Philosophy: A Fading Subject? How can Universities Adapt to Philosophy's Change in Popularity?

Authors: Yihan(Andrew) Chen, Thy Pham

Abstract

Universities need to continuously adjust their course offerings and programs to cater to the changing demand of incoming students every year. To advise universities on how to scale their philosophy department and programs, we discover the future of the philosophy's popularity through forecasting. We predict the popularity values for the search term "philosophy" on google for every month in the year of 2016 based on the data we have from the year 2004 to 2015 using time series analysis. Then we compare those predictions to the actual values from the year 2016 to judge the prediction model's accuracy. Finally, we predict the popularity values of every month in 2025 to forecast the future of the subject's popularity. When we predicted the 2016 popularity values, they all were within 0 to 5 points of the actual values in 2016. When we continued to forecast results to 2025, we found the numbers to all be below 50. Over time, the popularity of the search term "philosophy" decreases, and implies a declining interest in the subject. This supports the idea that universities should be sizing down their philosophy departments and decreasing the number of philosophy courses.

I. Introduction

Academic relevancy of subjects changes over time. So universities must adapt and change their curriculum and course offerings to accommodate that, but how would universities know which academic fields are in demand?

In this case, we will answer this question for the subject of philosophy: we determine if people will still be interested in this topic in the future through forecasting future predictions based on prior data. With our findings, universities will be able to judge how much money they want to invest in their philosophy programs. Since this is a rather niche topic, there is no existing literature that our proposed solution relates to, or can reference.

We will gauge the interest in philosophy using the google search trend data from 2004-2015, which contains the term's popularity over time (0-100, 100 being peak popularity). With the data, we will summarize its features, perform trend and residual analysis, discuss any significant findings, and conclude with our main discoveries answering the question.

II. Data Description

We are looking at the google search term "philosophy" trend with our value being popularity, quantified by what proportion of peak popularity the search term has at a given time. We extracted this data from the "google.com/trends" website which has values from 2004 to 2024. For our study, we will be using the data from 2004-2015.

II-1. Summary Statistics

The following tables show the 5 point summary of the search term popularity and the data's standard deviation and sample size.

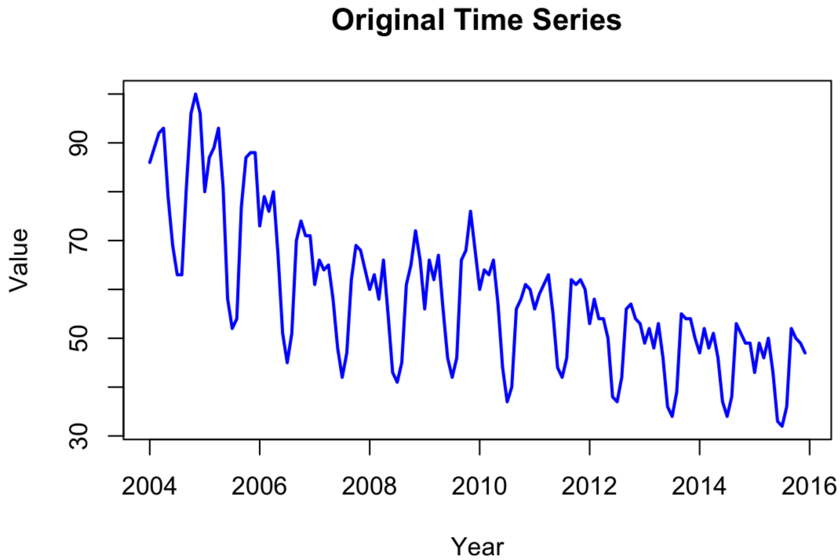
Min	1st Quartile	Median	Mean	3rd Quartile	Max
32	48	56.5	58.85	66	100

Standard Deviation	Sample size
15.11771	144

It seems that popularity of the search term, philosophy, is moderate most of the time. It is usually about half as popular, or a little more, as its peak popularity. Based on the standard deviation of about 15, it seems that there is a large spread of values. This may indicate inconsistent popularity.

