

Exam 3 (Solution)

Prateek Kumar prateekk@mtu.edu

December 18, 2018

Exam 3: Where's Waldo

The data set includes an experiment with 120 participants, each of whom completed 45 repeated measurements of a visual search task across a number of conditions. In this task, they were shown a target, and then asked to find the target in a complex screen. Sometimes the target would be in that screen, and other times it would not (the 'type' column). They would first determine if the target was there or not, and then click on the target. There were a number of different targets (probeFile) and different screens (baseimage). Also we could measure the overall size of the target (size) in pixels, and centrality, in pixels from the center of the screen. We measured overall time to find the target (responsetime), and whether their response was correct (corr) for each trial.

```
##recode the factors that look like numbers
data <- read.csv("training-pooled.csv") #reading the csv file
#head(data) #displaying the top few rows of the data

# converting subnum, probeCond and probe into factors because
data$subnum <- as.factor(data$subnum)
data$probeCond <- as.factor(data$probeCond)
data$probe <- as.factor(data$probe)
```

We read the dataset("training-pooled.csv") and we converted subnum, probeCond and probe into factors because factors represent a very efficient way to store character values, because each unique character value is stored only once.

1. Identifying outliers and influential points and transforming dependent measure

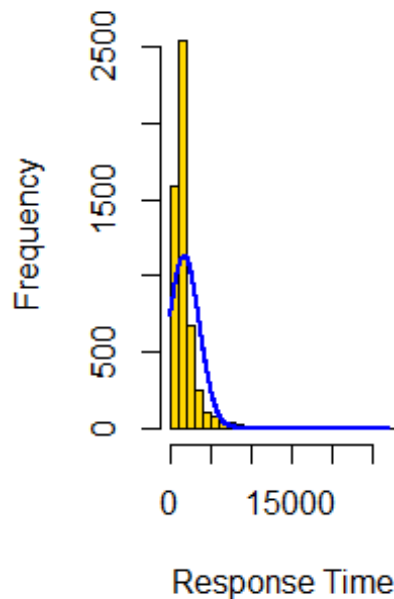
To begin, we want to use response time as the dependent measure.

```
par(mfrow=c(1,2))

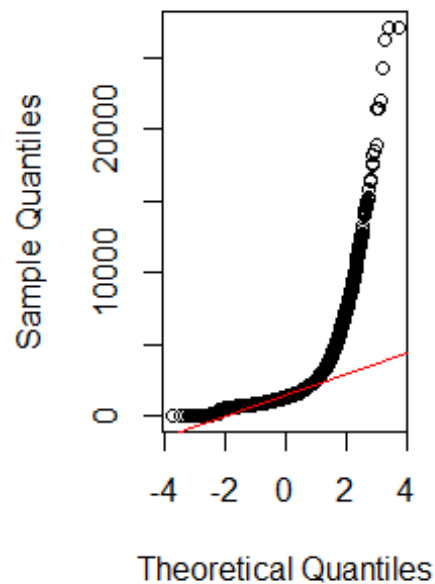
library(rcompanion)
plotNormalHistogram(data$responsetime, breaks = 20, col = "gold", border = 1,
                    main = "Histogram of Response Time", xlab = "Response
Time")

qqnorm(data$responsetime)
qqline(data$responsetime, col = 2)
```

Histogram of Response T



Normal Q-Q Plot

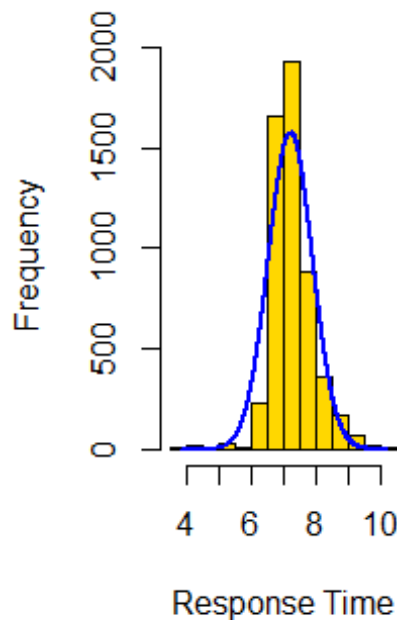


However, we suspect that it is non-normal. The histogram shows that the data is right skewed i.e. we have positive skewness so in order to normalize it we will use log transform.

