Replication code for “Wealth, Officeholding, and Elite Demand for Slavery in Antebellum Georgia”

Jason Poulos
poulos@berkeley.edu

August 20, 2014

This file replicates the results for the analyses of the treatment effects on officeholding and support for slavery (i.e., observed test statistics, p values, and confidence intervals). First, import the data from the internet:

```
lottery <- source_DropboxData(file = "lottery.csv", key = "tdi7rb4fzbpr5l2",
    stringsAsFactors = FALSE)

## Downloading data from:  https://dl.dropboxusercontent.com/s/tdi7rb4fzbpr5l2/lottery.csv
##
## SHA-1 hash of the downloaded data file is:
## c872ee6f3fd3112afd4c2d3086d8309eb6866e2d

assembly <- source_DropboxData(file = "assembly.csv", key = "w9ozk3499dtrzla",
    stringsAsFactors = FALSE)

## Downloading data from:  https://dl.dropboxusercontent.com/s/w9ozk3499dtrzla/assembly.csv
##
## SHA-1 hash of the downloaded data file is:
## d08fcf079d9c009602aba0170f8a11a539bd8ae2
```

1 Treatment effect on officeholding

Count the number of prizes in the lottery. Create a vector representing a box of tickets equal to the number of draws. On Each ticket is a participant ID.

```
# count # of prizes
prizes <- 2 * length(lottery$draw[lottery$draw == "PP"]) + length(lottery$draw[lottery$draw ==
    "P"]) + length(lottery$draw[lottery$draw == "BP"]) + length(lottery$draw[lottery$draw ==
    "PB"])
# create a box vector
box <- c(lottery$row.no, lottery$row.no[nchar(lottery$draw) == 2])
```

Create a new data frame that excludes widows, orphans, and prior officeholders. Have a column for participant ID number, binary treatment indicator, draw result, and outcome.

```
# subset data
outcome <- subset(lottery, lottery$orphan != 1 & lottery$widow != 1 & lottery$prior.office !=
    1, select = c("row.no", "treat", "draw", "office.holder"))
```

Define a function that takes arguments for the outcome variable, treatment, draws, box, and number of prizes, to calculate average treatment effect using difference–in–means (weighted by inverse propensity score).

```
# define treatment effect function
est.ate <- function(y, treat, draw, box, prizes) {
    # calculate prob(Z=1)
    prob.treat <- ifelse(nchar(draw) == 2, 2 * (prizes/length(box)), prizes/length(box))
    # create vector of weights
    weight <- treat/prob.treat + (1 - treat)/(1 - prob.treat)
    # return ATE
    return(weighted.mean(y[treat == 1], weight[treat == 1]) - weighted.mean(y[treat ==
        0], weight[treat == 0]))
}
```

Define the permutation test. Note that permutation treatment and control group sizes will vary depending on how many participants registered for two draws win two prizes.

```
Permutation.Test <- function(box, outcome, y, prizes, L = 10000, two.sided = FALSE) {
    # specify outcome
    outcome$y <- y
    # create vector to store permutation t-stats
    new.t.stats <- rep(0, L)
    # iterate permutation test L times
    for (i in 1:L) {
        # create permutation assignment vector
        Z <- sample(c(rep(1, prizes), rep(0, length(box) - prizes)), replace = FALSE)
        # permutation assignment IDs (remove second entry)
        new.t <- sort(unique(box[Z == 1]))
        new.c <- sort(unique(box[!box %in% new.t]))
        perm.treat <- rbind(cbind(new.t, 1), cbind(new.c, 0))
        colnames(perm.treat) <- c("row.no", "perm.treat")
        # link to observed outcomes
        perm.outcome <- merge(outcome, perm.treat, by = "row.no")
        # calculate permutation test statistic
        new.t.stats[i] <- est.ate(perm.outcome$y, perm.outcome$perm.treat, perm.outcome$draw,
            box, prizes)
    }
    # calculate p value
    if (two.sided) {
        pvalue <- sum(abs(new.t.stats) >= abs(est.ate(outcome$y, outcome$treat,
            outcome$draw, box, prizes)))/L
    } else {
        pvalue <- sum(new.t.stats >= est.ate(outcome$y, outcome$treat, outcome$draw,
            box, prizes))/L
    }
    # return p-value and permutation vectors
    return(list(p = pvalue, perm.t.stats = new.t.stats))
}
```