II-2. Plot of Popularity Over Time

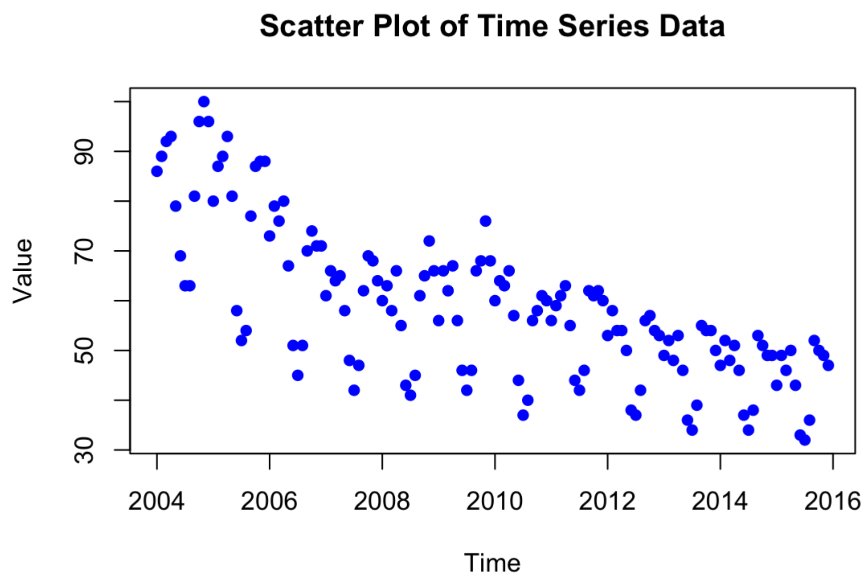
The following plot shows the data plotted as a time series with the value indicating popularity score, over time.



The time series seems to have a seasonality of 1 year, and an overall negative trend over time. For each period, it often spikes once in popularity score, followed by two smaller spikes shortly after before it drops down at the end of the period. The drops appear to be about 20-30 points each time. However, the drops become smaller over time.

II-3. Scatter Plot of Popularity Over Time

The following scatter plot shows each extracted data point of popularity score over time which helps us see the spread and trend of the data.



We can more clearly see there seems to be a linear trend downwards as time moves on similar to the prior plot. There also seems to be dense clustering at higher values and larger spread at lower values across the years. This could be indicating some abnormal variance as the spread between the points are not consistent.

III. Data Analysis

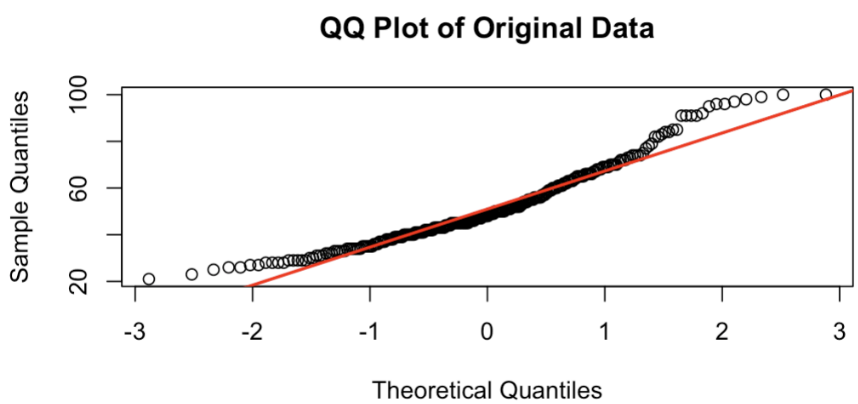
First, in order to eventually fit an ARMA function and forecast predictions, we checked if it's necessary to stabilize the variance of our data, or if transformations to symmetry were necessary through analyzing the time series's normality.

The first step in the analysis involved loading and preprocessing the dataset. The dataset was read from a CSV file while skipping the first row to ensure proper formatting. Column names were then assigned to make the dataset more readable. The dataset was then converted into a time series object with a monthly frequency starting from January 2004 to December 2015.

III-1 Initial Data Transformations

From the previous section, we have some guesses on the data distribution; there seems to be abnormal variance. Though, to further check for the data's distribution, we tested for normality.

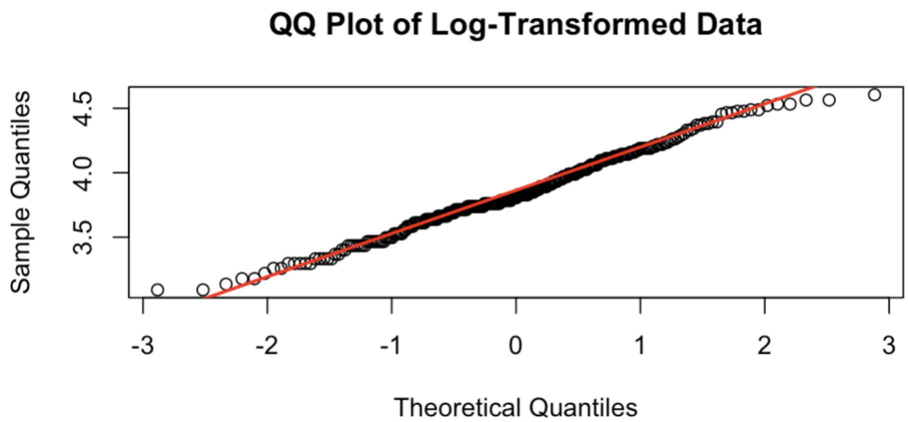
The following QQ plot below shows the distribution of the residuals for the original data with the best fit line, which checks for normality.



