Population

Jin Myung Kim

2025-07-19

Table of contents

1 Introduction

In an effort to reduce both domestic consumption and regional opium production, the Chinese government launched the Opium Replacement Program (ORP) in 2006. This initiative was designed to provide funding for Chinese companies to invest in opium-cultivating regions of Laos and Myanmar by supporting the transition of local farmers to cash crops such as rubber, maize, and sugarcane, thereby fostering broader socio-economic development in affected areas (J. Lu and Dwyer 2023; UNODC 2015; TNI 2010). Initially proposed as part of China's broader "Going Out" strategy in the early 2000s to encourage overseas investment, the ORP was officially initiated in 2004 and subsequently formalized and expanded in 2006 when it was incorporated into China's five-year plan with dedicated funding and policy support. Its stated aim is to replace opium cultivation in the Golden Triangle with licit, high-value cash crops, thus improving local livelihoods and promoting economic cooperation. To incentivize participation, Chinese companies involved in the program are offered tax exemptions within import quotas, along with subsidies and interest-free loans (TNI 2010; Cohen 2009; J. N. Lu 2019). Approximately 7.2 million USD (50 million RMB) in central funding, supplemented by 4.3 million USD (30 million RMB) from provincial matching funds, was allocated to the program. Companies must manage large-scale investment projects covering over 600 hectares (10,000 mu) to qualify, and these funds support the establishment of plantations for crops such as rubber, maize, and sugarcane.

Despite the policy's initial aim of nudging firms to invest directly to the opium farmers, firms had lots of autonomy throughout the investment activity. To be more specific, firms established locations where it favors firms' business operations rather than cooperate directly with opium farmers. By conducting several interviews with different stakeholders, J. N. Lu (2019) argues that the ORP attempts ''replacement by displacement''. That is, the replacement of opium cultivation is achieved by drawing opium farmers out of opium fields by providing labor opportunities in alternative crop plantations which are often established at regions near road and with low elevation.

1.1 Experimental Design

Main Hypothesis: The introduction of the ORP program decreased the opium cultivation.

Ideal experimental design: Randomly pick villages (that cultivate opium) and introduce ORP in some villages and leave rest of the villages untouched.

Issue:

- Randomization: The treatment are not received randomly. Villages are treated based on whether the village is suitable for business operations (i.e. low elevation, near road, etc.). Thus, we might want to characterize locations suitable for ORP alternative crop plantation, using distance to road, slope, and elevation.

Refined Hypothesis: Controlling for geographic accessibility and baseline opium risk, village exposed to ORP experienced a greater reduction in opium cultivation compared to similar, non-treated villages.

Mechanism:

- First-order Impact

  1. Local laborers shift from opium cultivation to wage labor on these plantations.
  2. Farmers/opium farmers migrate to ORP zones for better opportunities (replacement by displacement).

- Second-order Impact

  1. Schooling and child labor: with stable ORP income, some households pull children from labor to school (Sviatschi 2022; UNODC 2005)

ORP likely impacted population dynamics in Laos, especially through the relocation of labor from upland opium-growing areas to lowland agricultural and economic centers. We will explore such dynamics in the following section to better characterize such characteristics and check the possibility of the ''replacement by displacement'' mechanism.

Population and Housing Census. To understand these changes, we first analyze Laos' Population and Housing Census (2005) to establish a pre-ORP baseline. We assess how population, population density, and poverty levels relate to terrain characteristics (elevation, slope), road accessibility, and how they evolved between 2005 and 2015.

World Pop. Then we will use WorldPop data, which is trained using machine learning technique to simulate the population distribution between 2000 and 2020. This will add additional information on the population dynamics to the Population Housing Census information which is limited to only 2005 and 2015.

ORP Program. Then we will check whether the observed population dynamic is affected by ORP programs using official ORP records.

2 Population and Housing Census

2.1 Baseline Patterns in 2005 and 2015

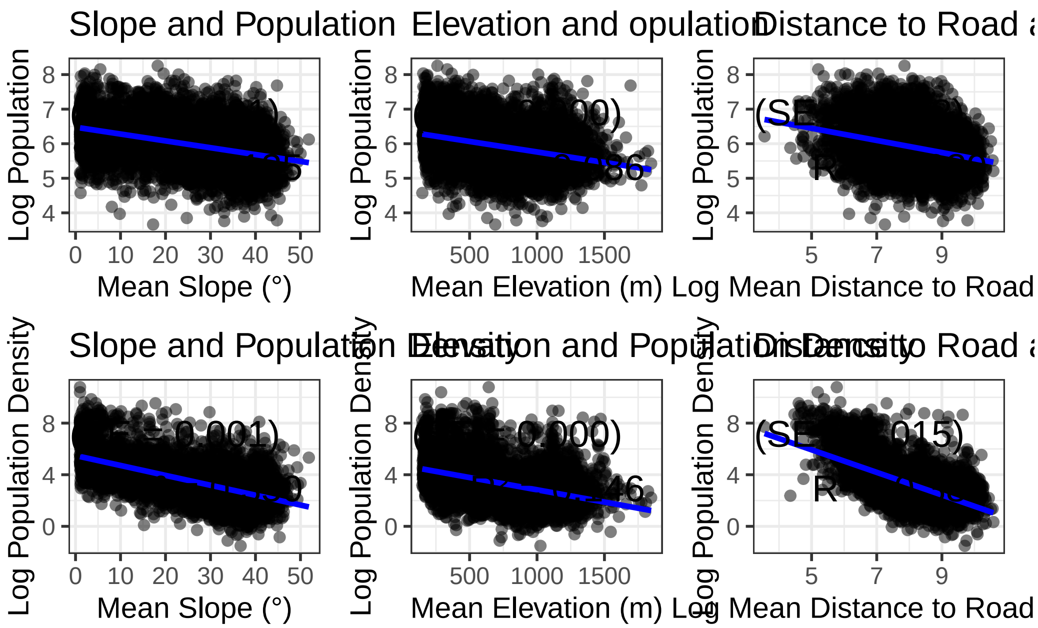
- **Population and density were higher in accessible, low-elevation/slope areas**
  - Villages located at lower elevations, with gentler slopes, and closer to national or provincial roads had significantly higher population and population density. Among these characteristics, slope showed the strongest negative association, followed by elevation and road distance.
  - *Slope, elevation, and distance to road should be treated as core baseline covariates.*

```
# Fit linear model
model_slope <- lm(log(population) ~ mean_slope, data = v_2005)
summary_slope <- summary(model_slope)

# Extract values
coef_slope <- summary_slope$coefficients["mean_slope", "Estimate"]
se_slope   <- summary_slope$coefficients["mean_slope", "Std. Error"]
r2_slope   <- summary_slope$r.squared

# Format label
label_slope <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_slope, se_slope, r2_slope)

# Plot
p1 <- ggplot(v_2005, aes(x = mean_slope, y = log(population))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_slope, hjust = 1.1, vjust = 1.5, size = 5
  labs(
    title = "Slope and Population",
    x = "Mean Slope (°)",
    y = "Log Population"
  ) +
  theme_bw()
```

