# Data wrangling

## AVOCADO

### Block and randomization experimental design.

- Three blocks and 12 subjects each

```
## Read and format data
da <- fread('avocado_blackness.csv')
#convert wide to long extract data attribute
#and create new columns
df <-(melt(da, id.vars=c("hue_index")))
df$block<- substring(df$variable,0,1)
df$treatment <- substring(df$variable,2,2)
df$id <- substring(df$variable,3,4)
d_avo_raw <- df[,c("block","treatment","id","hue_index","value")]
```

- Figure A1 shows empirical cumulative distribution of black ratio as function of hue
- Figure A1 shows suggests that there exist both fixed effect for block and individual level.
  - Having diffrent background when taking picture of avocado or differences in lighting could have caused the fixed effect.

```
d_avo_raw %>% ggplot(aes(x = hue_index, y = value, color=treatment)) +
  geom_point(aes(color = treatment)) + facet_wrap(~block,ncol = 1) +
  xlim(20,35) +
  xlab("Hue") + ylab("Black Ratio") + theme(plot.title = element_text(hjust = 0.5))
```

### Get individual hue count data

The higher the value of hue, the stronger the filter effect. For an example,

```
## Get the length of data, hue_range
len <- dim(da)[2]

## empty matrix that will get the frequency data
avo_frequency <- as.matrix(0:49)

for (i in colnames(da)){
  if (i == "hue_index"){
   }
  else{
   ##Do something
```

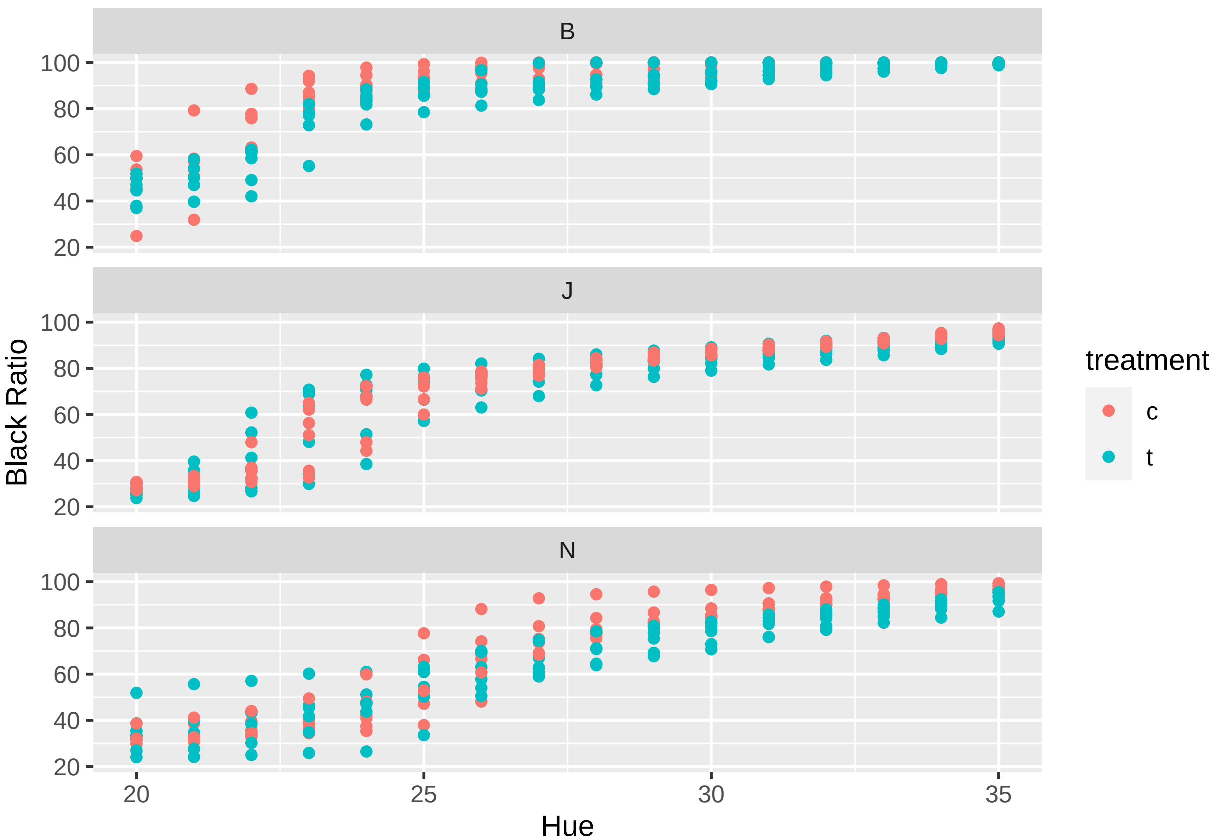


Figure 1: Cumulative distribution of black ratio

```r
    temp <- getIncre(as.matrix(da[[i]]))
    temp <- as.matrix(temp)
    ##get the increment
    avo_frequency  <- cbind(avo_frequency ,temp)
    ##start adding them
  }
}

#now add column names
d_avo <- data.frame(avo_frequency)
colnames(d_avo) <- colnames(da)
```