From the plot, we can see the residuals of the original time series do not seem to follow a linear model since the data points do not fit the line well at the ends. This suggests that the original time series does not follow a normal distribution. Thus we transform the data in an attempt to achieve normality.

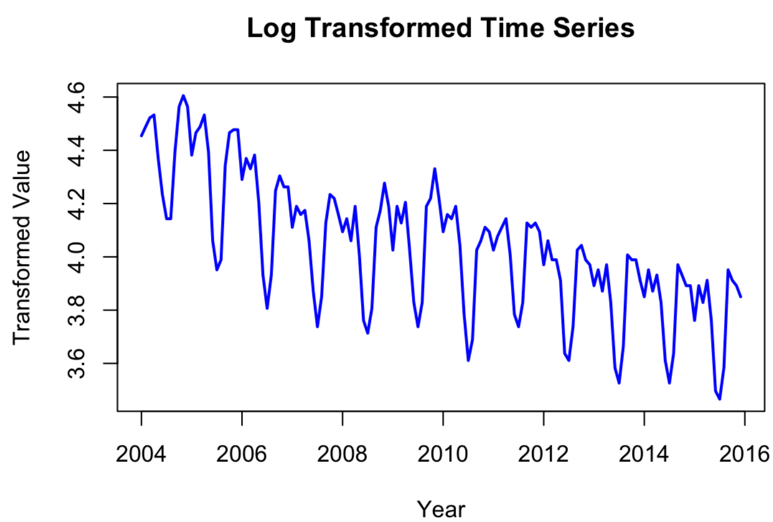
The lambda value obtained from the Box-Cox transformation is close to 0, which suggests that a log transformation is reasonable.

To assess normality again after transformation, a QQ plot was generated for the transformed data.



From the figure above, almost all of the data points lay on the linear best fit line. This indicates that the data's residuals are likely normally distributed, justifying the use of log transformation. Thus we assume that the transformed data is normal and proceed with the rest of our data analysis.

The following plot shows the log transformed time series analysis which can be visually compared to the original time series for analysis.

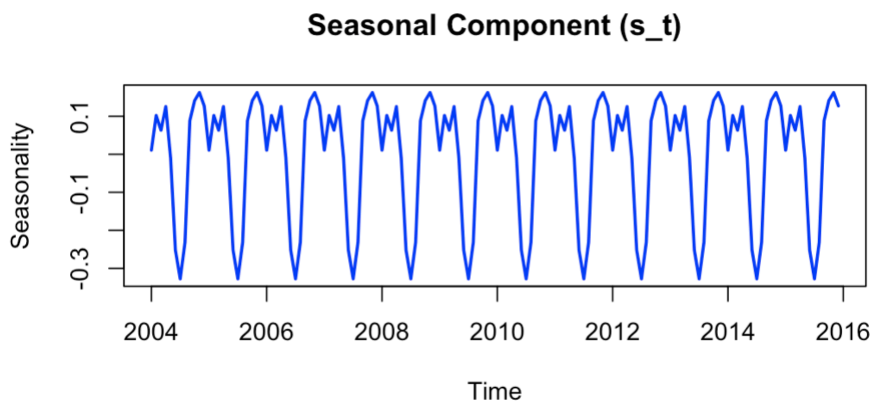


We then started modeling with a classical seasonal decomposition of a time series using the additive model.

III-2. Analyzing the “smooth” component

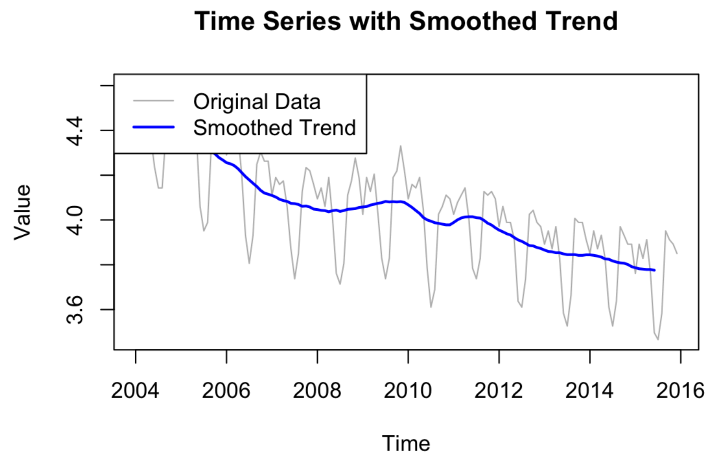
The time series was decomposed into trend, seasonal, and residual components using an additive model.

