Midwest Home Price Index v.s. National Average from 1991-2016

Alex Zajichek

May 1, 2017

Applied Time Series Analysis, Spring 2017

1 Introduction

The *home price index* (HPI), recorded by the Federal Housing Finance Agency (FHFA), is a measure which quantifies the movement of single-family detached home prices by determining the average price changes in repeat sales or refinancing of the same properties. The FHFA provides an in-depth description of the methodology used in obtaining these data, which can be accessed on their website [1]. Using data from the 1^{st} quarter of 1991 through the 2^{nd} quarter of 2016, it was of interest to explore the similarities and differences of HPI for the Midwestern region of the U.S with the national average. Specifically, the average HPI of the four Midwestern states: Illinois, Iowa, Minnesota, and Wisconsin, as well as Wisconsin's individual HPI, will be examined over the time period. Using this data, the 'best' model for each region will be obtained by a selection criterion to determine if the series' are all generated by the same type of process. Then, with the optimal model, HPI forecasts will be made for 2016-2019, which will be compared and contrasted.

2 Data exploration

Figure 1 displays the time series for the national, Midwest, and Wisconsin HPI. Each series has the same pattern overall by (linearly) increasing from 1991 through roughly 2005, and then a large spike from 2005-2010 followed by a downfall. This phenomenon can likely be explained by the US housing bubble that peaked in the mid-2000's followed by the 2008 housing crisis where there was a large decline in home prices leading to mortgage delinquencies [2].

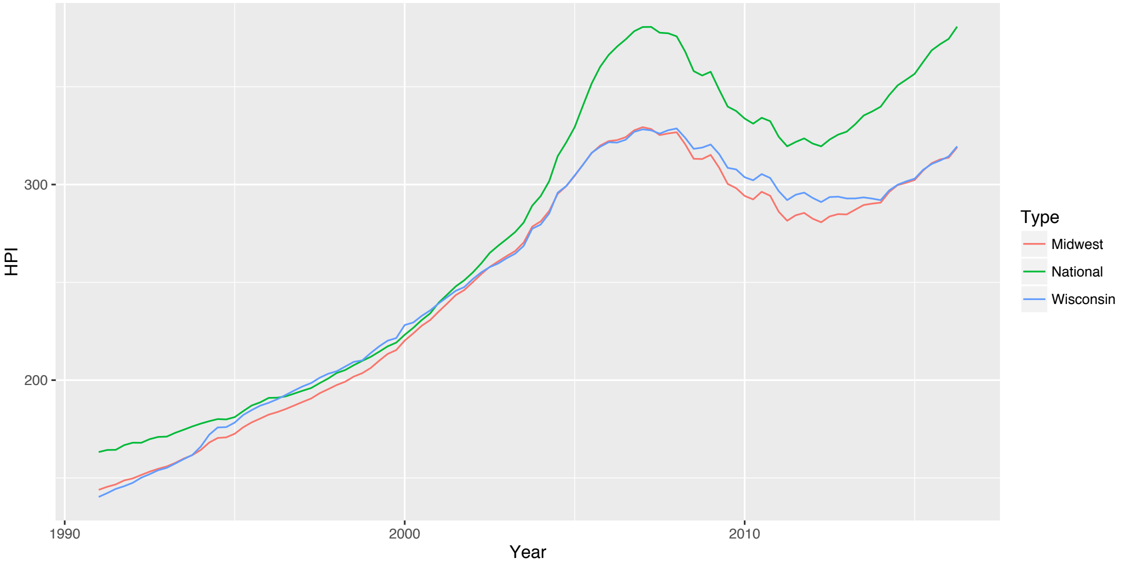

Figure 1: Original quarterly HPI for each of the three regions

Figure 1 also shows that the effect of the price peak in the mid-2000's was much greater on the eastern and western parts of the country than the Midwest. This is shown by the similarity of HPI in all three series from 1991-2005, and then a large increase for the US average relative to the other two. The decline, though, seems to be similar across the three series.

3 Obtaining stationarity

In order to choose models for this data, a *weakly* stationary process must first be obtained. Among other conditions, the mean, variance and autocorrelation at a particular time point must not depend on that time. Figure 1 clearly shows an increase in HPI with time, implying $E[N_t], E[M_t]$, and $E[W_t]$ depend on time where N_t, M_t, and W_t are the HPI for national average, Midwest, and Wisconsin, respectively, at time t. Notice that the series' are relatively flat from 1991-2008. After that, all three series appear to become much more variable. This observation also calls into question the homogeneity of the variance in a particular series. Each of these problems will be considered in the proceeding sections.

3.1 Variance stabilization

One possibility to stabilize the variability in a process is to use the *Box-Cox* transformation procedure, which is defined in the following:

$$y^* = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ log(y) & \text{if } \lambda = 0 \end{cases}$$

The `BoxCox.ar` function in the `TSA` package from R was used to get a 95% confidence interval for the transformation parameter, λ, for each series. Figure 2 shows the point estimates of λ along with the intervals. Since all values in the interval are plausible, 'nice' values for λ that were contained in the interval were chosen. Namely, the transformed series' are as follows:

$$N_t^* = 1 - N_t^{-1} \qquad M_t^* = \frac{1 - M_t^{-1.5}}{1.5} \qquad W_t^* = \frac{1 - W_t^{-0.5}}{0.5}$$

We will reference these transformed series' collectively as Y_t^*.

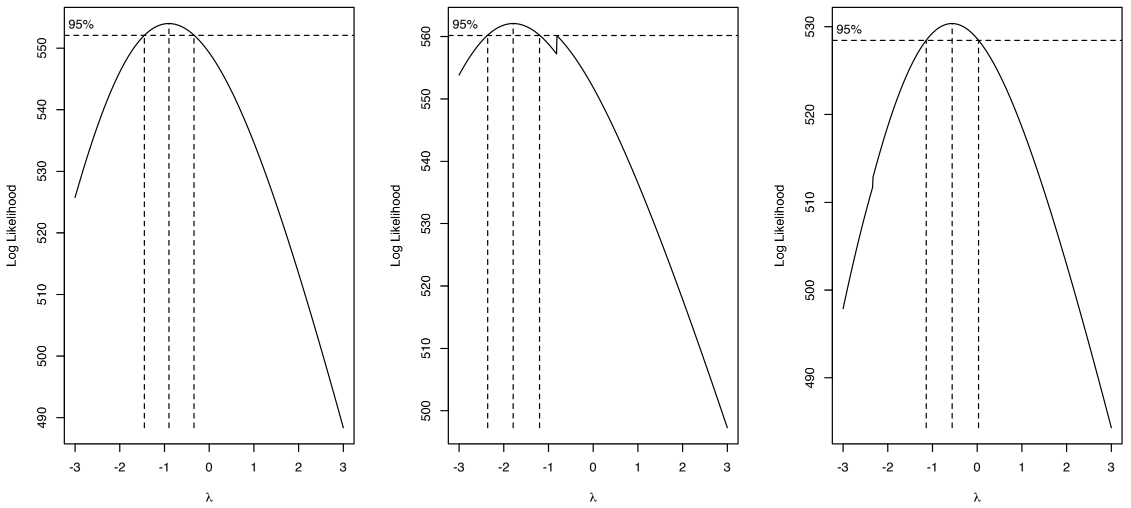

Figure 2: From left to right, the likelihood for a Box-Cox variance stabilizing parameter are shown for the national, Midwest, and Wisconsin series', respectively.