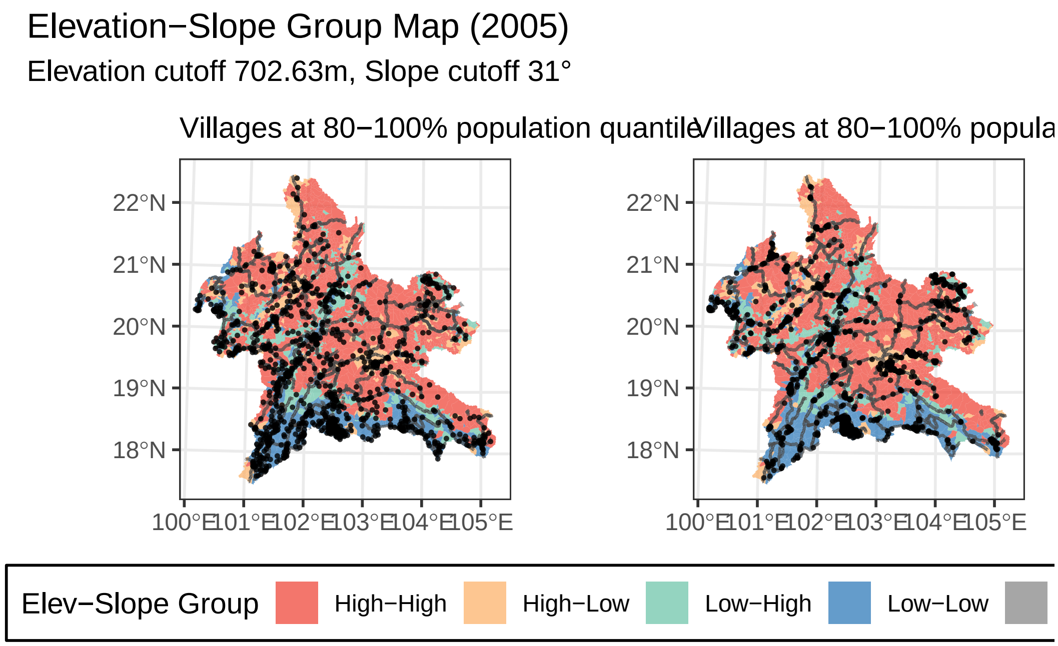
```r
# Fit linear model
model_elev <- lm(log(population) ~ mean_elev, data = v_2005)
summary_elev <- summary(model_elev)

# Extract values
coef_elev <- summary_elev$coefficients["mean_elev", "Estimate"]
se_elev   <- summary_elev$coefficients["mean_elev", "Std. Error"]
r2_elev   <- summary_elev$r.squared

# Format label
label_elev <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_elev, se_elev, r2_elev)

# Plot
p2 <- ggplot(v_2005, aes(x = mean_elev, y = log(population))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_elev, hjust = 1.1, vjust = 1.5, size = 5)
  labs(
    title = "Elevation and opulation",
    x = "Mean Elevation (m)",
    y = "Log Population"
  ) +
  theme_bw()

# Fit linear model with log-transformed mean_dist
model_log_dist <- lm(log(population) ~ log(mean_dist), data = v_2005)
summary_log_dist <- summary(model_log_dist)

# Extract values
coef_log_dist <- summary_log_dist$coefficients["log(mean_dist)", "Estimate"]
se_log_dist   <- summary_log_dist$coefficients["log(mean_dist)", "Std. Error"]
r2_log_dist   <- summary_log_dist$r.squared

# Format annotation label
label_log_dist <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_log_dist, se_log_dist, r2_l

# Plot
p3 <- ggplot(data = v_2005, aes(x = log(mean_dist), y = log(population))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_log_dist, hjust = 1.1, vjust = 1.5, size =
  labs(
```

```r
    title = "Distance to Road and Population",
    x = "Log Mean Distance to Road (m)",
    y = "Log Population"
  ) +
  theme_bw()

# Fit linear model
model_slope <- lm(log(population_density) ~ mean_slope, data = v_2005)
summary_slope <- summary(model_slope)

# Extract values
coef_slope <- summary_slope$coefficients["mean_slope", "Estimate"]
se_slope   <- summary_slope$coefficients["mean_slope", "Std. Error"]
r2_slope   <- summary_slope$r.squared

# Format label
label_slope <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_slope, se_slope, r2_slope)

# Plot
p4 <- ggplot(v_2005, aes(x = mean_slope, y = log(population_density))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_slope, hjust = 1.1, vjust = 1.5, size = 5
  labs(
    title = "Slope and Population Density",
    x = "Mean Slope (°)",
    y = "Log Population Density"
  ) +
  theme_bw()

# Fit linear model
model_elev <- lm(log(population_density) ~ mean_elev, data = v_2005)
summary_elev <- summary(model_elev)

# Extract values
coef_elev <- summary_elev$coefficients["mean_elev", "Estimate"]
se_elev   <- summary_elev$coefficients["mean_elev", "Std. Error"]
r2_elev   <- summary_elev$r.squared

# Format label
label_elev <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_elev, se_elev, r2_elev)
```

