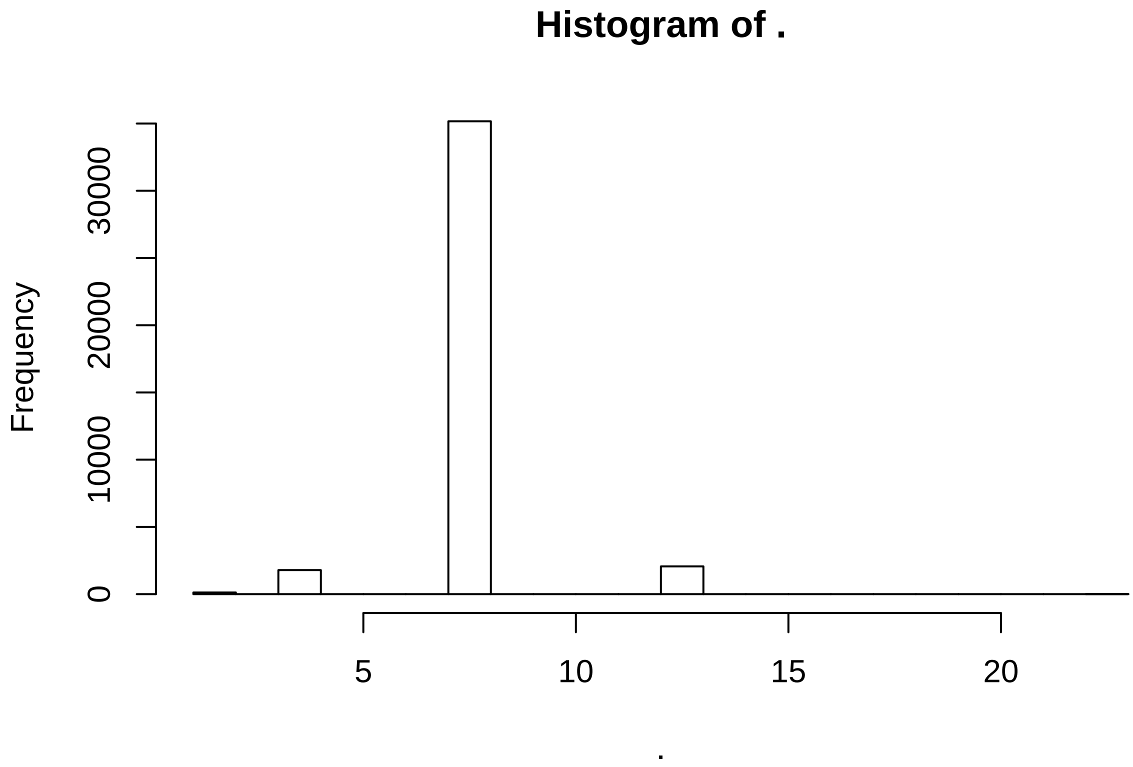
# Causal Inference Course Final Project

Ayush Kanodia and Mitchell Linegar

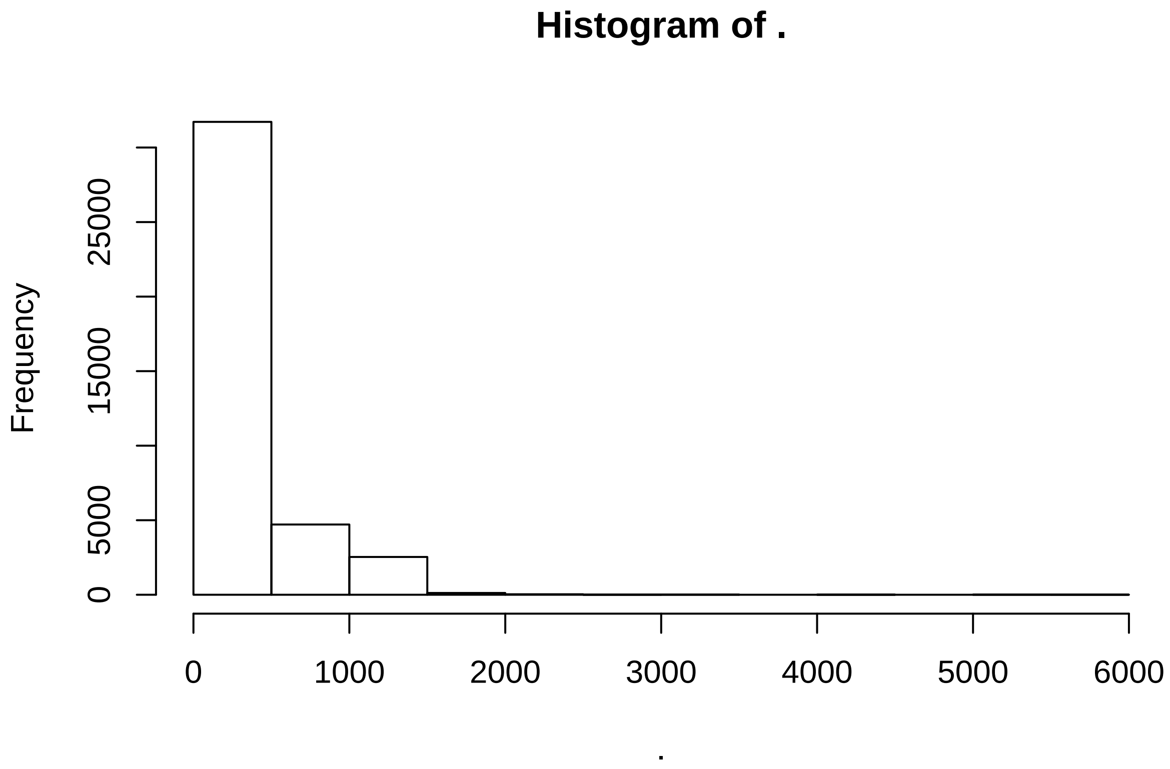
2020-06-05

## Contents

```r
# trip-level data
freadom <- fread(sprintf("%s/treatment_effects.csv", data_dir))


# cleaning
# some W not really continuous treatment - just different levels
freadom[,W_reading_time] %>% hist()
```



```r
freadom$W_wordcount %>% hist()
```



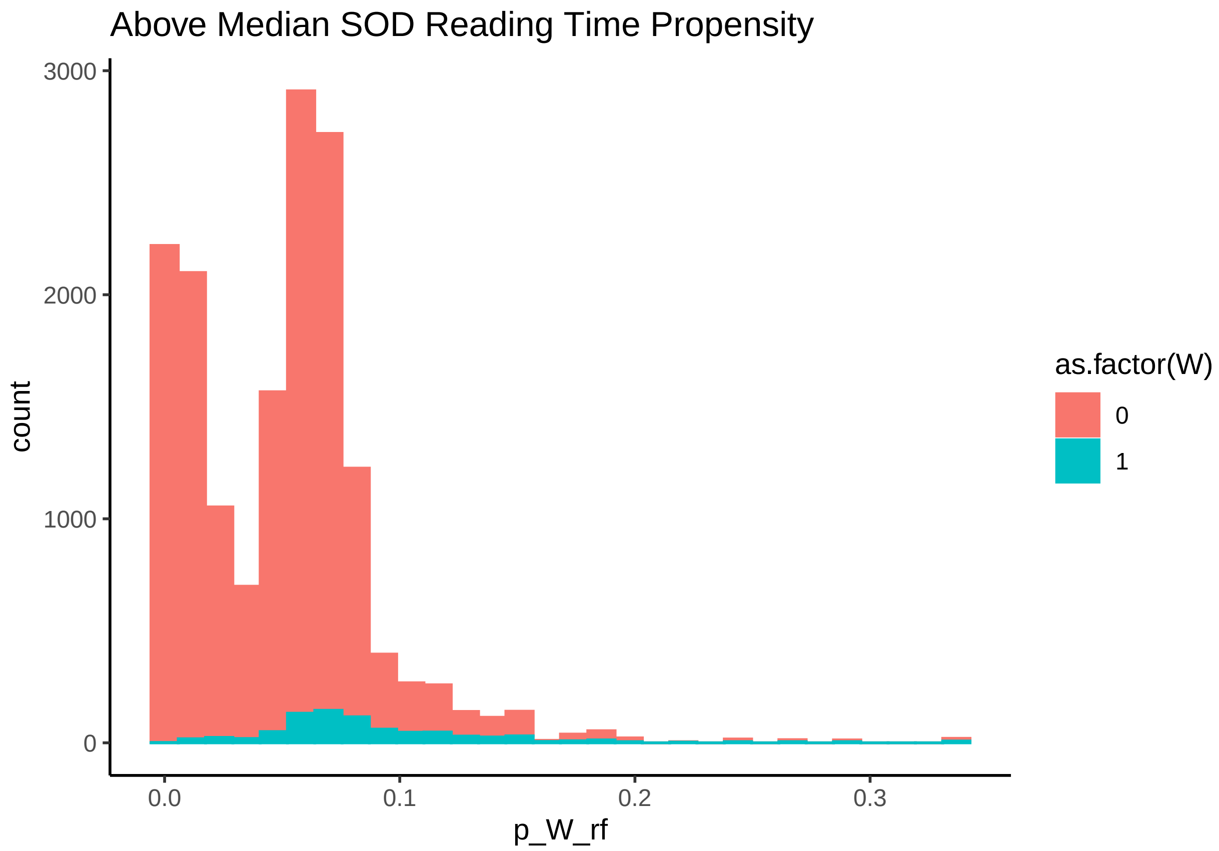
```r
# for now only look at 'medium' and 'high' levels of W_reading_time
freadom <- freadom[W_reading_time==12.5 | W_reading_time==7.5]
# look only at people who actually opened the SOD
freadom <- freadom[W_utility != 0]
# create binary treatment variables
freadom[, W_reading_time_high := ifelse(W_reading_time==max(W_reading_time), 1, 0)]
freadom[, W_wordcount_high := ifelse(W_wordcount>median(W_wordcount), 1, 0)]
# freadom[, W_wordcount_AB := ifelse(W_wordcount > median(W_wordcount), "A", "B")]
# make Utility (completion) binary
freadom[, `:=`(W_utility_high = ifelse(W_utility==3, 1, 0),
               Y_utility_high = ifelse(Y_utility==3, 1, 0))]


# make words read quartiles
freadom[, W_wordcount_qtile := ntile(W_wordcount, 4)]
```

```
# create words read count
freadom[, Y_wordsread := case_when(Y_utility==0 ~ 0,
                                 Y_utility==1 ~ W_wordcount * .25,
                                 Y_utility==2 ~ W_wordcount * .5,
                                 Y_utility==3 ~ W_wordcount)]

covariate_names <- colnames(freadom)[!grepl("W|Y", colnames(freadom), ignore.case = FALS

# construct various data sets we'll need
df_time_finish <- freadom[,c(covariate_names, "W_reading_time_high", "Y_utility"), with=
df_time_finish %>% setnames_W_Y()

df_wcount_finish <- freadom[,c(covariate_names, "W_wordcount_high", "Y_utility"), with=F
df_wcount_finish %>% setnames_W_Y()

df_time_lag <- freadom[,c(covariate_names, "W_reading_time_high", "Y_next_view_lag"), wi
df_time_lag %>% setnames_W_Y()

df_wcount_lag <- freadom[,c(covariate_names, "W_wordcount_high", "Y_next_view_lag"), wit
df_wcount_lag %>% setnames_W_Y()

df_finish_lag <- freadom[,c(covariate_names, "W_utility_high", "Y_next_view_lag"), with=
df_finish_lag %>% setnames_W_Y()

df_wcount_qtile_wordsread <- freadom[,c(covariate_names, "W_wordcount_qtile", "Y_wordsre
df_wcount_qtile_wordsread %>% setnames_W_Y()

# aggregate level data
freadom_agg <- fread(sprintf("%s/treatment_effects_agg.csv", data_dir))
```

