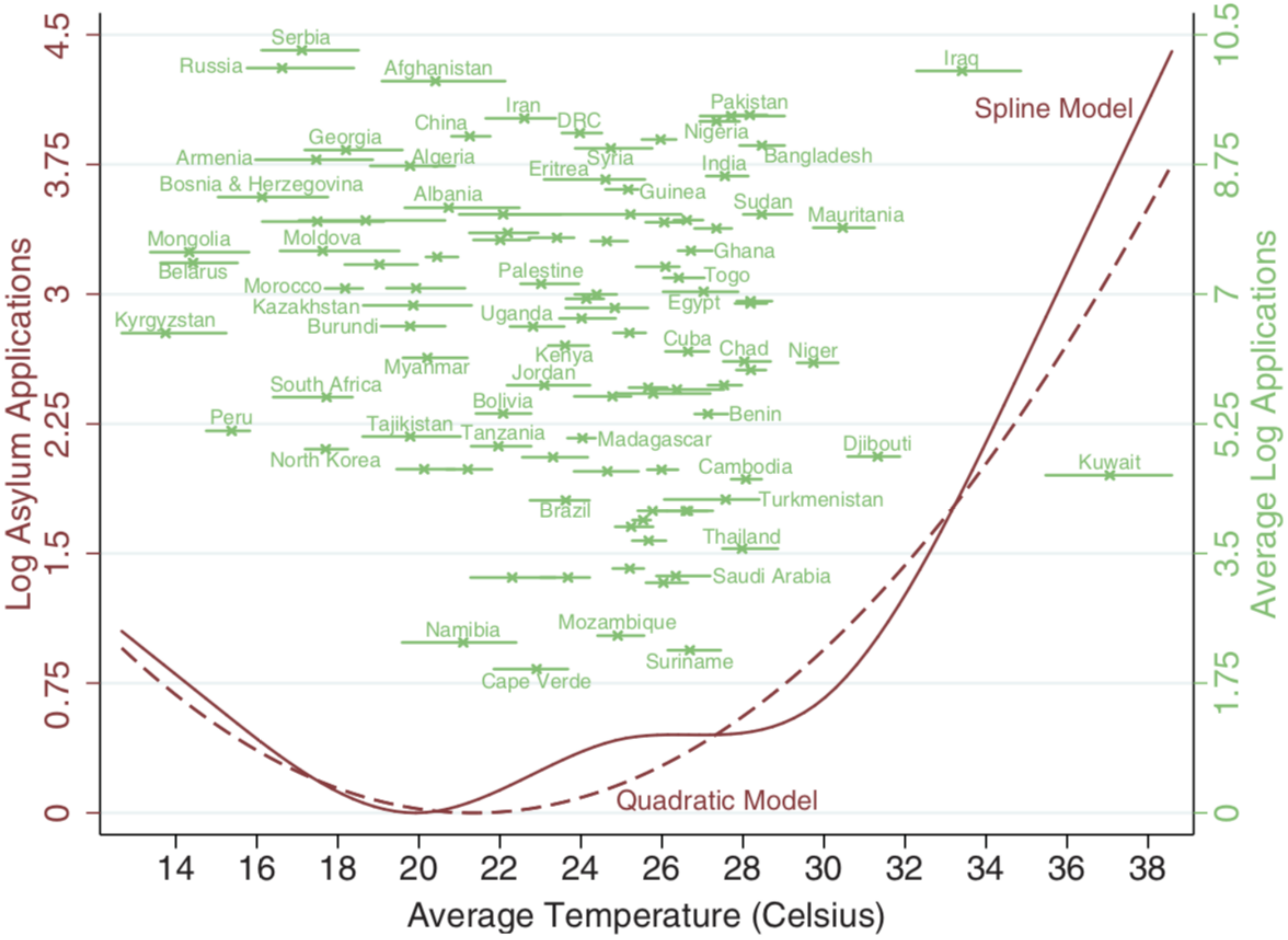
DS421 Replication - Missirian & Schlenker (2017)

Yiyi, Arfa and Elif

9/25/2019

Summary

We were interested in the green part of Figure 1, which captures the relationship between average temperatures and log asylum applications for each country.



In this figure, the ‘x’ marks represent each country’s average temperature and average log asylum applications. The horizontal lines represent the range of each country’s annual average temperature over the period of study.

We reproduce this portion of the plot and create an additional plot indicating the range of each country’s log asylum applications over the period of study.

