Model Emergency!

Modeling patterns in Emergency Medical Calls

Jephte Guerrier, Steve Hulac, Chris Larson

Spring 2019

911 Emergency Medical Service (EMS) call volume varies significantly by season, day of week, and time of day. While impossible to predict entirely, there are undoubtedly some correlations between call volume and other factors. A statistical model to predict incident volume would help to inform efficient ambulance staffing.

Background

The Denver Health Paramedic Division (DHPD) is the sole provider of Advanced Life Support care and emergency transport for the City and County of Denver. Other small areas around Denver covered by DHPD include the cities of Glendale, Sheridan, and Englewood, along with some areas of unincorporated Arapahoe County. Each ambulance is staffed by two experienced paramedics who work 10-hour shifts that begin and end at rotating times during the day. The city is covered by a dynamic dispersal model, where the ambulances are posted on street-corners and moved at frequent intervals to maintain coverage for the response area.

This staffing model offers the capability of maintaining staffing levels that follow anticipated response volumes. Establishing a model that could estimate predicted volume could help to maximize staffing efficiency for the 911 EMS service. Data exists for hourly EMS incident volume for DHPD back through 2002. To aid in this staffing efficiency, we propose a model that explains hourly assigned incident volume with the following predictor variables that are known far enough in advance to allow for staffing changes.

Methods

We set out to predict hourly call volume from a number of factors discussed below. We settled on 1-hours periods as our unit of analysis. This provided enough resolution discover patterns across days and sports games, as well as being able to merge with sports game data. We felt this was also general enough to make recommendations about staffing. The data sources we had overlapped to include 2011-2018. Ultimately this gave us 8*365*24+48(leap days) = 70128 observations with which to build our model.

Since different EMS agencies can use different response models, we chose to use assigned incidents as our Y variable, rather than units assigned. This helps to reduce variation based on responses (such as sending multiple units to the same call).

Data

We used historical call volume along with weather and local sports game data to predict call volume. We chose date elements that correlated with the variation in call volume:

- Hour of day
- Day of week (Binned as follows)
    - Monday-Thursday
    - Friday
    - Saturday-Sunday
- Pick (4-month period that corresponds with paramedic shift picks)
    - Spring (January-April)
    - Summer (May-August)
    - Fall (September-December)
- Year

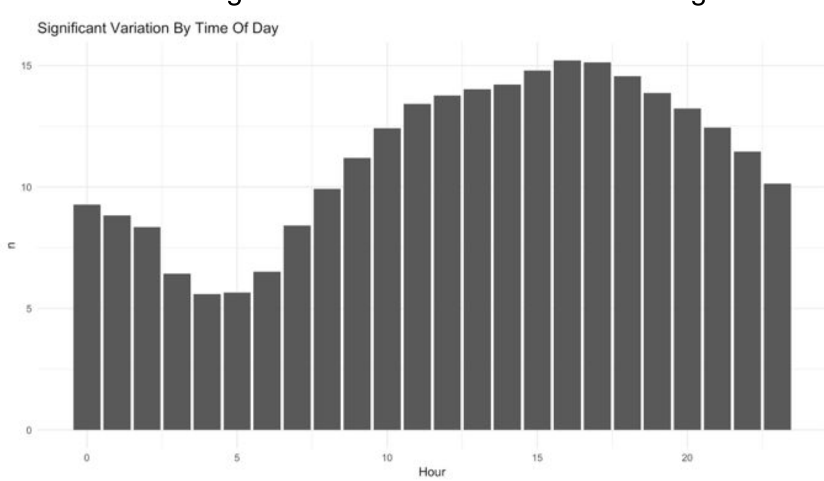
We pulled weather data from NOAA including hourly temperature recordings and precipitation.

