본죽 매출액의 계절성 파악 및 ARIMA모델을 활용한 예측모델설정

2015111593
경영학부
김정의

목차

Ⅰ.서론

 요식업에서 매출액을 예측할 수 있다면 효율적으로 인력을 배치하고 식자재 등을 적절한 시기에 구매하여 재료의 신선함을 올리는 등 얻을 수 있는 이점이 매우 많을 것이다. 본죽에서 아르바이트를 하며 얻은 2017년부터 2019년의 본죽 위례점의 일별 매출데이터를 바탕으로 매출액을 예상할 수 있는 모델을 만들고자한다.

 그전에 모델을 설정하기에 앞서 매출액에 7일의 계절성이 있는지를 확인해보고자 한다. 7일의 계절성의 여부를 확인하는 것은 가게를 운영함에 있어 매우 중요한 요소이다. 일반적으로 음식점들은 요일별로 적정인원을 배치한다. 위 가게의 경우 일반적인 음식점들처럼 주말에 매출액이 높을 것으로 예상을 하고 주말에 3명 평일에 2명의 인원을 배치하였다. 하지만 오히려 평일에 매출액이 높거나 주말에 매출이 적은 등 의외의 경우가 발생하는 등 패턴을 정확히 확인할 수 없었다. 가게에서 재료를 손질하고, 청소를 하기 위해 반드시 7일에서 이틀은 3명의 인원이 필요한 날이 존재하였는데 이 이틀을 어느 요일에 배정하는 것이 최고의 선택일까? 또한 특정 요일이 매출이 많고 계절성이 정말 존재한다면 이를 고려한 모델링을 통해 좀 더 정확한 예측모델을 만들 수 있을 것이다.

 본 논문에서는 이러한 주말이 다른 요일에 비하여 계속 매출이 주기적으로 높다면 발생하게 될 주기 7일의 계절성 존재여부를 시각화를 통하여 알아보고자 한다. 주기성이 만약 발견된다면 계절성을 고려한 SARIMA와 같은 모델을 고려하고 그렇지 않을 경우 기본적인 ARIMA모델을 통하여 모델을 만들어보고자 한다.

 본론 1장에서는 우선 데이터의 시기, 특징들을 간단히 설명하고자 한다. 또한 시각화를 통하여 매출데이터에 추세, 계절성 등이 존재하는지를 확인해볼 것이다. 2장에서는 모델을 만들 것 이다. ACF, PACF 그래프를 통하여 차분의 차수와 ARIMA의 모수를 결정하고 최종적으로 잔차 분석을 통하여 모델의 기본가정 성립여부를 확인할 것이다. 3장에서는 만들어진 모델을 테스트 데이터를 통하여 비교하고자 한다. 마지막으로 결론부분에서 모델링의 결과를 종합하고 본 연구를 통하여 얻은 결론들을 정리해보고자 한다.

Ⅱ.본론

1장 데이터 설명 및 탐색적 자료분석

가. 사용데이터 설명

2017년부터 2019년까지 본죽 위례점의 일 매출액 데이터를 이용하여 모델링하였다. 다만 결측치가 있는 날들이 존재하였으며 1월 1일, 5월 5일 과 같이 휴일이 있는 날의 주는 포스기의 오류로 인해 매출이 제대로 기록되지 않았다. 이러한 주들을 빼고 2017년 1월 9일 월요일을 시작점으로 하여 2017년에는 287일, 2018년에는 266일 ,2019년에 266일을 합하여 총 819개의 시점을 바탕으로 모델링하였다. 약 3년치의 데이터를 바탕으로 모델링을 한 후 2020년 1월 6일~ 1월 19일까지 약 2주간의 매출액을 예측해보고 이를 실제 데이터와 비교해보고자 한다.

나. 시각화를 통한 탐색적 자료분석

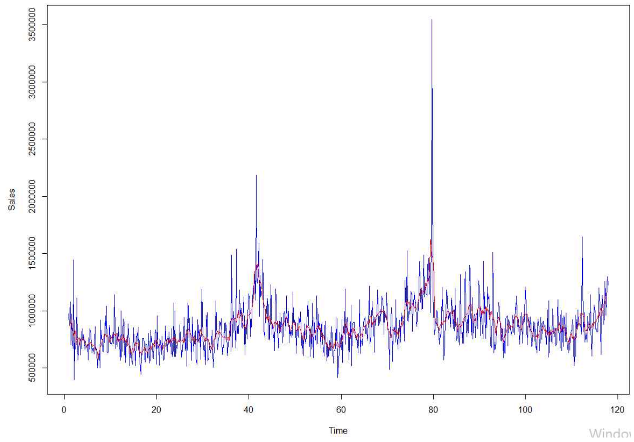
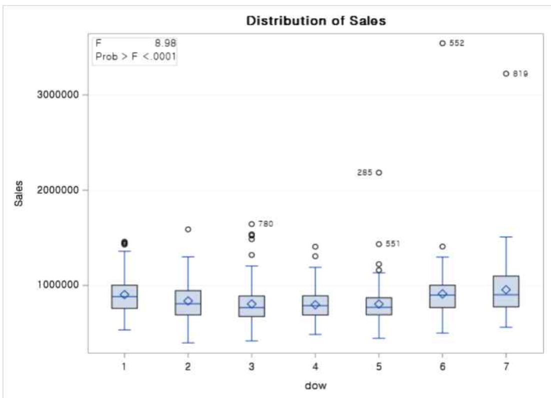