- `d_avo_frequency` (see below) contains frequency of pixles whose color changed when `hue` was incremented by 1

```r
#convert wide to long extract data attribute
#and create new columns
dff <-(melt(d_avo, id.vars=c("hue_index")))
dff$block<- substring(dff$variable,0,1)
dff$treatment <- substring(dff$variable,2,2)
dff$id <- substring(dff$variable,3,4)
d_avo_frequency <- dff[,c("block","treatment","id","hue_index","value")]
```

- 20 HUE indidate percent of avocado whose that became black when hue was changed from 19 to 20. (bad)
- 25 HUE indicate percent of avocado that turn black when hue was increased from 24 to 25. (still good)
- 30 HUE indicate percent of avocado that turn black when hue changed from 29 to 30.

```r
d_avo_frequency %>% ggplot(aes(x = hue_index, y = value, color=treatment)) +
  geom_point(aes(color = treatment)) + facet_wrap(~block,ncol = 1) + xlim(20,35) +
  xlab("Hue") + ylab("Count") + theme(plot.title = element_text(hjust = 0.5))
```

### Create avocado pdf

```r
#now df_avo cotains percent of pixles whose color changed when
# head(d_avo)

## empty matrix that will get the frequency data
avo_pdf <- as.matrix(0:49)

for (i in colnames(d_avo)){
  if (i == "hue_index"){
   }
  else{
    ##Do something
    temp <- sum(d_avo[[i]])
    temp <- d_avo[[i]]/temp
    ##start adding them
```

