

# Coots/Deschler Initial Analysis

Initial results table:

	No text	Planmaking	Consequences	Combination	Overall
Appeared	118119	3446	3655	1845	127065
Failed to Appear	80909	1606	1495	716	84726
Appearance rate	0.593	0.682	0.71	0.72	0.6

Treating each treatment independently but with the same large control group, let  $\pi_{jk}$  be the proportion of individuals in the principal stratum where  $(Y_i(0), Y_i(1)) = (j, k)$ . Then, for any set of constraints on the strata, we know that the possible range of the graduation rate,  $G$  is  $[\min(\pi_{11}), \max(1 - \pi_{00})]$ . Additionally, we can write the possible range if we make the monotonicity assumption. Using the repro, we can then write the range of possible appearance rates under each treatment type. Additionally, we can write the possible range if we make the monotonicity assumption  $Y_i(1) \geq Y_i(0)$ . For various reasons with this data, we believe the monotonicity assumption is not a good one to make. There are a multitude of theoretical mechanisms by which a text message could cause individuals to *not* appear in court, thus resulting in  $Y_i(0) > Y_i(1)$ . These could include, but are not limited to, messages scaring individuals due to the consequential nature, TODO.

Treatment	No assumption	Monotonicity assumption
Planmaking	[.275, 1]	[.593, .682]
Consequences	[.303, 1]	[.593, .710]
Combination	[.314, 1]	[.593, .721]

## Individualized Treatment Rules

The main focus of our analysis is to identify various individualized treatment rules (“ITRs”), and compare them to the experimental results using techniques from Imai and Li (2021).

- Simple ITR - by borough (what is the best we can do?)
- Simple ITR - by racial composition (what is the best we can do?)
- ITR from Imai and Strauss (2011)?
- ML ITR

## Exploratory stuff

### Subgroup analysis

The original dataset did not contain a variable for the race of an individual, but did contain the percentage of the population in each individual’s ZIP code that was either black or Hispanic. We can construct a approximation of an individual’s race by coding someone as, for example, black if more than 50% of the population in that ZIP code was black. We can do the same based on the percentage below the poverty line column. One could argue that we should actually be using a lower threshold than 50% given that communities of color and poor communities tend to be overpoliced, but without further data, we stick to the 50% threshold. The table below summarizes the appearance rates for various demographics in the data, using the heuristic just described.

Demo.	Group	Overall	Planmaking	Consequences	Combination
Gender	Male	0.594	0.683	0.709	0.711
	Female	0.645	0.678	0.728	0.782
Race	Black	0.567	0.622	0.662	0.673
	Hispanic	0.561	0.634	0.679	0.694
	Black or Hispanic	0.578	0.652	0.679	0.702
Poverty	Below Poverty Line	0.601	0.683	0.711	0.702

We can likewise break down the appearance rate by NYC borough:

Borough	Overall	Any Treatment	Planmaking	Consequences	Combination
The Bronx	0.577	0.665	0.648	0.673	0.688
Brooklyn	0.572	0.673	0.655	0.678	0.7
Manhattan	0.582	0.697	0.664	0.717	0.721
Queens	0.616	0.747	0.742	0.747	0.754
Staten Island	0.876	0.913	0.913	0.916	0.906

TODO interpret

## For First Write-up

### PAPE Matrix testing

Test PAPE matrix where we treat Staten Island with **planmaking**, the Bronx with **consequences**, and Brooklyn with **combination**. Queens and Manhattan are control groups. Each borough's treatment forms the ITR.

## First Causal Forest Attempt - Binary

```
library(grf)
# covariate set, leave one borough out
# offense-based covariates
X_offense <- c("StatenIsland", "Bronx", "Brooklyn", "Queens", "Park", "Alcohol", "Marijuana",
               "DisorderlyActive", "DisorderlyPassive", "Disorderly", "Bike", "Noise",
               "MotorVehicle", "Urination")
# individual-based covariates
X_perp <- c("Female", "Age", "PastFTA", "NbPastFTA", "NbPastSummons", "PastSummons",
            "MedianIncZip", "PctBlackZip", "PctHispanic", "PctBelowPovZip", "DOW")
# all covariates
X_full <- c(X_offense, X_perp)

# fit forests
trees <- 1000
forest_bin_full <- causal_forest(df[X_full], as.vector(df$success),
                                as.vector(df$Treatment), num.trees = trees)
forest_bin_offense <- causal_forest(df[X_offense], as.vector(df$success),
                                    as.vector(df$Treatment), num.trees = trees)
forest_bin_perp <- causal_forest(df[X_perp], as.vector(df$success),
                                 as.vector(df$Treatment), num.trees = trees)

# create dataframe of predictions
forests <- list(forest_bin_full, forest_bin_offense, forest_bin_perp)
```

```

names <- c("All", "Offense", "Perpetrator")
res_df <- data.frame(matrix(NA, nrow = nrow(df), ncol = 0))
for (n in 1:length(names)) {
  f <- forests[[n]]
  pred <- predict(f, estimate.variance = TRUE)
  taus <- pred$predictions
  vars <- pred$variance.estimates
  res_df[paste(names[n], "_Est", sep = '')] <- taus
  res_df[paste(names[n], "_Var", sep = '')] <- vars
}
# write csv of all forests
write.csv(res_df, paste("~/Desktop/HKS/G1 Fall/Gov 2003/Final Paper/text-itr/binary_", trees,
                        "trees_results.csv", sep = ''), row.names = FALSE)