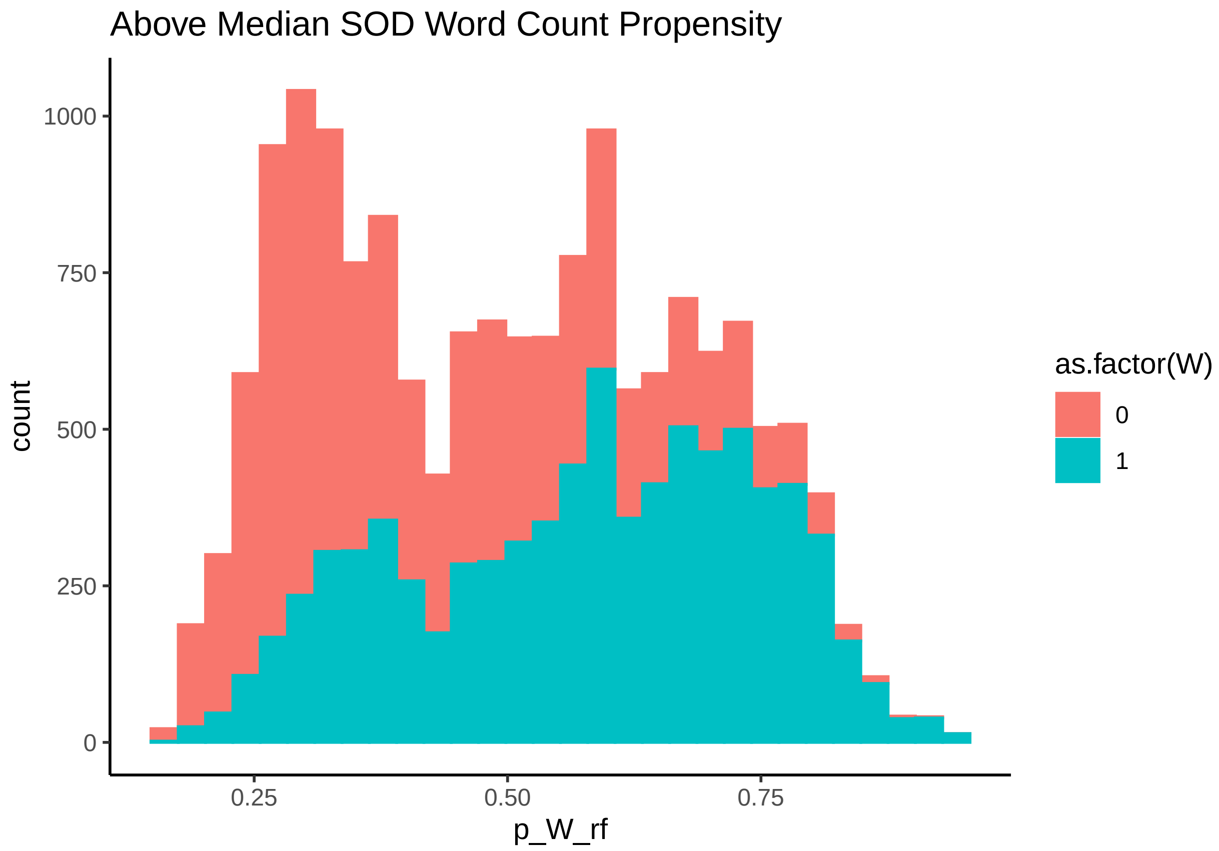
```
# don't need to do any filtering here because everything is an aggregate

# create binary treatment variables
freadom_agg[, W_reading_time_high := ifelse(W_reading_time>median(W_reading_time), 1, 0)
freadom_agg[, W_wordcount_high := ifelse(W_wordcount>median(W_wordcount), 1, 0)]

# make Utility (completion) binary
freadom_agg[, `:=`(W_utility_high = ifelse(W_utility>1.5, 1, 0),
               Y_utility_high = ifelse(Y_utility>1.5, 1, 0))]

# make words read quartiles
freadom_agg[, W_wordcount_qtile := ntile(W_wordcount, 4)]

# create words read count
freadom_agg[, Y_wordsread := Y_utility/3 * W_wordcount]

# construct aggregate data
agg_df_time_finish <- freadom_agg[,c(covariate_names, "W_reading_time_high", "Y_utility"
agg_df_time_finish %>% setnames_W_Y()

agg_df_wcount_finish <- freadom_agg[,c(covariate_names, "W_wordcount_high", "Y_utility")
agg_df_wcount_finish %>% setnames_W_Y()

agg_df_time_lag <- freadom_agg[,c(covariate_names, "W_reading_time_high", "Y_next_view_l
agg_df_time_lag %>% setnames_W_Y()

agg_df_wcount_lag <- freadom_agg[,c(covariate_names, "W_wordcount_high", "Y_next_view_la
agg_df_wcount_lag %>% setnames_W_Y()
```

```
agg_df_finish_lag <- freadom_agg[,c(covariate_names, "W_utility_high", "Y_next_view_lag"
agg_df_finish_lag %>% setnames_W_Y()


agg_df_wcount_qtile_wordsread <- freadom_agg[,c(covariate_names, "W_wordcount_qtile", "Y
agg_df_wcount_qtile_wordsread %>% setnames_W_Y()


#### ANALYSIS PER DATASET ####
```