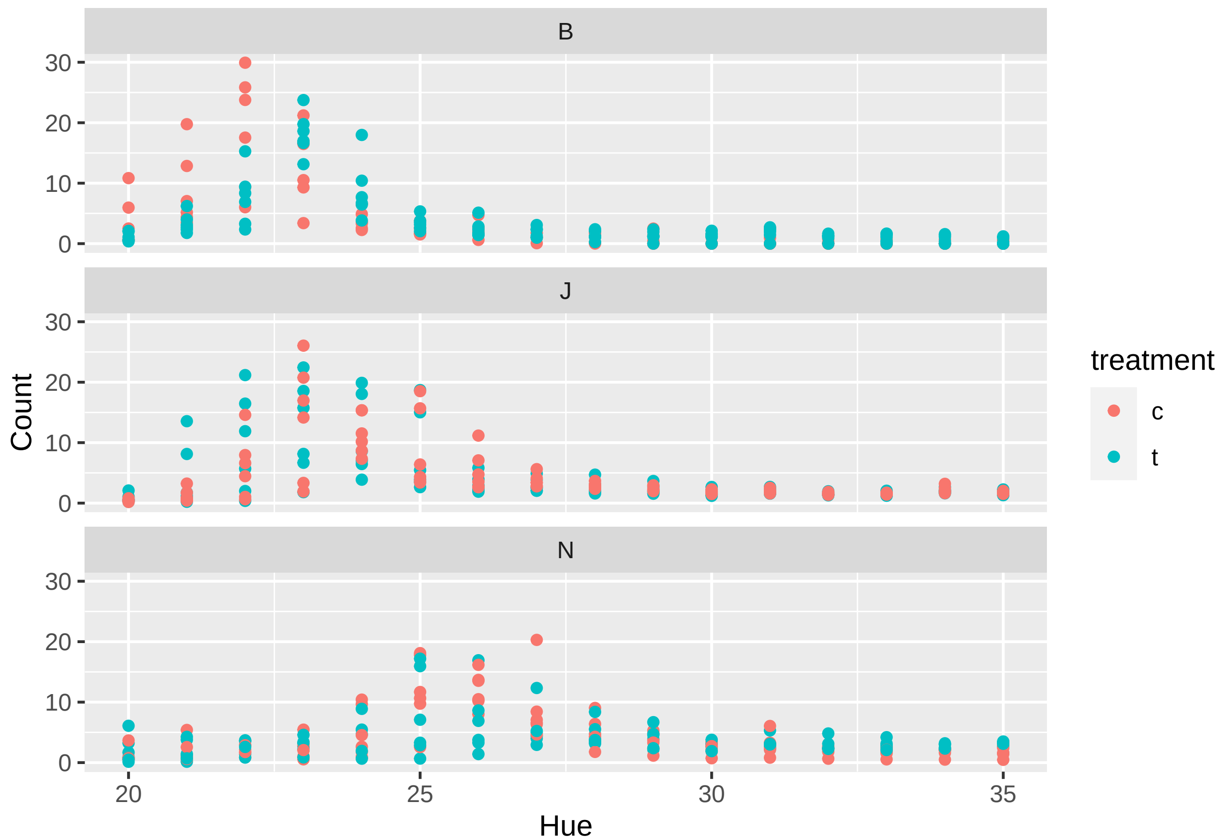


Figure 2: Hue frequency

```
  avo_pdf   <- cbind(avo_pdf  ,temp)
 }
}

#now add column names
d_avo_pdf <- data.frame(avo_pdf)
colnames(d_avo_pdf) <- colnames(da)
# head(d_avo_pdf)

#convert wide to long extract data attribute
#and create new columns
dff <-(melt(d_avo_pdf, id.vars=c("hue_index")))
dff$block<- substring(dff$variable,0,1)
dff$treatment <- substring(dff$variable,2,2)
dff$id <- substring(dff$variable,3,4)
d_avo_pdf_long <- dff[,c("block","treatment","id","hue_index","value")]
# head(d_avo_pdf_long)
```

## Create sample data based on the pdf

```
BT <- d_avo_pdf_long %>% filter(block=="B" & treatment =="t") %>%
      group_by(hue_index) %>% dplyr::summarize(Mean = mean(value, na.rm=TRUE))

BC <- d_avo_pdf_long %>% filter(block=="B" & treatment =="c")%>%
      group_by(hue_index) %>% dplyr::summarize(Mean = mean(value, na.rm=TRUE))

NT <- d_avo_pdf_long %>% filter(block=="N" & treatment =="t")%>%
      group_by(hue_index) %>% dplyr::summarize(Mean = mean(value, na.rm=TRUE))

NC <- d_avo_pdf_long %>% filter(block=="N" & treatment =="c")%>%
      group_by(hue_index) %>% dplyr::summarize(Mean = mean(value, na.rm=TRUE))

JT <- d_avo_pdf_long %>% filter(block=="J" & treatment =="t")%>%
      group_by(hue_index) %>% dplyr::summarize(Mean = mean(value, na.rm=TRUE))

JC <- d_avo_pdf_long %>% filter(block=="J" & treatment =="c")%>%
      group_by(hue_index) %>% dplyr::summarize(Mean = mean(value, na.rm=TRUE))

#treatment
s1 <- get_ind_data(BT)
t1 <- data.frame(block="B", control = "treatment", value = s1)

s2 <- get_ind_data(JT)
t2 <- data.frame(block="J", control = "treatment", value = s2)

s3 <- get_ind_data(NT)
t3 <- data.frame(block="N", control = "treatment", value = s3)

three_treats <- rbind(t1,t2,t3)

#control
```

```r
s4 <- get_ind_data(BC)
t4 <- data.frame(block="B", control = "control", value = s4)

s5 <- get_ind_data(JC)
t5 <- data.frame(block="J", control = "control", value = s5)

s6 <- get_ind_data(NC)
t6 <- data.frame(block="N", control = "control", value = s6)
three_control <- rbind(t4,t5,t6)

data <- rbind(three_treats,three_control)

p1 <- data%>% ggplot(.,aes(x=value)) +
           geom_density(aes(fill=control),adjust=1.5,alpha=0.3)  +
           facet_wrap(~block, ncol = 1) +
           xlim(20, 45) +
           theme(
                  legend.position="top",
                  panel.spacing = unit(0.1, "lines"),
                  axis.ticks.x=element_blank(),
                  plot.title = element_text(hjust = 0.5)
                ) +
        ggtitle("emprical pdf") +
        xlab("Hue") + ylab("Probability")


p2 <- data %>% ggplot(.,aes(x=value, colour = control)) + stat_ecdf() +
  facet_wrap(~block, ncol = 1) +
           xlim(20, 45) +
           theme(
                  legend.position="top",
                  panel.spacing = unit(0.1, "lines"),
                  axis.ticks.x=element_blank(),
                  plot.title = element_text(hjust = 0.5)
                ) +
        ggtitle("empirical cdf") +
        xlab("Hue") + ylab("Probability")

p1 | p2
```

**Test based on the maximum distance between empirical distributions**

```r
#control <- getIncre(df1)
#treatment <- getIncre(df2)
control <- BT$Mean
treatment <- BC$Mean

#sharp null distribution
par(mfrow=c(3,1))
invisible(capture.output(get_ks_permutation(BT$Mean,BC$Mean,5000)))
```