Figure 3 shows each series after its respective transformation. The variance appears to have been stabilized for later time points relative to the earlier ones.

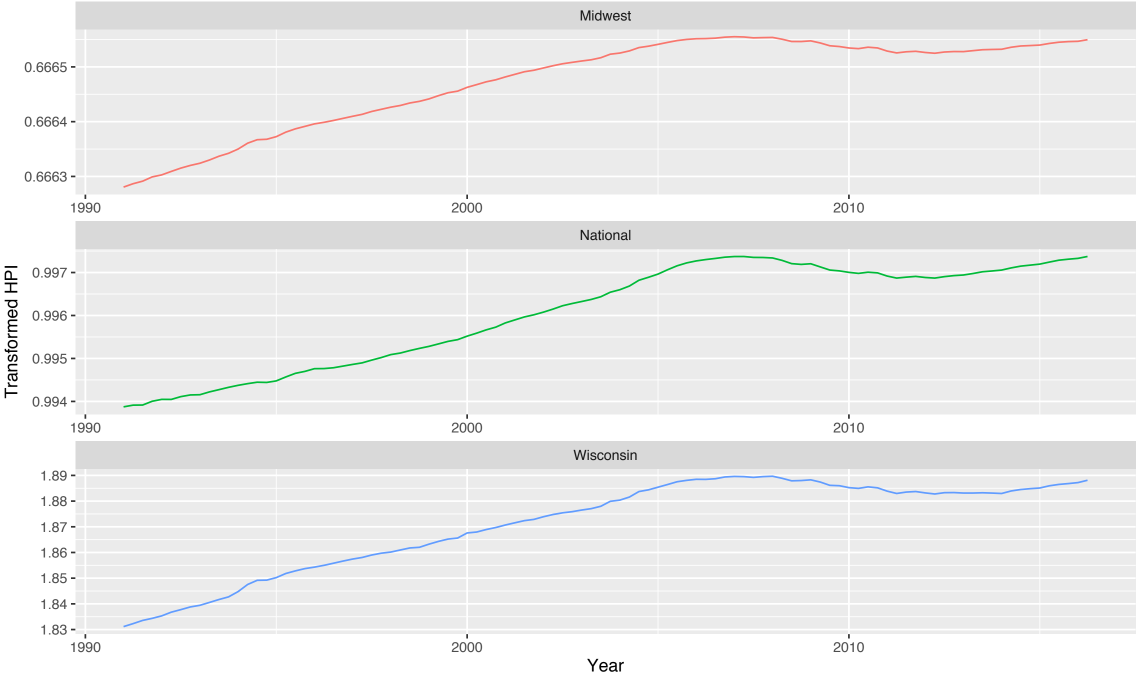

Figure 3: Quarterly HPI for each region after using the Box-Cox transformation.

3.2 Augmented Dickey-Fuller test

Let e_t be a stationary process. Then for a given series, Y_t, the *Augmented Dickey-Fuller* (ADF) test has the following form:

Suppose $Y_t = \alpha Y_{t-1} + e_t$, then

$$H_o : \alpha = 1 \qquad H_A : |\alpha| < 1 \text{ (stationary)}$$

Therefore, if H_o is not rejected, the test concludes that taking the *first difference*, $\nabla Y_t = Y_t - Y_{t-1}$, is necessary to obtain stationarity. It does not, however, guarantee it. The test can be run again on ∇Y_t to check whether a *second difference* is to be taken, denoted as $\nabla^2 Y_t = \nabla Y_t - \nabla Y_{t-1}$. Table 1 displays the *p-values* for the ADF test on the transformed series', Y_t^*, as well as the first and second differences.

Series	Y_t^*	∇Y_t^*	$\nabla^2 Y_t^*$
National	0.5237	0.6559	< 0.01
Midwest	0.4488	0.7507	< 0.01
Wisconsin	0.5581	0.6327	< 0.01

Table 1: P-values from the *Augmented Dickey-Fuller* for the transformed data, first difference, and second difference, respectively, for each series.

Figure 4 displays the resulting time series plots after taking the first and second differences. In all three series, the second difference was necessary to achieve stationary processes. Therefore, the final stationary processes for the national, Midwest, and Wisconsin series are $\nabla^2 N_t^*, \nabla^2 M_t^*, \nabla^2 W_t^*$, respectively.

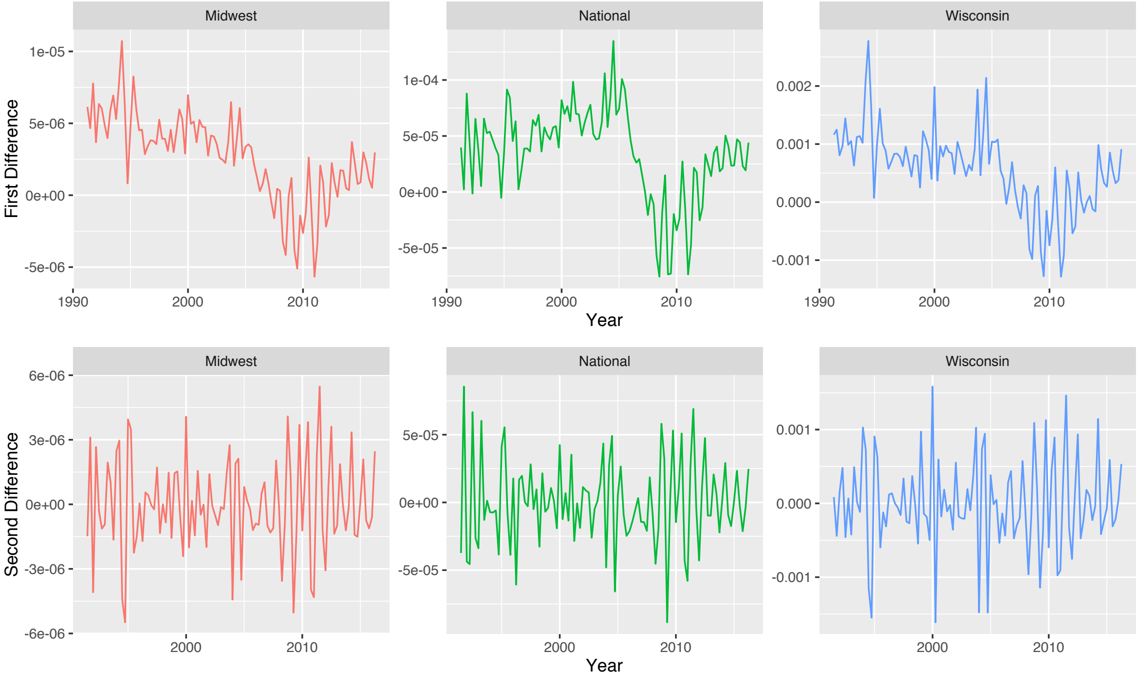

Figure 4: Plots for each of the three series after taking the *first difference* (top), and the *second difference* (bottom)

4 Pooling potential models

Now that (approximate) stationarity has been obtained, strategies to gather potential models for the data will be considered. By understanding the theoretical behavior of various model specifications, visualizations and summaries of the data can be examined to gain insight on what process may truly be generating the data. Since a goal of this analysis is to determine similarities and differences in the 'best' model chosen for each series, all potential models will be pooled collectively, and considered in the selection. Therefore, it will be possible for the chosen model for a given series to not have been suggested from the plotting evidence.