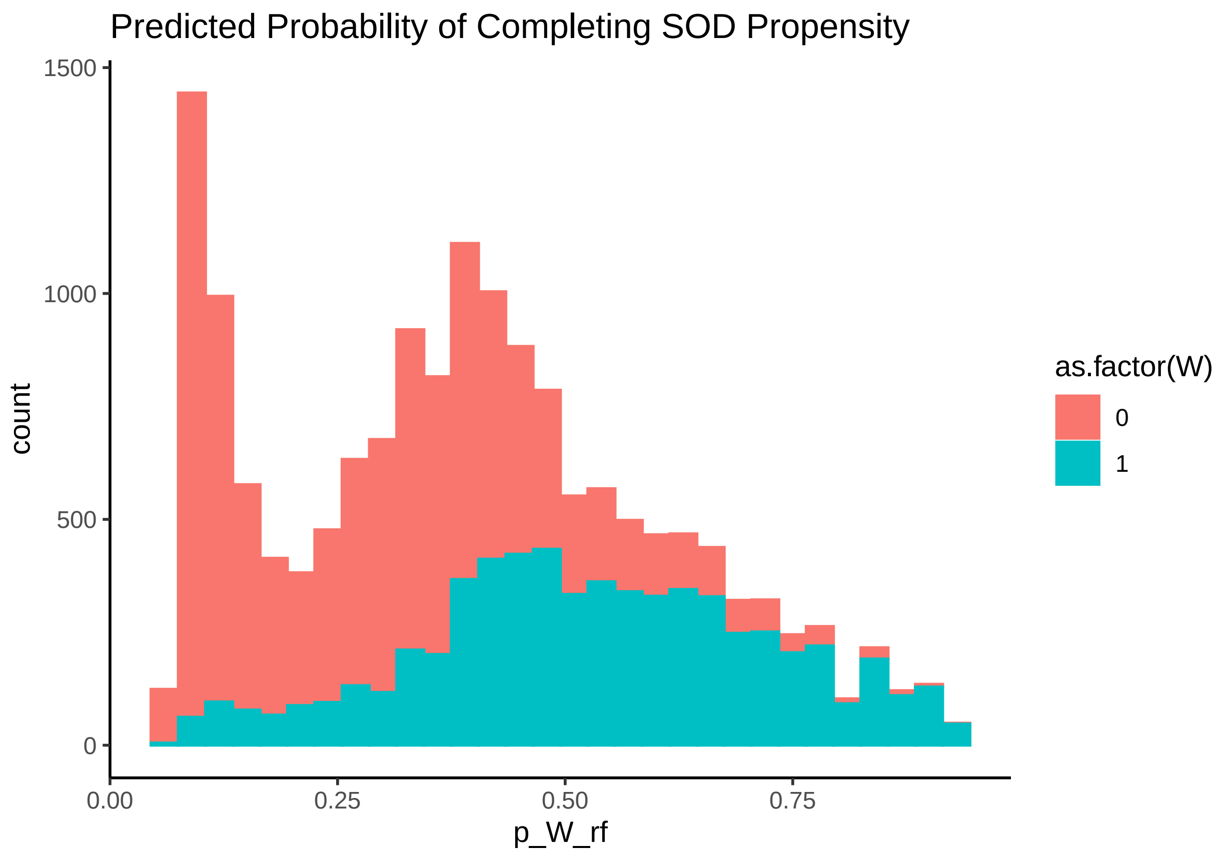
```
# Calculate and Plot Overlap
df_time_finish_overlap <- calculate_propensities(df_time_finish)
df_time_finish_overlap %>% ggplot(aes(x=p_W_rf,color=as.factor(W),fill=as.factor(W)))+
  ggtitle("Above Median SOD Reading Time Propensity")
```



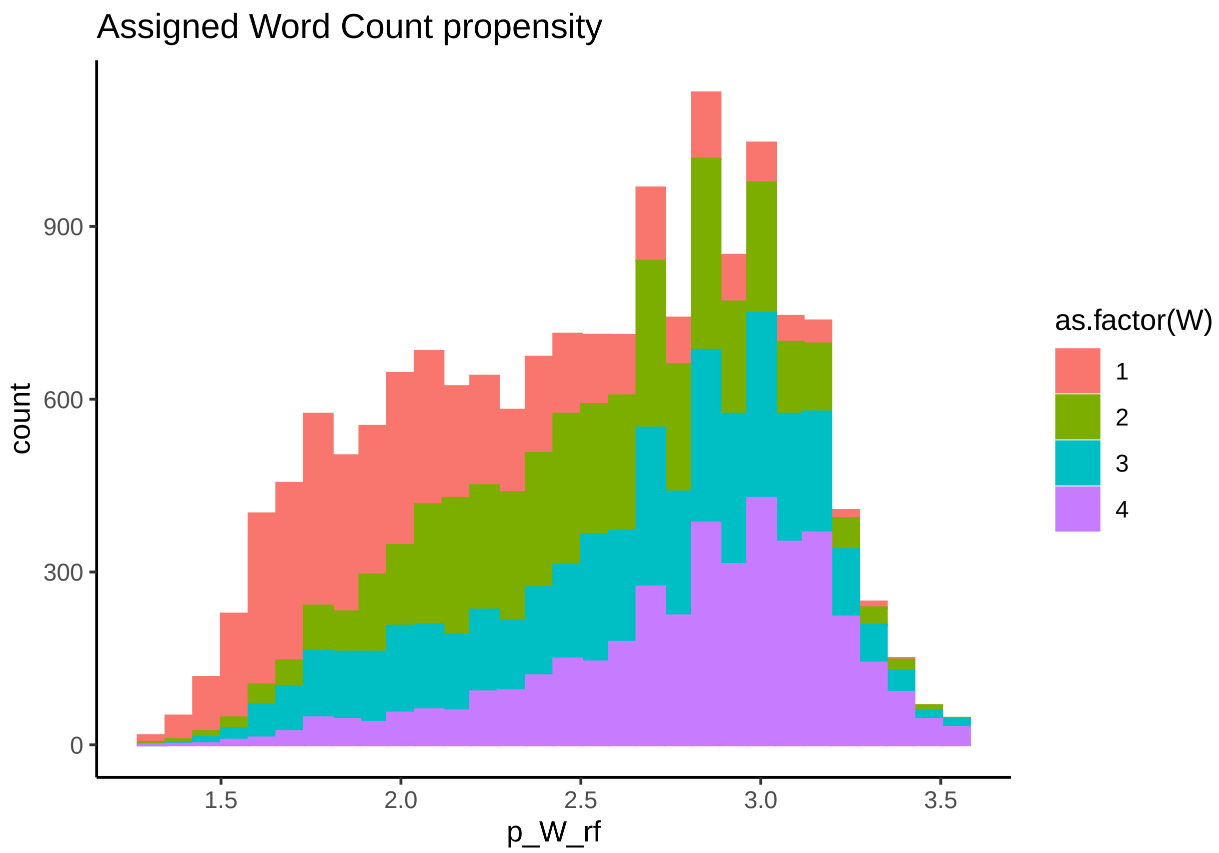
```
df_wcount_finish_overlap <- calculate_propensities(df_wcount_finish)
df_wcount_finish_overlap %>% ggplot(aes(x=p_W_rf,color=as.factor(W),fill=as.factor(W)))
  ggtitle("Above Median SOD Word Count Propensity")
```



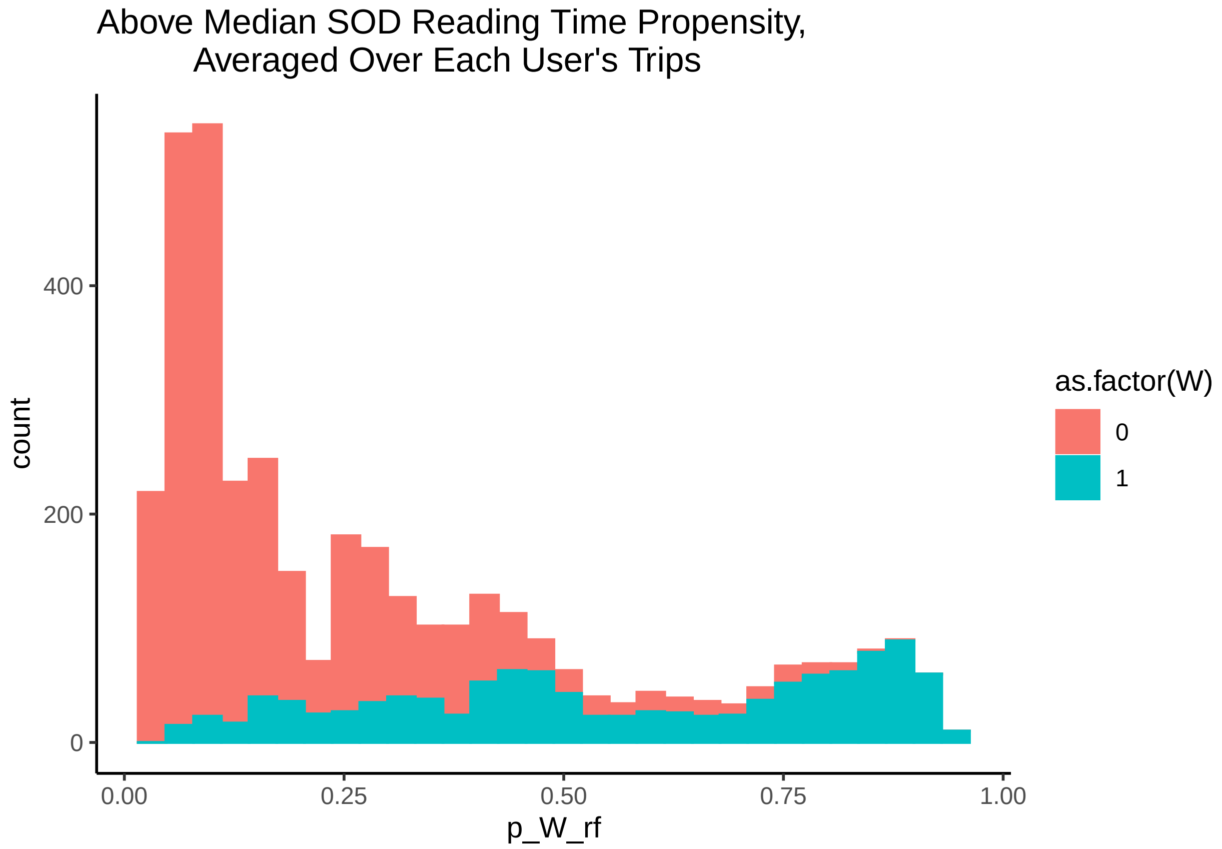
```
df_finish_lag_overlap <- calculate_propensities(df_finish_lag)
df_finish_lag_overlap %>% ggplot(aes(x=p_W_rf,color=as.factor(W),fill=as.factor(W)))+ g
  ggtitle("Predicted Probability of Completing SOD Propensity")
```



