Targeting for Email Campaign

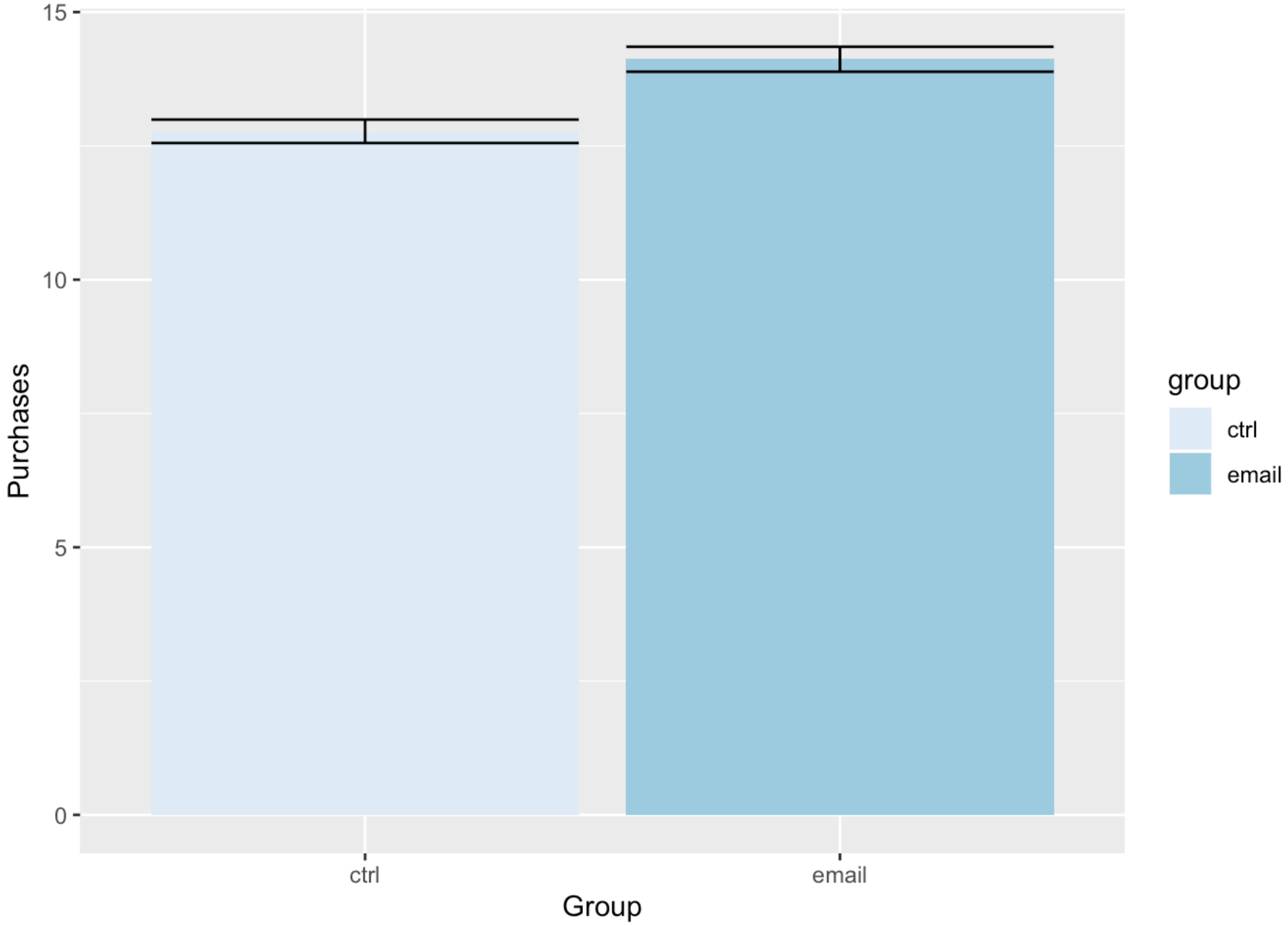
Amy Wang

Part A - Average Causal Effect

```
#Depict average effects in table
dt = data.table(d)
dagg = dt[,.(open=mean(open), click=mean(click), purch=mean(purch), seOpen=sd(open)/s
qrt(.N),
             seClick=sd(click)/sqrt(.N), sePurch=sd(purch)/sqrt(.N), .N), by=.(grou
p)]
dagg
```

```
##    group      open     click    purch      seOpen    seClick   sePurch
## 1: email 0.7957912 0.1345898 14.11913 0.002037245 0.00172474 0.2330652
## 2:  ctrl 0.0000000 0.0000000 12.77266 0.000000000 0.00000000 0.2186123
##        N
## 1: 39156
## 2: 39156
```

```
#depict averafe effects in graph
dodge = position_dodge(width=1); ##to form constant dimensions
ggplot(aes(x=group,y=purch,ymax=purch+sePurch,ymin=purch-sePurch,fill=group),data=dag
g)+
  geom_bar(position=dodge,stat="identity") +
    scale_fill_brewer(palette="Blues") +
  geom_errorbar(position=dodge)+
  labs(x="Group",y="Purchases")
```



It seems like customers receiving emails purchase more on average. Next conduct specific causal analysis to investigate this assumption.

```
summary(lm(purch~group, data=d))
```

```
##
## Call:
## lm(formula = purch ~ group, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -14.12  -14.12  -12.77  -12.77 1798.38
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.7727     0.2260  56.528  < 2e-16 ***
## groupemail    1.3465     0.3195   4.214 2.52e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.71 on 78310 degrees of freedom
## Multiple R-squared:  0.0002267,  Adjusted R-squared:  0.0002139
## F-statistic: 17.76 on 1 and 78310 DF,  p-value: 2.515e-05
```

