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

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

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, andgeographic 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.

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.

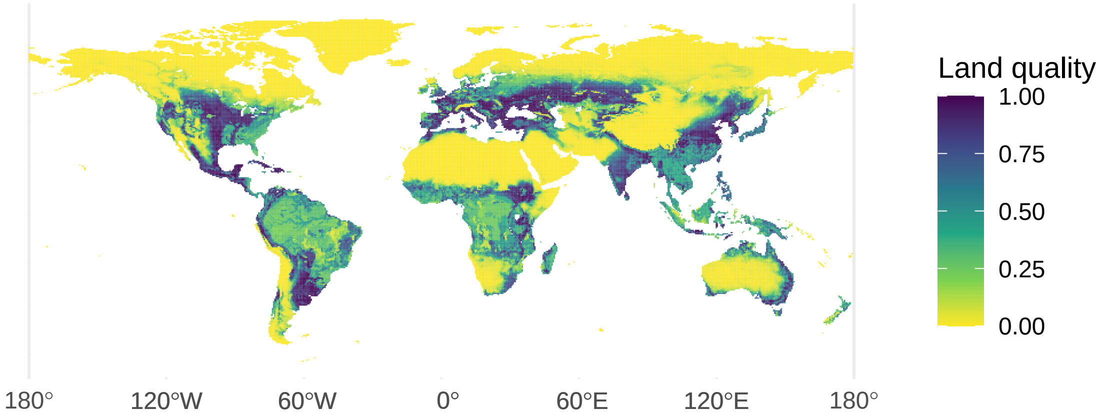

Figure 1: Land quality for agriculture across countries

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.

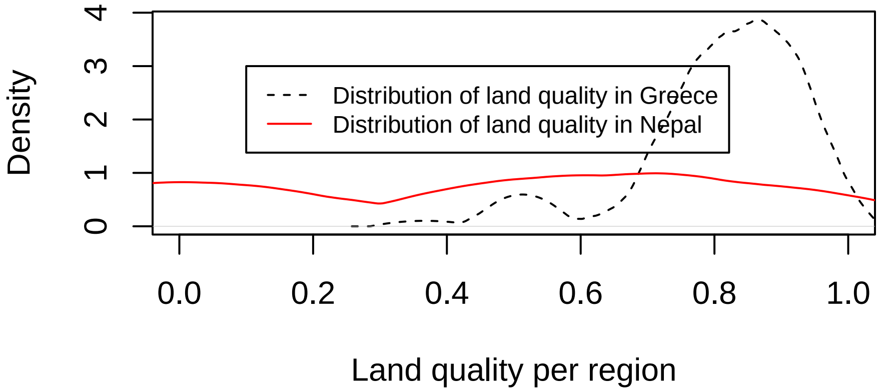

Figure 2: Kernel density of land quality in Greece and Nepal

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 variablity of the geographic data, the log of the country's 1995 population, human migration distance from Africa, and distance from a large body of water. While some other variables in the provided dataset have missing values for some countries, 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:

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

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

Table 2: Main specification for the cross-country analysis. Italics indicate significance at the 99% level.

	(1) Coef.	(1) S.E.	(2) Coef.	(2) S.E.	(3) Coef.	(3) S.E.	(4) Coef.	(4) S.E.	(5) Coef.	(5) S.E.
sdElev			*0.310*	*0.113*	*0.256*	*0.079*	*0.291*	*0.089*	*0.275*	*0.101*
sdSuitable			*0.340*	*0.084*	*0.177*	*0.061*	*0.208*	*0.058*	*0.211*	*0.060*
avgElev			-0.249	0.113	-0.111	0.106	-0.104	0.118	-0.085	0.113
avgSuitable			-0.179	0.069	-0.069	0.065	-0.029	0.068	0.006	0.064
absLat	*-0.479*	*0.070*	*-0.547*	*0.061*	-0.058	0.192	-0.033	0.214	-0.131	0.201
avgPrecip					*0.468*	*0.086*	*0.447*	*0.088*	*0.479*	*0.088*
avgTemp					0.270	0.197	0.385	0.213	0.404	0.183
lnArea					*0.517*	*0.067*	*0.482*	*0.073*	*0.464*	*0.074*
seaDist					0.053	0.065	0.063	0.062	0.073	0.064
migrationDist					*-0.281*	*0.063*	-0.518	0.199	-0.513	0.218
lnPopDens1995							-0.118	0.087	0.023	0.072
lnPopDens1500									-0.235	0.105
entryYear									-0.108	0.066
agriTran									0.134	0.094

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 X_i.

2.1 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. Unsurprisingly, given the increasing number of features, the observed R^2 also increases with each model. 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. Mean precipitation, log of area, and migratory distance from Ethiopia were also significant predictors whenever they were included.

