ga6

December 6, 2018

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import os
        from datetime import datetime
        import re
        from clean_data import get_filenames, get_clean_data
        from Assignment6_functions import get_dummy_features, get_permute, get_prcp_features, g
        import statsmodels.api as sm
        from cryptorandom.cryptorandom import SHA256
```

0.0.1 Please do not repeatedly run the code chuck below, reading and cleaning data is very time consuming.

```
In [ ]: #please refer to clean_data.py for more detail on function get_clean_data
        #please make sure the raw data (.dta folder) are in the same directory with this noteb
        df = get_clean_data()
        df_west = get_clean_data(selected_area='west')
        df_east = get_clean_data(selected_area='east')
```

Poisson Model for the Whole Alameda

```
In [2]: # get the data
        TMAX_data = pd.read_csv('../group_assignment3/TMAX_data.csv')
        # get the temp bins
        temp_bins = get_temp_features(TMAX_data)
        temp_bins.head()
```

Out[2]:

	30-39F	40-49F	50-59F	60-69F	70-79F	80-89F	90-99F	>100F
0	0.0	0.0	34.0	26.0	0.0	0.0	0.0	0.0
1	0.0	0.0	16.0	41.0	3.0	0.0	0.0	0.0
2	0.0	0.0	12.0	36.0	13.0	0.0	0.0	0.0
3	0.0	0.0	8.0	33.0	18.0	2.0	0.0	0.0
4	0.0	0.0	4.0	31.0	21.0	4.0	1.0	0.0

```
In [3]: # get the data
        PRCP_data = pd.read_csv("../group_assignment3/PRCP_data.csv")
        # get the prcp bins
```

```
prcp_bins = get_prcp_features_whole(PRCP_data)
prcp_bins.head()
```

Out[3]:

	0mm	1-4mm	5-14mm	15-29mm	>30mm
0	29.0	13.0	10.0	7.0	1.0
1	32.0	14.0	9.0	4.0	1.0
2	34.0	22.0	4.0	1.0	0.0
3	43.0	16.0	1.0	1.0	0.0
4	54.0	6.0	1.0	0.0	0.0

In [4]:
```
# dummy variables theta phi
theta,phi = get_dummy_features(TMAX_data)
```

In [5]:
```
# get independent variables
variables = pd.concat([temp_bins, prcp_bins, theta, phi], axis=1)
variables.head()

# get the response variable
df = pd.read_csv("all_alameda_crime.csv").iloc[:,1:3]
crime = df['crime_sum']
crime = crime.iloc[1:].reset_index(drop=True)

# fit the model
poisson_model = sm.GLM(crime,variables,family=sm.families.Poisson())
poisson_results = poisson_model.fit()

# show result
poisson_results.summary()
```

Out[5]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:              crime_sum   No. Observations:                  359
Model:                            GLM   Df Residuals:                      306
Model Family:                 Poisson   Df Model:                           52
Link Function:                    log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -11604.
Date:                Thu, 06 Dec 2018   Deviance:                       19300.
Time:                        10:21:54   Pearson chi2:                 2.08e+04
No. Iterations:                     9   Covariance Type:             nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
30-39F        -0.0148      0.003     -5.422      0.000      -0.020      -0.009
40-49F        -0.0053      0.001     -3.614      0.000      -0.008      -0.002
50-59F        -0.0007      0.001     -0.518      0.604      -0.004       0.002
60-69F        -0.0032      0.001     -2.289      0.022      -0.006      -0.000
70-79F        -0.0005      0.001     -0.334      0.739      -0.003       0.002
```

