LA Traffic Data Analysis 🚗

Using open-source data to analyze collision patterns in Los Angeles

Jai Bansal Apr 13, 2020 · 16 min read ★

Image by Jsig9 from Pixabay

Traffic is an issue that's familiar to pretty much everyone. As a 7-year Los Angeles resident, I've sat in more than my fair share of gridlock, seemingly regardless of time of day or day of week. That's why I was so interested when I stumbled upon a traffic collision data set maintained by the city of Los Angeles. This data is cool for several reasons. While it doesn't directly measure traffic, it measures a closely-related proxy. It's not a stretch to hypothesize that more traffic correlates with more collisions which directly cause more traffic.

maintained by the city of Los Angeles and is freely available to the public.

After browsing the data, I settled on 3 major questions I wanted to attempt to answer:

- How do traffic collision patterns vary by time of day, day of week, and time of year?
- How are collisions distributed geographically? Is it possible to identify high-risk areas or intersections?
- Is it possible to predict the number of collisions in a given time frame?

Before getting into the questions above, let's learn more about the data set.

📈 The Data

The data begins in January 2010 and is updated weekly. In my particular case, I use data from January 2010 - July 2019, which ends up being ~500K rows. Each row corresponds to a collision. This data is transcribed from original paper traffic reports, so it's very likely that there are errors. Below is a sample of some of the key fields:

dr_no	date_occ	time_occ	area_name	vict_age	vict_sex	location_1
100100767	2010-03-31	400	Central	21	M \n,	\n(34.0695, -118.2324)
100100831	2010-04-18	140	Central	50	F \n,	\n(34.0424, -118.2718)
100101705	2010-12-16	2045	Central	49	M \n,	\n(34.0551, -118.2545)
100104019	2010-01-01	1400	Central	47	M \n,	\n(34.0503, -118.2504)
100104025	2010-01-01	1700	Central	36	M \n,	\n(34.0369, -118.2522)

Selected fields from LA collision data set

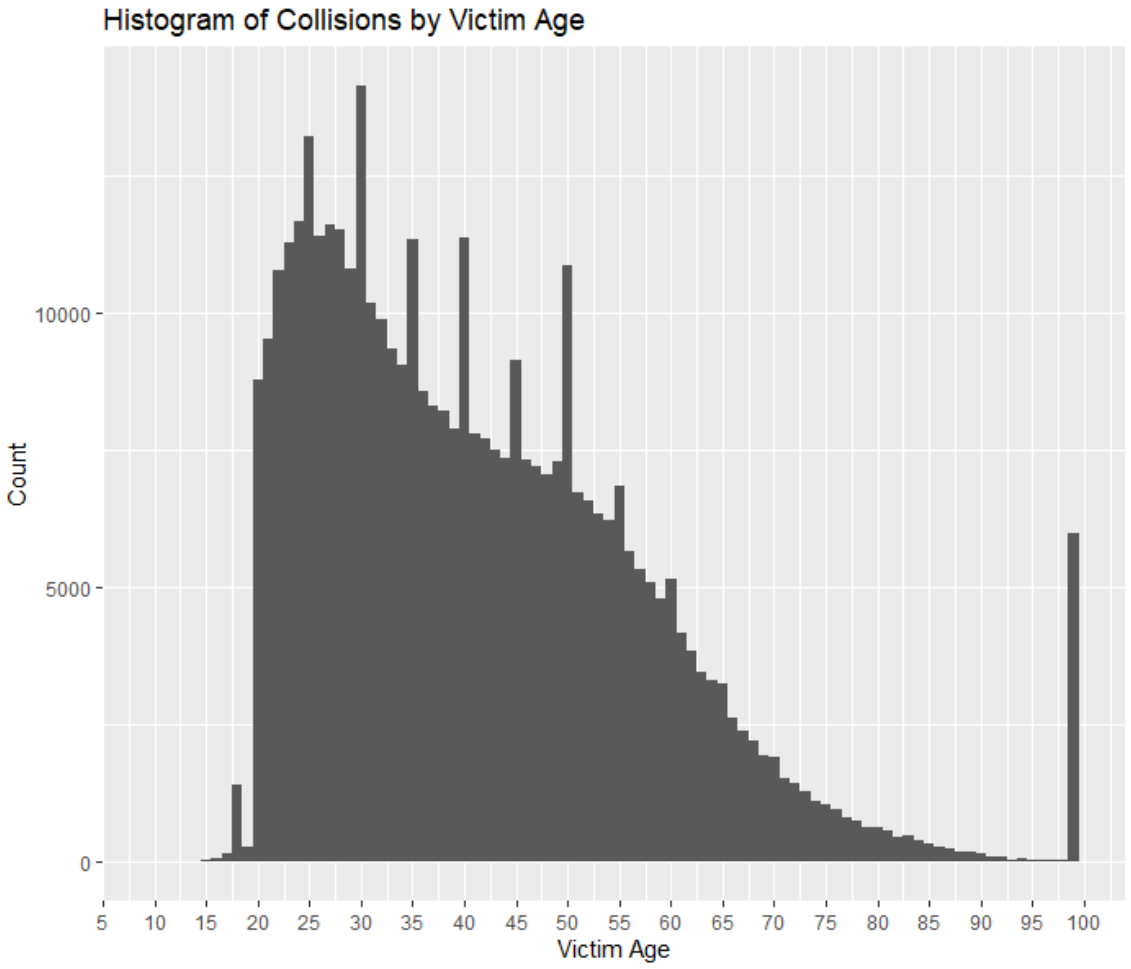
The availability of these fields inspired the key questions listed above.

As with any data set, this one needs cleaning before starting any analysis. There are a few columns with only one value, reflecting the fact that all rows in this data correspond to traffic collisions.

There are also multiple fields with the approximate street names of collisions (not shown above). These text fields need cleaning, specifically removing extra spaces. Similarly, in the image above we can see the latitude/longitude coordinates contained in a string. I extract these coordinates in separate columns for later use.

The next step is to check for null or missing values. ~16% of collisions (~78K) don't have an associated victim age. There is also a small number (~400) of collisions that do not have valid latitude/longitude coordinates and will be excluded from the mapping section in part 2.

exploration. This sort of analysis is typically useful. Even without a specific goal in mind, I often find helpful trends or insights. I'll start by plotting a few of the variables in the data. By the way, all of this work was done in R and the code is linked at the bottom of the post!



Number of collisions by victim age. Note the spike at age 99!

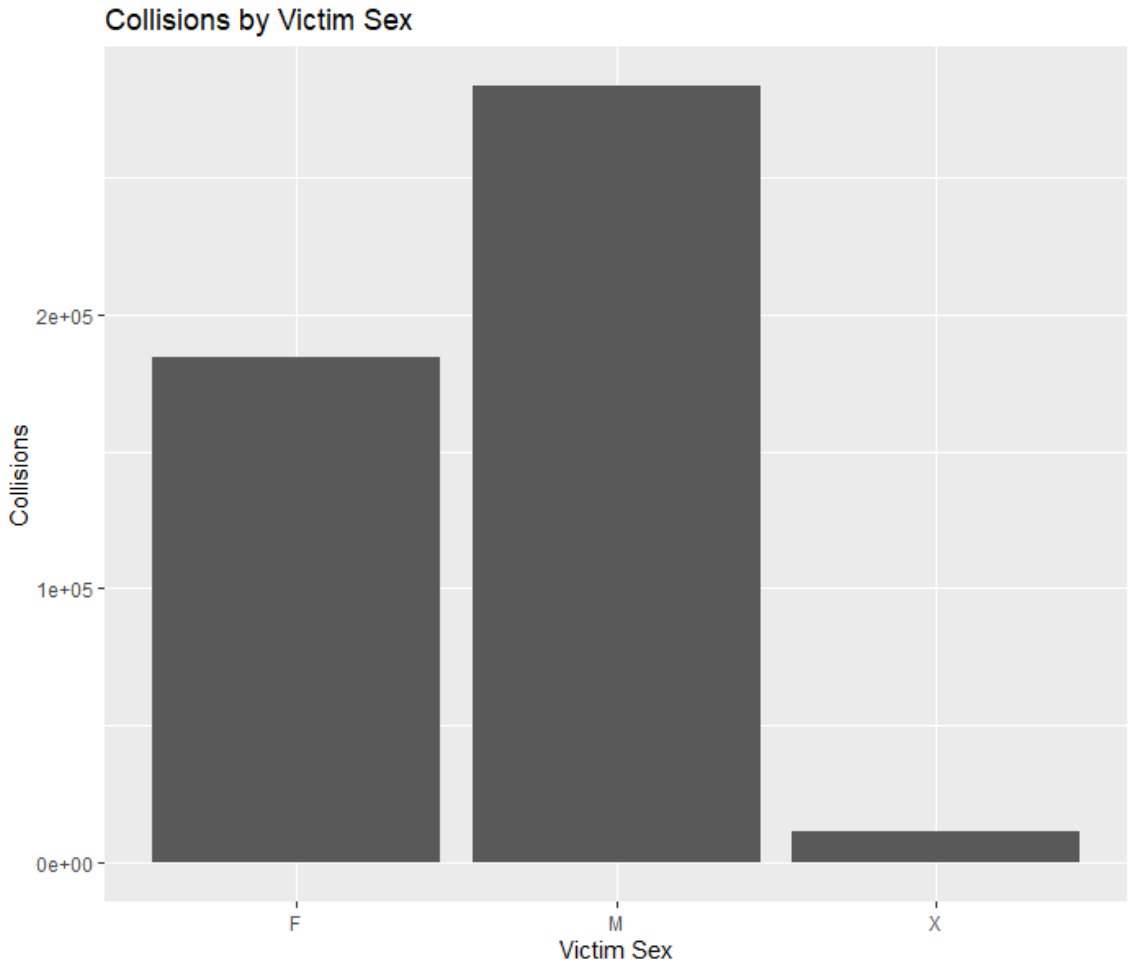
Here's what jumps out at me:

- There are hardly any victims below age 15.
- Most victims are in their 20s. The number of collision victims per age generally decreases after age 30.
- There are spikes at most multiples of 5 (25, 30, 35, etc). This suggests that some ages are estimated and that official identification (such as a driver's license) isn't always used in collision reports.
- Age 99 seems to be a catch-all bucket. It seems unlikely that there are actually as many age-99 victims as are shown above.

This plot also raises questions:

I emailed the data owner about these questions, but unfortunately haven't heard back. I'll add an update if I get a response.

Next, let's look at collisions by gender.



Number of collisions by gender. "X" represents unknown gender.

- This plot tells us that given that a collision occurred, the victim is much more likely to be male than female.

This work would be more interesting if I had the total number of drivers by gender, allowing for a collisions per capita measure. This will be a recurring shortcoming of this data and would be one of my main extensions of this analysis.

The next thing I want to do is look at the lowest and highest collision days in the data. I only include the top and bottom 0.5% on either end so I can review the results manually. Here are the lowest collision days:

```
## # A tibble: 17 x 4
##    date_occ   daily_count day       hypothesis
##    <chr>            <int> <chr>     <chr>
##  1 2010-01-02          79 Saturday  New Year
##  2 2010-02-15          80 Monday    Presidents Day
```

```
##  7 2012-11-22         82 Thursday  Thanksgiving
##  8 2012-12-25         68 Tuesday   Christmas
##  9 2012-12-30         79 Sunday    New Years
## 10 2013-01-21         81 Monday    MLK Day
## 11 2013-11-28         71 Thursday  Thanksgiving
## 12 2013-12-25         80 Wednesday Christmas
## 13 2013-12-28         81 Saturday  Christmas / New Year
## 14 2013-12-29         86 Sunday    Christmas / New Year
## 15 2014-01-12         86 Sunday    Random Sunday
## 16 2015-01-19         79 Monday    MLK Day
## 17 2018-12-25         83 Tuesday   Christmas
```

Days with the lowest number of collisions. “daily_count” is the number of collisions on the corresponding day.

- Most low-collision days fall on or around holidays. Intuitively, this makes sense as there are likely less people driving on (certain) holidays.
- Most low-collision days occur before early 2014. We’ll see later in this post that monthly collisions start rising after 2014.