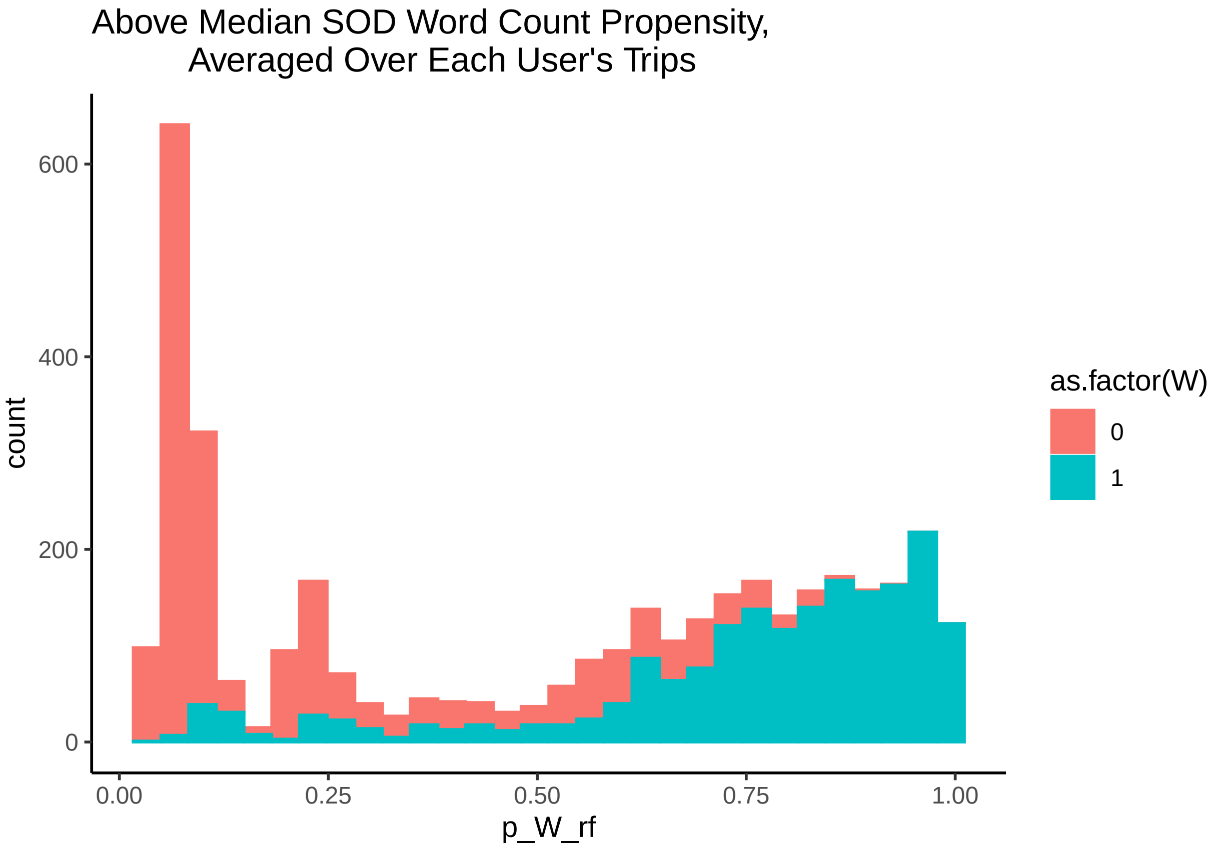
```
df_wcount_qtile_wordsread_overlap <- calculate_propensities(df_wcount_qtile_wordsread)
df_wcount_qtile_wordsread_overlap %>% ggplot(aes(x=p_W_rf,color=as.factor(W),fill=as.fa
  ggtitle("Assigned Word Count propensity")
```



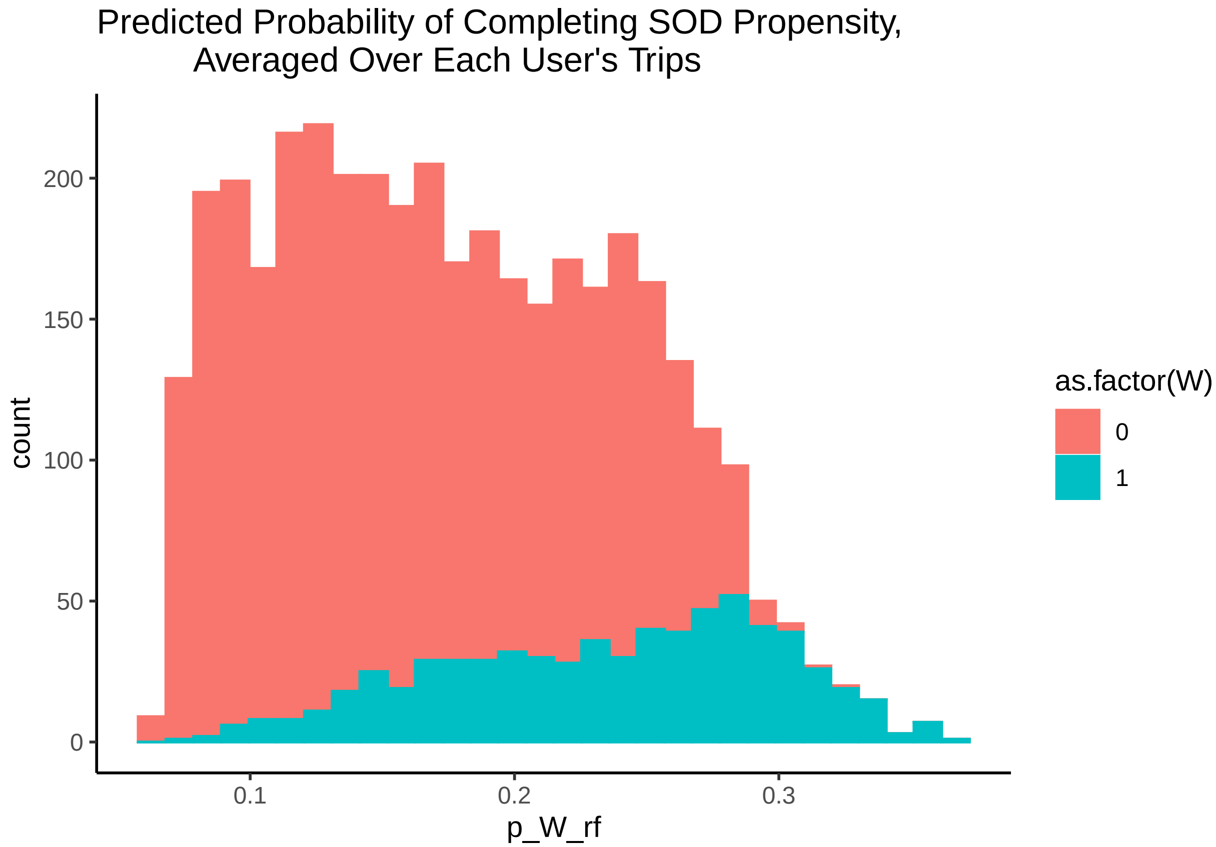
```
# aggregate data


agg_df_time_finish_overlap <- calculate_propensities(agg_df_time_finish)
agg_df_time_finish_overlap %>% ggplot(aes(x=p_W_rf,color=as.factor(W),fill=as.factor(W)
  ggtitle("Above Median SOD Reading Time Propensity,
          Averaged Over Each User’s Trips")
```



