Lab 1

```r
# Load the required libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.0.0 --
## v broom        1.0.3     v rsample      1.1.1
## v dials        1.1.0     v tune         1.0.1
## v infer        1.0.4     v workflows    1.1.2
## v modeldata    1.1.0     v workflowsets 1.0.0
## v parsnip      1.0.3     v yardstick    1.1.0
## v recipes      1.0.4
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```r
library(grf)
library(plotrix)
```

```
##
## Attaching package: 'plotrix'
##
## The following object is masked from 'package:scales':
##
##     rescale
```

```r
#Get Airbnb data
airbnb <- read_csv('data/airbnb-project-msba-sampled-10k.csv')
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 153995 Columns: 100
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (51): listing_url, state, city, name, summary, space, description, pict...
## dbl  (32): id, high_booking, host_id, latitude, longitude, accommodates, bat...
## lgl  (13): host_is_superhost, is_location_exact, requires_license, host_has_...
## date  (4): date, host_since, first_review, last_review
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Coding all observations after April, 2016 as 1 and 0 otherwise
df<-
  airbnb%>%
  mutate(treatment = ifelse(date>="2016-04-01",1,0))%>%
  relocate(treatment,date)
df$price = as.numeric(gsub("\\$", "", df$price))
```

```
## Warning: NAs introduced by coercion
```

```
df$cleaning_fee = as.numeric(gsub("\\$", "", df$cleaning_fee))
```

```
## Warning: NAs introduced by coercion
```

```
df$security_deposit = as.numeric(gsub("\\$", "", df$security_deposit))
```

```
## Warning: NAs introduced by coercion
```

```
df$extra_people = as.numeric(gsub("\\$", "", df$extra_people))
df$host_acceptance_rate = as.numeric(gsub("%","",df$host_acceptance_rate))
```

```
## Warning: NAs introduced by coercion
```

```
df$host_response_rate = as.numeric(gsub("%","",df$host_response_rate))
```

```
## Warning: NAs introduced by coercion
```

```
#Data frame for causal impact on price
df1<-df%>%
  select(cleaning_fee,instant_bookable,security_deposit,extra_people,city,high_booking,treatment,price)

df1 <- df1%>% drop_na()

#Data frame for causal impact on review_rating
df2<-df%>%
  select(cleaning_fee,instant_bookable,city,high_booking,treatment,price,host_response_time,review_scor

df2$host_response_time=gsub("N/A",NA,df2$host_response_time)
df2 <- df2%>% drop_na()
```

Causal Forest for Price

```
# Split the data into training and testing
set.seed(3.14159)
df1.split <- initial_split(df1)
df1.train <- training(df1.split)
df1.test <- testing(df1.split)

df1.train <-
  df1.train%>%
  relocate(price)
```

```
# Isolate the "treatment" as a matrix => Treatment is the deployment of Air Bnb's complaint line which
comp <- as.matrix(df1.train$treatment)
```

```
# Isolate the outcome as a matrix => Outcome is the nightly price
price <- as.matrix(df1.train$price)
```

```
# Use model.matrix to create a predictor matrix from the training data
X <- model.matrix(lm(price ~ -1 + cleaning_fee+factor(instant_bookable)+
                       security_deposit+extra_people+
                       factor(city)+factor(high_booking), data = df1.train))
```

```
# Estimate the causal forest
cf <- causal_forest(X, price, comp, num.trees = 5000, seed = 3.14159)
```

```
# Use model.matrix to create a predictor matrix from the testing data
X.test <- model.matrix(lm(price ~ -1 + cleaning_fee+factor(instant_bookable)+
                       security_deposit+extra_people+
                       factor(city)+factor(high_booking), data = df1.test))
```

```
# Calculate the effects in the test data
df1.test <-
  predict(cf, X.test) %>%
  select(predictions) %>%
  bind_cols(df1.test)
```

```
#Question 3: What is the causal impact of Airbnb's new complaint line on the prices?
imp<-df1.test%>%
  select(predictions,treatment)%>%
  filter(treatment==1)

head(imp,10)
```

```
##    predictions treatment
## 1    -9.379481         1
## 2   -21.666455         1
## 3     3.197681         1
## 4   -59.989891         1
## 5   -16.875216         1
## 6   -17.392098         1
## 7   -25.995192         1
## 8     1.684150         1
## 9   -18.243531         1
## 10    2.005823         1
```

```
#Average Treatment Effect
df1.test%>%
  select(predictions,treatment)%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATE = TE[2]-TE[1])
```

```
## # A tibble: 1 x 1
##    CATE
##   <dbl>
## 1 0.563
```

```
#Standard error of the mean
std.error(predict(cf, X.test))
```

```
## predictions
##   0.1638863
```

```
#Average Treatment effect on treated
df1.test%>%
  select(predictions,treatment)%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATET = TE[2])
```

```
## # A tibble: 1 x 1
##   CATET
##   <dbl>
## 1 -23.1
```

```
df1.test%>%
  select(predictions,treatment)%>%
  filter(treatment==1)%>%
  summarize(std_error= std.error(predictions))
```

```
##   std_error
## 1 0.1651704
```

```
#CATE and CATET for New York City, NY
#CATE
df1.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="new-york-city")%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATE = TE[2]-TE[1])
```

```
## # A tibble: 1 x 1
##    CATE
##   <dbl>
## 1  3.64
```

```
#Standard Error
df1.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="new-york-city")%>%
  summarize(std_error= std.error(predictions))
```

```
##   std_error
## 1 0.3600469
```

```
#CATET
df1.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="new-york-city")%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATET = TE[2])
```

```
## # A tibble: 1 x 1
##   CATET
##   <dbl>
## 1 -21.0
```

```r
#Standard Error
df1.test%>%
  select(predictions,treatment,city)%>%
  filter(treatment==1)%>%
  filter(city=="new-york-city")%>%
  summarize(std_error= std.error(predictions))
```

```
##   std_error
## 1  0.365714
```

```r
#CATE and CATET for Austin, TX
#CATE
df1.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="austin")%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATE = TE[2]-TE[1])
```

```
## # A tibble: 1 x 1
##    CATE
##   <dbl>
## 1  15.0
```

```r
#Standard Error
df1.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="austin")%>%
  summarize(std_error= std.error(predictions))
```

```
##   std_error
## 1 0.6946432
```

```r
#CATET
df1.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="austin")%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATET = TE[2])
```

```
## # A tibble: 1 x 1
##   CATET
##   <dbl>
## 1 -28.7
```

```r
#Standard Error
df1.test%>%
  select(predictions,treatment,city)%>%
  filter(treatment==1)%>%
  filter(city=="austin")%>%
  summarize(std_error= std.error(predictions))
```

```
##   std_error
## 1 0.6960752
```

Causal Forest for Review_Score_rating

```
# Split the data into training and testing
set.seed(3.14159)
df2.split <- initial_split(df2)
df2.train <- training(df2.split)
df2.test <- testing(df2.split)

df2.train <-
  df2.train%>%
  relocate(review_scores_rating)
```

```
# Isolate the "treatment" as a matrix => Treatment is the deployment of Air Bnb's complaint line which
comp <- as.matrix(df2.train$treatment)
```