그림 <1> 파란선은 실제 예측값, 빨간선은 추세 그래프

파란선은 약 3년의 데이터를 그래프로 시각화한 결과이다. 여기에 추세성분을 이동평균법을 이용하여 빨간선으로 표시하였다. 빨간선을 살펴보면 겨울에 해당하는 시기에 매출이 상승하고 겨울이 끝나가면 매출이 다시 하락하는 패턴을 보였다. 죽이 뜨거운 음식이기 때문에 추운 겨울에 더 잘 팔리기 때문에 생긴 결과라 생각하였다. 다만 매출액의 변동이 커 겨울이라고 하더라도 여름보다 매출액이 적은 경우가 꽤 존재



Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	2.4567747E12	409462454072	8.93	<.0001
Error	812	3.7221815E13	45839673274		
Corrected Total	818	3.9678589E13			

<그림 2> 매출데이터의 박스 플롯. **<그림 3> ANOVA 결과.**

하는 것을 확인할 수 있었다.
  다음으로 매출액이 7일의 주기를 가지는지를 확인하기 위하여 각 요일에 대하여 매출액의 분포를 확인해보았다. 박스 플롯을 그려본 결과 주말이 다른 요일에 비해 매출액이 비교적 높은 것을 확인할 수 있었다. 또한 ANOVA를 통해 각 요일별수준의 유의성을 검정해보았을 때도 p-value의 값이 매우 낮아 요일별수준은 유효하다고 말할 수 있었다.
 하지만 주말이 대체적으로 매출액이 높은 것은 사실이지만 월요일의 경우도 매출액이 높아 토요일과 거의 비슷하였다. 또한 각 요일별 매출액의 변동이 매우 큰 것을 확인할 수 있었다.

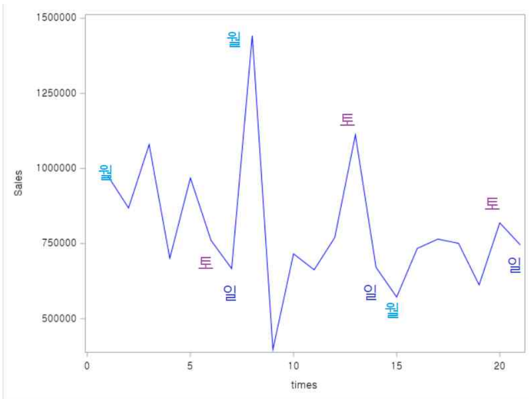


그림 <4> 21일(3주) 간격으로 살펴본 매출액

 데이터가 너무 많았기 때문에 자세히 보기 위해 그림 <4>에서처럼 21일씩 3주 간격으로 시각화를 해보았다. 대부분의 시점에서 계절형 패턴이 보이지 않는 경우가 많았고 한 주안에서 항상 토, 일, 월 이 매출액이 많다고 말할 수 없었다. 즉, 요일별 수준에 어느 정도 차이가 있는 것은 맞지만 주기적으로 차이를 보이는 것은 아니기 때문에 이를 7일의 계절적 주기를 가진다고 말할 순 없었다. 즉, 시각화를 통해 생각해봤을 때 데이터에 계절성이 있다고 말할 수는 없었다. 다만 모델을 설정할 때 7일의 계절성을 사용한 모델을 시도는 해보는 것이 좋다고 생각하였다.

2장 모델 설정

가. 차분을 통한 정상화

 앞서 시각화를 통해 7일의 계절성분은 확신할 수 없었지만 ①겨울에 매출이 증가하는 주기 1년의 계절성분은 확인할 수 있었다. 또한 그림<5>와 같이 ACF 그래프를