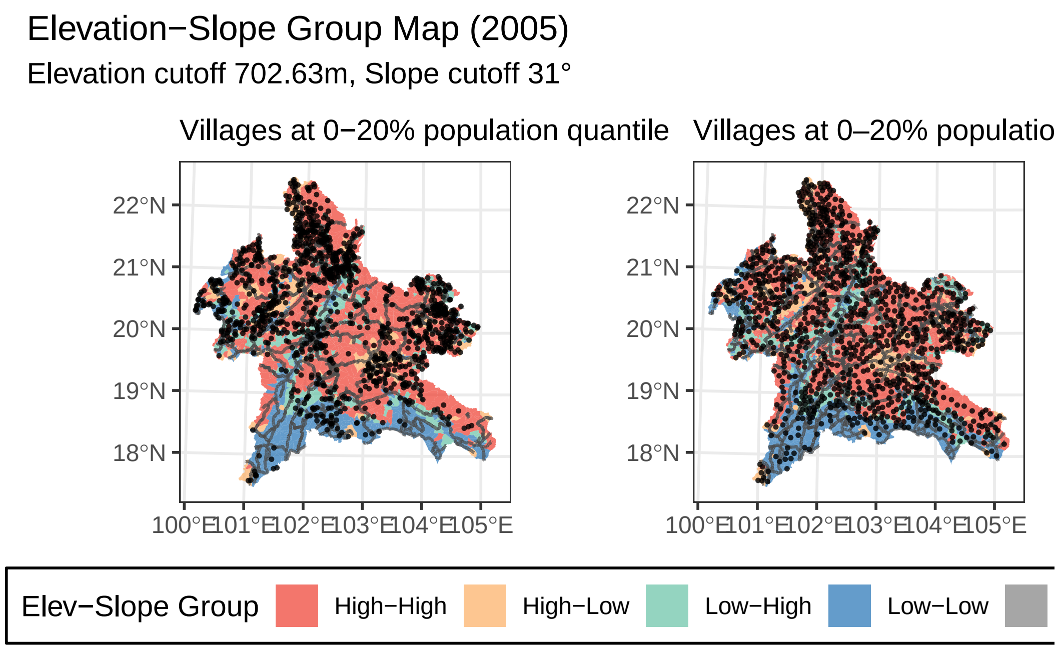
```r
# Plot
p5 <- ggplot(v_2005, aes(x = mean_elev, y = log(population_density))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_elev, hjust = 1.1, vjust = 1.5, size = 5)
  labs(
    title = "Elevation and Population Density",
    x = "Mean Elevation (m)",
    y = "Log Population Density"
  ) +
  theme_bw()

# Fit linear model with log-transformed mean_dist
model_log_dist <- lm(log(population_density) ~ log(mean_dist), data = v_2005)
summary_log_dist <- summary(model_log_dist)

# Extract values
coef_log_dist <- summary_log_dist$coefficients["log(mean_dist)", "Estimate"]
se_log_dist   <- summary_log_dist$coefficients["log(mean_dist)", "Std. Error"]
r2_log_dist   <- summary_log_dist$r.squared

# Format annotation label
label_log_dist <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_log_dist, se_log_dist, r2_l

# Plot
p6 <- ggplot(data = v_2005, aes(x = log(mean_dist), y = log(population_density))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_log_dist, hjust = 1.1, vjust = 1.5, size =
  labs(
    title = "Distance to Road and Population Density",
    x = "Log Mean Distance to Road (m)",
    y = "Log Population Density"
  ) +
  theme_bw()

(p1 + p2 + p3) / (p4 + p5 + p6)
```



Slope and Population
Elevation and opulation
Distance to Road a
Log Population
Mean Slope (°)
Mean Elevation (m)
Log Mean Distance to Road
Slope and Population Density
Log Population Density



Elevation−Slope Group Map (2005)
Elevation cutoff 702.63m, Slope cutoff 31°
Villages at 80−100% population quantile
Villages at 80−100% popula
22°N
21°N
20°N
19°N
18°N
100°E
101°E
102°E
103°E
104°E
105°E
Elev−Slope Group
High−High
High−Low
Low−High
Low−Low



In order to simultaneously consider the relationship between population and population density with elevation and slope, we categorized the villages in Northern Laos by following classification:

- Slope: ''High Slope Village'' if the average slope of the village is greater than 31 (median average slope) and "Low Slope Village" otherwise.
- Elevation: ''High Elevation Village'' if the average mean of the village is greater than 702.63 meter and ''Low Elevation Village'' otherwise.

According to Administration et al. (1992) and UNODC (2001) opium poppy cultivation are known to thrive in elevation above 700 meters and slope between 20 to 40 degree. This coincides with the above classification.

Note: Figures above did not considered distance to road since there is a strong correlation between distance to road and slope/elevation.

```
# Fit linear model
model_slope <- lm(log(mean_dist) ~ mean_slope, data = v_2005)
summary_slope <- summary(model_slope)

# Extract values
coef_slope <- summary_slope$coefficients["mean_slope", "Estimate"]
se_slope   <- summary_slope$coefficients["mean_slope", "Std. Error"]
r2_slope   <- summary_slope$r.squared
```

```r
# Format label
label_slope <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_slope, se_slope, r2_slope)

# Plot
p1 <- ggplot(v_2005, aes(x = mean_slope, y = log(mean_dist))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_slope, hjust = 1.1, vjust = 1.5, size = 5
  labs(
    title = "Relationship Between Slope and Distance to Road",
    x = "Mean Slope (°)",
    y = "Log Distance to Road (m)"
  ) +
  theme_bw()

# Fit linear model
model_elev <- lm(log(mean_dist) ~ mean_elev, data = v_2005)
summary_elev <- summary(model_elev)

# Extract values
coef_elev <- summary_elev$coefficients["mean_elev", "Estimate"]
se_elev   <- summary_elev$coefficients["mean_elev", "Std. Error"]
r2_elev   <- summary_elev$r.squared

# Format label
label_elev <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_elev, se_elev, r2_elev)

# Plot
p2 <- ggplot(v_2005, aes(x = mean_elev, y = log(mean_dist))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_elev, hjust = 1.1, vjust = 1.5, size = 5)
  labs(
    title = "Relationship Between Elevation and Distance to Road",
    x = "Mean Elevation (m)",
    y = "Log Distance to Road (m)"
  ) +
  theme_bw()

p1 + p2
```

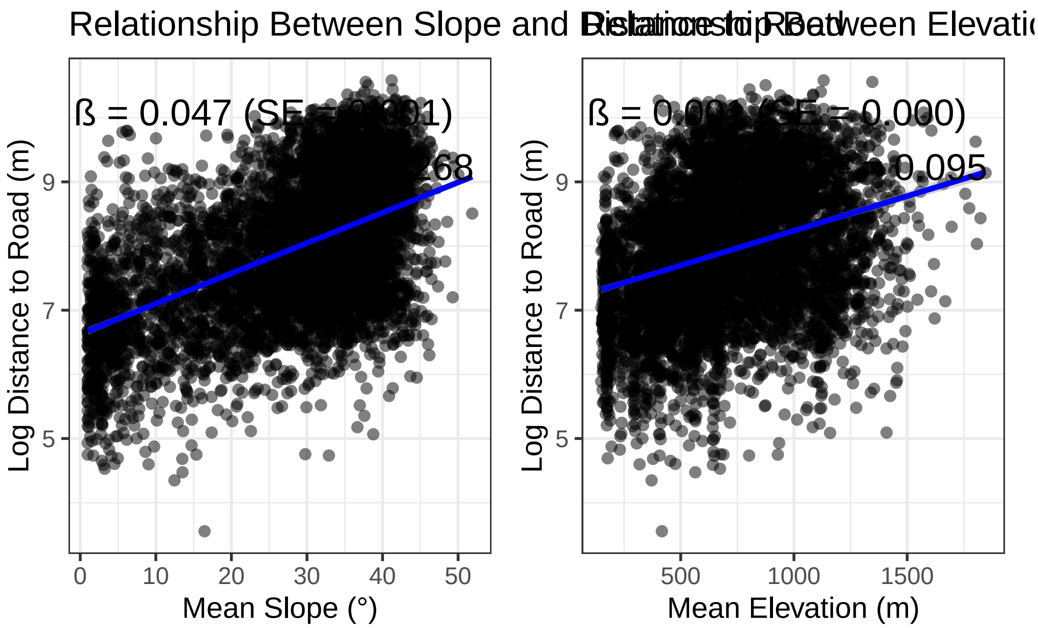


Table 1: Number of Villages by Elevation-Slope Group and Population Density Quantile

elevation_slope_group	0–20%	20–40%	40–60%	60–80%	80–100%
High-High	718	522	380	216	79
High-Low	133	182	192	194	162
Low-High	143	212	223	201	86
Low-Low	120	195	320	498	785
NA	0	2	0	2	1