Load libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readstata13)
library(ggplot2)
library(tidyr)
```

Load data

```
# loads data from
baseline <- read.dta13("data/baselineData.dta")
country <- read.dta13("data/countryData.dta")
nex <- read.dta13("data/NEXdata.dta")
climate <- read.dta13("data/uniformClimateChange.dta")
additional <- read.dta13("data/additionalData.dta")
```

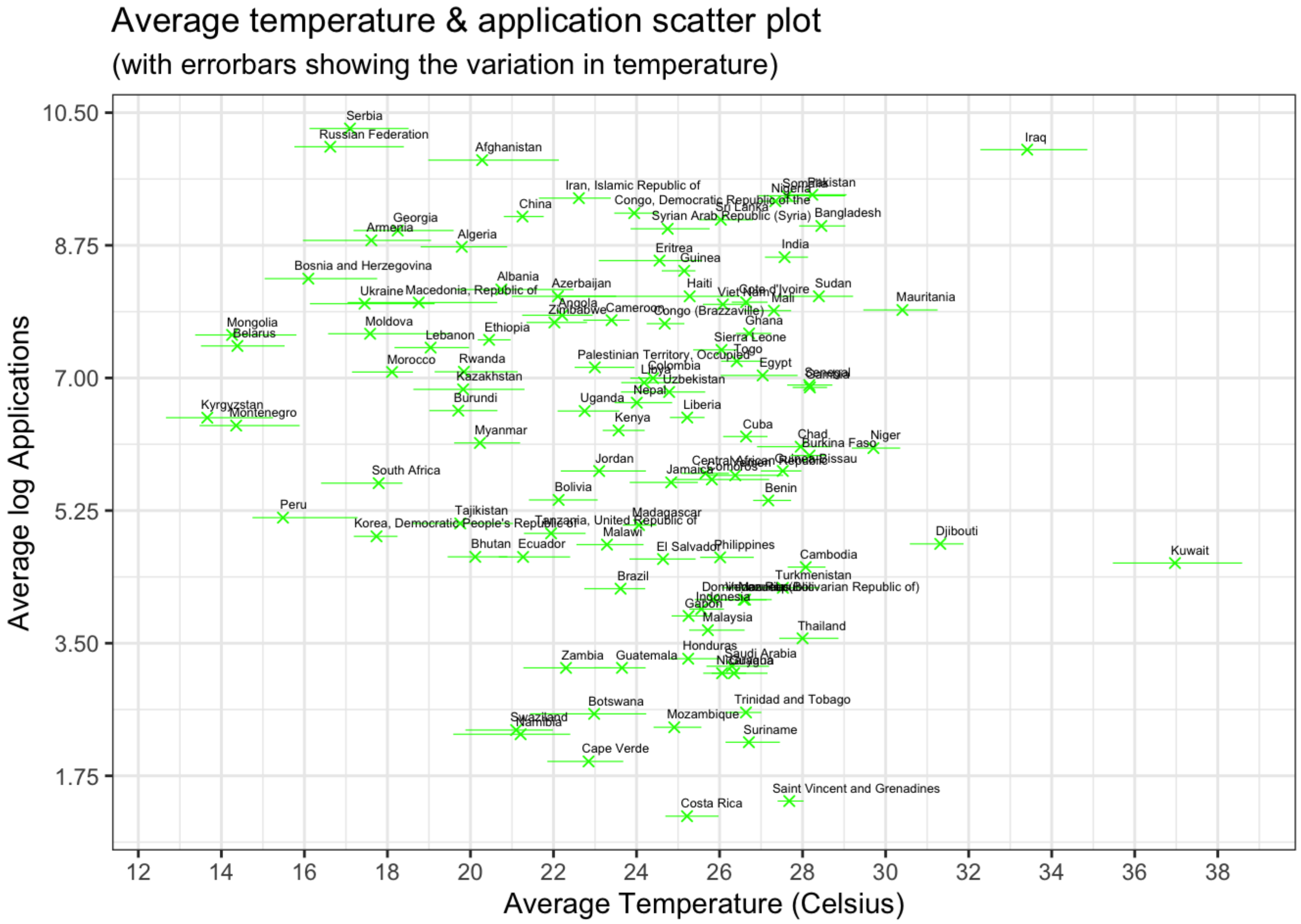
```
summary <- baseline %>%
  select(originCountryName, year, applications, tAvgDEL) %>%
  mutate(log_app = log(applications)) %>%
  group_by(originCountryName) %>%
  summarise(min_log_app = min(log_app, na.rm = TRUE),
            mean_log_app = mean(log_app, na.rm = TRUE),
            max_log_app = max(log_app, na.rm = TRUE),
            min_temp = min(tAvgDEL, na.rm = TRUE),
            mean_temp = mean(tAvgDEL, na.rm = TRUE),
            max_temp = max(tAvgDEL, na.rm = TRUE)) %>%
  filter(mean_log_app > 0)
```

