

Project_analysis_V2.R

dylanloader

Sat Apr 8 13:45:43 2017

```
# Statistics 368 project analysis.
# Set required libraries.
library(MASS)
library(Rcmdr)

## Warning: package 'Rcmdr' was built under R version 3.3.2

## Loading required package: splines

## Loading required package: RcmdrMisc

## Loading required package: car

## Warning: package 'car' was built under R version 3.3.2

## Loading required package: sandwich

## The Commander GUI is launched only in interactive sessions

library(plyr)

# randomization done before running data for ordering.
# In our case randomization within each block is fine.
# So we set a block and randomize the order within it.
set.seed(9012)
treatment_selections <- c('tg4_fin','tg4_no_fin','nth1_fin','nth1_no_fin',
                          'as_fin','as_no_fin','no_treat','just_fin')
block1_order <- sample(treatment_selections)
block2_order <- sample(treatment_selections)
block3_order <- sample(treatment_selections)
block1_order

## [1] "tg4_no_fin" "as_fin"      "no_treat"    "nth1_fin"    "as_no_fin"
## [6] "nth1_no_fin" "just_fin"    "tg4_fin"

block2_order

## [1] "nth1_no_fin" "tg4_no_fin"  "no_treat"    "as_no_fin"   "as_fin"
## [6] "just_fin"    "nth1_fin"    "tg4_fin"

block3_order

## [1] "nth1_fin"    "tg4_no_fin"  "as_fin"      "just_fin"    "no_treat"
## [6] "nth1_no_fin" "as_no_fin"   "tg4_fin"
```

#Data Entry from CSV.

#####

```
No_treat_B1 = 0.6840
No_treat_B2 = 0.8514
No_treat_B3 = 0.8805
just_fin_B1 = 0.6765
just_fin_B2 = 0.6133
just_fin_B3 = 0.6843
tg4_fin_B1 = 0.6087
tg4_fin_B2 = 0.6133
tg4_fin_B3 = 0.7173
tg4_no_fin_B1 = 0.6450
tg4_no_fin_B2 = 1.042
tg4_no_fin_B3 = 1.5018
nth1_fin_B1 = 0.6450
nth1_fin_B2 = 0.7051
nth1_fin_B3 = 0.6690
nth1_no_fin_B1 = 0.6738
nth1_no_fin_B2 = 0.7035
nth1_no_fin_B3 = 0.8517
as_fin_B1 = 0.6444
as_fin_B2 = 0.5364
as_fin_B3 = 0.6100
as_no_fin_B1 = 0.7159
as_no_fin_B2 = 0.8324
as_no_fin_B3 = 1.1386
```

Create ordered blocks

```
block_1 <- c(tg4_no_fin_B1,as_fin_B1,No_treat_B1,nth1_fin_B1,as_no_fin_B1,
             nth1_no_fin_B1,just_fin_B1,tg4_fin_B1)
block_2 <- c(nth1_no_fin_B2,tg4_no_fin_B2,No_treat_B2,as_no_fin_B2,as_fin_B2,
             just_fin_B2,nth1_fin_B2,tg4_fin_B2)
block_3 <- c(nth1_fin_B3,tg4_no_fin_B3,as_fin_B3,just_fin_B3,No_treat_B3,
             nth1_no_fin_B3,as_no_fin_B3,tg4_fin_B3)
```

Begin analysis

First we need to build the data frame

```
blocks <- c(1,1,1,1,1,1,2,2,2,2,2,2,3,3,3,3,3,3)
length(blocks)
```

```
## [1] 18
```

```
measures <- c(0.6765,0.6840,0.6444,0.7159,0.6450,0.6738,0.6087,0.6450,
              0.6133,0.8514,0.5364,0.8324,0.7051,0.7035,0.6133,1.042,
              0.6843,0.8805,0.6100,1.1386,0.6690,0.8517,0.7173,1.5018)
```

