

ARE213 Problem Set #1B

Peter Alstone & Frank Proulx

October 17, 2013

1 Problem #1

1.1 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?

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 model 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

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?

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.

Table 1: Comparison of three linear models for birth weight

	Birth Weight		
	1) PS1A LM	2) 1+spline	3) 2+interaction
tobacco2	195.101*** (4.764)	199.060*** (4.774)	252.395*** (84.171)
dmage	2.716*** (0.338)		
ns(dmage, df = 3)1		147.297*** (10.062)	85.607** (37.532)
ns(dmage, df = 3)2		342.494*** (36.880)	386.881** (177.619)
ns(dmage, df = 3)3		-10.419 (25.826)	-55.186 (104.185)
dmar2	-146.507*** (4.538)	-123.962*** (4.797)	57.854 (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*** (10.921)	3,040.520*** (16.392)	2,989.989*** (76.157)
N	3 114,610	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.

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 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 is sufficient for our purposes.

2.2 Part B

This estimation assumes unconfoundedness and a constant treatment effect.

The estimated average treatment effect is given by

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

This suggests that (pursuant to these assumptions) the average effect of smoking on birthweight 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

Table 2: Logistic function coefficients for propensity score models

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

Notes:

5

***Significant at the 1 percent level.

**Significant at the 5 percent level.

*Significant at the 10 percent level.

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
<i>Notes:</i>	***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.

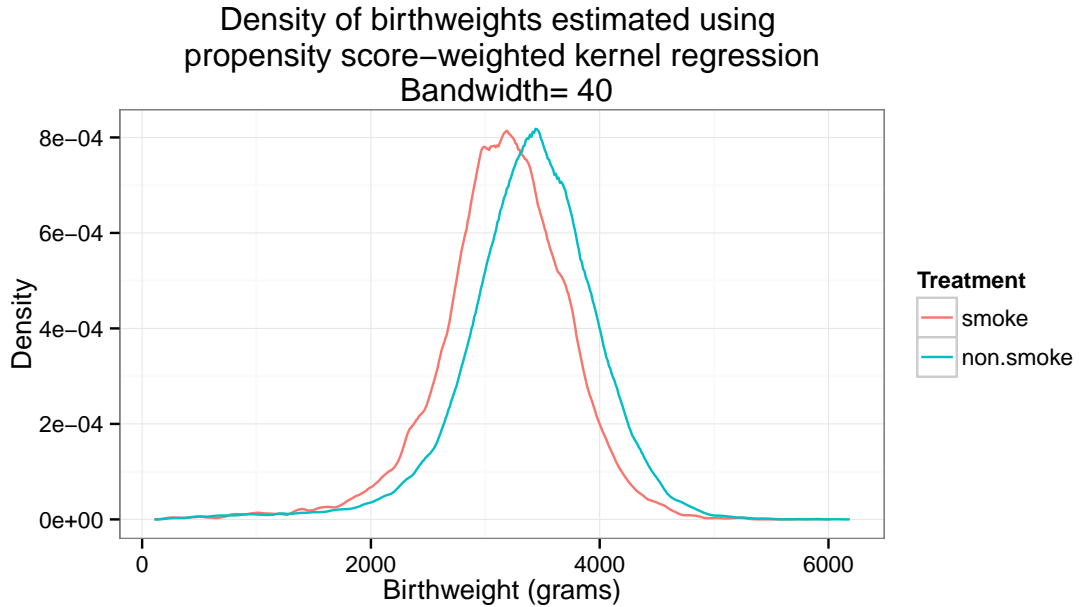


Figure 1: Birthweight density function estimates produced using Epanechnikov kernel estimator for smokers and non-smokers.

We also estimate the density at $x=3000$ grams by hand. We first calculate the average propensity score for both smokers and non-smokers with infants at 3000 grams. The dataset has 12 smokers and 47 non-smokers with infants born at this weight. The average propensity score for these smokers is 0.27 and for non-smokers it is 0.20. These values stand to reason- the people who

have opted in to smoking in this class have been predicted as more likely to be smokers.

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.