```
summary <- summary %>%
  mutate(app_range = max_log_app - min_log_app, temp_range = max_temp - min_temp)
summary
```

```
## # A tibble: 110 x 9
##    originCountryNa… min_log_app mean_log_app max_log_app min_temp mean_temp
##    <chr>                  <dbl>        <dbl>       <dbl>    <dbl>     <dbl>
##  1 Afghanistan             8.91         9.87       10.9      19.0      20.3
##  2 Albania                 7.10         8.17       10.0      19.7      20.7
##  3 Algeria                 8.15         8.73        9.31     18.8      19.8
##  4 Angola                  6.96         7.83        8.95     21.2      22.2
##  5 Armenia                 8.46         8.81        9.00     16.0      17.6
##  6 Azerbaijan              7.58         8.08        8.46     21.0      22.1
##  7 Bangladesh              8.64         9.01        9.65     27.9      28.5
##  8 Belarus                 6.86         7.42        8.02     13.5      14.4
##  9 Benin                   4.58         5.38        6.26     26.8      27.2
## 10 Bhutan                  3.95         4.64        5.40     19.5      20.1
## # … with 100 more rows, and 3 more variables: max_temp <dbl>,
## #   app_range <dbl>, temp_range <dbl>
```

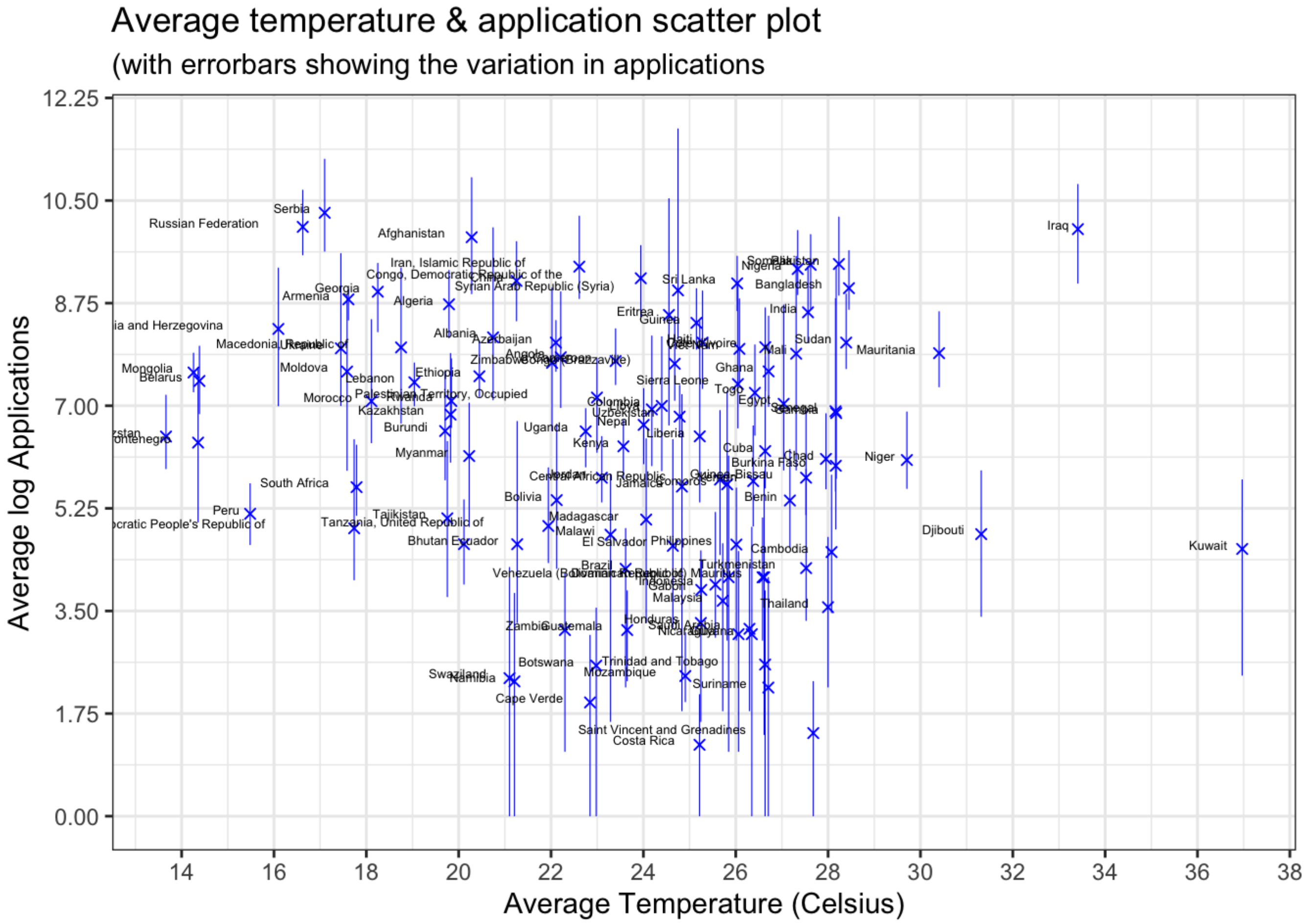
Scatter plot with range of temperature for each country

