

Data building

Peasant unrest and imperial repression

Stepan Polikanov

```
## Building data for the analysis
## Dependencies: data_raw folder

source(here::here("utilities", "check_packages.R"))
source(here::here("utilities", "functions.R"))

conflicts_prefer(sfnetworks::activate)
conflicts_prefer(dplyr::filter)
conflicts_prefer(dplyr::lag)
conflicts_prefer(dplyr::select)
```

```
# Repression data

## deprivation
deprivation <- read_excel(here("data", "data_raw", "grigoriadis_repression",
                              "data_result",
                              "results_deprivation_geolocated.xlsx"))

## okhrana
okhrana <- read_excel(here("data", "data_raw", "grigoriadis_repression",
                              "data_result",
                              "results_okhrana_geolocated.xlsx"),
                      guess_max = 13956)

# Unrest data

## Replication data from "Does Reform Prevent Rebellion? Evidence From
## Russia's Emancipation of the Serfs" - https://doi.org/10.7910/DVN/29018
finkel <- read_table(here::here("data", "data_raw",
                              "finkel_replication", "2014.07.12.tab"),
                    sep = "\t", header = TRUE, na.strings = 9999)

## Replication data from "Collective Action and Representation in Autocracies:
## Evidence from Russia's Great Reforms" - https://doi.org/10.1017/S0003055417000454
dower_variables <- read_dta(here("data", "data_raw", "dower_replication",
                              "variables_replication.dta"))
dower_events <- read_dta(here("data", "data_raw", "dower_replication",
                              "event_data_replication.dta"))

# occupations - province
province_level <- read_rds(here("data", "data_raw", "master_merged.rds"))

# serfdom data
serfdom_data_province <- read_dta(here("data", "data_raw",
                              "20160144_data",
                              "replication_province_Serfdom.dta"))

province_gis <- read_sf(here::here("data", "data_raw", "provinces_1897.gpkg"))

dictionary <- read_excel(here("data", "data_hand", "dictionary.xlsx"))
```

```
occupations <- read_excel(here("data", "data_raw",
                              "occupations1897_1904_cleaned.xlsx"))

occupations_entire_shares <- occupations |>
  filter(type == "0 = entire",
         province_id != "0") |>
  select(province,
         agriculture_1897 = agriculture,
         agr_arable_farming_1897 = agr_arable_farming,
         manufacturing_1897 = manufacturing,
         agriculture_1904 = `1904_agriculture`,
```

```

    agr_arable_farming_1904 = `1904_agr_arable_farming`,
    manufacturing_1904 = `1904_manufacturing`,
    population_1897 = `1897_pop`,
    population_1904 = `1904_both`,
    households_1897 = households,
    households_1904 = `1904_households`) |>
mutate(agriculture_share_1897 = agriculture_1897 / households_1897,
    agr_arable_farming_share_1897 = agr_arable_farming_1897 / households_1897,
    manufacturing_share_1897 = manufacturing_1897 / households_1897,
    agriculture_share_1904 = agriculture_1904 / households_1904,
    agr_arable_farming_share_1904 = agr_arable_farming_1904 / households_1904,
    manufacturing_share_1904 = manufacturing_1904 / households_1904)

urbanization <- occupations |>
  filter(province_id != "0") |>
  select(province, population_1897 = `1897_pop`,
    population_1904 = `1904_both`, type) |>
  pivot_wider(names_from = type,
    values_from = c(population_1897, population_1904)) |>
  transmute(province,
    urbanization_1897 = `population_1897_1 = urban` /
      `population_1897_0 = entire`,
    urbanization_1904 = `population_1904_1 = urban` /
      `population_1904_0 = entire`)

occupation_urbanization <- merge(occupations_entire_shares, urbanization,
  by = "province")