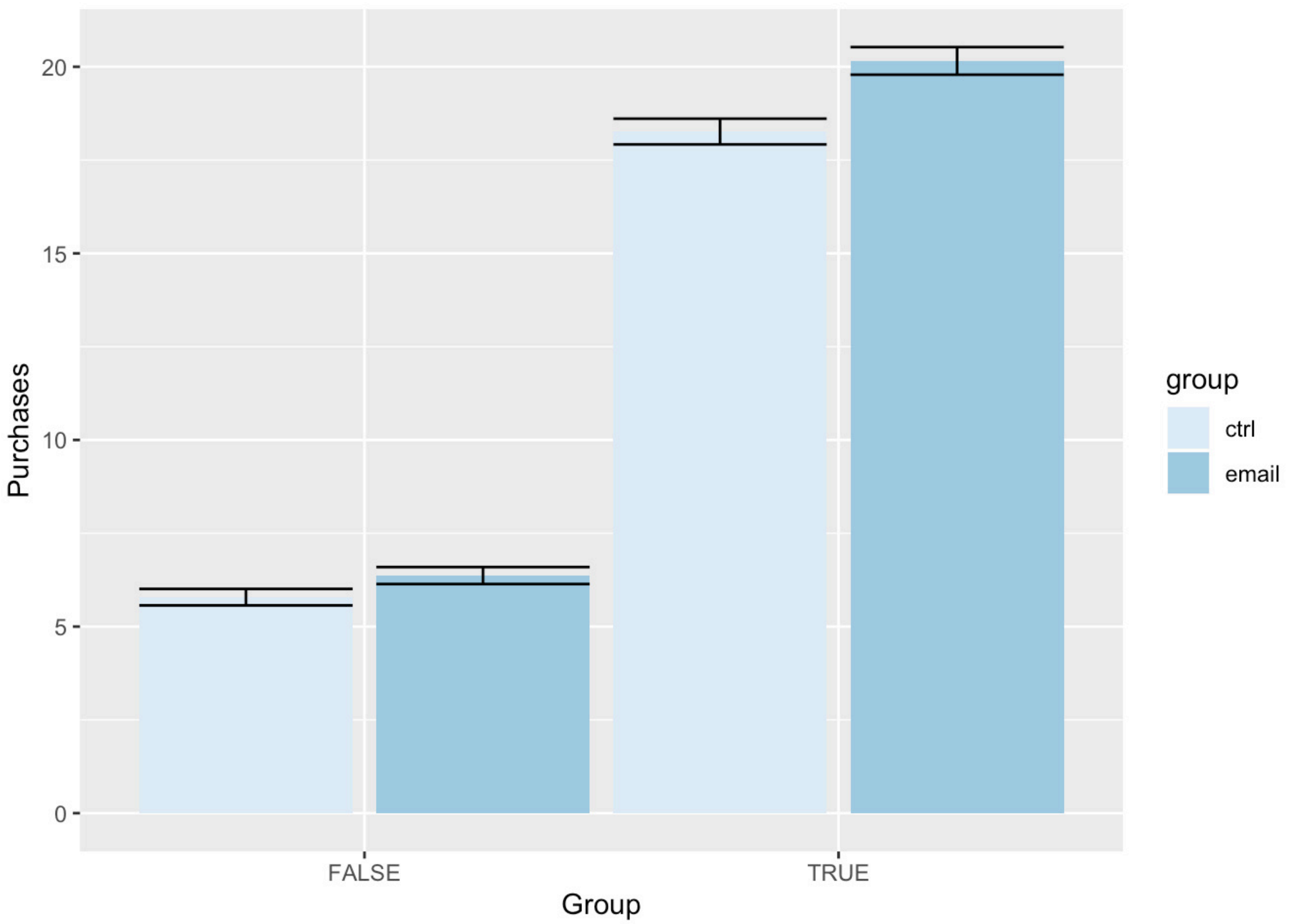
Part B - Slice & Dice

```
#Slice by recent purchase
d$recentPurch = d$last_purch < 75
dt = data.table(d)
dagg.recentPurch = dt[,.(open = mean(open), click=mean(click), purch = mean(purch),se
Open = sd(open)/sqrt(.N), seClick=sd(click)/sqrt(.N), sePurch = sd(purch)/sqrt(.N),.
N),by = .(group,recentPurch)]
dagg.recentPurch
```

```
##    group recentPurch      open     click     purch      seOpen     seClick
## 1: email       FALSE 0.6660834 0.1164188  6.367036 0.003601863 0.002449507
## 2:  ctrl        TRUE 0.0000000 0.0000000 18.264458 0.000000000 0.000000000
## 3:  ctrl       FALSE 0.0000000 0.0000000  5.788434 0.000000000 0.000000000
## 4: email        TRUE 0.8968243 0.1487438 20.157464 0.002050371 0.002398499
##      sePurch     N
## 1: 0.2271649 17145
## 2: 0.3451824 21920
## 3: 0.2210987 17236
## 4: 0.3698775 22011
```

Recent buyers buy more on average; Emails have a larger effect on recent buyers.

```
dodge = position_dodge(width=1); ##to form constant dimensions
ggplot(aes(fill=group,y=purch,x=recentPurch,ymax=purch+sePurch,ymin=purch-sePurch),data=dagg.recentPurch)+
  geom_bar(position=dodge,stat="identity") +
  geom_errorbar(position=dodge)+
scale_fill_brewer(palette="Blues")+
  labs(x="Group",y="Purchases")
```



```
summary(lm(purch~group*recentPurch, data=d))
```

```
##
## Call:
## lm(formula = purch ~ group * recentPurch, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -20.16  -18.26   -6.37   -5.79 1792.34
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  5.7884     0.3369  17.180   <2e-16 ***
## groupemail                   0.5786     0.4771   1.213   0.2252
## recentPurchTRUE             12.4760     0.4503  27.705   <2e-16 ***
## groupemail:recentPurchTRUE   1.3144     0.6370   2.063   0.0391 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.23 on 78308 degrees of freedom
## Multiple R-squared:  0.02152,    Adjusted R-squared:  0.02149
## F-statistic: 574.2 on 3 and 78308 DF,  p-value: < 2.2e-16
```

