预测 2022 年社会消费品零售总额的损失-基于季节性 ARIMA 模型

摘　要

社会消费品零售总额是衡量人们消费水平的重要指标，也是国民经济体系中的一个重要指标。预测 2022 年疫情损失的社会消费品零售总额，可以将其应用于的对整体经济损失的一个关键要素。

首先通过对以往月度统计数据进行 STL 季节性分解，得到时间序列的季节波动情况。然后先对原时间序列进行一阶差分，再对差分后序列进行季节差分，得到平稳的时间序列。再通过季节性 ARIMA 模型进行参数估计，并得到 2022 年未受到疫情干预的预测值。接着利用 2020 年前的数据建立 ARIMA 模型，得到 2020 年未受疫情影响的预测值。通过 2020 年预测值和实际值，建立在疫情干预下的干预分析模型。继而应用该模型得到 2022 年在疫情干预下的预测值。最终结合 ARIMA 模型的预测值，预测 2022 年社会消费品零售总额的损失。并通过 R 语言进行建模和计算以及绘图。

关键词: 社会消费品零售总额，季节性 ARIMA 模型，时间序列分析，干预分析模型

目 录

一、 模型假设

1. 模型假设一：2022 年 3 月起疫情对社会消费品零售总额产生干预
2. 模型假设二：2022 年疫情对经济的干预情况与 2020 年相同
3. 模型假设三：2022 年四月为损失最严重的一个月，五月起逐步恢复

二、 主要符号与说明

序号	符号	符号说明
1	X_t	社会消费品零售总额的时间序列
2	T_t	趋势因子时间序列
3	S_t	季节因子时间序列
4	R_t	随机误差因子时间序列
5	$\hat{X}_t$	对 X_t 一阶差分后的时间序列
6	ϵ_t	白噪声序列
7	p	自回归阶数
8	q	移动平均阶数
9	X'	对 $\hat{X}_t$ 季节差分后的时间序列
10	B	滞后算子
11	lag	滞后阶数
12	ω	疫情干预系数
13	Z_t	疫情发生后产生损失的时间序列
...	...	

三、 数据分析

3.1 数据来源

本次建模采用的数据为 2012 年 1 月起至 2022 年 4 月的月度数据，来自国家统计局。由于在 1-2 月只有总和的数据，在此简单假设两月每天的零售总额相同，则一月份总额在 1-2 月总和的占比在闰年和平年的占比分别为 51.7% , 52.5% ，得到表格如下：

表 3-1: 社会商品零售总额月度数据 (单位: 亿元)

	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
1	17171.0	19866.4	21563.2	25216.7	26984.3	30453.8	31151.7	34712.0	27390.6	36641.8	37957.3
2	16497.6	17943.4	20717.5	22775.8	25926.0	27505.9	29930.1	31352.0	24739.2	33095.0	36468.7
3	15650.2	17641.2	19800.6	22722.8	25114.1	27863.7	29193.6	31725.7	26449.9	35484.1	34233.1
4	15603.1	17600.3	19701.2	22386.7	24645.8	27278.5	28541.9	30586.1	28177.8	33152.6	29483.1
5	16714.8	18886.3	21249.8	24194.8	26610.7	29459.2	30359.1	32955.7	31972.8	35945.1	
6	16584.9	18826.7	21166.4	24280.3	26857.4	29807.6	30841.6	33878.1	33525.9	37585.8	
7	16314.9	18513.2	20775.8	24338.8	26827.4	29609.8	30733.7	33073.3	32202.5	34925.1	
8	16658.9	18886.2	21133.9	24893.4	27539.6	30329.7	31542.3	33896.3	33570.6	34394.9	
9	18226.6	20653.3	23042.4	25270.6	27976.4	30870.3	32005.4	34494.9	35294.7	36833.0	
10	18933.8	21491.3	23967.2	28278.9	31119.2	34240.9	35534.4	38104.3	38576.5	40453.9	
11	18476.7	21011.9	23474.7	27937.3	30958.5	34108.2	35259.7	38093.8	39514.2	41043.2	
12	20334.2	23059.7	25801.3	28634.6	31757.0	34734.1	35893.5	38776.7	40566.0	41268.9	

根据表格中所给数据，由于 2022 年三月份时吉林，上海等各省市集中爆发疫情，因此选择 2022 年 2 月份以前的数据作为训练集，以时间为横轴，绘出折线图如下:

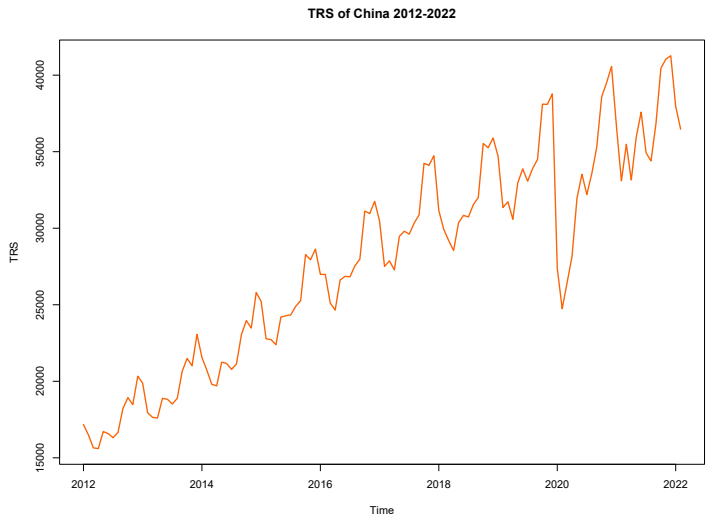

图 3-1: 社会消费品零售总额 2012-2022

3.2 季节性分解

可以看出社会消费品零售总额的变化呈明显的季节波动，并且呈现逐年上升趋势，所以考虑通过 STL 算法用 Loess 平滑化后将时间序列数据分解为趋势因子 (trend compoents)T_t，季节因子 (season compoents)S_t，和随机误差因子 (remmainder compoents)R_t[1]:

$$X_t = T_t + S_t + R_t \tag{1}$$

得到分解图如下:

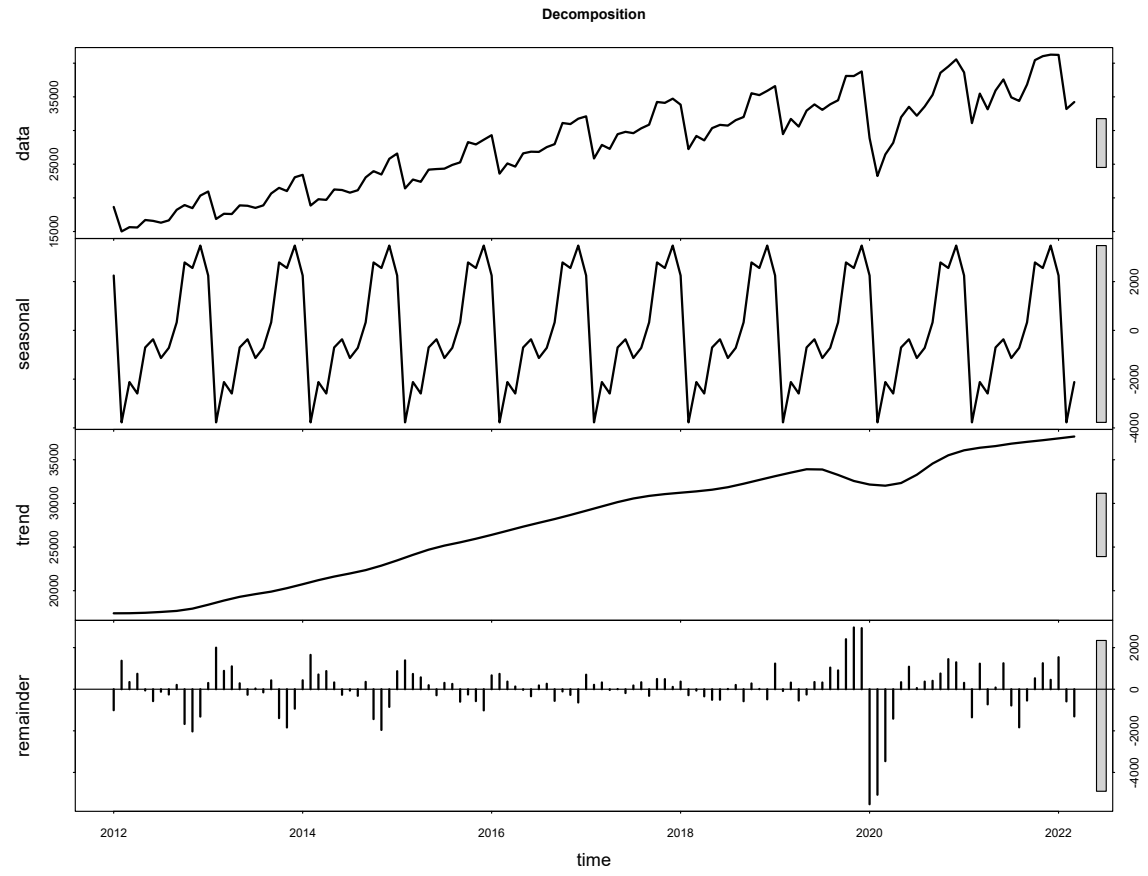

图 3-2: 按加法模型分解后

从趋势因子来看，总体趋势稳步提升直至 2020 疫情爆发前，到 2020 年末基本恢复正常发展水平，持续稳步发展到 2022 年 2 月。

从季节因子来看，在春节前，即年末 12 月份，消费达到顶峰，之后逐步下降至四月份到达最低点，之后开始逐步攀升直至 12 月份。

四、 ARIMA 模型建立

4.1 数据检验

数据平稳性检验

若时间序列 X_t 满足如下条件:

(1) 均值 $E(X_t) = \mu$, 均值 μ 是与时间 t 无关的常数

(2) 方差 $Var(X_t) = \sigma^2$, 方差 σ 是与时间 t 无关的常数

(3) 协方差 $Cov(X_t, X_{t+k}) = \gamma^2$, 协方差只与间隔 t 有关

则称时间序列 X_t 是平稳的。

由表3-1中可明显看出均值随时间 t 增长，可以猜测原序列应为非平稳序列。采用 ADF 检验原序列的平稳性，ADF 检验通过一下三个模型检验:

$$\begin{aligned}\Delta X_t &= \delta X_{t-1} + \sum_{i=1}^{m} \beta_i X_{t-i} + \epsilon_t \\ \Delta X_t &= \alpha + \delta X_{t-1} + \sum_{i=1}^{m} \beta_i X_{t-i} + \epsilon_t \\ \Delta X_t &= \alpha + \beta_t + \delta X_{t-1} + \sum_{i=1}^{m} \beta_i X_{t-i} + \epsilon_t\end{aligned} \tag{2}$$

三个模型原假设都是 $H_0 : \delta = 0$. 若拒绝 H_0 则为平稳序列，否则为非平稳序列。通过 ADF 临界值表判断是否接受 H_0

为验证猜想对原序列做 ADF 检验, 得到结果如下:

表 4-2: Add caption

Augmented Dickey-Fuller Test	
Lag Order:	1
Dickey-Fuller:	0.3394
P Value	0.7218

由于 $p - value > 0.05$ 所以无法拒绝原假设, 因此原序列是非平稳的。为了将原序列转化为平稳序列处理，因为从图3-1看出原序列应该有随时间线性增加的趋势，考虑对原序列做一阶差分[4]，得到新序列 $\hat{X}_t$ 如下图:

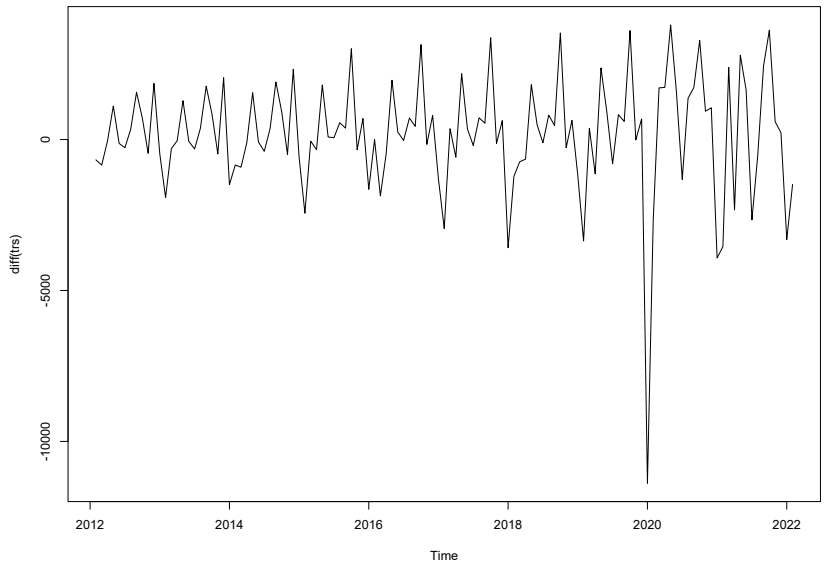

图 4-3: 对原序列做一阶差分后

在对差分后的序列做 ADF 检验：

表 4-3: ADF 检验

Augmented Dickey-Fuller Test	
Lag Order:	1
Dickey-Fuller:	-7.5267
P Value	0.01

由于 $p < 0.05$ 所以拒绝原假设，差分后的序列是平稳的，即通过一阶差分去掉了原序列线性的趋势因子。

随机性检验

尽管 $\hat{X}_t$ 为平稳序列，但是如白噪声等纯随机序列也是平稳序列，若 $\hat{X}_t$ 是纯随机序列，则没有建模研究价值价值，于是采用 $Ljung-Box$ 检验随机性

假设 $H_0 : \rho_1 = \rho_2 = \cdots = \rho_n = 0$ 则对所有的 $k > 0$, 样本的自相关系数服从:

$$\hat{\rho}_k \approx N(0, \frac{1}{n}) \tag{3}$$

其中 n 为样本量, 通过检验统计量：

$$Q_{LB}(m) = n(n+2) \sum_{k=1}^{m} \frac{{\hat{\rho}_k}^2}{n-k} \chi^2(m) \tag{4}$$

得到的 $Ljung-Box$ 检验结果为：

表 4-4: $Ljung-Box$ 检验

Ljung-Box test	
X-squared	494.39
df	6
p-value	< 2.2e-16

由于 $p < 0.05$ 所以拒绝原假设，则 $\hat{X}_t$ 为非随机序列，可进行下一步建模。

4.2 无季节性的 ARIMA 模型建立

给定一个差分 d 阶的时间序列 y_t，$ARIMA(p, d, q)$ 模型如下：

$$y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t \tag{5}$$

或者写为[4]

$$(1-\phi_1 B-\cdots-\phi_p B^p)(1-B)^d y_t = c+(1+\theta_1 B+\cdots+\theta_q B^q)\varepsilon_t$$

其中 ε_t 是白噪声序列, p 是自回归的阶数，q 是移动平均的阶数。

平稳序列的自相关函数 ACF 与时间间隔 k 有关, 并通过 ACF 相关系数决定 q:

$$\rho_h = \rho(y_t, y_{t+k}) = \frac{Cov(y_t, y_{t+k})}{\sigma_t \sigma_{t+k}} \tag{6}$$

ACF 图显示了 y_t 与 y_{t-k} 之间相关性，但是滞后阶数 $1, 2, \cdots, k-1$ 之间存在依赖关系[4]，例如若 y_t 与 y_{t-1} 自相关，那么 y_t 与 y_{t-2} 一定自相关，因为他们都通过与 y_{t-1} 直接相关，而间接相关，为了分离 $y_{t-1}, y_{t-2}, \cdots, y_{t-k+1}$ 的干扰，直接得到 y_t 与 y_{t-k} 之间的相关性，通过 $PACF$ 估计 P 值，由于历史白噪声 ε_{t-k} 通过影响历史观测值来间接影响当前 y_t 所以用 ACF 估计 q 值，绘出一阶差分后 $\hat{X}_t$ 的 ACF 和 $PACF$ 图（临界值 $\frac{\pm 1.96}{T}$ 以用虚线标出):

