A Time Series Analysis of Daily Bike Rentals in a Metropolitan System

Mingheng Su, Andrea Zhao, Khushi Adukia and Turner Scott

University of Waterloo

STAT 443: Forecasting

Dr. Reza Ramezan

Fall 2025

Contents

Chapter 1

Introduction

1.1 Motivations

Early bike-sharing programs can be traced back to Amsterdam (duh) in the 1960s, where community members aspired to provide pollution free transportation to the disadvantaged through free bikes scattered across the city. While a noble cause, the end result was the theft or vandalism of all 50 bikes within a month of the program's inception. Despite early iterations' failures, bike-shares have grown to become commonplace in many cities across the world and still embody the ideals set out by the bike-sharing forefathers. Namely, a focus on affordable, climate conscious transportation for the masses. With that in mind, there are a number of innovations which have made these bike-sharing programs possible, such as bike docks and profitable business models for these programs. Both of which rely on accurate bike sharing usage data and forecasting to ensure there are adequate bikes, docks and pricing schemes for a city's demand. With these motivations in mind, our group set out to determine the time factors that affect bike-share usage and provide a forecast for future demand in Washington, D.C. to ensure that supply will meet the needs of the program.

1.2 Context, Scope, and Data Observation

Our dataset was retrieved from UC Irvine Machine Learning Repository [2] and was collected in conjunction with the Capital Bike Share program in Washington, D.C. [1]. The data were sampled in a daily scheme from the two-year period between 2011 and 2012, recording the numbers of bike rentals, temperatures, and wind speeds, etc., over the whole 731 days. For the purpose of time series analysis, we extract only the bike rentals and time data and keep exactly 730 of the observations for model training. Our objective and strategy will be structured as the following procedures in order to achieve the best modeling performance:

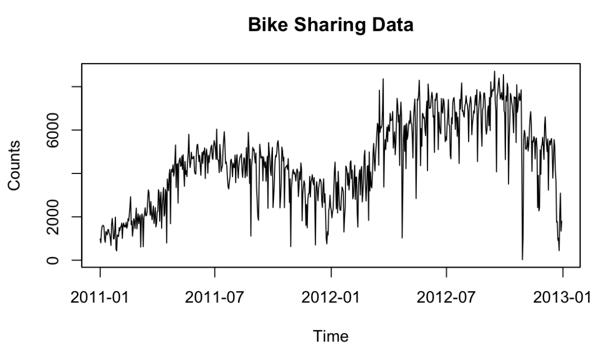
- Describe historical bike-rental patterns and identify any potential/transparent trends, seasonality, and model choices
- Apply multiple regression, smoothing, and Box-Jenkins methods on the data and

– Assess the fit, prediction power, and accuracy of models with different filtering methods and ultimately choose the best model that characterize the bike-rental trend

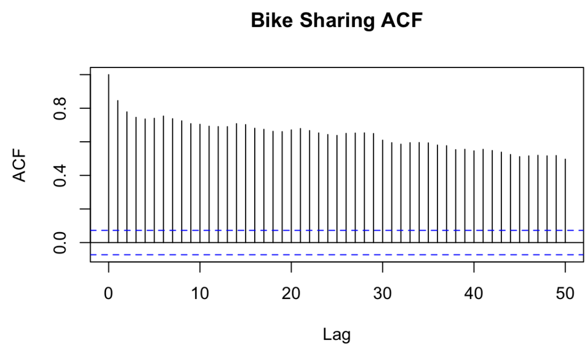
– Produce short-term forecasts using the selected models and provide comprehensive and rigorous statistical inference

- Discuss any social implications, analyze possible future reactions, and provide recommendations for the government, public, operators and users of Washington, D.C.'s bike sharing program

We measure the performance of models using adjusted R^2 (for fit of regression), mean squared error (MSE), residual error, AICc/BIC for analyzing fit; prediction error, 5-fold cross-validation or average prediction squared error (APSE) for analyzing prediction power; ACF plot, Ljung-Box test for addressing residual correlation; WW-plot and Shapiro-Wilk for residual normality, and runs test for randomness of the residuals. In addition, model complexity might also be considered as a factor. Let's start with a glance at the plot of the data:



(a) Bike-sharing Data



(b) Bike-sharing ACF

The data points reveal dramatic variability which could be a potential challenge to our model fitting. It also contains multiple extreme points that might be required for removal or weighting. Despite the large variability, the plot shows no fanning shape which is a good sign of constant variance. There is an obvious trend indicated by data plot and the linear decay in ACF. In the plot, we observe an overall upward trend between early 2011 and mid 2012. The two "humps" suggest an annual seasonal pattern such that Bike rentals steadily increases from winter to summer (and drops from summer to winter). When approaching the end of 2012, the bike rentals fall back to the same level as the starting of 2011 and 2012. We could reasonably guess that whatever how large the bike rentals grow in summer, they always decline to a low amount due the severe weather, and rise to a higher level in the next summer. However, this is just our guess and we will see whether the statistical forecasting reveals this trend. The ACF exhibits a clear linear decay, as well as some weak weekly seasonality which we ignore since it is not significant.

Intuitively, regression and Holt-Winters smoothing might not be good choices on these large-noise data. For regression, it might only capture the rough trend and provide rough prediction; For Holt-Winters method, the large noise might interrupt its steps. A final choice of model is going to be determined in the following chapter.

Chapter 2

Model Fitting, Examination, Selection

Our possible choices are Holt-Winters smoothing, regression, and differencing. As a start of our model selection process, we decided to use regression to model the bike sharing dataset because we observed strong and clear trends as well as annual seasonality in the data.

2.1 Regression

As the data seems to have a linear trend plus an annual seasonality, we try to fit a weighted regression model

$$y = \beta_0 + \beta_1 t + \beta_2 \cos\left(\frac{2\pi t}{365}\right) + \epsilon \tag{2.1.1}$$

The plot of fitted values, prediction and intervals is

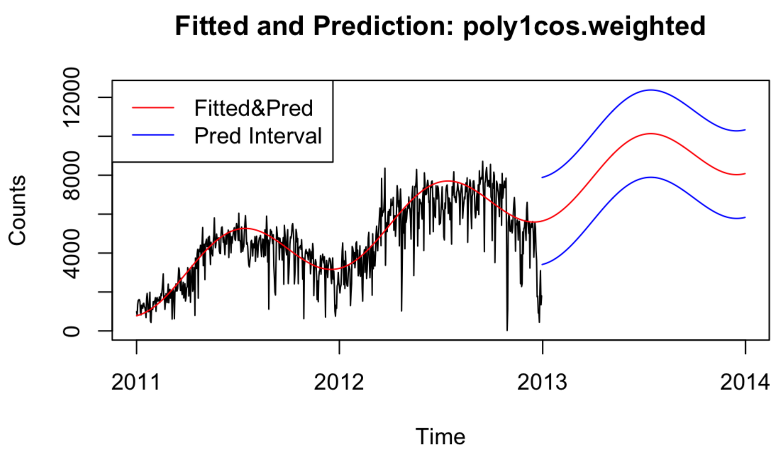


Figure 2.1.1: poly1cos.weighted

where

	Est.	Std. Err.	p-value
$\hat{\beta}_0$	7.7566579	0.0231303	<2e-16
$\hat{\beta}_1$	0.0015896	0.0000568	<2e-16
$\hat{\beta}_2$	-0.4130774	0.0168144	<2e-16

Table 2.1: poly1.cos.weighted

We are not focusing on the whole-year prediction, which is just meant to provide a sense of the future trend; we are only interested in predicting the first three months. In general it suggests an upward trend in the upcoming months.

Outlier manipulation played a significant role in our work since the huge number of extreme points significantly affects the fit, residual normality, and prediction power. We first considered removing those points that are far from their neighbors[2.1.1], a.k.a outliers. Since our data is non-stationary, the interquartile range (IQR) method for outlier detection is not appropriate for detecting outliers in the raw data. On the other hand, removing data points creates *artificial* data which distorts our purpose of analysis. Instead, we tried using a weight function to downweight data points that are far from their neighbors. Throughout our model selection process, we examined loss, prediction error, AICc/BIC, and k-fold cross-validation error, as well as visually checked the residual fit. To avoid overfitting, we deliberately limited the number of parameters in our models. Below are the residual plots and model performance:

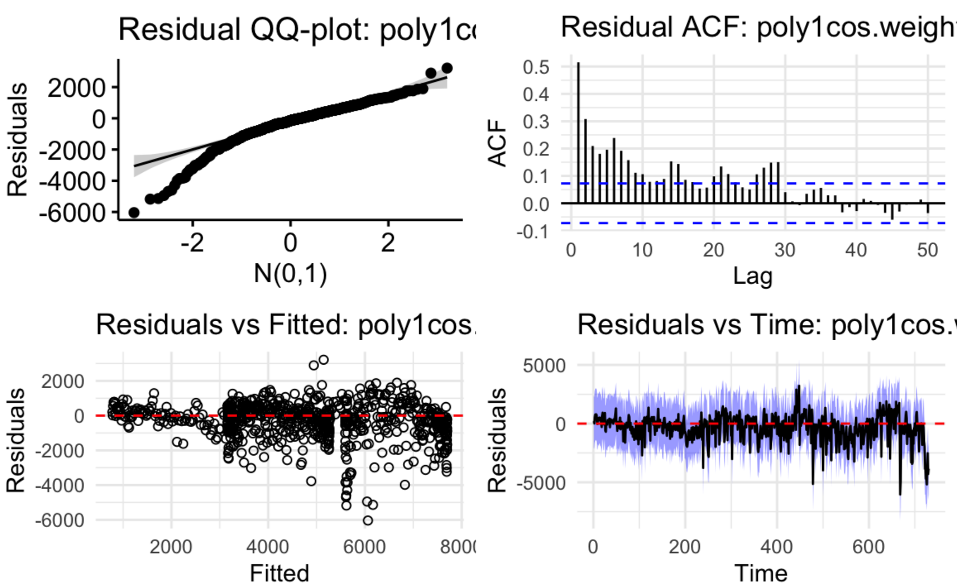


Figure 2.1.2: poly1cos.weighted residuals

The table below shows the model examination results in the order of: *Residual Standard Error, Prediction Error on the 10th day, AICc, BIC, adjusted* R^2*, 5-fold cross-validation Error, p values for*

[2.1.1]”far” means the distance to their neighbors exceed some threshold, which is set to be 1000 in our project

Shapiro-Wilk and runs tests.

