ARE213 Problem Set #1B

Peter Alstone & Frank Proulx

October 17, 2013

1 Problem #1

1.1 Part A

Wrong functional form. In Problem Set 1A we used linear (i.e., $y = \beta x + \epsilon$) estimators to make "corrections" while the true functional form of the relationships between the covariates we included in the modern were certainly not linear. By imposing a linear function on a non-linear data generating process (described by the CEF), we introduce misspecification bias in the model.

Omitted Variables Bias. We were able to use variables included in the dataset in our linear model, but not the unobserved variables that may be important for control. If omitted variables exist that both determine outcomes related to birth weight and are correlated with smoking status we will over- or under-estimate the effect (depending on the characteristics of the omission).

1.2 Part B

We can attempt to reduce the magnitude of the first source of bias mentioned above (functional form) by introducing non-parametric series estimators as a replacement for linear regression. To implement this we used a natural cubic spline with two knots on the "dmar" variable (maternal age) in the regression from PS1A. The tobacco use (treatment) and marital status remain as factors. We also implemented a version of the model with interactions between the splined maternal age term and the two discrete terms. The

summary of the results are in Table 1. The ATE for the model we used in PS1A for tobacco use was 200 grams (rounded from an exact estimate of 195 grams). This is essentially unchanged with the addition splines to the maternal age relationship (an exact estimate of 199 grams). Adding interaction terms results in an ATE for tobacco use of 220 grams.

The benefits to applying splines in this case is that the regression model more closely matches the reality of the data, which show that birth weight's relationship to maternal age has a peak and is not monotonically increasing. The drawback is that the true functional form is only obscured in this approach. While the interaction terms result in an ATE that is different from the one in a non-interacting model, the interpretation becomes much more difficult. In a policymaking environment the addition of splines and interactions would represent a potential roadblock to the essential message, which remains unchanged, which is that birth weight is reduced in mothers who use tobacco (by about 200 grams).

2 Problem #2

The Propensity Score Method (PSM) uses a "surrogate" normalized metric (p-score) as a replacement for the observable controls that would normally be used to condition the estimates of a treatment response to the variable in question. The p-score is defined as a normalized score that represents the likelihood a sample selects into treatment conditioned on observables. Because it collapses all the dimensions into a 0:1 continuum PSM avoids the curse of dimensionality encountered with large nonparametric regression models, where it can be difficult to find neighbors or "bandwidth-mates" in n-dimensional space.

2.1 Part A

To calculate the propensity score, we estimated a logit model of mother's tobacco use (0=non-smoker, 1=smoker) as determined by the predetermined covariates shown in Table 2. Model #1 shows the full model using all of the covariates suspected to be predetermined. Model #2 is a reduced form of the same model, preserving just those covariates that were significant at the 1% level in Model #1.

To test whether the propensity scores predicted by these two models are

Table 1:

	Birth Weight		
	(1)	(2)	(3)
tobacco2	195.101***	199.060***	252.395***
	(4.764)	(4.774)	(84.171)
dmage	2.716***		
	(0.338)		
ns(dmage, df = 3)1		147.297***	85.607**
		(10.062)	(37.532)
ns(dmage, df = 3)2		342.494***	386.881**
		(36.880)	(177.619)
ns(dmage, df = 3)3		−10.419	−55.186
		(25.826)	(104.185)
dmar2	−146.507***	−123.962***	57.854
	(4.538)	(4.797)	(85.517)
tobacco2:ns(dmage, df = 3)1			101.145**
			(41.432)
tobacco2:ns(dmage, df = 3)2			−68.050
			(194.021)
tobacco2:ns(dmage, df = 3)3			33.670
			(109.995)
tobacco2:dmar2			−267.716***
			(94.824)
ns(dmage, df = 3)1:dmar2			−135.516***
			(50.824)
ns(dmage, df = 3)2:dmar2			−398.948*
			(214.796)
ns(dmage, df = 3)3:dmar2			−275.447*
			(165.568)
tobacco2:ns(dmage, df = 3)1:dmar2			42.062
			(59.539)
tobacco2:ns(dmage, df = 3)2:dmar2			619.892***
			(239.750)
tobacco2:ns(dmage, df = 3)3:dmar2			391.482**
			(192.669)
Constant	3,170.698***	3,040.520***	2,989.989***
	(10.921)	(16.392)	(76.157)
N	114,610	114,610	114,610
R^2	0.037	0.039	0.040
Adjusted R^2	0.037	0.039	0.040

Notes: ***Significant at the 1 percent level.
**Significant at the 5 percent level.
*Significant at the 10 percent level.

Table 2: Logistic function coefficients for propensity score models

	Mother Tobacco-Use Status	
	(1)	(2)
Mother's Race not White or Black	-1.956^{***}	-1.954^{***}
	(0.134)	(0.133)
Mother's Years of Education	-0.817^{***}	-0.818^{***}
	(0.028)	(0.028)
Marital status	-0.205^{***}	-0.204^{***}
	(0.005)	(0.005)
Father's age	-1.256^{***}	-1.251^{***}
	(0.022)	(0.021)
Father's Years of Education	0.029^{***}	0.030^{***}
	(0.002)	(0.001)
Father Mexican	-0.131^{***}	-0.131^{***}
	(0.005)	(0.005)
Father Puerto Rican	-1.961^{***}	-1.957^{***}
	(0.173)	(0.173)
Father Cuban	-1.267^{***}	-1.268^{***}
	(0.058)	(0.058)
Father Central or South American	-0.567	-0.567
	(0.364)	(0.364)
Father Race Other or Unknown Hispanic	-1.933^{***}	-1.932^{***}
	(0.205)	(0.205)
Plurality of Infant	-0.890^{***}	-0.889^{***}
	(0.120)	(0.120)
Sex of Infant	-0.148^{***}	
	(0.054)	
Mother's age	-0.019	
	(0.017)	
dmage	0.003	
	(0.002)	
Constant	2.873^{***}	2.707^{***}
	(0.088)	(0.064)
N	114,610	114,610
Log Likelihood	-44,310.690	-44,315.790
Akaike Inf. Crit.	88,651.370	88,655.580

