kidmid behavior

```r
require(ggplot2)
```

```
## Loading required package: ggplot2
```

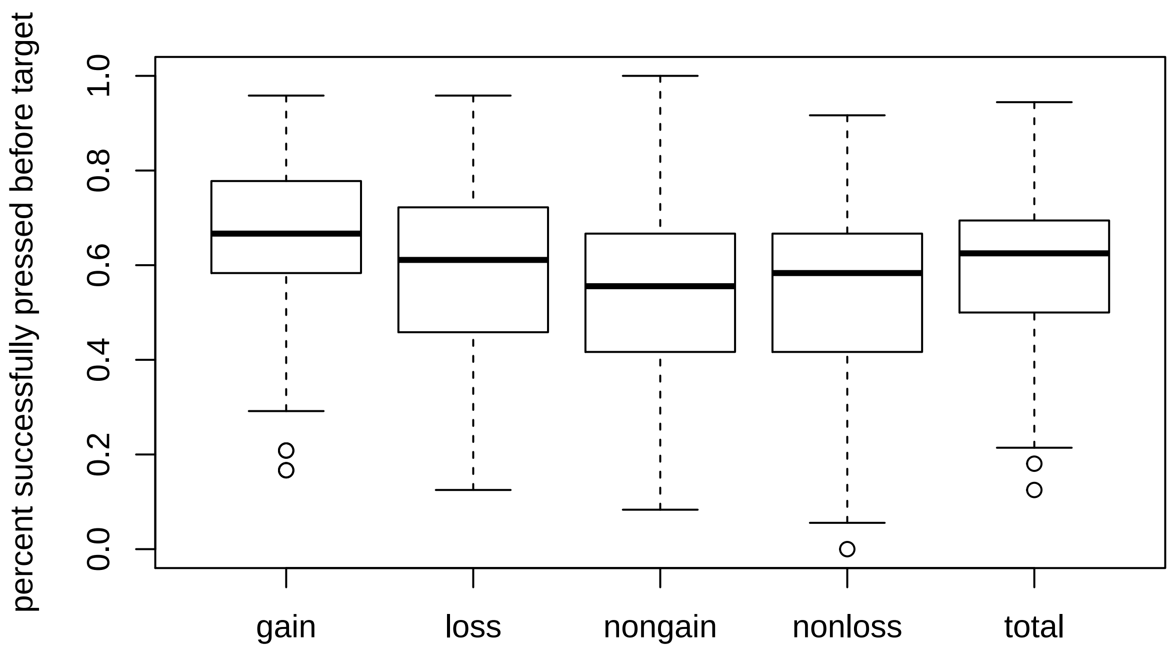
```r
setwd("~/Documents/ELS/KIDMID/Analysis/behavior")
d = read.csv("all_behavior_28-Dec-2014.csv", header=TRUE) #change to updated version

n = length(d$subID)
accuracy = data.frame(d$gain_acc,d$loss_acc,d$nongain_acc,d$nonloss_acc,d$total_acc)
rt = data.frame(d$rt_gain,d$rt_loss,d$rt_nongain,d$rt_nonloss,d$rt_mean)
counts = data.frame(d$gain_count, d$loss_count, d$nongain_count, d$nonloss_count, d$unsucc_gain_count, d
#mean_rt = (d$rt_gain + d$rt_loss + d$rt_nongain + d$rt_nonloss)/4
#rt = data.frame(d$rt_gain,d$rt_loss,d$rt_nongain,d$rt_nonloss,mean_rt)
```

Summary statistics

–73 subjects

Accuracy

```r
plot1 = boxplot(accuracy, names = c("gain","loss","nongain","nonloss","total"),ylab=("percent successful
```



The total accuracy (59.1%) is lower than what is expected. Participants should be successfully pressing the button before the target-offset approximately 66% of the time.

Gain: mean = 64.6%; median = 66.7%; min = 16.7%; max = 95.8%

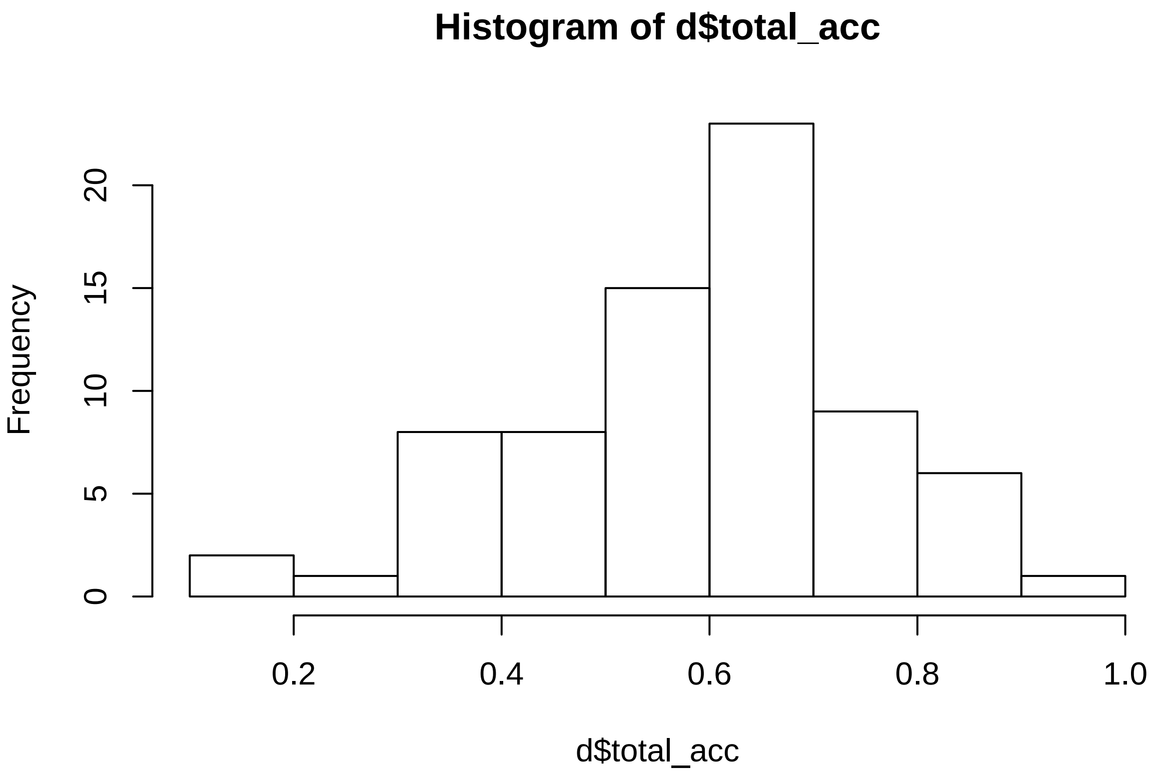
Loss: mean = 59.1%; median = 61.1%; min = 12.5%; max = 95.8%

Nongain: mean = 54.1%; median = 55.6%; min = 8.3%; max = 100%

Nonloss: mean = 54.6%; median = 58.3%; min = 0%; max = 91.7%

All trials: mean = 59.1%; median = 62.5%; min = 12.5%; max = 94.4%

```r
hist(d$total_acc)
```



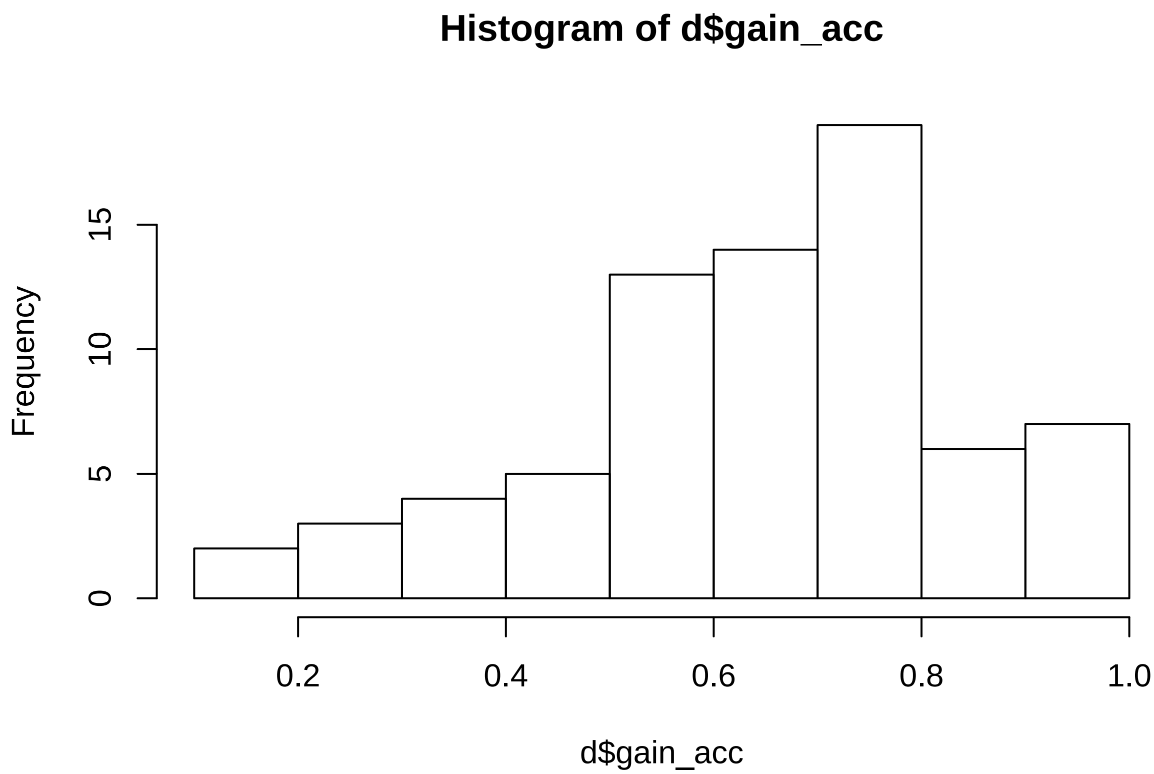

```r
summary(d$total_acc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.125   0.500   0.625   0.591   0.694   0.944
```

```r
total.out = boxplot.stats(d$total_acc, do.out=TRUE)
total.out$out
```

```
## [1] 0.1806 0.1250
```

```r
hist(d$gain_acc)
```



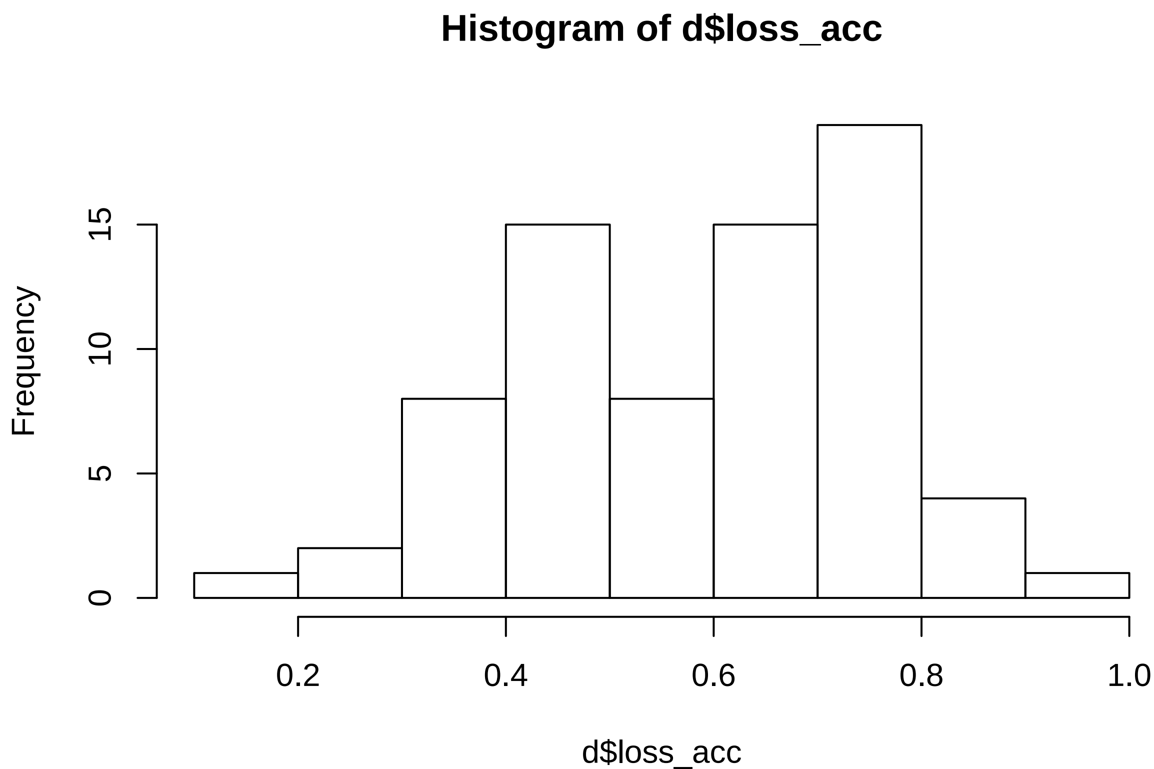

```r
summary(d$gain_acc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.167   0.583   0.667   0.646   0.778   0.958
```

```r
gain_acc.out = boxplot.stats(d$gain_acc, do.out=TRUE)
gain_acc.out$out
```

```
## [1] 0.1667 0.1667 0.2083 0.2083
```

```r
hist(d$loss_acc)
```



