ARE 213 PS 1b

S. Sung, H. Husain, T. Woolley, A. Watt

2021-10-18

Contents

Packages (omitted)

Data cleaning from PS 1a

```
data = read.dta('ps1.dta')
missing_codes = read.csv('missing_codes.csv')
mvars = as.character(missing_codes$varname)
missing_codes$num_missing = as.integer(0)
```

```
for (row in 1:nrow(missing_codes)) {
  var = as.character(missing_codes[row, "varname"])
  code  = as.numeric(missing_codes[row, "missing_code"])
  nmissing = as.integer(sum(data[, var] == code))
  missing_codes$num_missing[missing_codes$varname==var] = nmissing
  data[, var] = na_if(data[, var], code)
}
# Convert all variables with <7 unique values to factor (and 3 additional variables)
factor_vars = c("isllb10", "birmon", "weekday")
for (var in colnames(data)) {
  if (length(unique(data[!is.na(data[, var]), var])) < 7 || var %in% factor_vars) {
    data[, var] = factor(data[, var])
  }
}
# label data
variable_labels_df = read.csv('variable_labels.csv')
variable_labels <- setNames(as.character(variable_labels_df$label), variable_labels_df$varname)
data <- Hmisc::upData(data, labels = variable_labels)

# Dataframe with missing dropped

df = data[complete.cases(data), ] %>%
  sample_n(10000)
# Treatment reference level
df$tobacco <- relevel(df$tobacco, ref = "2")  # reference level: 2 = no for tobacco use during pregnanc
```

Problem 1

In Problem Set 1a, you used linear regression to relate infant health outcomes and maternal smoking during pregnancy.

Part (a)

Under the assumption of random assignment conditional on the observables, what are the sources of misspecification bias in the estimates generated by the linear model estimated in Problem Set 1a?

In problem set 1(a), we input observable characteristic in a linear fashion into our crude estimation with OLS. However, it is possible that CEF isn't linear in parameters, leading to bias. Also, even if we have a linear CEF, it's possible we didn't include important interaction terms.

In order to assume linearity, we need (1) joint normality with observables, indicator for maternal smoking, and our outcome variable of interest or (2) saturated model. In their absence, we could have used other non-linear estimation methods discussed in class.

Part (b)

Now, consider a series estimator. Estimate the smoking effects using a flexible functional form for the control variables (e.g., higher order terms and interactions). What are the benefits and drawbacks to this approach?

```
df1b = df %>% select(dbrwt, fmaps, omaps, tobacco,
                     csex, mrace3, preterm, dmage, dfage, dmeduc, dfeduc,
                     ormoth, orfath, disllb, dtotord, dmar, adequacy, nprevist)
```

```r
# indicator vars (no higher order terms)
vars1 = names(Filter(is.factor, select(df1b, -c(dbrwt, fmaps, omaps))))
# quantitative vars (need to create higher order terms)
vars2 = names(Filter(is.integer, select(df1b, -c(dbrwt, fmaps, omaps))))

birthweight = df1b$dbrwt
fiveminapgar = df1b$fmaps
oneminapgar = df1b$omaps

x = df1b %>% select(-c(dbrwt, fmaps, omaps))
# Create dummies from factor variables, all interactions, and squared continuous vars
formula1 = as.formula(paste("~ .^2 +", paste0("I(",vars2,"^2)", collapse=' + ') ))
xx <- model.matrix(formula1, x)[, -1]

# Series Regression
reg_1b_dbrwt = lm(birthweight ~ xx)
reg_1b_fmaps = lm(fiveminapgar ~ xx)
reg_1b_omaps = lm(oneminapgar ~ xx)

# rename the coefficients
names(reg_1b_dbrwt$coefficients) <- gsub("xx","",names(reg_1b_dbrwt$coefficients))
names(reg_1b_fmaps$coefficients) <- gsub("xx","",names(reg_1b_fmaps$coefficients))
names(reg_1b_omaps$coefficients) <- gsub("xx","",names(reg_1b_omaps$coefficients))
results_df = data.frame(type = 'Series Regression',
                        ate = reg_1b_dbrwt$coefficients['tobacco1'],
                        att = NA,
                        number_vars = ncol(xx)+1)
```

```r
stargazer(reg_1b_dbrwt, reg_1b_fmaps, reg_1b_omaps,
          type = 'latex',
          keep = "tobacco1(?!:)",
          perl=TRUE, header = FALSE, table.placement = '!h',
          title = "Series Regressions",
          align = TRUE, no.space = TRUE, font.size = "small",
          notes = c("For brevity, only the original variables are presented;",
                    "300+ second order / interaction terms are not shown."))
```

Table 1: Series Regressions

	Dependent variable:		
	birthweight	fiveminapgar	oneminapgar
	(1)	(2)	(3)
tobacco1	−251.113	−0.241	−0.045
	(209.763)	(0.269)	(0.482)
Observations	10,000	10,000	10,000
R^2	0.168	0.060	0.050
Adjusted R^2	0.145	0.034	0.024
Residual Std. Error (df = 9728)	540.467	0.692	1.243
F Statistic (df = 271; 9728)	7.233***	2.307***	1.904***

Note: *p<0.1; **p<0.05; ***p<0.01
For brevity, only the original variables are presented;
300+ second order / interaction terms are not shown.

Part (c)

Use the LASSO to determine which covariates (and higher order terms) to include in your regression from part (b). Do you end up dropping some covariates that you had thought might be necessary to include?

For brevity, I continue on with the exercise with only one dependent variable (dbrwt), skipping the others (fmaps and omaps). Running a simple lasso with all the included varibles from 1(b), and arbitrarily setting number of variables to be included as 20, we chose lambda = 14.23. This is the lowest lambda iteration that results in a model with 20 non-zero coefficients.

Under this setting, predicted impact of smoking on birthweight is -76.9.

We ended up dropping `dmage` and 'dfage', or age of parents, which we expected to have some correlation with infant's general health and/or weight. On the other hand, variables such as 'csex' and 'preterm', which we expected to have a stronger relationship with infant weight (i.e. male infant might be bigger on average, and previous baby's birth weight might be predictive of the baby of interest), are not dropped as expected.

One problem, however, with what we have done below is that while an interaction term that includes 'dmage' is included, the standalone variable 'dmage' is dropped, which might be undesirable.

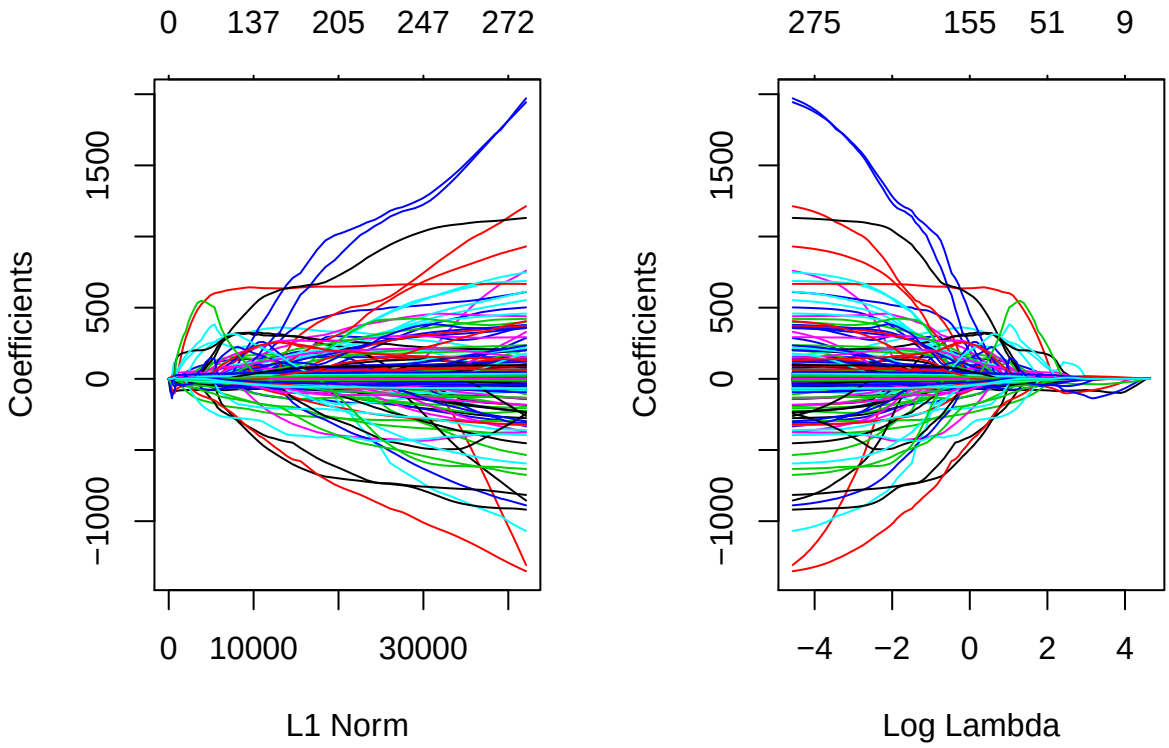
```
# use glmnet with alpha=1 for lasso
reg_1c = glmnet(xx, birthweight, family="gaussian", alpha=1)
# print results (Df = # of variables, %Dev = R^2)
print(reg_1c)
```

```
##
## Call:  glmnet(x = xx, y = birthweight, family = "gaussian", alpha = 1)
##
##       Df  %Dev  Lambda
## 1     0  0.00 104.200
## 2     2  0.56  94.940
## 3     4  1.34  86.510
## 4     5  2.29  78.820
## 5     5  3.11  71.820
## 6     7  3.85  65.440
## 7     9  4.70  59.630
## 8     9  5.58  54.330
## 9    10  6.32  49.500
## 10    9  6.93  45.110
## 11    9  7.43  41.100
## 12   10  7.85  37.450
## 13   11  8.20  34.120
## 14   10  8.50  31.090
## 15   12  8.76  28.330
## 16   13  9.01  25.810
## 17   17  9.27  23.520
## 18   18  9.52  21.430
## 19   20  9.76  19.530
## 20   23 10.01  17.790
## 21   27 10.24  16.210
## 22   27 10.43  14.770
## 23   29 10.62  13.460
## 24   32 10.79  12.260
## 25   36 10.94  11.170
## 26   41 11.29  10.180
## 27   45 11.66   9.276
```

```
## 28   47 12.01   8.452
## 29   50 12.33   7.701
## 30   51 12.60   7.017
## 31   53 12.84   6.394
## 32   57 13.04   5.826
## 33   59 13.23   5.308
## 34   62 13.39   4.837
## 35   67 13.52   4.407
## 36   69 13.64   4.015
## 37   77 13.75   3.659
## 38   84 13.86   3.334
## 39   87 13.98   3.037
## 40   94 14.10   2.768
## 41  104 14.21   2.522
## 42  115 14.38   2.298
## 43  119 14.54   2.094
## 44  125 14.70   1.908
## 45  131 14.86   1.738
## 46  130 14.99   1.584
## 47  134 15.08   1.443
## 48  137 15.20   1.315
## 49  143 15.27   1.198
## 50  147 15.37   1.092
## 51  155 15.43   0.995
## 52  156 15.51   0.906
## 53  166 15.56   0.826
## 54  170 15.63   0.752
## 55  178 15.70   0.686
## 56  185 15.77   0.625
## 57  188 15.84   0.569
## 58  190 15.91   0.519
## 59  198 15.96   0.472
## 60  200 16.01   0.431
## 61  205 16.06   0.392
## 62  207 16.10   0.358
## 63  212 16.15   0.326
## 64  218 16.20   0.297
## 65  226 16.25   0.270
## 66  227 16.29   0.246
## 67  232 16.34   0.224
## 68  232 16.37   0.204
## 69  234 16.40   0.186
## 70  237 16.42   0.170
## 71  243 16.44   0.155
## 72  242 16.47   0.141
## 73  247 16.50   0.128
## 74  252 16.52   0.117
## 75  252 16.54   0.107
## 76  254 16.55   0.097
## 77  256 16.57   0.089
## 78  257 16.58   0.081
## 79  259 16.60   0.074
## 80  260 16.61   0.067
## 81  263 16.61   0.061
```

```
## 82  263 16.62   0.056
## 83  265 16.63   0.051
## 84  266 16.63   0.046
## 85  266 16.64   0.042
## 86  266 16.64   0.038
## 87  267 16.65   0.035
## 88  267 16.65   0.032
## 89  269 16.65   0.029
## 90  271 16.66   0.026
## 91  272 16.66   0.024
## 92  274 16.66   0.022
## 93  275 16.67   0.020
## 94  275 16.67   0.018
## 95  275 16.67   0.017
## 96  276 16.67   0.015
## 97  277 16.67   0.014
## 98  277 16.68   0.013
## 99  280 16.68   0.011
## 100 280 16.68   0.010
```

```r
# Limit the model to num_vars number of variables
num_vars = 20
# choose lowest lambda iteration that results in num_vars non-zero variables
i = max(which(abs(reg_1c$df - num_vars) == min(abs(reg_1c$df - num_vars))))
lambda = reg_1c$lambda[[i]]

# Plot what the coefficients are doing as we increase lambda
op <- par(mfrow=c(1,2))
plot(reg_1c, "norm", label=TRUE)
plot(reg_1c, "lambda", label=TRUE)
```




```
par(op)
```

In both graphs, each curve corresponds to a variable. They show the path of its coefficient against (in the left plot) the L1-norm of the whole coefficient vector as lambda varies and (in the right plot) the values of log-lamba. The top axis indicates the number of nonzero coefficients at the current lambda, which is the effective degrees of freedom (df) for the lasso. We see clearly that as lambda increases more coefficients goes to zero, as intended in model selection.