Res. Err.	Pred. Err. (day 10)	AICc	BIC	adj R^2	5-fold CV	Normality	Randomness
1136.516	2236.792	15005.54	15023.86	0.955	1665690	1.762354e-18	1.576682e-24

Table 2.2: poly1.cos.weighted performance

The MSE can be calculated as $1136.516^2 \approx 1291669$. The large variability unavoidably causes large residual and prediction errors, APSE, and MSE for regression models, but this can be mitigated using weighted data. The AICc and BIC are so large that they indicate the inadequacy of this model. The adjusted R^2 is high since the errors are reduced due to the scaled data but that does not necessarily imply a good fit.

We now focus on the residual diagnostics. The Q-Q plot indicates the skewness of residuals, especially in the lower quantiles (heavy lower tail), and the small p-value of the Shapiro–Wilk test also confirms the non-normality. This model cannot characterize the drop in bike rentals during winter months, and reveals poor fit in the end of 2012 and 2013, which might be a reason for the skewness and thus non-normality of residuals. We also tried a Box–Cox transformed model which improves but does not resolve the skewness issue. The prediction intervals are thus invalid since they are constructed upon the normality assumption. The residual ACF plot shows significant correlations within the first 30 lags, which is not a sign of white noise, and we could also observe some minor non-randomness in the residuals vs time plot but we are still glad to see the constant variance exhibited.

Overall, this model does not meet our expectation for using regression. Furthermore, other regression models have very similar performances but are more complex than this model, and that's why we choose this model for our analysis. The other regression models can be found in appendix A.1. We conclude that regression on this dataset does not meet the expected requirements of an adequate model. The major reason is that bike rentals depend significantly on features other than time, such as temperature, weather, and holidays, and other factors. The uncertainty associated with these unspecified factors results in the huge variability in the data and thus the failure of regression. The inclusion of those unspecified factors could potentially improve the model fit, but that is beyond the scope of fitting time series data. Nonetheless, it still reflects the ascending trend in bicycle commuting that is a great mitigation to climate change.

2.2 Holt-Winters Smoothing

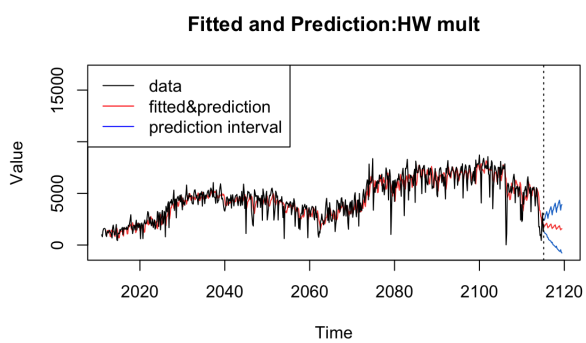
We have a raw series of daily bike-share counts that display both a long-term trend and a clear annual seasonal pattern, with low usage during the winter months and high usage in summer, repeated over two consecutive years (2011-2012). The amplitude of the seasonal effect grows as the level of the series increases, which suggests that the seasonal component should act multiplicatively instead of additively.

To capture these features, we modeled this data using a multiplicative Holt-Winters exponential smoothing model with seasonal period $p = 365$ days. In this way, we treat the level, trend, and seasonal indices as evolving over time, and we can update them recursively using the Holt-Winters algorithm. The seasonal period corresponds to one calendar year.

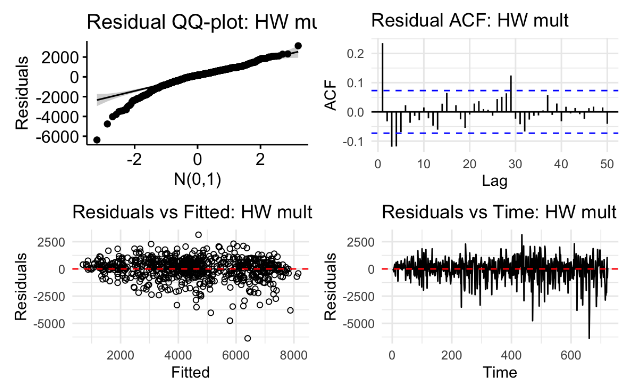
Initially, we experimented with a weekly seasonal period $p = 7$, which captures the recurring differences between weekdays and weekends. However, due to the main scientific interest of our project which is to understand and forecast annual demand patterns, we are more interested in $p = 365$ specification which produces a smoother yearly pattern and yields a forecast that better aligns with the behavior we expect in future years.

Nonetheless, since apse of Holt-Winters method requires at least two periods (years) of data, we are not able to run apse if choose $p = 365$ so that the prediction power cannot be determined. We first try $p = 7$.

The plot shows the data and fitted values[2.2.1]. It also includes a 30-day forecast with prediction intervals.



(a) HW mult



(b) HW mult residuals

Pred. Se.	MSE	APSE	Normality	Randomness
1451.699	956499.3	7819302	3.375413e-18	5.40851e-08

Table 2.3: HW mult performance

First of all, its MSE is 956499.3, which is much smaller than regression model's 1291669 (2.2) and the fitted values are closer to the data. However, the APSE of 7819302 is dramatically larger than regression model's 1665690[2.2.2] and it has very poor prediction power compared to regression. As we can observe from the prediction plot, the predicted values carry no trend information as the training data has and produce a downward *straight line*. This model is definitely not a good choice for forecasting.

The residuals of this multiplicative Holt–Winters model do not provide an adequate fit, but overall it outperforms the regression model. Similar to the regression model, the Q–Q plot shows a heavy tail in the residuals, which are not normal, hence the prediction intervals are unreliable since they are based on the normality assumption. The small p-values of Shapiro and runs tests also demonstrate

[2.2.1] Please ignore the wrongly labeled time axis. They are supposed to be days starting at year 2011.

[2.2.2] Just for a rough comparison since K-fold CV error is the average of APSEs on different validation sets.

the non-normality and non-randomness. However, the residuals are less correlated than the regression model's and look stationary, since they have constant mean and no significant fanning shape. Although they are not white noise yet, they can be modeled by a stationary process such as MA(4) or MA(29) as suggested by the cutoff lag 4 and 29 in ACF plot, or ARMA since there exists a damped sine wave in the ACF. The residual vs fitted plot looks random and exhibits no obvious relationship between them. In general, this model has a great fit.

Overall, the multiplicative Holt-Winters model with period 7 can be an effective *detrendizing and deseasonalizing* agent that removes the non-constant mean and produces filtered residuals for fitting stationary processes. A direct comparison is the multiplicative model with period 365 shown in ???A.3. Since it carries longer memory (up to one year before), it can model the annual seasonality better, but has an inferior fit with poor residuals, which is why we place it in the appendix. The reason is still due to the large variability. There might be an existing annual seasonal effect, for instance, the same day in the next year carries the same effect associated with seasonality as today. However, the information of this effect is interrupted by the large noise so it is not very accurate. Another reason is that we only have two years of data which is insufficient for training the seasonal factors. Therefore, the multiplicative Holt-Winters model with period 365 can only be used to provide rough forecasting but is not a preferred choice of our final model.