Try transformation

```
power_measures <- (measures)^(-2)
```

```

#Asin does not work for numbers over 1 produces NaNs
#asin_measures <- asin(measures)
fin_factors <- c((rep(c('On', 'Off'), each = 1, times = 12)))

fin_factorz <- as.factor(fin_factors)
# Legend for Pastes:
#paste_factors <- c(1,1,2,2,3,3,4,4,
#                   1,1,2,2,3,3,4,4,
#                   1,1,2,2,3,3,4,4)
paste_factors <- c(rep(c('CTRL', 'AS', 'NTH', 'TG'), each = 2, times = 3))

paste_factorz <- as.factor(paste_factors)

block_factors <- c(rep(c('5000', '10,000', '15,000'), each = 8, times = 1))

block_factorz <- as.factor(block_factors)

data_frame <- data.frame(measures, power_measures,
                          paste_factorz, block_factorz,
                          fin_factorz)

# Start ANOVA here
# Begin with residual analysis
A <- fin_factorz
B <- block_factorz
C <- paste_factorz

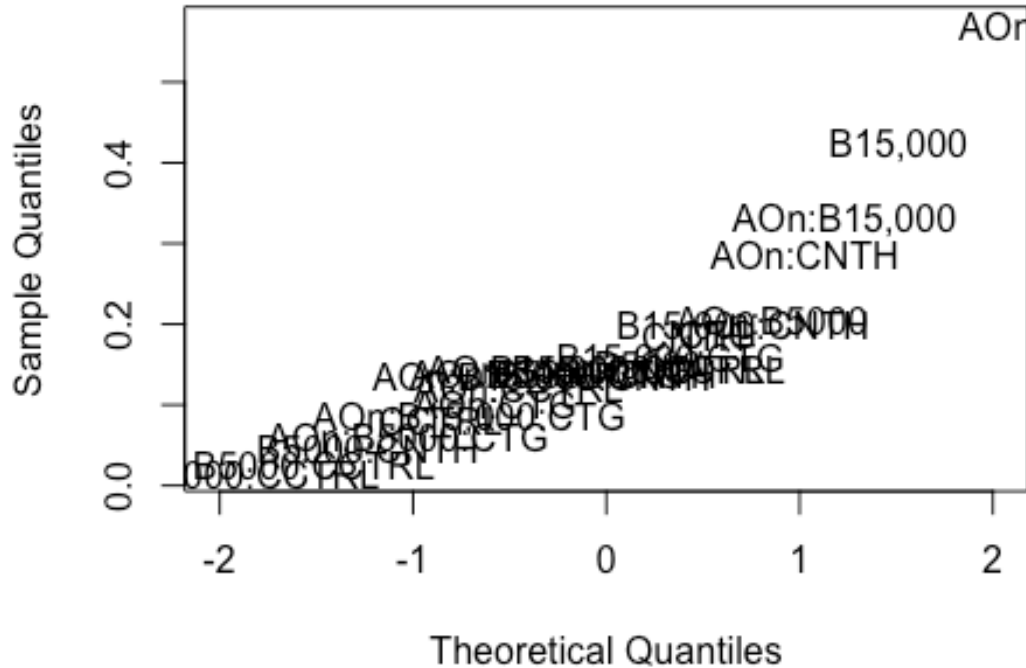
complete_model <- lm(measures ~ ((A+B+C)^3))

#complete_model

# we wish to test with a half normal plot the factors of interest.
complete_model.effects = abs(complete_model$effects[-1])
qq <- qqnorm(complete_model.effects, type="n")
text(qq$x, qq$y, labels = names(complete_model.effects))

```

Normal Q-Q Plot



```
# From the half normal plot we see FF is the most significant and
# several block factors are also significant which is to be expected
# aand block interactions are not of interest at this time. We are more
# interested in fin and pase interaction which there is some evidence of.
```

```
# set the model that with the factor combinations of interest.
full_model <- lm(measures ~ block_factorz + paste_factorz + fin_factorz +
paste_factorz*fin_factorz)
```