Notes: ***Significant at the 1 percent level.
**Significant at the 5 percent level.
*Significant at the 10 percent level.

significantly different we perform a likelihood ratio test between the full and reduced model. This test yields the following output:

Likelihood ratio test for MLE method: Chi-squared 3 d.f. = 10.21274, P value = 0.01684173

The likelihood ratio test result (i.e., $p < 0.05$ but not < 0.01) indicates that the reduced-form model and full model are significantly different from each other at the 0.05 confidence level. Given this and the fact that the reduced-form model has a higher AIC, we will work with the reduced-form model of propensity scores throughout the remainder of the assignment.

2.2 Part B

In this section, we estimate a linear model controlling for the propensity score.

The estimated coefficients for this model are given in Table 3.

The estimated average treatment effect is given by

$$\delta_1 + \delta_2 \overline{p}(X_i) = -237.099 + (0.1594 * 90.427) = -222.69 \tag{1}$$

This method assumes that treatment effects are not heterogenous with respect to X_i, that treatment effects *are* heterogenous with respect to $p(X_i)$, and that we have unconfoundedness (i.e. the decision to smoke is randomly assigned, conditioned on exogenous variables).

Pursuant to these assumptions, the estimated ATE of smoking on birthweight based on the corrected regression method is a reduction of 223 grams.

2.3 Part C

We estimate the average treatment effect of smoking on birthweight to be 222 grams when using the reweighting approach. This approach involved taking a weighted average of all observations using the (normalized inverse) of the propensity score as a weighting factor. This is consistent with the estimate produced using the regression approach.

We estimate the average treatment on the treated to be 240 grams. This was performed by using the summed propensity scores as a weighting function. This suggests that (depending on relevant assumptions being met), the mothers who smoke are more affected by smoking on average in terms of infant birthweight than the general population.

Table 3: Model of effects of tobacco use on birthweight using propensity score as a control

	Mother Tobacco-Use Status
Delta1	-241.076^{***}
	(16.738)
Beta	-237.099^{***}
	(9.180)
Delta2	90.427^{***}
	(34.496)
Constant	$3,445.897^{***}$
	(3.022)
N	114,610
R^2	0.025
Adjusted R^2	0.025
Residual Std. Error	577.939
F Statistic	963.616

Notes: ***Significant at the 1 percent level.
**Significant at the 5 percent level.
*Significant at the 10 percent level.

2.4 Part D

Here we estimate the density function with a kernel density estimator, using the density() function in R. We estimate the density function separately for the smoking and non-smoking members of the sample, and weight their responses with the propensity scores normalized to the subsample (e.g. $p(X_i)/\sum j = 1^{N_{smokers}} p(X_j)$)

We estimate the density function using the Epanechnikov kernel and bandwidths ranging from 10 grams to 100 grams in increments of 10 grams. Figure 1 shows the density function estimated with a bandwidth of 40 grams. This bandwidth appears to be a good compromise between washing out some of the noise at lower bandwidths while preserving the underlying CEF.

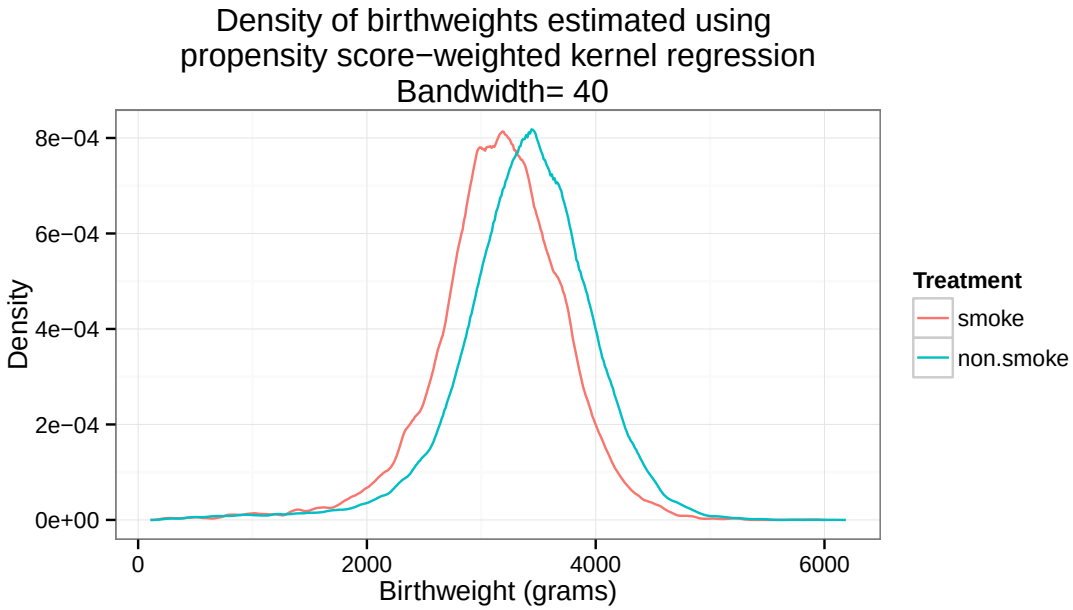

Figure 1: Birthweight density function estimates produced using Epanechnikov kernel estimator for smokers and non-smokers.