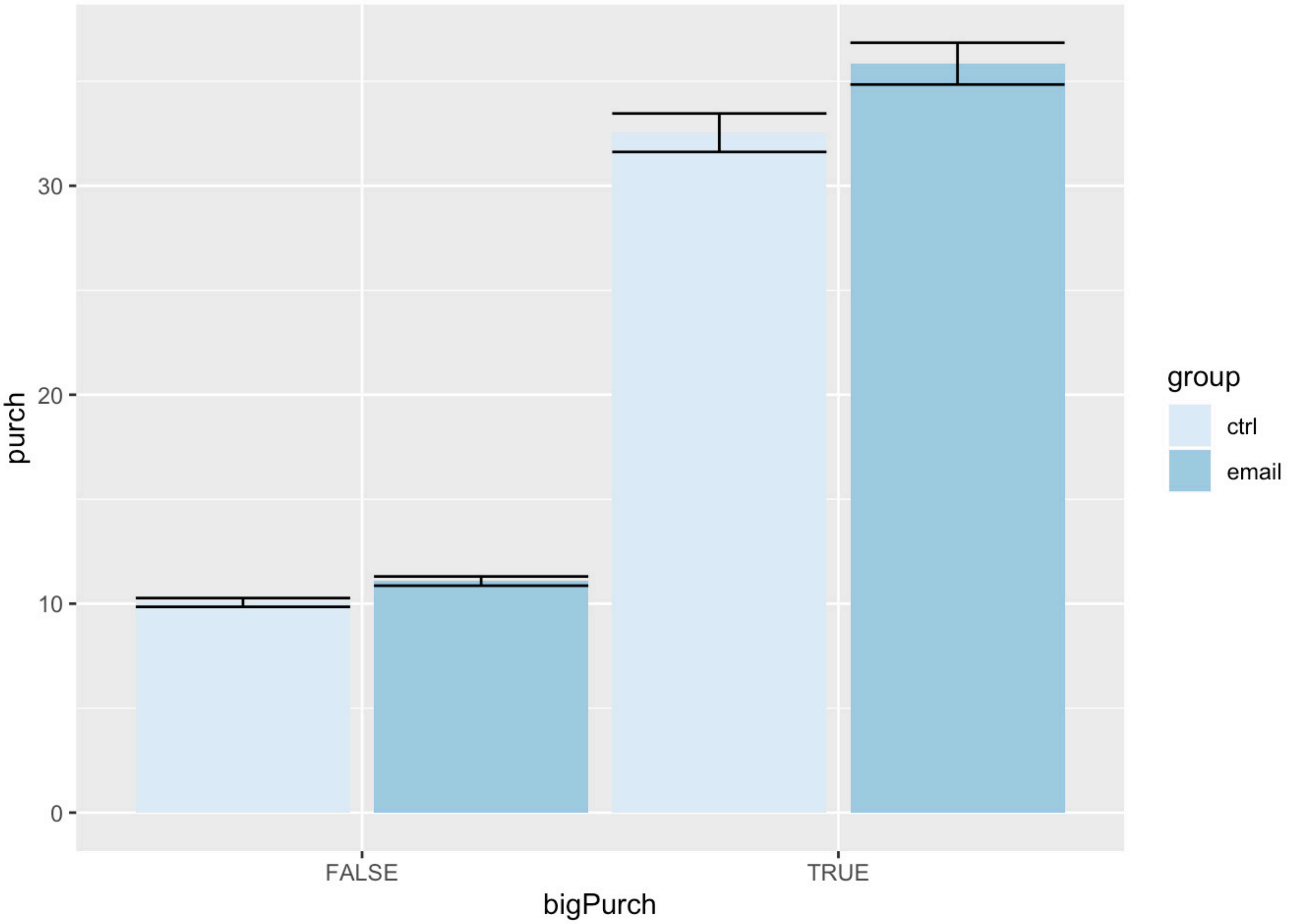
Now look at past purchase value:

```
d$bigPurch = d$past_purch > 300
dt = data.table(d)
dagg.bigPurch = dt[,.(open = mean(open), click=mean(click), purch = mean(purch),seOpe
n = sd(open)/sqrt(.N), seClick=sd(click)/sqrt(.N), sePurch = sd(purch)/sqrt(.N),.N),b
y = .(group,bigPurch)]
dagg.bigPurch
```

```
##    group bigPurch      open     click    purch       seOpen     seClick
## 1: email    FALSE 0.7675739 0.1269975 11.08268 0.0022788404 0.001796457
## 2:  ctrl    FALSE 0.0000000 0.0000000 10.05970 0.0000000000 0.000000000
## 3: email     TRUE 0.9977088 0.1889190 35.84742 0.0006900996 0.005650011
## 4:  ctrl     TRUE 0.0000000 0.0000000 32.54195 0.0000000000 0.000000000
##      sePurch     N
## 1: 0.2209772 34355
## 2: 0.2099934 34431
## 3: 1.0004333  4801
## 4: 0.9200842  4725
```

