R Notebook

Code

Hide

```
rm(list=ls())
library(ggplot2)
library(tidyr)
```

```

Attaching package: ‘tidyr’

The following objects are masked from ‘package:Matrix’:

    expand, pack, unpack
```

Hide

```
library(predictrace)
library(rethnicity)
```

```
== WARNING! ================================================================
==
The method provided by this package has its limitations and anyone must use them
cautiously and responsibly. You should also agree that you only intend to use the method
for academic research purpose and not for commercial use. You would also agree NOT to
discriminate anyone based on race or color or any characteristic, with the information
provided by this package. Please refer to the documentation for details:
https://fangzhou-xie.github.io/rethnicity/index.html
============================================================================
==
```

Hide

```
library(psych)
```

```

Attaching package: ‘psych’

The following objects are masked from ‘package:ggplot2’:

    %+%, alpha
```

Hide

```
library(dplyr)
```

```

Attaching package: ‘dplyr’

The following objects are masked from ‘package:stats’:

    filter, lag

The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

Hide

```
library(grf)
library(rpart)
library(glmnet)
```

```
Loaded glmnet 4.1-4
```

Hide

```
library(splines)
library(MASS)
```

```

Attaching package: ‘MASS’

The following object is masked from ‘package:dplyr’:

    select
```

Hide

```
library(lmtest)
```

```
Loading required package: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric
```

Hide

```
library(sandwich)
library(ggplot2)
library(tidyverse)
```

```
Registered S3 methods overwritten by 'dbplyr':
  method         from
  print.tbl_lazy
  print.tbl_sql
── Attaching packages ─────────────────────────────────────────────── tidyverse 1.3.1
──
✔ tibble  3.1.7     ✔ stringr 1.4.0
✔ readr   2.1.2     ✔ forcats 0.5.1
✔ purrr   0.3.4
── Conflicts ────────────────────────────────────────────────── tidyverse_conflicts()
──
✖ psych::%+%()    masks ggplot2::%+%()
✖ psych::alpha()  masks ggplot2::alpha()
✖ tidyr::expand() masks Matrix::expand()
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
✖ tidyr::pack()   masks Matrix::pack()
✖ MASS::select()  masks dplyr::select()
✖ tidyr::unpack() masks Matrix::unpack()
```

Hide

```
library(tidymodels)
```

```
── Attaching packages ─────────────────────────────────────────────── tidymodels 0.2.0
──
✔ broom        0.8.0     ✔ rsample      0.1.1
✔ dials        0.1.1     ✔ tune         0.2.0
✔ infer        1.0.0     ✔ workflows    0.2.6
✔ modeldata    0.1.1     ✔ workflowsets 0.2.1
✔ parsnip      0.2.1     ✔ yardstick    0.0.9
✔ recipes      0.2.0
── Conflicts ────────────────────────────────────────────────── tidymodels_conflicts()
──
✖ psych::%+%()      masks ggplot2::%+%()
✖ scales::alpha()   masks psych::alpha(), ggplot2::alpha()
✖ scales::discard() masks purrr::discard()
✖ tidyr::expand()   masks Matrix::expand()
✖ dplyr::filter()   masks stats::filter()
✖ recipes::fixed()  masks stringr::fixed()
✖ dplyr::lag()      masks stats::lag()
✖ tidyr::pack()     masks Matrix::pack()
✖ dials::prune()    masks rpart::prune()
✖ MASS::select()    masks dplyr::select()
✖ yardstick::spec() masks readr::spec()
✖ recipes::step()   masks stats::step()
✖ tidyr::unpack()   masks Matrix::unpack()
✖ recipes::update() masks Matrix::update(), stats::update()
• Learn how to get started at https://www.tidymodels.org/start/
```

Hide

```
library(fastDummies)
library(xtable)
```

Hide

```
# Auxiliary function to computes adjusted p-values
# following the Romano-Wolf method.
# For a reference, see http://ftp.iza.org/dp12845.pdf page 8
#  t.orig: vector of t-statistics from original model
#  t.boot: matrix of t-statistics from bootstrapped models
romano_wolf_correction <- function(t.orig, t.boot) {
  abs.t.orig <- abs(t.orig)
  abs.t.boot <- abs(t.boot)
  abs.t.sorted <- sort(abs.t.orig, decreasing = TRUE)

  max.order <- order(abs.t.orig, decreasing = TRUE)
  rev.order <- order(max.order)

  M <- nrow(t.boot)
  S <- ncol(t.boot)

  p.adj <- rep(0, S)
  p.adj[1] <- mean(apply(abs.t.boot, 1, max) > abs.t.sorted[1])
  for (s in seq(2, S)) {
    cur.index <- max.order[s:S]
    p.init <- mean(apply(abs.t.boot[, cur.index, drop=FALSE], 1, max) > abs.t.sorted[s])
    p.adj[s] <- max(p.init, p.adj[s-1])
  }
  p.adj[rev.order]
}

# Computes adjusted p-values for linear regression (lm) models.
#    model: object of lm class (i.e., a linear reg model)
#    indices: vector of integers for the coefficients that will be tested
#    cov.type: type of standard error (to be passed to sandwich::vcovHC)
#    num.boot: number of null bootstrap samples. Increase to stabilize across runs.
# Note: results are probabilitistic and may change slightly at every run.
#
# Adapted from the p_adjust from from the hdm package, written by Philipp Bach.
# https://github.com/PhilippBach/hdm_prev/blob/master/R/p_adjust.R
summary_rw_lm <- function(model, indices=NULL, cov.type="HC2", num.boot=10000) {

  if (is.null(indices)) {
    indices <- 1:nrow(coef(summary(model)))
  }
  # Grab the original t values.
  summary <- coef(summary(model))[indices,,drop=FALSE]
  t.orig <- summary[, "t value"]

  # Null resampling.
  # This is a trick to speed up bootstrapping linear models.
  # Here, we don't really need to re-fit linear regressions, which would be a bit slow.
  # We know that betahat ~ N(beta, Sigma), and we have an estimate Sigmahat.
  # So we can approximate "null t-values" by
  #  - Draw beta.boot ~ N(0, Sigma-hat) --- note the 0 here, this is what makes it a *nu
ll* t-value.
  #  - Compute t.boot = beta.boot / sqrt(diag(Sigma.hat))
```

```
  Sigma.hat <- vcovHC(model, type=cov.type)[indices, indices]
  se.orig <- sqrt(diag(Sigma.hat))
  num.coef <- length(se.orig)
  beta.boot <- mvrnorm(n=num.boot, mu=rep(0, num.coef), Sigma=Sigma.hat)
  t.boot <- sweep(beta.boot, 2, se.orig, "/")
  p.adj <- romano_wolf_correction(t.orig, t.boot)

  result <- cbind(summary[,c(1,2,4),drop=F], p.adj)
  colnames(result) <- c('Estimate', 'Std. Error', 'Orig. p-value', 'Adj. p-value')
  result
}
```

Hide

```
setwd(getwd())

ICLR_2018 = read.csv("../ICLR_2018.csv")
ICLR_2019 = read.csv("../ICLR_2019.csv")
ICLR_2017_authors = read.csv("../ICLR_2017_author.csv")
ICLR_2017_conversations = read.csv("../iclr2017_conversations.csv")