4.1 Autocorrelation

For a given lag, k, the autocorrelation (ACF) is the correlation between a process at time t, Y_t, and Y_{t-k}. Theoretical derivations show that *moving average* models, denoted $MA(q)$, have significant autocorrelation through $lag - q$, and then zero beyond. *Autoregressive* models, denoted $AR(p)$, have exponentially decaying autocorrelation when $p = 1$, and decaying cyclical behavior when $p = 2$. Keeping these ideas in mind while examining the plots give strategy to selecting reasonable models for the data. Figure 5 displays the autocorrelation plots for each series.

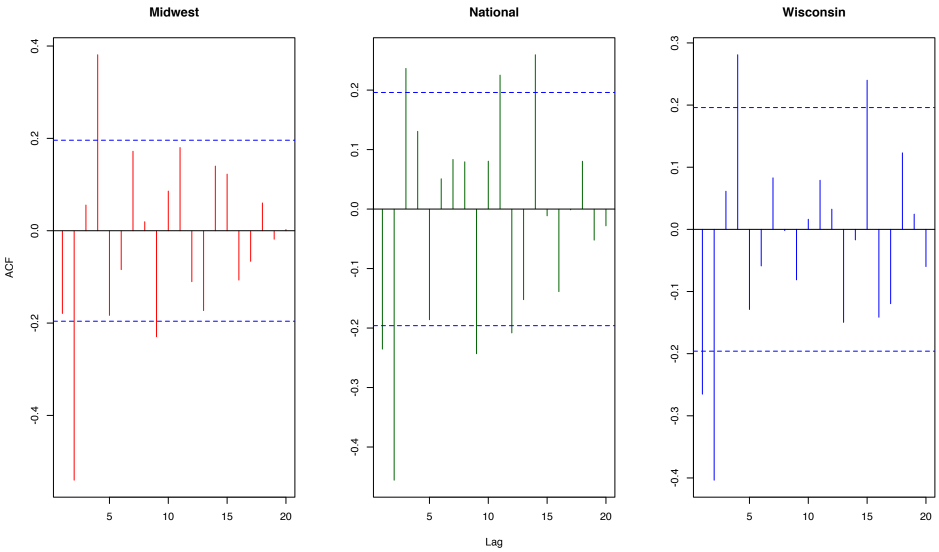

Figure 5: Autocorrelation for each (stationary) series after differencing twice.

Each of the plots have significant autocorrelation at $lag - 2$, as well as $lag - 4$. There also appears to be cyclical behavior in the plots, suggesting $AR(2)$ as a possibility. Therefore, $MA(2), MA(3), MA(4)$, and $AR(2)$ will be considered.

4.2 Partial-autocorrelation

It may be of interest to understand dependencies of lags *after* accounting for those in between. This would allow the compounding effect of additional lags on correlation to be marginalized, giving a more direct indication of the relationship between observations at various time differences. The partial-autocorrelation (PACF) does this. Again, by theoretical derivations, it is known that the PACF is zero after the true order of an $AR(p)$ model, and that there tends to be a gradual decay of the PACF for $MA(q)$ models. Using the same approach, this can be used as insight into the model pooling.

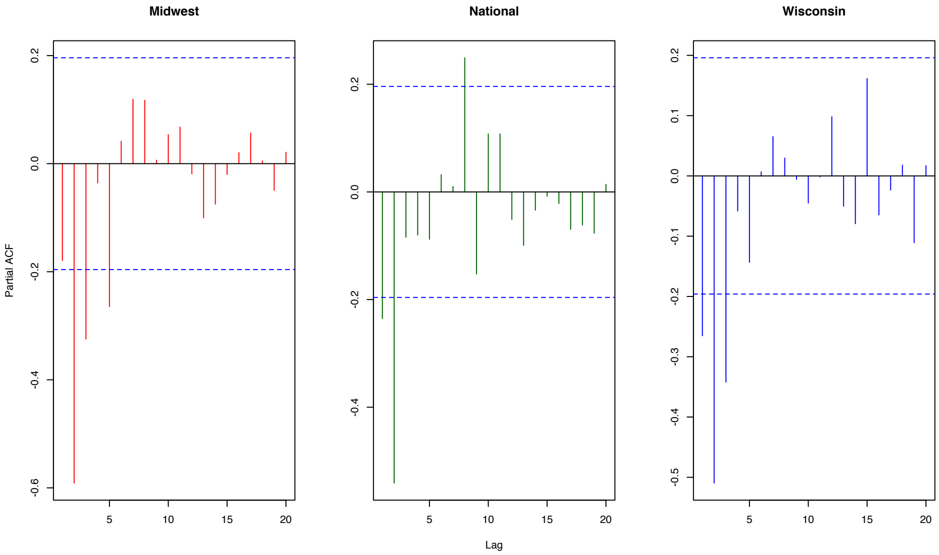

Figure 6: Partial-autocorrelation for each (stationary) series after differencing twice.

Figure 6 displays the PACF for each of the three series'. By examination, $AR(2), AR(3)$, and $AR(5)$ will be considered as possible models.

4.3 Other tools

There are other useful ways to gather potential models for a time series. Figure 7 displays a result of the `armasubsets` function in R, which determines well-suited models for the data by using a selection criterion on various subsets of parameters. This is especially useful to examine importance of seasonal effects when everything in between is not necessary to be accounted for. The *extended autocorrelation* plot (EACF) is a way to obtain models that combine $AR(p)$ and $MA(q)$, known as $ARMA(p, q)$. The appendix contains the tables produced by the `eacf` function in R.

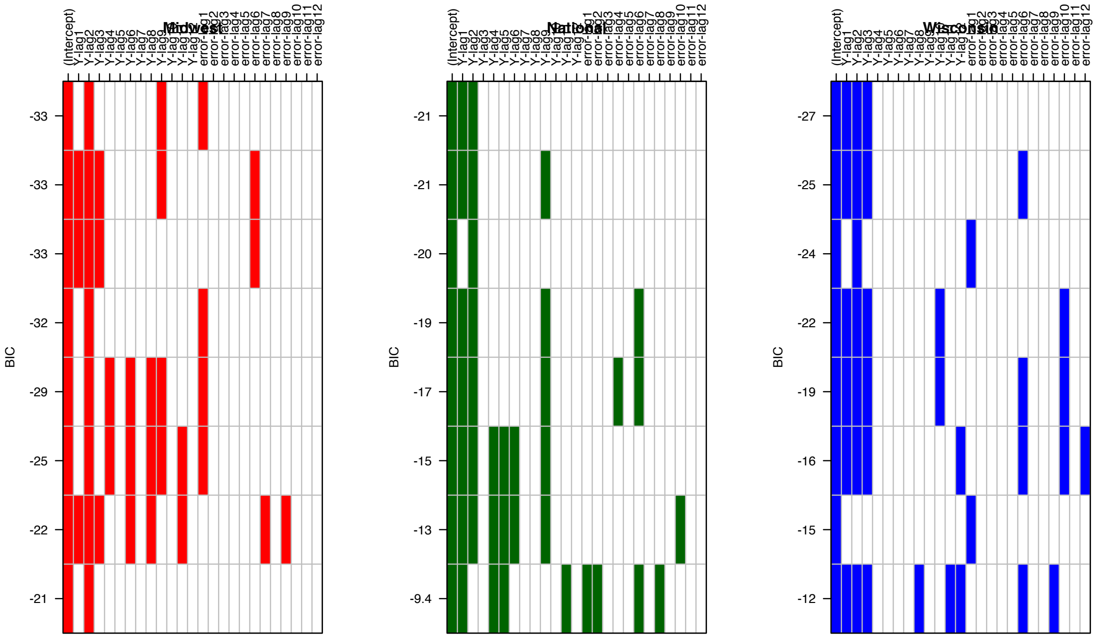

Figure 7: Plots produced by `armasubsets` suggesting potential models for each series according to the BIC criterion.