Plot.

```
dodge = position_dodge(width=1); ##to form constant dimensions
ggplot(aes(fill=group,y=purch,x=bigPurch,ymax=purch+sePurch,ymin=purch-sePurch),data=
dagg.bigPurch)+
  geom_bar(position=dodge,stat="identity") +
  geom_errorbar(position=dodge)+
scale_fill_brewer(palette="Blues")
```



```
  labs(x="Past Purchase More Than 300",y="Purchase")
```

```
## $x
## [1] "Past Purchase More Than 300"
##
## $y
## [1] "Purchase"
##
## attr(,"class")
## [1] "labels"
```

```
summary(lm(purch~group*bigPurch, data=d))
```

```
##
## Call:
## lm(formula = purch ~ group * bigPurch, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -35.85  -11.08  -10.06  -10.06 1801.42
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              10.0597     0.2373  42.387  < 2e-16 ***
## groupemail                1.0230     0.3358   3.046  0.00232 **
## bigPurchTRUE             22.4822     0.6832  32.907  < 2e-16 ***
## groupemail:bigPurchTRUE   2.2825     0.9629   2.370  0.01777 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.04 on 78308 degrees of freedom
## Multiple R-squared:  0.03014,    Adjusted R-squared:  0.0301
## F-statistic: 811.1 on 3 and 78308 DF,  p-value: < 2.2e-16
```

This one is good because all coefficients are significant. Now take a look at visits:

```
d$manyVisits = d$visits > 15
dt = data.table(d)
dagg.manyVisits = dt[,.(open = mean(open), click=mean(click), purch = mean(purch),seO
pen = sd(open)/sqrt(.N), seClick=sd(click)/sqrt(.N), sePurch = sd(purch)/sqrt(.N),.
N),by = .(group,manyVisits)]
dagg.manyVisits
```

```
##    group manyVisits      open     click    purch     seOpen     seClick
## 1: email      FALSE 0.7950427 0.1346218 13.94065 0.00204375 0.001728071
## 2:  ctrl      FALSE 0.0000000 0.0000000 12.65104 0.00000000 0.000000000
## 3: email       TRUE 1.0000000 0.1258741 62.81385 0.00000000 0.027836271
## 4:  ctrl       TRUE 0.0000000 0.0000000 49.28292 0.00000000 0.000000000
##      sePurch     N
## 1: 0.2313535 39013
## 2: 0.2185488 39026
## 3: 8.5278967   143
## 4: 4.5954611   130
```

```
summary(lm(purch~group*manyVisits, data=d))
```

```
##
## Call:
## lm(formula = purch ~ group * manyVisits, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -62.81  -13.94  -12.65  -12.65 1798.56
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 12.6510     0.2260  55.988  < 2e-16 ***
## groupemail                   1.2896     0.3196   4.035 5.46e-05 ***
## manyVisitsTRUE              36.6319     3.9216   9.341  < 2e-16 ***
## groupemail:manyVisitsTRUE  12.2413     5.4189   2.259   0.0239 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.64 on 78308 degrees of freedom
## Multiple R-squared:  0.00351,    Adjusted R-squared:  0.003472
## F-statistic: 91.95 on 3 and 78308 DF,  p-value: < 2.2e-16
```

This one looks good too. But how many customers visit more than 15 times?

```
sum(d$visits > 15)
```

```
## [1] 273
```

273 out of 78312 customers. This is way too few. Targeting at them would be meaningless from the economic perspective.

Try other slicing methods.

```
d$syrahOrNot = d$syrah > 0
summary(lm(purch~group*syrahOrNot, data=d))
```

```
##
## Call:
## lm(formula = purch ~ group * syrahOrNot, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -14.15  -14.15  -12.68  -12.68 1798.35
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                12.6842     0.2402  52.803  < 2e-16 ***
## groupemail                  1.4687     0.3400   4.320 1.56e-05 ***
## syrahOrNotTRUE              0.7678     0.7077   1.085    0.278
## groupemail:syrahOrNotTRUE  -1.0539     0.9955  -1.059    0.290
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.71 on 78308 degrees of freedom
## Multiple R-squared:  0.0002438,  Adjusted R-squared:  0.0002055
## F-statistic: 6.366 on 3 and 78308 DF,  p-value: 0.0002611
```

```
d$cabOrNot = d$cab > 0
summary(lm(purch~group*cabOrNot, data=d))
```

```
##
## Call:
## lm(formula = purch ~ group * cabOrNot, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -17.41  -12.78  -11.70  -11.70 1799.72
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              11.6981     0.2671  43.797  < 2e-16 ***
## groupemail                1.0865     0.3781   2.873  0.00406 **
## cabOrNotTRUE              3.7620     0.4998   7.527 5.23e-14 ***
## groupemail:cabOrNotTRUE   0.8618     0.7057   1.221  0.22203
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.67 on 78308 degrees of freedom
## Multiple R-squared:  0.002046,   Adjusted R-squared:  0.002008
## F-statistic: 53.53 on 3 and 78308 DF,  p-value: < 2.2e-16
```

```
d$chardOrNot = d$chard > 0
summary(lm(purch~group*chardOrNot, data=d))
```

```
##
## Call:
## lm(formula = purch ~ group * chardOrNot, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -21.81  -10.83   -9.67   -9.67 1801.67
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  9.6686     0.2679  36.094  < 2e-16 ***
## groupemail                   1.1640     0.3792   3.070  0.00214 **
## chardOrNotTRUE              10.4507     0.4915  21.262  < 2e-16 ***
## groupemail:chardOrNotTRUE    0.5229     0.6943   0.753  0.45138
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.44 on 78308 degrees of freedom
## Multiple R-squared:  0.01225,    Adjusted R-squared:  0.01221
## F-statistic: 323.6 on 3 and 78308 DF,  p-value: < 2.2e-16
```

```
d$savOrNot = d$sav_blanc > 0
summary(lm(purch~group*savOrNot, data=d))
```

```
##
## Call:
## lm(formula = purch ~ group * savOrNot, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -17.56  -12.73  -11.98  -11.98 1794.94
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               11.9785     0.2672  44.824  < 2e-16 ***
## groupemail                 0.7541     0.3781   1.995  0.04609 *
## savOrNotTRUE               2.7750     0.4995   5.555 2.78e-08 ***
## groupemail:savOrNotTRUE    2.0504     0.7060   2.904  0.00368 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.68 on 78308 degrees of freedom
## Multiple R-squared:  0.001812,   Adjusted R-squared:  0.001774
## F-statistic: 47.39 on 3 and 78308 DF,  p-value: < 2.2e-16
```

The regressions above have one common issue: insignificance, except the last one. But since the total past purchase brings a reliable regression, why don’t we use it?