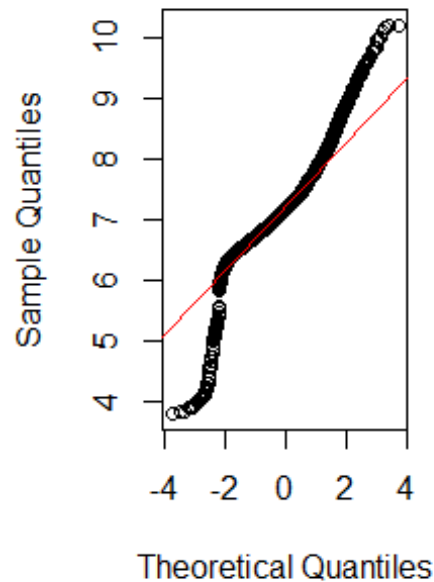
```
RT_log = log(data$responsetime) #Log transform
par(mfrow=c(1,2))
plotNormalHistogram(RT_log, breaks = 20, col = "gold", border = 1,
                     main = "Histogram of Response Time", xlab = "Response
Time")

qqnorm(RT_log)
qqline(RT_log, col = 2)
```

Histogram of Response T



Normal Q-Q Plot



Now our data is normalized as we can see in the histogram but looking at the qqnorm plot we can suspect some highly influential outliers so we need to remove them because those values are distant from remaining observations. As a result, they can potentially skew or bias our analysis performed on the dataset. It is therefore very important to detect and adequately deal with outliers.

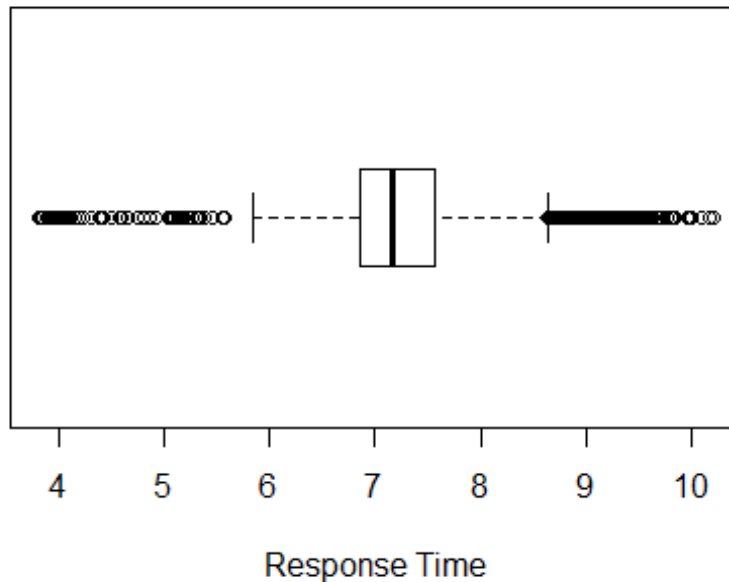
```
#adding the value to the data frame
```

```
data$responsetime_log <- RT_log
```

```
#Detecting outliers overall, univariate approach
```

```
outlier_values <- boxplot.stats(data$responsetime_log)$out # outlier values.  
boxplot(data$responsetime_log, main="Response Time Boxplot", boxwex=0.5,  
horizontal = T, xlab = "Response Time")
```

Response Time Boxplot



```
#mtext(paste("Outliers: ", paste(outlier_values, collapse=" ")), cex=0.6)
```

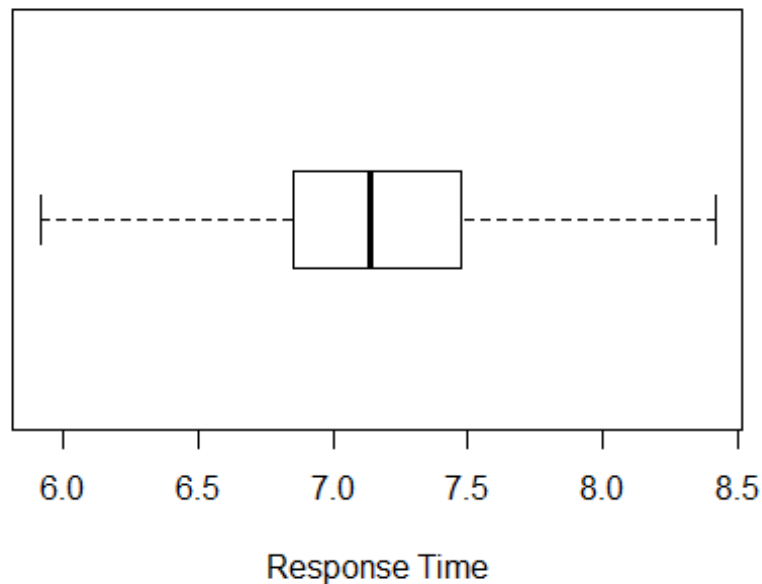
Firstly we check for the overall outliers. On looking at the box plot we see the outlier points which are outside the whiskers of the boxplot which is the 1.5 times the IQR.

```
#converting the outliers to NA values
```

```
while(length(outlier_values)!=0){  
  outlier_values <- boxplot.stats(data$responsetime_log)$out  
  i=j=0  
  for(i in data$responsetime_log){  
    j=j+1  
    if(i %in% outlier_values){  
      data$responsetime_log[j] <- NA  
    }  
  }  
}
```

```
boxplot(data$responsetime_log, main="Response Time boxplot", boxwex=0.5,  
horizontal = T, xlab = "Response Time")
```