5 Determination of final model

The final pool of models for the three differenced series that were gathered in Section 4 are:

$$AR(2) \quad AR(3) \quad AR(5) \quad MA(3) \quad MA(4) \quad ARMA(2,3) \quad ARMA(2,6) \quad ARMA(9,1)$$

There are various methods and criterion to use to make a decision on the optimal model from the pool. In the proceeding subsection, the *corrected Akaike's information criterion* will be considered.

5.1 Best model by AICc

The traditional AIC statistic is a broadly used, and relatively effective, model selection criterion that penalizes models for having too many parameters. Therefore, the goal with any criterion is to have the optimal balance between goodness-of-fit, and parsimony. One issue with AIC is that it uses an asymptotic assumption, meaning it assumes the sample size is extremely large. It is fairly robust for moderately sized samples, but if n is small, it becomes very misleading. The $AICc$ is a correction to the AIC in that it relieves the large-sample assumption, making it much more appealing for smaller data. It can be defined as follows:

If $L(\hat{\boldsymbol{\theta}}|data)$ is the data likelihood evaluated at the optimizer, n is the sample size, and our model is of the form $ARMA(p,q)$, then

$$AICc = -2log(L(\hat{\boldsymbol{\theta}}|data)) + \frac{2n(p+q+1)}{n-2-(p+q)}$$

The goal would then be to choose the model with the *smallest AICc*. Table 2 displays the *AICc* of the three series' for each potential model determined from Section 4.

Series	AR(2)	AR(3)	AR(5)	MA(3)	MA(4)	ARMA(2,3)	ARMA(2,6)	ARMA(9,1)
National	**-1826.012**	-1824.055	-1821.335	-1820.946	-1822.218	**-1826.98**	-1824.153	-1824.934
Midwest	-2362.055	-2370.101	**-2373.637**	-2363.714	-2372.163	-2367.418	-2368.114	**-2373.077**
Wisconsin	-1221.921	**-1232.339**	-1230.323	-1228.139	-1228.611	-1228.54	-1227.224	**-1232.435**

Table 2: Corrected AIC values for potential models of each series. Bolded values are within two of minimum of the corresponding row.

For each row in Table 2, the minimum AICc is bolded along with measures within 2 of it. As a general rule, model criterion within 2 of the minimum value should also be considered [3]. Now that the pool of models is significantly narrowed, the predictive ability will be explored to choose the final one.

5.2 Predictive ability with MAD

Since one of the goals of this project is to forecast the HPI for each of three series, it is of interest to determine the predictive ability of the potential models. By holding out the last 5 observations in the series, we can look at the *mean absolute difference* (MAD) which will be defined as

$$MAD = \frac{\sum_{i=1}^{n} |\nabla^2 Y_t^* - \widehat{\nabla^2 Y_t^*}|}{n}$$

where $\widehat{\nabla^2 Y_t^*}$ is the predicted value of the second difference. The model that minimizes this quantity will be chosen for each series. Table 3 displays the MAD of each model. Note that these were multiplied by 10000 due to extremely small values.

National	MAD	Midwest	MAD	Wisconsin	MAD
AR(2)	0.125	AR(5)	0.00994	AR(3)	2.072
ARMA(2,3)	0.142	ARMA(9,1)	0.00987	ARMA(9,1)	4.080

Table 3: Mean absolute difference of the last 5 observations (× 10000) for the best two models according to AICc for each series.

Based on these calculations, the following models will be used:

$$\nabla^2 N_t^* \sim AR(2) \qquad \nabla^2 M_t^* \sim ARMA(9,1) \text{ w/ } \phi_2, \phi_9, \theta_1 \neq 0 \qquad \nabla^2 W_t^* \sim AR(3)$$

5.3 Residual diagnostics

Before proceeding to fit the model to the original data, the residuals will be examined to verify that model assumptions are met for those chosen in section 5.2. Our hope is that the models chosen have accounted for all significant correlation structures, leaving the residuals to be independent, and normally distributed. To make the assessment, we will look at quantile-quantile plots, residual density plots, autocorrelation and partial-autocorrelation plots, and perform the Ljung-Box test. All diagnostics will be done on the *standardized* residuals. Figure 8 displays the quantile-quantile plot for each series.

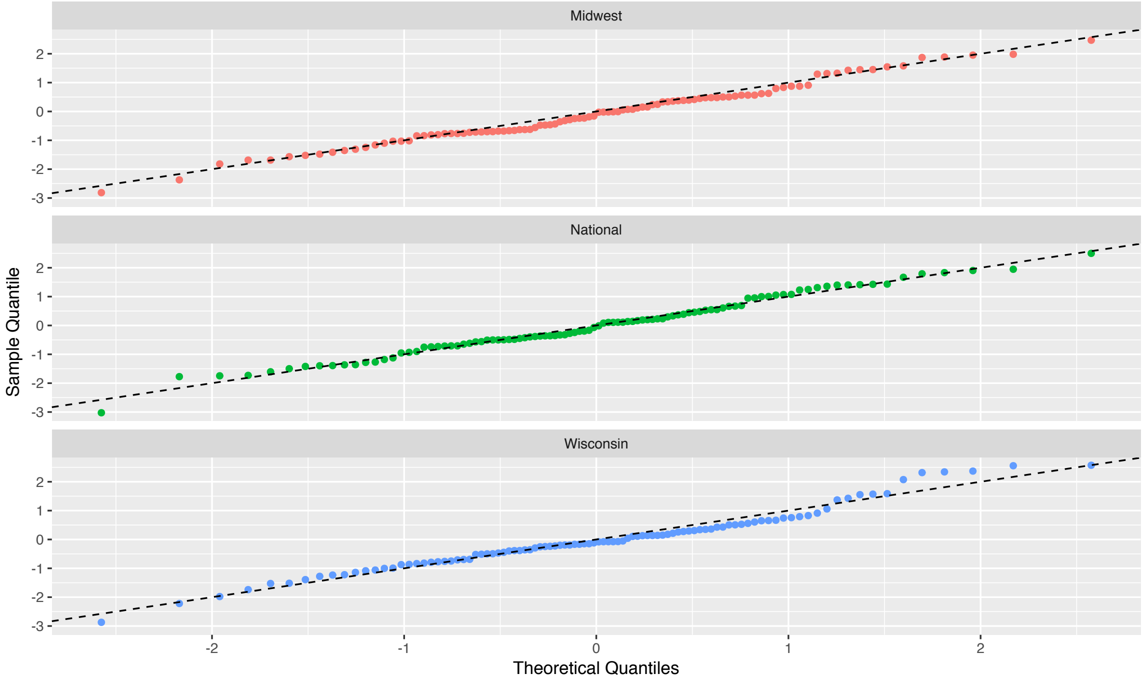

Figure 8: Quantile-quantile plots of the standardized residuals for the final model chosen on each series.

Here, the standardized residuals are plotted against their theoretical normal quantiles. Therefore, if the points stray far from the line on the plot, there will be evidence of a violation of the normality assumption. In all three plots, the residuals appear to fit the theoretical quantiles very well, with a slightly heavy upper-tail in the Wisconsin series. This is apparent in Figure 9, which displays the empirical densities of the standardized residuals.