확인했을 때 ②자기상관계수의 값이 서서히 하락하는 등 ACF 그래프에서

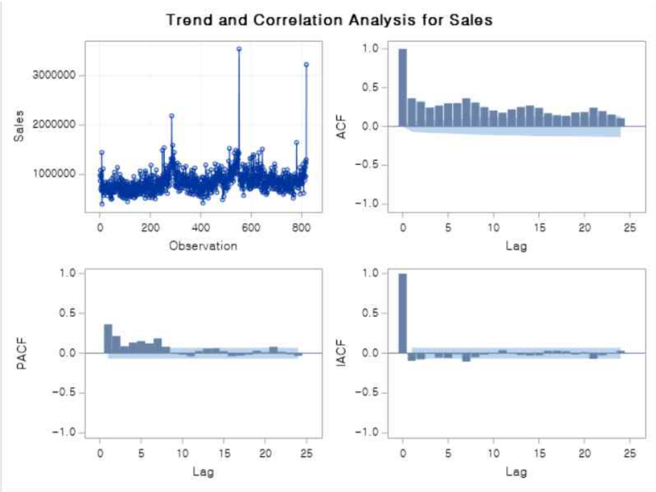
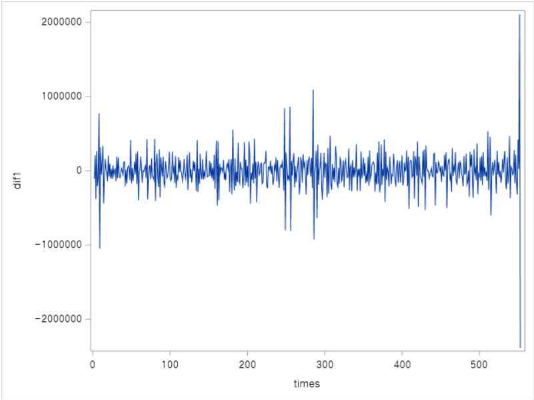


그림 <5> 매출 데이터의 ACF, PACF 그래프

패턴이 나타나는 것을 확인할 수 있었다. 따라서 추가적인 검정을 할 필요가 없이 추세가 있는 것이 분명하여 데이터에 비정상성이 있다고 판단하였고 차분을 통하여 정상성을 갖도록 하였다. 우선 가장 기본적인 1차 차분을 하고 그 결과를 시각화 해보았다.



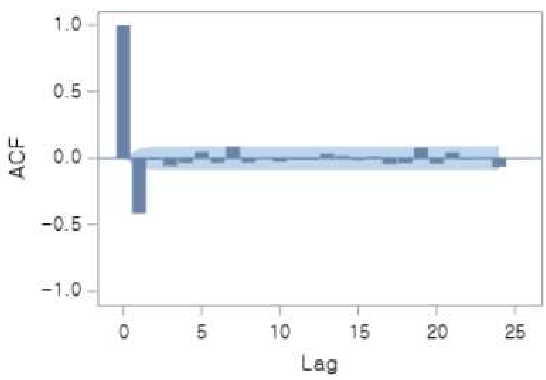


그림 <6> 1차 차분 후의 그래프

그림 <7> 1차 차분 후 ACF 그래프

차분을 진행한 결과 왼쪽 그림과 같이 패턴을 찾아볼 수 없었으며 ACF 그래프에서도 LAG 1시점에서 -0.4 의 값이 나온 후 0에 가깝게 절단되는 모습을 볼 수 있었다. 따라서 1차 차분만으로도 충분히 데이터가 정상성을 가지게 변환되었기 때문에 1차 차분을 한 모형을 택하기로 하였으며 추가적으로 차분을 진행할 필요 또한 없다고 생각하였다.

나. ARIMA Parameter 설정

1차 차분을 한 데이터를 사용하였으므로 ARIMA(p,1,q)에서 적절한 P와 Q를 설정해 주어야 했다.

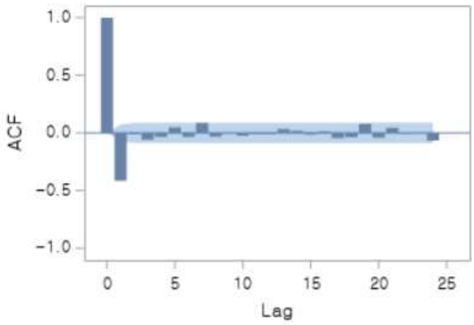


그림 <8> 1차 차분 후 ACF 그래프

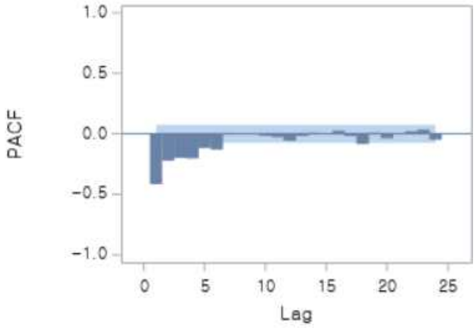


그림 <9> 1차 차분 후 PACF 그래프

다시 ACF 그래프와 PACF 그래프를 살펴보았을 때 ACF 그래프에서 1시점이후 0으로 절단되는 모습을 보이고 PACF 그래프에서 지수적으로 상관계수 값이 감소하는 그래프의 뒤집어진 모습을 확인할 수 있었다. 따라서 ARIMA(0,1,1)을 우선 고려해보기로 하였다. 또한 PACF의 경우 6시점까지는 유의한 값을 보여주는 것을 고려하여 비록 ACF 그래프에서 지수적으로 감소한다고 보기는 어려웠지만 ARIMA(6,1,0)도 고려하기로 하였다.

