README for "The Effects of Combat Deployments on Veterans' Outcomes"

by Jesse Bruhn, Kyle Greenberg, Matthew Gudgeon, Evan K. Rose, and Yotam Shem-Tov

Journal of Political Economy

Overview

This readme file documents the code required to replicate the tables and figures in the published paper and online appendices. The readme also contains information about the proprietary data used in this article.

Data Availability

The empirical results from "The Effects of Combat Deployments on Veterans' Outcomes" are derived from proprietary administrative data from the United States Army, the Department of Veterans Affairs (VA), the Social Security Administration (SSA), the joint VA and Department of Defense Mortality Data Repository (DoD MDR), Experian Credit Bureau, LexisNexis, and the National Student Clearinghouse (NSC). These restricted-use data are currently stored on servers located in the Army's Office of Economic and Manpower Analysis (OEMA) in accordance with data sharing agreements between the respective agencies and OEMA that restrict access to approved research partners.

For access to the raw administrative data required to replicate the results from this paper, approved OEMA research partners may make a direct application to OEMA (https://oema.army.mil). Other researchers may request to become OEMA research partners or they may make separate applications to the U.S. Army, the VA, the SSA, the DoD MDR, Experian, LexisNexis, and the NSC. Please contact Kyle Greenberg (kyle.greenberg@westpoint.edu) for additional information about how to apply for the data used in this project.

Details on Source Data

This folder contains the code necessary to replicate all results from this paper. These codes can be run (by those with appropriate access) on an internal OEMA data folder.

To run the code for this paper on an internal folder, the following datasets must be stored in a subfolder titled "raw_dta":

Baseline Enlisted Personnel Data Sets and Crosswalks

- raw_dta/ace_panel_v3_unit_deploy_nopii
- raw_dta/ace_panel_v3_unit_deploy_nopii_first

- raw_dta/ace_panel_v3_unit_deploy_nopii_last
- raw_dta/clean_uic_to_dmsl
- raw_dta/time_cons_dmsl
- raw_dta/pmos-to-conspmos-jan2021-panel.dta
- raw_dta/cpi_wide

Outcomes and Covariates Data Sets

- raw_dta/ace_first_v2_unit_deploy_nopii
- raw_dta/analysis_sample_phypr_cleaned
- raw_dta/analysis_sample_phypr_cleaned_v2
- raw_dta/arr_data
- raw_dta/bar_or_misc_in
- raw_dta/bud_data
- raw_dta/cas_unit_deploy_nopii_v3
- raw_dta/cas_unit_deploy_nopii_wide_v3
- raw_dta/clean_incarcerations_wide
- raw_dta/cleaned_alerts_subset1_wide_feb2022
- raw_dta/credit_coverage_2003_2017_wide_v2
- raw_dta/credit_extract_202012_v2_nopii
- raw_dta/credit_outcomes_2003_2017_v2_wide
- raw_dta/demo
- raw_dta/DPSO_return_to_OEMA_202106_nopii
- raw_dta/inc_data
- raw_dta/ln_pid_xwalk
- raw_dta/mepcom_last_unit_deploy_nopii
- raw_dta/mil_pay_outcomes_analysis_samp
- raw_dta/promo_mos_outcomes_analysis_samp
- raw_dta/rec_any_forc_2019
- raw_dta/share_dep_by_dmsl_tc_month_v2
- raw_dta/share_dep_by_dmsl_tc_month_whole_samp_v2

- raw_dta/spd_spell_panel
- raw_dta/ssa_death_ssi_unit_deploy_nopii
- raw_dta/ssdi_unit_deploy_cleaned_nopii
- raw_dta/suicide_rates (Figure 1b data is from https://www.mentalhealth.va.gov/docs/data-sheets/2021/VA_National_2001-2021_Appendix_508.xlsx)
- raw_dta/waivers_unit_deploy_wide_nopii
- raw_dta/unit_deploy_nsc_enrollments_v5_wide_nopii
- raw_dta/unit_deploy_nsc_graduations_v5_wide_nopii
- raw_dta/unit_deploy_vapgib_nopii
- raw_dta/vadc_diag_analysis_samp_ajh
- raw_dta/VADCpanel_cleaned_wide
- raw_dta/waivers_unit_deploy_wide_nopii

Descriptions of Programs/Code

The codes for this project are saved in a folder named "do" that contains the following contents:

Build Main Analysis Data Set

- master.do
- build_base_data.do
- prep_base_dta.do
- clean_dspo_ndi_death_data.do
- merge_outcomes.do
- final_prep.do
- Run the do files below if including women in sample
  - merge_outcomes_incw.do
  - final_prep_incw.do

Build 2001-Forward Data Set

- build_cohortdecomp_data.do

- prep_cohortdecomp_data.do
- merge_outcomes_cohortdecomp.do
- final_prep_cohortdecomp.do

Build Tables and Figures

- tables.do
- appendix_tables.do
- figures.do
- appendix_figures.do
  - prep_vcv.py
  - est_frandsen.m
  - testjfe.m
  - army_bounds.py

These codes can be run (by those with appropriate access) on an internal OEMA data folder.

In order to run simply, set the current directory at the top of master.do to the directory housing the "do" subfolder containing these dofiles (and the "raw_dta" subfolder containing the above data), and run the master.do do file.

This will call the data cleaning subcodes build_base_data.do, prep_base_dta.do, clean_dspo_ndi_death_data.do, merge_outcomes.do, final_prep.do, merge_outcomes_incw.do, final_prep_incw.do, build_cohortdecomp_data.do, prep_cohortdecomp_data.do, merge_outcomes_cohortdecomp.do, final_prep_cohortdecomp.do in that order.

It will next call the analysis subcodes tables.do, appendix_tables.do, figures.do, appendix_figures.do, in that order, which produce all the main and appendix table and figures.

For a color version of the main figures, one could also replace figures.do with figures_color.do

All this code runs in Stata v17 with two exceptions. After running the relevant Stata code for Figure A.2, the extensive margin bounds pasted on the top right of the figure are generated from created_dta/acr_weight_input.dta using the provided army_bounds.py code using gurobi.

Table A3 is generated by taking the excel files generated in the Stata portion of the Table A3 code (excel_varcov_cov_combat_death_by_32, excel_beta_cov_combat_death_by_32, excel_varcov_cov_noncombat_death_by_32, excel_beta_cov_noncombat_death_by_32, excel_varcov_cov_anyvadc_in_32, excel_beta_cov_anyvadc_in_32), running them through a cleaning code in python, prep_vcv.py, and then running the matlab code est_frandsen.m which calls testjfe.m

These two pieces need to be run separately after running all the Stata code via master.do if interested in replicating them.

Figure A.11 is generated from 1000 bootstrap trials and is slow to run. The slow part is left commented out and bypassed using the provided created_dta/bootstrappedoutput.dta in the interest of runtime.