ICLR_2018 = ICLR_2018[c(names(ICLR_2018)[1:14])]
ICLR_2019 = ICLR_2019[c(names(ICLR_2019)[1:14])]
```

Hide

```
authors = rbind(ICLR_2017_authors[c('year','authors','forum')],ICLR_2018[c('year','autho
rs','forum')],ICLR_2019[c('year','authors','forum')])
authors = unique(authors)
authors = as.data.frame(authors)

authors = extract(authors,authors,c("First_Name","Last_Name"), "([^ ]+) (.*)")

authors = na.omit(authors)
```

Hide

```
authors['race'] = predict_ethnicity(firstnames = authors$First_Name, lastnames = authors
$Last_Name, method = "fullname")[7]
```

Hide

```
yvalue = c(length(unique(ICLR_2017_authors$forum)),length(unique(ICLR_2018$forum)), leng
th(unique(ICLR_2019$forum)))
xvalue = as(c(2017,2018,2019), 'integer')
number_of_papers = data.frame(xvalue,yvalue)
yvalue
```

```
[1]  490  911 1419
```

Hide

```
ggplot(number_of_papers, aes(x=xvalue, y=yvalue, color = xvalue)) +
  geom_line() +
  geom_point(size = yvalue/150, alpha= 0.5) +
  ggtitle("Number of Submissions to ICLR 2017 - 2019") +
  labs(x = "Conference Year", y = "Number of Submission") +
  scale_x_discrete(limit = c(2017, 2018, 2019)) +
  geom_text(
    label=yvalue,
    nudge_x=0.2, nudge_y=0.1,
    check_overlap=T) +
  theme(legend.position="none")
```

```
Warning: Continuous limits supplied to discrete scale.
Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

Hide

```
#ggsave("submission_n.png")
```

Hide

```
df = authors %>% group_by(year,race) %>%
  dplyr::summarise(n = n()) %>%
  mutate(freq = round(n  / sum(n ), 3))
```

```
`summarise()` has grouped output by 'year'. You can override using the `.groups`
argument.
```

Hide

```
df
```

year	race	n	freq
<int>	<chr>	<int>	<dbl>
2017	asian	918	0.509
2017	black	179	0.099
2017	hispanic	201	0.111
2017	white	507	0.281
2018	asian	1940	0.556
2018	black	313	0.090
2018	hispanic	301	0.086
2018	white	933	0.268
2019	asian	3428	0.607
2019	black	382	0.068

1-10 of 12 rows Previous **1** 2 Next

Hide

```
ggplot(df, aes(x = year, y = freq)) +
  geom_line(aes(color = race, linetype = race)) +
  scale_color_manual(values = c("darkred", "steelblue",'green','purple')) +
  ggtitle("Percent of Authors by Racial Identity from ICLR 2017 - 2019")+
  scale_x_discrete(limit = c(2017, 2018, 2019)) +
  geom_point()
```

```
Warning: Continuous limits supplied to discrete scale.
Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

Hide

```
#ggsave("freq_authors.png")
ggplot(df, aes(x = year, y = n)) +
  geom_line(aes(color = race, linetype = race)) +
  scale_color_manual(values = c("darkred", "steelblue",'green','purple')) +
  ggtitle("Number of Authors by Racial Identity from ICLR 2017 - 2019") +
  scale_x_discrete(limit = c(2017, 2018, 2019)) +
  geom_point()
```

```
Warning: Continuous limits supplied to discrete scale.
Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

Hide

```
#ggsave("N_authors.png")
```

Hide

```
ICLR_2017_reviews = ICLR_2017_conversations[c("year","forum", "rating"   ,"confidence"
 ,  "review" )]
ICLR_2017_reviews = ICLR_2017_reviews %>%
  na_if("") %>%
  na.omit

ICLR_2018_reviews = ICLR_2018[c("year" , "forum"  ,  "rating"  ,  "confidence", "review"
)]
ICLR_2019_reviews = ICLR_2019[c("year" , "forum"  ,  "rating"  ,  "confidence", "review"
)]

ICLR_reviews = rbind(ICLR_2017_reviews,ICLR_2018_reviews,ICLR_2019_reviews)

ICLR_reviews['rating_number'] = as(sub(":.*","",ICLR_reviews$rating),'numeric')

ICLR_reviews['confidence_number'] = as(sub(":.*","",ICLR_reviews$confidence),'numeric')
ICLR_reviews['review_length'] = nchar(ICLR_reviews$review)

ICLR_reviews = ICLR_reviews %>% mutate(review_length_quantile = ntile(review_length, 4))
ICLR_reviews = unique(ICLR_reviews)
```

Hide

```
xtable(ICLR_reviews %>% group_by(year) %>%
  summarise(rating_average = mean(`rating_number`, na.rm=T), expert_average = mean(`conf
idence_number`, na.rm=T), review_length_average = mean(`review_length`, na.rm=T)))
```

```
% latex table generated in R 4.2.0 by xtable 1.8-4 package
% Sun Jun  5 01:34:50 2022
\begin{table}[ht]
\centering
\begin{tabular}{rrrrr}
  \hline
 & year & rating\_average & expert\_average & review\_length\_average \\
  \hline
1 & 2017 & 5.69 & 3.83 & 1873.30 \\
  2 & 2018 & 5.43 & 3.80 & 2315.58 \\
  3 & 2019 & 5.44 & 3.75 & 2522.51 \\
   \hline
\end{tabular}
\end{table}
```

Hide

```
ICLR_summary_stats = ICLR_reviews %>% group_by(year) %>%
  summarise(rating_average = mean(`rating_number`, na.rm=T), expert_average = mean(`conf
idence_number`, na.rm=T), review_length_average = mean(`review_length`, na.rm=T), rating
_sd = sd(`rating_number`, na.rm = T))

ggplot(ICLR_summary_stats, aes(x=year, y=rating_average)) + geom_point() +
  geom_errorbar(aes(ymin=rating_average-rating_sd, ymax=rating_average+rating_sd), width
=.2,
                position=position_dodge(0.05)) +
  scale_x_discrete(limit = c(2017, 2018, 2019)) +
  geom_vline(xintercept = 2017.5, linetype="dotted",
                color = "red", size=1.5) +
  geom_line() +
  ggtitle("Average Review Score from ICLR 2017 - 2019")
```

```
Warning: Continuous limits supplied to discrete scale.
Did you mean `limits = factor(...)` or `scale_*_continuous()`?
```

Hide

```
#ggsave("ratings.png")
```

Hide

```
rm(ICLR_analysis_ARI)
```

```
Warning in rm(ICLR_analysis_ARI) : object 'ICLR_analysis_ARI' not found
```

Hide

```
ICLR_analysis_ARI = ICLR_reviews %>% filter(year %in% c(2017,2018))

ICLR_analysis_ARI = full_join(x=ICLR_analysis_ARI,y=authors,by="forum")

#audit data
#ICLR_reviews %>% filter(forum == 'B1-q5Pqxl')
#ICLR_analysis_ARI %>% filter(forum == 'B1-q5Pqxl')
#authors %>% filter(forum == 'B1-q5Pqxl')


ICLR_analysis_ARI = ICLR_analysis_ARI %>% mutate(W = if_else(year.x==2017,0,1))

ICLR_analysis_ARI = na.omit(ICLR_analysis_ARI)
ICLR_analysis_ARI = unique(ICLR_analysis_ARI)

ICLR_analysis_ARI%>% group_by(W) %>% summarise(n())
```