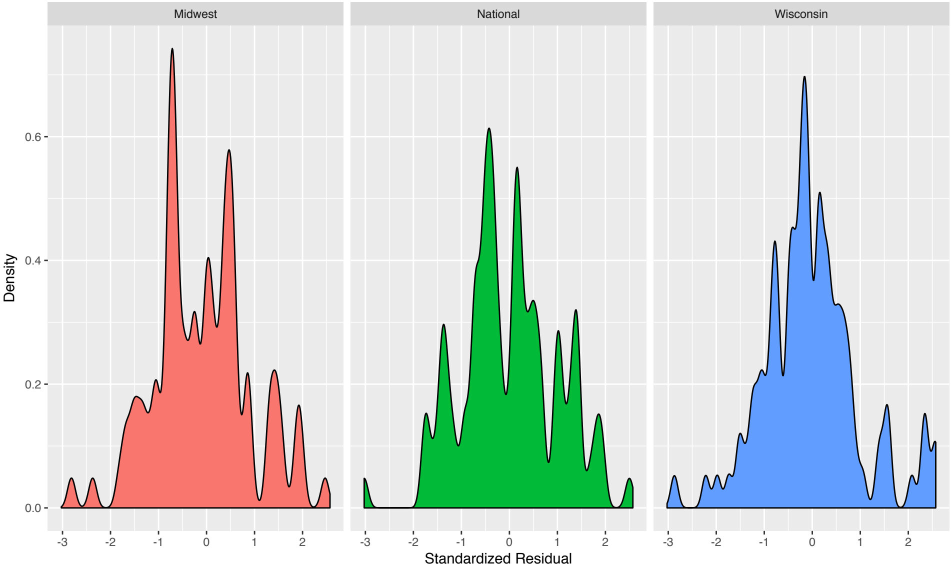

Figure 9: Density plots of the standardized residuals for the final model chosen on each series. A flexible bandwidth was chosen to ensure an accurate check for normality.

A relatively small bandwidth was chosen to see the detail of the distribution of residuals, which makes for a very ‘choppy’ distribution. Nevertheless, the overall shape of a normal distribution is present, and does not appear to suggest otherwise. The Shapiro-Wilk test is a possibility to formally test for normality, but it is very sensitive to outliers. The p-values from this test were .598, .585, and .020 for the national, Midwest, and Wisconsin series, respectively. Although the test says otherwise for Wisconsin, we will assume normality holds.

Figure 10 displays the time plots of the residuals, autocorrelation, and p-values for the Ljung-Box test for each series. To confirm that randomness is present, the *runs test* can be performed, which the p-values were 0.918, 0.840, and 0.652 for the three series, respectively. This indicates no dependency of residuals on previous observations.

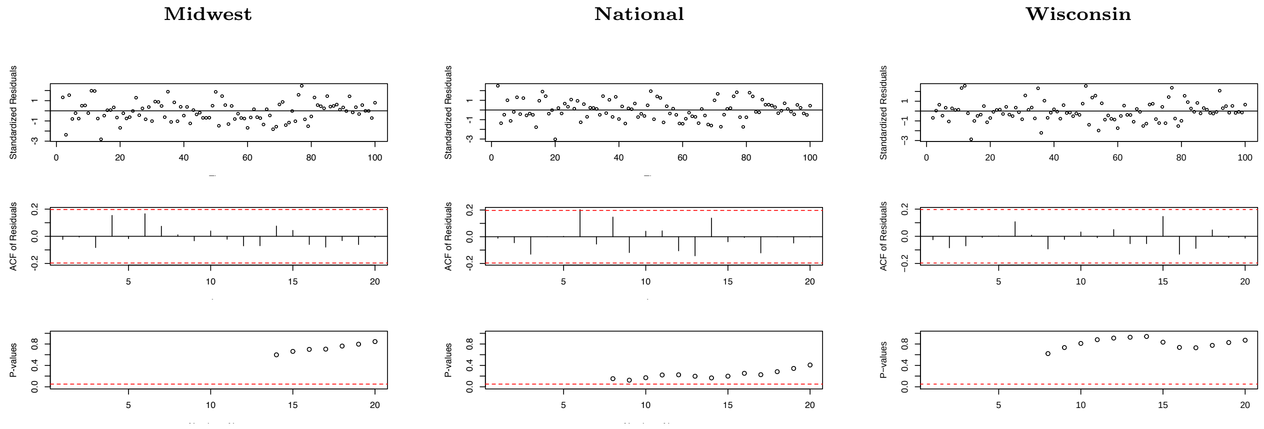

Figure 10: Model diagnostic plots produced by `tsdiag` for each series containing a time plot and autocorrelation plot for the standardized residuals, as well as the Ljung-Box test for a number of lags.

The autocorrelation plots in Figure 10, and the partial-autocorrelation plots in Figure 11 can also be examined. In all models, there does not appear to be any significant autocorrelation for either type.

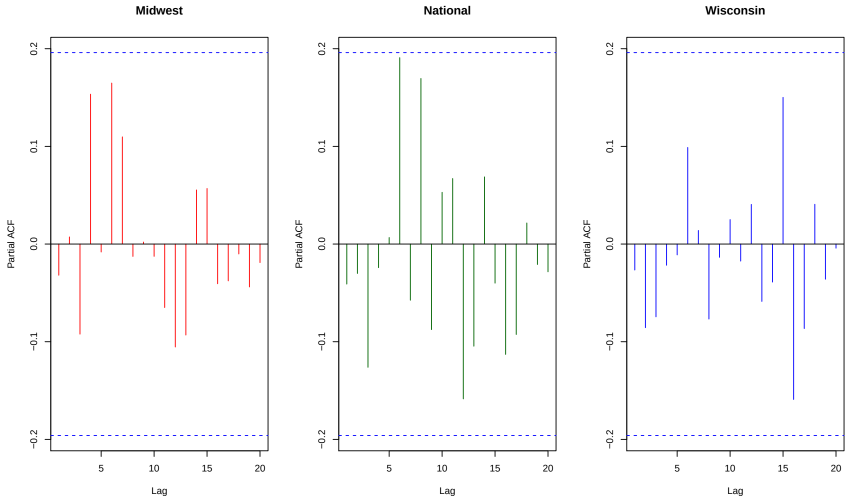

Figure 11: Partial-autocorrelation plots of standardized residuals for the final model chosen on each series.

The final diagnostic to perform is the Ljung-Box test. The hypotheses for the test are as follows:

$$H_0 : \rho_1 = \rho_2 = ... = \rho_k = 0 \qquad H_A : \text{At least one } \rho_i \neq 0$$

where ρ_i is the autocorrelation at *lag-i*. The bottom-most plot in Figure 10 displays the p-values for this test for $k = 8, ..., 20$ ($k = 8$ is the minimum out of the three) for each model. The red line is the 0.05 threshold to reject the null hypothesis. In all cases, there is not evidence to suggest that any ρ_i is non-zero for any model.

6 Forecasting HPI

The following models have been determined for the power-transformed data, in which the `arima` function was used to fit the models:

$$N_t^* \sim ARIMA(2, 2, 0) \qquad M_t^* \sim ARIMA(9, 2, 1) \text{ w/ } \phi_2, \phi_9, \theta_1 \neq 0 \qquad W_t^* \sim ARIMA(3, 2, 0)$$

In order to obtain forecasts, the defined transformations in Section 3 were inverted. Figure 12 displays each of the original series with forecasts for the 2^{nd} quarter of 2016 through the 2^{nd} quarter of 2019. Attached to the forecasts are 95% prediction intervals. These are also displayed in Figure 13, which is a zoomed plot on the forecasted HPI. Note that because the forecasts are *not* invariant to transformations, the confidence bounds are approximate.