2.1.1 Critique of Assumptions

2.2 Robustness Check

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

Model	Continental Indicators	Observations	R-Squared
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

2.2.1 Table 2A

```
##
## Call:
## glm.nb(formula = numLang ~ absLat + sdSuitable + sdElev + avgElev +
##     avgSuitable + avgPrecip + avgTemp + lnArea + seaDist + migrationDist +
##     lnPopDens1995 + lnPopDens1500 + entryYear + agriTran + africa +
##     europe + americas + asiaPac, data = data, na.action = na.exclude,
##     init.theta = 1.934771646, link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2512  -1.0161  -0.4081   0.4434   3.5187
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    6.548728   4.046319   1.618 0.105568
## absLat        -0.005966   0.017186  -0.347 0.728507
## sdSuitable     3.252227   0.967268   3.362 0.000773 ***
## sdElev         1.313302   0.413671   3.175 0.001500 **
## avgElev       -0.418495   0.328995  -1.272 0.203359
## avgSuitable    0.373707   0.401258   0.931 0.351679
## avgPrecip      0.010091   0.002181   4.626 3.73e-06 ***
## avgTemp        0.078466   0.030102   2.607 0.009143 **
## lnArea         0.528278   0.073553   7.182 6.86e-13 ***
## seaDist        0.253463   0.256217   0.989 0.322541
## migrationDist -0.157628   0.045867  -3.437 0.000589 ***
## lnPopDens1995  0.041208   0.093715   0.440 0.660141
## lnPopDens1500 -0.110614   0.095104  -1.163 0.244794
## entryYear     -0.003063   0.001842  -1.662 0.096436 .
## agriTran      -0.016956   0.063494  -0.267 0.789428
## africa        -0.308054   0.459205  -0.671 0.502322
## europe         0.130597   0.374891   0.348 0.727570
## americas       1.050666   0.678607   1.548 0.121558
## asiaPac        0.719299   0.382401   1.881 0.059971 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.9348) family taken to be 1)
##
##     Null deviance: 563.15  on 141  degrees of freedom
## Residual deviance: 144.04  on 123  degrees of freedom
##   (14 observations deleted due to missingness)
## AIC: 1112.7
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.935
##           Std. Err.:  0.250
##
##  2 x log-likelihood:  -1072.729
```

NOTE: Matches!

```
##
## Call:
## lm(formula = lnLang ~ absLat + dispElev + dispSuitable + avgElev +
##     avgSuitable + avgPrecip + avgTemp + lnArea + seaDist + migrationDist +
##     lnPopDens1995 + lnPopDens1500 + entryYear + agriTran + africa +
##     europe + americas + asiaPac, data = data, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.95035 -0.55786 -0.03237  0.52694  2.61099
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.852240   4.263804   1.373 0.172393
## absLat        -0.009741   0.019113  -0.510 0.611195
## dispElev       0.428598   0.134333   3.191 0.001802 **
## dispSuitable   1.314678   0.384770   3.417 0.000859 ***
## avgElev       -0.233907   0.326719  -0.716 0.475394
## avgSuitable   -0.046338   0.422776  -0.110 0.912901
## avgPrecip      0.010779   0.002425   4.445 1.94e-05 ***
## avgTemp        0.081696   0.033444   2.443 0.015995 *
## lnArea         0.351279   0.094939   3.700 0.000324 ***
## seaDist        0.269178   0.273772   0.983 0.327429
## migrationDist -0.126082   0.048136  -2.619 0.009920 **
## lnPopDens1995  0.059685   0.103261   0.578 0.564316
## lnPopDens1500 -0.226042   0.100970  -2.239 0.026973 *
## entryYear     -0.003279   0.001926  -1.702 0.091224 .
## agriTran       0.065991   0.067525   0.977 0.330345
## africa        -0.118792   0.497360  -0.239 0.811624
## europe         0.597281   0.403522   1.480 0.141385
## americas       0.375154   0.725772   0.517 0.606152
## asiaPac        0.457586   0.431350   1.061 0.290849
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8533 on 123 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.7323, Adjusted R-squared:  0.6931
## F-statistic: 18.69 on 18 and 123 DF,  p-value: < 2.2e-16
```