The seasonal component was extracted and plotted separately below to examine periodic fluctuations.



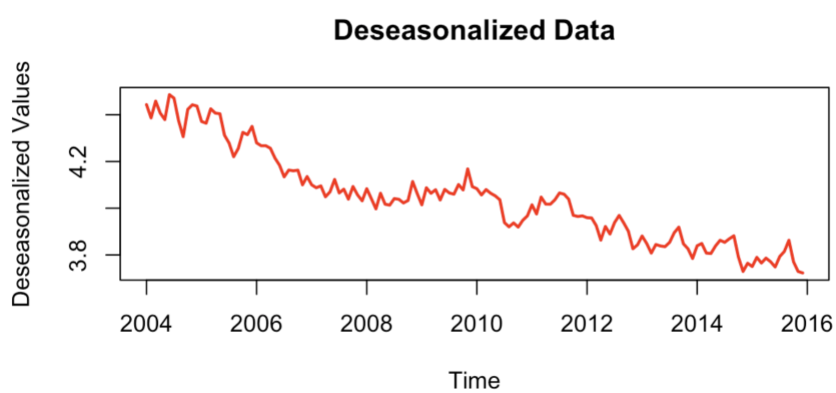
From the seasonal component visualization, we can see the seasonal period is about 1 year and cycles starting mid year.

Then, a moving average smoothing technique with a window size of 12 months was applied to estimate the trend component. The smoothed trend was then overlaid on the original data to visualize long-term movement as shown below.



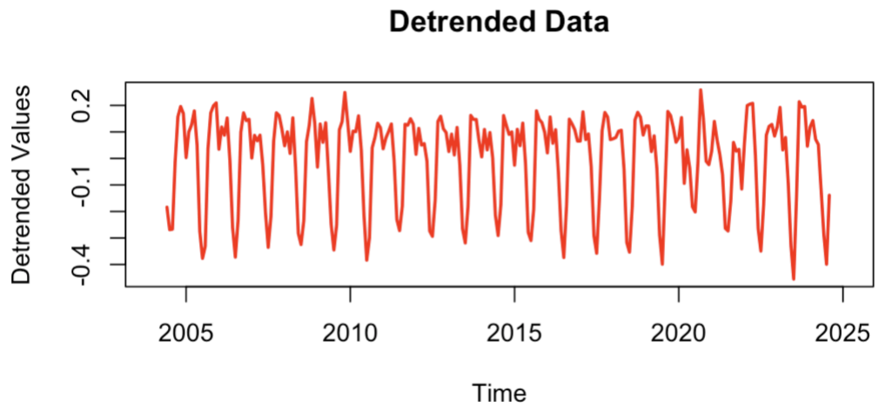
From the smoothed trend, we can see that as time goes on there is a negative linear dependency in popularity of the search term “philosophy”.

To assess the impact of seasonality, the deseasonalized data was computed by removing the seasonal component from the transformed series.



From the deseasonalized data plot above, we can again confirm the search term’s popularity decline, but now we can also see there seems to be a steep decline around 2004 to 2008 before it mellows out until 2010 and keeps dropping in popularity afterwards.

Similarly, detrended data was obtained by subtracting the trend component.

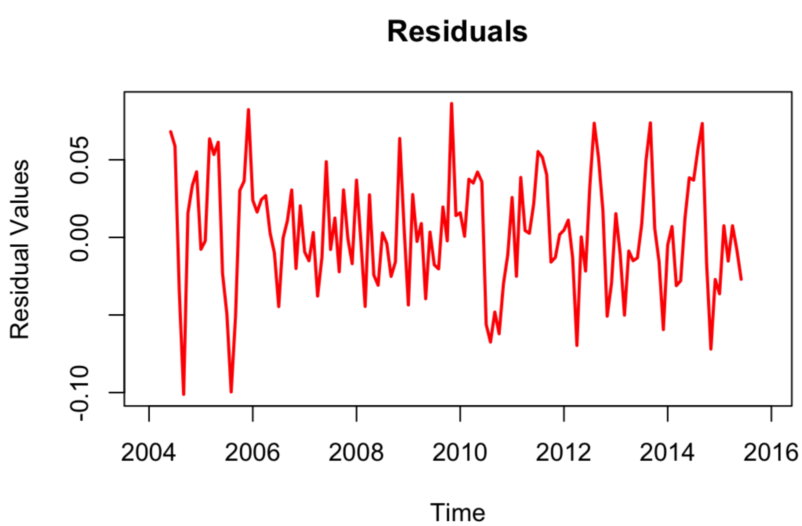


The detrended data in the figure above shows that the seasonality is very consistent as the increases and decreases look nearly identical and very cyclical over time.

