STAT 429 Final Project

Samuel Wu (samuel54), Molin Yang (moliny2), Rongsheng Zhang(rz36)

12-10-2025

Abstract

This project is about the car market regarding Used Vehicle CPI moves over time in the U.S. Our goal for this project is to find which model predicts the future car market most accurately. The data that we implemented into our project is the monthly data from 2000-2025, consisting of four predictors. They are New Vehicle CPI, Motor Fuel CPI, Federal Funds Rate as well as COVID dummy variable. In this project we attempted four models

- Model 1: Model 1: OLS multiple linear regression.
- Model 2: WLS regression to fix heteroscedasticity.
- Model 3: Lagged-predictor regression based on CCF analysis.
- Model 4: ARIMA model after differencing to make the series stationary.

Through out our attempts, we discovered in some linear regressions that the residuals were not normal, autocorrelation still existed as well as predictive performance wasn't great. Later we discovered that ARIMA worked better due to the fact that after differencing, the series became stationary, model selection including AIC as well as BIC pointed to ARIMA(0,1,2). Also, residuals were tighter, more stable, and predictions were more accurate.

We ended with the conclusion that ARIMA(0, 1, 2) was clearly the best-performing model and it produced the most reliable short-term forecasts of Used Vehicle CPI.

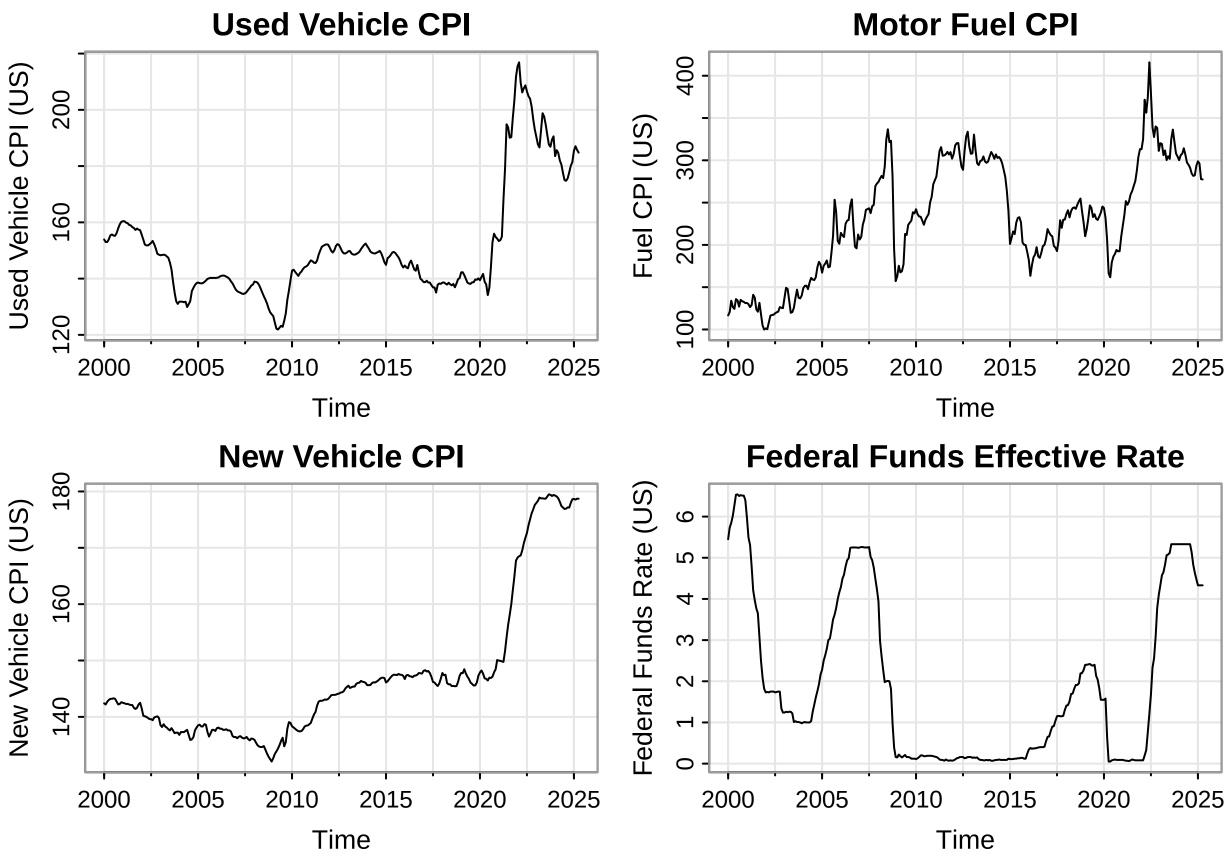
Introduction

In recent years, movements in consumer prices have become a central concern for households, firms, and policymakers in the United States. Among the various components of the Consumer Price Index (CPI), the index for used cars and trucks is of particular interest. Used vehicles are a relatively inexpensive way for many households to acquire transportation, and huge swings in their prices can immediately affect the cost of commuting and access to employment, as well as broader household budgets. Following the COVID-19 pandemic, the prices of used vehicles rose unusually rapidly and then partly reversed course. These fluctuations were cited extensively as one of the more visible contributors to the recent inflationary episode. This places great importance on understanding how the prices of used vehicles change over time and relate to overall macroeconomic conditions.

Our goal is to identify a model that can accurately estimate and forecast future values of the Used Vehicle CPI. We begin with multiple linear regression model approach, using New Vehicle CPI, Motor Fuel CPI, the Federal Funds Effective Rate, and a COVID-19 dummy variable as predictors. The New Vehicle CPI and Motor Fuel CPI are closely related to Used Vehicle CPI, while the Federal Funds Effective Rate and the COVID-19 dummy variable can reflect macroeconomic fluctuations. We evaluate the efficiency of this regression model and then propose a better ARIMA model which can significantly improve prediction accuracy.

We obtained all four data from Federal Reserve Economic Data (FRED). Each variable is recorded monthly from January 2000 to April 2025. We created time series objects in R and combined four variables together into a unified dataset.

Methods and Model Summary



Before fitting the model, we first displayed the time series plots of the response variable and the predictors. Since none of the series exhibit random fluctuations around a horizontal line, they are clearly non-stationary.

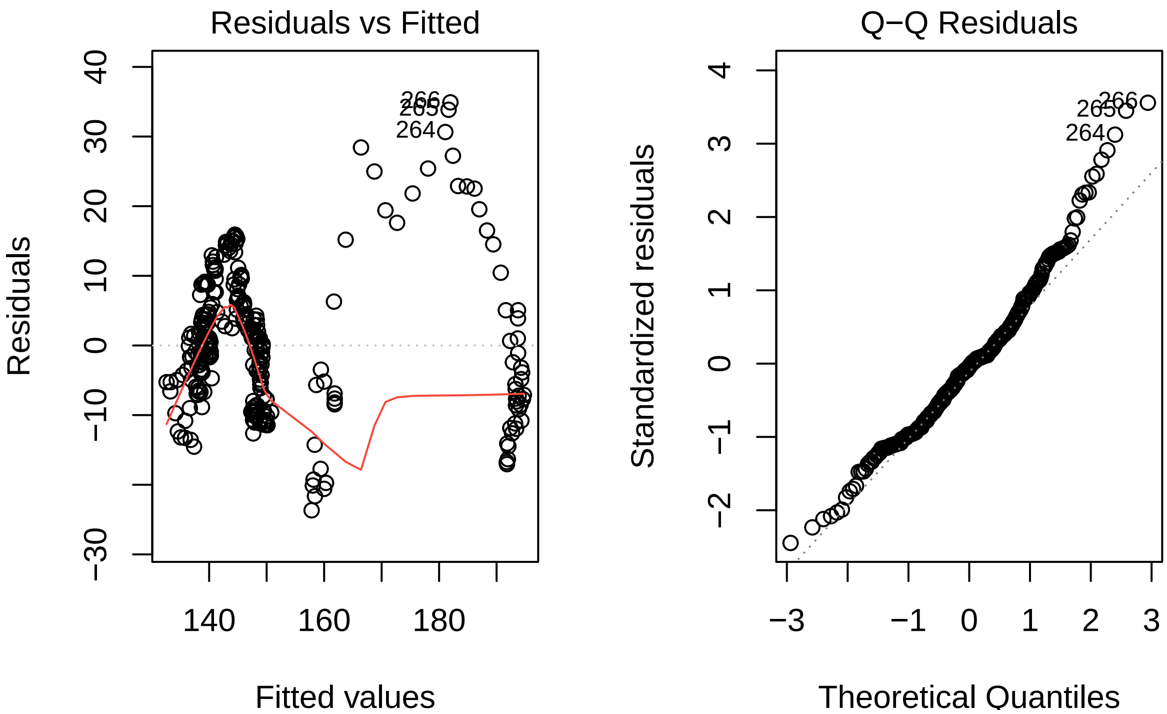
Model 1 (MLR with COVID dummy using OLS)

We first applied a multiple linear regression framework using the OLS method. The model is written in the following form:

$$\mathrm{CPI}_{\mathrm{used(t)}} = \beta_0 + \beta_1 \mathrm{CPI}_{\mathrm{new(t)}} + \beta_2 \mathrm{CPI}_{\mathrm{fuel(t)}} + \beta_3 \mathrm{FFER}_{(t)} + \beta_4 \mathrm{COVID}_{(t)} + \epsilon$$

```
##
## Call:
## lm(formula = used_cpi ~ new_cpi + fuel_cpi + fedfunds + covid,
##     data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.670  -7.127  -0.453   4.875  34.909
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.882950  11.990306  -0.657    0.511
```

```
## new_cpi      1.056579   0.091388  11.561  < 2e-16 ***
## fuel_cpi     0.005736   0.010505   0.546    0.585
## fedfunds     0.212106   0.308909   0.687    0.493
## covid        9.927836   2.515964   3.946 9.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.918 on 299 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.7507
## F-statistic: 229.1 on 4 and 299 DF,  p-value: < 2.2e-16
```



However, this model did not pass the tests for the error assumptions (constant variance and normality). In addition, only the New Vehicle CPI and the covid predictor appeared to be significant.

Model 2 (MLR with COVID dummy using WLS)

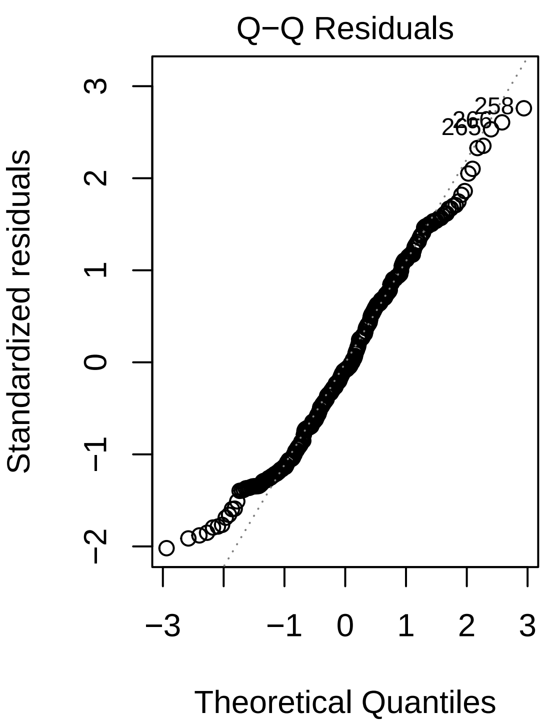
To revise our model, we applied the WLS method to address the heteroscedasticity problem. This approach introduces weights into the model-fitting process so that observations with smaller residuals receive larger weights, which helps stabilize the regression coefficients and address for heteroscedasticity. Similar to the first model, our second model is written in similar form:

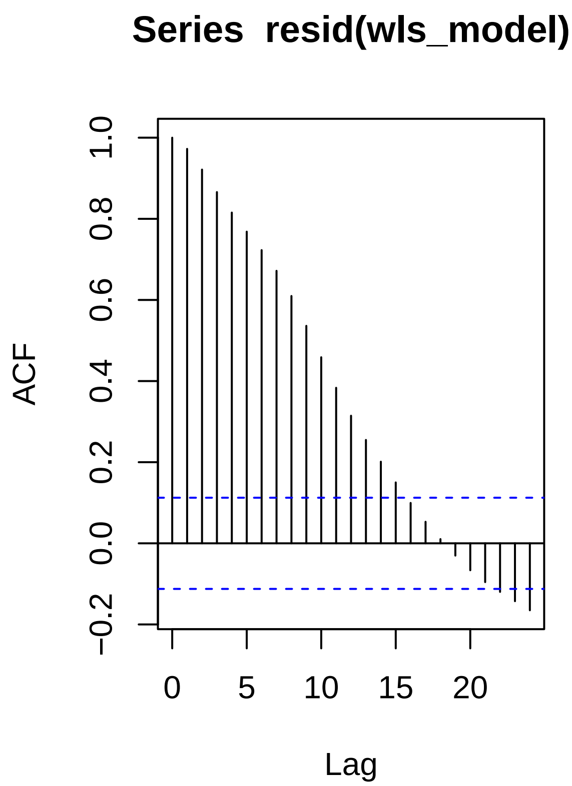
$$\mathrm{CPI}_{\mathrm{used(t)}} = \beta_0 + \beta_1 \mathrm{CPI}_{\mathrm{new(t)}} + \beta_2 \mathrm{CPI}_{\mathrm{fuel(t)}} + \beta_3 \mathrm{FFER}_{(t)} + \beta_4 \mathrm{COVID}_{(t)} + \epsilon$$

```
##
## Call:
## lm(formula = used_cpi ~ new_cpi + fuel_cpi + fedfunds + covid,
##     data = dat, weights = wt)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3925 -0.9062 -0.0985  0.8912  3.3071
##
## Coefficients:
```

```
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.026577  11.735751  -2.388  0.01755 *
## new_cpi       1.215372   0.086073  14.120  < 2e-16 ***
## fuel_cpi     -0.009485   0.008125  -1.167  0.24402
## fedfunds      0.815244   0.252607   3.227  0.00139 **
## covid         6.034583   2.511593   2.403  0.01688 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.214 on 299 degrees of freedom
## Multiple R-squared:  0.6489, Adjusted R-squared:  0.6442
## F-statistic: 138.1 on 4 and 299 DF,  p-value: < 2.2e-16
```

```
##
##  studentized Breusch-Pagan test
##
## data:  wls_model
## BP = 1.3971, df = 4, p-value = 0.8447
```