Part C - Targeting & Scoring

```
#Build Uplift Model
model <- lm(purch ~ group*(last_purch < 30) + group*(past_purch > 300) +
                group*(visits > 15) + group*(chard > 0) + group*(sav_blanc>0) +
                group*(syrah>0) + group*(cab>0), data=d)

summary(model)$coef
```

```
##                                   Estimate Std. Error    t value
## (Intercept)                      4.9318194  0.3656784 13.4867670
## groupemail                       0.0993849  0.5190053  0.1914911
## last_purch < 30TRUE             10.9450169  0.4920736 22.2426406
## past_purch > 300TRUE            18.8219369  0.7666887 24.5496475
## visits > 15TRUE                 17.9632883  3.8788351  4.6311039
## chard > 0TRUE                    4.6210015  0.5347539  8.6413605
## sav_blanc > 0TRUE                1.1991496  0.4925658  2.4344966
## syrah > 0TRUE                    0.8427783  0.6909828  1.2196804
## cab > 0TRUE                      2.2700456  0.4920585  4.6133648
## groupemail:last_purch < 30TRUE   1.1814986  0.6953087  1.6992431
## groupemail:past_purch > 300TRUE  1.4870314  1.0834076  1.3725503
## groupemail:visits > 15TRUE      12.4416626  5.3603020  2.3210749
## groupemail:chard > 0TRUE        -0.2253784  0.7574679 -0.2975419
## groupemail:sav_blanc > 0TRUE     1.9715620  0.6959084  2.8330770
## groupemail:syrah > 0TRUE        -1.1309491  0.9720740 -1.1634394
## groupemail:cab > 0TRUE           0.8351022  0.6950658  1.2014721
##                                     Pr(>|t|)
## (Intercept)                      2.081851e-41
## groupemail                       8.481414e-01
## last_purch < 30TRUE             2.903249e-109
## past_purch > 300TRUE            1.388471e-132
## visits > 15TRUE                  3.643048e-06
## chard > 0TRUE                    5.657157e-18
## sav_blanc > 0TRUE                1.491474e-02
## syrah > 0TRUE                    2.225897e-01
## cab > 0TRUE                      3.968284e-06
## groupemail:last_purch < 30TRUE   8.927737e-02
## groupemail:past_purch > 300TRUE  1.698961e-01
## groupemail:visits > 15TRUE       2.028536e-02
## groupemail:chard > 0TRUE         7.660536e-01
## groupemail:sav_blanc > 0TRUE     4.611406e-03
## groupemail:syrah > 0TRUE         2.446548e-01
## groupemail:cab > 0TRUE           2.295717e-01
```

```
#try new cus to see lift
new_cust <- data.frame(chard=rep(38.12, 2), sav_blanc=rep(0, 2),
                       syrah=rep(0, 2), cab=rep(0, 2),
                       past_purch=rep(38.12,2), last_purch=rep(19,2),
                       visits=rep(3,2))
(pred <- predict(model, cbind(group=c('email', 'ctrl'), new_cust)))
```

```
##        1        2
## 21.55334 20.49784
```

```
(lift <- pred[1] - pred[2])
```

```
##        1
## 1.055505
```

```
new_cust <- data.frame(chard=rep(27.50, 2), sav_blanc=rep(0, 2),
                       syrah=rep(100.38, 2), cab=rep(0, 2),
                       past_purch=rep(127.88,2), last_purch=rep(19,2),
                       visits=rep(40,2))
(pred <- predict(model, cbind(group=c('email', 'ctrl'), new_cust)))
```

```
##        1        2
## 51.67012 39.30390
```