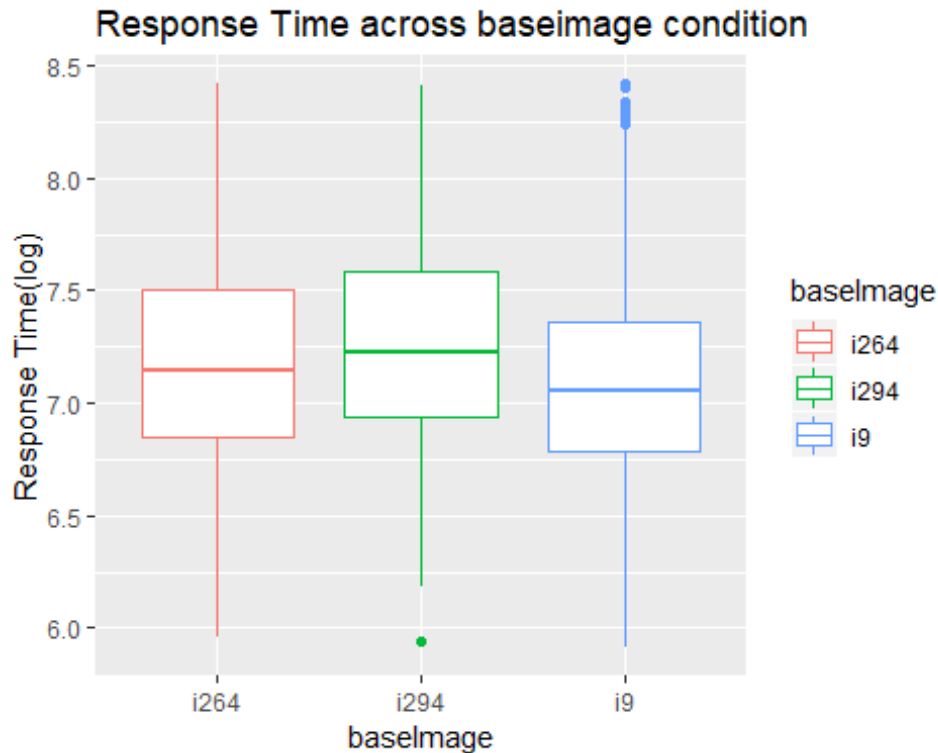
Response Time boxplot



Now, we removed the overall outliers and we can see the changes in our boxplot. Now we don't have any points outside the whiskers of the boxplot.

Now we need to check for the outliers for each baseimage condition as well.

```
#Detecting outliers for each baseimage condition, bivariate approach  
#boxplot(RT_log ~ baseImage, data = data, main="Response Time across  
baseimage condition") # clear pattern is noticeable.  
library(ggplot2)  
ggplot(data, aes(x = baseImage, y = responsetime_log, color = baseImage)) +  
  geom_boxplot() +  
  labs(title="Response Time across baseimage condition", x = "baseImage", y =  
    "Response Time(log)")  
  
## Warning: Removed 376 rows containing non-finite values (stat_boxplot).
```



Using the bivariate approach we can see that we have outliers in the “i294” and “i9” baseimage conditions.

```
#We see we have outliers in i294 and i9 so removing their outliers
#i264
data_i294 <- data[data$baseImage %in% c("i294"), ]
outlier_values <- boxplot.stats(data_i294$responsetime_log)$out
while(length(outlier_values)!=0){
  data_i294 <- data[data$baseImage %in% c("i294"), ]
  outlier_values <- boxplot.stats(data_i294$responsetime_log)$out
  i=j=0
  for(i in data$responsetime_log){
    j=j+1
    if(data$baseImage[j] %in% c("i294")){
      if(i %in% outlier_values){
        data$responsetime_log[j] <- NA
      }
    }
  }
}

#i9
data_i9 <- data[data$baseImage %in% c("i9"), ]
outlier_values <- boxplot.stats(data_i9$responsetime_log)$out
while(length(outlier_values)!=0){
  data_i9 <- data[data$baseImage %in% c("i9"), ]
  outlier_values <- boxplot.stats(data_i9$responsetime_log)$out
  i=j=0
  for(i in data$responsetime_log){
```

