Quant II

Lab 13: Multiple endpoints, missing data, power analysis

Giacomo Lemoli

May 4, 2023

Today's plan

- Multiple outcomes

Today's plan

- Multiple outcomes
- Missing data

Today's plan

- Multiple outcomes
- Missing data
- Power analysis

Review of concepts

- Testing multiple hypotheses increases the probability of type I (false Positives) or type II error (false Negatives)

Review of concepts

- Testing multiple hypotheses increases the probability of type I (false Positives) or type II error (false Negatives)
- We want to test multiple hypotheses: different behavioral outcomes, votes in multiple elections etc

Review of concepts

- Testing multiple hypotheses increases the probability of type I (false Positives) or type II error (false Negatives)
- We want to test multiple hypotheses: different behavioral outcomes, votes in multiple elections etc
- Popular strategies:

Review of concepts

- Testing multiple hypotheses increases the probability of type I (false Positives) or type II error (false Negatives)
- We want to test multiple hypotheses: different behavioral outcomes, votes in multiple elections etc
- Popular strategies:
  - Summarize different measures in a single one

Review of concepts

- Testing multiple hypotheses increases the probability of type I (false Positives) or type II error (false Negatives)
- We want to test multiple hypotheses: different behavioral outcomes, votes in multiple elections etc
- Popular strategies:
  - Summarize different measures in a single one
  - Adjust p-values for multiple comparisons

Principal Component Analysis

- Reduce dimensionality of the outcomes by extracting shared variation along different dimensions

Principal Component Analysis

- Reduce dimensionality of the outcomes by extracting shared variation along different dimensions
- Returns estimates of "principal components" that are orthogonal to each other

Principal Component Analysis

- Reduce dimensionality of the outcomes by extracting shared variation along different dimensions
- Returns estimates of "principal components" that are orthogonal to each other
- We can use the principal components scores (usually the first one) as a single outcome that "summarizes" the shared variation

Principal Component Analysis

- Reduce dimensionality of the outcomes by extracting shared variation along different dimensions
- Returns estimates of "principal components" that are orthogonal to each other
- We can use the principal components scores (usually the first one) as a single outcome that "summarizes" the shared variation
- Stata: `pca`. R: `prcomp`, `princomp`

Example

The effect of military repression on support for democracy

AJPS AMERICAN JOURNAL of POLITICAL SCIENCE

The Geography of Repression and Opposition to Autocracy

Maria Angélica Bautista University of Chicago
Felipe González Pontificia Universidad Católica de Chile
Luis R. Martínez University of Chicago
Pablo Muñoz FGV EPGE Brazilian School of Economics and Finance
Mounu Prem Universidad del Rosario

Abstract: *State repression is a prominent feature of nondemocracies, but its effectiveness in quieting dissent and fostering regime survival remains unclear. We exploit the location of military bases before the coup that brought Augusto Pinochet to power in Chile in 1973, which is uncorrelated to precoup electoral outcomes, and show that counties near these bases experienced more killings and forced disappearances at the hands of the government during the dictatorship. Our main result is that residents of counties close to military bases both registered to vote and voted "No" to Pinochet's continuation in power at higher rates in the crucial 1988 plebiscite that bolstered the democratic transition. Potential mechanisms include informational frictions on the intensity of repression in counties far from bases and shifts in preferences caused by increased proximity to the events. Election outcomes after democratization show no lasting change in political preferences.*

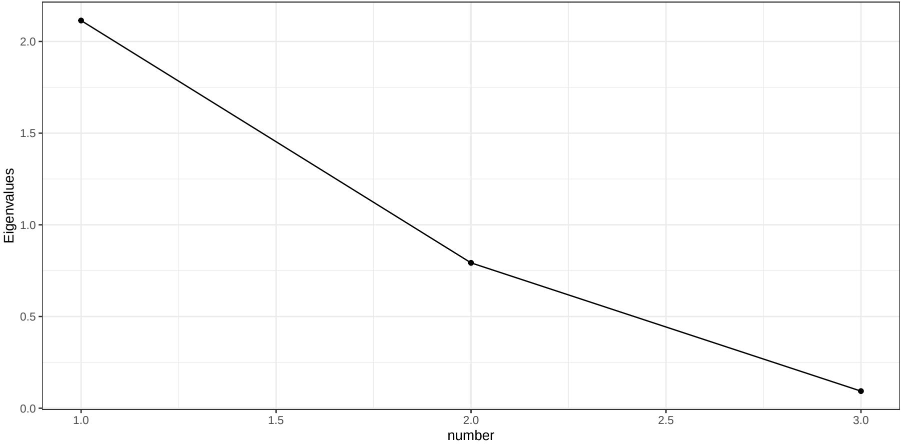
Principal Component Analysis

```r
vars <- c("Share_reg70_w2", "VoteShareNo", "VoteShareNo_pop70")
pca <- prcomp(d[,vars], center=T, scale=T)
pca
```

```
## Standard deviations (1, .., p=3):
## [1] 1.4540286 0.8902977 0.3052389
##
## Rotation (n x k) = (3 x 3):
##                          PC1         PC2        PC3
## Share_reg70_w2    -0.5861655 -0.55108749  0.5938961
## VoteShareNo       -0.4585680  0.82998321  0.3175582
## VoteShareNo_pop70 -0.6679261 -0.08620006 -0.7392187
```