We also estimate the density at $x = 3000$ grams by hand. We used the weighting scheme by inverse propensity scores combined with an Epanechkov kernel estimator. The code almost works, and is below.

```
h <- 30

```

```
# added identity function to kernel (cuts off outside bw)
kernel.epa <- function(u){
  if(abs(u)<1){return(0.75*(1-u*u))
  }else{return(0)}
}

smokelist <- with(ps1.data.clean, (which(tobacco.rescale == 1)))
nonsmokelist <- with(ps1.data.clean, (which(tobacco.rescale == 0)))

data.sm <- with(ps1.data.clean[smokelist,], data.frame(birth.weight = dbrwt,
     weight = 1/propensityreduced))

data.ns <- with(ps1.data.clean[nonsmokelist,], data.frame(birth.weight =
    dbrwt, weight = 1/(1-propensityreduced)))

ps3k.sm <- with(ps1.data.clean, mean(propensityreduced[which(dbrwt == 3000 &
     tobacco.rescale == 1)]))
ps3k.ns <- with(ps1.data.clean, mean(propensityreduced[which(dbrwt == 3000 &
     tobacco.rescale == 0)]))

# find smoking at 3000

# find vector of contributions to kernel weight, weighted by inverse p-score
for(i in 1:length(data.sm$birth.weight)){
  data.sm$ksum[i] <- data.sm$weight[i] * kernel.epa((data.sm$birth.weight[i]
       - 3000) / h)
}

reasonable.weight <- which(data.sm$weight < 10)

# sum contributions and divide by number of obs.,
kernel.3000.sm <- sum(data.sm$ksum) / (sum(data.sm$weight[reasonable.weight
    ]) * length(data.sm$ksum) * h)

# this almost works but doesn't quite match the plots.  :(
```

kd–by–hand.R

2.5 Part E

Figure 2 shows how the character of the kernel density estimator is effected by the choice of bandwidth for the problem at hand. While all the kernel densities tend to tell the same story, those with "middle" bandwidth are a good balance between the choppy nature of small bandwidth and over-smoothing of large bandwidth.

2.6 Part F

The weighting approach used in part (c) deals with the fact that some people have a higher likelihood of selecting into treatment by weighting the outcomes

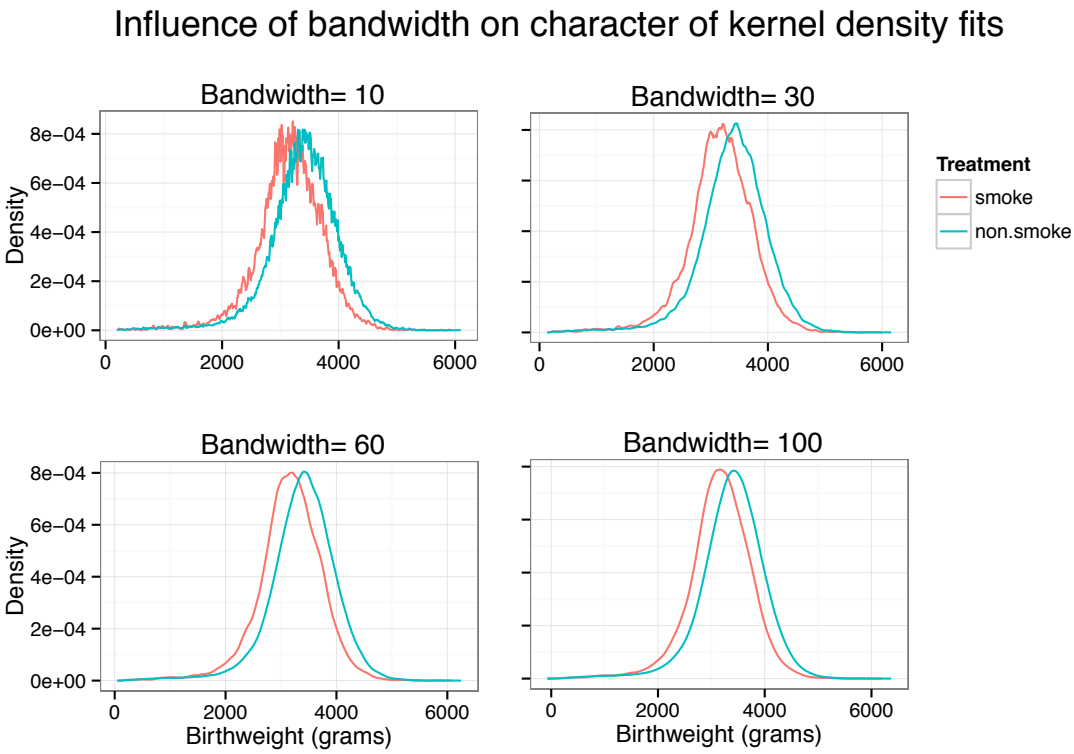

Figure 2: Comparison of bandwidth choices for birthweight density function estimates produced using Epanechnikov kernel estimator for smokers and non-smokers.

of the treated individuals by the inverse of their propensity for the treatment. In essence, this means that the more likely a treated individual would be to select into treatment (based on the estimated propensity score, i.e. "are they behaving how we expect them to behave in terms of choosing to smoke?"), the less weight their outcome receives, and vice versa. In this way, this approach is supposed to normalize out individuals' self-selection into treatment against the likelihood that they would opt in. This is desirable because it provides a means of estimating the unobserved counterfactual outcomes.

