STAT 230A Final Project
Replication of Michalopoulos: The Origins of Ethnolinguistic Diversity

Andrej Leban, andrej_leban@berkeley.edu
Isaac Schmidt, ischmidt20@berkeley.edu

Contents

1 Paper Summary & Summary Statistics Table

1.1 Paper Summary

The paper by Michalopoulos (Michalopoulos 2012) aims to explain ethnolinguistic diversity within and across countries by assuming that a proxy quantity—the number of languages per square kilometer—is determined by a selection of various economic, historical, and geographic variables. It determines that *variation in regional land quality* and *variation in elevation* are the most significant determinants of linguistic diversity. The hypothesis underpinning this examination is that differences in local land characteristics induce different levels of human capital across locations, which in turn, gives rise to

localized ethnicities that are characterized by separate languages. The results of the empirical study presented are found to be consistent with this hypothesis.

The empirical results are obtained separately by three regressions:

- **Cross-country**: this takes the current political borders as the unit within which covariates such as the number of languages are counted.
- **Virtual countries**: To account for the arbitrary nature of some political boundaries with respect to ethnolinguistic groupings, the world is split into arbitrary *virtual countries* and the regression is performed again.
- **Adjacent regions**: To account for a potentially high "baseline" effect in some regions, adjacent regions are compared directly, which neutralizes region-specific fixed effects and focuses on the effect of the variables under consideration. We will not replicate this analysis.

1.2 Exploratory Data Analysis and Summary Table

The data comes from multiple sources: the standard geographic data was sourced from the *Geographically Based Economic Data database*, the data on land quality for agriculture comes from *Ramankutty et al. (2002)*, and the data on the distribution of languages comes from the *World Language Mapping System*. Fortunately, the data provided by the author was already processed and cleaned to the extent used in the paper, so all we did was rename columns to more descriptive names.

The paper lacks a true summary table and shows a couple of EDA figures instead. We replicate two of those figures, and then display our own summary table of the features used in the paper's first regression—the *cross-country model*. Figure 1 shows the distribution of land suitability for agriculture across the world at a resolution of .5-by.5 decimal degrees. The dependent variable represents the probability that a particular grid cell may be cultivated.

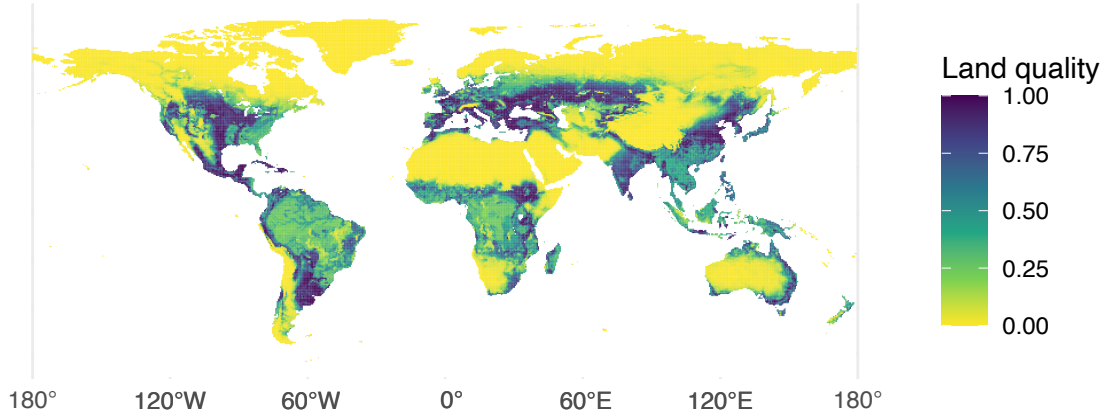


Figure 1: Land quality for agriculture across countries

Figure 2 shows the distribution of land quality within two countries selected in the paper—Greece and Nepal, obtained with a kernel density estimate using the Epanechnikov kernel.

Table 1 shows summary statistics of important variables for the first model. The dependent variable is `numLang`, which is the number of languages whose "traditional homeland" intersects with the country's boundary. Additional covariates are measures of centrality and variability of the geographic data, the log of the country's 1995 population, human migration distance from Africa, and the distance from a

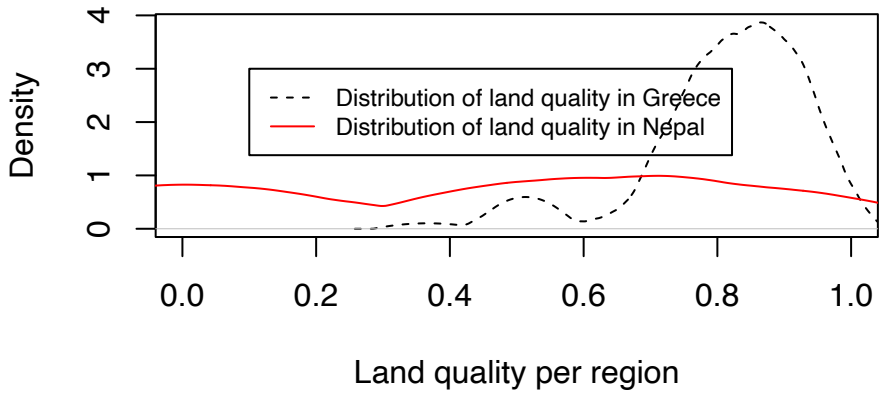


Figure 2: Kernel density of land quality in Greece and Nepal

Table 1: Summary statistics for covariates in cross-country analysis

	numLang	sdElev	sdSuitable	avgElev	avgSuitable	absLat	avgPrecip	avgTemp	lnArea	seaDist	migrationDist	lnPopDens1995
min	1.00	0.01	0.00	0.03	0.00	0.64	4.00	-6.37	-3.24	0.01	0.10	-10.22
median	10.00	0.25	0.18	0.42	0.44	24.18	77.11	20.93	0.61	0.18	5.79	-3.07
max	462.00	1.95	0.41	2.52	0.96	67.79	278.16	28.74	4.73	1.98	26.67	-0.25
mean	35.69	0.36	0.18	0.57	0.44	27.14	91.23	17.86	0.52	0.34	8.69	-3.27
sd	73.41	0.36	0.10	0.49	0.25	17.68	63.84	8.49	1.55	0.38	6.89	1.46

large body of water. While some other variables in the provided dataset have missing values for some countries, we note that all variables included in the first regression are known for all countries.

2 Analysis 1: Cross-Country

The first model regresses the (log) number of languages within each country on the features described above. Michalopolous presents five different regression models, each containing a different number of covariates. The model, as described in the original paper, is the following:

$$\ln(\text{numLang}_i) = \beta_0 + \beta_1 * \text{absLat}_i + \beta_2 * \text{sdElev}_i + \beta_3 * \text{sdSuitable}_i + \beta_4 * X_i + \epsilon_i \quad (1)$$

The first model only includes absolute latitude, the second model adds the mean and standard deviation of both elevation and land quality within each country, and the remaining models add additional covariates represented by X_i.

2.1 Statement of Assumptions

The canonical assumptions of a linear model are that Equation 1 actually is the data-generation process, and that the error terms ϵ_i are normal with mean 0, and constant variance σ^2. Of course, these assumptions are rarely actually true, but fortunately, they can be relaxed slightly.

In the original paper, Michalopolous reported "robust" standard errors for the estimated coefficients, following Eicker-Huber-White's formula. The author used the default behavior of Stata's `robust` command,

Table 2: Main specification for the cross-country analysis. Italics indicate significance at the 1% level.

Variable	(1)	(2)	(3)	(4)	(5)
Variation in elevation		*0.310*	*0.256*	*0.291*	*0.275*
		(0.113)	*(0.079)*	*(0.089)*	*(0.101)*
Variation in land quality		*0.340*	*0.177*	*0.208*	*0.211*
		(0.084)	*(0.061)*	*(0.058)*	*(0.060)*
Mean elevation		-0.249	-0.111	-0.104	-0.085
		(0.113)	(0.106)	(0.118)	(0.113)
Mean land quality		-0.179	-0.069	-0.029	0.006
		(0.069)	(0.065)	(0.068)	(0.064)
Absolute latitude	*-0.479*	*-0.547*	-0.058	-0.033	-0.131
	(0.070)	*(0.061)*	(0.192)	(0.214)	(0.201)
Mean precipitation			*0.468*	*0.447*	*0.479*
			(0.086)	*(0.088)*	*(0.088)*
Mean temperature			0.270	0.385	0.404
			(0.197)	(0.213)	(0.183)
Ln(Area)			*0.517*	*0.482*	*0.464*
			(0.067)	*(0.073)*	*(0.074)*
Distance from the sea			0.053	0.063	0.073
			(0.065)	(0.062)	(0.064)
Migratory distance from East Africa			*-0.281*	-0.518	-0.513
			(0.063)	(0.199)	(0.218)
Ln(Population density in 1995)				-0.118	0.023
				(0.087)	(0.072)
Ln(Population density in 1500)					-0.235
					(0.105)
Year of independence					-0.108
					(0.066)
Timing of transition to agriculture					0.134
					(0.094)

which includes the HC1 correction, as described in Section 6.4.1 of Peng Ding's lecture notes (Ding 2022). Such standard errors relax the homoskedasticity requirement—$\text{Var}(\epsilon_i) = \sigma^2$—as well as the assumption of normality.

Thus, the only maintained assumptions are that the linear form in 1 holds, and that the error terms are independent with mean 0.

2.2 Replication

As the code and the data files were provided completely by the author, we were able to replicate the results perfectly. Table 2 perfectly replicates Table 1 in the original paper, and Table 3 displays additional information about each model. Note that all variables, including indicators, were standardized by Michalopolous, so we did so here as well. As mentioned above, the reported standard errors follow the EHW formula with HC1 correction, so they are generally slightly larger than those one would obtain from a homoskedastic model. Unsurprisingly, given the increasing number of features, the observed R^2 also increases with each model.

Table 3: Information for each model in cross-country analysis.

Model	Continental Indicators	Observations	R^2
1	No	156	0.23
2	No	156	0.40
3	No	156	0.67
4	No	156	0.69
5	Yes	142	0.73

The interpretation of these results is much the same as in the original paper. In all four models, variation in elevation and variation in land quality were useful predictors of the log number of languages, as originally hypothesized by the author.

The effects of the geographic variables are also noteworthy. Naturally, the effect of absolute latitude becomes insignificant once precipitation and temperature are introduced, as those two are highly correlated with distance from the equator. Between models 1.1 and 1.2, and also 1.2 and 1.3, the observed R^2 makes sizable jumps, indicating that these geographic features are very useful in explaining linguistic diversity. About the distance distance from the sea coefficient, the author has this intriguing interpretation:

> ... areas that are increasingly isolated from the sea have been experiencing limited population mixing and thus should, on average, display higher ethnolinguistic fractionalization. It should be noted, however, that mean distance from the coast also captures the vulnerability of different areas to both the incidence and the intensity of invasion and colonization. Thus, the coefficient should be interpreted cautiously.

As the coefficient was never much more than one standard error above zero anyway, it is easy to ignore this effect entirely.

The final model introduces variables related to a country's history. The log of population density in 1500 does have a significant effect (at larger thresholds), and the author suggests "conditional on geographic characteristics, contemporary ethnic diversity may have been influenced by a country's historical levels of development." A mechanism that would track with this explanation is that the time of transition to a "modern" nation state, which necessarily reduces the ethnic diversity of a country, is naturally tied with its historic development level.

However, it would also seem plausible that countries that were denser in 1500 would have greater ethnolinguistic diversity today, simply due to having more people to split. It could also be that this new feature is simply "stealing" the effect of the 1990 density feature, as the two are very strongly correlated. Therefore, we consider it questionable that such "historical levels of development" have much effect on modern diversity.

Another thing to note is that we noticed some inaccuracies with the provided years of independence. For example, the United States was listed as 1816, as opposed to 1776 or 1783. Other long-existing countries, such as Portugal and Denmark, were also given this 1816 value. Additionally, former Soviet republics were all (correctly) given a year of 1991, despite many of those having been independent countries with established notions of ethnic identity long before being absorbed into the Soviet Union. This further emphasizes that independence year in general is almost arbitrary — it would have been surprising if it had a significant relationship with the outcome.