Call `Permutation.Test`.

```
# call function
results <- Permutation.Test(box, outcome, y = outcome$office.holder, prizes)
```

Report observed t–stat, the p–value, and create histogram of test statistics from the permutation distribution.

```
# what is observed t-stat?
observed.t.stat <- est.ate(outcome$office.holder, outcome$treat, outcome$draw,
    box, prizes)
observed.t.stat

## [1] 0.01275

# report p-value
options(scipen = 9999)
results$p

## [1] 0.0003

# create histogram of permutation distribution
hist.plot <- qplot(results$perm.t.stats, geom = "histogram", xlab = expression(delta),
    ylab = "Count", binwidth = 1e-04) + labs(title = "Treatment effect on officeholding") +
    geom_vline(aes(xintercept = round(observed.t.stat, 3)), colour = "red",
        linetype = "longdash") + scale_x_continuous(breaks = c(-0.01, 0, round(observed.t.stat
    3)), labels = c("-0.01", "0", round(observed.t.stat, 3)))
```

Estimate randomization confidence intervals. Assume a constant treatment effect.

```
# define function
perm.CI <- function(alpha = 0.05, box, outcome, y, prizes, l = 1000, L = 100) {
    # define treatment and control
    treat <- outcome$treat
    # create vector to store CIs
    CI <- rep(0, l)
    for (i in 1:l) {
        # choose constant treatment effect
        delta.c <- sample(seq(-1, 1, by = 0.0000001), 1, replace = FALSE)
        # subtract from all of treated outcomes
        y.delta <- ifelse(treat == 1, y - delta.c, y)
        # run permuation test
        results <- Permutation.Test(box, outcome, y = y.delta, prizes, L, two.sided = TRUE)
        # if result not significant, delta.c is in confidence interval
        CI[i] <- ifelse(results$p > alpha, delta.c, NA)
    }
    return(range(CI, na.rm = TRUE))
}
# report(100-alpha)% CI
perm.CI(alpha = 0.05, box, outcome, y = outcome$office.holder, prizes)

## [1] 0.007288 0.019670
```

Create table showing officeholder totals by treatment group.

```
tableNominal(vars = outcome[c("office.holder")], group = outcome$treat, prec = 3,
    cumsum = FALSE, cap = "Distribution of the outcome variable, by treatment assignment, for s
    lab = "outcomes")
```

Variable	**Levels**	$\mathbf{n}_0$	$\%_0$	$\mathbf{n}_1$	$\%_1$	$\mathbf{n}_{\text{all}}$	$\%_{\text{all}}$
office.holder	0	17832	97.4	3176	96.1	21008	97.2
	1	475	2.6	129	3.9	604	2.8
	all	18307	100.0	3305	100.0	21612	100.0

Table 1: Distribution of the outcome variable, by treatment assignment, for sample excluding widows, orphans, and pre-treatment officeholders. ‘Officeholder’ indicates whether participant held office between 1806 and 1846, inclusive. *p* value is calculated using a randomization test for weighted difference of means between treatment and control groups.

2 Sensitivity analysis: above cutoff

```
# Restrict participants to median (and above) surname length
outcome2 <- subset(lottery, orphan != 1 & widow != 1 & prior.office != 1 & surname.length >=
    6, select = c("row.no", "treat", "draw", "office.holder"))
# call perm test
results2 <- Permutation.Test(box, outcome2, y = outcome2$office.holder, prizes)
results2$p

## [1] 0.033

# what is observed t-stat?
observed.t.stat2 <- est.ate(outcome2$office.holder, outcome2$treat, outcome2$draw,
    box, prizes)
observed.t.stat2
```

```
## [1] 0.007771

# create histogram of permutation distribution
hist.plot2 <- qplot(results2$perm.t.stats, geom = "histogram", xlab = expression(delta),
    ylab = "Count", binwidth = 0.0001) + labs(title = "Treatment effect on officeholding: Abov
    geom_vline(aes(xintercept = round(observed.t.stat2, 3)), colour = "red",
        linetype = "longdash") + scale_x_continuous(breaks = c(-0.01, 0, round(observed.t.stat
    3)), labels = c("-0.01", "0", round(observed.t.stat2, 3)))
# report(100-alpha)%
perm.CI(alpha = 0.05, box, outcome2, y = outcome2$office.holder, prizes)

## [1] 0.003272 0.015046
```