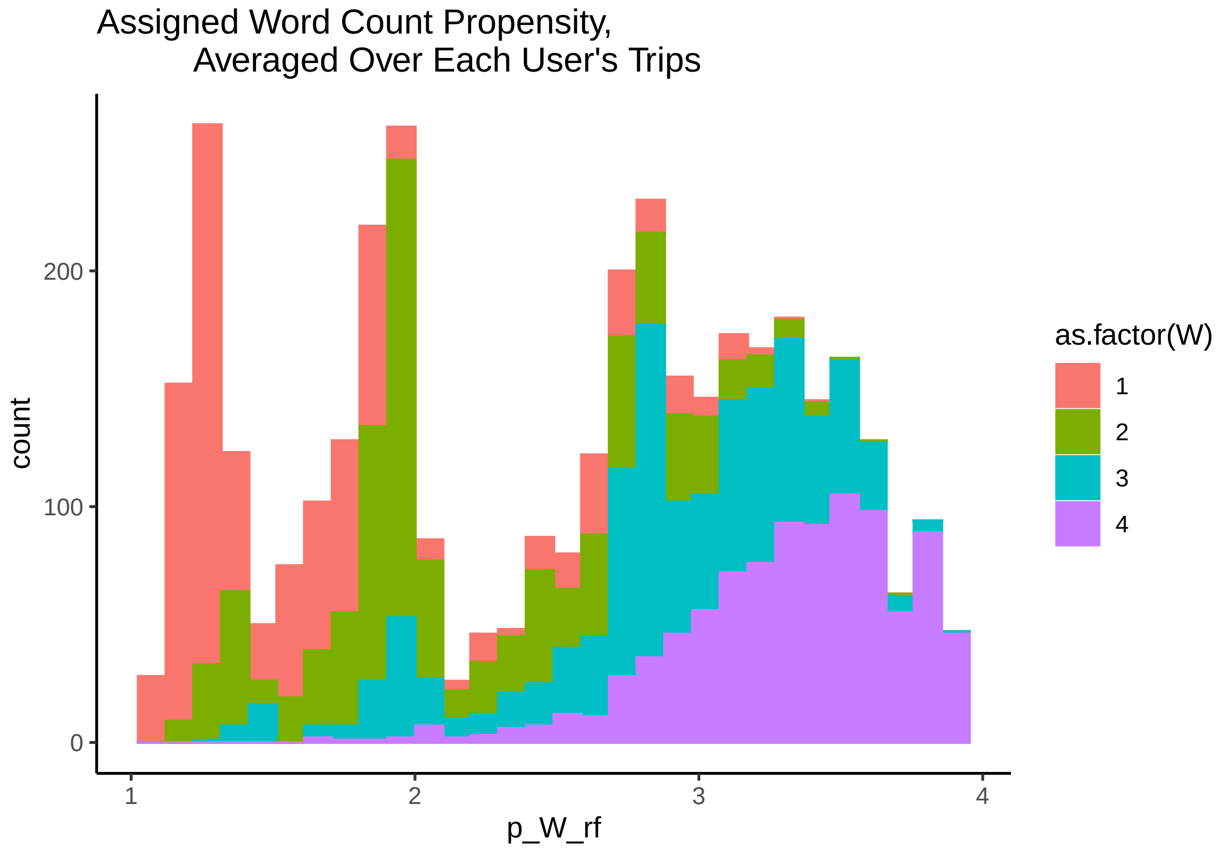
```
agg_df_wcount_finish_overlap <- calculate_propensities(agg_df_wcount_finish)
agg_df_wcount_finish_overlap %>% ggplot(aes(x=p_W_rf,color=as.factor(W),fill=as.factor(
  ggtitle("Above Median SOD Word Count Propensity,
          Averaged Over Each User’s Trips")
```



```
agg_df_finish_lag_overlap <- calculate_propensities(agg_df_finish_lag)
agg_df_finish_lag_overlap %>% ggplot(aes(x=p_W_rf,color=as.factor(W),fill=as.factor(W))
  ggtitle("Predicted Probability of Completing SOD Propensity,
          Averaged Over Each User’s Trips")
```



```
agg_df_wcount_qtile_wordsread_overlap <- calculate_propensities(agg_df_wcount_qtile_word
agg_df_wcount_qtile_wordsread_overlap %>% ggplot(aes(x=p_W_rf,color=as.factor(W),fill=as
  ggtitle("Assigned Word Count Propensity,
          Averaged Over Each User’s Trips")
```



```
# Subset if necessary
df_time_finish_overlap <- df_time_finish_overlap[p_W_rf %between% propensity_bound & p_W
df_wcount_finish_overlap <- df_wcount_finish_overlap[p_W_rf %between% propensity_bound &
df_time_lag_overlap <- df_time_lag_overlap[p_W_rf %between% propensity_bound & p_W %betw
df_wcount_lag_overlap <- df_wcount_lag_overlap[p_W_rf %between% propensity_bound & p_W %
df_finish_lag_overlap <- df_finish_lag_overlap[p_W_rf %between% propensity_bound & p_W %
df_wcount_qtile_wordsread_overlap <- df_wcount_qtile_wordsread_overlap[p_W_rf %between%



agg_df_time_finish_overlap <- agg_df_time_finish_overlap[p_W_rf %between% propensity_bou
agg_df_wcount_finish_overlap <- agg_df_wcount_finish_overlap[p_W_rf %between% propensity
agg_df_time_lag_overlap <- agg_df_time_lag_overlap[p_W_rf %between% propensity_bound & p
agg_df_wcount_lag_overlap <- agg_df_wcount_lag_overlap[p_W_rf %between% propensity_bound
agg_df_finish_lag_overlap <- agg_df_finish_lag_overlap[p_W_rf %between% propensity_bound
```

```
agg_df_wcount_qtile_wordsread_overlap <- agg_df_wcount_qtile_wordsread_overlap[p_W_rf %b
```

# 1 User-Session Level Analysis

## 1.1 Effect of Higher than Average Estimated Reading Time on SOD Completion

```
df_time_finish_models <- run_all_models_on_df(df_time_finish_overlap)
```

[1] 801 [1] 12200

```
round(as.matrix(df_time_finish_models), 3) %>% knitr::kable(format = "latex")
```

| | ATE | lower_ci | upper_ci | ci_length |
|---|---:|---:|---:|---:|
| Difference_in_Means | -0.103 | -0.169 | -0.038 | 0.131 |
| logistic_propensity_weighted_regression | -0.034 | -0.109 | 0.041 | 0.149 |
| IPW_logistic | 0.051 | -0.132 | 0.234 | 0.366 |
| AIPW_linear_plus_logistic | -0.050 | -0.122 | 0.023 | 0.145 |
| IPW_forest | -0.190 | -0.370 | -0.009 | 0.361 |
| AIPW_ate_causal_forest | -0.050 | -0.115 | 0.015 | 0.130 |
| AIPW_linear_plus_forest | -0.050 | -0.116 | 0.017 | 0.133 |

```
#### WORD COUNT FINISH SOD ANALYSIS ####
```

## 1.2 Effect of Higher than Average Estimated Word Count on SOD Completion

```r
df_wcount_finish_models <- run_all_models_on_df(df_wcount_finish_overlap)
```

```
[1] 8002 [1] 8005
```

```r
round(as.matrix(df_wcount_finish_models), 3) %>% knitr::kable(format = "latex")
```

| | ATE | lower_ci | upper_ci | ci_length |
|---|---:|---:|---:|---:|
| Difference_in_Means | -0.087 | -0.115 | -0.059 | 0.056 |
| logistic_propensity_weighted_regression | -0.040 | -0.070 | -0.010 | 0.060 |
| IPW_logistic | -0.010 | -0.081 | 0.062 | 0.143 |
| AIPW_linear_plus_logistic | -0.040 | -0.069 | -0.012 | 0.057 |
| IPW_forest | -0.040 | -0.110 | 0.029 | 0.139 |
| AIPW_ate_causal_forest | -0.029 | -0.058 | 0.000 | 0.058 |
| AIPW_linear_plus_forest | -0.033 | -0.060 | -0.005 | 0.055 |

```r
#### READING TIME LAG SOD ANALYSIS ####
```

## 1.3 Effect of Higher than Average Estimated Reading Time SOD on Time to Next Session

```
df_time_lag_models <- run_all_models_on_df(df_time_lag_overlap)
```

[1] 801 [1] 12186

```
round(as.matrix(df_time_lag_models), 3) %>% knitr::kable(format = "latex")
```

| | ATE | lower_ci | upper_ci | ci_length |
|---|---|---|---|---|
| Difference_in_Means | -0.022 | -0.190 | 0.147 | 0.337 |
| logistic_propensity_weighted_regression | 0.137 | -0.097 | 0.370 | 0.466 |
| IPW_logistic | 0.224 | -0.083 | 0.531 | 0.614 |
| AIPW_linear_plus_logistic | 0.103 | -0.119 | 0.325 | 0.444 |
| IPW_forest | -0.065 | -0.302 | 0.173 | 0.474 |
| AIPW_ate_causal_forest | 0.127 | -0.045 | 0.298 | 0.343 |
| AIPW_linear_plus_forest | 0.077 | -0.091 | 0.245 | 0.337 |

```
#### WORD COUNT LAG SOD ANALYSIS ####
```

## 1.4 Effect of Higher than Average Word Count SOD on Time to Next Session

```
df_wcount_lag_models <- run_all_models_on_df(df_wcount_lag_overlap)
```

[1] 8002 [1] 8005

```
round(as.matrix(df_wcount_lag_models), 3) %>% knitr::kable(format = "latex")
```

| | ATE | lower_ci | upper_ci | ci_length |
|---|---|---|---|---|
| Difference_in_Means | -0.012 | -0.085 | 0.061 | 0.146 |
| logistic_propensity_weighted_regression | 0.100 | 0.019 | 0.182 | 0.162 |
| IPW_logistic | 0.129 | 0.027 | 0.230 | 0.204 |
| AIPW_linear_plus_logistic | 0.094 | 0.015 | 0.172 | 0.157 |
| IPW_forest | 0.082 | -0.009 | 0.174 | 0.183 |
| AIPW_ate_causal_forest | 0.066 | -0.014 | 0.146 | 0.160 |
| AIPW_linear_plus_forest | 0.091 | 0.021 | 0.161 | 0.140 |

```
#### WORD COUNT QUARTILE SOD WORDS READ ANALYSIS ####
```