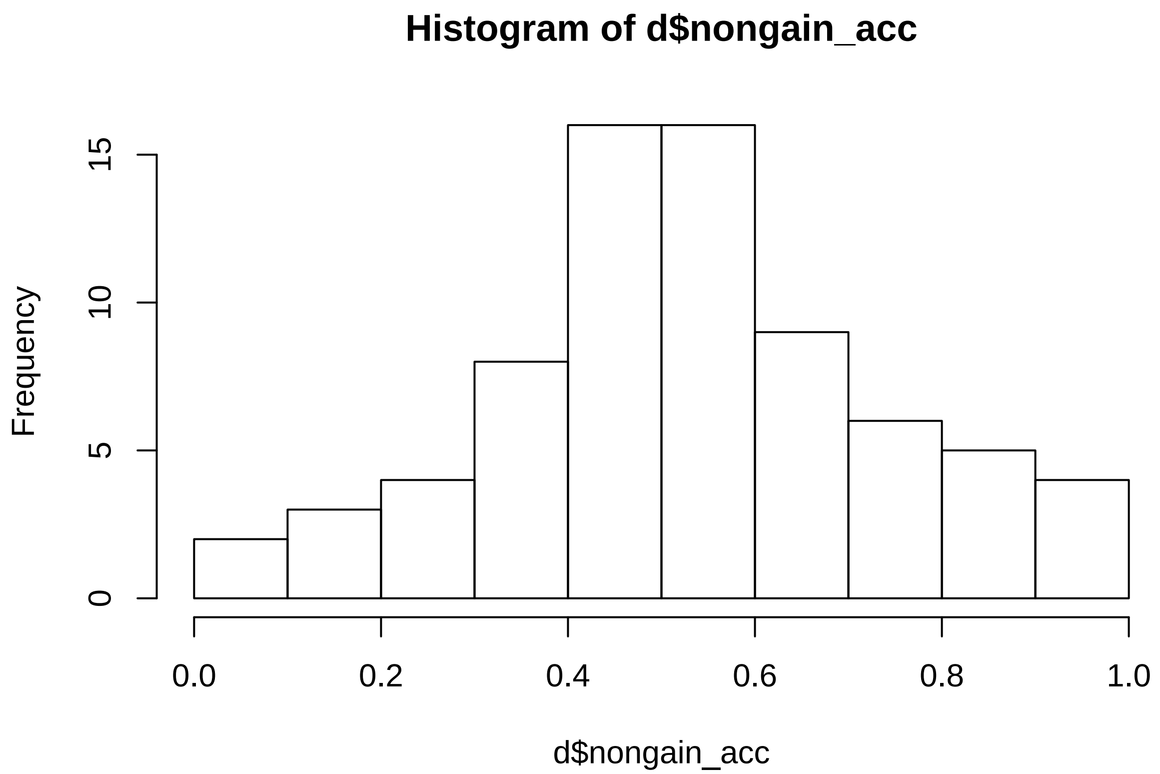

```
summary(d$loss_acc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.125   0.458   0.611   0.591   0.722   0.958
```

```
loss_acc.out = boxplot.stats(d$loss_acc, do.out=TRUE)
loss_acc.out$out
```

```
## numeric(0)
```

```
hist(d$nongain_acc)
```



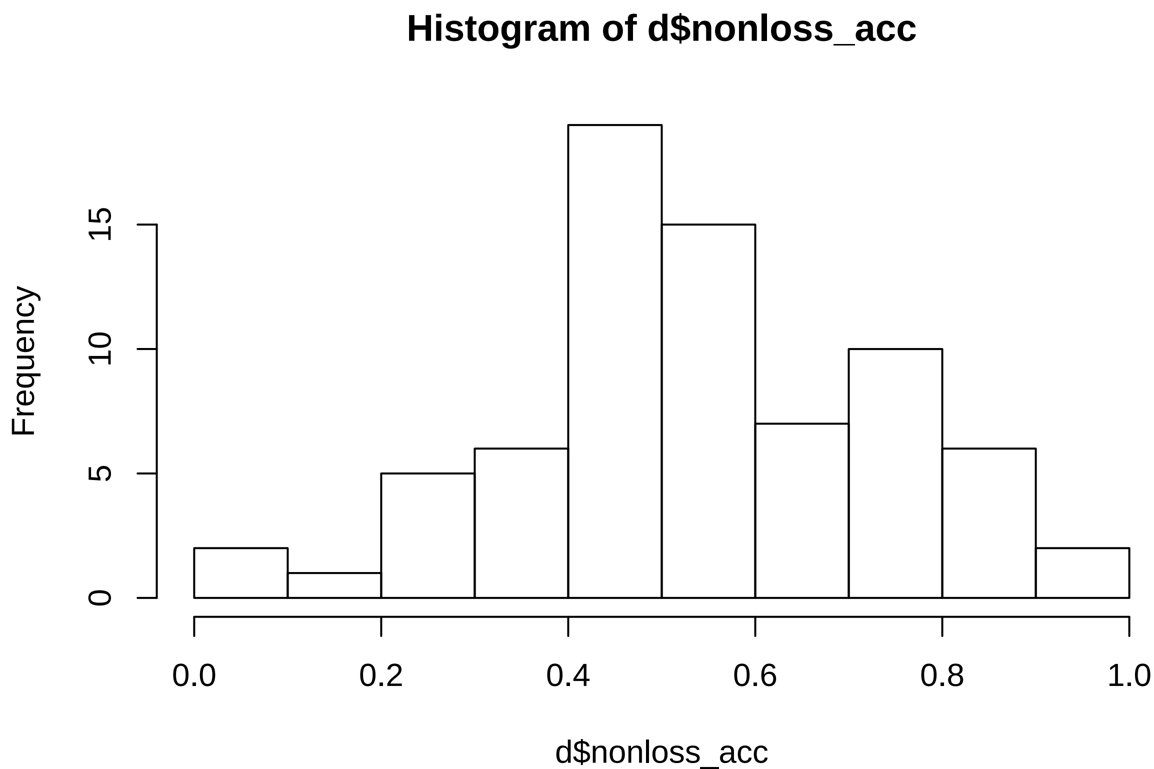

```r
summary(d$nongain_acc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0833  0.4170  0.5560  0.5410  0.6670  1.0000
```

```r
nongain_acc.out = boxplot.stats(d$nongain_acc, do.out=TRUE)
nongain_acc.out$out
```

```
## numeric(0)
```

```r
hist(d$nonloss_acc)
```




```r
summary(d$nonloss_acc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.417   0.583   0.546   0.667   0.917
```

```r
nonloss_acc.out = boxplot.stats(d$nonloss_acc, do.out=TRUE)
nonloss_acc.out$out
```

```
## [1] 0
```

Missed Trials

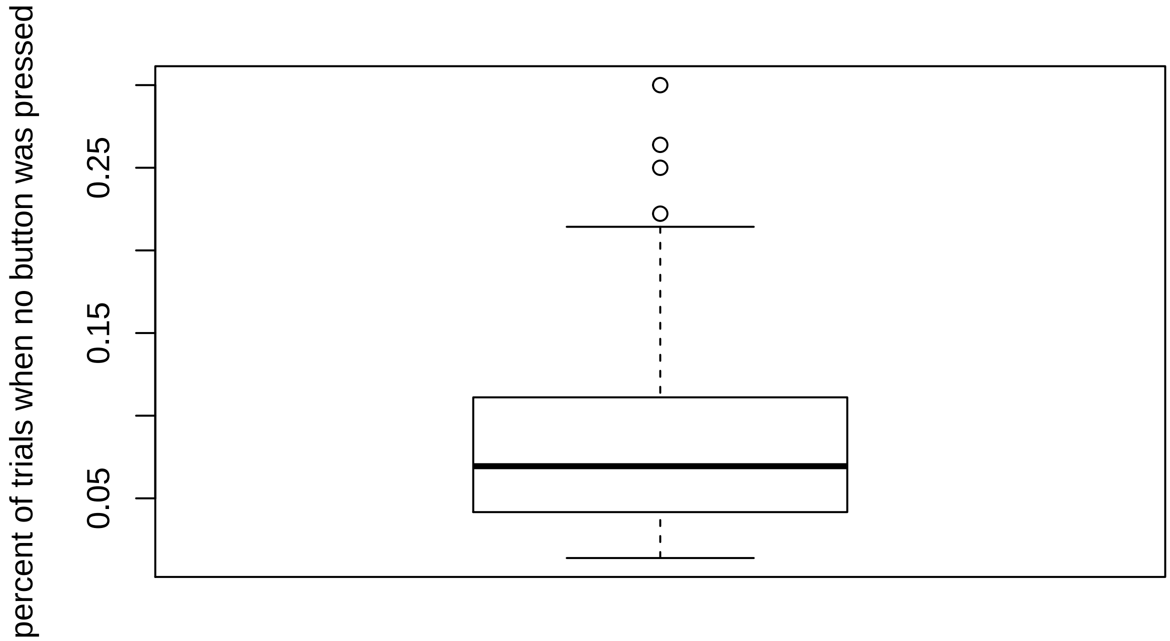
```r
summary(d$missed_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    3.00    5.00    6.55    8.00   21.00
```

```r
summary(d$missed_percent) #same as missed_count
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0139  0.0417  0.0694  0.0911  0.1110  0.3000
```

```r
plot2 = boxplot(d$missed_percent,ylab=("percent of trials when no button was pressed"))
```

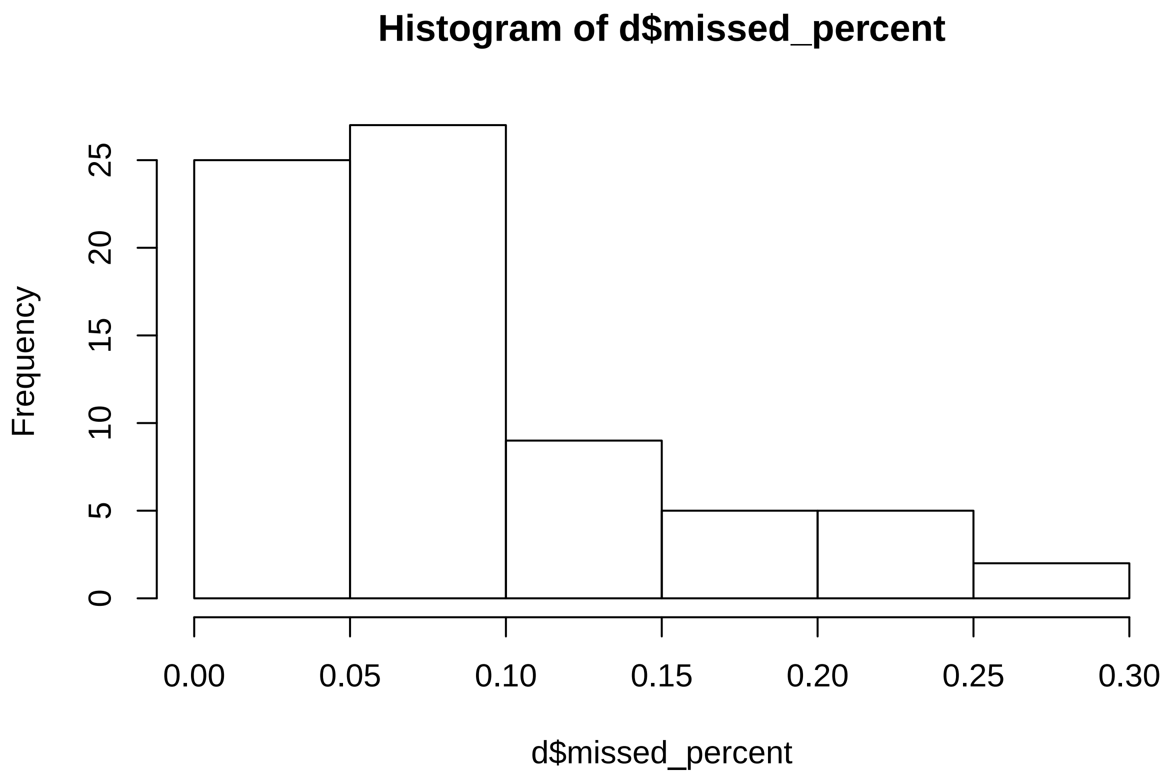


Outliers

```r
missed.out = boxplot.stats(d$missed_percent,do.out = TRUE)
missed.out$out
```

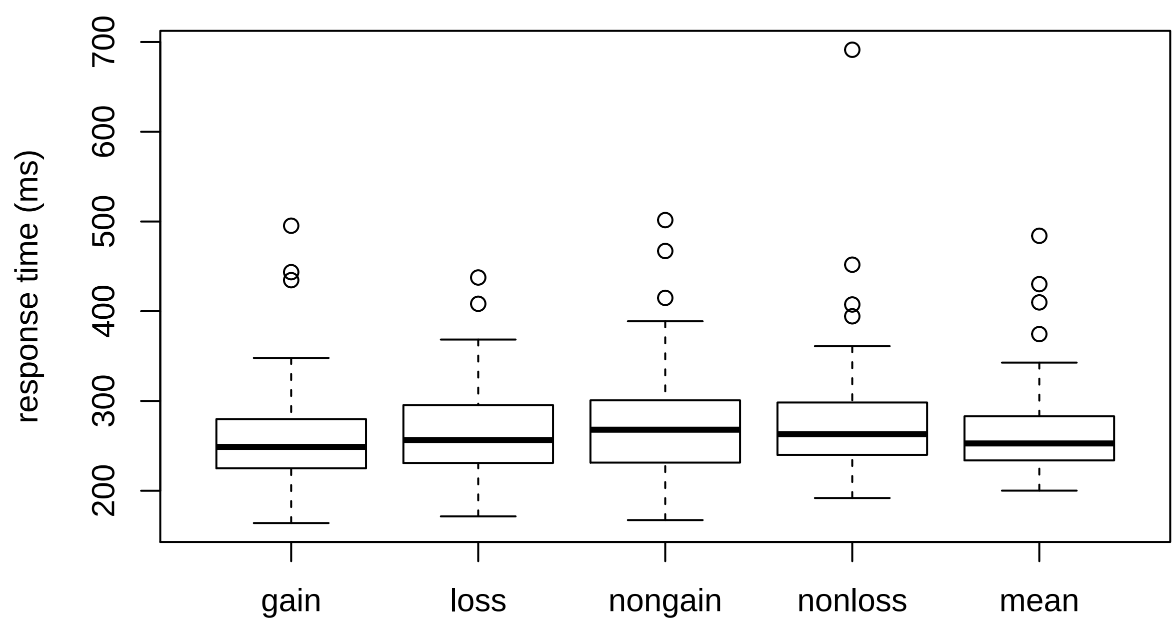
```
## [1] 0.2222 0.3000 0.2639 0.2500
```

```r
hist(d$missed_percent)
```



