Behavioral Health & Criminal Justice: An Analysis of Nebraska's Sarpy County

by Trevor Jurjevich

May 2024

University of Nebraska at Omaha

Master of Science in Data Science (Mathematics Concentration)

Academic Advisor: Dr. Fabio Vitor, UNO

External Advisors: Kim Kalina & Nick Chadwell, Region 6 Behavioral Healthcare

Abstract

Much research has been done to study the relationship between incarceration and mental health. Although partially related to external factors, such as the COVID-19 pandemic, the consensus is these two populations intersect to an alarming degree. Region 6 Behavioral Healthcare is an Omaha-based governmental organization responsible for monitoring and developing mental health and substance abuse services for providers in the greater Omaha area. Teaming up with Region 6, the initial goal is to analyze both Sarpy County Department of Corrections (SCDC) data, as well as Department of Behavioral Health centralized datasets (CDS) to discover potential disproportionalities and underlying correlations. Following this exploratory data analysis, several ARIMA-based and Prophet models are created to forecast weekly SCDC bookings and monthly mental health encounters for Sarpy County residents. Many of the results from exploratory analysis, as well as observed trends from model forecasts, fall right in line with Region 6's expectations given the current state of behavioral health coverage in Nebraska. Forecasts specifically related to mental health encounters also act as assets toward the planning of future programs and initiatives.

Table of Contents

1. Introduction

Many individuals take advantage of mental health services, and even more so in recent years with the destigmatization of seeking help. However, for most of these people seeking various types of treatment, the complexity of the very system they are a part of goes unnoticed. The Nebraska state legislature is responsible for divvying up federal block grant money to various branches, one of which is the Department of Health & Human Services (DHHS). DHHS can be further divided into several entities, one of which is the Division of Behavioral Health (DBH). Region 6 Behavioral Healthcare (referred to simply as Region 6 throughout the remainder of this report) is one of six "regions" within the state of Nebraska that fall under the umbrella of DBH, and they are responsible for "planning, developing, funding, monitoring, and evaluating" various services for mental health providers within their coverage area, which include Cass, Dodge, Douglas, Sarpy, and Washington counties (Region 6, 2018). Figure 1 displays Nebraska's six DBH regions, with Region 6 situated on the eastern border of the state.

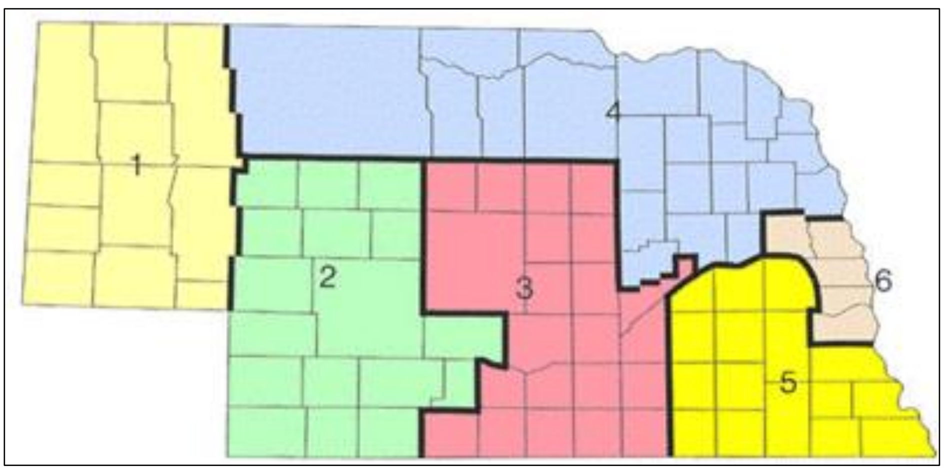


Figure 1. Map of Nebraska's six behavioral health regions. Region 6 (tan) oversees Douglas & Sarpy Counties, two of Nebraska's most populous counties (Nebraska Disaster Behavioral Health, 2022).

Region 6's counties cover urban and rural areas alike, which means understanding their consumer population becomes even more important. With Omaha (making up the overwhelming majority of Douglas County—Nebraska's most populous county) practically maxed out in terms of its future growth, eyes are increasingly on other portions of the region to study both similarities and differences between consumers and their received services. Sarpy County is considered one of the quickest growing counties in terms of both economic and population growth ("Sarpy's economic growth continues", 2021), and is the primary focus of this project.

As mentioned previously, Region 6 deals directly with mental health providers and their services. One other topic they introduce is the role that mental illness plays within the justice system, and more specifically, local jails. Region 6 partners up with the National Association of Counties (NACo) for the *Stepping Up Initiative,* in which the ultimate goal is to reduce the number of individuals suffering from mental illness in local jails. It is known that nearly two million people each year who suffer from a serious mental illness are booked into jails. The hope is that decreasing these bookings will in turn reduce other asymmetries between this particular sub-population and those not suffering from serious mental illnesses, such as time spent in jail and the consistency of bookings.

This project partners with Region 6 to examine several datasets pertaining to provider encounters for those residing in Sarpy County, as well as jail booking data from the Sarpy County Department of Corrections. The initial phase of this project conducts exploratory analysis on the provided datasets. Several topics are covered in this early phase, including demographic comparisons within the datasets to estimates from the American Community Survey, as well as determining statistically significant mean and proportion differences between dataset subgroups.

The second phase of the project revolves around forecasting three different numbers: the population of Sarpy County, the number of weekly jail bookings in Sarpy County Department of Corrections, and the number of monthly mental health encounters stemming from Sarpy County residents. Several model types are used to create forecasts, which include exponential smoothing, ARIMA, SARIMA, ARIMA with exogenous variables, and *Facebook*'s Prophet. Early analysis and forecasting aim to determine short- and long-term trends among the various provided time series. Additional benefits from forecasting are also discussed, as well as different approaches that could be taken for eventual research to determine predictions relating to the jail and provider populations.

2. Project Datasets

A handful of datasets are either provided from Region 6 or downloaded via the United States Census Bureau website. The first dataset is a master list of all bookings from July 1st, 2022, through December 31st, 2023, from the Sarpy County Department of Corrections (SCDC). Distinct bookings can be identified from the *BookingID* field, while distinct individuals can be identified from either the *InmateID* or *InmateName* field. Although this jail booking information is made publicly available online (personal identifiable information is therefore not an issue), *InmateID* is used to identify individuals throughout early analysis, forecasting, and therefore, the presentation of research findings.

Each row in this dataset corresponds to a *InmateID/BookingID/PrimaryCharge* combination, where *PrimaryCharge* contains the specific booking charge that an individual is assigned. Obviously, individuals can be booked on more than one charge during an arrest, so oftentimes there are multiple rows for a particular *BookingID*. Some other important fields in this dataset that are referenced during later analysis include *BirthDate, Race* (possible values are “A”, “B”, “H”, “I”, “P”, “U”, or “W”), *Sex* (possible values are “M”, “F”, “U”), *ZIP, BookingDate*, *ReleaseDate*, *ChargeCrimeType* (codes corresponding to felony type, misdemeanor type, etc.), and *SMI* (“serious mental illness”). It is important to note that the *SMI* field is joined to this dataset by a separate dataset with inmate names and whether or not they are considered to have a serious mental illness. Data is input by hand into SCDC’s system, so there are unfortunately many erroneous values throughout the data.

The second type of dataset provided by Region 6 contains provider encounters that took place within Region 6’s assigned counties (Cass, Dodge, Douglas, Sarpy, and Washington). These datasets are divided by fiscal year, where fiscal year [Year] contains encounters from July 1st, [Year]-1 through June 1st, [Year]. Fiscal years 2019 through 2023 are made available for analysis and forecasting.

Before explaining the structure of these datasets, it is important to describe how they are compiled. Region 6 pulls these encounters from the Division of Behavioral Health (DBH), and the datasets themselves are referred to as centralized datasets (CDS). When an individual gets admitted into a provider that falls under Region 6’s umbrella, three separate indices are added to DBH’s database in a *StartEnd* field for that single encounter: Admit, Start, and End. These can be thought of as timestamps in the individual's encounter. With that being said, the *EncounterID/StartEnd* index to use for data analysis primarily depends on the scope of analysis wanting to be completed. For example, when looking at something like demographic frequencies across all encounters, the focus would be on the *Admit* row. It is possible that field values can change for a record across these three sub-indices, so it is assumed that looking at the original *Admit* information is the best "root" source for fields that have the potential to change across the period of the encounter/treatment.

Another idea to understand with these datasets is that encounters can extend across multiple fiscal years. This occurs when an individual has an admission date falling under one fiscal year, but their encounter has not ended within that same fiscal year. If this fact is not taken into consideration, any analysis/forecasting based on admission date will be observing 2+ encounters, when in reality there is only one encounter that happens to be listed across multiple FY datasets. Some notable fields that come up for analysis include *EncounterID, ConsumerID* (unique key for each individual)*, date_admission, date_discharge, ServiceType* (general type of service provided, either "Mental Health" or "Substance Use Disorder")*, ServiceFullName* (specific type of service provided), *Date_Birth, Sex, Race* and *HadTrauma* (whether or not the individual experienced trauma).

The final types of datasets used for analysis and forecasting come directly from the United States Census Bureau. Two sets of numbers are used from these datasets: official decennial populations (standard once-every-ten-year Census) and American Community Survey (ACS) population estimates, which are conducted once a year to fill gaps that are otherwise unknown if observing only decennial Census data.

2.1 Early Exploratory Analysis

There are many queries that can be explored pertaining to the content of these datasets. Given the scope of this project, as well as the nature of the datasets, demographic frequency comparisons are the focus of this early analysis. Utilizing 2022 ACS Sarpy County data, one can compare estimated breakouts between the entire Sarpy County population and the breakouts that present themselves within the aforementioned datasets. Additionally, various supplemental tests of independence are explored to determine possible correlations among variables within datasets.

Recall that the jail dataset contains Sarpy County Department of Correction bookings, and spans 18 months from July 2022 through December 2023. There are a total of 5421 *BookingID*s from 4183 unique *InmateID*s. The total number of bookings per individual in this time span ranges from 1 to 14, with a heavy right-skewness. Tables 1-3 provide breakouts for age, gender, and race. Frequencies and percentage makeup are grouped by those labeled as having an SMI, those without an SMI, and a combined total of the two (to represent the entire jail population). The final two columns contain frequencies and percentage makeups from the 2022 ACS estimates for the same aggregate groupings. Frequencies for age are grouped on the *BookingID* level, whereas gender and race are grouped on the *BookingID/InmateID* level (age has the potential to increase for those with more than one booking, whereas gender and race theoretically should not change across bookings for an *InmateID*).

Table 1. SCDC vs. 2022 ACS Age Breakouts.

	SMI		**Non-SMI**		**Overall**		**2022 ACS Estimates**	
	N	%	*N*	%	*N*	%	*N*	%
0-14	0	0%	0	0%	0	0%	42545	21.7%
15-19	11	3.5%	242	4.8%	253	4.7%	13966	7.1%
20-24	64	20.2%	914	17.9%	978	18.0%	12369	6.3%
25-34	97	30.6%	1776	34.8%	1873	34.6%	26327	13.4%
35-44	106	33.4%	1279	25.1%	1385	25.6%	31120	15.8%
45-54	29	9.2%	602	11.8%	631	11.7%	22867	11.6%
55-59	7	2.2%	138	2.7%	145	2.7%	10442	5.3%
60-64	1	0.3%	86	1.7%	87	1.6%	11513	5.9%
65-74	2	0.6%	59	1.2%	61	1.1%	15170	7.7%
75-84	0	0%	2	0%	2	0%	8192	4.2%
85+	0	0%	2	0%	2	0%	2042	1.0%

Table 2. SCDC vs. 2022 ACS Gender Breakouts.

	SMI		**Non-SMI**		**Overall**		**2022 ACS Estimates**	
	N	%	*N*	%	*N*	%	*N*	%
Male	115	69.7%	2891	72.0%	3006	71.9%	99789	50.8%
Female	50	30.3%	1126	28.0%	1176	28.1%	96764	49.2%

Table 3. SCDC vs. 2022 ACS Race Breakouts.

	SMI		Non-SMI		Overall		2022 ACS Estimates	
	N	*%*	*N*	*%*	*N*	*%*	*N*	*%*
White	139	84.2%	2921	74.9%	3060	75.3%	157318	91.5%
Black/African American	23	13.9%	889	22.8%	912	22.4%	8023	4.7%
American Indian/Alaska Native	3	1.8%	46	1.2%	49	1.2%	1192	0.7%
Asian	0	0%	42	1.1%	42	1.0%	5326	3.1%
Native Hawaiian/Pacific Islander	0	0%	2	0%	2	0%	113	0%

Age breakouts do not appear different from each other when comparing between the SMI and Non-SMI groups. However, there are noticeable differences between the overall and ACS groupings. Almost 80% of jail bookings are coming from the age 20-44 population, whereas Sarpy County's makeup of the same age grouping is only around 35%. The gender splits also present some discrepancies between the two data sources. Sarpy County presents a roughly equal male-to-female ratio, but that same ratio for those present in the jail dataset is notably skewed toward the male population. Although this large difference certainly stands out, it is not necessarily unexpected. According to recent U.S. Department of Justice research, the male-to-female ratio for local jails on the national level has been hovering around 85% to 15% for the last decade, with 2021 coming in at 87% to 13% (Zeng 2022). Similarly, there is a sizeable difference in the White & Black/African American makeup between the overall dataset and ACS numbers. The same U.S. Department of Justice research shows an even higher percentage of Black/African American inmates and a lower percentage of White inmates. While the difference in this comparison appears large, it is not something unexpected when compared to jails on the national level.

Demographics, characteristics, and other factors are also a topic of interest between those labeled as having a serious mental illness and those without one. One potential difference is the average number of bookings for an individual labeled as having an SMI vs. those without one (Non SMI). Typically, one could use a two-sample t test to determine whether the SMI and Non-SMI average bookings are significantly different, however, that cannot be performed. Both the SMI and Non-

SMI lists of *InmateID* are heavily right-skewed, that is, many individuals have only a single booking within this time frame. This shift in data breaks one of the assumptions of two sample t-tests in that the data should be approximately normally distributed. To combat this, a non-parametric sample distribution test can be used to determine statistical significance between the two groups' sample means. For this test, as well as any other subsequent statistical tests, a significance level α of 0.05 (confidence level of 0.95) will be implied. With that being said, a Mann-Whitney U test—whose null hypothesis states the two groups average number of bookings are equal—determines that the SMI and Non-SMI groups have a distinct underlying average number of bookings (SMI=1.92 bookings, Non-SMI=1.27 bookings; test statistic=447400, p-value=3.35e-28).

Also of interest is the average number of days between bookings. For obvious reasons, this can only be calculated by looking at individuals with more than one booking during this 18 month time frame. The number of days between bookings of the SMI/Non-SMI groups also does not appear normally distributed, so a second Mann-Whitney U test is conducted to determine possible sample mean differences. However, this second test does not attribute a difference based on labeled group (SMI=126.56 days, Non-SMI=146.74 days; test statistic= 27253.5, p-value=0.22). This lack of statistical significance can likely be attributed to the low sample size of SMI inmates with more than one booking.

Charge type is also of interest when it comes to comparing the two groups within the dataset. In the raw dataset, the *ChargeCrimeType* field contains codes of what charge the individual is being booked on. These can take on a long list of different values, but they can be grouped into their own separate "buckets" based on the first character of the value ("I__": infraction, "M__": misdemeanor, "F__": felony, etc). Some brief data cleaning is performed to separate the "most severe" charge crime types by record into these buckets, and one can see counts of these records in Table 4.

Table 4. Grouped by SMI status, the number of bookings and their "most severe" charge.

	Felony	**Infraction**	**Misdemeanor**	**Missing**	**Other**
SMI	214	1	70	27	5
Non-SMI	1902	79	2413	603	107

Frequencies provide a good overview of breakouts on the SMI status level, but a statistical test can tell with greater certainty if there are differences between the discrete distributions of the two groups. Using a Chi-Square test of independence, it is determined that SMI status is indicative of the most severe charge crime type (test statistic=117.55, p-value=1.78e-24). This finding could indicate the difference in underlying behavior between individuals among the two groups, and perhaps the idea that SMI-labeled individuals are prone to commit more severe offenses. *The Stepping Up Initiative* also aims to reduce the disproportionality of jail stay durations between those with mental illnesses and those without. This idea can easily be explored within this dataset using the previous most severe charge grouping. Performing a similar crime type grouping by record, as well as pulling in the difference between the record's *ReleaseDate* and *BookingDate*, the analysis finds averages for each of the cells from Table 4. The results can be seen in Table 5, which displays these duration averages and the subsequent Mann-Whitney U test statistic/p-value.

Table 5. Grouped by SMI status, the average number of days individuals stayed in SCDC based on "most severe" charge.

	Felony	**Infraction**	**Misdemeanor**	**Missing**	**Other**
SMI	47.13	0	25.36	16.37	11.60
Non-SMI	21.40	0.41	5.47	4.86	8.07
Mann-Whitney U results	Tstat=240366, pvalue=3.14e-17	Tstat=30.5, pvalue=0.61	Tstat=108153, pvalue=5.60e-08	Tstat=12548.5, pvalue=7.58e-07	Tstat=390, pvalue=0.03

One can see that felonies and misdemeanors have significantly different average lengths of stay between the SMI/Non-SMI groups. SMI-labeled individuals are spending more than twice the amount of time in SCDC than their Non-SMI counterparts when their booking's most severe charge is a felony, and roughly five times longer when the most severe charge is a misdemeanor. Although the length of stay often depends on the severity of the *specific* charge the individual is being booked on, SMI-labeled individuals appear to be staying longer in jail for similar types of charges.