III-3. Analyzing the residuals

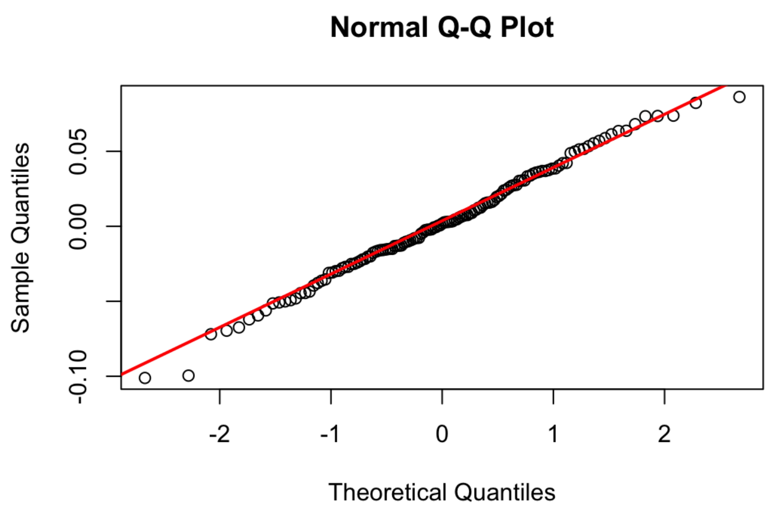
In this section, we will look at the residuals plot and check the residuals for whiteness to make sure our data meets assumptions necessary for a model we will fit later.

The residuals, representing the rough component, were computed by subtracting both trend and seasonality from the transformed data, shown in the plot below.



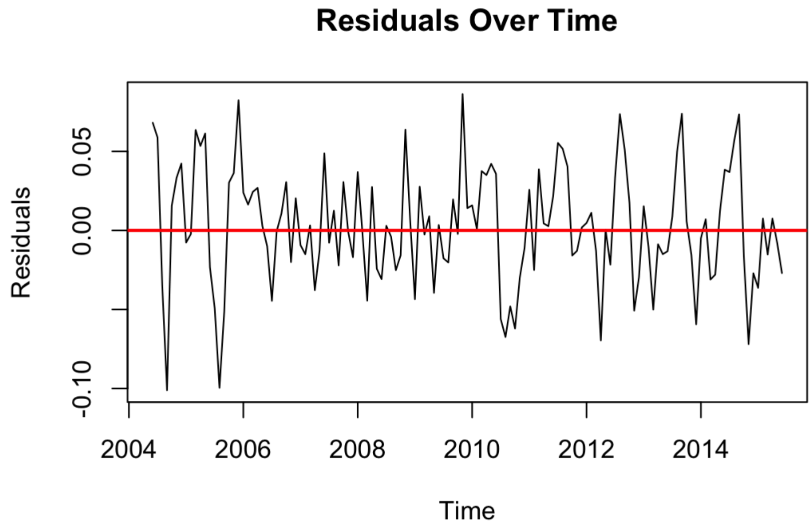
The stationarity of the residuals was checked using the Augmented Dickey-Fuller (ADF) test. Since the p-value of this test returned as 0.01 when our chosen alpha is 0.05, we reject the null hypothesis in favor of the alternative, which states the residuals are stationary. Thus, the test results validate the assumption needed for time series modeling.

Then a QQ plot was used to examine normality of the residuals.



According to the QQ plot, the residuals show normal distribution, which satisfies the assumption.

Additionally, we check for constant variance through the residuals over time chart given below.



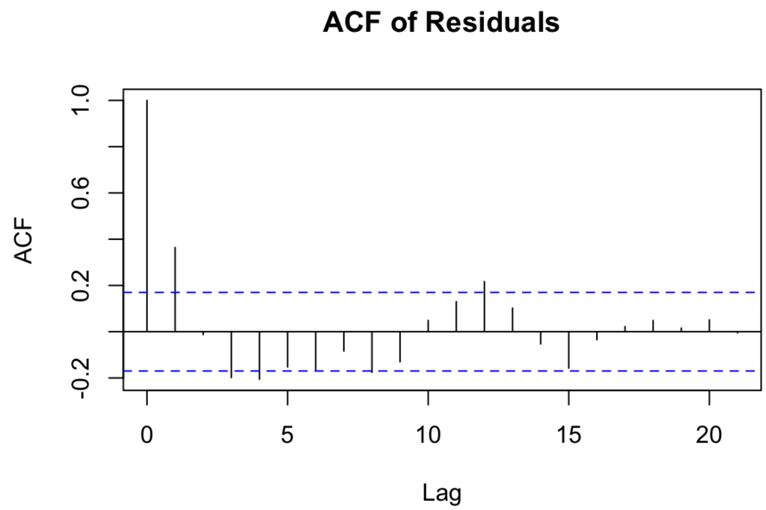
Since the residuals are scattered randomly around 0, we can assume that the data meets the assumption of constant variance.