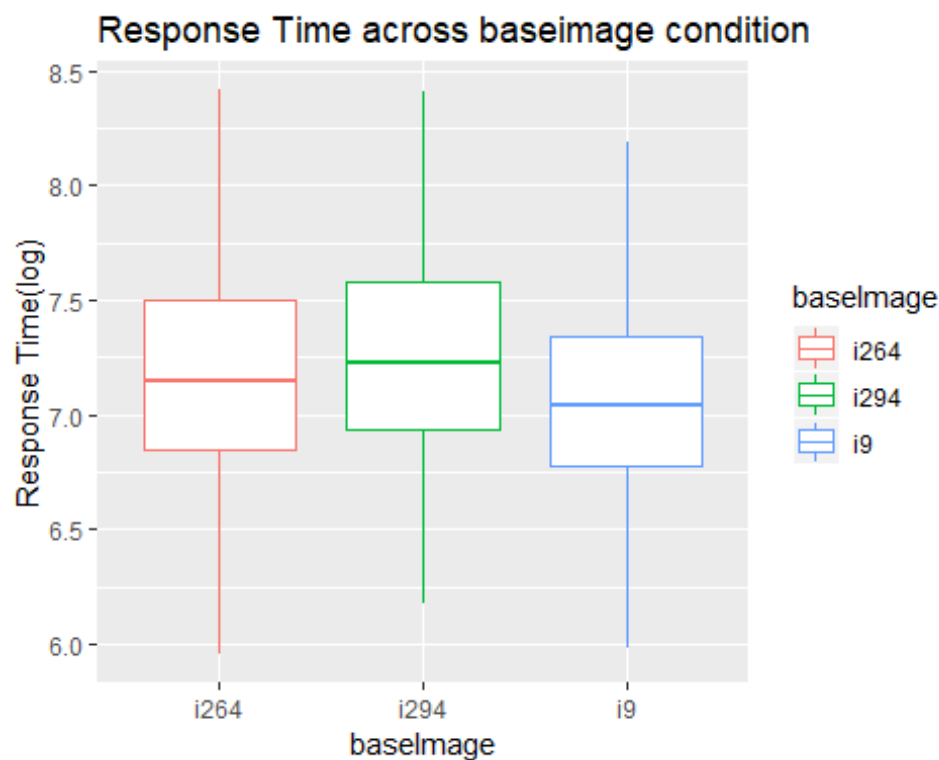
```

j=j+1
if(data$baseImage[j] %in% c("i9")){
  if(i %in% outlier_values){
    data$responsetime_log[j] <- NA
  }
}
}}

ggplot(data, aes(x = baseImage, y = responsetime_log, color = baseImage)) +
  geom_boxplot() +
  labs(title="Response Time across baseimage condition",x = "baseImage", y =
"Response Time(log)")

## Warning: Removed 414 rows containing non-finite values (stat_boxplot).

```



We now removed the outliers with respect to each baseImage condition as well. Now again we can see that we do not have any points outside the whiskers.

So now we are free from the outliers. Check the histogram and the qqnorm plots again.

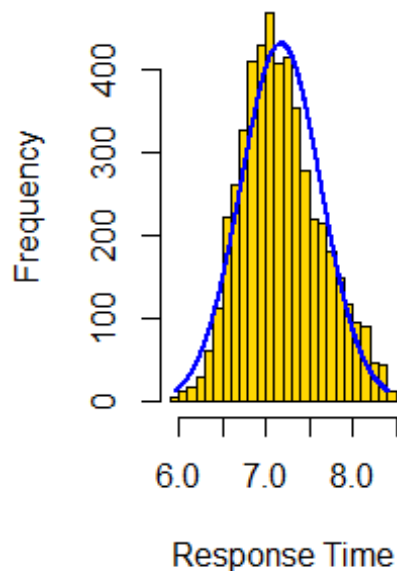
```

#After removing all the outliers
par(mfrow=c(1,2))
plotNormalHistogram(data$responsetime_log, breaks = 20, col = "gold", border
= 1,
                    main = "Histogram of Response Time", xlab = "Response
Time")

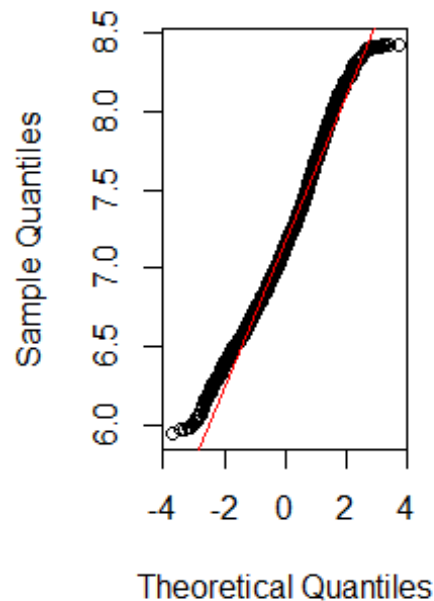
```

```
qqnorm(data$responsetime_log)
qqline(data$responsetime_log, col = 2)
```

Histogram of Response T



Normal Q-Q Plot



We can now see from the histogram that our data is properly normalized and our normal qq plot confirms it.