```

```

# Load raw data (adjust path as needed)
inequality <- read_excel(here("data", "data_raw", "land_inequality.xlsx"),
  skip = 3)

names(inequality) <- sapply(names(inequality), clean_land_column_name)

# Step 1: Drop peasants_incl_communes columns
inequality <- inequality |>
  select(~matches("\\.peasants_incl_communes\\."),
    ~matches("\\.all\\."))

# Step 2: Extract land and owner columns
land_cols <- names(inequality)[grepl("^land_areas\\. ", names(inequality))]
owner_cols <- names(inequality)[grepl("^n_owners\\. ", names(inequality))]

# Step 3: Reshape to long format
land_long <- inequality |>
  select(province, all_of(land_cols)) |>
  pivot_longer(~province, names_to = "var", values_to = "land_area")

owner_long <- inequality |>
  select(province, all_of(owner_cols)) |>
  pivot_longer(~province, names_to = "var", values_to = "owners")

land_parts <- lapply(land_long$var, extract_parts) |> bind_rows()
owner_parts <- lapply(owner_long$var, extract_parts) |> bind_rows()

land_long <- bind_cols(land_long, land_parts) |> select(-var, -category)
owner_long <- bind_cols(owner_long, owner_parts) |> select(-var, -category)

joined <- left_join(land_long, owner_long, by = c("province", "estate",
  "bracket"))

joined <- joined |>
  mutate(
    midpoint = sapply(bracket, bracket_midpoints),
    total_owners = owners,
    total_land = land_area,
    land_per_owner = ifelse(owners > 0, land_area / owners, 0)
  )

gini_by_province <- joined |>
  group_by(province) |>
  summarise(land_gini = compute_gini(cur_data_all()), .groups = "drop")

```

```

land_shares <- joined |>
  left_join(select(inequality, province, province_land = total_land),
    by = "province") |>
  filter(estate %in% c("nobility", "peasants_excl_communes",
    "peasants_communes_only")) |>
  group_by(province, estate) |>

```

```

summarise(total_land = sum(total_land, na.rm = TRUE), .groups = "drop",
           province_land = first(province_land)) |>
pivot_wider(names_from = estate, values_from = total_land,
            values_fill = 0) |>
mutate(peasant_share = (peasants_excl_communes + peasants_communes_only) /
       province_land,
       nobility_share = nobility / province_land) |>
select(province, peasant_share, nobility_share)

inequality2 <- read_excel(here("data", "data_raw", "land_inequality.xlsx"),
                        sheet = 2)

nadelands <- inequality2 |>
mutate(state_to_private = state_church_lands_1000_des*1000/total_land,
       nadel_to_private = nadel_lands/total_land)

```

```

finkel <- read.table(here::here("data", "data_raw", "finkel_replication",
                              "2014.07.12.tab"),
                   sep = "\t", header = TRUE, na.strings = 9999)

finkel_gub <- read.table(here::here("data", "data_raw", "finkel_replication",
                              "gub_id.tab"),
                       sep = "\t", header = TRUE)

finkel_clean <- finkel |>
group_by(Guberniyai) |>
summarize(events = n()) |>
left_join(finkel_gub, by = join_by("Guberniyai" == "gub")) |>
drop_na(name)

```

```

serfdom1858 <- serfdom_data_province |>
group_by(mapid, province_name, sh_serfs1858) |>
summarise(sh_serfs1858 = if_else(max(sh_serfs1858) == min(sh_serfs1858),
                               max(sh_serfs1858), NA),
          grain_prod = mean(grain_prod[year <= 1861], na.rm = T)) |>
drop_na(mapid) |>
mutate(mapid = as.character(mapid))

```

```

okhrana_province <- okhrana |>
filter(!is.na(gub)) |>
group_by(gub) |>
summarise(
  total_cases = n(),
  accused_cases = sum(data_source == "accused", na.rm = TRUE),
  revolutionary_cases = sum(data_source == "revolutionaries", na.rm = TRUE),
  observed_cases = sum(data_source == "observation", na.rm = TRUE)) |>
arrange(desc(total_cases))

okhrana_full <- merge(okhrana_province, dictionary,
                    by.x = "gub", by.y = "okhrana", all.x = TRUE) |>
merge(finkel_clean, by.x = "finkel", by.y = "name", all.x = TRUE) |>
merge(province_gis, by.x = "ristat_ru", by.y = "prov_RU", all.x = TRUE) |>
merge(serfdom1858, by.x = "Gub_ID", by.y = "mapid", all.x = TRUE) |>
merge(land_shares, by.x = "inequality", by.y = "province", all.x = TRUE) |>
merge(gini_by_province, by.x = "inequality", by.y = "province", all.x = TRUE) |>
merge(occupation_urbanization, by.x = "occupations", by.y = "province",
     all.x = TRUE)

```

```

write_rds(okhrana_full, here("data", "data_built", "okhrana_full.rds"))

```