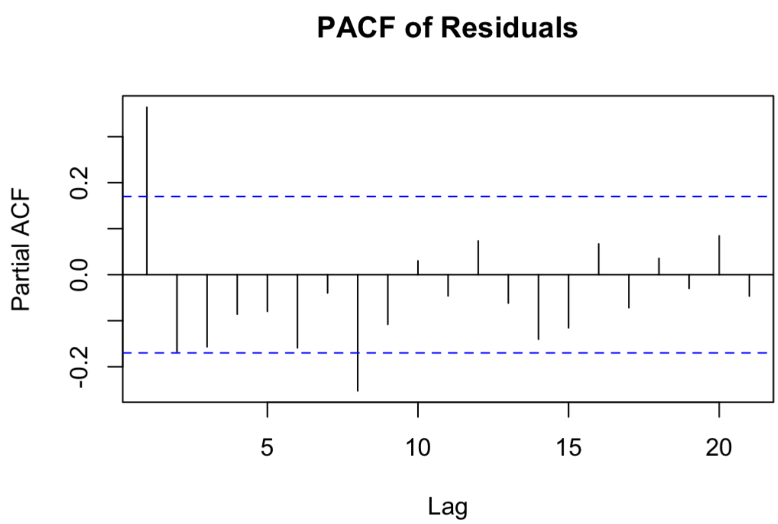
III-4. Analyzing the "rough" component

In this section, we will fit the appropriate ARMA model to the residuals obtained after the analysis of the residuals to help us predict future data points for popularity score.

For this analysis, we have chosen the Akaike Information Criterion (AIC) as the model selection criterion. AIC is well-suited for this project because it prioritizes predictive accuracy while balancing model complexity. Compared with Bayesian Information Criterion (BIC), AIC is more appropriate when the primary focus is on minimizing prediction error. By using AIC, we can ensure that the selected model optimally balances fit and complexity while maximizing predictive performance

First, the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots were generated to identify dependency structures within the residuals.



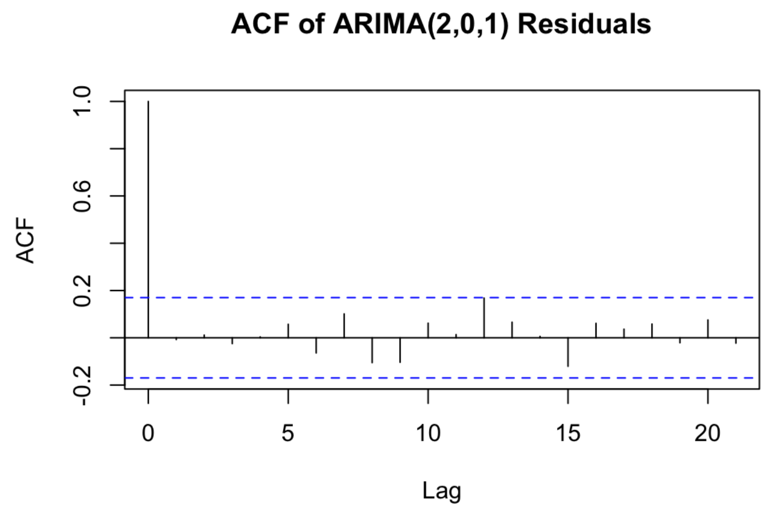


From the ACF plot of the residuals above, we see that the first lag is significant, suggesting a Moving Average (MA) process of order 1 (MA(1)). Similarly, in the PACF plot of residuals in the last section, the first two lags are significant which indicate an Autoregressive (AR) process of order 2 (AR(2)).

