GV330: Summative Assignment

Replication of Metrics Management and Bureaucratic Accountability

48626

Thu/15/May

1 Part 1: Introduction

Eckhouse (2021) looks at: How does metrics management, specifically CompStat, affect the behaviour of bureaucrats such as police officers? It matters because metrics management is increasingly used across government agencies to improve performance and accountability, but it may unintentionally incentivise behaviors like data manipulation and an emphasis on easy, short-term tasks to statpad arrests over meaningful, long-term work that better serves the public.

The article makes causal claims, arguing that adopting metrics management causes: increases in minor arrests, data manipulation (measured through rape reports being labeled "unfounded" more) and no decrease in serious crime, despite the focus on accountability

Eckhouse (2021) compiles a panel dataset of 47 major U.S. cities from 1990 to 2013, tracking the adoption of CompStat and combining it with annual police data on arrests, reported crimes, and clearance rates. The dataset includes 1,081 city-year observations and distinguishes between serious (Part 1) and minor (Part 2) crimes.

Eckhouse uses the binary implementation of metrics management, whether a city has adopted CompStat in a given year (0/1). This is the independent variable.

There are 3 outcome variables used:

1. Minor (Part 2) arrests: to capture shifts towards statpadding behaviours; whether police officers prioritise less serious crimes to increase their arrests stats.
2. Data manipulation: to capture efforts to falsely improve reported performance. This is operationalised by measuring the percentages of raped labelled "unfounded".
3. Serious crime (Part 1) incidence: capture the success or failure of serious crime reduction. Operationalised as reported part 1 crimes (e.g. murder, rape, robbery) and part 1 arrests.

Primary data sources:

1. FBI Uniform Crime Reporting (UCR) Program – for annual data on arrests, reported offenses (Part 1 and Part 2), and clearance rates.
2. Original hand-coded dataset – built from newspaper articles, police department websites, and public records to document the timing of CompStat adoption across cities.

For statistical methodology, the author uses fixed-effects regression models. This approach compares each city to itself over time, before and after adopting CompStat. It includes agency fixed effects, which control for unchanging characteristics specific to each city or police department, such as geographic location or long-standing institutional practices, and year fixed effects to account for nationwide trends over time. This helps isolate whether changes in policing outcomes are likely due to CompStat, rather than differences between cities or broader societal shifts.

Year was included as a factor variable (i.e., treated as a set of year-specific fixed effects) rather than as a continuous variable. While this effectively controls for national trends across years, it assumes that the effect of time is purely categorical and ignores potential temporal dynamics.

The study finds that adopting CompStat significantly increased the number of minor (Part 2) arrests and the share of rape cases classified as "unfounded", suggesting shifts toward stat-padding behaviours and potential data manipulation. However, there was no evidence that CompStat reduced serious crime (Part 1), indicating that the policy unintentionally changed police behaviour in ways that didn't improve public safety outcomes.

For robustness checks, the authors:

- **Conducted placebo tests**: The study examines whether "unfounded" rates for auto theft and murder changed after CompStat adoption. They did not, which supports the interpretation that increases in rape unfounding reflect data manipulation rather than improved record-keeping.
- **Conducted sensitivity analysis excluding New York City**: To ensure that results are not driven by a single influential case, the analysis is repeated without New York City. The findings remain consistent, reinforcing the validity of the results.
- **Used demographic controls**: The models include controls for changes in racial composition, such as the percentage of Black and White residents, to account for demographic shifts that might influence policing outcomes.
- **Did pre-trend checks**: The author tests for pre-existing trends in crime rates before CompStat adoption. This helps validate the parallel trends assumption, a key requirement for causal inference in panel data analysis.
- **Used wild bootstrapping of standard errors**: A resampling technique used to calculate more robust standard errors, especially in the presence of heteroskedasticity or clustering. This ensures that statistical significance is not overstated due to unreliable error estimates, so the conclusions on the effects of variables, such as CompStat, remain robust.
- **Jackknife tests**: Linked to sensitivity analysis, the study systematically removes one city at a time and re-runs the analysis to assess how much each city influences the overall results. This confirms that no single city is driving the observed effects. This effect is most prominent when NYC is removed.

2 Part 2: Computational reproducibility

Tables 3, 5 and 6 were selected to be replicated:

- Table 3 examines the effect of CompStat on the number of Part 2 (non-serious) arrests. This is central to the paper's core argument - that metrics-based management systems like CompStat lead to increased policing activity focused on less serious offenses.
- Table 5 evaluates whether CompStat adoption affects the share of rape cases classified as "unfounded." It also includes placebo tests using auto theft and murder as robustness checks. This table is critical in assessing a key unintended consequence of metrics-driven policing: the potential manipulation of crime data to improve performance metrics.
- Table 6 investigates the relationship between CompStat and Part 1 (serious) crime and arrest rates, including models with and without New York City due to its disproportionate influence. It complements Table 3 by assessing whether increased focus on low-level offenses under CompStat coincides with reduced attention to serious crime - a behavioural shift that may undermine overall public safety.

Table 4 from Eckhouse (2021) was omitted from this replication because it examines the share of arrests that are for Part 2 (minor) crimes, while Table 3, which was replicated in full, already captures the volume of Part 2 arrests both with and without New York City. Although Table 4 offers a proportion-based view, its insights are largely derivative of Table 3, as any increase in the share of Part 2 arrests depends on an increase in their volume or a relative decrease in other arrests.

Furthermore, Table 3 included the New York City robustness check, making it more comprehensive for assessing how CompStat influenced arrest patterns. Hence, table 4 was excluded to avoid redundancy.

Explanations for result deviations will be explained towards the end, summarising the replicability.

2.1 Table 3 Results

Table 1: Effect of Compstat on Number of Part 2 Arrests, Excluding NYC

	PART2arrests Part 2 Arrests		
	(1)	(2)	(3)
Compstat	4,028.177*	3,559.404*	3,581.072*
	(1,825.814)	(1,797.633)	(1,748.641)
Part 1 incidents	0.042**	0.036**	0.039**
	(0.007)	(0.007)	(0.007)
Total Population		0.166**	0.125**
		(0.028)	(0.028)
Black population (%)			320,364.200**
			(60,284.390)
White population (%)			19,492.110
			(50,858.280)
Observations	1,058	1,058	1,058
Note:			†p<0.1; *p<0.05; **p<0.01
			All regressions include year and agency fixed effects.

The Compstat coefficient on the number of part 2 arrests, excluding NYC, all had lower standard errors in my replication compared to the author's results (1,845.195, 1,814.232, 1,752.528).

Standard error of black population was higher in mine than the author's (59,901.810), and lower for white population (author's was 54,041.780).

Table 2: Effect of Compstat on Number of Part 2 Arrests, Including NYC

	PART2arrests Part 2 Arrests		
	(1)	(2)	(3)
Compstat	20,471.060**	20,294.410**	20,176.530**
	(6,082.105)	(5,543.778)	(5,540.931)
Part 1 incidents	−0.304**	−0.209**	−0.206**
	(0.019)	(0.018)	(0.018)
Total Population		0.823**	0.796**
		(0.052)	(0.055)
Black population (%)			398,618.400*
			(183,513.100)
White population (%)			94,309.340
			(160,192.800)
Observations	1,081	1,081	1,081
Note:			†p<0.1;*p<0.05; **p<0.01 All regressions include year and agency fixed effects.