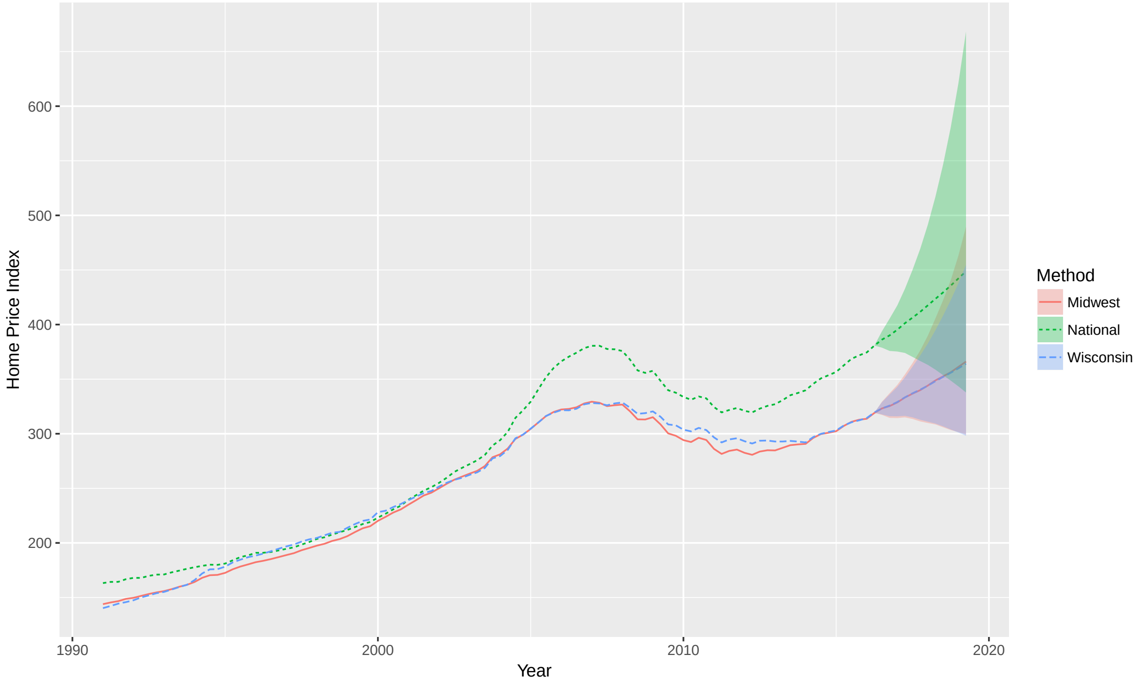

Figure 12: HPI forecasts with approximate 95% prediction intervals for 2016-Q2 through 2019-Q2.

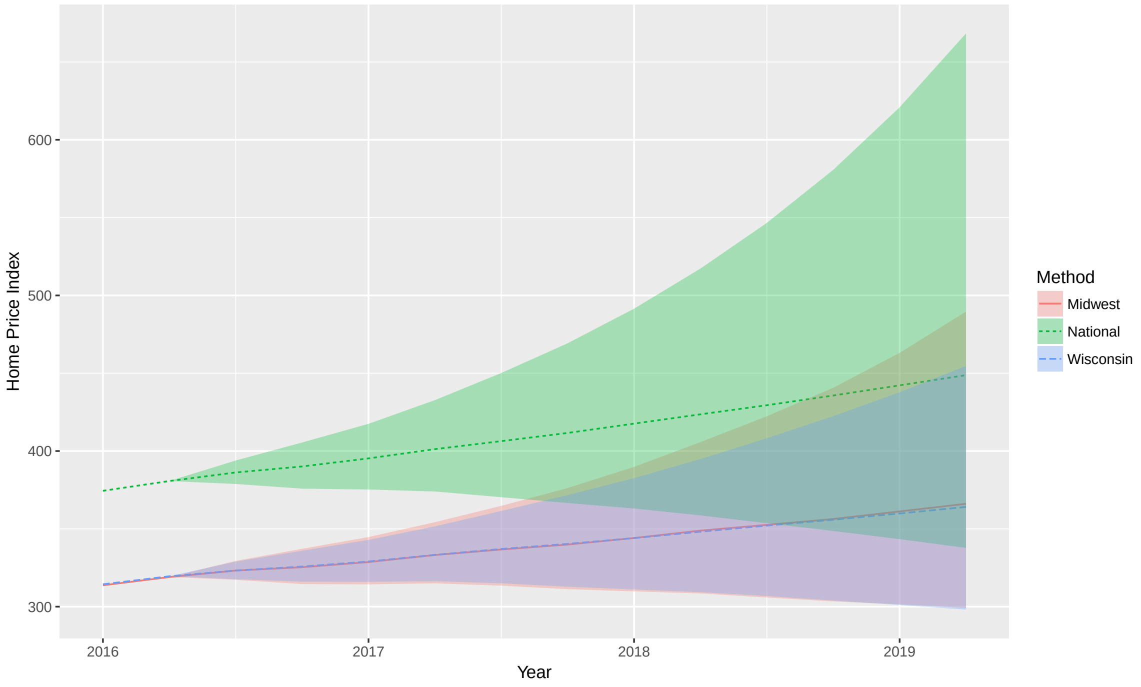

Figure 13: Zoomed plot of the 3-year forecast to assess predictions.

As expected, the margin of error significantly increased as forecasts were made farther out in all models. The margin of error for the national series, in particular, increased at a larger rate than the Midwest and Wisconsin series. The forecasts of the latter two were merely identical. Table 4 displays the point estimates of HPI at each time point, where an asterisk indicates non-overlapping prediction intervals between the national series and the other two.

Year	Quarter	Midwest	National	Wisconsin
2016	3	323.20	386.23*	323.23
	4	325.36	390.09*	325.78
2017	1	328.66	395.28*	329.03
	2	333.20	401.23*	333.30
	3	336.73	406.37*	337.07
	4	339.92	411.63	340.35
2018	1	344.13	417.63	344.05
	2	348.88	423.59	348.16
	3	352.56	429.44	352.09
	4	356.32	435.69	355.87
2019	1	361.21	442.21	359.89
	2	366.10	448.76	364.11

Table 4: HPI forecasts for the following 3 years for each region. The asterisk(*) indicates non-overlapping 95% prediction intervals for the National series with both the Midwest and Wisconsin series'. Note that all estimates for the latter two are very similar, with almost identical prediction intervals.

As noted, each HPI forecast for the Midwest and Wisconsin series' are within 1 of each other, and the national series forecasts are consistently larger by a margin around 70.

7 Conclusion

The objective of this project was to individually model the Home Price Index for Wisconsin and the Midwest, and compare these with the national average. Specifically, it was of interest to determine if the same model could be used (with different parameter estimates). According to the AICc and the MAD, none of the three models chose the same one to be optimal, though it was very close.

The similarity of the forecasts of HPI for Wisconsin and the Midwest provide evidence that housing prices among the states of Iowa, Illinois, Minnesota, and Wisconsin are similar, and change in a similar pattern. The national average HPI is fairly larger than the Midwest, which is expected since housing is known to be more expensive in larger cities on the east and west coasts. Another interesting observation is that after the peak and recession in the 2000's, the price index has just recently returned to the original, steady rate of increase that it had from 1991-2005, though there is much more separation between the Midwest and the rest of the country.