Table 2: Number of Villages by Elevation-Slope Group and Population Quantile

elevation_slope_group	0-20%	20-40%	40-60%	60-80%	80-100%
High-High	511	486	420	324	174
High-Low	214	194	164	153	138
Low-High	214	211	184	167	89
Low-Low	177	217	347	467	710
NA	2	1	1	1	0

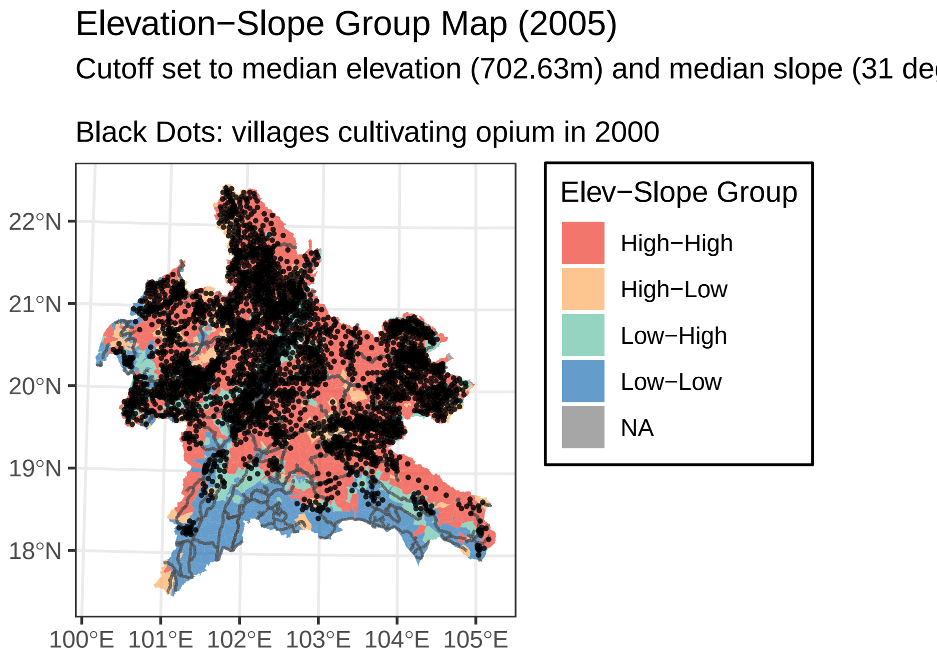


Table 3: Summary of Opium Cultivation by Elevation-Slope Group

elevation_slope_group	n_opium_v	n_no_opium_v	total_v	perc_opium	perc_no_opium
High-High	1625	290	1915	0.85	0.15
High-Low	688	175	863	0.80	0.20
Low-High	629	236	865	0.73	0.27
Low-Low	644	1274	1918	0.34	0.66
NA	0	5	5	0.00	1.00

From above Elevation-Slope Maps and Tables above, we observe that villages with higher population/population density tend to be located at low-elevation and low-slope region. In contrast, villages with lower population/population density tend to be located at high-elevation and high-slope villages. We also observe that village that have opium cultivation risk tend to be in high-elevation and high-slope region since the median elevation and median slope coincides with the topographic condition suitable for opium cultivation.

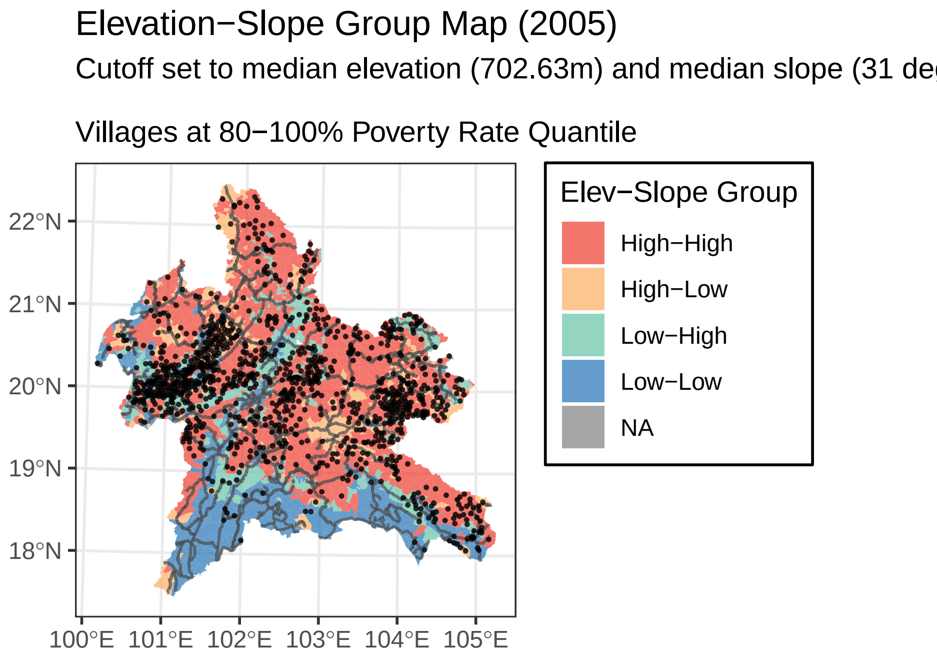


Table 4: Number of Villages by Elevation-Slope Group and Poverty Rate Quantile

elevation_slope_group	0-20%	20-40%	40-60%	60-80%	80-100%
High-High	79	207	417	565	647
High-Low	143	224	182	165	149
Low-High	92	182	189	204	198
Low-Low	801	498	324	177	118
NA	0	1	2	1	1

It can also been observed that villages with higher poverty rate tend to be in high-elevation and high-slope regions while villages with lower porverty rate tend to be in low-elevation and low-slope regions.