NOTE: matches or exceeds in significance, as well

```
##
## Call:
## lm(formula = lnLang ~ absLat + sdElev + sdClimate + avgElev +
##     climate + avgPrecip + avgTemp + lnArea + seaDist + migrationDist +
##     lnPopDens1995 + lnPopDens1500 + entryYear + agriTran + africa +
##     europe + americas + asiaPac, data = data, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9917 -0.5666 -0.0940  0.5144  2.7451
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.940308   4.284013   1.387  0.16806
## absLat        -0.011158   0.020088  -0.555  0.57959
## sdElev         1.045156   0.460917   2.268  0.02510 *
## sdClimate      2.505066   0.806127   3.108  0.00234 **
## avgElev       -0.230713   0.348637  -0.662  0.50936
## climate        0.660906   0.382811   1.726  0.08678 .
## avgPrecip      0.012137   0.002521   4.814 4.26e-06 ***
## avgTemp        0.078744   0.034420   2.288  0.02386 *
## lnArea         0.498980   0.078574   6.350 3.76e-09 ***
## seaDist        0.373275   0.274461   1.360  0.17631
## migrationDist -0.120027   0.048952  -2.452  0.01561 *
## lnPopDens1995  0.031198   0.101433   0.308  0.75893
## lnPopDens1500 -0.246419   0.102847  -2.396  0.01808 *
## entryYear     -0.003376   0.001917  -1.761  0.08077 .
## agriTran       0.095310   0.067456   1.413  0.16020
## africa        -0.162473   0.501039  -0.324  0.74628
## europe         0.624611   0.408883   1.528  0.12918
## americas       0.205573   0.728266   0.282  0.77821
## asiaPac        0.451329   0.428861   1.052  0.29468
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8583 on 123 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.7291, Adjusted R-squared:  0.6894
## F-statistic: 18.39 on 18 and 123 DF,  p-value: < 2.2e-16
```

NOTE: matches!

```
##
## Call:
## lm(formula = lnLang ~ absLat + sdElev + sdSoil + avgElev + soil +
##     avgPrecip + avgTemp + lnArea + seaDist + migrationDist +
##     lnPopDens1995 + lnPopDens1500 + entryYear + agriTran + africa +
##     europe + americas + asiaPac, data = data1.4, na.action = na.exclude)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.87797 -0.50870 -0.04869  0.49590  2.85469
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.324986   4.326770   1.000  0.31947
## absLat        -0.012498   0.019370  -0.645  0.52000
## sdElev         1.308730   0.439485   2.978  0.00350 **
## sdSoil         3.785250   1.349989   2.804  0.00587 **
## avgElev       -0.168396   0.353490  -0.476  0.63465
## soil           0.653355   0.499626   1.308  0.19342
## avgPrecip      0.012728   0.002396   5.313 4.88e-07 ***
## avgTemp        0.076052   0.033807   2.250  0.02625 *
## lnArea         0.502703   0.079264   6.342 3.92e-09 ***
## seaDist        0.250308   0.282993   0.885  0.37815
## migrationDist -0.112495   0.049314  -2.281  0.02426 *
## lnPopDens1995  0.061876   0.101543   0.609  0.54341
## lnPopDens1500 -0.244267   0.103564  -2.359  0.01992 *
## entryYear     -0.002706   0.001948  -1.389  0.16737
## agriTran       0.100822   0.068029   1.482  0.14089
## africa        -0.002027   0.497823  -0.004  0.99676
## europe         0.608691   0.403548   1.508  0.13403
## americas       0.117636   0.740684   0.159  0.87407
## asiaPac        0.412763   0.433711   0.952  0.34312
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.863 on 123 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.7261, Adjusted R-squared:  0.6861
## F-statistic: 18.12 on 18 and 123 DF,  p-value: < 2.2e-16
```

NOTE: matches!

2.2.2 Table 2B

```
TODO: write up a bit about the datasets used here
```

```
##
## Call:
## lm(formula = elf ~ abs_lat, data = data2bStd, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8470 -0.7628  0.1310  0.7781  1.9440
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.940e-17  7.800e-02   0.000        1
## abs_lat     -3.690e-01  7.827e-02  -4.714 5.76e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.9327 on 141 degrees of freedom
## Multiple R-squared:  0.1362, Adjusted R-squared:   0.13
## F-statistic: 22.22 on 1 and 141 DF,  p-value: 5.764e-06
```

OK

```
##
## Call:
## lm(formula = elf ~ abs_lat + sd_emean + sd_climsuit + emean +
##     mean_climsuit, data = data2bStd, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.87650 -0.78488  0.01382  0.70933  2.08944
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.415e-16  7.605e-02   0.000  1.00000
## abs_lat       -4.337e-01  8.338e-02  -5.202 7.04e-07 ***
## sd_emean      -1.105e-01  1.199e-01  -0.922  0.35815
## sd_climsuit    2.938e-01  1.053e-01   2.789  0.00603 **
## emean          9.261e-02  1.113e-01   0.832  0.40670
## mean_climsuit  5.326e-02  8.675e-02   0.614  0.54024
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9094 on 137 degrees of freedom
## Multiple R-squared:  0.2021, Adjusted R-squared:  0.173
## F-statistic: 6.942 on 5 and 137 DF,  p-value: 8.292e-06
```