And here are the highest collision days:

```
# A tibble: 17 x 3
   date_occ   daily_count day
   <chr>            <int> <chr>
 1 2010-02-05         202 Friday
 2 2010-12-17         219 Friday
 3 2015-10-09         213 Friday
 4 2015-11-20         208 Friday
 5 2016-02-17         218 Wednesday
 6 2016-12-15         223 Thursday
 7 2017-09-29         204 Friday
 8 2017-11-04         202 Saturday
 9 2017-11-17         230 Friday
10 2017-12-01         219 Friday
11 2018-01-08         212 Monday
12 2018-01-09         204 Tuesday
13 2018-02-16         217 Friday
14 2018-03-02         205 Friday
15 2018-10-12         216 Friday
16 2018-11-30         202 Friday
17 2018-12-07         207 Friday
```

Days with the highest number of collisions. “daily_count” is the number of collisions on the corresponding day.

- Most high-collision days are Fridays occurring after 2015.
- We’ll see later in this post that Fridays typically have the highest number of collisions out of any day of the week.

These results bring up a few questions and thoughts outside the scope of this analysis.

- Why are only some holidays associated with low-collision days? For example, MLK Day shows up twice as a low-collision day but Independence Day never does.
- Why don’t holidays like Memorial Day or Labor Day show up as either low or high collision days?

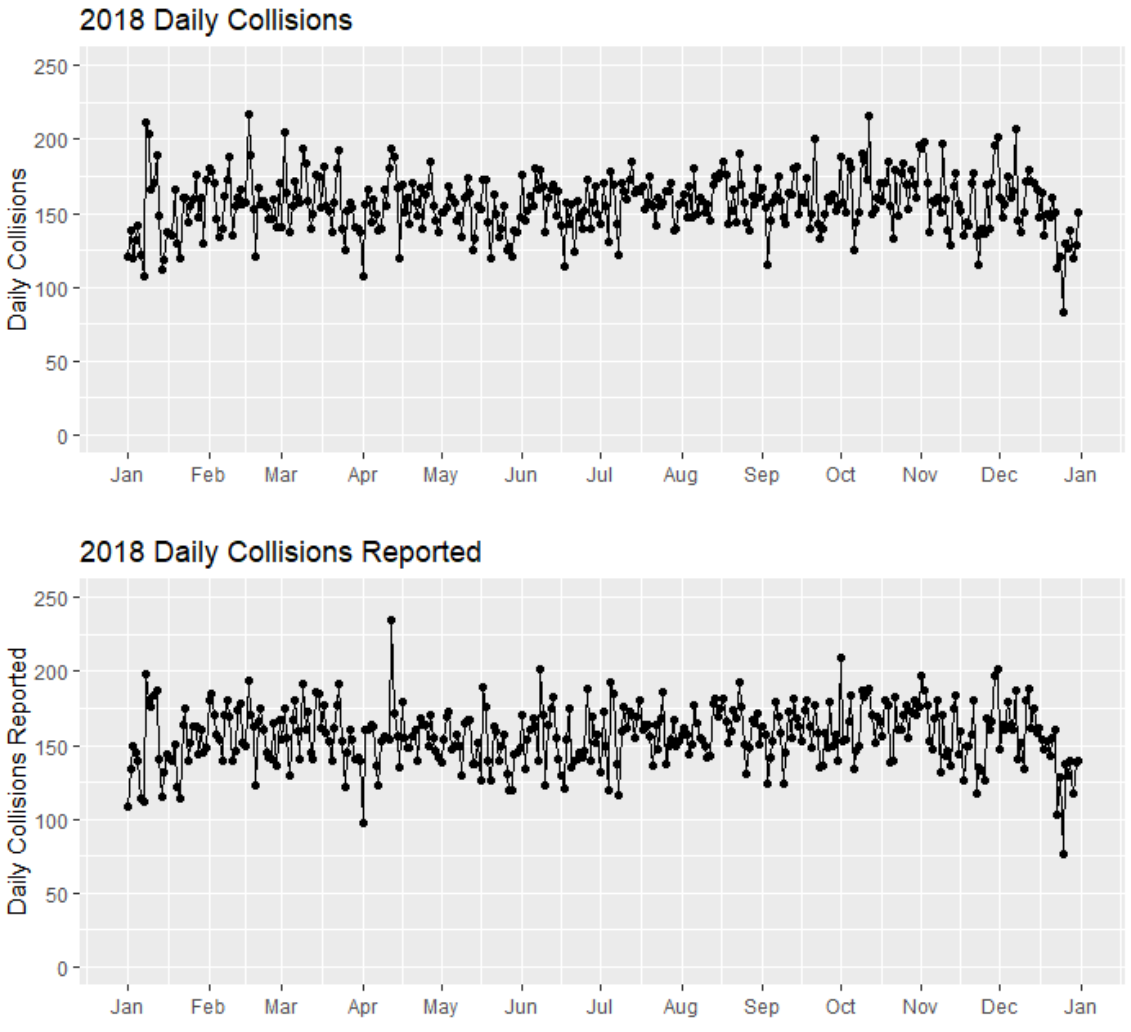
- Would it be interesting to look at weather conditions on high-collision days?

With this descriptive analysis in mind, I'm ready to tackle the key questions I outlined at the beginning of this post. The first one is about analyzing collisions over time.

🕒 Collisions by Time

In this section, I analyze how collision patterns vary by time of day, day of week, and time of year.

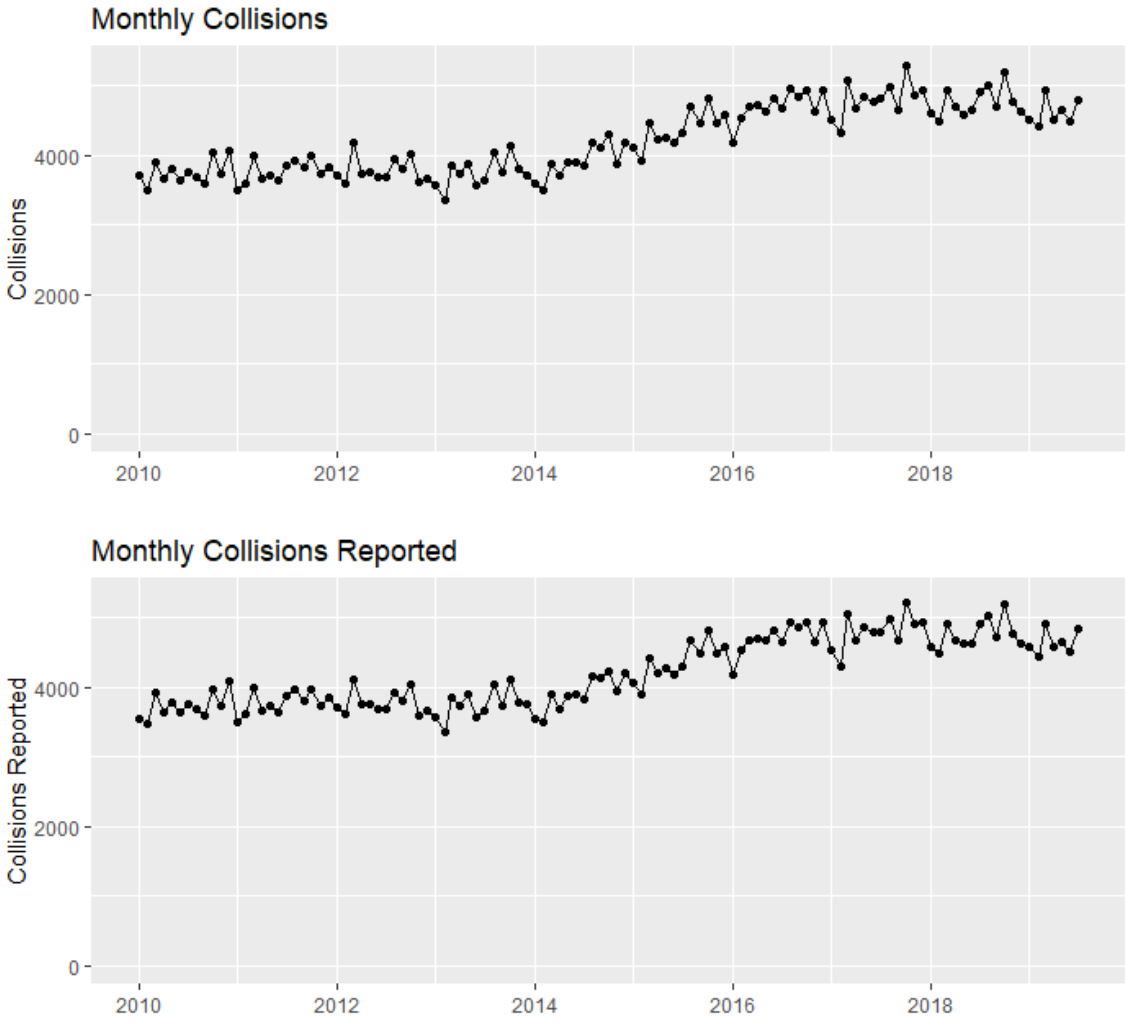
First, I plot daily collisions for 2018. Plotting all of the data starting from 2010 is too chaotic and so I focus only on 2018 for many parts of this section. In addition to collision date, the data has a field for the reporting date. This is the date the collision was actually reported to police. In most cases, the reporting date is the same day or one day after the collision date.



2018 daily collisions and collisions reported. Can you spot the differences?

- These differences make me think that there may be administrative dynamics at play regarding when collisions are reported or processed.
- The outliers in the data don't have an obvious pattern.

Let's see a similar plot aggregated by month. At this level, I can include the entire time frame from 2010–2019.

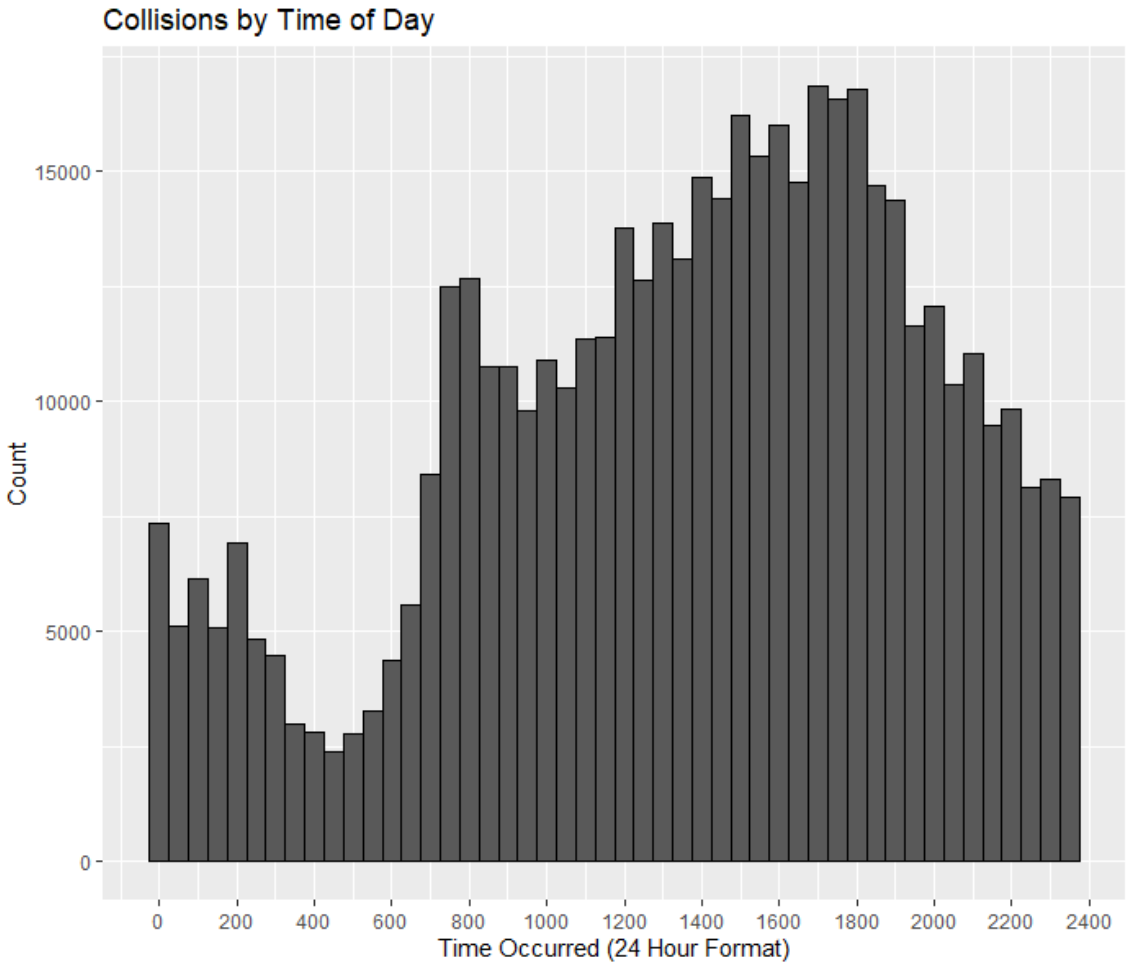


Monthly collisions and collisions reported

- At the monthly level, these two quantities seem to track each other more closely than at the daily level.
- Monthly collisions were roughly constant from 2010 - 2014, rose from 2014 - 2017, and were roughly constant from 2017 - 2019. Remember that the data used for this analysis ends in July 2019.