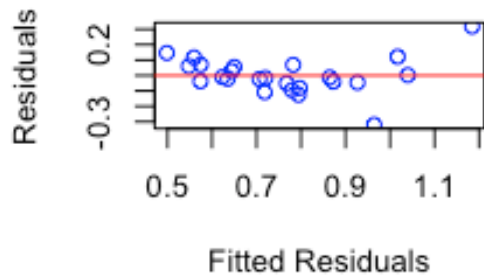
```
#####  
#Residual analysis for full model.  
# Residuals for Power model  
par(mfrow=c(2,2))  
plot(full_model$fitted,full_model$resid, col="blue",xlab= 'Fitted Residuals',  
      ylab = 'Residuals', main = 'Fitted Residuals vs. Residuals')  
abline(h=0, col="red")  
plot(c(block_factorz), full_model$resid, xlab="Blocks",col="blue",  
      ylab = 'Residuals', main = 'Block Factor Residual Plot')  
abline(h=0, col="red")  
plot(c(paste_factorz), full_model$resid, xlab="Paste",col="blue",  
      ylab = 'Residuals', main = 'Paste Factor Residual Plot')  
abline(h=0, col="red")  
plot(c(fin factorz), full_model$resid, xlab="Fin", col="blue",
```

```

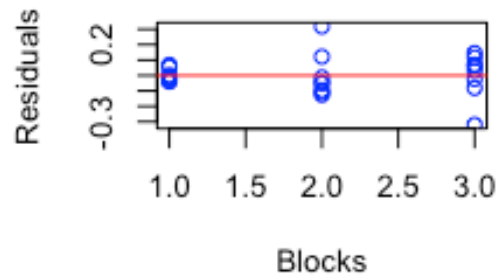
    ylab = 'Residuals', main = 'Fin Factor Residual Plot')
abline(h=0, col="red")