OK

```
##
## Call:
## lm(formula = elf ~ abs_lat + sd_emean + sd_climsuit + emean +
##     mean_climsuit + precav + tempav + lnareakm2 + distc + migdist +
##     lnpop95 + lpd1500 + yrentry + agritran + africa + europe +
##     americas + reg_eap, data = data2bStd, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.26948 -0.35908 -0.03879  0.46489  1.64637
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -9.380e-17  6.163e-02   0.000  1.00000
## abs_lat       -3.968e-01  2.931e-01  -1.354  0.17830
## sd_emean       3.630e-01  1.339e-01   2.712  0.00765 **
## sd_climsuit    2.309e-01  9.454e-02   2.442  0.01601 *
## emean         -3.009e-01  1.386e-01  -2.171  0.03187 *
## mean_climsuit  2.179e-01  1.093e-01   1.994  0.04834 *
## precav         1.798e-01  1.327e-01   1.355  0.17795
## tempav        -2.990e-02  2.454e-01  -0.122  0.90321
## lnareakm2      3.036e-02  1.129e-01   0.269  0.78852
## distc          2.815e-01  9.610e-02   2.929  0.00405 **
## migdist       -1.220e-01  2.781e-01  -0.439  0.66176
## lnpop95       -2.242e-02  1.208e-01  -0.186  0.85306
## lpd1500       -2.685e-01  1.343e-01  -1.999  0.04777 *
## yrentry        1.463e-01  9.602e-02   1.524  0.13002
## agritran      -7.968e-02  1.389e-01  -0.574  0.56726
## africa         8.347e-02  2.013e-01   0.415  0.67912
## europe         1.385e-01  1.490e-01   0.930  0.35441
## americas      -3.633e-01  2.268e-01  -1.602  0.11168
## reg_eap       -1.264e-01  1.157e-01  -1.093  0.27670
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.737 on 124 degrees of freedom
## Multiple R-squared:  0.5257, Adjusted R-squared:  0.4569
## F-statistic: 7.635 on 18 and 124 DF,  p-value: 5.106e-13
```

OK

```
##
## Call:
## lm(formula = elf3 ~ abs_lat + sd_emean + sd_climsuit + emean +
##     mean_climsuit + precav + tempav + lnareakm2 + distc + migdist +
##     lnpop95 + lpd1500 + yrentry + agritran + africa + europe +
##     americas + reg_eap, data = data2bStd, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5888 -0.5884 -0.1338  0.5310  2.0736
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.852e-16  7.373e-02   0.000  1.00000
## abs_lat       -1.156e-01  3.506e-01  -0.330  0.74220
## sd_emean       3.559e-01  1.601e-01   2.223  0.02804 *
## sd_climsuit    2.914e-01  1.131e-01   2.576  0.01116 *
## emean         -3.521e-01  1.658e-01  -2.123  0.03573 *
## mean_climsuit -1.414e-01  1.307e-01  -1.082  0.28146
## precav         4.553e-01  1.588e-01   2.868  0.00486 **
## tempav         2.476e-01  2.936e-01   0.843  0.40071
## lnareakm2     -2.473e-01  1.351e-01  -1.830  0.06963 .
## distc          4.520e-01  1.150e-01   3.931  0.00014 ***
## migdist       -5.348e-01  3.327e-01  -1.607  0.11051
## lnpop95       -9.337e-02  1.445e-01  -0.646  0.51945
## lpd1500       -2.393e-01  1.607e-01  -1.490  0.13886
## yrentry        5.755e-02  1.149e-01   0.501  0.61722
## agritran       1.545e-01  1.662e-01   0.930  0.35437
## africa        -1.706e-01  2.408e-01  -0.708  0.48006
## europe         1.397e-01  1.782e-01   0.784  0.43448
## americas       8.734e-02  2.713e-01   0.322  0.74805
## reg_eap        1.381e-01  1.384e-01   0.998  0.32021
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8816 on 124 degrees of freedom
## Multiple R-squared:  0.3212, Adjusted R-squared:  0.2227
## F-statistic:  3.26 on 18 and 124 DF,  p-value: 5.303e-05
```