For the CompStat coefficient, the standard errors in my replication had higher standard errors than the author's.

Standard errors of black and white population variables were lower in my replication than the author's.

In models 2 and 3, the standard errors of Part 1 incidents was a little bit higher in the author's (0.019), and standard error for total population was a little bit lower in the author's (0.053).

All else, with regards to coefficient values, statistical significance number of observations, other standard errors were replicated exactly, both including and excluding NYC. Meaning these deviations in standard errors didn't impact on the statistical significance of each variable.

2.2 Table 5 Results

The author had results for all 3 - auto theft, murder, rape - in one stargazer table, I chose to put them in 3 separate ones due to syntax issues I ran it.

Table 3: Effect of Compstat on Share of Auto Thefts Declared Unfounded

	Share Unfounded	
	unfoundautopct	
	No Demographics	With Demographics
	(1)	(2)
Compstat	0.005	0.003
	(0.003)	(0.003)
Total Population		-0.00000^{*}
		(0.000)
Black Population (%)		0.201^{+}
		(0.104)
White Population (%)		0.366^{**}
		(0.090)
Observations	900	900

Note: †p<0.1; *p<0.05; **p<0.01
All regressions include year and agency fixed effects.
8 cities report no data on unfoundedness and are excluded.

The standard error of white population coefficient on unfounded auto-theft was lower in my replication (0.090) than the author's (0.091).

Same with black population: 0.104 in my replication, 0.107 in the author's, but didn't impact on statistical significance.

Table 4: Effect of Compstat on Share of Rapes Declared Unfounded

	Share Unfounded	
	unfoundrapepct	
	No Demographics	With Demographics
	(1)	(2)
Compstat	0.020**	0.016**
	(0.005)	(0.005)
Total Population		−0.00000**
		(0.00000)
Black Population (%)		0.311^{+}
		(0.170)
White Population (%)		−0.242^{+}
		(0.145)
Observations	900	900
Note:		†p<0.1;*p<0.05; **p<0.01
		All regressions include year and agency fixed effects.
		8 cities report no data on unfoundedness and are excluded.

The standard error of black population on unfounded rapes in my replication (0.170) was lower than the author's (0.176).

The same for white population: 0.145 in my replication, and 0.154 in the author's The white population variable had a statistical significant coefficient at the 10% level from my replication, but not from the authors', which had no level of statistical significance.

This means the change in standard error had an impact on the statistical significance of the white population variable.

Table 5: Effect of Compstat on Share of Murders Declared Unfounded

	Share Unfounded	
	unfoundmurderpct	
	No Demographics	With Demographics
	(1)	(2)
Compstat	−0.006	−0.007
	(0.005)	(0.005)
Total Population		−0.000
		(0.000)
Black Population (%)		0.196
		(0.155)
White Population (%)		0.246^{+}
		(0.138)
Observations	900	900

Note: †p<0.1; *p<0.05; **p<0.01
All regressions include year and agency fixed effects.
8 cities report no data on unfoundedness and are excluded.

The standard error of the black population variable coefficient on unfounded murders in my replication (0.155) had a slightly lower standard error than the author's (0.157). All else are the same.

2.3 Table 6 Results

Table 6: Effect of Compstat on Number of Part 1 Incidents

	YEAR.ACT.ALL.FIELDS	
	No Demographics	With Demographics
	(1)	(2)
Compstat	243.249	−170.393
	(3,189.516)	(3,050.832)
Total Population	−0.116**	−0.109**
	(0.010)	(0.010)
Black population (%)		531,332.500**
		(106,629.700)
White population (%)		−114,194.600
		(90,506.010)
Observations	1,085	1,085
Note:		†p<0.1; *p<0.05; **p<0.01
		All regressions include year and agency fixed effects.

No differences in coefficients, statistical significance, and observations, but some differences in standard errors.

The author had higher standard errors for: CompStat, black population and white population. However, this didn't impact on statistical significance

Table 7: Effect of Compstat on Number of Part 1 Arrests (All Cities)

	PART1arrests		
	No Demographics	With Demographics	With Demographics + Part 1 Incidents
	(1)	(2)	(3)
Compstat	2,422.845*	2,334.490*	2,359.768*
	(1,195.633)	(1,177.967)	(982.613)
Total Population	−0.057**	−0.055**	−0.031**
	(0.004)	(0.004)	(0.003)
Black population (%)		153,564.700**	35,307.180
		(39,927.350)	(32,803.510)
White population (%)		−24,760.220	1,587.616
		(35,323.760)	(28,200.890)
Part 1 Incidents			0.222**
			(0.010)
Observations	1,081	1,081	1,081

Note: †p<0.1; *p<0.05; **p<0.01
All regressions include year and agency fixed effects.

For the effect of CompStat on Part 1 arrests in the model with no demographic controls and the model with demographic controls, the coefficient was statistically significant at the 5% level from my results (shown by the singular * in the table above, instead of a dagger), but was only at the 10% level from the author's.

This likely arises due to differences in standard errors, which are present in all variables, except for total population and Part 1 incidents.

Coefficients and all other levels of statistical significance were replicated.

Table 8: Effect of Compstat on Number of Part 1 Arrests (Excluding NYC)

	PART1arrests		
	No Demographics	With Demographics	With Demographics + Part 1 Incidents
	(1)	(2)	(3)
Compstat	391.243	359.220	93.064
	(709.485)	(703.551)	(675.735)
Total Population	0.015**	0.015**	0.011**
	(0.003)	(0.003)	(0.003)
Black population (%)		54,313.570*	29,942.660
		(23,083.150)	(22,449.460)
White population (%)		−32,329.960	−12,291.480
		(19,940.750)	(19,385.050)
Part 1 Incidents			0.100**
			(0.011)
Observations	1,058	1,058	1,058
Note:			†p<0.1; *p<0.05; **p<0.01
			All regressions include year and agency fixed effects.

A lot of differences in the standard errors of the variable coefficients, except for total population and Part 1 incidents.

Statistical significance of each variable, coefficients and number of observations were replicated.

2.4 Replication summary

The results were well-replicated, especially with regards to coefficients and number of observations. There were mild disparities, particularly in the standard errors of some variable coefficients, which in turn affected a few of the statistical significance levels.

The differences in standard errors may be due to necessary modifications I made while replicating the `bootr` function - for instance, renaming internal variables (like assigning `wild_sign` instead of overwriting `wild`) or adjusting error handling. While these changes were functionally equivalent, they may have introduced slight numerical variation.

However, the standard error values themselves were generally very close to those reported by the author, and in cases where standard errors were small (e.g. less than 1), the result was identical. This suggests that minor deviations in more variable coefficients are likely due to random variation in the resampling process rather than fundamental methodological differences.

In a few instances, variables that were statistically significant at one level (such as 1%) in the original paper became statistically significant at another level (such as 5%) in my replication, even though the direction and magnitude of the coefficients remained consistent.

These differences, while minor, highlight the sensitivity of inference to standard error estimation and reinforce the importance of transparent and reproducible code, especially for methods involving non-deterministic procedures like bootstrapping.