80-89F	-0.0010	0.001	-0.723	0.470	-0.004	0.002
90-99F	-0.0061	0.001	-4.195	0.000	-0.009	-0.003
>100F	0.0132	0.003	5.116	0.000	0.008	0.018
0mm	0.0006	0.002	0.292	0.770	-0.004	0.005
1-4mm	8.164e-06	0.002	0.004	0.997	-0.004	0.004
5-14mm	0.0001	0.002	0.061	0.952	-0.004	0.004
15-29mm	0.0024	0.002	1.107	0.268	-0.002	0.007
>30mm	-0.0216	0.002	-8.716	0.000	-0.026	-0.017
1980	2.7439	0.062	44.194	0.000	2.622	2.866
1981	2.7630	0.061	44.933	0.000	2.642	2.884
1982	2.7182	0.062	44.096	0.000	2.597	2.839
1983	2.6374	0.062	42.842	0.000	2.517	2.758
1984	2.6576	0.062	42.703	0.000	2.536	2.780
1985	2.6891	0.062	43.608	0.000	2.568	2.810
1986	2.7536	0.061	44.810	0.000	2.633	2.874
1987	2.7309	0.062	44.366	0.000	2.610	2.852
1988	2.7681	0.062	44.533	0.000	2.646	2.890
1989	2.7625	0.062	44.880	0.000	2.642	2.883
1990	2.6832	0.062	43.583	0.000	2.563	2.804
1991	2.8034	0.062	45.490	0.000	2.683	2.924
1992	2.7888	0.062	44.833	0.000	2.667	2.911
1993	2.8039	0.062	45.500	0.000	2.683	2.925
1994	2.7527	0.062	44.651	0.000	2.632	2.874
1995	2.2525	0.062	36.558	0.000	2.132	2.373
1996	2.6882	0.062	43.215	0.000	2.566	2.810
1997	2.6610	0.062	43.229	0.000	2.540	2.782
1998	2.6237	0.062	42.544	0.000	2.503	2.745
1999	2.4715	0.062	40.023	0.000	2.350	2.593
2000	2.3849	0.062	38.302	0.000	2.263	2.507
2001	2.4605	0.062	39.893	0.000	2.340	2.581
2002	2.4972	0.062	40.502	0.000	2.376	2.618
2003	2.4843	0.062	40.261	0.000	2.363	2.605
2004	2.4456	0.062	39.259	0.000	2.323	2.568
2005	2.4302	0.062	39.426	0.000	2.309	2.551
2006	2.4917	0.062	40.375	0.000	2.371	2.613
2007	2.4650	0.062	40.001	0.000	2.344	2.586
2008	2.4385	0.062	39.166	0.000	2.316	2.561
2009	2.3989	0.062	38.895	0.000	2.278	2.520
Jan	6.5599	0.159	41.290	0.000	6.249	6.871
Feb	6.4786	0.149	43.489	0.000	6.187	6.771
Mar	6.5685	0.149	44.190	0.000	6.277	6.860
Apr	6.5283	0.155	42.226	0.000	6.225	6.831
May	6.5416	0.154	42.345	0.000	6.239	6.844
Jun	6.4934	0.154	42.039	0.000	6.191	6.796
Jul	6.5102	0.155	42.119	0.000	6.207	6.813
Aug	6.4967	0.158	41.067	0.000	6.187	6.807
Sep	6.4500	0.155	41.715	0.000	6.147	6.753
Oct	6.5238	0.154	42.231	0.000	6.221	6.827

```
Nov              6.5179      0.155     42.187      0.000       6.215       6.821
Dec              6.5813      0.155     42.468      0.000       6.278       6.885
===============================================================================
"""
```

Poisson Model for East Alameda

In [6]:

```python
TMAX_data_east = pd.read_csv('../group_assignment4/TMAX_data_east.csv').iloc[:,1:4]
# drop 20-29F because this is not a feature in other models
TMAX_data_east = TMAX_data_east.drop(TMAX_data_east[TMAX_data_east['temp_bins']=='20-29
# get temp bins
temp_bins = get_temp_features(TMAX_data_east)
temp_bins.head()
```

Out[6]:

	30-39F	40-49F	50-59F	60-69F	70-79F	80-89F	90-99F	>100F
0	0.0	8.0	40.0	12.0	0.0	0.0	0.0	0.0
1	0.0	0.0	31.0	28.0	1.0	0.0	0.0	0.0
2	0.0	0.0	17.0	32.0	9.0	3.0	0.0	0.0
3	0.0	0.0	8.0	26.0	20.0	6.0	1.0	0.0
4	0.0	0.0	2.0	21.0	24.0	11.0	3.0	0.0

In [7]:

```python
# get the data
PRCP_data_east = pd.read_csv("../group_assignment4/PRCP_data_east.csv").iloc[:,1:4]
# get prcp bins
prcp_bins = get_prcp_features(PRCP_data_east)
prcp_bins.head()
```

Out[7]:

	0mm	1-4mm	5-14mm
0	0.0	60.0	0.0
1	0.0	60.0	0.0
2	0.0	61.0	0.0
3	0.0	61.0	0.0
4	0.0	61.0	0.0

In [8]:

```python
# dummy variables theta phi
theta,phi = get_dummy_features(TMAX_data_east)
```

In [9]:

```python
# get independent variables
variables = pd.concat([temp_bins, prcp_bins, theta, phi], axis=1)

# get the response variable
df_east = pd.read_csv("east_alameda_crime.csv").iloc[:,1:3]
crime_east = df['crime_sum']
crime_east = crime_east.iloc[1:].reset_index(drop=True)

# fit the model
east_poisson_model = sm.GLM(crime_east,variables,family=sm.families.Poisson())
east_poisson_results = east_poisson_model.fit()
crime_east_hat = east_poisson_results.predict(variables)
# show result
east_poisson_results.summary()
```

```
Out[9]: <class 'statsmodels.iolib.summary.Summary'>
"""
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:              crime_sum   No. Observations:                  359
Model:                            GLM   Df Residuals:                      307
Model Family:                 Poisson   Df Model:                           51
Link Function:                    log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -11651.
Date:                Thu, 06 Dec 2018   Deviance:                       19394.
Time:                        10:21:54   Pearson chi2:                 2.08e+04
No. Iterations:                     4   Covariance Type:             nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
30-39F        -0.0030      0.002     -1.320      0.187      -0.008       0.001
40-49F        -0.0034      0.002     -1.708      0.088      -0.007       0.000
50-59F        -0.0013      0.002     -0.659      0.510      -0.005       0.003
60-69F        -0.0025      0.002     -1.264      0.206      -0.006       0.001
70-79F        -0.0017      0.002     -0.872      0.383      -0.006       0.002
80-89F         0.0012      0.002      0.623      0.533      -0.003       0.005
90-99F        -0.0024      0.002     -1.193      0.233      -0.006       0.002
>100F         -0.0039      0.002     -1.924      0.054      -0.008    7.43e-05
0mm           -0.1288      0.008    -16.725      0.000      -0.144      -0.114
1-4mm          0.0007      0.004      0.181      0.856      -0.007       0.008
5-14mm        -0.0903      0.008    -11.259      0.000      -0.106      -0.075
1980           2.7345      0.061     44.548      0.000       2.614       2.855
1981           2.7494      0.061     45.238      0.000       2.630       2.868
1982           2.7061      0.061     44.597      0.000       2.587       2.825
1983           2.6387      0.061     43.430      0.000       2.520       2.758
1984           2.6535      0.061     43.249      0.000       2.533       2.774
1985           2.7001      0.061     44.462      0.000       2.581       2.819
1986           2.7421      0.061     45.206      0.000       2.623       2.861
1987           2.7260      0.061     44.941      0.000       2.607       2.845
1988           2.7606      0.061     45.027      0.000       2.640       2.881
1989           2.7548      0.061     45.399      0.000       2.636       2.874
1990           2.6753      0.061     44.084      0.000       2.556       2.794
1991           2.7942      0.061     46.033      0.000       2.675       2.913
1992           2.7872      0.061     45.473      0.000       2.667       2.907
1993           2.7892      0.061     45.950      0.000       2.670       2.908
1994           2.7465      0.061     45.262      0.000       2.628       2.865
1995           2.2527      0.061     37.094      0.000       2.134       2.372
1996           2.6565      0.061     43.334      0.000       2.536       2.777
1997           2.6397      0.061     43.506      0.000       2.521       2.759
1998           2.6148      0.061     43.032      0.000       2.496       2.734
1999           2.4715      0.061     40.701      0.000       2.353       2.591
2000           2.3730      0.061     38.708      0.000       2.253       2.493
2001           2.4483      0.061     40.311      0.000       2.329       2.567
```

```
2002              2.4917      0.061     41.030      0.000       2.373       2.611
2003              2.4931      0.061     40.997      0.000       2.374       2.612
2004              2.4343      0.061     39.696      0.000       2.314       2.555
2005              2.4175      0.061     39.821      0.000       2.299       2.537
2006              2.4974      0.061     41.070      0.000       2.378       2.617
2007              2.4646      0.061     40.620      0.000       2.346       2.584
2008              2.4171      0.061     39.397      0.000       2.297       2.537
2009              2.3708      0.061     39.049      0.000       2.252       2.490
Jan               6.5601      0.156     42.027      0.000       6.254       6.866
Feb               6.4663      0.146     44.149      0.000       6.179       6.753
Mar               6.5548      0.146     44.789      0.000       6.268       6.842
Apr               6.5140      0.152     42.769      0.000       6.215       6.813
May               6.5160      0.152     42.791      0.000       6.218       6.814
Jun               6.4644      0.152     42.429      0.000       6.166       6.763
Jul               6.4811      0.153     42.472      0.000       6.182       6.780
Aug               6.4644      0.156     41.378      0.000       6.158       6.771
Sep               6.4125      0.153     42.018      0.000       6.113       6.712
Oct               6.4940      0.152     42.614      0.000       6.195       6.793
Nov               6.4982      0.152     42.668      0.000       6.200       6.797
Dec               6.5752      0.152     43.128      0.000       6.276       6.874
==============================================================================
"""
```

