Project

2025-04-28

```
library(forecast)

## Warning: 程辑包'forecast'是用 R 版本 4.3.3 来建造的

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

library(ggplot2)
library(tseries)

## Warning: 程辑包'tseries'是用 R 版本 4.3.3 来建造的

data <- read.csv("C://Users//Anna Liu//Desktop//Columbia//Actuarial//PM
//group//project_data(2)(2).csv")



str(data)

## 'data.frame':    672 obs. of  16 variables:
##  $ observation_date  : chr  "1970/1/1" "1970/2/1" "1970/3/1" "1970/4
/1" ...
##  $ UMCSENT_interp    : num  78.6 78.1 77.2 76.3 75.4 ...
##  $ hourly_earning    : num  3.31 3.33 3.36 3.36 3.37 3.4 3.41 3.44 3.
45 3.45 ...
##  $ BBKMGDP           : num  -1.01 1.91 -1.39 -1.6 2.52 ...
##  $ CPI               : num  37.8 38 38.2 38.5 38.6 38.8 39 39 39.2 3
9.4 ...
##  $ CPILFESL          : num  39.6 39.8 40.1 40.4 40.5 40.8 40.9 41.1
41.3 41.5 ...
##  $ discouraged       : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ FEDFUNDS          : num  8.98 8.98 7.76 8.1 7.95 7.61 7.21 6.62 6.
29 6.2 ...
##  $ num_losers        : int  1245 1403 1511 1602 1829 1805 1859 1944
2078 2233 ...
##  $ PCEPI             : num  19.4 19.5 19.5 19.6 19.7 ...
##  $ PCE               : num  629 634 632 636 642 ...
##  $ unemployment_level: int  3201 3453 3635 3797 3919 4071 4175 4256
4456 4591 ...
##  $ X                 : logi  NA NA NA NA NA NA ...
##  $ monthly_average   : num  101 104 110 110 109 ...
##  $ IC                : num  1103571 1034000 1190000 1422571 1356429
 ...
```

```
##  $ CC                : num  1370161 1421000 1492161 1675200 1828581
 ...
```

```
head(data)
```

```
##   observation_date UMCSENT_interp hourly_earning   BBKMGDP  CPI CPIL
FESL
## 1         1970/1/1       78.63333           3.31 -1.014119 37.8
39.6
## 2         1970/2/1       78.10000           3.33  1.906225 38.0
39.8
## 3         1970/3/1       77.20000           3.36 -1.392509 38.2
40.1
## 4         1970/4/1       76.30000           3.36 -1.604366 38.5
40.4
## 5         1970/5/1       75.40000           3.37  2.520264 38.6
40.5
## 6         1970/6/1       76.13333           3.40  5.741429 38.8
40.8
##   discouraged FEDFUNDS num_losers  PCEPI   PCE unemployment_level  X
## 1          NA     8.98       1245 19.377 628.7               3201 NA
## 2          NA     8.98       1403 19.454 634.0               3453 NA
## 3          NA     7.76       1511 19.502 632.3               3635 NA
## 4          NA     8.10       1602 19.601 636.0               3797 NA
## 5          NA     7.95       1829 19.663 642.4               3919 NA
## 6          NA     7.61       1805 19.714 646.3               4071 NA
##   monthly_average      IC      CC
## 1        100.9633 1103571 1370161
## 2        104.4704 1034000 1421000
## 3        109.5552 1190000 1492161
## 4        109.5705 1422571 1675200
## 5        108.5677 1356429 1828581
## 6        109.3033 1261714 1851167
```

```
data$observation_date <- as.Date(data$observation_date)


ic_ts <- ts(data$IC, start = c(1970, 1), frequency = 12)
cc_ts <- ts(data$CC, start = c(1970, 1), frequency = 12)

print(ic_ts)
```

```
##             Jan        Feb        Mar        Apr        May        J
un
## 1970  1103571.4  1034000.0  1190000.0  1422571.4  1356428.6  1261714.
3
## 1971  1319857.1  1136142.9  1303428.6  1227714.3  1298428.6  1258285.
7
## 1972  1195714.3  1085285.7  1162000.0  1132571.4  1183142.9  1242142.
9
```

```
## 1973  1012000.0   886285.7  1007428.6  1026857.1  1055857.1  1028285.
7
## 1974  1359857.1  1273285.7  1385285.7  1252714.3  1305285.7  1334142.
9
## 1975  2423714.3  2166857.1  2418428.6  2236142.9  2205428.6  2102857.
1
## 1976  1649142.9  1442714.3  1609428.6  1593857.1  1737142.9  1691000.
0
## 1977  1844714.3  1744714.3  1587857.1  1587000.0  1676571.4  1539428.
6
## 1978  1558285.7  1530857.1  1529285.7  1392285.7  1437428.6  1418142.
9
## 1979  1577571.4  1438000.0  1587142.9  1762857.1  1531142.9  1560000.
0
## 1980  1833000.0  1734000.0  1951000.0  2313285.7  2752142.9  2593428.
6
## 1981  1830714.3  1715000.0  1808714.3  1757285.7  1848142.9  1868000.
0
## 1982  2401857.1  2135000.0  2476857.1  2511428.6  2599857.1  2566285.
7
## 1983  2190428.6  1953285.7  2130285.7  2129285.7  2068714.3  1881714.
3
## 1984  1550571.4  1408714.3  1530714.3  1581285.7  1590285.7  1553428.
6
## 1985  1653285.7  1558428.6  1694714.3  1702285.7  1740714.3  1681571.
4
## 1986  1623857.1  1487428.6  1693428.6  1658142.9  1706000.0  1588000.
0
## 1987  1543428.6  1395000.0  1474428.6  1439285.7  1466142.9  1381857.
1
## 1988  1498428.6  1317285.7  1355000.0  1340285.7  1387857.1  1318000.
0
## 1989  1282428.6  1230857.1  1425571.4  1327857.1  1438000.0  1445428.
6
## 1990  1603571.4  1423285.7  1525428.6  1548142.9  1582714.3  1549142.
9
## 1991  1985428.6  1939571.4  2209000.0  1997714.3  1965000.0  1833000.
0
## 1992  1948000.0  1833857.1  1894571.4  1804285.7  1838571.4  1790000.
0
## 1993  1530000.0  1358000.0  1562000.0  1504428.6  1522000.0  1463857.
1
## 1994  1606857.1  1390285.7  1487857.1  1475714.3  1583857.1  1450714.
3
## 1995  1472857.1  1357428.6  1502857.1  1515857.1  1635857.1  1562428.
6
## 1996  1658142.9  1558142.9  1721857.1  1529142.9  1524571.4  1468428.
6
## 1997  1492714.3  1272714.3  1414714.3  1410571.4  1423571.4  1400857.
1
```

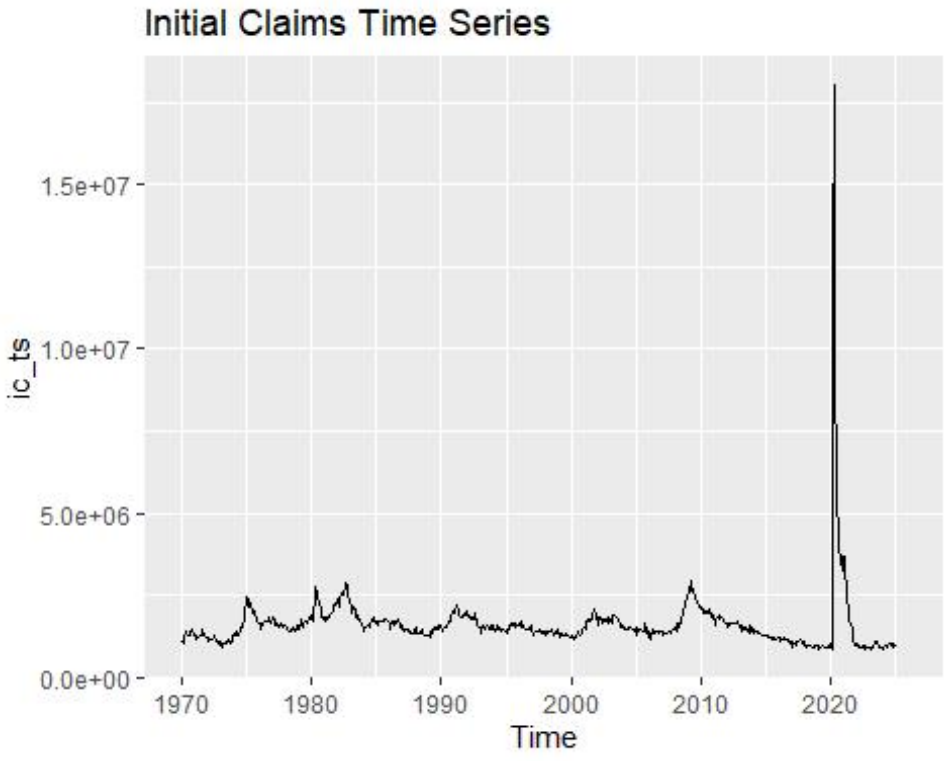
```
## 1998  1427714.3  1276000.0  1393714.3  1327428.6  1382571.4  1463428.
6
## 1999  1436142.9  1207285.7  1345285.7  1310571.4  1340857.1  1253428.
6
## 2000  1286000.0  1208428.6  1204428.6  1171857.1  1252857.1  1239142.
9
## 2001  1526714.3  1489571.4  1716571.4  1695857.1  1759142.9  1696000.
0
## 2002  1809142.9  1587428.6  1860571.4  1828714.3  1784714.3  1667857.
1
## 2003  1765571.4  1663714.3  1882428.6  1870428.6  1875428.6  1810714.
3
## 2004  1597571.4  1492142.9  1510285.7  1479285.7  1504714.3  1480857.
1
## 2005  1499571.4  1254428.6  1472142.9  1376714.3  1452857.1  1391714.
3
## 2006  1306857.1  1168571.4  1327000.0  1318142.9  1456857.1  1317428.
6
## 2007  1399000.0  1296714.3  1364142.9  1366714.3  1354857.1  1352428.
6
## 2008  1506142.9  1432714.3  1622428.6  1548285.7  1622571.4  1639000.
0
## 2009  2572571.4  2576000.0  2916857.1  2661142.9  2715142.9  2553428.
6
## 2010  2139857.1  1943857.1  2087285.7  1988857.1  2041571.4  1970285.
7
## 2011  1894857.1  1609428.6  1787857.1  1820857.1  1871285.7  1790285.
7
## 2012  1668714.3  1509571.4  1618714.3  1633571.4  1656714.3  1626714.
3
## 2013  1566142.9  1397000.0  1567428.6  1481285.7  1544000.0  1476571.
4
## 2014  1452714.3  1330714.3  1420428.6  1387857.1  1383285.7  1340285.
7
## 2015  1270000.0  1205000.0  1256857.1  1213857.1  1219000.0  1180428.
6
## 2016  1241000.0  1101857.1  1175142.9  1135000.0  1211571.4  1129571.
4
## 2017  1074571.4   923571.4  1019000.0   987142.9  1066142.9  1068000.
0
## 2018  1033000.0   886285.7   998714.3   946285.7   984000.0   953714.
3
## 2019   992428.6   878571.4   932714.3   907428.6   968714.3   965571.
4
## 2020   943428.6   873571.4 11971142.9 18020000.0  9094857.1  6341571.
4
## 2021  3681428.6  3003000.0  2946571.4  2430571.4  2022857.1  1731000.
0
## 2022  1053428.6   881428.6   950142.9   912428.6   931714.3   915857.
1
```

```
## 2023   906142.9   866000.0  1002857.1   928857.1  1017714.3  1075000.
0
## 2024   933857.1   872142.9   953857.1   908857.1   986142.9  1013428.
6
## 2025   956428.6   905714.3         NA         NA         NA
NA
##             Jul        Aug        Sep        Oct        Nov        D
ec
## 1970  1251000.0  1295857.1  1381428.6  1468000.0  1389000.0  1352428.
6
## 1971  1307714.3  1454857.1  1340714.3  1317428.6  1233142.9  1203000.
0
## 1972  1259428.6  1126857.1  1096714.3  1113285.7  1055142.9  1082857.
1
## 1973  1080714.3  1130714.3  1060285.7  1072857.1  1093857.1  1277428.
6
## 1974  1385000.0  1515571.4  1549000.0  1771000.0  1998000.0  2291428.
6
## 1975  1971428.6  1999571.4  1899714.3  1857285.7  1701428.6  1650571.
4
## 1976  1728857.1  1734428.6  1741000.0  1809285.7  1674571.4  1648428.
6
## 1977  1611000.0  1628714.3  1559571.4  1601285.7  1519142.9  1588285.
7
## 1978  1557000.0  1514285.7  1389428.6  1448857.1  1455428.6  1560000.
0
## 1979  1693571.4  1731857.1  1654428.6  1775857.1  1783857.1  1897000.
0
## 1980  2478000.0  2307571.4  2056000.0  1913142.9  1763285.7  1814857.
1
## 1981  1933857.1  1978142.9  2037285.7  2184571.4  2206428.6  2398142.
9
## 1982  2537142.9  2797000.0  2860428.6  2849714.3  2531571.4  2314000.
0
## 1983  1797857.1  1918571.4  1754142.9  1769857.1  1665714.3  1635428.
6
## 1984  1605714.3  1724285.7  1704571.4  1861142.9  1730000.0  1689714.
3
## 1985  1689142.9  1789428.6  1749714.3  1791571.4  1682714.3  1696142.
9
## 1986  1639142.9  1797000.0  1692428.6  1677428.6  1555857.1  1565714.
3
## 1987  1435571.4  1420000.0  1320714.3  1328000.0  1318142.9  1408857.
1
## 1988  1435285.7  1388714.3  1293428.6  1288142.9  1250857.1  1295714.
3
## 1989  1505428.6  1476428.6  1445000.0  1595714.3  1450000.0  1581000.
0
## 1990  1626142.9  1714857.1  1691714.3  1889000.0  1927857.1  2007000.
0
```