The mental health encounters datasets are also examined with a similar approach. Encounters and their corresponding data range from July 1st, 2018 through Jun 30th, 2023. Over the course of this period, there were 6188 encounters from 3719 distinct consumers. To get a sense of the Sarpy County consumer population, it is essential to observe demographic and socioeconomic frequencies. Before any analysis is done, some data cleaning and manipulation needs to be performed. First off, the datasets need to be combined into a single master dataset. Encounters then must be “deduped” to account for the *EncounterID*s that potentially reside across multiple fiscal years. Once this is completed, encounters can be filtered further to only include those with a Sarpy County residence. Lastly, any *date_admission* prior to July 1st, 2018 can be removed, since those encounters do not fall within the fiscal year 2018 through 2023 window. Comparisons are also made against the same ACS 2022 estimates, however, since this encounter data extends five years the frequencies are also split up by calendar year. Doing this allows two forms of measurement: one being an in-sample time-dependent comparison across the span of five calendar years, and the other being a direct comparison between calendar year 2022 and the ACS 2022 estimates for Sarpy County. The latter of comparisons can be seen in Table 6.

Table 6. CDS Calendar Year 2022 vs. 2022 ACS Demographics Breakouts.

		CDS CY 2022		2022 ACS Estimates	
		N	*%*	*N*	*%*
Age	**0-14**	71	8.05%	42545	21.65%
	15-19	155	17.57%	13966	7.11%
	20-24	101	11.45%	12369	6.29%
	25-34	248	28.12%	26327	13.39%
	35-44	197	22.34%	31120	15.83%
	45-54	66	7.48%	22867	11.63%
	55-59	15	1.70%	10442	5.31%
	60-64	13	1.47%	11513	5.86%
	65-74	5	0.57%	15170	7.72%
	75-84	10	1.13%	8192	4.17%
	85+	1	0.11%	2042	1.04%
Gender	**Male**	277	53.17%	99789	50.77%
	Female	244	46.83%	96764	49.23%
Race	**White**	392	76.41%	157318	80.04%
	Black/African American	53	10.33%	8023	4.08%
	American Indian/Alaska Native	11	2.14%	1192	0.61%
	Asian	6	1.17%	5326	2.71%
	Native Hawaiian/Pacific Islander	0	0.00%	113	0.06%
	Two or More Races	9	1.75%	17463	8.88%
	Other	42	8.19%	7118	3.62%

The distribution is like that of the jail dataset, where it is notably dense around the 20-44 age range. However, the 15-19 year old grouping also makes up a significant portion of the 2022 encounters window relative to their ACS estimate. Those identifying as Black/African American are also overrepresented among 2022 encounters relative to their ACS estimate, with roughly a 6% difference between the two sources. The average age between each calendar year does not change drastically, but rather, it tends to fluctuate around the 30-years old mark.

Service type is useful in determining the scope of services that an individual has received. *EncounterID*s can have this field labeled as “Mental Health” or “Substance Use Disorder.” Looking at the early portion of the available data, calendar year 2018 encounters have a split of 73% to 27% in favor of Mental Health service type. Over time this gap appears to increase, reaching a high in calendar year 2021 of 84% to 15%, and a most-recent split in calendar year 2023 of 82% to 18%.

HadTrauma is a field taking on one of four values: “yes”, “no”, “unknown”, or “missing.” Although a good portion of encounters are labeled “unknown” or “missing,” the other two options are still examined across all available calendar years. From 2018 to 2019, individuals report that they experienced trauma about 20% of the time, but there is a 6% jump starting in 2020. There is no change from 2020 to 2021, but 2022 and 2023 encounters practically revert back to 20% that was seen a couple of years ago. The COVID-19 pandemic became prominent in the U.S. in early 2020. This pandemic brought great struggle for millions of Americans, and many individuals experienced loss of family and friends. Not only that, but many were struggling to deal with the isolation that came with lockdowns, as well as any physical or mental debilitations that potentially stuck around from their personal COVID-19 contraction. All these struggles would theoretically translate to a higher rate of self-reported trauma, but encounters are not confirming that hypothesis.

3. Forecasting Approaches

When it comes to analyzing time series data, there are a variety of different models that can be used to fit, test against, and forecast such data. Exponential smoothing is a common approach, which in theory uses a decreasing trend of weighted averages of historical data points. For multivariate time series, vector autoregression can be used to determine relationships and effects of one variable on another. With current day computer and software capabilities, convolutional and recurrent neural networks can generate excellent forecasts for future data points. The primary forecasting approach used for this report is the family of ARIMA models, which include an array of different input parameters and capabilities.

3.1 An Introduction to Autoregressive Integrated Moving Average Models

Autoregressive Integrated Moving Average (ARIMA) models are a relatively simple, but effective way to model time-dependent data points. The underlying goal of ARIMA models is to determine any possible autocorrelation present in a univariate time series. For instance, there could be a strong correlation between the total sales at a local furniture store between Saturday and the Friday immediately before. That is, whenever sales are high on Fridays then they also tend to be high the following day. A standard ARIMA model can be split up into three different portions:

$$\underline{AR}_1 \quad \underline{I}_2 \quad \underline{MA}_3$$

The first component, AR, corresponds to any autoregressive components within the time series of concern (i.e. when a current data point is regressed upon a prior data point at lag p). The third component, MA, corresponds to any correlative components between a current error term and prior error term at lag q. The second component, I, corresponds to the magnitude of differencing d to which the time series exhibits traits of stationarity. ARIMA(p,d,q) models need to follow two basic assumptions: the data should have an approximate zero mean and constant variance. The combination of these two traits makeup the definition of a stationary dataset. Oftentimes, the time series appears to be trending upward or downward, and its variance can fluctuate unpredictably around the general trend of the data. To convert a non-stationary dataset into one that is stationary, one can difference the data d times (to eliminate the series trend) and/or transform the data in some non-linear manner (e.g. log, square, square root; done to create equal variance). A base ARIMA(p,d,q) model often catches the underlying autocorrelation structure, but there are additional components that can be added to make the model more robust.

Seasonal ARIMA (SARIMA) models account for the idea of seasonality in the series. These models include the same p, d, and q parameters as an ARIMA, but also include four additional ones: P, D, Q, and m. The first three are the same concept as their lowercase counterparts from the standard ARIMA, however, m is the timespan of the repetitive seasonal pattern (e.g. m=4 for monthly data possessing quarterly seasonality, m=7 for daily data possessing weekly seasonality). Another branch of ARIMA also includes the standard ARIMA parameters, but then includes additional exogenous regressors. In reference to the previous example, that same furniture store could present a strong correlation between total daily sales and the high temperature for that day. The use and implementation of these various types of ARIMA models often depend on what the time series values look like, so it is worth testing out multiple approaches to see if one generally fits better than others.

3.2 Software and Dependencies

The Python programming language is used to perform all data analysis, create visualizations, and determine forecasts for this project. Python version 3.12.2 is downloaded and used in unison with the Anaconda 2023-09 distribution for package management. Spyder IDE version 5.4.3 is downloaded and utilized to handle code implementation within multiple scripts. A full list of downloaded packages and their version numbers can be found in Appendix 1.

3.3 Sarpy County Population Growth

To go alongside eventual projections of the Sarpy County jail population and mental health related encounters from Sarpy County, a brief examination and forecast of Sarpy county's population is performed. As referenced earlier, U.S. Census Bureau datasets are downloaded to obtain two sets of numbers: decennial Census populations, as well as yearly ACS population estimates. Populations from 2010 forward are used for population growth estimates.

Three different methods are used for population growth estimates. The first method is known as arithmetic projection. This simple projection method determines the difference in population between each year and takes the average of those differences to come up with the expected absolute growth of the population. The second method is very similar, although instead of taking averages of absolute growth, averages are taken of percentage growth from year to year, which is known as geometric projection. Clearly, geometric projection is more capable of picking up non-linear growth (either positive or negative), since the denominator used to calculate percent change is time dependent. The dataset begins with the 2010 decennial Census (158,840) and ends with the 2022 ACS projection for 2022 (196,553). Figure 2 tracks the growth within Sarpy County, and distinguishes the data source of each year's population.

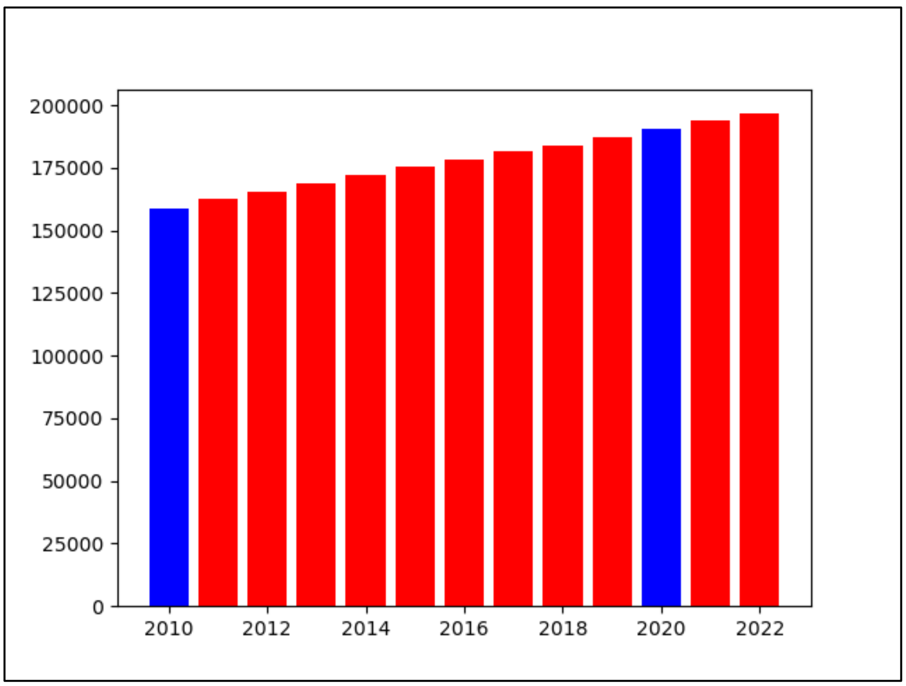


Figure 2. Sarpy County population, 2010-2022. Blue bars represent official U.S. Census Bureau populations, whereas red bars represent ACS estimates.

Growth within Sarpy County, at least in this time period, seems linear. Absolute growth rate is calculated to be increasing on average 3,143 each year. The percentage growth rate is then determined to also increase at an average rate of 1.79%. To derive out-of-sample population estimates, both average growth rates are applied to the prior year's listed/previously forecasted population. These 2023-2025 population estimates can be seen in Figure 3.

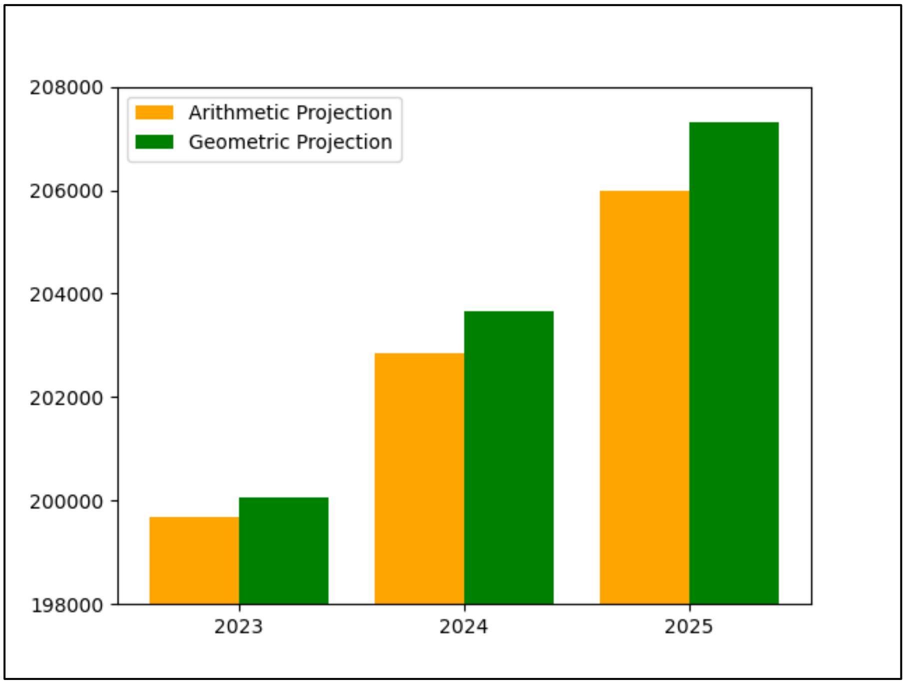


Figure 3. Sarpy County population projections for 2023-2025 with arithmetic & geometric projections.

While these methods are quite useful to pick up an overall trend of the data, they are rarely used for serious population forecasting today. Current methods often revolve around including more advanced metrics, such as birth/death rates and migration patterns. Not to mention, arithmetic and geometric projection fail to pick up on potential non-deterministic components. Like ARIMA, another time series forecasting model that is useful for considering previous values is exponential smoothing. This form of model uses weighted averages of previous data points. In its simplest form, simple exponential smoothing, α coefficients are used to assign weights to prior data points. The magnitude of these weights generally correspond to the influence of previous data points. An α that is generally small will make "older" data points more impactful on a forecast, while a generally larger α will place emphasis on more recent data points. Three types of ES models are considered for forecasting Sarpy County population through 2025. Python's *statsmodels* module is utilized for these various ES models. The first 10 data points are used for model fitting, and the last 2 are used for testing. First, a simple exponential smoothing (SES) model is modeled to the data. Default settings are used for this SES model. The calculated initial level l is 158,840, and the smoothing level α is 0.995. Forecasts for SES can be drawn from the following sets of equations:

$$p_t = l_t$$

$$l_t = \alpha y_t + (1 - \alpha) l_{t-1}$$

where p_t is the prediction/forecast at time t, l_t is the level at time t, y_t is the population at time t, and α is the smoothing level. Sum of squared errors (SSE) for this model is 102508129.097, Akaike Information Criterion (AIC) is 180.523, and Bayesian Information Criteria (BIC) is 181.319. The downside of this model is it acts in theory as an MA(1) model. If the goal is to forecast unique populations more than one year in advance, this type of model will not achieve that. It will continue to forecast the same one-step forecast, and do so by referencing the model's last level component.

The second ES model explored is a damped Holt's ES model. Damping in this case refers to the "flattening" of predicted values over time. In theory, the rate of population growth will diminish at some point in the future (e.g. drop in available housing within the county). Adding in this damping portion will ensure the population estimates continue increasing, but at a slower rate over time. Default settings are also used for this second ES model. The calculated initial level l and trend b are 156,220 and 1.02 respectively. The smoothing level α and trend β are 0.83 and 0.11 respectively. Lastly, the damping trend Φ is 0.99. Forecasts for this damped trend model can be created from the following set of equations:

$$p_{t+h} = l_t + b_t(\phi + \phi^2 + \cdots + \phi^{h-1} + \phi^h)$$

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + \phi b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + \phi\beta(1 - \beta)$$

The SSE for the damped model comes out to 1,243,661.159, the AIC is 137.992, and the BIC is 139.982. Clearly, this model gives a significant improvement in terms of how the fitted values compare to the actual data points, as evidenced by the lower SSE. The current implementation of this model has the damping trend toward the upper end of possible values (0,1). A value of 0.99 certainly forces the model to incorporate the least amount of continuous linearity for long-term forecasts, so that could be manipulated in future research.

The third and final type of ES model is a standard Holt's ES model. The only primary difference from the prior model to this one is the removal of the damping trend. This type of model will provide forecasts that are increasingly linear for all h-step forecasts. Default settings are applied to this third model. The smoothing level α and trend β are 0.83 and 0.19 respectively. The initial level l and trend b are 158,840 and 3,728 respectively. SSE drops from this model to 14,386,179.03, and AIC and BIC drop to 162.923 and 164.514, respectively. The following sets of equations represent the mathematical formulations used to create forecasts for a Holt's ES model:

$$p_{t+h} = l_t + hb_t$$

$$l_t = \alpha y_t + l_{t-1}(1 - \alpha)$$

$$b_t = \beta(l_t - l_{t-1}) + b_{t-1}(1 - \beta)$$

$$b_t = \beta(l_t - l_{t-1}) + b_{t-1}(1 - \beta)$$

This model does not appear to fit the test data as well. When referring to the list of geometric growths from year to year, it is evident that the early in-sample years are hovering around 2% growth, and that slowly makes its way down to anywhere between 1.4% and 1.8%. This behavior fits exactly the narrative that Holt's damped trend tries to capture. All three models can be seen in Figure 4. Interestingly, although the model corresponding to Holt's damped trend fits the test data points better than standard Holt's ES, one can see the opposite when comparing the two in-sample test values from 2021 and 2022.