3 Part 3: Potential Improvements

The original paper relies on linear models with factor variables, such as year, which may be too rigid to capture non-linear patterns with those factor variables in the data. A more flexible approach - such as using a Generalised Additive Model (GAM) - can better account for potential non-linear relationships. For example, the effect of CompStat may not scale linearly with population size, and the relationship between time and outcomes may evolve in a smooth, continuous way rather than abruptly across discrete years. To support this flexibility, the year variable should be redefined as a numeric variable and modelled using a smooth function (e.g., `s(YEAR)` in a GAM). This allows the model to estimate gradual temporal trends without imposing a separate parameter for each year, which also improves interpretability and flexibility. GAMs also retain the ability to include fixed effects for cities through parametric terms, making them well-suited for panel data like this. This approach can yield better model fit (as reflected in a higher deviance explained and adjusted R-squared) and may reveal non-linear effects that the linear models miss.

The original methodology appears to rely on a traditional two-way fixed effects DiD framework, which assumes a common treatment time across units. However, CompStat adoption varied across cities—some implemented it as early as 1995, while others only adopted it in the 2000s. A staggered DiD framework better accommodates this real-world treatment timing heterogeneity. This improves internal validity by avoiding biased estimates that can result from incorrectly pooling units at different stages of treatment. Recent methodological advancements (e.g., S. Freedman et al., 2023) offer tools for implementing staggered DiD with proper attention to treatment timing, dynamic effects, and causal interpretation. Adopting these methods would strengthen the credibility of the paper's causal claims.

The wild bootstrap procedure used in the original code is computationally intensive and slows down replication significantly - especially when applied across many models, variables and observations. This poses barriers to reproducibility, scalability (e.g., adding more cities or longer time periods), and uptake by other researchers or policymakers. Optimising this code - for example, by using faster bootstrapping packages - would substantially reduce runtime and enhance usability by being able to conduct the study across more cities and longer time periods, which can challenge or strengthen the author's results. It may also allow for more extensive robustness checks or simulation studies.

4 Part 4: Improvement Implementation

To explore whether a more flexible modeling approach improves the fit and interpretability of the results, I estimated a generalised additive model (GAM) as an extension to the author's original linear regression. In particular, I replaced the linear effects of population, racial composition, and year with smooth functions using the `mgcv` package in R. Year was converted to numeric form to enable smoothing, and the `k = 5` argument (i.e. maximum number of degrees of freedom) was used for `POP` (population variable) to prevent excessive flexibility and reduce overfitting risk.

```
# Create GAM
library(mgcv) # GAM library
set.seed(95428) # use the author's seed
# create year variable that is numeric
maindata$YEAR_Numeric <- as.numeric(as.character(maindata$YEAR))
gam_pt2arrests <- gam(PART2arrests ~  HASCOMPSTAT + s(POP, k =5) + s(popblackpct) +
s(popwhitepct) + as.factor(AGENCY) + s(YEAR_Numeric), data = maindata)
# Population with max 5 degrees of freedom to prevent over-complexity
# over-complexity leads to an overfitting model
```

Table 9: Effect of Compstat on Part 2 Arrests (GAM) - including NYC

	Dependent variable:
	PART2arrests
Compstat	13,940.070***
	(4,422.713)
Observations	1,081
Note:	*p<0.1; **p<0.05; ***p<0.01

Looking at CompStat (extracted from the model), it remains a statistically significant predictor for Part2Arrests. The coefficient of CompStat - `13,940,070` - is lower than the author's models, where it was over 20,000.

```
##                        edf   Ref.df           F      p-value
## s(POP)          3.532509 3.861194 343.0289357 0.0000000000
## s(popblackpct)  1.000000 1.000000   5.3378193 0.0210660136
## s(popwhitepct)  1.852098 2.449742   0.2171676 0.7685653283
## s(YEAR_Numeric) 5.312759 6.445800   4.9926157 0.0000358906
```

The numerical predictors with the smooth function revealed statistical significance at the 5% level, with the exception of `popwhitepct` (as shown by the p-values).

`YEAR_Numeric` being statistically significant from the model supports my decision to convert `YEAR` to a numeric variable and model it with a smooth function.

The existence of such a relationship, such as long-term temporal dynamics, wasn't able to be captured from the author's linear regression model, because year was a factor.

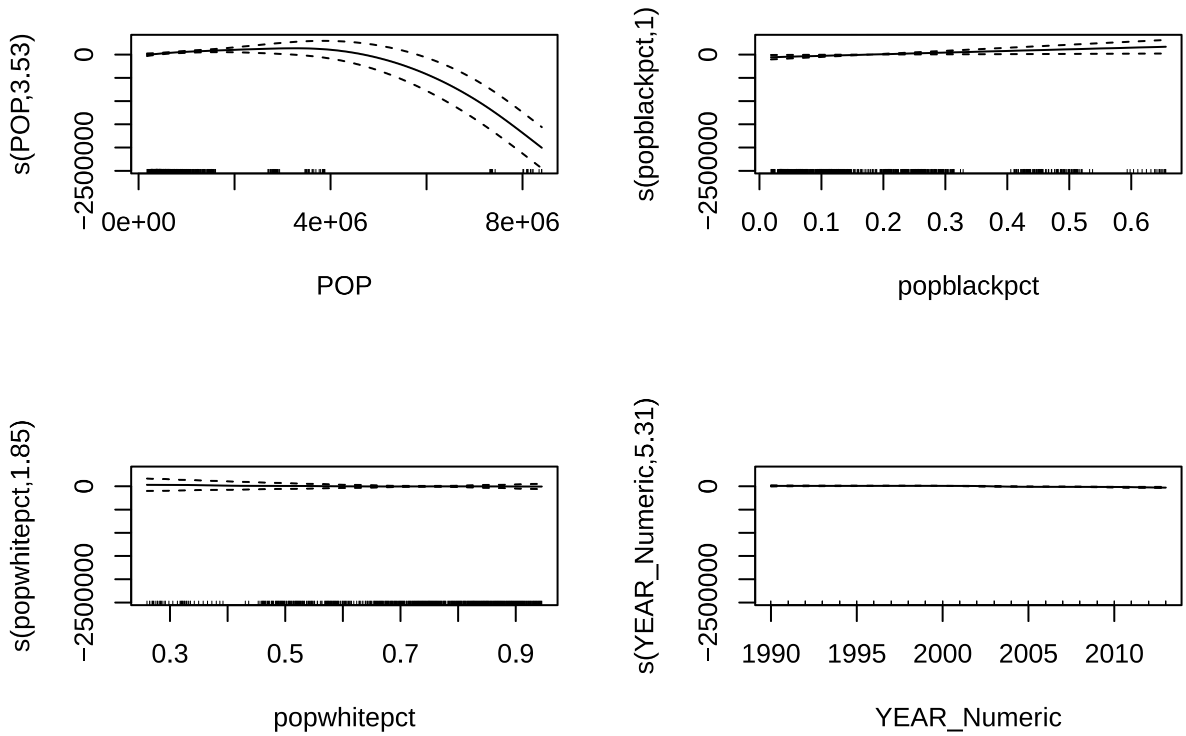
```
## GAM R-Squared Results:
## Adjusted R-squared: 0.809
## Deviance Explained: 81.9%
```

```
## Linear Regression R-squared results:
## R-squared: 0.630
## Adjusted R-squared: 0.604
```

The adjusted R-squared value of the GAM is much higher at `0.809`, whereas it's `0.6041` for the simple linear regression model. Deviance explained of the GAM is also higher in the GAM (`81.9%`) compared to the linear regression (60.4% from adjusted R-squared). This means the GAM explains much more of the deviance in the model's predictive power, capturing more of the real patterns in the data, leaving less unexpected "randomness" than the linear regression model.