The potential downside to the weighting scheme is that outlier values (very high or low propensity scores) for those who went "against" their propensity (i.e., for a low score, they selected to treatment or vice versa)

will have very high weight (approaching infinity in the limit). It is possible to mitigate this issue by trimming the data to only consider people with propensity scores between 10 % and 90%, for instance.

2.7 Part G

Using a range of propensity scoring methods we find that the average effect of smoking on birth weight is approximately 220 grams (lower birth weights for mothers who smoke). Both regression adjustment and reweighting methods resulted in essentially the same result, which matches the result of using cubic splines and interaction in a regression model. These findings, controlling for self-selection, result consistently across methods in an ATE 10% higher than in the simple linear case presented in Problem Set 1A.

The estimates produced in part (b) depend on the assumptions that the treatment effect heterogeneity is linear in the propensity score, that the treatment effect heterogeneity is *not* linear in the covariates, and that the decision to smoke is randomly assigned conditional on the exogenous variables. The estimates produced in part (c) also assume that the decision to smoke is randomly assigned conditional on the exogenous variables (unconfoundedness) but do not need assume that the treatment effect heterogeneity is linear in the propensity score.

3 Problem # 3

Here we use a blocking approach to estimate the Average Treatment Effect. This entails subdividing all of the observations into evenly spaced blocks based on propensity scores. The average treatment effect is estimated within each blocks by calculating the average birthweights of children to smokers and to non-smokers separately and taking the difference. Any blocks with 0 smokers or 0 non-smokers are discarded, as no "match" has been made here. Finally, the average treatment across blocks is taken, weighting each block based on the number of observations within the block. The block sizes produced using this method are displayed in Figure 3. In this plot, the histograms are plotted on top of each other to demonstrate how the smokers and non-smokers are distributed with respect to each other.

The Average Treatment Effect estimated using the blocking method is 217 grams. This is again very close to the other values that we have estimated.

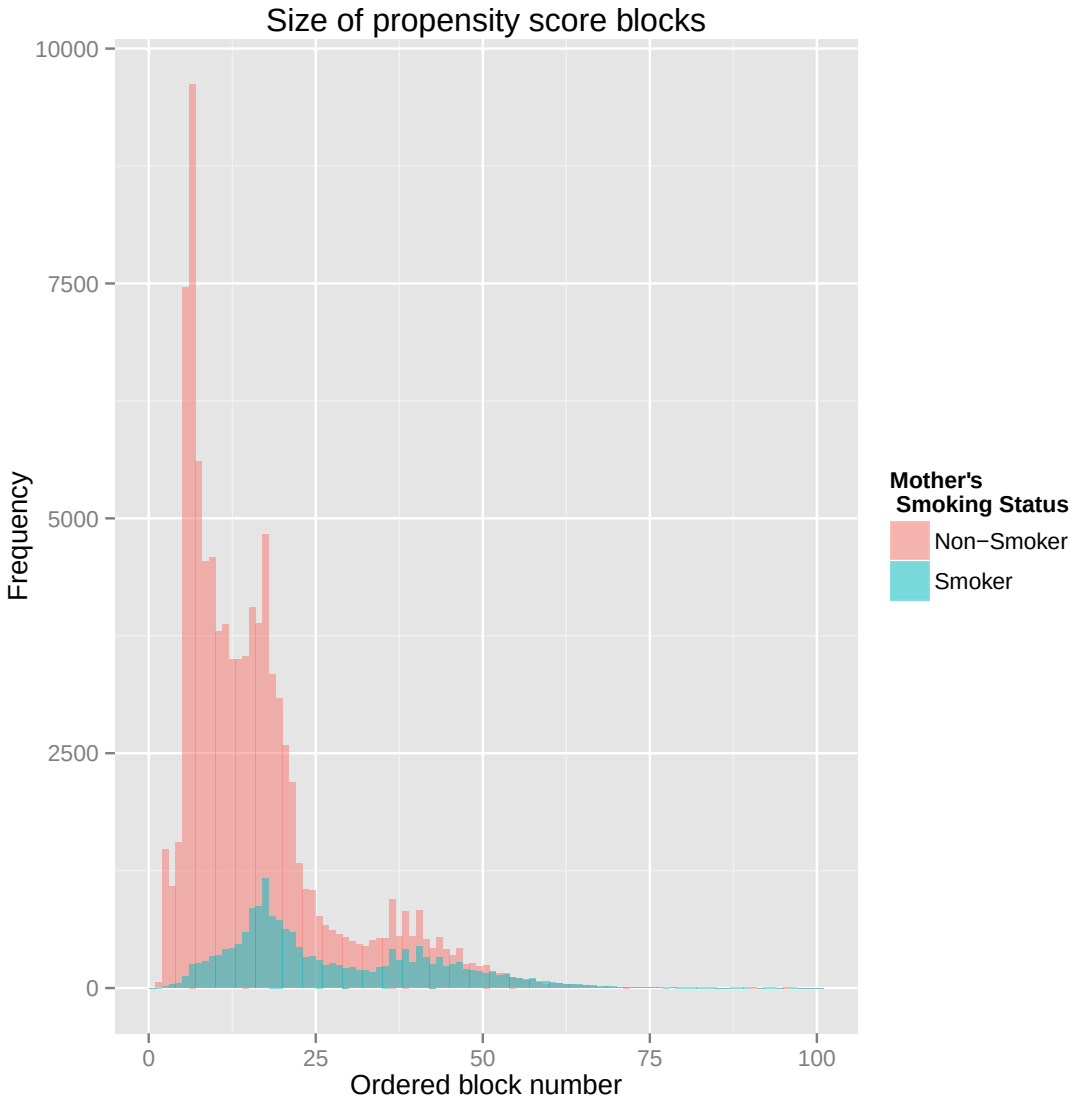

Figure 3: Overlaid plot of block group sizes, before cleaning.