Poisson Model for West Alameda

In [10]:
```python
# get the data
TMAX_data_west = pd.read_csv('../group_assignment4/TMAX_data_west.csv').iloc[:,1:4]

# get temp bins
temp_bins = get_temp_features(TMAX_data_west)
temp_bins.head()
```

Out[10]:

	30-39F	40-49F	50-59F	60-69F	70-79F	80-89F	90-99F	>100F
0	0.0	0.0	32.0	28.0	0.0	0.0	0.0	0.0
1	0.0	0.0	16.0	42.0	2.0	0.0	0.0	0.0
2	0.0	0.0	14.0	41.0	6.0	0.0	0.0	0.0
3	0.0	0.0	9.0	44.0	8.0	0.0	0.0	0.0
4	0.0	0.0	4.0	43.0	12.0	2.0	0.0	0.0

In [11]:
```python
#get the data
PRCP_data_west = pd.read_csv("../group_assignment4/PRCP_data_west.csv").iloc[:,1:4]

# get prcp bins
prcp_bins = get_prcp_features(PRCP_data_west)
prcp_bins.head()
```

Out[11]:

	0mm	1-4mm	5-14mm
0	0.0	59.0	1.0
1	0.0	59.0	1.0

```
2  0.0   61.0     0.0
3  0.0   61.0     0.0
4  0.0   61.0     0.0
```

In [12]:
```
# dummy variables theta phi
theta,phi = get_dummy_features(TMAX_data_west)
```

In [13]:
```
# get independent variables
variables = pd.concat([temp_bins, prcp_bins, theta, phi], axis=1)

# get the response variable
df_west = pd.read_csv("west_alameda_crime.csv").iloc[:,1:3]
crime_west = df['crime_sum']
crime_west = crime_west.iloc[1:].reset_index(drop=True)

# fit the model
west_poisson_model = sm.GLM(crime_west,variables,family=sm.families.Poisson())
west_poisson_results = west_poisson_model.fit()
crime_west_hat = west_poisson_results.predict(variables)
# show result
west_poisson_results.summary()
```