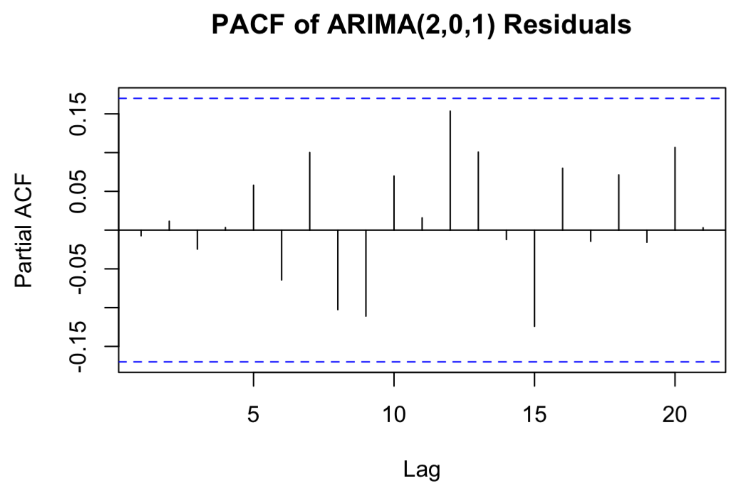
To confirm this, we fit several other ARIMA models and compare their Akaike Information Criterion (AIC) values. Lower-order models such as ARIMA(1,0,0) (AR(1)) and

ARIMA(1,0,1) (ARMA(1,1)) leave significant autocorrelations in the residuals, indicating they do not fully capture the underlying structure. Higher-order models introduce unnecessary complexity and lead to an increase in the AIC value, making them less optimal. Among all tested models, ARIMA(2,0,1) (ARMA(2,1)) achieves the smallest AIC value, significantly lower than both lower- and higher-order alternatives. Thus, ARIMA(2,0,1) is selected as the final model, as it effectively captures the residual structure while minimizing AIC.

To check this selection, we constructed the ACF and PACF plots of the model's residuals.

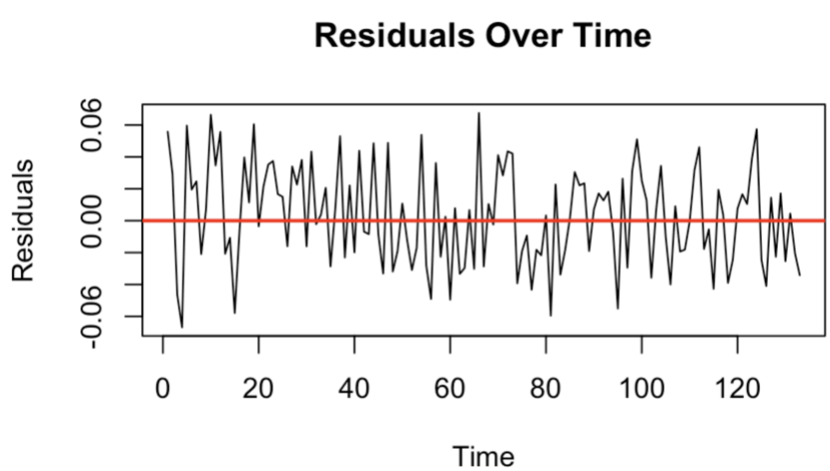


In the ACF plot for ARMA(2,1), the residuals after lag 0 are equivalent to 0, suggesting the rest is white noise. Thereby, it shows that the model captured all dependencies in the time series; there are no more patterns we need to take in account for our fitted model.



Since there are no spikes in the PACF plot of ARMA(2,1), it implies that the residuals are a white noise and there is no uncaptured structure.

For a final check, we take a look at the residuals over time plot to confirm that residuals conform to white noise.



Above the residuals are randomly scattered around 0, so we can assume the model's residuals conform to white noise.

Alas, as the model passed all the checks for appropriate representation of the time series, we can now use this model to predict future values of the “philosophy” search term popularity.

III-5. Predicting Future Values

Before predicting the future value, to assess the uncertainty in the estimated parameters of the ARMA(2,1) model, we compute 95% confidence intervals for the autoregressive (AR) and moving average (MA) coefficients. The confidence intervals provide a range within which the true parameter values are likely to fall, helping us evaluate the reliability of the model estimates.

We extract the estimated AR coefficients and the MA coefficient from the fitted ARMA model. By applying the confidence interval formula, we obtain confidence intervals for phi_1, phi_2, and theta. These intervals help us assess the statistical significance of the parameters. Based on the calculation result, the interval does not contain zero, so it suggests that the corresponding coefficient is likely nonzero, indicating its importance in the model.

We will use the predicted trend, seasonality, and residuals to predict the next 12 future values, one for each month in the year of 2016, of the “philosophy” search term popularity. Then we will compare our predictions with the actual data of that year.

We first fit a linear model to predict the trend component. Then repeat the last seasonal cycle for 12 months because we are trying to find the next 12 months’ values. After, we get the predicted residuals using the Maximum Likelihood Estimation prediction algorithm, and combine all the predicted components to find the final 12 future values of the “philosophy” search term popularity.