2.3 Differencing

As the data contains a linear trend and seasonality of period 365, we apply one regular and one seasonal differencing step

$$Y_t = \nabla\nabla_{365} X_t = (1-B)(1-B^{365})X_t \tag{2.3.1}$$

where X_t denotes the bike rental data and Y_t denotes the double differenced data. In the model fitting process, it turns out that this model does not converge at all. However, X_t is indeed stationary after one regular differencing

$$Y_t = \nabla X_t = (1-B)X_t \tag{2.3.2}$$

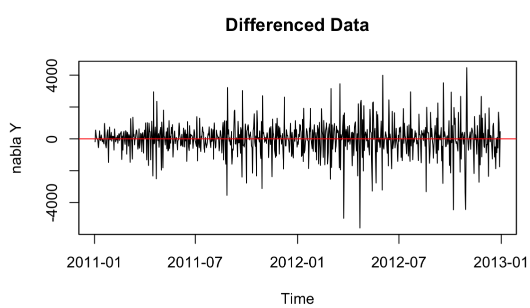
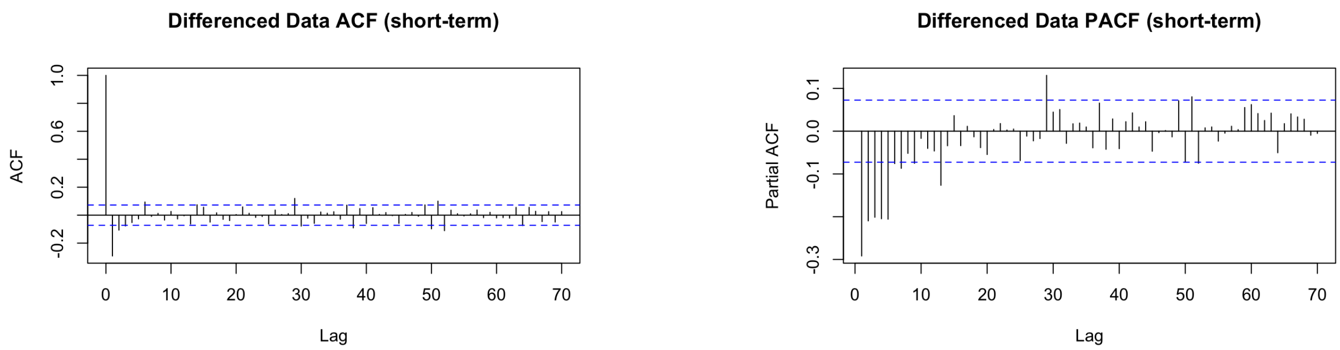


Figure 2.3.1: Y_t

Except for some possible randomly occurring spikes, there is no significant correlation after lag

70[2.3.1]. Take a closer look at



(a) Y_t ACF (b) Y_t PACF

There is strong evidence of ARMA process. In the ACF, there is a cutoff after lag 1, and in the PACF, there is an exponential decay. This is a sign of an MA process, possibly MA(1) or MA(3). As well, there is an exponential decay in the ACF and cutoff after lags 5 and 13 in the PACF, which suggests an AR process, possibly AR(5) or AR(13). No significant seasonal lags observed. Since X_t regularly differenced once, its possible options are ARIMA(5,1,1), ARIMA(5,1,2), ARIMA(13,1,3), etc. Note that both the ACF and PACF have significant correlations at lag 29, but that could be false positive since they are far from the last significant lag. The red line represents the fitted values and predictions, and the blue lines are prediction intervals.

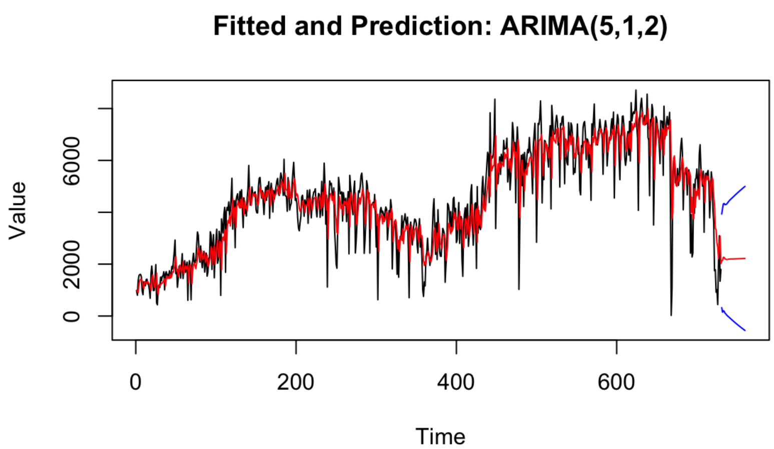


Figure 2.3.3: ARIMA(5,1,2)

Due to the small p, q in context of ARIMA process, the prediction converges very fast to the mean and this model can only predict a few steps ahead, so it might have poor prediction power. Now we look at residuals.

[2.3.1] See appendix A.2 for larger lags.

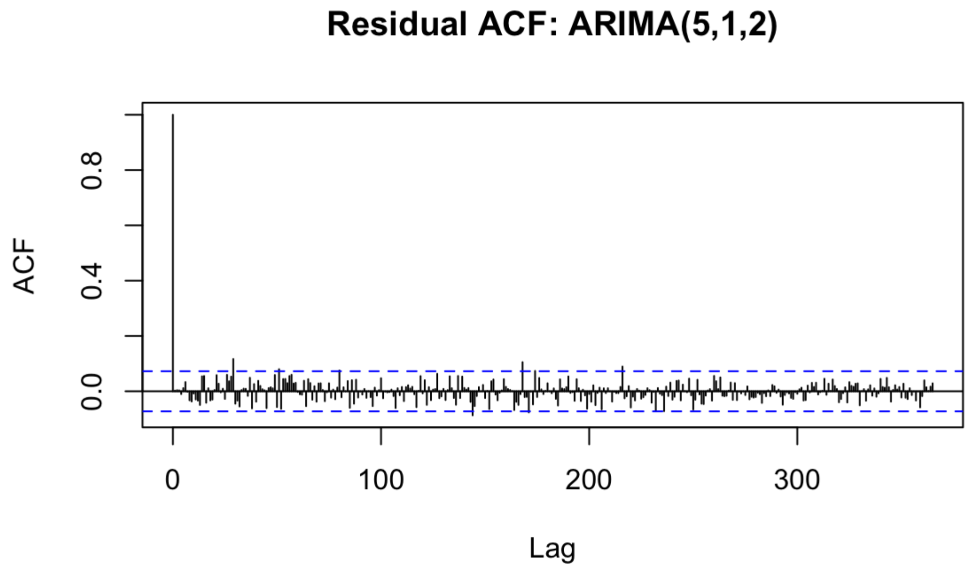
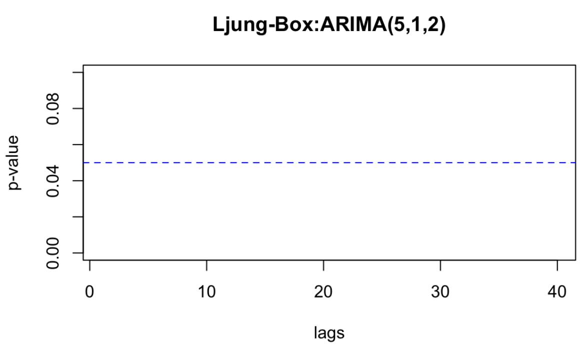
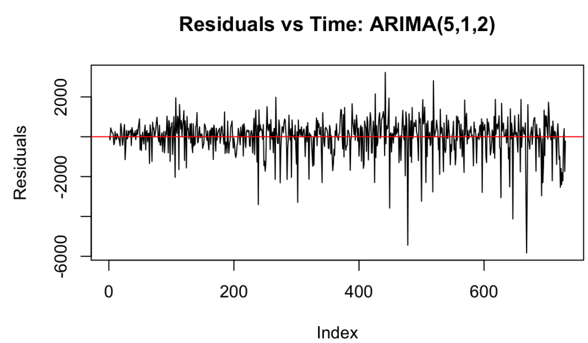


Figure 2.3.4: ARIMA(5,1,2) Residual ACF

No significant correlation except for lag 29. Other significant correlations are at large lags and possibly false positives. We need to take care of lag 29 since this correlation is also revealed in ACF and PACF of the data. We use Ljung-Box test to verify this. The left plot shows the p-values of Ljung-Box test at 40 lags, limited to $[0, 0.1]$, and on the right is a plot of residuals.



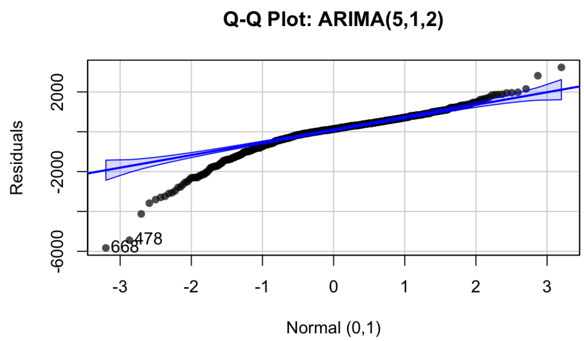
(a) ARIMA(5,1,2) Ljung-Box Test



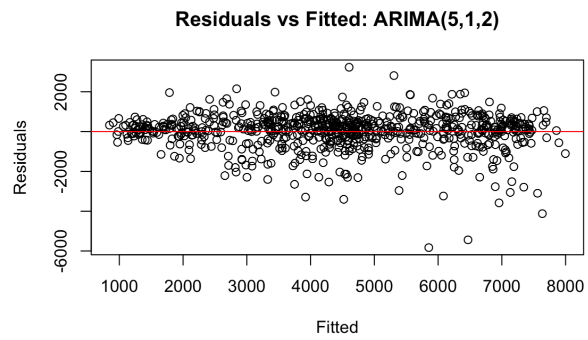
(b) ARIMA(5,1,2) Residuals vs Time

The blank board means all p-values at these 40 lags are at least 0.1, which reveals no significant correlation among the residuals. This also verifies that lag 29 is a false positive. This is a milestone within our project.

Now look at normality and randomness of the residuals.



(a) ARIMA(5,1,2) qq-plot



(b) ARIMA(5,1,2) Residual vs Fitted

Same as the other two models, the residuals have a heavy tail and are not normal. To reduce the non-normality, we conducted a Box–Cox transformation (see appendix A.2) but it turned out not to be very helpful and the λ is around 0.9. Inspirationally, the residuals vs time plot seems to be a random scatter, showing no relationship between them.