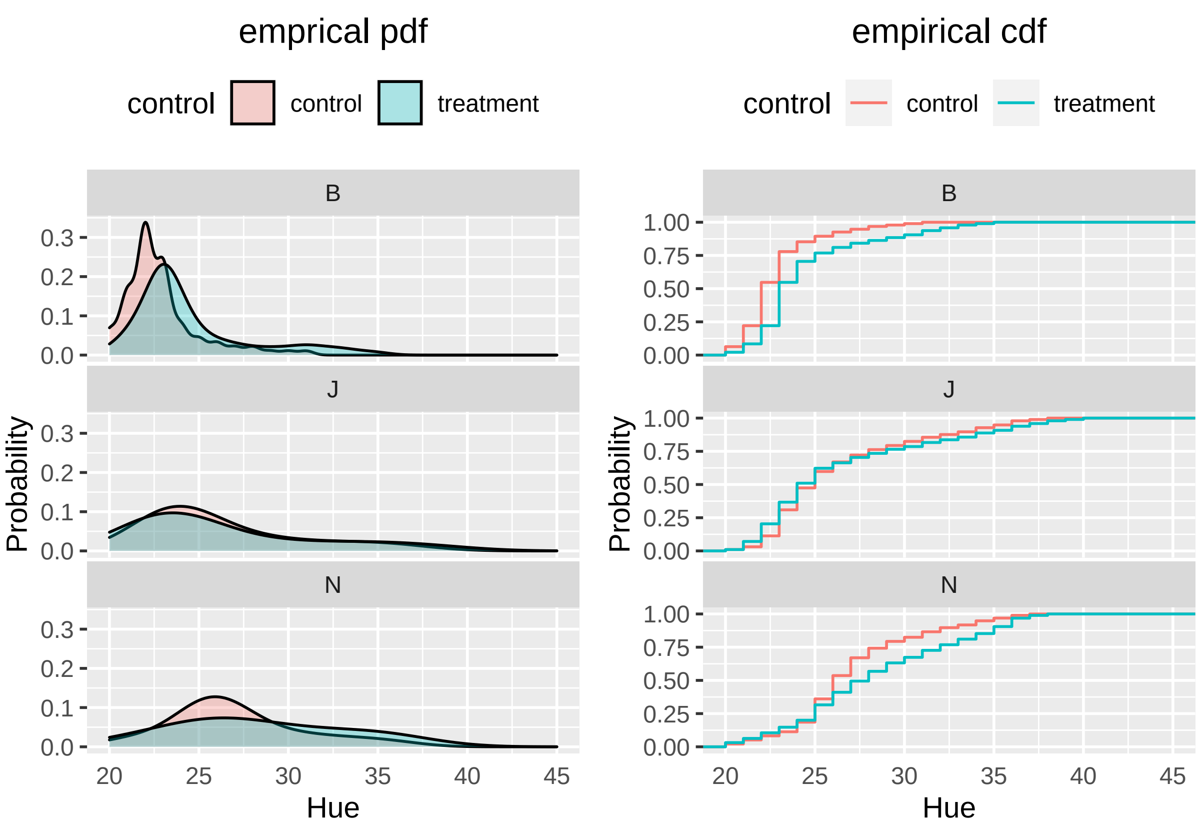


Figure 3: Empirical distribution

```
invisible(capture.output(get_ks_permutation(JT$Mean,JC$Mean,5000)))
invisible(capture.output(get_ks_permutation(NT$Mean,NC$Mean,5000)))
```