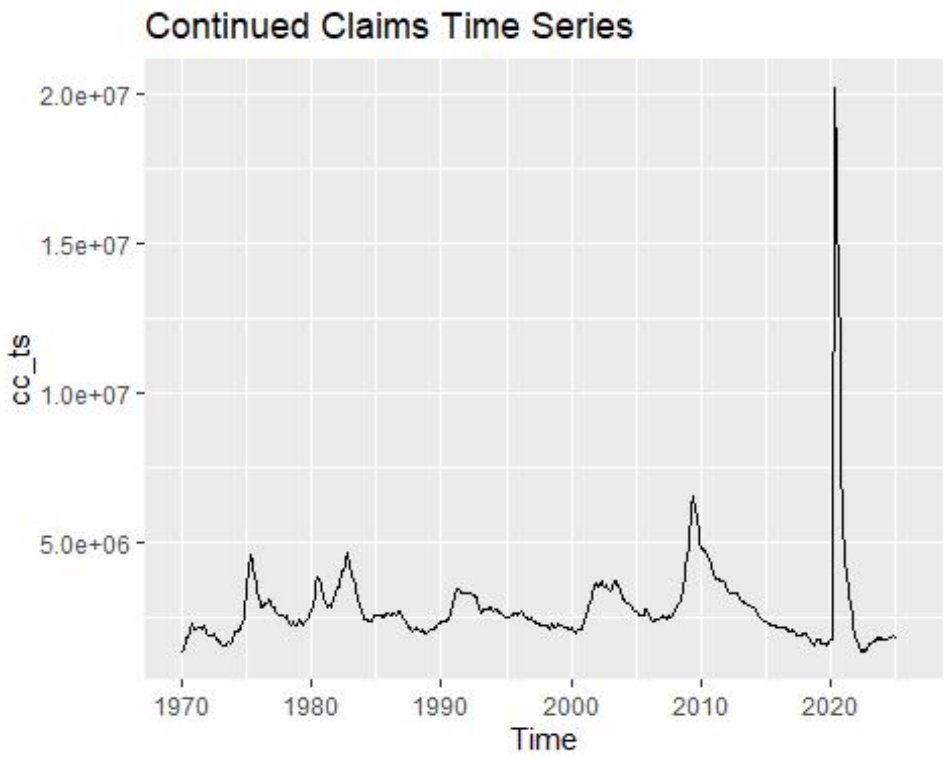
```
## 1991  1863000.0  1907714.3  1819142.9  1889142.9  1907714.3  2012142.
9
## 1992  1959142.9  1792285.7  1776571.4  1661571.4  1549000.0  1499571.
4
## 1993  1563142.9  1507142.9  1450714.3  1561142.9  1458428.6  1439857.
1
## 1994  1520714.3  1490428.6  1410142.9  1477857.1  1403428.6  1430857.
1
## 1995  1653000.0  1578285.7  1556428.6  1648857.1  1611571.4  1610142.
9
## 1996  1485428.6  1472714.3  1466714.3  1498857.1  1443428.6  1562714.
3
## 1997  1410714.3  1467142.9  1349571.4  1375000.0  1350000.0  1386000.
0
## 1998  1457142.9  1372142.9  1299142.9  1380428.6  1357142.9  1407285.
7
## 1999  1326000.0  1287285.7  1254428.6  1271571.4  1209714.3  1236142.
9
## 2000  1321571.4  1378571.4  1286000.0  1345857.1  1436571.4  1536714.
3
## 2001  1763857.1  1764714.3  1898428.6  2106428.6  1937142.9  1798857.
1
## 2002  1709857.1  1746285.7  1753428.6  1803857.1  1671714.3  1825428.
6
## 2003  1808714.3  1774857.1  1687142.9  1673714.3  1556285.7  1584000.
0
## 2004  1528714.3  1502714.3  1448571.4  1477857.1  1414857.1  1459142.
9
## 2005  1432857.1  1404857.1  1673857.1  1524714.3  1359571.4  1398571.
4
## 2006  1414000.0  1391142.9  1356000.0  1398714.3  1401857.1  1433571.
4
## 2007  1387571.4  1414571.4  1343000.0  1454714.3  1452000.0  1545000.
0
## 2008  1806000.0  1908000.0  1999571.4  2114428.6  2260428.6  2462000.
0
## 2009  2496142.9  2479714.3  2335142.9  2322428.6  2118000.0  2125714.
3
## 2010  2036571.4  2093714.3  1938857.1  1977428.6  1826428.6  1851571.
4
## 2011  1822285.7  1816714.3  1781571.4  1776857.1  1666142.9  1663857.
1
## 2012  1649000.0  1652000.0  1641000.0  1632571.4  1717857.1  1577285.
7
## 2013  1527571.4  1473285.7  1361142.9  1573142.9  1396714.3  1522000.
0
## 2014  1331428.6  1343142.9  1261714.3  1269428.6  1259571.4  1264428.
6
## 2015  1217285.7  1222571.4  1152571.4  1182428.6  1154000.0  1199000.
0
```

```
## 2016  1152714.3  1163000.0  1069142.9  1140428.6  1053285.7  1119285.
7
## 2017  1107142.9  1130571.4  1149857.1  1075285.7  1056857.1  1076857.
1
## 2018   928000.0   916285.7   883857.1   946428.6   975285.7   988428.
6
## 2019   922142.9   936142.9   897714.3   942142.9   969142.9  1040000.
0
## 2020  5823285.7  3972000.0  3600714.3  3528428.6  3254428.6  3671142.
9
## 2021  1638000.0  1584428.6  1501857.1  1235571.4   998857.1   970857.
1
## 2022   969000.0   951000.0   849285.7   895428.6   904142.9   920571.
4
## 2023  1035142.9  1068428.6   932857.1   933571.4   933714.3   911142.
9
## 2024  1048000.0  1020285.7   974000.0  1028857.1   939142.9   976857.
1
## 2025         NA         NA         NA         NA         NA
NA
```

```
autoplot(ic_ts) + ggtitle("Initial Claims Time Series")
```




```
autoplot(cc_ts) + ggtitle("Continued Claims Time Series")
```



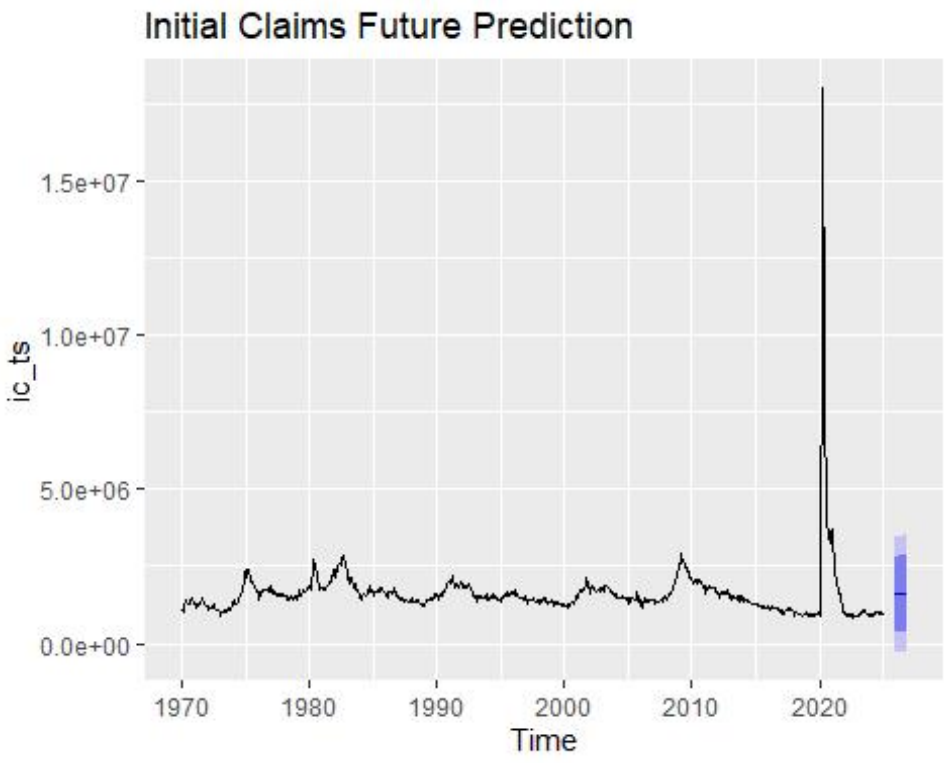

```
ic_model <- auto.arima(ic_ts)
cc_model <- auto.arima(cc_ts)


summary(ic_model)

## Series: ic_ts
## ARIMA(3,0,2) with non-zero mean
##
## Coefficients:
##          ar1     ar2     ar3     ma1     ma2        mean
##       0.4256  0.0109  0.1801  0.8876  0.0668  1613025.14
## s.e.  0.8528  0.3845  0.1183  0.8708  0.7536    97311.99
##
## sigma^2 = 2.443e+11:  log likelihood = -9616.93
## AIC=19247.86   AICc=19248.03   BIC=19279.32
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MAS
E
## Training set -108.7424 492063.7 176999.8 -2.476309 10.14929 0.438541
2
##                      ACF1
## Training set 0.001299218

summary(cc_model)
```

```
## Series: cc_ts
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1     ma1       mean
##       0.8646  0.5468  2827939.0
## s.e.  0.0204  0.0363   227513.7
##
## sigma^2 = 2.758e+11:  log likelihood = -9658.57
## AIC=19325.14   AICc=19325.2   BIC=19343.12
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE
      ACF1
## Training set 406.2501 523950.7 107453.5 -1.18604 3.176813 0.1388508
0.01281172
```

```
ic_forecast <- forecast(ic_model, h=12)
cc_forecast <- forecast(cc_model, h=12)
```

