STAT 230A Replication Project

Danny Wu

Group 03

May 10, 2022

1 Summary

In *Temperature and Decisions: Evidence from 207,000 Court Cases*, authors Anthony Heyes and Soodeh Saberian investigate the link between outdoor temperature to decisions made by "experienced professional decision-makers, working in good-quality, climate-controlled, indoor spaces." (Heyes, Saberian 2019) Specifically, they examine just under 207,000 court cases conducted by 266 immigration judges at 43 courthouse locations and conclude with high significance, robustness, and substantial effect size that variations in climate can damage well-being by influencing decisions. Their results signify that a 10-degree increase in temperature reduces the probability of a favorable decision (such as the granting of asylum in the context of immigration court) by 1.075 percent. This corresponds to a roughly 6.55 percent decrease in the overall "grant" rate, which on average is 16.4 percent. They conclude that the effect is particularly pronounced for female judges. In addition, they extend their analysis to decisions made in 18,461 Parole Suitability Hearings at the 39 locations of the California Department of Corrections and Rehabilitation, arriving at similar results and further solidifying their conclusions.

The authors have three sets of data, the first of which is case-level administrative data from asylum applications in the United States. These were conducted in immigration courts from January 2000 to September 2004. In total, there were 206,924 decisions made by all 266 judges across 43 courthouses in major cities, with each courthouse corresponding to its geographical region. This decision data is merged with hand-collected data on the gender of judges, of which 34% are female. Within this dataset, 16% of cases were granted asylum, with a large variation in grant rate between judges. Some judges granted fewer than 4% of cases, while others granted over 57%. In practice, the assignments to judges are randomized, however, the authors do not verify this in the data, as their methods do not employ the use of randomization for identification. This is a nonissue, as the setting of dates and of who presides over each case is done well in advance (over a year), so there is likely to be little confounding between how cases are assigned. All courtrooms are in climate-controlled buildings, which as the authors describe is at a level "typical of good-quality U.S. Federal government buildings in the twenty-first century." They assume that the effects of external temperature on internal actions have already been adjusted for the needed adaptation from outside weather to climate-controlled weather.

Next, the authors obtain data from the Board of Parole Hearing (BPH) with the California Department of Corrections and Rehabilitation. This contains 18,461 decisions made by 12 BPH commissioners across 39 venues in California. Each hearing determines whether or not inmates are to be released from prison back into society on parole and contains information about the date of hearing, the identity of the commissioners, the location, and the outcome.

Lastly, the authors obtain environmental data to answer their main question regarding whether adjudication responds to the outdoor temperature on the day it is evaluated. They merge the judge data with temperature and a wide variety of other environmental controls, as the locations of asylum decisions are widely dispersed in different cities that are subject to diverse weather conditions. First, they gather temperature and weather data from the National Oceanic and Atmospheric Administration. For each courthouse, they assign weather information from the closest monitoring stations, which in no case is further than 20 miles away. The average distance is 9.35 miles with a standard deviation of 6.33 miles. From this data, they only select weather between 6:00 AM and 4:00 PM, as this is what the authors consider to be the "active" time that people are around and getting ready for work. The authors excluded exposure after the courts close as they reason that this logically cannot affect proceedings. We reproduce the author's summary statistics in Table 1, as well as the distribution of cases over 6AM to 4AM temperature bins in Figure 1

2 Replication of Results

2.1 Statistical Model and Assumptions

The authors estimate the following linear probability model:

$$g_{it} = \beta_0 + \beta_i \text{temp}_{it} + W_{it}\beta_2 + P_{it}\beta_3 + X_{it}\beta_4 + \gamma_i + \Psi_{ct} + \theta_t + \epsilon_{it}$$

g_{it} is a binary variable which indicates whether a judge's decision on application i on date t is granted. temp_{it} is the key variable of interest, which is the average temperature from 6 AM to 4 PM on the date of the court case, with most of the discussion of results centering around β_1. W_{it} represents a vector of weather controls in the vicinity of the courthouse for application i on date t, and P_{it} is a vector for pollution

Table 1: Summary Statistics

	Mean	SD
Grant Indicator	0.164	0.371
Temperature (°F)	57.371	15.720
Heat Index (°F)	57.776	16.412
Air Pressure (pa)	29.693	0.759
Dew point (°F)	49.371	17.202
Precipitation (mm)	0.003	0.014
Wind speed (km/h)	4.557	3.441
Sky cover (percent)	55.44	27.56
Ozone (ppm)	0.022	0.012
CO (ppm)	0.918	0.496
$\text{PM}_{2.5}$ (μ/m^3)	14.957	11.570

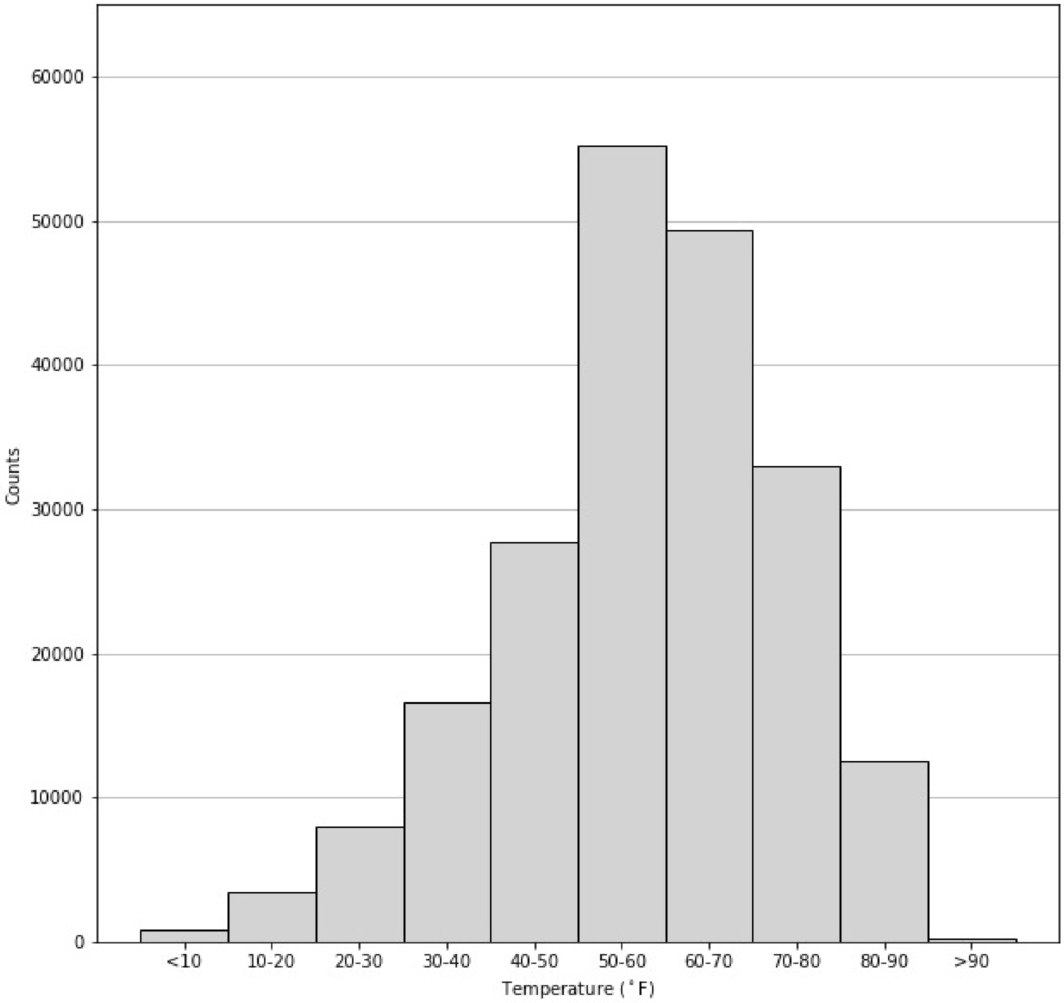
controls with similar indexing. X_{it} is a vector of court and application controls, specifically the category of application and the nationality of the applicant. γ_i represents judge fixed effects to account for variation in grant probabilities between judges, and Ψ_{ct} is a vector of city-by-month fixed effects where c indexes over the cities that court-houses are located in, and t represents the month. Lastly, θ_t includes time fixed effects which controls for day of week and year trends in the data.