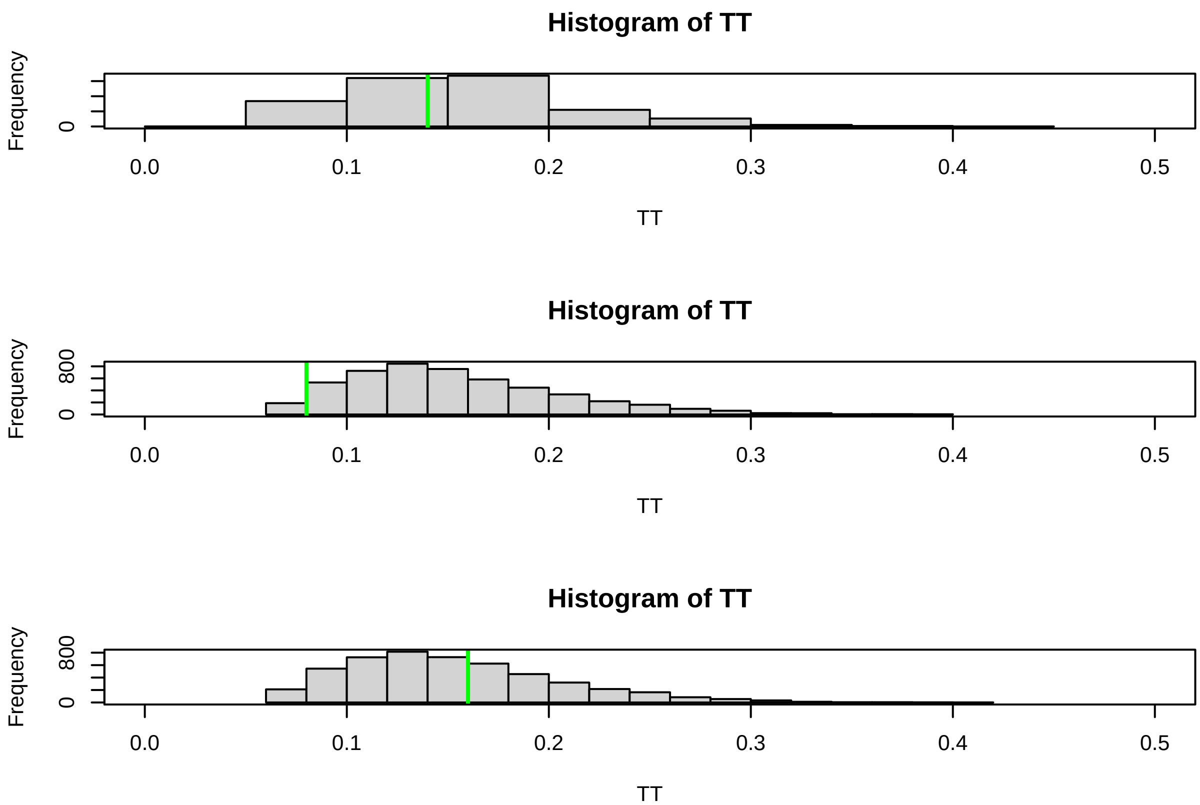


Figure 4: Result of KS permutation test

```
par(mfrow=c(1,1))
Z <- c(control,treatment)
n <- length(control )
m <- length(treatment)
N <- length(Z)
```

No significant treatment effect was observed

**hue_turn**

```
#raw data converted to long format
# head(d_avo_raw)

#arrange the data by block, treatement, id and hue_index
d <- as.data.table(d_avo_raw)
d <- d %>% filter(hue_index > 18 & hue_index < 44)
db <- d[order(rank(block), treatment,id,hue_index)]
```

```
# head(db)
# dim(db)/36
```

The following shows two samples shown during the presentation. These figure sugguest that even hue_turn does not capture the treatment effect of our interest.

```
d_temp <- as.data.table(d_avo_raw)
# d_temp
d <- d_temp %>% filter(hue_index > 18 & hue_index < 44) %>%
           filter (block == "B", id == 2 )
db <- d[order(rank(block), treatment,id,hue_index)]

d <- d_temp %>% filter(hue_index > 18 & hue_index < 44) %>%
           filter (block == "J", id == 11 )
dj <- d[order(rank(block), treatment,id,hue_index)]

# tail(db)
# db
p1 <- db %>% ggplot(.,aes(x=hue_index, y = value, colour = treatment)) + geom_line() +
  facet_wrap(~block, ncol = 1) +
           xlim(20, 45) +
           theme(
                  legend.position="top",
                  panel.spacing = unit(0.1, "lines"),
                  axis.ticks.x=element_blank(),
                  plot.title = element_text(hjust = 0.5)
                ) +
        ggtitle("Block B treatment") +
        xlab("Hue") + ylab("Probability")

p2 <- dj %>% ggplot(.,aes(x=hue_index, y = value, colour = treatment)) + geom_line() +
  facet_wrap(~block, ncol = 1) +
           xlim(20, 45) +
           theme(
                  legend.position="top",
                  panel.spacing = unit(0.1, "lines"),
                  axis.ticks.x=element_blank(),
                  plot.title = element_text(hjust = 0.5)
                ) +
        ggtitle("Block J treatment ") +
        xlab("Hue") + ylab("Probability")

p1/p2
```

### Code for mannual confirmation

```
d <- as.data.table(d_avo_raw)
d <- d %>% filter(hue_index > 18 & hue_index < 44)
db <- d[order(rank(block), treatment,id,hue_index)]
i = 23
```