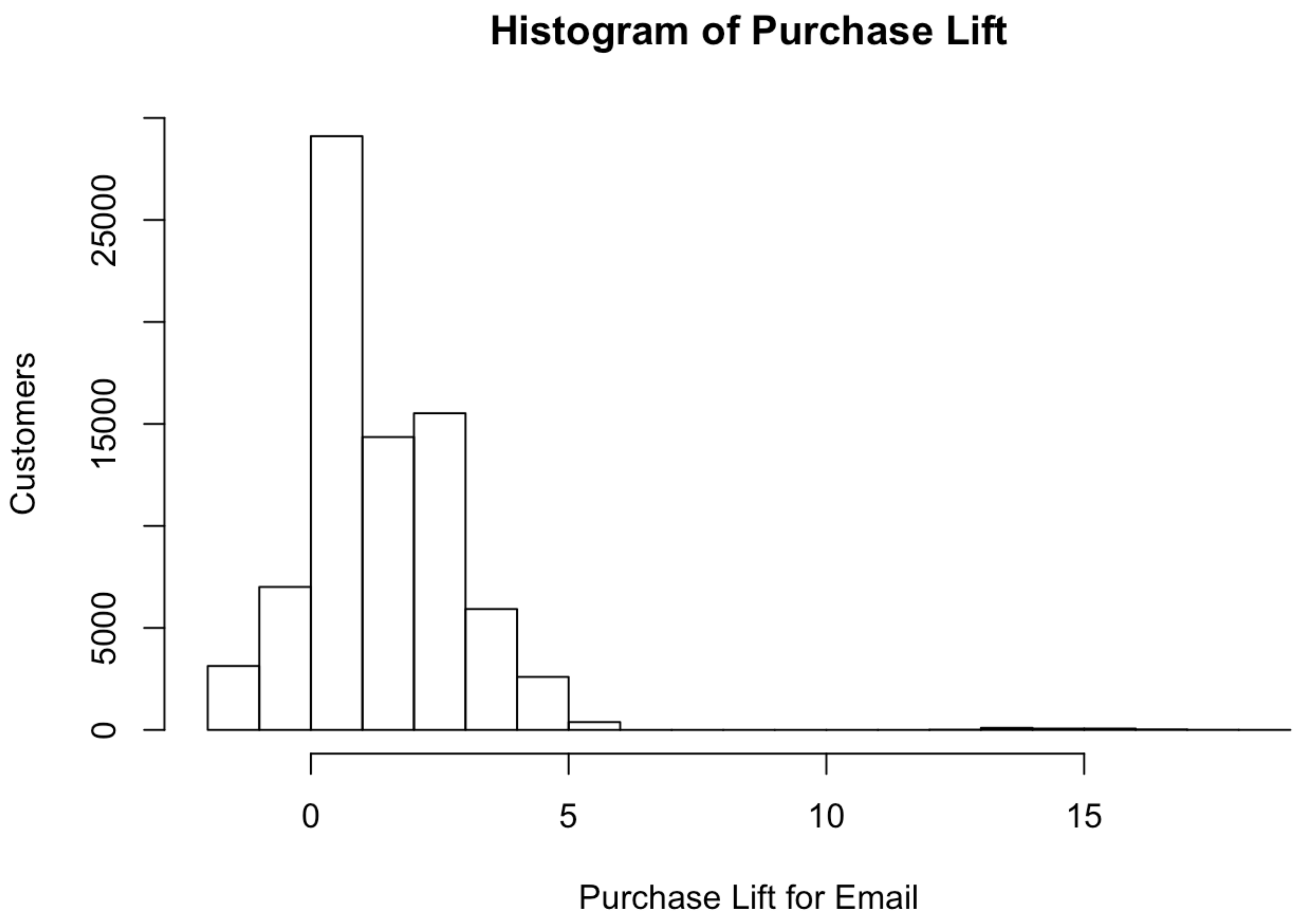
```
(lift <- pred[1] - pred[2])
```

```
##        1
## 12.36622
```

```
cust1 <- d[,7:13]
cust1$group <- 'email'
cust2 <- d[,7:13]
cust2$group <- 'ctrl'
pred <- predict(model, cust1)-predict(model, cust2)
profit <- sum(pred[(pred*0.30-0.10) >= 0])
target <- sum((pred*0.30-0.10) >= 0)
target
```

```
## [1] 49376
```

```
#estimated profit 101,131
#target 49376

hist(pred,
     main="Histogram of Purchase Lift",
     xlab="Purchase Lift for Email", ylab="Customers")
```



Build causal forest.

```
set.seed(4)
d$email <- d$group == 'email'
treat <- d$email
response <- d$purch
baseline <- d[, c("last_purch", "visits", "chard",
                  "sav_blanc", "syrah", "cab")]

tmp=proc.time()[3]
cf <- causal_forest(baseline, response, treat)
tmp = proc.time()[3]-tmp
print(cf)
```

```
## GRF forest object of type causal_forest
## Number of trees: 2000
## Number of training samples: 78312
## Variable importance:
##     1     2     3     4     5     6
## 0.243 0.048 0.284 0.254 0.048 0.124
```