So, why do monthly collisions rise in the plot above? Did the number of people living and driving in LA rise from 2014 - 2017? Could it be related to the rise of ride-sharing

Next, I analyze the distribution of collisions throughout the day.



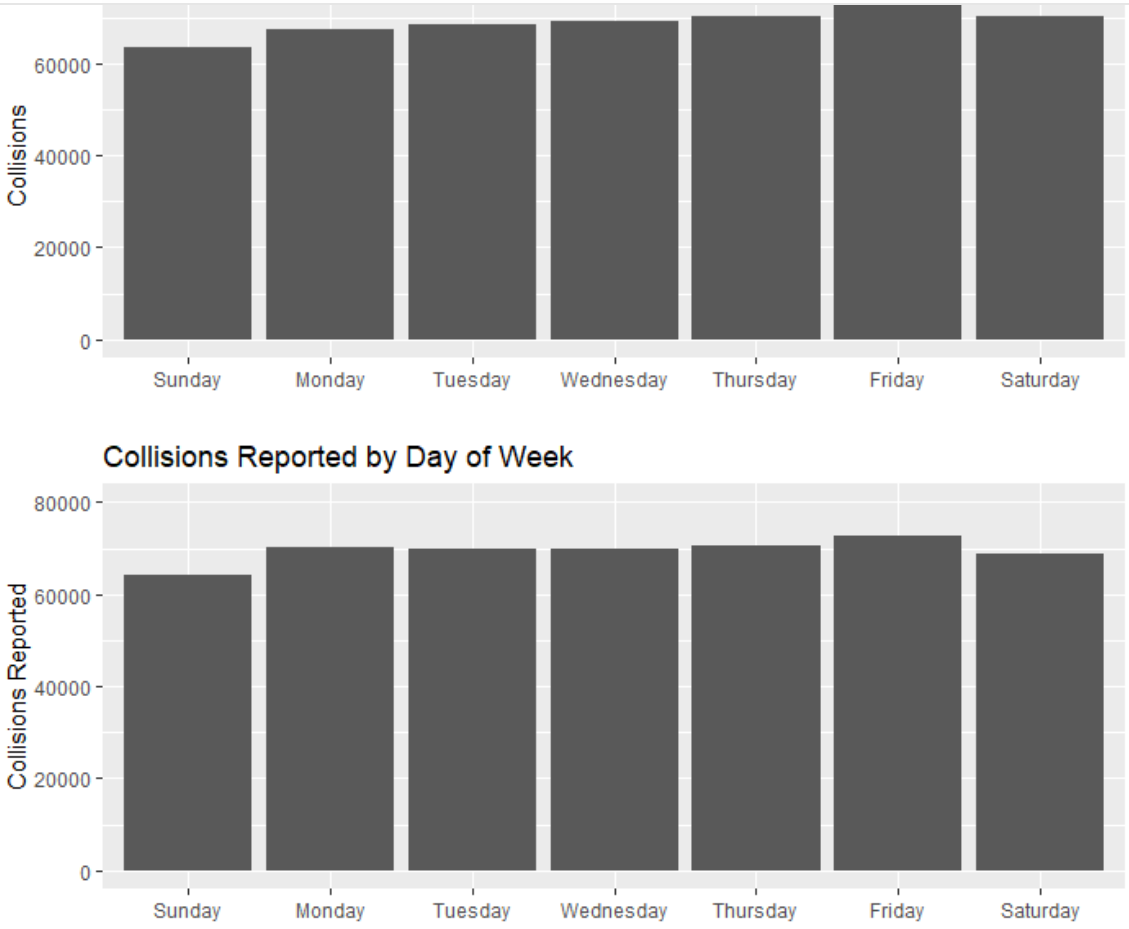
Collisions by time of day

Collisions are:

- sharply increasing from ~4am to ~8am
- decreasing from ~8am to ~930am
- generally increasing from ~930am to ~6pm
- sharply decreasing from ~8pm to ~4am
- at their daily minimum [maximum] at ~4am [~5pm]

These results likely mirror the number of vehicles on the road. It seems intuitive that many collisions occur during the evening rush hour. As I mentioned before, having access to a measure of total vehicles on the road per hour would allow me to calculate collisions per capita. There are no hourly timestamps available for when collisions are reported, so I can't plot that field by hour.

The next step is to examine collisions by day of week.



Collisions and collisions reported by day of week

Collisions are:

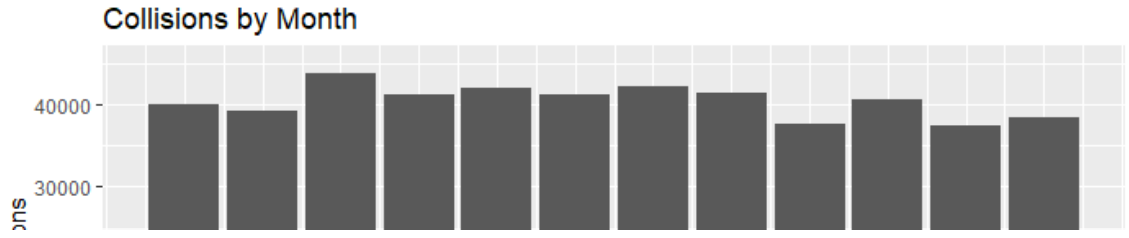
- increasing from Sunday - Friday, with a sharp increase from Thursday to Friday
- at their weekly minimum [maximum] on Sunday [Friday]

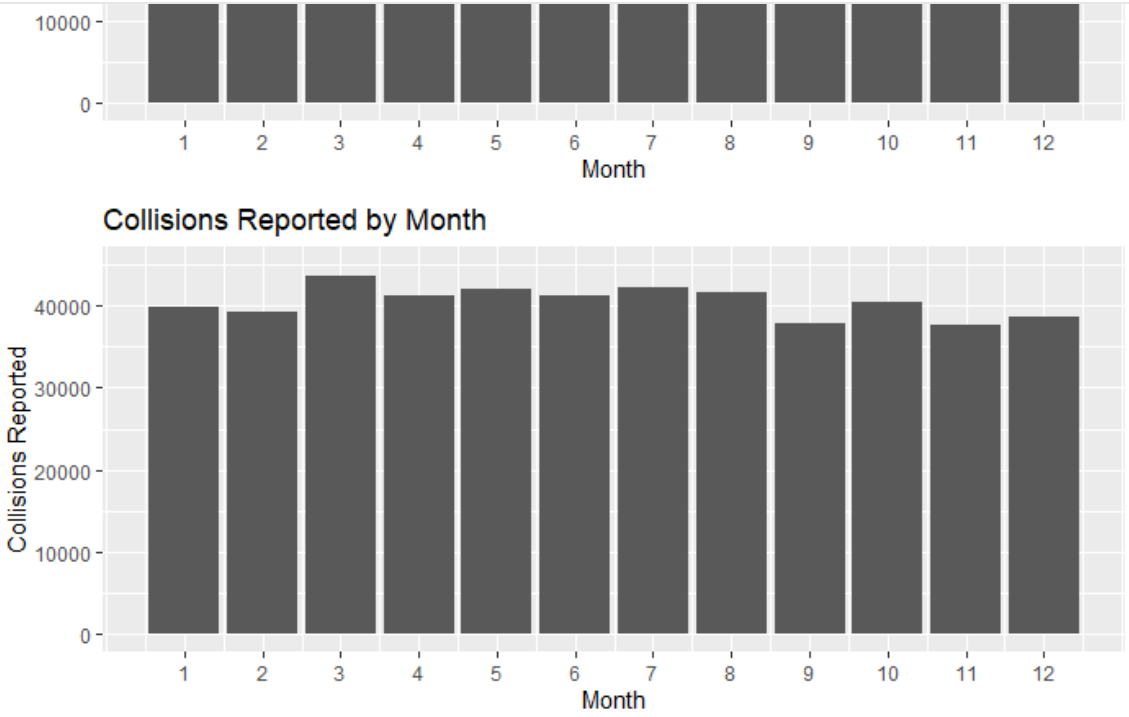
The end of the working week is the obvious hypothesis for the high number of Friday collisions. I haven't come up with any others so far!

For collisions reported:

- Sunday [Friday] still have the least [most] collisions
- However, collisions reported per weekday are essentially constant

Finally, I plot results by month.





Collisions and collisions reported by month. These fields are similar at this resolution.

Collisions are:

- generally constant from April to August
- generally lower from September to February, especially from September to December
- highest in March

So, collisions tend to be lower in colder months. One possible explanation for this is less tourists visiting LA in the colder months. Regarding the high number of collisions in March, my initial hypothesis was Daylight Savings Time. However, none of the highest collision days were on this date. Perhaps this result stems from spring break tourists?

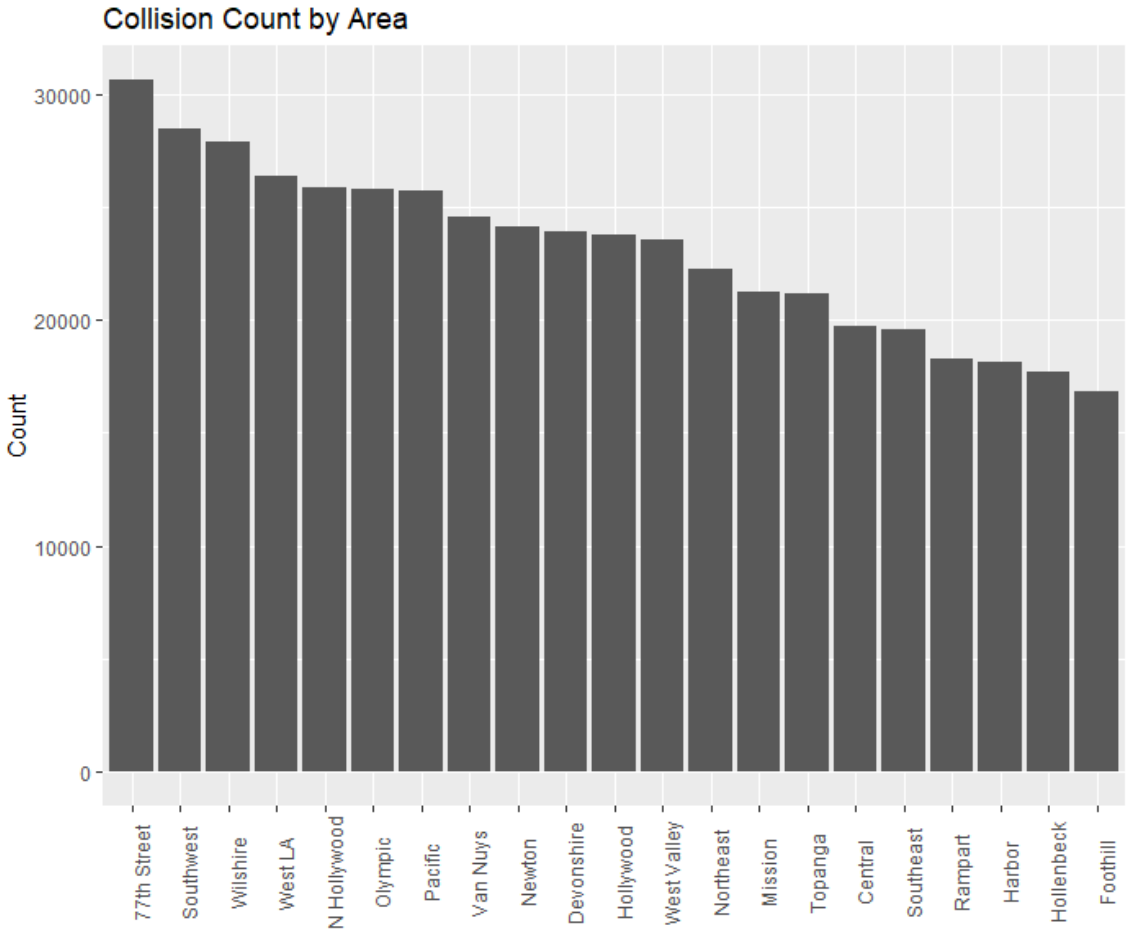
This concludes my analysis of the temporal patterns of collisions. Here's my summary of the results above.