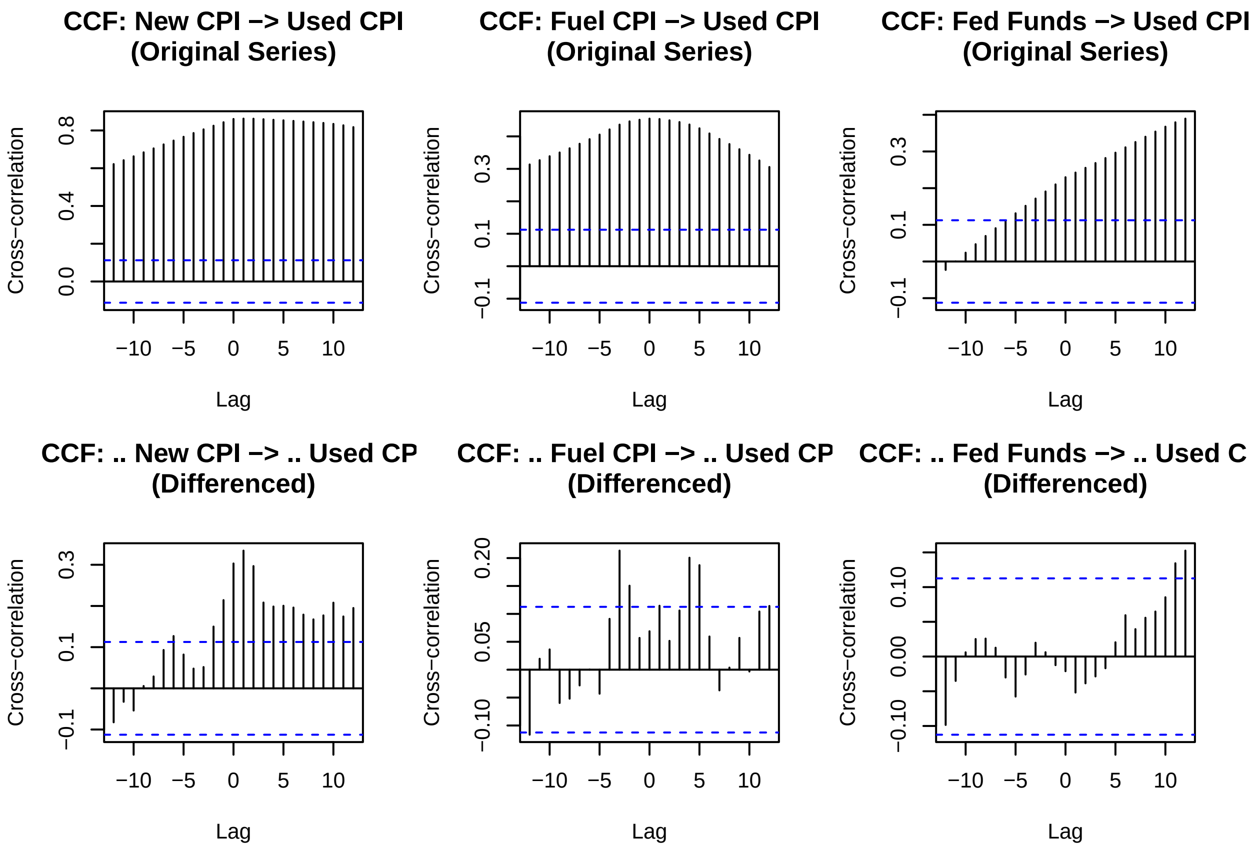
According to the Breusch–Pagan test, the resulting p-value is 0.8447, which is greater than 0.05. Thus, we conclude that the heteroscedasticity problem has been addressed. However, the QQ plot shows that the residual points fail to fall on a straight line, indicating that the residuals still do not satisfy the normality assumption. Besides this, the residual ACF plot suggests that a significant pattern from the graph still exists, which means that further improvement for elimination of this pattern is needed.

Model 3 (MLR with lagged predictors)

To propose a better model, we applied cross-correlation function (CCF) analysis.

```
## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ New CPI → Δ Used CPI' in 'mbcsToSbcs': dot
```

```
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ New CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ New CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ New CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fuel CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fuel CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fuel CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fuel CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fed Funds → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fed Funds → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fed Funds → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fed Funds → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>
```



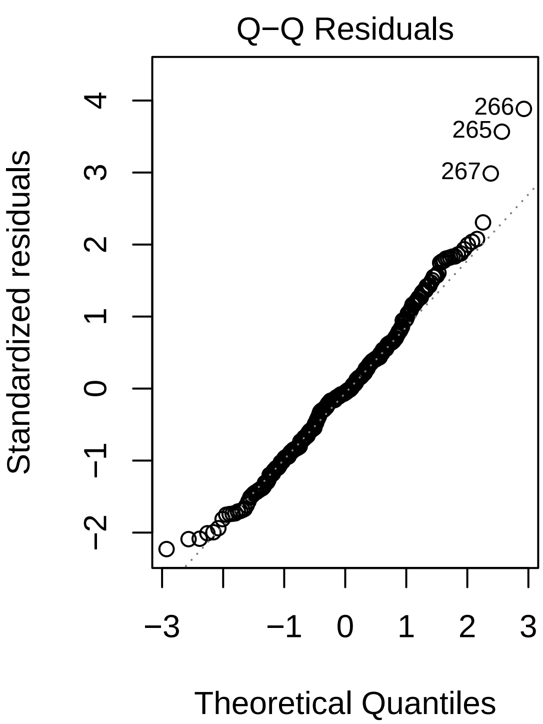

```
## === Optimal Lags from Differenced Series CCF ===
## New CPI: Lag 1 months, Correlation = 0.214
## Fuel CPI: Lag 3 months, Correlation = 0.213
## Fed Funds: Lag 12 months, Correlation = -0.099
```

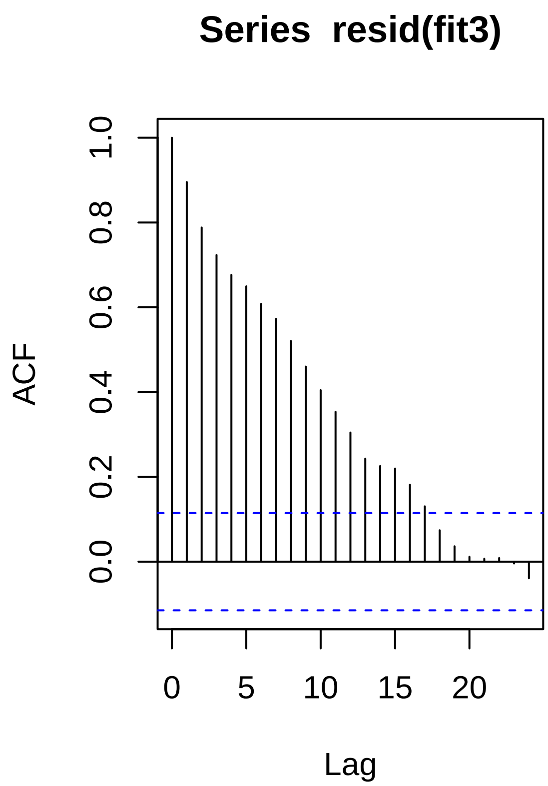
We first processed the data by differencing both the predictor variables and the response variable. After differencing, although not perfect, all series appear closer to being stationary. We then presented the CCF plots using the differenced data for Used Vehicle CPI, Motor Fuel CPI, New Vehicle CPI, and the Federal Funds Effective Rate. Next, we identified the lags at which the CCF values reached their maximum. The results are: lag for New Vehicle CPI = 1; lag for Motor Fuel CPI = 3; lag for Federal Funds Rate = 12. We then refit the model using these lagged, differenced predictors to estimate the differenced Used Vehicle CPI. Our third model is written in the following form:

$$\nabla \mathrm{CPI}_{\mathrm{used(t)}} = \beta_0+\beta_1\nabla \mathrm{CPI}_{\mathrm{new(t\text{-}1)}}+\beta_2\nabla \mathrm{CPI}_{\mathrm{fuel(t\text{-}3)}}+\beta_3\nabla \mathrm{FFER}_{(t-12)}+\beta_4\mathrm{CPI}_{\mathrm{new(t)}}+\beta_5\mathrm{CPI}_{\mathrm{fuel(t)}}+\beta_6\mathrm{FFER}_{(t)}+\epsilon$$

```
##
## Call:
## lm(formula = used_cpi ~ new_cpi + new_cpi_lag1 + fuel_cpi + fuel_cpi_lag3 +
##     fedfunds + fedfunds_lag12, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.290  -6.381  -0.543   5.159  36.020
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -45.65844    7.36980  -6.195 2.03e-09 ***
```

```
## new_cpi          7.05699    0.98794   7.143 7.63e-12 ***
## new_cpi_lag1    -5.73826    0.99988  -5.739 2.43e-08 ***
## fuel_cpi         0.04840    0.02183   2.217   0.0274 *
## fuel_cpi_lag3   -0.04212    0.02139  -1.969   0.0499 *
## fedfunds        -0.77780    0.45426  -1.712   0.0879 .
## fedfunds_lag12   0.96717    0.41370   2.338   0.0201 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.396 on 285 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.7889, Adjusted R-squared:  0.7845
## F-statistic: 177.5 on 6 and 285 DF,  p-value: < 2.2e-16
```





From the graph, we can see that QQ plot indicates that the residuals are having equal variance; however, please note that ACF still shows a pattern which needs to be treated in the next model.

Model 4 (ARIMA Model)

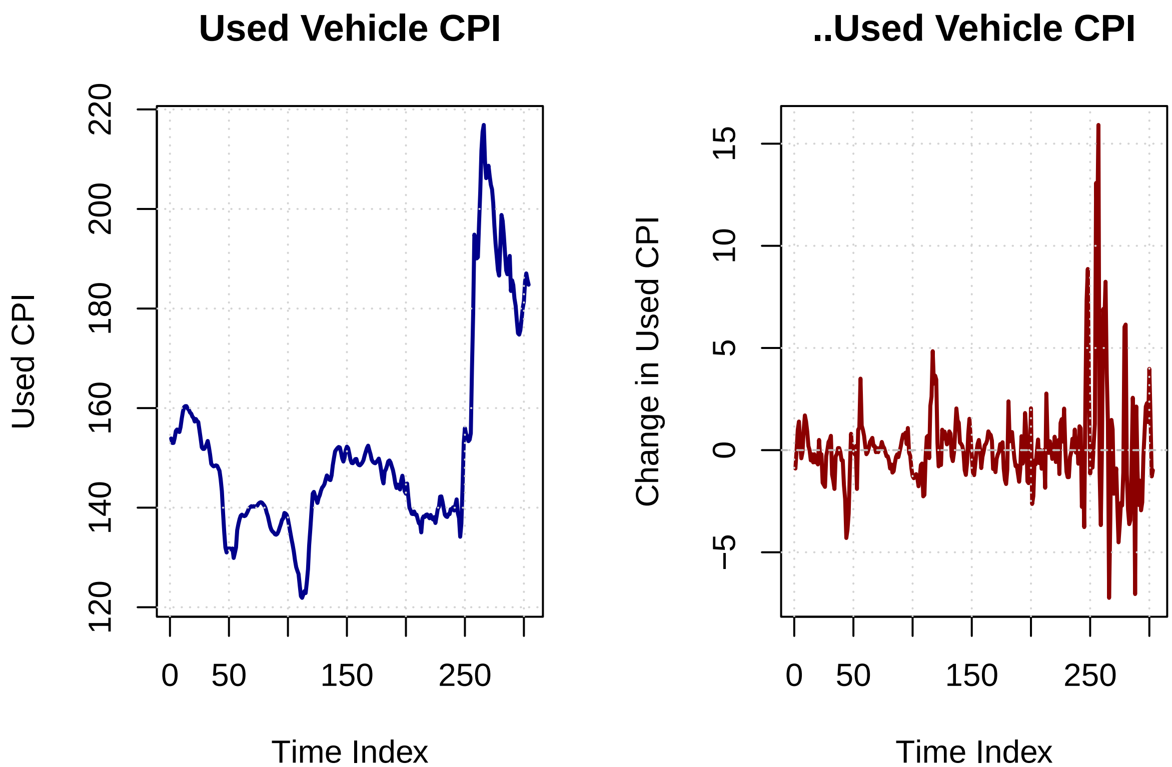
Finally, we applied the ARIMA modeling approach. Before proceeding, we first checked whether the differenced Used Vehicle CPI is stationary. We used the ADF test for this stationarity check. Note that the null hypothesis of the ADF test states that the series is non-stationary.

```
## === ADF Test on Original Series ===

## Test Statistic: -2.349

## P-value: 0.4292

## Conclusion: Series is NON-STATIONARY

## Warning in adf.test(diff_used_full): p-value smaller than printed p-value
```

```
## === ADF Test on Differenced Series ===

## Test Statistic: -4.2475

## P-value: 0.01

## Conclusion: Series is STATIONARY

##
## >>> DECISION: d = 1 (need one difference)
```

According to the test result, the p-value is 0.01, which is less than 0.05. Hence, we reject the null hypothesis and conclude that the Used Vehicle CPI series becomes stationary after regular differencing. Therefore, we set d=1.

```
## Warning in title(...): conversion failure on 'ΔUsed Vehicle CPI' in
## 'mbcsToSbcs': dot substituted for <ce>

## Warning in title(...): conversion failure on 'ΔUsed Vehicle CPI' in
## 'mbcsToSbcs': dot substituted for <94>
```