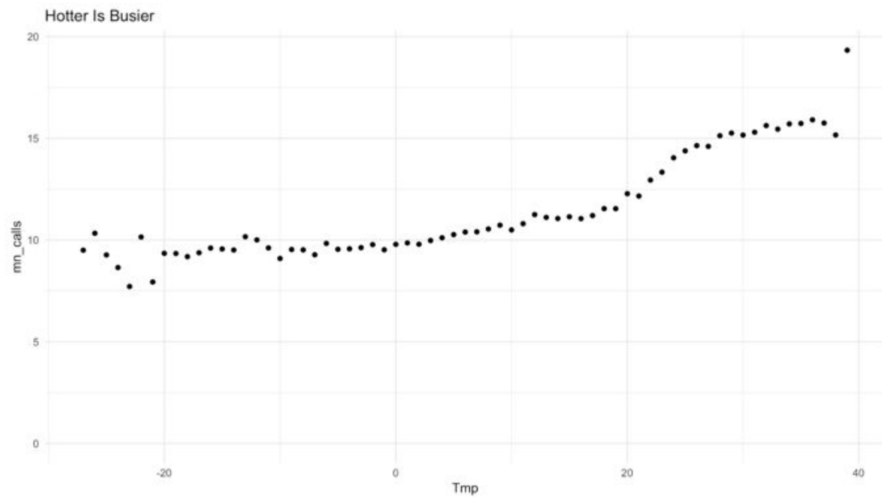
We created categorical variables for if there was a home game for the following teams, flagging positive for all hours from 1 hour before the game started to 1 hour after the game ended.

- Colorado Rockies
- Denver Nuggets
- Denver Broncos

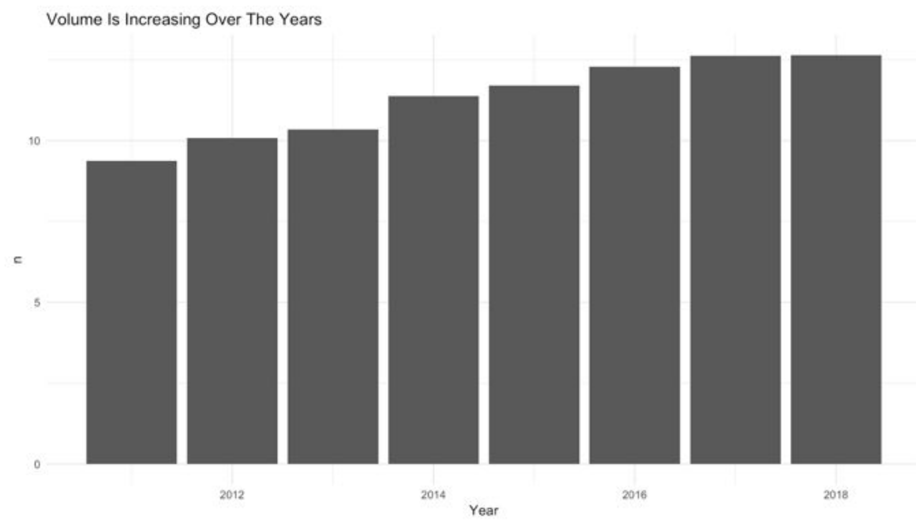
Initial observations

We first noticed significant variations in volume relating to Hour and temperature.

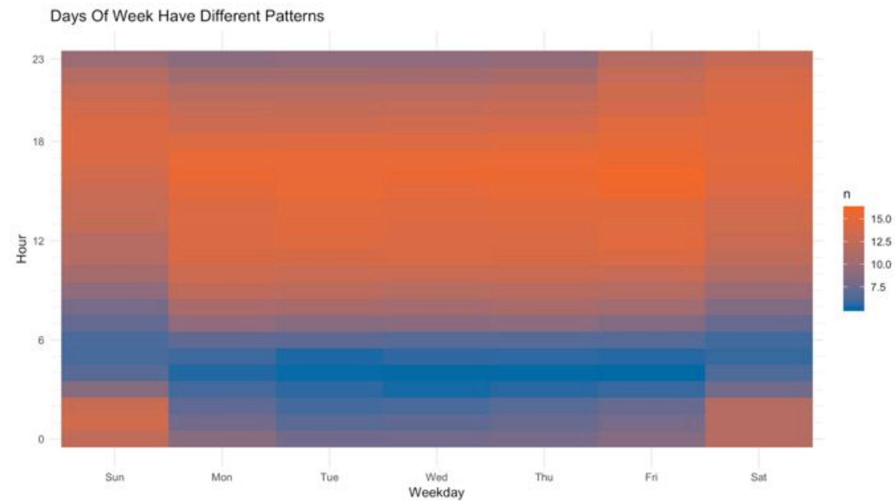




There was also a steadily increasing trend in volume over years.