4 Problem #4

Because low birth weights (¡2500 grams) are of particular concern, we again use the blocking method to estimate the Average Treatment Effect of smoking on the probability of having a low birth weight. The steps taken are very

similar to those taken in Problem #3. However, now we have calculated an indicator variable for every observation in the dataset corresponding to low birth weights. Within each block, we take the proportion of smokers and non-smokers who have given birth to low birth weight infants as the probability of this occuring. The within block treatment effect is thus the difference in probability of the low birth-weight event occuring. Using this method, we estimate an Average Treatment Effect of smoking on low birthweight of 3.78%, which is to say that smokers are 3.78% more likely than non-smokers to produce babies in the low birth weight range.

5 Problem #5

We explored a dataset describing the birth outcomes for babies born in 1993 in Pennsylvania using the National Natality Data Files. The effect we were exploring was the impact of maternal smoking on health outcomes, primarily with a focus on birth weight and APGAR scores, two metrics that are good indicators for overall natal health. For the purposes of this work we take the "treatment" to be smoking during the pregnancy only, so any covariate that occurs before the pregnancy is thought to be predetermined and not effected by the incidence of smoking. Covariates that have to do with the pregnancy (such as visits to the Doctor) are confounded by the treatment.

In PS1A we used the linear model to explore the data and found that APGAR scores were not sensitive to tobacco use in a significant way. Birth weight, however, showed a trend. The agnostic regression of differences in means without any accounting for covariates resulted in an ATE of 240 grams reduced birth weight for mothers who smoke. There are, however, several predetermined covariates that we found were important to correct for, namely maternal age and marital status (others were considered but these two were the most important). After correction for these factors (without interaction) the ATE is estimated at 200 grams.

In PS1B we used non-parametric methods to further explore the data and create more robust estimates of the ATE for smoking. Introducing a spline model for the maternal age - birth weight relationship did not result in a different ATE from the best one using parametric methods but adding interaction terms to the spline model did change the estimated ATE, from 200 to 220 grams. Using a range of propensity score matching methods we confirmed that 220 grams is a reasonable estimate for the ATE and also

estimate that the likelihood of very low birth weight is increased by 4 % by smoking.

Overall, the best estimate of ATE smoking is 220 grams, which was the result from a range of non-parametric approaches. These approaches allow us to make a reasonable estimate of what the ATE would be if smoking were randomly assigned (which it is not). The assumptions required for the non-parametric approaches (summarized above) are reasonable given the dataset. The best estimate, however, is probably 200 grams. This includes one significant figure and is confirmed by every statistical approach we took. For the purposes of public health policymaking this estimate has a reasonable level of precision.