W	n()
<dbl>	<int>
0	5534
1	10532

2 rows

Hide

```
NA
```

Hide

```
ICLR_analysis_ARI %>% filter(review_length >= 10000) %>% summarise(n())
```

n()
<int>
38

1 row

Hide

```
treated_review_length = ICLR_analysis_ARI %>% filter(W == 1) %>% dplyr::select(review_le
ngth,review_length_quantile)

treated_review_length$treatment = 'Treated'

control_review_length = ICLR_analysis_ARI %>% filter(W == 0) %>% dplyr::select(review_le
ngth,review_length_quantile)

control_review_length$treatment = 'Control'

ICLR_review_length = rbind(control_review_length, treated_review_length)

ggplot(ICLR_review_length, aes(x=review_length,fill= treatment)) + geom_histogram(alpha
 = .2)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Hide

```
ggplot(ICLR_review_length, aes(x=review_length_quantile,fill= treatment)) + geom_histogr
am(alpha = .2)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Hide

```
#names(ICLR_analysis)
#c("rating","confidence","review","rating number","confidence number","review length","F
irst_Name","Last_Name","race")

#remove race when doing model comparison with g
#covariates = c("confidence_number","review_length_quantile")
covariates = c("confidence_number","review_length_quantile","race")
XX <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))), data=ICLR_ana
lysis_ARI)

set.seed(1)

forest <- causal_forest(
             X=XX,
             W=ICLR_analysis_ARI[,"W"],
             Y=ICLR_analysis_ARI[,"rating_number"]
             )


forest.ate <- average_treatment_effect(forest)


forest.ate
```

```
   estimate     std.err
-0.24751592  0.02620021
```

Hide

```
hist(forest$W.hat, main="Estimated propensity scores \n(causal forest with interaction d
ata)", xlim=c(-.1, 1.1))
```

Hide

```
print(forest)
```

```
GRF forest object of type causal_forest
Number of trees: 2000
Number of training samples: 16066
Variable importance:
    1     2     3     4     5     6
0.000 0.372 0.292 0.079 0.191 0.066
```

Hide

```
# Available in randomized settings and observational settings with unconfoundedness+over
lap

# A list of vectors indicating the left-out subset
covariates = c("confidence_number","review_length_quantile")
n <- nrow(ICLR_analysis_ARI)
n.folds <- 5
indices <- split(seq(n), sort(seq(n) %% n.folds))
treatment <- "W"
# Preparing data
W <- ICLR_analysis_ARI[,"W"]
Y <- ICLR_analysis_ARI[,"rating_number"]

# Matrix of (transformed) covariates used to estimate E[Y|X,W]
fmla.xw <- formula(paste("~ 0 +", paste0("bs(", covariates, ", df=3)", "*", treatment, c
ollapse=" + ")))
XW <- model.matrix(fmla.xw, ICLR_analysis_ARI)
# Matrix of (transformed) covariates used to predict E[Y|X,W=w] for each w in {0, 1}
data.1 <- ICLR_analysis_ARI
data.1[,treatment] <- 1
XW1 <- model.matrix(fmla.xw, data.1)  # setting W=1
data.0 <- ICLR_analysis_ARI
data.0[,treatment] <- 0
XW0 <- model.matrix(fmla.xw, data.0)  # setting W=0

# Matrix of (transformed) covariates used to estimate and predict e(X) = P[W=1|X]
fmla.x <- formula(paste(" ~ 0 + ", paste0("bs(", covariates, ", df=3)", collapse=" + "
)))
XX <- model.matrix(fmla.x, ICLR_analysis_ARI)

# (Optional) Not penalizing the main effect (the coefficient on W)
penalty.factor <- rep(1, ncol(XW))
penalty.factor[colnames(XW) == treatment] <- 0

# Cross-fitted estimates of E[Y|X,W=1], E[Y|X,W=0] and e(X) = P[W=1|X]
mu.hat.1 <- rep(NA, n)
mu.hat.0 <- rep(NA, n)
e.hat <- rep(NA, n)
for (idx in indices) {
  # Estimate outcome model and propensity models
  # Note how cross-validation is done (via cv.glmnet) within cross-fitting!
  outcome.model <- cv.glmnet(x=XW[-idx,], y=Y[-idx], family="gaussian", penalty.factor=p
enalty.factor)
  propensity.model <- cv.glmnet(x=XX[-idx,], y=W[-idx], family="binomial")

  # Predict with cross-fitting
  mu.hat.1[idx] <- predict(outcome.model, newx=XW1[idx,], type="response")
  mu.hat.0[idx] <- predict(outcome.model, newx=XW0[idx,], type="response")
  e.hat[idx] <- predict(propensity.model, newx=XX[idx,], type="response")
}

# Commpute the summand in AIPW estimator
```

```
aipw.scores <- (mu.hat.1 - mu.hat.0
                + W / e.hat * (Y -  mu.hat.1)
                - (1 - W) / (1 - e.hat) * (Y -  mu.hat.0))

# Tally up results
ate.aipw.est <- mean(aipw.scores)
ate.aipw.se <- sd(aipw.scores) / sqrt(n)
ate.aipw.tstat <- ate.aipw.est / ate.aipw.se
ate.aipw.pvalue <- 2*(pnorm(1 - abs(ate.aipw.tstat)))
ate.aipw.results <- c(estimate=ate.aipw.est, std.error=ate.aipw.se, t.stat=ate.aipw.tsta
t, pvalue=ate.aipw.pvalue)
print(ate.aipw.results)
```

```
      estimate      std.error         t.stat         pvalue
 -1.071005e+00   4.823925e-02  -2.220194e+01  9.163442e-100
```

Hide

```
X <- model.matrix(formula("~ 0 + review_length_quantile + confidence_number"), ICLR_anal
ysis_ARI)
W <- ICLR_analysis_ARI$W
Y <- ICLR_analysis_ARI$rating_number
#png(file="ARI_covariates_confidence_review_len.png")
par(mfrow=c(1,2))
for (w in c(0, 1)) {
  plot(X[W==w,1] + rnorm(n=sum(W==w), sd=.1), X[W==w,2] + rnorm(n=sum(W==w), sd=.1),
       pch=ifelse(Y, 23, 21), cex=1, col=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)),
       bg=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)), main=ifelse(w, "Treated", "Untreate
d"), xlab="confidence_number", ylab="review_length_quantile")
}
```

Hide

```
X <- model.matrix(formula("~ 0 + confidence_number + race"), ICLR_analysis_ARI)
W <- ICLR_analysis_ARI$W
Y <- ICLR_analysis_ARI$rating_number
par(mfrow=c(1,2))
for (w in c(0, 1)) {
  plot(X[W==w,1] + rnorm(n=sum(W==w), sd=.1), X[W==w,2] + rnorm(n=sum(W==w), sd=.1),
       pch=ifelse(Y, 23, 21), cex=1, col=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)),
       bg=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)), main=ifelse(w, "Treated", "Untreate
d"), xlab="confidence_number", ylab="race")
}
```