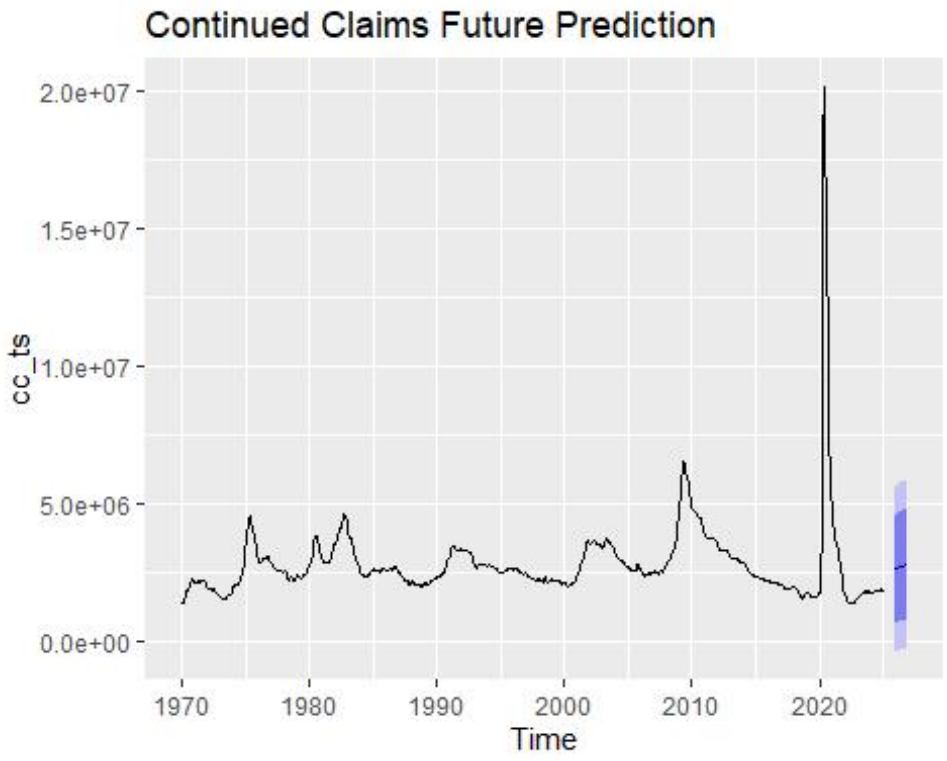
```
autoplot(ic_forecast) + ggtitle("Initial Claims Future Prediction")
```

```
## Warning: Removed 10 rows containing missing values (`geom_line()`).
```




```
autoplot(cc_forecast) + ggtitle("Continued Claims Future Prediction")
```

```
## Warning: Removed 10 rows containing missing values (`geom_line()`).
```




```
library(forecast)
library(ggplot2)
library(tseries)


data <- read.csv("C://Users//Anna Liu//Desktop//Columbia//Actuarial//PM
//group//project_data(2)(2).csv")


data$observation_date <- as.Date(data$observation_date)
data$discouraged <- NULL
data$X <- NULL


data_clean <- na.omit(data)


ic_ts <- ts(data_clean$IC, start = c(1970, 1), frequency = 12)
cc_ts <- ts(data_clean$CC, start = c(1970, 1), frequency = 12)

print(ic_ts)

##             Jan        Feb        Mar        Apr        May        J
un
```

```
## 1970  1103571.4  1034000.0  1190000.0  1422571.4  1356428.6  1261714.
3
## 1971  1319857.1  1136142.9  1303428.6  1227714.3  1298428.6  1258285.
7
## 1972  1195714.3  1085285.7  1162000.0  1132571.4  1183142.9  1242142.
9
## 1973  1012000.0   886285.7  1007428.6  1026857.1  1055857.1  1028285.
7
## 1974  1359857.1  1273285.7  1385285.7  1252714.3  1305285.7  1334142.
9
## 1975  2423714.3  2166857.1  2418428.6  2236142.9  2205428.6  2102857.
1
## 1976  1649142.9  1442714.3  1609428.6  1593857.1  1737142.9  1691000.
0
## 1977  1844714.3  1744714.3  1587857.1  1587000.0  1676571.4  1539428.
6
## 1978  1558285.7  1530857.1  1529285.7  1392285.7  1437428.6  1418142.
9
## 1979  1577571.4  1438000.0  1587142.9  1762857.1  1531142.9  1560000.
0
## 1980  1833000.0  1734000.0  1951000.0  2313285.7  2752142.9  2593428.
6
## 1981  1830714.3  1715000.0  1808714.3  1757285.7  1848142.9  1868000.
0
## 1982  2401857.1  2135000.0  2476857.1  2511428.6  2599857.1  2566285.
7
## 1983  2190428.6  1953285.7  2130285.7  2129285.7  2068714.3  1881714.
3
## 1984  1550571.4  1408714.3  1530714.3  1581285.7  1590285.7  1553428.
6
## 1985  1653285.7  1558428.6  1694714.3  1702285.7  1740714.3  1681571.
4
## 1986  1623857.1  1487428.6  1693428.6  1658142.9  1706000.0  1588000.
0
## 1987  1543428.6  1395000.0  1474428.6  1439285.7  1466142.9  1381857.
1
## 1988  1498428.6  1317285.7  1355000.0  1340285.7  1387857.1  1318000.
0
## 1989  1282428.6  1230857.1  1425571.4  1327857.1  1438000.0  1445428.
6
## 1990  1603571.4  1423285.7  1525428.6  1548142.9  1582714.3  1549142.
9
## 1991  1985428.6  1939571.4  2209000.0  1997714.3  1965000.0  1833000.
0
## 1992  1948000.0  1833857.1  1894571.4  1804285.7  1838571.4  1790000.
0
## 1993  1530000.0  1358000.0  1562000.0  1504428.6  1522000.0  1463857.
1
## 1994  1606857.1  1390285.7  1487857.1  1475714.3  1583857.1  1450714.
3
```

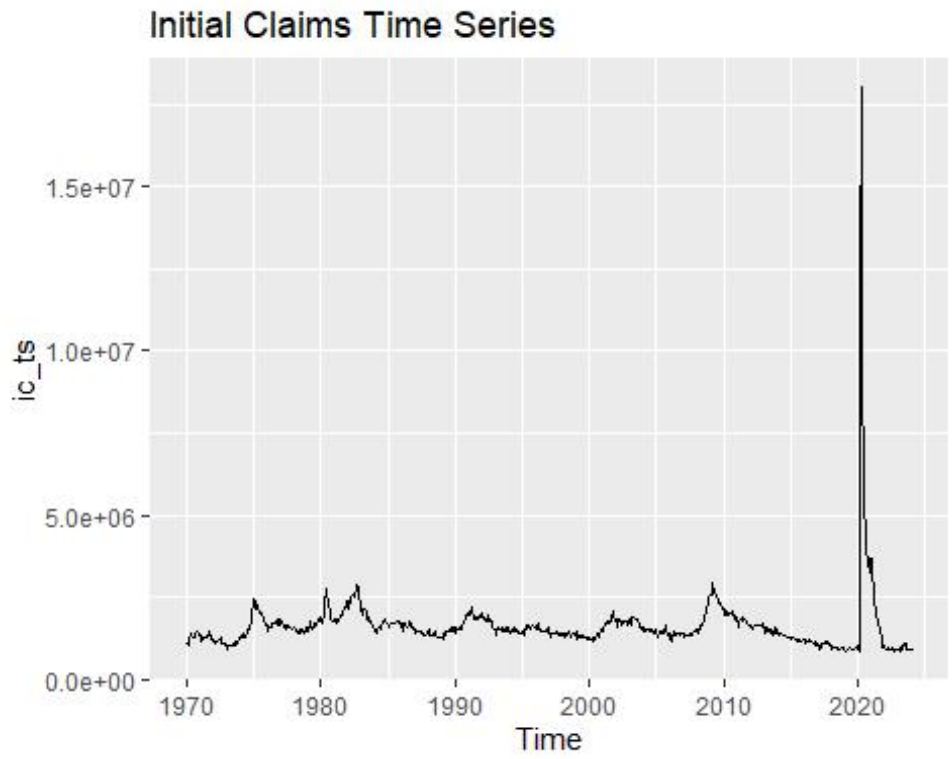
```
## 1995  1472857.1  1357428.6  1502857.1  1515857.1  1635857.1  1562428.
6
## 1996  1658142.9  1558142.9  1721857.1  1529142.9  1524571.4  1468428.
6
## 1997  1492714.3  1272714.3  1414714.3  1410571.4  1423571.4  1400857.
1
## 1998  1427714.3  1276000.0  1393714.3  1327428.6  1382571.4  1463428.
6
## 1999  1436142.9  1207285.7  1345285.7  1310571.4  1340857.1  1253428.
6
## 2000  1286000.0  1208428.6  1204428.6  1171857.1  1252857.1  1239142.
9
## 2001  1526714.3  1489571.4  1716571.4  1695857.1  1759142.9  1696000.
0
## 2002  1809142.9  1587428.6  1860571.4  1828714.3  1784714.3  1667857.
1
## 2003  1765571.4  1663714.3  1882428.6  1870428.6  1875428.6  1810714.
3
## 2004  1597571.4  1492142.9  1510285.7  1479285.7  1504714.3  1480857.
1
## 2005  1499571.4  1254428.6  1472142.9  1376714.3  1452857.1  1391714.
3
## 2006  1306857.1  1168571.4  1327000.0  1318142.9  1456857.1  1317428.
6
## 2007  1399000.0  1296714.3  1364142.9  1366714.3  1354857.1  1352428.
6
## 2008  1506142.9  1432714.3  1622428.6  1548285.7  1622571.4  1639000.
0
## 2009  2572571.4  2576000.0  2916857.1  2661142.9  2715142.9  2553428.
6
## 2010  2139857.1  1943857.1  2087285.7  1988857.1  2041571.4  1970285.
7
## 2011  1894857.1  1609428.6  1787857.1  1820857.1  1871285.7  1790285.
7
## 2012  1668714.3  1509571.4  1618714.3  1633571.4  1656714.3  1626714.
3
## 2013  1566142.9  1397000.0  1567428.6  1481285.7  1544000.0  1476571.
4
## 2014  1452714.3  1330714.3  1420428.6  1387857.1  1383285.7  1340285.
7
## 2015  1270000.0  1205000.0  1256857.1  1213857.1  1219000.0  1180428.
6
## 2016  1241000.0  1101857.1  1175142.9  1135000.0  1211571.4  1129571.
4
## 2017  1074571.4   923571.4  1019000.0   987142.9  1066142.9  1068000.
0
## 2018  1033000.0   886285.7   998714.3   946285.7   984000.0   953714.
3
## 2019   992428.6   878571.4   932714.3   907428.6   968714.3   965571.
4
```

```
## 2020   943428.6   873571.4 11971142.9 18020000.0  9094857.1  6341571.
4
## 2021  3681428.6  3003000.0  2946571.4  2430571.4  2022857.1  1731000.
0
## 2022  1053428.6   881428.6   950142.9   912428.6   931714.3   915857.
1
## 2023   906142.9   866000.0  1002857.1   928857.1  1017714.3  1075000.
0
## 2024   933857.1   872142.9

##             Jul        Aug        Sep        Oct        Nov        D
ec
## 1970  1251000.0  1295857.1  1381428.6  1468000.0  1389000.0  1352428.
6
## 1971  1307714.3  1454857.1  1340714.3  1317428.6  1233142.9  1203000.
0
## 1972  1259428.6  1126857.1  1096714.3  1113285.7  1055142.9  1082857.
1
## 1973  1080714.3  1130714.3  1060285.7  1072857.1  1093857.1  1277428.
6
## 1974  1385000.0  1515571.4  1549000.0  1771000.0  1998000.0  2291428.
6
## 1975  1971428.6  1999571.4  1899714.3  1857285.7  1701428.6  1650571.
4
## 1976  1728857.1  1734428.6  1741000.0  1809285.7  1674571.4  1648428.
6
## 1977  1611000.0  1628714.3  1559571.4  1601285.7  1519142.9  1588285.
7
## 1978  1557000.0  1514285.7  1389428.6  1448857.1  1455428.6  1560000.
0
## 1979  1693571.4  1731857.1  1654428.6  1775857.1  1783857.1  1897000.
0
## 1980  2478000.0  2307571.4  2056000.0  1913142.9  1763285.7  1814857.
1
## 1981  1933857.1  1978142.9  2037285.7  2184571.4  2206428.6  2398142.
9
## 1982  2537142.9  2797000.0  2860428.6  2849714.3  2531571.4  2314000.
0
## 1983  1797857.1  1918571.4  1754142.9  1769857.1  1665714.3  1635428.
6
## 1984  1605714.3  1724285.7  1704571.4  1861142.9  1730000.0  1689714.
3
## 1985  1689142.9  1789428.6  1749714.3  1791571.4  1682714.3  1696142.
9
## 1986  1639142.9  1797000.0  1692428.6  1677428.6  1555857.1  1565714.
3
## 1987  1435571.4  1420000.0  1320714.3  1328000.0  1318142.9  1408857.
1
## 1988  1435285.7  1388714.3  1293428.6  1288142.9  1250857.1  1295714.
3
```