2. ANOVA for base image condition

For this question, we have to pretend that we did not have repeated measures (ignoring the subnum variable). We build our anova model with the transformed Response Time as a dependent measure.

Here we are considering only the correct responses so we have to get rid of the values who have the correlation value as 0.

#Finding the columns which are of factor type

```
sapply(data, class) == "factor"
```

##	subnum	trial	baseImage	stimCode
##	TRUE	FALSE	TRUE	FALSE
##	type	probeCond	probe	x
##	TRUE	TRUE	TRUE	FALSE
##	y	probeFile	clickResp	corr
##	FALSE	TRUE	FALSE	FALSE
##	time1	time2	responsetime	size
##	FALSE	FALSE	FALSE	FALSE


```
##          Centrality responsetime_log
##          FALSE                     FALSE

#For each of the factors, recoding with polynomial contrasts inorder to
examine interactions.
#contrasts(data$subnum) <- contr.poly(levels(data$subnum))
contrasts(data$baseImage) <- contr.poly(levels(data$baseImage))
contrasts(data$type) <- contr.poly(levels(data$type))
contrasts(data$probeCond) <- contr.poly(levels(data$probeCond))
contrasts(data$probe) <- contr.poly(levels(data$probe))
contrasts(data$probeFile) <- contr.poly(levels(data$probeFile))

#filtering out those whose corr values is 0.
data_Q2 <- data[data$corr %in% c(1), ]
```

We firstly for each of the factors, recoded with polynomial contrasts so that we can examine interactions. Now including baseimage, type, and a baseImage x type interaction, and size and centrality as numerical covariates we build our model.

```
m_01 <- aov(responsetime_log ~ baseImage + type + baseImage:type + size +
Centrality, data = data_Q2)
summary(m_01)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
baseImage	2	24.9	12.433	67.418	< 2e-16	***
type	1	0.1	0.097	0.529	0.467270	
size	1	1.4	1.392	7.548	0.006030	**
Centrality	1	6.9	6.906	37.447	1.01e-09	***
baseImage:type	2	2.8	1.385	7.508	0.000555	***
Residuals	4797	884.6	0.184			

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

We now use a type-II ANOVA to determine which variables are significant.

```
library(car)

## Loading required package: carData

Anova(m_01)
```

Anova Table (Type II tests)					
	Sum Sq	Df	F value	Pr(>F)	
baseImage	0.26	2	0.6930	0.5001174	
type	0.09	1	0.4852	0.4861240	
size	0.42	1	2.2554	0.1332150	
Centrality	7.01	1	38.0267	7.551e-10	***
baseImage:type	2.77	2	7.5079	0.0005552	***
Residuals	884.64	4797			

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

#This shows that base image is not significant and size is just over significant value
```

On looking at the results of Type-II Anova and examining the p-values we see that only the centrality and baseImage and type interaction are significant with the p-values < 0.05

Now we conduct Type-III Anova on the model.

```
Anova(m_01, type = "III")

## Anova Table (Type III tests)
##
## Response: responsetime_log
##          Sum Sq   Df  F value    Pr(>F)
## (Intercept) 342.21   1 1855.6442 < 2.2e-16 ***
## baseImage    0.35    2   0.9405 0.3905059
## type         0.07    1   0.3528 0.5525775
## size         0.42    1   2.2554 0.1332150
## Centrality   7.01    1  38.0267 7.551e-10 ***
## baseImage:type 2.77    2   7.5079 0.0005552 ***
## Residuals   884.64 4797
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Also on conducting a type-III ANOVA, we get the same result i.e. centrality and type x baseimage interaction is only significant.

For the Type-II model we now compute post-hoc test for baseImage and the baseimage by type interaction to determine which levels differed from each other.

