STATS531 Midterm Project \n Time Series Analysis on Fatal Car Accidents in Michigan

Geovanna Caballero, Annie Tai, Paul Zhang

2025-02-11

Introduction

Data Description

The data for this study is obtained from the National Highway Traffic Safety Administration (NHTSA) and covers the years 2008 to 2022 [1]. The data is aggregated on a monthly basis and includes the number of fatal crashes and the number of people killed in fatal crashes in both Michigan and the overall United States.

Exploratory Analysis

```
psych::describe(crashes_MI_ts)
```

Some Descriptive Statistics

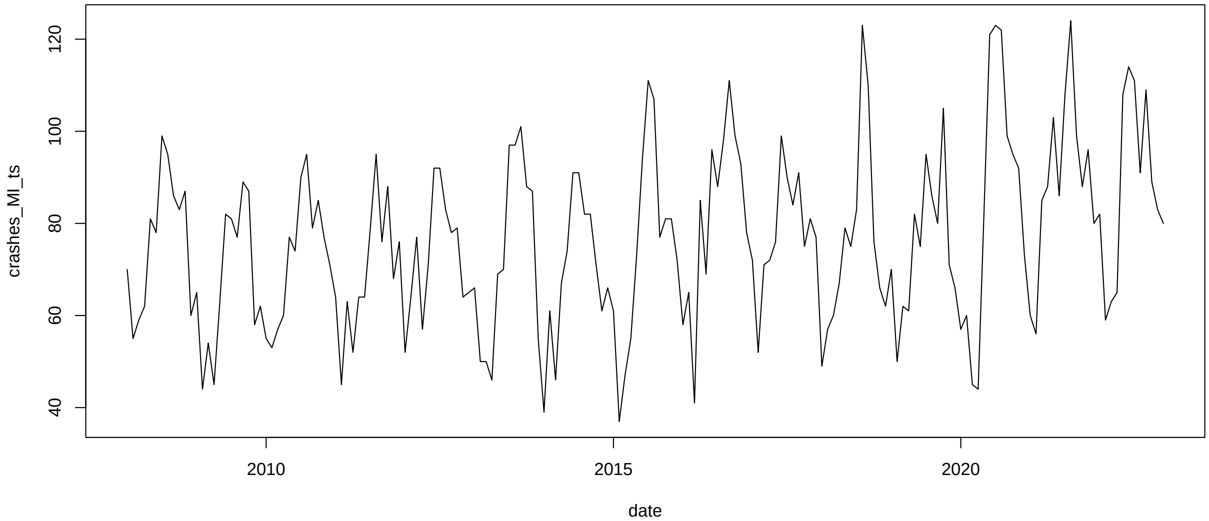
```
##    vars   n  mean    sd median trimmed   mad min max range skew kurtosis  se
## X1    1 180 76.54 18.77     77   75.92 20.02  37 124    87 0.28    -0.33 1.4
```

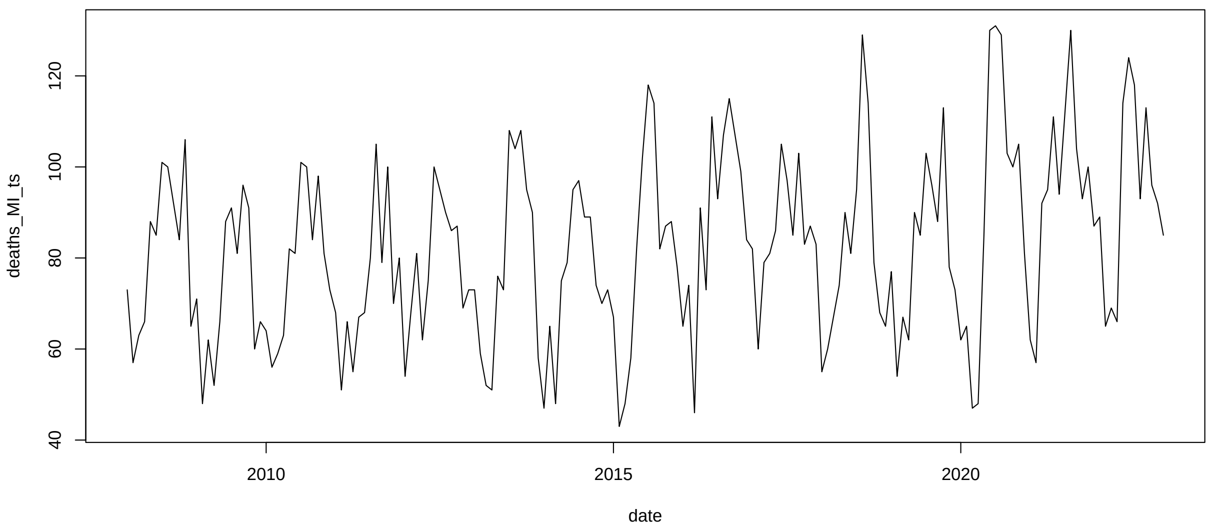
```
psych::describe(deaths_MI_ts)
```

```
##    vars   n  mean    sd median trimmed   mad min max range skew kurtosis   se
## X1    1 180 82.47 19.85     82   81.93 22.24  43 131    88 0.23    -0.51 1.48
```

```
# psych::describe(crashes_US_ts)
# psych::describe(deaths_US_ts)
```

```
par(mfrow = c(2, 1))
plot(date, crashes_MI_ts, type = "l")
plot(date, deaths_MI_ts, type = "l")
```






```r
# plot(date, crashes_US_ts, type = "l")
# plot(date, deaths_US_ts, type = "l")
```