The authors identify the effect off within-location, within-month variation. Their main assumption is that by controlling for location and time effects, the realization of outdoor temperature on a particular given day is as good as random. Since these changes to temperature are basically exogenous, this allows for insight into the effect of temperature on the probability of an affirmative decision.

The authors note that while they have a rich set of fixed effects, there might still be some unobserved characteristics that are important determinants of outcomes that were omitted. However they reason that controlling for location and time fixed effects is plausibly enough to make these omitted characteristics uncorrelated with the case-day temperature, hence the estimates for β_1 and are unbiased and the associated standard errors remain undisturbed.

Lastly, they acknowledge that the data could potentially be serially and spatially correlated. As a result, their standard errors are clustered by city-month, which accounts for both of these issues.

Figure 1: Histogram of cases over 6AM to 4PM temperature bins.



2.2 Main Results

We reproduce the authors' results in Table 2. Although using the same datasets as the authors and following the same procedure, albeit using the `pandas` package in Python for auxiliary data cleaning and the `felm` package in R to handle the fixed effects linear regression estimation, we end up with a slightly larger number of observations (207,207 in comparison to 206,924). This is an increase of only 283 observations, or an increase of less than 0.1%. Perhaps as a result of this discrepancy of units, our point estimates are on average around 2% smaller than those produced by the authors. However, our point estimates all have the same sign, and our standard errors all remain close to the original results. Furthermore, the F-statistics all increase for our model results, indicating that there is stronger evidence that the weather variables are jointly significant. As a result of this, our p-values for the F- tests are also smaller. The authors also run a regression where they include the 3 lag and lead terms and plot the coefficients and confidence intervals, which we reproduce in Figure 2.

Since our independent variables are divided by 1000, this first converts the unit of our point estimates from proportions to percentages out of 100. The additional factor of 10 then considers a unit change of our independent variable as a change of 10 degrees Fahrenheit. Hence the results imply that a 10 degree increase in daily average temperature from 6AM to 4PM will decrease the probability of an affirmative decision by roughly 1.03%. Considering that the average affirmative decision baseline is 16.4%, this implies a 6.55 percent decrease in grant rate. To rephrase this, when the weather is 10 degrees hotter on any given day, the probability of having one's asylum application granted will decrease by 6.55%.

Table 2: Main Results

	Base	1 Day Lag	1 Day Lead	1 Day Lag/Lead
	(1)	(2)	(3)	(4)
$\text{Temperature}_t/1,000$	-1.038^{***}	-1.443^{***}	-1.187^{***}	-1.590^{***}
	(0.327)	(0.403)	(0.381)	(0.484)
$\text{Temperature}_{t-1}/1,000$		0.357		0.367
		(0.276)		(0.276)
$\text{Temperature}_{t+1}/1,000$			0.123	0.143
			(0.259)	(0.259)
F-Statistic for Weather Variables	3.587	3.451	3.261	3.03
p-value	0.00172	0.00128	0.00213	0.00245
Observations	207,207	207,207	207,207	207,207
Note:				*p<0.1; **p<0.05; ***p<0.01

The authors believe that their main model includes the most natural set of time fixed effects. However, they also report various permutations of fixed effect variables to examine the stability of their main model. We reproduce these permutations of fixed effects in Table 3, confirming the authors' results that the main selection of fixed effects are valid.

2.3 Appraisal of Statistical Assumptions

The primary point of critique on the author's assumptions is their use of a linear probability model. Binary outcomes and continuous explanatory variables have an inherently nonlinear relationship, as the probability of occurrence is bounded. Hence, although using this model for inference using the current data set gives us estimates of the effect of weather on outcome decisions, it is plausible to believe that out-of-sample analysis might have coefficients that are not accurately represented by the author's results. This linearity is a very

Figure 2: Timing of Exposure: 6AM to 4PM.

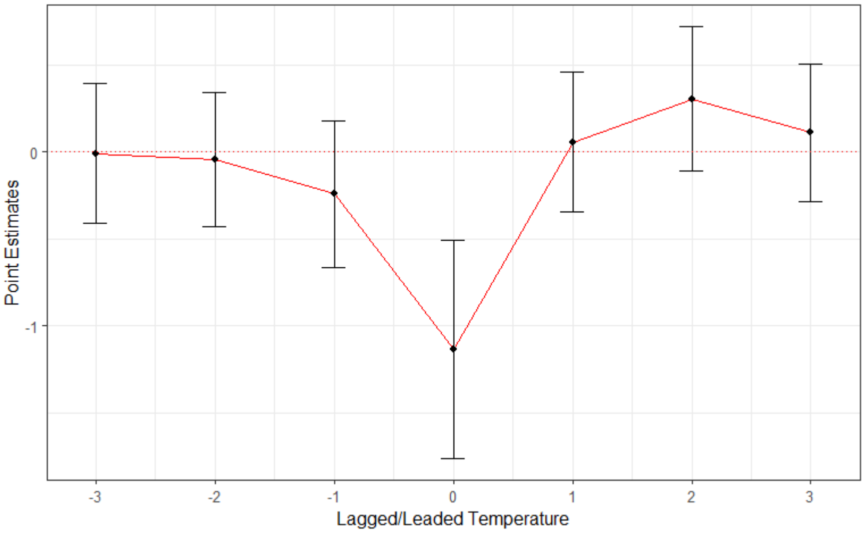


Table 3: Alternate Fixed Effects

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
$\text{Temperature}_t/1{,}000$	-0.727^{***} (0.269)	-0.737^{***} (0.272)	-0.788^{***} (0.269)	-0.815^{***} (0.248)	-1.031^{***} (0.277)	-0.897^{***} (0.214)	-1.082^{***} (0.269)	-0.933^{***} (0.283)
Nationality FEs	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
Day of Week FEs	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE
Type of Application FEs	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
Judge FEs	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE
City-Month FEs	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE
Judge-Month FEs	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
City FEs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE
Year FEs	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE
Year-Month FEs	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
Date FEs	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
Observations	207,207	207,207	207,207	207,207	207,207	207,207	207,207	207,207

Note: $^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$

strong assumption that the authors are making. However, in the context of advising policy makers, it is the most easily interpreted. Perhaps in the larger picture, the sign, magnitude, and significance are of greater importance than the accuracy of the point estimates. Nonetheless, it could be imagined that this linear probability model does not capture the true causal relationship between temperature and decision outcomes.

Table 4: Robustness Results

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
Temperature	-0.907^{***}	-1.067^{**}	-2.074	-1.567^{***}	-1.656^{***}			-0.773^{*}	-0.989^{***}
	(0.268)	(0.420)	(1.269)	(0.378)	(0.399)			(0.463)	(0.311)
Heat Index						-0.431^{**}	-2.007^{***}		
						(0.197)	(0.758)		
Observations	207,207	135,428	13,997	134,051	111,475	207,204	29,458	104,266	172,743
Note:									*p<0.1; **p<0.05; ***p<0.01

3 Robustness

In the following section, we replicate the suite of robustness tests the author conducts. In their paper, they consider a variety of effects that might confound the relationship between temperature and decisions. First, they consider the impact of pollution, and reason that if certain measures of pollution are confounding, then dropping the vector of pollution controls should drastically change the coefficient on temperature in the regression. In column (1) of Table 4, we replicate their results and obtain similar estimates (their coefficient is -0.910, ours is -0.907, a difference of 0.003). Since the magnitude and sign are relatively undisturbed compared to the original point estimate (-1.038), it is likely that pollution is not a large confounder as the coefficients do not change drastically.

Decisions made in California consist of almost 32% of the observations (roughly 71,000 of the 207,000 decisions). In order to check if there is something different going on in California that is swaying the results of their regression, the authors exclude all decisions made in California and rerun their main specification. We achieve similar results to the authors in column (2) of Table 4 (a coefficient of -1.067 compared to -1.159) and conclude that there are no idiosyncrasies associated with California.