Finally, the author includes five indicator variables in the last regression that map to the continent the country is located in. This is done in an attempt to better model both geographic (the author mentions

Africa as being less geographically varying as a whole), as well as continent-wide historic effects.

2.3 Critique of Assumptions

As stated before, this model only requires two main assumptions:

1. The functional form of 1 is correct.
2. The error terms are independent.

Both assumptions are hard to take fully for granted. Starting with the second, it is likely that there is some spatial correlation between neighboring countries, leading to dependence among the error terms. However, such correlation could have already been sufficiently modeled by including geographic features such as migration distance and average temperature. Additionally, the virtual country analysis, which we will reanalyze in Section 4, shows that the results of the model still hold after abstracting away the established country boundaries.

To informally test the whether or not the linearity assumption holds, Figure 3 shows the residual plot for the fifth model. While there is no curved relationship in the plot, it is clear that the residuals tend to increase as the dependent variable increases. This means that there is likely some other feature or combination of features which significantly impacts the log number of languages, which this model does not include. We only show the residuals for the fifth model, but given it is the best specified model, it is expected that the residual plots for the other models would look even worse.

Somewhat coincidentally, the residuals do look roughly homoskedastic, even though that assumption was relaxed. Another diagnostic plot that is commonly used is a Normal Q-Q plot of the residuals, but as this model does not require normality, we will not include such a plot here.

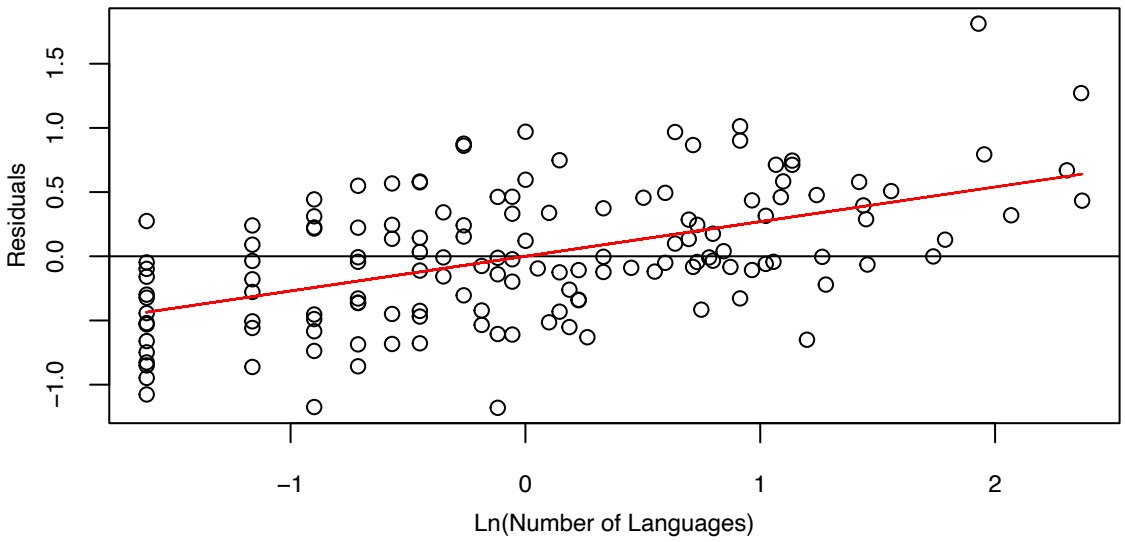


Figure 3: Residual Plot for Model 1.5

3 Robustness Check of Analysis 1

3.1 Table 2A

Table 4 recreates the first series of robustness check for the model described above. Since the dependent variable is a count, it makes sense to employ a regression model that models that fact directly. The author

Table 4: Table 2A—Robustness Checks for the Cross-Country Analysis. Italics indicate significance at the 1% level.

Variable	Negative binomial	OLS1	OLS2	OLS3
Variation in elevation	*1.313*		1.045	*1.309*
	(0.378)		(0.473)	*(0.445)*
Variation in land quality	*3.252*			
	(0.833)			
Dispersion of elevation		*0.429*		
		(0.130)		
Dispersion of land quality		*1.315*		
		(0.399)		
Variation in climatic suitability			*2.505*	
			(0.749)	
Mean climatic suitability			0.661	
			(0.349)	
Variation in soil suitability				*3.785*
				(1.324)
Mean soil suitability				0.653
				(0.474)

Table 5: Information for each model in Table 2A: cross-country robustness check

Model	Continental Indicators	Observations	R^2	Log pseudolikelihood
1	Yes	142	-	-536.36
2	Yes	142	0.73	-
3	Yes	142	0.73	-
4	Yes	142	0.73	-

argues for the *negative binomial* generative model due to the supposed *overdispersion* in the number of languages. The results are presented in the first column.

The second robustness check substitutes the measure of variance — standard deviation — of the two most crucial covariates (elevation and land quality) with a bit more "robust measure" — "dispersion," which the author defines as the range between the minimum and maximum values.

The third and fourth columns attempt to substitute the variation in land quality with alternative metrics: the means and variations in climatic and soil suitability, respectively. These are isolated components of the composite land suitability metric used in the original regression.

All the regressions in this table include the continental fixed effects mentioned previously. The regressions with alternative covariates confirm that the variation in land quality (or its counterpart) is the most impactful factor in determining ethnolinguistic diversity within a country. Furthermore, our replication finds the published results to hold exactly.

Table 5 summarizes the control groups used and the results of the robustness check regressions. Interestingly, the findings are robust to the substitution of the most important covariates with alternative measures, since the R^2 coefficient matches that of the last regression in the original analysis (with the same number of covariates), and is completely stable across all comparable models using alternative covariates.

3.2 Table 2B

Table 6 introduces a new dependent variable in place of the (logarithm) of the languages spoken — *ELF - Ethnolinguistic fractionalization*. This is the probability that two randomly drawn individuals would belong to different ethnolinguistic groups and is taken from an updated version of an old Soviet work - *Atlas Narodov Mira* (Atlas of the World's Nations) in the first three columns. Additionally, climatic data is again used in lieu of the land suitability metric used in the initial model.

Of note here is that, absent a continental fixed-effect variable, the variation in elevation coefficient actually flips its sign while becoming insignificant, while the situation is again reversed once the fixed effect is introduced. This seems to track with the author's explanation that Africa, for example, is less varying in elevation in general, so the numerical effect of the latter needs to be adjusted per-continent.

In columns 4-7, the same metric is reconstructed from the *Ethnologue* dataset using increasing fineness in defining ethnolinguistic groups via the aggregation of language trees. Despite the introduction of additional geographic features, the variations in elevation and land quality remain highly significant. Additionally, fractionalization is found to be positively impacted by the average amount of precipitation a country receives and its distance from the coast, while latitude and migratory distance from East Africa impact it negatively.

The replication again obtains complete agreement in the values, confidence intervals, and the R^2 coefficient with the published results. The noticeably small values of R^2 across the models perhaps indicate faults with the dependent variable; for the latter columns, this is a largely arbitrary level of aggregation in the clustering trees, so it's unsurprising the agreement represented by R^2 can even decrease in comparison to the third column using the "original" ELF.

4 Analysis 2: Virtual Country (Re-Analysis)

The second analysis Michalopolous presents in his paper is essentially a repeat of the the first, but instead aggregating geographic and ethnic information over "virtual" countries as opposed to real ones. The stated motivation for this is "to investigate whether the relationship between geography and ethnic diversity holds true at an arbitrary level of aggregation."

As with the previous analysis, the geographic features are derived from a dataset of cells, each of size .5-by.5 decimal degrees. However, instead of aggregating these cells at the country level as before, we now split up the world into blocks of size 2.5-by-2.5 decimal degrees, with each block containing 25 cells. Each block is precisely a "*virtual country.*"

To obtain the number of languages in each virtual country, Michalopolous simply intersected the shapefile provided in the World Language Mapping System with the newly-formed grid. However, probably due to the proprietary nature of the WLMS, the "number of languages" variable was withheld from the public data download, meaning we could not exactly replicate the analysis.

Fortunately, we stumbled across the *Geo-referencing of Ethnic Groups* (GREG) dataset (Weidmann, Rød, and Cederman 2010), which contains a shapefile of the locations of 928 ethnic groups across the world. As Michalopolous was only using linguistic diversity as a proxy for ethnic diversity, we decided it would be useful to model ethnic diversity directly, to see if the original paper's results held up with the GREG data.

Table 6: Table 2B—Linguistic Fractionalization across Countries. Italics indicate significance at the 1% level.

Variable	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Variation in elevation		-0.111	0.363	0.356	*0.472*	*0.426*	*0.413*
		(0.124)	(0.142)	(0.169)	*(0.160)*	*(0.149)*	*(0.156)*
Variation in climatic suitability		*0.294*	*0.231*	*0.291*	*0.293*	0.215	0.156
		(0.094)	*(0.084)*	*(0.103)*	*(0.103)*	(0.086)	(0.089)
Mean elevation		0.093	-0.301	-0.352	*-0.475*	*-0.462*	-0.367
		(0.127)	(0.148)	(0.171)	*(0.169)*	*(0.166)*	(0.167)
Mean climatic suitability		0.053	0.218	-0.141	-0.062	-0.213	-0.024
		(0.083)	(0.108)	(0.128)	(0.121)	(0.108)	(0.110)
Absolute latitude	*-0.369*	*-0.434*	-0.397	-0.116	-0.185	-0.064	-0.124
	(0.080)	*(0.081)*	(0.331)	(0.358)	(0.326)	(0.295)	(0.311)
Mean precipitation			0.180	*0.455*	0.404	*0.487*	*0.375*
			(0.151)	*(0.174)*	(0.167)	*(0.142)*	*(0.140)*
Mean temperature			-0.030	0.248	0.181	0.316	0.302
			(0.266)	(0.296)	(0.281)	(0.274)	(0.280)
Ln(area)			0.030	-0.247	-0.186	-0.174	-0.015
			(0.125)	(0.146)	(0.153)	(0.132)	(0.130)
Distance from the sea			*0.281*	*0.452*	*0.414*	*0.326*	0.229
			(0.086)	*(0.118)*	*(0.109)*	*(0.098)*	(0.093)
Migratory distance from East Africa			-0.122	-0.535	-0.280	-0.359	-0.540
			(0.256)	(0.289)	(0.296)	(0.287)	(0.287)
Ln(Population density in 1995)			-0.022	-0.093	0.006	0.007	-0.090
			(0.103)	(0.141)	(0.141)	(0.137)	(0.132)
Ln(Population density in 1500)			-0.268	-0.239	-0.214	-0.211	-0.166
			(0.123)	(0.144)	(0.150)	(0.135)	(0.137)
Year of independence			0.146	0.058	0.111	0.061	0.046
			(0.096)	(0.122)	(0.118)	(0.108)	(0.104)
Timing of transition to agriculture			-0.080	0.154	0.196	0.338	0.155
			(0.131)	(0.168)	(0.155)	(0.135)	(0.135)

Table 7: Information for each model in Table 2B: Linguistic Fractionalization across Countries

Model	Continental Indicators	Observations	R^2
1	No	143	0.14
2	No	143	0.20
3	Yes	143	0.53
4	Yes	143	0.32
5	Yes	143	0.35
6	Yes	143	0.44
7	Yes	143	0.48

4.1 Data Cleaning

The original GREG dataset required some manipulation to get it in a format suitable to swap in for the WLMS. Each polygon was labeled with up to three ethnic groups, so we had to melt and then dissolve the polygons such that each polygon represented only one ethnic group, and each ethnic group was only assigned to one polygon. For details, see the appendix which shows the geoprocessing steps performed with the `geopandas` module in Python.

In the original paper, Michalopolous described the steps he took to filter the virtual countries on criteria mostly based on the amount of "coverage" each country had in the WLMS data. If a large portion of a virtual country was an area that contained no languages — for example, the Sahara Desert — that virtual country was excluded from the analysis. The public data download, which contained the virtual countries after this filter had been applied, contained 1,888 virtual countries. Due to differences in coverage between WLMS and GREG, applying the same criteria to GREG would have resulted in 2,476 countries. Including these additional countries would have required obtaining the other features for these areas, and as Michalopolous did not document this procedure well, we decided that this was not feasible. The end result was that our dataset only included the intersection of the sets derived WLMS and GREG, which excluded the ~600 countries from the dataset derived from GREG, but also about 30 countries that had enough coverage in WLMS, but did not in GREG.