Model Selection

```r
adf.test(crashes_MI_ts)
```

Stationarity Tests

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  crashes_MI_ts
```

```
## Dickey-Fuller = -9.8884, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

```r
adf.test(deaths_MI_ts)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  deaths_MI_ts
## Dickey-Fuller = -9.9405, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Both time series have passed the ADF test.

Updates I have found another statistical test called the KPSS Test[1]. It seems to be an advanced version of stationary test since it takes trends into consideration. The trend doesn't have to be linear.

The KPSS test may be helpful since our data apparently have nonlinear trends.

The null hypothesis, the alternative hypothesis for the test are as follows:

- H_0: The time series is a trend-stationary process (A stochastic process from which an underlying trend (function solely of time) can be removed, leaving a stationary process[2].)
- H_1: The time series is a unit root process.

```r
kpss.test(crashes_MI_ts, null = "Trend")
```

```
## Warning in kpss.test(crashes_MI_ts, null = "Trend"): p-value greater than
## printed p-value
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  crashes_MI_ts
## KPSS Trend = 0.043981, Truncation lag parameter = 4, p-value = 0.1
```

```r
kpss.test(deaths_MI_ts, null = "Trend")
```

```
## Warning in kpss.test(deaths_MI_ts, null = "Trend"): p-value greater than
## printed p-value
```

```
##
##  KPSS Test for Trend Stationarity
##
## data:  deaths_MI_ts
## KPSS Trend = 0.035296, Truncation lag parameter = 4, p-value = 0.1
```

Both of p-values exceed 0.05, indicating trend stationary.

ARMA Models, Before Detrending The following code block generates the AIC table, given a stationary time series. It's borrowed from the lecture notes[3].

```r
aic_table <- function(data,P,Q){
table <- matrix(NA,(P+1),(Q+1))
for(p in 0:P) {
    for(q in 0:Q) {
```

[1] https://en.wikipedia.org/wiki/KPSS_test
[2] https://en.wikipedia.org/wiki/Trend-stationary_process
[3] https://ionides.github.io/531w25/05/slides.pdf, pp.29

```r
    # table[p+1,q+1] <- arima2::arima(data,order=c(p,0,q))$aic
    table[p + 1, q + 1] = tryCatch({
        arima(data, order = c(p, 0, q))$aic},
        error = function(e) {NA}
        )
    }
}
dimnames(table) <- list(paste("AR",0:P, sep=""),
paste("MA",0:Q,sep=""))
table
}
require(knitr)
```

```
## Loading required package: knitr
```

```r
crashes_table = aic_table(crashes_MI_ts, 4, 3)
kable(crashes_table, digits=2)
```

Crashes

	MA0	MA1	MA2	MA3
AR0	1569.42	1503.41	1469.59	1466.14
AR1	1468.59	1470.27	1463.78	1465.29
AR2	1469.99	1471.80	1419.19	1406.59
AR3	1455.02	1445.95	NA	1419.80
AR4	1450.13	1446.46	1448.27	1405.65

ARMA(2, 3) vs ARMA(4, 3). The AIC's are close.

```r
crashes_arma23 = arima(crashes_MI_ts, order = c(2, 0, 3))
crashes_arma43 = arima(crashes_MI_ts, order = c(4, 0, 3))
```

Likelihood Ratio Test We have two nested parameter spaces and therefore we can try LRT.