Out[13]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:              crime_sum   No. Observations:                  359
Model:                            GLM   Df Residuals:                      308
Model Family:                 Poisson   Df Model:                           50
Link Function:                    log   Scale:                          1.0000
Method:                          IRLS   Log-Likelihood:                -11715.
Date:                Thu, 06 Dec 2018   Deviance:                       19521.
Time:                        10:21:57   Pearson chi2:                 2.09e+04
No. Iterations:                     4   Covariance Type:             nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
30-39F        -0.0107      0.003     -4.248      0.000      -0.016      -0.006
40-49F        -0.0056      0.001     -5.010      0.000      -0.008      -0.003
50-59F         0.0013      0.001      1.208      0.227      -0.001       0.003
60-69F        -0.0012      0.001     -1.118      0.263      -0.003       0.001
70-79F         0.0010      0.001      0.941      0.347      -0.001       0.003
80-89F        -0.0003      0.001     -0.277      0.782      -0.002       0.002
90-99F        -0.0052      0.001     -4.410      0.000      -0.008      -0.003
>100F          0.0302      0.003      9.670      0.000       0.024       0.036
0mm            0.0042      0.003      1.618      0.106      -0.001       0.009
1-4mm          0.0039      0.003      1.480      0.139      -0.001       0.009
5-14mm         0.0014      0.003      0.485      0.628      -0.004       0.007
```

```
1980            2.6652      0.062      42.904      0.000       2.543       2.787
1981            2.6756      0.062      43.501      0.000       2.555       2.796
1982            2.6229      0.062      42.591      0.000       2.502       2.744
1983            2.5515      0.062      41.391      0.000       2.431       2.672
1984            2.5661      0.062      41.277      0.000       2.444       2.688
1985            2.6005      0.062      42.174      0.000       2.480       2.721
1986            2.6666      0.061      43.394      0.000       2.546       2.787
1987            2.6387      0.062      42.887      0.000       2.518       2.759
1988            2.6850      0.062      43.238      0.000       2.563       2.807
1989            2.6879      0.062      43.654      0.000       2.567       2.809
1990            2.5948      0.062      42.137      0.000       2.474       2.715
1991            2.7133      0.062      44.074      0.000       2.593       2.834
1992            2.7086      0.062      43.579      0.000       2.587       2.830
1993            2.7253      0.062      44.209      0.000       2.604       2.846
1994            2.6655      0.062      43.230      0.000       2.545       2.786
1995            2.1533      0.062      34.928      0.000       2.032       2.274
1996            2.5906      0.062      41.611      0.000       2.469       2.713
1997            2.5693      0.062      41.694      0.000       2.449       2.690
1998            2.5262      0.062      40.908      0.000       2.405       2.647
1999            2.3927      0.062      38.716      0.000       2.272       2.514
2000            2.2845      0.062      36.707      0.000       2.162       2.406
2001            2.3751      0.062      38.482      0.000       2.254       2.496
2002            2.4026      0.062      38.941      0.000       2.282       2.524
2003            2.3968      0.062      38.852      0.000       2.276       2.518
2004            2.3580      0.062      37.837      0.000       2.236       2.480
2005            2.3357      0.062      37.893      0.000       2.215       2.457
2006            2.4003      0.062      38.885      0.000       2.279       2.521
2007            2.3837      0.062      38.693      0.000       2.263       2.504
2008            2.3542      0.062      37.820      0.000       2.232       2.476
2009            2.2898      0.062      37.147      0.000       2.169       2.411
Jan             6.2982      0.159      39.649      0.000       5.987       6.609
Feb             6.2334      0.149      41.837      0.000       5.941       6.525
Mar             6.3333      0.149      42.588      0.000       6.042       6.625
Apr             6.2950      0.155      40.699      0.000       5.992       6.598
May             6.3254      0.155      40.940      0.000       6.023       6.628
Jun             6.2899      0.154      40.730      0.000       5.987       6.593
Jul             6.3115      0.154      40.861      0.000       6.009       6.614
Aug             6.2920      0.158      39.807      0.000       5.982       6.602
Sep             6.2523      0.155      40.458      0.000       5.949       6.555
Oct             6.3246      0.154      40.936      0.000       6.022       6.627
Nov             6.2920      0.155      40.706      0.000       5.989       6.595
Dec             6.3326      0.155      40.857      0.000       6.029       6.636
==============================================================================
"""
```

PRNG random number generator

In [14]:
```python
# pip install cryptorandom
from cryptorandom.cryptorandom import SHA256
```