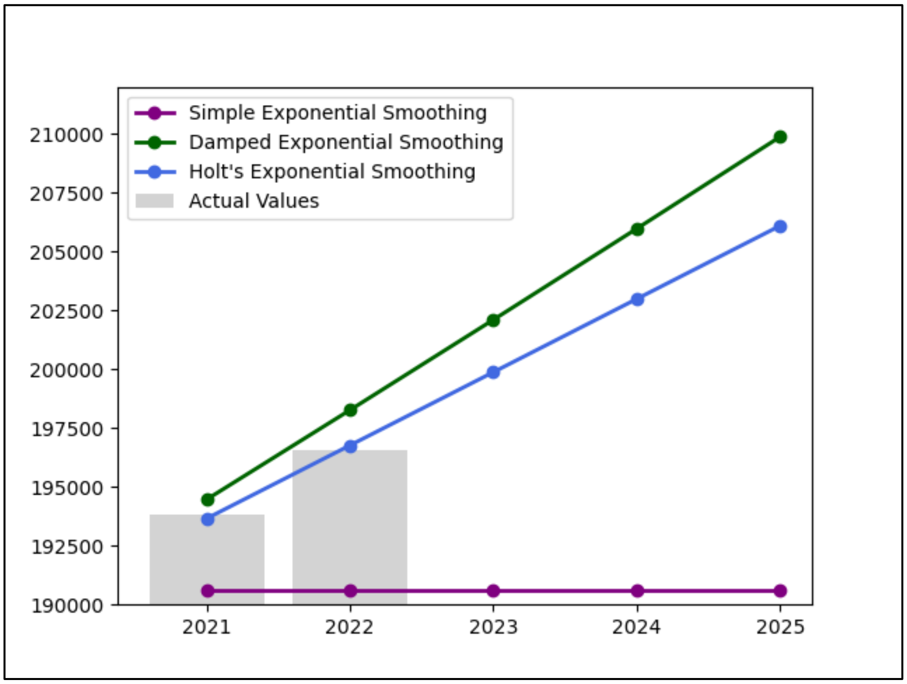


Figure 4. 2021-2022 Sarpy County Population compared to two in-sample forecasts. Three additional out-of-sample forecasts are then provided for 2023-2025.

The Holt's ES model aligns almost exactly to the 2021-2022 test values. The damped ES model does a fair job at fitting to the test data points. For future research, it would be interesting to observe the year(s) that population begins to taper off within the damped model. This is ultimately defined through the Φ coefficient, so manipulating that coefficient will determine the onset of dampening. Lastly, the simple ES model appears to be the worst of the three models created. As stated previously, simple ES would be redundant if forecasts were needed more than one year in advance, but the hope with this approach is to approximately match the first in-sample forecast (population estimate for 2021) actual value—but it ultimately fails to do so. In summary, Sarpy County's growth is without a doubt on a continuous upward trend. Based on population values from 2010-2022, arithmetic and geometric projections are useful for generating a direction and strength of trend, however, since they are calculated by equally weighting the change year-to-year, more recent trend changes (or earlier trends that have since dissipated) could be neglected. To combat this, the family of ES (sans simple ES) models apply exponentially shrinking weights in combination with a damping parameter Φ and/or smoothing trend β to favor population values from more recent years.

3.4 Sarpy County Jail Bookings

Next in line to model is the number of jail bookings for SCDC. One aggregate from the initial jail dataset is modeled, which is the number of total bookings for SCDC. The ARIMA model family is used to fit and forecast these aggregates.

3.4.1 Data Cleaning

It is important to manipulate and clean up the root dataset to prepare it for getting fed into any sort of model. First and foremost, the data grouping level needs to be determined. Models will likely differ between different frequencies of data (especially if there is any seasonal autocorrelation), but the two main focuses for determining grouping frequency are variance and the number of aggregated totals. Recall that the SCDC dataset contains bookings from July 2022 through December 2023. Daily batches over this time frame provide 548 data points with a variance of 743.39. Weekly batches provide 79 data points with a variance of 115.16. Monthly batches provide 18 data points with a variance of 1074.14. After the grouping criteria is decided, the aggregates are finalized. Model lag terms can finally be determined, as well as the coefficients tied to these lag terms.

3.4.2 ARIMA Model Fitting & Performance

Recall from earlier the assumption of stationarity for ARIMA models. Stationary datasets are said to possess zero/constant mean and constant variance. Since the idea of stationarity is needed for assuring validity in an ARIMA model, one doesn't rely solely on "looking" at the data to confirm or deny the weak stationarity assumptions. One common statistical test used to determine stationarity is the Augmented Dickey Fuller test, whose null hypothesis states a unit root is present within time series values (i.e. the time series is not stationary). The *adfuller* function within the *statsmodels* module helps perform this statistical test on this dataset, as well as the datasets to follow. The top plot in Figure 5 shows the original weekly bookings along with the test statistic and p-value from the AD Fuller test.

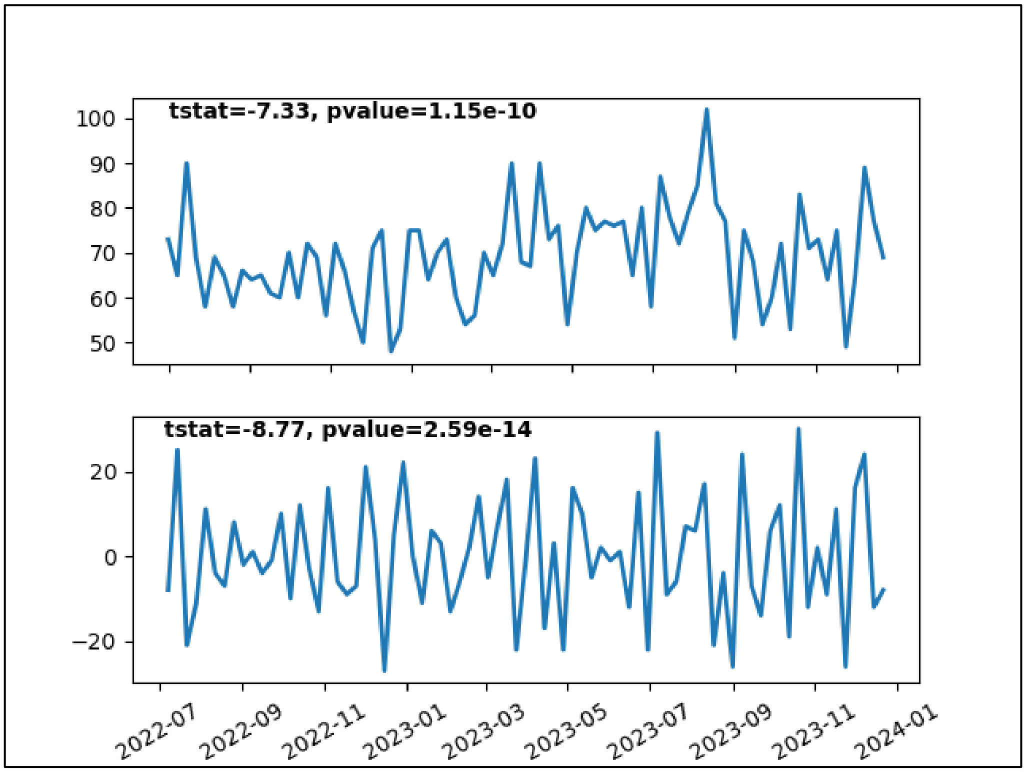


Figure 5. Number of SCDC bookings (top) and the single-differenced number of bookings (bottom).

The bottom plot shows the single-differenced number of bookings. Both AD Fuller test p-values are less than the predetermined significance level α=0.05, confirming that the null hypothesis can be rejected and both datasets are stationary. Autocorrelation and partial autocorrelation plots are then used to determine the autoregressive and/or moving average terms that could provide significant value toward accurately forecasting the number of weekly jail bookings through an ARIMA model. The *statsmodels* module contains two functions, *plot_acf* and *plot_pacf*, which are utilized for ARIMA term predictions throughout the remainder of this research. Both plots are initially applied to the standard booking aggregates, but little to no correlation is evident. When plotted against the single-differenced dataset it is easy to observe spikes in the plots. The ACF and PACF plots can be seen in Figures 6.

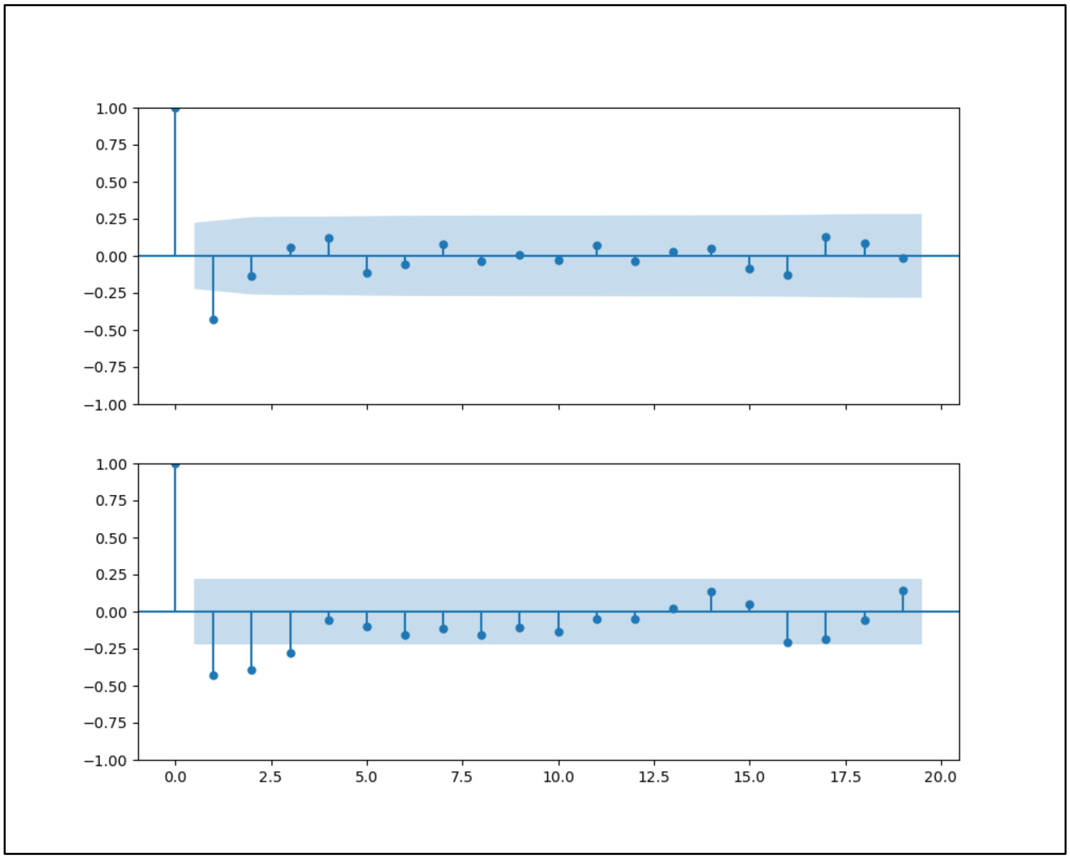


Figure 6. ACF (top) and PACF (bottom) plots for single-differenced jail booking data.

Many time series datasets do not present clear-cut lag terms to incorporate into an ARIMA model. The beauty of creating ARIMA models lies in the fact that ACF/PACF plots can be interpreted a multitude of ways. Spikes can be deemed significant to one person, but maybe not to another. At first glance, the ACF plot for this data shows a significant spike (outside the shaded blue Bartlett confidence interval region) at lag=1, and based on that, one could conclude the PACF is exponentially decreasing. This conclusion lines up with a single-differenced MA(1) model. However, it could also be argued that there is exponential decay within the ACF plot, and lags 1-3 in the PACF plot contain significant spikes—which would correspond to a single-differenced ARIMA(3,1,0) model.

When in doubt, it is best to fit all models that have potential to have strong predictive power. The two formerly mentioned models are fit to the single-differenced data. These two model summaries can be seen in Table 7, and the mathematical notation with calculated parameters are provided directly thereafter.

Table 7. SCDC ARIMA model results.

Model	AIC	BIC	MSE	MAE	MAPE	MRSE
ARIMA(0,1,1)	589.551	594.239	184.517	9.307	13.901%	13.584
ARIMA(3,1,0)	597.556	606.931	191.013	9.482	14.083%	13.821

$$ARIMA(0,1,1)\text{:}\ \ \nabla Y_t = e_t - \theta e_{t-1} \rightarrow Y_t = Y_{t-1} + e_t - \theta e_{t-1} \rightarrow Y_t = Y_{t-1} + e_t - 0.88e_{t-1}$$

$$ARIMA(3,1,0)\text{:}\ \ \nabla Y_t = \phi_1 \nabla Y_{t-1} + \phi_2 \nabla Y_{t-2} + \phi_3 \nabla Y_{t-3} + e_t \rightarrow Y_t = (1 + \phi_1)Y_{t-1} + (\phi_2 - \phi_1)Y_{t-2} + (\phi_3 - \phi_2)Y_{t-3} - \phi_3 Y_{t-4} + e_t \rightarrow Y_t = 0.29Y_{t-1} + 0.14Y_{t-2} + 0.28Y_{t-3} + 0.28Y_{t-4} + e_t$$

Y_t corresponds to the time series aggregate fitted value (Y_{t-h} corresponds to the time series aggregate fitted value at lag *h*), and e_t corresponds to an I.I.D. random variable with zero-mean. Initial observations of these models indicate they are an overall good fit. All ARIMA models for this research are summarized through six model performance estimators to determine goodness of fit: AIC, BIC, mean squared error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and mean root squared error (MRSE). MAPE and MAE will be referenced the most due to their ability of being easily understood by audiences of various technical backgrounds.

Residual analysis plays a large part in confirming an ARIMA model is correctly fitting to a time series. This step also ensures a model is not failing to pick up lag terms that were not initially included. Diagnostic methods used in this report are drawn from Cryer & Chan's (2011) *Time Series Analysis With Applications in R*. Various diagnostic plots can be seen in Figures 7 and 8. Diagnostic plots for the remaining jail booking models can be found in Appendices 3 and 7.

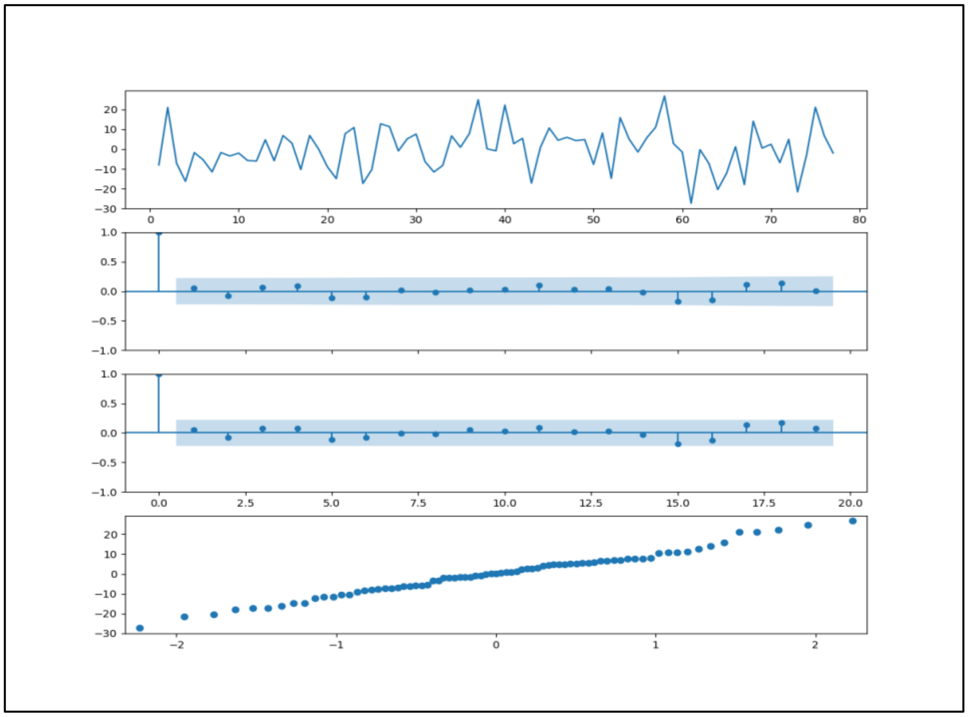


Figure 7. ARIMA(0,1,1) diagnostic plots for the SCDC booking dataset.

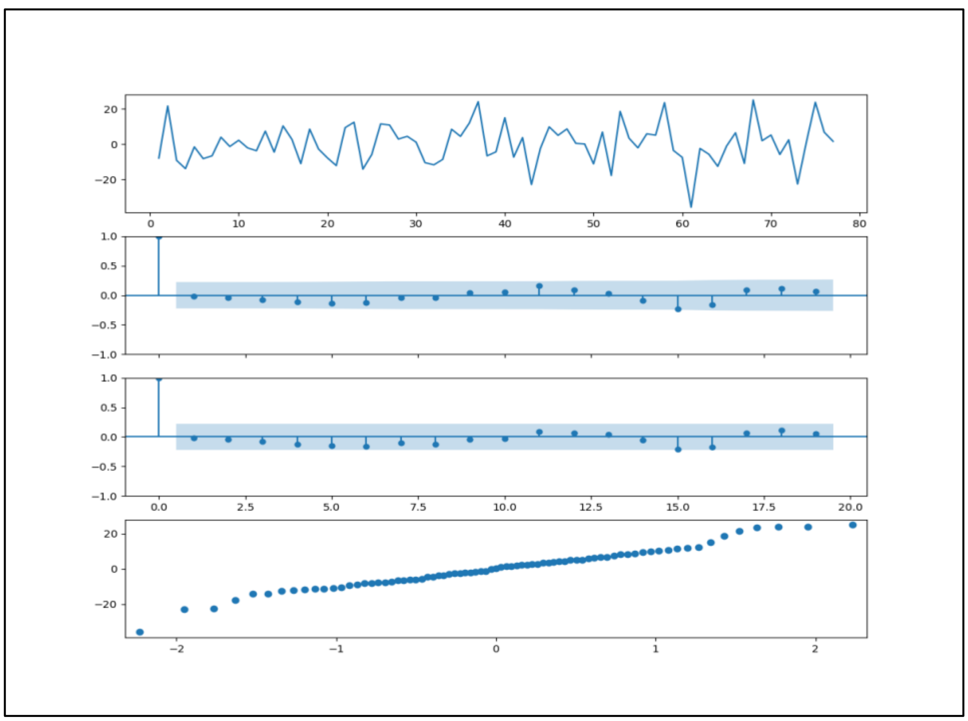


Figure 8. ARIMA(3,1,0) diagnostic plots for the SCDC booking dataset.