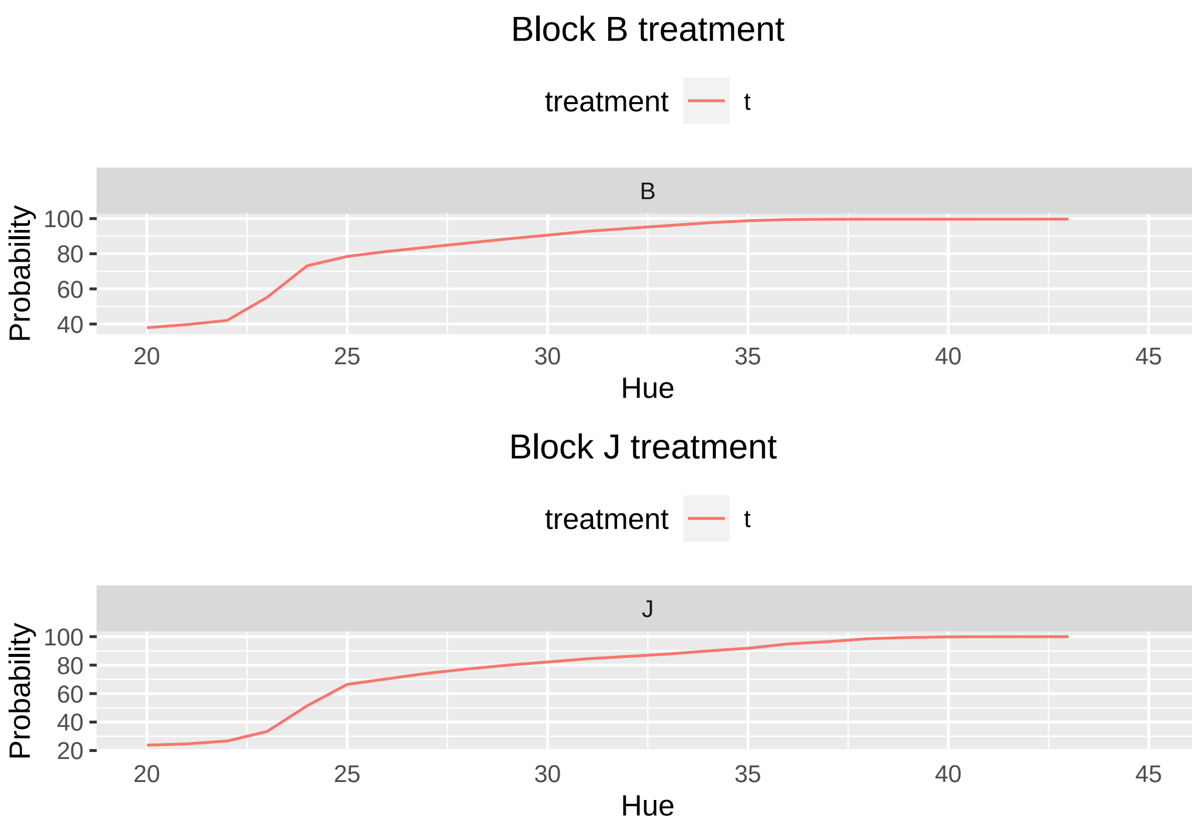


Figure 5: Comparision of good and bad avocado black cdf

```r
id <- i
start <- id + (id-1)*24
end <- id*25
d_temp <- as.data.frame(db[start:end,])
dg <- d_temp$value

d_temp[1,"block"]
```

```
## [1] "J"
```

```r
#abrupt change detection point
dg.amoc=cpt.mean(dg)
v = cpts(dg.amoc)
par(mfrow=c(1,1))
plot(dg,xaxt='n', ,
        xlab="Hue",
        ylab="Black Ratio")
abline(v=v,col="red")
```