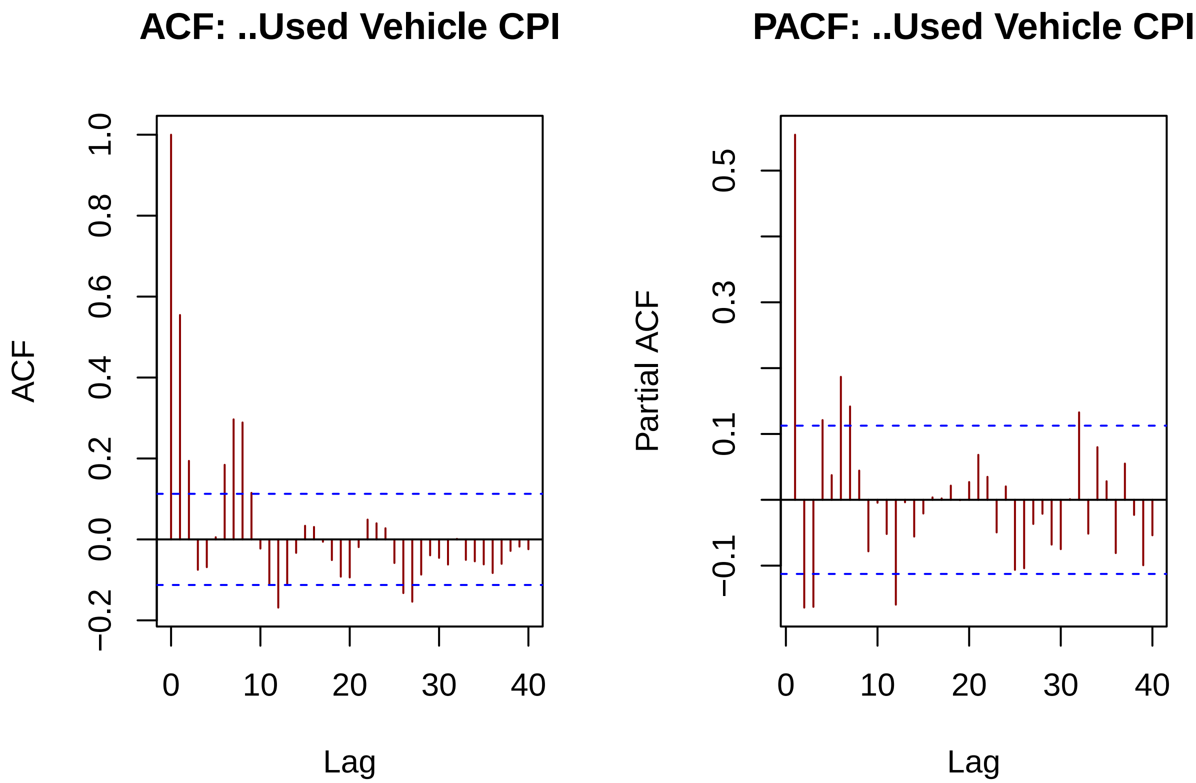
We also displayed the differenced Used Vehicle CPI and observed random fluctuations around the horizontal line at zero, supporting our conclusion from the ADF test.

The next step is to determine suitable values for p and q.

```
## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'ACF: ΔUsed Vehicle CPI' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'ACF: ΔUsed Vehicle CPI' in 'mbcsToSbcs': dot substituted
## for <94>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'PACF: ΔUsed Vehicle CPI' in 'mbcsToSbcs': dot
```

```
## substituted for <ce>
```

```
## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'PACF: ΔUsed Vehicle CPI' in 'mbcsToSbcs': dot
## substituted for <94>
```



We presented the ACF and PACF plots for the series after one regular differencing.

We observe that both the ACF and PACF values decrease quickly and fall within the significance threshold marked by the blue dotted lines. However, none of the ACF or PACF values lie exactly on the zero line. Therefore, we cannot determine precise values for p and q from the plots alone. Instead, we proposed a set of possible p and q values: (p=1, q=0); (p=0, q=1); (p=1, q=1); (p=2, q=0); (p=0, q=2); (p=2, q=2); (p=3, q=0); (p=0, q=3). We then used these proposed sets of values to build the corresponding ARIMA models.

```
##
## === Fit possible ARIMA models ===
```

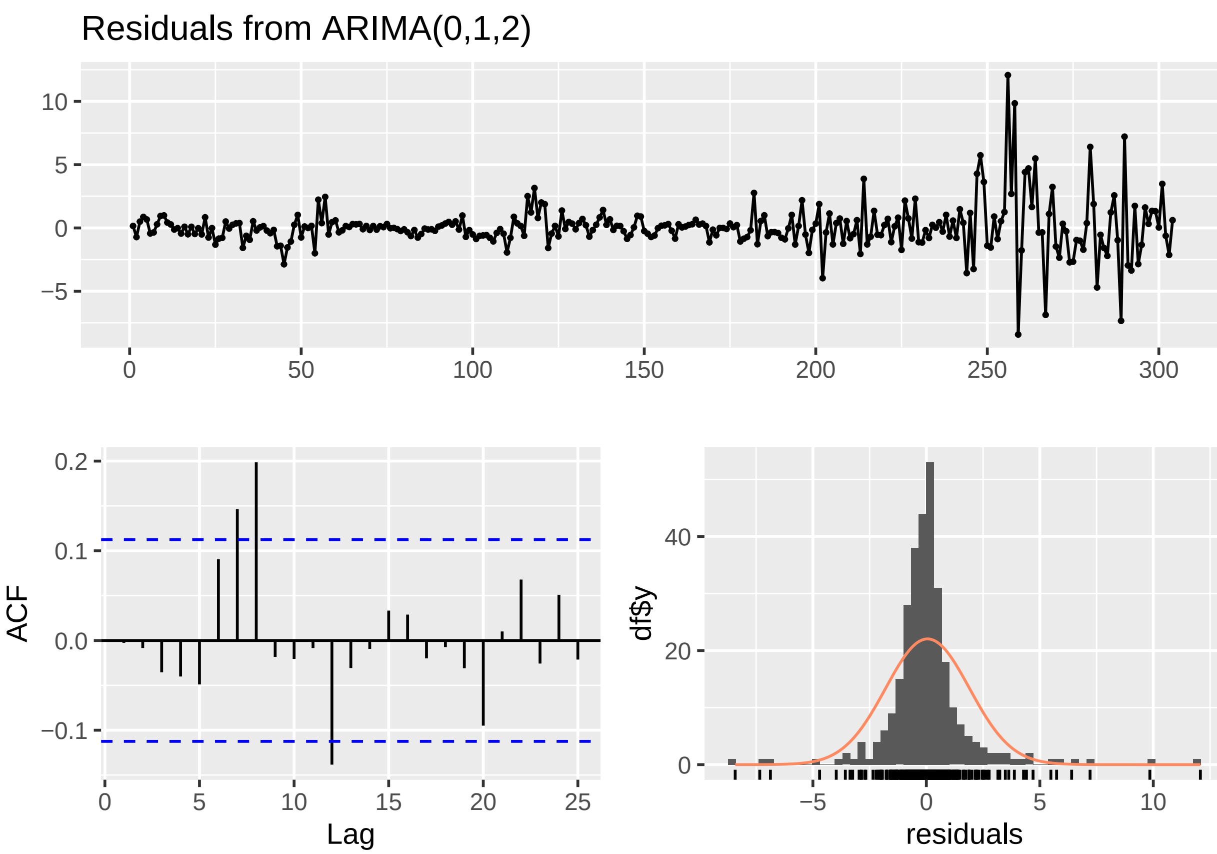
```
##
## ARIMA Model Comparison:
```

```
##          Model      AIC      BIC Log_Likelihood
## 1 ARIMA(0,1,2) 1239.144 1250.286      -616.5722
## 2 ARIMA(0,1,3) 1241.116 1255.970      -616.5578
## 3 ARIMA(3,1,0) 1242.744 1257.599      -617.3719
## 4 ARIMA(2,1,2) 1243.068 1261.637      -616.5339
## 5 ARIMA(2,1,0) 1248.924 1260.065      -621.4618
## 6 ARIMA(1,1,1) 1252.233 1263.374      -623.1166
## 7 ARIMA(1,1,0) 1255.037 1262.464      -625.5184
## 8 ARIMA(0,1,1) 1274.646 1282.073      -635.3230
```

```
##
## *** Most Optimal ARIMA Model: ARIMA(0,1,2) ***
```

```
## Series: dat$used_cpi
## ARIMA(0,1,2)
##
## Coefficients:
##          ma1     ma2
##       0.6444  0.3590
## s.e.  0.0537  0.0541
##
## sigma^2 = 3.445:  log likelihood = -616.57
## AIC=1239.14   AICc=1239.22   BIC=1250.29
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.05120644 1.846776 1.065576 0.0303407 0.671188 0.7972187
##                      ACF1
## Training set -0.002600269
```

Next, we compared their AIC and BIC values to select the most suitable combination of p and q. Based on this comparison, the most appropriate model is ARIMA(p=0, d=1, q=2).