- Collisions and collisions reported vary substantially at the daily level, but not at the monthly level.
- Monthly collisions were roughly constant from 2010 - 2014, rose from 2014 - 2017, and were roughly constant from 2017 - mid-2019 (the end date of this analysis).
- The number of collisions is lowest [highest] at ~4 AM [~5 PM].
- The number of collisions is lowest [highest] on Sunday [Friday].
- The number of collisions is lowest [highest] in September-December [March].

Next up is looking at collisions by geography.

🌎 Collisions by Geography

In addition to latitude/longitude coordinates, the data has multiple fields describing where a collision occurred. I'll start out by looking at these fields. First, I'll plot the distribution of collisions by `area`.



Collisions counted by the "area" column

Some areas obviously have more collisions than others. But without additional information such as size or traffic density per `area`, this graphic isn't too informative.

The data also includes fields called `location` and `cross_street`. `location` is the main street a collision occurred on, while `cross_street` is the nearest cross street. I'll look at the 10 most common values for these fields and their combination.

location	count	percentage
<chr>	<int>	<dbl>

4	SEPULVEDA BL	5493	1.1
5	VERMONT AV	5346	1.1
6	VICTORY BL	5114	1.1
7	SUNSET BL	5112	1.1
8	FIGUEROA ST	4707	1
9	ROSCOE BL	4395	0.9
10	OLYMPIC BL	4280	0.9

“location” with the most collisions

The 10 most common `location` are some of the longest and most trafficked roads in LA. These top 10 streets account for >10% of total collisions. There are >25K total `location`, so there’s a very long tail. Now, I’ll look at the `cross_street` field.

	cross_street	count	percentage
	<chr>	<int>	<dbl>
1	""	21813	4.5
2	VERMONT AV	3633	0.7
3	FIGUEROA ST	3418	0.7
4	WESTERN AV	3413	0.7
5	SHERMAN WY	2873	0.6
6	SEPULVEDA BL	2761	0.6
7	VICTORY BL	2758	0.6
8	BROADWAY	2694	0.6
9	3RD ST	2615	0.5
10	PICO BL	2602	0.5

“cross_street” with the most collisions

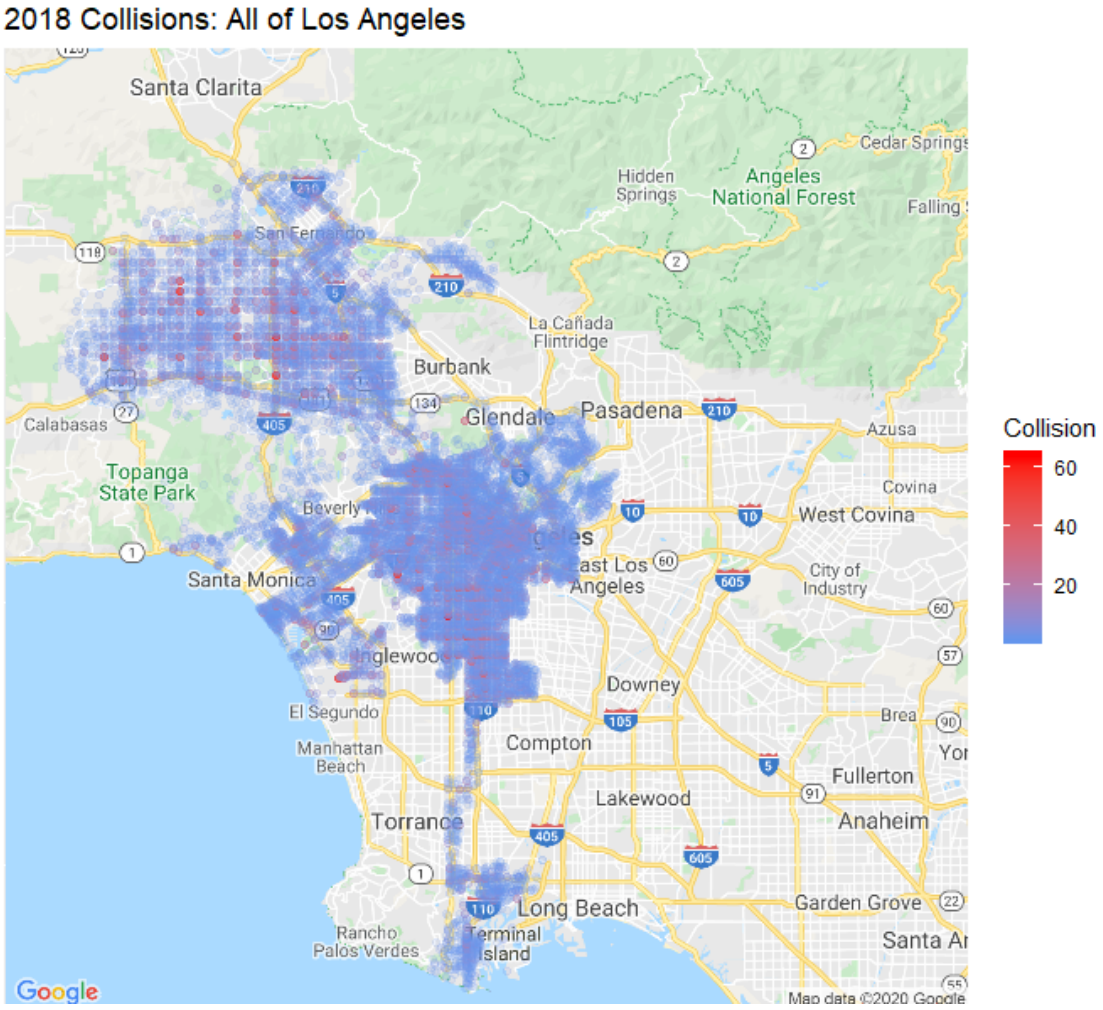
5% of collisions have no associated `cross_street`. Otherwise, this list has a lot of overlap with the previous list. The obvious next step is to look at the most common intersections for collisions by combining these two fields.

	location	cross_street	count	percentage
	<chr>	<chr>	<int>	<dbl>
1	SHERMAN WY	SEPULVEDA BL	247	0.05
2	TAMPA AV	NORDHOFF ST	240	0.05
3	VAN NUYS BL	ROSCOE BL	220	0.05
4	SHERMAN WY	WOODMAN AV	215	0.04
5	SHERMAN WY	WHITSETT AV	210	0.04
6	ROSCOE BL	VAN NUYS BL	195	0.04
7	SLAUSON AV	WESTERN AV	195	0.04
8	BURBANK BL	SEPULVEDA BL	191	0.04
9	MANCHESTER AV	FIGUEROA ST	182	0.04
10	SHERMAN WY	BELLAIRE AV	177	0.04

Intersections with the most collisions

exceptions: the components of row 2 (Tampa Ave. and Nordhoff St.) don't appear in either the most common `location` or `cross_street`. Even the most collision-prone intersections account for only a small proportion of overall collisions.

Now it's time to take advantage of the geographic coordinates in the data and start mapping collisions. As a reminder, there are a small number of collisions that do not have valid latitude/longitude data and so are excluded from this section.

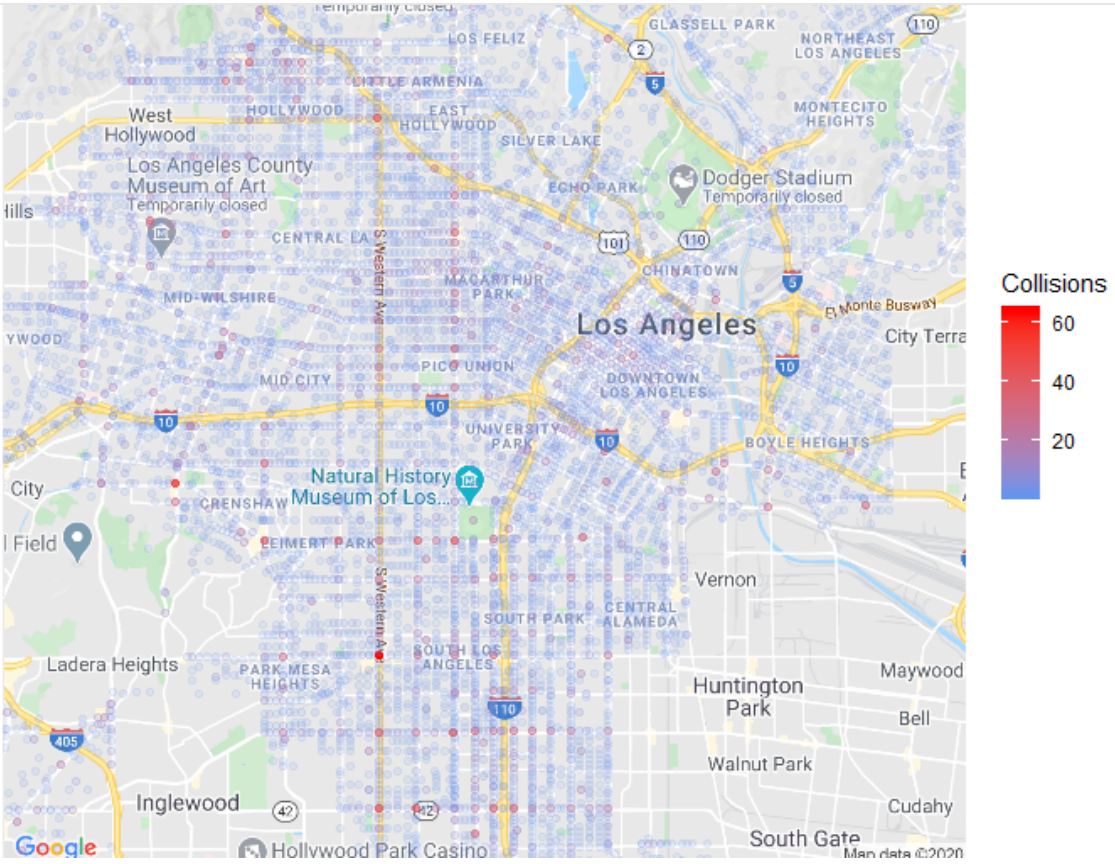


2018 collisions for all of Los Angeles. Each point is a unique latitude/longitude coordinate.

- The spatial distribution of points shows the interesting shape of LA.
- Blue [Red] points indicate latitude/longitude coordinates with a low [high] number of 2018 collisions.
- Even on this zoomed-out map, high collision coordinates are visible in the Valley (the northern part of the map) and the central and eastern parts of the city.

This map is pretty cluttered. For a better view, I'll zoom in to a window showing much of central and downtown Los Angeles.