Finally, the actual regressions performed in the paper used a dataset that was further filtered down. That is, there must have been at least 3000 people living in the virtual country in 1995, and at least 10 of the 25 cells that comprise a virtual country had to have been completely covered by the WLMS dataset. We applied both of these criteria here as well when reproducing the regressions.

4.2 Replication

The model specification for this analysis is almost exactly the same as before, except now each unit i is a virtual country, and "numLang" is really the number of ethnic groups:

$$\ln(\text{numLang}_i) = \beta_0 + \beta_1 * \text{absLat}_i + \beta_2 * \text{sdElev}_i + \beta_3 * \text{sdSuitable}_i + \beta_4 * X_i + \epsilon_i \tag{2}$$

Additionally, regressions 2.5, 2.6, and 2.7 (the second number corresponds to the columns in Table 8) are performed only on virtual countries meeting a certain criterion. Regression 2.5 looks only at virtual countries located in the tropics, 2.6 looks at countries not located in the tropics, and 2.7 filters to virtual countries that are located entirely within a real country.

Table 8 shows the results of regressing the log of number of ethnic groups on different sets of features, reproducing Table 4 of the original paper. Note that the features here are very similar to the ones used in the cross-country analysis. The only differences are that features directly relating to real countries, such as independence year, have been swapped for features describing the position of the virtual country in relation to real countries. Table 9 shows information about each model, including the observed R^2 in the original WLMS regression, and our GREG regression.

Here, the reported standard errors are cluster-robust, with the clusters defined by the real country in which the centroid of each virtual country falls. Whether Stata applies any corrections by default is unclear, but the base formula should be the same as the one described in Section 24.4.1 of Peng Ding's lecture notes (Ding 2022), which we implemented in R using the `sandwich` and `lmtest` packages. Michalopolous did not justify his decision to cluster by real country, and the reader would not know that he did so without

Table 8: Main specification for the virtual country analysis. Italics indicate significance at the 1% level.

Variable	(1)	(2)	(3)	(4)	(5)	(6)	(7)
Variation in elevation		*0.156*	*0.130*	*0.089*	-0.058	*0.158*	*0.145*
		(0.056)	*(0.047)*	*(0.033)*	(0.050)	*(0.042)*	*(0.041)*
Variation in land quality		*0.202*	*0.136*	*0.180*	0.103	*0.213*	*0.180*
		(0.054)	*(0.045)*	*(0.030)*	(0.053)	*(0.038)*	*(0.042)*
Mean elevation		-0.059	-0.111	*-0.138*	0.065	-0.108	-0.176
		(0.042)	(0.064)	*(0.051)*	(0.161)	(0.079)	(0.079)
Mean land quality		0.042	0.065	0.072	0.068	0.093	0.071
		(0.076)	(0.039)	(0.037)	(0.061)	(0.046)	(0.056)
Absolute latitude	*-0.196*	-0.146	*-0.382*	*-0.553*	-0.070	-0.154	*-0.728*
	(0.063)	(0.089)	*(0.144)*	*(0.162)*	(0.109)	(0.190)	*(0.265)*
Mean precipitation			*0.162*	0.036	0.134	-0.033	-0.026
			(0.056)	(0.046)	(0.108)	(0.048)	(0.070)
Mean temperature			-0.332	*-0.472*	0.111	-0.229	-0.492
			(0.150)	*(0.136)*	(0.174)	(0.178)	(0.215)
Ln(Area)			-0.096	0.001	0.013	0.030	-0.005
			(0.056)	(0.029)	(0.058)	(0.033)	(0.041)
Distance from the sea			*0.208*	*0.156*	*0.179*	*0.149*	*0.206*
			(0.044)	*(0.029)*	*(0.065)*	*(0.035)*	*(0.038)*
Water area			-0.006	-0.027	-0.066	-0.007	-0.013
			(0.028)	(0.023)	(0.035)	(0.031)	(0.038)
Within-country indicator			-0.091	-0.088	0.025	*-0.151*	
			(0.054)	(0.040)	(0.063)	*(0.050)*	
Number of countries			*0.293*	*0.260*	*0.294*	*0.229*	
			(0.038)	*(0.044)*	*(0.066)*	*(0.059)*	
Migratory distance from Ethiopia			-0.118	-0.378	-0.991	-0.118	-0.482
			(0.091)	(0.175)	(0.716)	(0.223)	(0.252)
Ln(Population density in 1995)				0.008	*0.267*	-0.051	0.009
				(0.039)	*(0.070)*	(0.049)	(0.053)

Table 9: Information for each model in virtual country analysis.

Model	Country Indicators	Observations	WLMS R^2	GREG R^2
1	No	1449	0.31	0.04
2	No	1449	0.36	0.12
3	No	1449	0.53	0.34
4	Yes	1449	0.70	0.56
5	Yes	447	0.73	0.65
6	Yes	1002	0.56	0.54
7	Yes	860	0.66	0.49

checking the footnotes or his code. However, it seems a reasonable decision, considering that virtual countries within the same real country are certainly related beyond any similarities in their features.

Additionally, models 2.4 through 2.7 include a fixed effect for each country. Again, exactly how to replicate the Stata code was not obvious, but the `lm_robust` function from the `estimatr` package appeared to work. This technique essentially just includes one indicator variable for each real country in the model, and then ignores the estimates for those variables in the output. Per Michalopolous:

> Such inclusion of powerful controls, not possible in a cross-country framework, allows me to explicitly take into account any systematic elements related to the nation-building process of current states and thus produce reliable estimates of the effect of geographic heterogeneity on ethnic diversity.

4.3 Comparison

One major difference between GREG and WLMS is that the footprint of each unique ethnic group in GREG is larger than that for each unique language in the WLMS. This makes sense considering that GREG only contains about 900 entries, yet there are a few thousand unique languages in the WLMS. As a result, the dependent variable is generally a lot smaller in our replication compared to that in the original paper. Michalopolous reports the median number of languages per virtual country as 3, yet here, more than half of the virtual countries contain only one ethnic group. A possible solution to adjust for this would be to simply reduce the size of the virtual countries, but this was infeasible due to our inability to recalculate the rest of the features.

As for the actual results, the first noticeable difference in the R^2 coefficients. In all models, the R^2 is considerably lower in our replication compared to the original, although the differences are smaller for the models including country fixed effects. For example, absolute latitude on its own explains just 4% of the variation in log number of ethnic groups, compared to 31% in the original. However, the coefficient is still significant at the 1% level and has a negative sign, as it does in the original.

Beyond the worse fits overall, the inference around the coefficients does not differ much between the two models. If a variable is significant at the 1% level in one model, there is a good chance it is significant at at least the 5% or 10% level in its counterpart, or vice versa. Variation in elevation and variation in land quality continue to have a large relationship with the outcome, lending further evidence to the author's original hypothesis. The number of real countries intersected by a virtual country is also a really strong predictor in both sets of models. Michalopolous ponders that this "may be suggestive of the effect of state formation on ethnic diversity and/or an artifact of modern states having drawn political borders along ethnic boundaries." This certainly seems reasonable, but given how diminished the effects of some of the other variables are in our replication, it is surprising that this one is still so large. Perhaps this is another artifact of the aforementioned reduced granularity of GREG compared to WLMS: ethnic groups are even more strongly correlated with national boundaries than languages are.

However, there are a few noteworthy inconsistencies. One is that distance from the sea is significant for every model here, but only for model 2.5 (tropical locations) in the original. Another oddity of the tropical model is that the sign for variation in elevation flips to negative, and log of population density becomes a strong predictor. One reason for this could be differences in how virtual countries in tropical areas were filtered in the GREG dataset compared to WLMS. An alternative, real world, explanation is that the densely-populated regions in the tropics are usually found at higher altitudes (e.g., in Kenya), where some of the downsides of the climate (such as malaria, etc.) are mitigated. Since densely populated,

Table 10: Table 5A—Robustness Checks for the Virtual Country Analysis. Italics indicate significance at the 1% level.

Variable	Negative binomial	OLS1	OLS2
Variation in elevation	0.221	*0.309*	*0.345*
	(0.136)	*(0.095)*	*(0.115)*
Variation in land quality	0.704	*1.198*	*1.211*
	(0.335)	*(0.210)*	*(0.232)*

urban areas usually correspond to a lessened number of separate ethnic groups, the sign of the coefficient is less surprising; for there to be a variation in elevation, one needs to have some elevation in the region in the first place, so the tropical regions without such better habitable areas are not affected.

5 Robustness Check for Analysis 2

5.1 Robustness Check 1

Table 10 shows the first series of robustness checks for our second model. The starting point for all is regression 2.4 in the previous section; only the usual two most decisive covariates are shown, however (as in the original work).

The first regression is, similarly as in the first model, a negative binomial. To obtain the cluster-robust standard error, we use the `geem` routine from the eponymous package. The clusters are again defined to be real-world countries in which the centroid of each virtual country falls. However, the routine does not include an optimization subroutine for the dispersion parameter. Thus, the negative binomial regression is first performed with the usual `glm` call to obtain the MLE for the dispersion parameter, which is then plugged into the `geem` call as a given.

The second and third are robust OLS (`lm_robust` from `estimatr`) regressions against the log of the number of ethnic groups in the country. The change here is that the second one includes all virtual countries irrespective of their population (the original filtered them by having at least 3000 inhabitants), while the third ups that threshold to 50000 inhabitants.

The original paper also included two additional regressions, which filtered the original dependent variable — the number of speakers — by size. Since we don't have that data and are using a substitute variable — the number of ethnic groups — we were unable to replicate those two regressions. All the regressions include the per-country fixed-effect variables.

The results show a very good match in terms of both the significance and magnitude of the two variables under consideration for the two ordinary least squares regressions, while the negative binomial one is less able to reproduce the significance obtained from using the WLMS dataset. The R^2 coefficients (c.f. Table 11) are, while again lower than in the original, still reasonably high. The higher likelihood of the negative binomial when compared to the original is likely mostly due to the lower number of observations and should not be taken as very authoritative.

Nonetheless, the robustness checks on an alternative dataset seem to confirm the strong relationship these two explanatory variables have with ethnolinguistic diversity.

Table 11: Information for each model in Table 5A: cross-country robustness check

Model	Country Indicators	Observations	R^2	Log pseudolikelihood
1	Yes	1449	-	-2591.32
2	Yes	1638	0.53	-
3	Yes	1153	0.60	-

Table 12: Table 5B—Robustness Checks for the Virtual Country Analysis. Italics indicate significance at the 1% level.

Variable	(1)	(2)	(3)	(4)	(5)
Variation in elevation	0.288	*0.308*		0.230	*0.347*
	(0.134)	*(0.101)*		(0.100)	*(0.099)*
Variation in land quality	*1.398*	*1.231*			
	(0.270)	*(0.219)*			
Dispersion of elevation			*0.108*		
			(0.034)		
Dispersion of land quality			*0.435*		
			(0.071)		
Variation in climatic suitability				*1.130*	
				(0.184)	
Mean climatic suitability				*0.345*	
				(0.079)	
Variation in soil suitability					*0.966*
					(0.185)
Mean soil suitability					*0.223*
					(0.078)

Table 13: Information for each model in Table 5B: cross-country robustness check

Model	Country Indicators	Observations	R^2
1	Yes	942	0.54
2	Yes	1448	0.61
3	Yes	1449	0.56
4	Yes	1449	0.56
5	Yes	1449	0.56

5.2 Robustness Check 2

Table 12 presents the results of the second series of robustness checks on the second model. All are ordinary least squares regressions with per-country fixed effects, while the errors reported are cluster-robust. As in section Table 2B, the two most important covariates — the variations in elevation and land quality — are substituted first with their dispersions, and the latter subsequently with its component parts.

The first regression is identical to regression 2.4 in the previous section in terms of covariates, but the data is filtered so only the virtual countries which have complete coverage on all of their underlying 25 cells are included. The second regression adds an additional fixed effect for each percentile of the size distribution of the virtual countries: we can stipulate that this is to better capture the case if the dynamics are not stable across varying orders of magnitude in country size.