Response Time

```
plot3 = boxplot(rt, names = c("gain","loss","nongain","nonloss","mean"),ylab=("response time (ms)"))
```



Gain: mean = 257.9 ms; median = 248.86 ms; min = 164 ms; max = 495.29 ms

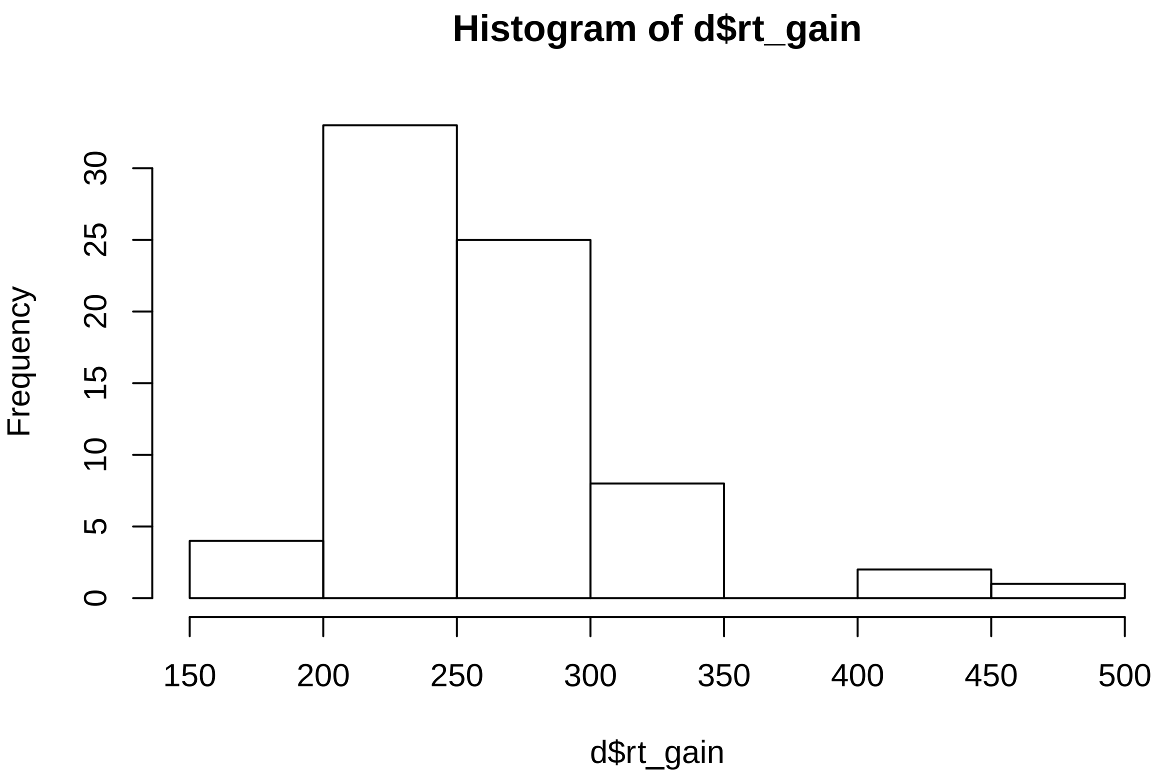
Loss: mean = 264.65 ms; median = 256.52 ms; min = 171.44 ms; max = 437.59 ms

Nongain: mean = 275.41 ms; median = 268 ms; min = 167.3 ms; max = 501.56 ms

Nonloss: mean = 277.74 ms; median = 263.08 ms; min = 191.89 ms; max = 691.33 ms

All trials: mean = 267.04 ms; median = 252.68 ms; min = 200.12 ms; max = 484.08 ms

```
hist(d$rt_gain)
```



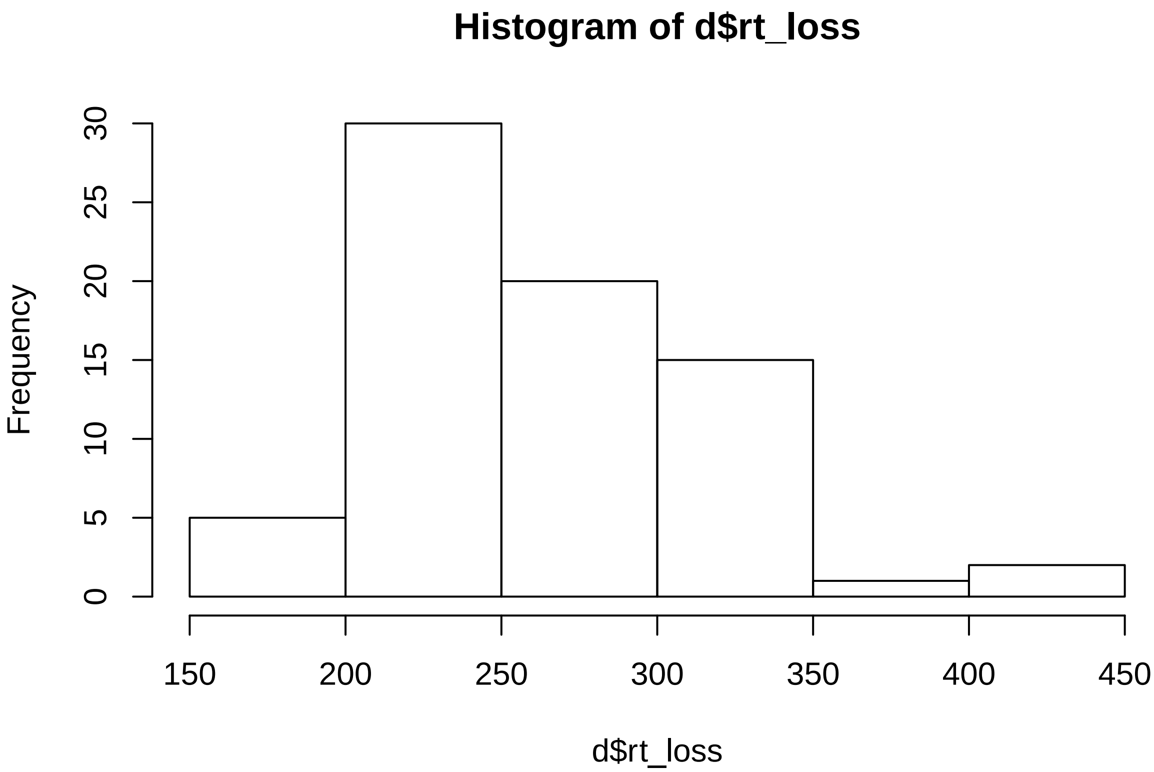

```
summary(d$rt_gain)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     164     225     249     258     280     495
```

```
gain_rt.out = boxplot.stats(d$rt_gain, do.out=TRUE)
gain_rt.out$out
```

```
## [1] 495.3 434.6 443.5
```

```r
hist(d$rt_loss)
```



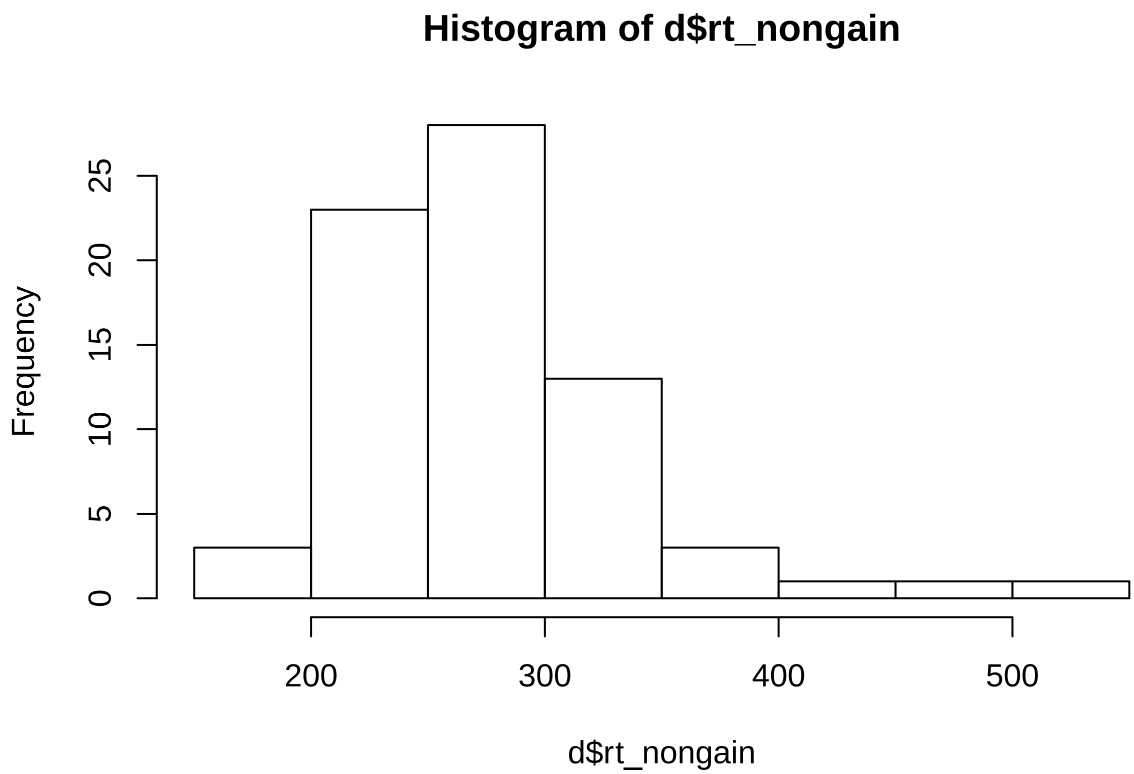

```r
summary(d$rt_loss)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     171     231     257     265     295     438
```

```r
rt_loss.out = boxplot.stats(d$rt_loss, do.out=TRUE)
rt_loss.out$out
```

```
## [1] 437.6 408.3
```

```r
hist(d$rt_nongain)
```



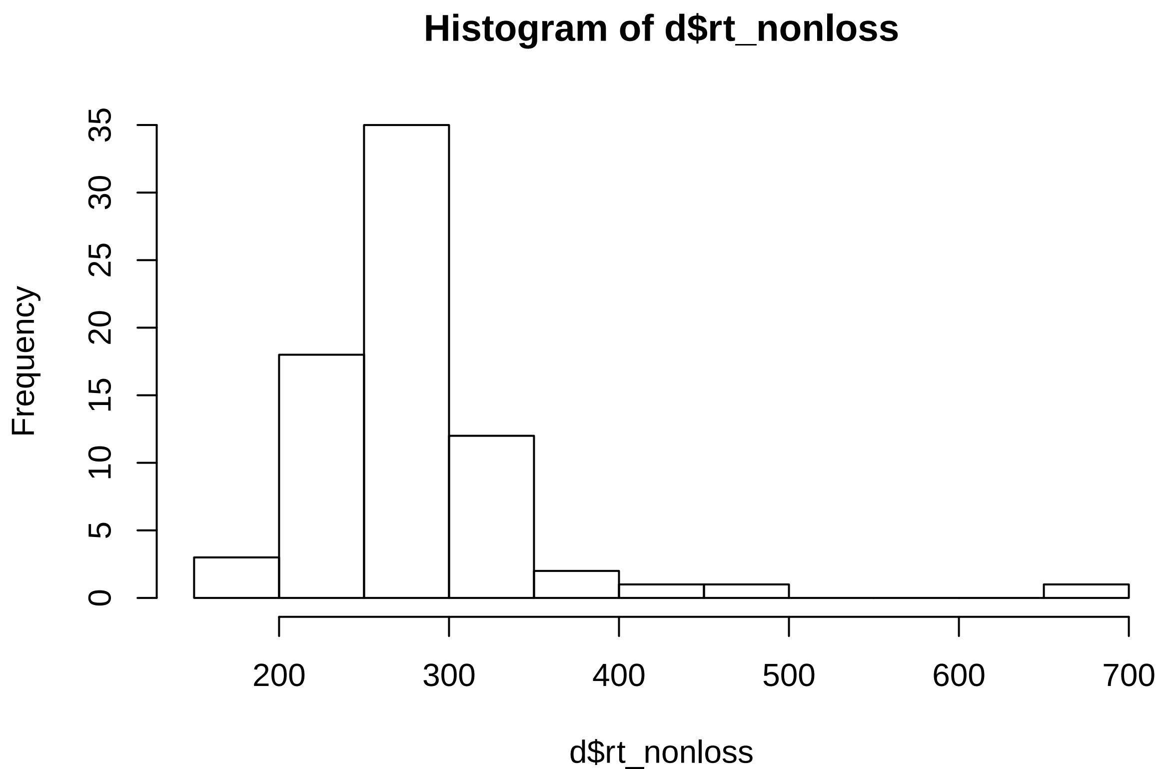

```r
summary(d$rt_nongain)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     167     231     268     275     301     502
```

```r
rt_nongain.out = boxplot.stats(d$rt_nongain, do.out=TRUE)
rt_nongain.out$out
```

```
## [1] 501.6 414.9 467.1
```

```r
hist(d$rt_nonloss)
```