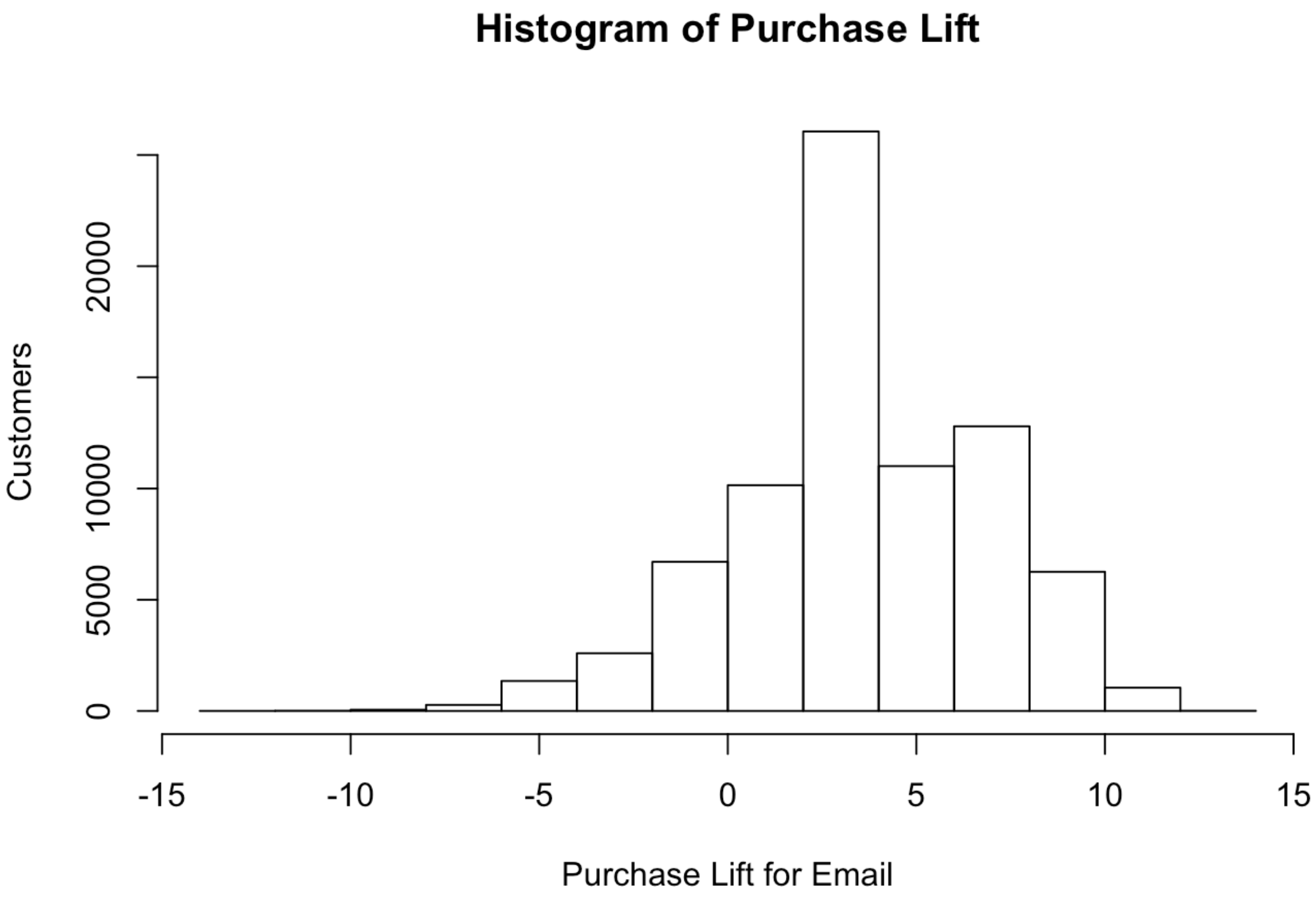
```
average_treatment_effect(cf, method="AIPW")
```

```
##  estimate   std.err
## 1.2950301 0.3097506
```

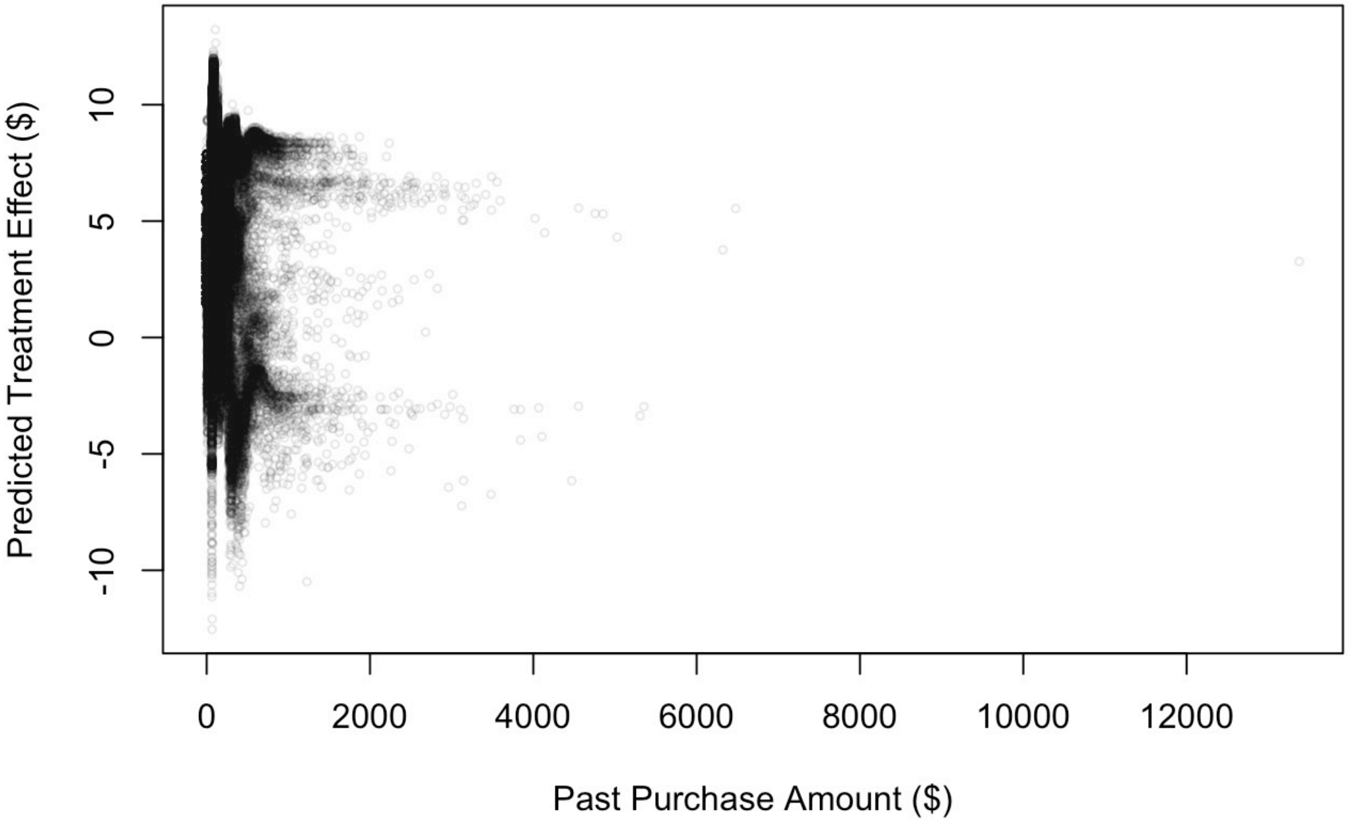
```
pred <- predict(cf, cust1[,2:7])
profit <- sum(pred[(pred*0.30-0.10) >= 0])
target <- sum((pred*0.30-0.10) >= 0)
target
```

```
## [1] 65737
```