```
# set seed
r = SHA256(seed=123456)
```

Test Statistics: RMS Error

```
In [15]: # calculate rms error
         exp = rms(crime_east.append(crime_west),crime_east_hat.append(crime_west_hat))
         exp
```

```
Out[15]: 681.2314926372698
```

Permutation Test

Null Hypothesis: East/West Alameda are consistent with a single model that show the relationship between crime and whether.

Alternative Hypothesis: East/West Alameda have different relationship between crime and whether.

```
In [18]: obs = []
         # get 1000 observed rms error from the permuted models
         for i in range(1000):
             # generate permutation index
             r.setstate(baseseed=123456, counter = 2*i)
             p = get_permute(360, r=r)
             extras = get_permute(152,r=r)  # throwing out the rest bits in that counter to ge
             permute_yearmonth = TMAX_data_east['YearMonth'].unique()[p]
             permute_crime = np.arange(0,359,1)[p[1:]] # because we are fitting model from 198

             # make a copy of the TMAX PRCP and crime data for permutation
             TMAX_data_east_per = TMAX_data_east.copy()
             TMAX_data_west_per = TMAX_data_west.copy()
             PRCP_data_east_per = PRCP_data_east.copy()
             PRCP_data_west_per = PRCP_data_west.copy()
             crime_east_per = crime_east.copy()
             crime_west_per = crime_west.copy()

             # permute
             for i in permute_yearmonth:
                 TMAX_data_east_per[TMAX_data_east_per['YearMonth']==i] = TMAX_data_west[TMAX_d
                 TMAX_data_west_per[TMAX_data_west_per['YearMonth']==i] = TMAX_data_east[TMAX_d
                 PRCP_data_east_per[PRCP_data_east_per['YearMonth']==i] = PRCP_data_west[PRCP_d
                 PRCP_data_west_per[PRCP_data_west_per['YearMonth']==i] = PRCP_data_east[PRCP_d
             for i in permute_crime:
                 crime_east_per[i]= crime_west[i]
                 crime_west_per[i]= crime_east[i]

             # fit east model
             temp_bins = get_temp_features(TMAX_data_east_per)
             prcp_bins = get_prcp_features(PRCP_data_east_per)
             theta,phi = get_dummy_features(TMAX_data_east_per)
```

```python
    variables = pd.concat([temp_bins, prcp_bins, theta, phi], axis=1)
    east_poisson_model = sm.GLM(crime_east_per,variables,family=sm.families.Poisson()
    east_poisson_results = east_poisson_model.fit()
    # get predicted east crime
    crime_east_hat = east_poisson_results.predict(variables)

    # fit west model
    temp_bins = get_temp_features(TMAX_data_west_per)
    prcp_bins = get_prcp_features(PRCP_data_west_per)
    theta,phi = get_dummy_features(TMAX_data_west_per)
    variables = pd.concat([temp_bins, prcp_bins, theta, phi], axis=1)
    west_poisson_model = sm.GLM(crime_west_per,variables,family=sm.families.Poisson()
    west_poisson_results = west_poisson_model.fit()
    # get predicted west crime
    crime_west_hat = west_poisson_results.predict(variables)

    # calculate and append observed test statistics: rms error
    obs.append(rms(crime_east_per.append(crime_west_per),crime_east_hat.append(crime_w