References

[1] Calhoun, Charles A. (1996). OFHEO House Price Indexes HPI Technical Description. Office of Federal Housing Enterprise Oversight. https://www.fhfa.gov/PolicyProgramsResearch/Research/PaperDocuments/1996-03_HPI_TechDescription_N508.pdf

[2] United States housing bubble, Wikipedia. Last updated: April 10, 2017.
https://en.wikipedia.org/wiki/United_States_housing_bubble

[3] Cavanaugh, Joseph E. (2016). Lecture II: The Akaike Information Criterion. The University of Iowa

Appendix

```
#Applied Time Series Project
#Comparison of the Home Price Index of WI to Midwest to USA
#Spring 2017, Osnat Stramer
#Author: Alex Zajichek

setwd("~/Documents/School/Iowa-M.S. Statistics/Spring 2017/Applied Time Series/Project")

#Plotting all 3 series together
load(file = "natl_HPI.Rda", verbose = T)
load(file = "midwst_HPI.Rda", verbose = T)
load(file = "WI_HPI.Rda", verbose = T)
all_series <- data.frame("Type" = c(rep("National", 102), rep("Midwest", 102), rep("Wisconsin", 102)),
"Year" = rep(natl_HPI$Year, 3), "HPI" = c(natl_HPI$Index_NSA, midwst_HPI$HPI, WI_HPI$index_nsa))
library(ggplot2)
ggplot(all_series) + geom_line(aes(x = Year, y = HPI, colour = Type))

library(TSA)
#Step 1: Variance Stabilizing Tranformations
par(mfrow = c(1,3))
BC1 <- BoxCox.ar(natl_HPI$Index_NSA, method = ’ols’, lambda = seq(-3,3,.01))
BC2 <- BoxCox.ar(midwst_HPI$HPI, method = ’ols’, lambda = seq(-3,3,.01))
BC3 <- BoxCox.ar(WI_HPI$index_nsa, method = ’ols’, lambda = seq(-3,3,.01))
BC1$mle #-0.9
BC1$ci #-1.45 -0.34 <-- use -1.0
BC2$mle #-1.79
BC2$ci #-2.36 -1.20 <-- use -1.5
BC3$mle #-0.56
```

```
BC3$ci #-1.14 0.03 <-- use -0.5

#Step 2: Obtain stationary series
national <- (natl_HPI$Index_NSA^-1 - 1)/(-1)
midwest <- (midwst_HPI$HPI^(-1.5) - 1)/(-1.5)
wisc <- (WI_HPI$index_nsa^(-0.5) - 1)/(-0.5)
transed <- data.frame("Type" = c(rep("National", 102), rep("Midwest", 102), rep("Wisconsin", 102)),
"Year" = rep(natl_HPI$Year, 3), "HPI" = c(national, midwest, wisc))
ggplot(transed) + geom_line(aes(x = Year, y = HPI, colour = Type)) +
facet_wrap(~Type, scales = "free", nrow=3) + ylab("Transformed HPI") + theme(legend.position="none")

    #Linear trend does not seem to hold, difference will be taken
adf.test(national) #pvalue for Y_t: 0.5237
adf.test(midwest) #pvalue for Y_t: 0.4488
adf.test(wisc) #pvalue for Y_t: 0.5581

library(TSA)
#First difference
nat_diff <- diff(national);
adf.test(nat_diff) #0.6559
mid_diff <- diff(midwest);
adf.test(mid_diff) #.7507
WI_diff <- diff(wisc);
adf.test(WI_diff) #0.6327
#Second difference
nat_diff2 <- diff(nat_diff);
adf.test(nat_diff2) #<0.01
mid_diff2 <- diff(mid_diff);
adf.test(mid_diff2) #<0.01
WI_diff2 <- diff(WI_diff);
adf.test(WI_diff2) #<0.01

library(ggplot2)
transed <- data.frame("Type" = c(rep("National", 101), rep("Midwest", 101), rep("Wisconsin", 101)),
"Year" = rep(natl_HPI$Year[-1], 3), "HPI" = c(nat_diff, mid_diff, WI_diff))
p1 <- ggplot(transed) + geom_line(aes(x = Year, y = HPI, colour = Type)) +
 facet_wrap(~Type, scales = "free", ncol=3) + ylab("First Difference") +
 theme(legend.position="none")
transed2 <- data.frame("Type" = c(rep("National", 100), rep("Midwest", 100),
rep("Wisconsin", 100)), "Year" = rep(natl_HPI$Year[-c(1,2)], 3), "HPI" = c(nat_diff2,
mid_diff2, WI_diff2))
p2 <- ggplot(transed2) + geom_line(aes(x = Year, y = HPI, colour = Type)) +
facet_wrap(~Type, scales = "free", ncol=3) + ylab("Second Difference") +
theme(legend.position="none")
gridExtra::grid.arrange(p1,p2)

#Step 3: Find pool of models for second diff
par(mfrow=c(1,3))
acf(mid_diff2, col = "red", main = "Midwest", xlab = "") #AR(2) MA(4)
acf(nat_diff2, col = "darkgreen", main = "National", ylab = "") #AR(2) MA(3)
acf(WI_diff2, col = "blue", main = "Wisconsin", ylab = "", xlab = "") #AR(2) MA(2) MA(4)
par(mfrow=c(1,3))
pacf(mid_diff2, col = "red", main = "Midwest", xlab = "") #AR(3) #AR(5)
pacf(nat_diff2, col = "darkgreen", main = "National", ylab = "") #AR(2)
pacf(WI_diff2, col = "blue", main = "Wisconsin", ylab = "", xlab = "") #AR(3)
eacf(nat_diff2) #ARMA(2,3)
```

```
AR/MA
0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x x o o o o o x o x  o  o  x
1 x x x o o o o o x o o  o  o  x
2 o o o o o o o o o o o  o  o  o
3 x o o o o o o o o o o  o  o  o
4 x o x o o x o o o o o  o  o  o
5 x o x o o x o o o o o  o  o  o
6 o o x o o x o o o o o  o  o  o
7 o x x o o x o o o o o  o  o  o
eacf(mid_diff2) #ARMA(2,6)
AR/MA
0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 o x o x o o o o x o o  o  o  o
1 x x o x x o o o x o x  o  x  o
2 x o x x o x o o o o o  o  o  o
3 o o x x o x o o o o o  o  o  o
4 o x x x o x o o o o o  o  o  o
5 x x x o o x o o o o o  o  o  o
6 x o x o o x o o o o o  o  o  o
7 x o x x o o o o o o o  o  o  o
eacf(WI_diff2)  #MA(4)
AR/MA
0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 x x o x o o o o o o o  o  o  o
1 x x o x o o o o o o o  o  o  o
2 x x x x o o o o o o o  o  o  o
3 o x o x o o o o o o o  o  o  o
4 x x o x o o o o o o o  o  o  o
5 o x o o o o o o o o o  o  o  o
6 o x o o o o o o o o o  o  o  o
7 x x o o o o o o o o o  o  o  o
par(mfrow = c(1,3))
plot(armasubsets(mid_diff2, nar = 12, nma = 12), col = "red"); title("Midwest")
plot(armasubsets(nat_diff2, nar = 12, nma = 12), col = "darkgreen"); title("National")
plot(armasubsets(WI_diff2, nar = 12, nma = 12), col = "blue"); title("Wisconsin")