While high R-squared values can sometimes indicate overfitting, especially in complex models, the use of penalised smooth terms, such as with `YEAR_Numeric` having maximum 5 degrees of freedom in the GAM helps control for this risk by automatically limiting model flexibility. Moreover, the smooth terms captured meaningful non-linear trends without excessive complexity, supporting a balance between fit and generalisability.

These results further support my decision to convert YEAR to a numeric and model those 4 predictors - POP, popblackpct, popwhitepct, YEAR_numeric - with a smooth function. Having smooth, non-linear dynamics captured in the GAM meant the model had greater explanatory power, as shown by its increase in R-squared values.



These plots represent the partial effects of the smooth terms from the GAM, visualising the relationships between the numerical variables and Part 2 arrests. These plots help visualise potential non-linearities in the effects of these variables.

- Population (`POP`): The smooth function of population reveals a non-linear relationship with the number of Part 2 arrests. Initially, the effect increases with population size, but beyond a certain threshold, the relationship reverses — suggesting that in very large cities, population growth may be associated with fewer additional arrests. This curvature would not have been captured under the linear specification used in the original paper.
- Percentage of Black Population (`popblackpct`): The effect of Black population proportion on Part 2 arrests is mostly linear and positive, with a slight upward slope. While the curvature is modest, the effect is statistically significant, suggesting extra policing for less serious crimes in neighbourhoods with higher black populations.
- Percentage of White Population (`popwhitepct`): The plot is nearly flat and statistically insignificant, suggesting no discernible non-linear effect on arrest counts in this context.
- Year (modeled as `YEAR_Numeric`): Although the smooth function for year appears flat, it is statistically significant. This implies the model detects subtle year-on-year variation not captured by simply using year as a set of dummy variables (as in the author's model). Modeling year as a continuous smooth function reduces overparameterisation and provides a clearer picture of temporal trends.

The GAM approach improves on the author's linear specification by relaxing the assumption that all covariates have constant, linear effects. For instance, the relationship between city population and arrests is unlikely to be perfectly linear across all scales. By using smooth terms, we allow the data to determine the shape of these relationships. This approach offers more flexibility, avoids potential mis-specification, and provides better model fit - as supported by the GAM's higher deviance explained and adjusted R-squared. It also reveals insights (e.g., a slight concave, diminishing returns to population size) that would be obscured by traditional linear modeling.