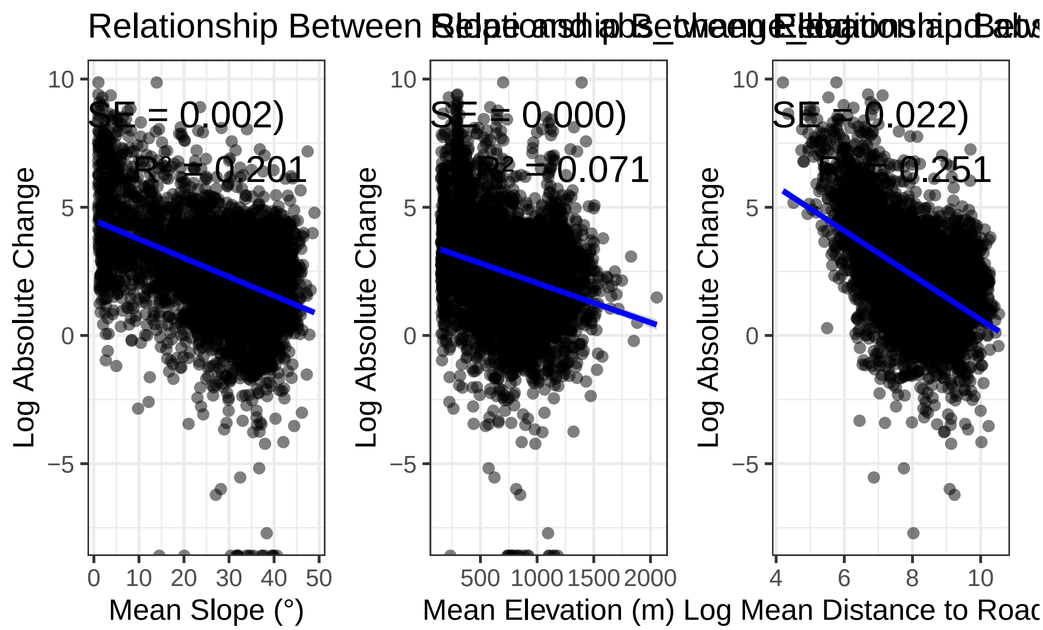
2.1.1 Summary

Topographic characteristics (elevation and slope) are associated with both demographic concentration, baseline exposure to opium poppy cultivation, and poverty.

Similar analysis using 2015 Population and Housing Census yields similar conclusion.

2.2 Patterns of Change (2005-2015)

- **Villages in low elevation, flat terrain, and near roads experienced more change (both increase and decrease)**
  - Both increases and decreases in population density were concentrated in accessible areas.
  - *Suggests that there areas were more exposed to demographic activity*

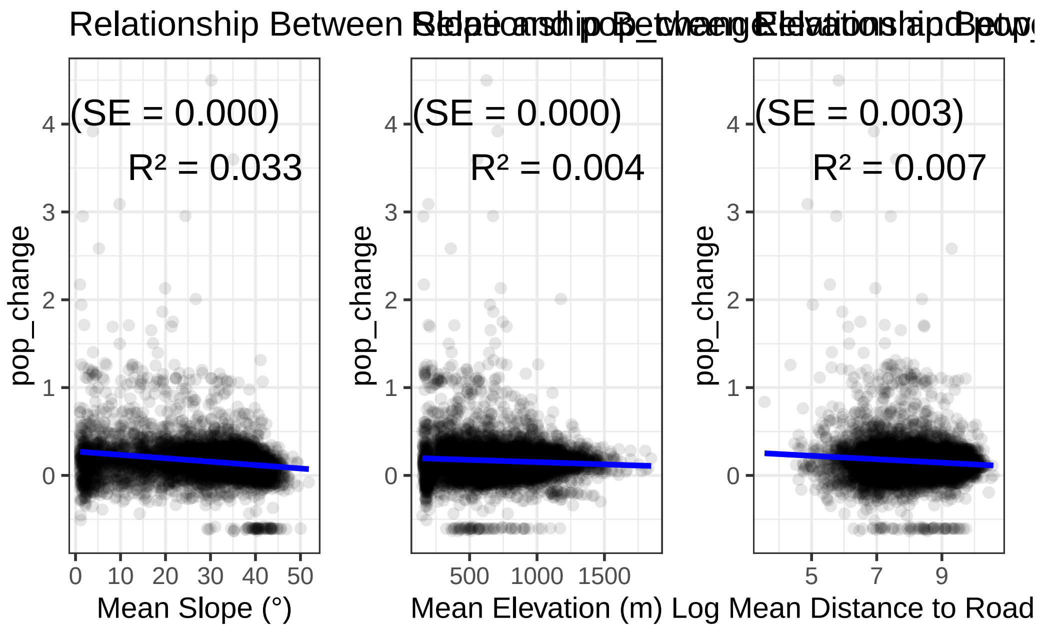



```r
temp <- matched_0515 %>%
  mutate(pop_change = (pop_2010 - pop_2001) / pop_2001)

# Fit linear model
model_slope <- lm(pop_change ~ mean_slope, data = temp)
summary_slope <- summary(model_slope)

# Extract values
coef_slope <- summary_slope$coefficients["mean_slope", "Estimate"]
se_slope   <- summary_slope$coefficients["mean_slope", "Std. Error"]
r2_slope   <- summary_slope$r.squared

# Format label
label_slope <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_slope, se_slope, r2_slope)
```

```
# Plot
p1 <- ggplot(temp, aes(x = mean_slope, y = pop_change)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_slope, hjust = 1.1, vjust = 1.5, size = 5)
  labs(
    title = "Relationship Between Slope and pop_change",
    x = "Mean Slope (°)",
    y = "pop_change"
  ) +
  theme_bw()

# Fit linear model
model_elev <- lm(pop_change ~ mean_elev, data = temp)
summary_elev <- summary(model_elev)

# Extract values
coef_elev <- summary_elev$coefficients["mean_elev", "Estimate"]
se_elev   <- summary_elev$coefficients["mean_elev", "Std. Error"]
r2_elev   <- summary_elev$r.squared

# Format label
label_elev <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_elev, se_elev, r2_elev)

# Plot
p2 <- ggplot(temp, aes(x = mean_elev, y = pop_change)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_elev, hjust = 1.1, vjust = 1.5, size = 5)
  labs(
    title = "Relationship Between Elevation and pop_change",
    x = "Mean Elevation (m)",
    y = "pop_change"
  ) +
  theme_bw()

# Fit linear model with log-transformed mean_dist
model_log_dist <- lm(pop_change ~ log(mean_dist), data = temp)
summary_log_dist <- summary(model_log_dist)

# Extract values
coef_log_dist <- summary_log_dist$coefficients["log(mean_dist)", "Estimate"]
```

```
se_log_dist   <- summary_log_dist$coefficients["log(mean_dist)", "Std. Error"]
r2_log_dist   <- summary_log_dist$r.squared

# Format annotation label
label_log_dist <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_log_dist, se_log_dist, r2_l

# Plot
p3 <- ggplot(temp,
       aes(x = log(mean_dist), y = pop_change)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_log_dist, hjust = 1.1, vjust = 1.5, size =
  labs(
    title = "Relationship Between Log Distance to Road and pop_change",
    x = "Log Mean Distance to Road (m)",
    y = "pop_change"
  ) +
  theme_bw()

p1 + p2 + p3
```

```
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
```