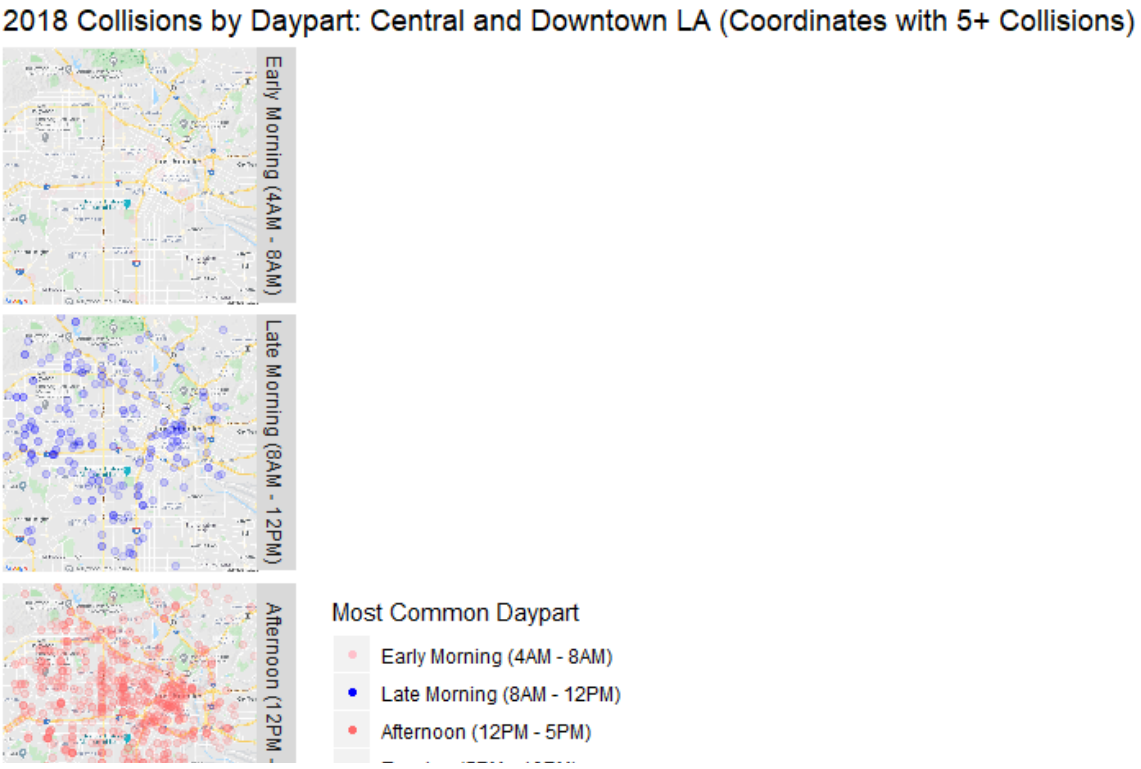
2018 Collisions: Central and Downtown LA

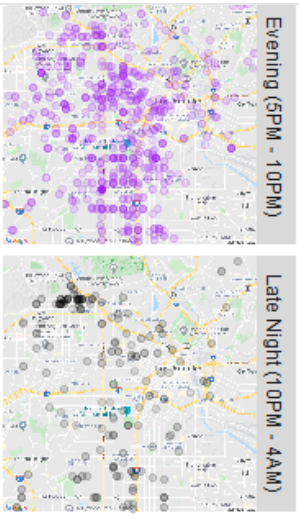


2018 collisions for selected areas of Los Angeles

- A number of medium and high collision coordinates are clearly visible. Many of these high collision points tend to occur on various intersections of the same street.

The previous maps showed overall collisions. How does this data look if I add time of day?

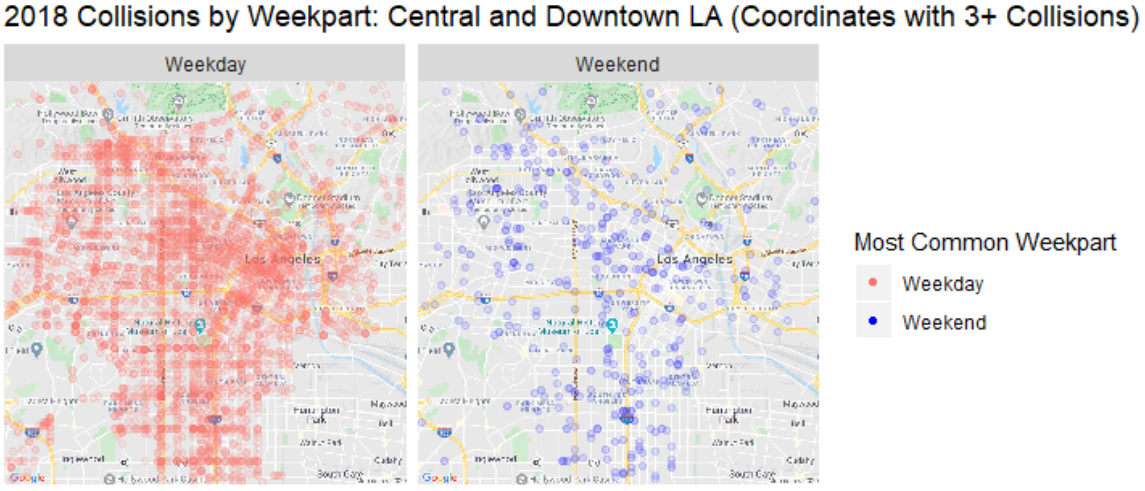




2018 collisions by most common daypart for selected areas of Los Angeles

- This plot only includes coordinates with 5+ collisions. Each coordinate is assigned to the daypart in which most of its collisions occur. Coordinates with ties between dayparts are removed.
- There are no coordinates with a majority of collisions occurring in the `Early Morning`. This makes sense given what I found in the Collisions by Time analysis.
- Many coordinates have most of their collisions in the `Afternoon` and `Evening`. This result also matches the results of the Collisions by Time section.
- Interestingly, there's a cluster of coordinates where `Late Night` collisions are common.

Let's look at a similar map broken out by weekday/weekend.



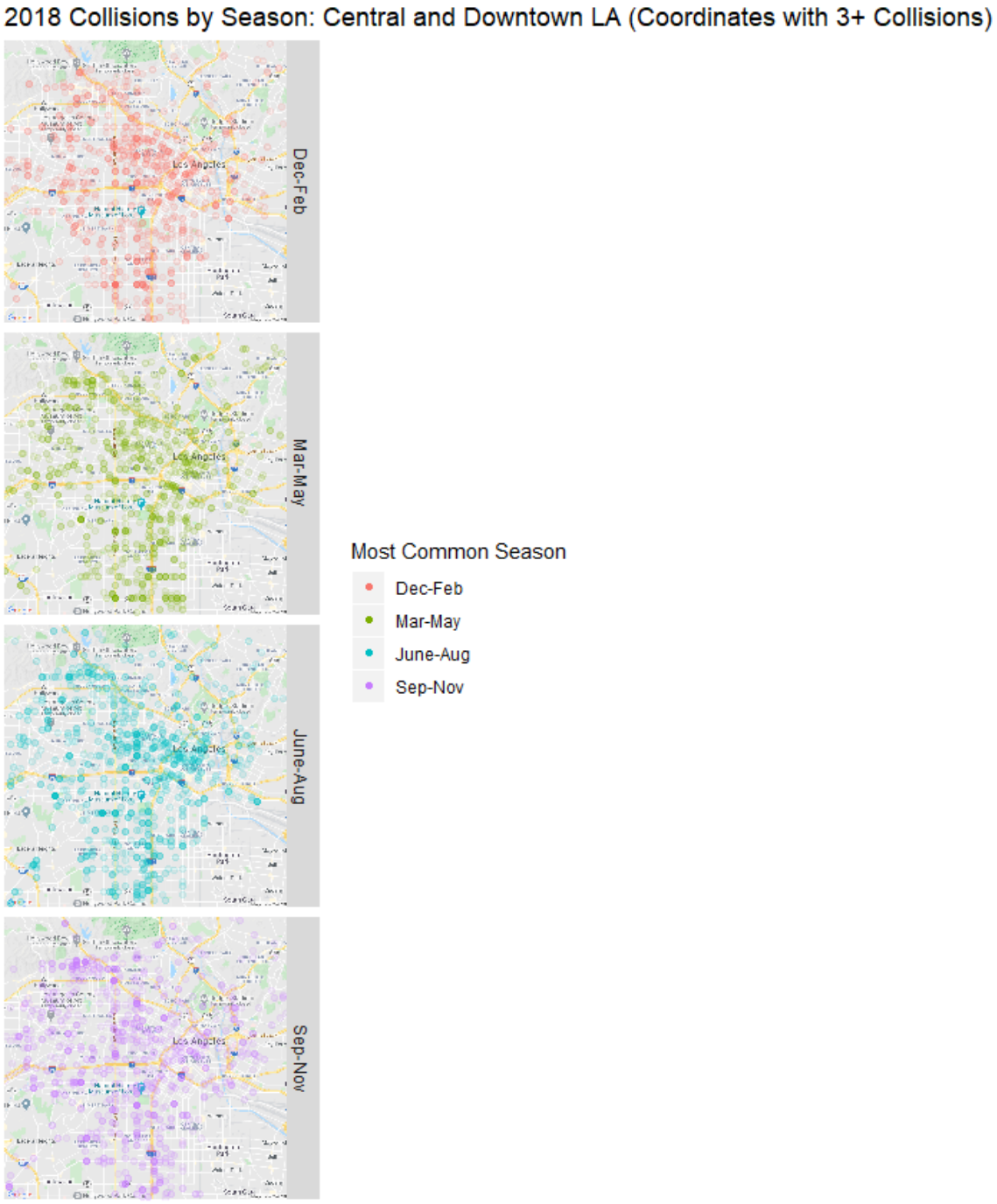
2018 collisions by most common weekpart for selected areas of Los Angeles

- This plot only includes coordinates with 3+ collisions. Each coordinate is assigned to the weekpart in which most of its collisions occur. Coordinates with ties between

There are obviously more days and collisions in the `Weekday` bucket.

- This map identifies areas with many weekend collisions. It might also be interesting to include part or all of Friday in the `Weekend` bucket.

To conclude this section, I plot collisions by time of year.



2018 collisions by most common season for selected areas of Los Angeles

- This plot only includes coordinates with 3+ collisions. Each coordinate is assigned to the season in which most of its collisions occur. Coordinates with ties between seasons are removed.

- Given that Los Angeles doesn't have distinct seasons like fall or winter, there may be other ways to split up the year for a plot like this.

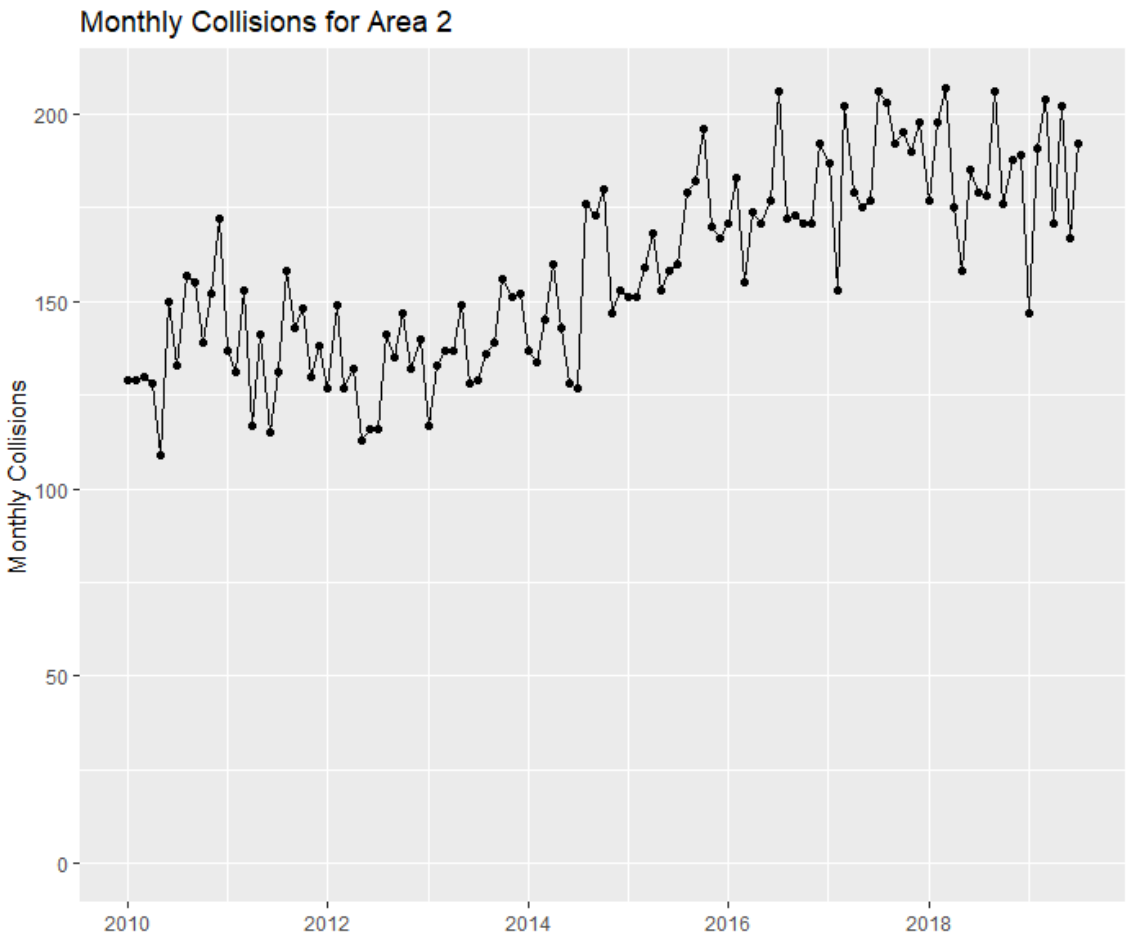
These are my conclusions for the Collisions by Geography section:

- Using the `location` and `cross_street` fields, in addition to mapping, it is possible to identify the most accident-prone coordinates in Los Angeles.
- The most common dayparts for collisions are the `Afternoon` and `Evening`.
- Many more collisions occur during the weekdays than the weekend. Mapping collisions by weekpart shows areas where weekend collisions are more common.
- More collisions happen in the summer than winter months.