To confirm the goodness of fit for the ARIMA models, four main plots are observed. The model errors (residuals) are observed over time to ensure the model is not consistently over- or underestimating portions of the modeling window (top plots in Figures 7 & 8). Next, the ACF and PACF plots are redrawn to confirm that there is no autocorrelation still present within the residuals (middle plots in Figures 7 & 8). If large spikes are still present outside the shaded blue region, the model would likely need to be reevaluated. Lastly, the bottom plots in Figures 7 and 8 show quantile-quantile plots. If residuals are drawn from an approximately normal distribution, the data points in this plot would follow a diagonal line from the bottom-left to the top-right corner of the plot. Obviously, residuals will never be perfectly normally distributed, so an approximate diagonal line is sufficient and expected.

Both models pass diagnostics based on these visual observations. Two additional statistical tests are also utilized: Shapiro's normality test and the Ljung-Box test for serial autocorrelation. Both will confirm the same ideas that are being tested through plotting, so they are included for added diagnostic checking strength. The ARIMA(0,1,1) model's Shapiro test statistic and p-value are 0.989 and 0.761 respectively (H_0: residuals are drawn from normal distribution), and the Ljung-Box test passes for lags 1-10 (p-values > 0.05; H_0: residuals exhibit no autocorrelation for fixed number of lags L). Similarly, the ARIMA(3,1,0) model's Shapiro test returns a test statistic and p-value of 0.980 and 0.279 respectively, and the Ljung-Box test also fails to confirm serial autocorrelation.

Two additional models are created for this set of jail booking data. However, they include only a seasonal component. Initial thoughts are that weekly batches would express a correlation with that same weekly batch from the month prior. This translates to a SARIMA model with a frequency of 4 (~4 weeks within a month). A SARIMA$(0,0,0)(1,1,0)_4$ and a SARIMA$(0,0,0)(0,1,1)_4$ are created, and their sets of performance metrics can be seen in Table 8.

Table 8. SCDC SARIMA model results.

Model	AIC	BIC	MSE	MAE	MAPE	RMSE
SARIMA$(0,0,0)(1,1,0)_4$	589.193	593.801	439.068	13.113	19.009%	20.954
SARIMA$(0,0,0)(0,1,1)_4$	576.817	581.425	414.286	12.568	18.118%	20.354

$$SARIMA(0,0,0)(1,1,0)_4\text{:}\ \ \nabla_S Y_t = \Theta \nabla_S Y_{t-4} + e_t \rightarrow Y_t = (1-\Theta)Y_{t-4} - \Theta Y_{t-8} + e_t \rightarrow Y_t = 1.42Y_{t-4} - 0.42Y_{t-8} + e_t$$

$$SARIMA(0,0,0)(0,1,1)_4\text{:}\ \ \nabla_S Y_t = e_t - \Phi e_{t-4} \rightarrow Y_t = Y_{t-4} + e_t - \Phi e_{t-4} \rightarrow Y_t = Y_{t-4} + e_t - 0.85e_{t-4}$$

Diagnostic plots look as expected. The Shapiro test statistic and p-value for the autoregressive single-differenced model are 0.989 and 0.769 respectively, and the Ljung-Box p-values are all greater than 0.05. For the moving average single-differenced model, the Shapiro test statistic and p-value are 0.986 and 0.581 respectively, and the Ljung-Box p-values also confirm no autocorrelation among residuals. Based on these checks, the two seasonal ARIMA models also pass the diagnostic tests and appear to be suitable models.

All four ARIMA models look to be good models for the booking dataset, but there is one potential concern to discuss. These models fit all weekly data values from data point 1 to data point 79. However, the successful ARIMA models are fit to those datapoints, and those datapoints only. One common approach to ensure ARIMA models are consistently accurate for the current set of available time series values is to perform a type of cross-validation (CV). In a standard CV application with machine learning models, a random sample of the independent variable matrix can be assigned to a training dataset, and the non-selected row vectors are assigned to a test dataset. This approach cannot be applied to time-dependent data points, since there would be no assurance that the *m* training dataset values are consecutive in nature (similarly for the *n* test dataset values).

One way to perform CV with time series data is to calculate forecast horizon accuracy metrics across multiple subsets of the time series data starting with time t=1 and continuing to time t=q, where q cannot exceed the number of data points minus the forecast horizon length. Because the first two ARIMA models supply better all-around performance metrics, those two have CV run against them to test their general performance. Both models are separated into five splits, or folds. The ARIMA(0,1,1) only forecasts a single value to test against (due to having only a single MA term), whereas the forecast window is not defined for the ARIMA(3,1,0) (default *n_splits* value of 5). Figure 9 provides an example of cross-validating a time series model with eight available data points. Figures 10 and 11 show the Python implementation and results of cross-validating the ARIMA(0,1,1) model, while Figures 12 and 13 show Python implementation and results of cross-validating the ARIMA(3,1,0) model.

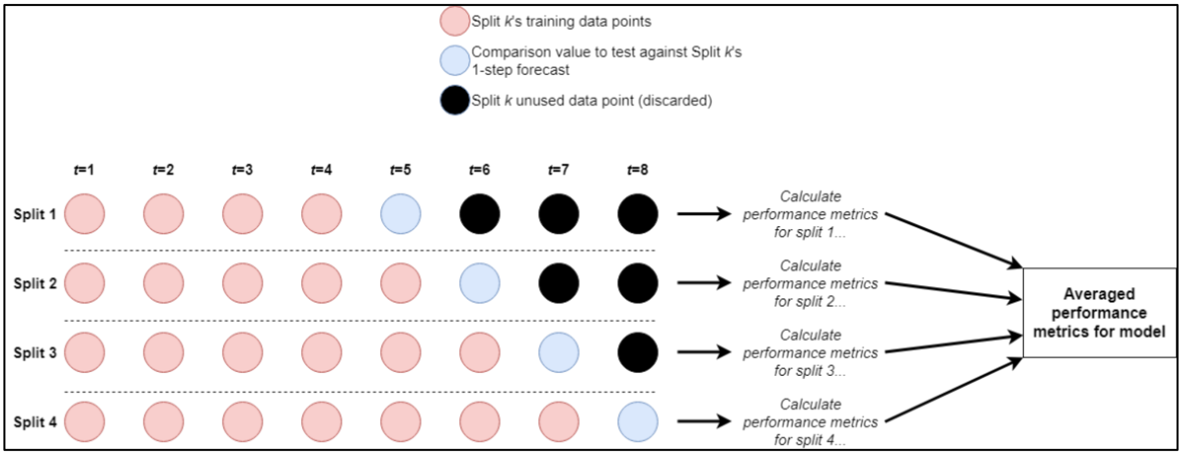


Figure 9. Example time series cross-validation. 4-fold over 8 data points that each generate a 1-step forecast for testing against actual value.

```python
arima011_mae_scores, arima011_mse_scores, arima011_mape_scores, arima011_rmse_scores = [], [], [], []
cv = TimeSeriesSplit(test_size=1)   # test size of 1 due to MA(1) component
for train, test in cv.split(jailData):
    trainData, testData = jailData['NumBookings'].iloc[train], jailData['NumBookings'].iloc[test]
    mod = sm.tsa.ARIMA(trainData, order = (0,1,1)).fit()
    forecast = mod.forecast(1)
    mse, mae, mape, rmse = mean_squared_error(testData, forecast), mean_absolute_error(testData, forecast), mean_absolute_percentage_error(testData, forecast), np.sqrt(mean_squared_error(testData, forecast))
    arima011_mse_scores.append(mse)
    arima011_mae_scores.append(mae)
    arima011_mape_scores.append(mape)
    arima011_rmse_scores.append(rmse)

#print results
print(f"Average MAE: {np.mean(arima011_mae_scores)}")
print(f"Average MSE: {np.mean(arima011_mse_scores)}")
print(f"Average MAPE: {np.mean(arima011_mape_scores)}")
print(f"Average RMSE: {np.mean(arima011_rmse_scores)}")
```

Figure 10. Python code for cross-validating ARIMA(0,1,1) model.

```
Average MAE: 10.924562139732405
Average MSE: 200.21550840328823
Average MAPE: 0.16770499756798746
Average RMSE: 10.924562139732405
```

Figure 11. Model performance metrics of cross-validated ARIMA(0,1,1) model.

```python
arima310_mae_scores, arima310_mse_scores, arima310_mape_scores, arima310_rmse_scores = [], [], [], []
cv = TimeSeriesSplit()   # don't need test size limit
for train, test in cv.split(jailData):
    trainData, testData = jailData['NumBookings'].iloc[train], jailData['NumBookings'].iloc[test]
    mod = sm.tsa.ARIMA(trainData, order = (3,1,0)).fit()
    forecast = mod.forecast(steps=len(testData))
    mse, mae, mape, rmse = mean_squared_error(testData, forecast), mean_absolute_error(testData, forecast), mean_absolute_percentage_error(testData, forecast), np.sqrt(mean_squared_error(testData, forecast))
    arima310_mse_scores.append(mse)
    arima310_mae_scores.append(mae)
    arima310_mape_scores.append(mape)
    arima310_rmse_scores.append(rmse)

#print results
print(f"Average MAE: {np.mean(arima310_mae_scores)}")
print(f"Average MSE: {np.mean(arima310_mse_scores)}")
print(f"Average MAPE: {np.mean(arima310_mape_scores)}")
print(f"Average RMSE: {np.mean(arima310_rmse_scores)}")
```

Figure 12. Python code for cross-validating ARIMA(3,1,0) model.

```
Average MAE: 9.2112687198475
Average MSE: 129.97447495636717
Average MAPE: 0.13502434543964265
Average RMSE: 11.158779011663672
```

Figure 13. Model performance metrics of cross-validated ARIMA(3,1,0) model.

Although the ARIMA(0,1,1) produces better performance metrics when fitted to the entire available dataset (July 2022 – December 2023), the ARIMA(3,1,0) metrics surpass those of the ARIMA(0,1,1) when solely taking the CV analysis into account. If Region 6 wishes to use the ARIMA(0,1,1) as it currently stands for deriving forecasts for weekly SCDC jail bookings, they would likely get a better estimate than using the ARIMA(3,1,0). However, since booking data can theoretically be requested from SCDC on demand, the ARIMA(3,1,0) model would be the better approach since that model tends to test well across many forecast horizons of the data.

To wrap up work with this jail dataset, forecasts are created for each of the four models. As expected, the ARIMA(0,1,1) offers one legitimate forecast for the first out-of-sample week. The ARIMA(3,1,0) fluctuates slightly for approximately two months and flattens out to around 75 monthly bookings beginning April 2024. The two seasonal models fluctuate within the 4-week period as apparent through their implementation, although the ARIMA(0,0,0)(1,1,0)$_4$ has a greater variance due to more influential Φ coefficients. Figure 14 shows these forecasts through the end of 2025. Individualized forecasts along with their 95% prediction intervals can be referenced in Appendices 4 and 8.

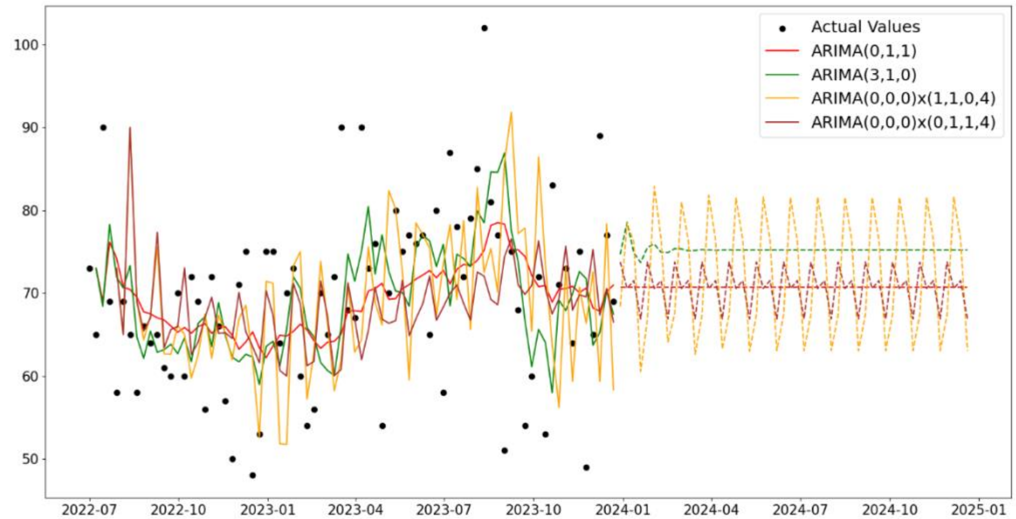


Figure 14. SCDC model forecasts: 2024 – 2025. Seasonal ARIMA models are easily recognizable through their wave-like forecasts, whereas standard ARIMA models tend to "flatten out" over a long period of time.

3.5 Mental Health Encounters

The weekly jail bookings prove to be a generally straight-forward dataset to fit models to. Once the values are differenced a single time, there are only a handful of ARIMA-based models that are be accurate and pass diagnostics. The mental health encounters dataset also primarily utilizes ARIMA-based models, although this dataset exhibits a definite mean shift unlike the previously observed datasets. Medicaid expansion rolled out in Nebraska in October 2020, causing Region 6's providers to experience a rapid decline in treatment from consumers. Those under the age of 65 who possess an income up to 138% of the poverty level were then eligible to be covered under Medicaid (Norris, 2024). A large portion of Nebraskans decided to take advantage of this coverage, which explains the drop in consumer/encounter data made available to Region 6. This apparent intervention limits the number of plausible standard ARIMA models, so various model types are explored in upcoming sections.

3.5.1 Data Cleaning

The brief data structure of the mental health encounters data is introduced in section 2 of this paper. To recap, there are five available datasets—each one corresponding to a fiscal year between 2019 and 2023. The idea of deduping is also touched on, since encounters could technically extend across multiple fiscal years until the encounter was completed (i.e. had a closing *discharge_date*). Similar data manipulation is utilized for the modeling portion of this dataset as the early exploratory analysis portion. All datasets are first appended into a master data frame. The master data frame is then pivoted across the *EncounterID* index to display *date_admission* values across all fiscal years that the encounter is listed under. This allows for two custom deduping functions to go through and pick out each unique *EncounterID* and the sole *admission_date* that it corresponds to. Records are then filtered to only include those who have a *County_residence* of "Sarpy", as well as an *admission_date* prior to or on July 1st, 2018.

Unlike the jail bookings, the grouping level is heavily favored based on the length of the forecast horizon. These mental health encounter model(s) are intended to forecast through the end of 2025. Grouping on "small" levels will inevitably reduce the model prediction power (especially in MA models where forecasts are residual-based). With that being said, the same groupings are explored as earlier. When grouped on the daily level, there are 1826 aggregates with a variance of 7.462. When grouped on the weekly level, there are 262 aggregates with a variance of 79.105. Lastly, monthly aggregates provide 60 values with a variance of 972.710. Although the monthly aggregates have a much higher variance, they still provide enough data points for model fitting. To get predicted values through the end of 2025, one would need 540 daily-level forecasts, 72 weekly-level forecasts, or 18 monthly-level forecasts. Monthly batches are decided for forecasting purposes, although significant autocorrelation structure could have been evident within daily and weekly batches.

3.5.2 ARIMA Model Fitting & Performance

Once the aggregate level is finalized, the final cleaned dataset is grouped to that level. Figure 15 shows the original number of encounters, single-differenced number of encounters, and twice-differenced number of encounters, along with their AD Fuller test statistic/p-value to determine stationarity. It is clear that the original values have a downward trend which violates the idea of a zero/constant mean, and is also evident through the p-value associated with the AD Fuller test. Differencing the data seemingly eliminates the trend component, so both single- and twice-differenced data are the primary sets of values used to create models for monthly encounters.