Screeplot

```
data.frame(number=c(1:3), Eigenvalues = (pca$sdev)^2) %>%
  ggplot(aes(x=number, y=Eigenvalues)) + geom_point() + geom_line() + theme_bw()
```



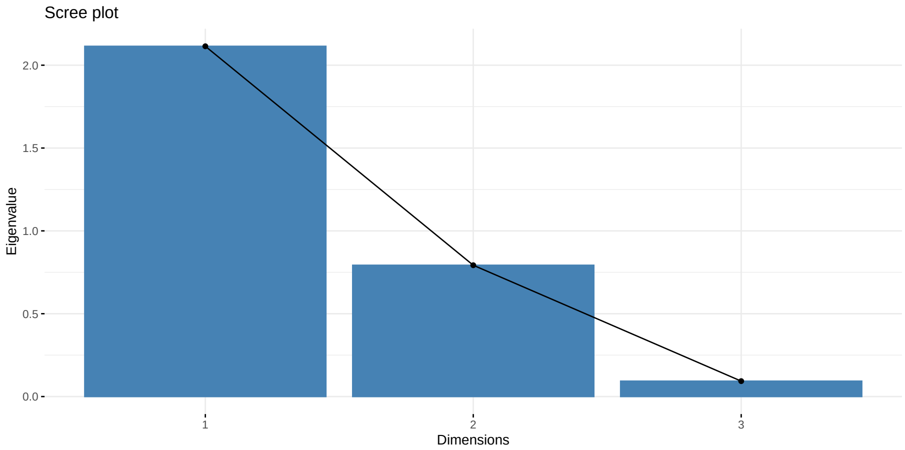
Principal Component Analysis

- Can also pass the PCA output through the package `factoextra`

Principal Component Analysis

- Can also pass the PCA output through the package `factoextra`

```
library(factoextra)
fviz_eig(pca, choice = "eigenvalue")
```



Principal Component Analysis

- First factor can be used as outcome in the regression

Principal Component Analysis

- First factor can be used as outcome in the regression
- Factors are estimates. For correct inference, incorporate their uncertainty in inference on the effect

Principal Component Analysis

- First factor can be used as outcome in the regression
- Factors are estimates. For correct inference, incorporate their uncertainty in inference on the effect
- Can bootstrap (sample code in this week's folder)

Inverse Covariance Weighting

- Summarizes outcome variables along just one dimension

Inverse Covariance Weighting

- Summarizes outcome variables along just one dimension
- Weighted by inverse covariance in order to downplay common information across variables

Inverse Covariance Weighting

- Summarizes outcome variables along just one dimension
- Weighted by inverse covariance in order to downplay common information across variables
- Code by Cyrus: Stata and R code

p-value adjustment

- Most standard p-value adjustment methods implemented in R by `p.adjust`

```
# Generate p-values from different tests
set.seed(1)
p <- sort(runif(10, 0.03, 0.07))
d <- data.frame(p = p,
                bonferroni = p.adjust(p, method = "bonferroni"))
d
```

```
##             p bonferroni
## 1  0.03247145  0.3247145
## 2  0.03806728  0.3806728
## 3  0.04062035  0.4062035
## 4  0.04488496  0.4488496
## 5  0.05291413  0.5291413
## 6  0.05516456  0.5516456
## 7  0.05643191  0.5643191
## 8  0.06593559  0.6593559
## 9  0.06632831  0.6632831
## 10 0.06778701  0.6778701
```

Implementing p-value adjustments

- Summary of Stata packages for adjustment options by David McKenzie

Implementing p-value adjustments

- Summary of Stata packages for adjustment options by David McKenzie
- Sample code from McKenzie in this week's folder

Missing data

- Random missingness

Missing data

- Random missingness
- Non-parametric methods: bounds

Missing data

- Random missingness
- Non-parametric methods: bounds
- Parametric methods: imputation