5 Code appendix

```
# this chunk contains code that sets global options for the entire .Rmd.
# we use include=FALSE to suppress it from the top of the document, but it will still appear in the app
knitr::opts_chunk$set(echo=FALSE, warning=FALSE, message=FALSE, linewidth=60)
setwd("/Users/shujaali/gv330_2425/summative-replication-shuja-ali298") # Using my WD
# Load libraries
library(stargazer)
library(ggplot2)
library(ggthemes)
library(dplyr)
library(magrittr)
library(gtools)
library(sampling)
library(extraDistr)
# Load data
maindata <- read.csv(file = "eckhouse_metricsmanagement-data.csv")
# Copied from the author with some differences (explained further down)
bootr <- function(lm.model, lm.data, n.b = 100, wild = F, dist = "normal", clusters = NULL, mainv = 1, s
  set.seed(95428)
  require(sampling)
  require(extraDistr)
  require(gtools)

  residuals <- resid(lm.model)
  lm.formula <- as.formula(formula(lm.model))
  lm.formula <- as.formula(paste("Y.star ~ ", lm.formula[3], sep = ""))

  betas.bs <- matrix(nrow = 1, ncol=1)
  betas.bs <- as.data.frame(betas.bs)
  colnames(betas.bs) <- "(Intercept)"

  if (!is.null(clusters)) {
    for(i in 1:n.b) {
      clustervar <- unique(lm.data[, clusters])
      blocks <- sample(1:length(clustervar) + 1, replace = T)
      sampler <- NA

      for (j in 1:length(blocks)) {
        block.extract <- which(lm.data[, clusters] == clustervar[blocks[j]])
        sampler <- c(sampler, block.extract)
      }

      sampler <- sampler[-1]
      sampler <- sampler[1:nrow(lm.data)]

      e.star.cluster <- residuals[sampler]

      if (wild == T) {
        if (dist == "rademacher") {
          # A difference from the author. The author assigned to wild, I chose to assign to a new varia
          wild_sign <- rsign(length(e.star.cluster))
        } else if (dist == "normal") {
```

```
        wild_sign <- rnorm(length(e.star.cluster), mean = 0, sd = 1)
      } else stop("dist argument not supported")

      e.star.cluster <- e.star.cluster * wild_sign
    }

    Y.star <- model.matrix.lm(lm.model, na.action = 'na.pass') %*% coef(lm.model) + e.star.cluster
    lm.data$Y.star <- Y.star
    beta.star <- coef(lm(lm.formula, data = lm.data, na.action = 'na.omit'))

    betas.bs <- smartbind(betas.bs, as.data.frame(t(beta.star)))
  }

} else if (is.null(clusters)) {
  for(i in 1:n.b) {
    e.star <- residuals[sample(1:length(residuals), replace=TRUE)]

    if (wild == T) {
      if (dist == "rademacher") {
        wild_sign <- rsign(length(e.star))
      } else if (dist == "normal") {
        wild_sign <- rnorm(length(e.star), mean = 0, sd = 1)
      } else stop("dist argument not supported")
      # Using stop, where the author used return() to make errors clear

      e.star <- e.star * wild_sign
    }

    Y.star <- model.matrix.lm(lm.model, na.action = 'na.pass') %*% coef(lm.model) + e.star
    lm.data$Y.star <- Y.star
    beta.star <- coef(lm(lm.formula, data = lm.data, na.action = 'na.omit'))

    betas.bs <- smartbind(betas.bs, as.data.frame(t(beta.star)))
  }
}

betas.bs <- betas.bs[2:nrow(betas.bs), ]

v <- mainv + 1
mean.boot <- mean(betas.bs[, v], na.rm = T)
sd.boot <- sd(betas.bs[, v], na.rm = T)

sigmin <- (sig/2)
sigmax <- 1-(sig/2)

confint.boot <- quantile(betas.bs[, v], probs = c(sigmin, sigmax))

booted <- list(betas.bs, mean.boot, sd.boot, confint.boot)

return(booted)
}
# This was copied directly from the author (right before sesT3_inclNYC part)
part2arrests <- lm(PART2arrests ~  HASCOMPSTAT + POP + as.factor(AGENCY) + as.factor(YEAR) , data = mai
```

```
part2arrests.pt1control <- lm(PART2arrests ~  HASCOMPSTAT + POP + YEAR.ACT.ALL.FIELDS  + as.factor(AGENC
part2arrests.pt1controldemog <- lm(PART2arrests ~  HASCOMPSTAT + POP + YEAR.ACT.ALL.FIELDS  + popblackp


part2arrestsDropNYC <- lm(PART2arrests ~  HASCOMPSTAT + POP + as.factor(AGENCY) + as.factor(YEAR), data
part2arrestsDropNYC.pt1control <- lm(PART2arrests ~  HASCOMPSTAT + POP + YEAR.ACT.ALL.FIELDS  + as.fact
part2arrestsDropNYC.pt1controldemog <- lm(PART2arrests ~  HASCOMPSTAT + POP + YEAR.ACT.ALL.FIELDS  + po

part2arrests.boot <- bootr(lm.model = part2arrests, lm.data = maindata, n.b = 1000, wild = TRUE, dist =
part2arrests.pt1control.boot <- bootr(lm.model = part2arrests.pt1control, lm.data = maindata, n.b = 100
part2arrests.pt1controldemog.boot <- bootr(lm.model = part2arrests.pt1controldemog, lm.data = maindata,

part2arrestsDropNYC.boot <- bootr(lm.model = part2arrestsDropNYC, lm.data = subset(maindata, AGENCY !=
part2arrestsDropNYC.pt1control.boot <- bootr(lm.model = part2arrestsDropNYC.pt1control, lm.data = subse
part2arrestsDropNYC.pt1controldemog.boot <- bootr(lm.model = part2arrestsDropNYC.pt1controldemog, lm.da


#run bootr for each model to generate bootstrapped standard error estimates for population (second vari
part2arrests.boot.pop <- bootr(lm.model = part2arrests, lm.data = maindata, n.b = 1000, wild = T, dist
part2arrests.pt1control.boot.pop <- bootr(lm.model = part2arrests.pt1control, lm.data = maindata, n.b =
part2arrests.pt1controldemog.boot.pop <- bootr(lm.model = part2arrests.pt1controldemog, lm.data = mainda

part2arrestsDropNYC.boot.pop <- bootr(lm.model = part2arrestsDropNYC, lm.data = subset(maindata, AGENCY
part2arrestsDropNYC.pt1control.boot.pop <- bootr(lm.model = part2arrestsDropNYC.pt1control, lm.data = s
part2arrestsDropNYC.pt1controldemog.boot.pop <- bootr(lm.model = part2arrestsDropNYC.pt1controldemog, l



#run bootr for each model to generate bootstrapped standard error estimates for part 1 arrests (third v

part2arrests.pt1control.boot.pt1 <- bootr(lm.model = part2arrests.pt1control, lm.data = maindata, n.b =
part2arrests.pt1controldemog.boot.pt1 <- bootr(lm.model = part2arrests.pt1controldemog,
                                               lm.data = maindata, n.b = 1000, wild = T, dist = "radema

part2arrestsDropNYC.pt1control.boot.pt1 <- bootr(lm.model = part2arrestsDropNYC.pt1control, lm.data = s
part2arrestsDropNYC.pt1controldemog.boot.pt1 <- bootr(lm.model = part2arrestsDropNYC.pt1controldemog,
                                                      lm.data = subset(maindata, AGENCY != "NEW YORK"),
                                                      n.b = 1000, wild = T, dist = "rademacher", mainv

#run bootr for each model to generate bootstrapped standard error estimates for % of population that is
part2arrests.pt1controldemog.boot.popblackpct <- bootr(lm.model = part2arrests.pt1controldemog, lm.data
part2arrestsDropNYC.pt1controldemog.boot.popblackpct <- bootr(lm.model = part2arrestsDropNYC.pt1control
                                                              lm.data = subset(maindata, AGENCY != "NEW
                                                              n.b = 1000, wild = T, dist = "rademacher"


#run bootr for each model to generate bootstrapped standard error estimates for % of population that is