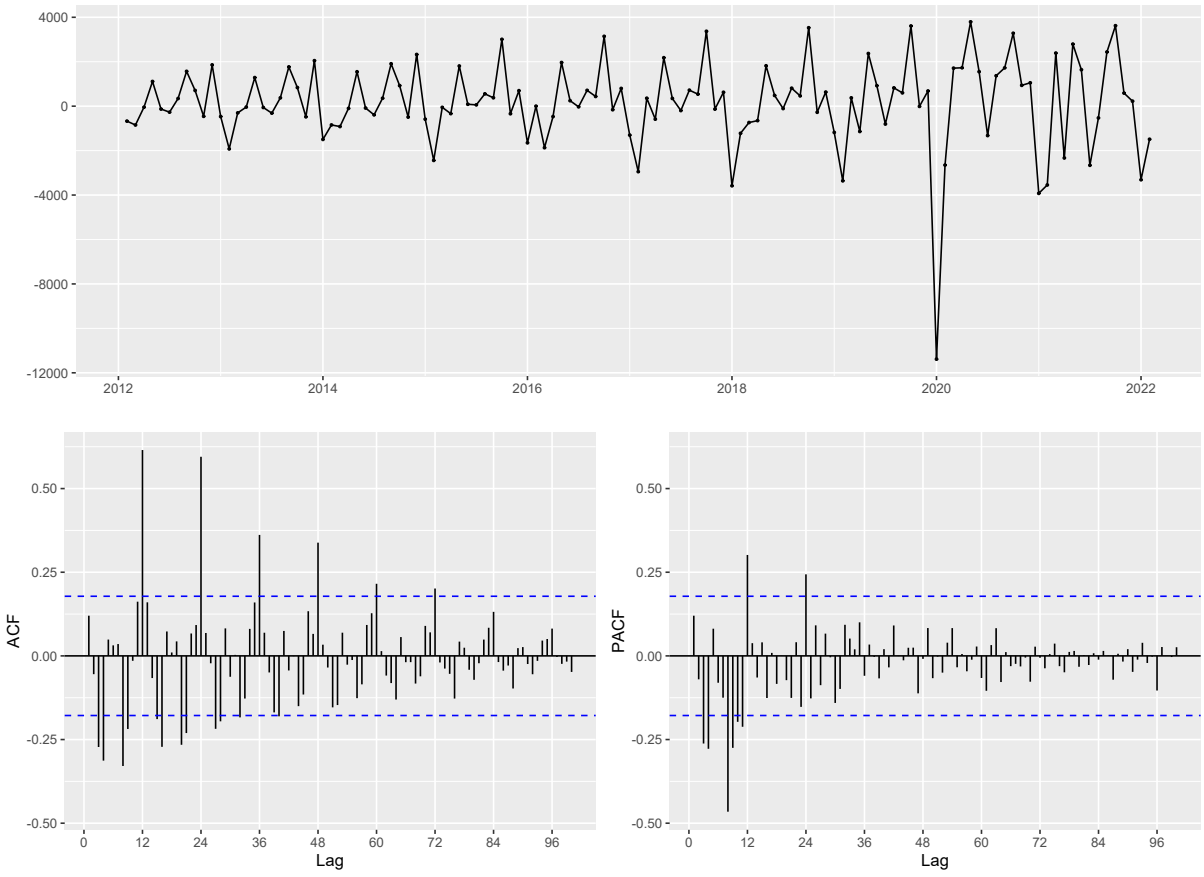

图 4-4: ACF 和 PACF

4.3 季节性的 ARIMA 模型

通过图4-4发现在滞后阶数 lag 值很高时才出现拖尾，会导致参数过多发生过拟合现象，而且通过图3-2得到消费总额应该是呈现明显季节波动所以考虑将一阶差分序列 $\hat{X}_t$ 分解为季节部分和剩下的非季节部分[4]:

$$\text{ARIMA} \quad (p,d,q) \qquad (P,D,Q)_m$$

其中 $m=12$ 为观测周期，可以将模型写成季节部分与非季节部分的乘积，例如对于 $ARIMA(1,1,1)(1,1,1)_m$ 模型:

$$(1-\phi_1 B)\,(1-\Phi_1 B^{12})\,(1-B)\,(1-B^4)y_t = (1+\theta_1 B)\,(1+\Theta_1 B^4)\varepsilon_t \tag{7}$$

4.4 参数估计

为先消除季节型波动，对一阶差分后 $\hat{X}_t$ 做季节性差分，得到 $X'_t = \hat{X}_t - \hat{X}_{t-12}$，绘出相关图像:

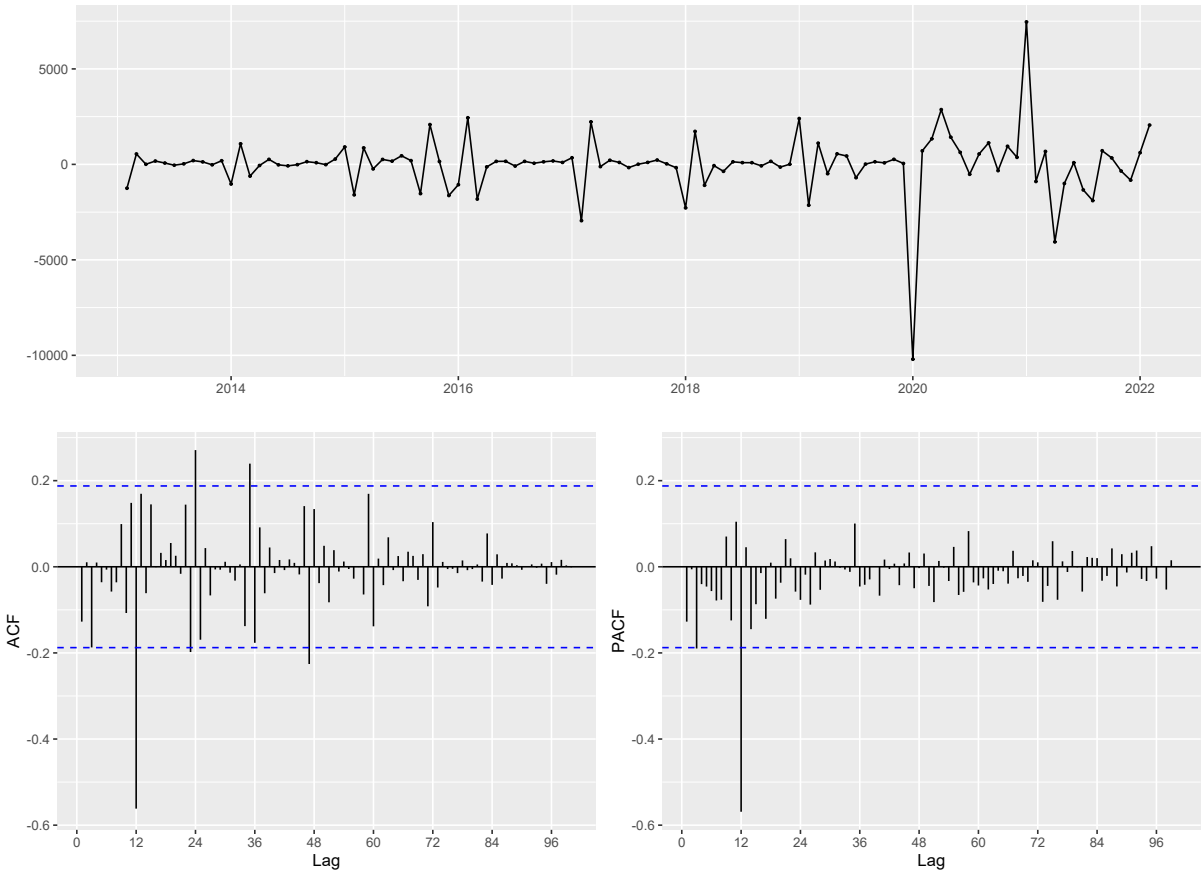

图 4-5: 季节差分后 ACF 和 PACF

4.5 模型的选择

从相关系数图看出相比于图4-4, 相关系数的的衰减速度加快很多，对于季节部分的 ACF 和 PACF 图 (即 $Lag = 12n$，n 对应季节部分的滞后系数)，都在 lag=12 后产生拖尾。因此可确定4.3中的季节部分相应系数 $P = 1, Q = 1$

但对于非季节部分的 ACF 和 PACF 图较难判断在何种滞后系数后产生拖尾，因此利用 AIC,AICc,BIC 准则定量的确定在何种系数下的模型最优:

$$\text{AIC} = -2log(L) + 2(p + q + k + 1)$$

其中 L 是似然数据的似然函数，最后一项为参数个数 (包含了余项的方差)$k = 0$ 若 $c = 0$,$k = 1$ 若 $c \neq 0$ 对于 ARIMA 模型而言，修正过的 AIC 值可以被表示为[4]:

$$\text{AICc} = \text{AIC} + \frac{2(p + q + k + 1)(p + q + k + 2)}{T - p - q - k - 2}$$

并且贝叶斯信息准则 (BIC) 如下:

$$\text{BIC} = \text{AIC} + [\log(T) - 2](p + q + k + 1)$$

通过枚举 p,q 的值得到相应模型 AIC,AICc,BIC 如下:

表 4-5: Add caption

相应的 ARIMA 模型	AIC	AICc	BIC
(0,1,0)(1,1,1)[12]	1876.32	1876.55	1884.4
(0,1,1)(1,1,1)[12]	1878.32	1878.7	1889.08
(0,1,2)(1,1,1)[12]	1877.96	1878.54	1891.41
(0,1,3)(1,1,1)[12]	1876.22	1877.04	1892.36
(1,1,1)(1,1,1)[12]	1880.27	1880.86	1893.73
(1,1,2)(1,1,1)[12]	1873.59	1874.41	1889.74
(1,1,3)(1,1,1)[12]	1875.51	1876.62	1894.35
(2,1,1)(1,1,1)[12]	1873.53	1874.36	1889.68
(2,1,2)(1,1,1)[12]	1875.02	1876.13	1893.86
(3,1,0)(1,1,1)[12]	1879.02	1879.84	1895.16
(3,1,1)(1,1,1)[12]	1875.53	1876.64	1894.37
(3,1,2)(1,1,1)[12]	1876.98	1878.79	1901.2

从表4-5中看出，$ARIMA(2,1,1)(1,1,1)_{12}$ 是最优的 ARIMA 模型。

4.6 残差检验

为说明残差纯随机变量，对残差 ϵ_t 做 Ljung-Box test 检验:

表 4-6: 残差 $Ljung-Box$ 检验结果

Ljung-Box test	
df	19
p-value	0.90

$p > 0.05$ 无法拒绝原假设，所得残差为白噪声序列, 残差之间不存在自相关性。并且得到的残差图4-6, 残差基本符合正态分布要求:

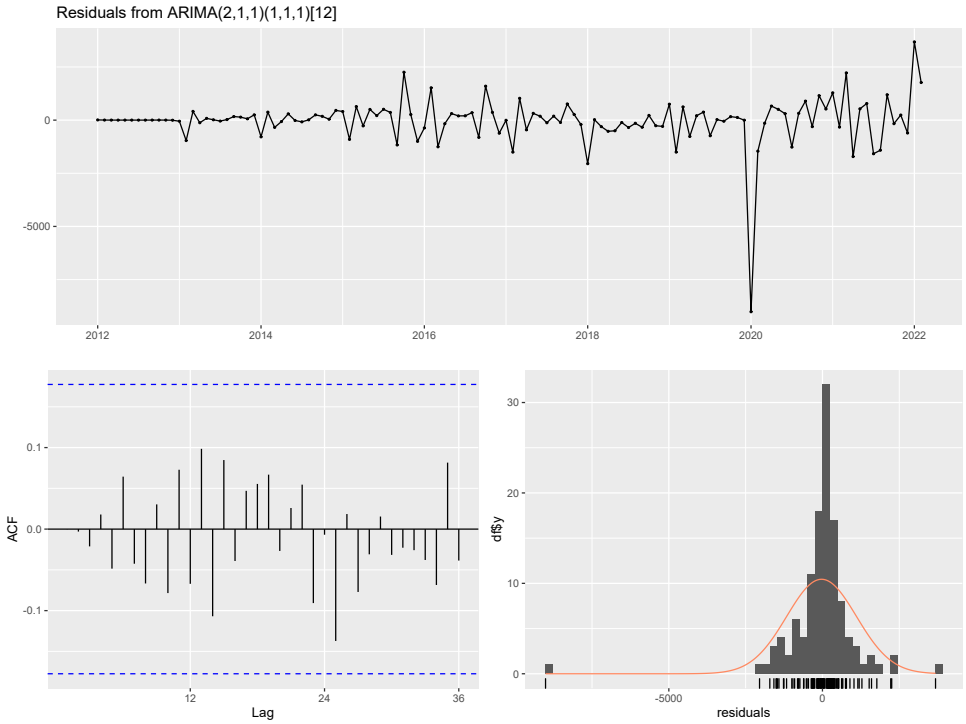

图 4-6: $ARIMA(2,1,1)(1,1,1)_{12}$ 的残差图

为进一步说明，绘出正态 Q-Q 图4-7, 所以残差符合正态分布要求:

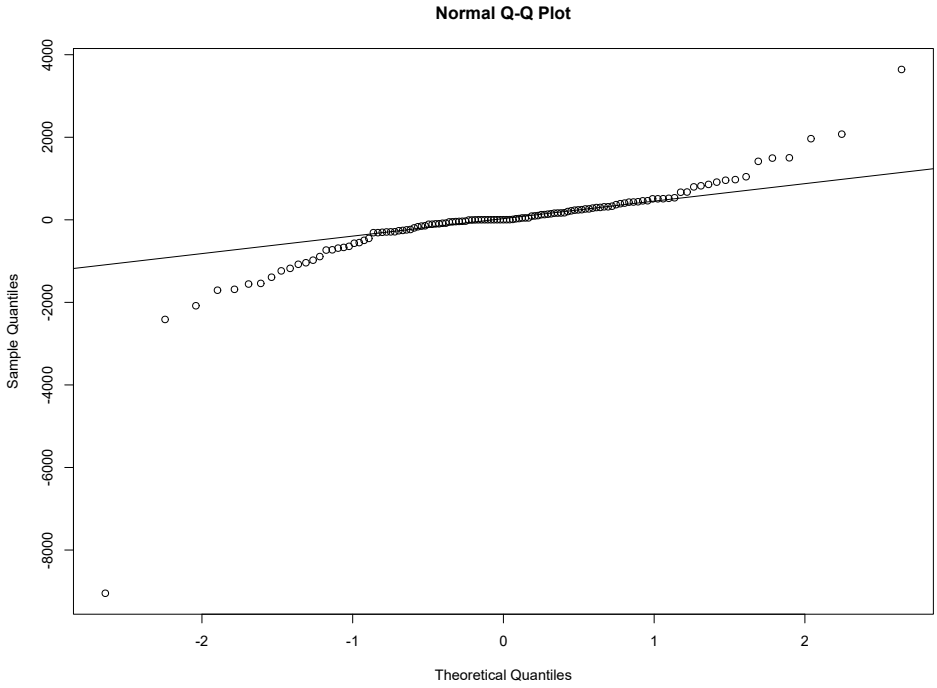

图 4-7: $ARIMA(2,1,1)(1,1,1)_{12}$ 的残差 Q-Q 图

4.7 未受疫情干预的预测

选取的训练集为表3-1中 2022 年 2 月以前 (包括二月). 用得到的 ARIMA 模型对训练集进行拟合, 以 12 个月划分序列。通过所有之前的数据拟合当前年份的数据 (起始两年除外), 拟合得到的 12 步拟合结果图4-8:

图 4-8: $ARIMA$ 模型得到的 12 步拟合值

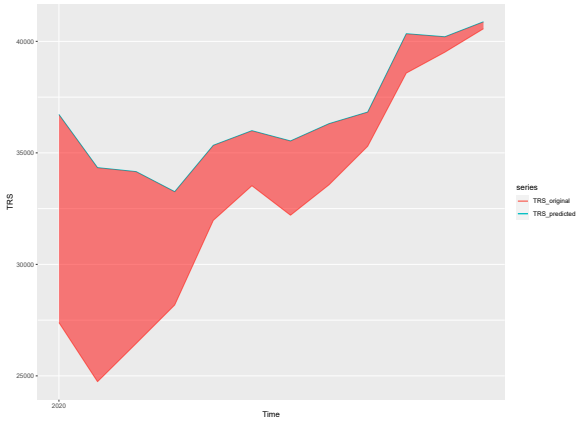

图 4-9: 2020 年社会消费品零售总额的损失

从图像上看, 除去 2020 年初有所偏差外, 其余部分都能较好拟合。由此也能从图中得出 2020 年疫情带来的社会消费品零售总额的损失为 47934.05（亿元）, 为图4-9中阴影部分。最终用此模型预测自 2022 年 3 月起 10 个月的预测结果如下, 并给出 80%和95%

的置信区间。

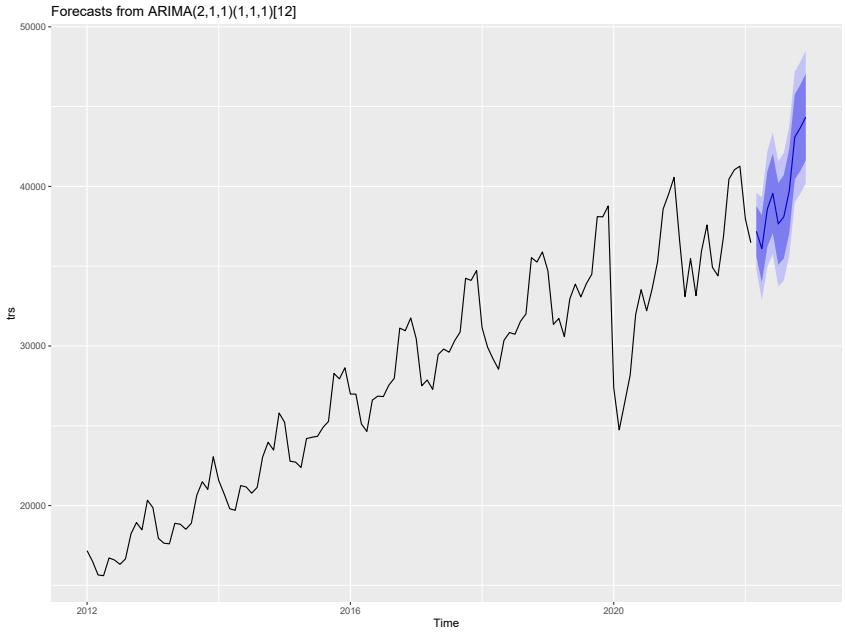

图 4-10: $ARIMA(2,1,1)(1,1,1)_{12}$ 模型对从 3 月起至 2022 年 12 月的预测值

五、 干预分析模型的建立

5.1 模型建立

假设疫情对经济的影响是突然开始，并且持续的，对持续性干预变量

$$S_t^T = \begin{cases} 0 & \text{疫情发生前 t<T} \\ 1 & \text{疫情发生后 t>=T} \end{cases}$$

设 ω 为干预未知的干预系数，Z_t 为疫情发生后所产生的损失的时间序列，通过一阶差分获得平稳序列，则干预后的模型可写为

$$Z_t = \frac{\omega S_t^T}{\delta Z_{t-1}} \quad 0 < \delta < 1 \tag{8}$$

经过变换，实际上为 1 阶自回归模型[5]

$$Z_t = \delta Z_{t-1} + \omega$$

通过 2020 年的损失的社会消费品零售总额的数据4-9，将疫情对经济的冲击分为两个阶段，先是损失逐步扩大的过程，后是经济逐步恢复的时期。只考虑当损失逐渐恢复的一段，即从 2020 年 2 月至 12 月的损失。用最小二乘法的到参数的估计值，$\delta = 0.7328, \omega = 72.1654$

5.2 残差检验

绘出回归拟合图像和残差图如下, 从残差分布曲线看，残差基本符合正态分布要求, 且通过 Ljung-Box 检验。:

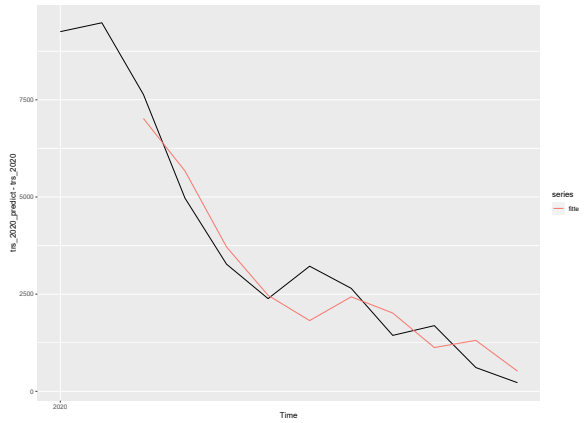

图 5-11: 2020 年社会消费品零售总额的损失图 (红色为回归结果)

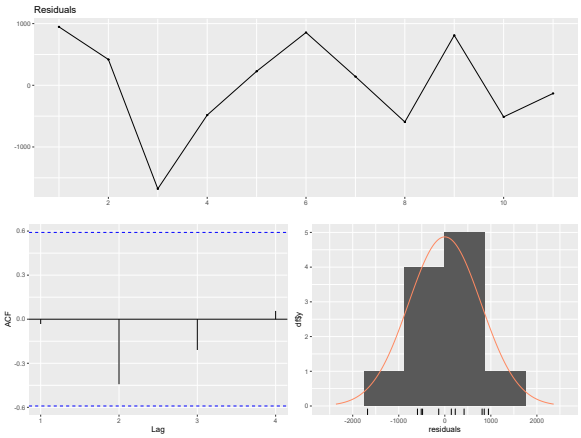

图 5-12: 回归结果的残差图

5.3 预测 2022 年的损失

假设 5 月份经济损失有望得到改善，即四月份为损失最严重的时期，那么且疫情不在反弹。那么预测从今年三月份开始到年末的社会消费品零售总额损失如下:

表 5-7: 2022 年 3 月起社会消费品零售总额的损失（单位：亿元）

月份	3	4	5	6	7	8	9	10	11	12
损失	3924	8587	6364	4735	3542	2667	2026	1557	1213	961

损失的社会消费品零售总额共计 35576.72 亿元损失，绘出曲线图如下:

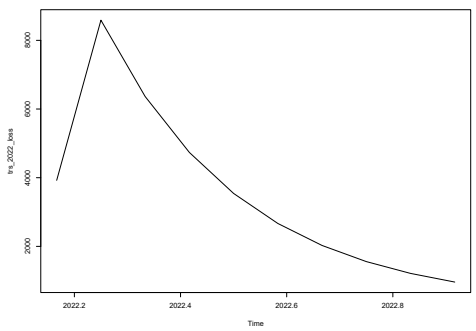

图 5-13: 2022 年 3 月起社会消费品零售总额的损失

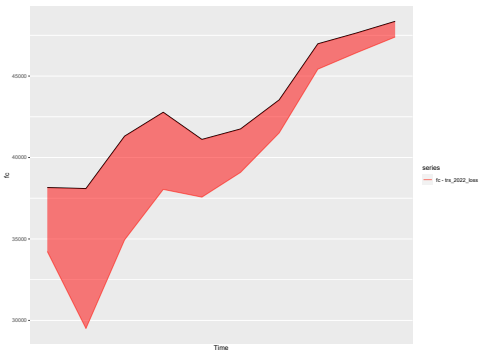

图 5-14: 红色面积为预测损失

六、 模型的评价

6.1 优点

（1）使用的模型简单，只需通过自身的变量进行建模，不需要依靠过多外部变量就能完成建模，减少数据收集的难度

（2）先对数据进行 STL 季节性分解，能够对原数据的规律有较为直观的认识，发现数据同时具有线性递增和季节波动的特点。

（3）检验充分，对数据的平稳性和随机性检验，以及多次对模型的残差进行检验，说明模型比较可靠

（4）在确定 p, q 参数时考虑全面，通过网格化枚举的方式，利用 AIC, AICc, BIC 准则选取最优的模型，一定程度上避免了直接通过 ACF 和 PACF 图主观选取参数。

6.2 缺点

（1）没有充分的理由假设 2020 年和 2022 年疫情对经济的干预相同，因此，直接利用通过 2020 年的干预分析模型预测 2022 年的经济损失不具有充分说服力。

（2）ARIMA 模型预测的误差会随时间的增加逐渐增大，所以对于 10 个月的预测具有一定的误差存在。

课程学习感悟

选择这门课程的主要目的是想了解数学是如何应用于解决生活中的实际问题的。数学建模的基本步骤中最重要的两点就是参数估计和假设检验两部分。接着课上介绍很多不同的模型，比如多元分析，线性规划，微分方程，还有图论相关的模型。课上比较强调的是线性回归模型。虽然在中学就已经接触最小二乘法求回归直线的问题，但是基于此却能衍生出很多的应用，比如机器学习中使用最多的方法之一就是线性回归。

另外通过作业的练习 R 语言的基本使用。体会到 R 语言的极大的便利性，首先开源、免费、轻量，平时要引进新的 package 也很方便，package 里面的函数功能也很强大，基本上几条命令就能搞定，比起其他语言很容易就能上手。

通过这次大作业也得到了充分的锻炼，从开始的选题，数据收集，编程，到最后写论文，都不断的学到了新的知识和技能。学到了很多时间序列分析相关的知识，熟悉时序分析相关的库函数，到最后使用 Latex 撰写论文, 最终得到令自己满意的结果。总之，学习这门课感觉收获颇丰，希望以后有机会能参加相关的比赛，进一步提高自己建模水平。

参考文献

[1] CLEVELAND R B, CLEVELAND W S, MCRAE J E, et al. Stl: A seasonal-trend decomposition[J]. J. Off. Stat, 1990, 6(1): 3-73.

[2] YANG L, WEI C, JIANG X, et al. Estimating the economic effects of the early covid-19 emergency response in cities using intracity travel intensity data[J]. International Journal of Disaster Risk Science, 2022, 13(1): 125-138.

[3] TSAY R S. An introduction to analysis of financial data with r[M]. [S.l.]: Wiley Publishing, 2012.

[4] HYNDMAN R J, ATHANASOPOULOS G. For further information on stationarity and differencing see[M]. [S.l.: s.n.], 2018.

[5] 杨新洪. 新冠肺炎疫情与“烟斗形”经济运行模式分析及模型测算——以广东省 2020 年前三季度经济走势为例 [J]. 深圳社会科学, 2021, 4(1): 18-39.

Forecast the loss of total retail sales of consumer goods in 2022 - Based on a seasonal ARIMA model

Abstract

The total retail sales of social consumer goods(TRS) is an important barometer to measure people's consumption level, and it is also an important barometer in the national economic. TRS lost in 2022 can be applied to be a key factor of the overall economic loss.

This article first decomposes the STL seasonal decomposition of the previous monthly statistics to obtain the seasonal fluctuations of the time sequence. After that make a first-order difference on the original time sequence, and then the seasons differential on the sequence of the differential sequence to obtain a smooth time sequence. Then perform parameter estimation through seasonal Arima models, and get the prediction value of without epidemic intervention in 2022. Then use the data before 2020 to establish the ARIMA model to obtain the predicted value that has not been affected by the epidemic in 2020. Utilize 2020 Forecast values and actual values to do intervention analysis modeling based on epidemic intervention. Then apply the model to get the predictive value in 2022 during pandemic intervention. Finally, the loss of TRS in 2022 can be forecasted by combining with the prediction value of the Arima model. Use the R language model and calculate and plot.