Create table showing officeholder totals by treatment group.

```
tableNominal(vars = outcome2[c("office.holder")], group = outcome2$treat, prec = 3,
    cumsum = FALSE, cap = "Distribution of the outcome variable, by treatment assignment, for
    lab = "outcomes-subset-surname")
```

Variable	Levels	n_0	$\%_0$	n_1	$\%_1$	n_{all}	$\%_{all}$
office.holder	0	11923	97.3	2167	96.5	14090	97.2
	1	334	2.7	79	3.5	413	2.9
	all	12257	100.0	2246	100.0	14503	100.0

Table 2: Distribution of the outcome variable, by treatment assignment, for sample of participants having a surname length of 6 characters or greater. See notes for Table 1.

3 Sensitivity analysis: treatment–on–the–treated (TOT)

```
# subset data
outcome3 <- subset(lottery, lottery$orphan != 1 & lottery$widow != 1 & lottery$prior.office !=
    1, select = c("row.no", "treat", "draw", "office.holder", "RGB", "county",
    "prize", "prize2"))
# create dummy for treatment received
outcome3$tot <- outcome3$treat
outcome3$tot[outcome3$RGB == 1] <- 0
# create dummy for 2 draws
outcome3$twodraw <- ifelse(nchar(outcome3$draw) == 2, 1, 0)
# call IV
results3 <- ivreg(office.holder ~ tot | treat, data = outcome3)
results3.2 <- ivreg(office.holder ~ twodraw + tot | twodraw + treat, data = outcome3)
results3.3 <- ivreg(office.holder ~ twodraw + county + tot | twodraw + county +
    treat, data = outcome3)
```

```r
# get confidence interval
confint(results3.3, "tot", level = 0.95)

##         2.5 %  97.5 %
## tot 0.007901 0.02127

# report treatment effect
results3.3$coefficients["tot"]

##     tot
## 0.01459
```

Create table showing officeholder totals by treatment group and compliance status.

```r
# partition groups for table (0 = control; 1 = complier; 2= noncomplier)
outcome3$partition <- outcome3$treat
outcome3$partition[outcome3$RGB == 1] <- 2
# create table
tableNominal(vars = outcome3[c("office.holder")], group = c(outcome3$partition),
    prec = 3, cumsum = FALSE, cap = "Distribution of the outcome variable, by treatment assign
    lab = "outcomes-subset-compliance")
```

Variable	**Levels**	$\mathbf{n_0}$	$\%_0$	$\mathbf{n_1}$	$\%_1$	$\mathbf{n_2}$	$\%_2$	$\mathbf{n_{all}}$	$\%_{all}$
office.holder	0	17832	97.4	2933	96.0	243	96.8	21008	97.2
	1	475	2.6	121	4.0	8	3.2	604	2.8
	all	18307	100.0	3054	100.0	251	100.0	21612	100.0

Table 3: Distribution of the outcome variable, by treatment assignment and compliance status, for analysis of treatment–on–the–treated (TOT). See notes for Table.

4 Sensitivity analysis: three treatments

Create dummies for actually receiving Ba, Wa, and Wi land.

```r
# use same lottery as noncomplier analysis
outcome4 <- outcome3
# create treatment dummies
outcome4$Ba <- ifelse(substr(outcome4$prize, 1, 2) == "Ba" | substr(outcome4$prize2,
    1, 2) == "Ba", 1, 0)
outcome4$Ba[is.na(outcome4$Ba)] <- 0
outcome4$Wa <- ifelse(substr(outcome4$prize, 1, 2) == "Wa" | substr(outcome4$prize2,
    1, 2) == "Wa", 1, 0)
outcome4$Wa[is.na(outcome4$Wa)] <- 0
outcome4$Wi <- ifelse(substr(outcome4$prize, 1, 2) == "Wi" | substr(outcome4$prize2,
    1, 2) == "Wi" | substr(outcome4$prize, 1, 2) == "W", 1, 0)
```

```
outcome4$Wi[is.na(outcome4$Wi)] <- 0
# create TOT dummies
outcome4$Ba.tot <- outcome4$Ba
outcome4$Ba.tot[outcome4$RGB == 1] <- 0
outcome4$Wa.tot <- outcome4$Wa
outcome4$Wa.tot[outcome4$RGB == 1] <- 0
outcome4$Wi.tot <- outcome4$Wi
outcome4$Wi.tot[outcome4$RGB == 1] <- 0
```