```

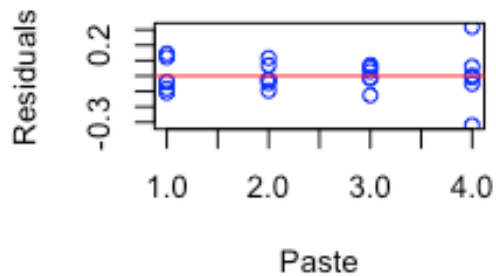
Fitted Residuals vs. Residual



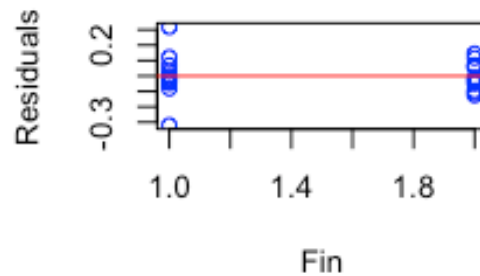
Block Factor Residual Plot



Paste Factor Residual Plot



Fin Factor Residual Plot

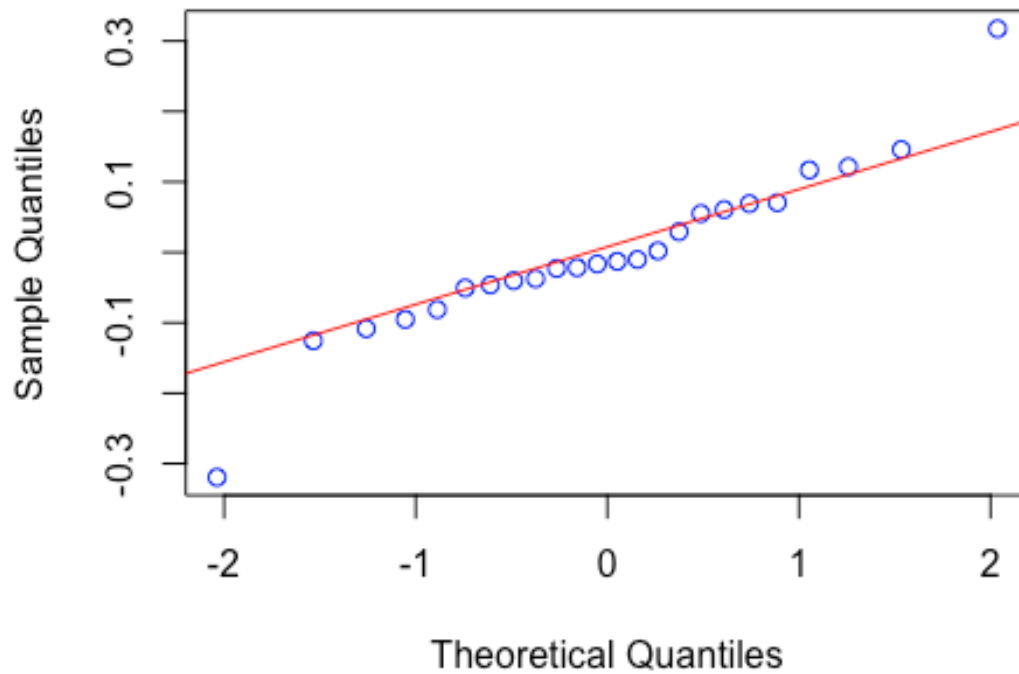


```

par(mfrow = c(1,1))
qqnorm(full_model$resid,col="blue", main = 'Normal Q-Q Plot')
qqline(full_model$resid, col=2)

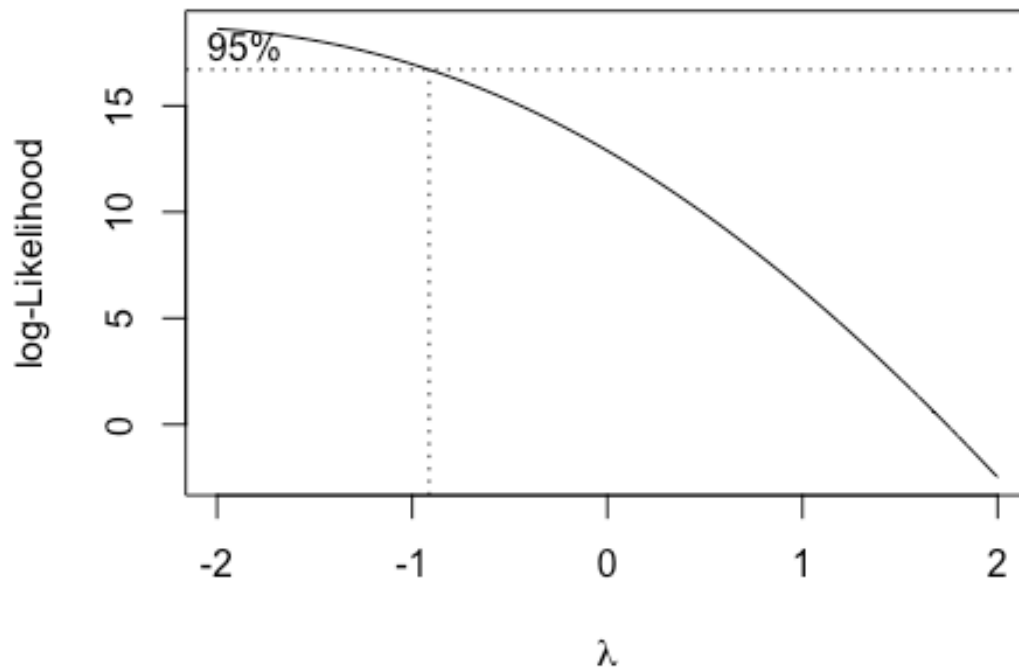
```