#ARMA(9, 1) with zeros
#AR(2) AR(3)

#Step 3: Fit models and determine AICc
AICc <- function(model, params, n) {
    -2*logLik(model) + (2*(params + 1)*n)/(n - params - 2)
}
#AICc       #National   #Midwest    #Wisconsin
#AR(2)      -1826.012*   -2362.055   -1221.921
#AR(3)      -1824.055   -2370.101   -1232.339*
#AR(5)      -1821.335   -2373.637*   -1230.323
#MA(3)      -1820.946   -2363.714   -1228.139
#MA(4)      -1822.218   -2372.163   -1228.611
#ARMA(2,3)  -1826.98*    -2367.418   -1228.54
#ARMA(2,6)  -1824.153   -2368.114   -1227.224
#ARMA(9,1)  -1824.934   -2373.077*   -1232.435*

#Best two models for each can be delved further
```

```
#Step 4: Choose final model for each based prediction on power transformed data
library(TSA)
prediction_error <- function(model, newdata) {
    preds <- predict(model, length(newdata))$pred
    mean(abs(newdata - preds))*1000
}
#National ARIMA(2,0,0) ARIMA(2,0,3)
inds <- 1:82
prediction_error(arima(nat_diff2[inds], order = c(2,0,0),include.mean = F), nat_diff2[-inds])
prediction_error(arima(nat_diff2[inds], order = c(2,0,3),include.mean = F), nat_diff2[-inds])
# 0.0224, 0.0282

#Midwest ARIMA(5,0,0) ARIMA(9,0,1) w/ zeros
prediction_error(arima(mid_diff2[inds], order = c(5,0,0), include.mean = F), mid_diff2[-inds])
prediction_error(arima(mid_diff2[inds], order = c(9,0,1), include.mean = F, fixed = c(0,NA,0,0,0,0,0,0,N
#0.0023, 0.0017

#Wisconsin AR(3) ARMA(9,1) w/ zeros
prediction_error(arima(WI_diff2[inds], order = c(3,0,0), include.mean = F), WI_diff2[-inds])
prediction_error(arima(WI_diff2, order = c(9,0,1), include.mean = F, fixed = c(0,NA,0,0,0,0,0,0,NA,NA)),
#.4939, .6254


#Final models
mod_nat <- arima(nat_diff2, order = c(2,0,0), include.mean = F)
mod_mid <- arima(mid_diff2, order = c(9,0,1), include.mean = F, fixed = c(0,NA,0,0,0,0,0,0,NA,NA))
mod_wi <- arima(WI_diff2, order = c(3,0,0), include.mean = F)

#Step 5: Residual diagnostics
library(ggplot2)
library(TSA)
res_nat <- rstandard(mod_nat)
res_mid <- rstandard(mod_mid)
res_wi <- rstandard(mod_wi)
res <- data.frame("Type" = c(rep("National", 100), rep("Midwest", 100), rep("Wisconsin", 100)),
"Residual" = c(res_nat, res_mid, res_wi))
ggplot(res) + stat_qq(aes(sample = Residual, colour = Type)) +
geom_abline(intercept = 0, slope = 1, linetype = 2) + facet_wrap(˜Type, nrow =3)+
theme(legend.position="none") + xlab("Theoretical Quantiles") + ylab("Sample Quantile")
ggplot(res)  + geom_density(aes(Residual, fill = Type), adjust = 1/4) +
facet_wrap(˜Type) + ylab("Density") +
xlab("Standardized Residual")+ theme(legend.position="none")
shapiro.test(res_nat) #.598
shapiro.test(res_mid) #.5846
shapiro.test(res_wi) #.01995
runs(res_nat)$pvalue #0.918
runs(res_mid)$pvalue #0.84
runs(res_wi)$pvalue #0.652
tsdiag(mod_nat)
tsdiag(mod_mid)
tsdiag(mod_wi)
par(mfrow=c(1,3))
pacf(res_mid, main = "Midwest", col = "red")
pacf(res_nat, main = ’National’, col = "darkgreen")
pacf(res_wi, main = "Wisconsin", col = "blue")
par(mfrow=c(1,3))
```

```
acf(res_mid, main = "Midwest", col = "red")
acf(res_nat, main = 'National', col = "darkgreen")
acf(res_wi, main = "Wisconsin", col = "blue")

#Step 6: Fit models and forecast new HPI
nat <- arima(national, order = c(2,2,0), include.mean = F)
nat_preds <- predict(nat, 12)
nat_fit <- nat_preds$pred
nat_lower <- nat_fit - 2*nat_preds$se
nat_upper <- nat_fit + 2*nat_preds$se

mid <- arima(midwest, order = c(9,2,1), include.mean = F, fixed = c(0,NA,0,0,0,0,0,0,NA,NA))
mid_preds <- predict(mid, 12)
mid_fit <- mid_preds$pred
mid_lower <- mid_fit - 2*mid_preds$se
mid_upper <- mid_fit + 2*mid_preds$se

wi <- arima(wisc, order = c(3,2,0), include.mean = F)
wi_preds <- predict(wi, 12)
wi_fit <- wi_preds$pred
wi_lower <- wi_fit - 2*wi_preds$se
wi_upper <- wi_fit + 2*wi_preds$se

inv_trans <- function(y, lambda) {
    (y*lambda + 1)^(1/lambda)
}
#Transforming back to orginial units
nat_fit <- inv_trans(nat_fit, -1)
nat_lower <- inv_trans(nat_lower, -1)
nat_upper <- inv_trans(nat_upper, -1)

mid_fit <- inv_trans(mid_fit, -1.5)
mid_lower <- inv_trans(mid_lower, -1.5)
mid_upper <- inv_trans(mid_upper, -1.5)

wi_fit <- inv_trans(wi_fit, -.5)
wi_lower <- inv_trans(wi_lower, -.5)
wi_upper <- inv_trans(wi_upper, -.5)

prediction_intervals <- data.frame("Year" = rep(c(natl_HPI$Year, seq(2016.50, 2019.25,.25)),3)
           ,"Method" = c(rep("Midwest", 114), rep("National", 114), rep("Wisconsin", 114))
           , "Lower" = c(rep(0,102),mid_lower,rep(0,102),nat_lower,rep(0,102),wi_lower)
           , "HPI" = c(midwst_HPI$HPI, mid_fit,natl_HPI$Index_NSA,nat_fit,WI_HPI$index_nsa,wi_fit)
           , "Upper" = c(rep(0,102),mid_upper,rep(0,102),nat_upper,rep(0,102),wi_upper)
           )
pi <- prediction_intervals
pi$Lower[pi$Lower == 0] <- pi$HPI[pi$Lower == 0]
pi$Upper[pi$Upper == 0] <- pi$HPI[pi$Upper == 0]
prediction_intervals <- pi
ggplot(prediction_intervals) + geom_line(aes(x = Year, y = HPI, colour = Method, linetype = Method)) +
   geom_ribbon(aes(x = Year, ymin = Lower, ymax = Upper, colour = Method, fill = Method), alpha=.3, colo
ylab("Home Price Index")
ggplot(prediction_intervals[prediction_intervals$Year >= 2016,]) + geom_line(aes(x = Year, y = HPI, colo
    geom_ribbon(aes(x = Year, ymin = Lower, ymax = Upper, colour = Method, fill = Method), alpha=.3, col
    ylab("Home Price Index")
```