Repeat IV analysis using the full specification.

```
# call IV
results4.Ba.3 <- ivreg(office.holder ~ twodraw + county + Ba.tot | twodraw +
    county + Ba, data = outcome4)
results4.Wa.3 <- ivreg(office.holder ~ twodraw + county + Wa.tot | twodraw +
    county + Wa, data = outcome4)
results4.Wi.3 <- ivreg(office.holder ~ twodraw + county + Wi.tot | twodraw +
    county + Wi, data = outcome4)
# get confidence interval
confint(results4.Ba.3, "Ba.tot", level = 0.95)

##           2.5 %  97.5 %
## Ba.tot 0.007408 0.02553

confint(results4.Wa.3, "Wa.tot", level = 0.95)

##           2.5 %  97.5 %
## Wa.tot 0.004291 0.03677

confint(results4.Wi.3, "Wi.tot", level = 0.95)

##            2.5 %  97.5 %
## Wi.tot -0.003508 0.01572

# report treatment effects
results4.Ba.3$coefficients["Ba.tot"]

##  Ba.tot
## 0.01647

results4.Wa.3$coefficients["Wa.tot"]

##  Wa.tot
## 0.02053

results4.Wi.3$coefficients["Wi.tot"]

##   Wi.tot
## 0.006108
```

Create table showing treatment assignment and compliance status.

```
# partition groups for table (0 = control; 1 = complier; 2= noncomplier)
outcome4$partition <- outcome4$treat
outcome4$partition[outcome4$RGB == 1] <- 2
# create table
tableNominal(vars = outcome4[c("Ba", "Wa", "Wi")], group = c(outcome4$partition),
    prec = 3, cumsum = FALSE, cap = "Distribution of treatment assignment, by compliance statu
    lab = "three-counties")
```

Variable	Levels	n_0	$\%_0$	n_1	$\%_1$	n_2	$\%_2$	n_{all}	$\%_{all}$
Ba	0	18275	99.8	1690	55.3	226	90.0	20191	93.4
	1	32	0.2	1364	44.7	25	10.0	1421	6.6
	all	18307	100.0	3054	100.0	251	100.0	21612	100.0
Wa	0	18278	99.8	2523	82.6	52	20.7	20853	96.5
	1	29	0.2	531	17.4	199	79.3	759	3.5
	all	18307	100.0	3054	100.0	251	100.0	21612	100.0
Wi	0	18280	99.8	1833	60.0	213	84.9	20326	94.0
	1	27	0.1	1221	40.0	38	15.1	1286	6.0
	all	18307	100.0	3054	100.0	251	100.0	21612	100.0

Table 4: Distribution of treatment assignment, by compliance status.

5 Estimation of treatment effect on support for slavery

Calculate observed test statistic, randomization *p* value, and confidence interval.

```
# call perm test
results.assembly <- Permutation.Test(box, assembly, y = assembly$vote.index,
    prizes)
results.assembly$p

## [1] 0.2209

# what is observed t-stat?
observed.t.assembly <- est.ate(assembly$vote.index, assembly$treat, assembly$draw,
    box, prizes)
observed.t.assembly

## [1] 0.06943

# create histogram of permutation distribution
hist.plot4 <- qplot(results.assembly$perm.t.stats, geom = "histogram", xlab = expression(delta
    ylab = "Count", binwidth = 0.001) + labs(title = "Treatment effect on demand for slavery")
    geom_vline(aes(xintercept = round(observed.t.assembly, 3)), colour = "red",
        linetype = "longdash") + scale_x_continuous(breaks = c(-0.2, 0, 0.2,
    round(observed.t.assembly, 3)), labels = c("-0.2", "0", "0.2", round(observed.t.assembly,
    3)))
# report(100-alpha)% CI
perm.CI(alpha = 0.05, box, assembly, y = assembly$vote.index, prizes)

## [1] -0.1148  0.2672
```