OK

```
##
## Call:
## lm(formula = elf5 ~ abs_lat + sd_emean + sd_climsuit + emean +
##     mean_climsuit + precav + tempav + lnareakm2 + distc + migdist +
##     lnpop95 + lpd1500 + yrentry + agritran + africa + europe +
##     americas + reg_eap, data = data2bStd, na.action = na.exclude)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6182 -0.5485 -0.1193  0.6637  1.8739
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.995e-16  7.210e-02   0.000 1.000000
## abs_lat       -1.851e-01  3.429e-01  -0.540 0.590204
## sd_emean       4.723e-01  1.566e-01   3.016 0.003104 **
## sd_climsuit    2.928e-01  1.106e-01   2.647 0.009165 **
## emean         -4.749e-01  1.622e-01  -2.928 0.004055 **
## mean_climsuit -6.188e-02  1.279e-01  -0.484 0.629229
## precav         4.043e-01  1.553e-01   2.604 0.010330 *
## tempav         1.807e-01  2.871e-01   0.629 0.530190
## lnareakm2     -1.862e-01  1.321e-01  -1.409 0.161262
## distc          4.138e-01  1.124e-01   3.681 0.000346 ***
## migdist       -2.804e-01  3.254e-01  -0.862 0.390511
## lnpop95        5.586e-03  1.413e-01   0.040 0.968537
## lpd1500       -2.137e-01  1.571e-01  -1.360 0.176243
## yrentry        1.107e-01  1.123e-01   0.985 0.326432
## agritran       1.956e-01  1.625e-01   1.203 0.231118
## africa        -8.903e-04  2.355e-01  -0.004 0.996990
## europe         1.646e-01  1.743e-01   0.945 0.346619
## americas      -1.426e-01  2.653e-01  -0.537 0.591952
## reg_eap        3.300e-02  1.353e-01   0.244 0.807742
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8622 on 124 degrees of freedom
## Multiple R-squared:  0.3509, Adjusted R-squared:  0.2567
## F-statistic: 3.724 on 18 and 124 DF,  p-value: 6.471e-06
```

OK

```
##
## Call:
## lm(formula = elf7 ~ abs_lat + sd_emean + sd_climsuit + emean +
##     mean_climsuit + precav + tempav + lnareakm2 + distc + migdist +
##     lnpop95 + lpd1500 + yrentry + agritran + africa + europe +
##     americas + reg_eap, data = data2bStd, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.60444 -0.50238  0.02369  0.57650  1.80411
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.751e-16  6.706e-02   0.000 1.000000
## abs_lat       -6.412e-02  3.189e-01  -0.201 0.840982
## sd_emean       4.260e-01  1.456e-01   2.925 0.004093 **
## sd_climsuit    2.154e-01  1.029e-01   2.094 0.038278 *
## emean         -4.620e-01  1.508e-01  -3.063 0.002688 **
## mean_climsuit -2.134e-01  1.189e-01  -1.795 0.075131 .
## precav         4.871e-01  1.444e-01   3.373 0.000991 ***
## tempav         3.158e-01  2.670e-01   1.183 0.239188
## lnareakm2     -1.745e-01  1.229e-01  -1.420 0.158115
## distc          3.264e-01  1.046e-01   3.121 0.002240 **
## migdist       -3.594e-01  3.026e-01  -1.188 0.237242
## lnpop95        7.032e-03  1.314e-01   0.053 0.957421
## lpd1500       -2.111e-01  1.461e-01  -1.445 0.151068
## yrentry        6.106e-02  1.045e-01   0.585 0.559945
## agritran       3.381e-01  1.511e-01   2.237 0.027073 *
## africa         9.670e-03  2.190e-01   0.044 0.964855
## europe         1.604e-02  1.621e-01   0.099 0.921321
## americas      -1.658e-01  2.468e-01  -0.672 0.502771
## reg_eap       -2.955e-03  1.259e-01  -0.023 0.981309
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8019 on 124 degrees of freedom
## Multiple R-squared:  0.4385, Adjusted R-squared:  0.357
## F-statistic:  5.38 on 18 and 124 DF,  p-value: 4.415e-09
```

OK

```
##
## Call:
## lm(formula = elf9 ~ abs_lat + sd_emean + sd_climsuit + emean +
##     mean_climsuit + precav + tempav + lnareakm2 + distc + migdist +
##     lnpop95 + lpd1500 + yrentry + agritran + africa + europe +
##     americas + reg_eap, data = data2bStd, na.action = na.exclude)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.66758 -0.54364  0.05585  0.54828  1.74016
```

```
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.912e-16  6.471e-02   0.000  1.00000
## abs_lat       -1.244e-01  3.078e-01  -0.404  0.68667
## sd_emean       4.133e-01  1.406e-01   2.941  0.00391 **
## sd_climsuit    1.561e-01  9.928e-02   1.572  0.11839
## emean         -3.674e-01  1.456e-01  -2.524  0.01286 *
## mean_climsuit -2.386e-02  1.148e-01  -0.208  0.83562
## precav         3.755e-01  1.394e-01   2.694  0.00804 **
## tempav         3.019e-01  2.577e-01   1.172  0.24354
## lnareakm2     -1.462e-02  1.186e-01  -0.123  0.90207
## distc          2.290e-01  1.009e-01   2.269  0.02497 *
## migdist       -5.396e-01  2.921e-01  -1.847  0.06706 .
## lnpop95       -9.025e-02  1.269e-01  -0.711  0.47817
## lpd1500       -1.662e-01  1.410e-01  -1.179  0.24084
## yrentry        4.645e-02  1.008e-01   0.461  0.64579
## agritran       1.554e-01  1.459e-01   1.065  0.28881
## africa        -1.430e-01  2.114e-01  -0.677  0.49991
## europe        -4.793e-02  1.564e-01  -0.306  0.75980
## americas      -2.222e-01  2.381e-01  -0.933  0.35264
## reg_eap       -8.756e-02  1.215e-01  -0.721  0.47235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7739 on 124 degrees of freedom
## Multiple R-squared:  0.477,  Adjusted R-squared:  0.4011
## F-statistic: 6.284 on 18 and 124 DF,  p-value: 1.027e-10
```