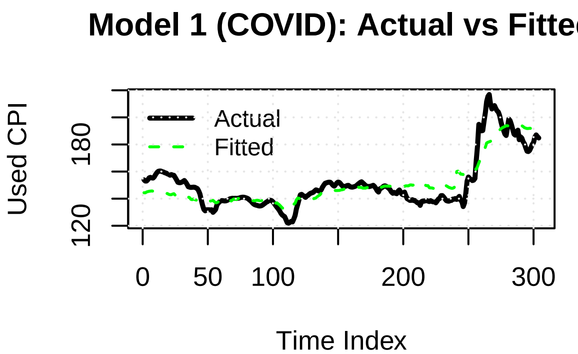


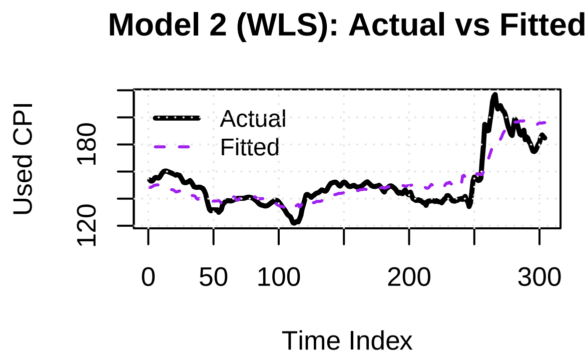

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,2)
## Q* = 23.55, df = 8, p-value = 0.002725
##
## Model df: 2.   Total lags used: 10
```

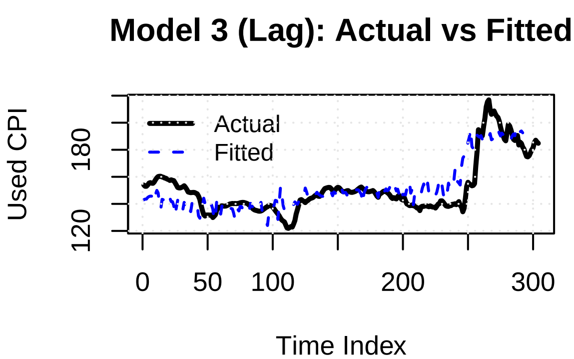
According to the check resiudals result, though the Ljung-Box shows that p-value = $0.002 < 0.05$, expressing there are still weak auto-correlation, but considering that the out of sample MAPE is only 0.67% with high precision. Also, residual ACF plot shows that most lags are within confidence interval. Besides this, the residuals are approximately normal with equal variance. Therefore, we think that ARIMA(0,1,2) is sufficient in practice.

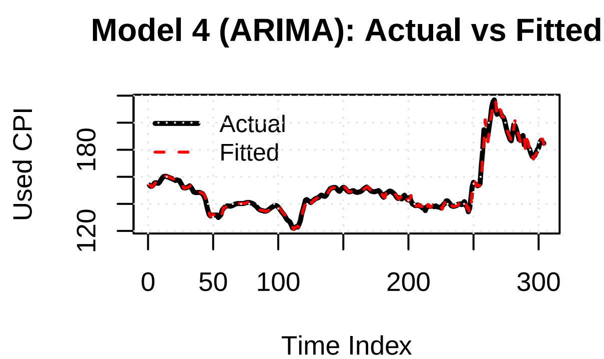
Result

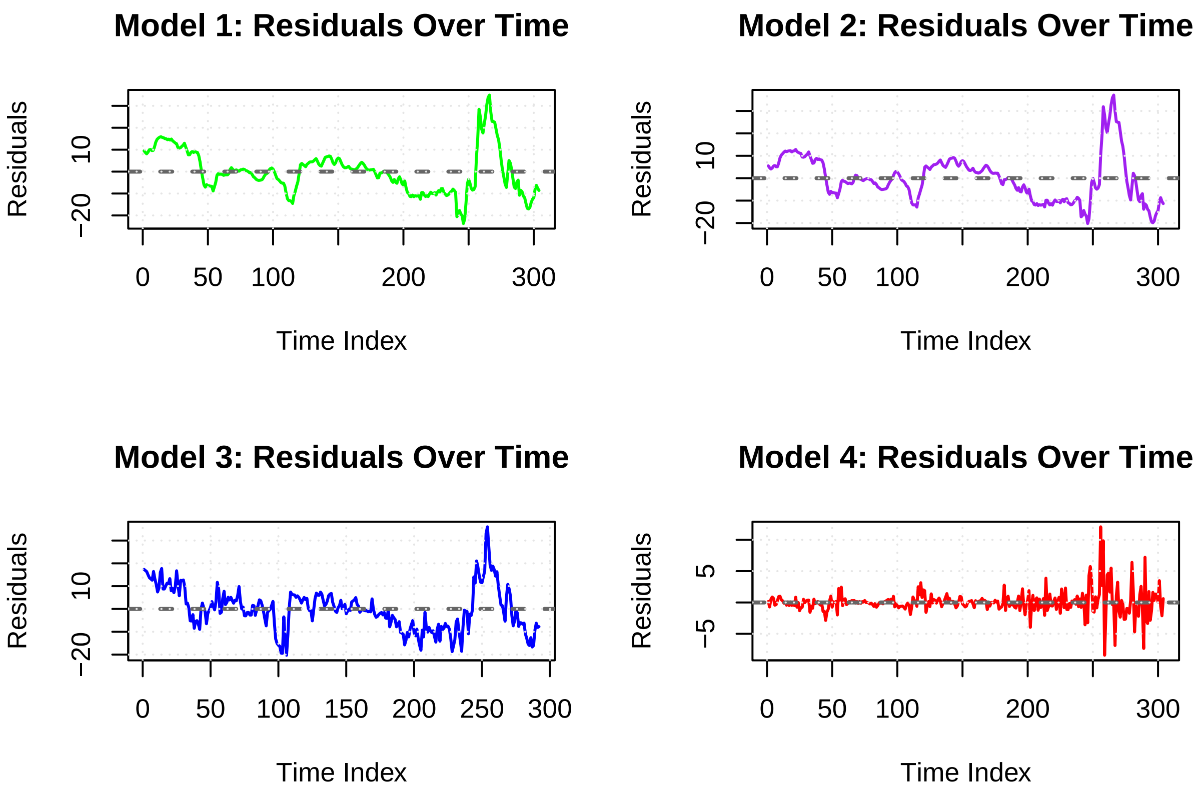
We applied four models—multiple linear regression using OLS, multiple linear regression using WLS, multiple linear regression with lagged predictors, and an ARIMA model—to forecast five future values beyond the last observation in our dataset (after April 2025). The graphical comparisons of the forecast results are displayed below.



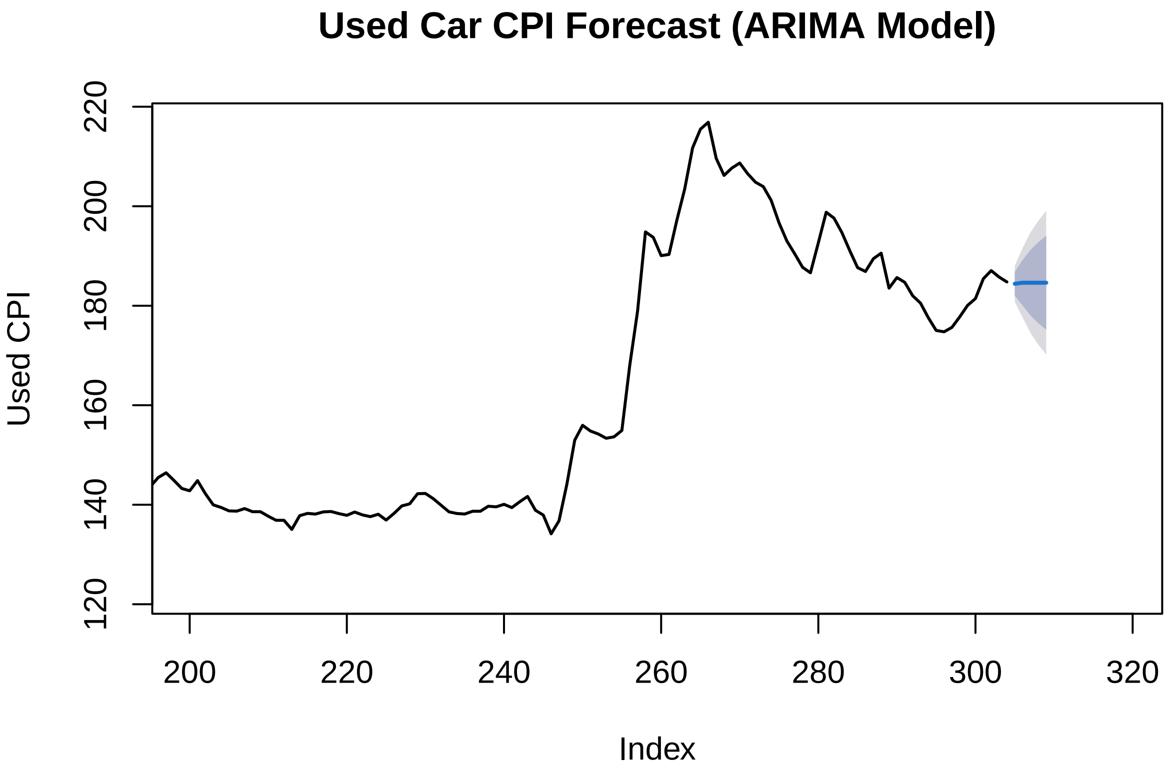




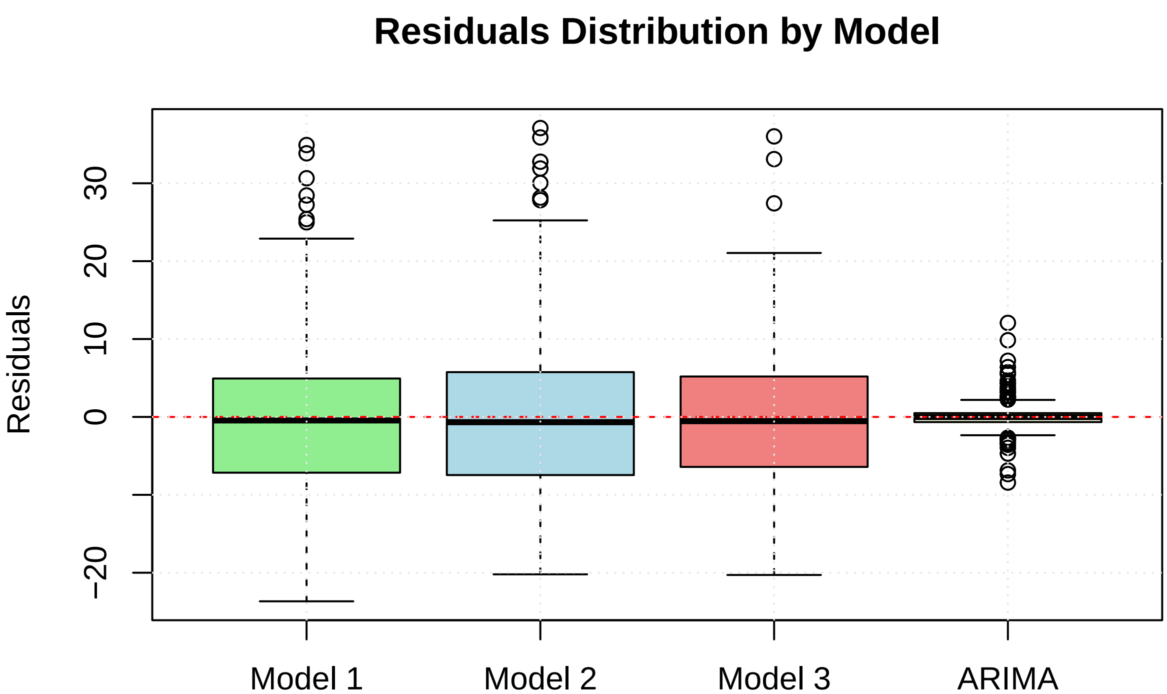





```
##
## === ARIMA Model Forecast ===
##
## 5 future months forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 305       184.4098 182.0313 186.7883 180.7722 188.0474
## 306       184.6285 180.0509 189.2062 177.6277 191.6294
## 307       184.6285 178.0208 191.2362 174.5229 194.7341
## 308       184.6285 176.4818 192.7752 172.1692 197.0878
## 309       184.6285 175.1906 194.0665 170.1944 199.0627
```



Across all four models, the ARIMA model provides forecasts that align most closely with the observed trend and demonstrate the smallest prediction errors. In contrast, the three linear regression models show noticeably poorer predictive performance.



This plot shows that all three linear regression models have wide, skewed residual spreads with big outliers, while the ARIMA model's residuals are much tighter and centered, indicating far better error stability.

Therefore, based on overall accuracy and graphical visualizations, we conclude that the ARIMA(p=0,d=1,q=2) model is the most appropriate model for forecasting Used Vehicle CPI.

Discussion

In this project, our goal was to explore different modeling approaches to estimate and forecast the Used Vehicle CPI. We considered four models: (1) multiple linear regression using OLS, (2) multiple linear regression using WLS, (3) multiple linear regression with lagged predictors, and (4) an ARIMA model.

Our comparison shows that the ARIMA model significantly performs better than all three linear regression models in short-term forecasting. This result is expected, as ARIMA models are specifically designed to capture temporal patterns that traditional linear regression models cannot adequately represent. Among the three proposed linear regression models, incorporating lagged predictors improve the predicting ability of the model.

However, this study has several limitations. First, our analysis relies on a relatively small set of predictors, and additional economic indicators—such as consumer sentiment measures—may further improve model performance. Second, our forecast horizon includes only five future observations; evaluating model performance over a longer horizon may lead to different conclusions. Third, since the ARIMA model does not incorporate contemporaneous relationships among related CPI components, it may limit its interpretability from an economic perspective.

Future work could extend the analysis to more complex linear regression models that can be used for long-term forecasting, possible models can include interaction terms, incorporating more macroeconomic predictors, and perform variable selection to remove redundant variables. We may also add ARIMAX in order to eliminate still-existing autocorrelation from the ARIMA model.

Group Member Contributions

The research question and all methodological decisions were collectively discussed and equally contributed by all three group members.

For the coding parts, Rongsheng mainly contributed to the ARIMA model, while Samuel and Molin mainly contributed to the multiple linear regression model and the determination of suitable lags.

For the written report, Rongsheng mainly contributed to generating graphical outputs and writing their interpretations. Molin mainly contributed to writing introduction, methods, and interpretations. Samuel mainly contributed to writing result, discussion, as well as making slides and presentation drafts.

Citation

Consumer Price Index for All Urban Consumers: Used Cars and Trucks in U.S. City Average (CUSR0000SETA02)

https://fred.stlouisfed.org/series/CUSR0000SETA02

Consumer Price Index for All Urban Consumers: Fuel Oil and Other Fuels in U.S. City Average (CUSR0000SEHE)

https://fred.stlouisfed.org/series/CUSR0000SEHE

Consumer Price Index for All Urban Consumers: New Vehicles in U.S. City Average (CUUR0000SETA01)

https://fred.stlouisfed.org/series/CUUR0000SETA01

Federal Funds Effective Rate (FEDFUNDS)

https://fred.stlouisfed.org/series/FEDFUNDS

Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
```

```r
# load packages
library(tidyverse)
library(lubridate)
library(astsa)
library(MASS)
library(lmtest)
library(forecast)
library(tseries)
library(fpp2)

# train data
used_raw <- read_csv("CUSR0000SETA02.csv") # Used Cars & Trucks CPI
```

```
## Rows: 304 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): CUSR0000SETA02
## date (1): observation_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
new_raw  <- read_csv("CUSR0000SETA01.csv") # New Vehicles CPI
```

```
## Rows: 304 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): CUSR0000SETA01
## date (1): observation_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
fuel_raw <- read_csv("CUSR0000SETB.csv")   # Motor Fuel CPI
```

```
## Rows: 304 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): CUSR0000SETB
## date (1): observation_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
rate_raw <- read_csv("FEDFUNDS.csv")       # Federal Funds Effective Rate
```

```
## Rows: 304 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): FEDFUNDS
## date (1): observation_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# Quick previews
head(used_raw)
```

```
## # A tibble: 6 x 2
##   observation_date CUSR0000SETA02
##   <date>                    <dbl>
## 1 2000-01-01                 154.
## 2 2000-02-01                 153
## 3 2000-03-01                 153
## 4 2000-04-01                 154
## 5 2000-05-01                 155.
## 6 2000-06-01                 156.
```

```r
head(new_raw)
```

```
## # A tibble: 6 x 2
##   observation_date CUSR0000SETA01
##   <date>                    <dbl>
## 1 2000-01-01                 142.
## 2 2000-02-01                 142.
## 3 2000-03-01                 143.
## 4 2000-04-01                 143
## 5 2000-05-01                 143.
## 6 2000-06-01                 143.
```

```r
head(fuel_raw)
```

```
## # A tibble: 6 x 2
##   observation_date CUSR0000SETB
##   <date>                  <dbl>
## 1 2000-01-01               117.
## 2 2000-02-01               121.
## 3 2000-03-01               134
## 4 2000-04-01               126.
## 5 2000-05-01               124.
## 6 2000-06-01               136.
```

```r
head(rate_raw)
```

```
## # A tibble: 6 x 2
##   observation_date FEDFUNDS
##   <date>              <dbl>
## 1 2000-01-01           5.45
## 2 2000-02-01           5.73
## 3 2000-03-01           5.85
## 4 2000-04-01           6.02
## 5 2000-05-01           6.27
## 6 2000-06-01           6.53
```

```r
# test data
used_test_raw <- read_csv("CUSR0000SETA02_test.csv")
```

```
## Rows: 5 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): CUSR0000SETA02
## date (1): observation_date
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
new_test_raw <- read_csv("CUSR0000SETA01_test.csv")
```

```
## Rows: 5 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): CUSR0000SETA01
## date (1): observation_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
fuel_test_raw <- read_csv("CUSR0000SETB_test.csv")
```

```
## Rows: 5 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): CUSR0000SETB
## date (1): observation_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

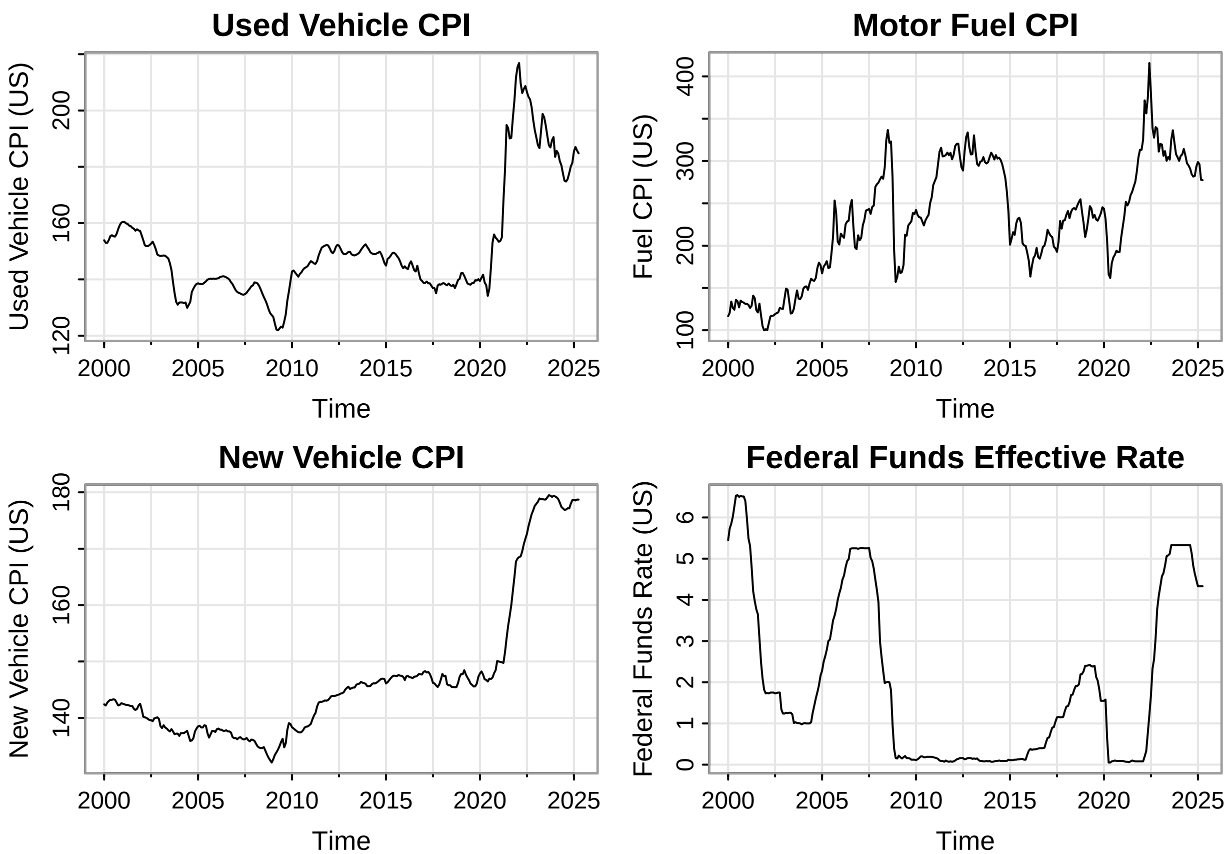
```r
rate_test_raw <- read_csv("FEDFUNDS_test.csv")
```