```r
lrt = function(model0, model1, df){
    chi_sq = 2 * (model1$loglik - model0$loglik)
    pval = pchisq(chi_sq, df, lower.tail = FALSE)
    cat(sprintf("Test Statistic: %.4f\nDOF: %d\np-value: %.4f", chi_sq, df, pval))
    }
```

```r
lrt(crashes_arma23, crashes_arma43, 2)
```

```
## Test Statistic: 4.9350
## DOF: 2
## p-value: 0.0848
```

Fitting ARMA(4, 3) is not necessary.

Diagnostics for ARMA(2,3):

- Check the AR roots (the code also comes from the lecture notes[4]):

```r
AR_roots <- polyroot(c(1,-coef(crashes_arma23)[c("ar1","ar2")]))
abs(AR_roots)
```

[4] https://ionides.github.io/531w25/05/slides-annotated.pdf, p.32

```
## [1] 1.000436 1.000436
```

That's an edge case.

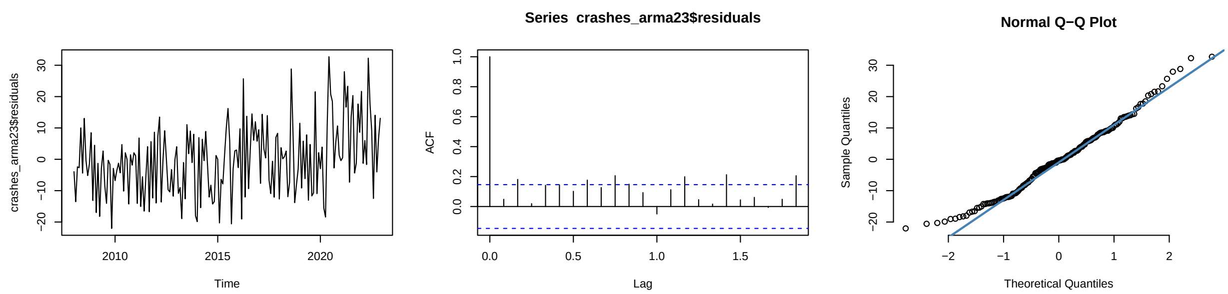
- Ljung-Box Test

```r
Box.test(crashes_arma23$residuals, lag = 20, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  crashes_arma23$residuals
## X-squared = 60.657, df = 20, p-value = 5.633e-06
```

- Residual Plots

```r
par(mfrow = c(1, 3))
plot(crashes_arma23$residuals)
acf(crashes_arma23$residuals)
qqnorm(crashes_arma23$residuals, pch = 1, frame = FALSE)
qqline(crashes_arma23$residuals, col = "steelblue", lwd = 2)
```



positive autocorrelation; right skew

Deaths The deaths series.

```r
deaths_table = aic_table(deaths_MI_ts, 4, 3)
kable(deaths_table, digits=2)
```

	MA0	MA1	MA2	MA3
AR0	1589.61	1533.72	1497.16	1496.03
AR1	1501.34	1503.34	1494.17	1496.14
AR2	1503.33	1502.35	1447.75	1439.67
AR3	1485.65	1478.34	1436.02	1449.48
AR4	1483.45	1479.97	NA	1440.10

ARMA(3, 2)

```r
deaths_arma32 = arima(deaths_MI_ts, order = c(3, 0, 2))
summary(deaths_arma32)
```

```
##
## Call:
## arima(x = deaths_MI_ts, order = c(3, 0, 2))
##
```

```
## Coefficients:
##          ar1      ar2     ar3      ma1     ma2  intercept
##       2.0129  -1.4898  0.2833  -1.7181  0.9996    82.5378
## s.e.  0.0758   0.1310  0.0756   0.0276  0.0292     1.3284
##
## sigma^2 estimated as 150.4:  log likelihood = -711.01,  aic = 1436.02
##
## Training set error measures:
##                        ME     RMSE      MAE       MPE    MAPE      MASE
## Training set -0.09294278 12.26535 10.01929 -2.738693 12.8855 0.7368335
##                     ACF1
## Training set -0.0374875
```

Diagnostics

- Check AR roots

```
AR_roots <- polyroot(c(1,-coef(deaths_arma32)[c("ar1","ar2","ar3")]))
abs(AR_roots)
```

```
## [1] 1.000148 1.000148 3.529283
```

- Ljung-Box Test

```
Box.test(deaths_arma32$residuals, lag = 20, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  deaths_arma32$residuals
## X-squared = 48.244, df = 20, p-value = 0.0003931
```

- Residual Plots

```
par(mfrow = c(1, 3))
plot(deaths_arma32$residuals)
acf(deaths_arma32$residuals)
qqnorm(deaths_arma32$residuals, pch = 1, frame = FALSE)
qqline(deaths_arma32$residuals, col = "steelblue", lwd = 2)
```

light-tailed on the left

ARMA models, After Detrending Let's see what happens after we detrend the data. (I read some docs[5] be4 writing the following code.)

[5]https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/stl

```
crashes_MI_decomposed = stl(crashes_MI_ts, s.window = "periodic")
crashes_MI_detrended = crashes_MI_ts - crashes_MI_decomposed$time.series[, "trend"]

deaths_MI_decomposed = stl(deaths_MI_ts, s.window = "periodic")
deaths_MI_detrended = deaths_MI_ts - deaths_MI_decomposed$time.series[, "trend"]
```

Crashes, detrended ARMA(2, 2) has the lowest AIC, but ARMA(2, 3) is close.