```
#For the Type-II model, compute post-hoc tests for baseImage and the baseimage by type interaction to determine which levels differed from each other
TukeyHSD(aov(responsetime_log ~ baseImage + baseImage:type, data = data_Q2))

## Tukey multiple comparisons of means
## 95% family-wise confidence level
##
## Fit: aov(formula = responsetime_log ~ baseImage + baseImage:type, data = data_Q2)
##
## $baseImage
##          diff          lwr          upr    p adj
## i294-i264  0.07414541  0.0380996  0.11019122 4.4e-06
## i9-i264    -0.10226921 -0.1374897 -0.06704874 0.0e+00
## i9-i294    -0.17641462 -0.2124500 -0.14037924 0.0e+00
##
## $`baseImage:type`
```

	diff	lwr	upr	p adj
## i294:ABSENT-i264:ABSENT	0.06120107	0.002572907	0.119829231	0.0348191
## i9:ABSENT-i264:ABSENT	-0.06580160	-0.122276041	-0.009327159	0.0116169
## i264:PRESENT-i264:ABSENT	0.01143704	-0.049979973	0.072854051	0.9949401
## i294:PRESENT-i264:ABSENT	0.09969247	0.038404341	0.160980605	0.0000532
## i9:PRESENT-i264:ABSENT	-0.13923119	-0.200068211	-0.078394174	0.0000000
## i9:ABSENT-i294:ABSENT	-0.12700267	-0.185931168	-0.068074170	0.0000000
## i264:PRESENT-i294:ABSENT	-0.04976403	-0.113444913	0.013916854	0.2250467
## i294:PRESENT-i294:ABSENT	0.03849140	-0.025065190	0.102047998	0.5141190
## i9:PRESENT-i294:ABSENT	-0.20043226	-0.263553956	-0.137310567	0.0000000
## i264:PRESENT-i9:ABSENT	0.07723864	0.015534863	0.138942415	0.0048738
## i294:PRESENT-i9:ABSENT	0.16549407	0.103918577	0.227069569	0.0000000
## i9:PRESENT-i9:ABSENT	-0.07342959	-0.134556096	-0.012303089	0.0081551
## i294:PRESENT-i264:PRESENT	0.08825543	0.022117482	0.154393385	0.0019972
## i9:PRESENT-i264:PRESENT	-0.15066823	-0.216388368	-0.084948095	0.0000000
## i9:PRESENT-i294:PRESENT	-0.23892367	-0.304523376	-0.173323955	0.0000000

We did TukeyHSD test and see that for all the three baseImages(i264, i294 and i9) the p-value is significant for each of the three. And then checking for each baseImage and type interaction we see that the p-value is significant for all except for i264:PRESENT-i264:ABSENT, i264:PRESENT-i294:ABSENT and i294:PRESENT-i294:ABSENT interactions.

Now, checking with the overall effect sizes of the model

#Report the post-hoc tests, and the overall effect sizes

library(sjstats)

Warning in checkMatrixPackageVersion(): Package version inconsistency detected.

TMB was built with Matrix version 1.2.15

Current Matrix version is 1.2.14

Please re-install 'TMB' from source using `install.packages('TMB', type = 'source')` or ask CRAN for a binary version of 'TMB' matching CRAN's 'Matrix' package

anova_stats(m_01)

	term	df	sumsq	meansq	statistic	p.value	etasq	partial.etasq
## 1	baseImage	2	24.866	12.433	67.418	0.000	0.027	0.027
## 2	type	1	0.097	0.097	0.529	0.467	0.000	0.000
## 3	size	1	1.392	1.392	7.548	0.006	0.002	0.002
## 4	Centrality	1	6.906	6.906	37.447	0.000	0.008	0.008
## 5	baseImage:type	2	2.769	1.385	7.508	0.001	0.003	0.003
## 6	Residuals	4797	884.637	0.184	NA	NA	NA	NA
	omegasq	partial.omegasq	cohens.f	power				
## 1	0.027		0.027	1.000				
## 2	0.000		0.000	0.112				
## 3	0.001		0.001	0.784				
## 4	0.007		0.008	1.000				
## 5	0.003		0.003	0.944				
## 6	NA		NA	NA				

Above we have the effect size values for each of the predictors and the interaction used to build the model. These values are useful beyond significance tests (p-values), because they estimate the magnitude of effects, independent from sample size.

1. Looking at the p-values we see that all the variables are significant except for type.
2. The etasq values give the variability accounted by each of the variables, it can be interpreted as the percentage of the percentage of variance accounted by each. We see that baseImage accounts for the highest variability in the model which is 2.7%
3. Partial etasq value strongly depends on the variability of the residuals, that is also highest for baseImage
4. Omegasq and Partial Omegasq values are also highest for baseImage.

On examining the above result we see that all of our predictors are significant except for the type because we get variance as 0 for that predictor.

```
m_01
## Call:
## aov(formula = responsetime_log ~ baseImage + type + baseImage:type +
## size + Centrality, data = data_Q2)
##
## Terms:
##          baseImage      type      size Centrality baseImage:type
## Sum of Squares    24.8658    0.0975    1.3920     6.9058         2.7691
## Deg. of Freedom      2        1        1        1        2
##          Residuals
## Sum of Squares    884.6366
## Deg. of Freedom   4797
##
## Residual standard error: 0.4294352
## Estimated effects may be unbalanced
```

Now we calculate the overall variance of the model. We use the sum of Squares value in order to do that.