6 Code

We used R to complete this assignment. The code is below:

```
# PROBLEM SET 1B
# ARE 213 Fall 2013
## TO DO:
# Figure out ATT in 2c
# Get a legend on the kerndensity plots to make the "beautiful and
    publication ready" DONE
# Figure the kernel regression by hand problem
# problem 5
# writing up a number of problems (I will get more done on this in the
    morning - need to go take a midterm).

# Frank's Directory
#setwd("/media/frank/Data/documents/school/berkeley/fall13/are213/are213/ps1
    ")

# Peter's Directory
#setwd("~/Google Drive/ERG/Classes/ARE213/are213/ps1")


# Packages --------
library(foreign) #this is to read in Stata data
library(Hmisc)
library(psych)
library(stargazer)
library(ggplot2) # for neato plotting tools
library(plyr) # for nice data tools like ddply
library(epicalc) # For likelihood ratio test
library(car) # "companion for applied regression" - recode fxn, etc.
library(gmodels) #for Crosstabs
library(splines) # for series regression
library(np) #nonparametric regression
library(rms) #regression modeling tools
library(effects)
```

```
library(reshape)

# Homebrewed functions
source("../util/are213-func.R")
source("../util/watercolor.R") # for watercolor plots

# Data -------
ps1.data <- read.dta(file="ps1.dta")

var.labels <- attr(ps1.data, "var.labels")

# Data Cleaning Step
full.record.flag <- which(ps1.data$cardiac != 9 &
                            ps1.data$cardiac != 8 &
                            ps1.data$lung != 9 &
                            ps1.data$lung != 8 &
                            ps1.data$diabetes !=9 &
                            ps1.data$diabetes !=8 &
                            ps1.data$herpes != 9 &
                            ps1.data$herpes != 8 &
                            ps1.data$chyper != 9 &
                            ps1.data$chyper != 8 &
                            ps1.data$phyper != 9 &
                            ps1.data$phyper != 8 &
                            ps1.data$pre4000 !=9 &
                            ps1.data$pre4000 !=8 &
                            ps1.data$preterm != 9 &
                            ps1.data$preterm != 8 &
                            ps1.data$tobacco != 9 &
                            ps1.data$cigar != 99 &
                            ps1.data$cigar6 !=6 &
                            ps1.data$alcohol != 9 &
                            ps1.data$drink != 99 &
                            ps1.data$drink5 !=5 &
                            ps1.data$wgain !=99
)

ps1.data$full.record <- FALSE # initialize column as F
ps1.data$full.record[full.record.flag] <- TRUE #reassign level to T for full
     records

ps1.data.clean <- subset (ps1.data, full.record == TRUE)
ps1.data.missingvalues <- subset(ps1.data, full.record == FALSE)

# Problem 1a : Describes PS1a results. -------
# Problem 1b --------------
# This is using a series estimator. I think smooth.spline() is the right
    function to use, but let me know if you think we should be doing kernel
    regression instead. I'm also not sure how to go about adding interaction
     terms.  I think a kernel regression is more appropriate here...mostly
    because I don't know the spline function and there seems to be a good
    package ("np") for running kernel regression.

# SPLINE FIT FOR # CIGS

# sm.flex <- with(ps1.data.clean, smooth.spline(cigar, y=dbrwt, nknots=10,
    spar = 0.7, tol = 0.0001)) # Fits a smooth line to the data
```

```
# sm.flex.df <- data.frame(sm.flex$x, sm.flex$y) #converts the fitted values into a data frame for ggplot
#
# splineplot <- ggplot(sm.flex.df, aes(x = sm.flex.x, y=sm.flex.y))
# splineplot <- splineplot +
#   geom_point(data=ps1.data.clean, aes(x = cigar, y = dbrwt), pch = 1) +
#   geom_line(color='red') +
#   labs(x = 'Cigarettes smoked per day by mother', y= 'Birthweight')
# splineplot
#
# ggsave(filename = 'img/splineplot.pdf')

# Using Series estimator with splines on maternal age.

ps1.data.clean$tobacco <- as.factor(ps1.data.clean$tobacco)
ps1.data.clean$dmar <- as.factor(ps1.data.clean$dmar)

wsp.ps1a <- lm(dbrwt ~ tobacco + dmage + dmar, data=ps1.data.clean)
wsp <- lm(dbrwt ~ tobacco + ns(dmage, df=3) + dmar, data=ps1.data.clean)
wsp.int <- lm(dbrwt ~ tobacco * ns(dmage, df=3) * dmar, data=ps1.data.clean)

stargazer(wsp.ps1a, wsp, wsp.int, out = 'splineresults.tex', style="qje", label = 'tab:splineresults', no.space = TRUE, dep.var.labels = "Birth Weight")



# This is the ATE with a splines regression on Age
summary(effect("tobacco", wsp))

# This is the ATE with a complex, interacting splines regression on AGe
summary(effect("tobacco", wsp.int))

# Problem 2a --------
ps1.data.clean$tobacco.rescale <- with(ps1.data.clean, recode(tobacco, "2='0'", as.numeric.result=TRUE)) #rescales the tobacco use variable to be 0/1, where 0=no and 1 = yes
ps1.data.clean$dmar.rescale <- with(ps1.data.clean, recode(dmar, "2='0'"))

smoke.propensity.all <- glm(tobacco.rescale ~ as.factor(mrace3) + dmeduc + dmar.rescale + dfage + dfeduc + as.factor(orfath) + dplural + csex + dmage, data=ps1.data.clean, family = binomial()) ## Did I miss any predetermined covariates here? No.

smoke.propensity.reduced <- glm(tobacco.rescale ~ as.factor(mrace3) + dmeduc + dmar.rescale + dfage + dfeduc + as.factor(orfath), data=ps1.data.clean, family = binomial())


stargazer(smoke.propensity.all, smoke.propensity.reduced,
          type = "latex",
          covariate.labels = c("Mother's Race not White or Black", "Mother's Years of Education", "Marital status", "Father's age", "Father's Years of Education", "Father Mexican", "Father Puerto Rican", "Father Cuban", "Father Central or South American", "Father Race Other or Unknown Hispanic", "Plurality of Infant", "Sex of Infant", "Mother's age"),
```

```
          style ="qje",
          align = TRUE,
          label = "tab:propensities",
          title = "Logistic function coefficients for propensity score
              models",
          dep.var.labels = "Mother Tobacco-Use Status",
          no.space = TRUE,
          out = "propensityscores.tex"
           )

ps1.data.clean$propensityfull <- predict(smoke.propensity.all, type = "
    response")
ps1.data.clean$propensityreduced <- predict(smoke.propensity.reduced, type =
     "response")

detach("package:rms")
sink(file = "lrtest.tex", append = FALSE)
lrtest(smoke.propensity.all, smoke.propensity.reduced) #Test whether the two
     scores are statistically different
sink()

require(rms)

#Problem 2b - Estimating a regression model using propensity scores --------

sm.propensityregression <- lm(dbrwt ~ propensityreduced * tobacco.rescale,
    ps1.data.clean)

#calculation of average treatment effect:
coefficients(sm.propensityregression)[2] + coefficients(sm.
    propensityregression)[4]*mean(ps1.data.clean$propensityreduced)

tobacco.effects <- (effect("tobacco.rescale", sm.propensityregression))

stargazer(sm.propensityregression,
          type = "latex",
          covariate.labels = c("Delta1", "Beta", "Delta2", "Constant"),
          style ="qje",
          align = TRUE,
          font.size="footnotesize",
          label = "tab:propensitymodel",
          title = "Model of effects of tobacco use on birthweight using
              propensity score as a control",
          dep.var.labels = "Mother Tobacco-Use Status",
          out = "propensityscoremodel.tex"
)

#Problem 2c - Using reweighting with propensity scores --------

ps1.data.clean$tobacco.rescale.n <- as.numeric(levels(ps1.data.clean$tobacco
    .rescale))[ps1.data.clean$tobacco.rescale]

ps1.data.clean$weightingterm1 <- with(ps1.data.clean, ((tobacco.rescale.n*
    dbrwt)/propensityreduced))
ps1.data.clean$weightingterm2 <- with(ps1.data.clean, (((1-tobacco.rescale.n
    )*dbrwt)/(1-propensityreduced)))