```
# Isolate the outcome as a matrix => Outcome is the rating of Air Bnb.
rating <-as.matrix(df2.train$review_scores_rating)
```

```
# Use model.matrix to create a predictor matrix from the training data
X2 <- model.matrix(lm(review_scores_rating ~ -1+
                      cleaning_fee+factor(instant_bookable)+factor(city)+factor(high_booking)+treatment
                      data= df2.train))
```

```
# Estimate the causal forest
rcf <- causal_forest(X2, rating, comp, num.trees = 5000, seed = 3.14159)
```

```
# Use model.matrix to create a predictor matrix from the testing data
X2.test<-model.matrix(lm(review_scores_rating ~ -1+
                      cleaning_fee+factor(instant_bookable)+factor(city)+factor(high_booking)+treatment
                      data= df2.test))
```

```
# Calculate the effects in the test data
df2.test <-
  predict(rcf, X2.test) %>%
  select(predictions) %>%
  bind_cols(df2.test)
```

```
#Question 2: What is the causal impact of Airbnb's new complaint line on the prices?
imp2<-df2.test%>%
  select(predictions,treatment)%>%
  filter(treatment==1)

head(imp2,10)
```

```
##    predictions treatment
## 1   2.77429210         1
## 2   0.49460852         1
## 3   6.83919856         1
## 4   0.86905324         1
## 5   1.24301911         1
## 6  -0.08427279         1
## 7   4.04300682         1
## 8  -3.45974889         1
## 9  -1.60411594         1
## 10  3.10904340         1
```

```r
#Average Treatment Effect
df2.test%>%
  select(predictions,treatment)%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATE = TE[2]-TE[1])
```

```
## # A tibble: 1 x 1
##    CATE
##   <dbl>
## 1 0.409
```

```r
#Standard error of the mean
std.error(predict(rcf, X2.test))
```

```
## predictions
##  0.02303521
```

```r
#Average Treatment effect on treated
df2.test%>%
  select(predictions,treatment)%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATET = TE[2])
```

```
## # A tibble: 1 x 1
##   CATET
##   <dbl>
## 1  1.01
```

```r
#Standard Error
df2.test%>%
  select(predictions,treatment)%>%
  filter(treatment==1)%>%
  summarize(std_error=std.error(predictions))
```

```
##    std_error
## 1 0.02335306
```

```r
#CATE and CATET for New York City, NY
#CATE
df2.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="new-york-city")%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATE = TE[2]-TE[1])
```

```
## # A tibble: 1 x 1
##    CATE
##   <dbl>
## 1 0.967
```

```r
#Standard Error
df2.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="new-york-city")%>%
  summarize(std_error=std.error(predictions))
```

```
##    std_error
## 1 0.05472127
```

```
#CATET
df2.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="new-york-city")%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATET = TE[2])
```

```
## # A tibble: 1 x 1
##   CATET
##   <dbl>
## 1  1.44
```

```
#Standard Error
df2.test%>%
  select(predictions,treatment,city)%>%
  filter(treatment==1)%>%
  filter(city=="new-york-city")%>%
  summarize(std_error=std.error(predictions))
```

```
##    std_error
## 1 0.05723709
```

```
#CATE and CATET for Austin, TX
#CATE
df2.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="austin")%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATE = TE[2]-TE[1])
```

```
## # A tibble: 1 x 1
##    CATE
##   <dbl>
## 1 0.911
```

```
#Standard Error
df2.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="austin")%>%
  summarize(std_error=std.error(predictions))
```

```
##    std_error
## 1 0.08890026
```

```
#CATET
df2.test%>%
  select(predictions,treatment,city)%>%
  filter(city=="austin")%>%
  group_by(treatment)%>%
  summarize(TE = mean(predictions))%>%
  summarize(CATET = TE[2])
```

```
## # A tibble: 1 x 1
##   CATET
##   <dbl>
## 1  1.17
```

```
#Standard Error
df2.test%>%
  select(predictions,treatment,city)%>%
  filter(treatment==1)%>%
  filter(city=="austin")%>%
  summarize(std_error=std.error(predictions))
```

```
##    std_error
## 1 0.08999777
```