```
ggplot(summary, aes(x = mean_log_app, y = mean_temp)) +
  geom_point(shape = 4, color = "green") +
  geom_errorbar(mapping = aes(x = mean_log_app, ymin = min_temp, ymax = max_temp), size=
0.2, color = "green") +
  geom_text(aes(label=originCountryName),hjust=0.1, vjust=-1, size=1.7) +
  coord_flip() +
  theme_bw() +
  scale_y_continuous(name = "Average Temperature (Celsius)", breaks = seq(0,40, 2)) +
  scale_x_continuous(name = "Average log Applications", breaks = seq(0,15,1.75)) +
  labs(title = "Average temperature & application scatter plot", subtitle = "(with error
bars showing the variation in temperature)")
```



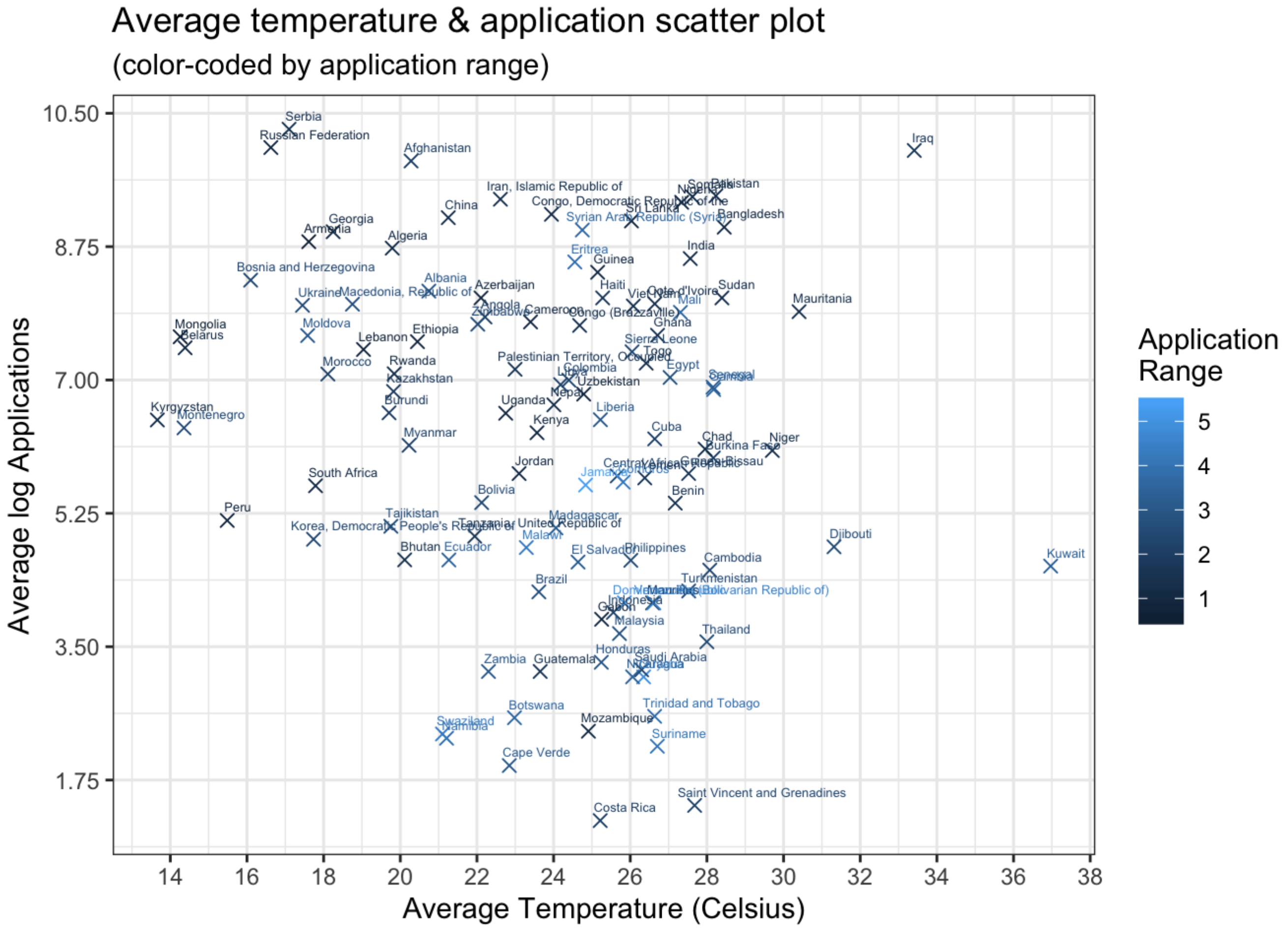
Scatter plot with range of log asylum applications for each country