🧠 Predicting Collisions

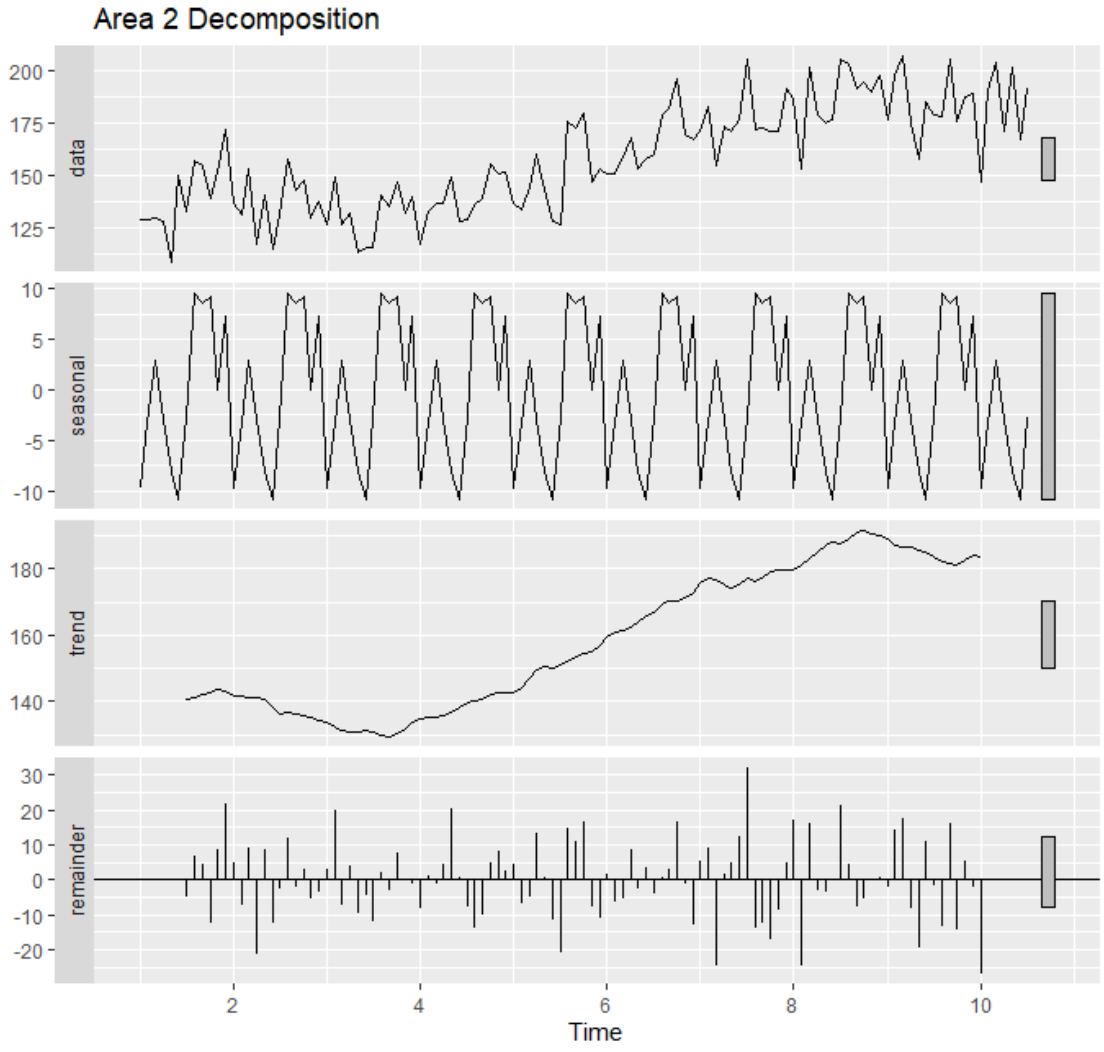
The final section deals with trying to predict collisions. I specifically try to predict the number of collisions that will occur per month and `area`.

I'll start by looking at an example of the collision time series for a single `area`:



- The trend for `area` 2 generally matches the trend of overall monthly collisions (this is in the Collisions by Time analysis).

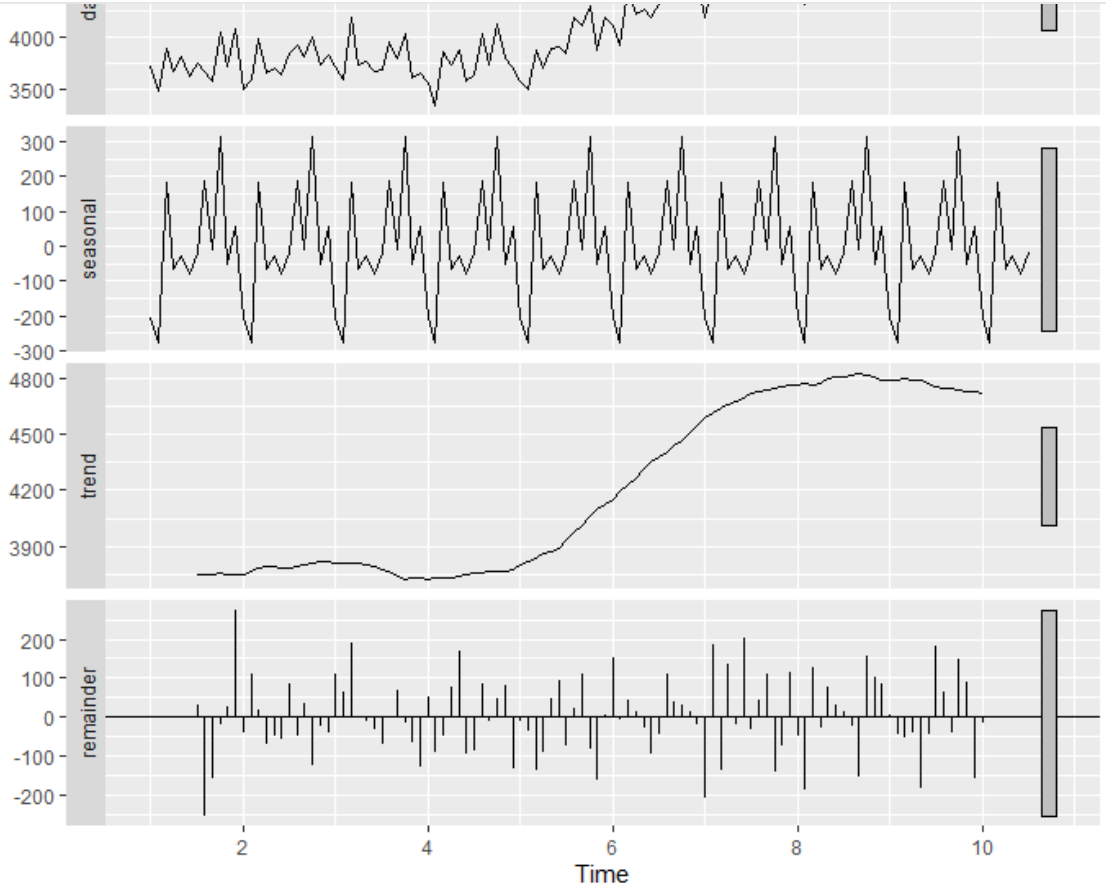
Let's decompose this time series into trend, seasonality, and remainder components. As an aside, I'll be focusing on the analysis results in this post and won't delve into the theory of the time series methods I'm using. However, there are lots of resources available online if you'd like to learn more!



Trend, seasonality, and remainder components for "area" 2 (Rampart)

- Keep the shape of the seasonality curve in mind. I'll compare it against the decomposition of overall monthly collisions next.
- The trend looks similar to what we saw in the previous graphic.

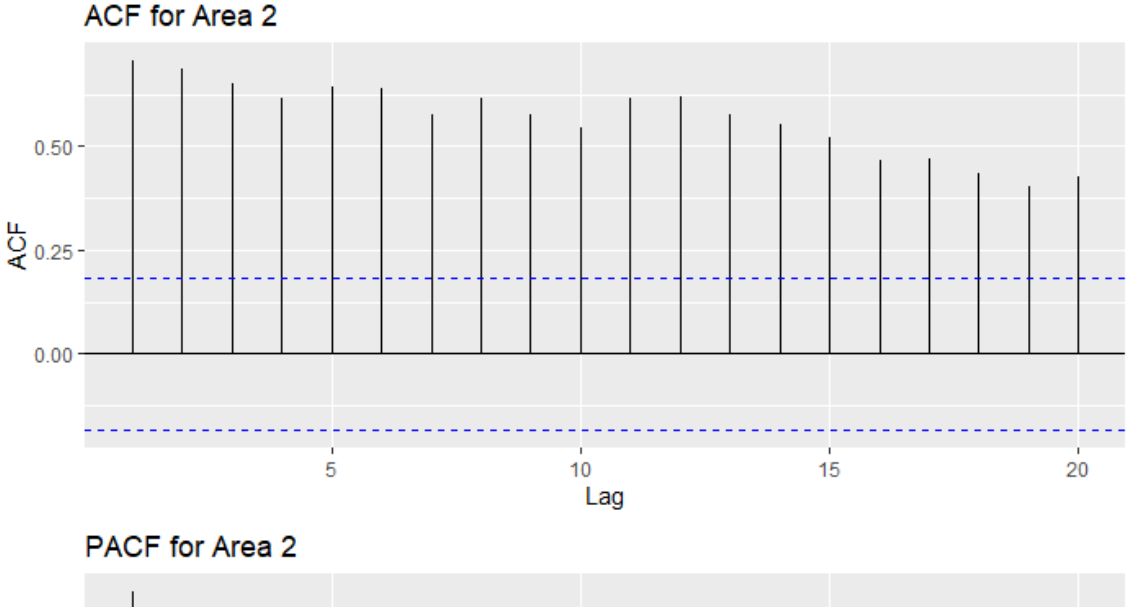
Let's compare the `area` 2 decomposition to the overall monthly collisions decomposition.

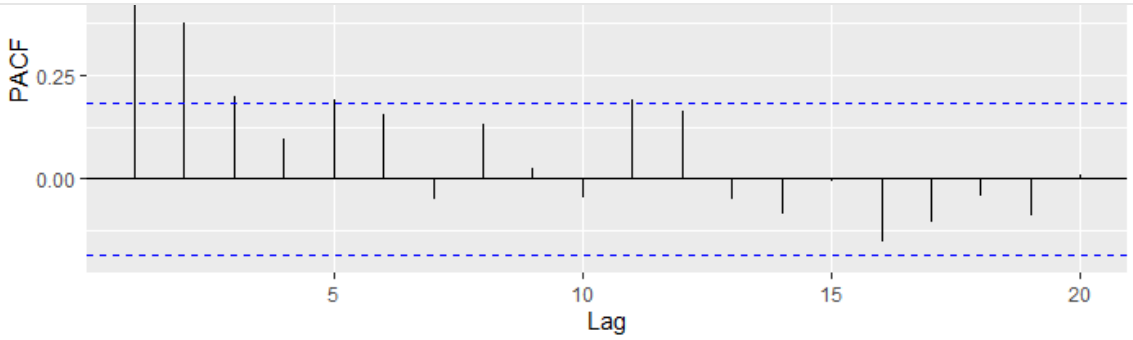


Trend, seasonality, and remainder components for overall data

- The seasonality curve for the overall data looks very different as compared to `area 2`! This indicates that different areas can have different dynamics.

Next, I'll look at the auto-correlation function (ACF) and partial auto-correlation function (PACF) for `area 2`. The ACF analyzes how correlated lagged values of collisions are to the current value. The PACF shows how much previously unexplained variance each lag explains.





ACF and PACF for “area” 2 (Rampart)

- These plots indicate that `area` 2 collision values are correlated with their lags. However, after the first two lagged values, additional lags are not accounting for much unexplained variance.
- So, any model predicting `area` 2 collisions should include at least 2 lagged terms.

I try a few different model types to predict collisions:

- 3 and 6 month moving average models (MA)
- Auto Regressive Integrated Moving Average (ARIMA)
- Prophet library

MA models average past values to generate predictions and are the simplest type of time series model. ARIMA models can use past values, differencing, and previous errors. Prophet is an additive forecasting model where non-linear trends are fit with yearly and monthly seasonality.