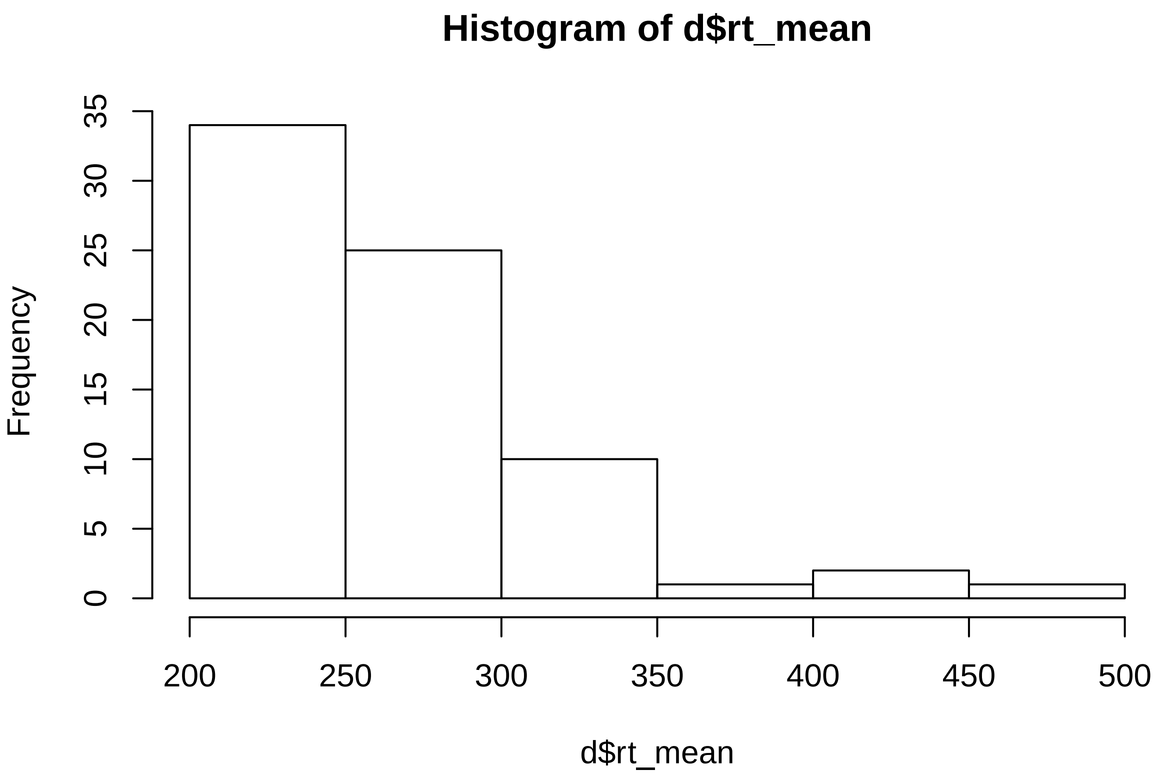

```r
summary(d$rt_nonloss)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##     192     240     263     278     298     691
```

```r
rt_nonloss.out = boxplot.stats(d$rt_nonloss, do.out=TRUE)
rt_nonloss.out$out
```

```
## [1] 394.4 451.9 691.3 407.5
```

```r
hist(d$rt_mean)
```




```
summary(d$rt_mean)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     200     234     253     267     283     484
```

```
rt_mean.out = boxplot.stats(d$rt_mean, do.out=TRUE)
rt_mean.out$out
```

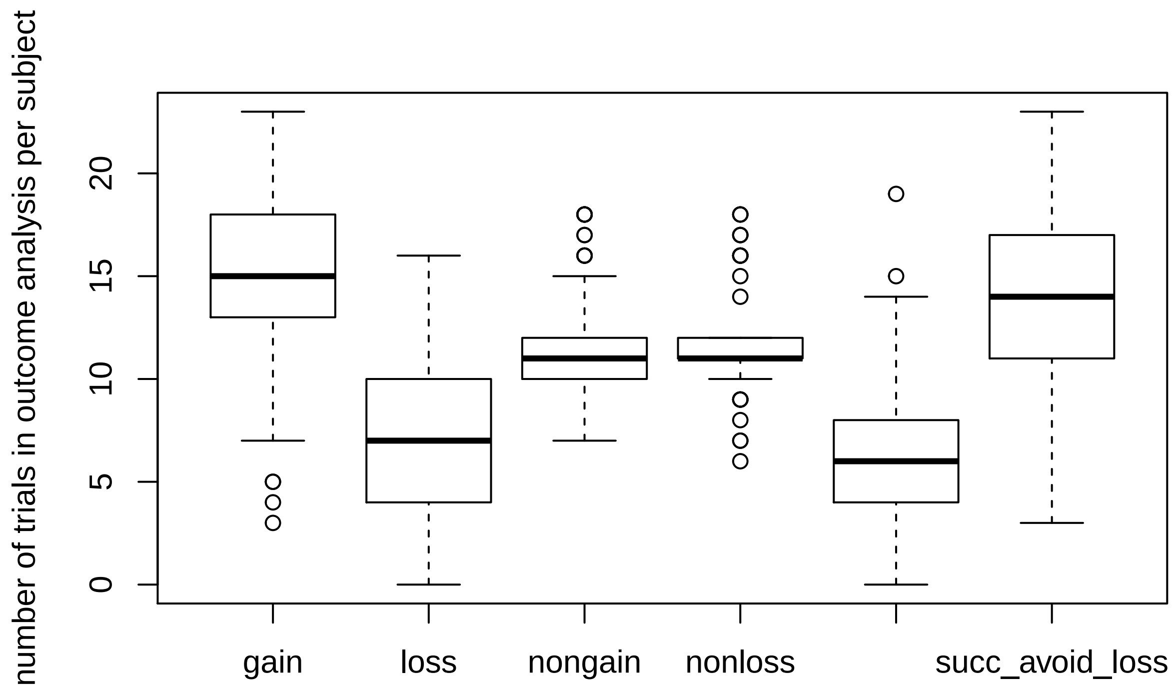
```
## [1] 374.5 430.2 484.1 409.9
```

After removing outliers (subs 18, 32, 39) mean total accuracy = 60.7%; mean gain accuracy = 66.6%; mean loss accuracy = 60.7%; mean nongain accuracy = 55.3%; mean nonloss accuracy = mean = 56.3%.

mean total rt = 267.04 ms; mean gain rt = 249.33 ms; mean loss rt = 258.64 ms; mean nongain rt = 269.65 ms; mean nonloss rt = 267.49 ms.

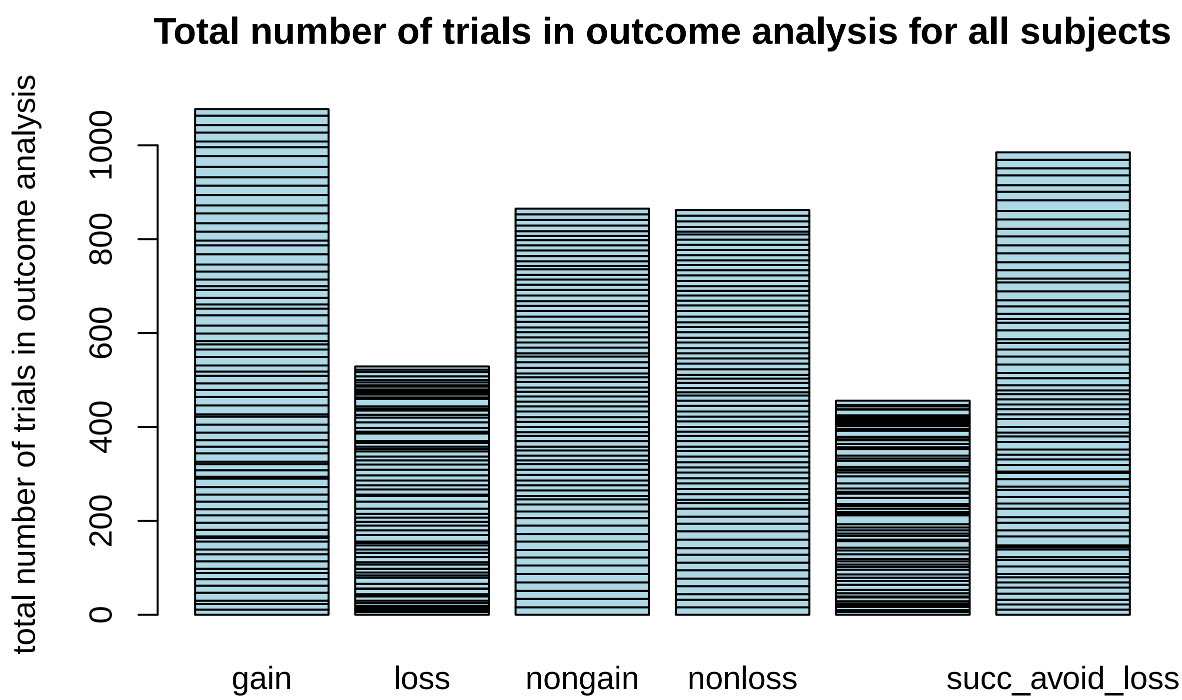
Number of trials going into the single-subject model

```
plot4 = boxplot(counts, names = c("gain","loss","nongain","nonloss","unsucc_gain","succ_avoid_loss"), yl
```



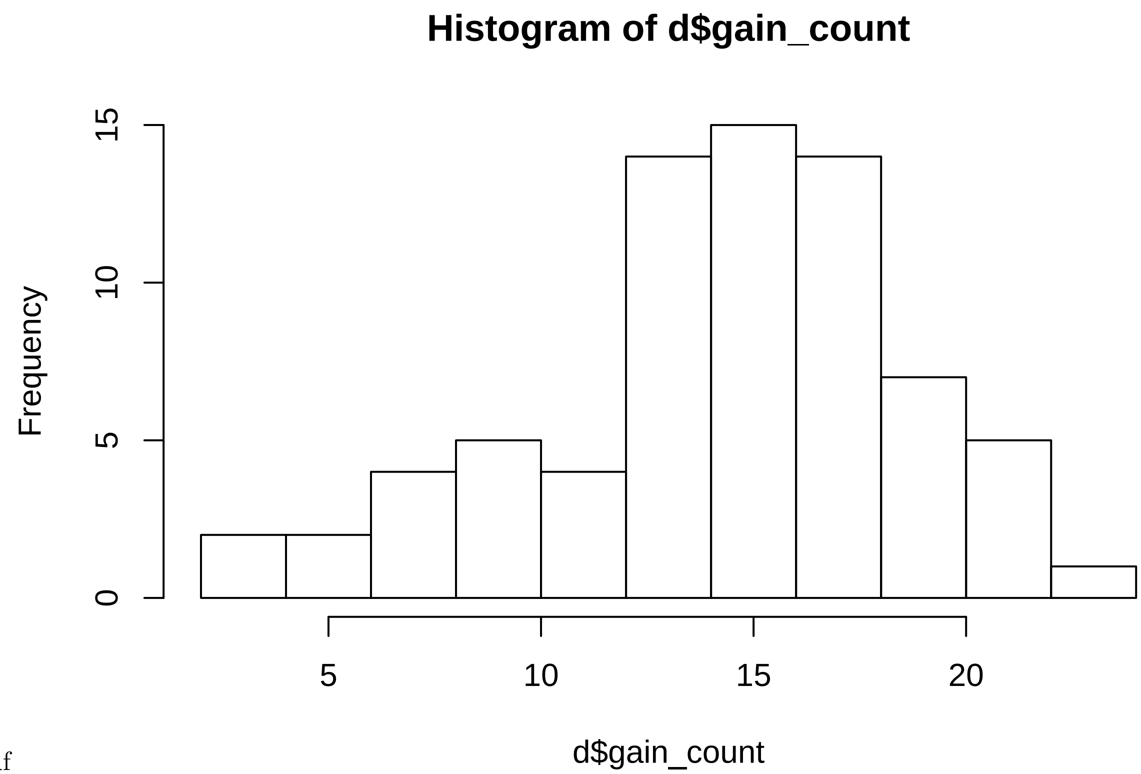
counts1.pdf

```
counts.m = as.matrix(counts)


barplot(counts.m,names.arg = c("gain","loss","nongain","nonloss","unsucc_gain","succ_avoid_loss"), ylab
```