```
ggplot(summary, aes(x = mean_temp, y = mean_log_app)) +
  geom_point(shape = 4, color = "blue") +
  geom_errorbar(mapping = aes(x = mean_temp, ymin = min_log_app, ymax = max_log_app), si
ze = 0.2, color = "blue") +
  geom_text(aes(label=originCountryName),hjust=1.4, vjust=0, size=1.7) +
  theme_bw() +
  scale_x_continuous(name = "Average Temperature (Celsius)", breaks = seq(0,40, 2)) +
  scale_y_continuous(name = "Average log Applications", breaks = seq(0,15,1.75)) +
  labs(title = "Average temperature & application scatter plot", subtitle = "(with error
bars showing the variation in applications")
```



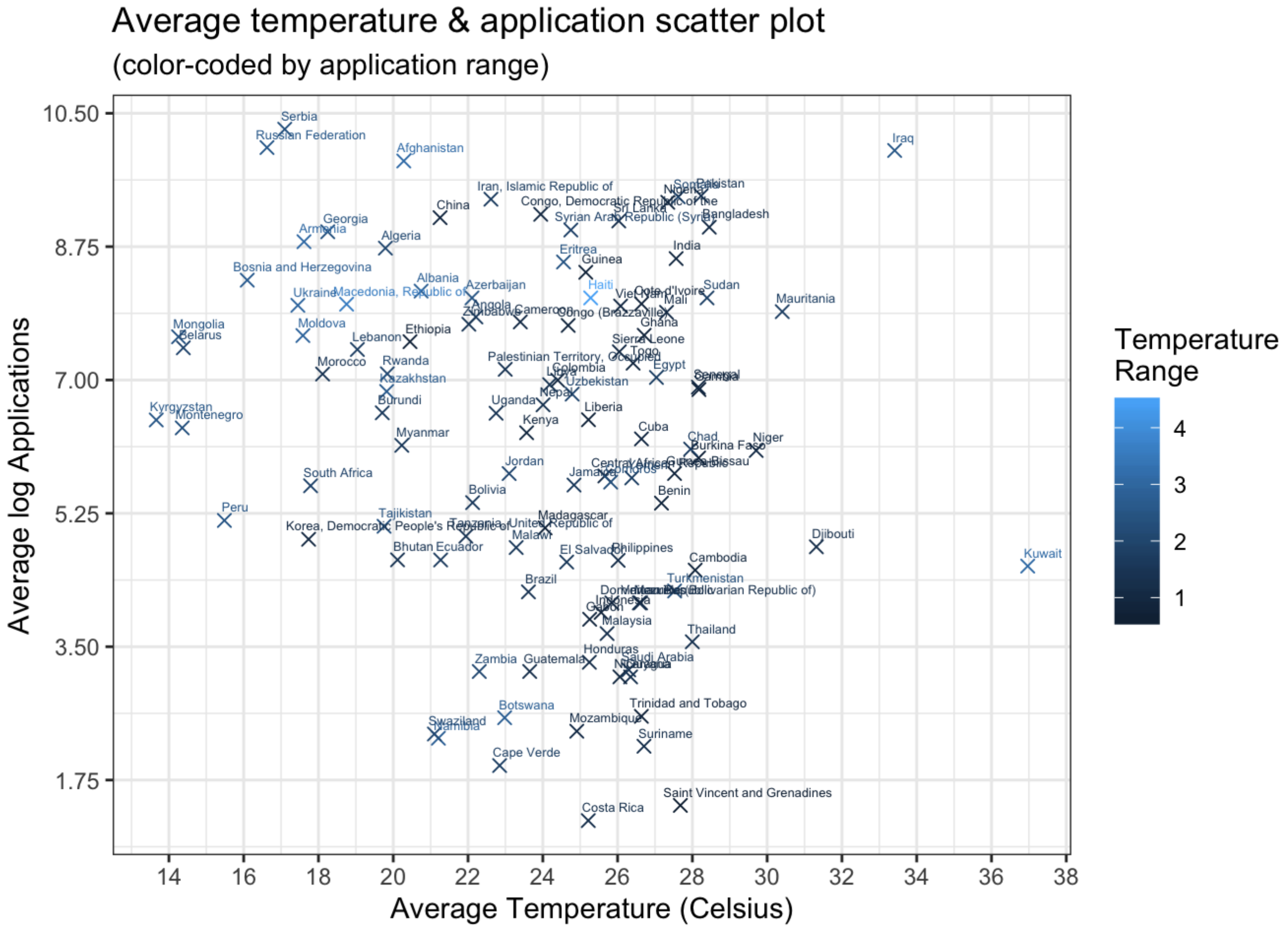
Scatter plot with range of log applications for each country (visualized using colorramp)