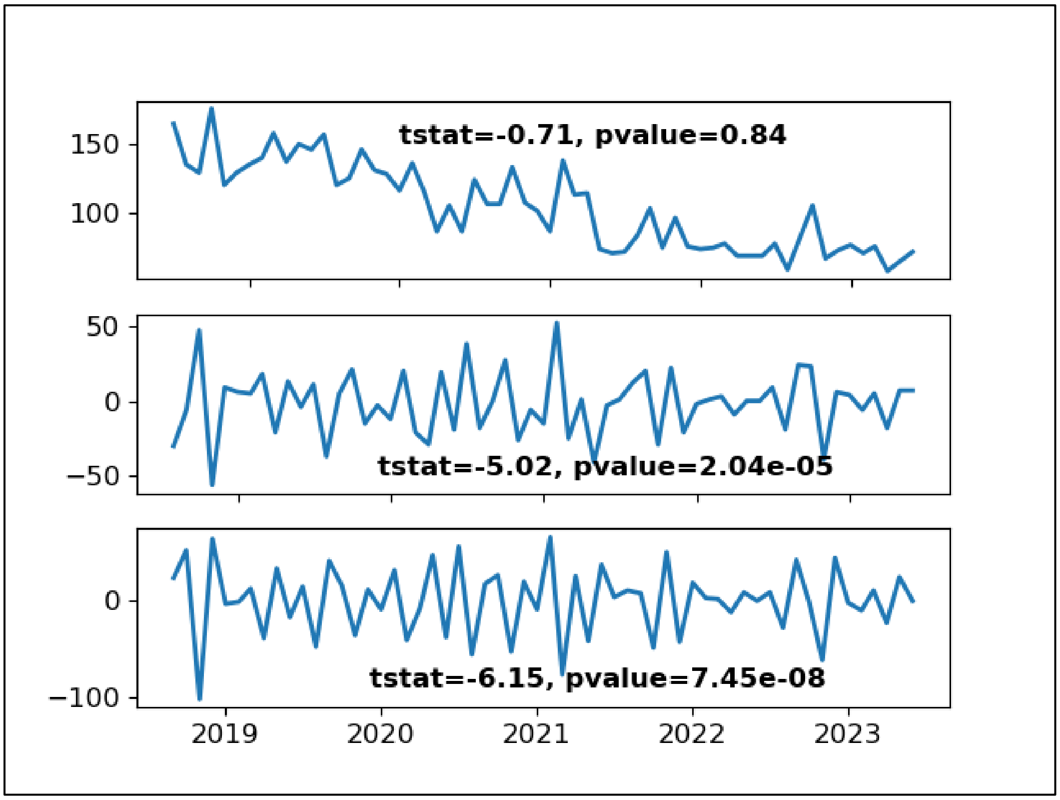


Figure 15. Original (top), single-differenced (middle), and twice-differenced (bottom) monthly mental health encounters.

The first set of models explored are standard and seasonal ARIMA models. Since single- and double-differenced data are stationary via AD Fuller tests, ACF and PACF plots are created from those sets of values to determine potential autoregressive and/or residual autocorrelation. For differenced data the ACF plot appears to have a significant spike at lag=1, but a spike is also just about at the confidence band lower bound at lag=6. One could also argue that the plot presents exponential decay in autocorrelation from lag=6 onward. When viewing the PACF plot for single-differenced data, lags 1 and 2 appear to be significant, lags 3 and 4 are just below the confidence band lower bound, and lag=6 is once again almost exactly at the confidence band lower bound.

Since there are earlier lags containing autocorrelation and a probable gap until the next significant one, there is likely some seasonality present in the single-differenced data.

The two successful seasonal ARIMA models used to fit this dataset are a SARIMA(0,1,1)(1,0,0)$_6$ and a SARIMA(0,1,1)(1,0,1)$_6$. Both models and their mathematical notation can be seen in Table 9, and their corresponding diagnostic plots are in Appendices 9, 11, and 13. Additionally, second-differenced data and its ACF/PACF plots are explored for potential autocorrelation. The ACF plot for this dataset shows much more prominence for lags 1 through 4. The PACF plot shows a very strong correlation with lag=1, and similar to the single-differenced data, a spike at the confidence band lower bound at lag=6. This lag=6 "spike" is ultimately disregarded, since various subsequent lags are fluttering around the upper and lower bound of the confidence band. Based on these observations, a single standard ARIMA is fit to the twice-differenced data, which is an ARIMA(4,2,0). Table 9 lists all successful standard and seasonal ARIMA models fit to the single- or twice-differenced monthly encounters dataset, followed by each model's mathematical representation.

Table 9. Standard & seasonal ARIMA models for mental health encounters.

Model	AIC	BIC	MSE	MAE	MAPE	RMSE
SARIMA(0,1,1)(1,0,0)$_6$	506.204	515.437	749.655	16.103	15.723%	27.380
SARIMA(0,1,1)(1,0,1)$_6$	508.326	516.636	736.663	15.501	15.126%	27.142
ARIMA(4,2,0)	523.529	533.831	1096.61	19.882	18.595%	33.115

$$SARIMA(0,1,1)(1,0,0)_6\text{:}\ \nabla Y_t = e_t + \Theta \nabla Y_{t-6} - \phi e_{t-1} \rightarrow Y_t = Y_{t-1} + \Theta Y_{t-6} - \Theta Y_{t-7} - \phi e_{t-1} + e_t \rightarrow Y_t = Y_{t-1} - 0.19Y_{t-6} + 0.19Y_{t-7} + 0.65e_{t-1} + e_t$$

$$SARIMA(0,1,1)(1,0,1)_6\text{:}\ \nabla Y_t = e_t + \Theta \nabla Y_{t-6} - \phi e_{t-1} - \Phi e_{t-6} \rightarrow Y_t = Y_{t-1} - \phi e_{t-1} + \Theta Y_{t-6} - \Theta Y_{t-7} - \Phi e_{t-6} + e_t \rightarrow Y_t = Y_{t-1} + 0.65e_{t-1} - 0.93Y_{t-6} + 0.93Y_{t-7} - 0.80e_{t-6} + e_t$$

$$ARIMA(4,2,0)\text{:}\ \nabla^2 Y_t = \theta_1 \nabla^2 Y_{t-1} + \theta_2 \nabla^2 Y_{t-2} + \theta_3 \nabla^2 Y_{t-3} + \theta_4 \nabla^2 Y_{t-4} + e_t \rightarrow Y_t = (2+\theta_1)Y_{t-1} + (\theta_2 - 2\theta_1 - 1)Y_{t-2} + (\theta_1 - 2\theta_2 + \theta_3)Y_{t-3} + (\theta_2 - 2\theta_3 + \theta_4)Y_{t-4} + (\theta_3 - 2\theta_4)Y_{t-5} + \theta_4 Y_{t-6} + e_t \rightarrow Y_t = 0.71Y_{t-1} + 0.41Y_{t-2} + 0.17Y_{t-3} + 0.10Y_{t-4} + 0.10Y_{t-5} - 0.49Y_{t-6} + e_t$$

Diagnostics pass for all three models. The Shapiro and Ljung-Box tests confirm residual normality and non-serial autocorrelation. Shapiro test statistics/p-values are 0.984/0.649, 0.989/0.885, and 0.978/0.362 for the models given in Table 9, and all three models have p-values greater than 0.05 for lags 1-10. Immediately, one can see the ARIMA(4,2,0) model does not fit the data as strong as the other two seasonal ARIMA models.

The two seasonal ARIMA models are off by roughly 15-16 encounters between the actual and fitted values, or about 15% error between the actual and fitted values. These residual model errors are acceptable, especially since they pass the diagnostic checklist. Despite being technically sound, the number of monthly encounters can likely be further refined to the point where residuals are exhibiting a MAE and MAPE of roughly 10 monthly encounters and 10% respectively.

ARIMAX models are also situated within the family of ARIMA models, however, they add additional exogenous regressor(s) into the fitting and prediction of the time series variable. These exogenous variables do not have to necessarily be "derived" from a regressor at lag=0, but that is the only option explored in this research.

3.5.3 ARIMAX Model Fitting & Performance

To begin the process of exploring exogenous variables, the first inclination is to implement a set of hard-coded regressors. These fields and their values correspond to three separate aspects of the original monthly encounters values. All of them have to do with the notable downward shift resulting from the October 2020 Medicaid expansion:

1) A binary field indicating points before (0) and on/after (1) April 1st, 2021, which is the decided intervention date that initiates a change in the monthly encounters process (refer to the top plot in Figure 15).
2) An increasing count of integers (1,2,…,*n*), where each integer corresponds to a particular month's data point (*n* is the number of months within the dataset).
3) An increasing count of integers (1,2,…*n-p*), where each integer corresponds to a month's data point, but *only* for those months on/after April 2021 (*n* is the total number of months within the dataset and *p* is the number of months from the first month to the month corresponding to April 2021)

All three of these are assigned to their own field in a Python data frame in order to be included within ordinary least-squares regression on the monthly encounters values. 1) is added to determine any potential change in mean between the time periods July 2018 – March 2021 and April 2021 – June 2023. 2) is added to determine trend strength across the *entire* July 2018 – June 2023 dataset, and 3) is added to determine trend *only* within April 2021 – June 2023. Figure 16 provides the Python code for creating and implementing these three additional regressor variables into an ordinary least-squares regression model.

```
monthlyAdmissions['medicaid_rollout'] = [0 if x < pd.to_datetime('2021-04-01') else 1 for x in monthlyAdmissions['MonthStart']]
monthlyAdmissions['data_point'] = list(range(1,len(monthlyAdmissions)+1))
monthlyAdmissions['medicaid_rollout_posttrend'] = [0 if a==0 else b-33 for a,b in zip(monthlyAdmissions['medicaid_rollout'],monthlyAdmissions['data_point'])]

ols = sm.OLS(monthlyAdmissions['NumAdmissions'], monthlyAdmissions[['medicaid_rollout','data_point','medicaid_rollout_posttrend']]).fit()
```

Figure 16. Ordinary least-squares regression model fitted with the three newly-created fields from the monthly encounters time series dataset (*medicaid_rollout, data_point,* and *medicaid_rollout_posttrend*).

All three prove to have sizeable regressive properties in relation to the number of monthly encounters, with coefficients/p-values of -89.52/0.003, 5.33/0.000, and -6.03/0.000. Once this is determined, one can observe potential autocorrelation among the OLS residuals. For this particular OLS regression, autocorrelation is present among residuals via ACF/PACF plot observations. This indicates that an ARIMAX model would be a desirable model choice.

After creating various ARIMAX models, only then is it determined that a single one of these OLS exogenous regressors, *medicaid_rollout,* should be included into an ARIMAX model. Incorporating all three regressors reveals a strong results in a **strict** OLS regression model (no Y_{t-h} variables are included), however, such an OLS model is not explored further. Table 10 shows the three ARIMAX models and their performance metrics, followed by their mathematical notation. Prior to creating ARIMAX models, residuals from the initial OLS procedure are also checked to confirm stationarity.

Table 10. ARIMAX models for mental health encounters.

Model	AIC	BIC	MSE	MAE	MAPE	RMSE
ARIMAX(2,0,0)	521.295	531.766	296.626	13.711	13.465%	17.223
ARIMAX([2],0,0)*	521.659	530.036	307.463	13.672	13.321%	17.535
ARIMAX(1,0,0)	524.050	532.428	319.668	13.750	13.445%	17.879

*Only one AR term (lag=2) included in model.

$$ARIMAX(2,0,0)\text{:}\ \ Y_t = \beta_0 + \beta_1 \cdot medicaidrollout_i + \theta_1 Y_{t-1} + \theta_2 Y_{t-2} + e_t \rightarrow Y_t = 126.22 - 48.21 \cdot medicaidrollout_i + 0.20Y_{t-1} + 0.28Y_{t-2} + e_t$$

$$ARIMAX([2],0,0)\text{:}\ \ Y_t = \beta_0 + \beta_1 \cdot medicaidrollout_i + \theta_2 Y_{t-2} + e_t \rightarrow Y_t = 126.72 - 49.80 \cdot medicaidrollout_i + 0.33Y_{t-2} + e_t$$

$$ARIMAX(1,0,0)\text{:}\ \ Y_t = \beta_0 + \beta_1 \cdot medicaidrollout_i + \theta_1 Y_{t-1} + e_t \rightarrow Y_t = 126.48 - 49.34 \cdot medicaidrollout_i + 0.27Y_{t-1} + e_t$$

Practically all metrics improve through the addition of the *medicaid_rollout* regression variable. There is not a large difference between the three models created (as one can see through the changes in the β_i and θ_i coefficients). MAE drops by roughly 2 monthly encounters, and MAPE decreases by about 2%.

Once exogenous regressors are explored, there are not many other ARIMA-based approaches to traverse. Other regressor columns could be inspected and would likely be useful in terms of predictive power for the number of Sarpy County resident monthly encounters. Various fields from the base dataset could be aggregated in an equivalent manner, such as counts within *Insurance* (would aggregates for "Medicaid", "Medicare", and/or "No Insurance" have any relation to the total monthly encounters, perhaps even at a lag≠0?). Nevertheless, Figure 17 provides forecasts through the end of 2025 for mental health encounters within the Sarpy County consumer population, utilizing both the standard/seasonal ARIMA and ARIMA with exogenous regressors (ARIMAX) models. The ARIMA(4,2,0) forecasts values with a decreasing trend, whereas the two seasonal ARIMAs remain stagnant for the long-term forecast window. This is likely the correct representation as Region 6 does not expect the trend to continue decreasing, but rather persist along a new baseline. As expected, the ARIMAX models account for the sudden drop in encounters starting with April 1st, 2021, encounters, hence the reason for the immediate decrease about halfway through the actual data points in the plot.

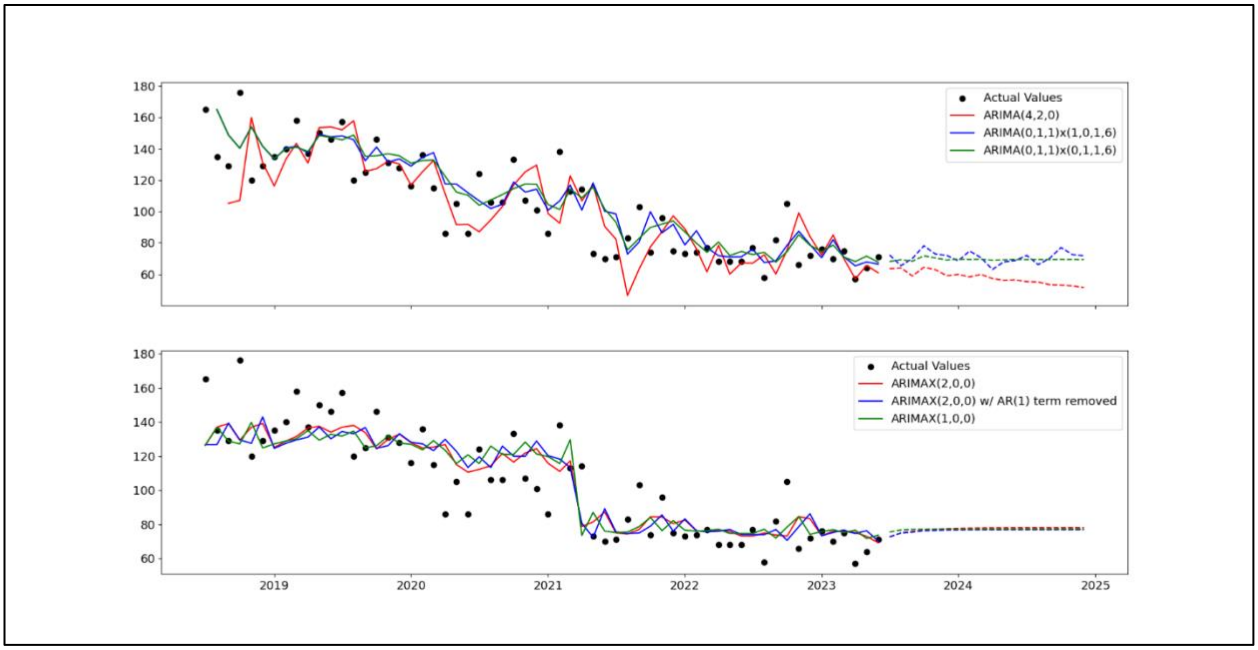


Figure 17. Forecasts for Sarpy County residents monthly mental health encounters through 2025. Standard/seasonal ARIMA models (top) and ARIMAX models (bottom) take on considerably different forms.

Future ARIMAX models could attempt to apply gradual intervention-related effects, as opposed to instantaneous effects from a single month. Each ARIMAX model's diagnostic plots and forecasts (along with 95% prediction intervals) can be found in Appendices 15 through 20.

3.5.4 A Prophet Approach

ARIMA models are some of the more reliable time series models. Many natural and business processes likely possess an underlying autocorrelation structure, which ARIMA models are solely comprised of. However, ARIMA models often have some disadvantages when it comes to modeling. One disadvantage is the ARIMA model terms are strictly linear (e.g. Y_{t-1}, Y_{t-2}, e_{t-6}). That is, model representations can only be expressed as constants multiplied by lagged errors or values of the time series. Another obvious disadvantage is assumptions need to be met for ARIMA models—or at least accurate ones. Recall each fitted ARIMA model discussed prior to this point. It was essential to confirm that the datasets had zero/constant mean and homoscedasticity. To achieve this for many "real-world" datasets, data transformation becomes a large part of the modelling process. Whether it be differencing one/many time(s), or carrying out a Box-Cox transformation, this pre-model creation step can eat up a lot of time for a data analysts and scientists (on top of all that, this time-consuming work will only prove worthy if a "successful" ARIMA model is found).

These disadvantages lead to another type of model being created strictly for the mental health encounters dataset. The theory and applications of *Facebook*'s Prophet was first proposed in a 2017 paper "Forecasting at scale" written by Sean Taylor and Bejamin Letham, two members of *Facebook*'s core data science team. In this paper, Taylor and Letham note similar disadvantages for ARIMA, ES, and other common time series models. Instead of deriving forecasts from weighted averages or correlations among lagged values, Taylor and Letham describe the generalized additive model (GAM) structure that Prophet draws from. GAMs can account for both linear and non-linear relationships between any regressor and dependent variable, proving their robustness for datasets with complex properties. The Prophet model can be simplified to the following equation:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

In this formulation, *g(t)* represents the general/long-term trend of the time series, *s(t)* represents "periodic changes" (e.g. seasonality within a week or year), and *h(t)* represents the potential effects that holidays play in the forecasted outcome variable *y(t)* (Taylor and Letham, 2017; ε_t represents normally distributed white noise). Two of the three components immediately fit the structure of the mental health encounters dataset. Based on the top plot in Figure 15, there is a subtle downward trend across monthly encounters from July 2018 to June 2023, which can be modeled accordingly through *g(t)*. Similarly, it is evident there is seasonality (at varying lags/periods) present from the ARIMA models. This seasonality aspect can also be incorporated through the aforementioned seasonality equation, *s(t)*. Given that holidays typically occur on a specific day, and the aggregates are grouped by month, it would not make sense to include *h(t)* within this approach.