Pred.Err	AICc	BIC	apse	mse	normality	randomness
1140.7811	16.513924	16.570337	7698797.8	846084.35	5.42E-21	0.6040962

Its MSE is better than that of both the Holt–Winters and regression models (2.3 and 2.2) which suggests that the differenced model has a better fit than them. However, its APSE is about 4 times that of regression model while slightly outperforming the Holt-Winters model. Its prediction might not be reliable.

Although the normality assumption is violated, we are glad to see the large p-value for the runs test. This indicates that the residuals are actually random, and since they are also uncorrelated, we can conclude that they are white noise (but not normally distributed).

We also tried various SARIMA models (in appendix A.2) with larger p, q, but they have very similar performances such as in AICc/BIC, MSE, or APSE. This model is chosen because it has a small number of parameters and gives relatively good APSE among those SARIMA models. It has a good fit, but due to its short memory, we'd rather use regression or multiplicative Holt-Winters model (with period) for long-term forecasting.

2.4 Conclusion and Statistical Inference

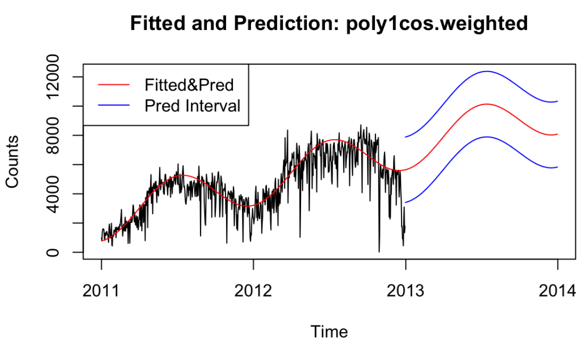
There is no *best* method or *best* model. For instance, the regression model has great prediction power but poor fit, while the differencing methods are good for fit but not prediction, etc. The choice of model should depend on the purpose of analysis, e.g. forecasting future trends or smoothing observed data. Unfortunately, none of these investigated models can characterize the drop in bike rentals during deep winter which is one of the reasons why their residuals have large lower tails.

Chapter 3

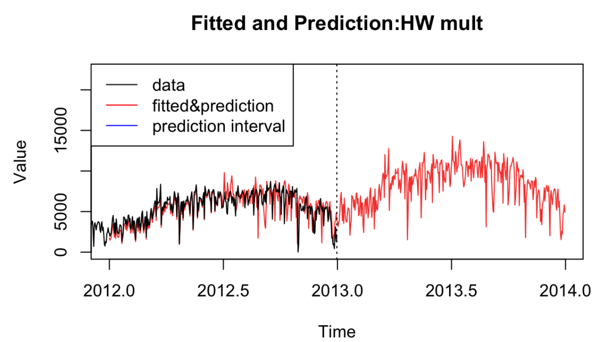
Forecasting, Social Implications, and Future Reactions

3.1 Forecasting

Although the differenced model has the best fit on the bike rentals data, it can only predict a few days ahead. As a result, we consider forecasting using the regression and Holt-Winters models. As a consequence of their poor residual normality, the prediction intervals are unreliable, and we only use them for a rough forecast. To avoid extrapolation issues, we are only looking at the first three months of 2013 and ingnoring all prediction afterward.



(a) Regression



(b) Holt-Winters

3.2 Social Implications and Future Reactions

In this project, we set out to understand how demand for Washington D.C.'s bike-share system evolves over time. We analyzed two years of daily rental counts, and confirmed what we suspected from the raw plots: bike-share usage follows a clear seasonal pattern, with ridership peaking during the summer months, and then dropping down in the wintertime. At the same time, the overall level of usage is trending upward, which suggests that cities operating bike-share programs are likely to continue experiencing growth in adoption, supporting the broader objectives of promoting greener transportation options and improving accessibility.

Understanding these dynamics is valuable for both operators and policymakers. Seasonal trends can inform decisions about fleet allocation, maintenance scheduling, and capacity planning, ensuring that enough bikes are available during peak periods while minimizing service disruptions. Additionally, the long-term upward trend in ridership provides support for continued investment in bike-share infrastructure, reinforcing the role of these programs in sustainable urban mobility.

To capture these patterns, we tried several families of models based solely on the given time and calendar structure. Regression models with trend and yearly seasonality, exponential smoothing via Holt-Winters, and Box-Jenkins time-series models. Each approach highlighted the same story in a slightly different way.

Regression and Holt-Winters models were able to reproduce the broad upward trend, with the recurring summer peaks and winter troughs, while Box-Jenkins models were better at capturing short-term day-to-day dependence. No single model provided a "perfect" forecast, especially when accounting for extreme winter days, however they painted a consistent picture together of rising demand with strong and repeatable seasonal swings.

From an operational standpoint, these findings have direct implications for how a bike-share system should be managed. For instance, the regularity of the seasonal pattern, i.e., the growth in summer, suggest that operators should plan to expand capacity and improve bike availability during peak months, by adding more bikes at busy stations or increasing re-balancing efforts when the weather is good. On the flip side, the predictable drop in winter usage creates a natural window for heavier maintenance, equipment upgrades, and dock repairs, as taking bikes out of service will inconvenience fewer riders. Finally, the long term upward trend also reinforces the case for continued investment in bike-share infrastructure, such as adding new stations and integrating bike-share more closely with public transit.

At the same time, however, our work highlights the limitations of using only strictly time-based information. Large day-to-day fluctuations that are driven by weather, holidays, and special events cannot be fully explained, or predicted without explicitly including those factors, and our models have obvious setbacks when considering extreme days such as snowstorms. Future analyses that incorporate weather, calendar effects, and longer time series would likely yield forecasts that are more precise and more useful for relevant operational decisions.

Nonetheless, even with these limitations, our results support a clear conclusion. Washington D.C.'s bike-share program has been successful, demand is growing, and cities that invest in similar systems can reasonably expect sustained and increasing usage over time.

Appendix A

Supplementary Models

A.1 Regression Models

For 5-fold cross validation error and APSE, we use 20% of the data, which is 146 observations for prediction and the rest 584 observations for training. Particularly, APSE uses the last 20% as validation set.

To achieve the best prediction power, all polynomials present in a model are orthogonally generated. We can thus assume no significant multicollinearity in the models and those relevant diagnostics such as VIF will not be included. Practically, the confidence bands for prediction are reasonably great (around 1.96 times residual standard deviation) across all models.

The distance from each data point to its neighbors is measured as

$$Dist_i = \begin{cases} \text{mean}(|X_i - X_{i-1}|, |X_i - X_{i+1}|) & i = 2, \ldots, n-1 \\ |X_1 - X_2| & i = 1 \\ |X_n - X_{n-1}| & i = n \end{cases}$$

and the weighting function w we used for the weighted model is

$$w_i = 1/Dist_i^2, \quad i = 1, \ldots, n$$

A.1.1 poly4

First start with a degree-4 polynomial as the baseline model,

$$y = \beta_0 + \beta_1 P_1(x) + \beta_2 P_2(x) + \beta_3 P_3(x) + \beta_4 P_4(x) + \epsilon \tag{A.1.1}$$

such that the $P_1(x), \ldots, P_4(x)$ denote 4 orthogonal polynomials with degrees $1, \ldots, 4$ respectively.

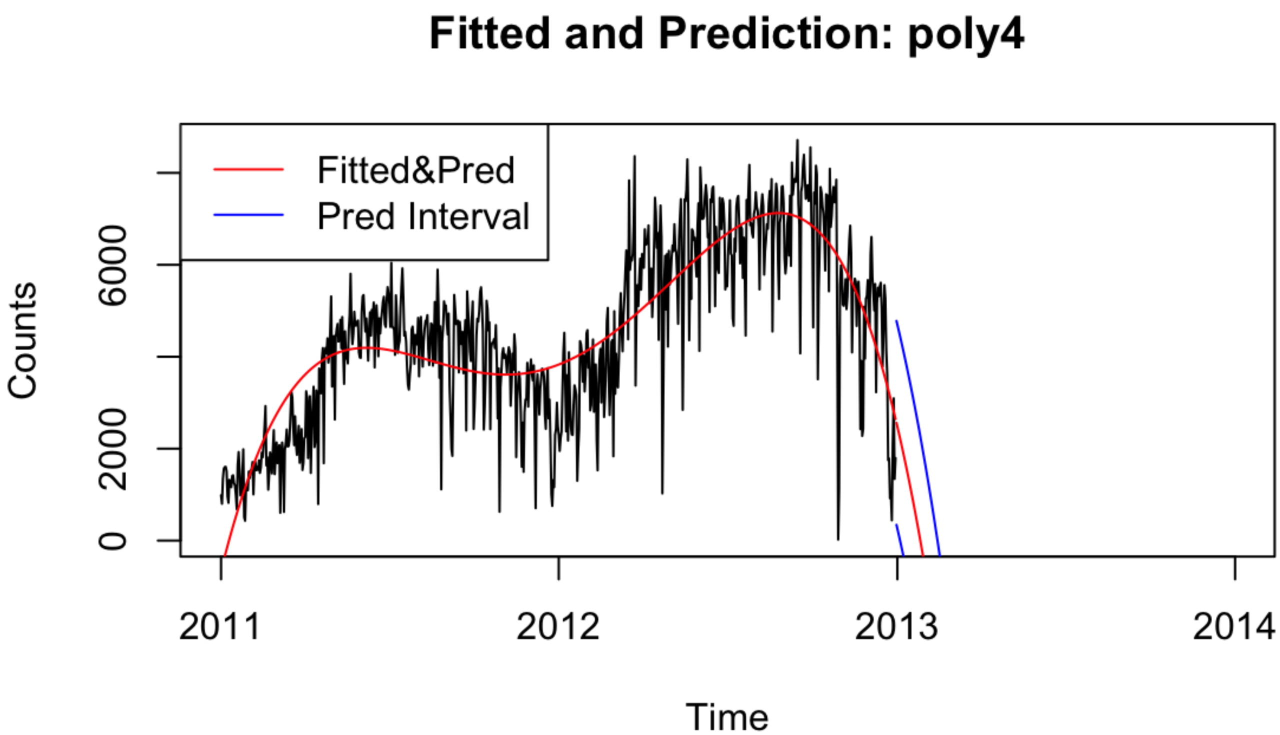


Figure A.1.1: Poly 4 plot

The degree-4 polynomial is not *flexible* enough to fit the curves, but the prediction interval is acceptable since the design matrix X is orthogonal.