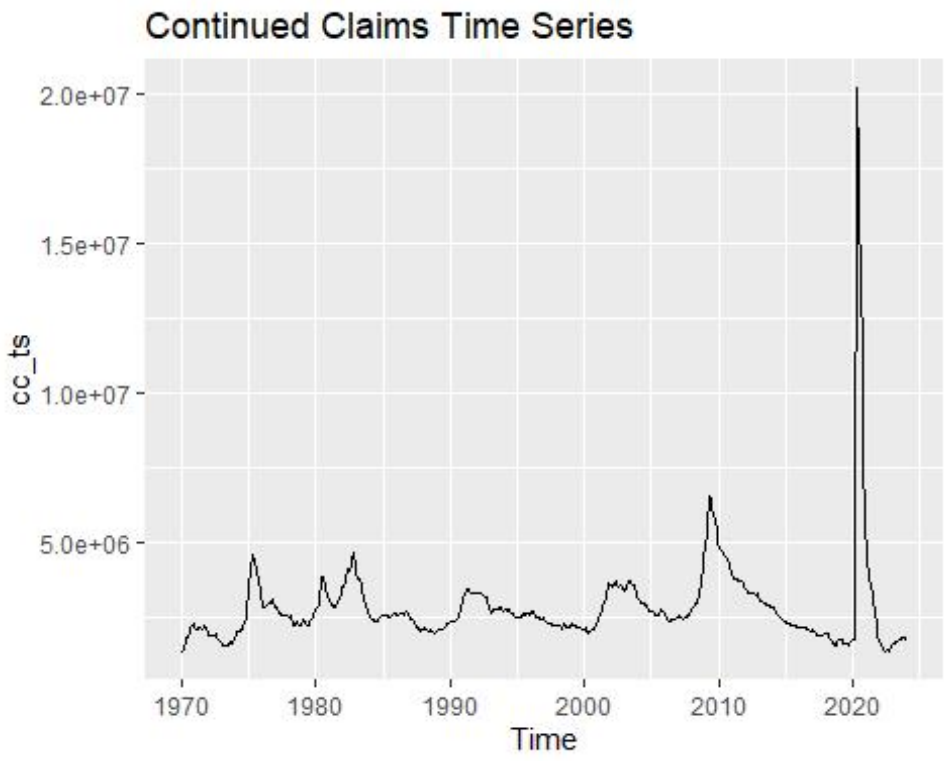
```
## 1989  1505428.6  1476428.6  1445000.0  1595714.3  1450000.0  1581000.
0
## 1990  1626142.9  1714857.1  1691714.3  1889000.0  1927857.1  2007000.
0
## 1991  1863000.0  1907714.3  1819142.9  1889142.9  1907714.3  2012142.
9
## 1992  1959142.9  1792285.7  1776571.4  1661571.4  1549000.0  1499571.
4
## 1993  1563142.9  1507142.9  1450714.3  1561142.9  1458428.6  1439857.
1
## 1994  1520714.3  1490428.6  1410142.9  1477857.1  1403428.6  1430857.
1
## 1995  1653000.0  1578285.7  1556428.6  1648857.1  1611571.4  1610142.
9
## 1996  1485428.6  1472714.3  1466714.3  1498857.1  1443428.6  1562714.
3
## 1997  1410714.3  1467142.9  1349571.4  1375000.0  1350000.0  1386000.
0
## 1998  1457142.9  1372142.9  1299142.9  1380428.6  1357142.9  1407285.
7
## 1999  1326000.0  1287285.7  1254428.6  1271571.4  1209714.3  1236142.
9
## 2000  1321571.4  1378571.4  1286000.0  1345857.1  1436571.4  1536714.
3
## 2001  1763857.1  1764714.3  1898428.6  2106428.6  1937142.9  1798857.
1
## 2002  1709857.1  1746285.7  1753428.6  1803857.1  1671714.3  1825428.
6
## 2003  1808714.3  1774857.1  1687142.9  1673714.3  1556285.7  1584000.
0
## 2004  1528714.3  1502714.3  1448571.4  1477857.1  1414857.1  1459142.
9
## 2005  1432857.1  1404857.1  1673857.1  1524714.3  1359571.4  1398571.
4
## 2006  1414000.0  1391142.9  1356000.0  1398714.3  1401857.1  1433571.
4
## 2007  1387571.4  1414571.4  1343000.0  1454714.3  1452000.0  1545000.
0
## 2008  1806000.0  1908000.0  1999571.4  2114428.6  2260428.6  2462000.
0
## 2009  2496142.9  2479714.3  2335142.9  2322428.6  2118000.0  2125714.
3
## 2010  2036571.4  2093714.3  1938857.1  1977428.6  1826428.6  1851571.
4
## 2011  1822285.7  1816714.3  1781571.4  1776857.1  1666142.9  1663857.
1
## 2012  1649000.0  1652000.0  1641000.0  1632571.4  1717857.1  1577285.
7
## 2013  1527571.4  1473285.7  1361142.9  1573142.9  1396714.3  1522000.
0
```

```
## 2014  1331428.6  1343142.9  1261714.3  1269428.6  1259571.4  1264428.
6
## 2015  1217285.7  1222571.4  1152571.4  1182428.6  1154000.0  1199000.
0
## 2016  1152714.3  1163000.0  1069142.9  1140428.6  1053285.7  1119285.
7
## 2017  1107142.9  1130571.4  1149857.1  1075285.7  1056857.1  1076857.
1
## 2018   928000.0   916285.7   883857.1   946428.6   975285.7   988428.
6
## 2019   922142.9   936142.9   897714.3   942142.9   969142.9  1040000.
0
## 2020  5823285.7  3972000.0  3600714.3  3528428.6  3254428.6  3671142.
9
## 2021  1638000.0  1584428.6  1501857.1  1235571.4   998857.1   970857.
1
## 2022   969000.0   951000.0   849285.7   895428.6   904142.9   920571.
4
## 2023  1035142.9  1068428.6   932857.1   933571.4   933714.3   911142.
9
## 2024
```

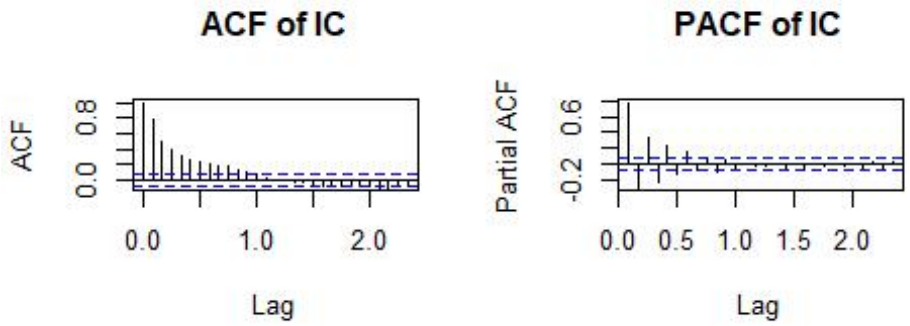
```r
autoplot(ic_ts) + ggtitle("Initial Claims Time Series")
```

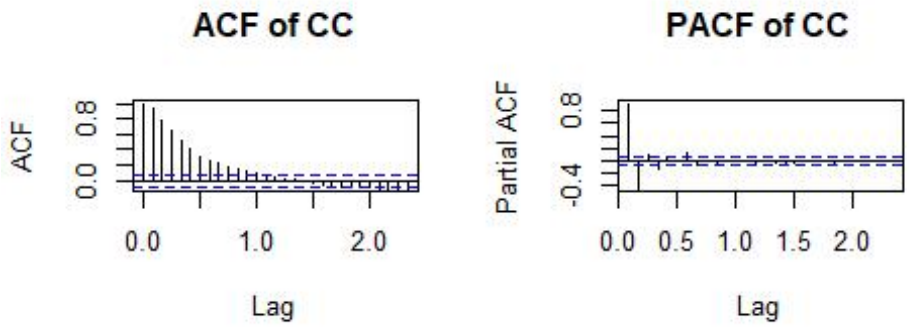



```r
autoplot(cc_ts) + ggtitle("Continued Claims Time Series")
```




```r
par(mfrow=c(2,2))
acf(ic_ts, main="ACF of IC")
pacf(ic_ts, main="PACF of IC")
acf(cc_ts, main="ACF of CC")
pacf(cc_ts, main="PACF of CC")
```





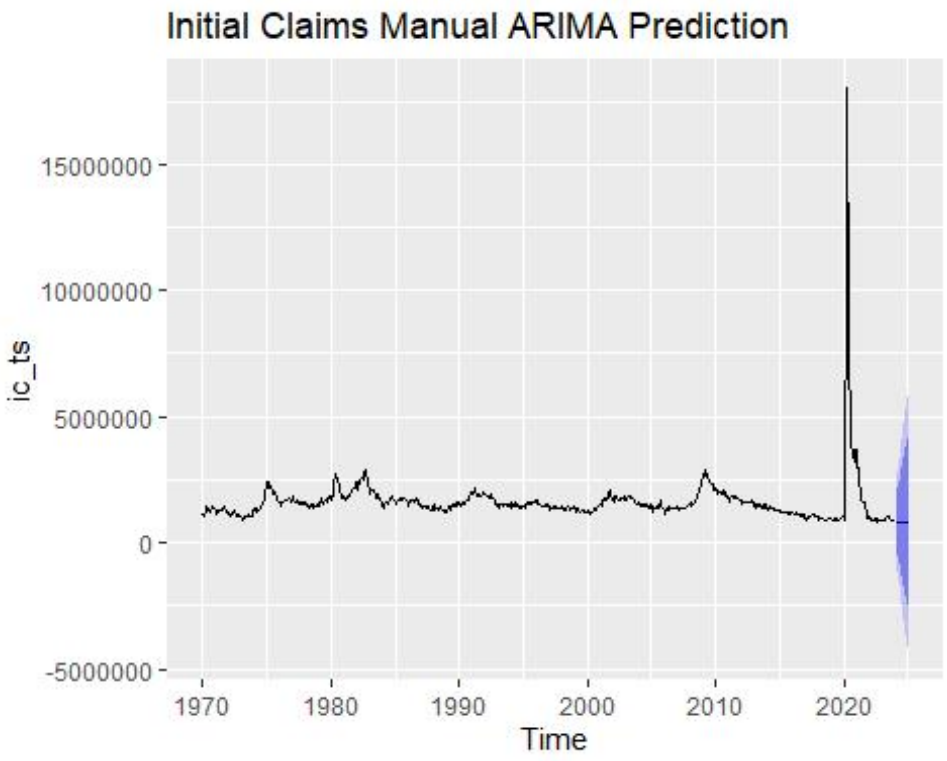

```r
par(mfrow=c(1,1))


ic_model_manual <- arima(ic_ts, order=c(1,1,1))
cc_model_manual <- arima(cc_ts, order=c(1,1,1))


summary(ic_model_manual)
```

```
##
## Call:
## arima(x = ic_ts, order = c(1, 1, 1))
##
## Coefficients:
##           ar1     ma1
##       -0.3305  0.8645
## s.e.   0.0430  0.0175
##
## sigma^2 estimated as 3.019e+11:  log likelihood = -9498.88,  aic = 1
9003.77
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
       ACF1
## Training set -282.108 548994.6 171112.3 -0.4136672 9.33175 1.319143
-0.08682375
```

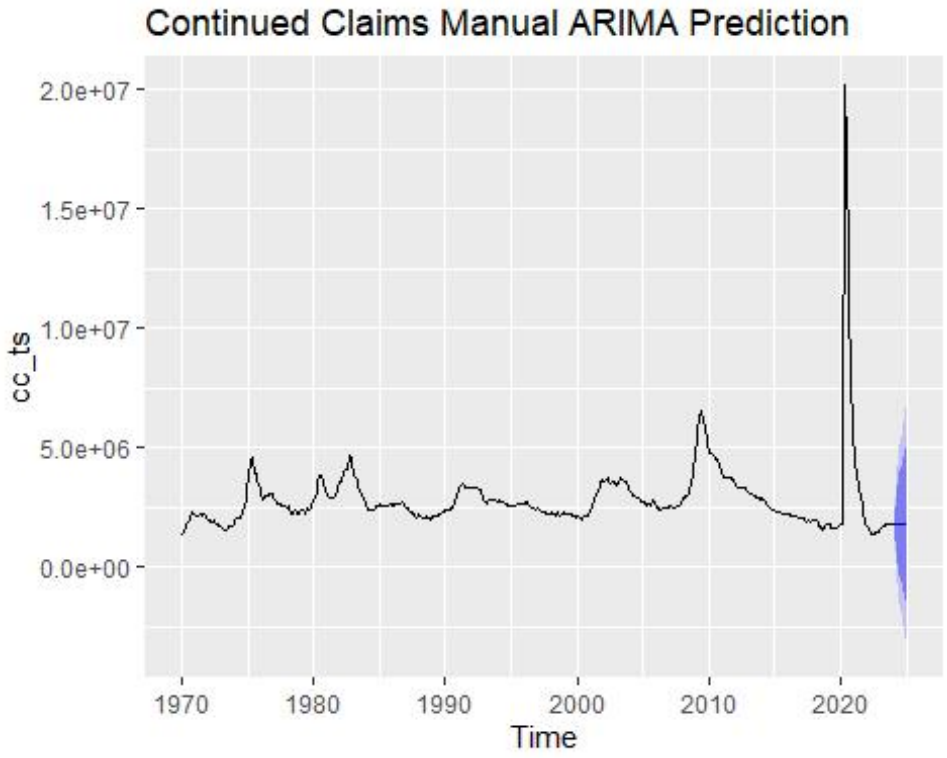
```r
summary(cc_model_manual)
```

```
##
## Call:
## arima(x = cc_ts, order = c(1, 1, 1))
##
## Coefficients:
##           ar1     ma1
##       -0.0991  0.5900
## s.e.   0.0944  0.0818
##
## sigma^2 estimated as 2.99e+11:  log likelihood = -9495.54,  aic = 18
997.08
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE     MAPE      MA
SE
## Training set 454.4881 546383.5 96382.48 -0.07777366 2.285327 0.81972
73
##                      ACF1
## Training set 0.004235727
```

```r
ic_forecast_manual <- forecast(ic_model_manual, h=12)
cc_forecast_manual <- forecast(cc_model_manual, h=12)


autoplot(ic_forecast_manual) + ggtitle("Initial Claims Manual ARIMA Pre
diction")
```




```
autoplot(cc_forecast_manual) + ggtitle("Continued Claims Manual ARIMA P
rediction")
```



```r
ic_auto_model <- auto.arima(ic_ts)
cc_auto_model <- auto.arima(cc_ts)

summary(ic_auto_model)
```

```
## Series: ic_ts
## ARIMA(3,0,2) with non-zero mean
##
## Coefficients:
##           ar1     ar2     ar3     ma1     ma2        mean
##       -0.4357  0.4629  0.1793  1.7778  0.8163  1621645.13
## s.e.   0.1060  0.0772  0.0503  0.1038  0.0833    87740.32
##
## sigma^2 = 2.469e+11:  log likelihood = -9446.02
## AIC=18906.03   AICc=18906.2   BIC=18937.37
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 642.4714 494558.3 175505.5 -2.579447 10.05998 0.4276046
##                      ACF1
## Training set -0.01262006
```

```r
summary(cc_auto_model)
```

```
## Series: cc_ts
## ARIMA(1,0,1) with non-zero mean
##
## Coefficients:
##          ar1     ma1       mean
##       0.8636  0.5468  2827299.4
## s.e.  0.0206  0.0366   234260.2
##
## sigma^2 = 2.808e+11:  log likelihood = -9489.3
## AIC=18986.6   AICc=18986.66   BIC=19004.51
##
## Training set error measures:
##                     ME   RMSE      MAE       MPE     MAPE      MASE
     ACF1
## Training set 2020.991 528641 108129.5 -1.130136 3.163341 0.1373794 0.
01332749
```

```r
ic_manual_model <- arima(ic_ts, order = c(1,1,1))
cc_manual_model <- arima(cc_ts, order = c(1,1,1))

summary(ic_manual_model)
```