As mentioned, regressions 3 - 5 follow section Table 2B in their choice of the substitutions for the two most important covariates: dispersions instead of variations, and climatic and soil suitability instead of the overall composite metric. All regressions include per-country fixed effects.

The results reasonably replicate those done on WLMS in terms of variable significance, except the variation in elevation is found to be less significant in regressions 1 and 4. This might be attributed to the fact that, as mentioned, GREG contains much fewer ethnic groups than WLMS does languages.

Table 13 shows the R^2 coefficient for the regressions. The values are comparable to the best models in the initial set of regressions. The best-performing model in terms of this metric is the second one, which includes fixed-effects that take the country size into consideration. This implies that, while it still holds that the two variations are the biggest explanatory factors for ethnolinguistic diversity, the dynamics themselves are not completely invariant to the size of the (virtual) country.

6 Conclusion

Our replication seems to give credence to the hypothesis put forth in the original paper:

> The empirical analysis conducted ... establishes that geographic variability, captured by variation in regional land quality and elevation, is a fundamental determinant of contemporary linguistic diversity. The findings are consistent with the proposed hypothesis that differences in land endowments gave rise to location-specific human capital, leading to the formation of localized ethnicities.

A representative example of the formation of separate ethnicities being induced by a high variance in the land quality and elevation is that of the latter two allowing for two separate populations in a relatively small geographic area to practice drastically different modes of subsistence. The (usually) low-lying, higher quality land is conductive to intensive farming, while the adjacent lower quality, hillier terrain permits only pastoralism. With time, these two populations develop separate customs and languages, leading to a separate sense of ethnic belonging.

7 References

Ding, Peng. 2022. "Linear Model and Extensions." University of California, Berkeley.

Michalopoulos, Stelios. 2012. "The Origins of Ethnolinguistic Diversity." *American Economic Review* 102 (4): 1508–39. https://doi.org/10.1257/aer.102.4.1508.

Weidmann, Nils B., Jan Ketil Rød, and Lars-Erik Cederman. 2010. "Representing Ethnic Groups in Space: A New Dataset." *Journal of Peace Research* 47 (4): 491–99. https://doi.org/10.1177/0022343310368352.