part2arrests.pt1controldemog.boot.popwhitepct <- bootr(lm.model = part2arrests.pt1controldemog,
                                                       lm.data = maindata, n.b = 1000,
                                                       wild = T, dist = "rademacher", mainv = 5)
part2arrestsDropNYC.pt1controldemog.boot.popwhitepct <- bootr(lm.model = part2arrestsDropNYC.pt1control
                                                              lm.data = subset(maindata, AGENCY != "NEW
                                                              n.b = 1000, wild = T, dist = "rademacher"
```

```
#collect standard errors
sesT3 <- as.data.frame(cbind(coef(summary(part2arrests))[, 2], coef(summary(part2arrests.pt1control))[,
                             coef(summary(part2arrests.pt1controldemog))[, 2], coef(summary(part2arrests
                             coef(summary(part2arrestsDropNYC.pt1control))[, 2], coef(summary(part2arre
sesT3["HASCOMPSTAT", ] <- c(part2arrests.boot[[3]], part2arrests.pt1control.boot[[3]], part2arrests.pt1c
                            part2arrestsDropNYC.boot[[3]], part2arrestsDropNYC.pt1control.boot[[3]], pa

sesT3["POP", ] <- c(part2arrests.boot.pop[[3]], part2arrests.pt1control.boot.pop[[3]], part2arrests.pt1c
                    part2arrestsDropNYC.boot.pop[[3]], part2arrestsDropNYC.pt1control.boot.pop[[3]], pa


sesT3["YEAR.ACT.ALL.FIELDS", ] <- c(NA, part2arrests.pt1control.boot.pt1[[3]], part2arrests.pt1controlde
                                    NA, part2arrestsDropNYC.pt1control.boot.pt1[[3]], part2arrestsDropNY

sesT3["popwhitepct", ] <- c(NA, NA, part2arrests.pt1controldemog.boot.popwhitepct[[3]],
                            NA, NA, part2arrestsDropNYC.pt1controldemog.boot.popwhitepct[[3]])


sesT3["popblackpct", ] <- c(NA, NA, part2arrests.pt1controldemog.boot.popblackpct[[3]],
                            NA, NA, part2arrestsDropNYC.pt1controldemog.boot.popblackpct[[3]])

# Was added in for the table splitting - mapping standard errors on each table, whereas the author had
sesT3_inclNYC <- sesT3[, 1:3]
sesT3_exclNYC <- sesT3[, 4:6]
# Table 5: share of rapes unfounded + placebos (murder, auto theft)
# Copied from the author
#drop agencies with no unfoundeds ever
unique(subset(maindata[ , c("AGENCY", "YEAR")], maindata$unfoundassaultpct == 0))
unique(subset(maindata$COMPSTATDATE, maindata$AGENCY == "MILWAUKEE"))

#DROP AUSTIN, NASHVILLE, DETROIT, CHICAGO, DENVER, BAKERSFIELD, VIRGINIA BEACH, MILWAUKEE -- cities mis

miunfound <- c("AUSTIN", "NASHVILLE", "DETROIT", "CHICAGO", "DENVER", "BAKERSFIELD", "VIRGINIA BEACH",


# run models
unfoundrape.pct <- lm(unfoundrapepct ~ HASCOMPSTAT + as.factor(YEAR) + as.factor(AGENCY), data = subset
unfoundrape.pct.demog <- lm(unfoundrapepct ~ HASCOMPSTAT + POP + popblackpct + popwhitepct + as.factor(
unfoundauto.pct <- lm(unfoundautopct ~ HASCOMPSTAT + as.factor(YEAR) + as.factor(AGENCY), data = subset
unfoundauto.pct.demog <- lm(unfoundautopct ~ HASCOMPSTAT + POP + popblackpct + popwhitepct + as.factor(
unfoundmurder.pct <- lm(unfoundmurderpct ~ HASCOMPSTAT + as.factor(YEAR) + as.factor(AGENCY), data = su
unfoundmurder.pct.demog <- lm(unfoundmurderpct ~ HASCOMPSTAT + POP + popblackpct + popwhitepct + as.fac



#use bootr to generate standard errors for main independent variable (Compstat)
unfoundrape.pct.boot <- bootr(lm.model = unfoundrape.pct, lm.data = subset(maindata, !as.character(AGEN
unfoundrape.pct.demog.boot <- bootr(lm.model = unfoundrape.pct.demog, lm.data = subset(maindata, !as.cha
unfoundauto.pct.boot <- bootr(lm.model = unfoundauto.pct, lm.data = subset(maindata, !as.character(AGEN
unfoundauto.pct.demog.boot <- bootr(lm.model = unfoundauto.pct.demog, lm.data = subset(maindata, !as.cha
unfoundmurder.pct.boot <- bootr(lm.model = unfoundmurder.pct, lm.data = subset(maindata, !as.character(
unfoundmurder.pct.demog.boot <- bootr(lm.model = unfoundmurder.pct.demog, lm.data = subset(maindata, !a
```

```
#use bootr to generate standard errors for population (second variable -- specified via order in bootr)
unfoundrape.pct.demog.boot.pop <- bootr(lm.model = unfoundrape.pct.demog, lm.data = subset(maindata, !a
unfoundauto.pct.demog.boot.pop <- bootr(lm.model = unfoundauto.pct.demog, lm.data = subset(maindata, !a
unfoundmurder.pct.demog.boot.pop <- bootr(lm.model = unfoundmurder.pct.demog, lm.data = subset(maindata

#use bootr to generate standard errors for % of population that is Black (third variable)
unfoundrape.pct.demog.boot.popblackpct <- bootr(lm.model = unfoundrape.pct.demog, lm.data = subset(main
unfoundauto.pct.demog.boot.popblackpct <- bootr(lm.model = unfoundauto.pct.demog, lm.data = subset(main
unfoundmurder.pct.demog.boot.popblackpct <- bootr(lm.model = unfoundmurder.pct.demog, lm.data = subset(


#use bootr to generate standard errors for % of population that is white (fourth variable)
unfoundrape.pct.demog.boot.popwhitepct <- bootr(lm.model = unfoundrape.pct.demog, lm.data = subset(main
unfoundauto.pct.demog.boot.popwhitepct <- bootr(lm.model = unfoundauto.pct.demog, lm.data = subset(main
unfoundmurder.pct.demog.boot.popwhitepct <- bootr(lm.model = unfoundmurder.pct.demog, lm.data = subset(

sesT5 <- as.data.frame(cbind(coef(summary(unfoundrape.pct))[, 2], coef(summary(unfoundrape.pct.demog))[
sesT5["HASCOMPSTAT", ] <- c(unfoundrape.pct.boot[[3]], unfoundrape.pct.demog.boot[[3]], unfoundauto.pct
sesT5["POP", ] <- c(NA, unfoundrape.pct.demog.boot.pop[[3]], NA, unfoundauto.pct.demog.boot.pop[[3]], N
sesT5["popblackpct", ] <- c(NA, unfoundrape.pct.demog.boot.popblackpct[[3]], NA, unfoundauto.pct.demog.
sesT5["popwhitepct", ] <- c(NA, unfoundrape.pct.demog.boot.popwhitepct[[3]], NA, unfoundauto.pct.demog.
# Table 6: crime and arrest rates
# Copied from the author until sesT6
part1incidents <- lm(YEAR.ACT.ALL.FIELDS ~ HASCOMPSTAT + POP + as.factor(AGENCY) + as.factor(YEAR) , da
part1incidents.demog <- lm(YEAR.ACT.ALL.FIELDS ~ HASCOMPSTAT  + POP + popblackpct + popwhitepct+ as.fac
part1arrests <- lm(PART1arrests ~ HASCOMPSTAT  +  POP + as.factor(AGENCY) + as.factor(YEAR), data = mai
part1arrests.demog <- lm(PART1arrests ~ HASCOMPSTAT  + POP + popblackpct + popwhitepct + as.factor(AGEN
part1arrests.demog.pt1control <- lm(PART1arrests ~ HASCOMPSTAT + POP + popblackpct + popwhitepct + YEAR


part1arrestsDropNYC <- lm(PART1arrests ~  HASCOMPSTAT + POP + as.factor(AGENCY) + as.factor(YEAR) , dat
part1arrestsDropNYC.demog <- lm(PART1arrests ~  HASCOMPSTAT + POP +popblackpct + popwhitepct +  as.fact
part1arrestsDropNYC.pt1controldemog <- lm(PART1arrests ~  HASCOMPSTAT + POP + popblackpct + popwhitepct


#use bootr to generate standard errors for main independent variable (Compstat)
part1incidents.boot <- bootr(lm.model = part1incidents, lm.data = maindata, n.b = 1000, wild = T, dist 
part1incidents.demog.boot <- bootr(lm.model = part1incidents.demog, lm.data = maindata, n.b = 1000, wil