마지막으로 앞써 시도해보기로 한 7일의 계절성을 반영한 모델을 만들어보았다.
처음에 7일의 계절성을 고려하여 주기 7을 가지는 SARIMA모델을 만들어보고자 하였다. 하지만 SARIMA모델을 만들고자 했을 때 SARIMA(a,1,b)(c,1,e)7에서 c와 e에 어떤 값을 넣더라도 계수 추정치의 p-value의 값이 1과 가까울 정도로 계수가 유의하지 않았다. 따라서 다음과 같은 모델을 만들어 보았다.

$$(1-\phi_7 B^7)(1-B)Z_t = (1-\theta_1 B)\epsilon_t\,, \epsilon_t \quad WN(0, \sigma_\epsilon^2)$$

이는 ARIMA(7,1,1) 모델에서 $Z_{t-1}, Z_{t-2}, Z_{t-3}, Z_{t-4}, Z_{t-5}, Z_{t-6}$ 을 제거한 형태로써 7일 전의 Z(매출액) 값인 Z_{t-7}을 반영하여 요일의 영향은 고려하면서 ϵ_{t-1} 를 통해 Z_t와 가까운 날의 매출액도 반영할 수 있을 것 이라고 생각하였다. 이러한 모델은 모수가 많이 필요한 ARIMA(7,1,1)과 다르게 7일 전의 Z_t를 통하여 요일의 영향은 고려하면서도 모델에 필요한 모수를 최소화 할 수 있을 것이라고 생각하였다. 이를 편의상 @(7,1,1)로 표시하였다.

추가적으로 ARIMA(0,1,1), ARIMA(6,1,0) 만이 아니라 비슷한 주변모델 예를 들어, ARIMA (0,1,2) , ARIMA(7,1,0) 등의 모델도 추가적으로 탐색해보기로 하였다.

모델을 적합해본 결과 후보로 둔 모든 모델에서 AR 또는 MA의 계수는 유의하더라도 상수항의 경우 모든 후보모델에서 유의하지 않은 것으로 나타났다. 이러한 점과

함께 1차 차분한 결과의 시각화그래프 그림 <6> 에서 0을 기준으로 무작위로 진동하는 패턴을 보여주기도 했기 때문에 상수항을 제거한 모형을 적합하였다. 이제 AIC 값을 통하여 모델의 성능을 비교해보았다.

ARIMA(p,1,q)	(0,1,1)	(0,1,2)	(1,1,1)	(6,1,0)	**(6,1,1)**	**(7,1,0)**	**(7,1,1)**	**@(7,1,1)**
AIC	22205	22204	22203	22194	**22187**	**22187**	**22189**	**22184**

표 <1> 각 모델의 AIC 값

그 결과 @(7,1,1) 의 경우 AIC 값이 22184로 가장 낮은 것으로 나타났다. 다만 AIC의 값이 22184와 비슷한 값을 가지는 상위 4개 모델에 포함되는 ARIMA(6,1,1), ARIMA(7,1,0), ARIMA(7,1,1) 세가지 모델도 추가적으로 탐색하였다.

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	0.12720	0.33720	0.38	0.7060	1
AR1,1	-0.68327	0.33658	-2.03	0.0423	1
AR1,2	-0.57419	0.26660	-2.15	0.0313	2
AR1,3	-0.56245	0.21838	-2.58	0.0100	3
AR1,4	-0.48275	0.20506	-2.35	0.0186	4
AR1,5	-0.38301	0.17433	-2.20	0.0280	5
AR1,6	-0.28346	0.13713	-2.07	0.0387	6
AR1,7	-0.07005	0.09510	-0.74	0.4613	7

그림 <10> ARIMA(7,1,1) 모델의 Parameter 추정 결과

우선 ARIMA(7,1,1) 의 경우 Lag=7에 해당하는 AR1,7 계수의 p-value의 값이 0.46으로 매우 높았기 때문에 제외하였다. ARIMA(6,1,1), ARIMA(7,1,0) ,ARIMA(7,1,1)의 경우 그림 <11>,그림 <12>, 그림 <13>에서 p-value의 값이 유의수준 0.05에서 모두 유의한 것을 확인할 수 있었다.

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	0.35250	0.11483	3.07	0.0021	1
AR1,1	-0.45712	0.11389	-4.01	<.0001	1
AR1,2	-0.39481	0.08996	-4.39	<.0001	2
AR1,3	-0.41525	0.07322	-5.67	<.0001	3
AR1,4	-0.34363	0.06751	-5.09	<.0001	4
AR1,5	-0.26497	0.05921	-4.48	<.0001	5
AR1,6	-0.18955	0.04684	-4.05	<.0001	6

그림 <11> ARIMA(6,1,1) 모델의 Parameter 추정 결과

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
AR1,1	-0.80914	0.03492	-23.17	<.0001	1
AR1,2	-0.67284	0.04353	-15.46	<.0001	2
AR1,3	-0.64219	0.04706	-13.64	<.0001	3
AR1,4	-0.55726	0.04832	-11.53	<.0001	4
AR1,5	-0.44555	0.04700	-9.48	<.0001	5
AR1,6	-0.33151	0.04363	-7.60	<.0001	6
AR1,7	-0.10191	0.03503	-2.91	0.0036	7