Existing research points to how cloud cover could influence mood, hence the authors test running their model on observations where decisions were made on ”clear sky” days - the cloud cover was less than 5 percent. Our results are in column (3) of Table 4 and we obtained a smaller coefficient than the authors (-2.074 compared to -2.738), however the overall conclusion remains the same. This suggests that there is a more pronounced negative effect of hotter temperatures on clear sky days, however, like the authors, our results are not significant on the 5 percent level. Although the authors make no attempt at explaining this higher result, a possible explanation could be how direct sunlight causes the perceived temperature to be higher than actual ambient levels.

Additionally, the authors explain that research has also pointed to how rain can influence people's moods.

The authors then take a subset of data where there was zero precipitation, arguing that these observations cannot have rain influencing the outcomes. Running the regression results in coefficients that retain the sign and significance, however with a larger magnitude and this can be seen in column (4) of Table 4. The authors get a coefficient of -1.304 while we have a slight increase at -1.567. The authors also extend to considering days that had no rain before or after to further solidify that rain could have no effect on these decisions. We replicate this in column (5) of Table 4 and have a rather large difference from the authors. Our coefficient ends up being -1.656 while the authors obtain a coefficient of -1.281. Similar to the clear skies results, rain is typically associated with cloudy weather. Hence, this is potentially just a rerun of that regression, with fewer observations. On days with no rain, the lack of clouds might make the temperature be perceived to be hotter, hence resulting in the larger coefficient.

After considering weather impacts on judge's decision making, the authors consider using a new measure called the Heat Index used by the US National Weather Service which combines both air temperature and relative humidity through a nonlinear algorithm. They posit that this better captures the way that temperature is experienced by humans since different weather conditions can cause the same temperature to be felt hotter or cooler. Simply put, how the weather feels can vary day to day even if the temperature is the same. By re-estimating the results using this heat index instead of just using base temperature, the authors obtain a lower and less significant result when considering all observations, and our replication matches this in column (6) of Table 4. The authors get a coefficient of -0.44 compared to our -0.431. This is in comparison to the original coefficient of -1.038. The overall effect of temperature is cut in half when using a new metric. The authors note that the heat index is typically only considered reliable on warm days, so they then consider only observations where the heat index was greater than 75. We match their results in column (7) of Table 4, obtaining a coefficient that is negative, slightly large in magnitude, and significant (-2.007 for us compared to -1.991).

Lastly, the authors consider discrepancies in judge's granting rates, as judges do not have specific quotas, and hence some judges are more likely to grant or deny than others. To convince themselves that outlier judges are not driving the results, they drop the first and last quartiles and deciles and reconduct the analysis. We replicated the same results in columns (8) and (9) of Table 4 with a coefficient of -0.773 compared to -0.707 for excluding the bottom and top quartiles and a coefficient of -0.989 compared to -1.064 for dropping the top and bottom deciles. Both of these are very close to our original estimate of the effect size at -1.038, and hence outlier judges are likely to not be an issue.

4 Reanalysis

In the following section, we attempt to reanalyze the authors' main result first leverage score analysis and then consider a different statistical assumption for the relationship between the covariates and the probability of a favorable decision by running a logit regression for a binary outcome instead of a linear probability model. The use of leverage scores is justified as it allows for us to examine which observations might be potentially driving the results, as outliers may cause the coefficients to be biased. A logit model is justified as the relationship between covariates and approval probability may not be linear as assumed by the authors' model. We instead choose to use a logit model, to see how shifting that assumption might alter the results. We then compute the average marginal effect of temperature on the outcome and compare this with the base result.

4.1 Leverage Score Analysis (Observation Level)

We re-analyze the author's results by first taking a look at the leverage scores of their data from their linear probability model. As seen from Figure 3, there is a *highly* skewed distribution of leverage scores, with a select subset of the data vastly larger than the majority of the data. In Figure 4, we only look at the leverage scores less than 0.005. Looking to the right of the figure is the empirical density plot of the leverage scores, and you can see it is no longer as highly skewed as before. Leverage scores indicate the impact each observation has with regards to the overall observation, and give a rough measure of how much the observation influences the fitted value, and hence the coefficients. Hence, we hypothesize that only a select subset is driving the results of the regression, and dropping them will drastically change the coefficients of interest.

We proceed by dropping high-leverage observations and re-running the base model the authors use with the remaining observations. Since various cutoffs can be used to determine what leverage score is considered to be "too high", we use three decreasing thresholds and examine the impact on the coefficient and confidence interval for temperature. First, we drop all observations with leverage scores greater than 0.05, then with scores greater than 0.01, and lastly with scores greater than 0.0025. Our results are summarized in Table 5. The first column lists the author's results. Our first cutoff at 0.05 removes 1162 observations and causes a *very* slight decrease in the coefficient. At the next cutoff of 0.01, we remove a total of just under 20,000 observations, and there is a further decrease. At the last cutoff of 0.05, we remove almost 50,000 observations, yet still, the results remain the same, albeit with wider confidence intervals due to the fewer observations. Even removing the most influential observations, the author's results remain very stable and still significant, reassuring the validity of their results.

Figure 3: Plot of all leverage scores.

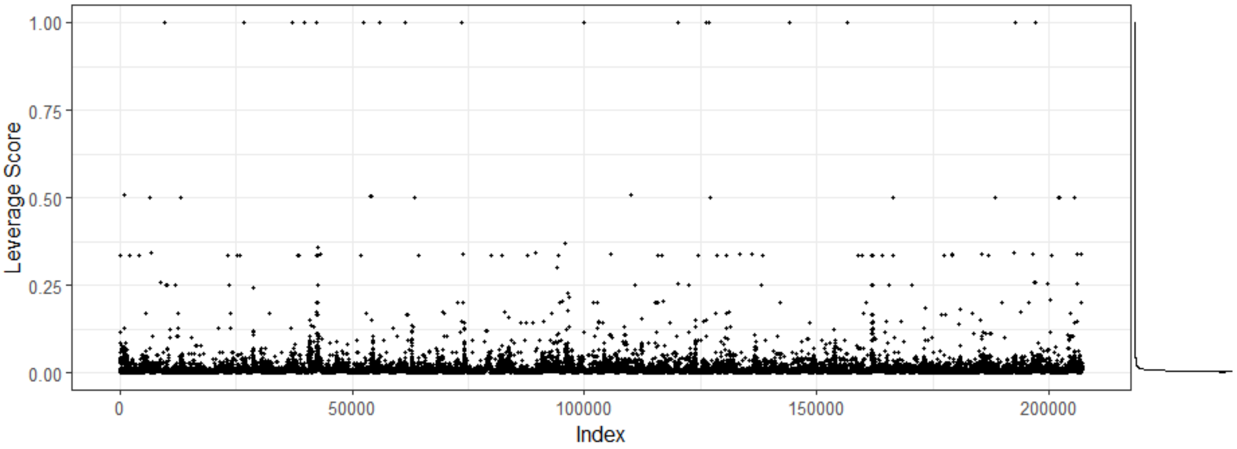


Figure 4: Plot of leverage scores less than 0.005.