OK

3 Analysis 2: Virtual Country (Re-Analysis)

The second analysis Michalopolous presents in his paper is essentially a repeat of the the previous, but aggregating over “virtual” countries instead of real ones, “in order 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 thus becomes 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 grid of countries. 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) (Weidmann, Rød, and Cederman 2010) dataset, which contains a shapefile of the locations of 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 try to see if the original paper's results held up with the GREG data.

The original GREG dataset required some manipulation to get it in a format suitable to swap in for the WLMS. 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. In other words, if a large portion of a virtual country was in an area which contained no languages—for example, the Sahara Desert—that virtual country was excluded from the analysis. The data download, which contained the virtual countries after the 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 we looked at the intersection of 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.

3.1 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 unique ethnic groups:

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

Additionally, regressions 2.5, 2.6, and 2.7 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.

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept) -2.6500e-15  1.0255e-01  0.0000 1.000000
## absLat1     -1.9566e-01  6.2818e-02 -3.1148 0.001877 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## t test of coefficients:
##
##                 Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  -2.0695e-15  1.1588e-01  0.0000 1.0000000
## sdElev1       1.5649e-01  5.5946e-02  2.7971 0.0052245 **
## sdSuitable1   2.0202e-01  5.4137e-02  3.7316 0.0001976 ***
## avgElev1     -5.9212e-02  4.1580e-02 -1.4241 0.1546481
## avgSuitable1  4.2408e-02  7.5600e-02  0.5609 0.5749219
## absLat1      -1.4645e-01  8.8857e-02 -1.6482 0.0995320 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## t test of coefficients:
##
##                   Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)    -4.9658e-15  7.5311e-02  0.0000  1.000000
## sdElev1         1.2950e-01  4.7178e-02  2.7450  0.006127 **
## sdSuitable1     1.3629e-01  4.4669e-02  3.0510  0.002322 **
## avgElev1       -1.1123e-01  6.3910e-02 -1.7404  0.082002 .
## avgSuitable1    6.4850e-02  3.9440e-02  1.6443  0.100337
## absLat1        -3.8227e-01  1.4397e-01 -2.6553  0.008012 **
## avgPrecip1      1.6189e-01  5.6146e-02  2.8834  0.003992 **
## avgTemp1       -3.3237e-01  1.4990e-01 -2.2173  0.026759 *
## lnArea1        -9.6029e-02  5.5662e-02 -1.7252  0.084706 .
## seaDist1        2.0797e-01  4.4284e-02  4.6964 2.902e-06 ***
## waterArea1     -6.4610e-03  2.7559e-02 -0.2344  0.814672
## withinCountry1 -9.0592e-02  5.3506e-02 -1.6931  0.090650 .
## numCountry1     2.9294e-01  3.8451e-02  7.6186 4.631e-14 ***
## migrationDist1 -1.1802e-01  9.0501e-02 -1.3041  0.192420
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## t test of coefficients:
##
##                   Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)    -6.5863e-15  7.4471e-02  0.0000 1.0000000
## sdElev1         1.1148e-01  5.0479e-02  2.2084 0.0273747 *
## sdSuitable1     1.2517e-01  4.5189e-02  2.7699 0.0056793 **
## avgElev1       -1.2337e-01  6.1989e-02 -1.9902 0.0467570 *
## avgSuitable1    1.7044e-02  4.5761e-02  0.3724 0.7096181
## absLat1        -4.4590e-01  1.3121e-01 -3.3984 0.0006964 ***
## avgPrecip1      1.3923e-01  5.6695e-02  2.4558 0.0141760 *
## avgTemp1       -3.9280e-01  1.3247e-01 -2.9652 0.0030753 **
## lnArea1        -1.1355e-01  5.3775e-02 -2.1115 0.0348992 *
## seaDist1        2.3187e-01  4.1947e-02  5.5278 3.848e-08 ***
## waterArea1     -1.1292e-02  2.8096e-02 -0.4019 0.6877972
## withinCountry1 -9.5373e-02  5.3658e-02 -1.7774 0.0757091 .
## numCountry1     2.9096e-01  3.9305e-02  7.4026 2.267e-13 ***
## migrationDist1 -9.6262e-02  9.4846e-02 -1.0149 0.3103152
## lnPopDens951    1.1070e-01  6.3766e-02  1.7360 0.0827726 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## t test of coefficients:
##
##                   Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)    -1.0292e-15  5.5080e-02  0.0000 1.0000000
## sdElev5        -1.6218e-01  6.3454e-02 -2.5559 0.0109335 *
## sdSuitable5     1.3272e-01  6.5636e-02  2.0220 0.0437927 *
## avgElev5        1.1355e-01  1.6038e-01  0.7080 0.4793060
## avgSuitable5    9.3581e-02  7.7099e-02  1.2138 0.2254921
## absLat5        -1.5785e-01  9.5186e-02 -1.6583 0.0979802 .
## avgPrecip5      8.5326e-02  1.1139e-01  0.7660 0.4440838
## avgTemp5        7.4479e-02  1.5762e-01  0.4725 0.6367968
## lnArea5        -6.9943e-02  5.0684e-02 -1.3800 0.1683092
## seaDist5        2.2080e-01  6.1157e-02  3.6103 0.0003417 ***
## waterArea5     -7.1364e-02  3.8858e-02 -1.8365 0.0669653 .
## withinCountry5 -4.9012e-02  6.6403e-02 -0.7381 0.4608550
## numCountry5     2.9148e-01  6.3716e-02  4.5747 6.239e-06 ***
## migrationDist5  2.1408e-01  9.6588e-02  2.2165 0.0271794 *
## lnPopDens955    3.5127e-01  7.5121e-02  4.6761 3.915e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## t test of coefficients:
##
##                   Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)     6.7295e-15  8.3360e-02  0.0000 1.0000000
## sdElev6         2.2909e-01  4.4006e-02  5.2059 2.348e-07 ***
## sdSuitable6     1.3547e-01  5.5319e-02  2.4489 0.0145040 *
## avgElev6       -1.7488e-01  6.0536e-02 -2.8888 0.0039517 **
## avgSuitable6    2.7062e-02  5.3837e-02  0.5027 0.6153047
## absLat6        -2.2458e-01  1.1102e-01 -2.0228 0.0433579 *
## avgPrecip6      1.7316e-02  7.4887e-02  0.2312 0.8171855
## avgTemp6       -3.6687e-01  1.0512e-01 -3.4901 0.0005042 ***
## lnArea6        -2.2147e-02  5.0349e-02 -0.4399 0.6601261
## seaDist6        1.9741e-01  5.9347e-02  3.3263 0.0009124 ***
## waterArea6      1.3515e-02  3.5721e-02  0.3783 0.7052559
## withinCountry6 -1.3159e-01  6.0650e-02 -2.1696 0.0302727 *
## numCountry6     2.6155e-01  5.2902e-02  4.9441 8.985e-07 ***
## migrationDist6 -1.2690e-01  9.8706e-02 -1.2857 0.1988606
## lnPopDens956    2.5362e-02  6.5902e-02  0.3848 0.7004388
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## t test of coefficients:
##
##                   Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)    -1.4554e-15  1.0267e-01  0.0000  1.000000
## sdElev7         1.2961e-01  7.1601e-02  1.8102  0.070621 .
## sdSuitable7     1.4192e-01  5.9297e-02  2.3934  0.016910 *
## avgElev7       -1.7774e-01  6.9501e-02 -2.5573  0.010721 *
## avgSuitable7   -3.8045e-03  7.5857e-02 -0.0502  0.960012
## absLat7        -6.1846e-01  2.0304e-01 -3.0460  0.002391 **
## avgPrecip7      8.3413e-02  8.4943e-02  0.9820  0.326387
## avgTemp7       -5.4181e-01  1.7264e-01 -3.1384  0.001757 **
## lnArea7        -1.5689e-01  7.3781e-02 -2.1264  0.033759 *
## seaDist7        2.6493e-01  6.0304e-02  4.3932 1.259e-05 ***
## waterArea7      1.1295e-03  3.8318e-02  0.0295  0.976491
## migrationDist7 -1.5649e-01  1.2825e-01 -1.2202  0.222742
## lnPopDens957    1.1153e-01  7.8205e-02  1.4261  0.154214
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.2 Robustness Check

4 References

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.