Hide

```
X <- model.matrix(formula("~ 0 + review_length_quantile + race"), ICLR_analysis_ARI)
W <- ICLR_analysis_ARI$W
Y <- ICLR_analysis_ARI$rating_number
par(mfrow=c(1,2))
for (w in c(0, 1)) {
  plot(X[W==w,1] + rnorm(n=sum(W==w), sd=.1), X[W==w,2] + rnorm(n=sum(W==w), sd=.1),
       pch=ifelse(Y, 23, 21), cex=1, col=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)),
       bg=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)), main=ifelse(w, "Treated", "Untreate
d"), xlab="review_length_quantile", ylab="race")
}
```

Untreated
race
0.0
0.5
1.0
1
2
3
4
review_length_quantile

Treated
race
0.0
0.5
1.0
1
2
3
4
review_length_quantile

Hide

```
set.seed(1)
group = 'race'
covariates = c("confidence_number","review_length_quantile", "race")
fmla <- formula(paste0("~ 0 +", paste0(covariates, collapse="+")))
XX <- model.matrix(fmla, ICLR_analysis_ARI)
W=ICLR_analysis_ARI[,"W"]
Y=ICLR_analysis_ARI[,"rating_number"]


forest.tau <- causal_forest(XX, Y, W)

tau.hat <- predict(forest.tau)$predictions
m.hat <- forest.tau$Y.hat  # E[Y|X] estimates
e.hat <- forest.tau$W.hat  # e(X) := E[W|X] estimates (or known quantity)
tau.hat <- forest.tau$predictions  # tau(X) estimates

# Predicting mu.hat(X[i], 1) and mu.hat(X[i], 0) for obs in held-out sample
# Note: to understand this, read equations 6-8 in this vignette
# https://grf-labs.github.io/grf/articles/muhats.html
mu.hat.0 <- m.hat - e.hat * tau.hat        # E[Y|X,W=0] = E[Y|X] - e(X)*tau(X)
mu.hat.1 <- m.hat + (1 - e.hat) * tau.hat  # E[Y|X,W=1] = E[Y|X] + (1 - e(X))*tau(X)

# Compute AIPW scores
aipw.scores <- tau.hat + W / e.hat * (Y -  mu.hat.1) - (1 - W) / (1 - e.hat) * (Y -  mu.
hat.0)

# Estimate average treatment effect conditional on group membership
fmla <- formula(paste0('aipw.scores ~ factor(', group, ')'))
ols <- lm(fmla, data=transform(ICLR_analysis_ARI[covariates], aipw.scores=aipw.scores))
summary_rw_lm(ols)
```

```
                      Estimate Std. Error Orig. p-value Adj. p-value
(Intercept)          -0.1779189 0.03573403  6.458432e-07       0.0001
factor(race)black    -0.1657387 0.09321692  7.542458e-02       0.1439
factor(race)hispanic -0.2720958 0.09217297  3.161659e-03       0.0083
factor(race)white    -0.1028437 0.06165488  9.532431e-02       0.1439
```

Hide

```
test_calibration(forest.tau)
```

```
Best linear fit using forest predictions (on held-out data)
as well as the mean forest prediction as regressors, along
with one-sided heteroskedasticity-robust (HC3) SEs:

                              Estimate Std. Error t value    Pr(>t)
mean.forest.prediction         1.00069    0.10300  9.7157 < 2.2e-16 ***
differential.forest.prediction  0.75081    0.11220  6.6914 1.142e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hide

```
best_linear_projection(forest.tau,ICLR_analysis_ARI[,covariates])
```

```
Best linear projection of the conditional average treatment effect.
Confidence intervals are cluster- and heteroskedasticity-robust (HC3):

                          Estimate  Std. Error t value Pr(>|t|)
(Intercept)              0.73752719  0.38557259  1.9128 0.055790 .
confidence_number2      -0.61918346  0.40394541 -1.5328 0.125335
confidence_number3      -0.97196064  0.38948874 -2.4955 0.012589 *
confidence_number4      -0.91813845  0.38935204 -2.3581 0.018380 *
confidence_number5      -1.18820584  0.39512547 -3.0072 0.002641 **
review_length_quantile2  0.20978171  0.06898097  3.0412 0.002361 **
review_length_quantile3 -0.05161598  0.07167919 -0.7201 0.471476
review_length_quantile4 -0.00036196  0.07979586 -0.0045 0.996381
raceblack               -0.15495465  0.09417114 -1.6455 0.099895 .
racehispanic            -0.26695208  0.09157396 -2.9152 0.003560 **
racewhite               -0.10460764  0.06026347 -1.7358 0.082612 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hide

```
authors_p = inner_join(ICLR_2017_authors[c('year','authors','forum')],ICLR_2018[c('year'
,'authors','forum')], by ='authors')
authors_p = inner_join(authors_p, ICLR_2019[c('year','authors','forum')], by ='authors')
authors_p = unique(authors_p)

authors_p = tidyr::extract(authors_p,authors,c("First_Name","Last_Name"), "([^ ]+) (.*)"
)

authors_p = na.omit(authors_p)

authors_p['race'] = predict_ethnicity(firstnames = authors_p$First_Name, lastnames = aut
hors_p$Last_Name, method = "fullname")[7]

authors_p = authors_p %>% dplyr::select(First_Name, Last_Name, race)

authors_p = left_join(authors_p,authors, by =c("First_Name","Last_Name"))
authors_p = authors_p %>% dplyr::select(-race.x)
authors_p = unique(authors_p)


authors_p = dplyr::rename(authors_p, race=race.y)


df = authors_p %>% group_by(race) %>%
  dplyr::summarise(n = n()) %>%
  mutate(freq = round(n  / sum(n ), 3))
df
```

race <chr>	n <int>	freq <dbl>
asian	270	0.582
black	29	0.062
hispanic	48	0.103
white	117	0.252

4 rows

Hide

```
prestigious_forums = unique(authors_p$forum)

ICLR_reviews = ICLR_reviews %>% mutate (presti_paper = ifelse(forum %in% prestigious_for
ums, 1, 0))

#only prestigious authors
rm(ICLR_analysis_ARIP)
```

```
Warning in rm(ICLR_analysis_ARIP) :
  object 'ICLR_analysis_ARIP' not found
```

Hide

```
ICLR_analysis_ARIP = ICLR_reviews %>% filter(year %in% c(2017,2018))

ICLR_analysis_ARIP = full_join(x=ICLR_analysis_ARIP,y=authors_p,by="forum")
#data audit
#ICLR_reviews %>% filter(forum == 'B1-q5Pqxl')
#ICLR_analysis_ARIP %>% filter(forum == 'B1-q5Pqxl')
#authors_p %>% filter(forum == 'B1-q5Pqxl')


ICLR_analysis_ARIP = ICLR_analysis_ARIP %>% mutate(W = if_else(year.x==2017,0,1))

ICLR_analysis_ARIP = na.omit(ICLR_analysis_ARIP)
ICLR_analysis_ARIP = unique(ICLR_analysis_ARIP)

ICLR_analysis_ARIP%>% group_by(W) %>% summarise(n())
```