Another important feature of Prophet used for this modeling approach is saturation and logistic decay. In many real-world datasets, there are assumed floors and/or ceilings that values can take on. In the case of these mental health encounters, monthly counts appear to taper off around 50 to 60 encounters per month. Based on trends from October 2020 onwards, Region 6 envisions that monthly encounters will not dip below 50. Given their obvious domain knowledge, a "floor" of 50 is implemented into a Prophet model. This saturation method is made possible through a piecewise logistic growth/decay function within the model's *g(t)* component. The added floor/ceiling doesn't necessarily have to be constant across time due to the idea of the piecewise linear function. This function allows for a time-dependent floor or ceiling, a constant floor or ceiling for some period of time (which is used for this research), or some combination of the two.

Prophet is strict on how it accepts input parameters. For the Python programming language, one must have a data frame containing the following two fields: *y* and *ds*. These represent the time series values and datetimes, respectively. To include saturation (and therefore, a floor/ceiling) within the process, two additional fields need to be within the data frame: *cap* and *floor*. *Floor* corresponds to the 50 encounters that Region 6 identifies as the lowest they could see these values reaching. *Cap* must also be included so that the piecewise logistic growth/decay function is bounded by both a minimum and maximum. A *cap* of 200 is defined in this instance, since that is slightly greater than the observed maximum of monthly encounters prior to the enaction of Medicaid expansion.

```
#Define ceiling/floor for logistic growth/decay
cap = 200
floor = 50

#Rename columns to Prophet's expectations.
admissionsProphet = monthlyAdmissions[['MonthStart','NumAdmissions']]
admissionsProphet.columns = ['ds','y']
admissionsProphet['floor'] = floor
admissionsProphet['cap'] = cap

#Create model instance with 95% CIs and logistic growth, then fit.
pModel3 = Prophet(growth='logistic', interval_width=0.95)
pModel3.fit(admissionsProphet)

#Create new dataframe with forecasts through 2025
future3 = pModel3.make_future_dataframe(periods=18, freq='M')
future3['floor'] = floor
future3['cap'] = cap
forecast3 = pModel3.predict(future3)
```

Figure 18. Initial Prophet model fitting and forecasting. Along with having to adjust column names, *cap* and *floor* must also be provided to fit a logistic growth/decay Prophet model.

Figure 18 shows the Python code used to create the Prophet model for monthly Sarpy County mental health encounters. Recall that the ARIMAX models only slightly improve many of the measures, including MAE and MAPE. This initial Prophet model outperforms even the ARIMAX models, which, at their moment of evaluation, appeared significantly better than ARIMA/SARIMA. MAE decreases by over 4 encounters, while MAPE decreases by almost 4%. Unlike ARIMA, AIC/BIC are not available for Prophet models. These two measures are Bayesian-based, rather than likelihood-based, which AIC/BIC use to determine a model's goodness of fit.

This Prophet model is the best choice out of all models created to forecast Sarpy County's monthly mental health encounters—at least up until this point. Given the "black box" nature of these Prophet models, data analysts and scientists are often left in the dark on the model's interpretability. How are discrete hyperparameters chosen, and how are continuous ones optimized? Most of the hyperparameters that one can alter within a Prophet model are actually recommended to be left untouched, and to be optimized by the software itself. Prophet documentation lists some of its hyperparameters that "can be tuned," which includes: *changepoint_prior_scale* (how flexible the general trend can be, primarily at changepoints), *seasonality_prior_scale* (like *changepoint_prior_scale,* how flexible the seasonality can be) and *seasonality_mode* ("additive" or "multiplicative"; is the variance in seasonal fluctuations changing over time?). These three hyperparameters are added to the list to be adjusted, in addition to one other one: *changepoints.* This hyperparameter defaults to *None*, but it is where one can include potential dates that a trend change is initiated. Recall that the ARIMAX models include the exogenous regressor, *medicaid_rollout,* to account for this exact change.

The four Prophet hyperparameters are ultimately decided to be tuned while simultaneously performing CV. Performing hyperparameter-tuning in unison with CV allows the different combinations of hyperparameters to be fit along many training subsets and forecast horizons, in hopes of finding parameters that fit future monthly encounter aggregates with higher accuracy. Prophet documentation recommends testing out values within the range [0.001, 0.5] for *changepoint_prior_scale,* values within the range [0.01, 10] for *seasonality_prior_scale,* and both "additive" & "multiplicative" inputs for *seasonality_mode* (the first two scales are logistic in nature, so equal increases/decreases will not correspond to equal adjustments). Five different *changepoints* are also added for CV: *None*, October 1st, 2020, November 1st, 2020, December 1st, 2020, and January 1st, 2021. All of these can be argued as potential points in time where the trend is altered by the impact of Medicaid expansion and the COVID-19 pandemic. This hyperparameter tuning and CV Python implementation can be found in Appendix 21.

This process takes roughly 15 minutes to complete using a parallel process grid-search method (200 models), and results appear conclusive for which set of hyperparameters are best for modeling the process. The following set of hyperparameters are optimal for 3 out of the 4 (MAE, MAPE, RMSE) performance metrics: c*hangepoint_prior_scale*=0.01, s*easonality_prior_scale*=0.01, *seasonality_mode*=”multiplicative”, *changepoints*=November 1st, 2020. A final Prophet model is finally created using this hyperparameter solution set on the entire dataset. Figure 19 shows fitted (dark blue line) vs. actual values (black points) in addition to a forecast through 2025 (with a 95% prediction interval band; light blue band) while Figure 20 displays the different components, *trend* and *yearly*, that the model uses for its seasonal *s(t)* component. Forecasts tend to level out at the 50 encounters mark as intended. The *trend* possesses logistic decay, and the *yearly* component shows the number of encounters tends to peak in October and stoop to a low in August.

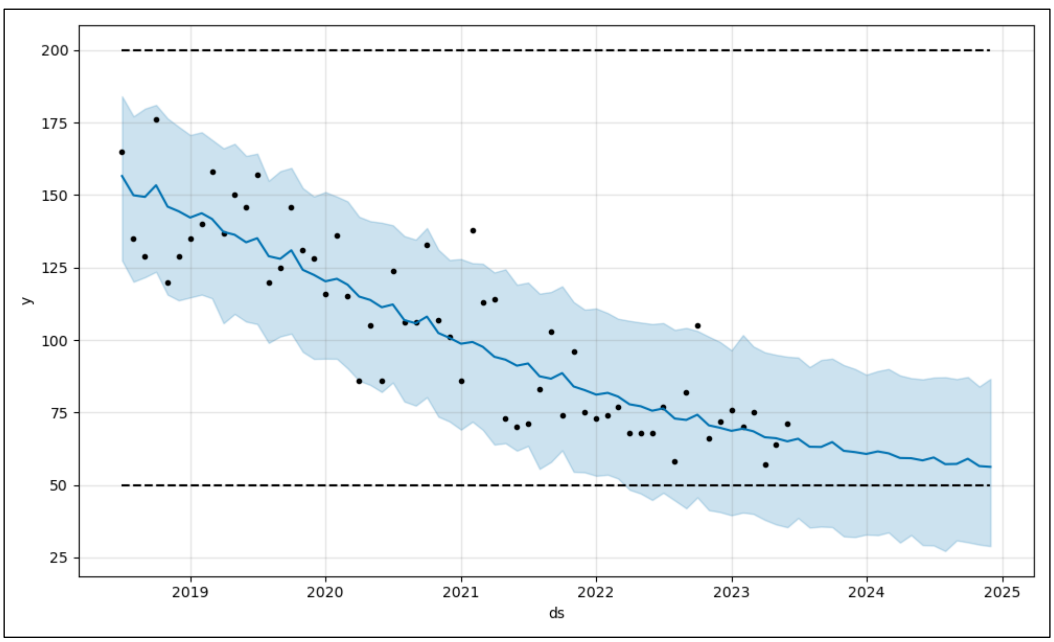


Figure 19. Final Prophet model with saturating minimum (ceiling=200, floor=50). Only two data points fall outside the 95% prediction interval range, indicating a generally reliable model.

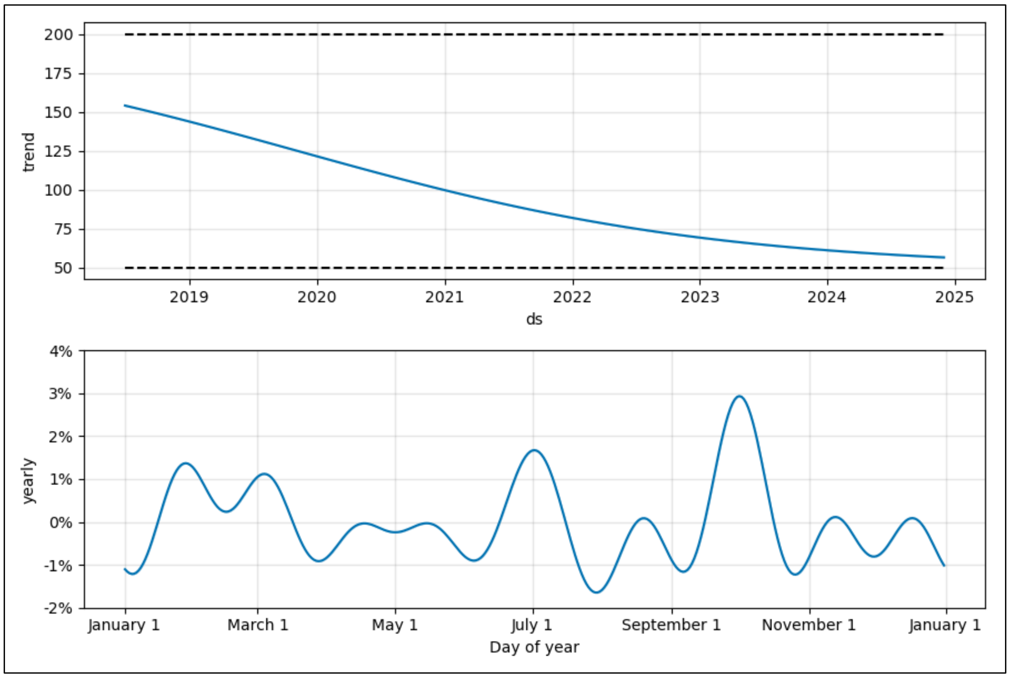


Figure 20. Overall (top) and yearly (bottom) trend components of final Prophet model.

This final Prophet model tests better across the board when compared to the various ARIMA-based models. MAE, MSE, MAPE, and RMSE are calculated at 11.599, 207.305, 11.611%, and 14.398, respectively. Although metrics worsen slightly from the Prophet model composed of default hyperparameters, this final Prophet model's metrics prove to reduce error when applied to many subsets of the dataset—as proven through the CV process.

4. Forecast Implications

There are many implications that arise from the previous model forecasts. First off, many of the forecast long-term trends match the general prior assumptions of Region 6—especially for the CDS datasets. Although Region 6 portions out time to analyze these CDS datasets in their entirety (all mental health encounters for providers under their coverage), a focused approach on Sarpy County residents has not been done. The number of monthly encounters for this subset of encounters closely resembles the entire dataset, although obviously on a much smaller scale. Decreases in encounters are certainly noticeable in the latter half of calendar year 2020 around the Medicaid expansion rollout, which could have also occurred through long-term effects as a result of the COVID-19 pandemic. For the entire dataset, the encounter decreases takes on a gradual decline from October 2020 through December 2022, where it then appears to reach a new baseline. When filtered to Sarpy County residents, there is still the mean adjustment that takes place after October 2020. However, this intervention is not as defined (hence the reason many *changepoints* are used in CV for the Prophet model). Regardless of any sort of model-implemented intervention, Region 6 is content to see both ARIMA and Prophet long-term forecasts for monthly encounters take on the trajectory that they expected.

Second, long-term encounters forecasts would provide Region 6 assurance they can proactively plan for changes that would affect service funding. In essence, Region 6 is the middleman between its' community providers and the state legislature—which is the source of service funding. Any dramatic decline, such as the one seen within the monthly encounters model, would subsequently leave Region 6 scrambling to develop new services/programs. If funding is not going toward services already established (and there are no "new" programs in development that can absorb the residual dollars), the state will likely assume Region 6's providers are receiving more than necessary. The reality is that mental health programs and initiatives cannot be created and put into place instantaneously, so it is vital to understand the direction long-term encounters. Prior knowledge of such a trend can provide sufficient time for Region 6 to devise plans, and spend these dollars that would otherwise sit idle.

Weekly booking forecasts, although not as direct, would benefit Region 6 for a couple of different reasons. Even though Sarpy County population growth is not factored into the forecasting models for SCDC data (since only a relative sliver of booking data is made available for modeling), it is assumed that bookings will generally increase as Sarpy County continues its' linear growth trend—as shown in section 3.3. Also, within the general booking population lies individuals labeled as having *SMI*s, which will inevitably increase along with the general booking aggregates over time. With more data points made available, forecasts would be useful to track the proportion of *SMI* individuals relative to the overall booking population. This relates back to the intersection that lies between individuals who are booked into local jails and those with serious mental illnesses. Understanding these proportions over time, as would be evident through fitted and forecasted data points, would be important to understand the effects of incarceration on mental health status (or vice versa).

5. Conclusion and Future Work

There are countless takeaways from the preceding analysis and forecasts. To begin, the SCDC jail bookings dataset prove to be a useful tool for determining differences between inmates with serious mental illnesses and those without. It is determined that the SMI group has a higher average number of bookings per individual than Non-SMI group, among other startling discrepancies. Much of the analysis and information gathered from the jail dataset demonstrates what the previously mentioned *Stepping Up Initiative* is trying to raise awareness toward. Those suffering from a serious mental illnesses tend to be more frequent visitors of local jails. These same individuals also find themselves staying longer in jails when compared to the non-SMI population for equally severe crimes.

The CDS datasets explore the Sarpy County resident population as a whole, and primarily those that are receiving treatment for mental health or substance abuse related issues. Frequencies and their corresponding percentages tend to differ significantly from the 2022 ACS estimates, indicating a difference in distribution between all Sarpy County residents and those consumers voluntarily/involuntarily receiving services from Region 6's providers.

To reiterate from the prior section, forecasts would provide a means for either confirming or rejecting the notion that the providers' allocated budgets will be sufficient. Expecting a sudden decline, like was the case from Medicaid expansion in late 2020 (along with effects from COVID-19 and a shrinking provider-level workforce), would allow Region 6 to shift their focus to other programs/resources that can absorb these unused funds.

The approaches used throughout this paper appear to be useful in terms of gauging general trends in jail bookings and mental health encounters. As always, having access to more data (i.e. a longer viewing window) would provide valuable data points to fit time series models upon. This is especially true for the CDS datasets which contained the obvious intervention sometime between late 2020 and early 2021. Gaining access to pre-fiscal year 2019 data could fill in various gaps that are otherwise unknown about the pre-intervention general trend. Would it have been the same as post-intervention? Would the underlying autocorrelation structure be the same pre-Medicaid expansion as post-Medicaid expansion? The jail dataset would also be more impactful if it contained more records over a longer period of time—perhaps years. Would the growth rate seen in Sarpy County's population over the last decade match the growth rate in weekly bookings at SCDC? How do bookings compare to neighboring counties that Region 6 covers, especially Douglas County?

Standard ARIMA, seasonal ARIMA, ARIMAX, and Prophet models are all explored as time series models for this project. The presented model residuals are able to pass a handful of diagnostic tests, including ACF/PACF re-plots, QQ plots, the Ljung-Box test, and the Augmented Dickey Fuller test. However, as is the case with most mathematical models, what they gain in one area they tend to lose in another. The models created for this report are excellent in terms of comprehensibility. For the most part, they are "easy" to implement, comprehend, and forecast new data points. With the ARIMA family of models in particular, it becomes somewhat cumbersome to determine significant features and estimate lag terms that should be included within a model. The models also become generally weak within the long-term predictions. Forecasted values tend to approach the process mean, or prediction intervals stretch to ranges that provide no insight whatsoever. Using a deep learning approach via recurrent neural networks (RNNs) could be an alternative to some of the standard machine learning models explored in this project. RNNs tend to be theoretically more complex, but if the data points can be modeled to a better degree for future weeks/months, it would be an interesting route to take for future research.

Regardless of model type, one thing is evident through analyzing these various datasets: mental health issues and incarceration are tough to ignore for many individuals. Not only that, but they tend to be persistent among the *same* individuals time and time again. Breaking this recurring cycle for the affected population is a major key toward improving their overall quality of life. This project's analysis aims to provide Region 6 confirmation of the successes within the behavioral health's public sector, and concurrently, what problems have yet to be addressed.

References

"Contacts | Nebraska Disaster Behavioral Health." *Nebraska Disaster Behavioral Health*, 2022, www.disastermh.nebraska.edu/contact/. Accessed 13 May 2024.

Cryer, Jonathan D, and Kung-Sik Chan. *Time Series Analysis with Applications in R*. New York London Springer New York London Springer, 2011.

Norris, Louise. "ACA Medicaid Expansion in Nebraska [Updated 2022 Guide]." *Healthinsurance.org*, 2024, www.healthinsurance.org/medicaid/nebraska/.

"Region 6." *Region 6 Behavioral Healthcare*, 21 Sept. 2018, www.regionsix.com/.