```

Calculate actual ITRs from the forests

```

trees = c(100, 500, 1000)
res <- read.csv(paste("~/Desktop/HKS/G1 Fall/Gov 2003/Final Paper/text-itr/binary_", trees,
                      "trees_results.csv", sep = ''))
names <- c("All", "Offense", "Perpetrator")
itrs <- data.frame(matrix(NA, nrow = nrow(res), ncol = 0))

for (t in trees) {
  for (n in names) {
    taus <- res[paste(n, "_Est", sep = '')]
    vars <- res[paste(n, "_Var", sep = '')]
    itrs[paste(n, "_simple_", t, sep = '')] <- as.numeric(taus > 0)
    itrs[paste(n, "_harmless_", t, sep = '')] <- as.numeric(taus - (1.96 * sqrt(vars)) > 0)
  }
}

write.csv(itrs, "~/Desktop/HKS/G1 Fall/Gov 2003/Final Paper/text-itr/binary_itrs.csv", row.names = FALSE)

```

Calculate PAPE for binary ITRs

```

itrs <- read.csv("~/Desktop/HKS/G1 Fall/Gov 2003/Final Paper/text-itr/binary_itrs.csv")
papes <- c()
sds <- c()
covariates <- c()
rule <- c()
numtrees <- c()
for (r in colnames(itrs)) {
  papelist <- PAPE(df$Treatment, itrs[,r], df$success, centered = FALSE)
  covariates <- c(covariates, strsplit(r, '_')[[1]][1])
  rule <- c(rule, strsplit(r, '_')[[1]][2])
  numtrees <- c(numtrees, strsplit(r, '_')[[1]][3])
  papes <- c(papes, papelist$pape)
  sds <- c(sds, papelist$sd)
}
pape_df = data.frame(covariates, numtrees, rule, papes, sds)

```

```
write.csv(pape_df, "~/Desktop/HKS/G1 Fall/Gov 2003/Final Paper/text-itr/binaryITR_pape.csv", row.names =
```

## Multi-arm Causal Forest Attempt

```
trees <- c(100, 500, 1000)
covs <- c("All", "Offense", "Perpetrator")

# create single treatment array, 1 is planmaking, 2 is consequences, 3 is combo
t <- 1000
treat_type <- df$planmaking + 2*df$consequences + 3*df$combination
treat_df <- data.frame(matrix(NA, nrow = nrow(df), ncol = 0))
for (cov in c("Perpetrator")){
  # select covariates and fit model
  if (c == "All") {
    multi_arm <- multi_arm_causal_forest(df[X_full], as.vector(df$success),
                                          factor(treat_type), num.trees = t)
  }
  else if (c == "Offense") {
    multi_arm <- multi_arm_causal_forest(df[X_offense], as.vector(df$success),
                                          factor(treat_type), num.trees = t)
  }
  else {
    multi_arm <- multi_arm_causal_forest(df[X_perp], as.vector(df$success),
                                          factor(treat_type), num.trees = t)
  }

  # create df of predictions
  preds <- predict(multi_arm, estimate.variance = T)
  taus <- preds$predictions[,]
  vars <- preds$variance.estimates
  simple_mat <- matrix(0, nrow = nrow(vars), ncol = ncol(vars))
  harmless_mat <- matrix(0, nrow = nrow(vars), ncol = ncol(vars))
  for (r in 1:nrow(vars)) {
    for (c in 1:ncol(vars)) {
      simple_mat[r,c] <- as.numeric(taus[r,c] > 0)
      harmless_mat[r,c] <- as.numeric(taus[r,c] - 1.96 * sqrt(vars[r,c]) > 0)
    }
  }
  colnames <- c("planmaking", "consequences", "combination")
  for (c in 1:length(colnames)) {
    treat_df[paste(cov, t, colnames[c], "harmless", sep = '_')] = harmless_mat[,c]
    treat_df[paste(cov, t, colnames[c], "simple", sep = '_')] = simple_mat[,c]
  }
}
write.csv(treat_df, paste("~/Desktop/HKS/G1 Fall/Gov 2003/Final Paper/text-itr/multiarm-itr/multiarm",
                          t, "trees.csv", sep = '_'))
```