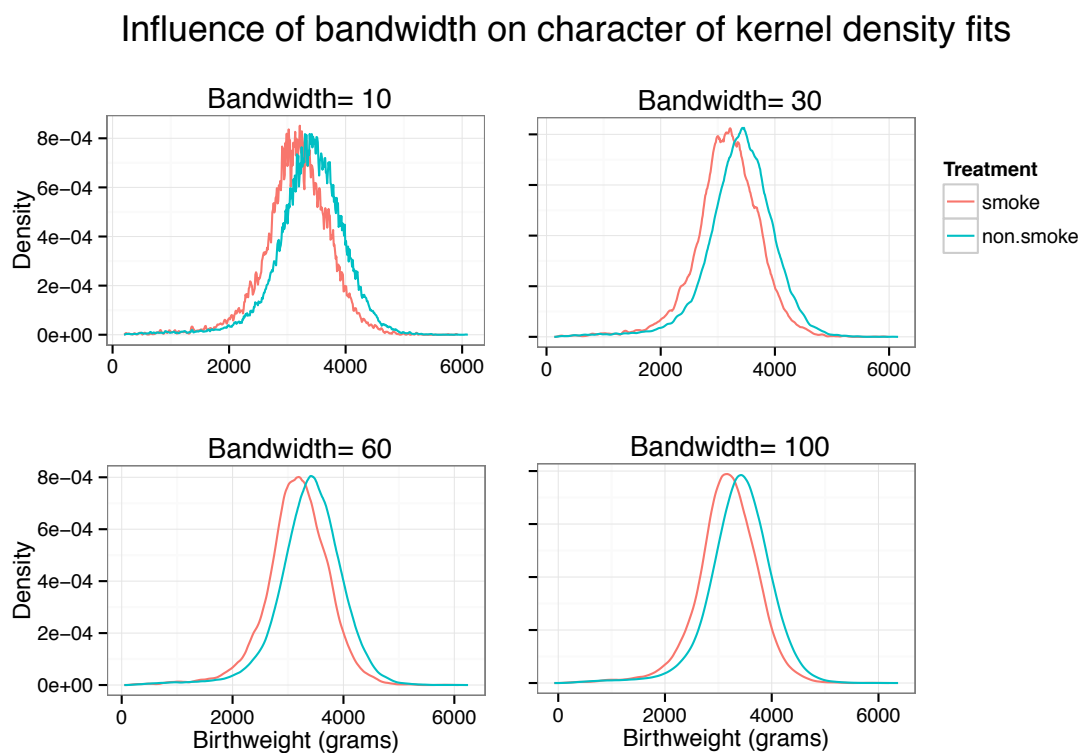


Figure 2: Comparison of bandwidth choices for birthweight density function estimates produced using Epanechnikov kernel estimator for smokers and non-smokers.

2.6 Part F

The weighting approach in Part C offers a way to use propensity scores in a regression model, so it is possible to “tell a story” with the results that is accessible to people who are comfortable with regression models. TODO: Add text here.

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 220 grams (lower weight for mothers who smoke). Both regression adjustment and reweighing methods resulted in essentially the same result, which matches the result of using cubic splines and interaction in a regression model.

TODO: Finish writing interpretation.