Create table showing outcomes by treatment group.

```
# custom stats
my.stats <- list("n", "min", "mean", "max", "s")
tableContinuous(vars = assembly[("vote.index")], group = assembly$treat, prec = 3,
    cumsum = FALSE, stats = my.stats, cap = "Distribution of the outcome variable, by treatmen
    lab = "outcomes-assembly")
```

Variable	Levels	n	Min	$\bar{x}$	Max	s
vote.index	0	138	0	0.661	1	0.422
	1	36	0	0.721	1	0.375
	all	174	0	0.674	1	0.412

Table 5: Distribution of the outcome variable, by treatment assignment, for lottery participants who held the office in the Georgia General Assembly, and who voted on at least one of the thirteen bills used to form the outcome variable. 'Support for slavery' is the mean of votes in favor of slavery for roll calls in which the member is present. p values are calculated using a randomization test for weighted difference of means between treatment and control groups.

5.1 Sensitivity analysis: three treatments

```
# use same lottery as previous analysis
assembly2 <- assembly
# create dummy for treatment received
assembly2$tot <- assembly2$treat
assembly2$tot[assembly2$RGB == 1] <- 0
# create dummy for 2 draws
assembly2$twodraw <- ifelse(nchar(assembly2$draw) == 2, 1, 0)
# create treatment dummies
assembly2$Ba <- ifelse(substr(assembly2$prize, 1, 2) == "Ba", 1, 0)
assembly2$Ba[is.na(assembly2$Ba)] <- 0
assembly2$Wa <- ifelse(substr(assembly2$prize, 1, 2) == "Wa", 1, 0)
assembly2$Wa[is.na(assembly2$Wa)] <- 0
assembly2$Wi <- ifelse(substr(assembly2$prize, 1, 2) == "Wi", 1, 0)
```

```
assembly2$Wi[is.na(assembly2$Wi)] <- 0
# create TOT dummies
assembly2$Ba.tot <- assembly2$Ba
assembly2$Ba.tot[assembly2$RGB == 1] <- 0
assembly2$Wa.tot <- assembly2$Wa
assembly2$Wa.tot[assembly2$RGB == 1] <- 0
assembly2$Wi.tot <- assembly2$Wi
assembly2$Wi.tot[assembly2$RGB == 1] <- 0
```