그림 <11> ARIMA(7,1,0) 모델의 Parameter 추정 결과

Maximum Likelihood Estimation					
Parameter	Estimate	Standard Error	t Value	Approx Pr > |t|	Lag
MA1,1	0.87686	0.01756	49.93	<.0001	1
AR1,1	0.16999	0.03558	4.78	<.0001	7

그림 <12> @(7,1,1) 모델의 Parameter 추정 결과

다. 잔차 분석

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	.	0	.	-0.002	0.002	0.003	-0.000	-0.009	-0.023
12	5.83	5	0.3229	-0.034	-0.013	-0.034	-0.032	-0.044	-0.031

그림 <13> ARIMA(6,1,1) 모델의 포르멘토 검정 결과

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	.	0	.	-0.002	-0.004	-0.002	-0.004	-0.015	-0.026
12	5.72	5	0.3349	-0.038	-0.037	-0.013	-0.023	-0.040	-0.029

그림 <14> ARIMA(7,1,0) 모델의 포르멘토 검정 결과

Autocorrelation Check of Residuals									
To Lag	Chi-Square	DF	Pr > ChiSq	Autocorrelations					
6	8.15	4	0.0861	0.058	0.021	-0.066	-0.017	0.012	0.036
12	17.29	10	0.0682	-0.006	0.058	-0.015	-0.047	-0.062	-0.037

그림 <15> @(7,1,1) 모델의 포르멘토 검정 결과

다음으로 잔차의 White Noise 가정을 확인하기 위하여 포르멘토 검정을 실시하였다. 비 계절성 데이터의 경우 고려하는 최대 시차를 10정도로 사용하는 것이 일반적이기 때문에 1차 차분을 통해 정상성을 가지게 된 데이터로써 Lag=12 까지 포르멘토 검정을 실시하였다.[1] 검정 결과 그림<13>, 그림<14>, 그림<15>에서 확인할 수 있듯이 모든 모델에서 잔차가 White Noise를 따른다는 귀무가설을 유의수준 0.05에서 기각할 수 없었다.

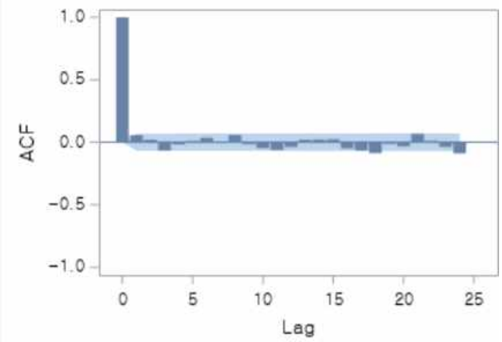


그림 <16> @(7,1,1) 잔차의 ACF

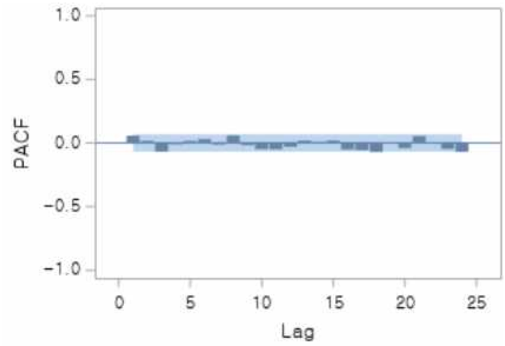


그림 <17> @(7,1,1) 잔차의 PACF

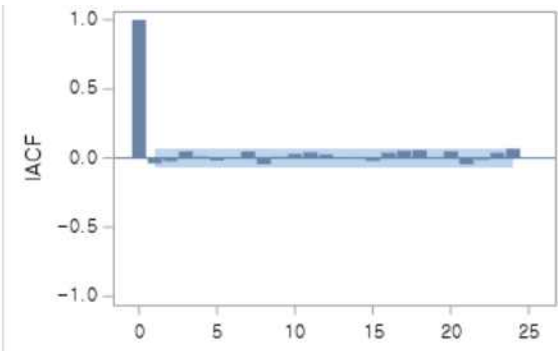


그림 <18> @(7,1,1) 잔차의 IACF

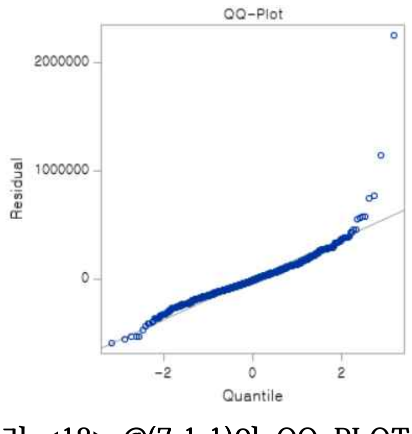


그림 <18> @(7,1,1)의 QQ-PLOT

또한 모든 모델에서 ACF, PACF, IACF 그래프에서 유의한 자기상관계수값이 발견되지 않았으며 잔차들의 QQ-plot을 확인했을 때 모든 모델에서 약 800개 데이터에서 몇몇 값들을 제외하곤 대체적으로 직선에 잘 적합하여 정규성을 만족하는 것까지 확인할 수 있었다.