8 Appendix: code

```
cells = read_sf(dsn = 'data_raw/Virtual_country', layer = 'virtual_cntrygrid')
countries = read_sf(dsn = 'countries', layer = 'countries')
data = read.dta13("data_raw/Tables1-3a.dta")
colnames(data) = c('countryCode', 'entryYear', 'countryName', 'avgTemp',
                   'avgPrecip', 'seaDist', 'avgElev', 'sdElev', 'absLat',
                   'dispElev', 'numLang', 'suitableCells', 'dispSuitable',
                   'climate', 'soil', 'sdClimate', 'sdSoil', 'sdSuitable',
                   'avgSuitable', 'pop95', 'area', 'lnLang', 'africa',
                   'europe', 'americas', 'lnPopDens1995', 'migrationDist',
                   'lnArea', 'pctIndigenous', 'lnPopDens1500',
                   'agriTran', 'asiaPac')
greeceCells = countries %>% filter(COUNTRY == 'Greece') %>%
  st_intersection(y = cells)
nepalCells = countries %>% filter(COUNTRY == 'Nepal') %>%
  st_intersection(y = cells)
plot(
  density(greeceCells$suit_new, kernel = "epanechnikov"),
  xlim = c(0, 1),
  xlab = 'Land quality per region',
  ylab = 'Density',
  main = '',
  lty = 2
)
lines(density(nepalCells$suit_new, kernel = "epanechnikov"), col = 'red')
legend(
  .1,
  3,
  legend = c(
    'Distribution of land quality in Greece',
    'Distribution of land quality in Nepal'
  ),
  col = c("black", "red"),
  lty = 2:1,
  cex = .75
)
count = function(x) {
  (sum( ~ is.na(x)))
}

sumTable <- data %>% select(
  c(
    'numLang',
    'sdElev',
    'sdSuitable',
    'avgElev',
    'avgSuitable',
    'absLat',
    'avgPrecip',
    'avgTemp',
```

```
    'lnArea',
    'seaDist',
    'migrationDist',
    'lnPopDens1995'
  )
) %>%
  summarise_each(
    funs(
      min = min,
      median = median,
      max = max,
      mean = mean,
      iqr = quantile(., 0.75) - quantile(., 0.25),
      sd = sd,
      n = sum(!is.na(.))
    )
  ) %>%
  gather(var, val) %>%
  separate(var, into = c("var", "stat"), sep = "_") %>%
  spread(var, val) %>% column_to_rownames(var = "stat") %>%
  select(
    c(
      'numLang',
      'sdElev',
      'sdSuitable',
      'avgElev',
      'avgSuitable',
      'absLat',
      'avgPrecip',
      'avgTemp',
      'lnArea',
      'seaDist',
      'migrationDist',
      'lnPopDens1995'
    )
  ) %>%
  mutate_if(is.numeric, ~ round(., 2)) %>% slice(5, 4, 2, 3, 7)
standardize = function(vec) {return ((vec - mean(vec, na.rm = TRUE)) / sd(vec, na.rm = TRUE))}

modelCols = c('entryYear', 'avgTemp', 'avgPrecip', 'seaDist', 'avgElev',
              'sdElev', 'absLat', 'numLang', 'dispSuitable', 'climate',
              'soil', 'sdClimate', 'sdSoil', 'sdSuitable', 'avgSuitable',
              'pop95', 'area', 'lnLang', 'lnPopDens1995', 'migrationDist',
              'lnArea', 'pctIndigenous', 'lnPopDens1500', 'agriTran',
              'americas', 'europe', 'africa', 'asiaPac')
# for (col in modelCols) {
#  data[,col] = standardize(data[,col])
# }
dataStd = data %>% mutate(across(!countryName & !countryCode , standardize))
model1.1 = lm(lnLang ~ absLat, dataStd)
```

```
model1.2 = lm(lnLang ~ sdElev + sdSuitable + avgElev + avgSuitable + absLat, dataStd)
model1.3 = lm(lnLang ~ sdElev + sdSuitable + avgElev + avgSuitable + absLat
            + avgPrecip + avgTemp + lnArea + seaDist + migrationDist, dataStd)
model1.4 = lm(lnLang ~ sdElev + sdSuitable + avgElev + avgSuitable + absLat
            + avgPrecip + avgTemp + lnArea + seaDist + migrationDist + lnPopDens1995
            + africa + europe + americas + asiaPac, dataStd)
missingData = is.na(dataStd$agriTran) | is.na(dataStd$entryYear) | is.na(dataStd$lnPopDens1500)
for (col in modelCols) {
  dataStd[!missingData, col] = standardize(dataStd[!missingData,col])
}

model1.5 = lm(lnLang ~ sdElev + sdSuitable + avgElev + avgSuitable + absLat
            + avgPrecip + avgTemp + lnArea + seaDist + migrationDist + lnPopDens1995
            + lnPopDens1500 + entryYear + agriTran
            + africa + europe + americas + asiaPac, dataStd, na.action = na.exclude)
models = paste0("model1.", 1:5)
coefs = sapply(models, function(model) {coeftest(get(model),
                                                 vcov = vcovHC(get(model), "HC1"))[, 1]}) %>%
  unlist() %>% data.frame()
coefs$model = substr(row.names(coefs), 1, 8)
coefs$column = substr(row.names(coefs), 10, nchar(row.names(coefs)))

ses = sapply(models, function(model) {coeftest(get(model),
                                               vcov = vcovHC(get(model), "HC1"))[, 2]}) %>%
  unlist() %>% data.frame()
ses$model = substr(row.names(ses), 1, 8)
ses$column = substr(row.names(ses), 10, nchar(row.names(ses)))

pvals = sapply(models, function(model) {coeftest(get(model),
                                                 vcov = vcovHC(get(model), "HC1"))[, 4]}) %>%
  unlist() %>% data.frame()
pvals$model = substr(row.names(pvals), 1, 8)
pvals$column = substr(row.names(pvals), 10, nchar(row.names(pvals)))

order = c('sdElev', 'sdSuitable', 'avgElev', 'avgSuitable', 'absLat',
          'avgPrecip', 'avgTemp', 'lnArea', 'seaDist', 'migrationDist',
          'lnPopDens1995', 'lnPopDens1500', 'entryYear', 'agriTran')

pvalsPivoted = pvals %>% pivot_wider(names_from = "model", values_from = '.') %>%
  slice(match(order, column))

tbl1 = rbind(coefs %>% pivot_wider(names_from = "model", values_from = '.'),
             ses %>% pivot_wider(names_from = "model", values_from = '.'))
tbl1$stat = c(rep('Estimate', 19), rep('SE', 19))
indices = c(rbind(match(order, tbl1$column), match(order, tbl1$column) + 19))
tbl1 = tbl1 %>% slice(indices)
tbl1format = data.frame(tbl1)
for (model in models) {
  estimRows = !is.na(tbl1[, model]) & (tbl1$stat == 'Estimate')
  seRows = !is.na(tbl1[, model]) & (tbl1$stat == 'SE')
```

```
  tbl1format[estimRows, model] = sprintf(fmt = "%.3f",
                                         tbl1[estimRows, model] %>% unlist() %>% as.numeric())
  tbl1format[seRows, model] = paste0("(", sprintf(fmt = "%.3f",
                                                  tbl1[seRows, model] %>%
                                                    unlist() %>% as.numeric()), ")")

  significant = rep(pvalsPivoted[, model] < .01, each = 2)
  significant[is.na(significant)] = FALSE
  tbl1format[estimRows, model] = cell_spec(tbl1format[estimRows, model],
                                           italic = significant[estimRows])
  tbl1format[seRows, model] = cell_spec(tbl1format[seRows, model], italic = significant[seRows])
}

tbl1format$name = c('Variation in elevation', NA, 'Variation in land quality', NA,
                    'Mean elevation', NA, 'Mean land quality', NA,
                    'Absolute latitude', NA, 'Mean precipitation', NA,
                    'Mean temperature', NA, 'Ln(Area)', NA,
                    'Distance from the sea', NA, 'Migratory distance from East Africa', NA,
                    'Ln(Population density in 1995)', NA, 'Ln(Population density in 1500)', NA,
                    'Year of independence', NA, 'Timing of transition to agriculture', NA
                    )

col.names = c("Variable", paste0("(", 1:5, ")"))

tbl1format %>% select(8, 2:6) %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE, linesep = c("", "\\addlinespace"),
               col.names = col.names, align = "r", escape = F,
               caption = "Main specification for the cross-country analysis. Italics indicate significance at the
  row_spec(seq(2, 28, 2), font_size = 8)
tbl1info = data.frame(
  model = 1:5,
  cont = c("No", "No", "No", "No", "Yes"),
  nobs = sapply(models, function(model) { nobs(get(model))}),
  rsq = sapply(models, function(model) {
    formatC(summary(get(model))$r.squared, digits = 2, format = 'f')
    }), row.names = NULL
  )
tbl1info %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE,
               col.names = c('Model', 'Continental Indicators', 'Observations', '$R^2$'),
               escape = F,
               caption = "Information for each model in cross-country analysis.")
yhat = model1.5$residuals
y = model1.5$model$lnLang
plot(y, yhat, xlab = "", ylab = "", cex.axis = .75)
title(ylab = "Residuals", xlab = "Ln(Number of Languages)", mgp = c(2, .5, 0), cex.lab = .75)
residModel = lm(yhat ~ y)
abline(0, 0)
lines(y, y * residModel$coefficients['y'], col = 'red', type = 'l')
robust1.1 <- glm.nb(numLang ~ absLat + sdSuitable + sdElev + avgElev + avgSuitable + avgPrecip +
```

```
                    avgTemp + lnArea + seaDist + migrationDist + lnPopDens1995 + lnPopDens1500 +
                    entryYear + agriTran + africa + europe + americas + asiaPac
                    , data, na.action = na.exclude)
robust1.2 <- lm(lnLang ~ absLat + dispElev + dispSuitable + avgElev + avgSuitable + avgPrecip +
                avgTemp +  lnArea +  seaDist + migrationDist +  lnPopDens1995 + lnPopDens1500 +
                entryYear + agriTran + africa + europe + americas + asiaPac,
                data, na.action = na.exclude)

robust1.3 <- lm(lnLang ~ absLat + sdElev + sdClimate + avgElev + climate + avgPrecip +  avgTemp +
                lnArea +  seaDist + migrationDist +  lnPopDens1995 +  lnPopDens1500 + entryYear +
                agriTran + africa + europe + americas + asiaPac,
                data, na.action = na.exclude)

# NOTE: the conditional doesn't remove anything from the `data` df
data1.4 <- data[(data$suitableCells > 9) & (data$lnArea > -10), ]

robust1.4 <- lm(lnLang ~ absLat + sdElev + sdSoil + avgElev + soil + avgPrecip +  avgTemp +
                lnArea +  seaDist + migrationDist +  lnPopDens1995 +  lnPopDens1500 + entryYear +
                agriTran + africa + europe + americas + asiaPac,
                data1.4, na.action = na.exclude)

models <- paste0("robust1.", 1:4)
coeffs <- lapply(models, function(model) {coeftest(get(model),
                                                   vcov = vcovHC(get(model), "HC1"))[, 1]})
names(coeffs) <- models

vars <- unique(unlist(lapply(coeffs, names)))

coeffDf <- data.frame(row.names = vars)
for(model in models) coeffDf[names(coeffs[[model]]), model] <- coeffs[[model]]

ses <- lapply(models, function(model) {coeftest(get(model), vcov = vcovHC(get(model),
                                                                          "HC1"))[, 2]})
names(ses) <- models
sesDf <- data.frame(row.names = vars)
for(model in models) sesDf[names(ses[[model]]), model] <- ses[[model]]

pvals <- lapply(models, function(model) {coeftest(get(model), vcov = vcovHC(get(model),
                                                                            "HC1"))[, 4]})
names(pvals) <- models
pvalsDf <- data.frame(row.names = vars)
for(model in models) pvalsDf[names(pvals[[model]]), model] <- pvals[[model]]

order <-c('sdElev', 'sdSuitable', 'dispElev', 'dispSuitable', 'sdClimate',
          'climate', 'sdSoil', 'soil')

coeffDf <- coeffDf[order, ]
sesDf   <- sesDf[order, ]
pvalsDf <- pvalsDf[order, ]
```

```
# interleave rows
coeffDf[, "stat"] <- "Estimate"
sesDf[, "stat"] <- "SE"
tbl2a <- gdata::interleave(coeffDf, sesDf)
# ensure a hard copy
tbl2aformat <- data.frame(tbl2a)

for (model in models) {
  estimRows <- !is.na(tbl2aformat[, model]) & (tbl2aformat$stat == 'Estimate')
  seRows <- !is.na(tbl2aformat[, model]) & (tbl2aformat$stat == 'SE')

  tbl2aformat[estimRows, model] <- sprintf(fmt = "%.3f", tbl2aformat[estimRows, model]
                                           %>% unlist() %>% as.numeric())
  tbl2aformat[seRows, model] <- paste0("(", sprintf(fmt = "%.3f", tbl2aformat[seRows, model]
                                            %>% unlist()
                                            %>% as.numeric()), ")")

  significant <- rep(pvalsDf[, model] < .01, each = 2)
  significant[is.na(significant)] <- FALSE

  tbl2aformat[estimRows, model] <- cell_spec(tbl2aformat[estimRows, model],
                                             italic = significant[estimRows])
  tbl2aformat[seRows, model] <- cell_spec(tbl2aformat[seRows, model],
                                          italic = significant[seRows])
}

tbl2aformat$name <- c('Variation in elevation', NA, 'Variation in land quality', NA,
                      'Dispersion of elevation', NA, 'Dispersion of land quality', NA,
                      'Variation in climatic suitability', NA, 'Mean climatic suitability', NA,
                      'Variation in soil suitability', NA, 'Mean soil suitability', NA)

col.names <- c("Variable", "Negative binomial", "OLS1", "OLS2", "OLS3")

tbl2aformat %>%
  select(name, 1:4, -stat) %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE, linesep = c("", "\\addlinespace"),
               col.names = col.names,
               row.names = F,
               align = "r", escape = F,
               caption = "Table 2A-Robustness Checks for the Cross-Country Analysis. Italics indicate significanc
  row_spec(seq(2, nrow(tbl2aformat), 2), font_size = 8)

tbl2ainfo <-  data.frame(
  model = 1:4,
  cont = c("Yes", "Yes", "Yes", "Yes"),
  nobs = sapply(models, function(model) { nobs(get(model))}),
  rsq = c("-", lapply(models[-1], function(model) {
    formatC(
      summary(get(model))$r.squared
      , digits = 2, format = 'f')
```

```
    })) %>% unlist,
  loglik = c(formatC(robust1.1$twologlik / 2, digits = 2, format = 'f'), rep("-", 3)),
  row.names = NULL
  )

tbl2ainfo %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE,
               col.names = c('Model', 'Continental Indicators', 'Observations', '$R^2$',
                             "Log pseudolikelihood"),
               escape = F,
               caption = "Information for each model in Table 2A: cross-country robustness check")
data2b <- read.dta13("data_raw/Table_3b.dta")

standardized <- list("lpd1500", "yrentry", "agritran", "elf", "elf3", "elf5", "elf7", "elf9",
                     "abs_lat", "sd_climsuit", "sd_emean", "emean", "mean_climsuit", "precav",
                     "tempav", "lnareakm2", "distc", "migdist", "lnpop95", "americas", "reg_eap",
                     "africa", "europe", "nmbr_climsuit")

notStdized <- names(data2b)[! names(data2b) %in%  standardized]
data2bStd <- data2b %>% mutate(across(! all_of(notStdized) , standardize))

robust1.2.1 <- lm(elf ~ abs_lat, data2bStd, na.action = na.exclude)

robust1.2.2 <- lm(elf ~ abs_lat + sd_emean + sd_climsuit + emean + mean_climsuit,
                  data2bStd, na.action = na.exclude)

robust1.2.3 <- lm(elf ~ abs_lat + sd_emean + sd_climsuit + emean + mean_climsuit +
                  precav + tempav + lnareakm2 + distc + migdist + lnpop95 + lpd1500 + yrentry +
                  agritran + africa + europe + americas + reg_eap
                  , data2bStd, na.action = na.exclude)

robust1.2.4 <- lm(elf3 ~ abs_lat + sd_emean + sd_climsuit + emean + mean_climsuit +
                  precav + tempav + lnareakm2 + distc + migdist + lnpop95 + lpd1500 + yrentry +
                  agritran + africa + europe + americas + reg_eap
                  , data2bStd, na.action = na.exclude)

robust1.2.5 <- lm(elf5 ~ abs_lat + sd_emean + sd_climsuit + emean + mean_climsuit +
                  precav + tempav + lnareakm2 + distc + migdist + lnpop95 + lpd1500 + yrentry +
                  agritran + africa + europe + americas + reg_eap
                  , data2bStd, na.action = na.exclude)

robust1.2.6 <- lm(elf7 ~ abs_lat + sd_emean + sd_climsuit + emean + mean_climsuit +
                  precav + tempav + lnareakm2 + distc + migdist + lnpop95 + lpd1500 + yrentry +
                  agritran + africa + europe + americas + reg_eap
                  , data2bStd, na.action = na.exclude)

robust1.2.7 <- lm(elf9 ~ abs_lat + sd_emean + sd_climsuit + emean + mean_climsuit +
                  precav + tempav + lnareakm2 + distc + migdist + lnpop95 + lpd1500 + yrentry +
                  agritran + africa + europe + americas + reg_eap
                  , data2bStd, na.action = na.exclude)
```

```
models <- paste0("robust1.2.", 1:7)
coeffs <- lapply(models, function(model) {coeftest(get(model), vcov = vcovHC(get(model),
                                                                              "HC1"))[, 1]})
names(coeffs) <- models

vars <- unique(unlist(lapply(coeffs, names)))

coeffDf <- data.frame(row.names = vars)
for(model in models) coeffDf[names(coeffs[[model]]), model] <- coeffs[[model]]

ses <- lapply(models, function(model) {coeftest(get(model), vcov = vcovHC(get(model),
                                                                          "HC1"))[, 2]})
names(ses) <- models
sesDf <- data.frame(row.names = vars)
for(model in models) sesDf[names(ses[[model]]), model] <- ses[[model]]

pvals <- lapply(models, function(model) {coeftest(get(model), vcov = vcovHC(get(model),
                                                                            "HC1"))[, 4]})
names(pvals) <- models
pvalsDf <- data.frame(row.names = vars)
for(model in models) pvalsDf[names(pvals[[model]]), model] <- pvals[[model]]

order <-c('sd_emean', 'sd_climsuit', 'emean', 'mean_climsuit', 'abs_lat',
          'precav', 'tempav', 'lnareakm2', 'distc', 'migdist', 'lnpop95', 'lpd1500',
          'yrentry', 'agritran')

coeffDf <- coeffDf[order, ]
sesDf   <- sesDf[order, ]
pvalsDf <- pvalsDf[order, ]

# interleave rows
coeffDf[, "stat"] <- "Estimate"
sesDf[, "stat"] <- "SE"
tbl2b <- gdata::interleave(coeffDf, sesDf)
# ensure a hard copy
tbl2bformat <- data.frame(tbl2b)

for (model in models) {
  estimRows <- !is.na(tbl2bformat[, model]) & (tbl2bformat$stat == 'Estimate')
  seRows <- !is.na(tbl2bformat[, model]) & (tbl2bformat$stat == 'SE')

  tbl2bformat[estimRows, model] <- sprintf(fmt = "%.3f", tbl2bformat[estimRows, model]
                                           %>% unlist() %>% as.numeric())
  tbl2bformat[seRows, model] <- paste0("(", sprintf(fmt = "%.3f", tbl2bformat[seRows, model]
                                            %>% unlist()
                                            %>% as.numeric()), ")")

  significant <- rep(pvalsDf[, model] < .01, each = 2)
  significant[is.na(significant)] <- FALSE
```

```
  tbl2bformat[estimRows, model] <- cell_spec(tbl2bformat[estimRows, model],
                                             italic = significant[estimRows])
  tbl2bformat[seRows, model] <- cell_spec(tbl2bformat[seRows, model],
                                          italic = significant[seRows])
}

tbl2bformat$name <- c('Variation in elevation', NA, 'Variation in climatic suitability', NA,
                      'Mean elevation', NA, 'Mean climatic suitability', NA,
                      'Absolute latitude', NA, 'Mean precipitation', NA,
                      'Mean temperature', NA, 'Ln(area)', NA,
                      'Distance from the sea', NA, 'Migratory distance from East Africa', NA,
                      'Ln(Population density in 1995)', NA, 'Ln(Population density in 1500)', NA,
                      'Year of independence', NA, 'Timing of transition to agriculture', NA
                      )

col.names <- c("Variable", paste0("(", 1:7, ")"))

tbl2bformat %>%
  select(name, 1:7, -stat) %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE, linesep = c("", "\\addlinespace"),
               col.names = col.names,
               row.names = F,
               align = "r", escape = F,
               caption = "Table 2B-Linguistic Fractionalization across Countries. Italics indicate significance a
  row_spec(seq(2, nrow(tbl2bformat), 2), font_size = 8)

tbl2binfo <-  data.frame(
  model = 1:7,
  cont = c(rep("No", 2), rep("Yes", 5)),
  nobs = sapply(models, function(model) { nobs(get(model))}),
  rsq = sapply(models, function(model) {
    formatC(summary(get(model))$r.squared, digits = 2, format = 'f')
    }),
  row.names = NULL
  )

tbl2binfo %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE,
               col.names = c('Model', 'Continental Indicators', 'Observations', '$R^2$'),
               escape = F,
               row.names = F,
               caption = "Information for each model in Table 2B: Linguistic Fractionalization across Countries
data2 = read.dta13('data_raw/Tables4-7b.dta')
greg = read.csv('greg.csv')
colnames(greg) = c('uniq_cnt25', 'number_suit_valid25', 'nmbrlang')

data2 = data2 %>% select(-c('nmbrlang', 'number_suit_valid25')) %>% merge(greg, by = 'uniq_cnt25')

data2$lnnmbrlang = log(data2$nmbrlang)
colnames(data2) = c('virtCode', 'countryCode', 'climate', 'soil',
```

```
                  'sdClimate', 'sdSoil', 'seaDist', 'avgElev', 'avgPrecip',
                  'avgTemp', 'sdElev', 'waterArea', 'avgSuitable',
                  'sdSuitable', 'popDens95', 'dispSuitable', 'area', 'withinCountry',
                  'numCountry', 'migrationDist', 'lnLang', 'totalPop95',