counts2.pdf

```
hist(d$gain_count)
```



outcome counts.pdf

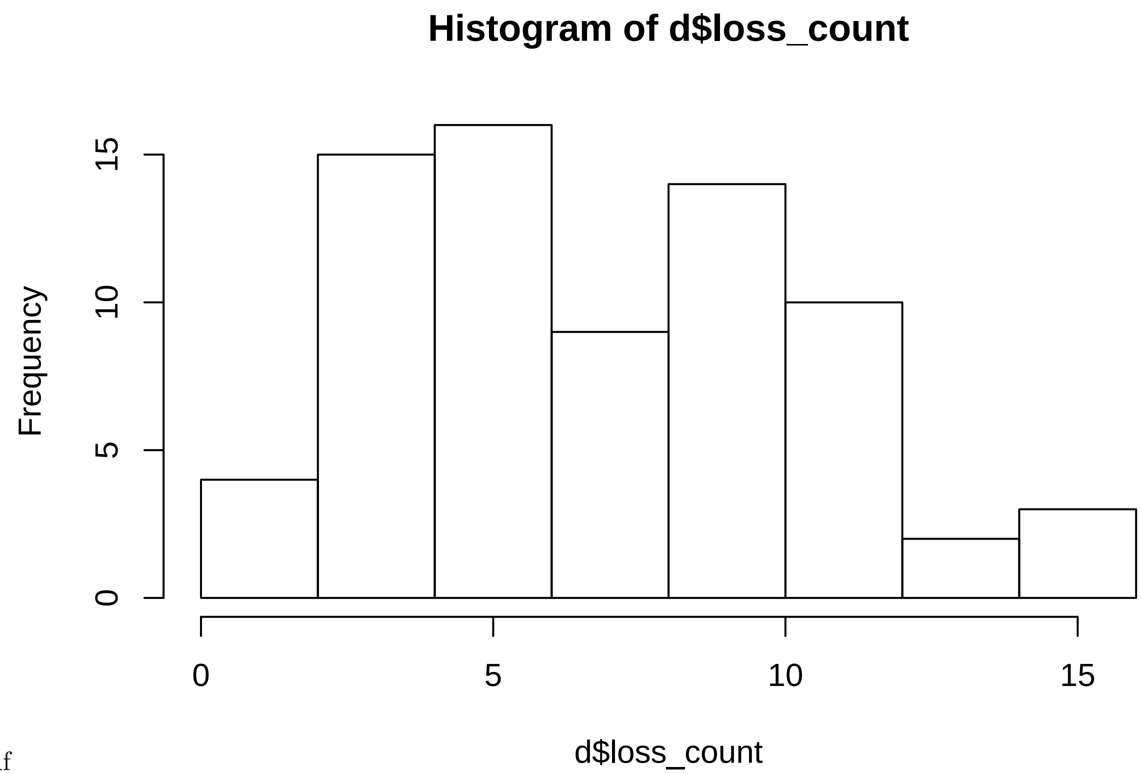
```r
summary(d$gain_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     3.0    13.0    15.0    14.8    18.0    23.0
```

```r
gain_count.out = boxplot.stats(d$gain_count, do.out=TRUE)
gain_count.out$out
```

```
## [1] 3 4 5 5
```

```r
hist(d$loss_count)
```



outcome counts.pdf

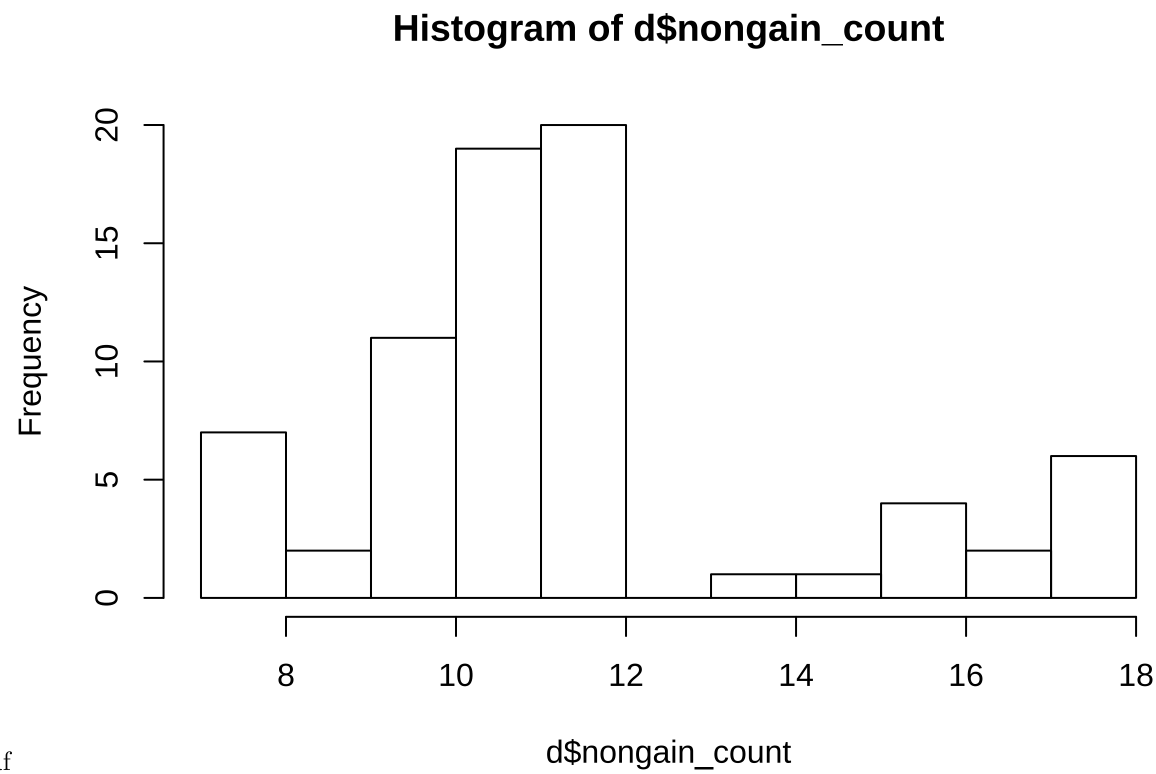
```
summary(d$loss_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    4.00    7.00    7.25   10.00   16.00
```

```
loss_count.out = boxplot.stats(d$loss_count, do.out=TRUE)
loss_count.out$out
```

```
## integer(0)
```

```
hist(d$nongain_count)
```



outcome counts.pdf

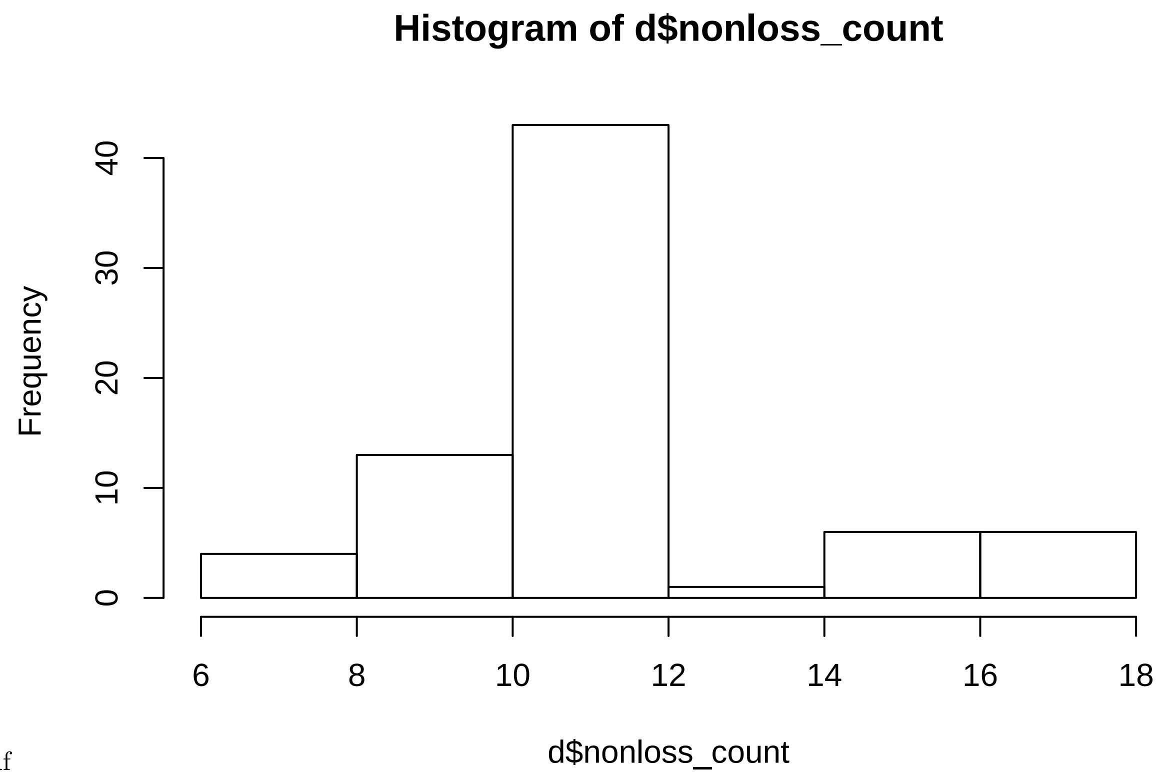
```
summary(d$nongain_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     7.0    10.0    11.0    11.8    12.0    18.0
```

```
nongain_count.out = boxplot.stats(d$nongain_count, do.out=TRUE)
nongain_count.out$out
```

```
##  [1] 16 18 17 18 18 18 17 16 18 16 18 16
```

```
hist(d$nonloss_count)
```



outcome counts.pdf

```
summary(d$nonloss_count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     6.0    11.0    11.0    11.8    12.0    18.0
```

```
nonloss_count.out = boxplot.stats(d$nonloss_count, do.out=TRUE)
nonloss_count.out$out
```

```
##  [1] 16 16 17 16 18 16 17 14 18 18 16 15 17  7  9  9  7  9  9  8  6
```

Grange and Task Version Statistics

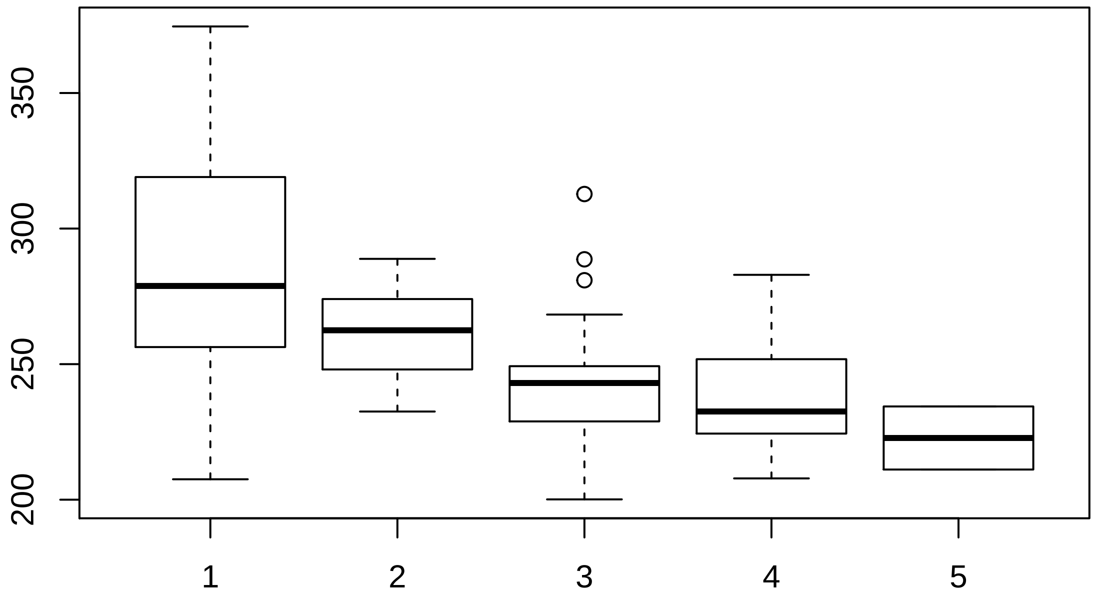
We made slight modifications to the kidmid task throughout the study. Each time the task was changed, we noted the version number. There are 4 versions - this variable is converted to a factor.

Additionally, the target times ("Grange") presented in the scanner were tailored to their predicted level of performance. There were 5 different levels of difficulty (1-5). This variable (Grange) is converted to an ordinal variable. Grange 1 represents the slowest target durations (easiest) and Grange 5 represents the fastest target durations (most difficult). The following statistics excludes outliers (subs 18, 32, and 39).

```
d1 = d[-c(14,26, 33),]

d1$version = as.factor(d1$version)
d1$Grange = ordered(d1$Grange)
```

Is there a relation between RT and Grange?

```r
plot(d1$Grange, d1$rt_mean)
```




```r
cor.test(as.numeric(d1$Grange),d1$rt_mean,use="all.obs",method="spearman") #sig
```

```
## Warning: Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  as.numeric(d1$Grange) and d1$rt_mean
## S = 86683, p-value = 4.684e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##     rho
## -0.5166
```

```r
cor.test(as.numeric(d1$Grange),d1$rt_gain,use="all.obs",method="spearman") #sig
```

```
## Warning: Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  as.numeric(d1$Grange) and d1$rt_gain
## S = 86485, p-value = 5.557e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##     rho
## -0.5132
```

```r
cor.test(as.numeric(d1$Grange),d1$rt_loss,use="all.obs",method="spearman") #sig
```

```
## Warning: Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  as.numeric(d1$Grange) and d1$rt_loss
## S = 82972, p-value = 8.67e-05
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##     rho
## -0.4517
```

```r
cor.test(as.numeric(d1$Grange),d1$rt_nongain,use="all.obs",method="spearman") #sig p = .001
```

```
## Warning: Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  as.numeric(d1$Grange) and d1$rt_nongain
## S = 78655, p-value = 0.001331
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##     rho
## -0.3762
```

```r
cor.test(as.numeric(d1$Grange),d1$rt_nonloss,use="all.obs",method="spearman") #sig p = .0002
```