즉, 모든 모델이 기본적인 가정을 잘 만족한다고 결론을 내렸다. 따라서 세 모델중

1) https://otexts.com/fppkr/residuals.html

에서 AIC 값이 22184로써 가장 낮고 모수들을 적게 사용하여 다른 모델에 비하여 간단한 @(7,1,1) 모델을 최종적으로 선택하였다.

$$(1-0.169B^7)(1-B)Z_t = (1-0.876B)\epsilon_t\ ,$$
$$\epsilon_t \quad WN(0,(187600)^2)$$

<최종 모델>

3장 모델 평가

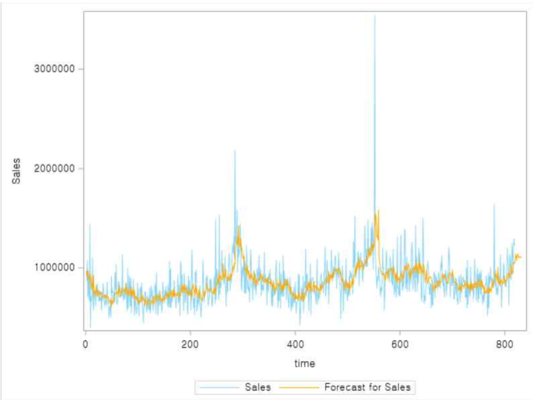


그림 <18> 실제 매출액과 예측값
실제값(하늘색), 예측값(주황색)

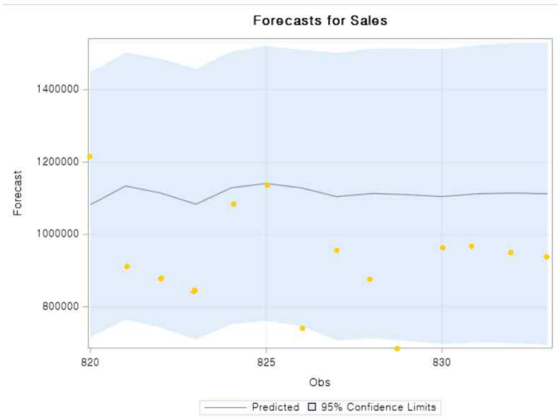


그림 <19> 실제매출액과 예측된 값
실제매출액(파란선), 예측값(주황색 점)

모델을 만들기 위해 사용하였던 시점들에 대하여 실제 매출액과 모델을 통해 예측한 값을 시각화 해보았다. 그림 <18>을 보면 주황색선이 실제 매출액인 하늘색의 값들을 잘 표현하고 있는 것을 확인할 수 있었다

이제 이 모델을 실제 매출액을 예측하는데 사용해보았다. 앞서 말한대로 2020년 1월 6일~ 1월 19일 까지 약 14일의 데이터를 예측해보았다. 그림 <19>를 보면 파란선으로 예측 값을 주황색 점으로 실제데이터를 표시하였다. 예측 결과 3가지의 문제점을 확인할 수 있었다.

첫째, 예측결과 실제 값들보다 다소 높게 예측하는 경향이 있다는 것을 확인할 수 있었다. 두 번째는 분산이 너무 크기 때문에 예측구간의 신뢰구간이 지나치게 넓다는 것이다. 마지막으로 가장 큰 문제는 예측구간에서 t의 값이 커질수록 점점 기울기 0의 직선형태를 가지는 것을 확인할 수 있었다. 모델이 7일전의 매출액과 하루 전 시점의 잔차를 사용하기 때문에 시간이 지나면서 기울기가 0에 가까운 값들을 이용하여

예측을 하게 되고 최종적으로는 결국에 하나의 값만을 계속 예측값으로 사용하는 모델이 나올 것이 분명하였다. 따라서 변동을 전혀 잡지 못하게 되므로 모델의 성능은 좋다고 말할 수 없었다.