                  'absLat', 'tropics', 'dispElev', 'lnArea',
                  'lnPopDens95', 'pctIndigenous', 'diffAvgElev',
                  'diffAvgPrecip', 'diffAvgTemp', 'diffAvgSuit',
                  'overlap', 'suitableCells', 'numLang')

modelCols = c('lnLang', 'sdElev', 'sdSuitable', 'avgElev', 'avgSuitable',
              'absLat', 'avgPrecip', 'avgTemp', 'lnArea', 'seaDist', 'waterArea',
              'withinCountry', 'numCountry', 'migrationDist', 'lnPopDens95')
condition = (data2$totalPop95 >= 3000) & (data2$suitableCells >= 10)
for (col in modelCols) {
  data2[condition, paste0(col, '1')] = standardize(data2[condition, col])
}
model2.1 = lm(lnLang1 ~ absLat1, data2 %>% filter(condition))
coefs = coeftest(model2.1, vcov = vcovCL, cluster = ~countryCode)[, 1]
ses = coeftest(model2.1, vcov = vcovCL, cluster = ~countryCode)[, 2]
pvals = coeftest(model2.1, vcov = vcovCL, cluster = ~countryCode)[, 4]
model2.2 = lm(lnLang1 ~ sdElev1 + sdSuitable1 + avgElev1 + avgSuitable1 + absLat1,
            data2 %>% filter(condition))
coefs = c(coefs, coeftest(model2.2, vcov = vcovCL, cluster = ~countryCode)[, 1])
ses = c(ses, coeftest(model2.2, vcov = vcovCL, cluster = ~countryCode)[, 2])
pvals = c(pvals, coeftest(model2.2, vcov = vcovCL, cluster = ~countryCode)[, 4])
model2.3 = lm(lnLang1 ~ sdElev1 + sdSuitable1 + avgElev1 + avgSuitable1 + absLat1
            + avgPrecip1 + avgTemp1 + lnArea1 + seaDist1 + waterArea1
            + withinCountry1 + numCountry1 + migrationDist1,
            data2 %>% filter(condition))
coefs = c(coefs, coeftest(model2.3, vcov = vcovCL, cluster = ~countryCode)[, 1])
ses = c(ses, coeftest(model2.3, vcov = vcovCL, cluster = ~countryCode)[, 2])
pvals = c(pvals, coeftest(model2.3, vcov = vcovCL, cluster = ~countryCode)[, 4])
model2.4 = lm_robust(lnLang1 ~ sdElev1 + sdSuitable1 + avgElev1 + avgSuitable1 + absLat1
            + avgPrecip1 + avgTemp1 + lnArea1 + seaDist1 + waterArea1
            + withinCountry1 + numCountry1 + migrationDist1 + lnPopDens951,
            data2 %>% filter(condition),
            fixed_effects = ~countryCode, se_type = "stata")
coefs = c(coefs, 0, model2.4$coefficients)
ses = c(ses, 0, model2.4$std.error)
pvals = c(pvals, 0, model2.4$p.value)
condition = (data2$totalPop95 >= 3000) & (data2$suitableCells >= 10) & (data2$tropics == 1)
for (col in modelCols) {
  data2[condition, paste0(col, '5')] = standardize(data2[condition, col])
}
model2.5 = lm_robust(lnLang5 ~ sdElev5 + sdSuitable5 + avgElev5 + avgSuitable5 + absLat5
            + avgPrecip5 + avgTemp5 + lnArea5 + seaDist5 + waterArea5
            + withinCountry5 + numCountry5 + migrationDist5 + lnPopDens955,
            data2 %>% filter(condition),
            fixed_effects = ~countryCode, se_type = "stata")
coefs = c(coefs, 0, model2.5$coefficients)
```

```
ses = c(ses, 0, model2.5$std.error)
pvals = c(pvals, 0, model2.5$p.value)
condition = (data2$totalPop95 >= 3000) & (data2$suitableCells >= 10) & (data2$tropics == 0)
for (col in modelCols) {
  data2[condition, paste0(col, '6')] = standardize(data2[condition, col])
}
model2.6 = lm_robust(lnLang6 ~ sdElev6 + sdSuitable6 + avgElev6 + avgSuitable6 + absLat6
            + avgPrecip6 + avgTemp6 + lnArea6 + seaDist6 + waterArea6
            + withinCountry6 + numCountry6 + migrationDist6 + lnPopDens956,
            data2 %>% filter(condition),
            fixed_effects = ~countryCode, se_type = "stata")
coefs = c(coefs, 0, model2.6$coefficients)
ses = c(ses, 0, model2.6$std.error)
pvals = c(pvals, 0, model2.6$p.value)
condition = (data2$totalPop95 >= 3000) & (data2$suitableCells >= 10) & (data2$withinCountry == 1)
for (col in modelCols) {
  data2[condition, paste0(col, '7')] = standardize(data2[condition, col])
}
model2.7 = lm_robust(lnLang7 ~ sdElev7 + sdSuitable7 + avgElev7 + avgSuitable7 + absLat7
            + avgPrecip7 + avgTemp7 + lnArea7 + seaDist7 + waterArea7
            + migrationDist7 + lnPopDens957,
            data2 %>% filter(condition),
            fixed_effects = ~countryCode, se_type = "stata")
coefs = c(coefs, 0, model2.7$coefficients)
ses = c(ses, 0, model2.7$std.error)
pvals = c(pvals, 0, model2.7$p.value)
models = paste0("model2.", 1:7)

names(coefs)[names(coefs) == ""] = "(Intercept)"
coefs = data.frame(coefs, row.names = paste0("model2.",
                                             cumsum(names(coefs) %in% c("(Intercept)", "")),
                                             ".", names(coefs)))
coefs$model = substr(row.names(coefs), 1, 8)
coefs$column = substr(row.names(coefs), 10, nchar(row.names(coefs)) - 1)

names(ses)[names(ses) == ""] = "(Intercept)"
ses = data.frame(ses, row.names = paste0("model2.",
                                         cumsum(names(ses) %in% c("(Intercept)", "")),
                                         ".", names(ses)))
ses$model = substr(row.names(ses), 1, 8)
ses$column = substr(row.names(ses), 10, nchar(row.names(ses)) - 1)

names(pvals)[names(pvals) == ""] = "(Intercept)"
pvals = data.frame(pvals, row.names = paste0("model2.",
                                             cumsum(names(pvals) %in% c("(Intercept)", "")),
                                             ".", names(pvals)))
pvals$model = substr(row.names(pvals), 1, 8)
pvals$column = substr(row.names(pvals), 10, nchar(row.names(pvals)) - 1)

order = c('sdElev', 'sdSuitable', 'avgElev', 'avgSuitable', 'absLat',
```

```
          'avgPrecip', 'avgTemp', 'lnArea', 'seaDist', 'waterArea',
          'withinCountry', 'numCountry', 'migrationDist', 'lnPopDens95')

pvalsPivoted = pvals %>% pivot_wider(names_from = "model", values_from = 'pvals') %>%
  slice(match(order, column))

tbl4 = rbind(coefs %>% pivot_wider(names_from = "model", values_from = 'coefs'),
             ses %>% pivot_wider(names_from = "model", values_from = 'ses'))
tbl4$stat = c(rep('Estimate', 15), rep('SE', 15))
indices = c(rbind(match(order, tbl4$column), match(order, tbl4$column) + 15))
tbl4 = tbl4 %>% slice(indices)
tbl4format = data.frame(tbl4)
for (model in models) {
  estimRows = !is.na(tbl4[, model]) & (tbl4$stat == 'Estimate')
  seRows = !is.na(tbl4[, model]) & (tbl4$stat == 'SE')

  tbl4format[estimRows, model] = sprintf(fmt = "%.3f", tbl4[estimRows, model] %>%
                                           unlist() %>% as.numeric())
  tbl4format[seRows, model] = paste0("(", sprintf(fmt = "%.3f", tbl4[seRows, model] %>%
                                                    unlist() %>% as.numeric()), ")")

  significant = rep(pvalsPivoted[, model] < .01, each = 2)
  significant[is.na(significant)] = FALSE
  tbl4format[estimRows, model] = cell_spec(tbl4format[estimRows, model],
                                           italic = significant[estimRows])
  tbl4format[seRows, model] = cell_spec(tbl4format[seRows, model], italic = significant[seRows])
}

tbl4format$name = c('Variation in elevation', NA, 'Variation in land quality', NA,
                    'Mean elevation', NA, 'Mean land quality', NA,
                    'Absolute latitude', NA, 'Mean precipitation', NA,
                    'Mean temperature', NA, 'Ln(Area)', NA,
                    'Distance from the sea', NA, 'Water area', NA,
                   'Within-country indicator', NA, 'Number of countries', NA,
                   'Migratory distance from Ethiopia', NA, 'Ln(Population density in 1995)',
                NA)

col.names = c("Variable", paste0("(", 1:7, ")"))

tbl4format %>% select(10, 2:8) %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE, linesep = c("", "\\addlinespace"),
               col.names = col.names, align = "r", escape = F,
               caption = "Main specification for the virtual country analysis. Italics indicate significance at t
  row_spec(seq(2, 28, 2), font_size = 8)
tbl4info = data.frame(
  model = 1:7,
  cont = c("No", "No", "No", "Yes", "Yes", "Yes", "Yes"),
  nobs = sapply(models, function(model) { nobs(get(model))}),
  rsqOG = formatC(c(.31, .36, .53, .70, .73, .56, .66), digits = 2, format = 'f'),
  rsq = sapply(models, function(model) {
```

```
    formatC(summary(get(model))$r.squared, digits = 2, format = 'f')
    }),
  row.names = NULL
  )
tbl4info %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE,
               linesep = c(""), align = 'r',
               col.names = c('Model', 'Country Indicators',
                             'Observations', 'WLMS $R^2$', 'GREG $R^2$'),
               escape = F,
               caption = "Information for each model in virtual country analysis.")
# NOTE: Need glm for the theta, and geem for the robust se
robust2.1.1glm <-  glm.nb(numLang ~ absLat + sdElev + sdSuitable +  avgElev + avgSuitable +
                          avgPrecip +  avgTemp +  lnArea +  seaDist +
                        waterArea +  withinCountry +  numCountry + migrationDist + lnPopDens95
                        + factor(countryCode),
                        data2 %>% filter((data2$suitableCells >= 10) & (data2$totalPop95 >= 3000)),
                        na.action = na.exclude)

robust2.1.1 <- geem(formula = numLang ~ absLat + sdElev + sdSuitable +  avgElev + avgSuitable +
                             avgPrecip +avgTemp +  lnArea +  seaDist + waterArea +
                             withinCountry +  numCountry + migrationDist + lnPopDens95,
                         id = countryCode,
                         nodummy = TRUE,
                         data = data2 %>% filter((data2$suitableCells >= 10) &
                                                   (data2$totalPop95 >= 3000)),
                         family = MASS::negative.binomial(theta = robust2.1.1glm$theta,
                                                          link = 'log'),
                         sandwich = TRUE,
                         corstr = "independence",
                         scale.fix = TRUE,
                         init.phi = robust2.1.1glm$theta
                         )
robust2.1.2 <- lm_robust(lnLang ~ absLat + sdElev + sdSuitable + avgElev + avgSuitable +
                          avgPrecip + avgTemp +  lnArea + seaDist + waterArea +
                          withinCountry +  numCountry +  migrationDist +  lnPopDens95,
                          data2 %>% filter((data2$suitableCells >= 10)),
                          fixed_effects = ~countryCode,
                          se_type = "stata")
robust2.1.3 <- lm_robust(lnLang ~ absLat + sdElev + sdSuitable + avgElev + avgSuitable +
                          avgPrecip + avgTemp +  lnArea + seaDist + waterArea +
                          withinCountry +  numCountry +  migrationDist +  lnPopDens95,
                          data2 %>% filter((data2$suitableCells >= 10) &
                                             (data2$totalPop95 >= 50000)),
                          fixed_effects = ~countryCode,
                          se_type = "stata")
models <- paste0("robust2.1.", 1:3)
coeffs <- c(list(setNames(robust2.1.1$beta, robust2.1.1$coefnames)[-1]),
            lapply(models[-1], function(model) { get(model)[["coefficients"]]})
            )
```

```
names(coeffs) <- models

vars <- unique(unlist(lapply(coeffs, names)))

coeffDf <- data.frame(row.names = vars)
for(model in models) coeffDf[names(coeffs[[model]]), model] <- coeffs[[model]]

ses <- c(list(sqrt(diag(robust2.1.1$var))[-1]),
         sapply(models[-1], function(model) get(model)["std.error"])
        )
names(ses) <- models
sesDf <- data.frame(row.names = vars)
for(model in models) sesDf[names(ses[[model]]), model] <- ses[[model]]

pvals <- c(list(setNames(summary(robust2.1.1)[[5]], robust2.1.1$coefnames)[-1]),
           sapply(models[-1], function(model) get(model)["p.value"])
        )
names(pvals) <- models
pvalsDf <- data.frame(row.names = vars)
for(model in models) pvalsDf[names(pvals[[model]]), model] <- pvals[[model]]

order <-c('sdElev', 'sdSuitable')

coeffDf <- coeffDf[order, ]
sesDf   <- sesDf[order, ]
pvalsDf <- pvalsDf[order, ]

# interleave rows
coeffDf[, "stat"] <- "Estimate"
sesDf[, "stat"] <- "SE"
tbl5a <- gdata::interleave(coeffDf, sesDf)
# ensure a hard copy
tbl5aformat <- data.frame(tbl5a)

for (model in models) {
  estimRows <- !is.na(tbl5aformat[, model]) & (tbl5aformat$stat == 'Estimate')
  seRows <- !is.na(tbl5aformat[, model]) & (tbl5aformat$stat == 'SE')

  tbl5aformat[estimRows, model] <- sprintf(fmt = "%.3f", tbl5aformat[estimRows, model] %>%
                                           unlist() %>% as.numeric())
  tbl5aformat[seRows, model] <- paste0("(", sprintf(fmt = "%.3f", tbl5aformat[seRows, model]
                                           %>% unlist()
                                           %>% as.numeric()), ")")

  significant <- rep(pvalsDf[, model] < .01, each = 2)
  significant[is.na(significant)] <- FALSE

  tbl5aformat[estimRows, model] <- cell_spec(tbl5aformat[estimRows, model],
                                           italic = significant[estimRows])
  tbl5aformat[seRows, model] <- cell_spec(tbl5aformat[seRows, model],
```

```
                                        italic = significant[seRows])
}

tbl5aformat$name <- c('Variation in elevation', NA, 'Variation in land quality', NA)

col.names <- c("Variable", "Negative binomial", "OLS1", "OLS2")

tbl5aformat %>%
  select(name, 1:3, -stat) %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE, linesep = c("", "\\addlinespace"),
               col.names = col.names,
               row.names = F,
               align = "r", escape = F,
               caption = "Table 5A-Robustness Checks for the Virtual Country Analysis. Italics indicate significa
  row_spec(seq(2, nrow(tbl5aformat), 2), font_size = 8)

tbl5ainfo <- data.frame(
  model = 1:3,
  cont = rep("Yes", 3),
  nobs = sapply(c("robust2.1.1glm", models[-1]), function(model) { nobs(get(model))}),
  rsq = c("-", lapply(models[-1], function(model) {
    formatC(
      summary(get(model))$r.squared
      , digits = 2, format = 'f')
    })) %>% unlist,
  loglik = c(formatC(robust2.1.1glm$twologlik / 2, digits = 2, format = 'f'), rep("-", 2)),
  row.names = NULL
  )

tbl5ainfo %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE,
               col.names = c('Model', 'Country Indicators', 'Observations', '$R^2$',
                             "Log pseudolikelihood"),
               escape = F,
               caption = "Information for each model in Table 5A: cross-country robustness check")
robust2.2.1 <- lm_robust(lnLang ~ absLat + sdElev + sdSuitable + avgElev + avgSuitable +
                         avgPrecip + avgTemp +  lnArea + seaDist + waterArea +
                         withinCountry +  numCountry +  migrationDist +  lnPopDens95,
                         data2 %>% filter((data2$suitableCells == 25) &
                                          (data2$totalPop95 >= 3000)),
                         fixed_effects = ~countryCode,
                         se_type = "stata")
data222 <- data2 %>% filter((data2$suitableCells >= 10) & (data2$totalPop95 >= 3000))
sizePercentile <- quantile(data222$area, probs = seq(0, 1, length.out = 101))
data222[, "sizePerc"] <- factor(cut(data222$area, sizePercentile))

# NOTE: need to put the size percentile in manually, else doesn't work
robust2.2.2 <- lm_robust(lnLang ~ absLat + sdElev + sdSuitable + avgElev + avgSuitable +
                         avgPrecip + avgTemp + lnArea + seaDist + waterArea +
                         withinCountry +  numCountry +  migrationDist +
```

```
                        lnPopDens95 + sizePerc,
                        data = data222,
                        # fixed_effects = ~ c(countryCode, sizePerc),
                        fixed_effects = ~ countryCode,
                        se_type = "stata")
robust2.2.3 <- lm_robust(lnLang ~ absLat + dispElev + dispSuitable + avgElev + avgSuitable +
                        avgPrecip + avgTemp +  lnArea + seaDist + waterArea +
                        withinCountry +  numCountry +  migrationDist +  lnPopDens95,
                        data2 %>% filter((data2$suitableCells >= 10) &
                                         (data2$totalPop95 >= 3000)),
                        fixed_effects = ~countryCode,
                        se_type = "stata")
robust2.2.4 <- lm_robust(lnLang ~ absLat + sdElev + sdClimate + avgElev + climate +  avgPrecip +
                        avgTemp +  lnArea + seaDist + waterArea +
                        withinCountry +  numCountry +  migrationDist +  lnPopDens95,
                        data2 %>% filter((data2$suitableCells >= 10) &
                                         (data2$totalPop95 >= 3000)),
                        fixed_effects = ~countryCode,
                        se_type = "stata")
robust2.2.5 <- lm_robust(lnLang ~ absLat + sdElev + sdSoil + avgElev + soil +  avgPrecip +
                        avgTemp +  lnArea + seaDist + waterArea +  withinCountry +
                        numCountry +  migrationDist +  lnPopDens95,
                        data2 %>% filter((data2$suitableCells >= 10) &
                                           (data2$totalPop95 >= 3000)),
                        fixed_effects = ~countryCode,
                        se_type = "stata")
models <- paste0("robust2.2.", 1:5)
coeffs <- c(lapply(models, function(model) { get(model)[["coefficients"]]}) )
names(coeffs) <- models

vars <- unique(unlist(lapply(coeffs, names)))

coeffDf <- data.frame(row.names = vars)
for(model in models) coeffDf[names(coeffs[[model]]), model] <- coeffs[[model]]

ses <- sapply(models, function(model) get(model)["std.error"])
names(ses) <- models
sesDf <- data.frame(row.names = vars)
for(model in models) sesDf[names(ses[[model]]), model] <- ses[[model]]

pvals <- sapply(models, function(model) get(model)["p.value"])
names(pvals) <- models
pvalsDf <- data.frame(row.names = vars)
for(model in models) pvalsDf[names(pvals[[model]]), model] <- pvals[[model]]

order <- c('sdElev', 'sdSuitable', 'dispElev', 'dispSuitable', 'sdClimate',
           'climate', 'sdSoil', 'soil')

coeffDf <- coeffDf[order, ]
sesDf   <- sesDf[order, ]
```

```
pvalsDf <- pvalsDf[order, ]

# interleave rows
coeffDf[, "stat"] <- "Estimate"
sesDf[, "stat"] <- "SE"
tbl5b <- gdata::interleave(coeffDf, sesDf)
# ensure a hard copy
tbl5bformat <- data.frame(tbl5b)

for (model in models) {
  estimRows <- !is.na(tbl5bformat[, model]) & (tbl5bformat$stat == 'Estimate')
  seRows <- !is.na(tbl5bformat[, model]) & (tbl5bformat$stat == 'SE')

  tbl5bformat[estimRows, model] <- sprintf(fmt = "%.3f", tbl5bformat[estimRows, model] %>%
                                             unlist() %>% as.numeric())
  tbl5bformat[seRows, model] <- paste0("(", sprintf(fmt = "%.3f", tbl5bformat[seRows, model]
                                             %>% unlist()
                                             %>% as.numeric()), ")")

  significant <- rep(pvalsDf[, model] < .01, each = 2)
  significant[is.na(significant)] <- FALSE

  tbl5bformat[estimRows, model] <- cell_spec(tbl5bformat[estimRows, model],
                                             italic = significant[estimRows])
  tbl5bformat[seRows, model] <- cell_spec(tbl5bformat[seRows, model],
                                          italic = significant[seRows])
}

tbl5bformat$name <- c('Variation in elevation', NA, 'Variation in land quality', NA,
                      'Dispersion of elevation', NA, 'Dispersion of land quality', NA,
                      'Variation in climatic suitability', NA, 'Mean climatic suitability', NA,
                      'Variation in soil suitability', NA, 'Mean soil suitability', NA)

col.names <- c("Variable", paste0("(", 1:5, ")"))

tbl5bformat %>%
  select(name, 1:5, -stat) %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE, linesep = c("", "\\addlinespace"),
               col.names = col.names,
               row.names = F,
               align = "r", escape = F,
               caption = "Table 5B-Robustness Checks for the Virtual Country Analysis. Italics indicate significa
  row_spec(seq(2, nrow(tbl5bformat), 2), font_size = 8)

tbl5binfo <- data.frame(
  model = 1:5,
  cont = rep("Yes", 5),
  nobs = sapply(models, function(model) { nobs(get(model))}),
  rsq = lapply(models, function(model) {
    formatC(
```

```
      summary(get(model))$r.squared
      , digits = 2, format = 'f')
    }) %>% unlist,
  row.names = NULL
  )

tbl5binfo %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE,
               col.names = c('Model', 'Country Indicators', 'Observations', '$R^2$'),
               escape = F,
               caption = "Information for each model in Table 5B: cross-country robustness check")
```