3 Problem # 3

4 Code

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

```
1 # PROBLEM SET 1B
2 # ARE 213 Fall 2013
3 ## TO DO:
4 # Figure out ATT in 2c
5 # Get a legend on the kerndensity plots to make the "beautiful and
  publication ready" DONE
6 # Figure the kernel regression by hand problem
7 # problem 5
8 # writing up a number of problems (I will get more done on this in the
  morning - need to go take a midterm).
9
10 # Frank's Directory
11 #setwd("/media/frank/Data/documents/school/berkeley/fall13/are213/are213/ps1
  ")
12
```

```

13 # Peter's Directory
14 #setwd("~/Google Drive/ERG/Classes/ARE213/are213/ps1")
15
16
17 # Packages -----
18 library(foreign) #this is to read in Stata data
19 library(Hmisc)
20 library(psych)
21 library(stargazer)
22 library(ggplot2) # for neato plotting tools
23 library(plyr) # for nice data tools like ddply
24 library(epicalc) # For likelihood ratio test
25 library(car) # "companion for applied regression" - recode fxn, etc.
26 library(gmodels) #for Crosstabs
27 library(splines) # for series regression
28 library(np) #nonparametric regression
29 library(rms) #regression modeling tools
30 library(effects)
31 library(reshape)
32
33 # Homebrewed functions
34 source("../util/are213-func.R")
35 source("../util/watercolor.R") # for watercolor plots
36
37 # Data -----
38 ps1.data <- read.dta(file="ps1.dta")
39
40 var.labels <- attr(ps1.data, "var.labels")
41
42 # Data Cleaning Step
43 full.record.flag <- which(ps1.data$cardiac != 9 &
44                           ps1.data$cardiac != 8 &
45                           ps1.data$lung != 9 &
46                           ps1.data$lung != 8 &
47                           ps1.data$diabetes !=9 &
48                           ps1.data$diabetes !=8 &
49                           ps1.data$herpes != 9 &
50                           ps1.data$herpes != 8 &
51                           ps1.data$chyper != 9 &
52                           ps1.data$chyper != 8 &
53                           ps1.data$phyper != 9 &
54                           ps1.data$phyper != 8 &
55                           ps1.data$pre4000 !=9 &
56                           ps1.data$pre4000 !=8 &
57                           ps1.data$preterm != 9 &
58                           ps1.data$preterm != 8 &
59                           ps1.data$tobacco != 9 &
60                           ps1.data$cigar != 99 &
61                           ps1.data$cigar6 !=6 &
62                           ps1.data$alcohol != 9 &
63                           ps1.data$drink != 99 &
64                           ps1.data$drink5 !=5 &
65                           ps1.data$wgain !=99
66 )
67
68 ps1.data$full.record <- FALSE # initialize column as F
69 ps1.data$full.record[full.record.flag] <- TRUE #reassign level to T for full

```

```

70         records
71 ps1.data.clean <- subset (ps1.data, full.record == TRUE)
72 ps1.data.missingvalues <- subset(ps1.data, full.record == FALSE)
73
74 # Problem 1a : Describes PS1a results. -----
75 # Problem 1b -----
76 # 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.
77
78 # SPLINE FIT FOR # CIGS
79
80 # 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
81 # sm.flex.df <- data.frame(sm.flex$x, sm.flex$y) #converts the fitted values
    into a data frame for ggplot
82 #
83 # splineplot <- ggplot(sm.flex.df, aes(x = sm.flex.x, y=sm.flex.y))
84 # splineplot <- splineplot +
85 #   geom_point(data=ps1.data.clean, aes(x = cigar, y = dbrwt), pch = 1) +
86 #   geom_line(color='red') +
87 #   labs(x = 'Cigarettes smoked per day by mother', y= 'Birthweight')
88 # splineplot
89 #
90 # ggsave(filename = 'img/splineplot.pdf')
91
92 # Using Series estimator with splines on maternal age.
93
94 ps1.data.clean$tobacco <- as.factor(ps1.data.clean$tobacco)
95 ps1.data.clean$dmr <- as.factor(ps1.data.clean$dmr)
96
97 wsp.ps1a <- lm(dbrwt ~ tobacco + dmr + dmr, data=ps1.data.clean)
98 wsp <- lm(dbrwt ~ tobacco + ns(dmr, df=3) + dmr, data=ps1.data.clean)
99 wsp.int <- lm(dbrwt ~ tobacco * ns(dmr, df=3) * dmr, data=ps1.data.clean)
100
101 stargazer(wsp.ps1a, wsp, wsp.int, style="qje", no.space = TRUE, dep.var.
    labels = "Birth Weight")
102
103
104
105 # This is the ATE with a splines regression on Age
106 summary(effect("tobacco", wsp))
107
108 # This is the ATE with a complex, interacting splines regression on AGE
109 summary(effect("tobacco", wsp.int))
110
111 # Problem 2a -----
112 ps1.data.clean$tobacco.rescale <- with(ps1.data.clean, recode(tobacco, "
    2='0'", as.numeric.result=TRUE)) #rescales the tobacco use variable to
    be 0/1, where 0=no and 1 = yes
113 ps1.data.clean$dmr.rescale <- with(ps1.data.clean, recode(dmr, "2='0'"))
114
115 smoke.propensity.all <- glm(tobacco.rescale ~ as.factor(mrace3) + dmeduc +