Here are the non-zero coefficients:

```
knitr::kable(reg_1c$beta[, i][reg_1c$beta[, i] != 0],
             caption=paste0("Lasso Regression for top ", num_vars, " vars (lambda = ", lambda, ")"),
             col.names = 'Non-zero Coefficients', align = "l", digits = 3)
```

Table 2: Lasso Regression for top 20 vars (lambda = 19.5252088080363)

	Non-zero Coefficients
tobacco1	-78.941
csex2	-87.593
mrace33	-117.832
preterm2	13.830
dmar2	-36.690
I(dislb^2)	0.000
tobacco1:csex2	-6.867
tobacco1:dfage	-2.666
tobacco1:dtotord	-8.152

	Non-zero Coefficients
mrace33:dmage	-0.509
mrace33:dfeduc	-1.837
mrace32:nprevist	-2.126
preterm2:dfeduc	1.876
preterm2:nprevist	18.916
dfage:disllb	0.000
dmeduc:ormoth2	-4.375
dfeduc:nprevist	0.059
ormoth2:dmar2	-4.013
orfath2:dmar2	-28.105
dtotord:adequacy2	6.147

Problem 2

Describe the propensity score approach to the problem of estimating the average causal effect of smoking when the treatment is randomly assigned conditional on the observables. How does it reduce the dimensionality problem of multivariate matching? Try a few ways to estimate the effects of maternal smoking on birthweight:

Part (a)

First create the propensity score. For our purposes let's use a logit specification. First specify the logit using all of the "predetermined" covariates (don't include interactions). Next, include only those "predetermined" covariates that enter significantly in the first logit specification. How comparable are the propensity scores? If they are similar does this imply that we have the "correct" set of covariates in the logit specification used for our propensity score?

```
# create the propensity score using logit
# using all of the "predetermined" covariates
df2a = df %>% select(tobacco, csex, mrace3, preterm,
                     dmage, dfage, dmeduc, dfeduc, ormoth, orfath,
                     disllb, dtotord, dmar, adequacy, nprevist)
reg_2a1 <- glm(tobacco ~ ., data = df2a, family = "binomial")
# Try logit with only the significant covariates
df2a2 = df2a %>% select(-csex, -adequacy)
reg_2a2 <- glm(tobacco ~ ., data = df2a2, family = "binomial")
```

Model (1) below is with all the predetermined variables and model (2) is removing the insignificant variables.

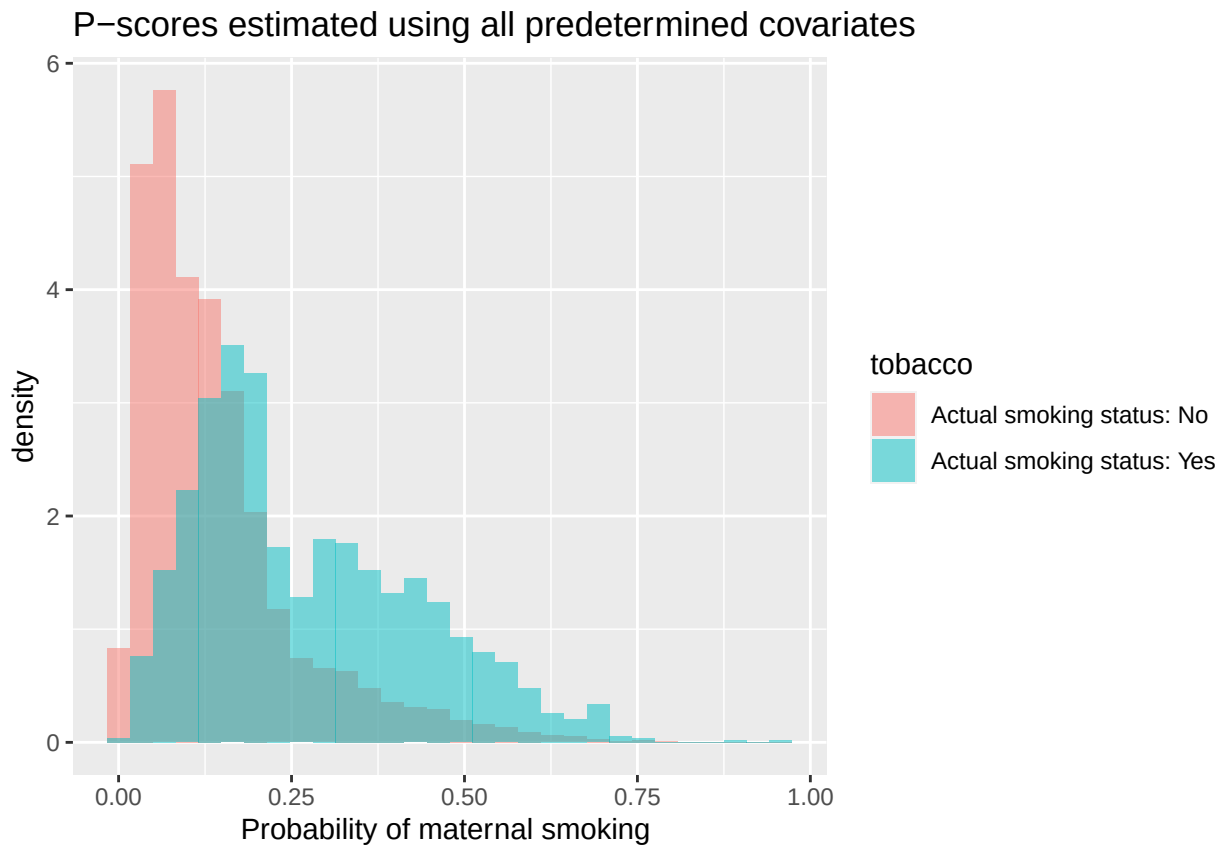
```
stargazer(reg_2a1, reg_2a2, title="Logit regressions", header=FALSE, single.row=TRUE, type=table_type)
```

```
labs <- paste("Actual smoking status:", c("Yes", "No"))
p1_df <- data.frame(p1_score = predict(reg_2a1, type = "response"), tobacco = reg_2a1$model$tobacco) %>%
  mutate(tobacco = ifelse(tobacco == 1, labs[1], labs[2]))
p2_df <- data.frame(p2_score = predict(reg_2a2, type = "response"), tobacco = reg_2a2$model$tobacco) %>%
  mutate(tobacco = ifelse(tobacco == 1, labs[1], labs[2]))

ggplot(p1_df, aes(x=p1_score, fill = tobacco)) +
  geom_histogram(position = "identity", alpha = 0.5,
                 mapping = aes(y = stat(density))) +
  xlab("Probability of maternal smoking ") +
  ggtitle("P-scores estimated using all predetermined covariates")
```

Table 3: Logit regressions

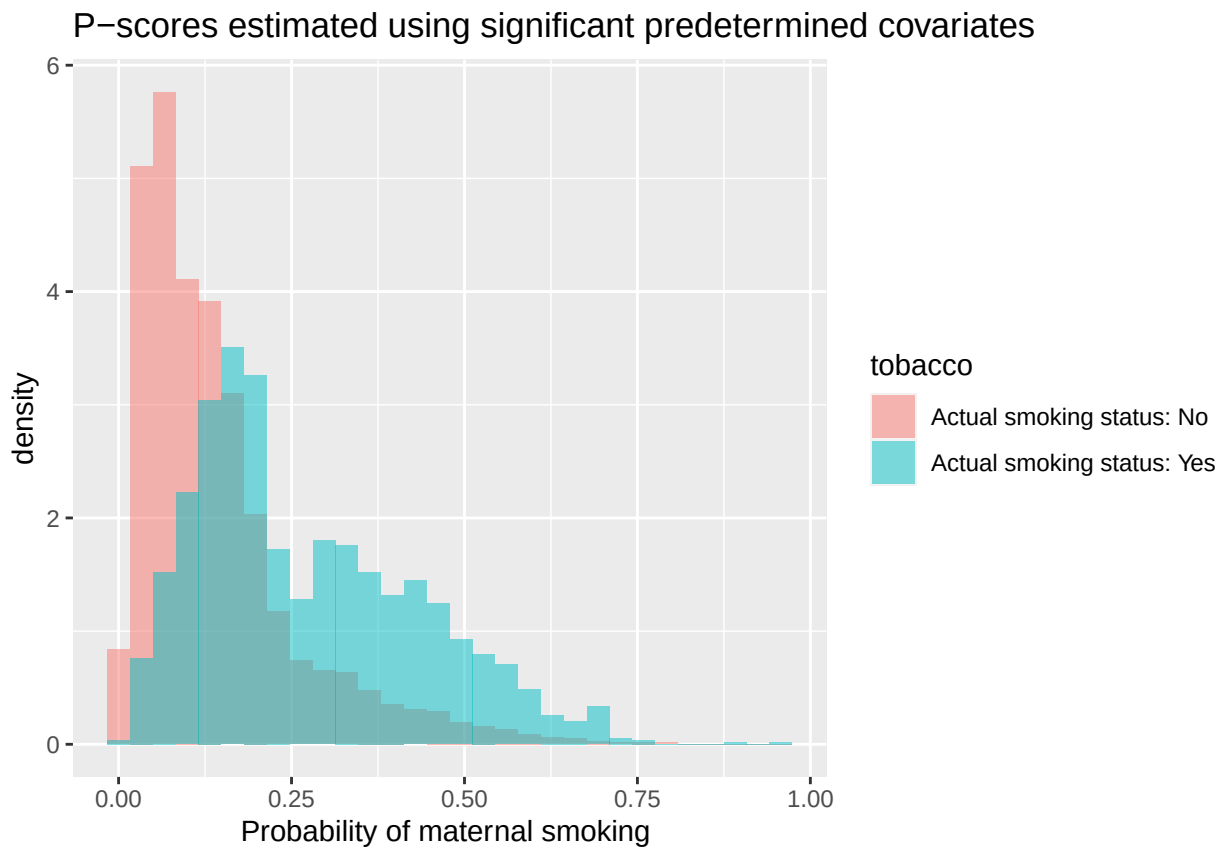
	Dependent variable:	
	tobacco	
	(1)	(2)
csex2	−0.064 (0.058)	
mrace32	−1.910*** (0.438)	−1.914*** (0.438)
mrace33	−0.965*** (0.095)	−0.967*** (0.095)
preterm2	−0.423** (0.213)	−0.422** (0.213)
dmage	−0.025*** (0.009)	−0.024*** (0.009)
dfage	0.040*** (0.007)	0.040*** (0.007)
dmeduc	−0.216*** (0.019)	−0.216*** (0.019)
dfeduc	−0.105*** (0.018)	−0.104*** (0.018)
ormoth1	0.501 (0.874)	0.511 (0.873)
ormoth2	−1.029*** (0.285)	−1.024*** (0.284)
ormoth3	−0.620 (1.123)	−0.608 (1.122)
ormoth4	−13.454 (222.562)	−13.445 (222.675)
ormoth5	−0.603 (0.417)	−0.610 (0.417)
orfath1	−15.644 (206.897)	−15.675 (206.628)
orfath2	−0.959*** (0.274)	−0.964*** (0.274)
orfath3	1.840** (0.856)	1.841** (0.856)
orfath4	−1.537** (0.641)	−1.551** (0.642)
orfath5	−0.017 (0.378)	−0.025 (0.378)
disllb	−0.0003*** (0.0001)	−0.0003*** (0.0001)
dtotord	0.094*** (0.024)	0.092*** (0.024)
dmar2	1.262*** (0.075)	1.255*** (0.075)
adequacy2	−0.075 (0.084)	
adequacy3	−0.099 (0.155)	
nprevist	−0.029*** (0.011)	−0.023*** (0.008)
Constant	2.386*** (0.356)	2.240*** (0.330)
Observations	10,000	10,000
Log Likelihood	−3,832.877	−3,833.915
Akaike Inf. Crit.	7,715.754	7,711.829
Note:		*p<0.1; **p<0.05; ***p<0.01