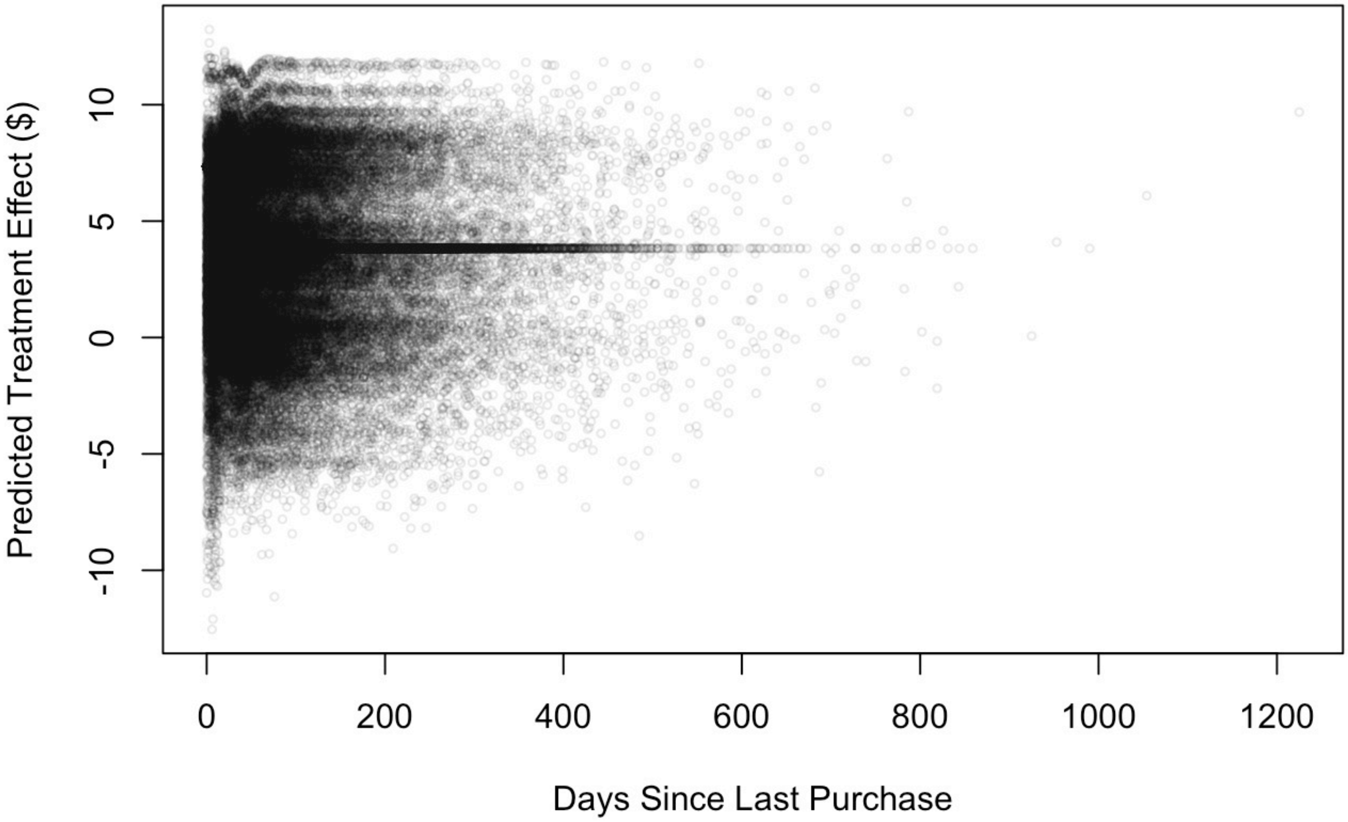
```
# estimated profit 308,614
# target 62650

# Predicted uplift for all customers in test
hist(pred$predictions,
     main="Histogram of Purchase Lift",
     xlab="Purchase Lift for Email", ylab="Customers")
```



```
# Uplift versus past purchase amount
trans_gray <- rgb(0.1, 0.1, 0.1, alpha=0.1)
plot(d$past_purch[1:nrow(d)], pred$predictions,
     cex=0.5, col=trans_gray,
     xlab="Past Purchase Amount ($)", ylab="Predicted Treatment Effect ($)")
```



```
# Uplift versus days since last purchase
trans_gray <- rgb(0.1, 0.1, 0.1, alpha=0.1)
plot(d$last_purch[1:nrow(d)], pred$predictions,
     cex=0.5, col=trans_gray,
     xlab="Days Since Last Purchase", ylab="Predicted Treatment Effect ($)")
```



```
# write a new file of predication results
pred_file <- data.frame(cbind(ID = d$user_id, score = (pred*0.30-0.10),
                              targeting_indicator = (pred*0.30-0.10) >= 0))
colnames(pred_file) <- c('ID', 'score', 'targeting_indicator')
write.csv(pred_file, file = 'pred_file.csv')
```

```
d$target = pred_file$targeting_indicator
dt = data.table(d)
dagg.target = dt[,.(percentage = .N/78312, open = mean(open), seOpen = sd(open)/sqr
t(.N), click=mean(click), seClick=sd(click)/sqrt(.N), past_purch = mean(past_purch),
sePastPurch = sd(past_purch)/sqrt(.N)),by = .(target)]
write.csv(dagg.target, 'target des.csv', sep='')
```

```
## Warning in write.csv(dagg.target, "target des.csv", sep = ""): attempt to
## set 'sep' ignored
```