Keywords:total retail sales of social consumer goods ; seasonal ARIMA ; time series analysis ; intervention analysis

附 录

程序一：对原时间序列季节性分解以及数据检验:

```
library(readr)
library(forecast)
library(tseries)
setwd("E:/Code/code/mathmodeling/Final_Project/DATA")

# 数据读取
month_total <- read_csv("TRS_China_trainning.csv",
  locale = locale(encoding = "GBK")
)
t <- month_total[["社会消费品零售总额（亿元）"]]
trs <- ts(t, start = c(2012, 1), frequency = 12)

# 绘出折线图
plot(trs,
  main = "TRS of China 2012-2022",
  col = "#fb6305",
  xlab = "Time",
  ylab = "TRS",
  lwd = 2
)

#stl季节性分解
fit <- stl(trs, s.window = "period")
plot(fit, main = "Decomposition", lwd = 2)

# 原数据平稳性检验
print(adf.test(t))

# 一阶差分与季节差分
diff_trs <- diff(trs)
ggtsdisplay(diff_trs, lag.max = 100)
diff_season <- diff(diff_trs, lag = 12)
ggtsdisplay(diff(diff_trs, lag = 12), lag.max = 100)
```

程序二：ARIMA 模型的选取和预测:

```
library(readr)
```

```r
library(forecast)
library(urca)
library(knitr)
library(AICcmodavg)
library(ggplot2)

#数据读取
setwd("E:/Code/code/mathmodeling/Final_Project/DATA")
month_total <- read_csv("TRS_China_trainning.csv",
  locale = locale(encoding = "GBK")
)
t <- month_total[["社会消费品零售总额（亿元）"]]
trs <- ts(t, start = c(2012, 1), frequency = 12)

#方格搜索合适的 p ，q ,Q
models <- list(Arima(trs,
  order = c(1, 1, 1),
  seasonal = list(order = c(1, 1, 1), period = 12)
))
model_names <- rep("(p,1,q)(1,1,1)[12]", 30)
model_names[1] <- "(1,1,1)(1,1,1)[12]"
for (p in 0:5) {
  for (q in 0:5) {
    models <- append(models,list(Arima(trs,
      order = c(p, 1, q),
      seasonal = list(order = c(1, 1, 1), period = 12)
    )))
    model_names[length(models)] <-
      gsub("p", as.character(p), model_names[length(models)])
    model_names[length(models)] <-
      gsub("q", as.character(q), model_names[length(models)])
  }
}
print(models)

# 选取出最优的模型为ARIMA(2,1,1)(1,1,1)[12]
fit <- Arima(trs,
  order = c(2, 1, 1),
  seasonal = list(order = c(1, 1, 1), period = 12)
)

# 残差检验
checkresiduals(fit)
qqnorm(fit$residuals)
qqline(fit$residuals)
```

```
# 预测
print(autoplot(trs, series = "Training data") +
  autolayer(fitted(fit, h = 12),
    series = "12-step fitted values"
  ) +
  ylab("Training data") + xlab("Time") +
  theme(text = element_text(family = "STHeiti")))
print(autoplot(forecast(fit, h = 10)))
print(forecast(fit,h = 10))
```

程序三：干预分析模型的建立和预测:

```
library(readr)
library(forecast)
library(urca)
library(knitr)
library(AICcmodavg)
library(ggplot2)
library(tsfgrnn)
setwd("E:/Code/code/mathmodeling/Final_Project/DATA")
month_total <- read_csv("TRS_China_trainning.csv",
  locale = locale(encoding = "GBK")
)
t <- month_total[["社会消费品零售总额（亿元）"]]
trs <- ts(t, start = c(2012, 1), frequency = 12)
fit <- Arima(trs,
  order = c(2, 1, 1),
  seasonal = list(order = c(1, 1, 1), period = 12)
)
# 测算2020年的疫情损失
trs_2020 <- window(trs,
  start = c(2020, 1),
  end = c(2020, 12)
)
trs_2020_predict <- window(fitted(fit, h = 12),
  start = c(2020, 1),
  end = c(2020, 12)
)
print(trs_2020)
print(trs_2020_predict)
print(aggregate(trs_2020_predict-trs_2020))
print(autoplot(trs_2020, series = "TRS_original") +
```

```
  autolayer(trs_2020_predict,
    series = "TRS_predicted"
  ) + geom_ribbon(
    aes(ymin = trs_2020, ymax = trs_2020_predict), fill = "red", alpha = 0.5
  ) +
  ylab("TRS") + xlab("Time") +
  theme(text = element_text(family = "STHeiti")))
print(autoplot(trs_2020_predict - trs_2020) +
  ylab("损失的消费品零售总额"))

# 分析干预模型的参数估计
loss <- (trs_2020_predict - trs_2020)
x <- as.vector(loss)[2:11]
loss <- as.vector(loss)[3:12]
fit_loss <- lm(loss ~ 1 + x)
print(fit_loss)
print(fitted.values(fit_loss))
fitte <- ts(fitted.values(fit_loss),
  start = c(2020, 3), frequency = 12
)
print(autoplot(trs_2020_predict - trs_2020) + autolayer(fitte))

# 残差检验
print(checkresiduals(fit_loss))

# 用得到的模型对2022年的损失进行预测
loss_2022 <- c(1:10)
loss_2022[1] = 3924.08
loss_2022[2] = 8587.446
for (i in c(3:10)) {
  loss_2022[i] = 0.7327 * loss_2022[i - 1] + 72.1654
}


#预测结果
print(sum(trs_2022_loss))
trs_2022_loss <- ts(loss_2022, frequency = 12, start = c(2022, 3))
plot(trs_2022_loss)
print(trs_2022_loss)


#绘出2022年的图
month_total <- read_csv("TRS_China_trainning.csv",
  locale = locale(encoding = "GBK")
)
```

```r
t <- month_total[["社会消费品零售总额（亿元）"]]
trs <- ts(t, start = c(2012, 1), frequency = 12)
fit <- Arima(trs,
  order = c(2, 1, 1),
  seasonal = list(order = c(1, 1, 1), period = 12),

)
fc <- forecast(fit, h = 10)
print(fc)
fc <- ts(c(
  38157.18, 38093.26, 41309.77, 42776.93, 41109.13,
  41751.93, 43536.51, 46978.47, 47642.45, 48357.27
), start = c(2022, 3), frequency = 12)

print(autoplot(fc) + autolayer(fc - trs_2022_loss)
  + geom_ribbon(aes(ymin = fc - trs_2022_loss, ymax = fc),
    fill = "red", alpha = 0.5
  ))
```