Ⅲ.결론

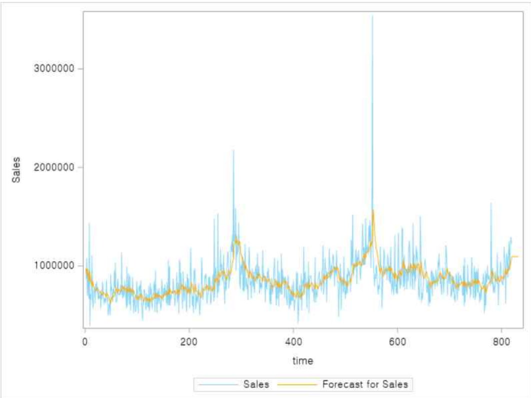


그림 <20>ARIMA(0,1,1)의 예측된 값

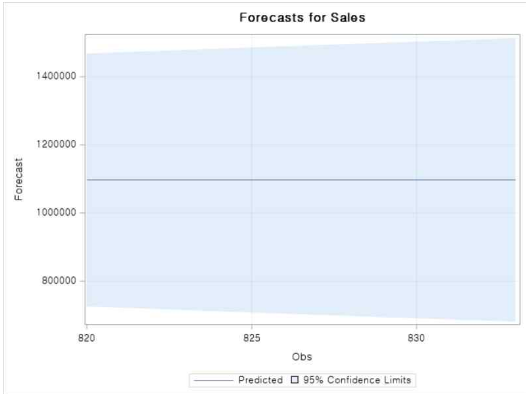


그림 <21>ARIMA(0,1,1)의 예측된 값

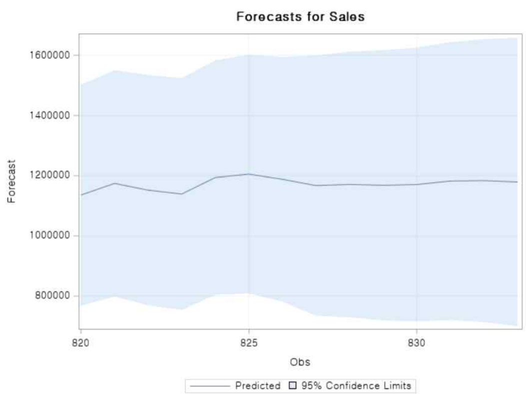


그림 <22> ARIMA(6,1,0)의 예측된 값

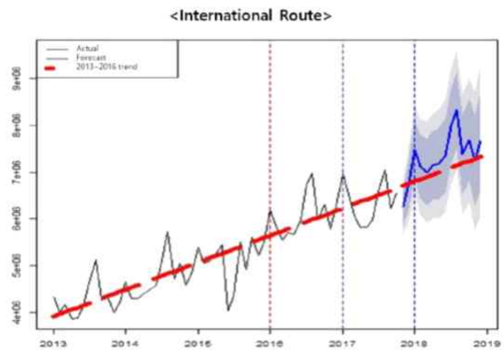


그림 <23> ARIMA를 통해 예측한 국제선여객수요(예시자료)

시각화를 통하여 보았을 때 요일의 영향성은 확인할 수 있었지만 계절성은 확인할 수 없었다. 하지만 최종적으로 7일 전의 값을 반영하는 모델이 최고의 결과를 보여주었다. 이를 어떻게 설명할 수 있을까? 결론부터 말을 하자면 “계절성은 존재하지 않지만 요일 수준을 고려한 것이 모델 성능에 작은 도움이 된다.” 정도로 해석할 수 있다. 그 이유로 최종모델에서 Z_{t-7}부분을 제거한 ARIMA(0,1,1)의 AIC 값이 22205이고 최종모델의 AIC 값이 22184로써 Z_{t-7}를 모델에 반영한 최종모델 이 유의미한 결

과를 가져온 것은 사실이지만 21정도 차이가 나기 때문에 아주 큰 차이라고 말할 수는 없기 때문이다. 즉, 모델이 잘 적합할 수 있었던 주 효과는 ϵ_{t-1} 에서 오는 것 이라고 말할 수 있다.

따라서 가게의 인원배치에 있어서 계절성은 존재하지 않기 때문에 인원을 매주 고정된 요일에 맞춰 배치하는 것은 좋지 않다. 하지만 현실적으로 직원들의 시간을 변칙적으로 조정할 수는 없으므로 매출이 다른 요일에 비하여 대부분 많았던 토, 일, 월에 3명을 배치하는 것이 좋다고 결론내릴 수 있다.

예측성능이 좋지 않았던 부분의 경우 이를 개선하기 위하여 두가지 방법을 생각하였다. 첫 번째로 기온, 날씨 등의 데이터를 추가적으로 반영하는 것이 좋다고 생각하였다. 그림 <1>에서 추운 겨울에 죽의 판매량이 증가하는 것을 확인 할 수 있었다. 이러한 점을 반영하여 기온, 날씨 등의 데이터를 추가적으로 사용한 다변량 시계열 모델을 만들면 예측 성능이 올라갈 수 있을 것이다.