part1arrests.boot <- bootr(lm.model = part1arrests, lm.data = maindata, n.b = 1000, wild = T, dist = "r
part1arrests.demog.boot <- bootr(lm.model = part1arrests.demog, lm.data = maindata, n.b = 1000, wild = 
part1arrests.demog.pt1control.boot <- bootr(lm.model = part1arrests.demog.pt1control, lm.data = maindat


part1arrestsDropNYC.boot <- bootr(lm.model = part1arrestsDropNYC, lm.data = subset(maindata, AGENCY != 
part1arrestsDropNYC.demog.boot <- bootr(lm.model = part1arrestsDropNYC.demog, lm.data = subset(maindata
part1arrestsDropNYC.pt1controldemog.boot <- bootr(lm.model = part1arrestsDropNYC.pt1controldemog, lm.da

#use bootr to generate standard errors for population (second variable -- specified via order in bootr)
part1incidents.boot.pop <- bootr(lm.model = part1incidents, lm.data = maindata, n.b = 1000, wild = T, d
part1incidents.demog.boot.pop <- bootr(lm.model = part1incidents.demog, lm.data = maindata, n.b = 1000,
part1arrests.boot.pop <- bootr(lm.model = part1arrests, lm.data = maindata, n.b = 1000, wild = T, dist 
part1arrests.demog.boot.pop <- bootr(lm.model = part1arrests.demog, lm.data = maindata, n.b = 1000, wil
part1arrests.demog.pt1control.boot.pop <- bootr(lm.model = part1arrests.demog.pt1control, lm.data = mai
```

```
part1arrestsDropNYC.boot.pop <- bootr(lm.model = part1arrestsDropNYC, lm.data = subset(maindata, AGENCY
part1arrestsDropNYC.demog.boot.pop <- bootr(lm.model = part1arrestsDropNYC.demog, lm.data = subset(mainc
part1arrestsDropNYC.pt1controldemog.boot.pop <- bootr(lm.model = part1arrestsDropNYC.pt1controldemog, lr


#use bootr to generate standard errors for % of population that is Black (third variable)
part1incidents.demog.boot.popblackpct <- bootr(lm.model = part1incidents.demog, lm.data = maindata, n.b
part1arrests.demog.boot.popblackpct <- bootr(lm.model = part1arrests.demog, lm.data = maindata, n.b = 1(
part1arrests.demog.pt1control.boot.popblackpct <- bootr(lm.model = part1arrests.demog.pt1control, lm.dat

part1arrestsDropNYC.pt1controldemog.boot.popblackpct <- bootr(lm.model = part1arrestsDropNYC.demog, lm.c
part1arrestsDropNYC.pt1controldemog.boot.popblackpct <- bootr(lm.model = part1arrestsDropNYC.pt1controlc


#use bootr to generate standard errors for % of population that is white (fourth variable)
part1incidents.demog.boot.popwhitepct <- bootr(lm.model = part1incidents.demog, lm.data = maindata, n.b
part1arrests.demog.boot.popwhitepct <- bootr(lm.model = part1arrests.demog, lm.data = maindata, n.b = 1(
part1arrests.demog.pt1control.boot.popwhitepct <- bootr(lm.model = part1arrests.demog.pt1control, lm.dat

part1arrestsDropNYC.pt1controldemog.boot.popwhitepct <- bootr(lm.model = part1arrestsDropNYC.demog, lm.c
part1arrestsDropNYC.pt1controldemog.boot.popwhitepct <- bootr(lm.model = part1arrestsDropNYC.pt1controlc

part1arrestsDropNYC.demog.boot.popblackpct <- bootr(lm.model = part1arrestsDropNYC.demog, lm.data = subs
part1arrestsDropNYC.demog.boot.popwhitepct <- bootr(lm.model = part1arrestsDropNYC.demog, lm.data = subs

#use bootr to generate standard errors for number of Part 1 incidents (fifth variable)
part1arrests.demog.pt1control.boot.pt1 <- bootr(lm.model = part1arrests.demog.pt1control, lm.data = mair

part1arrestsDropNYC.pt1controldemog.boot.pt1 <- bootr(lm.model = part1arrestsDropNYC.pt1controldemog, lr

# This sesT6 code required changes on my part, due to some syntax and missing variables in the author's
sesT6 <- as.data.frame(cbind(
    coef(summary(part1incidents))[, 2],
    coef(summary(part1incidents.demog))[, 2],
    coef(summary(part1arrests))[, 2],
    coef(summary(part1arrests.demog))[, 2],
    coef(summary(part1arrests.demog.pt1control))[, 2],
    coef(summary(part1arrestsDropNYC))[, 2],
    coef(summary(part1arrestsDropNYC.demog))[, 2],
    coef(summary(part1arrestsDropNYC.pt1controldemog))[, 2]
))

sesT6["HASCOMPSTAT", ] <- c(part1incidents.boot[[3]], part1incidents.demog.boot[[3]],
                           part1arrests.boot[[3]], part1arrests.demog.boot[[3]],
                           part1arrests.demog.pt1control.boot[[3]], part1arrestsDropNYC.boot[[3]],
                           part1arrestsDropNYC.demog.boot[[3]], part1arrestsDropNYC.pt1controldemog.boot

sesT6["POP", ] <- c(part1incidents.boot.pop[[3]], part1incidents.demog.boot.pop[[3]],
                            part1arrests.boot.pop[[3]], part1arrests.demog.boot.pop[[3]],
                            part1arrests.demog.pt1control.boot.pop[[3]], part1arrestsDropNYC.boot.pop[[3
                            part1arrestsDropNYC.demog.boot.pop[[3]], part1arrestsDropNYC.pt1controldemog
```

```r
sesT6["popblackpct", ] <- c(NA, part1incidents.demog.boot.popblackpct[[3]],
                    NA, part1arrests.demog.boot.popblackpct[[3]],
                    part1arrests.demog.pt1control.boot.popblackpct[[3]], NA,
                    part1arrestsDropNYC.demog.boot.popblackpct[[3]], part1arrestsDropNYC.pt1controldemog

sesT6["popwhitepct", ] <- c(NA, part1incidents.demog.boot.popwhitepct[[3]],
                            NA, part1arrests.demog.boot.popwhitepct[[3]],
                            part1arrests.demog.pt1control.boot.popwhitepct[[3]], NA,
                            part1arrestsDropNYC.demog.boot.popwhitepct[[3]], part1arrestsDropNYC.pt1con
# Table 3 without NYC, using Latex type output for PDF display. Copied directly with modifications
# Original version had 6 models in one stargazer call, but this caused formatting issues
# Split into two separate tables (All Cities and Excluding NYC)
stargazer(part2arrestsDropNYC, part2arrestsDropNYC.pt1control, part2arrestsDropNYC.pt1controldemog,
          omit = c("AGENCY", "YEAR\\)", "Constant"),
          title = "Effect of Compstat on Number of Part 2 Arrests, Excluding NYC",
          column.separate = c(3, 3), column.labels = c("Part 2 Arrests"),
          covariate.labels = c("Compstat", "Part 1 incidents", "Total Population", "Black population (\\
          dep.var.caption = "", digits = 3, keep.stat = "n",
          star.char = c("+", "*", "**"),
          star.cutoffs = c(0.1, 0.05, 0.01),
          notes = c("$^{\\dagger}$p$<$0.1;$^{*}$p$<$0.05; $^{**}$p$<$0.01", "All regressions include yea
          se = sesT3_exclNYC,
          type= "latex", header=FALSE,
          float.env = "table", table.placement = "H") # This was for placement in the pdf document. Wit
# Table 3 with NYC
stargazer(part2arrests, part2arrests.pt1control, part2arrests.pt1controldemog,
          omit = c("AGENCY", "YEAR\\)", "Constant"),
          title = "Effect of Compstat on Number of Part 2 Arrests, Including NYC",
          column.separate = c(3, 3), column.labels = c("Part 2 Arrests"),
          covariate.labels = c("Compstat", "Part 1 incidents", "Total Population", "Black population (\\
          dep.var.caption = "", digits = 3, keep.stat = "n",
          star.char = c("+", "*", "**"),
          star.cutoffs = c(0.1, 0.05, 0.01),