Taylor, Sean J, and Benjamin Letham. "Forecasting at Scale." *Practical Data Science for Stats*, 27 Sept. 2017, peerj.com/preprints/3190/, https://doi.org/10.7287/peerj.preprints.3190v2.

United States Census Bureau. "Explore Census Data." *United States Census Bureau*, 2023, data.census.gov/.

Zeng, Zhen. "Preliminary Data Release - Jails 2022 | Bureau of Justice Statistics." *Bjs.ojp.gov*, Sept. 2023, bjs.ojp.gov/preliminary-data-release-jails-2022. Accessed 13 May 2024.

"2021 Annual Report: Sarpy's Economic Growth Continues." *Sarpy County, NE*, 20 Dec. 2021, www.sarpy.gov/CivicAlerts.aspx?AID=113. Accessed 13 May 2024.

Appendix

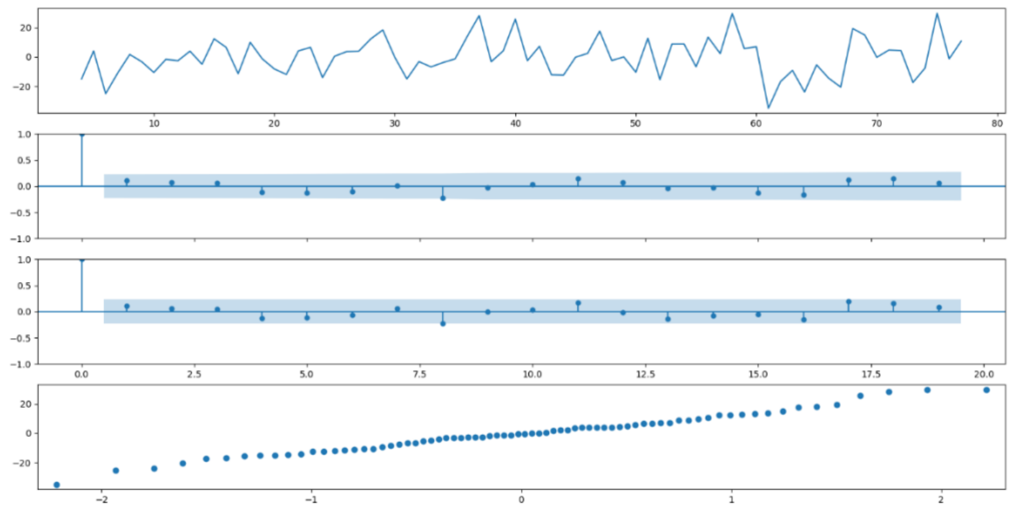
Appendix 1. Python modules used for analysis/forecasting and their corresponding versions.

Module	Version
numpy	1.24.3
pandas	2.0.3
statsmodels	0.14.0
matplotlib	3.7.2
scipy	1.11.1
prophet	1.1.5
scikit-learn	1.3.0

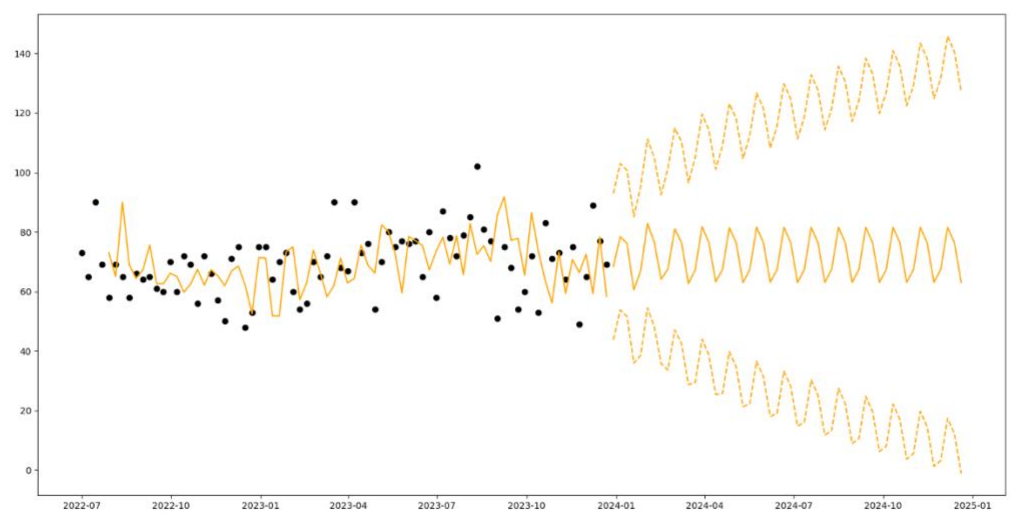
Appendix 2. Sarpy County Population exponential smoothing models Python implementation.

```
# Simple exponential smoothing
trend1 = SimpleExpSmoothing(populations[:-2]).fit()
es_estimate1 = trend1.forecast(2)
pop_residuals1 = trend1.resid

# Exponential smoothing
trend2 = ExponentialSmoothing(populations[:-2], trend='mul', damped_trend=True).fit(optimized=True)
es_estimate2 = trend2.forecast(2)
pop_residuals2 = trend2.resid

# Holt's Exponential Smoothing
trend3 = Holt(populations[:-2]).fit(optimized=True)
es_estimate3 = trend3.forecast(2)
pop_residuals3 = trend3.resid
```

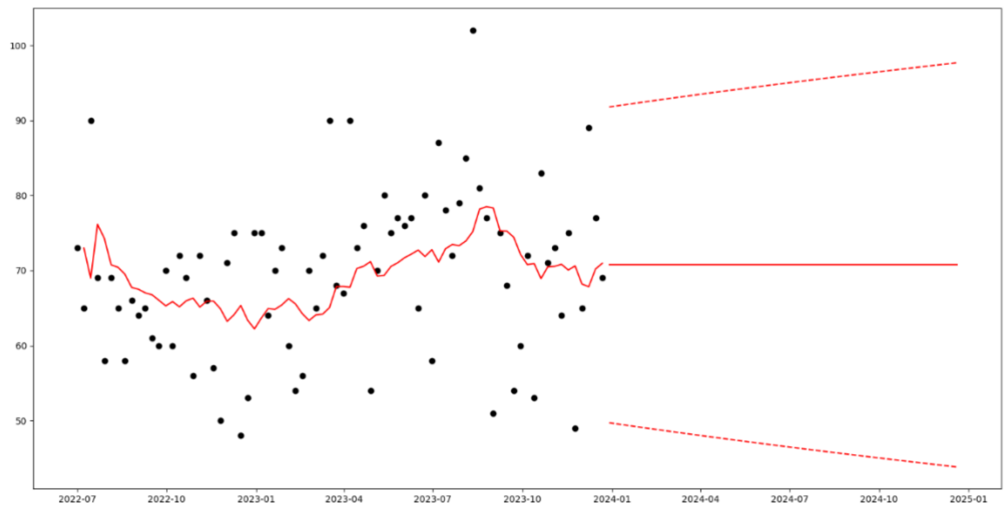
Appendix 3. Jail bookings diagnostic plots, SARIMA(0,0,0)(1,1,0)$_4$.



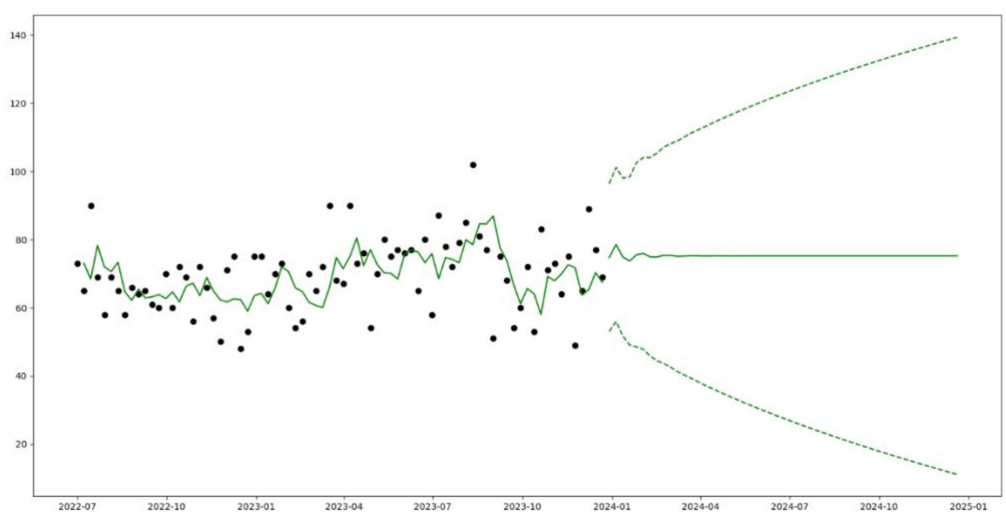
Appendix 4. Jail bookings forecast with 95% PI, SARIMA(0,0,0)(1,1,0)$_4$.



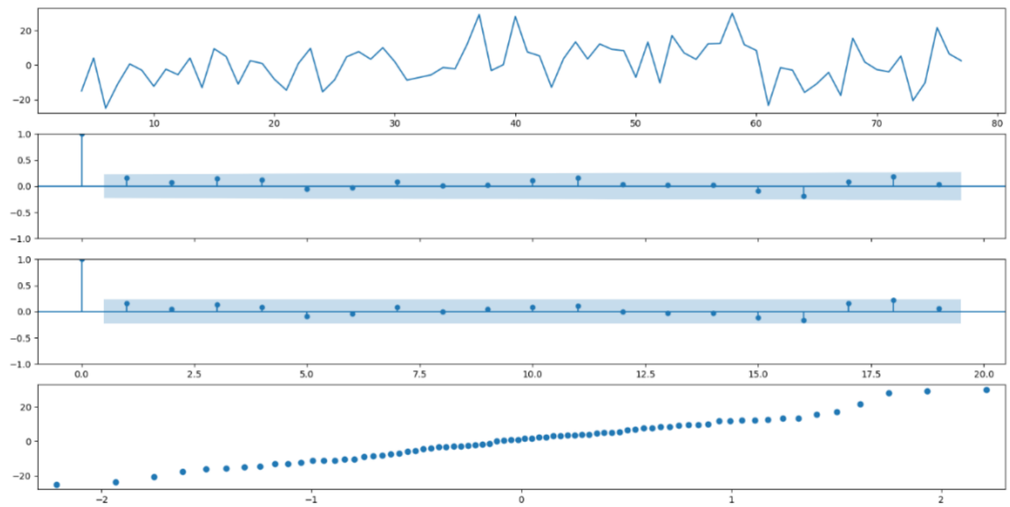
Appendix 5. Jail bookings forecast with 95% PI, ARIMA(0,1,1).



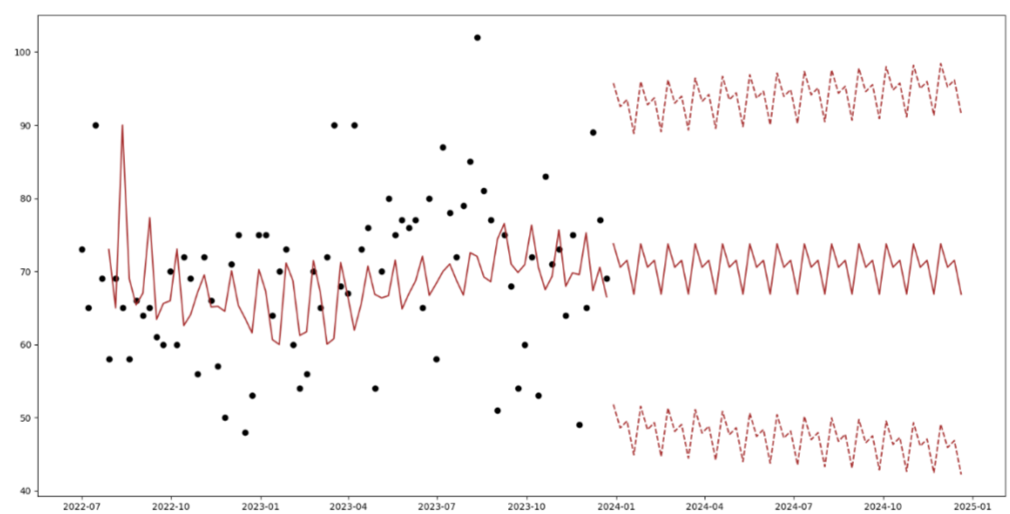
Appendix 6. Jail bookings forecast with 95% PI, ARIMA(3,1,0).



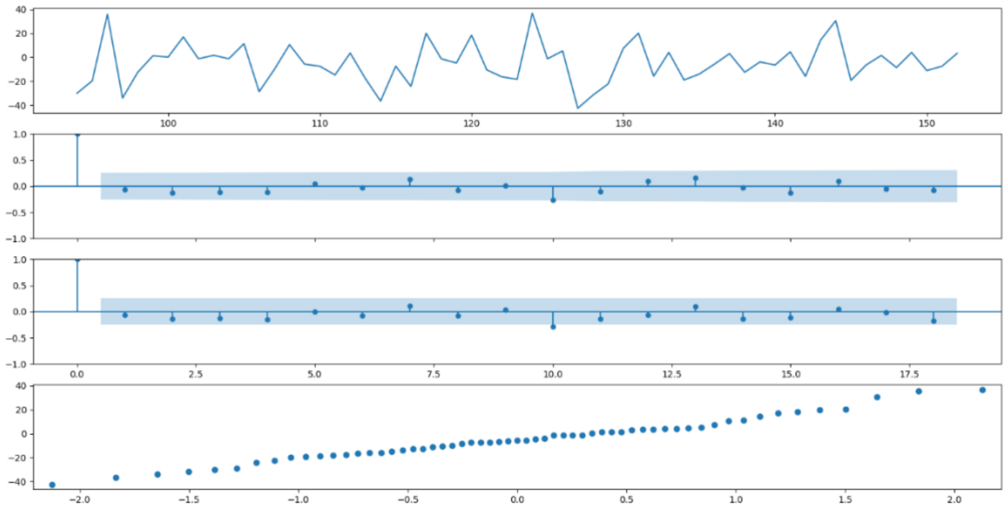
Appendix 7. Jail bookings diagnostic plots, SARIMA(0,0,0)(0,1,1)$_4$.



Appendix 8. Jail bookings forecast with 95% PI, SARIMA(0,0,0)(0,1,1)$_4$.

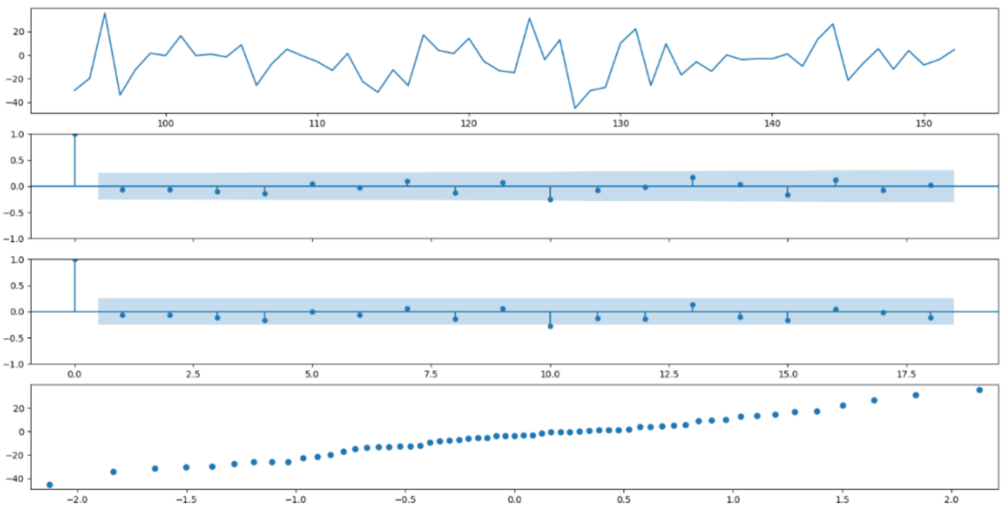


Appendix 9. CDS diagnostic plots, SARIMA(0,1,1)(1,0,0)$_6$.

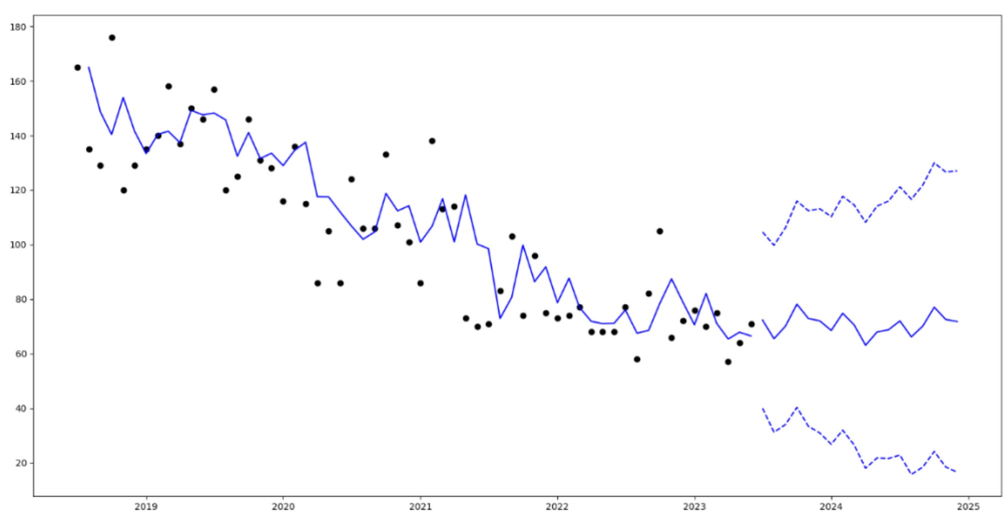


Appendix 10. CDS forecast with 95% PI, SARIMA(0,1,1)(1,0,0)$_6$.