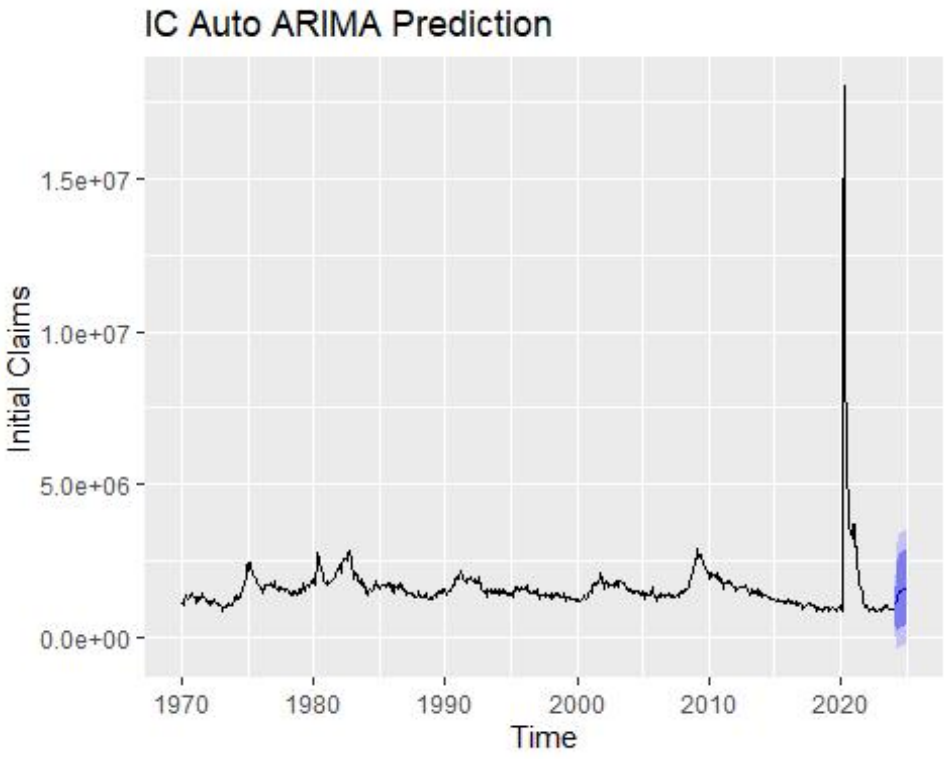
```
##
## Call:
## arima(x = ic_ts, order = c(1, 1, 1))
##
## Coefficients:
```

```
##           ar1     ma1
##       -0.3305  0.8645
## s.e.   0.0430  0.0175
##
## sigma^2 estimated as 3.019e+11:  log likelihood = -9498.88,  aic = 1
9003.77
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE    MAPE     MASE
       ACF1
## Training set -282.108 548994.6 171112.3 -0.4136672 9.33175 1.319143
-0.08682375
```

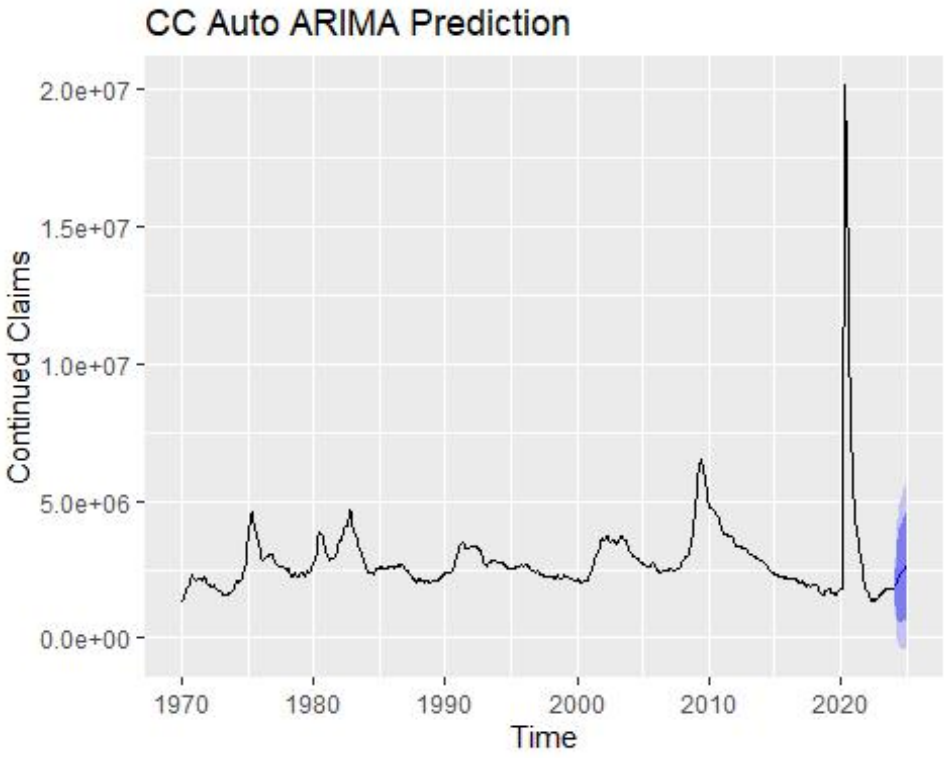
```r
summary(cc_manual_model)
```

```
##
## Call:
## arima(x = cc_ts, order = c(1, 1, 1))
##
## Coefficients:
##           ar1     ma1
##       -0.0991  0.5900
## s.e.   0.0944  0.0818
##
## sigma^2 estimated as 2.99e+11:  log likelihood = -9495.54,  aic = 18
997.08
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE     MAPE      MA
SE
## Training set 454.4881 546383.5 96382.48 -0.07777366 2.285327 0.81972
73
##                      ACF1
## Training set 0.004235727
```

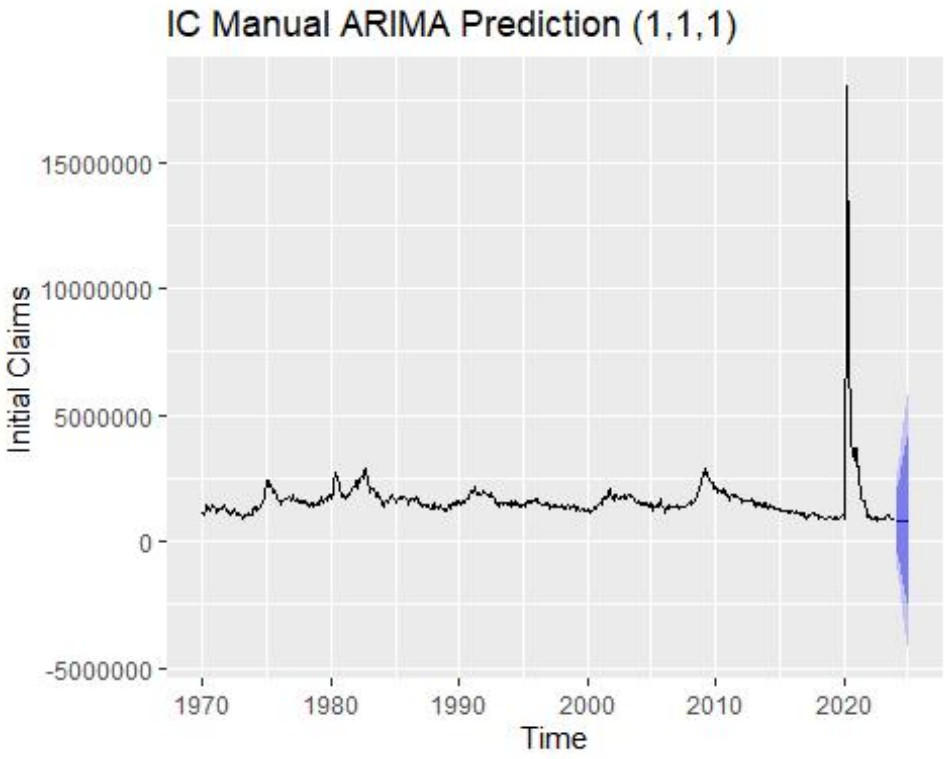
```r
ic_auto_forecast <- forecast(ic_auto_model, h = 12)
cc_auto_forecast <- forecast(cc_auto_model, h = 12)

ic_manual_forecast <- forecast(ic_manual_model, h = 12)
cc_manual_forecast <- forecast(cc_manual_model, h = 12)


autoplot(ic_auto_forecast) + ggtitle("IC Auto ARIMA Prediction") + xlab
("Time") + ylab("Initial Claims")
```