```
ggplot(summary, aes(x = mean_temp, y = mean_log_app, color = app_range)) +
  geom_point(shape = 4, size = 2) +
  theme_bw() +
  scale_x_continuous(name = "Average Temperature (Celsius)", breaks = seq(0,40, 2)) +
  scale_y_continuous(name = "Average log Applications", breaks = seq(0,15,1.75)) +
  geom_text(aes(label=originCountryName),hjust=0.1, vjust=-1, size=1.7) +
  labs(title = "Average temperature & application scatter plot", subtitle = "(color-code
d by application range)", color = "Application \nRange")
```



Scatter plot with range of temperature for each country (visualized using colorramp)

```
ggplot(summary, aes(x = mean_temp, y = mean_log_app, color = temp_range)) +
  geom_point(shape = 4, size = 2) +
  theme_bw() +
  scale_x_continuous(name = "Average Temperature (Celsius)", breaks = seq(0,40, 2)) +
  scale_y_continuous(name = "Average log Applications", breaks = seq(0,15,1.75)) +
  geom_text(aes(label=originCountryName),hjust=0.1, vjust=-1, size=1.7) +
  labs(title = "Average temperature & application scatter plot", subtitle = "(color-code
d by application range)", color = "Temperature \nRange")
```



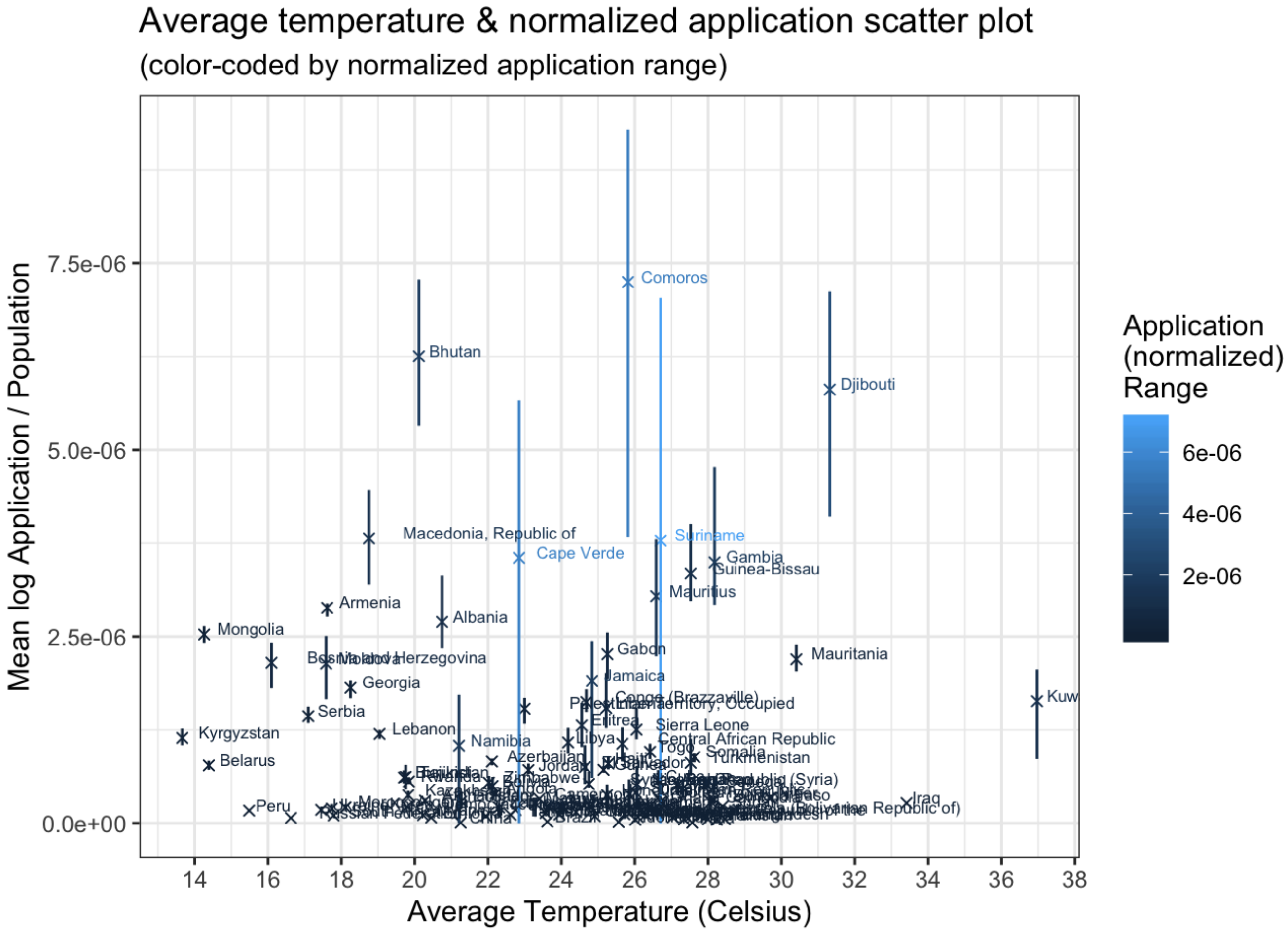
Exploring the relationship between average temperature and application normalized using total population.

```
merged <- left_join(baseline, country, by="originCodeISO")
summary2 <- merged %>%
  select(originCountryName, year, applications, tAvgDEL, population) %>%
  mutate(log_app = log(applications)) %>%
  group_by(originCountryName) %>%
  summarise(min_log_app = min(log_app, na.rm = TRUE),