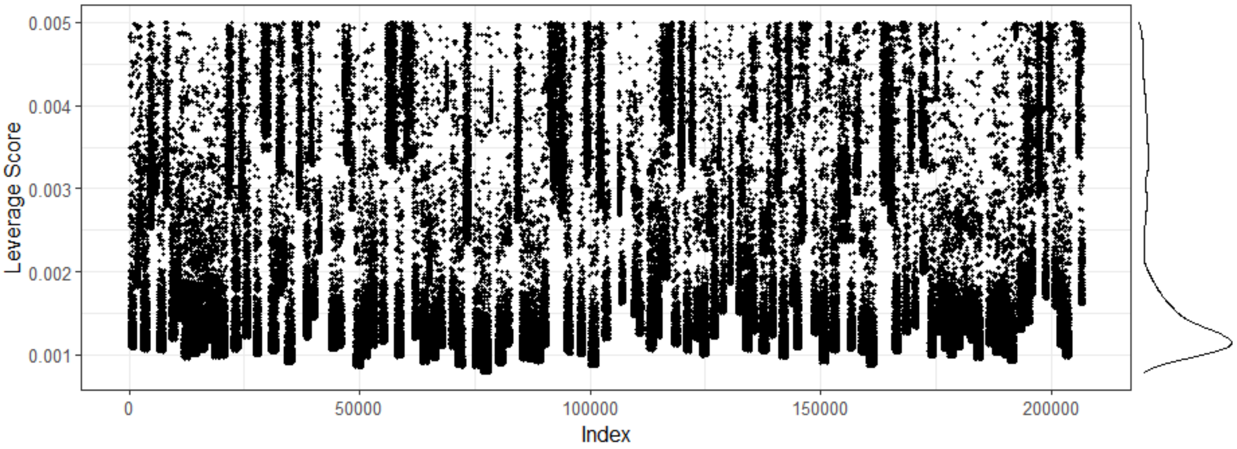


Table 5: Leverage Score Reanalysis

	≤ 1	≤ 0.05	≤ 0.01	≤ 0.005
	(1)	(2)	(3)	(4)
$\text{Temperature}_t/1,000$	-1.038^{***}	-1.051^{***}	-1.147^{***}	-1.052^{**}
	(0.327)	(0.326)	(0.362)	(0.420)
F-Statistic for Weather Variables	3.587	3.557	3.202	3.739
p-value	0.00172	0.00187	0.00473	0.00164
Observations	207,207	206,045	187,903	159,706
Note:				*p<0.1; **p<0.05; ***p<0.01

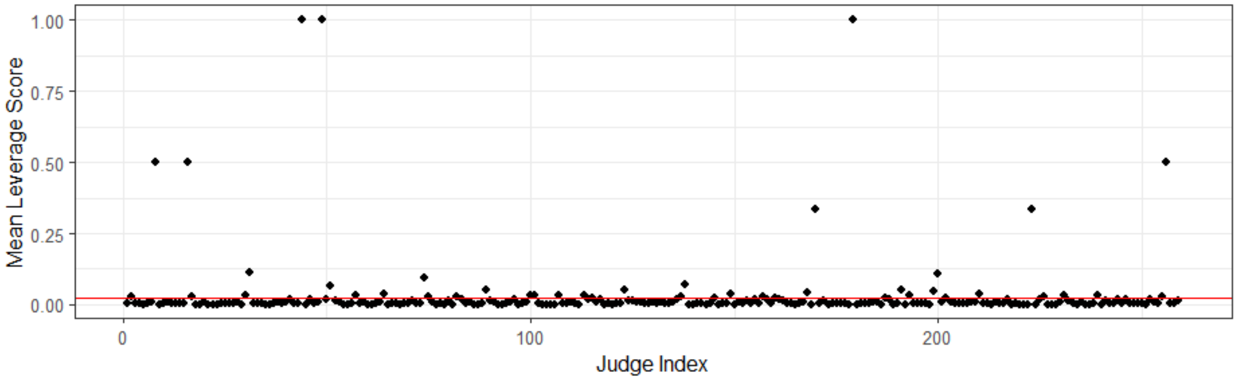
4.2 Leverage Score Analysis (Cluster Level)

Looking at Figure 4, one might notice the vertical groupings of leverage scores between sections in the indexing. The observations are grouped by judges and one can see that some judges have higher or lower minimums for their leverages. This motivated us to consider dropping groups of observations rather than only individual observations that have high leverage. Perhaps an entire group of observations impacts the results rather than one particular observation from a specific judge or a specific city. We plot the average leverage score from grouped data in three different groupings, the identity of a judge, the country of origin for each case, and lastly, the city in which the courthouse is located, which can be seen in Figure 5, 6, and 7. Using the same process as before, we set a threshold for too large of a leverage score, drop the observations that belong in that subset, and rerun the analysis. Our results are reported in Table 6. Once again, the authors' results are highly stable, and even removing highly influential groups of observations do not greatly affect the coefficient of interest.

Table 6: Grouped Leverage Score Reanalysis

	Judges	Nationality	Courthouse City
	(1)	(2)	(3)
$\text{Temperature}_t/1,000$	-1.092^{***}	-1.084^{***}	-1.109^{***}
	(0.335)	(0.325)	(0.338)
F-Statistic for Weather Variables	3.493	3.661	3.586
p-value	0.00217	0.00143	0.00183
Observations	202,998	205,815	202,288
Note:			*p<0.1; **p<0.05; ***p<0.01

Figure 5: Mean Leverage Scores grouped by Judge. Cutoff at 0.025



4.3 Logit Model

We re-analyze the author's assumption of a linear probability model and see whether their results are robust to a different assumption regarding the relationship between covariates and probability. Using a Logit model with the same preferred specification used in the authors' main result, we obtain a coefficient of -9.802 for daily average temperature. A logit model assumes that the relationship between the coefficients and the log-odds of the outcome is linear, hence we must convert this to the average marginal effect of temperature to compare this to the authors' results. After computing this value, we obtain an average marginal effect of -1.0972, with a standard error of 0.2267, which is slightly larger in magnitude and more significant when compared to the authors' result of a point estimate of -1.038 and a standard error of 0.326. However, this is still quite close, indicating that the authors' results are quite robust.

Code Appendix

[3]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', 50)
```

[4]:

```
data = pd.read_stata('matched.dta')
```

[5]:

```
temp = ['l2avgtemp', 'l3avgtemp', 'lavgtemp',  'temp6t4',  'le1avgtemp',␣
 ↪'le2avgtemp',
        'le3avgtemp', 'le3temp6t4', 'le2temp6t4',  'letemp6t4', 'ltemp6t4',␣
 ↪'l2temp6t4',
        'l3temp6t4', 'avgtemp', 'yearaftertemp', 'gtemp', 'yeartemp', 'heat']

for col in temp:
    new_col = col + str(10)
    data[new_col] = data[col]/10


weatherdaily =  ['avgtemp10', 'skycover', 'pressureavgsealevel', 'windspeed',␣
 ↪'precipitationwaterequiv', 'avgdewpt']
weatherdailyt = ['skycover', 'avgdewpt', 'pressureavgsealevel', 'windspeed',␣
 ↪'precipitationwaterequiv']
weathertemp =   ['press6t4', 'dew6t4', 'prcp6t4', 'wind6t4', 'skycover']
weather6t4 =    ['temp6t410', 'press6t4', 'dew6t4', 'prcp6t4', 'wind6t4',␣
 ↪'skycover' ]
heat =          ['heat10', 'press6t4', 'prcp6t4', 'wind6t4', 'skycover']
dailyheat =     ['dailyheat', 'skycover', 'pressureavgsealevel', 'windspeed',␣
 ↪'precipitationwaterequiv' ]
dummies =       ['dayofweek', ' nati', 'type', 'year', 'cm', 'chair']
pollutants =    ['ozone', 'co', 'pm25']
```

[ ]:

```
dropped_data = data.copy().dropna(how = 'any', subset = weather6t4 + pollutants)
```

[7]:

```
sum_stat_columns = ['res', 'temp6t4', 'heat'] + weathertemp + pollutants
mean = dropped_data[sum_stat_columns].mean()
sd = dropped_data[sum_stat_columns].std()
```

```
new_index = ['Grant Indicator',
             'Temperature ($^{\circ}$F)',
             'Heat Index ($^{\circ}$F)',
             'Air Pressure (pa)',
             'Dew point ($^{\circ}$F)',
             'Precipitation (mm)',
             'Wind speed (km/h)',
             'Sky cover (percent)',
             'Ozone (ppm)',
             'CO (ppm)',
             'PM$_{2.5}$ ($\mu$ / $m^3$)']
```

[8]:

```
sum_stat_table = pd.DataFrame({'Mean':mean.values, 'Standard Deviation':sd.
 ↪values}, index = new_index)
```

[9]:

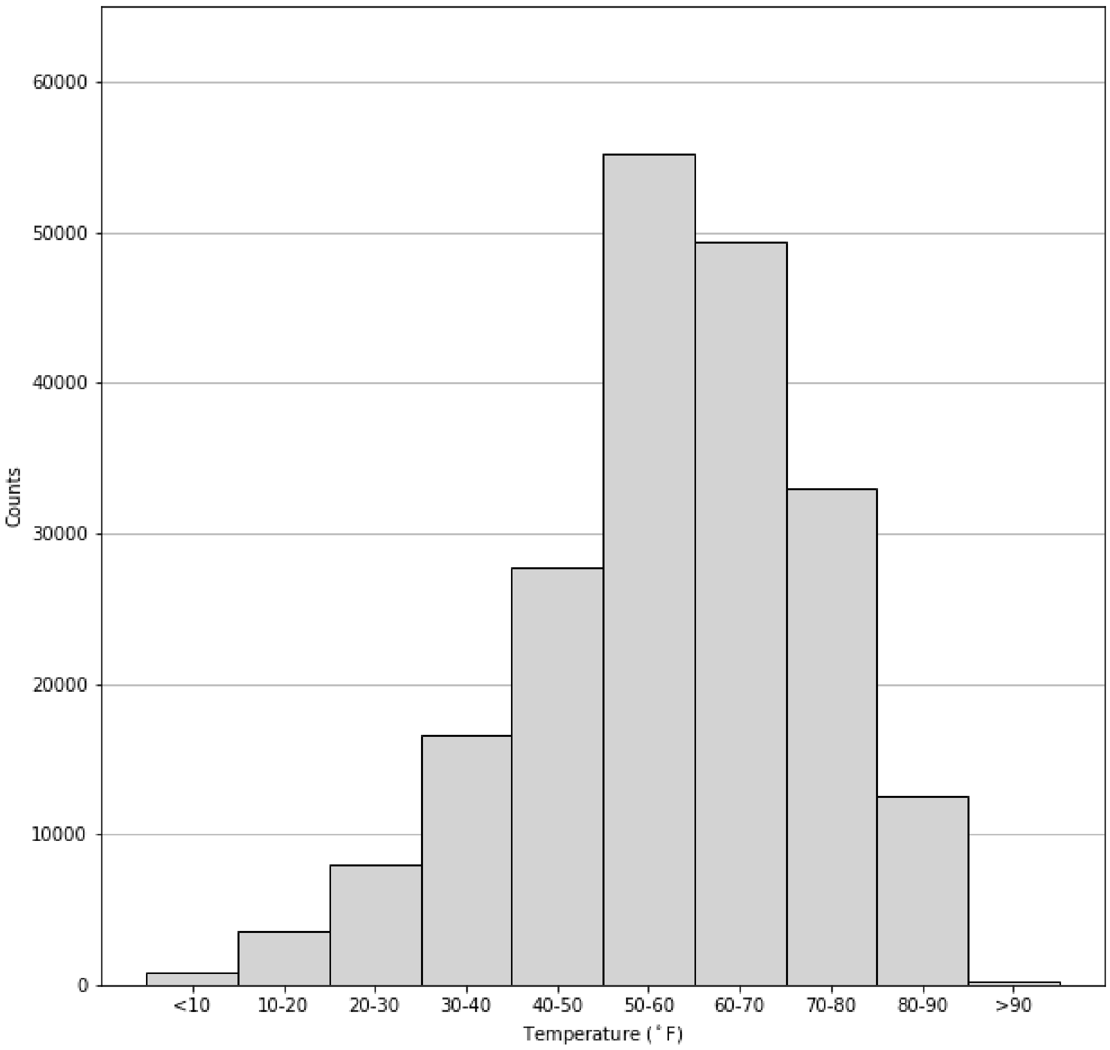
```
sum_stat_table
```

[9]:

```
                                  Mean  Standard Deviation
Grant Indicator               0.164261            0.370854
Temperature ($^{\circ}$F)    57.370564           15.720450
Heat Index ($^{\circ}$F)     57.776386           16.411543
Air Pressure (pa)            29.692633            0.759302
Dew point ($^{\circ}$F)      49.371086           17.202080
Precipitation (mm)            0.002822            0.013569
Wind speed (km/h)             4.556642            3.440875
Sky cover (percent)           0.554441            0.275636
Ozone (ppm)                   0.021971            0.011932
CO (ppm)                      0.917596            0.496442
PM$_{2.5}$ ($\mu$ / $m^3$)   14.956901           11.570034
```

[12]:

```
paper_fig3, ax = plt.subplots(figsize = (10,10))
ax.grid(axis = 'y', zorder=0)
ax.hist(dropped_data['temp6t4'], bins = np.arange(0,110,10), color =␣
 ↪'lightgrey', edgecolor = 'black', zorder=3)
ax.set_xlabel("Temperature ($^\circ$F)")
ax.set_xticks(np.arange(5,100,10))
ax.set_xticklabels(['<10'] + [str(i) + '-' + str(i+10) for i in range(10, 90,␣
 ↪10)] + ['>90'])
ax.set_ylabel("Counts")
ax.set_ylim([0, 65000]);
paper_fig3.savefig('paper_fig3.jpg')
```



60000
50000
40000
30000
20000
10000
0
Counts
<10
10-20
20-30
30-40
40-50
50-60
60-70
70-80
80-90
>90
Temperature ($^\circ$F)

Code Appendix (R)

```r
reg_eq = function(dependent, independent, fixed_effects, clustering) {
    independent_str = paste(independent, collapse = " + ")
    fixed_effects_str = paste(fixed_effects, collapse = " + ")
    eqn = paste(dependent, " ~ ", independent_str, "|", fixed_effects_str,
        "| 0 |", clustering)
    return(as.formula(eqn))
}
```

```r
data = read.csv("regression_data.csv")
```

```r
weatherdaily = c("avgtemp10", "skycover", "pressureavgsealevel",
    "windspeed", "precipitationwaterequiv", "avgdewpt")
weatherdailyt = c("skycover", "avgdewpt", "pressureavgsealevel",
    "windspeed", "precipitationwaterequiv")
weathertemp = c("press6t4", "dew6t4", "prcp6t4", "wind6t4", "skycover")
weather6t4 = c("temp6t410", "press6t4", "dew6t4", "prcp6t4",
    "wind6t4", "skycover")
heat = c("heat10", "press6t4", "prcp6t4", "wind6t4", "skycover")
dailyheat = c("dailyheat", "skycover", "pressureavgsealevel",
    "windspeed", "precipitationwaterequiv")
dummies = c("dayofweek", "nat_name", "c_asy_type", "year", "cm",
    "chair")
pollutants = c("ozone", "co", "pm25")
```

```r
# qui xi: reg res $weatherdaily $pollutants $dummies , vce
# (cluster cm)
base_dev = felm(reg_eq("res", c(weatherdaily, pollutants), dummies,
    "cm"), data = data)


# qui xi: reg res $weather6t4 $pollutants $dummies , vce
# (cluster cm)
base_6t4 = felm(reg_eq("res", c("deviations", weather6t4, pollutants),
    dummies, "cm"), data = data)


# qui xi: reg res ltemp6t410 $weather6t4 $dummies
# $pollutants , vce (cluster cm)
lag_6t4 = felm(reg_eq("res", c("ltemp6t410", weather6t4, pollutants),
    dummies, "cm"), data = data)


# qui xi: reg res letemp6t410 $weather6t4 $dummies
# $pollutants , vce (cluster cm)
lead_6t4 = felm(reg_eq("res", c("letemp6t410", weather6t4, pollutants),
    dummies, "cm"), data = data)


# qui xi: reg res ltemp6t410 temp6t410 letemp6t410 press6t4
# dew6t4 prcp6t4 wind6t4 skycover $dummies $pollutants ,
# vce (cluster cm)
all_6t4_one = felm(reg_eq("res", c("ltemp6t410", "letemp6t410",
    weather6t4, pollutants), dummies, "cm"), data = data)
```

```
# function to get F-Stat for weather variables as described
# in paper
get_waldstat_F <- function(model, variables) (round(waldtest(model,
    variables)["F"], 3))


# function to get pvals for F-Stat for weather variables as
# described in paper
get_waldstat_pF <- function(model, variables) (round(waldtest(model,
    variables)["p.F"], 5))
```