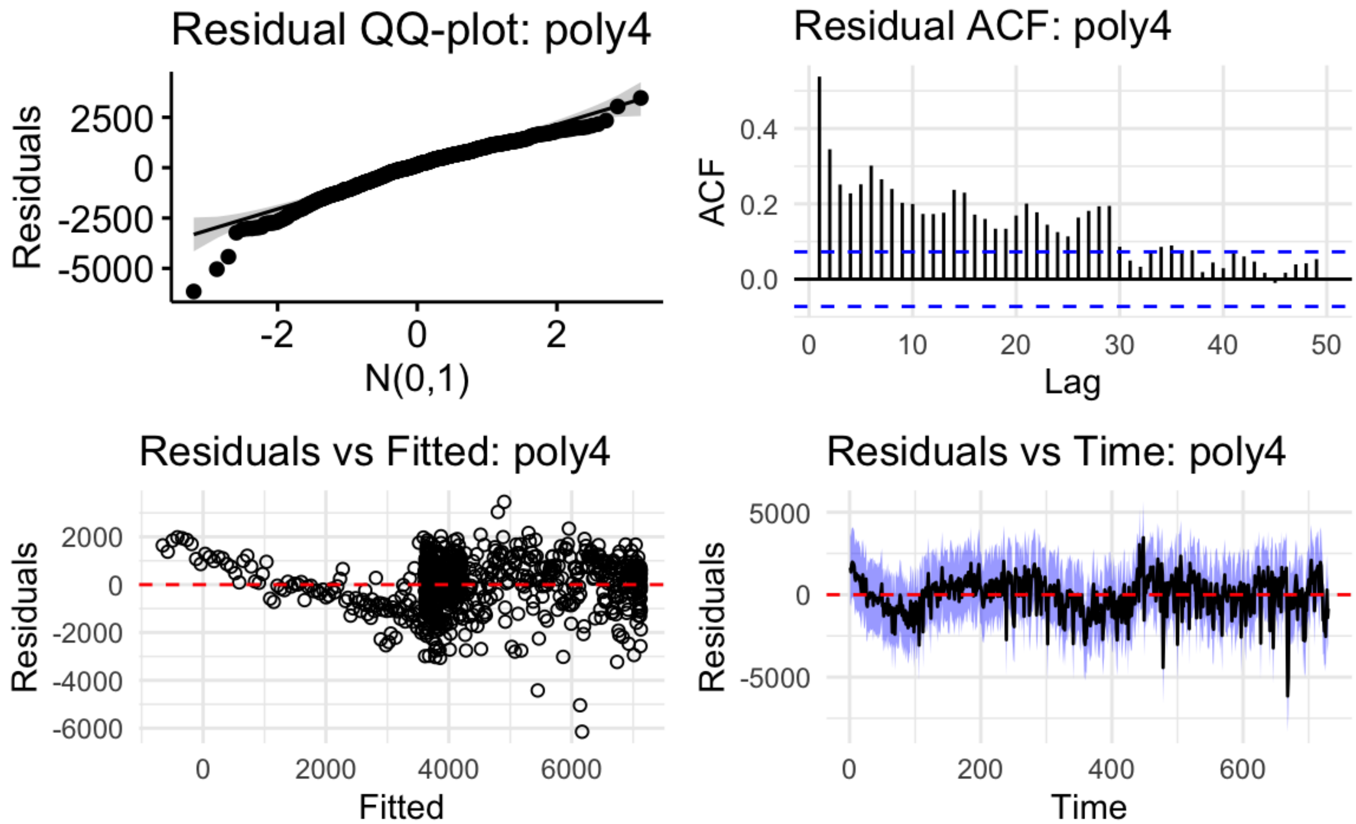


Figure A.1.2: Poly 4 residuals

The 95% confidence bands for residuals are shown as the blue region in the residual vs time plot. The residuals are not normal, not random, and are correlated such that they are not yet qualified as white noise. We will just look at higher degrees.

A.1.2 poly 4-10

We now fit and compare polynomials of degrees from 4 to 10. The models are

$$y = \beta_0 + \sum_{j=1}^{p} \beta_j P_j(t) + \epsilon, \quad p = 4, \dots, 10 \tag{A.1.2}$$

The plots are

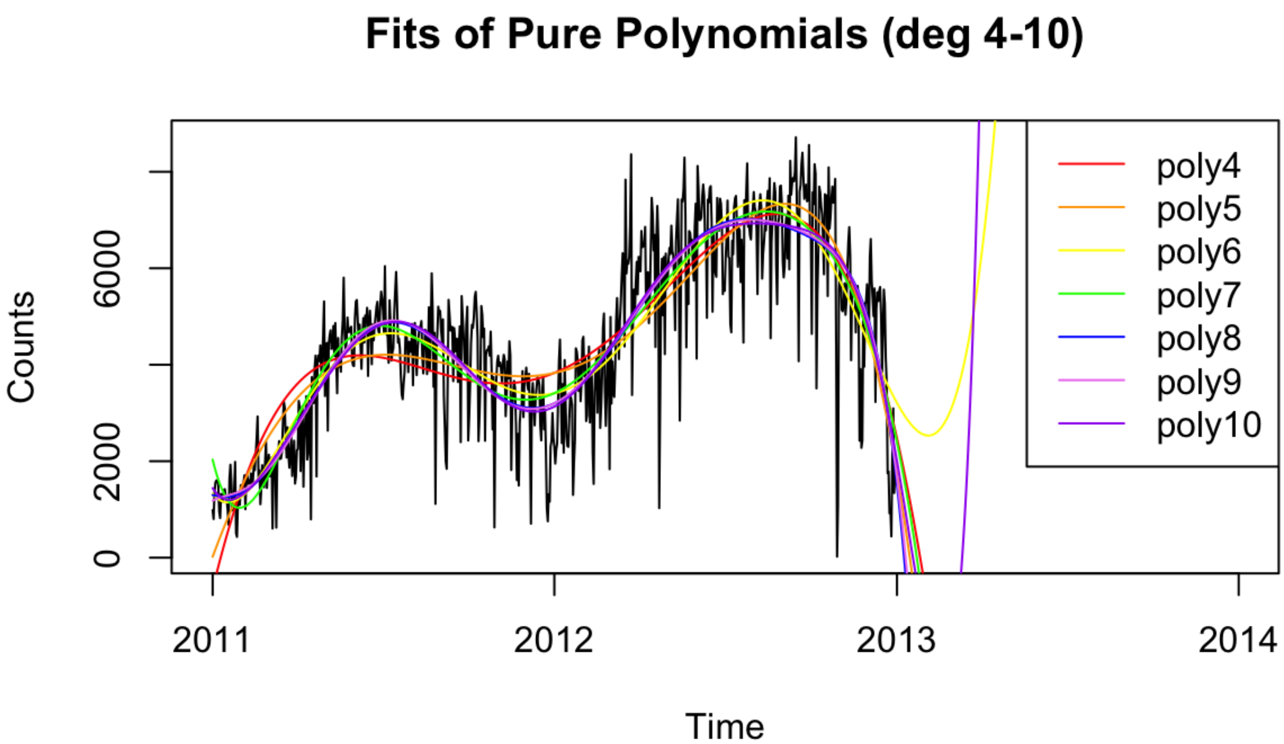


Figure A.1.3: poly 4-10

We have observed overfitting, e.g. the unnecessarily sharp curves in the predicted lines. Most polynomials predict a rapid downward trend that even go below zero, followed by a dramatically steep increase. They might have great fits on the original data but give unreliable predictions. Other analytical details are shown in the comparison table ????????????A.1.5. It is difficult, or at least inefficient, for pure-polynomial models to characterize the entire nature of the data. Simply increasing the degrees does not significantly enhance model fit but definitely exaggerates model complexity. Based on this, only low-degree polynomials will be included in later models.