```
## Rows: 5 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): FEDFUNDS
## date (1): observation_date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# train data cleaning
used <- used_raw %>%
mutate(observation_date = ymd(observation_date)) %>%
filter(observation_date >= ymd("2000-01-01"),
observation_date <= ymd("2025-04-01")) %>%
rename(used_cpi = CUSR0000SETA02)

new <- new_raw %>%
mutate(observation_date = ymd(observation_date)) %>%
filter(observation_date >= ymd("2000-01-01"),
observation_date <= ymd("2025-04-01")) %>%
rename(new_cpi = CUSR0000SETA01)

fuel <- fuel_raw %>%
mutate(observation_date = ymd(observation_date)) %>%
filter(observation_date >= ymd("2000-01-01"),
observation_date <= ymd("2025-04-01")) %>%
rename(fuel_cpi = CUSR0000SETB)

rate <- rate_raw %>%
```

```r
mutate(observation_date = ymd(observation_date)) %>%
filter(observation_date >= ymd("2000-01-01"),
observation_date <= ymd("2025-04-01")) %>%
rename(fedfunds = FEDFUNDS)

# test data cleaning

used_test <- used_test_raw %>%
mutate(observation_date = ymd(observation_date)) %>%
rename(used_cpi = CUSR0000SETA02)

new_test <- new_test_raw %>%
mutate(observation_date = ymd(observation_date)) %>%
rename(new_cpi = CUSR0000SETA01)

fuel_test <- fuel_test_raw %>%
mutate(observation_date = ymd(observation_date)) %>%
rename(fuel_cpi = CUSR0000SETB)

rate_test <- rate_test_raw %>%
mutate(observation_date = ymd(observation_date)) %>%
rename(fedfunds = FEDFUNDS)

# merge data
dat <- used %>%
left_join(new, by = "observation_date") %>%
left_join(fuel, by = "observation_date") %>%
left_join(rate, by = "observation_date") %>%
arrange(observation_date)

dat_test <- used_test %>%
left_join(new_test, by = "observation_date") %>%
left_join(fuel_test, by = "observation_date") %>%
left_join(rate_test, by = "observation_date") %>%
arrange(observation_date)

dat <- dat %>%
mutate(
t = row_number(),
covid = if_else(observation_date >= ymd("2020-01-01"), 1, 0)
)

dat_test <- dat_test %>%
mutate(
t = max(dat$t) + row_number(),
covid = 1
)

used_cpi_ts <- ts(dat$used_cpi, start = c(2000, 1), frequency = 12)
cpi_fuel_ts <- ts(dat$fuel_cpi, start = c(2000, 1), frequency = 12)
new_cpi_ts <- ts(dat$new_cpi, start = c(2000, 1), frequency = 12)
fed_funds_ts <- ts(dat$fedfunds, start = c(2000, 1), frequency = 12)
```

```r
par(mfrow = c(2,2))
tsplot(used_cpi_ts, col = 1, xlab = "Time", ylab = "Used Vehicle CPI (US)",
main = "Used Vehicle CPI")
tsplot(cpi_fuel_ts, col = 1, xlab = "Time", ylab = "Fuel CPI (US)",
main = "Motor Fuel CPI")
tsplot(new_cpi_ts, col = 1, xlab = "Time", ylab = "New Vehicle CPI (US)",
main = "New Vehicle CPI")
tsplot(fed_funds_ts, col = 1, xlab = "Time", ylab = "Federal Funds Rate (US)",
main = "Federal Funds Effective Rate")
```

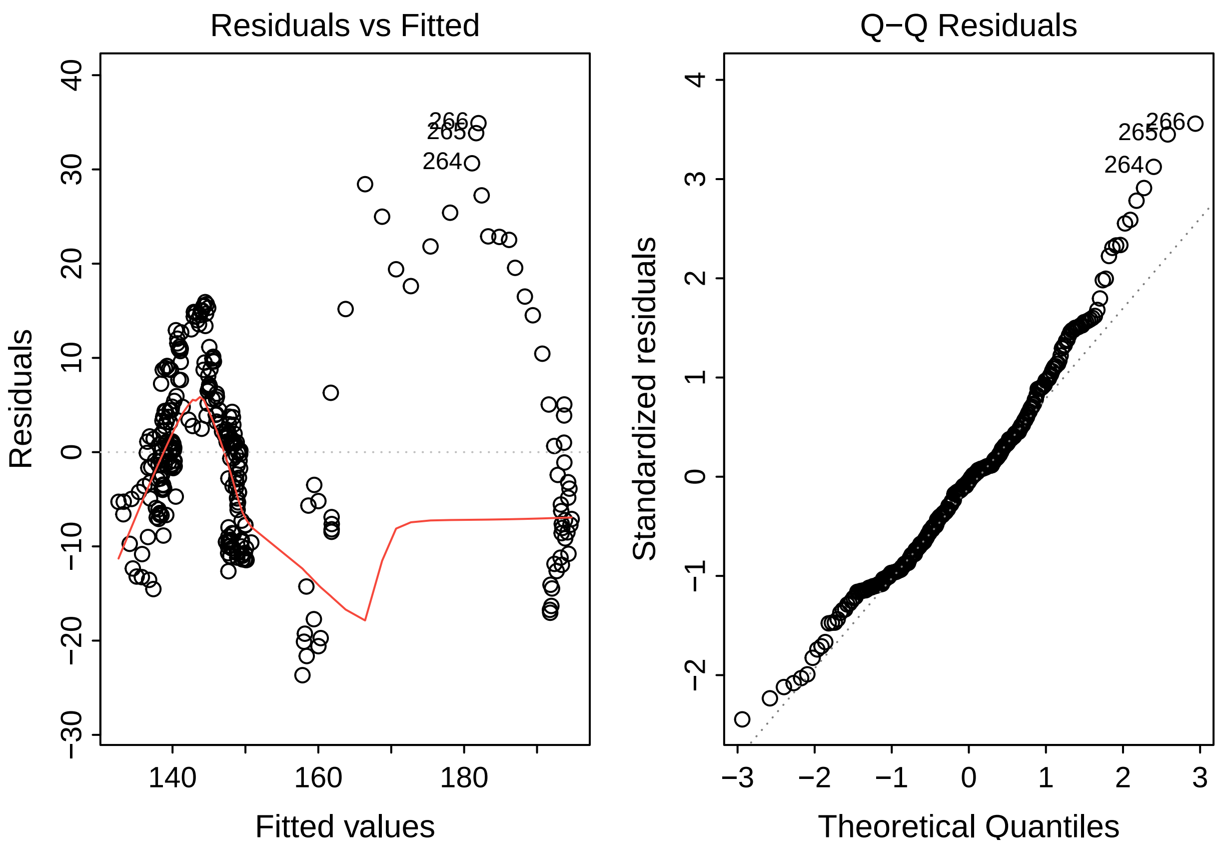



```r
fit1 <- lm(used_cpi ~ new_cpi + fuel_cpi + fedfunds + covid, data = dat)
summary(fit1)
```

```
##
## Call:
## lm(formula = used_cpi ~ new_cpi + fuel_cpi + fedfunds + covid,
##     data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -23.670  -7.127  -0.453   4.875  34.909
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.882950  11.990306  -0.657    0.511
## new_cpi      1.056579   0.091388  11.561  < 2e-16 ***
## fuel_cpi     0.005736   0.010505   0.546    0.585
```

```
## fedfunds     0.212106   0.308909   0.687    0.493
## covid        9.927836   2.515964   3.946 9.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.918 on 299 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.7507
## F-statistic: 229.1 on 4 and 299 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(1, 2))
plot(fit1, which = 1)
plot(fit1, which = 2)
```




```r
wt <- 1 / lm(abs(fit1$residuals) ~ fit1$fitted.values)$fitted.values^2

wls_model <- lm(used_cpi ~ new_cpi + fuel_cpi + fedfunds + covid,
                data = dat,
                weights = wt)

summary(wls_model)
```

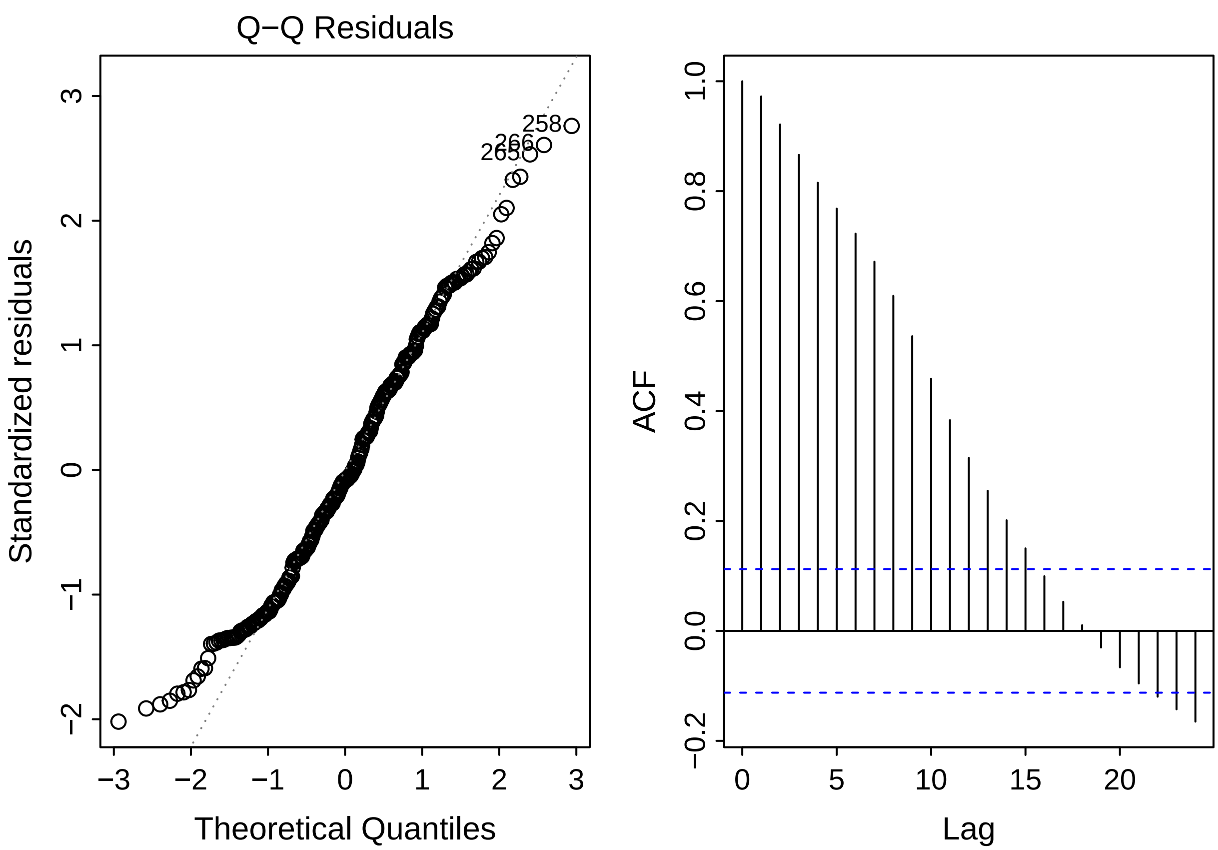
```
##
## Call:
## lm(formula = used_cpi ~ new_cpi + fuel_cpi + fedfunds + covid,
##     data = dat, weights = wt)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.3925 -0.9062 -0.0985  0.8912  3.3071
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28.026577  11.735751  -2.388  0.01755 *
## new_cpi       1.215372   0.086073  14.120  < 2e-16 ***
## fuel_cpi     -0.009485   0.008125  -1.167  0.24402
## fedfunds      0.815244   0.252607   3.227  0.00139 **
## covid         6.034583   2.511593   2.403  0.01688 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.214 on 299 degrees of freedom
## Multiple R-squared:  0.6489, Adjusted R-squared:  0.6442
## F-statistic: 138.1 on 4 and 299 DF,  p-value: < 2.2e-16
```

```
bptest(wls_model)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  wls_model
## BP = 1.3971, df = 4, p-value = 0.8447
```

```
par(mfrow = c(1, 2))
plot(wls_model, which = 2)
acf(resid(wls_model))
```



```r
diff_used <- diff(dat$used_cpi)
diff_new <- diff(dat$new_cpi)
diff_fuel <- diff(dat$fuel_cpi)
diff_rate <- diff(dat$fedfunds)

par(mfrow = c(2, 3))
# Original series CCF
ccf(dat$new_cpi, dat$used_cpi, lag.max = 12,
main = "CCF: New CPI → Used CPI\n(Original Series)",
ylab = "Cross-correlation")

ccf(dat$fuel_cpi, dat$used_cpi, lag.max = 12,
main = "CCF: Fuel CPI → Used CPI\n(Original Series)",
ylab = "Cross-correlation")

ccf(dat$fedfunds, dat$used_cpi, lag.max = 12,
main = "CCF: Fed Funds → Used CPI\n(Original Series)",
ylab = "Cross-correlation")
# Differenced series CCF (more reliable)

ccf_new_diff <- ccf(diff_new, diff_used, lag.max = 12,
main = "CCF: Δ New CPI → Δ Used CPI\n(Differenced)",
ylab = "Cross-correlation")
```