```
# esttab base_6t4 lag_6t4 lead_6t4 all_6t4_one using
# base_6t4.tex,replace keep(ltemp6t410 temp6t410
# letemp6t410 ) se brackets star(* 0.10 ** 0.05 *** 0.01
# mtitles('base' '1-Day lag' '1-Day lead' 'all')\t


stargazer(base_6t4, lag_6t4, lead_6t4, all_6t4_one, dep.var.labels = "Result",
    covariate.labels = c("Temperature$_t/1,000$", "Temperature$_{t-1}/1,000$",
        "Temperature$_{t+1}/1,000$"), column.labels = c("Base",
        "1 Day Lag", "1 Day Lead", "1 Day Lag and Lead"), type = "latex",
    keep = c(".*temp.*"), keep.stat = c("n"), order = c(4, 2,
        3), add.lines = list(c("F-Statistic for Weather Variables",
        c(get_waldstat_F(base_6t4, weather6t4), get_waldstat_F(lag_6t4,
            c("ltemp6t410", weather6t4)), get_waldstat_F(lead_6t4,
            c("letemp6t410", weather6t4)), get_waldstat_F(all_6t4_one,
            c("ltemp6t410", "letemp6t410", weather6t4)))), c("p-value",
        c(get_waldstat_pF(base_6t4, weather6t4), get_waldstat_pF(lag_6t4,
            c("ltemp6t410", weather6t4)), get_waldstat_pF(lead_6t4,
            c("letemp6t410", weather6t4)), get_waldstat_pF(all_6t4_one,
            c("ltemp6t410", "letemp6t410", weather6t4))))))
```

```
# qui xi: reg res l3avgtemp10 l2avgtemp10 lavgtemp10
# temp6t410 le1avgtemp10 le2avgtemp10 le3avgtemp10 press6t4
# dew6t4 prcp6t4 wind6t4 $dummies $pollutants , vce
# (cluster cm)
all_6t4_one = felm(reg_eq("res", c("l3avgtemp10", "l2avgtemp10",
    "lavgtemp10", "temp6t410", "le1avgtemp10", "le2avgtemp10",
    "le3avgtemp10", "press6t4", "dew6t4", "prcp6t4", "wind6t4",
    pollutants), dummies, "cm"), data = data)


figure_5_pt = summary(all_6t4_one)$coefficients[1:7, 1]
figure_5_se = summary(all_6t4_one)$coefficients[1:7, 2]
```

```
figure_5 = ggplot() + geom_line(aes(x = 1:7, y = figure_5_pt),
    col = "red") + geom_point(aes(x = 1:7, y = figure_5_pt)) +
    geom_errorbar(aes(x = 1:7, y = figure_5_pt, ymin = figure_5_pt -
        1.95 * figure_5_se, ymax = figure_5_pt + 1.95 * figure_5_se),
        width = 0.2, position = position_dodge(0.05)) + geom_hline(yintercept = 0,
    linetype = "dotted", col = "red") + theme_bw() + labs(x = "Lagged/Leaded Temperature",
    y = "Point Estimates") + scale_x_continuous(breaks = 1:7,
    labels = c(-3:3))



figure_5
```

```
alt_dummies = c("nat_name", "dayofweek", "c_asy_type", "chair",
    "cm", "jm", "city", "year", "ym", "date")
# qui reg res $weather6t4 $pollutants , vce (cluster cm)
base_6t4_nothing_dummies = 0
base_6t4_nothing = felm(reg_eq("res", c(weather6t4, pollutants),
    base_6t4_nothing_dummies, "cm"), data = data)


# qui reg res $weather6t4 $pollutants $dummies
base_1_dummies = dummies
base_1 = felm(reg_eq("res", c(weather6t4, pollutants), base_1_dummies,
    0), data = data)


# qui reg res $weather6t4 $pollutants
base_2_dummies = 0
base_2 = felm(reg_eq("res", c(weather6t4, pollutants), base_2_dummies,
    0), data = data)


# qui reg res $weather6t4 $pollutants i.nati , cluster (cm)
base_6t4_nati_dummies = c("nat_name")
base_6t4_nati = felm(reg_eq("res", c(weather6t4, pollutants),
    base_6t4_nati_dummies, "cm"), data = data)


# qui reg res $weather6t4 $pollutants i.nati i.dayofweek ,
# cluster (cm)
base_6t4_nati_dow_dummies = c("nat_name", "dayofweek")
base_6t4_nati_dow = felm(reg_eq("res", c(weather6t4, pollutants),
    base_6t4_nati_dow_dummies, "cm"), data = data)


# qui reg res $weather6t4 $pollutants i.nati i.type
# i.dayofweek , cluster (cm)
base_6t4_nati_dow_type_dummies = c("nat_name", "c_asy_type",
    "dayofweek")
base_6t4_nati_dow_type = felm(reg_eq("res", c(weather6t4, pollutants),
    base_6t4_nati_dow_type_dummies, "cm"), data = data)
```

```r
# qui reg res $weather6t4 $pollutants i.nati i.type
# i.dayofweek i.chair , cluster (cm)
base_6t4_nati_dow_type_j_dummies = c("nat_name", "c_asy_type",
    "dayofweek", "chair")
base_6t4_nati_dow_type_j = felm(reg_eq("res", c(weather6t4, pollutants),
    base_6t4_nati_dow_type_j_dummies, "cm"), data = data)


# qui reg res $weather6t4 $pollutants i.nati i.type
# i.dayofweek i.chair i.cm , cluster (cm)
base_6t4_nati_dow_type_j_cm_dummies = c("nat_name", "c_asy_type",
    "dayofweek", "chair", "cm")
base_6t4_nati_dow_type_j_cm = felm(reg_eq("res", c(weather6t4,
    pollutants), base_6t4_nati_dow_type_j_cm_dummies, "cm"),
    data = data)


# qui reg res $weather6t4 $pollutants i.nati i.type
# i.dayofweek i.chair i.ct i.ym , cluster (cm)
base_6t4_city_ym_dummies = c("nat_name", "c_asy_type", "dayofweek",
    "chair", "city", "ym")
base_6t4_city_ym = felm(reg_eq("res", c(weather6t4, pollutants),
    base_6t4_city_ym_dummies, "cm"), data = data)


# qui reg res $weather6t4 $pollutants i.nati i.type
# i.dayofweek i.chair i.cym , cluster (cm)
base_6t4_cym_dummies = c("nat_name", "c_asy_type", "dayofweek",
    "chair", "cym")
base_6t4_cym = felm(reg_eq("res", c(weather6t4, pollutants),
    base_6t4_cym_dummies, "cm"), data = data)


# qui reg res $weather6t4 $pollutants i.nati i.type
# i.dayofweek i.chair i.jm i.ct i.year , cluster (cm)
base_6t4_jm_ct_year_dummies = c("nat_name", "c_asy_type", "dayofweek",
    "jm", "city", "year")
base_6t4_jm_ct_year = felm(reg_eq("res", c(weather6t4, pollutants),
    base_6t4_jm_ct_year_dummies, "cm"), data = data)


# qui reg res $weather6t4 $pollutants i.nati i.type i.chair
# i.cm i.year i.date, cluster (cm)
base_6t4_date_dummies = c("date", "nat_name", "c_asy_type", "year",
    "cm", "chair")
base_6t4_date = felm(reg_eq("res", c(weather6t4, pollutants),
    base_6t4_date_dummies, "cm"), data = data)

# they don't have cym for some reason
models_afe = list(base_6t4_nati, base_6t4_nati_dow, base_6t4_nati_dow_type,
    base_6t4_nati_dow_type_j, base_6t4_nati_dow_type_j_cm, base_6t4_city_ym,
    base_6t4_jm_ct_year, base_6t4_date)
```