W <dbl>	n() <int>
0	403
1	496

2 rows

Hide

```
#auditing data to see if correct
#authors %>% filter(First_Name=="Angeliki", Last_Name=="Lazaridou")
#authors_p %>% filter(First_Name=="Angeliki", Last_Name=="Lazaridou")
#ICLR_2017_authors %>% dplyr::select(forum, year, authors)%>%  filter(authors == "Angeli
ki Lazaridou") %>% unique()
#ICLR_2018 %>% dplyr::select(forum, year, authors)%>% filter(authors=="Angeliki Lazarido
u") %>% unique()
#ICLR_2019 %>% dplyr::select(forum, year, authors)%>%  filter(authors=="Angeliki Lazarid
ou") %>% unique()
```

Hide

```
ICLR_analysis_ARI = ICLR_reviews %>% filter(year %in% c(2017,2018))
ICLR_analysis_ARI = full_join(x=ICLR_analysis_ARI,y=authors,by="forum")
ICLR_analysis_ARI = ICLR_analysis_ARI %>% mutate(W = if_else(year.x==2017,0,1))
ICLR_analysis_ARI = na.omit(ICLR_analysis_ARI)
ICLR_analysis_ARI = unique(ICLR_analysis_ARI)

set.seed(1)
group = 'presti_paper'
covariates = c("confidence_number","review_length_quantile", "race",'presti_paper')
fmla <- formula(paste0("~ 0 +", paste0(covariates, collapse="+")))
XX <- model.matrix(fmla, ICLR_analysis_ARI)
W=ICLR_analysis_ARI[,"W"]
Y=ICLR_analysis_ARI[,"rating_number"]


forest.tau <- causal_forest(XX, Y, W)

tau.hat <- predict(forest.tau)$predictions
m.hat <- forest.tau$Y.hat  # E[Y|X] estimates
e.hat <- forest.tau$W.hat  # e(X) := E[W|X] estimates (or known quantity)
tau.hat <- forest.tau$predictions  # tau(X) estimates

# Predicting mu.hat(X[i], 1) and mu.hat(X[i], 0) for obs in held-out sample
# Note: to understand this, read equations 6-8 in this vignette
# https://grf-labs.github.io/grf/articles/muhats.html
mu.hat.0 <- m.hat - e.hat * tau.hat        # E[Y|X,W=0] = E[Y|X] - e(X)*tau(X)
mu.hat.1 <- m.hat + (1 - e.hat) * tau.hat  # E[Y|X,W=1] = E[Y|X] + (1 - e(X))*tau(X)

# Compute AIPW scores
aipw.scores <- tau.hat + W / e.hat * (Y -  mu.hat.1) - (1 - W) / (1 - e.hat) * (Y -  mu.
hat.0)

# Estimate average treatment effect conditional on group membership
fmla <- formula(paste0('aipw.scores ~ factor(', group, ')'))
ols <- lm(fmla, data=transform(ICLR_analysis_ARI[covariates], aipw.scores=aipw.scores))
summary_rw_lm(ols)
```

```
                        Estimate Std. Error Orig. p-value Adj. p-value
(Intercept)           -0.1662739 0.02953447  1.834082e-08       0.0000
factor(presti_paper)1 -0.1297235 0.06231457  3.738088e-02       0.0346
```

Hide

```
test_calibration(forest.tau)
```

```
Best linear fit using forest predictions (on held-out data)
as well as the mean forest prediction as regressors, along
with one-sided heteroskedasticity-robust (HC3) SEs:

                              Estimate Std. Error t value    Pr(>t)
mean.forest.prediction        1.002609   0.128767  7.7862 3.661e-15 ***
differential.forest.prediction 0.838857   0.083417 10.0562 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hide

```
X <- model.matrix(formula("~ 0 + review_length_quantile + confidence_number"), ICLR_anal
ysis_ARIP)
W <- ICLR_analysis_ARIP$W
Y <- ICLR_analysis_ARIP$rating_number
par(mfrow=c(1,2))
for (w in c(0, 1)) {
  plot(X[W==w,1] + rnorm(n=sum(W==w), sd=.1), X[W==w,2] + rnorm(n=sum(W==w), sd=.1),
       pch=ifelse(Y, 23, 21), cex=1, col=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)),
       bg=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)), main=ifelse(w, "Treated", "Untreate
d"), xlab="confidence_number", ylab="review_length_quantile",ylim=c(0,5))
}
```

Hide

```
X <- model.matrix(formula("~ 0 + confidence_number + race"), ICLR_analysis_ARIP)
W <- ICLR_analysis_ARIP$W
Y <- ICLR_analysis_ARIP$rating_number
par(mfrow=c(1,2))
for (w in c(0, 1)) {
  plot(X[W==w,1] + rnorm(n=sum(W==w), sd=.1), X[W==w,2] + rnorm(n=sum(W==w), sd=.1),
       pch=ifelse(Y, 23, 21), cex=1, col=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)),
       bg=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)), main=ifelse(w, "Treated", "Untreate
d"), xlab="confidence_number", ylab="race")
}
```

Hide

```
X <- model.matrix(formula("~ 0 + review_length_quantile + race"), ICLR_analysis_ARIP)
W <- ICLR_analysis_ARIP$W
Y <- ICLR_analysis_ARIP$rating_number
par(mfrow=c(1,2))
for (w in c(0, 1)) {
  plot(X[W==w,1] + rnorm(n=sum(W==w), sd=.1), X[W==w,2] + rnorm(n=sum(W==w), sd=.1),
       pch=ifelse(Y, 23, 21), cex=1, col=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)),
       bg=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)), main=ifelse(w, "Treated", "Untreate
d"), xlab="review_length_quantile", ylab="race")
}
```

Hide

```
#names(ICLR_analysis)
#c("rating","confidence","review","rating number","confidence number","review length","F
irst_Name","Last_Name","race")

covariates = c("confidence_number","review_length_quantile", "race")
XX <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))), data=ICLR_ana
lysis_ARIP)

set.seed(1)

forest <- causal_forest(
              X=XX,
              W=ICLR_analysis_ARIP[,"W"],
              Y=ICLR_analysis_ARIP[,"rating_number"]
              )

forest.ate <- average_treatment_effect(forest)

forest.ate
```

```
  estimate    std.err
-0.2240945  0.1054972
```

Hide

```
hist(forest$W.hat, main="Estimated propensity scores \n(causal forest with )", xlim=c(-.
1, 1.1))
```

Hide

```
set.seed(1)
group = 'race'
fmla <- formula(paste0("~ 0 +", paste0(covariates, collapse="+")))
XX <- model.matrix(fmla, ICLR_analysis_ARIP)
W=ICLR_analysis_ARIP[,"W"]
Y=ICLR_analysis_ARIP[,"rating_number"]


forest.tau <- causal_forest(XX, Y, W)

tau.hat <- predict(forest.tau)$predictions
m.hat <- forest.tau$Y.hat  # E[Y|X] estimates
e.hat <- forest.tau$W.hat  # e(X) := E[W|X] estimates (or known quantity)
tau.hat <- forest.tau$predictions  # tau(X) estimates