```

```

116     dmar.rescale + dfage + dfeduc + as.factor(orfath) + dplural + csex +
117     dimage, data=ps1.data.clean, family = binomial()) ## Did I miss any
predetermined covariates here? No.
118
119
120 smoke.propensity.reduced <- glm(tobacco.rescale ~ as.factor(mrace3) + dmeduc
+ dmar.rescale + dfage + dfeduc + as.factor(orfath), data=ps1.data.
clean, family = binomial())
121
122
123 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"
)
130
131
132 ps1.data.clean$propensityfull <- predict(smoke.propensity.all, type = "
response")
133 ps1.data.clean$propensityreduced <- predict(smoke.propensity.reduced, type =
"response")
134
135 detach("package:rms")
136 sink(file = "lrtest.tex", append = FALSE)
137 lrtest(smoke.propensity.all, smoke.propensity.reduced) #Test whether the two
scores are statistically different
138 sink()
139
140 require(rms)
141
142 #Problem 2b - Estimating a regression model using propensity scores -----
143
144 sm.propensityregression <- lm(dbrwt ~ propensityreduced * tobacco.rescale,
ps1.data.clean)
145
146 #calculation of average treatment effect:
147 coefficients(sm.propensityregression)[2] + coefficients(sm.
propensityregression)[4]*mean(ps1.data.clean$propensityreduced)
148
149 tobacco.effects <- (effect("tobacco.rescale", sm.propensityregression))
150
151 stargazer(sm.propensityregression,
152 type = "latex",
153 covariate.labels = c("Delta1", "Beta", "Delta2", "Constant"),
154 style = "qje",
155 align = TRUE,
156 font.size="footnotesize",

```

```

157     label = "tab:propensitymodel",
158     title = "Model of effects of tobacco use on birthweight using
        propensity score as a control",
159     dep.var.labels = "Mother Tobacco-Use Status",
160     out = "propensityscoremodel.tex"
161 )
162
163 #Problem 2c - Using reweighting with propensity scores -----
164
165 ps1.data.clean$tobacco.rescale.n <- as.numeric(levels(ps1.data.clean$tobacco
    .rescale))[ps1.data.clean$tobacco.rescale]
166
167 term1 <- with(ps1.data.clean, sum((tobacco.rescale.n*dbrwt)/
    propensityreduced)/sum(tobacco.rescale.n/propensityreduced))
168 term2 <- with(ps1.data.clean, sum(((1-tobacco.rescale.n)*dbrwt)/(1-
    propensityreduced))/sum((1-tobacco.rescale.n)/(1-propensityreduced)))
169
170 weightingestimator <- term1-term2 #This should be the average treatment
    effect
171
172 term1.T <- with(subset(ps1.data.clean, tobacco.rescale.n==1), sum((tobacco.
    rescale.n*dbrwt)/propensityreduced)/sum(tobacco.rescale.n/
    propensityreduced))
173 term2.T <- with(subset(ps1.data.clean, tobacco.rescale.n==1), sum(((1-
    tobacco.rescale.n)*dbrwt)/(1-propensityreduced))/sum((1-tobacco.rescale.
    n)/(1-propensityreduced)))
174
175 weightingestimator.T <- term1.T-term2.T #This should be the average
    treatment on treated
176
177
178 # Problem 2d - Kernel Density Estimator
179 tot.propensity.nosm <- with(subset(ps1.data.clean, tobacco.rescale == 0),
    sum(propensityreduced))
180 tot.propensity.sm <- with(subset(ps1.data.clean, tobacco.rescale == 1), sum(
    propensityreduced))
181
182 kd.plot.fn <- function(h, plot.w = 5, plot.h = 3){
183 kerndensity.nosm <- with(subset(ps1.data.clean, tobacco.rescale == 0),
184     density(dbrwt, #if nobody smoked
185         kernel = "epanechnikov",
186         bw = h,
187         weights = propensityreduced/tot.propensity.
            nosm))
188 kerndensity.nosm.df <- data.frame(birth.weight = kerndensity.nosm$x, non.
    smoke = kerndensity.nosm$y)
189
190 kerndensity.sm <- with(subset(ps1.data.clean, tobacco.rescale == 1),
191     density(dbrwt, #if everybody smoked
192         kernel = "epanechnikov",
193         bw = h,
194         weights = propensityreduced/tot.propensity.sm)
    )
195 kerndensity.sm.df <- data.frame(birth.weight = kerndensity.sm$x, smoke =
    kerndensity.sm$y)
196
197 kdens <- join(kerndensity.sm.df, kerndensity.nosm.df, by="birth.weight",