Day of week had less variation, but showed differently hourly trends:



Model Fitting

Finding a model to fit the data was a major challenge in this project. We knew we had count data but finding a model was much more complex than just plug and play. In fact plugging in the raw data resulted in zero success in a good fitting model. We explored regular poisson, quasi-poisson, negative binomial, ZIPF, and logistic regression as ways to get a good fit with the full data set.

Examples of chi squared GOF for full data

Poisson

```
##      res.deviance    df p
## [1,]     131025.9 70709 0
```

Negative Binomial

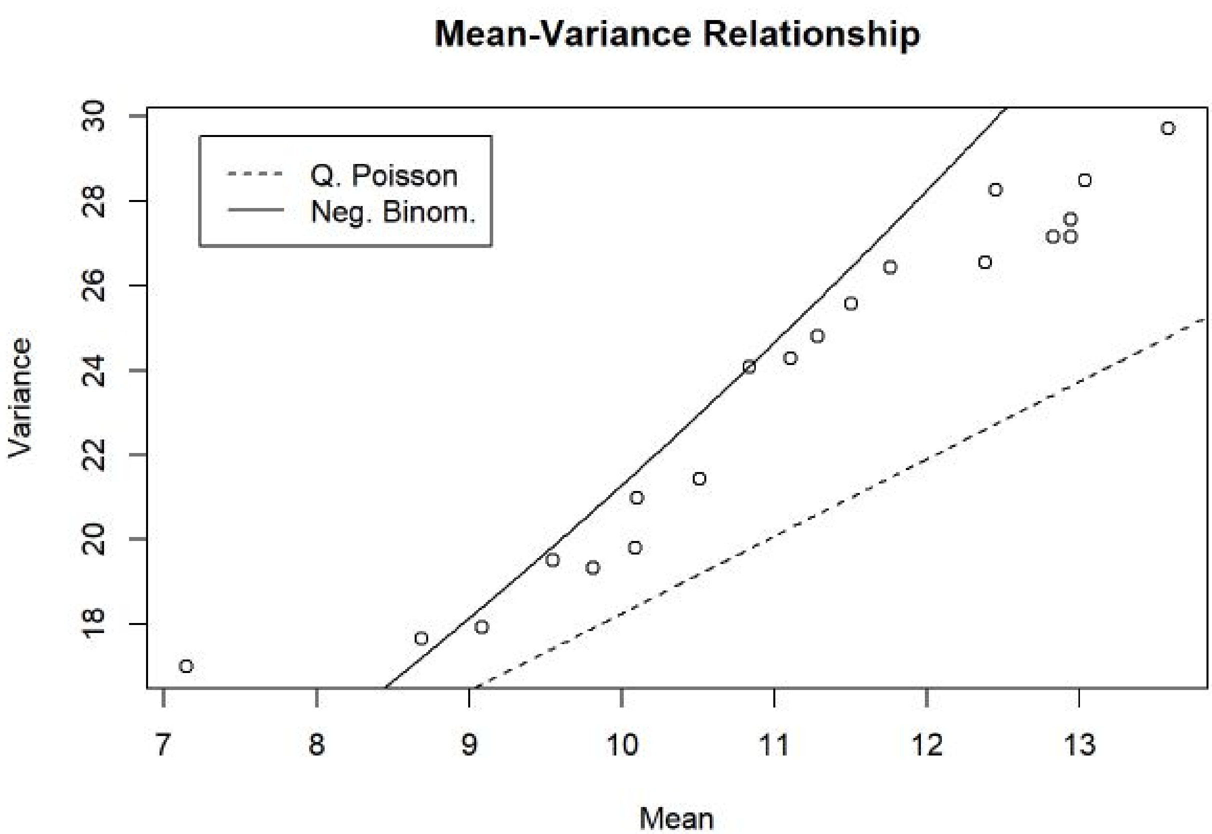
```
##      res.deviance    df            p
## [1,]     74688.62 70709 1.347968e-25
```

```
##      res.deviance    df            p
## [1,]     74766.89 70709 1.611664e-26
```

The results were similar across the board, no model considered constituted a good fit.