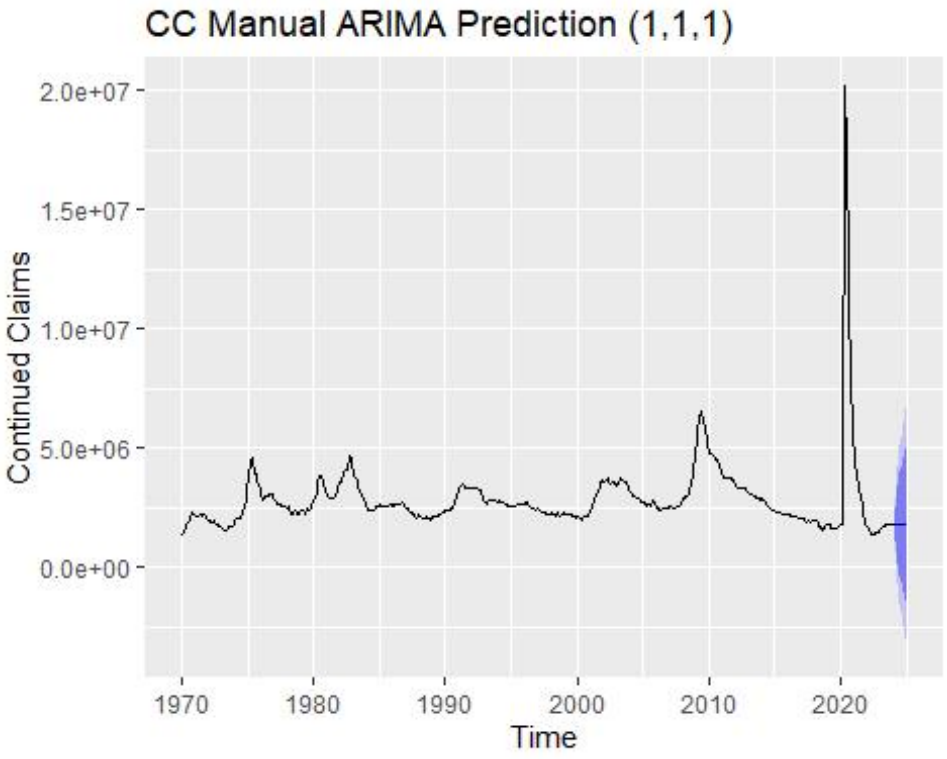

```
autoplot(cc_auto_forecast) + ggtitle("CC Auto ARIMA Prediction") + xlab
("Time") + ylab("Continued Claims")
```



```
autoplot(ic_manual_forecast) + ggtitle("IC Manual ARIMA Prediction (1,1,
1)") + xlab("Time") + ylab("Initial Claims")
```




```
autoplot(cc_manual_forecast) + ggtitle("CC Manual ARIMA Prediction (1,1,
1)") + xlab("Time") + ylab("Continued Claims")
```




```
cat("Initial Claims:\n")
```

```
## Initial Claims:
```

```
cat("Auto ARIMA AIC:", AIC(ic_auto_model), " BIC:", BIC(ic_auto_model),
 "\n")
```

```
## Auto ARIMA AIC: 18906.03  BIC: 18937.37
```

```
cat("Manual ARIMA AIC:", AIC(ic_manual_model), " BIC:", BIC(ic_manual_m
odel), "\n\n")
```

```
## Manual ARIMA AIC: 19003.77  BIC: 19017.19
```

```
cat("Continued Claims:\n")
```

```
## Continued Claims:
```

```
cat("Auto ARIMA AIC:", AIC(cc_auto_model), " BIC:", BIC(cc_auto_model),
 "\n")
```

```
## Auto ARIMA AIC: 18986.6  BIC: 19004.51
```

```
cat("Manual ARIMA AIC:", AIC(cc_manual_model), " BIC:", BIC(cc_manual_m
odel), "\n")
```

```
## Manual ARIMA AIC: 18997.08  BIC: 19010.5
```

```r
p_max <- 4
d_max <- 3
q_max <- 4


results <- data.frame(p=integer(), d=integer(), q=integer(), AIC=numeri
c())


for (p in 0:p_max) {
  for (d in 0:d_max) {
    for (q in 0:q_max) {
      tryCatch({
        model <- arima(ic_ts, order=c(p,d,q))
        results <- rbind(results, data.frame(p=p, d=d, q=q, AIC=AIC(mod
el)))
      }, error=function(e){})
    }
  }
}
```

```
## Warning in log(s2): 产生了 NaNs
## Warning in log(s2): 产生了 NaNs
```

```r
results <- results[order(results$AIC), ]


print(results)
```

```
##    p d q      AIC
## 83 4 1 3 18860.46
## 54 2 2 4 18862.12
## 67 3 1 4 18862.35
## 84 4 1 4 18863.57
## 49 2 1 4 18864.38
## 72 3 2 4 18871.05
## 71 3 2 3 18872.28
## 88 4 2 3 18873.59
## 89 4 2 4 18879.87
## 66 3 1 3 18881.42
## 65 3 1 2 18883.60
## 82 4 1 2 18884.99
## 30 1 1 4 18885.62
## 34 1 2 3 18887.53
## 15 0 2 4 18888.59
## 48 2 1 3 18888.63
## 35 1 2 4 18889.07
## 53 2 2 3 18889.23
## 94 4 3 4 18891.09
## 29 1 1 3 18892.05
```

```
## 47 2 1 2 18897.47
## 44 2 0 3 18899.54
## 93 4 3 3 18899.80
## 28 1 1 2 18904.79
## 62 3 0 2 18906.03
## 9  0 1 3 18906.66
## 61 3 0 1 18908.45
## 10 0 1 4 18908.66
## 25 1 0 4 18909.32
## 79 4 0 1 18909.77
## 87 4 2 2 18910.22
## 24 1 0 3 18910.70
## 77 3 3 4 18913.39
## 43 2 0 2 18913.70
## 76 3 3 3 18914.21
## 58 2 3 3 18915.56
## 23 1 0 2 18916.82
## 14 0 2 3 18917.70
## 5  0 0 4 18920.14
## 42 2 0 1 18922.00
## 40 1 3 4 18923.38
## 22 1 0 1 18924.99
## 70 3 2 2 18925.62
## 92 4 3 2 18926.25
## 52 2 2 2 18927.19
## 81 4 1 1 18929.58
## 86 4 2 1 18934.26
## 8  0 1 2 18936.70
## 20 0 3 4 18939.23
## 59 2 3 4 18940.76
## 64 3 1 1 18945.32
## 46 2 1 1 18947.04
## 80 4 1 0 18953.31
## 4  0 0 3 18956.06
## 3  0 0 2 18967.14
## 39 1 3 3 18970.23
## 78 4 0 0 18979.36
## 33 1 2 2 18983.29
## 69 3 2 1 18999.75
## 27 1 1 1 19003.77
## 51 2 2 1 19008.71
## 60 3 0 0 19010.04
## 63 3 1 0 19019.44
## 19 0 3 3 19024.33
## 45 2 1 0 19028.15
## 13 0 2 2 19038.39
## 7  0 1 1 19059.43
## 41 2 0 0 19083.21
## 91 4 3 1 19083.77
## 85 4 2 0 19101.75
```

```
## 2  0 0 1 19116.32
## 21 1 0 0 19138.76
## 32 1 2 1 19146.42
## 75 3 3 2 19157.54
## 12 0 2 1 19159.64
## 26 1 1 0 19166.80
## 18 0 3 2 19169.20
## 6  0 1 0 19179.72
## 74 3 3 1 19190.15
## 56 2 3 1 19190.55
## 57 2 3 2 19200.40
## 68 3 2 0 19208.96
## 50 2 2 0 19209.48
## 90 4 3 0 19368.50
## 38 1 3 2 19401.14
## 37 1 3 1 19457.06
## 17 0 3 1 19473.75
## 31 1 2 0 19477.33
## 11 0 2 0 19494.35
## 73 3 3 0 19523.51
## 55 2 3 0 19552.67
## 1  0 0 0 19751.17
## 36 1 3 0 19936.71
## 16 0 3 0 20015.39
```

```
results <- data.frame(p=integer(), d=integer(), q=integer(), AIC=numeri
c(), BIC=numeric())
```

```
for (p in 0:p_max) {
  for (d in 0:d_max) {
    for (q in 0:q_max) {
      tryCatch({
        model <- arima(ic_ts, order=c(p,d,q))
        results <- rbind(results, data.frame(p=p, d=d, q=q, AIC=AIC(mod
el), BIC=BIC(model)))
      }, error=function(e){})
    }
  }
}
```

```
## Warning in log(s2): 产生了 NaNs
## Warning in log(s2): 产生了 NaNs
```

```
best_aic <- results[order(results$AIC), ][1:5, ]
print("Top Best AIC combinations:")
```

```
## [1] "Top Best AIC combinations:"
```

```
print(best_aic)
```

```
##    p d q      AIC      BIC
## 83 4 1 3 18860.46 18896.27
## 54 2 2 4 18862.12 18893.44
## 67 3 1 4 18862.35 18898.15
## 84 4 1 4 18863.57 18903.85
## 49 2 1 4 18864.38 18895.70
```

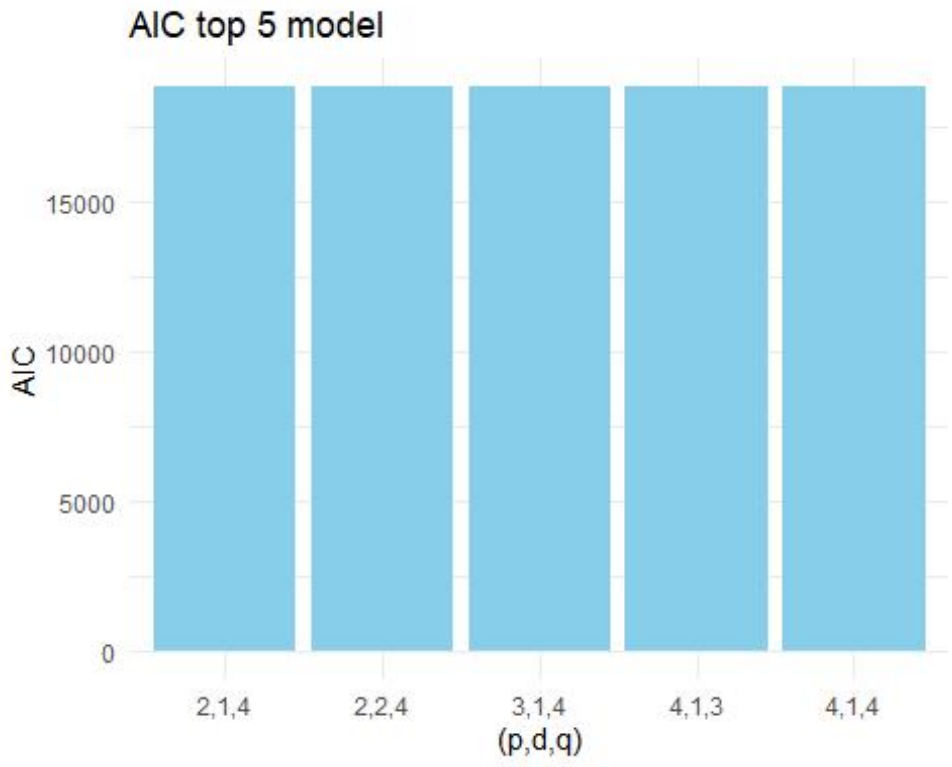
```
best_bic <- results[order(results$BIC), ][1:5, ]
print("Top Best BIC combinations:")
```

```
## [1] "Top Best BIC combinations:"
```

```
print(best_bic)
```

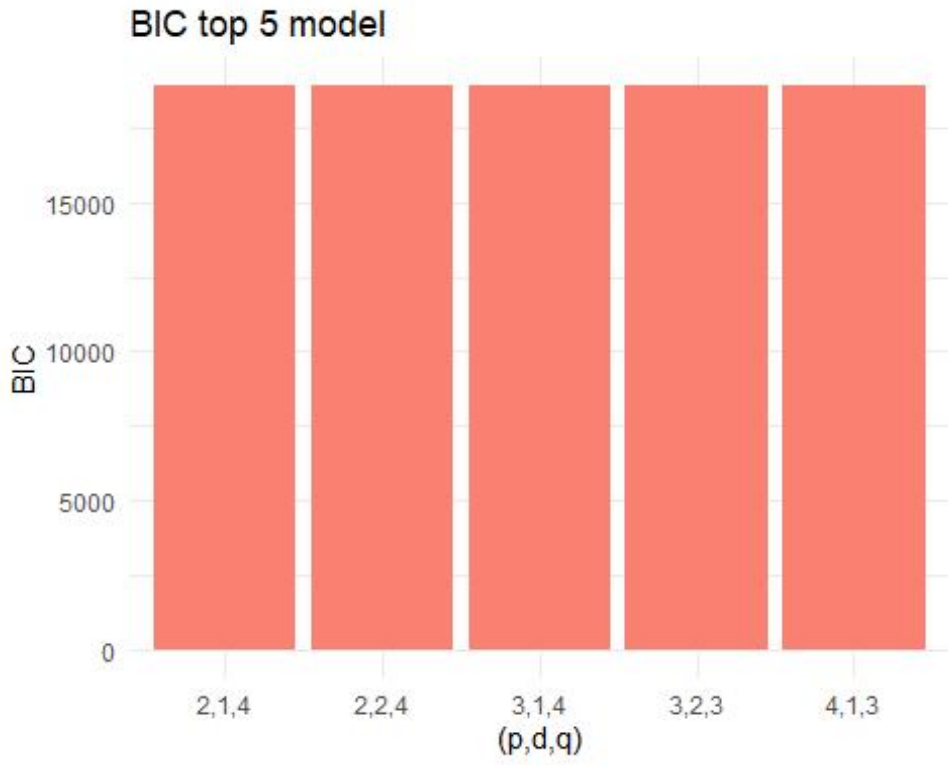
```
##    p d q      AIC      BIC
## 54 2 2 4 18862.12 18893.44
## 49 2 1 4 18864.38 18895.70
## 83 4 1 3 18860.46 18896.27
## 67 3 1 4 18862.35 18898.15
## 71 3 2 3 18872.28 18903.59
```

```
ggplot(best_aic, aes(x=paste(p,d,q,sep=","), y=AIC)) +
  geom_col(fill="skyblue") +
  labs(title="AIC top 5 model", x="(p,d,q)", y="AIC") +
  theme_minimal()
```