# Predicting mu.hat(X[i], 1) and mu.hat(X[i], 0) for obs in held-out sample
# Note: to understand this, read equations 6-8 in this vignette
# https://grf-labs.github.io/grf/articles/muhats.html
mu.hat.0 <- m.hat - e.hat * tau.hat        # E[Y|X,W=0] = E[Y|X] - e(X)*tau(X)
mu.hat.1 <- m.hat + (1 - e.hat) * tau.hat  # E[Y|X,W=1] = E[Y|X] + (1 - e(X))*tau(X)

# Compute AIPW scores
aipw.scores <- tau.hat + W / e.hat * (Y -  mu.hat.1) - (1 - W) / (1 - e.hat) * (Y -  mu.
hat.0)

# Estimate average treatment effect conditional on group membership
fmla <- formula(paste0('aipw.scores ~ factor(', group, ')'))
ols <- lm(fmla, data=transform(ICLR_analysis_ARIP[covariates], aipw.scores=aipw.scores))
summary_rw_lm(ols)
```

```
                       Estimate Std. Error Orig. p-value Adj. p-value
(Intercept)          -0.12462750  0.1401247     0.3740249       0.7200
factor(race)black    -0.59928829  0.4313341     0.1650618       0.4836
factor(race)hispanic -0.30047809  0.3540865     0.3963304       0.7200
factor(race)white    -0.08275372  0.2544635     0.7450992       0.7458
```

Hide

```
test_calibration(forest.tau )
```

```
Best linear fit using forest predictions (on held-out data)
as well as the mean forest prediction as regressors, along
with one-sided heteroskedasticity-robust (HC3) SEs:

                              Estimate Std. Error t value  Pr(>t)
mean.forest.prediction         0.93921    0.49229  1.9078 0.02837 *
differential.forest.prediction -0.81826    0.58163 -1.4068 0.92009
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hide

```
ICLR_analysis_Submission = ICLR_reviews


ICLR_analysis_Submission = ICLR_analysis_Submission %>% group_by(forum, year) %>%
  summarise(ave_rating = mean(rating_number), ave_review_len = mean(review_length), ave_
confidence_number = mean(confidence_number))
```

```
`summarise()` has grouped output by 'forum'. You can override using the `.groups`
argument.
```

Hide

```
ICLR_analysis_Submission = full_join(x=ICLR_analysis_Submission,y=authors,by="forum")

ICLR_analysis_Submission = na.omit(ICLR_analysis_Submission)

ICLR_analysis_Submission = dummy_cols(ICLR_analysis_Submission, select_columns = "race")
```

```
Registered S3 method overwritten by 'data.table':
  method           from
  print.data.table
```

Hide

```
ICLR_analysis_Submission = ICLR_analysis_Submission %>% group_by(forum) %>%
  summarise(year = year.x,
            ave_review_len = ave_review_len,
            ave_rating=ave_rating,
            ave_confidence_number = ave_confidence_number,
            asian_authors = sum(race_asian),
            black_authors = sum(race_black),
            hispanic_authors = sum(race_hispanic),
            white_authors = sum(race_white),
            total_authors = sum(race_asian,race_black,race_hispanic,race_white)) %>%
  unique()
```

```
`summarise()` has grouped output by 'forum'. You can override using the `.groups`
argument.
```

Hide

```
ICLR_analysis_Submission = na.omit(ICLR_analysis_Submission)
#ICLR_analysis_Submission
ICLR_analysis_Submission = ICLR_analysis_Submission %>% mutate(W = if_else(year==2017,0,
1))


ICLR_analysis_Submission$ave_review_length_quantile = ntile(ICLR_analysis_Submission$ave
_review_len, 4)


ICLR_analysis_Submission$racial_majority = colnames(ICLR_analysis_Submission[,c( "asian_
authors","black_authors", "hispanic_authors","white_authors")])[max.col(ICLR_analysis_Su
bmission[,c( "asian_authors","black_authors", "hispanic_authors","white_authors")])]
#ICLR_analysis_Submission %>% filter(First_Name=='Jake')

prestigious_forums = unique(authors_p$forum)

ICLR_analysis_Submission = ICLR_analysis_Submission %>% mutate (presti_paper = ifelse(fo
rum %in% prestigious_forums, 1, 0))

ICLR_analysis_Submission %>% filter(forum=='B1-Hhnslg')
```

forum <chr>	y... <int>	ave_review_len <dbl>	ave_rating <dbl>	ave_confidence_number <dbl>	asian_authors <int>	black_
B1-Hhnslg	2017	1460.667	5	4	0	

1 row | 1-7 of 14 columns

Hide

```
ICLR_analysis_Submission %>% filter(forum=='Hk8N3Sclg')
```

forum <chr>	y... <int>	ave_review_len <dbl>	ave_rating <dbl>	ave_confidence_number <dbl>	asian_authors <int>	black_
Hk8N3Sclg	2017	1646.333	7	3	2	

1 row | 1-7 of 14 columns

Hide

```
hist(ICLR_analysis_Submission$ave_review_length_quantile)
```

Hide

```
names(ICLR_analysis_Submission)
```

```
 [1] "forum"                      "year"
 [3] "ave_review_len"             "ave_rating"
 [5] "ave_confidence_number"      "asian_authors"
 [7] "black_authors"              "hispanic_authors"
 [9] "white_authors"              "total_authors"
[11] "W"                          "ave_review_length_quantile"
[13] "racial_majority"            "presti_paper"
```

Hide

```
ICLR_analysis_Submission_1718 = ICLR_analysis_Submission %>% filter(year %in% c(2017,201
8))
ICLR_analysis_Submission_1718 = as.data.frame(ICLR_analysis_Submission_1718)

names(ICLR_analysis_Submission_1718)
```

```
 [1] "forum"                      "year"
 [3] "ave_review_len"             "ave_rating"
 [5] "ave_confidence_number"      "asian_authors"
 [7] "black_authors"              "hispanic_authors"
 [9] "white_authors"              "total_authors"
[11] "W"                          "ave_review_length_quantile"
[13] "racial_majority"            "presti_paper"
```

Hide

```
covariates = c("ave_confidence_number","ave_review_length_quantile", "asian_authors","bl
ack_authors", "hispanic_authors","white_authors")
XX <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))), data=ICLR_ana
lysis_Submission_1718)

set.seed(1)

forest <- causal_forest(
              X=XX,
              W=ICLR_analysis_Submission_1718[,"W"],
              Y=ICLR_analysis_Submission_1718[,"ave_rating"]
              )

forest.ate <- average_treatment_effect(forest)

forest.ate
```

```
   estimate     std.err
-0.25257828  0.07744789
```

Hide

```
hist(forest$W.hat, main="Estimated propensity scores \n(causal forest with submission da
ta)", xlim=c(-.1, 1.1))
```