```r
get_indicator <- function(dummies) {
    output = c()
    for (cov in alt_dummies) {
        indicator = cov %in% dummies
        output = c(output, indicator)
    }
    return(output)
}

dummies_list = list(base_6t4_nati_dummies, base_6t4_nati_dow_dummies,
    base_6t4_nati_dow_type_dummies, base_6t4_nati_dow_type_j_dummies,
    base_6t4_nati_dow_type_j_cm_dummies, base_6t4_city_ym_dummies,
    base_6t4_jm_ct_year_dummies, base_6t4_date_dummies)
fe_indicators = NULL
for (afe in dummies_list) {
    output = get_indicator(afe)
    fe_indicators = rbind(fe_indicators, output)
}

fe_indicators = t(fe_indicators)

fe_vars = c("Nationality FEs", "Day of Week FEs", "Type of Application FEs",
    "Judge FEs", "City-Month FEs", "Judge-Month FEs", "City FEs",
    "Year FEs", "Year-Month FEs", "Date FEs")

stargazer_addlines = NULL

for (i in 1:length(fe_vars)) {
    stargazer_addlines[[i]] = c(fe_vars[i], fe_indicators[i,
        ])
}

stargazer(models_afe, dep.var.labels = "Result", covariate.labels = c("Temperature"),
    apply.coef = multiply.by.100, apply.se = multiply.by.100,
    type = "latex", keep = c("temp6t410"), keep.stat = c("n"),
    add.lines = stargazer_addlines)
```

```r
# qui reg res $weather6t4 $pollutants $dummies
no_pollution = felm(reg_eq("res", weather6t4, dummies, "cm"),
    data = data)


# Not California

CA_cities = c("SAN PEDRO", "SAN FRANCISCO", "SAN DIEGO", "LOS ANGELES",
    "LAS VEGAS", "LANCASTER", "IMPERIAL")
data$CA = data$city %in% CA_cities
no_CA = felm(reg_eq("res", c("deviations", weather6t4, pollutants),
    dummies, "cm"), data = subset(data, CA == FALSE))




# quietly reg res $weather6t4 $dummies $pollutants if
# skycover<0.1 , vce (cluster cm)

clearsky = felm(reg_eq("res", c("deviations", weather6t4, pollutants),
    dummies, "cm"), data = subset(data, skycover < 0.1))


# quietly reg res $weather6t4 $dummies $pollutants if
# precipitationwaterequiv==0 , vce (cluster cm)

norain = felm(reg_eq("res", c("deviations", weather6t4, pollutants),
    dummies, "cm"), data = subset(data, precipitationwaterequiv ==
    0))


# quietly reg res $weather6t4 $dummies $pollutants if
# precipitationwaterequiv==0 & lprcp==0, vce (cluster cm)

norain2 = felm(reg_eq("res", c("deviations", weather6t4, pollutants),
    dummies, "cm"), data = subset(data, precipitationwaterequiv ==
    0 & lprcp == 0))


# qui reg res $heat $pollutants $dummies, cluster (cm)

heat_6t4 = felm(reg_eq("res", c(heat, pollutants), dummies, "cm"),
    data = data)


# qui reg res $heat $dummies $pollutants if heat10>=0.075 ,
# cluster (cm)

heat_75 = felm(reg_eq("res", c(heat, pollutants), dummies, "cm"),
    data = subset(data, heat10 > 0.075))


# judges

judge_means = aggregate(data[, c("res", "ij_name")], list(data$ij_name),
    mean)[, 2:3]
names(judge_means)[1] = "rate"
data = merge(data, judge_means, by = "ij_name")
judge_cutoffs = quantile(judge_means$rate, c(0.1, 0.25, 0.75,
    0.9))


judge_quartile = felm(reg_eq("res", c("deviations", weather6t4,
    pollutants), dummies, "cm"), data = subset(data, rate > judge_cutoffs[2] &
    rate <= judge_cutoffs[3]))

judge_decile = felm(reg_eq("res", c("deviations", weather6t4,
    pollutants), dummies, "cm"), data = subset(data, rate > judge_cutoffs[1] &
    rate <= judge_cutoffs[4]))
```

```
models_robust = list(no_pollution, no_CA, clearsky, norain, norain2,
    heat_6t4, heat_75, judge_quartile, judge_decile)


stargazer(models_robust, dep.var.labels = "Result", covariate.labels = c("Temperature",
    "Heat Index"), type = "latex", keep = c("temp6t410", "heat10"),
    keep.stat = c("n"))
```

```
data = data %>%
    mutate(across(all_of(dummies), as.factor))
lm_model = lm(res ~ ., data = subset(data, select = c("res",
    "deviations", weather6t4, pollutants, dummies)))
hv = as.data.frame(hatvalues(lm_model))
names(hv)[1] = "leverage_scores"
```

```
hv$X = as.integer(rownames(hv))
leverage_plot = ggplot() + geom_point(aes(x = hv$X, y = hv$leverage_scores),
    size = 0.5) + labs(y = "Leverage Score", x = "Index") + theme_bw()
leverage_plot = ggMarginal(leverage_plot, margins = "y", size = 10)
leverage_plot
```

```
hv_subset = subset(hv, leverage_scores < 0.05)
leverage_plot2 = ggplot() + geom_point(aes(x = hv_subset$X, y = hv_subset$leverage_scores),
    size = 0.5) + labs(y = "Leverage Score", x = "Index") + theme_bw()
leverage_plot2 = ggMarginal(leverage_plot2, margins = "y", size = 10)
leverage_plot2
```

```
hv_subset = subset(hv, leverage_scores < 0.01)
leverage_plot3 = ggplot() + geom_point(aes(x = hv_subset$X, y = hv_subset$leverage_scores),
    size = 0.5) + labs(y = "Leverage Score", x = "Index") + theme_bw()
leverage_plot3 = ggMarginal(leverage_plot3, margins = "y", size = 10)
leverage_plot3
```

```
hv_subset = subset(hv, leverage_scores < 0.005)
leverage_plot4 = ggplot() + geom_point(aes(x = hv_subset$X, y = hv_subset$leverage_scores),
    size = 0.5) + labs(y = "Leverage Score", x = "Index") + theme_bw()
leverage_plot4 = ggMarginal(leverage_plot4, margins = "y", size = 10)
leverage_plot4
```

```
data$leverage_scores = hv$leverage_scores

dropped_leverage_0 = felm(reg_eq('res', c('deviations', weather6t4, pollutants), dummies,
                                 'cm'), data = subset(data, leverage_scores <= 1))

dropped_leverage_1 = felm(reg_eq('res', c('deviations', weather6t4, pollutants), dummies,
                                 'cm'), data = subset(data, leverage_scores <= 0.05))

dropped_leverage_2 = felm(reg_eq('res', c('deviations', weather6t4, pollutants), dummies,
                                 'cm'), data = subset(data, leverage_scores <= 0.01))

dropped_leverage_3 = felm(reg_eq('res', c('deviations', weather6t4, pollutants), dummies,
                                 'cm'), data = subset(data, leverage_scores <= 0.005))


stargazer(dropped_leverage_0, dropped_leverage_1, dropped_leverage_2, dropped_leverage_3,
          dep.var.labels = 'Result',
          covariate.labels = c('Temperature$_t/1,000$'),
          column.labels = c("<= 1","<= 0.05", "<= 0.01", "<= 0.005"),
          type = "latex",
          keep = c(".*temp.*"),
          keep.stat = c('n'),
          add.lines = list(c("F-Statistic for Weather Variables",
                             c(get_waldstat_F(dropped_leverage_0, weather6t4),
                               get_waldstat_F(dropped_leverage_1, weather6t4),
                               get_waldstat_F(dropped_leverage_2, weather6t4),
                               get_waldstat_F(dropped_leverage_3, weather6t4))),
                           c("p-value",
                             c(get_waldstat_pF(dropped_leverage_0, weather6t4),
                               get_waldstat_pF(dropped_leverage_1, weather6t4),
                               get_waldstat_pF(dropped_leverage_2, weather6t4),
                               get_waldstat_pF(dropped_leverage_3, weather6t4)))