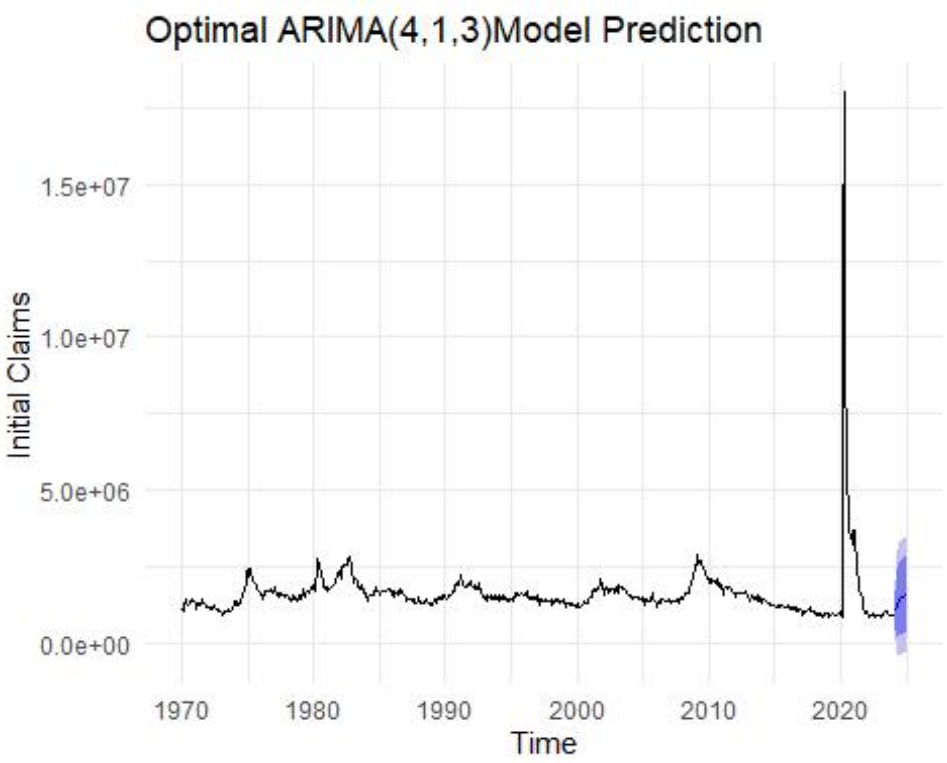



```
ggplot(best_bic, aes(x=paste(p,d,q,sep=","), y=BIC)) +
  geom_col(fill="salmon") +
```

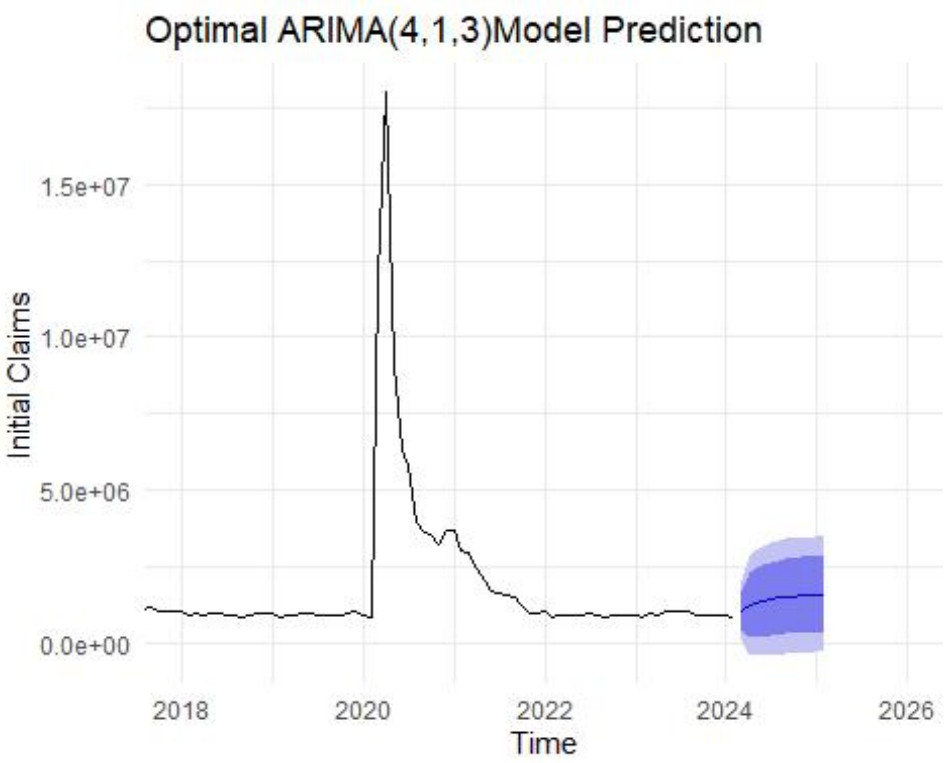
```
  labs(title="BIC top 5 model", x="(p,d,q)", y="BIC") +
  theme_minimal()
```



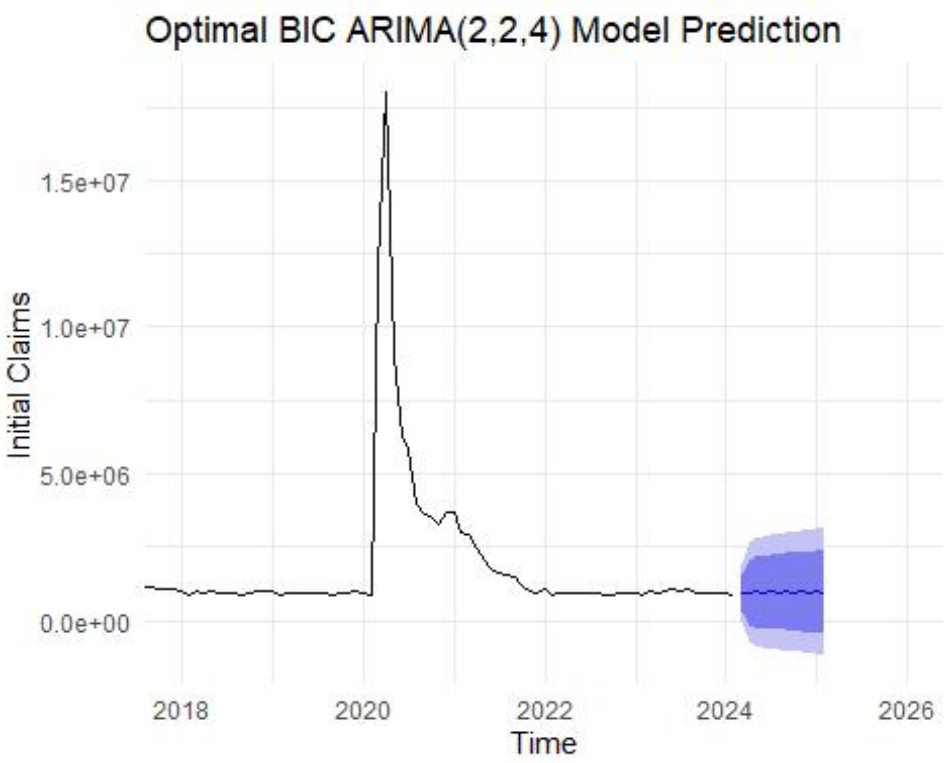

```
best_model_params <- results[which.min(results$AIC), ]


cat("The Optimal Model: (p,d,q) = (", best_model_params$p, ",", best_mo
del_params$d, ",", best_model_params$q, ")\n")

## The Optimal Model: (p,d,q) = ( 4 , 1 , 3 )

best_arima_model <- arima(ic_ts, order=c(best_model_params$p, best_mode
l_params$d, best_model_params$q))


best_forecast <- forecast(best_arima_model, h=12)


autoplot(best_forecast) +
  ggtitle(paste0("Optimal ARIMA(", best_model_params$p, ",", best_model
_params$d, ",", best_model_params$q, ")Model Prediction")) +
  xlab("Time") + ylab("Initial Claims") +
  theme_minimal()
```




```
autoplot(best_forecast) +
  ggtitle(paste0("Optimal ARIMA(", best_model_params$p, ",", best_model
_params$d, ",", best_model_params$q, ")Model Prediction ")) +
  xlab("Time") + ylab("Initial Claims") +
  theme_minimal() +
  coord_cartesian(xlim=c(2018, 2026))
```




```
best_bic_model_params <- results[which.min(results$BIC), ]


cat("Optimal BIC Model: (p,d,q) = (", best_bic_model_params$p, ",", bes
t_bic_model_params$d, ",", best_bic_model_params$q, ")\n")

## Optimal BIC Model: (p,d,q) = ( 2 , 2 , 4 )

best_bic_arima_model <- arima(ic_ts, order = c(best_bic_model_params$p,
 best_bic_model_params$d, best_bic_model_params$q))


best_bic_forecast <- forecast(best_bic_arima_model, h = 12)


autoplot(best_bic_forecast) +
  ggtitle(paste0("Optimal BIC ARIMA(", best_bic_model_params$p, ",", be
st_bic_model_params$d, ",", best_bic_model_params$q, ") Model Predictio
n")) +
  xlab("Time") + ylab("Initial Claims") +
  theme_minimal() +
  coord_cartesian(xlim = c(2018, 2026))
```




```
results <- data.frame(p=integer(), d=integer(), q=integer(), AIC=numeri
c(), BIC=numeric())

# 遍历所有(p,d,q)
for (p in 0:p_max) {
  for (d in 0:d_max) {
    for (q in 0:q_max) {
      tryCatch({
        model <- arima(cc_ts, order=c(p,d,q))
        results <- rbind(results, data.frame(p=p, d=d, q=q, AIC=AIC(mod
el), BIC=BIC(model)))
      }, error=function(e){})
    }
  }
}
```

```
## Warning in log(s2): 产生了 NaNs
## Warning in log(s2): 产生了 NaNs
## Warning in log(s2): 产生了 NaNs
```

```
best_aic <- results[order(results$AIC), ][1:5, ]
print("Top 5 AIC")
```

```
## [1] "Top 5 AIC"
```

```
print(best_aic)
```

```
##    p d q      AIC      BIC
## 94 4 2 3 18940.35 18976.15
## 74 3 2 3 18940.69 18972.00
## 95 4 2 4 18942.87 18983.14
## 68 3 1 2 18952.34 18979.19
## 69 3 1 3 18954.15 18985.48
```

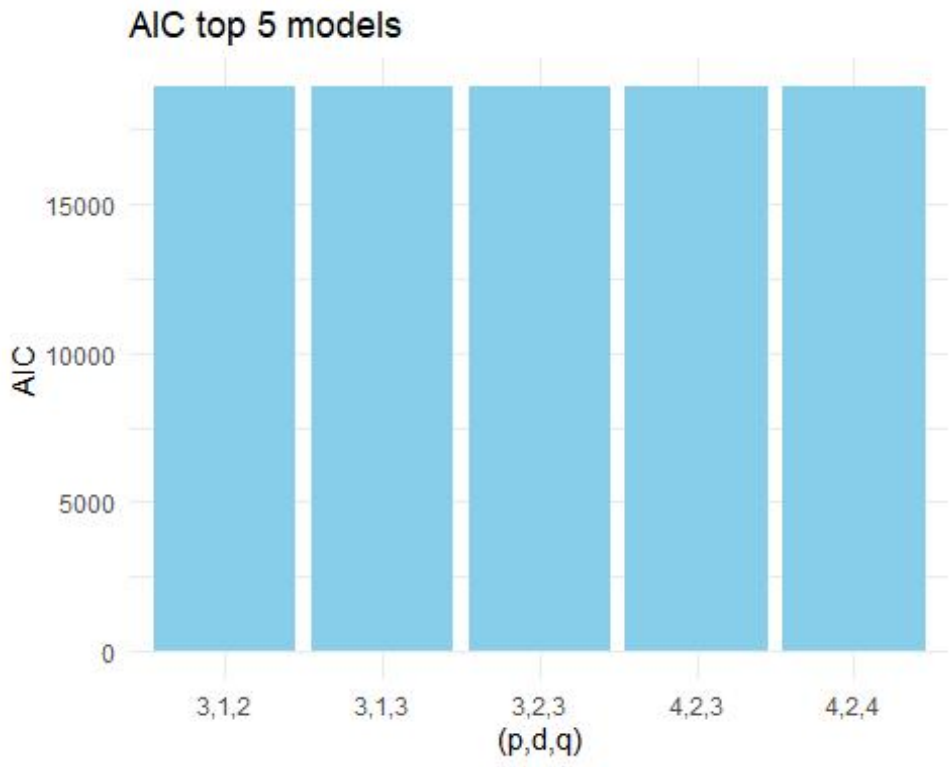
```
best_bic <- results[order(results$BIC), ][1:5, ]
print("Top 5 BIC")
```

```
## [1] "Top 5 BIC"
```

```
print(best_bic)
```