```
## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ New CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ New CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ New CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ New CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>
```

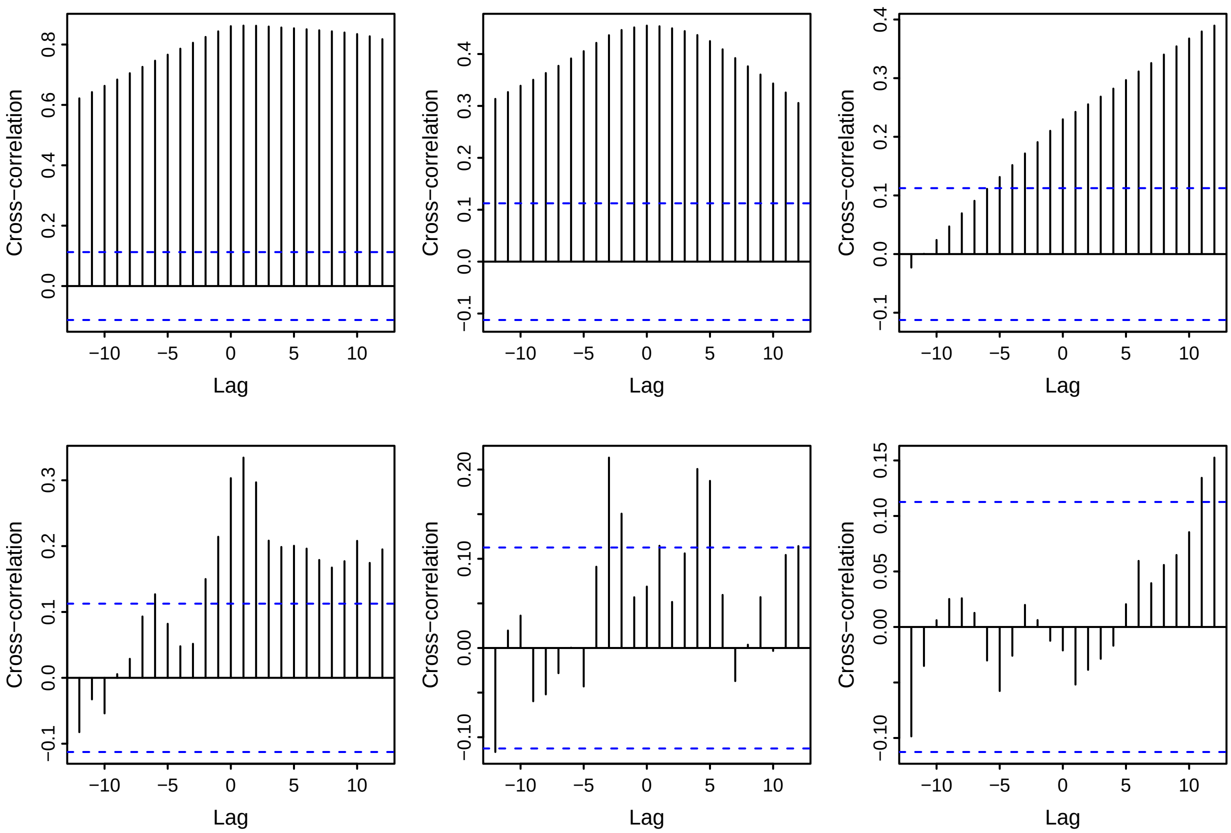
```r
ccf_fuel_diff <- ccf(diff_fuel, diff_used, lag.max = 12,
main = "CCF: Δ Fuel CPI → Δ Used CPI\n(Differenced)",
ylab = "Cross-correlation")
```

```
## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fuel CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fuel CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fuel CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
```

```
## conversion failure on 'CCF: Δ Fuel CPI → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>
```

```r
ccf_rate_diff <- ccf(diff_rate, diff_used, lag.max = 12,
main = "CCF: Δ Fed Funds → Δ Used CPI\n(Differenced)",
ylab = "Cross-correlation")
```

```
## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fed Funds → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fed Funds → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fed Funds → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'CCF: Δ Fed Funds → Δ Used CPI' in 'mbcsToSbcs': dot
## substituted for <94>
```




```r
find_best_lag <- function(ccf_obj, max_neg_lag = 12) {
correlations <- ccf_obj$acf[,,1]
lags <- ccf_obj$lag[,,1]
neg_idx <- which(lags < 0 & lags >= -max_neg_lag)

if(length(neg_idx) > 0) {
  best_idx <- neg_idx[which.max(abs(correlations[neg_idx]))]
```

```r
  return(list(lag = abs(lags[best_idx]),
  correlation = correlations[best_idx]))
}
  return(list(lag = 0, correlation = correlations[lags == 0]))
}

lag_new_diff <- find_best_lag(ccf_new_diff)
lag_fuel_diff <- find_best_lag(ccf_fuel_diff)
lag_rate_diff <- find_best_lag(ccf_rate_diff)

cat("=== Optimal Lags from Differenced Series CCF ===\n")
```

```
## === Optimal Lags from Differenced Series CCF ===
```

```r
cat("New CPI: Lag", lag_new_diff$lag, "months, Correlation =",
round(lag_new_diff$correlation, 3), "\n")
```

```
## New CPI: Lag 1 months, Correlation = 0.214
```

```r
cat("Fuel CPI: Lag", lag_fuel_diff$lag, "months, Correlation =",
round(lag_fuel_diff$correlation, 3), "\n")
```

```
## Fuel CPI: Lag 3 months, Correlation = 0.213
```

```r
cat("Fed Funds: Lag", lag_rate_diff$lag, "months, Correlation =",
round(lag_rate_diff$correlation, 3), "\n")
```

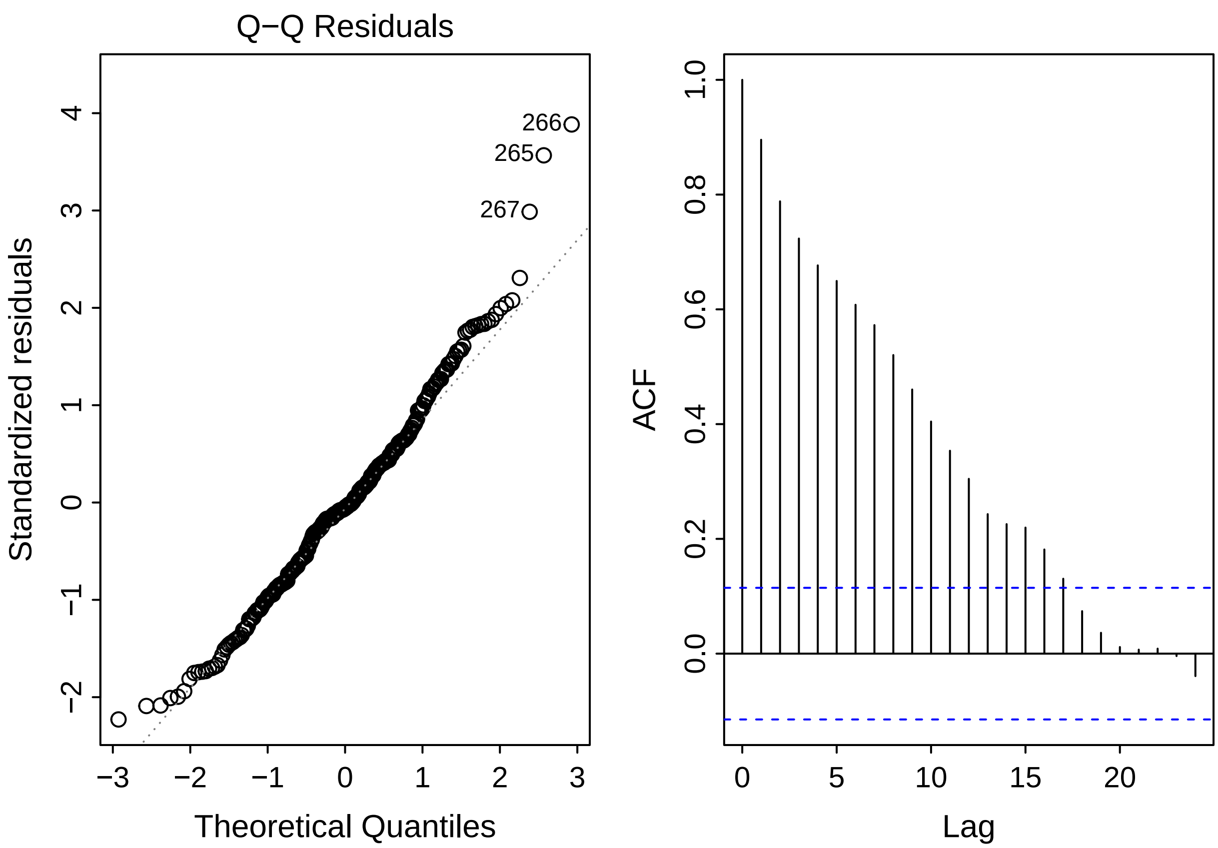
```
## Fed Funds: Lag 12 months, Correlation = -0.099
```

```r
dat <- dat %>%
mutate(
new_cpi_lag1 = lag(new_cpi, 1),
fuel_cpi_lag3 = lag(fuel_cpi, 3),
fedfunds_lag12 = lag(fedfunds, 12)
)

fit3 <- lm(used_cpi ~ new_cpi + new_cpi_lag1 + fuel_cpi + fuel_cpi_lag3+ fedfunds + fedfunds_lag12, data

summary(fit3)
```

```
##
## Call:
## lm(formula = used_cpi ~ new_cpi + new_cpi_lag1 + fuel_cpi + fuel_cpi_lag3 +
##     fedfunds + fedfunds_lag12, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.290  -6.381  -0.543   5.159  36.020
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -45.65844    7.36980  -6.195 2.03e-09 ***
## new_cpi          7.05699    0.98794   7.143 7.63e-12 ***
## new_cpi_lag1    -5.73826    0.99988  -5.739 2.43e-08 ***
## fuel_cpi         0.04840    0.02183   2.217   0.0274 *
## fuel_cpi_lag3   -0.04212    0.02139  -1.969   0.0499 *
## fedfunds        -0.77780    0.45426  -1.712   0.0879 .
```

```
## fedfunds_lag12   0.96717    0.41370   2.338   0.0201 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.396 on 285 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.7889, Adjusted R-squared:  0.7845
## F-statistic: 177.5 on 6 and 285 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(1, 2))
plot(fit3, which = 2)
acf(resid(fit3))
```




```r
adf_original <- adf.test(dat$used_cpi)
cat("=== ADF Test on Original Series ===\n")
```

```
## === ADF Test on Original Series ===
```

```r
cat("Test Statistic:", round(adf_original$statistic, 4), "\n")
```

```
## Test Statistic: -2.349
```

```r
cat("P-value:", round(adf_original$p.value, 4), "\n")
```

```
## P-value: 0.4292
```

```r
cat("Conclusion: Series is",
ifelse(adf_original$p.value > 0.05, "NON-STATIONARY", "STATIONARY"),
"\n\n")
```

```
## Conclusion: Series is NON-STATIONARY
```

```r
diff_used_full <- diff(dat$used_cpi)
adf_diff <- adf.test(diff_used_full)
```

```
## Warning in adf.test(diff_used_full): p-value smaller than printed p-value
```

```r
cat("=== ADF Test on Differenced Series ===\n")
```

```
## === ADF Test on Differenced Series ===
```

```r
cat("Test Statistic:", round(adf_diff$statistic, 4), "\n")
```

```
## Test Statistic: -4.2475
```

```r
cat("P-value:", round(adf_diff$p.value, 4), "\n")
```

```
## P-value: 0.01
```

```r
cat("Conclusion: Series is",
ifelse(adf_diff$p.value > 0.05, "NON-STATIONARY", "STATIONARY"),
"\n")
```

```
## Conclusion: Series is STATIONARY
```

```r
cat("\n>>> DECISION: d = 1 (need one difference)\n")
```

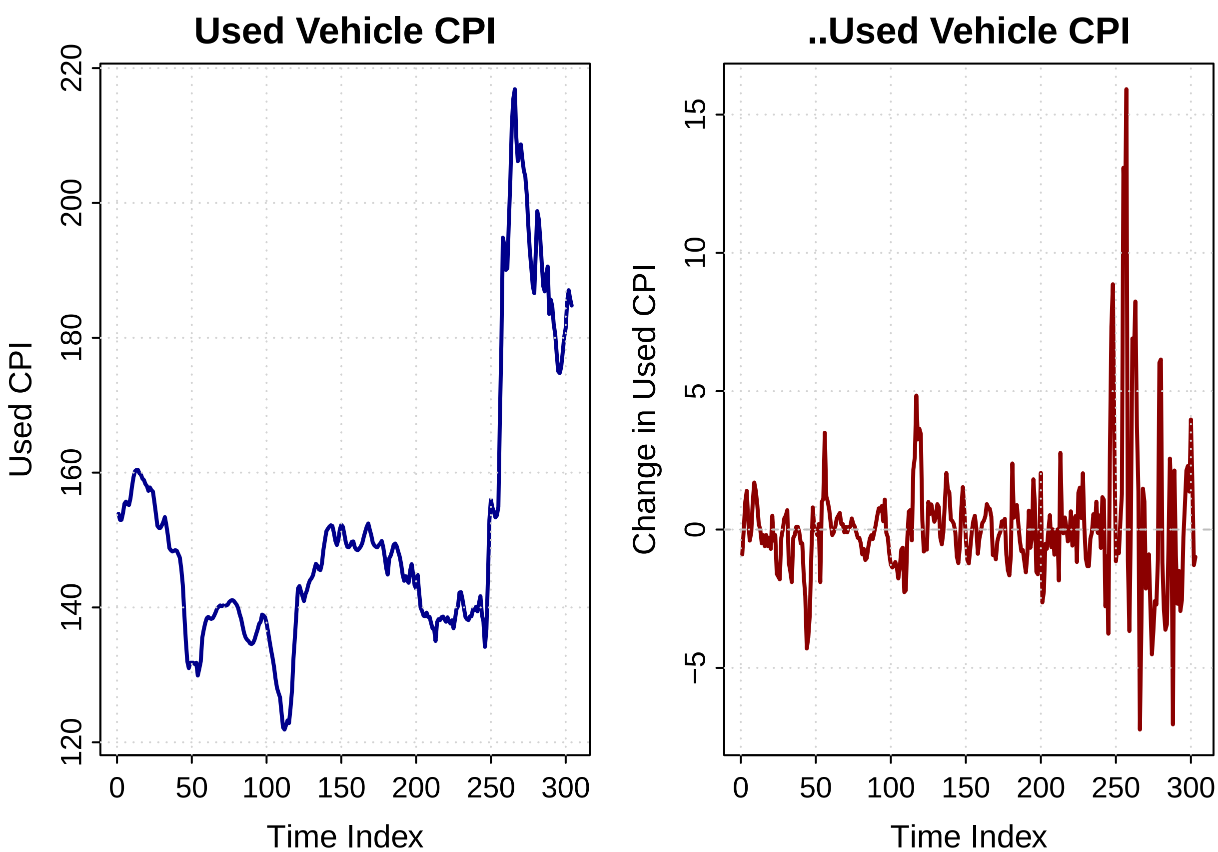
```
##
## >>> DECISION: d = 1 (need one difference)
```