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




```
ggplot(p1_df, aes(x=p1_score, fill = tobacco)) +
  geom_histogram(position = "identity", alpha = 0.5,
                 mapping = aes(y = stat(density))) +
  xlab("Probability of maternal smoking ") +
  ggtitle("P-scores estimated using significant predetermined covariates")
```

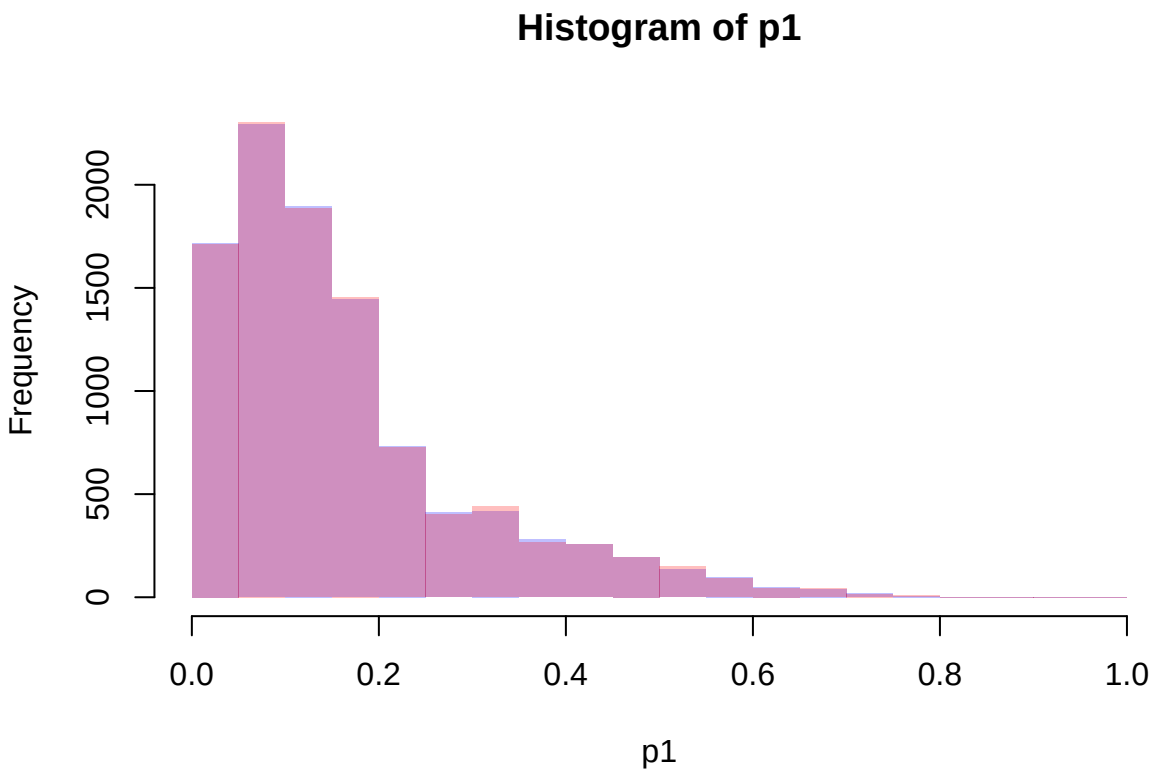
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Compare histograms of p-scores
p1 = reg_2a1 %>% predict(df2a, type = "response")
p2 = reg_2a2 %>% predict(df2a2, type = "response")
```

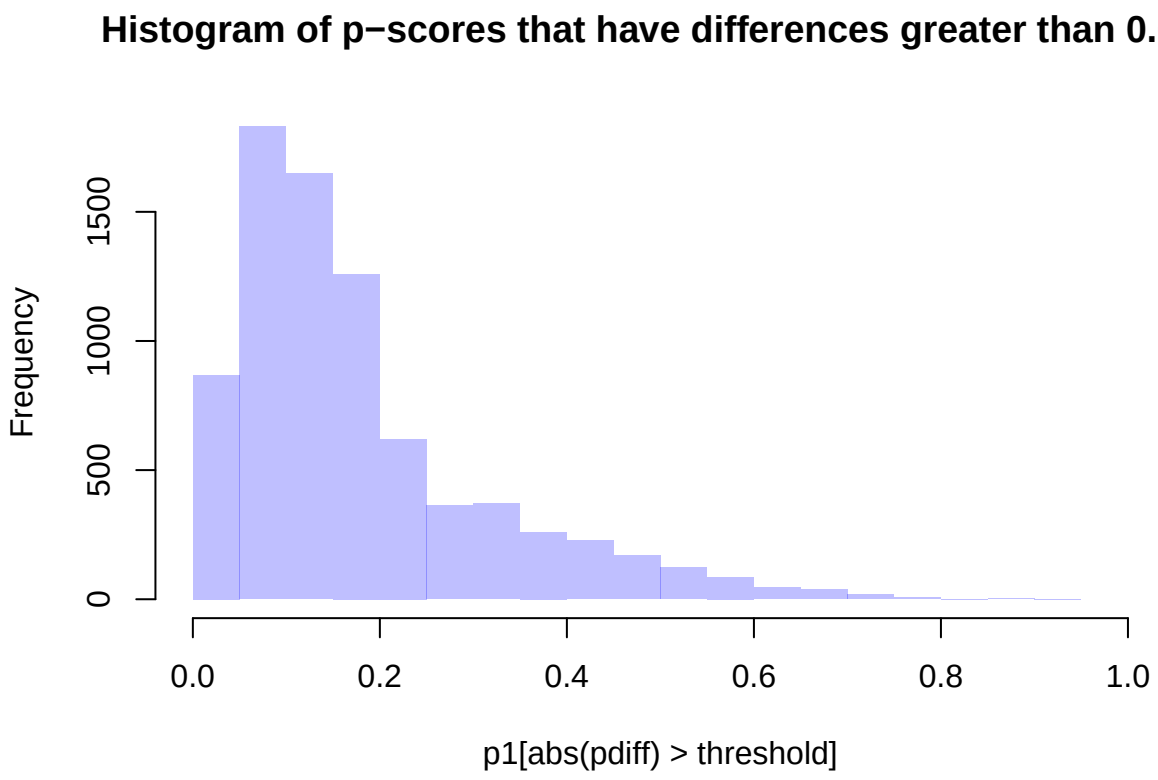
The histogram of the first predicted p-score is in orange, the second is in purple. So this fully-pink histogram is meant to show that there is nearly complete overlap in this histograms of the predicted p-scores.

```
plot(hist(p1, plot=F), col=rgb(0,0,1,1/4), border=NA, xlim=c(0,1))  # first histogram
plot(hist(p2, plot=F), col=rgb(1,0,0,1/4), border=NA, xlim=c(0,1), add=T)  # second
```

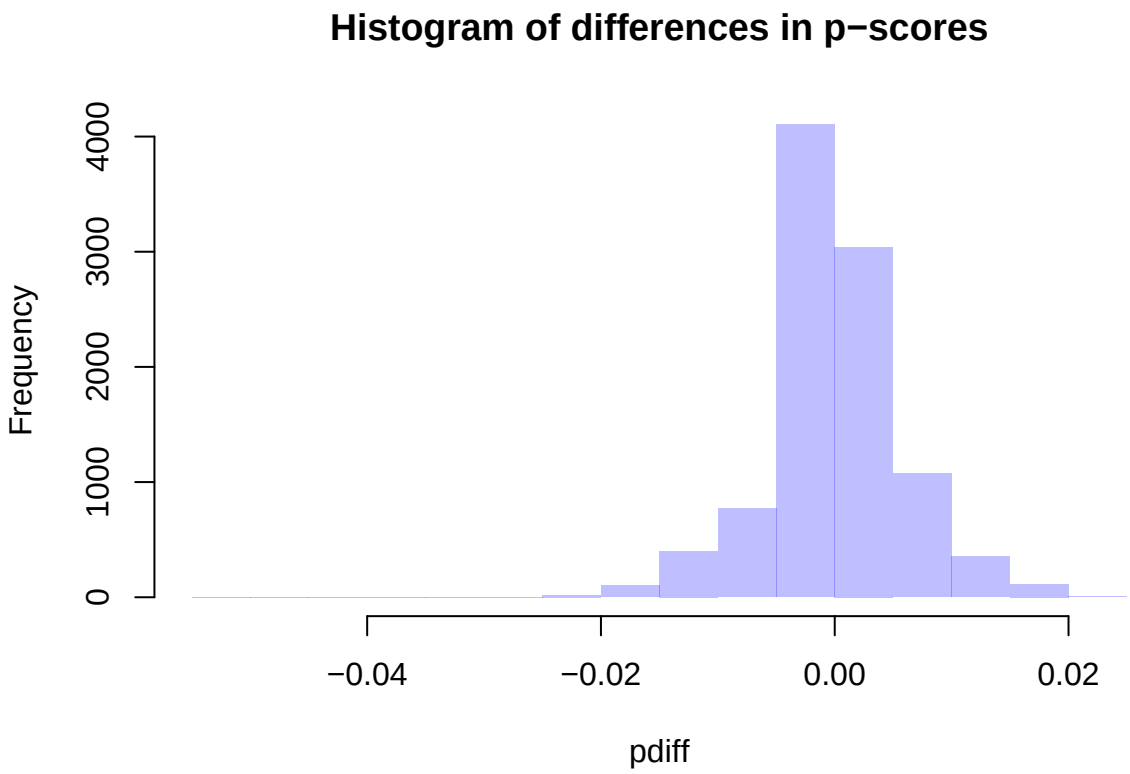


We can see where the differences are that are more than 0.001 and the shape of the differences.

```
threshold = 0.001
pdiff = p1-p2
plot(hist(p1[abs(pdiff) > threshold], plot=F), col=rgb(0,0,1,1/4), border=NA, xlim=c(0,1),
     main=paste('Histogram of p-scores that have differences greater than', threshold))
```

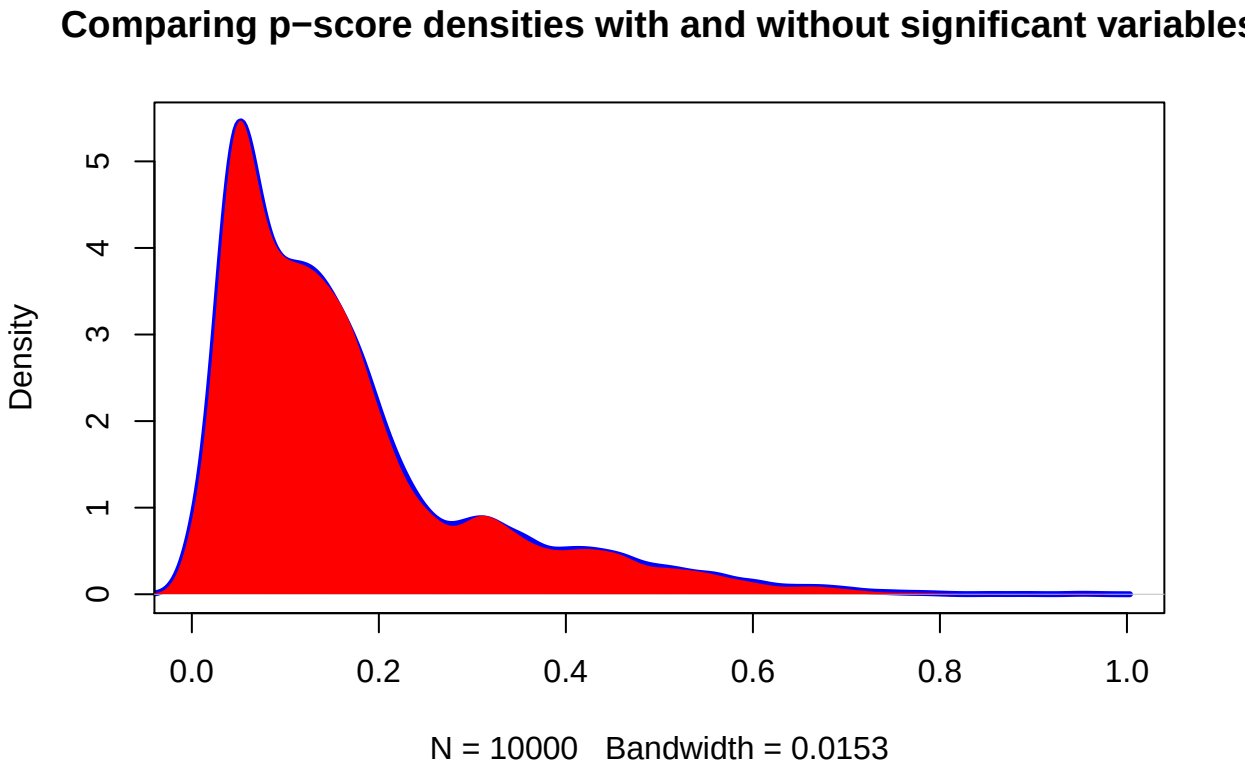



```
plot(hist(pdiff, plot=F), col=rgb(0,0,1,1/4), border=NA, main='Histogram of differences in p-scores')
```



Here's two density plots to reinforce the idea. The blue line is the density plot for the first p-score and the red filling is the density plot for the second.