두 번째는 ARIMA 이외의 모델을 사용하는 것 이다. 그림<18>에서 기존 데이터에 대하여 모델이 잘 적합하지만 그림 <19>의 테스트 데이터에 대하여 예측성능이 크게 떨어지는 것을 확인 할 수 있다. 모델이 과적합이 되었다고 말할 수 있지만 사실 이는 ARIMA를 통해 예측을 할 때 항상 떠오르는 문제 중 하나이다. 그림 <20>을 보면 @(7,1,1)과 비교하여 모수를 1개 적게 사용한 ARIMA(0,1,1)을 통해 기존 데이터에 대하여 모델을 적합하였을 때 적합이 잘 된 것을 확인할 수 있다. 하지만 그림<21>을 보면 새로운 구간에 대하여 예측을 할 경우 예측 결과가 하나의 값만을 예측함으로써 매우 안 좋은 것을 확인할 수 있다. 또한 그림 <22>를 보면 ARIMA(6,1,0)을 통해 예측을 한 결과로써 역시 예측결과가 매우 좋지 않은 것을 확인 할 수 있었다. 이는 모수의 많고 적음의 문제가 아니라고 생각하였다. ARIMA 모델의 경우 그림 <21>처럼 비교적 계절성이 뚜렷하거나 추세가 확실한 경우 좋은 성능을 보여주지만 보고서에서 사용한 데이터와 같이 무작위성이 심하고 추세를 정확히 파악하기 어려운 데이터의 경우 예측성능이 크게 나빠지게 된다. 이러한 ARIMA의 좋지 못한 예측 성능에 대하여 많은 보고서에서 매우 많이 발견할 수 있었다. 따라서 예측 성능을 올리기 위해서는 ARIMA만이 아닌 다른 모델을 추가적으로 탐색해보고 사용해 볼 것을 제안한다.

부록

1. 사용코드

데이터부르기

```
data Bon;
infile '/folders/myfolders/all.csv' dlm=',' Firstobs=2; /*구분자 : 띄어쓰기*/
input Year Month Date Sales weekend dow$ times;
run;
```

그림 <1>만 R사용

```
.libPaths('C:/Library')
install.packages('forecast')
library(forecast)
all <- read.csv('C:/Users/todtj/OneDrive/바탕 화면/기말보고서/all.csv')
a <- ts(all$X4[1:91],frequency=7)
b <- decompose(a,type=c('additive'))
trend <- trendcycle(b)
ts.plot(a,trend,ylab='Sales',col=c('blue','RED'))
```

ANOVA

```
proc anova data= Bon;
Class dow;
model Sales=dow;
means dow;
run;
```

ACF,PACF 그래프 확인

```
PROC ARIMA DATA=Bon;
IDENTIFY VAR=Sales NLAG=24;
RUN;
```

모델별 AIC 검증

```
/* ARIMA(0,1,1) */
proc arima data= Bon_DIFF;
identify var=Sales(1) nlag=24;
estimate p=(0) q=(1) plot method=ML noconstant;
run;

/* ARIMA(0,1,2) */
proc arima data= Bon_DIFF;
identify var=Sales(1) nlag=24;
estimate p=(0) q=2 plot method=ML noconstant;
run;

/* ARIMA(1,1,1) */
```

```
proc arima data= Bon_DIFF;
identify var=Sales(1) nlag=24;
estimate p=1 q=1 plot method=ML noconstant;
run;


/* ARIMA(6,1,0) */
proc arima data= Bon_DIFF;
identify var=Sales(1) nlag=24;
estimate p=6 q=(0) plot method=ML noconstant;
run;

/* ARIMA(6,1,1) */
proc arima data= Bon_DIFF;
identify var=Sales(1) nlag=24;
estimate p=6 q=(0) plot method=ML noconstant;
run;

/* ARIMA(6,1,1) */
proc arima data= Bon_DIFF;
identify var=Sales(1) nlag=24;
estimate p=6 q=1 plot method=ML noconstant;
run;

/* ARIMA(7,1,0) */
proc arima data= Bon_DIFF;
identify var=Sales(1) nlag=24;
estimate p=7 q=0 plot method=ML noconstant;
run;

/* ARIMA(7,1,1) */
proc arima data= Bon_DIFF;
identify var=Sales(1) nlag=24;
estimate p=7 q=1 plot method=ML noconstant;
run;

/* @(7,1,1)*/
proc arima data= Bon_DIFF;
identify var=Sales(1) nlag=24;
estimate p=(7) q=1 plot method=ML noconstant;
run;
```

최종선택된 @(7,1,1) 과 예측하기

```
/* ARIMA(7,1,1) */
proc arima data= Bon;
identify var=Sales(1) nlag=24;
estimate p=(7) q=1 plot method=ML noconstant;
forecast lead=14 out=fore;
run;

# 기존데이터에 대한 적합값 구하기

data fore;
set fore;
time=_n_;
run;

proc sgplot;
series x=time y= Sales/lineattrs=(color=skyblue);
series x= time y=forecast/ lineattrs=(color=orange);
run;

# 0,1,1 에대한 적합선
/* ARIMA(0,1,1) */
proc arima data= Bon;
identify var=Sales(1) nlag=24;
estimate p=0 q=(1) plot method=ML noconstant;
forecast lead=14 out=fore;
run;

data fore;
set fore;
time=_n_;
run;
proc sgplot;
series x=time y= Sales/lineattrs=(color=skyblue);
series x= time y=forecast/ lineattrs=(color=orange);
run;
```