```r
par(mfrow = c(1,2))
plot(dat$used_cpi, type = "l", lwd = 2,
main = "Used Vehicle CPI",
ylab = "Used CPI", xlab = "Time Index",
col = "darkblue")
grid()

diff_used_full <- diff(dat$used_cpi)
plot(diff_used_full, type = "l", lwd = 2,
main = "ΔUsed Vehicle CPI",
ylab = "Change in Used CPI", xlab = "Time Index",
col = "darkred")
```

```
## Warning in title(...): conversion failure on 'ΔUsed Vehicle CPI' in
## 'mbcsToSbcs': dot substituted for <ce>
```

```
## Warning in title(...): conversion failure on 'ΔUsed Vehicle CPI' in
## 'mbcsToSbcs': dot substituted for <94>
```

```r
grid()
abline(h = 0, col = "gray", lty = 2)
```

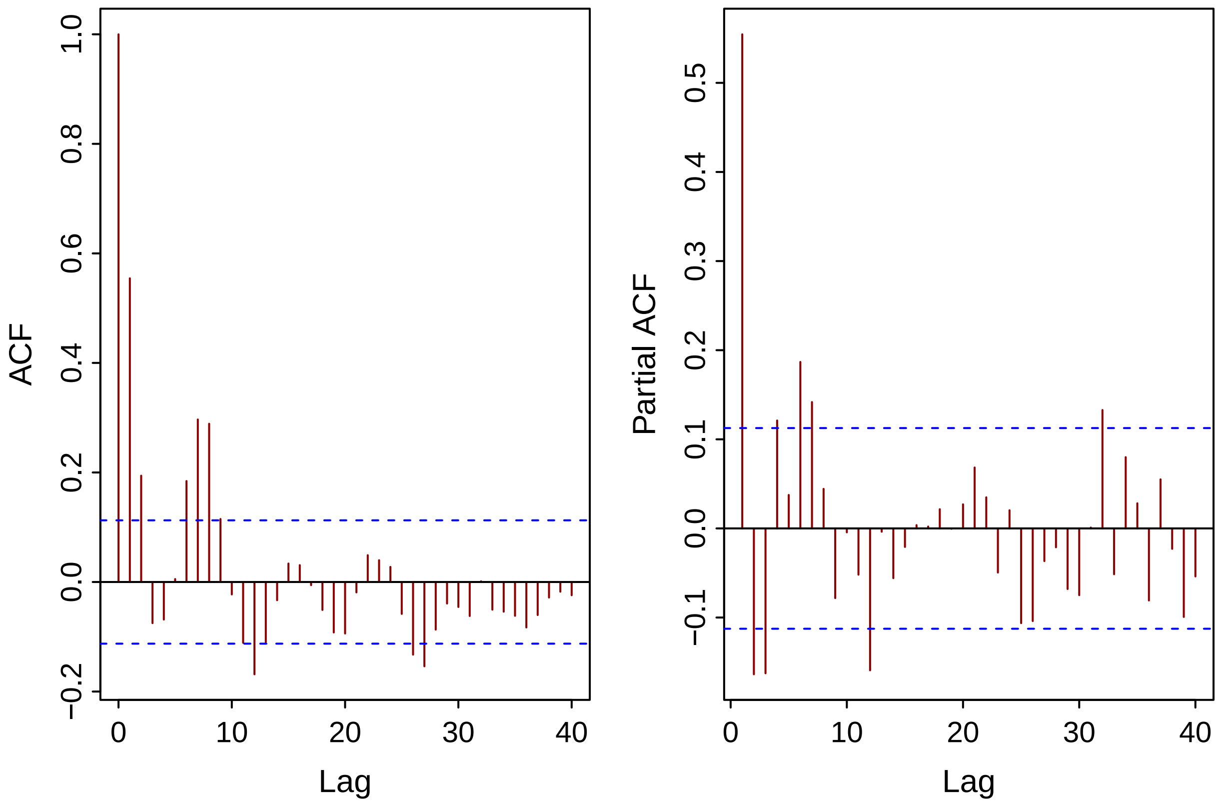



```r
par(mfrow = c(1, 2))

# ACF of differenced
acf(diff_used_full, main = "ACF: ΔUsed Vehicle CPI", lag.max = 40,
col = "darkred")
```

```
## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'ACF: ΔUsed Vehicle CPI' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'ACF: ΔUsed Vehicle CPI' in 'mbcsToSbcs': dot substituted
## for <94>
```

```r
# PACF of differenced
pacf(diff_used_full, main = "PACF: ΔUsed Vehicle CPI", lag.max = 40,
col = "darkred")
```

```
## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'PACF: ΔUsed Vehicle CPI' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in title(main %||% if (i == j) snames[i] else paste(sn.abbr[i], :
## conversion failure on 'PACF: ΔUsed Vehicle CPI' in 'mbcsToSbcs': dot
## substituted for <94>
```




```
cat("\n=== Fit possible ARIMA models ===\n")
```

```
##
## === Fit possible ARIMA models ===
```

```
candidate_models <- list(
  "ARIMA(1,1,0)" = c(1,1,0),
  "ARIMA(0,1,1)" = c(0,1,1),
  "ARIMA(1,1,1)" = c(1,1,1),
  "ARIMA(2,1,0)" = c(2,1,0),
  "ARIMA(0,1,2)" = c(0,1,2),
  "ARIMA(2,1,2)" = c(2,1,2),
  "ARIMA(3,1,0)" = c(3,1,0),
  "ARIMA(0,1,3)" = c(0,1,3)
)

arima_comparison <- data.frame(
  Model = character(),
  AIC = numeric(),
  BIC = numeric(),
  Log_Likelihood = numeric(),
  stringsAsFactors = FALSE
)

for(model_name in names(candidate_models)) {
  order_params <- candidate_models[[model_name]]

  tryCatch({
    fit <- Arima(dat$used_cpi, order = order_params)
```

```r
    arima_comparison <- rbind(arima_comparison, data.frame(
      Model = model_name,
      AIC = AIC(fit),
      BIC = BIC(fit),
      Log_Likelihood = fit$loglik
    ))
  }, error = function(e) {
    cat("Model", model_name, "Fail\n")
  })
}


arima_comparison <- arima_comparison %>% arrange(AIC)
cat("\nARIMA Model Comparison:\n")
```

```
##
## ARIMA Model Comparison:
```

```r
print(arima_comparison)
```

```
##          Model      AIC      BIC Log_Likelihood
## 1 ARIMA(0,1,2) 1239.144 1250.286      -616.5722
## 2 ARIMA(0,1,3) 1241.116 1255.970      -616.5578
## 3 ARIMA(3,1,0) 1242.744 1257.599      -617.3719
## 4 ARIMA(2,1,2) 1243.068 1261.637      -616.5339
## 5 ARIMA(2,1,0) 1248.924 1260.065      -621.4618
## 6 ARIMA(1,1,1) 1252.233 1263.374      -623.1166
## 7 ARIMA(1,1,0) 1255.037 1262.464      -625.5184
## 8 ARIMA(0,1,1) 1274.646 1282.073      -635.3230
```

```r
best_arima_name <- arima_comparison$Model[1]
best_order <- candidate_models[[best_arima_name]]
fit_arima_best <- Arima(dat$used_cpi, order = best_order)

cat("\n*** Most Optimal ARIMA Model:", best_arima_name, "***\n")
```

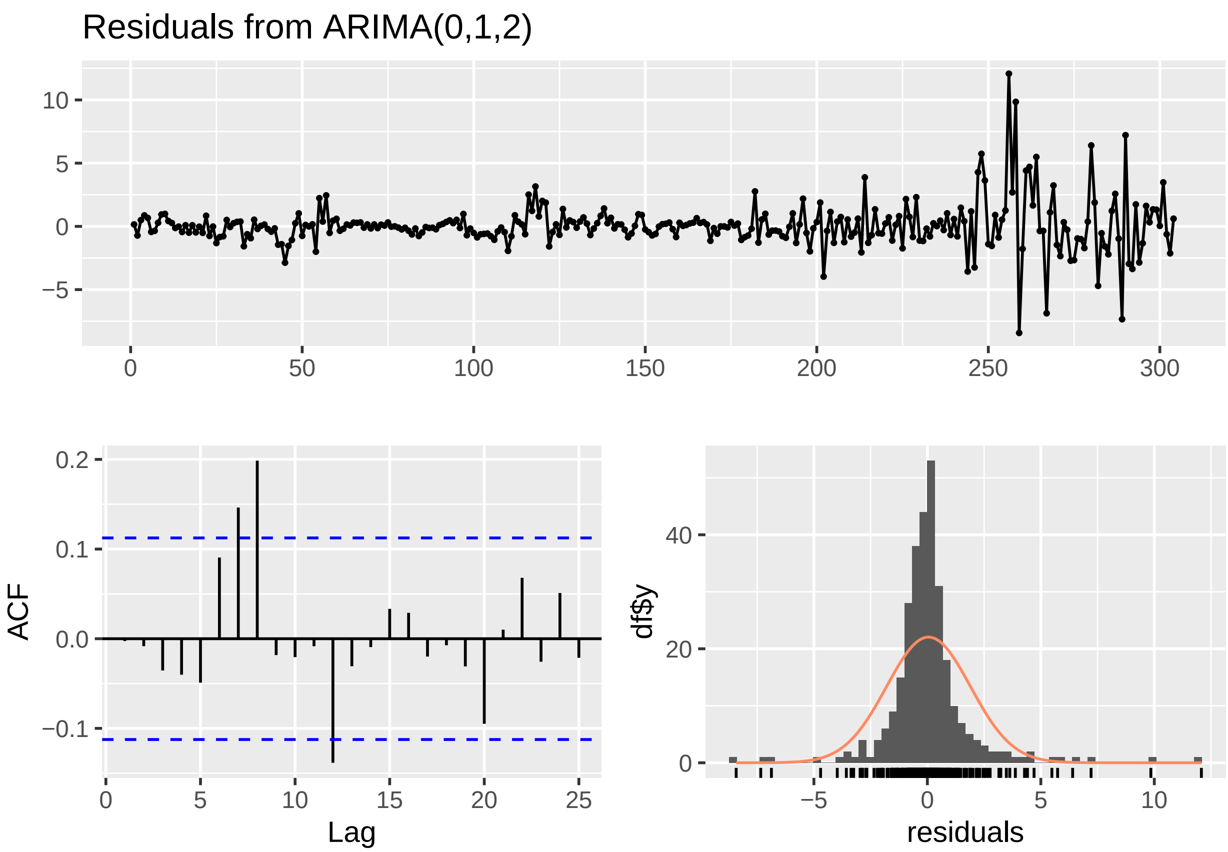
```
##
## *** Most Optimal ARIMA Model: ARIMA(0,1,2) ***
```

```r
print(summary(fit_arima_best))
```

```
## Series: dat$used_cpi
## ARIMA(0,1,2)
##
## Coefficients:
##          ma1     ma2
##       0.6444  0.3590
## s.e.  0.0537  0.0541
##
## sigma^2 = 3.445:  log likelihood = -616.57
## AIC=1239.14   AICc=1239.22   BIC=1250.29
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 0.05120644 1.846776 1.065576 0.0303407 0.671188 0.7972187
```