# pvalue
pvalue = (sum(obs <= exp for obs in obs)+1)/(1000+1)
pvalue
```

Out[18]: 0.012987012987012988

In [26]: `obs[-20:]`

```
Out[26]: [688.5458473764135,
 688.814814938401,
 687.090532883382,
 688.712486093707,
 689.0062780009308,
 689.7139500708561,
 681.7039233501821,
 688.5581783187492,
 688.4389995154324,
 686.7163316517501,
 684.7979763281726,
 687.657083823286,
 689.0783637798638,
 683.8960976179711,
 684.1749806022849,
 688.1411285824455,
 688.5087375195724,
 689.3625395694486,
 687.2377299957112,
 689.686652266526]
```

The P value keeps shrinking and is about 0.013 after 1000 runs, which is significant(<0.05).

So we reject the null. The relationships between crime and weather in east and west Alameda are different

0.1 Analytical Questions

randomization: I used getrandbits(k) in cryptorandom which generates k BINARY bits at one shot that approximate I.I.D. Bernoulli trails with $p = 1/2$. These binary bits indicate whether in YearMonth n, west/east would get their original data or they have to exchange their data. With package cryptorandom, we reset the counter to $2i$ and generate 360 binary bits in each 1000 loops. To make the result reproducible, we threw away the rest 512-360=152 bits in that counter for each loop.

assumption: We fitted the same model as Ranson did. Ranson assumed that the number of crimes C_{iym} in month m of year y in county i of state s has a Poisson distribution with probability density function given by

$$f(C_{iym}|X_{iym}) = exp(-\mu(X_{iym}))\mu(X_{iym})^(C_{iym})/C_{iym}!$$

where X_{iym} is the set of all observed covariates and $\mu(X_{iym}) \equiv E[C_{iym}|X_{iym}]$ is a link function that provides a parametric form for the conditional mean of C_{iym} given X_{iym}. Following the standard practice, Ranson assumes that $\mu(X_{iym})$ takes an exponential form:

$$\mu(X_{iym}) = exp(\sum_{j=1}^{11} \alpha_0^j T_{iym}^j + \sum_{k=1}^{5} \beta_0^k P_{iym}^k + \sum_{j=1}^{11} \alpha_0^j T_{i,y,m-1}^j + \sum_{k=1}^{5} \beta_0^k P_{i,y,m-1}^k + \phi_{sm} + \theta_{iy}$$

(From Ranson2014, P279) In our model, we did not include bins less than 30 for T_MAX because we don't have any tmax below 30 in our data In our model for the east/west part of Alameda, we did not include bins greater than 14mm for PRCP becuase we don't have any prcp above 14mm in both east and west data set.

justification for null hypothesis: To test whether it is necessary to fit two models for two parts of Alameda, we cannot compare the performace of a single model and that of two models directly becuase two models have more parameters. Therefore, we instead compare the RMS error of the east/west models with many randomly assigned east/west models. It's assumed that whether west/east would assigned with their original weather and crime or they have to exchange their data of YearMonth n is a bernoulli trail with probability $p = 1/2$. After we permute the data many times, we refit the models and compare RMS errors with the original RMS error. If the RMS error of the permuted models is typically greater than the original models, we can say that the relationship between weather and crime in the east/west Alameda is different. If the RMS error of the permuted models is typically greater than the original model, it evidents that the models are different, implying that changing some of the weather and crime pairs altered the relationship between weather and crime. If the relationship of two parts is the same, inter-changing weather and crime as a pair won't change the performace of the model a lot.

When estimated the pvalue we used this method

$$((\#random\ permutations\ with\ test\ statistic >= threshold) + 1)/(\#random\ permutations\ generated\ + 1)$$

. If the null hypothesis is true, the original data are one of the equally likely permutations of the data–exactly as likely as the permutations you generate. So you really have n+1 permutations, not n (where n is the number of permutations you generated deliberately); nature gave you one more permutation. With that choice, the estimated p-value is never smaller than 1/(n+1). (From prof. Stark's email on Nov 18, 2018)