Example

The effects of censorship in authoritarian regimes

AJPS AMERICAN JOURNAL of POLITICAL SCIENCE

Sometimes Less Is More: Censorship, News Falsification, and Disapproval in 1989 East Germany

Christian Gläßel University of Mannheim
Katrin Paula University of Mannheim

Abstract: *Does more media censorship imply more regime stability? We argue that censorship may cause mass disapproval for censoring regimes. In particular, we expect that censorship backfires when citizens can falsify media content through alternative sources of information. We empirically test our theoretical argument in an autocratic regime—the German Democratic Republic (GDR). Results demonstrate how exposed state censorship on the country's emigration crisis fueled outrage in the weeks before the 1989 revolution. Combining original weekly approval surveys on GDR state television and daily content data of West German news programs with a quasi-experimental research design, we show that recipients disapproved of censorship if they were able to detect misinformation through conflicting reports on Western television. Our findings have important implications for the study of censoring systems in contemporary autocracies, external democracy promotion, and campaigns aimed at undermining trust in traditional journalism.*

Baseline results (full data)

```
library(haven); library(estimatr); library(modelsummary); library(tidyverse); library(ggpubr)
d <- read_dta("Censorship.dta")

# Reproduce col 1 of table 1
mod <- lm_robust(ak_rating ~ censorship + liberalization, clusters = hh, se_type = "stata",
                 data = d)
modelsummary(mod, coef_omit = "Int", output = "markdown")
```

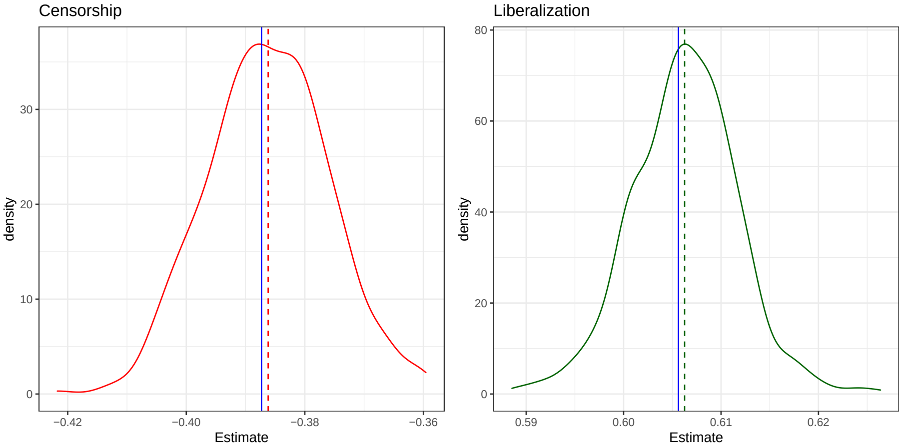
	Model 1
censorship	-0.387
	(0.036)
liberalization	0.606
	(0.021)
Num.Obs.	17551
R2	0.235
R2 Adj.	0.235
se_type	stata

Missing Completely at Random

```
set.seed(123)

nboot <- 500

cens <- lib <- rep(NA,nboot)

# 20% of outcome values are missing in the data
for (i in 1:nboot){
  d <- d %>% mutate(newout = ifelse(runif(nrow(d),0,1)<0.2, NA, ak_rating))
  fit <- update(mod, newout ~ .)
  cens[i] <- coef(fit)["censorship"]
  lib[i] <- coef(fit)["liberalization"]
}

plot1 <- ggplot(as.data.frame(cens)) + geom_density(aes(x=cens), col="red") +
  labs(x="Estimate", title = "Censorship") +
  geom_vline(xintercept = mean(cens), col="red", linetype = "dashed") +
  geom_vline(xintercept = coef(mod)["censorship"], col="blue") + theme_bw()

plot2 <- ggplot(as.data.frame(lib)) + geom_density(aes(x=lib), col="dark green") +
  labs(x="Estimate", title = "Liberalization") +
  geom_vline(xintercept = mean(lib), col="dark green", linetype = "dashed") +
  geom_vline(xintercept = coef(mod)["liberalization"], col="blue") + theme_bw()
```