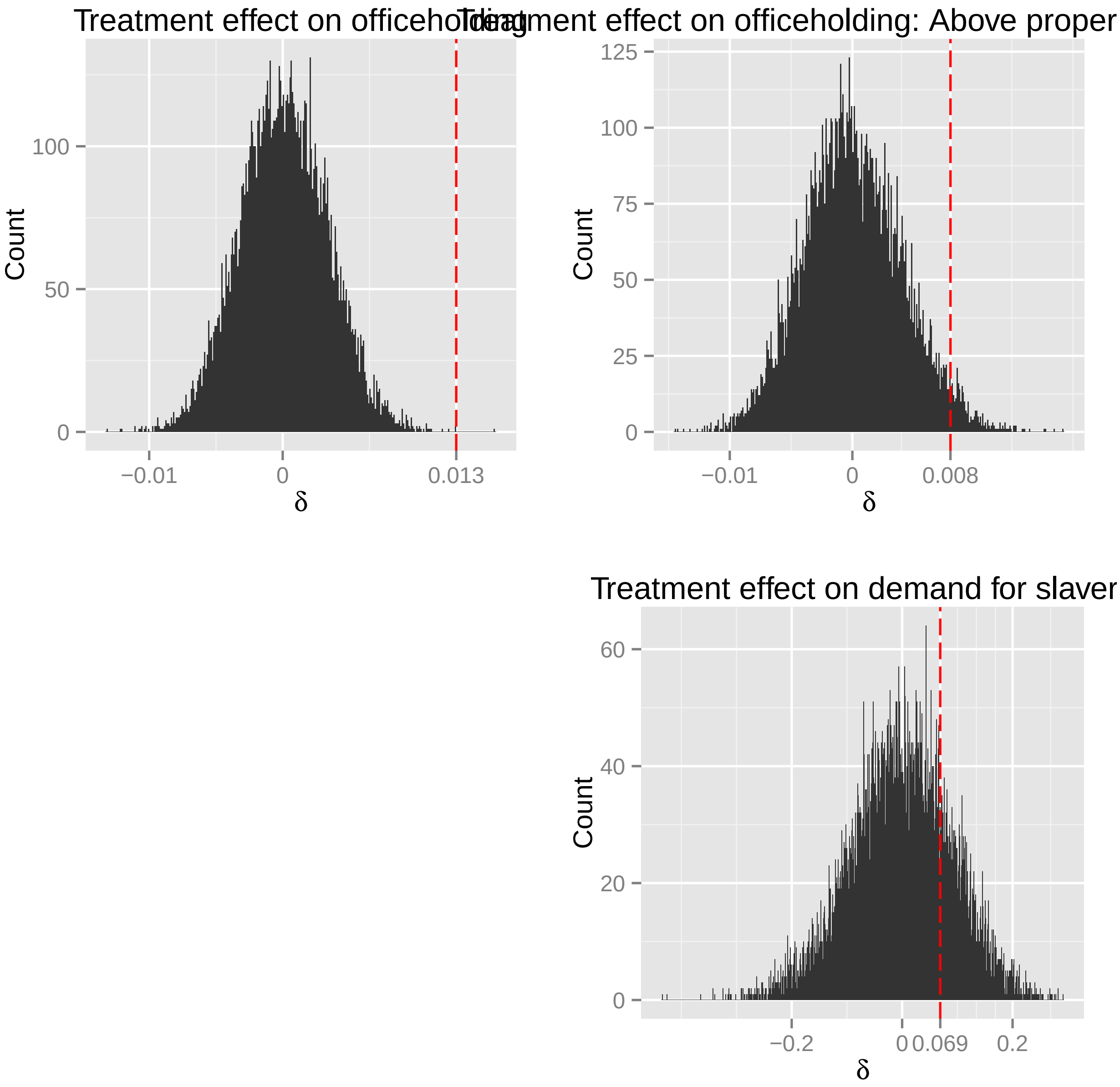
Repeat IV analysis.

```
# call IV
resuts.assembly.Ba.3 <- ivreg(vote.index ~ twodraw + county + Ba.tot | twodraw +
    county + Ba, data = assembly2)
resuts.assembly.Wa.3 <- ivreg(vote.index ~ twodraw + county + Wa.tot | twodraw +
    county + Wa, data = assembly2)
resuts.assembly.Wi.3 <- ivreg(vote.index ~ twodraw + county + Wi.tot | twodraw +
    county + Wi, data = assembly2)
# get confidence interval
confint(resuts.assembly.Ba.3, "Ba.tot", level = 0.95)

##           2.5 % 97.5 %
## Ba.tot -0.06622 0.4206

confint(resuts.assembly.Wa.3, "Wa.tot", level = 0.95)

##          2.5 % 97.5 %
## Wa.tot -0.3821 0.3792

confint(resuts.assembly.Wi.3, "Wi.tot", level = 0.95)

##          2.5 % 97.5 %
## Wi.tot -0.2615 0.2853

# report treatment effects
resuts.assembly.Ba.3$coefficients["Ba.tot"]

## Ba.tot
## 0.1772

resuts.assembly.Wa.3$coefficients["Wa.tot"]

##    Wa.tot
## -0.001449

resuts.assembly.Wi.3$coefficients["Wi.tot"]

##  Wi.tot
## 0.01194
```

6 Permutation plots

Combine permutation plots.

```r
vplayout <- function(x, y) viewport(layout.pos.row = x, layout.pos.col = y)
grid.newpage()
pushViewport(viewport(layout = grid.layout(2, 2)))
print(hist.plot, vp = vplayout(1, 1))

## Warning:  position_stack requires constant width:  output may be incorrect

print(hist.plot2, vp = vplayout(1, 2))

## Warning:  position_stack requires constant width:  output may be incorrect

print(hist.plot4, vp = vplayout(2, 2))

## Warning:  position_stack requires constant width:  output may be incorrect
```


Treatment effect on officeholding
Count
100
50
0
−0.01
0
0.013
δ
Treatment effect on officeholding: Above proper
Count
125
100
75
50
25
0
−0.01
0
0.008
δ
Treatment effect on demand for slaver
Count
60
40
20
0
−0.2
0
0.069
0.2
δ