Normal Q-Q Plot



```
# From the fitted values vs residuals we can see an almost parabolic relation  
# this suggests that the data should under go transformation  
# Since the measures are positive we may use the MLE based Box Cox  
# transform to find an asymptotically optimal alpha for the transformation  
# of response measures
```

```
bc <- boxcox(full_model)
```



```
alpha <- bc$x[which.max(bc$y)]
alpha

## [1] -2

# Finding an alpha of -2 from the B.C transform suggests that
# the optimal model is a reciprocal square.

# we define the adjust the measures with the aforementioned transform
# and set a power model.
power_model <- lm(power_measures ~ block_factorz + paste_factorz +
  fin_factorz + paste_factorz*fin_factorz)

#####
# Interaction plots
par(mfrow=c(2,2),mar=c(4.5,4.5,4.5,4.5))
colors <- c('red','blue','green','violet')

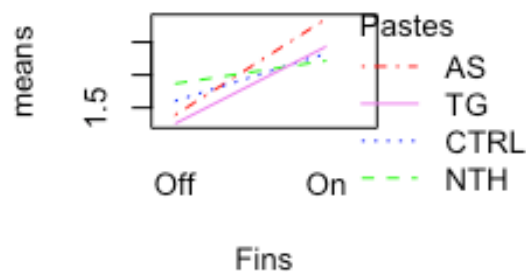
interaction.plot(factor(fin_factorz),trace.factor =
  paste_factorz,power_measures,
  xlab = 'Fins',trace.label = 'Pastes', ylab = 'means',col =
  colors, main = 'Fin-Paste Interaction Plot')
interaction.plot(factor(block_factorz),trace.factor =
```

```

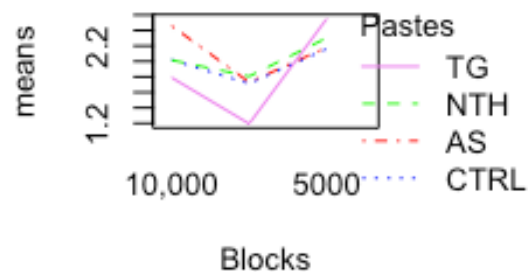
paste_factorz, power_measures,
      xlab = 'Blocks', ylab = 'means', trace.label = 'Pastes', col =
colors, main = 'Block-Paste Interaction Plot')
interaction.plot(factor(block_factorz), trace.factor =
fin_factorz, power_measures,
      xlab = 'Blocks', ylab = 'means', trace.label = 'Fins', col =
colors, main = 'Block-Fin Interaction Plot')
plot.new()

```

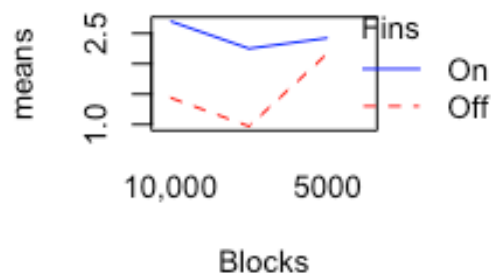
Fin-Paste Interaction Plot



Block-Paste Interaction Plot



Block-Fin Interaction Plot



```

# Residuals for Power model
par(mfrow=c(2,2))
plot(power_model$fitted, power_model$resid, col="blue", xlab= 'Fitted
Residuals',
      ylab = 'Residuals', main = 'Fitted Residuals vs. Residuals')
abline(h=0, col="red")
plot(c(block_factorz), power_model$resid, xlab="Blocks", col="blue",
      ylab = 'Residuals', main = 'Block Factor Residual Plot')
abline(h=0, col="red")
plot(c(paste_factorz), power_model$resid, xlab="Pastes", col="blue",
      ylab = 'Residuals', main = 'Paste Factor Residual Plot')
abline(h=0, col="red")
plot(c(fin_factorz), power_model$resid, xlab="Fin", col="blue",