5 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])
# }
# NOTE: Easier to say what we're not standardizing; also for the robustness checks we need to keep a non-standa
dataStd = data %>% mutate(across(!countryName & !countryCode , standardize))
```

```r
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')
tbl1 = coefs %>% pivot_wider(names_from = "model", values_from = '.') %>%
  merge(ses %>% pivot_wider(names_from = "model", values_from = '.'),
        by = 'column',
        suffixes = c('Estimate', 'SE')
  ) %>%
  merge(pvals %>% pivot_wider(names_from = "model", values_from = '.'),
        by = 'column') %>%
  slice(match(order, column)) %>%
  column_to_rownames(var = "column")
tbl1format = data.frame(tbl1)
for (model in models) {
  estimCol = paste0(model, 'Estimate')
  seCol = paste0(model, 'SE')
  notNA = !is.na(tbl1[, estimCol])
  tbl1format[notNA, estimCol] = as.character(formatC(tbl1format[notNA, estimCol], digits = 3, format = 'f'))
  tbl1format[notNA, seCol] = as.character(formatC(tbl1format[notNA, seCol], digits = 3, format = 'f'))
  significant = tbl1[notNA, model] <= .01

  tbl1format[notNA, estimCol] = cell_spec(tbl1format[notNA, estimCol], italic = significant)
  tbl1format[notNA, seCol] = cell_spec(tbl1format[notNA, seCol], italic = significant)
}

order = c(1, 6, 2, 7, 3, 8, 4, 9, 5, 10)
col.names = c(paste0("(", 1:5, ") Coef."), paste0("(", 1:5, ") S.E."))[order]

tbl1format %>% select(order) %>%
  knitr::kable(toprule = '', bottomrule = '', booktabs = TRUE, linesep = c(""),
             col.names = col.names, align = "r",
             caption = 'Main specification for the cross-country analysis. Italics indicate significance at the
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-Squared'),
               caption = "Information for each model in cross-country analysis.")
robust1.1 <- glm.nb(numLang ~ absLat + sdSuitable + sdElev + avgElev + avgSuitable + avgPrecip + avgTemp + lnAr
                    + 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)

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)

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', 'range', 'area', 'withinCountry',
                   'numCountry', 'migrationDist', 'lnLang', 'totalPop95',
                   'absLat', 'tropics', 'erange_gecon', '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))
coeftest(model2.1, vcov = vcovCL, cluster = ~countryCode)
model2.2 = lm(lnLang1 ~ sdElev1 + sdSuitable1 + avgElev1 + avgSuitable1 + absLat1,
            data2 %>% filter(condition))
coeftest(model2.2, vcov = vcovCL, cluster = ~countryCode)
model2.3 = lm(lnLang1 ~ sdElev1 + sdSuitable1 + avgElev1 + avgSuitable1 + absLat1
            + avgPrecip1 + avgTemp1 + lnArea1 + seaDist1 + waterArea1
            + withinCountry1 + numCountry1 + migrationDist1,
            data2 %>% filter(condition))
coeftest(model2.3, vcov = vcovCL, cluster = ~countryCode)
model2.4 = lm(lnLang1 ~ sdElev1 + sdSuitable1 + avgElev1 + avgSuitable1 + absLat1
            + avgPrecip1 + avgTemp1 + lnArea1 + seaDist1 + waterArea1
            + withinCountry1 + numCountry1 + migrationDist1 + lnPopDens951,
            data2 %>% filter(condition))
coeftest(model2.4, vcov = vcovCL, cluster = ~countryCode)
condition = (data2$totalPop95 >= 3000) & (data2$suitableCells >= 10) & (data2$tropics == 1)
for (col in modelCols) {
  data2[condition, paste0(col, '5')] = standardize(data2[condition, col])
}
model2.5 = lm(lnLang5 ~ sdElev5 + sdSuitable5 + avgElev5 + avgSuitable5 + absLat5
            + avgPrecip5 + avgTemp5 + lnArea5 + seaDist5 + waterArea5
            + withinCountry5 + numCountry5 + migrationDist5 + lnPopDens955,
            data2 %>% filter(condition))
coeftest(model2.5, vcov = vcovCL, cluster = ~countryCode)
condition = (data2$totalPop95 >= 3000) & (data2$suitableCells >= 10) & (data2$tropics == 0)
for (col in modelCols) {
  data2[condition, paste0(col, '6')] = standardize(data2[condition, col])
}
model2.6 = lm(lnLang6 ~ sdElev6 + sdSuitable6 + avgElev6 + avgSuitable6 + absLat6
            + avgPrecip6 + avgTemp6 + lnArea6 + seaDist6 + waterArea6
            + withinCountry6 + numCountry6 + migrationDist6 + lnPopDens956,
            data2 %>% filter(condition))
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(lnLang7 ~ sdElev7 + sdSuitable7 + avgElev7 + avgSuitable7 + absLat7
            + avgPrecip7 + avgTemp7 + lnArea7 + seaDist7 + waterArea7
            + migrationDist7 + lnPopDens957,
            data2 %>% filter(condition))
coeftest(model2.7, vcov = vcovCL, cluster = ~countryCode)
```