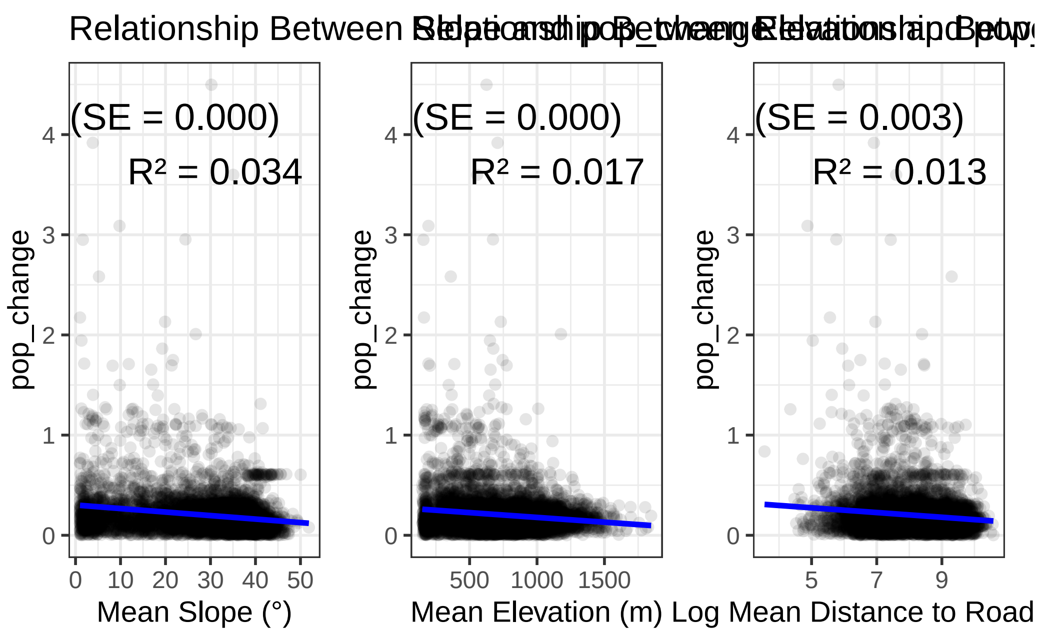



```r
temp <- matched_0515 %>%
  mutate(pop_change = abs((pop_2010 - pop_2001) / pop_2001))

# Fit linear model
model_slope <- lm(pop_change ~ mean_slope, data = temp)
summary_slope <- summary(model_slope)

# Extract values
coef_slope <- summary_slope$coefficients["mean_slope", "Estimate"]
se_slope   <- summary_slope$coefficients["mean_slope", "Std. Error"]
r2_slope   <- summary_slope$r.squared

# Format label
label_slope <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_slope, se_slope, r2_slope)

# Plot
p1 <- ggplot(temp, aes(x = mean_slope, y = pop_change)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_slope, hjust = 1.1, vjust = 1.5, size = 5
  labs(
    title = "Relationship Between Slope and pop_change",
    x = "Mean Slope (°)",
```

```
    y = "pop_change"
  ) +
  theme_bw()

# Fit linear model
model_elev <- lm(pop_change ~ mean_elev, data = temp)
summary_elev <- summary(model_elev)

# Extract values
coef_elev <- summary_elev$coefficients["mean_elev", "Estimate"]
se_elev   <- summary_elev$coefficients["mean_elev", "Std. Error"]
r2_elev   <- summary_elev$r.squared

# Format label
label_elev <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_elev, se_elev, r2_elev)

# Plot
p2 <- ggplot(temp, aes(x = mean_elev, y = pop_change)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_elev, hjust = 1.1, vjust = 1.5, size = 5)
  labs(
    title = "Relationship Between Elevation and pop_change",
    x = "Mean Elevation (m)",
    y = "pop_change"
  ) +
  theme_bw()

# Fit linear model with log-transformed mean_dist
model_log_dist <- lm(pop_change ~ log(mean_dist), data = temp)
summary_log_dist <- summary(model_log_dist)

# Extract values
coef_log_dist <- summary_log_dist$coefficients["log(mean_dist)", "Estimate"]
se_log_dist   <- summary_log_dist$coefficients["log(mean_dist)", "Std. Error"]
r2_log_dist   <- summary_log_dist$r.squared

# Format annotation label
label_log_dist <- sprintf("  = %.3f (SE = %.3f)\nR² = %.3f", coef_log_dist, se_log_dist, r2_l

# Plot
p3 <- ggplot(temp,
```

```
      aes(x = log(mean_dist), y = pop_change)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  annotate("text", x = Inf, y = Inf, label = label_log_dist, hjust = 1.1, vjust = 1.5, size =
  labs(
    title = "Relationship Between Log Distance to Road and pop_change",
    x = "Log Mean Distance to Road (m)",
    y = "pop_change"
  ) +
  theme_bw()

p1 + p2 + p3
```

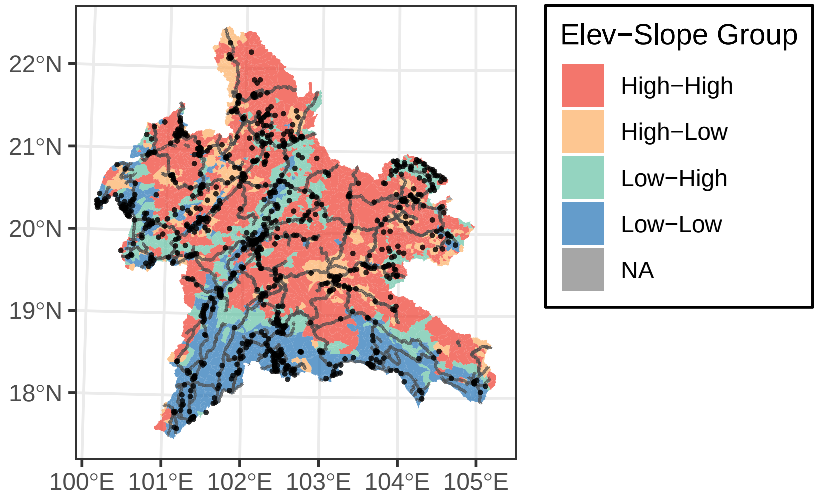
```
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
`geom_smooth()` using formula = 'y ~ x'
```



Elevation−Slope Group Map (2005)