```

```
weightingestimator <- with(ps1.data.clean, sum((weightingterm1/sum(tobacco.rescale.n/propensityreduced))-(weightingterm2/sum((1-tobacco.rescale.n)/(1-propensityreduced))))) #This should be the average treatment effect

ATTweight <- with(ps1.data.clean, sum(((propensityreduced*weightingterm1)/sum(propensityreduced))-(((1-propensityreduced)*weightingterm2)/sum(1-propensityreduced))))

# Problem 2d - Kernel Density Estimator
tot.propensity.nosm <- with(subset(ps1.data.clean, tobacco.rescale == 0), sum(propensityreduced))
tot.propensity.sm <- with(subset(ps1.data.clean, tobacco.rescale == 1), sum(propensityreduced))

kd.plot.fn <- function(h, plot.w = 5, plot.h = 3){
kerndensity.nosm <- with(subset(ps1.data.clean, tobacco.rescale == 0),
                         density(dbrwt, #if nobody smoked
                                 kernel = "epanechnikov",
                                 bw = h,
                                 weights = propensityreduced/tot.propensity.nosm))
kerndensity.nosm.df <- data.frame(birth.weight = kerndensity.nosm$x, non.smoke = kerndensity.nosm$y)

kerndensity.sm <- with(subset(ps1.data.clean, tobacco.rescale == 1),
                       density(dbrwt, #if everybody smoked
                               kernel = "epanechnikov",
                               bw = h,
                              weights = propensityreduced/tot.propensity.sm)
                                )
kerndensity.sm.df <- data.frame(birth.weight = kerndensity.sm$x, smoke = kerndensity.sm$y)

kdens <- join(kerndensity.sm.df, kerndensity.nosm.df, by="birth.weight", type = "full")

kdens.m <- melt.data.frame(kdens, id.vars="birth.weight", measure.vars = c("smoke", "non.smoke"), na.rm = TRUE)

kd.plot <- ggplot(kdens.m, aes(birth.weight, value, factor(variable)))
kd.plot <- kd.plot +
  geom_line(aes(color=variable)) +
  labs(title = paste("Density of birthweights estimated using \n propensity score-weighted kernel regression \n Bandwidth=", as.factor(h)), x = "Birthweight (grams)", y = "Density") +
  guides(color = guide_legend(title = "Treatment")) +
  theme_bw()


kd.plot

ggsave(width = plot.w, height = plot.h, file = paste0('img/kerndensity', h,'.pdf'), plot = kd.plot)

}
```

```
212
213
214 ##Problem 2d - calculating kernel value by 'hand' at dbrwt = 3000 --------
215 ##I can't figure out what to do here. Most of this is probably wrong but
        maybe something is right. Want to take a whack?
216
217
218 h <- 30
219
220 # added identity function to kernel (cuts off outside bw)
221 kernel.epa <- function(u){
222   if(abs(u)<1){return(0.75*(1-u*u))
223   }else{return(0)}
224 }
225
226 propensity3000.sm <- with(ps1.data.clean, mean(propensityreduced[which(dbrwt
        == 3000 & tobacco.rescale == 1)]))
227 propensity3000.nosm <- with(ps1.data.clean, mean(propensityreduced[which(
        dbrwt == 3000 & tobacco.rescale == 0)]))
228 ##for(i in 1:nrow(subset(ps1.data.clean, tobacco.rescale == 1))){
229 ##with(subset(ps1.data.clean, tobacco.rescale == 1),
230 ##     kern3000.sm.num <- kern3000.sm.num +
231 ##      kernel.epa(((propensity3000.sm-propensityreduced[i])/h)*dbrwt))
232 ## with(subset(ps1.data.clean, tobacco.rescale == 1),
233 ##      kern3000.sm.den <- kern3000.sm.den +
234 ##      kernel.epa((propensity3000.sm-propensityreduced[i])/h))
235 ## }
236 ## kern3000.sm <- kern3000.sm.num / kern3000.sm.den
237
238
239 ## kernel3000.sm <- with(subset(ps1.data.clean,tobacco.rescale == 1), data.
        frame(window = (3000 - dbrwt/h)))
240 ## kernel3000.sm$numerator <- with(subset(ps1.data.clean, tobacco.rescale ==
         1), kernel.epa(((3000/propensity3000.sm) - (dbrwt/propensityreduced))/h
        ))
241 ## kernel3000.sm$denominator <- with(subset(ps1.data.clean, tobacco.rescale
        == 1), kernel.epa(((3000/propensity3000.sm) - (dbrwt/propensityreduced))
        /h))
242
243 ## with(kernel3000.sm[window < 1 & window > -1], sum(numerator))/(nrow(
        kernel3000.sm[abs(window < 1)])*h)
244
245 #Problem 2e --------
246
247 for(bw in seq(10,100,10)){kd.plot.fn(bw)}
248
249 kd.plot.fn(40,7,4)
250
251
252
253 ### Problem 3
254 ## Using blocking estimator
255 # Divide smokers into ~100 equally spaced blocks
256 prop.max <- with(ps1.data.clean, max(propensityreduced))
257 prop.min <- with(ps1.data.clean, min(propensityreduced))
258 prop.binsize <- (prop.max - prop.min)/99
259
```

```
ps1.data.clean$blocknumber <- with(ps1.data.clean,
                                   round(propensityreduced/prop.binsize,
                                       digits = 0) + 1)

blocktreatmenteffects <- ddply(ps1.data.clean, .(blocknumber), summarize,
    smokers = sum(tobacco.rescale == 1), nonsmokers = sum(tobacco.rescale ==
     0), smokerdbrwt = mean(dbrwt[tobacco.rescale == 1]), nonsmokerdbrwt =
    mean(dbrwt[tobacco.rescale == 0]))

blocktreatmenteffects$badbin <- with(blocktreatmenteffects, as.numeric(
    smokers == 0 | nonsmokers == 0))

cleaned.blocks <- subset(blocktreatmenteffects, badbin == 0)
cleaned.blocks$avgtreatmenteffect <- with(cleaned.blocks, smokerdbrwt -
    nonsmokerdbrwt)
cleaned.blocks$weight <- with(cleaned.blocks, (smokers + nonsmokers)/sum(
    smokers + nonsmokers))
cleaned.blocks$weightedTE <- with(cleaned.blocks, weight *
    avgtreatmenteffect)

blocks.plot <- ggplot(data=ps1.data.clean, aes(x=blocknumber, fill = tobacco
    .rescale))
blocks.plot <- blocks.plot +
    geom_histogram(binwidth = 1, alpha = 0.5, position = "identity") +
    labs(title = 'Size of propensity score blocks', x = 'Ordered block
        number', y = 'Frequency') +
    scale_fill_discrete(name = "Mother's \n Smoking Status",
                        breaks = c("0", "1"),
                        labels = c("Non-Smoker", "Smoker"))
ggsave(filename = 'img/blockplot.pdf', plot=blocks.plot)

blocksATE <- sum(cleaned.blocks$weightedTE)

### Problem 4
ps1.data.clean$lowbrwt <- as.numeric(ps1.data.clean$dbrwt < 2500)

blocklowbrwt <- ddply(ps1.data.clean, .(blocknumber), summarize, smokers =
    sum(tobacco.rescale == 1), nonsmokers = sum(tobacco.rescale == 0),
    lowbrwtprob.sm = mean(lowbrwt[tobacco.rescale == 1]), lowbrwtprob.nosm =
     mean(lowbrwt[tobacco.rescale == 0]))

blocklowbrwt$badbin <- with(blocklowbrwt, as.numeric(smokers == 0 |
    nonsmokers == 0))

cleaned.blocks.lowbrwt <- subset(blocklowbrwt, badbin == 0)
cleaned.blocks.lowbrwt$ATE <- with(cleaned.blocks.lowbrwt, lowbrwtprob.sm -
    lowbrwtprob.nosm)
cleaned.blocks.lowbrwt$weight <- with(cleaned.blocks.lowbrwt, (smokers +
    nonsmokers)/sum(smokers + nonsmokers))
cleaned.blocks.lowbrwt$weightedTE <- with(cleaned.blocks.lowbrwt, weight *
    ATE)

blocks.lowbrwt.ATE <- sum(cleaned.blocks.lowbrwt$weightedTE)



### Output values that need to be typed in to TeX:
```

```
print(paste("The estimated average treatment effect using the reweighting approach is", round(weightingestimator, digits=0)))
print(paste("The estimated average treatment on the treated using the reweighting approach is", round(ATTweight, digits = 0)))
print(paste("ATE is", round(tobacco.effects$fit[1] - tobacco.effects$fit[2], digits=0), "based on regression adjustment with p-score."))
print(paste("The Average Treatment Effect predicted by the blocking method with birthweight treated as a continuous variable is", round(blocksATE, digits=0)))
print(paste("The Average Treatment Effect predicted by the blocking method of birthweights falling into the 'low' category of less than 2500 grams is a probability of", round(blocks.lowbrwt.ATE, digits = 4),". That is, smokers are approximately", round(100*blocks.lowbrwt.ATE, digits = 0), " percent more likely to have babies with weights less than 2500 grams."))
```