A quick view of the data showed why a poisson model was not a good fit. The data was overdispersed not meeting the assumption of equal mean and variance of poisson regression. I.E. the ratio of mean:Variance should be close to 1 for poisson, in our case it was approx. 2.38.

The first hint of using a Negative Binomial model came from graphically exploring the mean-variance relationship and a view of the observed versus expected counts:



Neither poisson or negative binomial looks great, but negative binomial clearly looks to be a better fit.

Observed Vs. Expected for Poisson and Negative Binomial:

Chi-squared p-value: 0 0

Chi-squared table:

	obscounts	theo Poisson	theo NegBin
<= 1	481	13.28360	418.8378
<= 2	1180	67.45583	875.7144
<= 3	2114	248.98918	1663.8067
<= 4	3025	689.29124	2617.1521
<= 5	3673	1526.56401	3603.3457
<= 6	4119	2817.38395	4489.9079
<= 7	4518	4456.87303	5175.1749
<= 8	4770	6169.11040	5602.4516
<= 9	4967	7590.35869	5759.7168
<= 10	5006	8405.13254	5670.3098
<= 11	5176	8461.24260	5380.0452
<= 12	5022	7807.91664	4944.7287
<= 13	4772	6650.80298	4420.2745
<= 14	4258	5260.51561	3856.1566
<= 15	3801	3883.46434	3291.9873
<= 16	3252	2687.70504	2756.5419
<= 17	2758	1750.71298	2268.4260
<= 18	2222	1077.02234	1837.6585
<= 19	1686	627.70181	1467.6109
<= 20	1259	347.54068	1156.9267
<= 21	966	183.26040	901.2050
<= 22	660	92.24189	694.3535
<= 24	767	64.90074	929.7652
> 24	443	15.52950	1112.9023

**Note the P-values at the top, both are zero suggesting neither poisson or negative binomial models are a good fit. On further inspection we can clearly see that the negative binomial model is much “closer” to being accurate.*

We looked at numerous methods to manipulate the data in a way to achieve a model that could pass a goodness of fit test. We shifted the response variable (hour), to make its distribution look normal, we attempted to transform the response variable. Both with limited success. We explored outliers and their potential removal. We looked at simply truncating the data set to capture the middle 95% of the data. It was here that we started to have success in finding a good fitting model.

Truncating the bottom 2.5% of data was the data manipulation method we used to form our final model. We were able to achieve a goodness of fit with Poisson regression as well, however that required also truncating the top 2.5% of data, which excluded too many data points for our liking. When keeping in mind the goal of setting staffing levels for ambulances needed, removal of the bottom 2.5% was an easy choice. This set the lower bound of calls per hour at 3 and for the size and EMS call volume of Denver there would never be a recommendation of less than 3 ambulances staffed at any particular hour. Using this method we were able to fit a Negative Binomial model while keeping in all of the outlying data points (including points that were potentially “bad data”). A total of 1778 observations were removed.

Negative Binomial GOF with bottom 2.5% removed

res.deviance	df	p
69200.14	69318	0.6236

The Final Model

After truncating the data and using cross validation for variable selection the final model included the following variables:

Year	Pick	Hour
Temp	Precipitation	Broncos
Mon-Thurs	Fri	Temp*Precipitation

A negative binomial regression was run with the following results:

	Estimate	Std. Error	z value	Pr(>|z|)	2.5 %	97.5 %
(Intercept)	-88.8499	1.0610	-83.7393	0.0000	-90.9329	-86.7672
Year	0.0452	0.0005	85.8271	0.0000	0.0442	0.0462
pickSpring	-0.0199	0.0031	-6.3947	0.0000	-0.0261	-0.0138
pickSummer	0.0027	0.0035	0.7572	0.4489	-0.0042	0.0096
Hour1	-0.0483	0.0093	-5.1841	0.0000	-0.0665	-0.0300
Hour2	-0.1034	0.0095	-10.9115	0.0000	-0.1220	-0.0848
Hour3	-0.3654	0.0103	-35.6017	0.0000	-0.3855	-0.3453
Hour4	-0.5025	0.0109	-46.0836	0.0000	-0.5239	-0.4812
Hour5	-0.4937	0.0108	-45.5384	0.0000	-0.5150	-0.4725
Hour6	-0.3450	0.0101	-34.0235	0.0000	-0.3649	-0.3251
Hour7	-0.0883	0.0094	-9.4075	0.0000	-0.1067	-0.0699
Hour8	0.0694	0.0090	7.6916	0.0000	0.0517	0.0871
Hour9	0.1804	0.0088	20.5201	0.0000	0.1632	0.1977
Hour10	0.2737	0.0086	31.6454	0.0000	0.2568	0.2907
Hour11	0.3435	0.0086	40.1325	0.0000	0.3267	0.3602
Hour12	0.3631	0.0085	42.5181	0.0000	0.3464	0.3799
Hour13	0.3755	0.0085	43.9774	0.0000	0.3587	0.3922
Hour14	0.3862	0.0085	45.2598	0.0000	0.3694	0.4029
Hour15	0.4267	0.0085	50.3480	0.0000	0.4101	0.4433
Hour16	0.4564	0.0084	54.1765	0.0000	0.4399	0.4729
Hour17	0.4551	0.0084	54.1045	0.0000	0.4386	0.4715
Hour18	0.4243	0.0084	50.3370	0.0000	0.4078	0.4408
Hour19	0.3825	0.0085	45.1391	0.0000	0.3659	0.3991
Hour20	0.3410	0.0085	39.9852	0.0000	0.3243	0.3578
Hour21	0.2866	0.0086	33.2840	0.0000	0.2697	0.3035
Hour22	0.2061	0.0088	23.5494	0.0000	0.1890	0.2233
Hour23	0.0851	0.0090	9.4785	0.0000	0.0675	0.1027
Temp	0.0046	0.0002	29.4210	0.0000	0.0043	0.0049
precipT	0.0154	0.0063	2.4632	0.0138	0.0031	0.0277
broncosy	-0.0863	0.0183	-4.7014	0.0000	-0.1224	-0.0504
M_ThTRUE	-0.0302	0.0028	-10.7664	0.0000	-0.0357	-0.0247
FriTRUE	0.0125	0.0039	3.1745	0.0015	0.0048	0.0202
Temp:precipT	-0.0023	0.0005	-4.1394	0.0000	-0.0033	-0.0012

The general form of the regression equation for this output is:

$$ln(\hat{Y}) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_n x_n$$

For ease of understanding the output, common practice suggests using the IRR (incident rate ratio) which comes from exponentiating the general form above.

$$\hat{Y} = Exp[\hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_n x_n]$$
$$= e^{\hat{\beta}_0} e^{\hat{\beta}_1 x_1} \ldots e^{\hat{\beta}_n x_n}$$