Prophet is much newer than ARIMA or MA models. You can find more info on it here:

Prophet

Prophet is a forecasting procedure implemented in R and Python. It is fast and provides completely automated forecasts…

facebook.github.io

I evaluate each model on the final 12 months of data (August 2018 to July 2019) with the following metrics:

- Mean absolute percentage error (MAPE): average absolute percentage difference between predictions and actual values
- Bias: average percentage difference between predictions and actual values

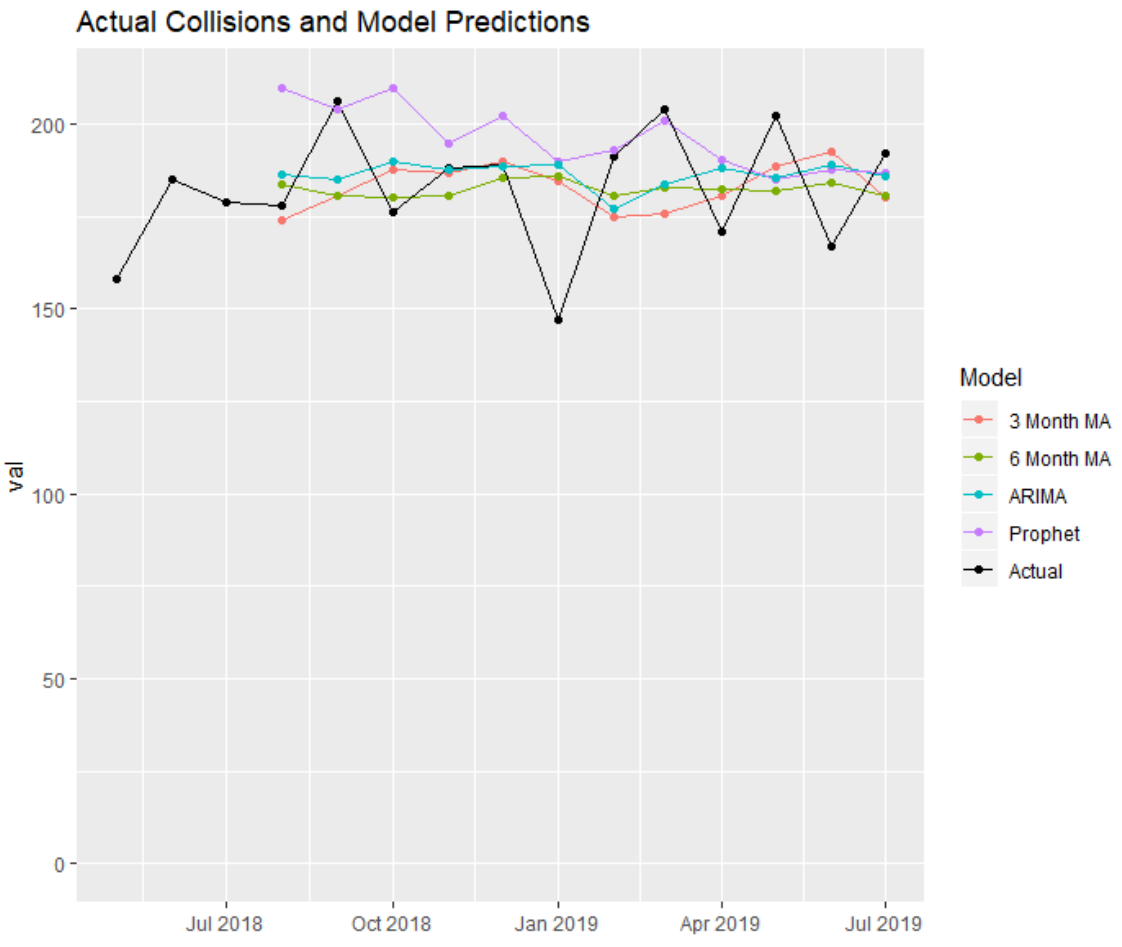
I auto-fit ARIMA models for each `area`. So each one can have a different p, d, and q. I also test multiple Prophet model specifications (and show the best results below). Here are the model MAPE and bias results averaged over all areas:

mean_m3_mape	mean_m6_mape	mean_arima_mape	mean_prophet_mape
0.0796	0.0797	0.0797	0.0915
mean_m3_bias	**mean_m6_bias**	**mean_arima_bias**	**mean_prophet_bias**
0.0113	0.0137	0.0234	0.069

Multiple model results averaged over all areas in the data

- Overall, I think these results are pretty good!
- The 3 and 6 month MA models have very similar performance.
- The ARIMA model has a similar MAPE to the MA models, but worse bias.
- The Prophet model has the worst results by far (even after tuning).
- It's surprising to me that the MA models (the simplest ones) have the best performance!

Let's look at model predictions for `area` 2 only.

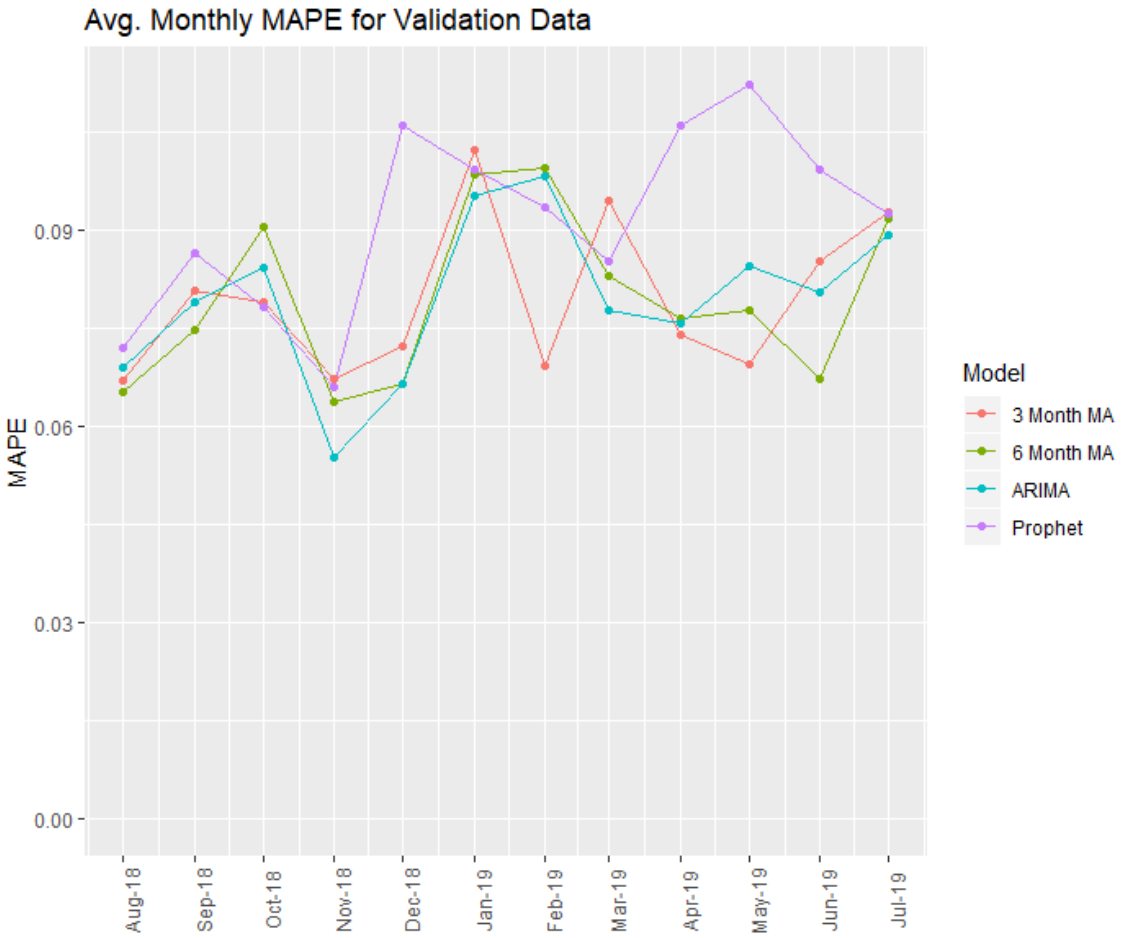


Model predictions vs actual collisions for "area" 2 (Rampart)

2019.

- The 6 month MA model predictions don't vary that much.
- The MA and ARIMA models seem to make conservative predictions that don't capture the fluctuating nature of the data.

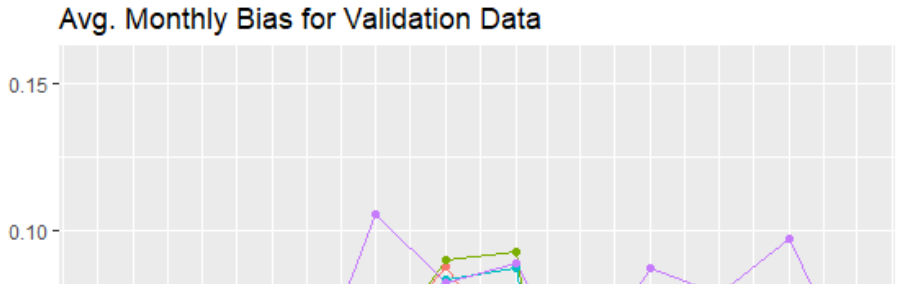
Next, I'll look at the average MAPE per month per model. This plot includes all areas.

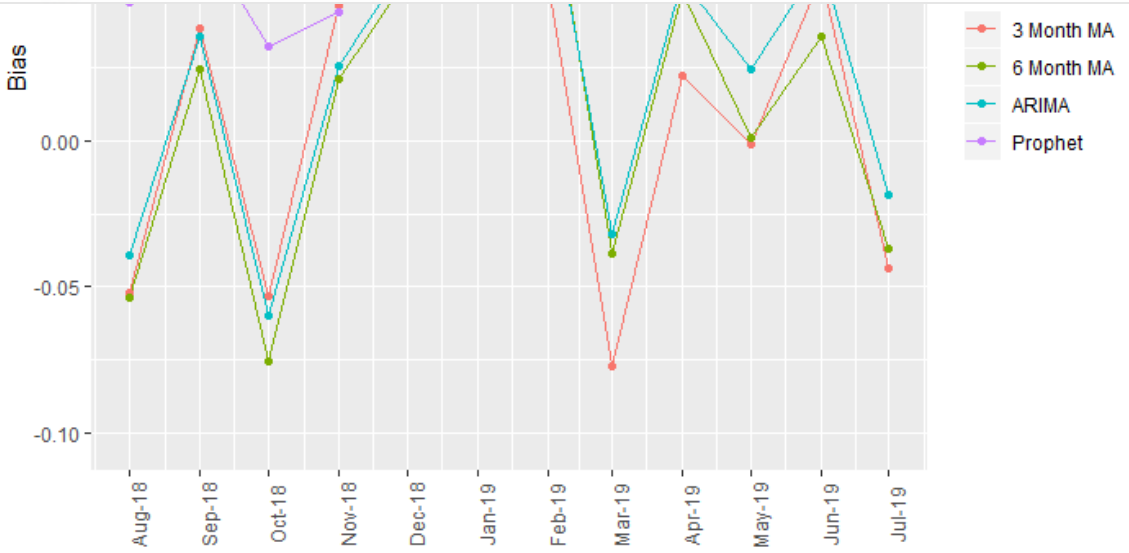


Average monthly MAPE for each model (all areas)

- For the first half of the validation data, the MAPE for the MA and ARIMA models move together.

It's also worth looking at the average bias per month per model.