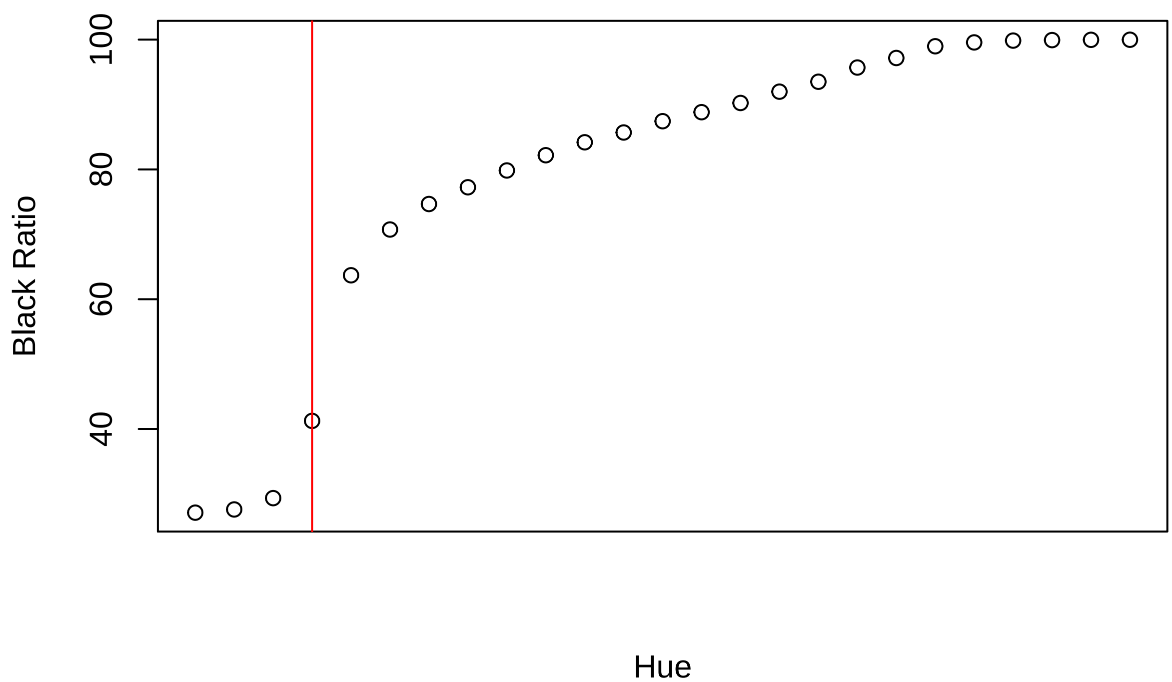


Figure 6: Example of abrupt change detection

```r
d <- as.data.table(d_avo_raw)
d <- d %>% filter(hue_index > 18 & hue_index < 44)
```

```r
db <- d[order(rank(block), treatment,id,hue_index)]

b_result <- data.frame(block = numeric(0),avocado_number= numeric(0),
                       treatment = numeric(0),
                       hue_turn = numeric(0) )



#hue index starts from 19
# par(mfrow=c(3,4))
i = 1
for (i in 1:36){
  start <- i + (i-1)*24
  end <- i*25
  d_temp <- as.data.frame(db[start:end,])
  dg <- d_temp$value
  block <- d_temp[1,"block"]
  treatment <- d_temp[1,"treatment"]
  id <- d_temp[1,"id"]
  #abrupt change detection point
  dg.amoc=cpt.mean(dg)
  v = cpts(dg.amoc)
  # print(v)
  abrupt <- v + 19
  # print(abrupt)
  b_result[i,] <- c(block,id,treatment,abrupt)
  # plot(dg)
  # abline(v=v,col="red")
}
```

Abrupt change detection in HUE

### Regression Estimator for Block Randomization

We have `3` blocks, `B`, `N`, and `J` with 12 samples in each block.

```r
# b_result

mod_b1 <- lm(hue_turn ~ as.factor(treatment) + as.factor(block),data = b_result)
mod_b2 <- lm(hue_turn ~ as.factor(treatment)*as.factor(block),data = b_result)

# mod_b2
coefficients(mod_b1)
```

```
##          (Intercept) as.factor(treatment)t     as.factor(block)J
##           22.2222222             0.8888889             1.5000000
##    as.factor(block)N
##            4.5000000
```

```r
coeftest(mod_b1, vcov = vcovHC(mod_b1, type = "HC1"))
```

```
##
## t test of coefficients:
##
##                         Estimate Std. Error t value  Pr(>|t|)
## (Intercept)             22.22222    0.29200 76.1042 < 2.2e-16 ***
## as.factor(treatment)t    0.88889    0.42853  2.0743 0.0461824 *
## as.factor(block)J        1.50000    0.41002  3.6584 0.0009047 ***
## as.factor(block)N        4.50000    0.52347  8.5966 7.994e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
coeftest(mod_b2, vcov = vcovHC(mod_b2, type = "HC1"))
```

```
##
## t test of coefficients:
##
##                                          Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                              22.16667    0.30732 72.1294 < 2.2e-16
## as.factor(treatment)t                     1.00000    0.34960  2.8604 0.0076326
## as.factor(block)J                         2.16667    0.52175  4.1527 0.0002506
## as.factor(block)N                         4.00000    0.50553  7.9126 7.862e-09
## as.factor(treatment)t:as.factor(block)J -1.33333    0.75277 -1.7712 0.0866826
## as.factor(treatment)t:as.factor(block)N  1.00000    1.02198  0.9785 0.3356562
##
## (Intercept)                              ***
## as.factor(treatment)t                    **
## as.factor(block)J                        ***
## as.factor(block)N                        ***
## as.factor(treatment)t:as.factor(block)J .
## as.factor(treatment)t:as.factor(block)N
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
stargazer(mod_b1, mod_b2, type = "text")
```

```
##
## ===================================================================================
##                                                      Dependent variable:
##                                          ------------------------------------------
##                                                           hue_turn
##                                                  (1)                     (2)
## -----------------------------------------------------------------------------------
## as.factor(treatment)t                         0.889**                   1.000
##                                               (0.429)                  (0.704)
##
## as.factor(block)J                             1.500***                 2.167***
##                                               (0.525)                  (0.704)
##
## as.factor(block)N                             4.500***                 4.000***
##                                               (0.525)                  (0.704)
##
## as.factor(treatment)t:as.factor(block)J                                 -1.333
##                                                                        (0.996)
```

```
##
## as.factor(treatment)t:as.factor(block)N                                      1.000
##                                                                             (0.996)
##
## Constant                                         22.222***                22.167***
##                                                   (0.429)                  (0.498)
##
## -------------------------------------------------------------------------------------
## Observations                                          36                       36
## R2                                                 0.716                    0.760
## Adjusted R2                                        0.689                    0.720
## Residual Std. Error                         1.286 (df = 32)          1.220 (df = 30)
## F Statistic                           26.846*** (df = 3; 32) 18.985*** (df = 5; 30)
## =====================================================================================
## Note:                                                  *p<0.1; **p<0.05; ***p<0.01
```