```
crashes_detrend_table = aic_table(crashes_MI_detrended, 4, 3)
kable(crashes_detrend_table, digits=2)
```

	MA0	MA1	MA2	MA3
AR0	1541.80	1485.47	1457.75	1456.75
AR1	1462.35	1463.52	1456.34	1457.91
AR2	1462.60	1388.38	1321.81	1322.02
AR3	1440.71	1353.18	1322.47	1325.80
AR4	1427.94	1342.58	1323.46	1325.13

```
crashes_detrend_arma22 = arima(crashes_MI_detrended, order = c(2, 0, 2))
crashes_detrend_arma23 = arima(crashes_MI_detrended, order = c(2, 0, 3))
```

Likelihood Ratio Tests

```
lrt(crashes_detrend_arma22, crashes_detrend_arma23, 1)
```

```
## Test Statistic: 1.7880
## DOF: 1
## p-value: 0.1812
```

ARMA(2, 2) is good.

Diagnostics

- Check the AR roots

```
AR_roots <- polyroot(c(1,-coef(crashes_detrend_arma22)[c("ar1","ar2")]))
abs(AR_roots)
```

```
## [1] 1.024448 1.024448
```

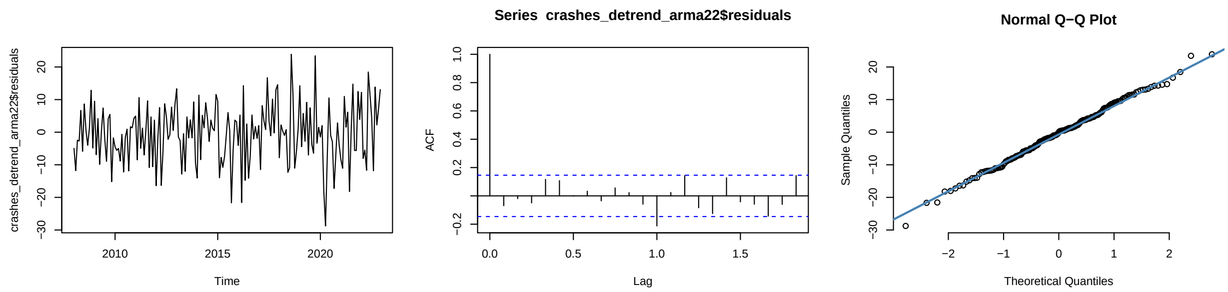
- Ljung-Box Test

```
Box.test(crashes_detrend_arma22$residuals, lag = 20, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  crashes_detrend_arma22$residuals
## X-squared = 33.929, df = 20, p-value = 0.02661
```

- Residual Plots

```
par(mfrow = c(1, 3))
plot(crashes_detrend_arma22$residuals)
acf(crashes_detrend_arma22$residuals)
qqnorm(crashes_detrend_arma22$residuals, pch = 1, frame = FALSE)
qqline(crashes_detrend_arma22$residuals, col = "steelblue", lwd = 2)
```



Deaths, detrended ARMA(2, 3)

```r
deaths_detrend_table = aic_table(deaths_MI_detrended, 4, 3)
kable(deaths_detrend_table, digits=2)
```

	MA0	MA1	MA2	MA3
AR0	1562.95	1516.24	1486.02	1486.72
AR1	1494.59	1496.43	1486.34	1488.29
AR2	1496.22	1496.31	1356.34	1355.52
AR3	1471.28	1387.10	1356.03	1357.76
AR4	1462.94	1380.57	1357.76	1358.91

```r
deaths_detrend_arma23 = arima(deaths_MI_detrended, order = c(2, 0, 3))
```

Diagnostics

- Check the AR roots

```r
AR_roots <- polyroot(c(1,-coef(deaths_detrend_arma23)[c("ar1","ar2")]))
abs(AR_roots)
```

```
## [1] 1.0197 1.0197
```

- Ljung-Box Test

```r
Box.test(deaths_detrend_arma23$residuals, lag = 20, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  deaths_detrend_arma23$residuals
## X-squared = 36.919, df = 20, p-value = 0.01197
```

- Residual Plots

```r
par(mfrow = c(1, 3))
plot(deaths_detrend_arma23$residuals)
acf(deaths_detrend_arma23$residuals)
qqnorm(deaths_detrend_arma23$residuals, pch = 1, frame = FALSE)
qqline(deaths_detrend_arma23$residuals, col = "steelblue", lwd = 2)
```

Conclusion

References

[1] https://cdan.dot.gov/query [2] https://ionides.github.io/531w25/08/slides.pdf, p.6