```
##                          ACF1
## Training set -0.002600269
```

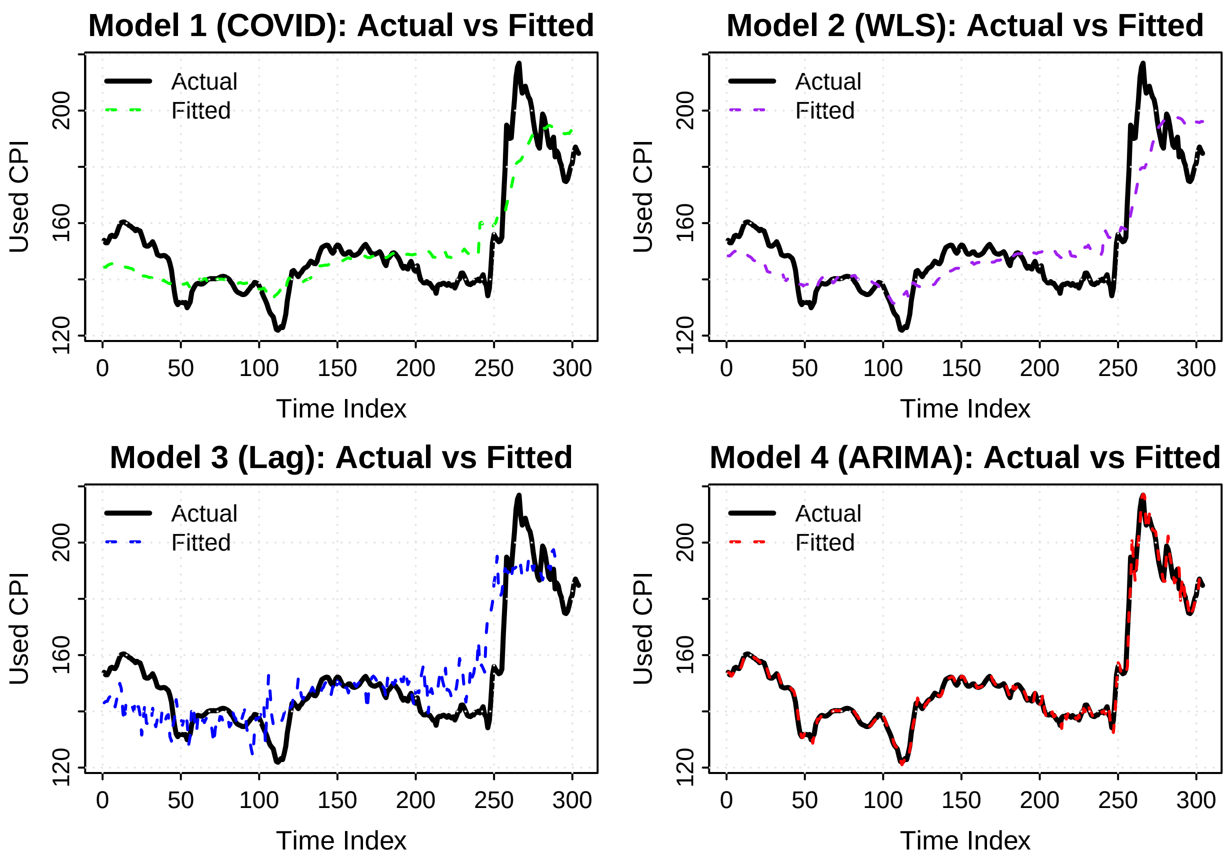
```r
checkresiduals(fit_arima_best)
```



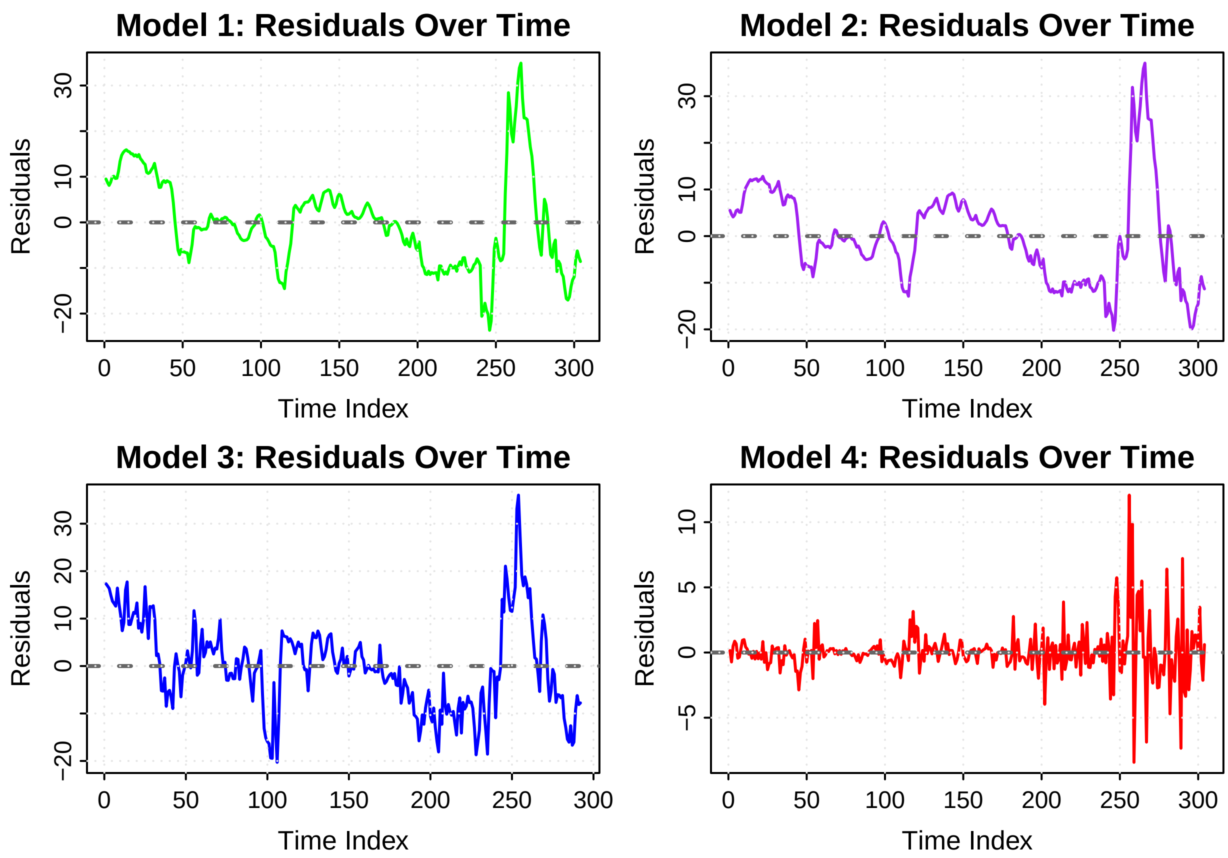

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,2)
## Q* = 23.55, df = 8, p-value = 0.002725
##
## Model df: 2.   Total lags used: 10
```

```r
par(mfrow = c(2,2))

arima_fitted <- fitted(fit_arima_best)
lm_fitted_m2 <- fitted(fit1)
lm_fitted_m3 <- fitted(wls_model)
lm_fitted_m4 <- fitted(fit3)

plot(dat$used_cpi, type = "l", lwd = 2.5, col = "black",
main = "Model 1 (COVID): Actual vs Fitted",
ylab = "Used CPI", xlab = "Time Index")
lines(lm_fitted_m2, col = "green", lwd = 1.5, lty = 2)
legend("topleft",
legend = c("Actual", "Fitted"),
col = c("black", "green"),
lwd = c(2.5, 1.5),
```

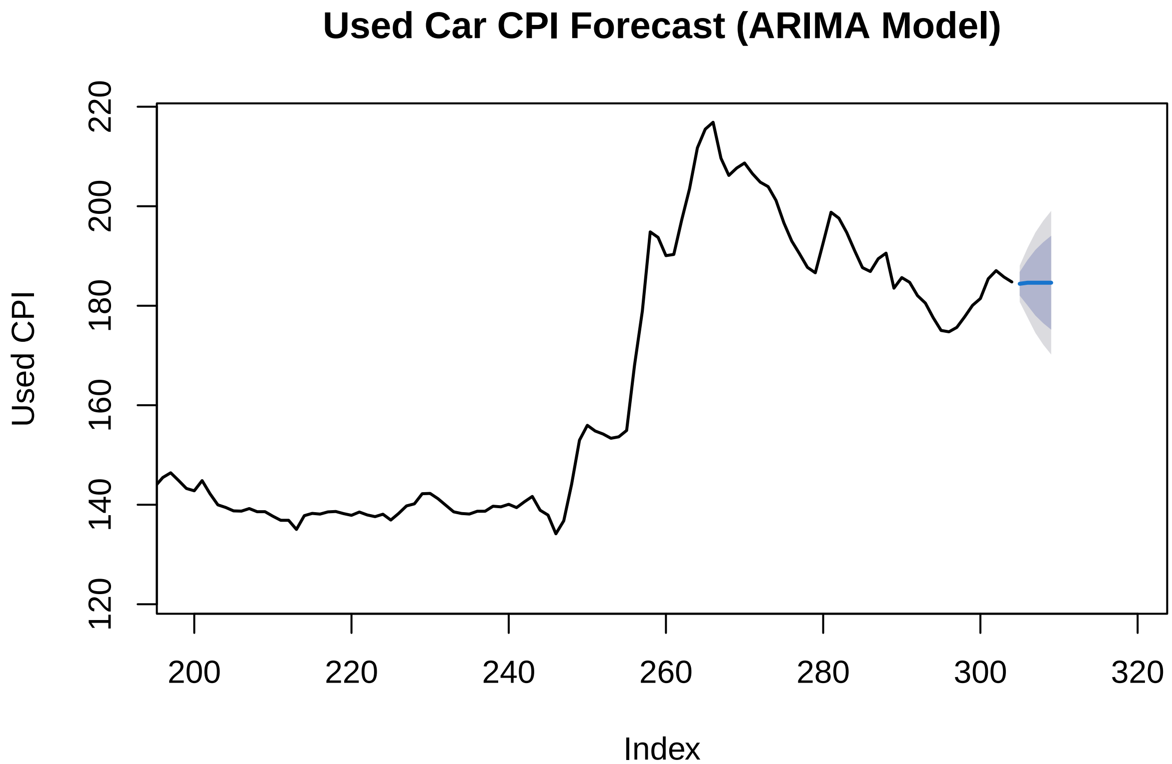
```r
lty = c(1, 2),
bty = "n", cex = 0.9)
grid(col = "gray90")

plot(dat$used_cpi, type = "l", lwd = 2.5, col = "black",
main = "Model 2 (WLS): Actual vs Fitted",
ylab = "Used CPI", xlab = "Time Index")
lines(lm_fitted_m3, col = "purple", lwd = 1.5, lty = 2)
legend("topleft",
legend = c("Actual", "Fitted"),
col = c("black", "purple"),
lwd = c(2.5, 1.5),
lty = c(1, 2),
bty = "n", cex = 0.9)
grid(col = "gray90")

plot(dat$used_cpi, type = "l", lwd = 2.5, col = "black",
main = "Model 3 (Lag): Actual vs Fitted",
ylab = "Used CPI", xlab = "Time Index")
lines(lm_fitted_m4, col = "blue", lwd = 1.5, lty = 2)
legend("topleft",
legend = c("Actual", "Fitted"),
col = c("black", "blue"),
lwd = c(2.5, 1.5),
lty = c(1, 2),
bty = "n", cex = 0.9)
grid(col = "gray90")

plot(dat$used_cpi, type = "l", lwd = 2.5, col = "black",
main = "Model 4 (ARIMA): Actual vs Fitted",
ylab = "Used CPI", xlab = "Time Index")
lines(arima_fitted, col = "red", lwd = 1.5, lty = 2)
legend("topleft",
legend = c("Actual", "Fitted"),
col = c("black", "red"),
lwd = c(2.5, 1.5),
lty = c(1, 2),
bty = "n", cex = 0.9)
grid(col = "gray90")
```



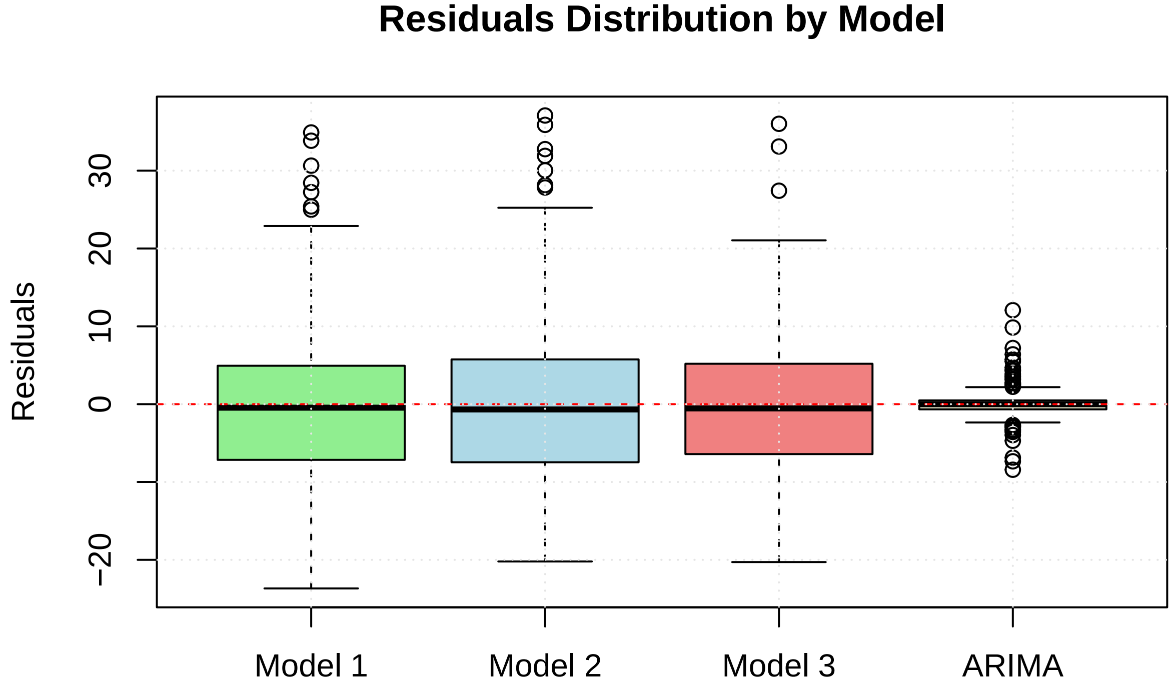

```r
par(mfrow = c(2,2))

plot(residuals(fit1), type = "l", col = "green", lwd = 1.5,
main = "Model 1: Residuals Over Time",
ylab = "Residuals", xlab = "Time Index")
abline(h = 0, lty = 2, col = "gray40", lwd = 2)
grid(col = "gray90")

plot(residuals(wls_model), type = "l", col = "purple", lwd = 1.5,
main = "Model 2: Residuals Over Time",
ylab = "Residuals", xlab = "Time Index")
abline(h = 0, lty = 2, col = "gray40", lwd = 2)
grid(col = "gray90")

plot(residuals(fit3), type = "l", col = "blue", lwd = 1.5,
main = "Model 3: Residuals Over Time",
ylab = "Residuals", xlab = "Time Index")
abline(h = 0, lty = 2, col = "gray40", lwd = 2)
grid(col = "gray90")

plot(residuals(fit_arima_best), type = "l", col = "red", lwd = 1.5,
main = "Model 4: Residuals Over Time",
ylab = "Residuals", xlab = "Time Index")
abline(h = 0, lty = 2, col = "gray40", lwd = 2)
grid(col = "gray90")
```




```
##
## === ARIMA Model Forecast ===
##
## 5 future months forecasts:
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 305       184.4098 182.0313 186.7883 180.7722 188.0474
## 306       184.6285 180.0509 189.2062 177.6277 191.6294
## 307       184.6285 178.0208 191.2362 174.5229 194.7341
## 308       184.6285 176.4818 192.7752 172.1692 197.0878
## 309       184.6285 175.1906 194.0665 170.1944 199.0627
```



Used Car CPI Forecast (ARIMA Model)
Used CPI
120
140
160
180
200
220
200
220
240
260
280
300
320
Index



Residuals Distribution by Model
Residuals
−20
0
10
20
30
Model 1
Model 2
Model 3
ARIMA