            mean_log_app = mean(log_app, na.rm = TRUE),
            max_log_app = max(log_app, na.rm = TRUE),
            min_temp = min(tAvgDEL, na.rm = TRUE),
            mean_temp = mean(tAvgDEL, na.rm = TRUE),
            max_temp = max(tAvgDEL, na.rm = TRUE),
            population = max(population, na.rm = TRUE)) %>%
  filter(mean_log_app > 0, population > 0)
## new attempt at graph
```

```
summary2 <- summary2 %>%
  mutate(norm_range = max_log_app/population - min_log_app/population)
summary2
```

```
## # A tibble: 103 x 9
##    originCountryNa… min_log_app mean_log_app max_log_app min_temp mean_temp
##    <chr>                  <dbl>        <dbl>       <dbl>    <dbl>     <dbl>
##  1 Afghanistan             8.91         9.87       10.9      19.0      20.3
##  2 Albania                 7.10         8.17       10.0      19.7      20.7
##  3 Algeria                 8.15         8.73        9.31     18.8      19.8
##  4 Angola                  6.96         7.83        8.95     21.2      22.2
##  5 Armenia                 8.46         8.81        9.00     16.0      17.6
##  6 Azerbaijan              7.58         8.08        8.46     21.0      22.1
##  7 Bangladesh              8.64         9.01        9.65     27.9      28.5
##  8 Belarus                 6.86         7.42        8.02     13.5      14.4
##  9 Benin                   4.58         5.38        6.26     26.8      27.2
## 10 Bhutan                  3.95         4.64        5.40     19.5      20.1
## # … with 93 more rows, and 3 more variables: max_temp <dbl>,
## #   population <dbl>, norm_range <dbl>
```