```
plot(density(p1), col=rgb(0,0,1), xlim=c(0,1), lwd=3,
     main='Comparing p-score densities with and without significant variables')  # first kernel
polygon(density(p2), col="red", border=NA, xlim=c(0,1))  # second kernel filled in
```



Note that the maximum difference between any two predicted p-scores is 0.050802.

Overall, the propensity scores are very similar. However, this does not imply that we have the "correct" set of covariates in the logit specification used for our propensity score. It merely shows that excluding the non-significant variables from the propensity score estimation won't affect the propensity score estimates. This doesn't suggest that we aren't missing any potentially important covariates.

Part (b)

Control directly for the estimated propensity scores using a regression analysis, and estimate an average treatment effect. State clearly the assumptions under which your estimate is correct.

```
# Control for p-score in regression analysis
df2b = df %>%
  select(dbrwt, tobacco, csex, mrace3, preterm,
         dmage, dfage, dmeduc, dfeduc, ormoth, orfath,
         disllb, dtotord, dmar, adequacy, nprevist) %>%
  mutate(pscore = p1)
reg_2b = lm(dbrwt ~ ., data=df2b)

# Estimate ATE
ATE_2b = reg_2b$coefficients["tobacco1"]
results_df = rbind(results_df,
                   data.frame(type = 'Pscore Regression with covariates',
                              ate = ATE_2b,
                              att = NA,
```

```
                              number_vars = ncol(df2b)))

reg_2bb = lm(dbrwt ~ tobacco + pscore, data=df2b)
ATE_2bb = reg_2bb$coefficients["tobacco1"]
results_df = rbind(results_df,
                   data.frame(type = 'Pscore Regression without covariates',
                              ate = ATE_2bb,
                              att = NA,
                              number_vars = 2+1))
```

The estimated ATE of tobacco use during pregnancy when including covariates and the pscore is -242.014 – smoking during pregnancy causes a 242.014 drop in grams of the birthweight of the child. When only including the pscore, the ATE is -242.886. This is correct if we have unconfoundedness and we assume a constant treatment effect.

Part (c)

As discussed in class, one can use the estimated propensity scores to reweight the outcomes of non-smokers and estimate the average treatment effect. Compute an estimate of the average treatment effect and the "effect of the treatment on the treated" by appropriate reweighting of the data.

Imbens tells us that, after dividing by the sum of the weights in each treatment group, the feasible propensity-score-weighted ATE is:

$$\hat{\tau}_{ATE} = \frac{\sum_{i=1}^{N} \frac{Y_i \cdot D_i}{\hat{p}(X_i)}}{\sum_{i=1}^{N} \frac{D_i}{\hat{p}(X_i)}} - \frac{\sum_{i=1}^{N} \frac{(1-D_i) \cdot Y_i}{1-\hat{p}(X_i)}}{\sum_{i=1}^{N} \frac{1-D_i}{1-\hat{p}(X_i)}}$$

```
# Reweight data using p-score to weight
df2c = df2b %>%
  mutate(pscore = predict(reg_2a1, ., type = "response")) %>%
  mutate(weight = ifelse(tobacco == 1, 1/pscore, 1/(1-pscore)))
t_norm1 = sum((df2c$tobacco==1)*df2c$weight)  # smokers
c_norm1 = sum((df2c$tobacco==2)*df2c$weight)  # non-smokers
df2c = df2c %>% mutate(weight3 = ifelse(tobacco == 1, 1/pscore/t_norm1, 1/(1-pscore)/c_norm1))

# Estimate ATE
weight_mean_smoker_all = sum((df2c$tobacco==1) * df2c$weight * df2c$dbrwt) / t_norm1
weight_mean_nonsmoker_all = sum((df2c$tobacco==2) * df2c$weight * df2c$dbrwt) / c_norm1
ATE2c = weight_mean_smoker_all - weight_mean_nonsmoker_all
ATE2c
```

```
## [1] -232.7225
```

The propensity-score-weighted estimated average treatment effect is -232.72.

Imbens (2004) in a review article states that the efficient TOT estimator of $\tau_{treated}$ is derived by weighting each observation by it's propensity score. This makes intuitive sense because we want to weight-up the observations that look like the treated observations and weight-down the observations that look like the untreated. After canceling out the propensity score in the first term and dividing by the sum of the weights

for each treatment group, we get:

$$\hat{\tau}_{ATT} = \frac{1}{N_T} \sum_{i:D_i=1} Y_i - \frac{\sum_{i:D_i=0} Y_i \frac{\hat{p}(X_i)}{1-\hat{p}(X_i)}}{\sum_{i:D_i=0} \frac{\hat{p}(X_i)}{1-\hat{p}(X_i)}} = \frac{\sum_{i=1}^{N} D_i Y_i}{\sum_{i=1}^{N} D_i} - \frac{\sum_{i=1}^{N} \frac{(1-D_i)\hat{p}(X_i)Y_i}{1-\hat{p}(X_i)}}{\sum_{i=1}^{N} \frac{(1-D_i)\hat{p}(X_i)}{1-\hat{p}(X_i)}}$$

```
# Reweight data for ATT
df2c = df2c %>%
  mutate(weight2 = ifelse(tobacco == 1, 1, pscore/(1-pscore)))
t_norm2 = sum((df2c$tobacco==1)*df2c$weight2)  # smokers
c_norm2 = sum((df2c$tobacco==2)*df2c$weight2)  # non-smokers

# Estimate ATT
weight_mean_smoker_treat = sum((df2c$tobacco==1) * df2c$weight2 * df2c$dbrwt) / t_norm2
weight_mean_nonsmoker_treat = sum((df2c$tobacco==2) * df2c$weight2 * df2c$dbrwt) / c_norm2
ATT2c = weight_mean_smoker_treat - weight_mean_nonsmoker_treat
ATT2c
```

```
## [1] -250.8515
```

```
results_df = rbind(results_df,
                   data.frame(type = 'Pscore weighted data',
                              ate = ATE2c,
                              att = ATT2c,
                              number_vars = NA))
```

The propensity-score-weighted estimated average treatment effect on the treated is -250.85. If we take this result as significantly different from the ATE, then the effect of smoking on birthweight is stronger for those who look like smokers than it is in the general population. This seems to suggest that there are other, possibly unobserved, characteristics that decrease birthweight that are more common in the population that look like non-smokers than in population that looks like smokers.

Part (d)

Estimate the counterfactual densities relevant for the above part with a kernel density estimator. That is, estimate the density of birthweight (or log birthweight) if everyone smoked and again if no one smoked. Hint: Consider directly applying the Hirano, Imbens, and Ridder propensity score reweighting scheme in the context of estimating the densities of the treated and control groups (rather than the means of the treated and control groups). Stata has very useful preprogrammed commands. In addition to using the preprogrammed Stata command to compute/graph the kernel density over the entire range of birthweight, please also calculate by hand the kernel estimator at birthweight equals 3,000 grams (and provide the code you wrote that shows the calculation of the kernel estimator at this single point). Play around with a bandwidth starting with half the default Stata bandwidth. Choose the same bandwidth for all the pictures, and produce a (beautiful, production quality) figure depicting both densities.

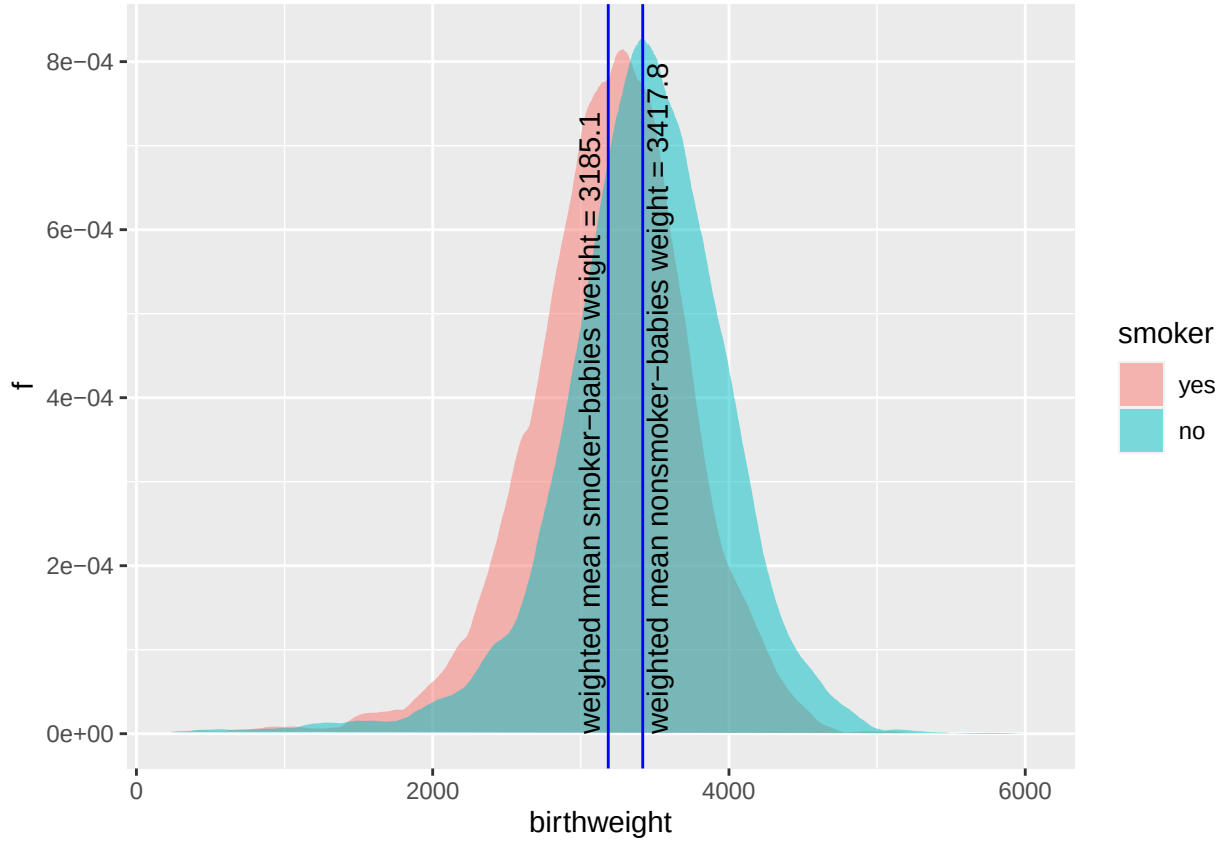
Morgan and Todd (2008) gives us the weights for ATT and ATC (average treatment effect on the controls). These come from weighting observations by $1 - p(X_i)$ instead of $p(X_i)$ to weight toward the control group:

$$\text{For } D_i = 1: \quad w_{i,ATT} = 1$$
$$\text{For } D_i = 0: \quad w_{i,ATT} = \frac{\hat{p}(X_i)}{1-\hat{p}(X_i)}$$
and
$$\text{For } D_i = 1: \quad w_{i,ATC} = \frac{1-\hat{p}(X_i)}{\hat{p}(X_i)}$$
$$\text{For } D_i = 0: \quad w_{i,ATC} = 1$$

We can use weighted kernel density estimation with the weights above:

```
# Stata default bandwidth
m = min(var(df2c$dbrwt), IQR(df2c$dbrwt)/1.349)
h_stata = 0.9*m/nrow(df2c)^(1/5)  # 44.14637 grams
df2d = df2c

# Create a vector of birthwieght values for plotting the density
x_vec = seq(min(df2d$dbrwt), max(df2d$dbrwt), length.out=1000)

# Define the kernel
kernel_fun <- function(x, h){ # Epanechnikov Kernel with bandwidth h
  weight = (3/4)*(1-(x/h)^2)*as.numeric(abs(x/h)<1)
  return(weight)
}

# Define the kernel density
density_fun <- function(x, df, h) { # Kernel density estimate with data X and weights w
  withCallingHandlers({
    # h = h_stata*2
    n = nrow(df)
    df = df %>% filter(between(dbrwt, x-h, x+h))
    if (nrow(df) == 0) {return(0)}
    kernel_vec = sapply(x - df$dbrwt, kernel_fun, h=h)
    f = sum(df$weight * kernel_vec) / (n*h) #
    return(f)
  }, warning=function(w) {
      if (startsWith(conditionMessage(w), "between() called on numeric"))
          invokeRestart("muffleWarning")
  })
}

