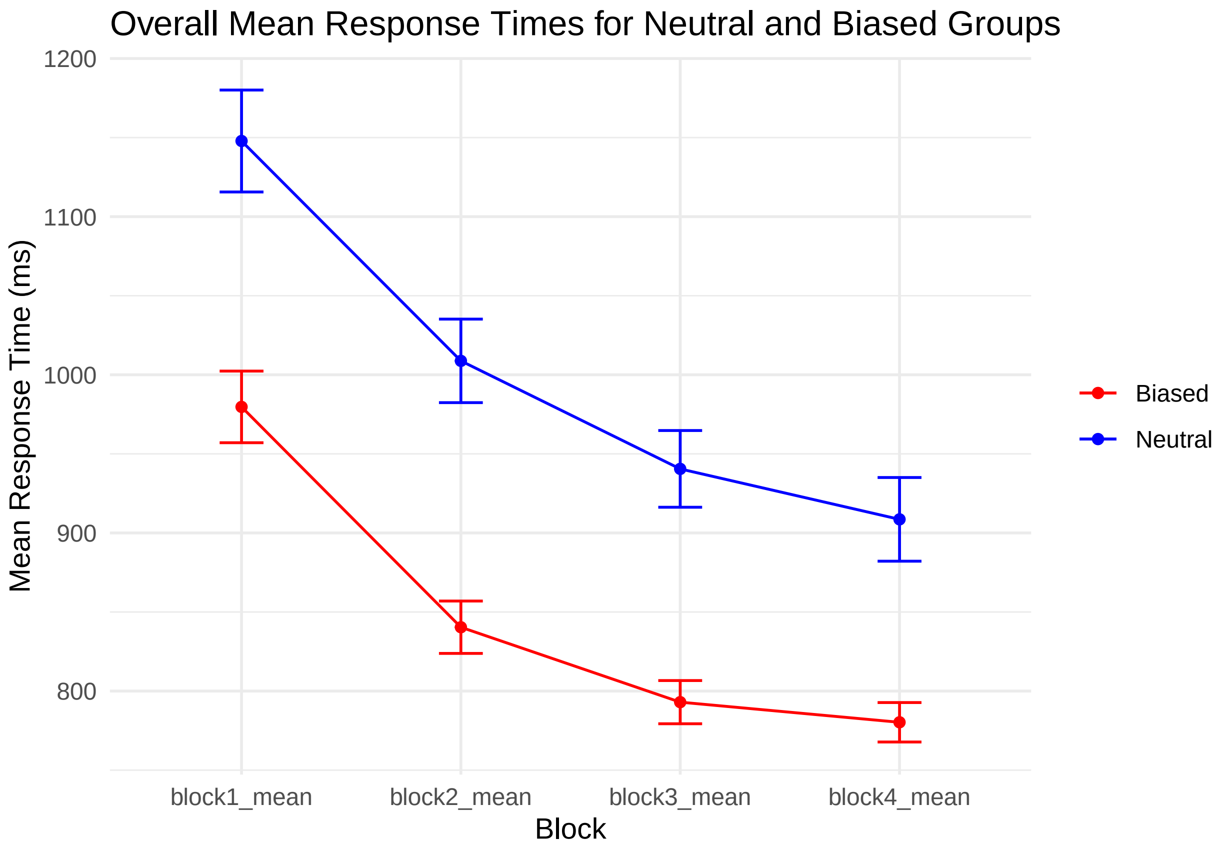
```
# Print the combined plot
print(combined_plot)
```



```
biased$Group <- "Biased"
neutral$Group <- "Neutral"

# Combine the two datasets into one
combined_data <- rbind(biased, neutral)

# Function to calculate standard error of the mean (SEM)
sem <- function(x) {
  return(sd(x, na.rm = TRUE) / sqrt(length(x)))
}

# Aggregate the data to calculate mean RT and SEM for each Block within each Group
mean_rt_scores <- combined_data %>%
  group_by(Block, Group) %>%
  summarise(
    Mean_RT = mean(RT, na.rm = TRUE),
    SEM = sem(RT)
  )
```

```
## ‘summarise()‘ has grouped output by ’Block’. You can override using the
## ‘.groups‘ argument.
```

```r
# Plotting overall mean response time with error bars for Neutral and Biased Groups
combined_plot <- ggplot(mean_rt_scores, aes(x = Block, y = Mean_RT, color = Group, group = Group)) +
  geom_line() +
  geom_point() +
  geom_errorbar(aes(ymin = Mean_RT - SEM, ymax = Mean_RT + SEM), width = 0.2) +
  labs(title = "Overall Mean Response Times for Neutral and Biased Groups", x = "Block", y = "Mean Respo
  scale_color_manual(values = c("red", "blue")) +
  theme_minimal() +
  theme(legend.title = element_blank())  # Optionally remove legend title

# Print the combined plot
print(combined_plot)
```



```r
parc_cardsort <- read.csv("CardSort_Summary(in).csv")
parc_demographics <- read.csv("Demographics_Summary(in).csv")

parc_merged <- merge(parc_demographics, parc_cardsort, by = "Subject")
maia_data <- read.csv("CardSort Data.csv")
lang_data <- read.csv("CardSortLanguage.csv")
lang_data = lang_data[, c("SubjectNumber", "NativeEnglish", "SecondLang", "WhatSecLang")]
lang_data$L1 = NA

for (subject in lang_data$SubjectNumber){
  lang_data$L1[lang_data$NativeEnglish == 1] <- "English"
  lang_data$L1[lang_data$NativeEnglish == 2] <- lang_data$WhatSecLang[lang_data$NativeEnglish == 2]
}

# Drop columns using base R
lang_data <- lang_data[, setdiff(names(lang_data), c("NativeEnglish", "SecondLang", "WhatSecLang"))]

maia_merged = merge(lang_data, maia_data, by= "SubjectNumber")

parc_merged = parc_merged[, c("Subject", "L1", "SymRespCount", "TxtRespCount","IncorrRespCount")]
maia_merged = maia_merged[, c("SubjectNumber", "L1", "SymRespCount", "TxtRespCount", "IncorrRespCount")]
names(maia_merged)[names(maia_merged) == "SubjectNumber"] <- "Subject"

merged <- rbind(maia_merged, parc_merged)

englishL1 <- merged[merged$L1 == "English", ]
unique_subjects_eng <- unique(englishL1$Subject)
num_sub_eng <- length(unique_subjects_eng)

not_englishL1 <- merged[merged$L1 != "English", ]
unique_subjects_not_eng <- unique(not_englishL1$Subject)
num_eng_not_L1 <- length(unique_subjects_not_eng)
```