8.1 GREG preprocessing

greg_nmbrlang

May 12, 2022

1 Appendix Part 2: GREG Database Wrangling

This notebook is included separately, because it contains the code used to transform the GREG dataset into values in a suitable format to swap in for the withheld WLMS data.

[1]:
```
import pandas as pd
import numpy as np
import geopandas as gpd

import matplotlib.pyplot as plt
import seaborn as sns
```

```
/opt/homebrew/lib/python3.9/site-packages/geopandas/_compat.py:111: UserWarning:
The Shapely GEOS version (3.10.2-CAPI-1.16.0) is incompatible with the GEOS
version PyGEOS was compiled with (3.10.1-CAPI-1.16.0). Conversions between both
will be slow.
  warnings.warn(
```

1.1 Import Shapefiles

[2]:
```
virtual = gpd.read_file('data_raw/Virtual_country')
virtual[['uniq_cnt25', 'point5_id', 'geometry']].head()
```

[2]:

	uniq_cnt25	point5_id	geometry
0	39	247867.0	POLYGON ((-86.00000 82.00000, -86.50000 82.000…
1	40	247868.0	POLYGON ((-85.50000 82.00000, -86.00000 82.000…
2	40	247869.0	POLYGON ((-85.00000 82.00000, -85.50000 82.000…
3	40	247870.0	POLYGON ((-84.50000 82.00000, -85.00000 82.000…
4	40	247871.0	POLYGON ((-84.00000 82.00000, -84.50000 82.000…

[3]:
```
greg = gpd.read_file('greg')
greg[['G1SHORTNAM', 'G2SHORTNAM', 'G3SHORTNAM', 'geometry']].head()
```

[3]:

	G1SHORTNAM	G2SHORTNAM	G3SHORTNAM	\
0	Curaçao Islanders	None	None	
1	English-speaking population of the Lesser Anti…	None	None	
2	Baloch	Brahui	None	
3	Persians	Afghans	None	

```
4                                          Afghans     Tajiks       None

                                                geometry
0  POLYGON ((-69.88223 12.41111, -69.94695 12.436…
1  MULTIPOLYGON (((-61.73889 17.54055, -61.75195 …
2  POLYGON ((64.03937 30.02453, 64.03937 30.11267…
3  POLYGON ((61.75456 30.78628, 61.75833 30.79028…
4  POLYGON ((61.62285 31.39536, 61.64841 31.46713…
```

Before proceeding must check that the two shapefiles follow the same coordinate reference system, in this case, WGS84.