Elevation cutoff 702.63m, Slope cutoff 31°

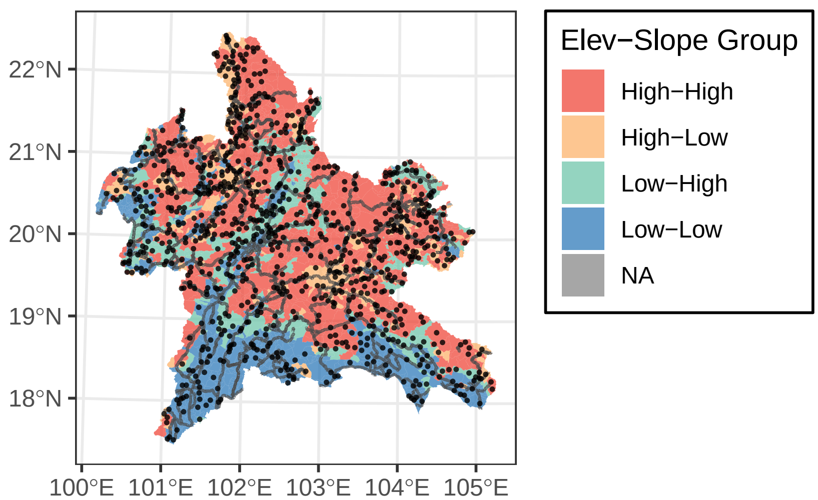
Black Dots: villages at 80−100% population density change quantil



Elevation−Slope Group Map (2005)

Elevation cutoff 702.63m, Slope cutoff 31°

Black Dots: villages at 40–60% population density change quantile



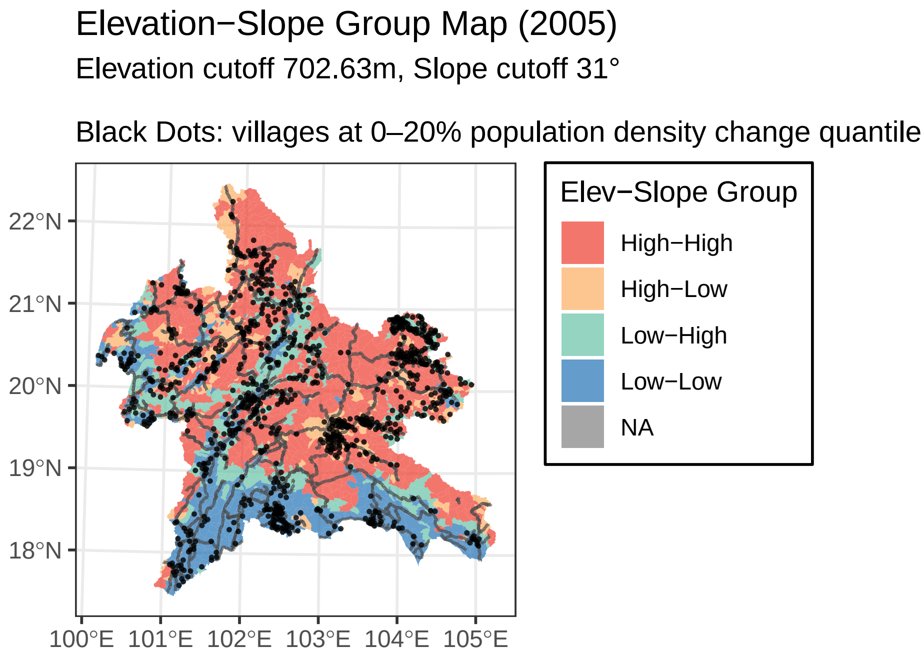
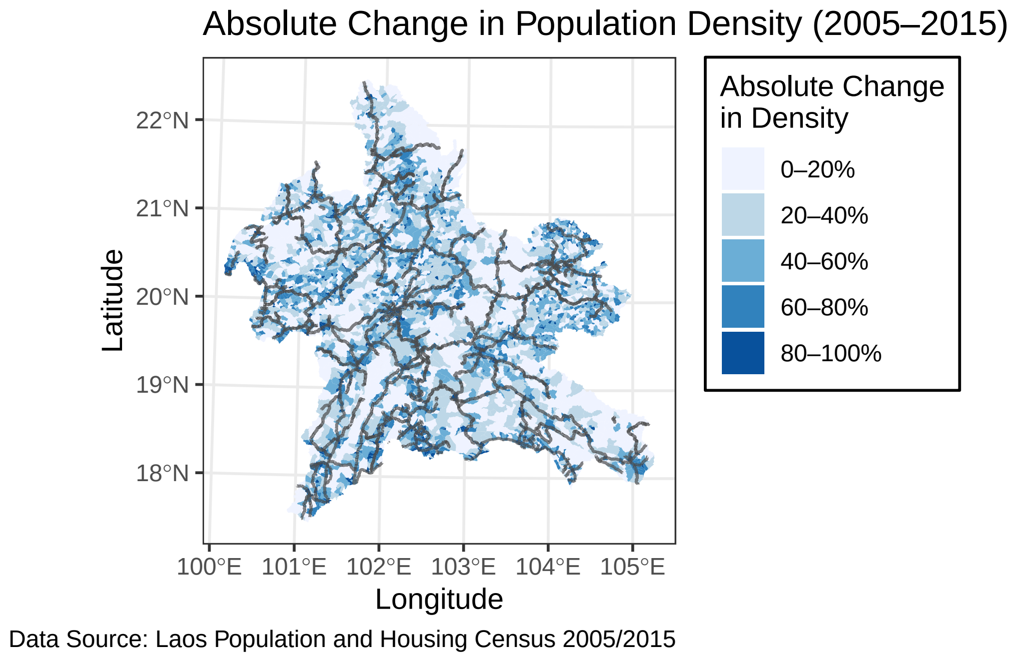


Table 5: Number of Villages by Elevation-Slope Group and Population Density Change Quantile

elevation_slope_group	0–20%	20–40%	40–60%	60–80%	80–100%
High-High	228	450	438	369	198
High-Low	162	135	161	155	122
Low-High	201	189	155	109	82
Low-Low	377	194	214	334	565
NA	0	0	0	1	0



- **Population and density changes showed spatial clustering**
  - It can be observed that villages that are in the highest quantile are also surrounded by villages in lower quantile. In other words, villages that experienced higher population density increase are close to villages that experienced lower population density increase.

Table 6: Summary Statistics of Population Density Change by Quantile (Census)

change_quantile	mean_val	median_val	minimum	maximum
0–20%	-52.369684	-14.347769	-4376.7847000	-6.0999966
20–40%	-2.316047	-2.082729	-6.0988216	0.4348898
40–60%	3.700253	3.443567	0.4367409	7.9297333
60–80%	17.080334	15.665905	7.9355316	32.0619580
80–100%	493.816808	84.209671	32.1726000	19378.7730000

From the above table, we can further observe that most of villages in lower 40% quantile experience decrease in population density.