Hide

```
covariates = c("ave_confidence_number","ave_review_length_quantile", "asian_authors","bl
ack_authors", "hispanic_authors","white_authors")
# Here, adding covariates and their interactions, though there are many other possibilit
ies.
fmla <- formula(paste("~ 0 +", paste(apply(expand.grid(covariates, covariates), 1, funct
ion(x) paste0(x, collapse="*")), collapse="+")))

# Using the propensity score estimated above
#check to see if you are using ATE causal forest ICLR_analysis_Submission_1718
e.hat <- forest$W.hat

XX <- model.matrix(fmla, ICLR_analysis_Submission_1718)
W <- ICLR_analysis_Submission_1718[,"W"]
pp <- ncol(XX)

# Unadjusted covariate means, variances and standardized abs mean differences
means.treat <- apply(XX[W == 1,], 2, mean)
means.ctrl <- apply(XX[W == 0,], 2, mean)
abs.mean.diff <- abs(means.treat - means.ctrl)

var.treat <- apply(XX[W == 1,], 2, var)
var.ctrl <- apply(XX[W == 0,], 2, var)
std <- sqrt(var.treat + var.ctrl)

# Adjusted covariate means, variances and standardized abs mean differences
means.treat.adj <- apply(XX*W/e.hat, 2, mean)
```

```
Warning in XX * W/e.hat :
  longer object length is not a multiple of shorter object length
```

Hide

```
means.ctrl.adj <- apply(XX*(1-W)/(1-e.hat), 2, mean)
```

```
Warning in XX * (1 - W)/(1 - e.hat) :
  longer object length is not a multiple of shorter object length
```

Hide

```
abs.mean.diff.adj <- abs(means.treat.adj - means.ctrl.adj)

var.treat.adj <- apply(XX*W/e.hat, 2, var)
```

```
Warning in XX * W/e.hat :
  longer object length is not a multiple of shorter object length
```

Hide

```
var.ctrl.adj <- apply(XX*(1-W)/(1-e.hat), 2, var)
```

```
Warning in XX * (1 - W)/(1 - e.hat) :
  longer object length is not a multiple of shorter object length
```

Hide

```
std.adj <- sqrt(var.treat.adj + var.ctrl.adj)

# Plotting
#png(file = "balance_ICLR_analysis_Submission_1718_ATE.png")
par(oma=c(0,4,0,0))
plot(-2, xaxt="n", yaxt="n", xlab="", ylab="", xlim=c(-.01, 1.3), ylim=c(0, pp+1), main=
"Standardized absolute mean differences \n Submission Level Data 2017-2018")
axis(side=1, at=c(-1, 0, 1), las=1)
```

Hide

```
lines(abs.mean.diff / std, seq(1, pp), type="p", col="blue", pch=19)
lines(abs.mean.diff.adj / std.adj, seq(1, pp), type="p", col="orange", pch=19)
```

Hide

```
legend("topright", c("Unadjusted", "Adjusted"), col=c("blue", "orange"), pch=19)
abline(v = seq(0, 1, by=.25), lty = 2, col = "grey", lwd=.5)
```

Hide

```
abline(h = 1:pp,  lty = 2, col = "grey", lwd=.5)
mtext(colnames(XX), side=2, cex=0.42, at=1:pp, padj=.4, adj=1, col="black", las=1, line=
.3)
```

Hide

```
abline(v = 0)
```

Hide

```
ICLR_analysis_Submission_p_1718 = ICLR_analysis_Submission_1718 %>% filter(presti_paper
== 1)
covariates = c("ave_confidence_number","ave_review_length_quantile", "asian_authors","bl
ack_authors", "hispanic_authors","white_authors")
XX <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))), data=ICLR_ana
lysis_Submission_p_1718)

set.seed(1)

forest <- causal_forest(
             X=XX,
             W=ICLR_analysis_Submission_p_1718[,"W"],
             Y=ICLR_analysis_Submission_p_1718[,"ave_rating"]
             #,num.trees = 100
             )


forest.ate <- average_treatment_effect(forest)


forest.ate
```

```
  estimate    std.err
-0.3084228  0.1432176
```

Hide

```
hist(forest$W.hat, main="Estimated propensity scores \n(causal forest with submission da
ta)", xlim=c(-.1, 1.1))
```

Estimated propensity scores
(causal forest with submission data)
Frequency
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0
forest$W.hat

Hide

```
ICLR_analysis_Submission_1819 = ICLR_analysis_Submission %>% filter(year %in% c(2019,201
8))

ICLR_analysis_Submission_1819 = ICLR_analysis_Submission_1819 %>% mutate(W = if_else(yea
r==2018,0,1))
ICLR_analysis_Submission_1819 = as.data.frame(ICLR_analysis_Submission_1819)

covariates = c("ave_confidence_number","ave_review_length_quantile", "asian_authors","bl
ack_authors", "hispanic_authors","white_authors")
XX <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))), data=ICLR_ana
lysis_Submission_1819)

set.seed(1)

forest <- causal_forest(
             X=XX,
             W=ICLR_analysis_Submission_1819[,"W"],
             Y=ICLR_analysis_Submission_1819[,"ave_rating"]
             #,num.trees = 100
             )

#what does the target sample mean and what I should use
forest.ate <- average_treatment_effect(forest)

#forest.ate <- average_treatment_effect(forest, target.sample="control")
forest.ate
```

```
    estimate      std.err
-0.009684859  0.052313303
```

Hide

```
hist(forest$W.hat)
```

Hide

```
hist(get_scores(forest))
```

Hide

```
ICLR_analysis_Submission_1819 = ICLR_analysis_Submission %>% filter(year %in% c(2019,201
8))

ICLR_analysis_Submission_1819 = ICLR_analysis_Submission_1819 %>% mutate(W = if_else(yea
r==2018,0,1))
ICLR_analysis_Submission_1819 = as.data.frame(ICLR_analysis_Submission_1819)
ICLR_analysis_Submission_p_1819 = ICLR_analysis_Submission_1819 %>% filter(presti_paper=
=1)

covariates = c("ave_confidence_number","ave_review_length_quantile", "asian_authors","bl
ack_authors", "hispanic_authors","white_authors")
XX <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))), data=ICLR_ana
lysis_Submission_p_1819)

set.seed(1)

forest <- causal_forest(
              X=XX,
              W=ICLR_analysis_Submission_p_1819[,"W"],
              Y=ICLR_analysis_Submission_p_1819[,"ave_rating"]
              #,num.trees = 100
              )

#what does the target sample mean and what I should use
forest.ate <- average_treatment_effect(forest)

#forest.ate <- average_treatment_effect(forest, target.sample="control")
forest.ate
```

```
  estimate    std.err
0.02085575 0.12526899
```

Hide

```
hist(forest$W.hat)
```

Hide

```
hist(get_scores(forest))
```

Hide

```
set.seed(1)
rm(HTE_Submission)
```

```
Warning in rm(HTE_Submission) : object 'HTE_Submission' not found
```