# 2 Number of Words Read

```
df_wcount_lag_models <- run_all_models_on_df(df_wcount_lag_overlap)
```

```
[1] 8002 [1] 8005
```

```
round(as.matrix(df_wcount_lag_models), 3) %>% knitr::kable(format = "latex")
```

| | ATE | lower_ci | upper_ci | ci_length |
|---|---|---|---|---|
| Difference_in_Means | -0.012 | -0.085 | 0.061 | 0.146 |
| logistic_propensity_weighted_regression | 0.100 | 0.019 | 0.182 | 0.162 |
| IPW_logistic | 0.129 | 0.027 | 0.230 | 0.204 |
| AIPW_linear_plus_logistic | 0.094 | 0.015 | 0.172 | 0.157 |
| IPW_forest | 0.082 | -0.009 | 0.174 | 0.183 |
| AIPW_ate_causal_forest | 0.057 | -0.021 | 0.135 | 0.155 |
| AIPW_linear_plus_forest | 0.091 | 0.021 | 0.161 | 0.140 |

```
#### WORD COUNT SOD LAG NEXT SESSION CATE ANALYSIS ####
```

# 3 Word Count CATE on Time to Next Session

We now estimate the CATE, and use it to construct quartiles of user-sessions. We then report the ATE as estimated with AIPW from our causal forest estimate across quartiles.

```
cf <- forest_from_df(df_wcount_lag_overlap)

oob_pred <- predict(cf, estimate.variance=TRUE)
df_wcount_lag_overlap$cate <- oob_pred$predictions

# Manually creating subgroups
num_tiles <- 4  # ntiles = CATE is above / below the median
df_wcount_lag_overlap$ntile <- factor(ntile(oob_pred$predictions, n=num_tiles))


# ATE estimates
df_wcount_lag_overlap_qtile <- df_wcount_lag_overlap[,.(
  avg_cf_cate = mean(cate)
),.(ntile)][order(ntile)]

df_wcount_lag_overlap_qtile
```

```
estimated_aipw_ate <- lapply(
  seq(num_tiles), function(w) {
    ate <- average_treatment_effect(cf, subset = df_wcount_lag_overlap$ntile == w)
  })
estimated_aipw_ate <- data.frame(do.call(rbind, estimated_aipw_ate))
estimated_aipw_ate$ntile <- factor(seq(num_tiles))
setDT(estimated_aipw_ate)
setnames(estimated_aipw_ate, c("aipw_estimate", "aipw_std.err", "ntile"))
# join CATE estimates together
```

```
df_wcount_lag_overlap_qtile <- df_wcount_lag_overlap_qtile[estimated_aipw_ate, on = .(nt
df_wcount_lag_overlap_qtile %>% knitr::kable(format = "latex")
```

| ntile | avg_cf_cate | aipw_estimate | aipw_std.err |
|---|---:|---:|---:|
| 1 | -0.139 | -0.123 | 0.076 |
| 2 | -0.006 | 0.021 | 0.032 |
| 3 | 0.054 | 0.011 | 0.051 |
| 4 | 0.366 | 0.382 | 0.111 |

# 4 User-level Analysis (Averaged over User-Sessions)

## 4.1 User Average Effect of Higher than Average Estimated Reading Time on SOD Completion

```
agg_df_time_finish_models <- run_all_models_on_df(agg_df_time_finish_overlap)
```

[1] 1114 [1] 2444

```
round(as.matrix(agg_df_time_finish_models), 3) %>% knitr::kable(format = "latex")
```

| | ATE | lower_ci | upper_ci | ci_length |
|---|---|---|---|---|
| Difference_in_Means | -0.129 | -0.176 | -0.082 | 0.095 |
| logistic_propensity_weighted_regression | -0.073 | -0.139 | -0.007 | 0.131 |
| IPW_logistic | -0.097 | -0.194 | 0.000 | 0.193 |
| AIPW_linear_plus_logistic | -0.080 | -0.139 | -0.022 | 0.116 |
| IPW_forest | -0.200 | -0.278 | -0.121 | 0.157 |
| AIPW_ate_causal_forest | -0.101 | -0.180 | -0.022 | 0.158 |
| AIPW_linear_plus_forest | -0.078 | -0.127 | -0.029 | 0.099 |

```
#### WORD COUNT FINISH SOD AGGREGATE ANALYSIS ####
```

## 4.2 User Average Effect of Higher than Average Estimated Word Count on SOD Completion

```
agg_df_wcount_finish_models <- run_all_models_on_df(agg_df_wcount_finish_overlap)
```

[1] 1716 [1] 1575

```
round(as.matrix(agg_df_wcount_finish_models), 3) %>% knitr::kable(format = "latex")
```

| | ATE | lower_ci | upper_ci | ci_length |
|---|---|---|---|---|
| Difference_in_Means | -0.124 | -0.175 | -0.073 | 0.102 |
| logistic_propensity_weighted_regression | -0.112 | -0.216 | -0.008 | 0.207 |
| IPW_logistic | -0.216 | -0.366 | -0.065 | 0.301 |
| AIPW_linear_plus_logistic | -0.088 | -0.182 | 0.007 | 0.189 |
| IPW_forest | 0.009 | -0.077 | 0.095 | 0.172 |
| AIPW_ate_causal_forest | -0.115 | -0.206 | -0.023 | 0.183 |
| AIPW_linear_plus_forest | -0.076 | -0.132 | -0.020 | 0.112 |

```
#### READING TIME LAG SOD AGGREGATE ANALYSIS ####
```

## 4.3 User Average Effect of Higher than Average Estimated Reading Time SOD on Time to Next Session

```r
agg_df_time_lag_models <- run_all_models_on_df(agg_df_time_lag_overlap)
```

```
[1] 1114 [1] 2444
```

```r
round(as.matrix(agg_df_time_lag_models), 3) %>% knitr::kable(format = "latex")
```

| | ATE | lower_ci | upper_ci | ci_length |
|---|---|---|---|---|
| Difference_in_Means | -1.107 | -1.309 | -0.905 | 0.404 |
| logistic_propensity_weighted_regression | -0.552 | -0.807 | -0.296 | 0.511 |
| IPW_logistic | -0.639 | -1.022 | -0.256 | 0.767 |
| AIPW_linear_plus_logistic | -0.558 | -0.787 | -0.328 | 0.458 |
| IPW_forest | -0.772 | -1.126 | -0.417 | 0.709 |
| AIPW_ate_causal_forest | -0.030 | -0.382 | 0.322 | 0.704 |
| AIPW_linear_plus_forest | -0.376 | -0.612 | -0.141 | 0.471 |

```r
#### WORD COUNT LAG SOD AGGREGATE ANALYSIS ####
```

## 4.4 User Average Effect of Higher than Average Word Count SOD on Time to Next Session

```
agg_df_wcount_lag_models <- run_all_models_on_df(agg_df_wcount_lag_overlap)
```

[1] 1710 [1] 1575

```
round(as.matrix(agg_df_wcount_lag_models), 3) %>% knitr::kable(format = "latex")
```

| | ATE | lower_ci | upper_ci | ci_length |
|---|---|---|---|---|
| Difference_in_Means | 0.079 | -0.163 | 0.321 | 0.484 |
| logistic_propensity_weighted_regression | -0.038 | -0.574 | 0.498 | 1.071 |
| IPW_logistic | -0.478 | -1.211 | 0.255 | 1.466 |
| AIPW_linear_plus_logistic | 0.058 | -0.475 | 0.590 | 1.065 |
| IPW_forest | 0.522 | 0.024 | 1.020 | 0.996 |
| AIPW_ate_causal_forest | 0.837 | 0.222 | 1.451 | 1.228 |
| AIPW_linear_plus_forest | 0.173 | -0.187 | 0.534 | 0.721 |

```
#### FINISH SOD LAG NEXT SESSION CATE AGGREGATE ANALYSIS ####
```

We now estimate the CATE, and use it to construct quartiles. We then report the ATE as estimated with AIPW from our causal forest estimate across quartiles.

```
cf <- forest_from_df(agg_df_wcount_lag_overlap)


oob_pred <- predict(cf, estimate.variance=TRUE)
agg_df_wcount_lag_overlap$cate <- oob_pred$predictions


# Manually creating subgroups
num_tiles <- 4  # ntiles = CATE is above / below the median
agg_df_wcount_lag_overlap$ntile <- factor(ntile(oob_pred$predictions, n=num_tiles))
```

```r
# ATE estimates
agg_df_wcount_lag_overlap_qtile <- agg_df_wcount_lag_overlap[,.(
  avg_cf_cate = mean(cate)
),.(ntile)][order(ntile)]

agg_df_wcount_lag_overlap_qtile
```

```r
estimated_aipw_ate <- lapply(
  seq(num_tiles), function(w) {
    ate <- average_treatment_effect(cf, subset = agg_df_wcount_lag_overlap$ntile == w)
  })
estimated_aipw_ate <- data.frame(do.call(rbind, estimated_aipw_ate))
estimated_aipw_ate$ntile <- factor(seq(num_tiles))
setDT(estimated_aipw_ate)
setnames(estimated_aipw_ate, c("aipw_estimate", "aipw_std.err", "ntile"))
# join CATE estimates together
agg_df_wcount_lag_overlap_qtile <- agg_df_wcount_lag_overlap_qtile[estimated_aipw_ate, o
agg_df_wcount_lag_overlap_qtile %>% knitr::kable(format = "latex")
```

| ntile | avg_cf_cate | aipw_estimate | aipw_std.err |
|---|---:|---:|---:|
| 1 | -0.820 | -1.571 | 0.508 |
| 2 | -0.181 | -0.160 | 0.458 |
| 3 | 0.232 | -0.202 | 0.394 |
| 4 | 0.983 | 1.886 | 0.509 |