[4]:
```
virtual.crs == greg.crs
```

[4]:
```
True
```

1.2 Transform GREG

The original GREG format is a number of regions, each of which has up to three ethnic groups attached to it. Ethnic groups may also be attached to different regions. This code chunk melts, and then dissolves, the original `greg` dataset, such that we have one entry per ethnic group.

[5]:
```
melted = pd.melt(greg, id_vars = ['geometry'], value_vars = ['G1SHORTNAM',␣
 ↪'G2SHORTNAM', 'G3SHORTNAM'], value_name = 'SHORTNAM')
ethnicGroups = melted[melted['SHORTNAM'].notna()].drop('variable', axis = 1).
 ↪dissolve(by = 'SHORTNAM', aggfunc = 'first', as_index = False)
ethnicGroups
```

[5]:
```
        SHORTNAM                                           geometry
0     Abazinians  MULTIPOLYGON (((41.83519 44.08370, 41.86445 44…
1        Abkhaz  MULTIPOLYGON (((41.73878 42.62086, 41.71329 42…
2      Achaguas  MULTIPOLYGON (((-74.02123 2.16973, -73.98634 2…
3        Achang  POLYGON ((97.84312 24.33767, 97.84467 24.36087…
4      Achinese  MULTIPOLYGON (((97.81446 2.77691, 97.86672 2.7…
..          …                                                  …
923      Zagawa  MULTIPOLYGON (((25.88538 14.53904, 25.83321 14…
924   Zakhchins  POLYGON ((91.54557 47.36713, 91.54557 47.43751…
925    Zapotecs  POLYGON ((-94.96082 16.37316, -95.03084 16.322…
926       Zoque  MULTIPOLYGON (((-93.18895 16.87464, -93.13737 …
927       Zulus  POLYGON ((31.11778 -29.67945, 31.00972 -29.872…

[928 rows x 2 columns]
```

1.3 Perform Intersection

This cell intersects the imported dataset of cells with the dataset of ethnic groups, derived from GREG.

[6]:
```
joined = gpd.overlay(virtual, ethnicGroups, how = 'intersection')
joined.head()
```

[6]:

	uniq_cnt25	point5_id	pop95	maize	pasture	suit_new	sorghum	allcrops	\
0	211	247281.0	0.0301	0.0	0.0	0.0000	0.0	0.0	
1	211	247282.0	0.0300	0.0	0.0	0.0000	0.0	0.0	
2	335	241416.0	0.0271	0.0	0.0	0.0001	0.0	0.0	
3	335	242134.0	0.0195	0.0	0.0	0.0001	0.0	0.0	
4	335	242135.0	0.0330	0.0	0.0	0.0001	0.0	0.0	

	SHORTNAM	geometry
0	Eskimos	MULTIPOLYGON (((-19.00000 81.71801, -19.14417 …
1	Eskimos	MULTIPOLYGON (((-19.00000 81.80707, -18.99083 …
2	Eskimos	POLYGON ((-72.00000 78.00000, -71.87679 78.000…
3	Eskimos	MULTIPOLYGON (((-73.00000 78.17449, -72.99834 …
4	Eskimos	POLYGON ((-72.34038 78.00000, -72.34695 78.003…

1.4 Coverage

These cells reduce each virtual country to *only contain cells in which the cell is completely covered by an ethnic group from GREG*, similar to our interpretation of the procedure described in Michalopolous.

First, we calculate the "area" of each small cell after it has been intersected with the transformed GREG dataset. Then we compare this area to the area of the full cell, and equaivalent areas indicate that the cell is completely covered.

[7]:
```
dissolved = joined[['point5_id', 'geometry']].dissolve('point5_id')
areasCell = dissolved.area.to_frame().rename(columns = {0: 'overlay'})
areasCell['full'] = virtual.set_index('point5_id').area
areasCell['complete'] = np.isclose(areasCell['overlay'], areasCell['full'])
areasCell
```

```
/opt/homebrew/lib/python3.9/site-packages/pygeos/set_operations.py:388:
RuntimeWarning: divide by zero encountered in unary_union
  result = lib.unary_union(collections, **kwargs)
/var/folders/l7/_yl1rg512jv095gql7v0v5r00000gn/T/ipykernel_50548/310672282.py:2:
UserWarning: Geometry is in a geographic CRS. Results from 'area' are likely
incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to a projected CRS
before this operation.

  areasCell = dissolved.area.to_frame().rename(columns = {0: 'overlay'})
/var/folders/l7/_yl1rg512jv095gql7v0v5r00000gn/T/ipykernel_50548/310672282.py:3:
UserWarning: Geometry is in a geographic CRS. Results from 'area' are likely
incorrect. Use 'GeoSeries.to_crs()' to re-project geometries to a projected CRS
before this operation.

  areasCell['full'] = virtual.set_index('point5_id').area
```

[7]:

point5_id	overlay	full	complete
49903.0	0.040962	0.25	False
49904.0	0.111268	0.25	False
49905.0	0.112302	0.25	False
49906.0	0.006557	0.25	False
50621.0	0.006873	0.25	False
…	…	…	…
242136.0	0.185729	0.25	False
242137.0	0.013326	0.25	False
242857.0	0.033278	0.25	False
247281.0	0.013842	0.25	False
247282.0	0.060065	0.25	False

[58073 rows x 3 columns]

This cell merges the overlay dataset calculated earlier with the coverage dataset, to determine whether each cell-ethnic group combination is of a cell with complete coverage.

[8]:
```
joinedCoverage = joined.merge(areasCell, left_on = 'point5_id', right_index =␣
↪True)
joinedCoverage[['uniq_cnt25', 'point5_id', 'complete']].head()
```

[8]:

	uniq_cnt25	point5_id	complete
0	211	247281.0	False
1	211	247282.0	False
2	335	241416.0	False
3	335	242134.0	False
4	335	242135.0	False

Finally, we group by the virtual country ID, and count the number of unique ethnic groups (SHORTNAM), along with the number of complete cells (point5_id).

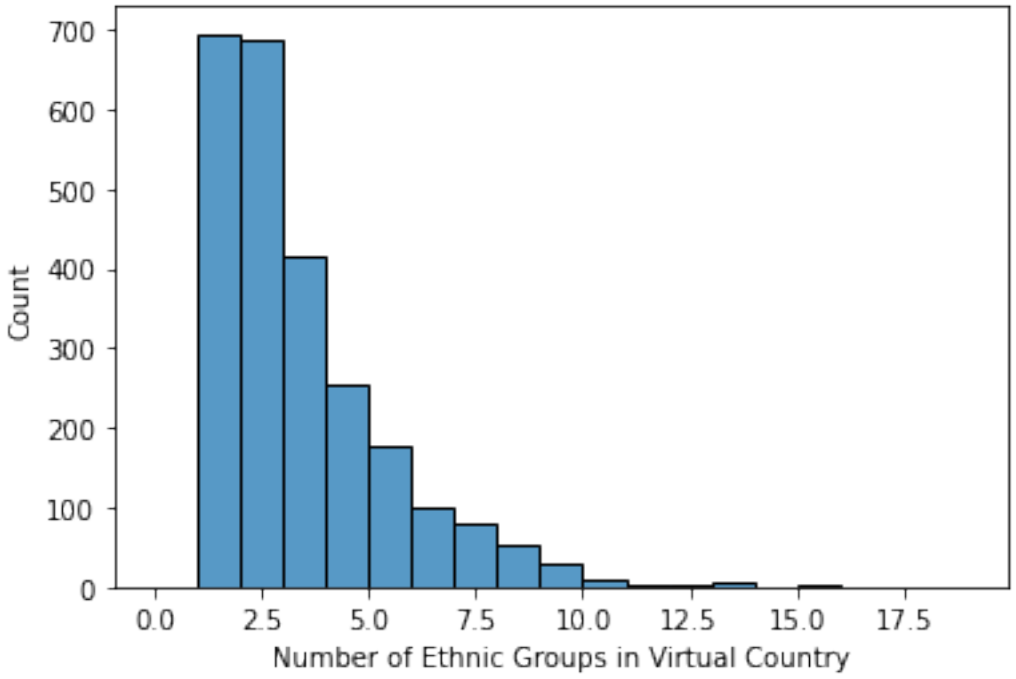
[9]:
```
countries = joinedCoverage[joinedCoverage['complete'] == True].
↪groupby('uniq_cnt25')[['point5_id', 'SHORTNAM']].nunique()
countries.to_csv('greg.csv')
```

[10]:
```
len(countries)
```

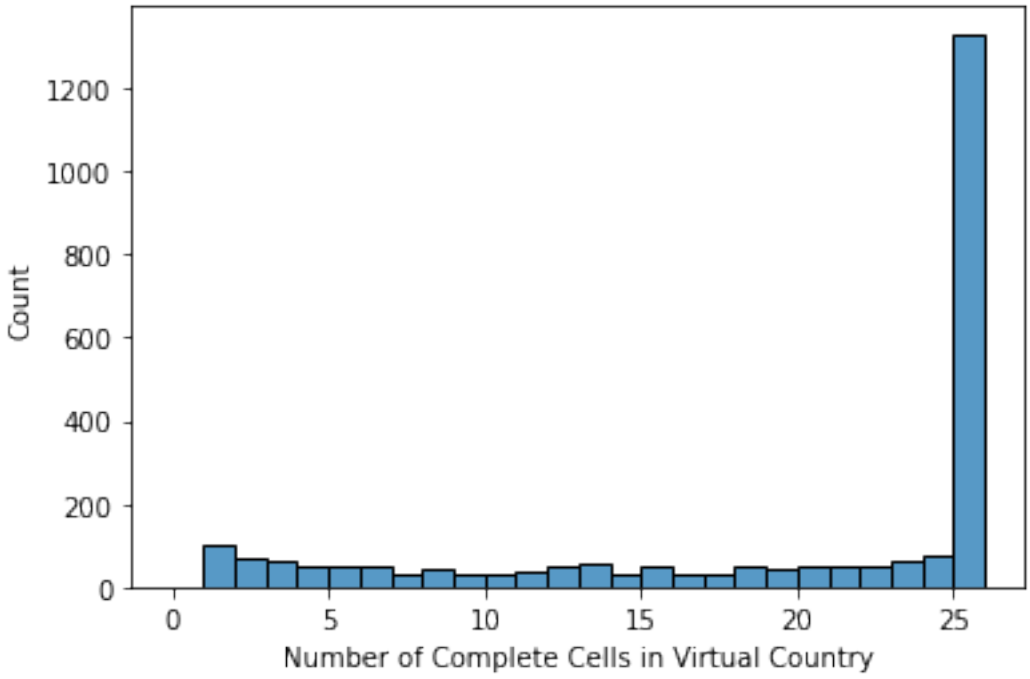
[10]:
```
2521
```

[11]:
```
ax = sns.histplot(x = countries['SHORTNAM'], bins = np.arange(0, 20))
ax.set_xlabel('Number of Ethnic Groups in Virtual Country')
```

[11]:
```
Text(0.5, 0, 'Number of Ethnic Groups in Virtual Country')
```



[13]:
```
ax = sns.histplot(x = countries['point5_id'], bins = np.arange(27))
ax.set_xlabel('Number of Complete Cells in Virtual Country');
```



1.5 Comparison to WLMS

When calculating number of ethnic groups per virtual country, we obtained 2521 countries with full coverage in at least one of its 25 cells. 1857 of these countries are included in the dataset derived from WLMS provided in the data download. 31 of the countries included in the data downloaded are *not* included in the 2521 countries we obtained.

[14]:
```
df = pd.read_stata('data_raw/Tables4-7b.dta')
df['uniq_cnt25'] = df['uniq_cnt25'].astype(int)
```

[15]:
```
len(df)
```

[15]: 1888

[16]:
```
countries.index.isin(df['uniq_cnt25']).sum()
```

[16]: 1857

[17]:
```
df['uniq_cnt25'].isin(countries.index).sum()
```

[17]: 1857