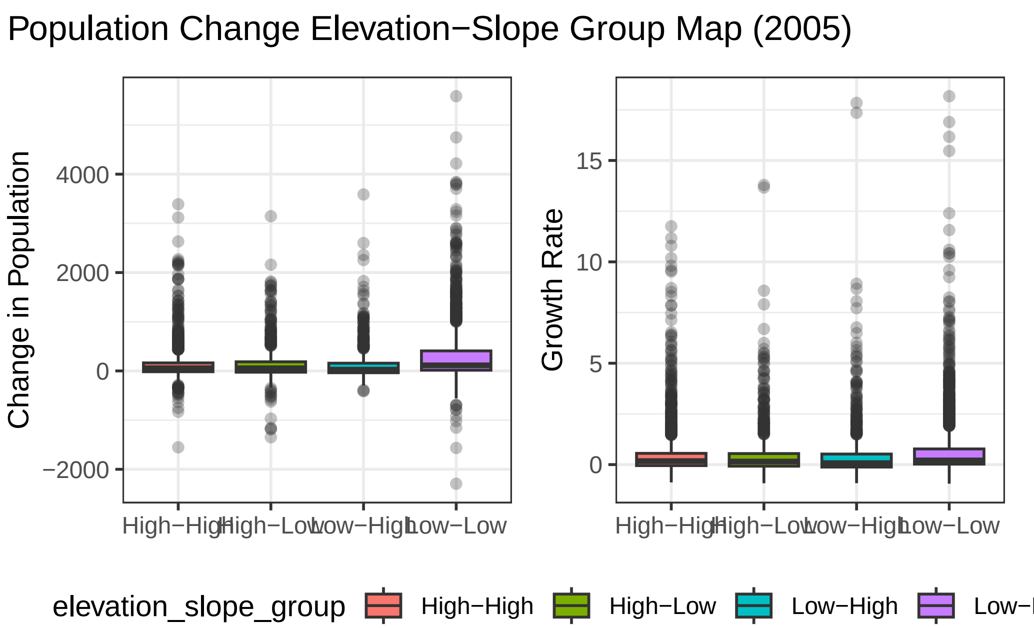
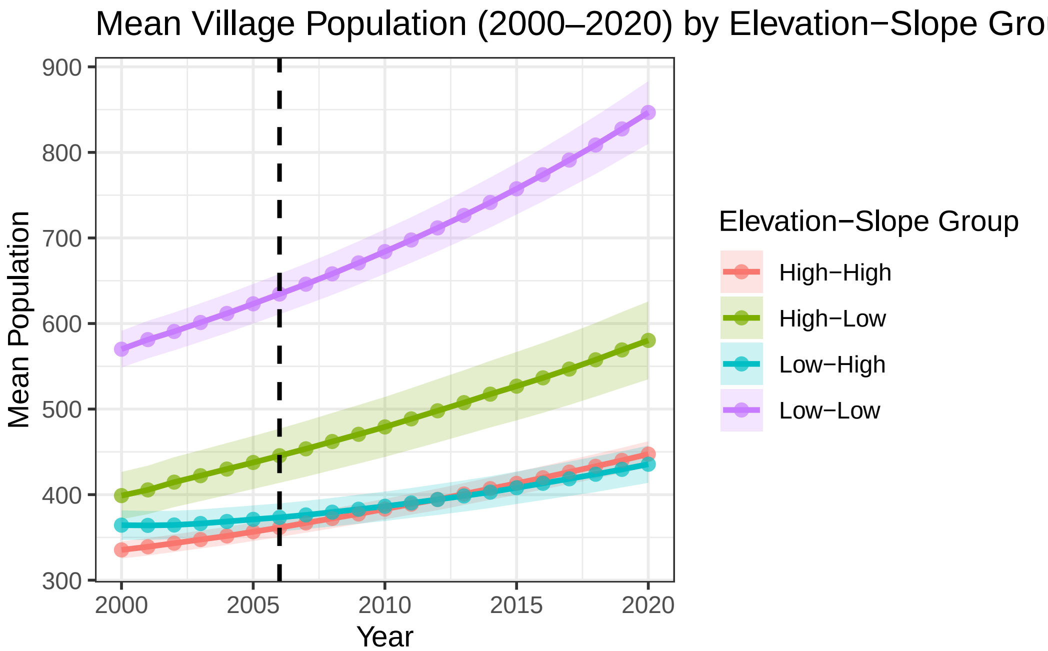


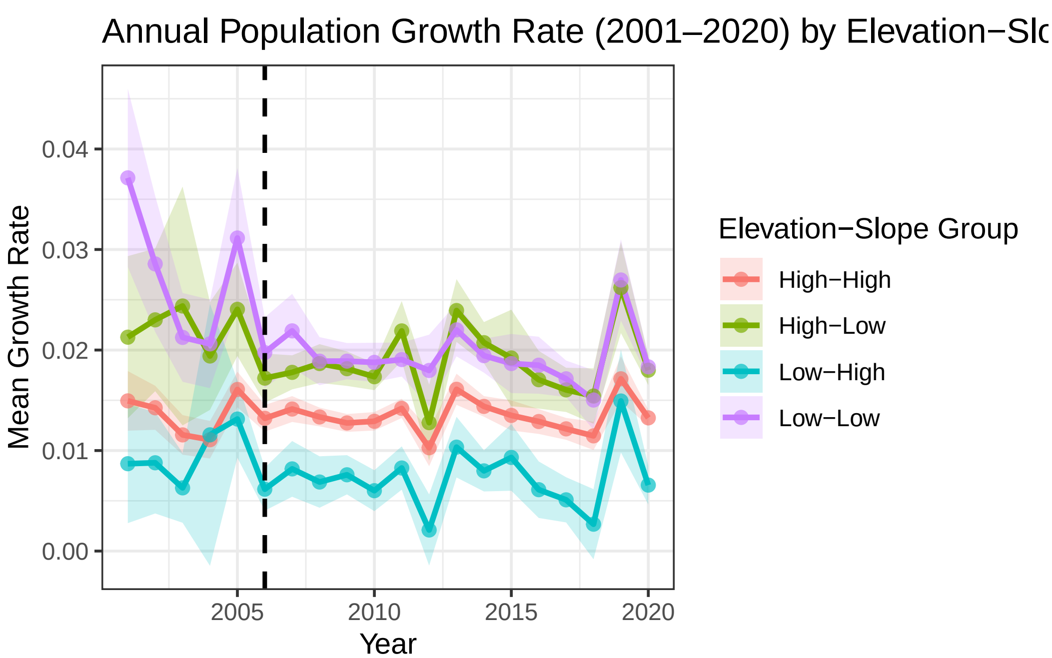
Table 7: Summary Statistics of Population Change by Elevation Slope
Matched)

elevation_slope_group	mean_c_population	median_c_population	sd_c_population	mean_g_populat
High-High	118.7104	56	309.5695	0.4909082
High-Low	134.3322	54	349.5559	0.5063868
Low-High	120.4988	26	328.6131	0.4934685
Low-Low	301.4241	114	560.9786	0.7401963

Using matched Population and Housing Census dataset, we can observe villages in low-low elevation-slope group experience higher population change and higher population growth rate compared to other elevation-slope group.

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```





Using WorldPop Population Count data, we can also observe that low-low elevation-slope group have both higher population and population growth rate compared to high-high group.