```

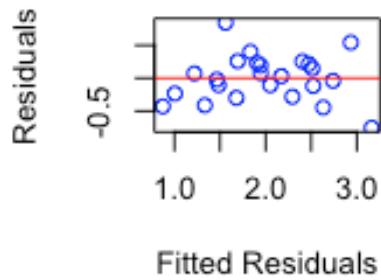


```

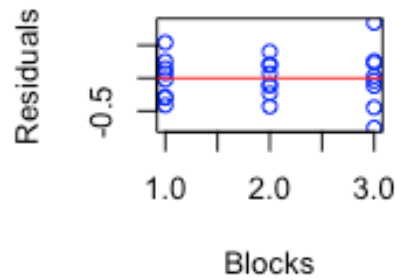
    ylab = 'Residuals', main = 'Fin Factor Residual Plot')
abline(h=0, col="red")

```

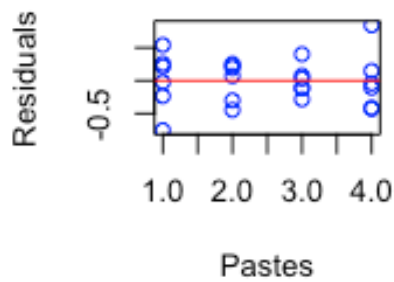
Fitted Residuals vs. Residuals



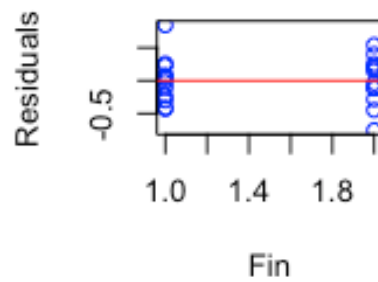
Block Factor Residual Plot



Paste Factor Residual Plot



Fin Factor Residual Plot

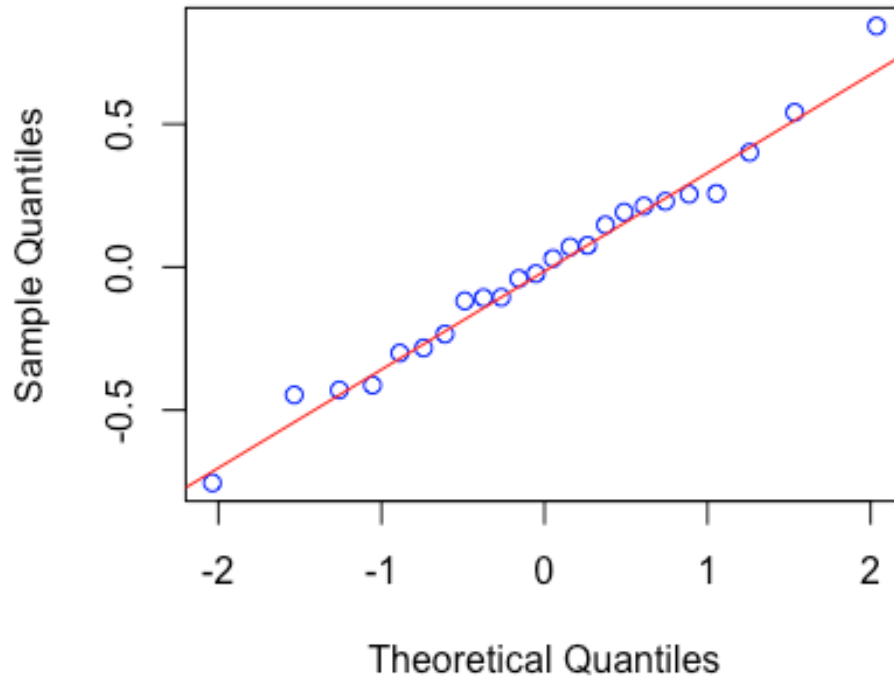


```

par(mfrow = c(1,1))
qqnorm(power_model$resid,col="blue", main = 'Normal Q-Q Plot')
qqline(power_model$resid, col=2)