          )
)
```

```
data$leverage_scores = hv$leverage_scores
cutoffs = c(1, 0.05, 0.01, 0.005)

dropped_obs = c()
point_estimates = c()
standard_errors = c()

for (cutoff in cutoffs) {
    dropped_data = subset(data, leverage_scores <= cutoff)
    dropped_obs = c(dropped_obs, nrow(data) - nrow(dropped_data))
    base_6t4 = felm(reg_eq("res", c("deviations", weather6t4,
        pollutants), dummies, "cm"), data = dropped_data)
    point_estimates = c(point_estimates, summary(base_6t4)$coefficients[2,
        1])
    standard_errors = c(standard_errors, summary(base_6t4)$coefficients[2,
        2])
}


xlabels = c()

for (i in 1:4) {
    label = paste("<= ", cutoffs[i], " (", dropped_obs[i], ")")
    xlabels = c(xlabels, label)
}

leverage_results = ggplot() + geom_line(aes(x = 1:4, y = point_estimates),
    col = "red") + geom_point(aes(x = 1:4, y = point_estimates),
    size = 4) + geom_errorbar(aes(x = 1:4, y = point_estimates,
    ymin = point_estimates - 1.95 * standard_errors, ymax = point_estimates +
        1.95 * standard_errors), width = 0.1, position = position_dodge(0.05)) +
    geom_hline(yintercept = 0, linetype = "dotted", col = "red") +
    theme_bw() + labs(x = "Leverage Score Cutoff (Dropped Observations)",
    y = "Point Estimates") + scale_x_continuous(breaks = 1:4,
    labels = xlabels, limits = c(0.8, 4.2))

leverage_results
```

```
data = data %>%
    mutate(across(all_of(dummies), as.factor))


logit_model = glm(res ~ ., data = subset(data, select = c("res",
    "deviations", weather6t4, pollutants, dummies)), family = binomial(link = logit))

saveRDS(logit_model, file = "logit_fit.rds")

summary(logit_model)

stargazer(logit_model, dep.var.labels = "Result", covariate.labels = c("Temperature"),
    type = "text", keep = c("temp6t410"), keep.stat = c("n"))
```

```
ape = margins(logit_model, variables = "temp6t410")
summary(ape)
```

```
judge_avg_leverage = aggregate(data[, c("leverage_scores")],
    list(data$ij_name), mean)
colnames(judge_avg_leverage)[1] = "judge"
colnames(judge_avg_leverage)[2] = "mean_leverage"

ggplot(data = judge_avg_leverage, aes(x = judge, y = mean_leverage)) +
    geom_point() + geom_hline(yintercept = 0.025, color = "red") +
    labs(x = "Judge Index", y = "Mean Leverage Score") + theme_bw()
```

```
origin_avg_leverage = aggregate(data[, c("leverage_scores")],
    list(data$nat_name), mean)
colnames(origin_avg_leverage)[1] = "origin"
colnames(origin_avg_leverage)[2] = "mean_leverage"

ggplot() + geom_point(aes(x = as.integer(rownames(origin_avg_leverage)),
    y = origin_avg_leverage$mean_leverage)) + labs(x = "Country Index",
    y = "Mean Leverage") + geom_hline(yintercept = 0.025, color = "red") +
    theme_bw()
```

```
city_avg_leverage = aggregate(data[, c("leverage_scores")], list(data$city),
    mean)
colnames(city_avg_leverage)[1] = "city"
colnames(city_avg_leverage)[2] = "mean_leverage"


ggplot() + geom_point(aes(x = as.integer(rownames(city_avg_leverage)),
    y = city_avg_leverage$mean_leverage)) + labs(x = "City Index",
    y = "Mean Leverage") + geom_hline(yintercept = 0.045, color = "red") +
    theme_bw()
```

```
low_lev_judges = subset(judge_avg_leverage, mean_leverage < 0.025)$judge
low_lev_origin = subset(origin_avg_leverage, mean_leverage < 0.025)$origin
low_lev_city = subset(city_avg_leverage, mean_leverage < 0.025)$city

judge_data = subset(data, ij_name %in% low_lev_judges)
origin_data = subset(data, nat_name %in% low_lev_origin)
city_data = subset(data, city %in% low_lev_city)




judge_model = felm(reg_eq('res', c('deviations', weather6t4, pollutants), dummies, 'cm'),
                   data = judge_data)

origin_model = felm(reg_eq('res', c('deviations', weather6t4, pollutants), dummies, 'cm'),
                    data = origin_data)

city_model = felm(reg_eq('res', c('deviations', weather6t4, pollutants), dummies, 'cm'),
                  data = city_data)

stargazer(judge_model, origin_model, city_model,
          dep.var.labels = 'Result',
          covariate.labels = c('Temperature$_t/1,000$'),
          column.labels = c("Judges","Nationality", "Courthouse City"),
          type = "latex",
          keep = c(".*temp.*"),
          keep.stat = c('n'),
          add.lines = list(c("F-Statistic for Weather Variables",
                             c(get_waldstat_F(judge_model, weather6t4),
                               get_waldstat_F(origin_model, weather6t4),
                               get_waldstat_F(city_model,weather6t4))),
                           c("p-value",
                             c(get_waldstat_pF(judge_model, weather6t4),
                               get_waldstat_pF(origin_model, weather6t4),
                               get_waldstat_pF(city_model, weather6t4)))

          )
)
```

Figure 6: Mean Leverage Scores grouped by County of Origin. Cutoff at 0.025

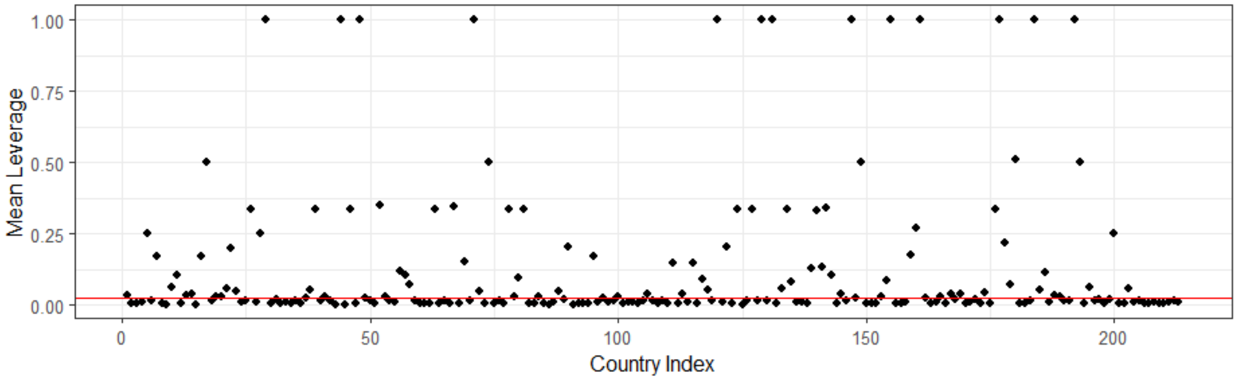
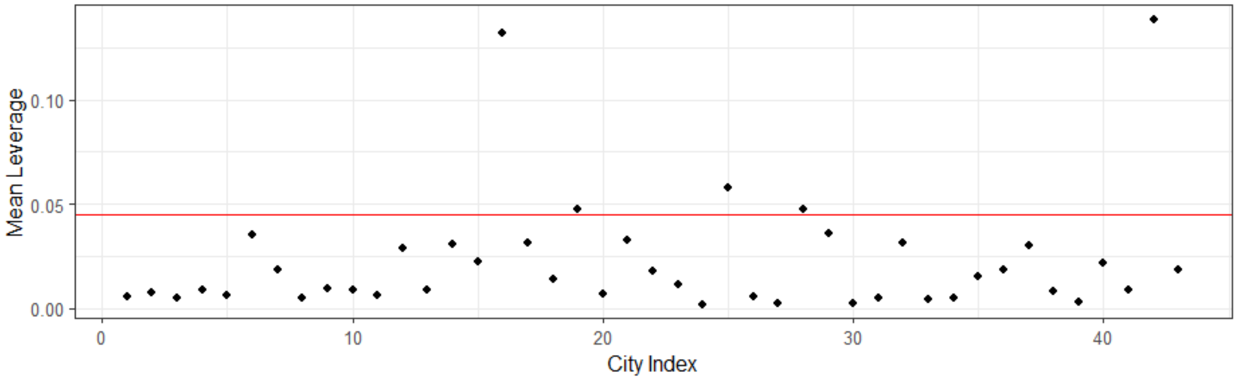


Figure 7: Mean Leverage Scores grouped by City. Cutoff at 0.045


Mean Leverage
City Index