```
#overall proportion of variance being predicted by this model, sum of sq
R_sq = (24.8658 + 0.0975 + 1.3920 + 6.9058 + 2.7691)/(24.8658 + 0.0975 +
1.3920 + 6.9058 + 2.7691 + 884.6366)
R_sq
## [1] 0.0391349
```

The overall proportion of variance being predicted by this model can be calculated directly from the sum-of-squares in the ANOVA model. We see that the model accounts for 4% of variance. So we can say that the model is not good because it accounts for a very low variance.

3. Repeated Measures

Now, we incorporated subject number(subnum) as a randomized factor, specifying error strata with Error(subnum/(baseImage*type)).

#For each of the factors, recoding with helmert contrasts inorder to examine interactions.

```
contrasts(data$subnum) <- contr.helmert(levels(data$subnum))
contrasts(data$baseImage) <- contr.helmert(levels(data$baseImage))
contrasts(data$type) <- contr.helmert(levels(data$type))
contrasts(data$probeCond) <- contr.helmert(levels(data$probeCond))
contrasts(data$probe) <- contr.helmert(levels(data$probe))
contrasts(data$probeFile) <- contr.helmert(levels(data$probeFile))
```

Now for each of the factors, I recoded with helmert contrasts so that we can examine interactions.

```
m_02 <- aov(responsetime_log ~ baseImage + type + baseImage*type + size +
Centrality + Error(subnum/(baseImage*type)), data = data)
summary(m_02)
```

```
##
## Error: subnum
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 119  180.7    1.519
##
## Error: subnum:baseImage
##           Df Sum Sq Mean Sq F value Pr(>F)
## baseImage  2  28.40  14.200   73.17 <2e-16 ***
## Residuals 238  46.19    0.194
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: subnum:type
##           Df Sum Sq Mean Sq F value Pr(>F)
## type       1  1.283   1.2831   6.632 0.0112 *
## Residuals 119 23.022    0.1935
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: subnum:baseImage:type
##           Df Sum Sq Mean Sq F value Pr(>F)
## baseImage:type  2  2.822   1.4112  11.33 2e-05 ***
## Residuals      238 29.652    0.1246
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Error: Within
##           Df Sum Sq Mean Sq F value Pr(>F)
## size       1    0.9    0.919   5.876 0.0154 *
```

```
## Centrality      1      7.4      7.428  47.492 6.26e-12 ***
## Residuals    4678    731.7      0.156
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To test the effect of the condition, in each case we look at the subnum:baseImage:type error strata. We get the significant p-value as its less than 0.05

Consequently, for true repetition of conditions, this error strata scheme will properly ignore repetitions within subject in the test, because we care about generalizing across the mean and we don't see an effect, we shouldn't get a huge benefit if we simply decide to measure response time multiple times per subject.

```
library(sjstats)
#anova_stats(m_02)
eta_sq(m_02)

##           term etasq                stratum
## 1      Residuals 0.172                subnum
## 2      baseImage 0.027      subnum:baseImage
## 3      Residuals 0.044      subnum:baseImage
## 4           type 0.001      subnum:type
## 5      Residuals 0.022      subnum:type
## 6 baseImage:type 0.003 subnum:baseImage:type
## 7      Residuals 0.028 subnum:baseImage:type
## 8           size 0.001                Within
## 9      Centrality 0.007                Within
```

We see from the above results the etasq value accounted for each predictor and the interaction. We see that the subnum stratum accounts for the highest variance in the model which is 17.2%. Also the subnum and baseImage interaction stratum accounts for the next highest variance which is 4.4%