# Calculate starting bandwidths (based on Stata formula)
m1 = min(var(filter(df2d, tobacco==1)$dbrwt), IQR(filter(df2d, tobacco==1)$dbrwt)/1.349)
h1 = 0.9*m/nrow(filter(df2d, tobacco==1))^(1/5)
m2 = min(var(filter(df2d, tobacco==2)$dbrwt), IQR(filter(df2d, tobacco==2)$dbrwt)/1.349)
h2 = 0.9*m/nrow(filter(df2d, tobacco==2))^(1/5)
# estimate counterfactual densities (takes about 16 seconds for both over 1000 point x_vec)
dens1_custom = sapply(x_vec, density_fun, df=filter(df2d, tobacco==1), h=2*h1)
dens2_custom = sapply(x_vec, density_fun, df=filter(df2d, tobacco==2), h=2*h2)

# Caclulate area under the curve to use as normalization constants for the PDFs
norm_cons1 = sum(diff(x_vec) * zoo::rollmean(dens1_custom, 2))
norm_cons2 = sum(diff(x_vec) * zoo::rollmean(dens2_custom, 2))
```

```
# Use normalization constants to scale up to true PDFs
dens_df = rbind(data.frame(birthweight=x_vec, f=(dens1_custom / norm_cons1), smoker="yes"),
                data.frame(birthweight=x_vec, f=(dens2_custom / norm_cons2), smoker="no"))

# Plot
mean_lines = data.frame(avg = c(weight_mean_smoker_all, weight_mean_nonsmoker_all),
    name = c(paste("weighted mean smoker-babies weight =", round(weight_mean_smoker_all, 1)),
             paste("weighted mean nonsmoker-babies weight =", round(weight_mean_nonsmoker_all, 1))))
p <- ggplot(dens_df, aes(x = birthweight, y = f)) +
  geom_polygon(aes(fill=smoker, group = smoker), alpha = 0.5) +
  geom_vline(data=mean_lines, mapping=aes(xintercept=avg), color="blue") +
  geom_text(data=mean_lines, mapping=aes(x=avg, y=0, label=name), size=4, angle=90, vjust=c(-0.4, 1.2),
p
```



Calculate the kernel estimator at birthweight equals 3,000 grams:

```
# for smokers
density_fun(3000, filter(df2d, tobacco==1), 2*h1) / norm_cons1
```

```
## [1] 0.0007113172
```

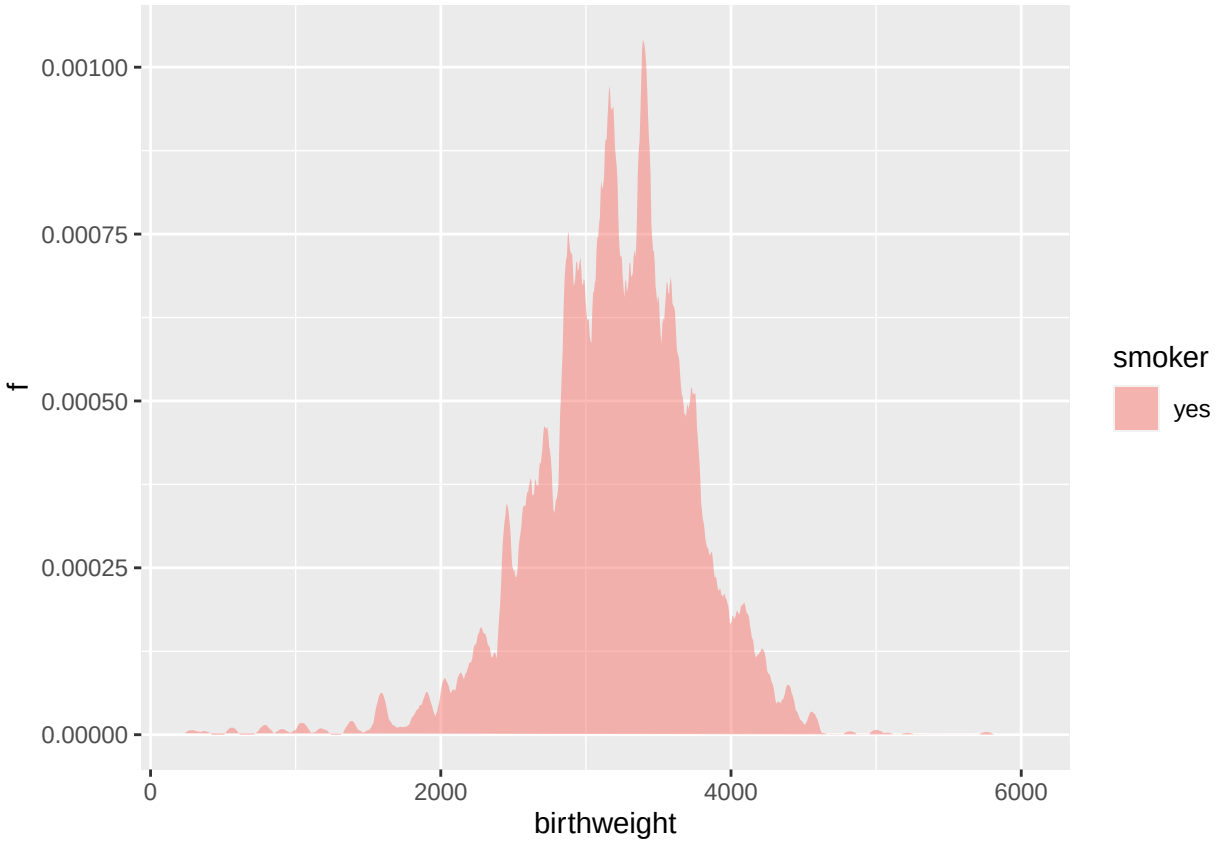
```
# for nonsmokers
density_fun(3000, filter(df2d, tobacco==2), 2*h2) / norm_cons1
```

```
## [1] 9.905399e-05
```

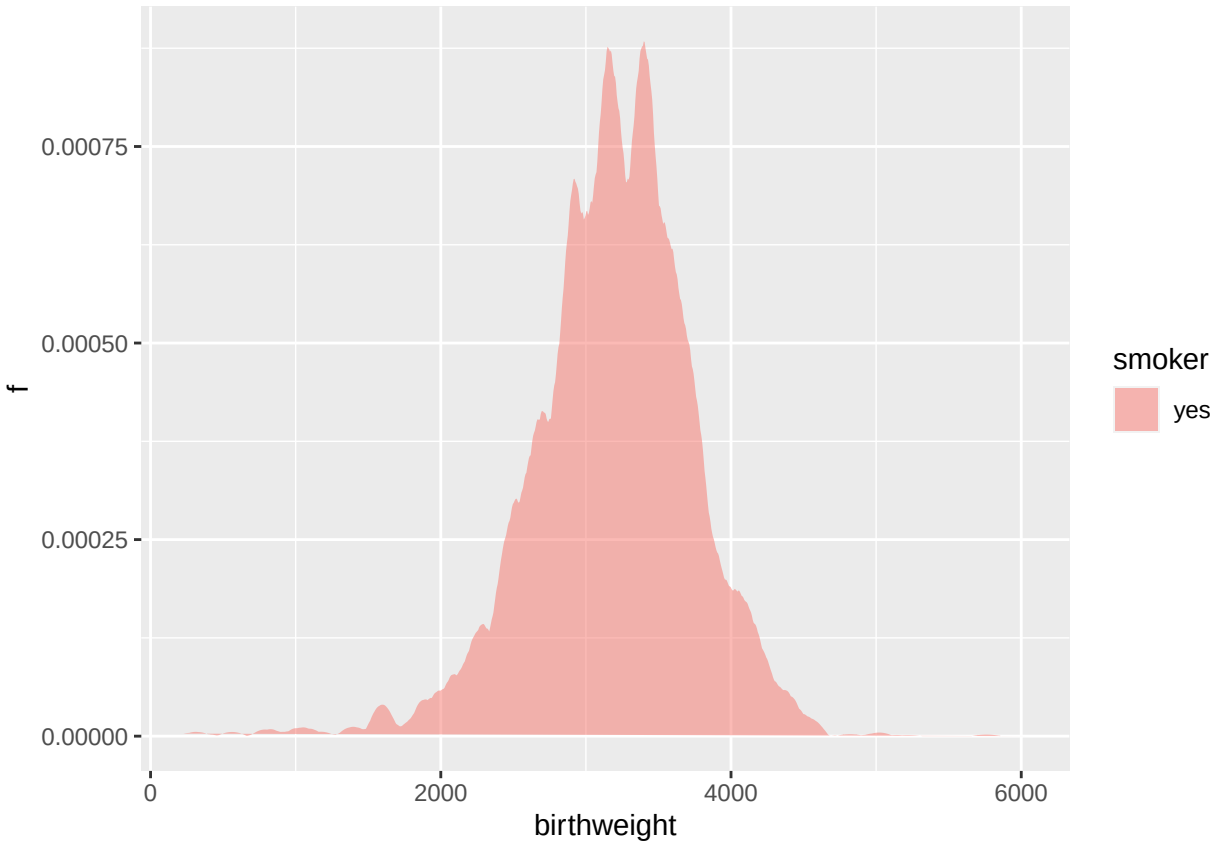
Part (e)

Take one of your densities and display an estimate of the density using different bandwidths as well as the one you settled on. What happens with bigger (smaller) bandwidths?

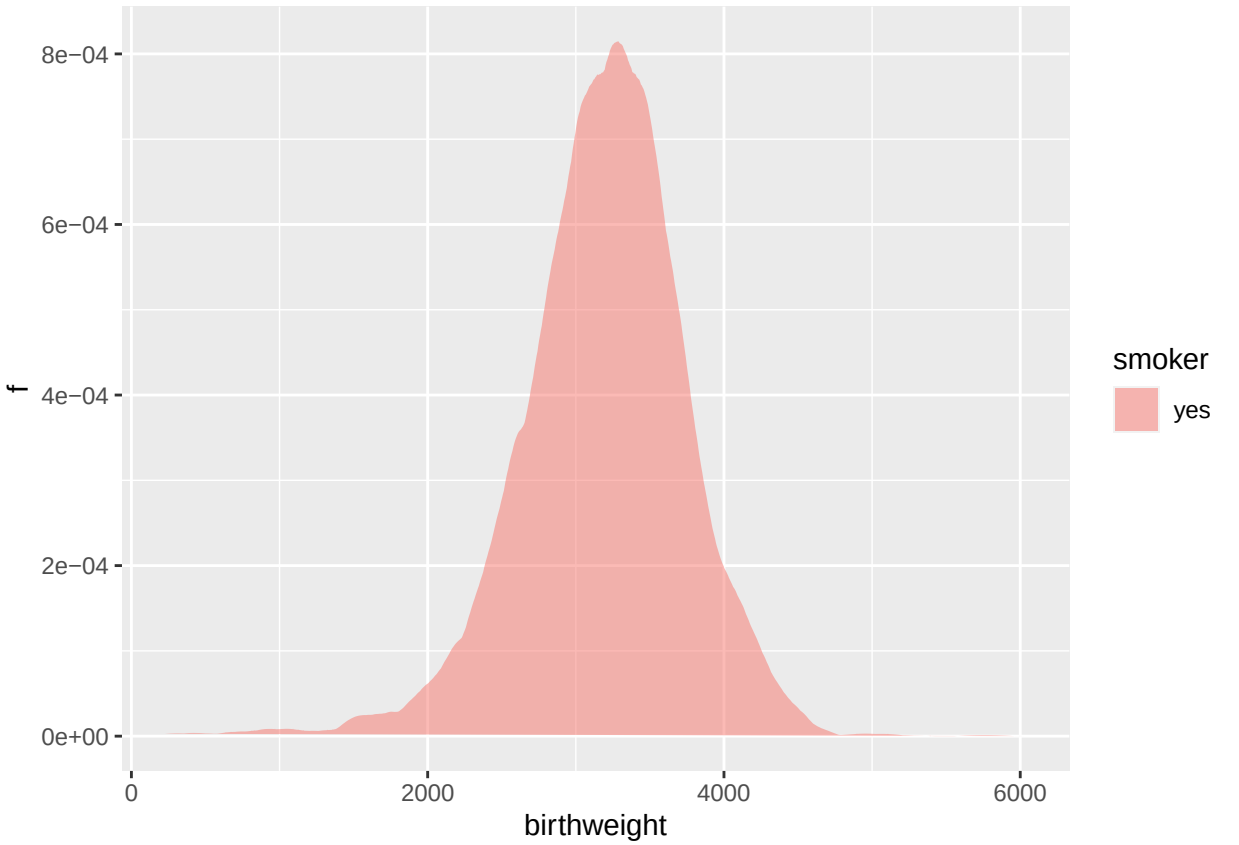
Smokers' density with bandwidth = 1/2 * Stata bandwidth



Smokers' density with bandwidth = 1 Stata bandwidth



Smokers' density with bandwidth = 2 Stata bandwidth



As we increase the bandwidth, the density esimtate gets smoother and smoother.

Part (f)

What are the benefits of the weighting approach (from part c)? What are the potential drawbacks? Pay particular attention to to the issue of people with extremely high and extremely low values of the propensity score.

Weighting on the propensity score, we can reduce the dimensionality of the covariate space to compare observations that have similar probabilities of selection into the treatment group. If we have good covariate overlap, this means we can estimate treatment effects based on comparisons of control and treatment observations that "look alike."

If we don't have good covariate overlap, however, we may have very few observations that are matched to the opposite treatment group. We expect the majority of the control observations to have a low propensity score (since it's estimating the probability of treatment based on covariates) and the majority of treatment observations to have high propensity scores. Without good covariate overlap, we would expect very few treatment observations to have very low p-scores (near 0) and very few control observations to have high p-scores (near 1).

Since our weights are $1/p(X)$ for treated observations, the weights for the rare treatment observations with p-scores near 0 can be extremely large and thus we might be relying on relatively few observations to estimate the treatment effect. Similarly, the weights are $1/(1-p(X))$ for control observations and we could be heavily weighting few observations with p-scores near 1. If we trust the outlier untreated observations, this may be reasonable, however the fact that they are outliers among the untreated is also a worrying sign such outliers may be unrepresentative as counterfactual controls.

Part (g)

Present your findings and interpret the results on the relationship between birthweight and smoking. For the estimates in parts (b) and (c), consider which of the following conditions must hold in order for that estimate to be valid:

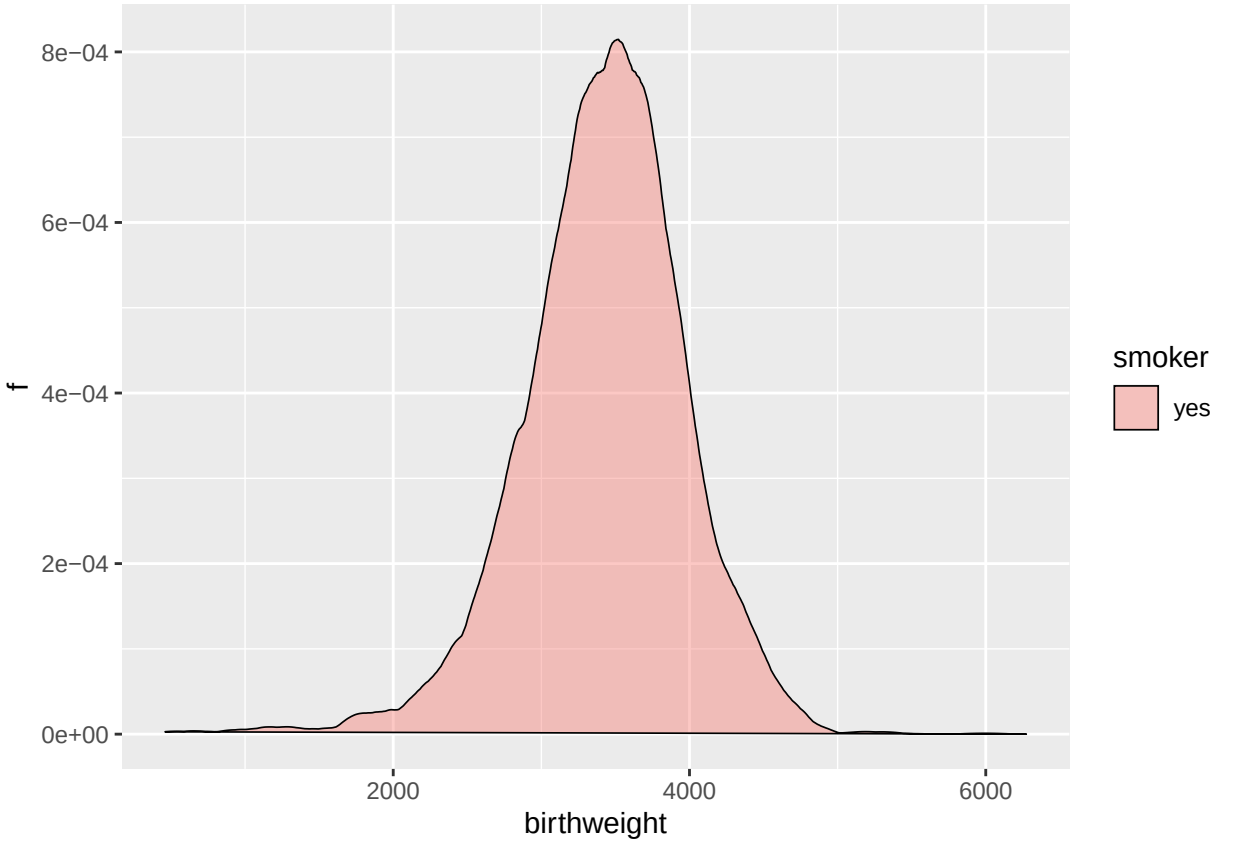
(i) The treatment effect heterogeneity is linear in the propensity score.
(ii) The treatment effect heterogeneity is not linear in the propensity score.
(iii) The decision to smoke is completely randomly assigned.
(iv) Conditional on the exogenous variables the decision to smoke is randomly assigned.

For part (b) where we estimate the ATE using a regression on the treatment and the pscore, we estimate an effect of -242.9 – that is, on average, we expect that smoking during pregnancy decreases the birthweight of the child by 242.9 grams for the general population (assuming our sample is representative of the general population). This requires that condition (iv) holds, and additionally, that the treatment effect is homogeneous since we are regressing on just the treatment indicator and the pscore.

For part (c), we use the pscore to weight the observations to create balanced set of covariates and estimate the average treatment effect to be -232.7 – on average, we expect that smoking during pregnancy decreases the birthweight of the child by 232.7 grams for the general population. We also estimate the average effect on the treated to be -250.9 – on average for women that smoke, we expect that smoking during pregnancy decreases the birthweight of the child by 232.7 grams. These results both require conditions (i) and (iv) to be valid.

We can assess the homogeneous treatment effect assumption by comparing the shape of the counterfactual distributions. Below, we have shifted the smokers' distribution up by the ATE estimated in part (c). We can see that the distributions are extremely close, giving evidence that the treatment effect is fairly homogenous in across the pscore values.