ps1b.R

```
# Econometrics helper functions for [R]
#
# Peter Alstone and Frank Proulx
# 2013
# version 1
# contact: peter.alstone AT gmail.com

# Category: Data Management -------------


# Category: Data Analysis ----------------

# Function: Find adjusted R^2 for subset of data
# This requires a completed linear model...pull out the relevant y-values and residuals and feed them to function
# [TODO @Peter] Improve function so it can simply evaluate lm or glm object, add error handling, general clean up.
adjr2 <- function(y,resid){
  r2 <- 1-sum(resid^2) / sum((y-mean(y))^2)
  return(r2)
} #end adjr2


# Category: Plots and Graphics -------------

## Function for arranging ggplots. use png(); arrange(p1, p2, ncol=1); dev.off() to save.
require(grid)
vp.layout <- function(x, y) viewport(layout.pos.row=x, layout.pos.col=y)
arrange_ggplot2 <- function(..., nrow=NULL, ncol=NULL, as.table=FALSE) {
  dots <- list(...)
  n <- length(dots)
  if(is.null(nrow) & is.null(ncol)) { nrow = floor(n/2) ; ncol = ceiling(n/nrow)}
  if(is.null(nrow)) { nrow = ceiling(n/ncol)}
  if(is.null(ncol)) { ncol = ceiling(n/nrow)}
  ## NOTE see n2mfrow in grDevices for possible alternative
  grid.newpage()
  pushViewport(viewport(layout=grid.layout(nrow,ncol) ) )
  ii.p <- 1
```

```
  for(ii.row in seq(1, nrow)){
    ii.table.row <- ii.row
    if(as.table) {ii.table.row <- nrow - ii.table.row + 1}
    for(ii.col in seq(1, ncol)){
      ii.table <- ii.p
      if(ii.p > n) break
      print(dots[[ii.table]], vp=vp.layout(ii.table.row, ii.col))
      ii.p <- ii.p + 1
    }
  }
}
```

../util/are213–func.R