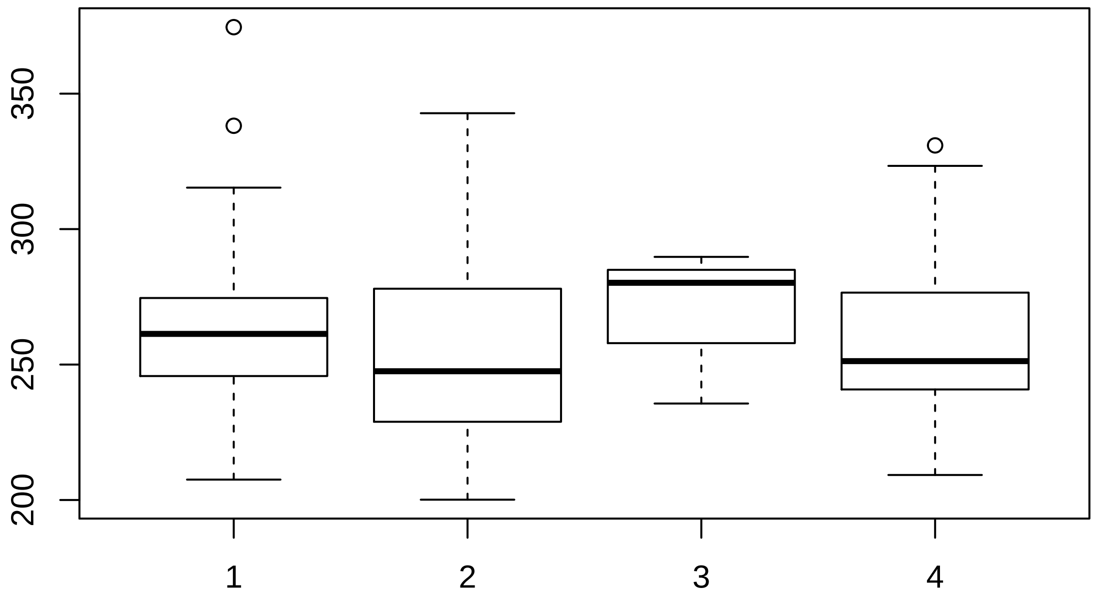
```
## Warning: Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  as.numeric(d1$Grange) and d1$rt_nonloss
## S = 81416, p-value = 0.00025
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##     rho
## -0.4245
```

Yes, highly significant for mean RT (rho = -.52, $p < .0001$), and p's $< .001$ for all conditions. The more difficult the task, the faster the RT. This is expected and uninteresting.

Is there a relation between RT and version?

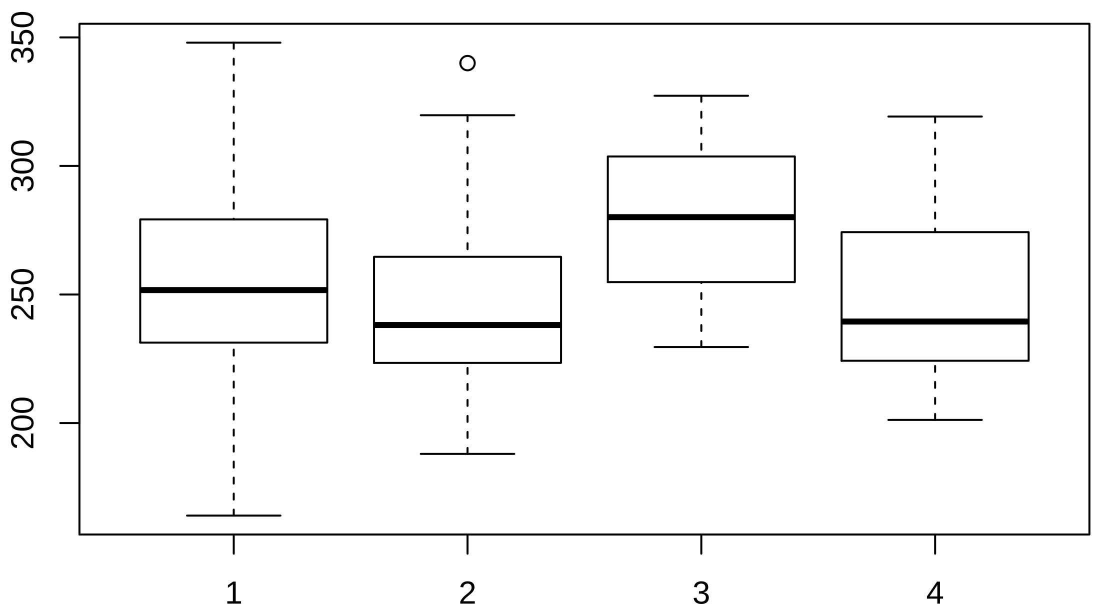
```r
plot(d1$version, d1$rt_mean)
```




```r
#summary(lm(d1$rt_mean~d1$version + d1$Grange))
#summary(lm(d1$rt_mean~d1$version))
oneway.test(d1$rt_mean ~ d1$version) #p = .67
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  d1$rt_mean and d1$version
## F = 0.531, num df = 3.000, denom df = 9.041, p-value = 0.6722
```

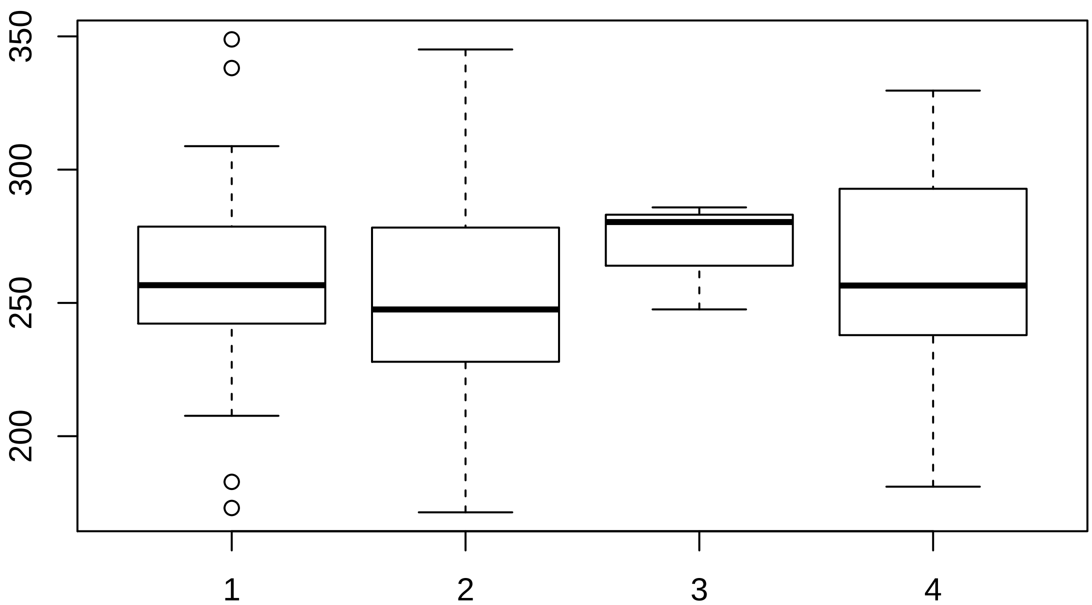
```r
plot(d1$version, d1$rt_gain)
```



```r
#summary(lm(d1$rt_gain~d1$version + d1$Grange))
oneway.test(d1$rt_gain ~ d1$version) #p .68
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  d1$rt_gain and d1$version
## F = 0.5246, num df = 3.000, denom df = 8.514, p-value = 0.6768
```

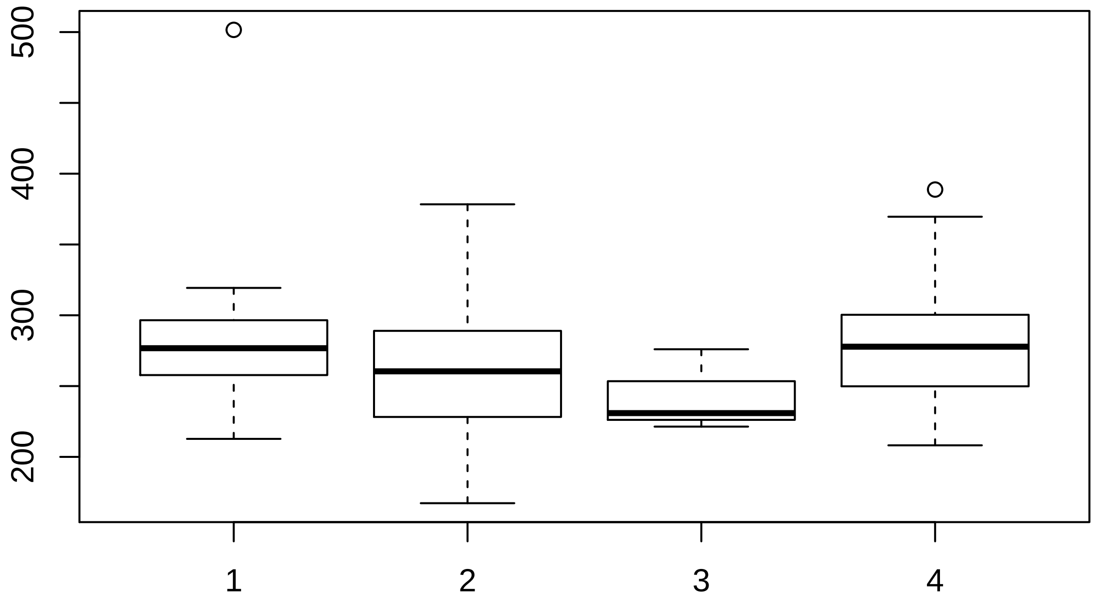
```r
plot(d1$version, d1$rt_loss)
```




```r
#summary(lm(d1$rt_loss~d1$version + d1$Grange))
oneway.test(d1$rt_loss ~ d1$version) #p = .73
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  d1$rt_loss and d1$version
## F = 0.4313, num df = 3.00, denom df = 10.66, p-value = 0.7349
```

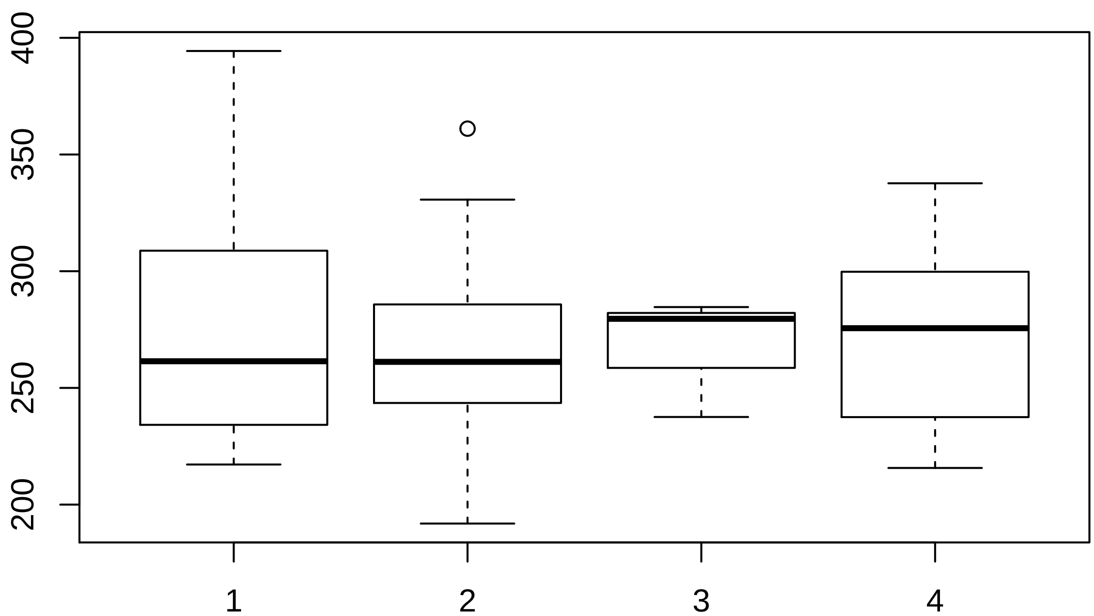
```r
plot(d1$version, d1$rt_nongain)
```




```r
#summary(lm(d1$rt_nongain~d1$version + d1$Grange))
oneway.test(d1$rt_nongain ~ d1$version) # p =.28
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  d1$rt_nongain and d1$version
## F = 1.495, num df = 3.000, denom df = 9.891, p-value = 0.2756
```

```r
plot(d1$version, d1$rt_nonloss)
```



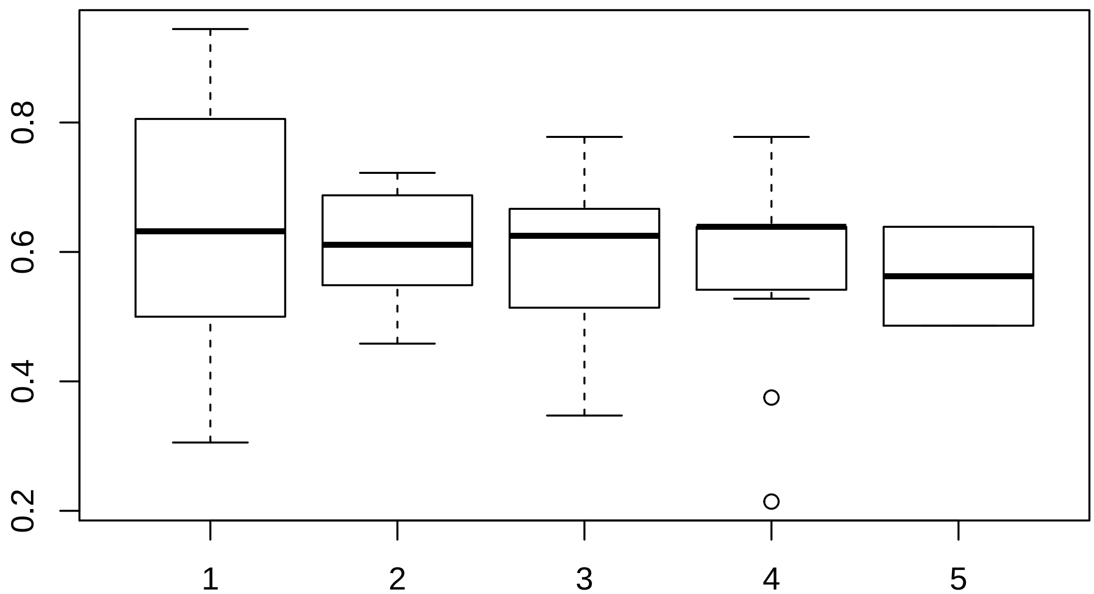

```r
#summary(lm(d1$rt_nonloss~d1$version + d1$Grange))
oneway.test(d1$rt_nonloss~ d1$version) # p = .80
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  d1$rt_nonloss and d1$version
## F = 0.3399, num df = 3.000, denom df = 9.625, p-value = 0.7971
```

No. One-way ANOVAs were performed to assess whether response time was significantly moderated by the version of the task. There is no relation between the version of the task and response times to all trials (p = .67), or response times within each condition (p's > .1).