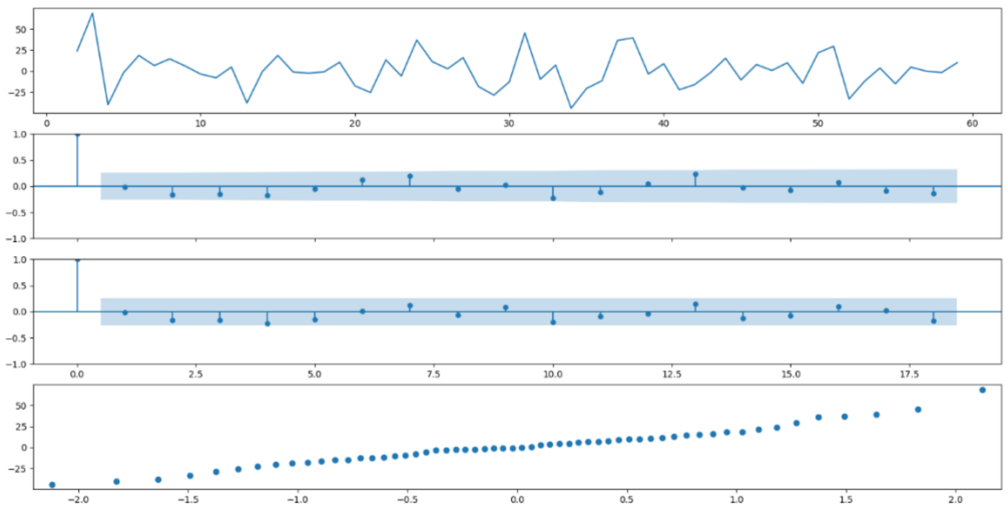
Appendix 11. CDS diagnostic plots, SARIMA(0,1,1)(1,0,1)$_6$.



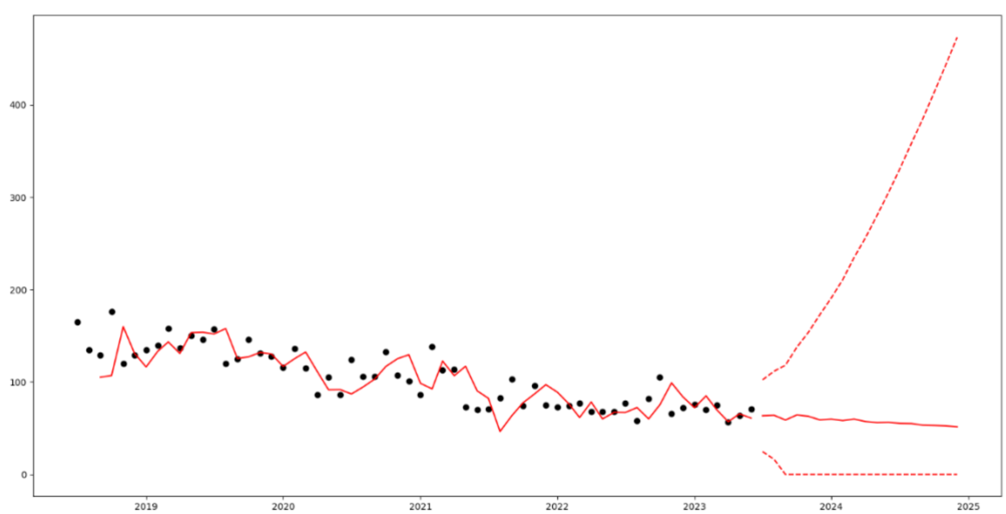
Appendix 12. CDS forecast with 95% PI, SARIMA(0,1,1)(1,0,1)$_6$.



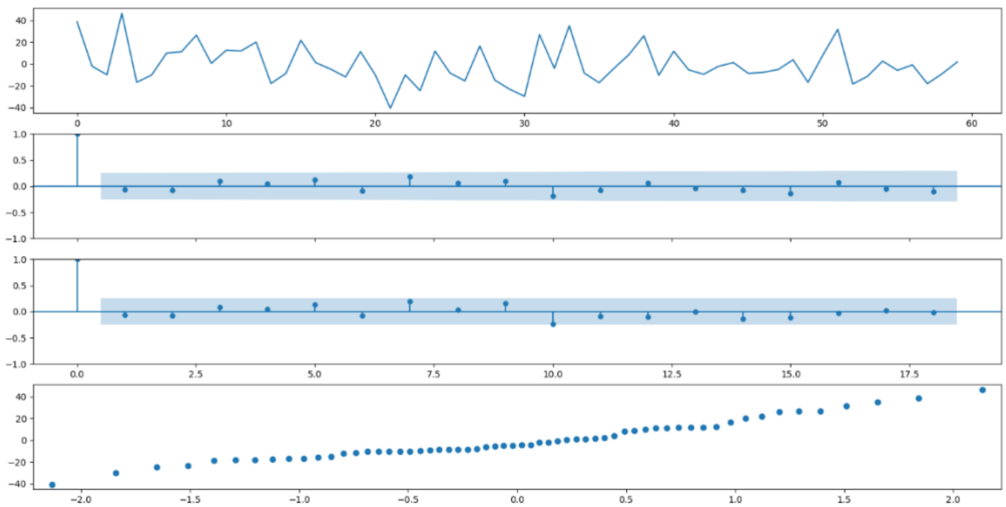
Appendix 13. CDS diagnostic plots, ARIMA(4,2,0).



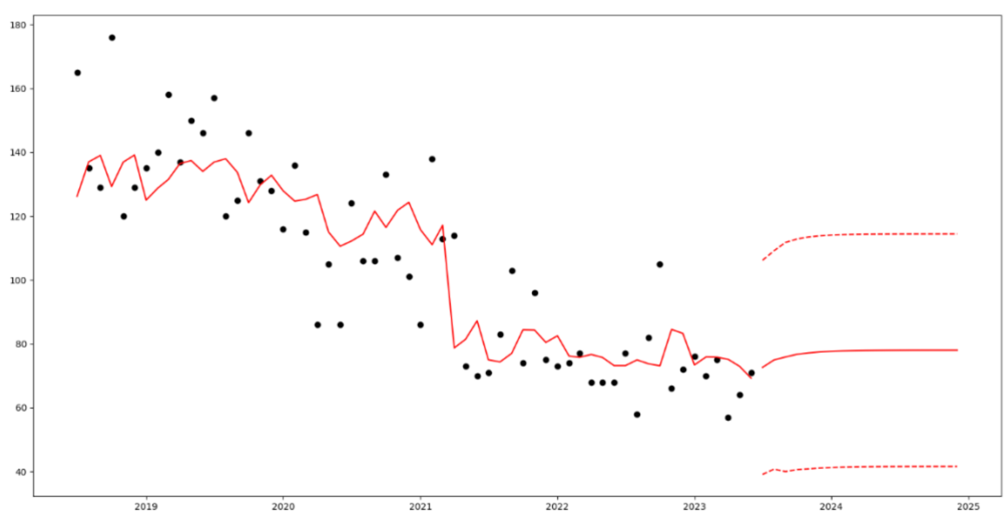
Appendix 14. CDS forecast with 95% PI, ARIMA(4,2,0).



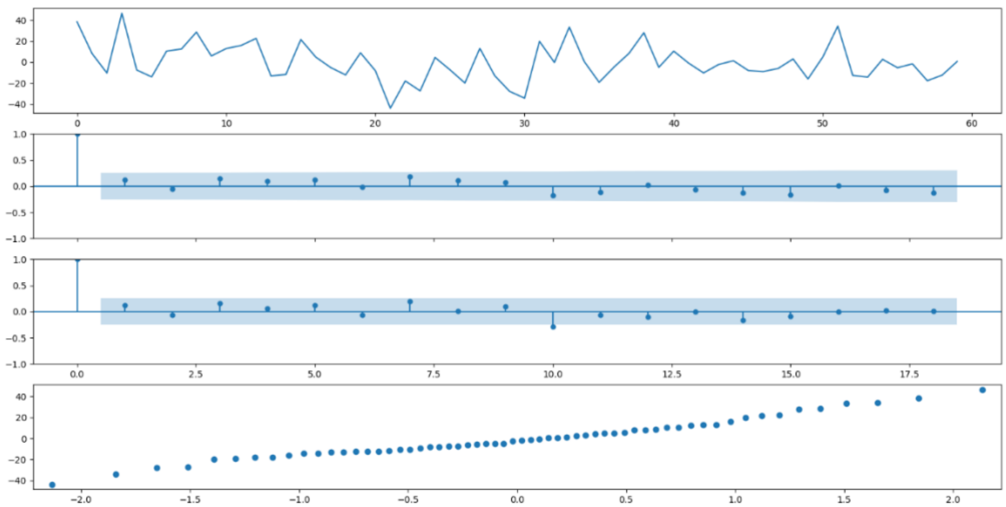
Appendix 15. CDS diagnostic plots, ARIMAX(2,0,0).



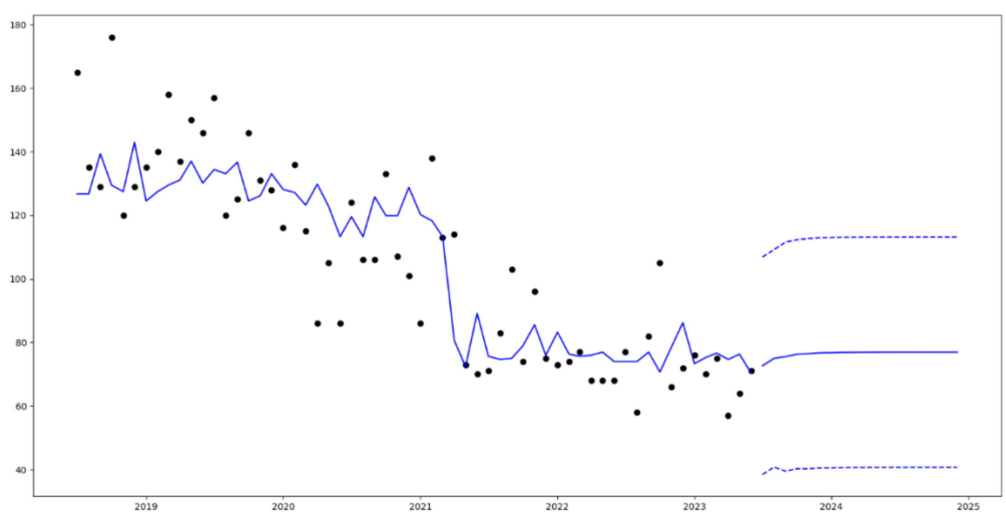
Appendix 16. CDS forecast with 95% PI, ARIMAX(2,0,0).



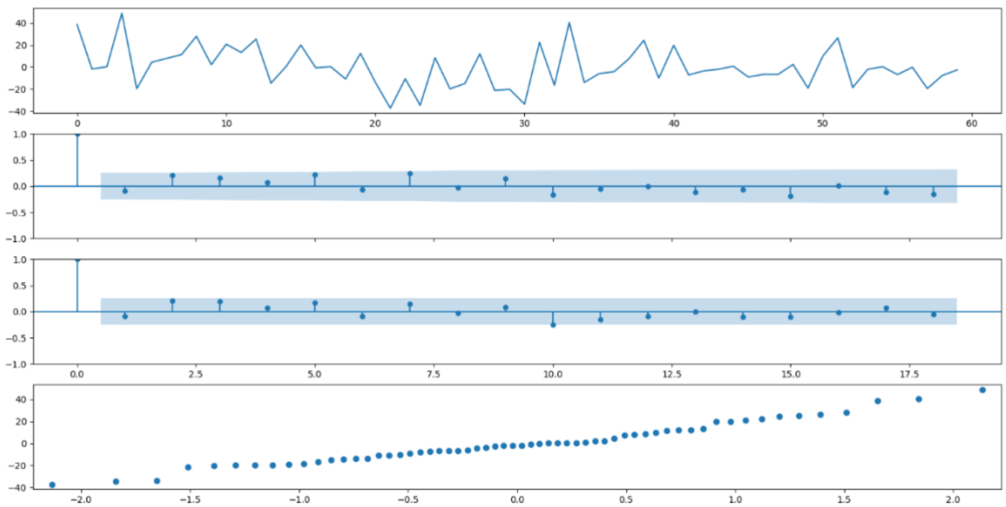
Appendix 17. CDS diagnostic plots, ARIMAX([2],0,0).



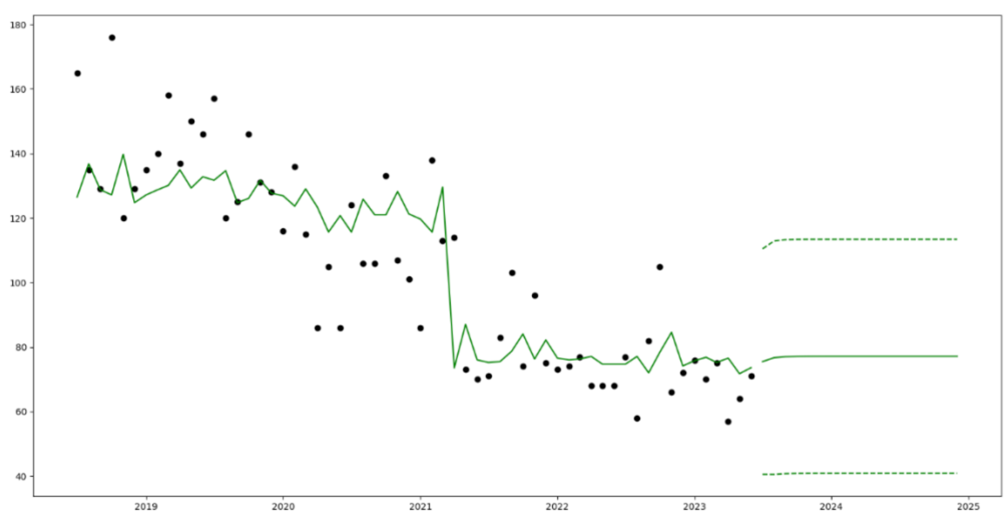
Appendix 18. CDS forecast with 95% PI, ARIMAX([2],0,0).



Appendix 19. CDS diagnostic plots, ARIMAX(1,0,0).



Appendix 20. CDS forecast with 95% PI, ARIMAX(1,0,0).



Appendix 21. CDS cross-validation Python implementation, Prophet.

```python
'''Perform cross validation in conjuction with parameter hypertuning of prophet model with enabled saturation'''

#create cutoff dates. each split will train data up to these points
cutoffs = pd.date_range(start='2021-01-01', end='2022-07-01', freq='1MS') - pd.to_timedelta(1, unit='day')

#create lists of parameters we want to test
seasonality_mode = ['additive','multiplicative']
changepoint_prior_scale = [0.001, 0.01, 0.1, 0.5]
seasonality_prior_scale = [0.01, 0.1, 0.5, 1, 10]
changepoints = [pd.to_datetime('2020-10-01'), pd.to_datetime('2020-11-01'), pd.to_datetime('2020-12-01'), pd.to_datetime('2021-01-01'), None]

params = {}
combos = [(sm, cps, sps, c) for cps in changepoint_prior_scale for sm in seasonality_mode for sps in seasonality_prior_scale for c in changepoints]
modNum = range(len(seasonality_mode) * len(changepoint_prior_scale) * len(seasonality_prior_scale) * len(changepoints))
modDiagnostics = []

for c, m in zip(combos, modNum):
    params[m] = {'seasonality_mode': c[0], 'changepoint_prior_scale': c[1], 'seasonality_prior_scale': c[2], 'changepoint': c[3]}

#Create function to calculate metrics across each split from cross-validation results
def partitionMetrics(vals):
    mse = mean_squared_error(vals['y'], vals['yhat'])
    rmse = np.sqrt(mean_squared_error(vals['y'], vals['yhat']))
    mae = mean_absolute_error(vals['y'], vals['yhat'])
    mape = mean_absolute_percentage_error(vals['y'], vals['yhat'])
    return pd.Series(dict(mse=mse, rmse=rmse, mae=mae, mape=mape))


#loop through grid search and perform cross-validation
for i in modNum:
    tmpMod = Prophet(growth='logistic', interval_width=0.95, seasonality_mode=params[i]['seasonality_mode'], changepoint_prior_scale=params[i]['changepoint_prior_scale'], seasonality_prior_scale=params[i]['seasonality_prior_scale'], changepoints=params[i]['changepoint']).fit(admissionsProphet)
    tmp_cv = cross_validation(tmpMod, cutoffs = cutoffs, horizon='180 days', parallel='processes')

    #remove rare cases where it forecasted 7th month
    tmp_cv['predStep'] = tmp_cv.groupby('cutoff')['ds'].rank(method='first',ascending=True)
    tmp_cv = tmp_cv[tmp_cv['predStep'] <= 6]

    #calculate MAE, MSE, RMSE, MAPE
    modDiagnostics.append(tmp_cv.groupby('cutoff').apply(partitionMetrics).reset_index()[['mse','mae','rmse','mape']].mean(axis=0).tolist())
    print(f'Model {i} completed.')


modDiagnostics = pd.DataFrame(modDiagnostics, columns=['MSE','MAE','RMSE','MAPE'])
```