A.1.3 poly1.cos

As we observed linear trend plus a sinusoidal wave with annual period, we consider

$$y = \beta_0 + \beta_1 t + \beta_2 \cos\left(\frac{2\pi t}{365}\right) + \epsilon \tag{A.1.3}$$

The fitted values and prediction are

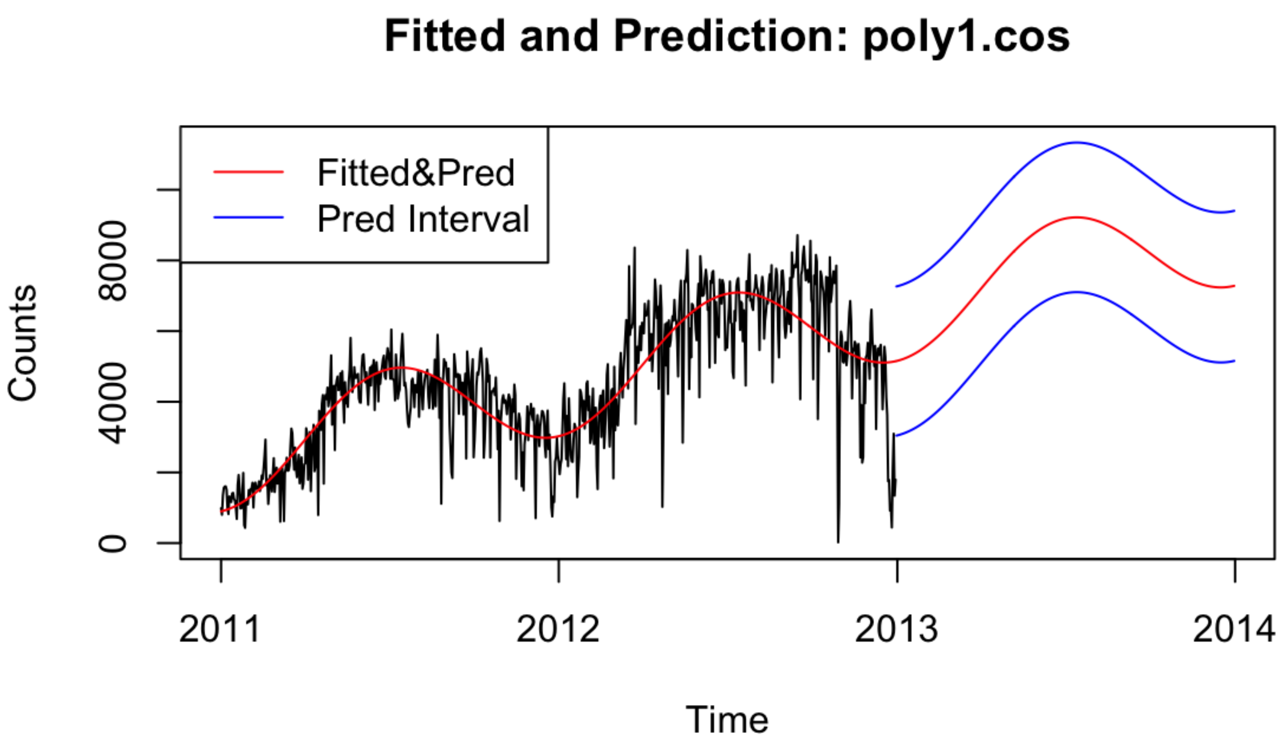


Figure A.1.4: poly1.cos

Note that we do not rely on the entire prediction but only look at the first a few months.

A.1.4 Summary

Model	ResErr	PredErr	AICc	BIC	adjR2	K-Fold	Normalit	Randomnes
poly4	1.1126E+03	2.2310E+03	1.2320E+04	1.2347E+04	6.7022E-01	1.2513E+06	3.6406E-12	7.5316E-39
poly5	1.0936E+03	2.2249E+03	1.2296E+04	1.2328E+04	6.8136E-01	1.2075E+06	2.2172E-12	2.0099E-41
poly6	1.0281E+03	2.1365E+03	1.2207E+04	1.2243E+04	7.1842E-01	1.0672E+06	4.7791E-14	2.9072E-22
poly7	1.0111E+03	2.1672E+03	1.2183E+04	1.2224E+04	7.2762E-01	1.0346E+06	6.4343E-15	1.5238E-23
poly8	9.9434E+02	2.2270E+03	1.2160E+04	1.2206E+04	7.3660E-01	9.9927E+05	2.5766E-16	2.9072E-22
poly9	9.9462E+02	2.3678E+03	1.2161E+04	1.2212E+04	7.3645E-01	1.0071E+06	2.8159E-16	1.2291E-21
poly10	9.9340E+02	2.5665E+03	1.2161E+04	1.2215E+04	7.3709E-01	1.0032E+06	7.6225E-17	7.3219E-25
month	1.0486E+03	2.0839E+03	1.2242E+04	1.2305E+04	7.0706E-01	1.1269E+06	1.0887E-18	6.7286E-23
poly1.weekmonth	1.0424E+03	2.0801E+03	1.2239E+04	1.2330E+04	7.1054E-01	1.1429E+06	6.2636E-19	1.5535E-25
poly1.cos	1.0727E+03	2.1112E+03	1.2265E+04	1.2283E+04	6.9343E-01	1.1548E+06	1.3777E-16	2.9072E-22
poly1.cos.week	1.0670E+03	2.1086E+03	1.2263E+04	1.2309E+04	6.9667E-01	1.1574E+06	1.3341E-16	6.7286E-23
poly1.fourier3	1.0244E+03	2.0245E+03	1.2202E+04	1.2243E+04	7.2044E-01	1.0580E+06	2.0429E-19	7.1928E-24
poly3.cos	1.0564E+03	2.0989E+03	1.2244E+04	1.2272E+04	7.0268E-01	1.1237E+06	6.3632E-15	2.9072E-22
poly3.fourier3	1.0010E+03	2.0140E+03	1.2171E+04	1.2221E+04	7.3303E-01	1.0143E+06	1.7475E-18	8.1536E-20
Best	9.9340E+02	2.0140E+03	1.2160E+04	1.2206E+04	7.3709E-01	9.9927E+05	NA	NA

Figure A.1.5: Model Comparison

From the above models, there are no significant differences between the model performances, especially in prediction error, AICc/BIC, adjusted R squared, and 5-fold cross validation errors. For this reason, we would rather choose the cheapest model poly1.cos

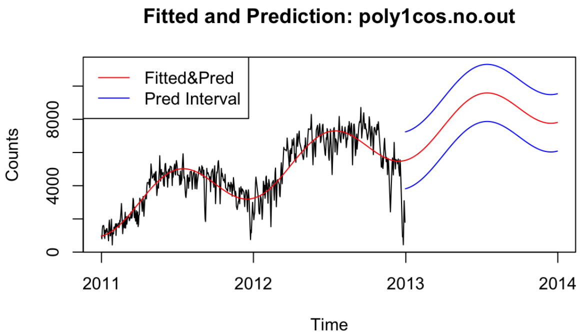
$$y = \beta_0 + \beta_1 t + \beta_2 \cos\left(\frac{2\pi t}{365}\right) + \epsilon$$

for future analysis.

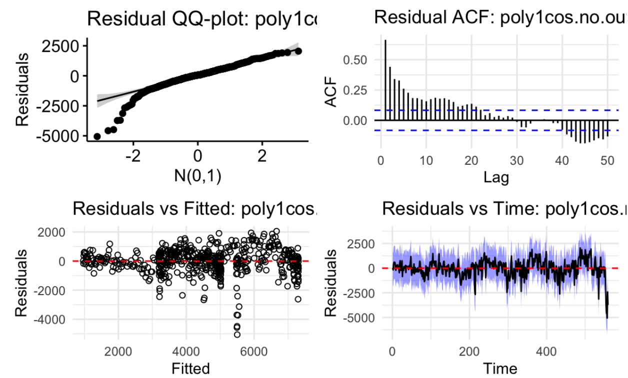
Now we can conclude that ordinary regression models are not sufficient for this dataset. We need manipulations such as weighting, box-cox transformation, and outlier removal.

A.1.5 Weighted, box-cox transformed, and outlier-removed regression models

The weighted regression model is presented in the main content of the report at 2.1.1. Now look at the fit with outliers removed. As we can see in A.1.6a, the data with outliers removed are more "condensed" but we are also losing much data as well as underlying trend information.