Hide

```
group = 'racial_majority'

covariates = c("ave_confidence_number","ave_review_length_quantile", 'racial_majority',
'presti_paper')

#ICLR_analysis_Submission_1718$racial_majority = as.factor(ICLR_analysis_Submission_1718
[,'racial_majority'])
#ICLR_analysis_Submission_1718$racial_majority = unclass(ICLR_analysis_Submission_1718$r
acial_majority)
#attr(,"levels") "asian_authors"    "black_authors"    "hispanic_authors" "white_author
s"
XX <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))), data=ICLR_ana
lysis_Submission_1718)
W=ICLR_analysis_Submission_1718[,"W"]
Y=ICLR_analysis_Submission_1718[,"ave_rating"]



forest.tau <- causal_forest(XX, Y, W)

tau.hat <- predict(forest.tau)$predictions
m.hat <- forest.tau$Y.hat  # E[Y|X] estimates
e.hat <- forest.tau$W.hat  # e(X) := E[W|X] estimates (or known quantity)
tau.hat <- forest.tau$predictions  # tau(X) estimates

# Predicting mu.hat(X[i], 1) and mu.hat(X[i], 0) for obs in held-out sample
# Note: to understand this, read equations 6-8 in this vignette
# https://grf-labs.github.io/grf/articles/muhats.html
mu.hat.0 <- m.hat - e.hat * tau.hat        # E[Y|X,W=0] = E[Y|X] - e(X)*tau(X)
mu.hat.1 <- m.hat + (1 - e.hat) * tau.hat  # E[Y|X,W=1] = E[Y|X] + (1 - e(X))*tau(X)

# Compute AIPW scores
aipw.scores <- tau.hat + W / e.hat * (Y -  mu.hat.1) - (1 - W) / (1 - e.hat) * (Y -  mu.
hat.0)

# Estimate average treatment effect conditional on group membership
fmla <- formula(paste0('aipw.scores ~ factor(', group, ')'))
ols <- lm(fmla, data=transform(ICLR_analysis_Submission_1718[covariates], aipw.scores=ai
pw.scores))

summary_rw_lm(ols)
```

```
                                        Estimate Std. Error Orig. p-value Adj. p-value
(Intercept)                            -0.1197922  0.1067461     0.2619638       0.5812
factor(racial_majority)black_authors    0.2259062  0.3095470     0.4656374       0.7184
factor(racial_majority)hispanic_authors -0.1934307  0.3473473     0.5776989       0.7184
factor(racial_majority)white_authors   -0.2425740  0.1877978     0.1966833       0.5442
```

Hide

```
hist(forest.tau$W.hat)
```

Hide

```
test_calibration(forest.tau)
```

```
Best linear fit using forest predictions (on held-out data)
as well as the mean forest prediction as regressors, along
with one-sided heteroskedasticity-robust (HC3) SEs:

                              Estimate Std. Error t value    Pr(>t)
mean.forest.prediction         1.05417    0.32581  3.2355 0.0006214 ***
differential.forest.prediction -0.68608    0.46004 -1.4914 0.9319540
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Hide

```
X <- model.matrix(formula("~ 0 + ave_review_length_quantile + ave_confidence_number"), I
CLR_analysis_Submission_1718)
W <- ICLR_analysis_Submission_1718$W
Y <- ICLR_analysis_Submission_1718$ave_rating
#png(file="Submission_covariates_confidence_review_len.png")
par(mfrow=c(1,2))
for (w in c(0, 1)) {
  plot(X[W==w,1] + rnorm(n=sum(W==w), sd=.1), X[W==w,2] + rnorm(n=sum(W==w), sd=.1),
       pch=ifelse(Y, 23, 21), cex=1, col=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)),
       bg=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)), main=ifelse(w, "Treated", "Untreate
d"), xlab="ave_confidence_number", ylab="ave_review_length_quantile")
}
```

Hide

```
X <- model.matrix(formula("~ 0 + ave_confidence_number + total_authors"), ICLR_analysis_
Submission_1718)
W <- ICLR_analysis_Submission_1718$W
Y <- ICLR_analysis_Submission_1718$ave_rating
par(mfrow=c(1,2))
for (w in c(0, 1)) {
  plot(X[W==w,1] + rnorm(n=sum(W==w), sd=.1), X[W==w,2] + rnorm(n=sum(W==w), sd=.1),
       pch=ifelse(Y, 23, 21), cex=1, col=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)),
       bg=ifelse(Y, rgb(1,0,0,1/4), rgb(0,0,1,1/4)), main=ifelse(w, "Treated", "Untreate
d"), xlab="confidence_number", ylab="total_authors")
}
```

Hide

```
set.seed(1)
rm(HTE_Submission)
```

```
Warning in rm(HTE_Submission) : object 'HTE_Submission' not found
```

Hide

```
group = 'racial_majority'
#group = 'racial_majority'
covariates = c("ave_confidence_number","ave_review_length_quantile", 'racial_majority')

ICLR_analysis_Submission_p_1718 = ICLR_analysis_Submission_1718 %>% filter(presti_paper
 ==1)

ICLR_analysis_Submission_p_1718$racial_majority = as.factor(ICLR_analysis_Submission_p_1
718[,'racial_majority'])
ICLR_analysis_Submission_p_1718$racial_majority = unclass(ICLR_analysis_Submission_p_171
8$racial_majority)
#attr(,"levels") "asian_authors"    "black_authors"    "hispanic_authors" "white_author
s"
XX <- model.matrix(formula(paste0("~", paste0(covariates, collapse="+"))), data=ICLR_ana
lysis_Submission_p_1718)
W=ICLR_analysis_Submission_p_1718[,"W"]
Y=ICLR_analysis_Submission_p_1718[,"ave_rating"]



forest.tau <- causal_forest(XX, Y, W)

tau.hat <- predict(forest.tau)$predictions
m.hat <- forest.tau$Y.hat  # E[Y|X] estimates
e.hat <- forest.tau$W.hat  # e(X) := E[W|X] estimates (or known quantity)
tau.hat <- forest.tau$predictions  # tau(X) estimates

# Predicting mu.hat(X[i], 1) and mu.hat(X[i], 0) for obs in held-out sample
# Note: to understand this, read equations 6-8 in this vignette
# https://grf-labs.github.io/grf/articles/muhats.html
mu.hat.0 <- m.hat - e.hat * tau.hat        # E[Y|X,W=0] = E[Y|X] - e(X)*tau(X)
mu.hat.1 <- m.hat + (1 - e.hat) * tau.hat  # E[Y|X,W=1] = E[Y|X] + (1 - e(X))*tau(X)

# Compute AIPW scores
aipw.scores <- tau.hat + W / e.hat * (Y -  mu.hat.1) - (1 - W) / (1 - e.hat) * (Y -  mu.
hat.0)

# Estimate average treatment effect conditional on group membership
fmla <- formula(paste0('aipw.scores ~ factor(', group, ')'))
ols <- lm(fmla, data=transform(ICLR_analysis_Submission_p_1718[covariates], aipw.scores=
aipw.scores))

summary_rw_lm(ols)
```

```
                            Estimate Std. Error Orig. p-value Adj. p-value
(Intercept)              -0.166285939  0.2008648    0.40851462       0.7332
factor(racial_majority)2 -0.216384594  0.6025944    0.71982048       0.9185
factor(racial_majority)3 -1.469848622  0.7540035    0.05232372       0.1658
factor(racial_majority)4  0.004874355  0.3227878    0.98796334       0.9872
```

Hide

```
hist(forest.tau$W.hat)
```

Hide

```
test_calibration(forest.tau)
```

```
Best linear fit using forest predictions (on held-out data)
as well as the mean forest prediction as regressors, along
with one-sided heteroskedasticity-robust (HC3) SEs:

                               Estimate Std. Error t value   Pr(>t)
mean.forest.prediction          1.13234    0.47584  2.3797 0.009022 **
differential.forest.prediction -12.99434    2.21407 -5.8690 1.000000
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```