```
dens_df2 = dens_df %>%
  mutate(birthweight = ifelse(smoker=='yes', birthweight - ATE2c, birthweight))

ggplot(dens_df2, aes(x = birthweight, y = f)) +
  geom_polygon(aes(fill=smoker), alpha = 0.4, color='black', size=0.3)
```



It is unclear to us if the pscore data weighting method relies on the assumption of heterogeneous treamtent effects, if this type of counterfactual density comparison is circular reasoning.

Problem 3

A potentially more informative way to describe how birth weight affects smoking is to estimate the "non-parametric" conditional mean of birth weight as a function of the estimated probability of smoking, separately for smokers and non-smokers on the same graph. To do so, divide the data from smokers into 100 approximately equally spaced bins based on the estimated propensity score. Do the same for nonsmokers. Use the blocking estimator we discussed in class. Interpret your findings and relate them to the results in (2b).

```
#Prepping dataset arranged by pscore
df3 <- df %>%
  select(dbrwt, tobacco) %>%
  mutate(pscore = p1, smoke = ifelse(tobacco == 1, 1,0)) %>%
  select(-tobacco) %>%
  arrange(pscore)

# Check Overlap Assumption
minmax <- df3 %>%
            group_by(smoke) %>%
            summarise(max = max(pscore), min = min(pscore))

#Creating bins separately for smokers and non-smokers
df3_smoke <- df3 %>%
```

```
  filter(smoke == 1) %>%
  mutate(pscore_bin = cut(pscore, breaks = 100, labels = seq(1,100)))

df3_nosmoke <- df3 %>%
  filter(smoke == 0) %>%
  mutate(pscore_bin = cut(pscore, breaks = 100, labels = seq(1,100)))

#Generating t_k (treatment effects for bins)
df3_smoke_mean <- df3_smoke %>%
  group_by(pscore_bin) %>%
  summarise(bin_mean_smoke = mean(dbrwt), N_smoke = n())

df3_nosmoke_mean <- df3_nosmoke %>%
  group_by(pscore_bin) %>%
  summarise(bin_mean_nosmoke = mean(dbrwt), N_nosmoke = n())

#Generating t (weighted total treatment effect)
df3_merged <- full_join(df3_nosmoke_mean, df3_smoke_mean, by = "pscore_bin") %>%
  filter(is.na(bin_mean_nosmoke) == 0 & is.na(bin_mean_smoke) == 0) %>%
  mutate(dbrwt_diff = bin_mean_smoke - bin_mean_nosmoke, N = N_smoke + N_nosmoke)

N_total_3 <- sum(df3_merged$N)

df3_merged <- df3_merged %>%
  mutate(t_k_weighted = dbrwt_diff * (N/N_total_3))

t_3 <- sum(df3_merged$t_k_weighted)
results_df = rbind(results_df,
                   data.frame(type = 'Blocking on pscore (continuous weight)',
                              ate = t_3,
                              att = NA,
                              number_vars = NA))
```

First, for each group of smokers and nonsmokers, we see that the range of estimated propensity score is (0, 0.9496) for nonsmokers and (0.0067, 0.9573) for smokers. The range is very similar, meaning overlap assumption is satisfied, and trimming for blocking is not needed. This also ensures that what we have done above, which is creating 100 equal bins between the minimum and maximum propensity score for each group, will be very similar to creating the same 100 bins for the two groups such that first bin is [0, 0.01] and so on.

Using the blocking estimator, we find a smoking treatment effect of -233.735

Problem 4

Low birth weight births (less than 2500 grams) are considered particularly undesirable since they comprise a large share of infant deaths. Redo question 3 using an indicator for low birth weight birth as the outcome of interest. Interpret your findings.

```
# Prepping dataset with indicator variable
df4 <- df3 %>%
  mutate(low_dbrwt = ifelse(dbrwt < 2500, 1, 0))

# Creating bins separately for smokers and non-smokers
df4_smoke <- df4 %>%
  filter(smoke == 1) %>%
```

```
  mutate(pscore_bin = cut(pscore, breaks = 100, labels = seq(1,100)))

df4_nosmoke <- df4 %>%
  filter(smoke == 0) %>%
  mutate(pscore_bin = cut(pscore, breaks = 100, labels = seq(1,100)))

# Generating t_k (treatment effects for bins)
df4_smoke_mean <- df4_smoke %>%
  group_by(pscore_bin) %>%
  summarise(bin_mean_smoke = mean(low_dbrwt), N_smoke = n())

df4_nosmoke_mean <- df4_nosmoke %>%
  group_by(pscore_bin) %>%
  summarise(bin_mean_nosmoke = mean(low_dbrwt), N_nosmoke = n())

# Generating t (weighted total treatment effect)
df4_merged <- full_join(df4_nosmoke_mean, df4_smoke_mean, by = "pscore_bin") %>%
  filter(is.na(bin_mean_nosmoke) == 0 & is.na(bin_mean_smoke) == 0) %>%
  mutate(low_dbrwt_diff = bin_mean_smoke - bin_mean_nosmoke, N = N_smoke + N_nosmoke)

N_total_4 <- sum(df4_merged$N)

df4_merged <- df4_merged %>%
  mutate(t_k_weighted = low_dbrwt_diff * (N/N_total_4))

t_4 <- sum(df4_merged$t_k_weighted)
results_df = rbind(results_df,
                   data.frame(type = 'Blocking on pscore (indicator for low weight)',
                              ate = t_4,
                              att = NA,
                              number_vars = NA))
```

Using the blocking estimator, we find a smoking treatment effect of 0.04. That is smokers are 4 percentage points more likely to have a baby with low birth weight than non-smokers.

Problem 5

Let's link matching back to regression. Consider the conditional expectation function $\mathbb{E}[birthweight \mid X]$, where X contains the following variables: `rectype pldel3 cntocpop stresfip dmage mrace3 dmar adequacy csex dplural`.

Part (a)

Develop a regression that you are confident estimates $\mathbb{E}[birthweight \mid X]$ as $N \to \infty$? Why are you confident that your regression gets the CEF right?

Saturated model for discrete regressors is a sufficient condition for a linear CEF. Hence, I would create a dummy variable for each unique combination of covariate values and run linear regression on these newly created dummy variables to estimate CEF.

Part (b)

Now run the regression you propose above, but add the treatment (your binary smoking variable) as the righthand side variable of interest. Prove that if the treatment effect of

smoking on birthweight is independent of the covariates in X, then exact matching and your regression estimate the same thing. You may assume the conditional independence assumption holds given the variables in X listed above.