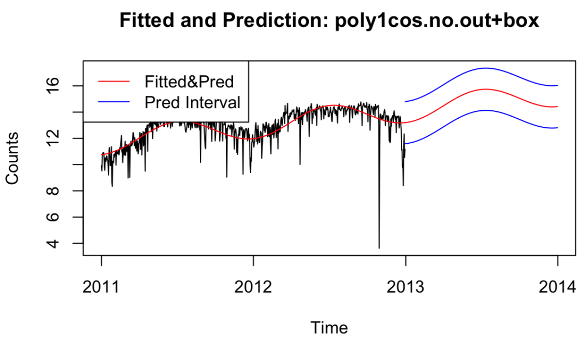


(a) poly1cos.no.out

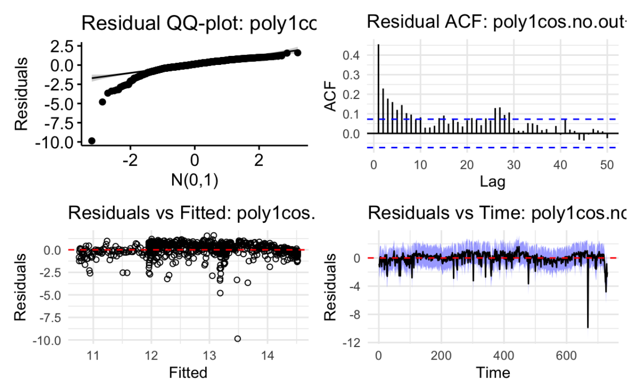


(b) poly1cos.no.out residuals

This is the box-cox transformed model with $\lambda = 0.1$:

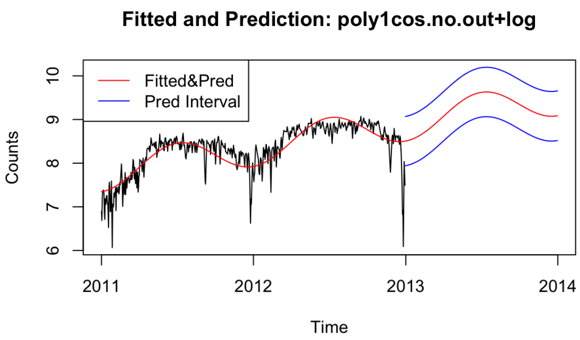


(a) poly1cos.boxcox

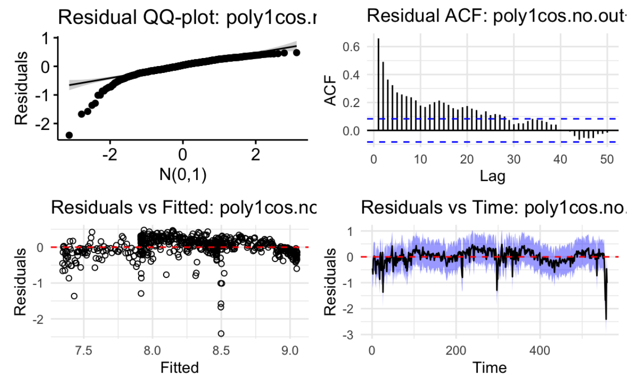


(b) poly1cos.boxcox residuals

Since the λ is close to zero or negative, we try the log transformation with outliers removed:



(a) poly1cos.no.out.log



(b) poly1cos.no.out.log residuals

This a summary of the above models:

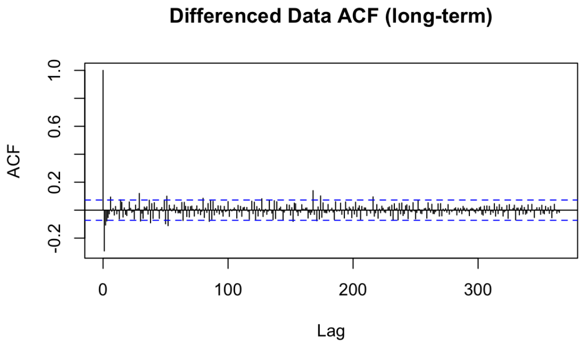
Model	ResErr	PredErr	AICc	BIC	adjR2	APSE	Normalit	Randomnes	lambd
poly1cos.no.out	869.7477362	1714.380694	9158.288593	9175.520989	0.794141814	NA	1.64E-16	1.94E-25	NA
poly1cos.no.out+box	0.816491062	1.606946722	1780.699852	1799.016858	0.588346794	0.676027834	9.97E-31	5.08E-32	0.1
poly1cos.no.out+log	0.286114637	0.563967447	192.4138842	209.6462799	0.717645059	NA	1.32E-23	1.38E-47	0

Figure A.1.9: Model Comparison

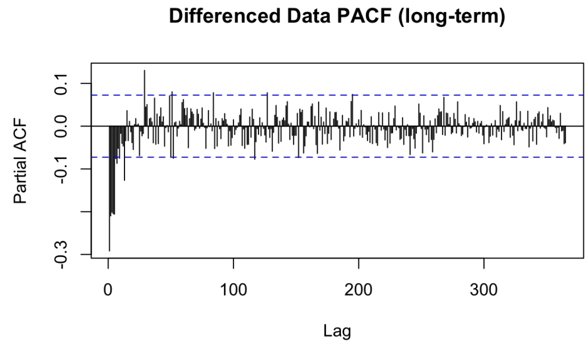
We can see that although the Q–Q plots are closer to normal, we are not able to eliminate the correlation in the residuals. A second-stage model must be considered if we want to apply regression on this dataset.

A.2 Differencing Models

We take a rough look at the long term relationship:

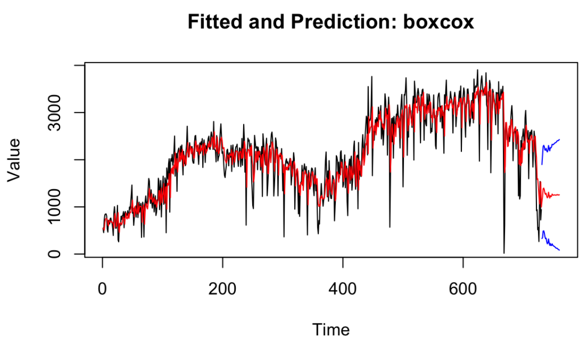


(a) Y_t ACF

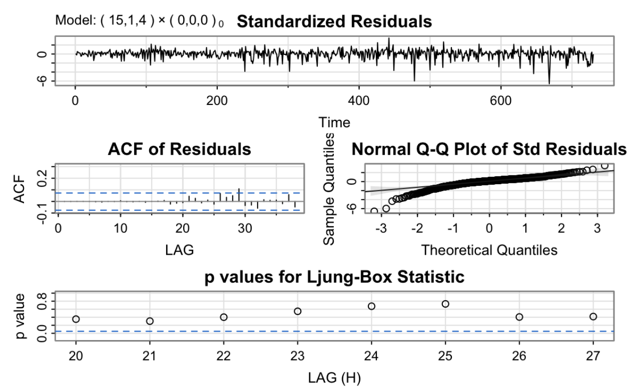


(b) Y_t PACF

The boxcox transformed model (with $\lambda \approx 0.86$):



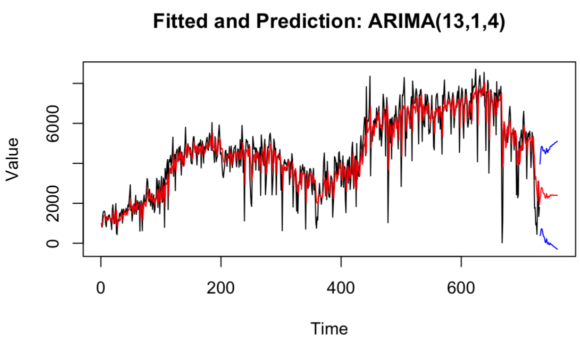
(a) Boxcox Arima



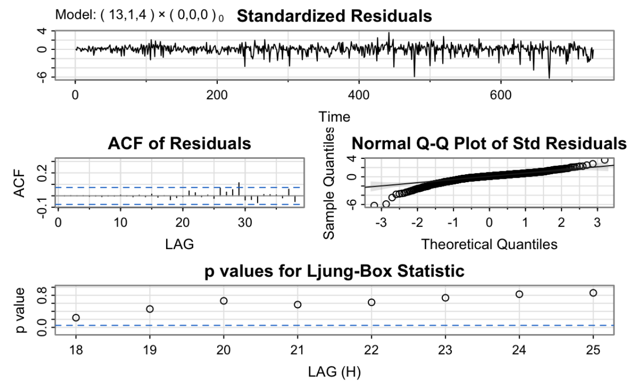
(b) Boxcox Arima Residuals

This is very similar to the untransformed model.

ARIMA(13,1,4)

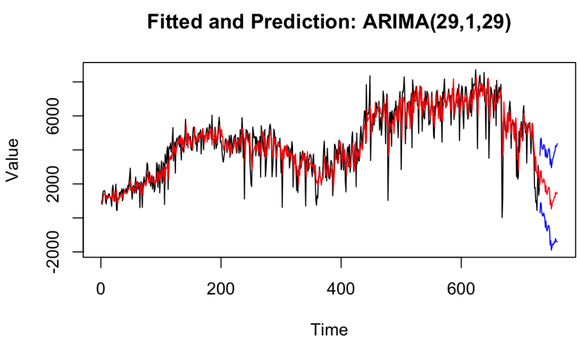


(a) Arima(13,1,4)

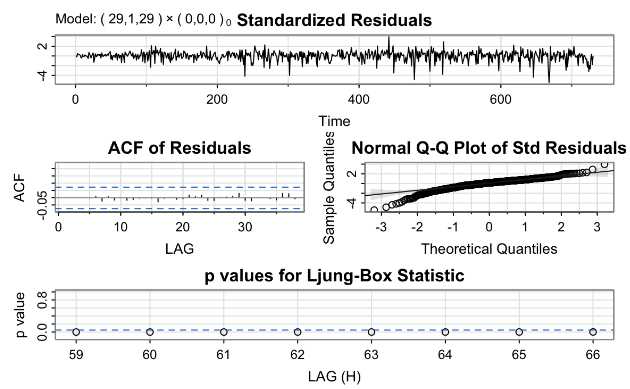


(b) Arima(13,1,4) Residuals

ARIMA(29,1,29)



(a) Arima(29,1,29)



(b) Arima(29,1,29) Residuals

This is a poor model which suggests that the correlation at lag 29 is just a false positive.