```

```

198     type = "full")
199 kdens.m <- melt.data.frame(kdens, id.vars="birth.weight", measure.vars = c("
    smoke", "non.smoke"), na.rm = TRUE)
200
201 kd.plot <- ggplot(kdens.m, aes(birth.weight, value, factor(variable)))
202 kd.plot <- kd.plot +
203   geom_line(aes(color=variable)) +
204   labs(title = paste("Density of birthweights estimated using \n propensity
    score-weighted kernel regression \n Bandwidth=", as.factor(h)), x = "
    Birthweight (grams)", y = "Density") +
205   guides(color = guide_legend(title = "Treatment")) +
206   theme_bw()
207
208 kd.plot
209 kd.plot
210
211 ggsave(width = plot.w, height = plot.h, file = paste0('img/kerndensity', h, '
    .pdf'), plot = kd.plot)
212
213 }
214
215
216
217
218 ##Problem 2d - calculating kernel value by 'hand' at dbrwt = 3000 -----
219 ##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?
220
221
222 h <- 30
223
224 # added identity function to kernel (cuts off outside bw)
225 kernel.epa <- function(u){
226   if(abs(u)<1){return(0.75*(1-u*u))
227   }else{return(0)}
228 }
229
230 propensity3000.sm <- with(ps1.data.clean, mean(propensityreduced[which(dbrwt
    == 3000 & tobacco.rescale == 1)]))
231 propensity3000.nosm <- with(ps1.data.clean, mean(propensityreduced[which(
    dbrwt == 3000 & tobacco.rescale == 0)]))
232 ##for(i in 1:nrow(subset(ps1.data.clean, tobacco.rescale == 1))){
233 ##with(subset(ps1.data.clean, tobacco.rescale == 1),
234 ##  kern3000.sm.num <- kern3000.sm.num +
235 ##    kernel.epa(((propensity3000.sm-propensityreduced[i])/h)*dbrwt))
236 ## with(subset(ps1.data.clean, tobacco.rescale == 1),
237 ##    kern3000.sm.den <- kern3000.sm.den +
238 ##    kernel.epa((propensity3000.sm-propensityreduced[i])/h))
239 ## }
240 ## kern3000.sm <- kern3000.sm.num / kern3000.sm.den
241
242
243 ## kernel3000.sm <- with(subset(ps1.data.clean,tobacco.rescale == 1), data.
    frame(window = (3000 - dbrwt/h)))
244 ## kernel3000.sm$enumerator <- with(subset(ps1.data.clean, tobacco.rescale ==
    1), kernel.epa(((3000/propensity3000.sm) - (dbrwt/propensityreduced))/h)

```

```

    ))
245 ## kernel3000.sm$denominator <- with(subset(ps1.data.clean, tobacco.rescale
    == 1), kernel.epa(((3000/propensity3000.sm) - (dbrwt/propensityreduced))
    /h))
246
247 ## with(kernel3000.sm[window < 1 & window > -1], sum(numerator))/(nrow(
    kernel3000.sm[abs(window < 1)]*h)
248
249 #Problem 2e -----
250
251 for(bw in seq(10,100,10)){kd.plot.fn(bw)}
252
253 kd.plot.fn(40,7,4)
254
255
256
257 ### Problem 3
258 ## Using blocking estimator
259 # Divide smokers into ~100 equally spaced blocks
260 prop.max <- with(ps1.data.clean, max(propensityreduced))
261 prop.min <- with(ps1.data.clean, min(propensityreduced))
262 prop.binsize <- (prop.max - prop.min)/99
263
264 ps1.data.clean$blocknumber <- with(ps1.data.clean,
265     round(propensityreduced/prop.binsize,
        digits = 0) + 1)
266
267 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]))
268
269 blocktreatmenteffects$badbin <- with(blocktreatmenteffects, as.numeric(
    smokers == 0 | nonsmokers == 0))
270
271 cleaned.blocks <- subset(blocktreatmenteffects, badbin == 0)
272 cleaned.blocks$avgtreatmenteffect <- with(cleaned.blocks, smokerdbrwt -
    nonsmokerdbrwt)
273 cleaned.blocks$weight <- with(cleaned.blocks, (smokers + nonsmokers)/sum(
    smokers + nonsmokers))
274 cleaned.blocks$weightedTE <- with(cleaned.blocks, weight *
    avgtreatmenteffect)
275
276 blocksATE <- sum(cleaned.blocks$weightedTE)
277
278 ### Problem 4
279 ps1.data.clean$lowbrwt <- as.numeric(ps1.data.clean$dbrwt < 2500)
280
281 blocklowbrwt <- ddply(ps1.data.clean, .(blocknumber), summarize, smokers =
    sum(tobacco.rescale == 1), nonsmokers = sum(tobacco.rescale == 0),
    lowbrwtprob.sm = mean(lowbrwt[tobacco.rescale == 1]), lowbrwtprob.nosm =
    mean(lowbrwt[tobacco.rescale == 0]))
282
283 blocklowbrwt$badbin <- with(blocklowbrwt, as.numeric(smokers == 0 |
    nonsmokers == 0))
284
285 cleaned.blocks.lowbrwt <- subset(blocklowbrwt, badbin == 0)