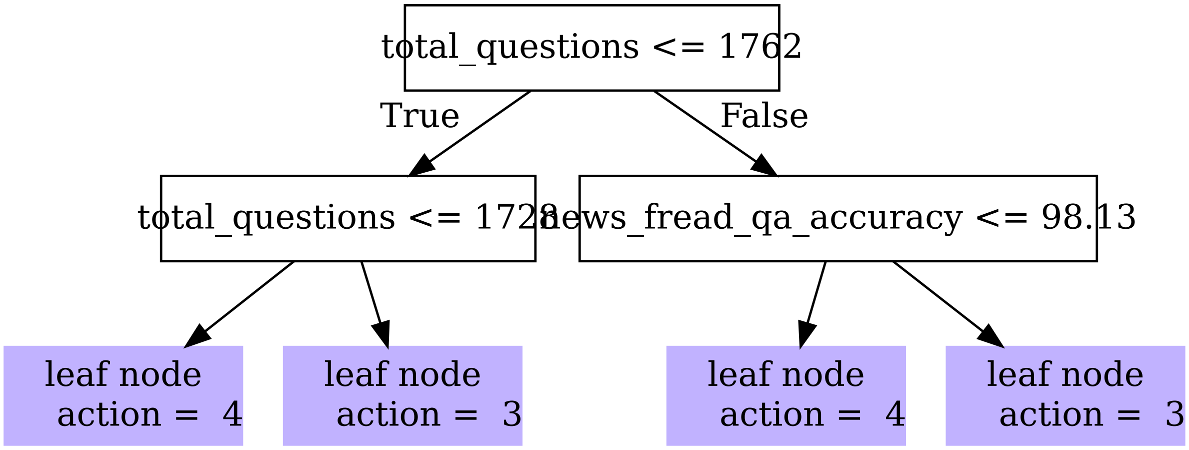
```
#### POLICY TREE ####
```

# 5 Optimal Policy Trees

```
X <- as.matrix(freadom[, c(covariate_names), with=FALSE])
# Y <- as.matrix(freadom[, c("Y_utility"), with=FALSE])
# W <- as.matrix(freadom[, c("W_wordcount_high"), with=FALSE])
# multi.forest <- multi_causal_forest(X = X, Y = Y, W = W)
```

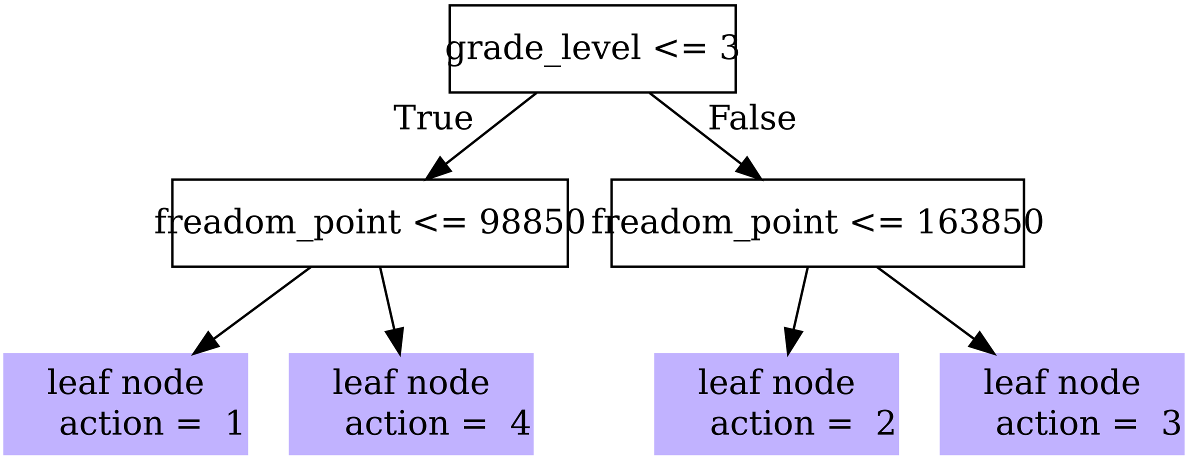
## 5.1 Policy Tree for Effect of SOD Word Count Quartile on Words Read per Session

```
multi.forest <- forest_from_df(freadom, multi_causal_forest, "W_wordcount_qtile", "Y_wor
Gamma.matrix <- double_robust_scores(multi.forest)

opt.tree <- policy_tree(X, Gamma.matrix, depth = 2)
plot(opt.tree, tweak = treeplot_font_size_mult, title = "Total Words Read")
```



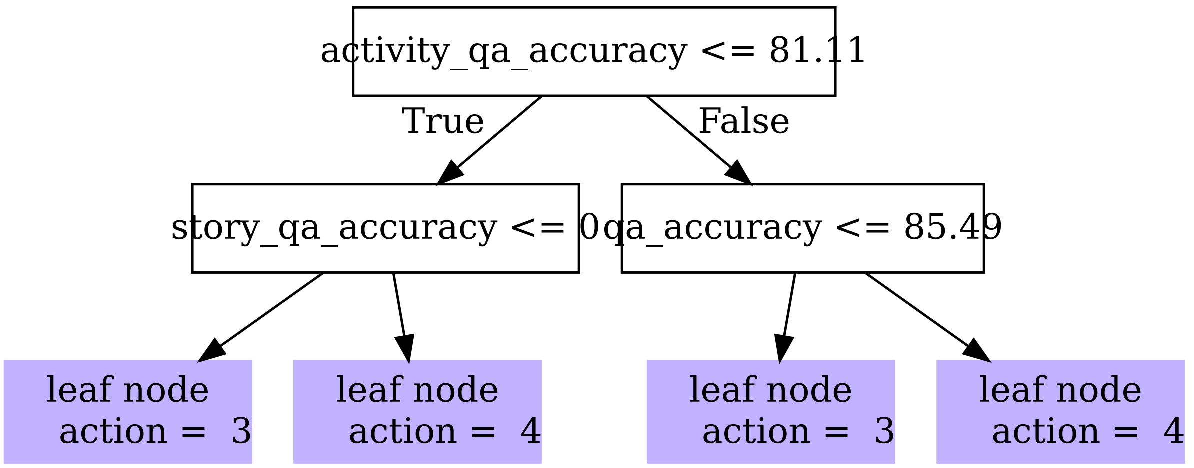
### 5.2 Policy Tree for Effect of SOD Word Count Quartile on SOD Completion

```
multi.forest <- forest_from_df(freadom, multi_causal_forest, "W_wordcount_qtile", "Y_uti
Gamma.matrix <- double_robust_scores(multi.forest)

opt.tree <- policy_tree(X, Gamma.matrix, depth = 2)
plot(opt.tree, tweak = treeplot_font_size_mult, title = "Total Words Read")
```



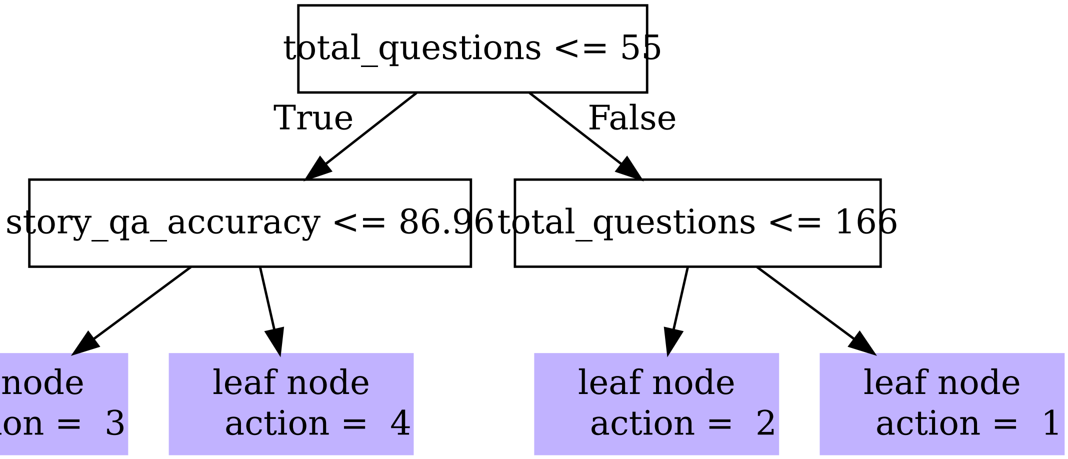
### 5.3 Policy Tree for User-Level Average Effect of SOD Word Count Quartile on Words Read per Session

```
X <- as.matrix(freadom_agg[, c(covariate_names), with=FALSE])
multi.forest <- forest_from_df(freadom_agg, multi_causal_forest, "W_wordcount_qtile", "Y
Gamma.matrix <- double_robust_scores(multi.forest)

opt.tree <- policy_tree(X, Gamma.matrix, depth = 2)
plot(opt.tree, tweak = treeplot_font_size_mult)
```



### 5.4 Policy Tree for User-Level Average Effect of SOD Word Count Quartile on SOD Completion

```r
X <- as.matrix(freadom_agg[, c(covariate_names), with=FALSE])
multi.forest <- forest_from_df(freadom_agg, multi_causal_forest, "W_wordcount_qtile", "Y
Gamma.matrix <- double_robust_scores(multi.forest)

opt.tree <- policy_tree(X, Gamma.matrix, depth = 2)
plot(opt.tree, tweak = treeplot_font_size_mult)
```


total_questions <= 55
True
False
story_qa_accuracy <= 86.96
total_questions <= 166
leaf node
action = 3
leaf node
action = 4
leaf node
action = 2
leaf node
action = 1

```
#### WRITEUP ####
```

# 6 Writeup Introduction

Improvements to childhood literacy have been linked to numerous positive outcomes, including economic and social benefits. In this paper, we use data from a mobile application aimed at improving childhood reading outcomes. The app is primarily used by schoolgoing children from junior kindergarten until grade 3 to read and interact with stories.