Model comparison:

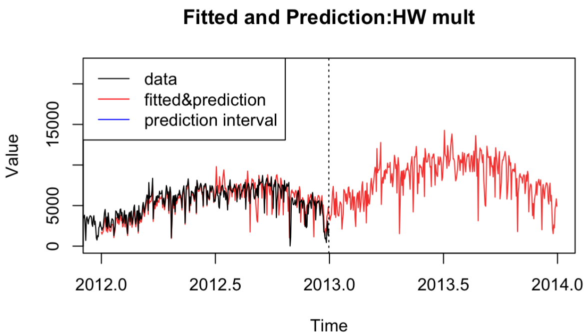
name	Pred.Err	AICc	BIC	apse	mse	normality	randomness
ARIMA(2,1,2)	1134.1724	16.512062	16.549739	7815633.3	851642.92	5.79E-21	0.50498467
ARIMA(2,1,5)	1135.6824	16.51319	16.569603	7641713.4	845438.18	6.64E-21	0.06404685
ARIMA(5,1,2)	1140.7811	16.513924	16.570337	7698797.8	846084.35	5.42E-21	0.6040962
ARIMA(13,1,3)	1133.426	16.52265	16.63485	7762009	831681.8	1.47E-20	0.6040962
ARIMA(13,1,4)	1133.6654	16.525542	16.643893	7732227.3	831684.89	1.48E-20	0.6040962
ARIMA(15,1,4)	1132.3995	16.529892	16.660534	7737019.9	830462.87	1.38E-20	0.6040962
ARIMA(15,1,1)	1146.0662	16.522654	16.634848	7771360.4	831700.23	1.22E-20	0.26652008
ARIMA(2,1,29)	1119.654	16.539587	16.743278	7867567.1	808238.14	6.60E-19	0.94095139
ARIMA(13,1,6)	1120.0619	16.518553	16.649195	7846254.6	820641.1	1.82E-20	0.50498467
ARIMA(29,1,29	1062.0444	16.550961	16.914359	7971430.5	726583.66	8.04E-18	0.71110629

Figure A.2.5: Model Comparison

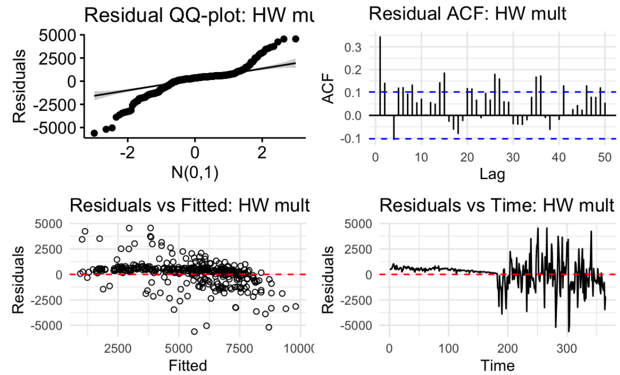
They have similar AICc/BIC and MSE, but ARIMA(5,1,2) and ARIMA(2,1,5) seem to have the best APSE. Considering the simplicity of ARIMA(5,1,2), we choose it as our final model. The residuals of most models are not normal but random and uncorrelated (except for overfitted models such as ARIMA(29,1,29)), hence white noise. The prediction is not very reliable, but the fit is good.

A.3 Holt-Winters Models

Visually, the 365-day Holt-Winters forecast reproduces the broad summer peaks and winter troughs for a future year in a way that the $p = 7$ model does not.



(a) HW mult 365



(b) HW mult 365 Residuals

The residual diagnostics reveal the inadequacy of this model, but it carries rough prediction information for the next year. Its prediction power cannot be assessed since it requires more than two periods of data. Overall, we are not using it for model analysis except for prediction.

Appendix B

Programming Preferences

In this section, we introduce conventional details for understanding our implementations.

B.1 Differencing

Sample usage of basic functions:

```
# load data
load(data)

# set up reusable sarima parameters
ARIMA.1.2.1 <- data.frame(name = 'ARIMA(1,2,1)',
                   p=1, d=2, q=1, P=0, D=0, Q=0, S=0)

# fit sarima model with these parameters
fitted.model <- fit.sarima(data, ARIMA.1.2.1)

# attributes
fitted.modelfitfitfitted
fitted.modelfitfitresiduals
...

# predict useing this sarima model
n.ahead <- 30
# same as the output of sarima.for()
pred.model <- pred.sarima(data, ARIMA.1.2.1, n.ahead)
```

Traditional `sarima()` function can only return residuals but no fitted values, and you need to manually enter parameters for sarima. The function `fit.sarima(data, params)` provides everything needed for analysis such as fitted values, and allows you to reuse the parameter dataframe to skip repeated parameter entering. As an alternative, we can just use a comprehensive function for the entire analysis:

```
    # for ordinary
    performance <- sarima.analysis(data, params, print.summary = TRUE,
                                   plot = TRUE, residual.plot = TRUE)
    # for box-cox transformed
    sarima.analysis.boxcox <- function(data, params, lambda,
                                       print.summary = TRUE,
                                       plot = TRUE,
                                       residual.plot = TRUE)
```

For the box-cox version, just pass the calculated λ to the function and it will calculate APSE, MSE in the original scale.

B.2 Regression

For regression, we used two versions of functions that are designed for different situations.

B.2.1 Algorithm 1: Simple

There are 2 major functions:

```
Fit.Model(model, model.name, weights=FALSE)
    # Takes in a dummy model and return a list
model.summary(model.fit, print.summary = FALSE, plot = FALSE, residual.plot = FALSE)
    # Takes in an object from Fit.Model and perform full analysis
```

Sample usage:

```
poly4 <- poly(t.full, degree = 4)
model.poly4 <- lm(y.dummy ~ poly4)
FIT <- Fit.Model(model.poly4, 'poly4', weights = bike.weights)
model.summary(FIT, print.summary = TRUE, plot = TRUE, residual.plot = TRUE)
```

For efficiency, some global variables are directly used in the function and not passed as arguments. The `poly4` is the degree-4 orthogonal polynomial matrix. Since prediction matrix needs the same structure as training matrix, we generate them in the same way. The `t.full` used here is a vector containing indices of both *training* and *predicted* data, and `y.dummy` is a placeholder of the same length that makes sure `lm` runs without error. Therefore, `poly4` is a concatenation of training and prediction matrices. `model.poly4` is a dummy model used to indicate parameters of the model and provide design matrix[B.2.1], then `Fit.Model()` will extract its matrix, separate into training and prediction matrices, fit true model with training data and return it along with the matrices. After these preparations, `model.summary()` performs whole analyses on the model and returns a list of summarized results.

B.2.2 Algorithm 2: Advanced

Sample usages:

```
# indices of non-outliers
idx.no <- no.outlier.indices(Bike, 1000)

# design matrix for model
poly1cos.no.data <- data.frame(t = t.full, cos = cos(2*pi/365*t.full))

# fit model with no outliers and no box-cox transformation
result1 <- model.analysis(poly1cos.no.data, name = 'poly1cos.no.out',
                       train.idx = idx.no,
                       box.cox.lambda = FALSE,
                       print.summary = TRUE, plot = TRUE, residual.plot = TRUE)

# fit model with outliers but box-cox transformed
result2 <- model.analysis(poly1cos.no.data, name = 'poly1cos.no.out+box',
                       train.idx = FALSE,
                       box.cox.lambda = seq(3, 8, 0.05),
                       print.summary = TRUE, plot = TRUE, residual.plot = TRUE)

# fit model with no outliers but log transformation
result3 <- model.analysis(poly1cos.no.data, name = 'poly1cos.no.out+log',
                       train.idx = FALSE,
                       box.cox.lambda = 0,
                       print.summary = TRUE, plot = TRUE, residual.plot = TRUE)
```

`no.outlier.indices(data, threshold)` excludes the indices of outliers that are defined in A.1. The `model.analysis(data, name, train.idx = FALSE, box.cox.lambda = FALSE, ...)` is a more advanced model fitting function that is particularly designated for outlier removal and Box-Cox transformation. It merges all functionalities of `Fit.Model(), model.summary()` into one function and finds the best box-cox λ as well as removing outliers. The user just needs to provide column data as in `poly1cos.no.data` for training & prediction and the optional indices for non-outliers, then it will show the entire analysis process. The `box.cox.lambda` could be a list of choices or a single number

[B.2.1] via R base function `model.matrix()`

for lambda, or set to `FALSE` to avoid Box-Cox transformation. In the model results and plots, the data will be the transformed ones if Box-Cox transformation is applied.

B.3 Holt-Winters Smoothing

```
hw.result <- hw.analysis(name, data, seasonal,
                         pred.days,
                         params = FALSE,
                         print.summary = TRUE,
                         plot = TRUE, plot.residuals = TRUE)
```

The user can provide customized initial parameters for α, β, γ. It calculates and returns MSE, APSE, and p-values for Shapiro and runs tests, as well as plotting residual diagnostics.

Bibliography

[1] Hadi Fanaee-T. *Bike Sharing*. 2013. URL: https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset.

[2] Kolby Nottingham Markelle Kelly Rachel Longjohn. *The UCI Machine Learning Repository*. 2023. URL: https://archive.ics.uci.edu.