```
# Select variables
df5a = df %>%
          select(dbrwt, tobacco, rectype, pldel3, cntocpop, stresfip,
                 dmage, mrace3, dmar, adequacy, csex, dplural) %>%
          mutate(stresfip = as.factor(stresfip), dmage = as.factor(dmage))

# Create factor variable (for dummies needed for regression) for unique combination of covariate values
df5a_unique = df5a %>%
                select(-c(dbrwt, tobacco)) %>%
                distinct() %>%
                arrange(rectype, pldel3, cntocpop, stresfip,
                  dmage, mrace3, dmar, adequacy, csex, dplural) %>%
                mutate(uniq = row_number())
df5a <- full_join(df5a, df5a_unique, by = c("rectype", "pldel3", "cntocpop",
                  "stresfip", "dmage", "mrace3", "dmar", "adequacy", "csex", "dplural"))
df5a <- df5a %>% mutate(uniq = as.factor(uniq))

# Run regression on dbrwt with saturated model (incl tobacco)
start_t = proc.time()
reg_5a <- lm(dbrwt ~ tobacco + uniq, data = df5a)
end_t = proc.time()
print('time to regress:')
```

```
## [1] "time to regress:"
```

```
print(end_t[3] - start_t[3])
```

```
## elapsed
##  44.006
```

```
ATE5b = reg_5a$coefficients['tobacco1']

results_df = rbind(results_df,
                   data.frame(type = 'Saturated regression',
                              ate = ATE5b,
                              att = NA,
                              number_vars = length(levels(df5a$uniq)) + 2))
```

```
stargazer(reg_5a,
          type = 'latex',
          keep = "tobacco1",
          perl=TRUE, header = FALSE, table.placement = '!h',
          title = "Saturated Regression",
          align = TRUE, no.space = TRUE, font.size = "small",
          notes = c("For brevity, only the treatment is presented;",
                    paste(length(levels(df5a$uniq)), "interaction terms are not shown.")))
```

Let Y = dbrwt, D = tobacco, and X = covariates listed for quesiton 5a. Assume $Y_i = \alpha + \beta D_i + \delta h(x_i) + v_i$ and thereby $Y_i = \alpha + \beta \tilde{D}_i + \delta h(x_i) + u_i$, where $\tilde{D}_i = D_i - \mathbb{E}[D_i|X_i]$ and $u_i = \epsilon_i + \beta \mathbb{E}[D_i|X_i]$. We showed, in class, that $\mathbb{E}[u_i \tilde{D}_i] = 0$ under un-confounded-ness and homogeneous treatment, and can get a consistent estimate of β if the CEF is known. While our CEF is not known, given we have linear CEF with a saturated model, we can simply include X's as additional control.

Table 4: Saturated Regression

	Dependent variable:
	dbrwt
tobacco1	-226.648^{***}
	(17.605)
Observations	10,000
R^2	0.380
Adjusted R^2	0.174
Residual Std. Error	531.145 (df = 7507)
F Statistic	1.844^{***} (df = 2492; 7507)
Note:	*p<0.1; **p<0.05; ***p<0.01
	For brevity, only the treatment is presented;
	2492 interaction terms are not shown.

Let matrix $Z = [D\ E[D|X_i]]$.

$$
\begin{aligned}
\tau_{(OLS\ 5b)} &= \frac{\mathbb{E}[Z_i Y_i]}{Var[Z_i]} \\
&= \frac{\mathbb{E}[\tilde{D}_i Y_i]}{Var[\tilde{D}_i]} && \text{given consistent estimate} \\
&= \frac{\mathbb{E}[\mathbb{E}[\tilde{D} Y_i | D_i, X_i]]}{Var[\tilde{D}_i]} \\
&= \frac{\mathbb{E}[\tilde{D}\mathbb{E}[Y_i | D_i, X_i]]}{Var[\tilde{D}_i]} \\
&= \frac{\mathbb{E}[\tilde{D}(\mathbb{E}[Y_i | D_i = 0, X_i] + \mathbb{E}[\tau_i | D_i, X_i] D_i)]}{Var[\tilde{D}_i]} \\
&= \frac{\mathbb{E}[\tilde{D}(\mathbb{E}[Y_i | D_i = 0, X_i] + \mathbb{E}[\tau_i | X_i] D_i)]}{Var[\tilde{D}_i]} && \text{CIA} \\
&= \frac{\mathbb{E}[\tilde{D}\mathbb{E}[Y_i | D_i = 0, X_i] + \tilde{D} D_i \mathbb{E}[\tau_i]]}{Var[\tilde{D}_i]} && \text{Trt independent of X} \\
&= \frac{\mathbb{E}[D_i \mathbb{E}[Y_i | D_i = 0, X_i] - \mathbb{E}[D_i | X_i]\mathbb{E}[Y_i | D_i = 0, X_i] + \tilde{D} D_i \mathbb{E}[\tau_i]]}{Var[\tilde{D}_i]} \\
&= \frac{\mathbb{E}[D_i \mathbb{E}[Y_i | D_i = 0, X_i]] - \mathbb{E}[\mathbb{E}[D_i | X_i]\mathbb{E}[Y_i | D_i = 0, X_i]] + \mathbb{E}[\tilde{D} D_i \mathbb{E}[\tau_i]]}{Var[\tilde{D}_i]} \\
&= \frac{\mathbb{E}[D_i]\mathbb{E}[Y_i | D_i = 0, X_i] - \mathbb{E}[D_i]\mathbb{E}[Y_i | D_i = 0, X_i] + \mathbb{E}[\tilde{D} D_i]\mathbb{E}[\tau_i]}{Var[\tilde{D}_i]} \\
&= \frac{\mathbb{E}[\tilde{D} D_i]\mathbb{E}[\tau_i]}{Var[\tilde{D}_i]} \\
&= \frac{[\mathbb{E}[D_i^2] - \mathbb{E}[D_i]^2]\mathbb{E}[\tau_i]}{Var[\tilde{D}_i]} \\
&= \frac{Var[D_i]\mathbb{E}[\tau_i]}{Var[D_i]} \\
&= \mathbb{E}[\tau_i]
\end{aligned}
$$

Hence, we know our regression from 5b estimates Average Treatment Effect.

Furthermore, we have assumed CIA holds, that conditioning on X's, materal tobacco is as good as randomly assigned. Hence, matching exactly on X's and comparing outcomes within the match across treatment and control groups should also derive Average Treatment Effect. Hence, our regression from 5b would estimate the same thing as exact matching.

Part (c)

Develop a weighted version of the exact matching estimator that estimates the same thing as the regression above (regardless of whether the treatment effect is independent of covariates).

$$\hat{\tau}_M = \frac{1}{N_T} \sum_{i:D_i=1} \left[y_i - \sum_{j:D_j=0} w_i(j) y_j \right]$$

$$\text{where } w_i(j) = \begin{cases} \frac{1}{N_{CC}(i)} & \text{if } i\&j \text{ share a cell} \\ 0 & \text{o.w.} \end{cases}$$

Given discrete covariates establishing cells for exact matching, we can set $w_i(j)$ equal to 0 for control observations j that do not get (exactly) matched to treatment observation i; for control observations with same covariates matched to i, we would set $w_i(j)$ equal to $\frac{1}{N_{CC}(i)}$, where $N_{CC}(i)$ is number of control observations that share the same covariate cell as observation i.

Part (d)

Estimate the weighted matching estimator you propose. Compare it to the regression estimate from part (b). Are they similar?

```
# Make two df5a--one of treated and one of untreated.
df5c <- df5a %>%
  select(dbrwt, tobacco, uniq)
df5c_t <- subset(df5c, tobacco ==1)
df5c_t <- df5c_t %>%
  mutate(dbrwt_t = dbrwt) %>%
  select(dbrwt_t, uniq)

df5c_c <- subset(df5c, tobacco ==2)
df5c_c <- df5c_c %>%
  mutate(dbrwt_c = dbrwt) %>%
  select(dbrwt_c, uniq)

# Get N treated and controls in each bucket
df5c_t2 <- df5c_t %>%
  group_by(uniq) %>%
  mutate(N_t = n())

df5c_c2 <- df5c_c %>%
  group_by(uniq) %>%
  mutate(count = n())

# Collapse controls down to uniq and take average of weight
df5c_c3 <- aggregate(df5c_c2$count, list(df5c_c2$uniq), mean)
df5c_c3 <- df5c_c3 %>%
```

```
  rename(uniq = Group.1, N_c = x)

df5c_c4 <- aggregate(df5c_c$dbrwt_c, list(df5c_c$uniq), mean)
df5c_c4 <- df5c_c4 %>%
  rename(uniq = Group.1, dbrwt_c = x)

df5c_c5 <- merge(df5c_c3, df5c_c4, by = "uniq", all.x = F, all.y = F)


# Merge the treated and collapsed control obs
df5c2 <- merge(df5c_t2, df5c_c5, by = "uniq", all.x = F, all.y = F)
df5c2 <- df5c2 %>%
  mutate(diff_wt = dbrwt_t - dbrwt_c)

summary(df5c2$diff_wt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3032.2  -589.4  -227.0  -226.0   171.0  2734.9
```

```
ATE5d = mean(df5c2$diff_wt)

results_df = rbind(results_df,
                   data.frame(type = 'Weighted exact matching',
                              ate = ATE5d,
                              att = NA,
                              number_vars = NA))
```

Part (e)

Is the sample size of your regression the same as the sample size of your matching estimator, or does the regression have more observations? If the regression has more observations, why don't these extra observations influence the treatment effect estimate?

The regression has more observations because it includes all the observation in the data while the matching estimator includes only those treated and untreated observations with a match in the untreated and treated group, respectively. However, this discrepancy do not influence the treatment effect estimate because treated and untreated observations without any untreated or treated match (respectively) are given a weight of zero.

Part (f)

Compute a standard error for your matching estimator using the formula from Imbens (2015). Specifically, note that your matching estimator should have a form

$$\frac{1}{N_t}\sum_{d_i=1} w_i y_i - \frac{1}{N_c}\sum_{d_i=0} w_i y_i$$

where $\sum_{d_i=1} w_i = N_t$ and $\sum_{d_i=0} w_i = N_c$. Then the conditional variance is approximately

$$\sum_i \left(\frac{d_i}{N_t^2} + \frac{1-d_i}{N_c^2} w_i^2 \hat{\sigma}^2_{d_i}(x_i) \right),$$

where $\hat{\sigma}^2_{d_i}(x_i) = \frac{1}{2}(y_i - y_{nn(i)})$, and $y_{nn(i)}$ is the nearest neighbor to observation i with the *same* treatment status. Figure out the implicit weights w_i in your estimator from part (d), and compute the conditional variance. Is it close to your regression coefficient variance?

Problem 6

Concisely and coherently summarize your overall results, providing some intuition. Write it like you would the conclusion of a paper. In this summary, describe whether you think your best estimate of the effects of smoking is credibly identified. State why or why not.

In conclusion, the treatment effect of mother's smoking habits on baby birthweight appears to be robust across a number of identification strategies. Each method–be it matching by propensity score or on covariates, or using a regression or direct matching estimator–produces extremely similar results. While surprising on the surface, the reason for this is simple–they all use essentially the same mechanics under the hood.

Each of these methods compares treated and untreated observations based on similar characteristics. In the case of propensity score matching, observations are weighted by their likelihood of being treated, giving more weight to untreated observations that by all other measures look like treated observations. In the case of exact matching, treated observations are paired with untreated observations based on relevant characteristics and their differences are computed. For these reasons, we believe the true effect of smoking on baby birthweight to be around -220.