```
ggplot(summary2, aes(x = mean_temp, y = mean_log_app/population, color = norm_range)) +
  geom_point(shape = 4) +
  geom_errorbar(mapping = aes(x = mean_temp, ymin = min_log_app/population, ymax = max_l
og_app/population)) +
  theme_bw() +
  geom_text(aes(label=originCountryName),hjust=-0.2, vjust=0, size=2.2) +
  scale_x_continuous(name = "Average Temperature (Celsius)", breaks = seq(0,40, 2)) +
  scale_y_continuous(name = "Mean log Application / Population") +
  labs(title = "Average temperature & normalized application scatter plot", subtitle = "
(color-coded by normalized application range)", color = "Application \n(normalized) \nRa
nge")
```



Trying to replicate the quadratic and spline models

```
knitr::include_graphics('TableS1.PNG')
```

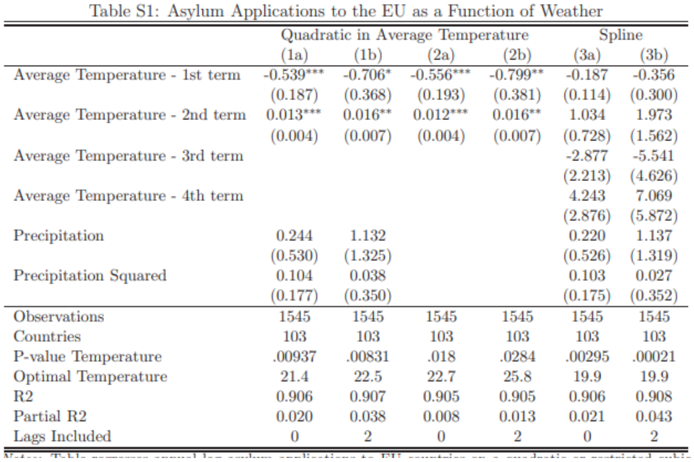


Table S1: Asylum Applications to the EU as a Function of Weather

	Quadratic in Average Temperature				Spline	
	(1a)	(1b)	(2a)	(2b)	(3a)	(3b)
Average Temperature - 1st term	-0.539***	-0.706*	-0.556***	-0.799**	-0.187	-0.356
	(0.187)	(0.368)	(0.193)	(0.381)	(0.114)	(0.300)
Average Temperature - 2nd term	0.013***	0.016**	0.012***	0.016**	1.034	1.973
	(0.004)	(0.007)	(0.004)	(0.007)	(0.728)	(1.562)
Average Temperature - 3rd term					-2.877	-5.541
					(2.213)	(4.626)
Average Temperature - 4th term					4.243	7.069
					(2.876)	(5.872)
Precipitation	0.244	1.132			0.220	1.137
	(0.530)	(1.325)			(0.526)	(1.319)
Precipitation Squared	0.104	0.038			0.103	0.027
	(0.177)	(0.350)			(0.175)	(0.352)
Observations	1545	1545	1545	1545	1545	1545
Countries	103	103	103	103	103	103
P-value Temperature	.00937	.00831	.018	.0284	.00295	.00021
Optimal Temperature	21.4	22.5	22.7	25.8	19.9	19.9
R2	0.906	0.907	0.905	0.905	0.906	0.908
Partial R2	0.020	0.038	0.008	0.013	0.021	0.043
Lags Included	0	2	0	2	0	2

Quadratic and spline fit

Unable to explain the negative values

```
baseline2 <- baseline %>%
  mutate(log_assylum_quad = (tAvgDEL*-0.539)+(tAvgDEL2*0.013)+(precDEL*0.244)+(precDEL2*
0.104)) %>%
  mutate(log_assylum_spline = (tSplineDEL1*-0.187)+(tSplineDEL2*1.034)+(tSplineDEL3*-2.8
77)+(tSplineDEL4*4.243)+(precDEL*0.220)+(precDEL2*0.103)) %>%
  mutate(log_applications = log(applications)) %>%
  select(log_applications, log_assylum_quad, log_assylum_spline)

head(baseline2)
```

```
##   log_applications log_assylum_quad log_assylum_spline
## 1               NA        -4.996904          -2.797324
## 2               NA        -5.062105          -2.859595
## 3        10.474043        -5.029436          -2.835687
## 4        10.895961        -4.996133          -2.813088
## 5        10.219903        -5.067005          -2.856897
## 6         9.449436        -5.099237          -2.872639
```