```

```

286 cleaned.blocks.lowbrwt$ATE <- with(cleaned.blocks.lowbrwt, lowbrwtprob.sm -
    lowbrwtprob.nosm)
287 cleaned.blocks.lowbrwt$weight <- with(cleaned.blocks.lowbrwt, (smokers +
    nonsmokers)/sum(smokers + nonsmokers))
288 cleaned.blocks.lowbrwt$weightedTE <- with(cleaned.blocks.lowbrwt, weight *
    ATE)
289
290 blocks.lowbrwt.ATE <- sum(cleaned.blocks.lowbrwt$weightedTE)
291
292
293
294 ### Output values that need to be typed in to TeX:
295 print(paste("The estimated average treatment effect using the reweighting
    approach is", round(weightingestimator, digits=0)))
296 print(paste("ATE is", round(tobacco.effects$fit[1] - tobacco.effects$fit[2],
    digits=0), "based on regression adjustment with p-score."))
297 print(paste("The Average Treatment Effect predicted by the blocking method
    with birthweight treated as a continuous variable is", round(blocks.ATE,
    digits=0)))
298 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

```

1 # Econometrics helper functions for [R]
2 #
3 # Peter Alstone and Frank Proulx
4 # 2013
5 # version 1
6 # contact: peter.alstone AT gmail.com
7
8 # Category: Data Management -----
9
10
11 # Category: Data Analysis -----
12
13 # Function: Find adjusted R^2 for subset of data
14 # This requires a completed linear model...pull out the relevant y-values
    and residuals and feed them to function
15 # [TODO @Peter] Improve function so it can simply evaluate lm or glm object,
    add error handling, general clean up.
16 adjr2 <- function(y,resid){
17   r2 <- 1-sum(resid^2) / sum((y-mean(y))^2)
18   return(r2)
19 } #end adjr2
20
21
22 # Category: Plots and Graphics -----
23
24 ## Function for arranging ggplots. use png(); arrange(p1, p2, ncol=1); dev.
    off() to save.
25 require(grid)
26 vp.layout <- function(x, y) viewport(layout.pos.row=x, layout.pos.col=y)
27 arrange_ggplot2 <- function(..., nrow=NULL, ncol=NULL, as.table=FALSE) {

```



```

28 dots <- list(...)
29 n <- length(dots)
30 if(is.null(nrow) & is.null(ncol)) { nrow = floor(n/2) ; ncol = ceiling(n/
    nrow)}
31 if(is.null(nrow)) { nrow = ceiling(n/ncol)}
32 if(is.null(ncol)) { ncol = ceiling(n/nrow)}
33 ## NOTE see n2mfrow in grDevices for possible alternative
34 grid.newpage()
35 pushViewport(viewport(layout=grid.layout(nrow,ncol) ) )
36 ii.p <- 1
37 for(ii.row in seq(1, nrow)){
38   ii.table.row <- ii.row
39   if(as.table) {ii.table.row <- nrow - ii.table.row + 1}
40   for(ii.col in seq(1, ncol)){
41     ii.table <- ii.p
42     if(ii.p > n) break
43     print(dots[[ii.table]], vp=vp.layout(ii.table.row, ii.col))
44     ii.p <- ii.p + 1
45   }
46 }
47 }

```

../util/are213-func.R