```
##                                                    type           ate       att
## tobacco1                                 Series Regression -251.11349348        NA
## tobacco11             Pscore Regression with covariates -242.01442003        NA
## tobacco12          Pscore Regression without covariates -242.88607515        NA
## 1                                   Pscore weighted data -232.72253851 -250.8515
## 11               Blocking on pscore (continuous weight) -233.73511002        NA
## 12        Blocking on pscore (indicator for low weight)    0.03984294        NA
## tobacco13                          Saturated regression -226.64800494        NA
## 13                              Weighted exact matching -225.99070945        NA
##           number_vars
## tobacco1          311
## tobacco11          17
## tobacco12           3
## 1                  NA
## 11                 NA
## 12                 NA
## tobacco13        2494
## 13                 NA
```

```
## , , att = -250.851460505855, number_vars = 3
##
##                                                 ate
## type                                             -251.113493475501
##   Series Regression                                                0
##   Pscore Regression with covariates                                0
##   Pscore Regression without covariates                             0
##   Pscore weighted data                                             0
##   Blocking on pscore (continuous weight)                           0
##   Blocking on pscore (indicator for low weight)                    0
##   Saturated regression                                             0
##   Weighted exact matching                                          0
##                                                 ate
## type                                             -242.886075151731
##   Series Regression                                                0
##   Pscore Regression with covariates                                0
##   Pscore Regression without covariates                             0
##   Pscore weighted data                                             0
##   Blocking on pscore (continuous weight)                           0
```

```
##   Blocking on pscore (indicator for low weight)                0
##   Saturated regression                                         0
##   Weighted exact matching                                      0
##                                                  ate
## type                                              -242.01442003439
##   Series Regression                                          0
##   Pscore Regression with covariates                          0
##   Pscore Regression without covariates                       0
##   Pscore weighted data                                       0
##   Blocking on pscore (continuous weight)                     0
##   Blocking on pscore (indicator for low weight)              0
##   Saturated regression                                       0
##   Weighted exact matching                                    0
##                                                  ate
## type                                              -233.735110021543
##   Series Regression                                          0
##   Pscore Regression with covariates                          0
##   Pscore Regression without covariates                       0
##   Pscore weighted data                                       0
##   Blocking on pscore (continuous weight)                     0
##   Blocking on pscore (indicator for low weight)              0
##   Saturated regression                                       0
##   Weighted exact matching                                    0
##                                                  ate
## type                                              -232.722538508639
##   Series Regression                                          0
##   Pscore Regression with covariates                          0
##   Pscore Regression without covariates                       0
##   Pscore weighted data                                       0
##   Blocking on pscore (continuous weight)                     0
##   Blocking on pscore (indicator for low weight)              0
##   Saturated regression                                       0
##   Weighted exact matching                                    0
##                                                  ate
## type                                              -226.648004943488
##   Series Regression                                          0
##   Pscore Regression with covariates                          0
##   Pscore Regression without covariates                       0
##   Pscore weighted data                                       0
##   Blocking on pscore (continuous weight)                     0
##   Blocking on pscore (indicator for low weight)              0
##   Saturated regression                                       0
##   Weighted exact matching                                    0
##                                                  ate
## type                                              -225.990709445074
##   Series Regression                                          0
##   Pscore Regression with covariates                          0
##   Pscore Regression without covariates                       0
##   Pscore weighted data                                       0
##   Blocking on pscore (continuous weight)                     0
##   Blocking on pscore (indicator for low weight)              0
##   Saturated regression                                       0
##   Weighted exact matching                                    0
##                                                  ate
```

```
## type                                                0.0398429384429508
##   Series Regression                                                  0
##   Pscore Regression with covariates                                  0
##   Pscore Regression without covariates                               0
##   Pscore weighted data                                               0
##   Blocking on pscore (continuous weight)                             0
##   Blocking on pscore (indicator for low weight)                      0
##   Saturated regression                                               0
##   Weighted exact matching                                            0
##
## , , att = -250.851460505855, number_vars = 17
##
##                                                  ate
## type                                              -251.113493475501
##   Series Regression                                                  0
##   Pscore Regression with covariates                                  0
##   Pscore Regression without covariates                               0
##   Pscore weighted data                                               0
##   Blocking on pscore (continuous weight)                             0
##   Blocking on pscore (indicator for low weight)                      0
##   Saturated regression                                               0
##   Weighted exact matching                                            0
##                                                  ate
## type                                              -242.886075151731
##   Series Regression                                                  0
##   Pscore Regression with covariates                                  0
##   Pscore Regression without covariates                               0
##   Pscore weighted data                                               0
##   Blocking on pscore (continuous weight)                             0
##   Blocking on pscore (indicator for low weight)                      0
##   Saturated regression                                               0
##   Weighted exact matching                                            0
##                                                  ate
## type                                              -242.01442003439
##   Series Regression                                                  0
##   Pscore Regression with covariates                                  0
##   Pscore Regression without covariates                               0
##   Pscore weighted data                                               0
##   Blocking on pscore (continuous weight)                             0
##   Blocking on pscore (indicator for low weight)                      0
##   Saturated regression                                               0
##   Weighted exact matching                                            0
##                                                  ate
## type                                              -233.735110021543
##   Series Regression                                                  0
##   Pscore Regression with covariates                                  0
##   Pscore Regression without covariates                               0
##   Pscore weighted data                                               0
##   Blocking on pscore (continuous weight)                             0
##   Blocking on pscore (indicator for low weight)                      0
##   Saturated regression                                               0
##   Weighted exact matching                                            0
##                                                  ate
## type                                              -232.722538508639
```

```
##   Series Regression                                          0
##   Pscore Regression with covariates                          0
##   Pscore Regression without covariates                       0
##   Pscore weighted data                                       0
##   Blocking on pscore (continuous weight)                     0
##   Blocking on pscore (indicator for low weight)              0
##   Saturated regression                                       0
##   Weighted exact matching                                    0
##                                                 ate
## type                                             -226.648004943488
##   Series Regression                                          0
##   Pscore Regression with covariates                          0
##   Pscore Regression without covariates                       0
##   Pscore weighted data                                       0
##   Blocking on pscore (continuous weight)                     0
##   Blocking on pscore (indicator for low weight)              0
##   Saturated regression                                       0
##   Weighted exact matching                                    0
##                                                 ate
## type                                             -225.990709445074
##   Series Regression                                          0
##   Pscore Regression with covariates                          0
##   Pscore Regression without covariates                       0
##   Pscore weighted data                                       0
##   Blocking on pscore (continuous weight)                     0
##   Blocking on pscore (indicator for low weight)              0
##   Saturated regression                                       0
##   Weighted exact matching                                    0
##                                                 ate
## type                                             0.0398429384429508
##   Series Regression                                           0
##   Pscore Regression with covariates                           0
##   Pscore Regression without covariates                        0
##   Pscore weighted data                                        0
##   Blocking on pscore (continuous weight)                      0
##   Blocking on pscore (indicator for low weight)               0
##   Saturated regression                                        0
##   Weighted exact matching                                     0
##
## , , att = -250.851460505855, number_vars = 311
##
##                                                 ate
## type                                             -251.113493475501
##   Series Regression                                          0
##   Pscore Regression with covariates                          0
##   Pscore Regression without covariates                       0
##   Pscore weighted data                                       0
##   Blocking on pscore (continuous weight)                     0
##   Blocking on pscore (indicator for low weight)              0
##   Saturated regression                                       0
##   Weighted exact matching                                    0
##                                                 ate
## type                                             -242.886075151731
##   Series Regression                                          0
```

```
##   Pscore Regression with covariates                              0
##   Pscore Regression without covariates                           0
##   Pscore weighted data                                           0
##   Blocking on pscore (continuous weight)                         0
##   Blocking on pscore (indicator for low weight)                  0
##   Saturated regression                                           0
##   Weighted exact matching                                        0
##                                                 ate
## type                                             -242.01442003439
##   Series Regression                                             0
##   Pscore Regression with covariates                             0
##   Pscore Regression without covariates                          0
##   Pscore weighted data                                          0
##   Blocking on pscore (continuous weight)                        0
##   Blocking on pscore (indicator for low weight)                 0
##   Saturated regression                                          0
##   Weighted exact matching                                       0
##                                                 ate
## type                                             -233.735110021543
##   Series Regression                                             0
##   Pscore Regression with covariates                             0
##   Pscore Regression without covariates                          0
##   Pscore weighted data                                          0
##   Blocking on pscore (continuous weight)                        0
##   Blocking on pscore (indicator for low weight)                 0
##   Saturated regression                                          0
##   Weighted exact matching                                       0
##                                                 ate
## type                                             -232.722538508639
##   Series Regression                                             0
##   Pscore Regression with covariates                             0
##   Pscore Regression without covariates                          0
##   Pscore weighted data                                          0
##   Blocking on pscore (continuous weight)                        0
##   Blocking on pscore (indicator for low weight)                 0
##   Saturated regression                                          0
##   Weighted exact matching                                       0
##                                                 ate
## type                                             -226.648004943488
##   Series Regression                                             0
##   Pscore Regression with covariates                             0
##   Pscore Regression without covariates                          0
##   Pscore weighted data                                          0
##   Blocking on pscore (continuous weight)                        0
##   Blocking on pscore (indicator for low weight)                 0
##   Saturated regression                                          0
##   Weighted exact matching                                       0
##                                                 ate
## type                                             -225.990709445074
##   Series Regression                                             0
##   Pscore Regression with covariates                             0
##   Pscore Regression without covariates                          0
##   Pscore weighted data                                          0
##   Blocking on pscore (continuous weight)                        0
```

```
##   Blocking on pscore (indicator for low weight)                0
##   Saturated regression                                         0
##   Weighted exact matching                                      0
##                                                          ate
## type                                                      0.0398429384429508
##   Series Regression                                            0
##   Pscore Regression with covariates                            0
##   Pscore Regression without covariates                         0
##   Pscore weighted data                                         0
##   Blocking on pscore (continuous weight)                       0
##   Blocking on pscore (indicator for low weight)                0
##   Saturated regression                                         0
##   Weighted exact matching                                      0
##
## , , att = -250.851460505855, number_vars = 2494
##
##                                                          ate
## type                                                      -251.113493475501
##   Series Regression                                            0
##   Pscore Regression with covariates                            0
##   Pscore Regression without covariates                         0
##   Pscore weighted data                                         0
##   Blocking on pscore (continuous weight)                       0
##   Blocking on pscore (indicator for low weight)                0
##   Saturated regression                                         0
##   Weighted exact matching                                      0
##                                                          ate
## type                                                      -242.886075151731
##   Series Regression                                            0
##   Pscore Regression with covariates                            0
##   Pscore Regression without covariates                         0
##   Pscore weighted data                                         0
##   Blocking on pscore (continuous weight)                       0
##   Blocking on pscore (indicator for low weight)                0
##   Saturated regression                                         0
##   Weighted exact matching                                      0
##                                                          ate
## type                                                      -242.01442003439
##   Series Regression                                           0
##   Pscore Regression with covariates                           0
##   Pscore Regression without covariates                        0
##   Pscore weighted data                                        0
##   Blocking on pscore (continuous weight)                      0
##   Blocking on pscore (indicator for low weight)               0
##   Saturated regression                                        0
##   Weighted exact matching                                     0
##                                                          ate
## type                                                      -233.735110021543
##   Series Regression                                            0
##   Pscore Regression with covariates                            0
##   Pscore Regression without covariates                         0
##   Pscore weighted data                                         0
##   Blocking on pscore (continuous weight)                       0
##   Blocking on pscore (indicator for low weight)                0
```

```
##   Saturated regression                                    0
##   Weighted exact matching                                 0
##                                                      ate
## type                                                  -232.722538508639
##   Series Regression                                       0
##   Pscore Regression with covariates                       0
##   Pscore Regression without covariates                    0
##   Pscore weighted data                                    0
##   Blocking on pscore (continuous weight)                  0
##   Blocking on pscore (indicator for low weight)           0
##   Saturated regression                                    0
##   Weighted exact matching                                 0
##                                                      ate
## type                                                  -226.648004943488
##   Series Regression                                       0
##   Pscore Regression with covariates                       0
##   Pscore Regression without covariates                    0
##   Pscore weighted data                                    0
##   Blocking on pscore (continuous weight)                  0
##   Blocking on pscore (indicator for low weight)           0
##   Saturated regression                                    0
##   Weighted exact matching                                 0
##                                                      ate
## type                                                  -225.990709445074
##   Series Regression                                       0
##   Pscore Regression with covariates                       0
##   Pscore Regression without covariates                    0
##   Pscore weighted data                                    0
##   Blocking on pscore (continuous weight)                  0
##   Blocking on pscore (indicator for low weight)           0
##   Saturated regression                                    0
##   Weighted exact matching                                 0
##                                                      ate
## type                                                  0.0398429384429508
##   Series Regression                                     0
##   Pscore Regression with covariates                     0
##   Pscore Regression without covariates                  0
##   Pscore weighted data                                  0
##   Blocking on pscore (continuous weight)                0
##   Blocking on pscore (indicator for low weight)         0
##   Saturated regression                                  0
##   Weighted exact matching                               0
```

```
knitr::kable(results_df,
             caption="Treatment Effect Estimates by Method",
             col.names = c('Method Type', 'ATE', 'ATT', 'Number of variables'), align = "l", digits = 3
             format = "latex")
```

Table 5: Treatment Effect Estimates by Method

	Method Type	ATE	ATT	Number of variables
tobacco1	Series Regression	-251.113	NA	311
tobacco11	Pscore Regression with covariates	-242.014	NA	17
tobacco12	Pscore Regression without covariates	-242.886	NA	3
1	Pscore weighted data	-232.723	-250.851	NA
11	Blocking on pscore (continuous weight)	-233.735	NA	NA
12	Blocking on pscore (indicator for low weight)	0.040	NA	NA
tobacco13	Saturated regression	-226.648	NA	2494
13	Weighted exact matching	-225.991	NA	NA