4. EzANOVA

We now use an ezANOVA to build the same model as above, excluding the size and centrality variables if the model won't incorporate continuous predictors.

```
library(ez)

mod.ez <- ezANOVA(data = data,
                  dv = .(responsetime_log),
                  wid = .(subnum),
                  within = .(baseImage, type),
                  detailed = T,
                  return_aov = T)
```

```
## Warning: Collapsing data to cell means. *IF* the requested effects are a
## subset of the full design, you must use the "within_full" argument, else
## results may be inaccurate.
```

```
mod.ez
```

```
## $ANOVA
```

```
##           Effect DFn DFd           SSn           SSd           F           p
## 1 (Intercept)    1 119 3.709224e+04 24.114417 1.830430e+05 1.658542e-191
## 2 baseImage      2 238 4.270564e+00  6.000569 8.469148e+01 1.668463e-28
## 3 type           1 119 1.782107e-01  3.197451 6.632494e+00 1.123783e-02
## 4 baseImage:type  2 238 3.919910e-01  4.118370 1.132655e+01 2.000656e-05
```

```
## p<.05 ges
```

```
## 1 * 0.99899189
```

```
## 2 * 0.10240823
```

```
## 3 * 0.00473851
```

```
## 4 * 0.01036388
```

```
##
```

```
## $`Mauchly's Test for Sphericity`
```

```
##           Effect           W           p p<.05
```

```
## 2 baseImage 0.9852218 0.4154394
```

```
## 4 baseImage:type 0.9962520 0.8012787
```

```
##
```

```
## $`Sphericity Corrections`
```

```
##           Effect           GGe           p[GG] p[GG]<.05           HFe           p[HF]
```

```
## 2 baseImage 0.985437 3.966716e-28 * 1.001908 1.668463e-28
```

```
## 4 baseImage:type 0.996266 2.059945e-05 * 1.013200 2.000656e-05
```

```
## p[HF]<.05
```

```
## 2 *
```

```
## 4 *
```

```
##
```

```
## $aov
```

```
##
```

```
## Call:
```

```
## aov(formula = formula(aov_formula), data = data)
```

```
##
```

```
## Grand Mean: 7.177534
```

```
##
```

```
## Stratum 1: subnum
```

```
##
```

```
## Terms:
```

```
##           Residuals
```

```
## Sum of Squares 24.11442
```

```
## Deg. of Freedom 119
```

```
##
```

```
## Residual standard error: 0.4501579
```

```
##
```

```
## Stratum 2: subnum:baseImage
```

```
##
```

```
## Terms:
```

```
##               baseImage Residuals
## Sum of Squares  4.270564  6.000569
## Deg. of Freedom      2      238
##
## Residual standard error: 0.1587844
## Estimated effects are balanced
##
## Stratum 3: subnum:type
##
## Terms:
##               type Residuals
## Sum of Squares  0.178211  3.197451
## Deg. of Freedom      1      119
##
## Residual standard error: 0.1639187
## Estimated effects are balanced
##
## Stratum 4: subnum:baseImage:type
##
## Terms:
##               baseImage:type Residuals
## Sum of Squares      0.391991  4.118370
## Deg. of Freedom      2      238
##
## Residual standard error: 0.131545
## Estimated effects are balanced
```

Focusing on the Mauchly's test in order to check the sphericity, we see that the test is not significant because the p-value is greater than 0.05.

In the cases above, GGe and HFe are both close to 1.0, so there is no large adjustment. We can then report the p-value from the corrections lines - in this case they differ.

	GGe	HFe
baseImage	0.985437	1.001908
baseImage:type	0.996266	1.013200

We can see above the p-values for both the GGe and HFe values.

5. Mixed-effects model

Finally, we use nlme(lme) to build the same model as a mixed effects model, treating subject as a randomized factor.

Creating the first model same as done in the previous question.

```
#First Model
library(lme4)

## Loading required package: Matrix

lmer <- lmer(responsetime_log ~ baseImage + type + baseImage*type +
(1|subnum) ,data = data)
```

Creating the second model without the interaction term.

```
#Second Model Without Interaction
lmer2 <- lmer(responsetime_log ~ baseImage + type + (1|subnum) ,data = data)
```

Comparing the two models with an anova()

```
anova(lmer2, lmer)

## refitting model(s) with ML (instead of REML)

## Data: data
## Models:
## lmer2: responsetime_log ~ baseImage + type + (1 | subnum)
## lmer: responsetime_log ~ baseImage + type + baseImage * type + (1 |
## lmer:      subnum)
##      Df    AIC    BIC logLik deviance  Chisq Chi Df Pr(>Chisq)
## lmer2  6 5690.3 5729.9 -2839.2   5678.3
## lmer   8 5676.6 5729.3 -2830.3   5660.6 17.734      2  0.000141 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We would here prefer the second model which does not include the interaction term. We confirmed this based upon the AIC and BIC value because the more negative the value the better the model.

```
library(nlme)

##
## Attaching package: 'nlme'

## The following object is masked from 'package:lme4':
##
##      lmList

am <- lme(responsetime_log ~ baseImage + type + baseImage*type, random =
(~1|subnum), data = data)
am2 <- lme(responsetime_log ~ baseImage + type, random = (~1|subnum), data =
data)
anova(am, am2)
```

```
## Warning in anova.lme(am, am2): fitted objects with different fixed effects.
```

```
## REML comparisons are not meaningful.
```

##	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
##	am	1	8	5726.333	5779.077	-2855.166		
##	am2	2	6	5722.660	5762.221	-2855.330	1 vs 2	0.3277722 0.8488

When we use lme4 (lmer) to build the same model we again see that the model without the interaction is better with low AIC and BIC value as compared to the other model.