(see page 77 of Analysis of Categorical data) `anova()` from the `stat` package performs type I test (i.e., sequentially adding the additional terms) while `Anova()` from `car` package performs type II test (i.e, )

```
#anova(long_mod, short_mod, test = 'F')
anova(mod_b2, mod_b1, test = 'F')
```

```
## Analysis of Variance Table
##
## Model 1: hue_turn ~ as.factor(treatment) * as.factor(block)
## Model 2: hue_turn ~ as.factor(treatment) + as.factor(block)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     30 44.667
## 2     32 52.889 -2   -8.2222 2.7612 0.0793 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# b_result
b_result %>% ggplot(aes(x = hue_turn, fill=treatment)) + geom_bar() +  facet_wrap(~block,ncol = 1)
```

```
temp <- as.data.table(b_result)
b_result2 <- temp[, .(round(mean(as.integer(hue_turn))),0), by = .(block,treatment)][,1:3]
# print(b_result2)
```

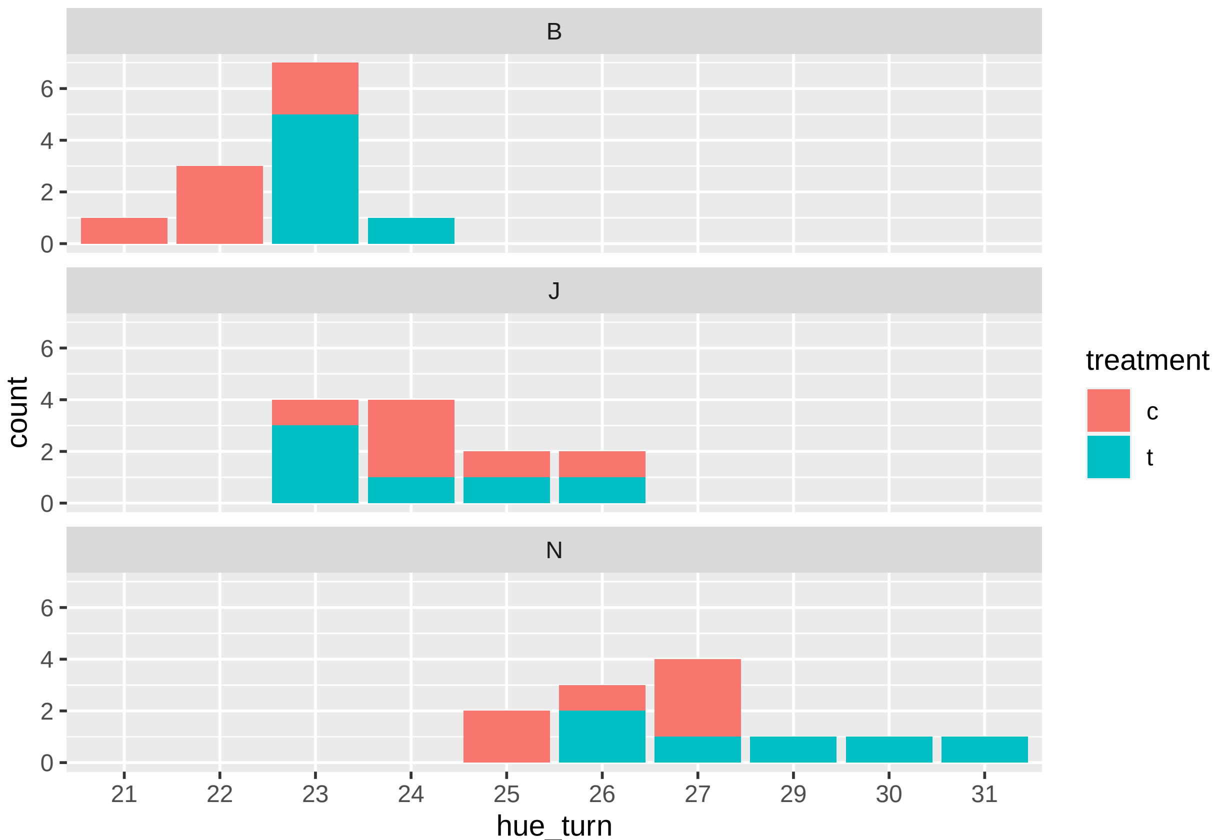


Figure 7: Comparision of hue_turn by block