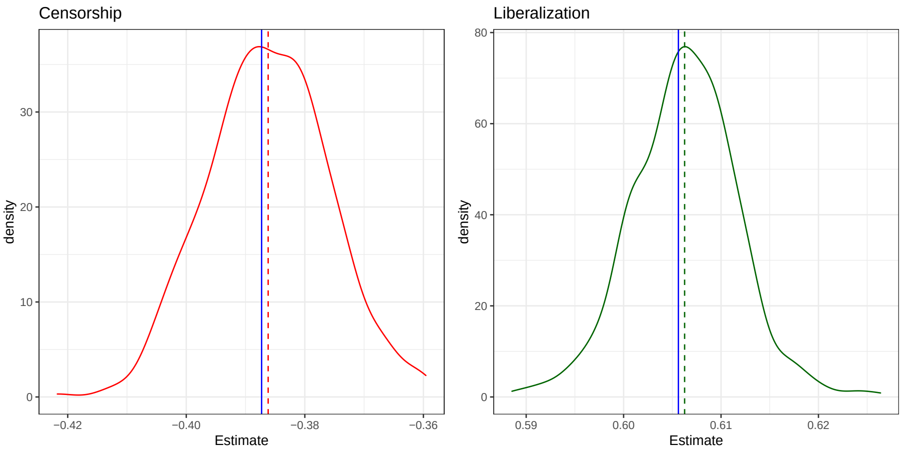
Missing Completely at Random

```
ggarrange(plot1, plot2)
```



Missing Completely at Random

```
ggarrange(plot1, plot2)
```

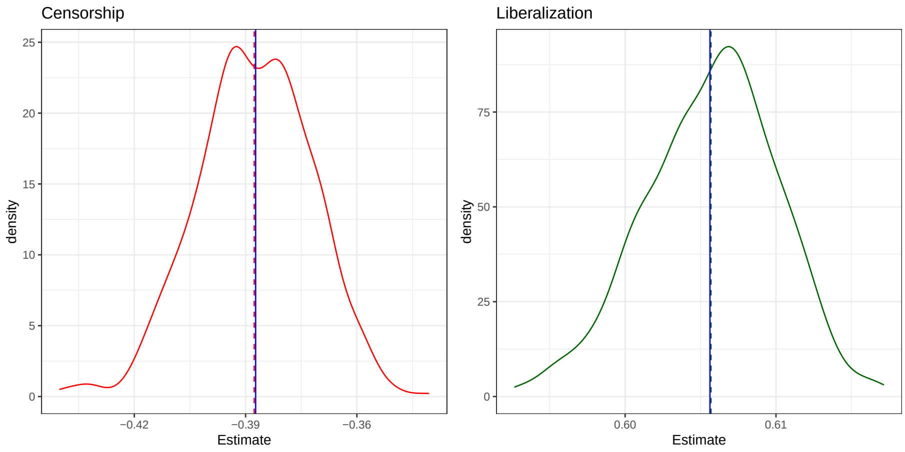


Missing at Random

```r
# Suppose during censorship people express negative attitudes by non-responding
set.seed(123)

nboot <- 500

cens <- lib <- rep(NA,nboot)

for (i in 1:nboot){
  d <- d %>% mutate(newout = ifelse(rnorm(nrow(d), mean = d$censorship)>1.15,
                                    NA, ak_rating))
  fit <- update(mod, newout ~ .)
  cens[i] <- coef(fit)["censorship"]
  lib[i] <- coef(fit)["liberalization"]
}

plot1 <- ggplot(as.data.frame(cens)) + geom_density(aes(x=cens), col="red") +
  labs(x="Estimate", title = "Censorship") +
  geom_vline(xintercept = mean(cens), col="red", linetype = "dashed") +
  geom_vline(xintercept = coef(mod)["censorship"], col="blue") + theme_bw()

plot2 <- ggplot(as.data.frame(lib)) + geom_density(aes(x=lib), col="dark green") +
  labs(x="Estimate", title = "Liberalization") +
  geom_vline(xintercept = mean(lib), col="dark green", linetype = "dashed") +
  geom_vline(xintercept = coef(mod)["liberalization"], col="blue") + theme_bw()
```

Missing at Random

```
ggarrange(plot1, plot2)
```



Point identification with missingness

- MCAR: in expectation the estimates are unbiased

Point identification with missingness

- MCAR: in expectation the estimates are unbiased
- MAR: can use covariates that determine missingness to impute the missing values of the outcome

Imputation with regression

```r
# Predicted values from the regression
d <- d %>% mutate(predicted = predict(fit, newdata = d))

# Replace missing outcomes with imputed values
d <- d %>% mutate(impy = ifelse(is.na(newout), predicted, newout))
```

In Stata you can use the post-estimation command `predict`

```stata
help predict
reg newout censorship liberalization, cluster(hh)
predict predicted, xb
replace newout = predicted if missing(newout) & !missing(predicted)
```

Multiple imputation with MICE

```
library(mice)
m1 <- mice(data = zap_labels(d[,c("newout", "censorship", "liberalization",
                                  "sed", "education")]),
           maxit = 5, printFlag = FALSE)
d_imp <- complete(m1)

length(d$newout[is.na(d$newout)])
```

```
## [1] 2992
```

```
length(d_imp$newout[is.na(d_imp$newout)])
```

```
## [1] 0
```