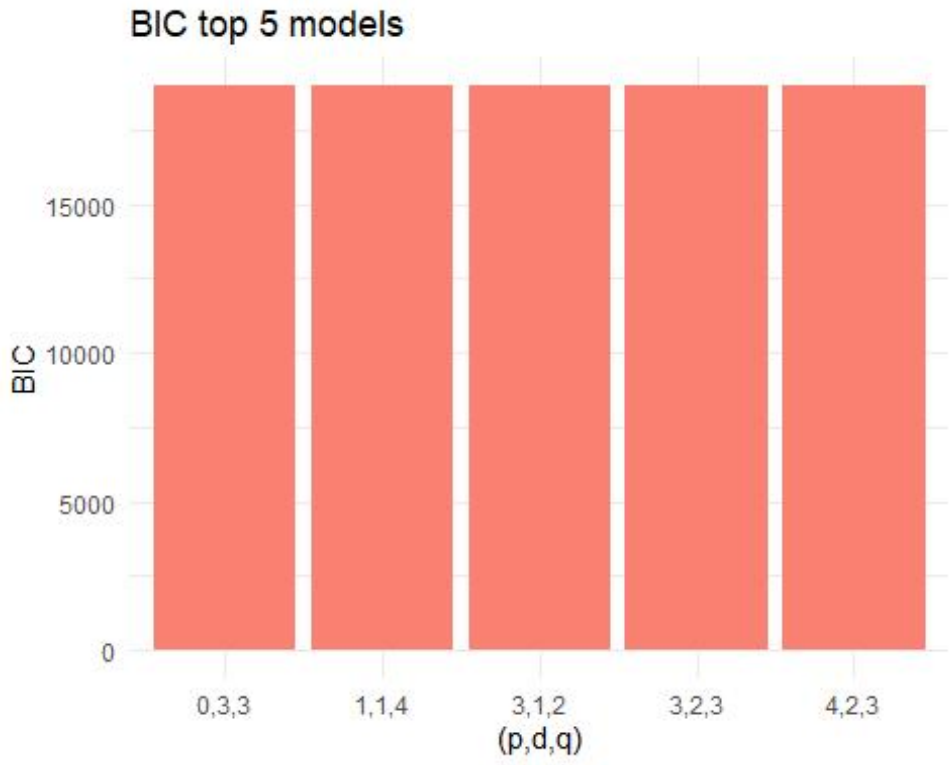
```
##    p d q      AIC      BIC
## 74 3 2 3 18940.69 18972.00
## 94 4 2 3 18940.35 18976.15
## 68 3 1 2 18952.34 18979.19
## 19 0 3 3 18963.94 18981.83
## 30 1 1 4 18955.43 18982.28
```

```
ggplot(best_aic, aes(x=paste(p,d,q,sep=","), y=AIC)) +
  geom_col(fill="skyblue") +
  labs(title="AIC top 5 models", x="(p,d,q)", y="AIC") +
  theme_minimal()
```

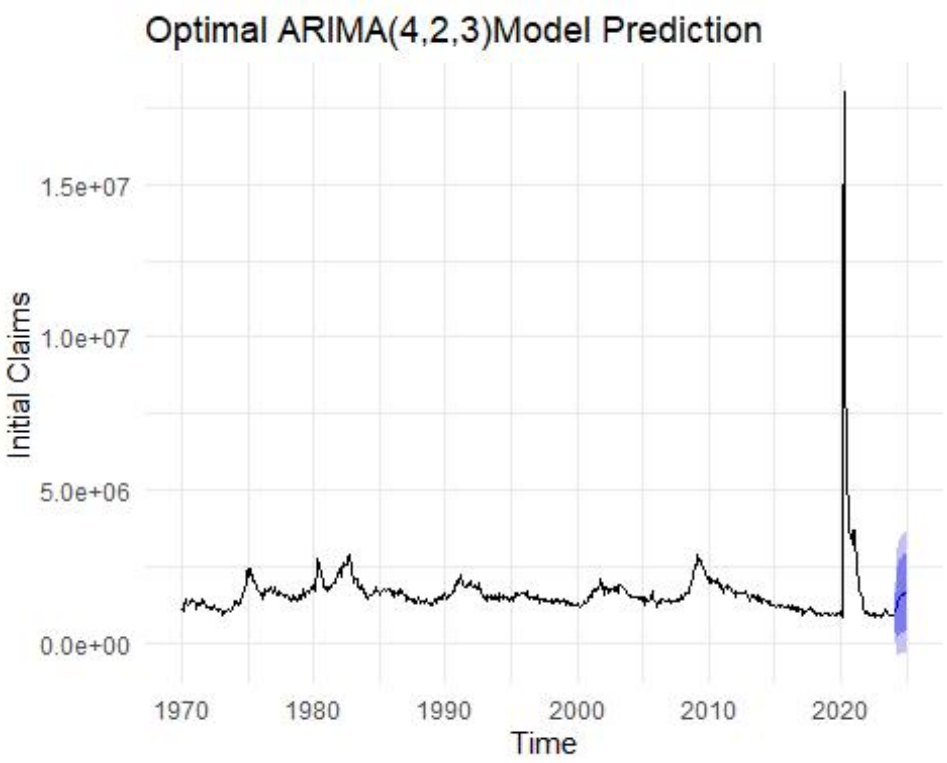



```
ggplot(best_bic, aes(x=paste(p,d,q,sep=","), y=BIC)) +
  geom_col(fill="salmon") +
```

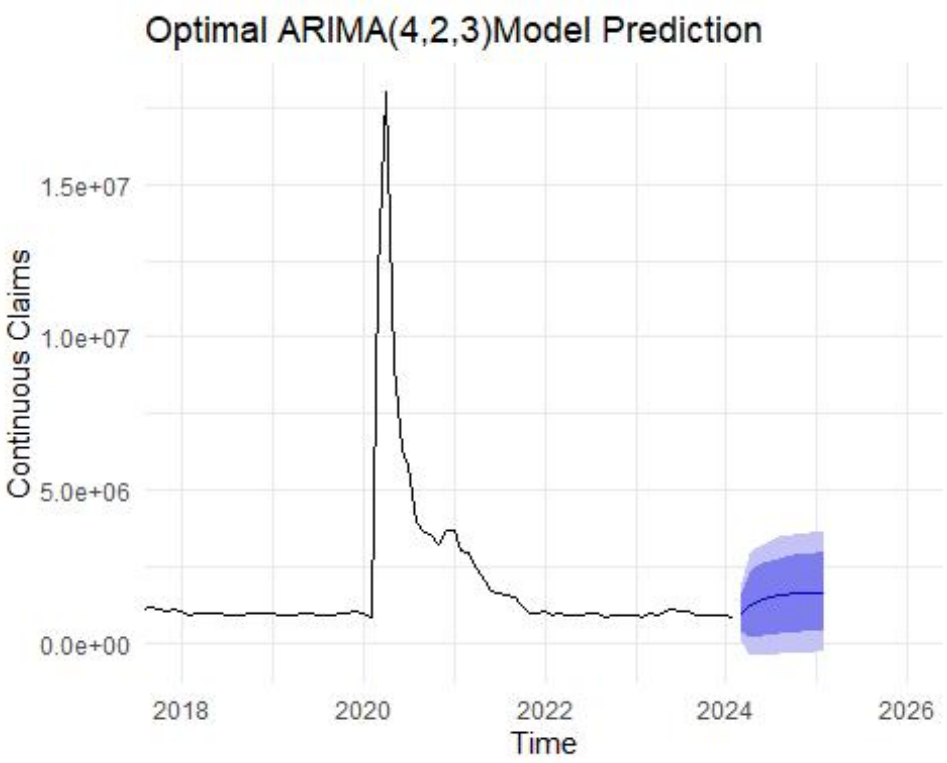
```
  labs(title="BIC top 5 models", x="(p,d,q)", y="BIC") +
  theme_minimal()
```



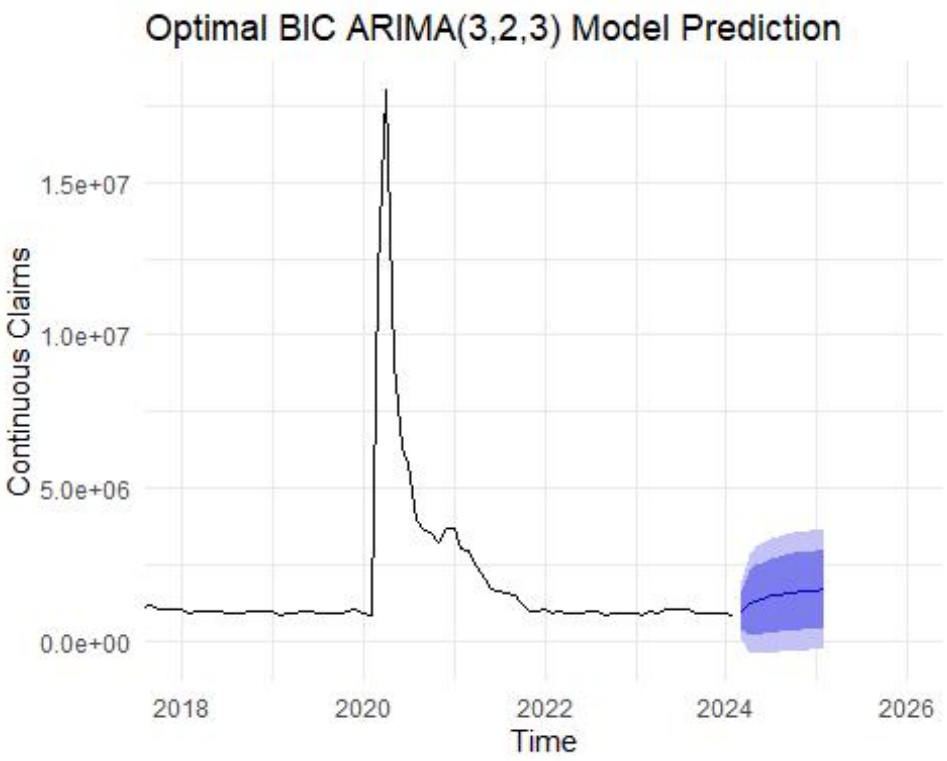

```
best_model_params <- results[which.min(results$AIC), ]

cat("The Best Model: (p,d,q) = (", best_model_params$p, ",", best_model
_params$d, ",", best_model_params$q, ")\n")

## The Best Model: (p,d,q) = ( 4 , 2 , 3 )

best_arima_model <- arima(ic_ts, order=c(best_model_params$p, best_mode
l_params$d, best_model_params$q))


best_forecast <- forecast(best_arima_model, h=12)


autoplot(best_forecast) +
  ggtitle(paste0("Optimal ARIMA(", best_model_params$p, ",", best_model
_params$d, ",", best_model_params$q, ")Model Prediction")) +
  xlab("Time") + ylab("Initial Claims") +
  theme_minimal()
```




```
autoplot(best_forecast) +
  ggtitle(paste0("Optimal ARIMA(", best_model_params$p, ",", best_model
_params$d, ",", best_model_params$q, ")Model Prediction ")) +
  xlab("Time") + ylab("Continuous Claims") +
  theme_minimal() +
  coord_cartesian(xlim=c(2018, 2026))
```




```
best_bic_model_params <- results[which.min(results$BIC), ]


cat("Optimal BIC Model: (p,d,q) = (", best_bic_model_params$p, ",", bes
t_bic_model_params$d, ",", best_bic_model_params$q, ")\n")

## Optimal BIC Model: (p,d,q) = ( 3 , 2 , 3 )

best_bic_arima_model <- arima(ic_ts, order = c(best_bic_model_params$p,
 best_bic_model_params$d, best_bic_model_params$q))


best_bic_forecast <- forecast(best_bic_arima_model, h = 12)


autoplot(best_bic_forecast) +
  ggtitle(paste0("Optimal BIC ARIMA(", best_bic_model_params$p, ",", be
st_bic_model_params$d, ",", best_bic_model_params$q, ") Model Predictio
n")) +
  xlab("Time") + ylab("Continuous Claims") +
  theme_minimal() +
  coord_cartesian(xlim = c(2018, 2026))
```




```r
fitted_values <- fitted(best_arima_model)

actual_values <- ic_ts


mse_in_sample <- mean((fitted_values - actual_values)^2)


cat("In-sample MSE =", mse_in_sample)
```

```
## In-sample MSE = 2.48743e+11
```