Is there a relation between accuracy and Grange?

```r
plot(d1$Grange, d1$total_acc)
```

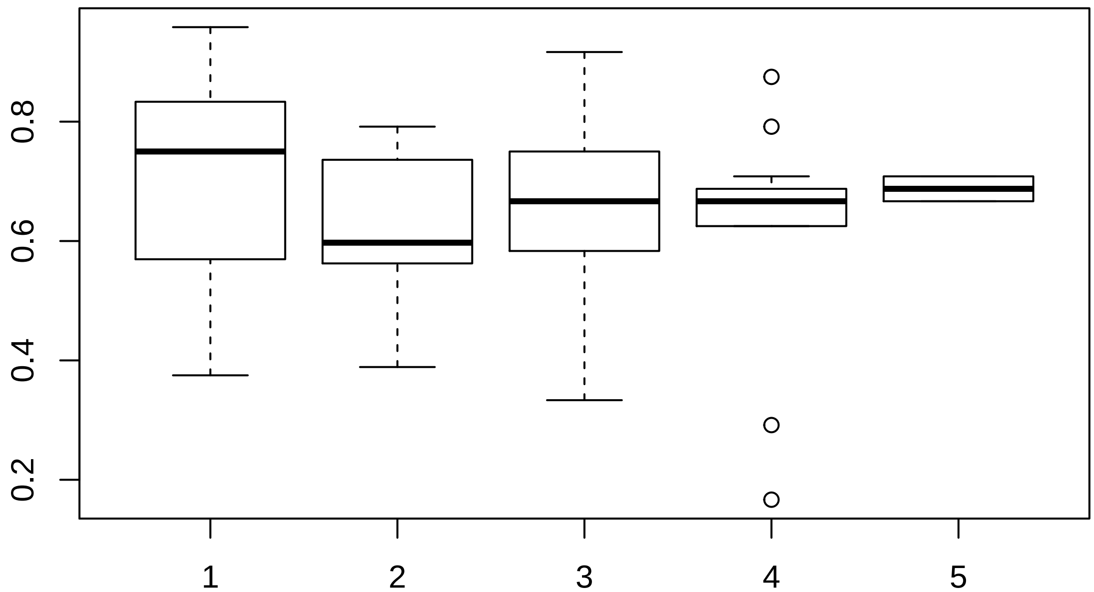



```r
cor.test(as.numeric(d1$Grange), d1$total_acc, method="spearman")
```

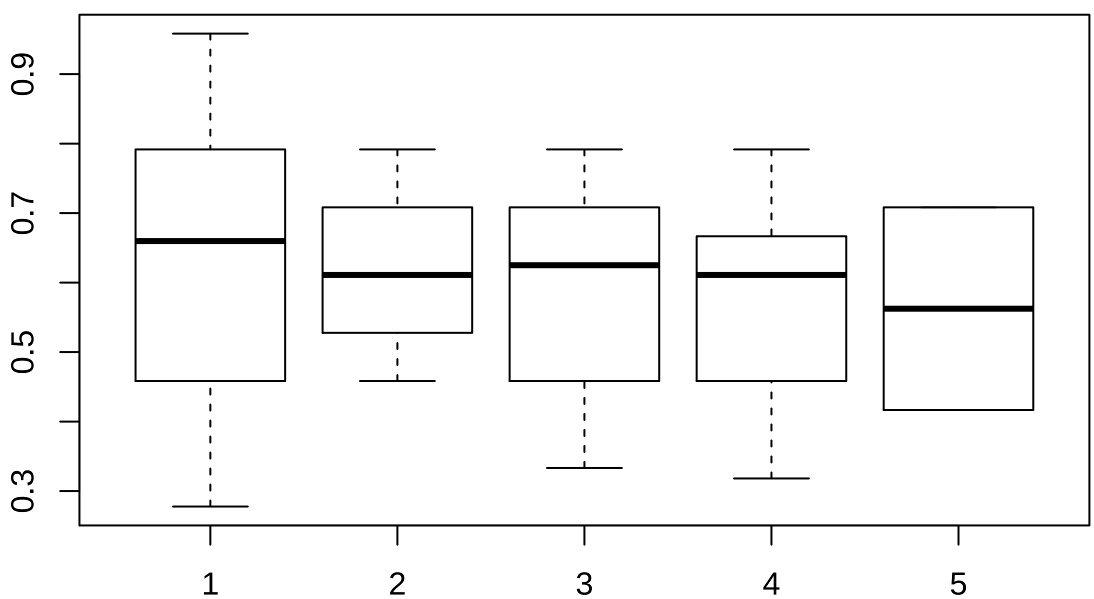
```
## Warning: Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  as.numeric(d1$Grange) and d1$total_acc
## S = 65525, p-value = 0.2264
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##     rho
## -0.1465
```

```r
plot(d1$Grange, d1$gain_acc)
```




```r
plot(d1$Grange, d1$loss_acc)
```




```r
cor.test(as.numeric(d1$Grange), d1$gain_acc, method="spearman")
```

```
## Warning: Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  as.numeric(d1$Grange) and d1$gain_acc
## S = 64861, p-value = 0.2658
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##     rho
## -0.1348
```

```r
cor.test(as.numeric(d1$Grange), d1$loss_acc, method="spearman")
```

```
## Warning: Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  as.numeric(d1$Grange) and d1$loss_acc
## S = 65422, p-value = 0.2322
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##     rho
## -0.1446
```

```r
cor.test(as.numeric(d1$Grange), d1$nongain_acc, method="spearman")
```

```
## Warning: Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  as.numeric(d1$Grange) and d1$nongain_acc
## S = 66073, p-value = 0.1971
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##    rho
## -0.156
```

```r
cor.test(as.numeric(d1$Grange), d1$nonloss_acc, method="spearman")
```

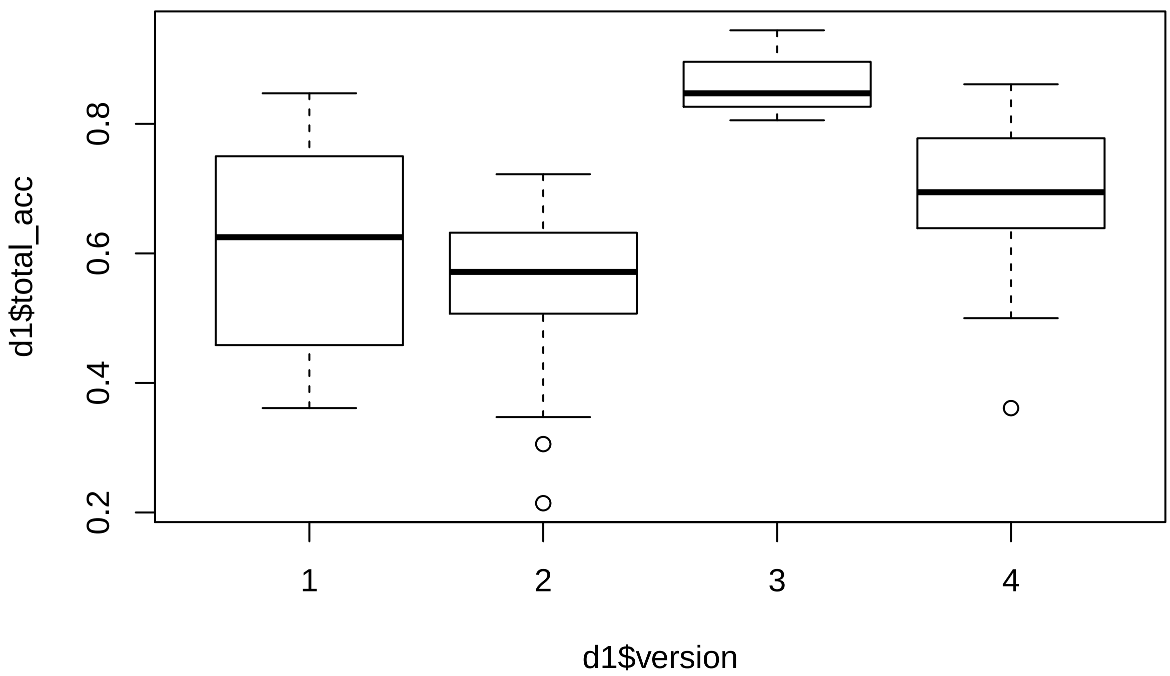
```
## Warning: Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  as.numeric(d1$Grange) and d1$nonloss_acc
## S = 68152, p-value = 0.1105
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##     rho
## -0.1924
```

No. There is no correlation between Grange (level of difficulty) and accuracy across all trials (rho = -.15, p = .23), gain trials (rho = -.13, p = .27), loss trials (rho = -.14, p =.23), nongain trials (rho = -.16, p = .20), and nonloss trials (rho = -.19, p = .11). This is a good check that across varying levels of difficulty prescribed by the GRange, participants performed with equivalent accuracy.

Is there a relation between accuracy and version?

```r
plot(d1$total_acc ~ d1$version)
```




```r
oneway.test(d1$total_acc ~ d1$version) #sig
```

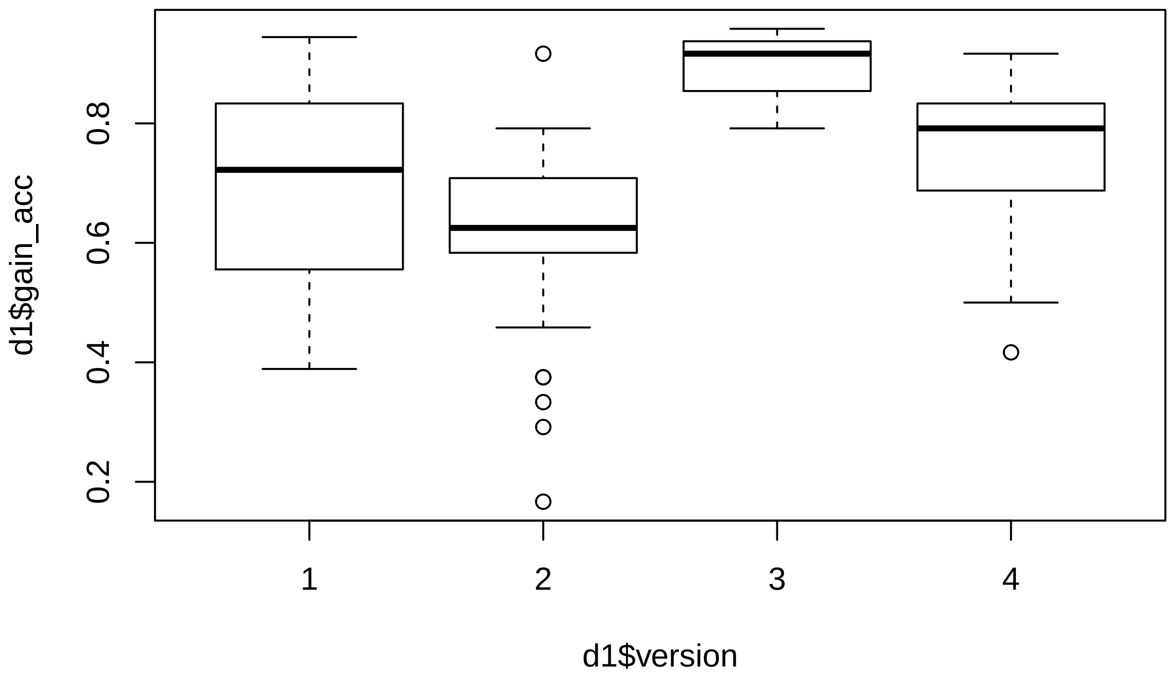
```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  d1$total_acc and d1$version
## F = 15.01, num df = 3.000, denom df = 9.763, p-value = 0.0005455
```

```r
summary(lm(d1$total_acc~d1$version + d1$Grange)) #sig
```

```
##
## Call:
## lm(formula = d1$total_acc ~ d1$version + d1$Grange)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3266 -0.0547  0.0074  0.0870  0.2400
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.60860    0.04485   13.57   <2e-16 ***
## d1$version2 -0.05096    0.04743   -1.07   0.2868
## d1$version3  0.25850    0.08456    3.06   0.0033 **
## d1$version4  0.07373    0.05229    1.41   0.1636
## d1$Grange.L -0.01149    0.06573   -0.17   0.8618
## d1$Grange.Q  0.00137    0.05614    0.02   0.9805
## d1$Grange.C  0.03281    0.04646    0.71   0.4827
## d1$Grange^4  0.00850    0.03545    0.24   0.8113
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.13 on 62 degrees of freedom
## Multiple R-squared:  0.27,   Adjusted R-squared:  0.187
## F-statistic: 3.27 on 7 and 62 DF,  p-value: 0.00508
```

```
plot(d1$gain_acc ~ d1$version)
```




```
oneway.test(d1$gain_acc ~ d1$version) #sig
```