Bounds

- If MAR assumption seems unjustified, we can estimate bounds on causal effects

Bounds

- If MAR assumption seems unjustified, we can estimate bounds on causal effects
- Manski bounds: also called “worst case” bounds

Bounds

- If MAR assumption seems unjustified, we can estimate bounds on causal effects
- Manski bounds: also called "worst case" bounds
- Fill missing outcomes using the bounds of potential outcomes

Manski bounds

Formally:

$$\beta^L \leq ATE \leq \beta^H$$

$$\beta^L = \{\mu_{1,obs} Pr[R_{1i} = 1] + y_1^L Pr[R_{1i=0}]\} -$$

$$\{\mu_{0,obs} Pr[R_{0i} = 1] + y_0^H Pr[R_{0i} = 0]\}$$

$$\beta^H = \{\mu_{1,obs} Pr[R_{1i} = 1] + y_1^H Pr[R_{1i=0}]\} -$$

$$\{\mu_{0,obs} Pr[R_{0i} = 1] + y_0^L Pr[R_{0i} = 0]\}$$

Manski bounds

Formally:

$$\beta^L \leq ATE \leq \beta^H$$

$$\beta^L = \{\mu_{1,obs} Pr[R_{1i} = 1] + y_1^L Pr[R_{1i=0}]\} - \{\mu_{0,obs} Pr[R_{0i} = 1] + y_0^H Pr[R_{0i} = 0]\}$$

$$\beta^H = \{\mu_{1,obs} Pr[R_{1i} = 1] + y_1^H Pr[R_{1i=0}]\} - \{\mu_{0,obs} Pr[R_{0i} = 1] + y_0^L Pr[R_{0i} = 0]\}$$

- $\{y_t^L, y_t^H\}$ straightforward when the outcome is bounded, e.g. binary (just replace every missing y with 0 or 1)
- Otherwise, they are not
- If the bounds are very large, the set of possible effects can be too large

Manski bounds

Implementation:

- Manually
- R: `ATbounds` (vignette), `attrition` by Alex Coppock (download from GitHub)

```
#install.packages("ATbounds")
#library(ATbounds)

#install.packages(devtools)
#library(devtools)
#install_github("acoppock/attrition")
```

Lee bounds

- Motivation: provide bounds on the causal effect even with unbounded outcome

Lee bounds

- Motivation: provide bounds on the causal effect even with unbounded outcome
- Trade-off: assume more structure on the data, but weaker than "exclusion restrictions" necessary for other strategies

Lee bounds

- Motivation: provide bounds on the causal effect even with unbounded outcome
- Trade-off: assume more structure on the data, but weaker than "exclusion restrictions" necessary for other strategies
- Relies on monotonicity assumption similar to that used for LATE identification, but in this case it serves to solve sample selection

Lee bounds

Assumptions:

$$(Y_{1i}^*, Y_{0i}^*, R_{1i}, R_{0i}) \perp D$$
$$R_{1i} \geq R_{0i}$$

Under these assumptions, Lee (2009) proves that we can identify bounds for the ATE *for the sub-population* $\{R_{0i} = 1, R_{1i} = 1\}$.

$$\Delta_0^{LB} = E[Y|D = 1, R = 1, Y \leq y_{1-p0}] - E[Y|D = 0, R = 1]$$
$$\Delta_0^{UB} = E[Y|D = 1, R = 1, Y \geq y_{p0}] - E[Y|D = 0, R = 1]$$
$$p_0 = \frac{Pr[R = 1|D = 1] - Pr[R = 1|D = 0]}{Pr[R = 1|D = 1]}$$

Lee bounds

Implementation:

- R: `leebounds` (install_github("vsemenova/leebounds"))
- Stata: `leebounds`

Lee bounds

```r
library(leebounds)

input <- d %>% mutate(sel = ifelse(is.na(newout),0,1)) %>%
  select(c(censorship, newout, sel)) %>%
  rename(treat = censorship, outcome = newout, selection = sel)

do.call("rbind", leebounds(input))
```

```
##                     62.58541%
## lower_bound        -1.1071905
## upper_bound        -0.5783773
## p0                  1.5978164
## trimmed_mean_upper  4.2069751
## trimmed_mean_lower  3.6781620
## mean_no_trim        3.0997846
## odds                0.1690535
## yp0                 4.0000000
## y1p0                4.0000000
## s0                  0.8769733
## s1                  0.5488574
## prop0               0.8553929
## prop1               0.1446071
```