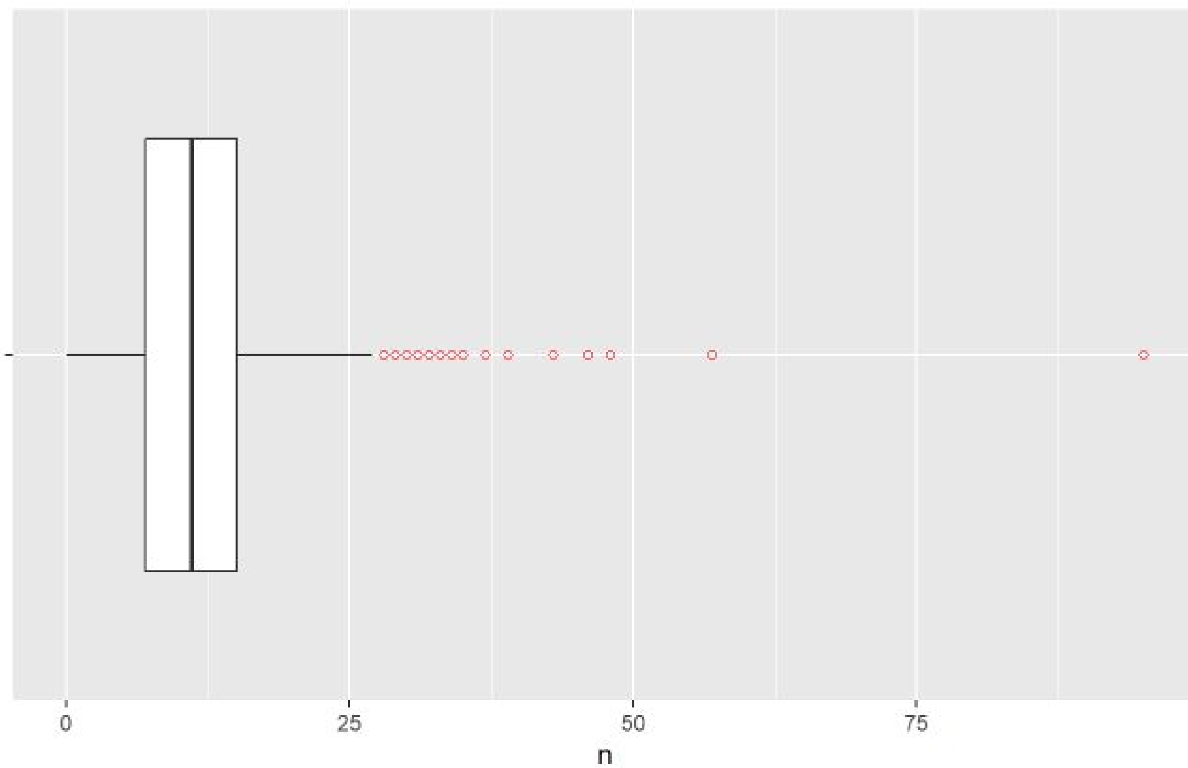
Note this is has a multiplicative effect instead of an additive one and results are reported in a percent change format instead of a change in the log count. Any coefficient values above 1 will represent an increase, below 1 will represent a decrease. For example a value of 1.45 would represent an expected increase of 45% and a value of 0.75 would represent and expected decrease of 25% versus the base case.

The exponentiated coefficients

Year	1.0462
pickSpring	0.9803
pickSummer	1.0027
Hour1	0.9529
Hour2	0.9017
Hour3	0.6939
Hour4	0.6050
Hour5	0.6103
Hour6	0.7082
Hour7	0.9155
Hour8	1.0718
Hour9	1.1978
Hour10	1.3148
Hour11	1.4098
Hour12	1.4378
Hour13	1.4557
Hour14	1.4713

Hour15	1.5322
Hour16	1.5784
Hour17	1.5763
Hour18	1.5285
Hour19	1.4659
Hour20	1.4064
Hour21	1.3319
Hour22	1.2289
Hour23	1.0888
Temp	1.0046
precipT	1.0155
broncosy	0.9174
M_ThTRUE	0.9703
FriTRUE	1.0125
Temp:precipT	0.9977

Outliers:




```
## outliers
## 28 29 30 31 32 33 34 35 37 39 43 46 48 57 95
## 42 22 15 16  8  5  1  1  1  1  1  1  1  1  1
```

Future Work

This data can be used in the future (along with traffic and response time data) to estimate the number of ambulances required in each hour of each week. This can help any agency to make their staffing model as efficient as possible.
Looking for patterns in wins or losses of the local sports teams is possible. Checking for patterns with the lunar cycle could help to support or disprove some myths among EMS providers.

Data sources

We collected data from the following sources:

- Denver Health Paramedics
- NOAA
- Denver Nuggets
- Denver Broncos
- Baseball-refererence.com

Appendix

R Code

K-fold validation and stepwise selection

```
set.seed(125) #for reproducibility

training.samples <- CallVolume$n %>%
  createDataPartition(p = 0.8, list = FALSE)  # Here 80% of the data used for training

train_control <- trainControl(method = "cv", number = 10) #split the data into 10 random samples

step_model <- train(n ~ Year + Hour + pick + Temp + precip + broncos + nuggets + M_Th + Fri + Temp * precip, data = CallVolume,
                    method = "leapBackward",
```

```
        tuneGrid = data.frame(nvmax = 1:9),
        trControl = train_control            #This spits out models with a maximum of
9 variables

step_model$results  #This gives the RMSE and MAE numbers of the model

summary(step_model$finalModel) #Helps selecting the best model

coef(step_model$finalModel, 9) #This spits out the model’s coefficients
```

Packages needed: tidyverse, caret, leaps, and MASS.