This work takes advantage of the application's "Story of the Day" (hereafter SOD) feature. Stories of the Day are featured prominently on the app, and users read the SOD on approximately 33% of days they use the app. Our analysis focuses entirely on these stories. Several SODs are available to be assigned to users on each day, and vary primarily by estimated reading time and word count. The assignment of story of the day is generic and not personalised, and different stories are shown each day. As a result, if we consider a user opening the app as an exogenous random decision on a given day, since the stories shown to students are different each day, this gives us exogenous treatments for length of stories shown to students in terms of reading time and number of words in story. We measure the effect of this treatment on reading outcomes. Our identifying motivation is that longer stories reduce the probability of a child reading a story.

Even if this effect is true on average, this does not mean that longer stories have negative effects on all users. As such, we examine CATEs across a variety of groups in Section 5.

Finally, in Section 6 we attempt to maximize aggregate reading time by identifying the optimal policy, and summarize possible gains from targeted assignment of Stories of the Day by length.

# 7 Data Description

## 7.1 User Covariates:

In our dataset, we have a bunch of covariates describing users. These include age, grade, statistics about usage such as books read, experience points gained on the app while using it, total time spent on the app, covariates about how well a child answered questions related to their readings, and their reading interests.

## 7.2 Treatment Definitions

We use the following treatments for analysis:

- Suggested Reading Time for a given story: The app includes suggested reading times for a story in one of five choices. We examine only stories where the estimated reading time was either 7.5 or 12.5 minutes, as estimated reading time is not continuous. These two values of estimated reading time account for 95% of all user-trips.

- Number of words for a given story: We parsed the stories shown on the app to get the number of words in each story, and we use this as a treatment variable. We divide observations into those where the number of words is below and above the median, giving us a treatment and a control. We also do another analysis with multiple treatments where the treatment is characterised by the quantile in which its number of words falls; giving us 4 treatment conditions.

We recall that as the assignment of the Story of the Day is at-random, so is the assignment of word count and estimated reading time. Note: We examine only users-trips where the user started reading the Story of the Day, as otherwise the user would have no estimate of the length or time required to read the story.

## 7.3 Outcomes

We examine three outcomes in our analysis: whether the user finished reading their assigned Story of the Day, the length of time until their next session, and the estimated number of words they read. In addition to conducting our analyses at the user-session level, we also estimate user-level aggregate treatment effects. For each user we aggregate over all sessions, averaging session-level binary treatment status and outcome.

# 8 Average Treatment Effect

First, we measure the average treatment effect of our various treatments on our various outcomes. For this, as we learnt in the class, we use various methods.

## 8.1 Models

For each set of models, we compare results from the following methods:
* Simple Difference in Means
* Logistic Propensity Weighted Linear Regression
* IPW Estimator using Logistic Propensities
* IPW Estimator using Regression Forest Propensities
* AIPW Estimator using Logistic Propensities and Linear Regression
* AIPW Estimator using Causal Forests for both outcome and propensity models

## 8.2 Results for each Treatment

### 8.2.1 Suggested Reading Time

At the user-session level, estimated reading time has no significant effect on a user's likelihood of completing their story of the day; confidence intervals include zero in almost all models in Table 1.1. On the other hand, Table 4.1 shows the estimated reading time has a significant and negative effect on the probability of a user completing the SOD at the user level: when assigned readings that take longer, users are less likely to complete an average SOD. This is true and significant across all model specifications.

We see that model results are similar with some variation in confidence intervals. The largest treatment effect in magnitude is reported by the IPW estimator with logistic propensities. The AIPW estimator with causal forests propensity model and a linear regression outcome model reports a more conservative treatment effect and also reports the minimum confidence interval length, while the difference in means estimator is a close second. However, we believe that the former is more reliable because it tries to control for confounding. We note that results across models are similar across treatments and outcomes. At the user-session level in Table 1.3, we do not see significant effects on time until next SOD started under any model.

At the user-level in Table 4.3, our causal forest model has confidence intervals that include zero, while all other models indicate a negative effect of higher average reading time on time until the next SOD attempt. Otherwise, the results are analogous to the effect on SOD completion, with AIPW estimators with Causal Forest Propensities giving the smallest Confidence Intervals.

### 8.2.2 Word Count

At the user-session level in Table 1.2, we see that longer SOD by word-count had a slightly negative (but statistically significant) effect on finishing SOD. This is the case for the difference in means estimate of the treatment effect as well as all AIPW methods, which are generally more reliable than the (conflicting evidence presented by) the IPW methods. This matches our initial hypothesis.
In Table 1.4 we see some evidence for the effect of SOD word count on time until next login: while the causal forest confidence intervals include zero, all other AIPW models are significant and show that longer passages by word-length increases the average time until the next session.

At the user-level in Table 4.2, users who were on average assigned longer readings by word count are generally less likely to finish their SOD. This is true of all AIPW methods other than the AIPW estimated with linear propensity scores. Table 4.4 shows that there is conflicting evidence for the effect of longer word-count SODs on time until the next SOD view: the causal forest shows a significant and positive relationship between the two, while almost all other methods have confidence intervals that include zero.

# 9 Conditional Average Treatment Effect

We now examine the CATE of word count on time until the next SOD attempt. These results are detailed in Table 5.1.

We experimented with a number of models for estimating CATEs, but only present those CATE estimates which were done using causal forests. We first use causal forests to estimate the CATE, and use our CATE estimates to construct quartiles of user-sessions. We then report the ATE as estimated with AIPW from our causal forest estimate across quartiles. We find significant effects in the first and fourth quartiles in opposite directions: for those users in the bottom quartile of estimated CATE, we find that an increase in word count has a negative and statistically significant effect on time until next SOD attempt; for the highest quartile we observe a significant, positive relationship between these two quantities. If our goal is to increase total amount read, this indicates that we may be able to do so by targeting users with SODs of different length.

# 10 Optimal Policy

In this section, we divide story word counts into 4 quartiles, leading to 4 different treatments. We use multi treatment causal forests to estimate treatment effects for each of these. We consider two outcomes; SOD completion and Number of Words Read by a user. We calculate treatment effects for these outcomes at both the user level as well as the user session level. As we learnt in class, our goal is to learn optimal policy assignments for students in this section. Here, a policy would denote a decision to show a user a story of a certain length.

## 10.1 Model

We use the Policy Tree model. This model does a greedy search over the covariate space, in the style of a regression forest, but exhausting all possible split points (with some approximation if desired). We show depth 2 policy trees learnt for policy allocation with this model.

## 10.2 Results

We note that the Policy Tree model tells us that for users with more experience, we should adopt the policy of showing them stories which are longer to maximize their reading utilities (section 5.2); the second level split shows this. This is an interesting finding which shows that more experienced users mature over time and prefer to read lengthier stories.

For number of words read as outcome, the model tells us that we should prefer to show lengthier stories to all users (quantile 3 and 4 in section 5.1) in general

The results at the user level are more mixed. As in section 5.3, the model suggests showing longer stories to users with higher qa accuracy to increase number of words read. Probably the most interesting plot is in section 5.4. Here, we note that the policy prescribed for students with less than 55 questions is quartile 3 for students with lower qa accuracy, and quartile 4 for the rest. However, for students who have answered an unusually high number of questions, the policy prescribed is to show shorter utility stories to maximize SOD completion. We hypothesize that this is possible because this class represents unusually enthusiastic students who wish to maximize number of stories read/questions completed, and the easiest way for them to do this is to read shorter stories. This suggests corrective plans for such strategies which may not be pedagogically the most beneficial.

# 11 Conclusion

In this paper, we analyzed the impact of exposing children to stories of varied lengths, and testing effects of this treatment on various reading outcomes. We note that there is, as we hypothesized, an increased completion of stories when the length of the stories is shorter. This is not true, however, in bringing back users to the app more often. At the same time, we also find that this increase is not true ubiquitously for all users. As we see with the CATE estimates, there are subpopulations for whom the treatment effect may work in the opposite direction. Further, as the Policy Tree analysis shows, we may want to particularly target different subpopulations in different ways; to some beginners we want to show stories which are shorter, so that they complete these easier stories and ride their wave of accomplishment to read more of these. For more advanced users, it is more beneficial to expose them to longer, more complex reads, to maximize reading. We utilized multiple methods for ATE, CATE and Optimal Policy Estimation, which we will return to in future work.