          notes = c("$^{\\dagger}$p$<$0.1;$^{*}$p$<$0.05; $^{**}$p$<$0.01", "All regressions include yea
          se = sesT3_inclNYC,
          type= "latex", header=FALSE,
          float.env = "table", table.placement = "H")
# Unfounded GTA

stargazer(
  unfoundauto.pct, unfoundauto.pct.demog,
  title = "Effect of Compstat on Share of Auto Thefts Declared Unfounded",
  dep.var.caption = "Share Unfounded",
  column.labels = c("No Demographics", "With Demographics"),
  omit = c("AGENCY", "YEAR", "Constant"),
  covariate.labels = c("Compstat", "Total Population", "Black Population (\\%)", "White Population (\\%
  digits = 3,
  keep.stat = "n",
  star.char = c("+", "*", "**"),
  star.cutoffs = c(0.1, 0.05, 0.01),
  notes = c(
    "$^{\\dagger}$p$<$0.1;$^{*}$p$<$0.05; $^{**}$p$<$0.01",
    "All regressions include year and agency fixed effects.",
```

```r
    "8 cities report no data on unfoundedness and are excluded."
  ), # stat significance keys
  notes.append = FALSE,
  se = sesT5[, 3:4],
  type = "latex",
  header=FALSE,
  float.env = "table", table.placement = "H" # PDF placement
)
# Unfounded rape
stargazer(
  unfoundrape.pct, unfoundrape.pct.demog,
  title = "Effect of Compstat on Share of Rapes Declared Unfounded",
  dep.var.caption = "Share Unfounded",
  column.labels = c("No Demographics", "With Demographics"),
  omit = c("AGENCY", "YEAR", "Constant"),
  covariate.labels = c("Compstat", "Total Population", "Black Population (\\%)", "White Population (\\%
  digits = 3,
  keep.stat = "n",
  star.char = c("+", "*", "**"),
  star.cutoffs = c(0.1, 0.05, 0.01),
  notes = c(
    "$^{\\dagger}$p$<$0.1;$^{*}$p$<$0.05; $^{**}$p$<$0.01",
    "All regressions include year and agency fixed effects.",
    "8 cities report no data on unfoundedness and are excluded."
  ), # stat significance keys
  notes.append = FALSE,
  se = sesT5[, 1:2],
  type = "latex",
  header=FALSE,
  float.env = "table", table.placement = "H" # PDF placement
)
# Unfounded murder stargazer table

stargazer(
  unfoundmurder.pct, unfoundmurder.pct.demog,
  title = "Effect of Compstat on Share of Murders Declared Unfounded",
  dep.var.caption = "Share Unfounded",
  column.labels = c("No Demographics", "With Demographics"),
  omit = c("AGENCY", "YEAR", "Constant"),
  covariate.labels = c("Compstat", "Total Population", "Black Population (\\%)", "White Population (\\%
  digits = 3,
  keep.stat = "n",
  star.char = c("+", "*", "**"),
  star.cutoffs = c(0.1, 0.05, 0.01),
  notes = c(
    "$^{\\dagger}$p$<$0.1;$^{*}$p$<$0.05; $^{**}$p$<$0.01",
    "All regressions include year and agency fixed effects.",
    "8 cities report no data on unfoundedness and are excluded."
  ), # stat significance keys
  notes.append = FALSE,
  se = sesT5[, 5:6],
  type = "latex",
  header=FALSE,
```

```
  float.env = "table", table.placement = "H" # PDF placement
)
# Table of CompStat effect on Part 1 incidents
stargazer(
  part1incidents, part1incidents.demog,
  title = "Effect of Compstat on Number of Part 1 Incidents",
  column.labels = c("No Demographics", "With Demographics"),
  covariate.labels = c("Compstat", "Total Population", "Black population (\\%)", "White population (\\%
  dep.var.caption = "",
  keep.stat = "n",
  omit = c("AGENCY", "YEAR\\)", "Constant"),
  se = sesT6[, 1:2],
  star.char = c("+", "*", "**"),
  star.cutoffs = c(0.1, 0.05, 0.01),
  notes = c("$^{\\dagger}$p$<$0.1; $^{*}$p$<$0.05; $^{**}$p$<$0.01", "All regressions include year and a
  notes.append = FALSE,
  type = "latex",
  header=FALSE,
  float.env = "table", table.placement = "H" # PDF placement
)
# Table on CompStat effect on part 1 arrests
stargazer(
  part1arrests, part1arrests.demog, part1arrests.demog.pt1control,
  omit = c("AGENCY", "YEAR\\)", "Constant"),
  title = "Effect of Compstat on Number of Part 1 Arrests (All Cities)",
  column.labels = c("No Demographics", "With Demographics", "With Demographics + Part 1 Incidents"),
  covariate.labels = c("Compstat", "Total Population", "Black population (\\%)", "White population (\\%
  dep.var.caption = "", digits = 3, keep.stat = "n",
  star.char = c("+", "*", "**"),
  star.cutoffs = c(0.1, 0.05, 0.01),
  notes = c("$^{\\dagger}$p$<$0.1; $^{*}$p$<$0.05; $^{**}$p$<$0.01", "All regressions include year and a
  notes.append = FALSE,
  se = sesT6[, 3:5],
  type = "latex",
  header=FALSE,
  float.env = "table", table.placement = "H" # PDF placement
)
# Table of effect of CompStat on Part 1 arrests without NYC
stargazer(
  part1arrestsDropNYC, part1arrestsDropNYC.demog, part1arrestsDropNYC.pt1controldemog,
  omit = c("AGENCY", "YEAR\\)", "Constant"),
  title = "Effect of Compstat on Number of Part 1 Arrests (Excluding NYC)",
  column.labels = c("No Demographics", "With Demographics", "With Demographics + Part 1 Incidents"),
  covariate.labels = c("Compstat", "Total Population", "Black population (\\%)", "White population (\\%
  dep.var.caption = "", digits = 3, keep.stat = "n",
  star.char = c("+", "*", "**"),
  star.cutoffs = c(0.1, 0.05, 0.01),
  notes = c("$^{\\dagger}$p$<$0.1; $^{*}$p$<$0.05; $^{**}$p$<$0.01", "All regressions include year and a
  notes.append = FALSE,
  se = sesT6[, 6:8],
  type = "latex",
  header=FALSE,
  float.env = "table", table.placement = "H"
```

```
)
# Create GAM
library(mgcv) # GAM library
set.seed(95428) # use the author's seed
# create year variable that is numeric
maindata$YEAR_Numeric <- as.numeric(as.character(maindata$YEAR))
gam_pt2arrests <- gam(PART2arrests ~  HASCOMPSTAT + s(POP, k =5) + s(popblackpct) +
s(popwhitepct) + as.factor(AGENCY) + s(YEAR_Numeric), data = maindata)
# Population with max 5 degrees of freedom to prevent over-complexity
# over-complexity leads to an overfitting model
# Create stargazer table for effect of CompStat on part 2 arrests in the GAM
stargazer(gam_pt2arrests, type = "latex",
          title = "Effect of Compstat on Part 2 Arrests (GAM) - including NYC",
          covariate.labels = c("Compstat"),
          omit = c("as.factor", "Intercept", "s()"),
          keep.stat = "n", digits = 3, header=FALSE,
          float.env = "table", table.placement = "H")
summary(gam_pt2arrests)$s.table
set.seed(95428) # use author's seed
gam_summary <- summary(gam_pt2arrests) # store GAM summary statistics

# Extract adjusted R-squared
adj_r_squared <- gam_summary$r.sq
dev_explained <- gam_summary$dev.expl

# Call back author's part2arrests variable - the lm model
lm_summary <- summary(part2arrests) # store summary to extract metrics

# Extract lm R-squared values
adj_r_squared_lm <- lm_summary$adj.r.squared
r_squared_lm <- lm_summary$r.squared

cat(sprintf("GAM R-Squared Results:\nAdjusted R-squared: %.3f\nDeviance Explained: %.1f%%", adj_r_square

cat(sprintf("Linear Regression R-squared results:\nR-squared: %.3f\nAdjusted R-squared: %.3f", r_square
plot(gam_pt2arrests, pages = 1)
# this chunk generates the complete code appendix.
# eval=FALSE tells R not to re-run (``evaluate'') the code here.
```