```

Normal Q-Q Plot



```
bartlett.test(anova(power_model))

##
##  Bartlett test of homogeneity of variances
##
## data:  anova(power_model)
## Bartlett's K-squared = 25.81, df = 4, p-value = 3.457e-05

# Bartlett's test for homoscedasticity is not robust against deviations from
# normality so we should try
# Levenes test which is robust against violations to the normality
# assumption.

leveneTest(power_measures, group = block_factorz)

## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group  2  4.3174 0.02688 *
##      21
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Heterogeneous variances (moderate rejection P=0.02688)
leveneTest(power_measures, group = paste_factorz)
```

```

## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 3  1.8518 0.1704
##      20

# Homogeneous variances (Do not reject P=0.1704)
leveneTest(power_measures, group = fin_factorz)

## Levene's Test for Homogeneity of Variance (center = median)
##      Df F value Pr(>F)
## group 1  2.1449 0.1572
##      22

# Homogenous variances (Do not reject P=0.1572)
bartlett.test(anova(power_model))

##
## Bartlett test of homogeneity of variances
##
## data:  anova(power_model)
## Bartlett's K-squared = 25.81, df = 4, p-value = 3.457e-05

# still Bartlett's test suggests heteroscedasticity of variances
# this is most likely to do with the failure of homoscedasticity
# with the block factor
shapiro.test(power_model$residuals)

##
## Shapiro-Wilk normality test
##
## data:  power_model$residuals
## W = 0.9866, p-value = 0.981

# This returns P=0.981 which suggests normality of the residuals.
#qqnorm(power_model$residuals)

# Since the assumptions for the ANOVA model seem to be valid we can begin
analysis
anova(power_model)

## Analysis of Variance Table
##
## Response: power_measures
##
##      Df Sum Sq Mean Sq F value    Pr(>F)
## block_factorz      2  1.9675    0.9838   4.8337 0.0253416 *
## paste_factorz      3  0.2520    0.0840   0.4128 0.7464158
## fin_factorz        1  5.1880    5.1880 25.4917 0.0001777 ***
## paste_factorz:fin_factorz  3  1.0964    0.3655   1.7958 0.1941700
## Residuals        14  2.8493    0.2035
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

*# Here we can see both the paste factor and paste-fin interactions
do not seem to be significant. So we remove the highest order term
in this case Paste-fin interaction and run the ANOVA model again.*

```
power_model_adj <- lm(power_measures ~ block_factorz + paste_factorz +  
fin_factorz)  
anova(power_model_adj)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: power_measures
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)  
## block_factorz  2 1.9675   0.9838   4.2385 0.0321040 *  
## paste_factorz  3 0.2520   0.0840   0.3620 0.7812583  
## fin_factorz    1 5.1880   5.1880  22.3526 0.0001944 ***  
## Residuals     17 3.9457   0.2321
```

```
## ---
```

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

*# Here we see that the paste factor is not significant so we remove this
factor and run the adjusted anova again*

```
power_model_final <- lm(power_measures ~ block_factorz + fin_factorz)  
anova(power_model_final)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: power_measures
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)  
## block_factorz  2 1.9675   0.9838   4.6871  0.02141 *  
## fin_factorz    1 5.1880   5.1880  24.7183 7.334e-05 ***  
## Residuals     20 4.1977   0.2099
```

```
## ---
```

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

*# finally we find a model which only contains significant factors
this suggests that both blocks and fins are significant.
from the anova output we can suggest that the fin factor is the
most important in determining the thermal conduction on the processor.
we should now find where the differences occur using Tukey's HSD*

```
final_anova <- aov(power_model_final)
```

```
TukeyHSD(final_anova, 'fin_factorz')
```

```
## Tukey multiple comparisons of means
```

```
## 95% family-wise confidence level
```

```
##
```

```
## Fit: aov(formula = power_model_final)
```

```
##
```

```
## $fin_factorz
```

```
##           diff      lwr      upr    p adj
```

```
## On-Off 0.9298776 0.539735 1.32002 7.33e-05
```

```
# using a family wise error rate of 95% we find that  $\theta$  is not in the  
confidence  
# interval suggesting that the use of a fin on the Raspberry Pi's processor  
# Causes a mean decrease in temperature and thus is worth applying. For fun
```