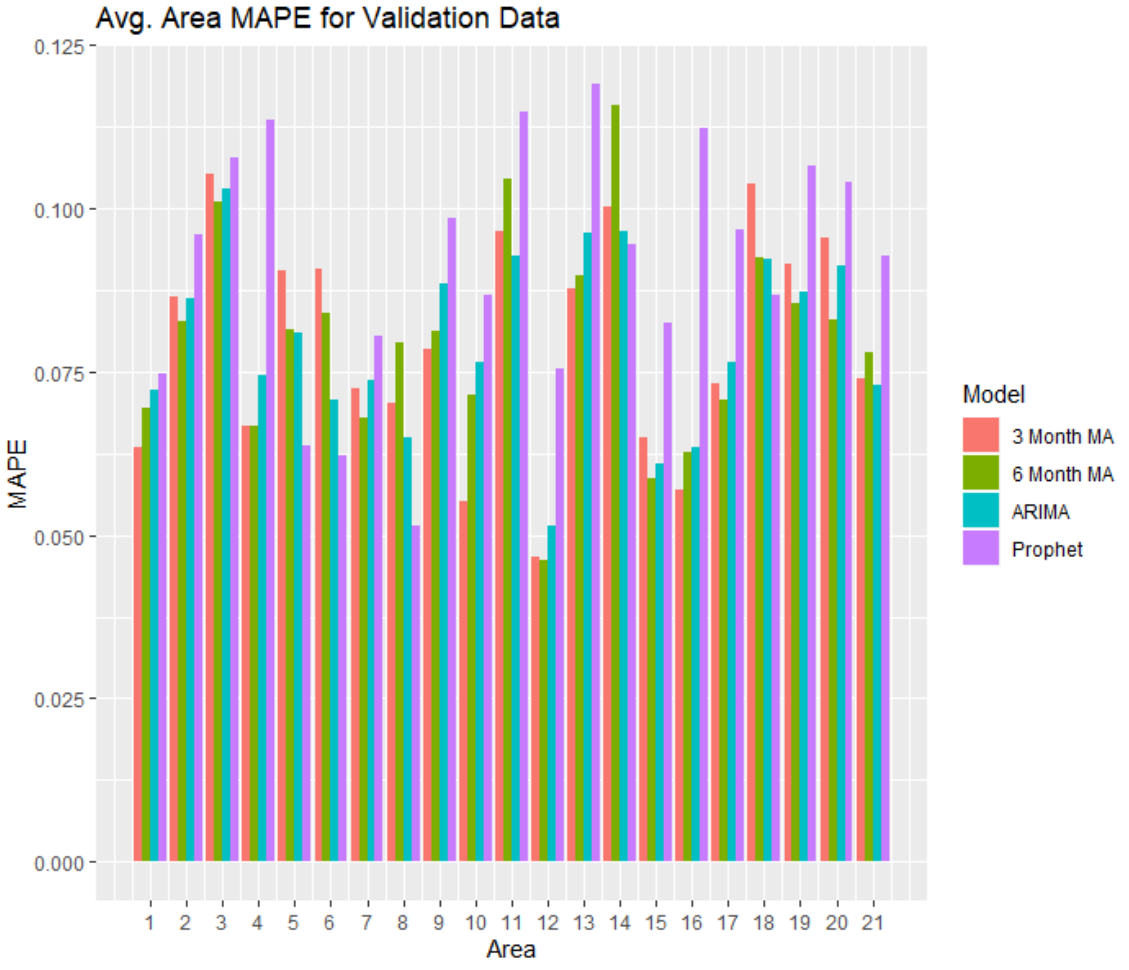




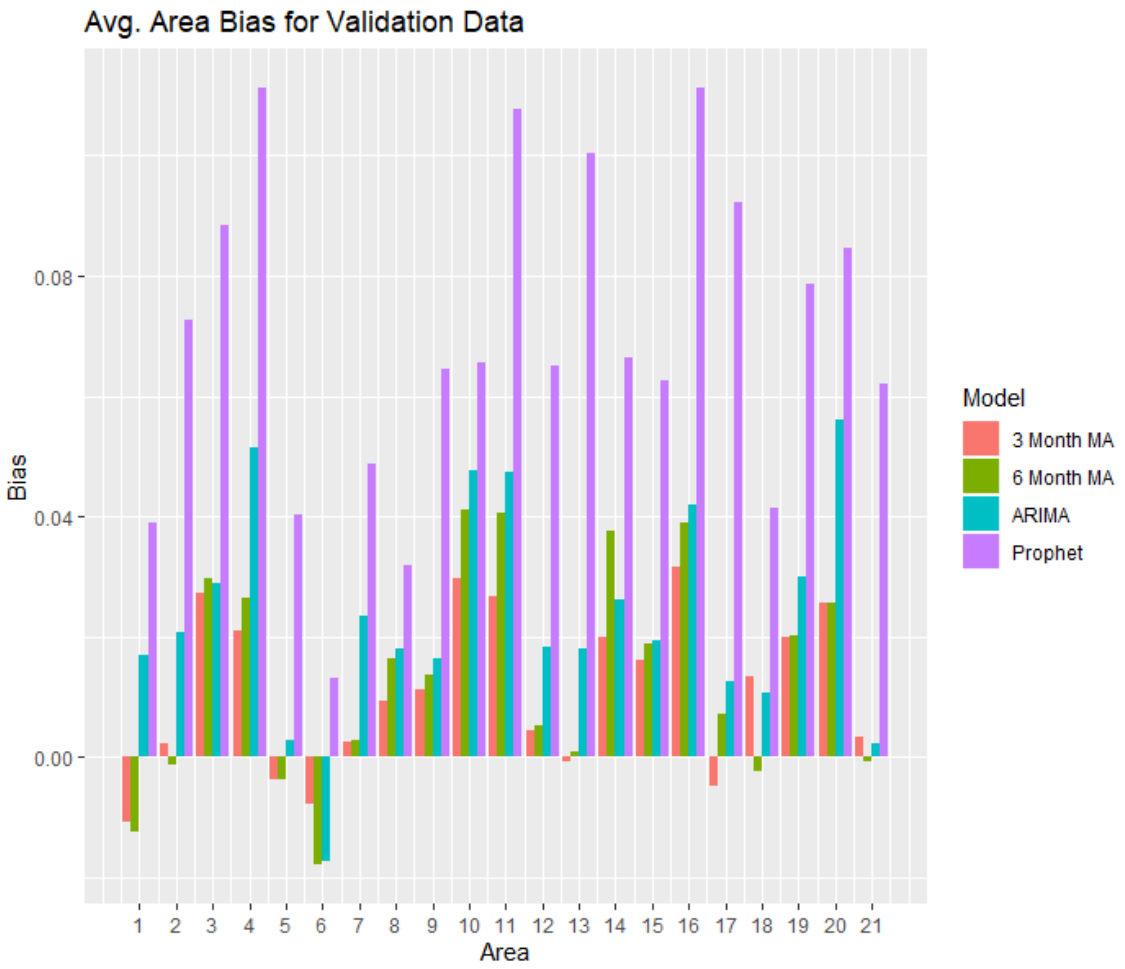
Average monthly bias for each model (all areas)

- The bias for the MA and ARIMA models move together.
- The Prophet model has positive bias (overpredicts) for the entire validation set.

Now, I'll look at the average MAPE and bias per model per `area`.



Average MAPE per model per "area"



Average bias per model per “area”

- Average bias also varies substantially by `area`.
- Surprisingly, most models have positive bias (which indicates over-prediction) for most areas!

Next, I’ll look at the worst `area`/month predictions per model.

- 3 Month MA Model

```
# A tibble: 1 x 7
  month      area  collisions m3_mape m6_mape prophet_mape arima_mape
  <date>     <chr>      <dbl>   <dbl>   <dbl>        <dbl>      <dbl>
  2018-09-01 14           233   0.260   0.162       0.0895      0.183
```

- 6 Month MA Model

```
# A tibble: 1 x 7
  month      area  collisions m3_mape m6_mape prophet_mape arima_mape
```

- ARIMA Model

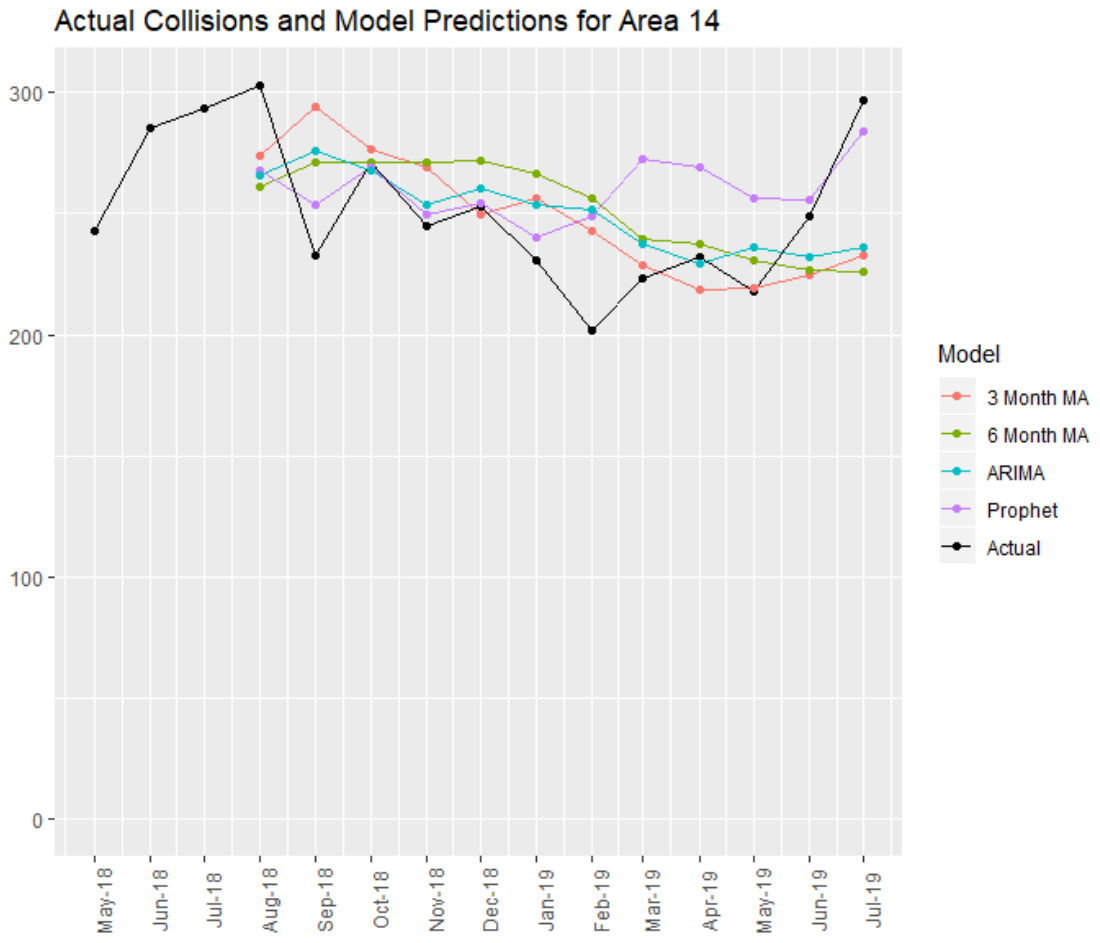
```
# A tibble: 1 x 7
  month      area  collisions m3_mape m6_mape prophet_mape arima_mape
  <date>     <chr>      <dbl>   <dbl>   <dbl>        <dbl>      <dbl>
2 2019-01-01 2            147   0.254   0.265        0.290      0.286
```

- Prophet Model

```
# A tibble: 1 x 7
  month      area  collisions m3_mape m6_mape prophet_mape arima_mape
  <date>     <chr>      <dbl>   <dbl>   <dbl>        <dbl>      <dbl>
2 2019-05-01 11           166   0.133   0.140        0.338      0.169
```

- `area` 14 shows up twice. Both January and February of 2019 show up.
- It's interesting to see cases where all models struggled (Jan 2019 in `area` 2) vs. cases where one model in particular struggled (Sept 2018 in `area` 14).

Based on the worst predictions, I'll zoom into `area` 14.



- The trend for `area` 14 is completely different than for `area` 2 above.
- All models except Prophet miss the spike in July 2019.

Here are my conclusions for the Collision Prediction section:

- Overall monthly performance is not bad (<10% MAPE and bias in most cases).
- However, MA models have the best performance which indicates that longer-term lagged data, differencing, and previous errors don't improve error rates. This is surprising, but suggests that the number of collisions per `area` /month is largely random within a certain range.
- Trends seem to vary by `area` . This should in theory be addressed by the ARIMA and Prophet methods which fit a separate model per `area` .
- To improve these models, I would want to dig into one or two areas in-depth and attempt to understand the trends. Getting data about the specific traffic patterns of an `area` would definitely help too. Additional data sources (like weather) could also be promising to explore.

🏁 Conclusion

This concludes my analysis on Los Angeles collision data! Feel free to get in touch if you have other approaches to these questions.

You can find all of the code I used in the GitHub repository below:

jai-bansal/los-angeles-collision-analysis

This repo contains materials surrounding an analysis of collision data in Los Angeles. Below is a description of the…

github.com

If you enjoyed this post, check out some of my other work below!

How to Use Random Seeds Effectively

This post is about an aspect of the machine process that doesn't typically get much attention: random seeds.

towardsdatascience.com

Techniques to bring your next training course idea to life

medium.com

Thanks for reading!