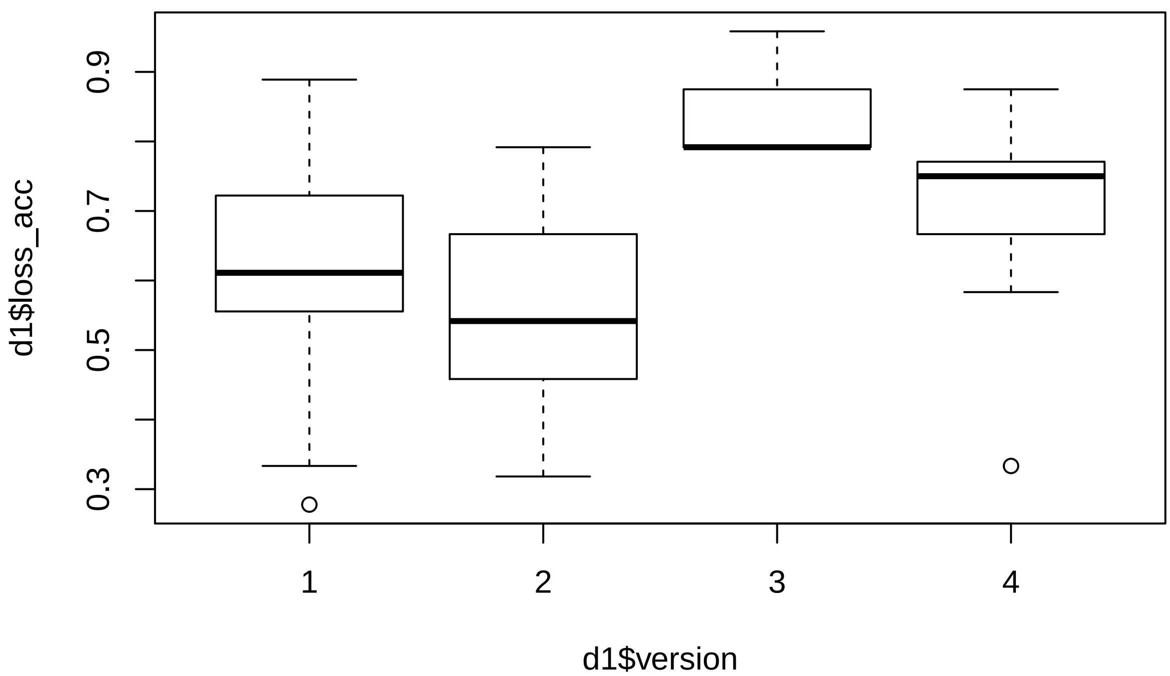
```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  d1$gain_acc and d1$version
## F = 8.76, num df = 3.000, denom df = 9.957, p-value = 0.00382
```

```
summary(lm(d1$gain_acc~d1$version + d1$Grange)) #version 2 and 3 are marginally different from version
```

```
##
## Call:
## lm(formula = d1$gain_acc ~ d1$version + d1$Grange)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3951 -0.0499  0.0306  0.0907  0.2834
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.7200     0.0533   13.50   <2e-16 ***
```

```
## d1$version2  -0.1073     0.0564   -1.90    0.062 .
## d1$version3   0.1798     0.1006    1.79    0.079 .
## d1$version4   0.0331     0.0622    0.53    0.597
## d1$Grange.L   0.0486     0.0782    0.62    0.536
## d1$Grange.Q   0.0457     0.0668    0.68    0.496
## d1$Grange.C   0.0382     0.0553    0.69    0.492
## d1$Grange^4   0.0626     0.0422    1.49    0.143
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.155 on 62 degrees of freedom
## Multiple R-squared:  0.227,  Adjusted R-squared:  0.14
## F-statistic: 2.61 on 7 and 62 DF,  p-value: 0.02
```

```r
plot(d1$loss_acc ~ d1$version)
```




```r
oneway.test(d1$loss_acc ~ d1$version) #sig
```

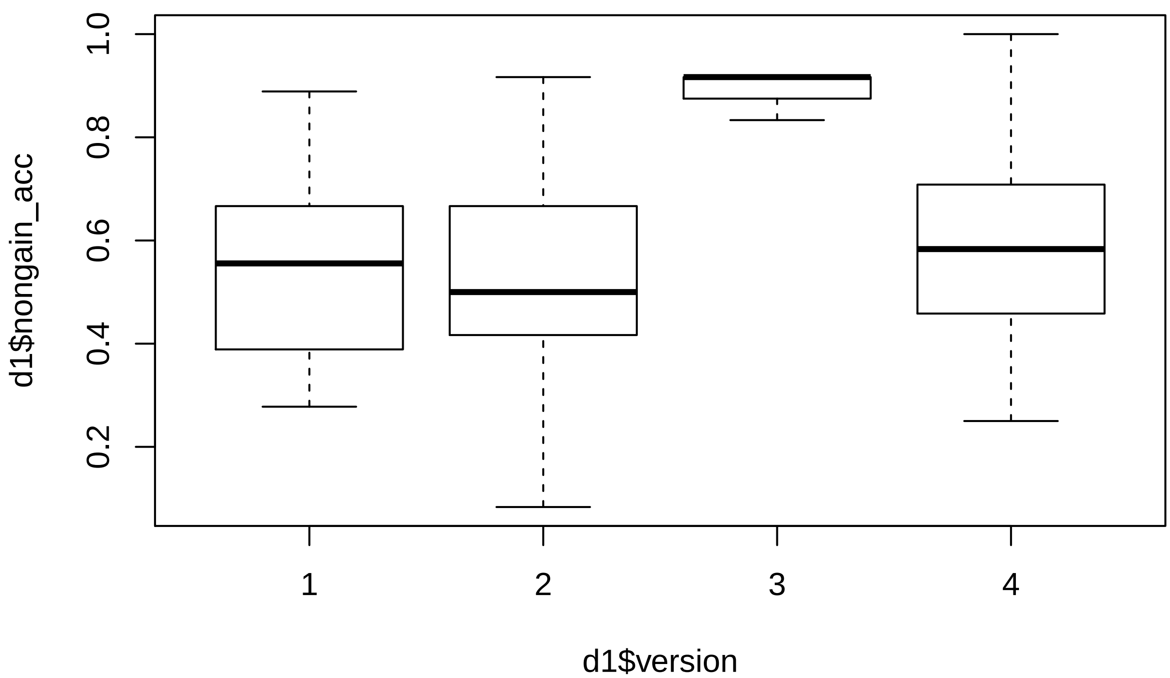
```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  d1$loss_acc and d1$version
## F = 10.21, num df = 3.00, denom df = 9.37, p-value = 0.002629
```

```r
summary(lm(d1$loss_acc~d1$version + d1$Grange)) #version 3 is more accurate on loss trials
```

```
##
## Call:
## lm(formula = d1$loss_acc ~ d1$version + d1$Grange)
##
```

```
## Residuals:
##      Min       1Q  Median      3Q     Max
## -0.3825 -0.0743  0.0068  0.1100  0.2826
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.62321    0.05050   12.34   <2e-16 ***
## d1$version2 -0.07450    0.05341   -1.39    0.168
## d1$version3  0.24096    0.09523    2.53    0.014 *
## d1$version4  0.08854    0.05888    1.50    0.138
## d1$Grange.L  0.00278    0.07401    0.04    0.970
## d1$Grange.Q -0.00362    0.06321   -0.06    0.955
## d1$Grange.C  0.04305    0.05232    0.82    0.414
## d1$Grange^4  0.00303    0.03992    0.08    0.940
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.147 on 62 degrees of freedom
## Multiple R-squared:  0.263,  Adjusted R-squared:  0.18
## F-statistic: 3.16 on 7 and 62 DF,  p-value: 0.00635
```

```
plot(d1$nongain_acc ~ d1$version)
```



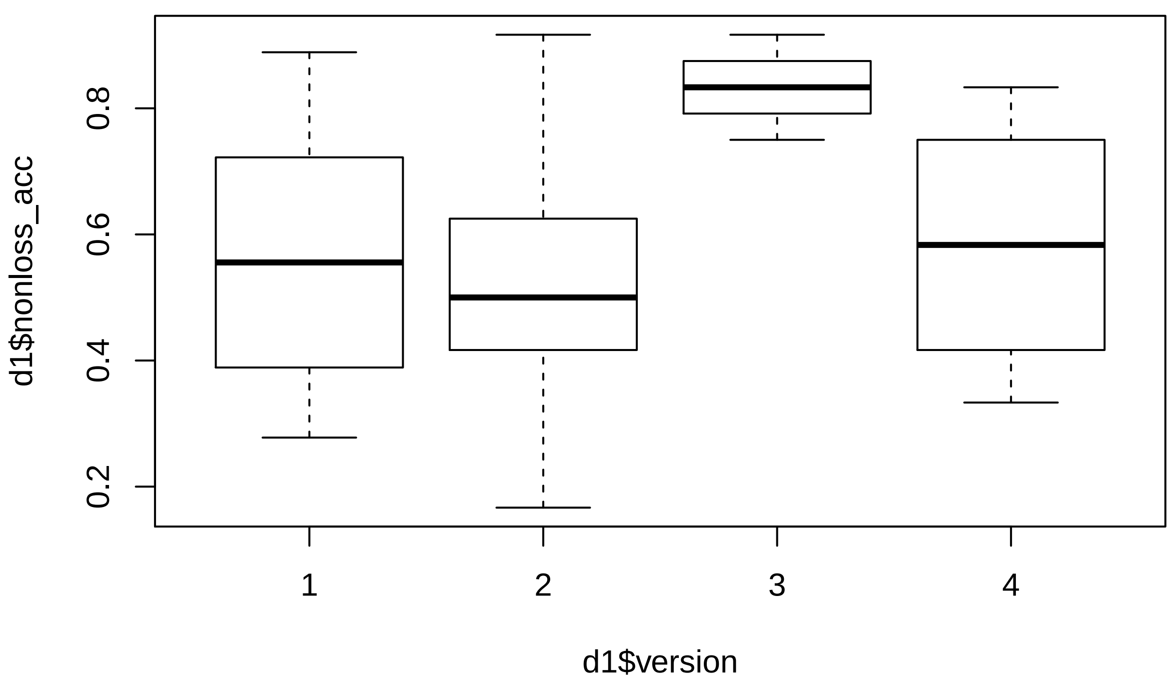

```
oneway.test(d1$nongain_acc ~ d1$version) #sig
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  d1$nongain_acc and d1$version
## F = 29.84, num df = 3.00, denom df = 16.51, p-value = 6.773e-07
```

```r
summary(lm(d1$nongain_acc~d1$version + d1$Grange)) #version 3 is more accurate on nongain trials
```

```
##
## Call:
## lm(formula = d1$nongain_acc ~ d1$version + d1$Grange)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4212 -0.1306  0.0052  0.1144  0.3786
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5318     0.0658    8.09  2.8e-11 ***
## d1$version2  -0.0143     0.0695   -0.21   0.8378
## d1$version3   0.3338     0.1240    2.69   0.0091 **
## d1$version4   0.0662     0.0767    0.86   0.3911
## d1$Grange.L  -0.0476     0.0964   -0.49   0.6233
## d1$Grange.Q   0.0176     0.0823    0.21   0.8309
## d1$Grange.C   0.0306     0.0681    0.45   0.6551
## d1$Grange^4  -0.0546     0.0520   -1.05   0.2976
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.191 on 62 degrees of freedom
## Multiple R-squared:  0.189,  Adjusted R-squared:  0.0977
## F-statistic: 2.07 on 7 and 62 DF,  p-value: 0.0605
```

```r
plot(d1$nonloss_acc ~ d1$version)
```



```r
oneway.test(d1$nonloss_acc ~ d1$version) #sig
```

```
##
##  One-way analysis of means (not assuming equal variances)
##
## data:  d1$nonloss_acc and d1$version
## F = 9.082, num df = 3.00, denom df = 10.71, p-value = 0.00278
```

```r
summary(lm(d1$nonloss_acc~d1$version + d1$Grange)) #version 3 is more accurate on nonloss trials
```

```
##
## Call:
## lm(formula = d1$nonloss_acc ~ d1$version + d1$Grange)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3710 -0.1210  0.0013  0.1239  0.3383
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.52354    0.06073    8.62  3.3e-12 ***
## d1$version2 -0.01774    0.06423   -0.28    0.783
## d1$version3  0.27498    0.11451    2.40    0.019 *
## d1$version4  0.04085    0.07081    0.58    0.566
## d1$Grange.L -0.13184    0.08900   -1.48    0.144
## d1$Grange.Q -0.08547    0.07601   -1.12    0.265
## d1$Grange.C  0.00181    0.06291    0.03    0.977
## d1$Grange^4 -0.01931    0.04800   -0.40    0.689
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.177 on 62 degrees of freedom
## Multiple R-squared:  0.16,   Adjusted R-squared:  0.0653
## F-statistic: 1.69 on 7 and 62 DF,  p-value: 0.128
```

Yes, version 3 is associated with greater accuracy (hit rate) than the other versions when controlling for Grange ($F(7,62) = 3.27$, $p = .005$).

Note: The 3rd version of the task has been administered 3 times, and all times this was done with Grange 1, so Grange is highly correlated with RT for the 3rd version. Also, the 3rd version added 50 ms to each target duration, which was not mirrored in the practice, so participants who received this version ($n = 3$) were significantly more accurate than those who received the other versions.