To assess the accuracy of our ARMA(2,1) model, we compared its predictions for 2016 with the actual observed values. The 12 predicted values’ are, from January to December in

2016: 42.10656, 46.60350, 44.78011, 47.31643, 40.93496, 31.82971, 29.26761, 32.00034, 43.81425, 45.97961, 46.77036, 44.96864. The actual 12 values from 2016, from January to December: 43, 49, 44, 46, 39, 31, 28, 34, 47, 46, 45, 43. The values for 2016 ranged from 29.27 to 47.32, which closely align with the actual data (28 to 49). Some deviations were noted, such as overestimation in July and August and underestimation in February and September. Despite minor errors, the general alignment between predicted and actual values indicates that the model is reliable. Therefore, we conclude that the model successfully captured seasonal fluctuations and overall trends.

Based on the validated model, we extended our forecast to 2025. The predictions for the year of 2025 from January to December: 39.99004, 44.26216, 42.52968, 44.93779, 38.87701, 30.22980, 27.79689, 30.39268, 41.61339, 43.67006, 44.42096, 42.70952. The predicted values for 2025 range from 27.80 to 44.94, showing a similar seasonal pattern to 2016, with expected dips in the summer months (June–August) and peaks in the beginning and end of the year. A key observation is that the predicted values for each month of 2025 are consistently lower than their counterparts in 2016. This aligns with the overall declining trend component identified in our time series analysis, suggesting a general downward shift in future values. Although the actual data for 2025 is unknown, this forecast serves as a projection based on historical patterns.

IV. Discussion

Through our data analysis, we found that our popularity data for the search term "philosophy" over 2004-2015 had both a trend and seasonality, which allowed us to map it to a time series function. From there, we were able to fit an ARMA model to our time series function in order to predict the next 12 popularity values in the next year, 2016. We found that our forecasted values followed a similar pattern to the actual values, but the predictions' changes in value were smaller than the actual values. This smoothing effect likely resulted from our trend extraction method, which involved moving averages. Since moving averages smooth out short-term fluctuations by averaging past values to find the trend.

An alternative approach we could have used is polynomial regression, which fits a polynomial function to model the trend. This method allows for more flexibility in capturing nonlinear growth or decline, potentially providing a more accurate representation of complex trend behaviors. However, polynomial regression has its own limitations—it is sensitive to overfitting, especially when using higher-degree polynomials, which can lead to unrealistic fluctuations in the estimated trend.

Ultimately, our choice of moving averages provided a stable and interpretable trend, but exploring polynomial regression could have allowed us to capture more intricate trend dynamics.

A possible limitation of our study could be from using the AIC model criteria. Since our goal was to forecast future predictions of the popularity score of the search term "philosophy", we went with the best model determined by AIC. However, AIC tends to favor more complex models because its penalty for additional parameters is relatively weaker compared to other criteria like BIC. Therefore, we may be selecting a slightly overfitted model that includes unnecessary factors in the model.

There is also a risk in long-term prediction: Our prediction for 2025 assumes that past trends and seasonal patterns continue into the future. However, search behavior is influenced by unpredictable external factors. Therefore, the further we extend our forecast, the greater the uncertainty in our predictions.

V. Conclusion

Our analysis of Google search trends for the word "philosophy" reveals a gradual decline in popularity over time. Based on historical data, the model suggests that search interest peaked during specific months, particularly at the beginning and end of each year, while dipping in the summer months. This seasonal pattern has remained consistent and is projected to continue into 2025, though at an overall lower level compared to 2016. Our prediction of the popularity of philosophy in 2025 from January to December is: 39.99004, 44.26216, 42.52968, 44.93779, 38.87701, 30.22980, 27.79689, 30.39268, 41.61339, 43.67006, 44.42096, 42.70952.

Since philosophy is a highly academically related term, we can reasonably assume that a significant portion of searches come from students and scholars engaging with the subject for academic purposes. The summer decline in search activity likely corresponds to academic breaks, when fewer students and researchers are actively engaged in coursework or scholarly pursuits. Given the overall downward trend, this could indicate a declining interest in philosophy as a field of study, possibly reflecting fewer students choosing it as a major or fewer academic discussions surrounding the subject. For universities, this would be a signal to size down their philosophy department and course offerings as the subject's demand will be decreasing in the future, evidenced by year 2025's popularity value all being below 50.

This decline may also be part of a broader societal shift in how people engage with philosophical topics. With the rise of short-form content, social media, and AI-driven information retrieval, fewer people may be turning to traditional searches to explore philosophical concepts. Additionally, philosophy often requires deep reflection, which may contrast with modern digital consumption patterns favoring quick, practical knowledge. While search frequency is decreasing, this does not necessarily imply that philosophy itself is losing relevance—it may simply indicate that discussions are happening in different formats and platforms beyond traditional search engines.

VI. R appendix

```
knitr::opts_chunk$set(echo = FALSE)
knitr::opts_chunk$set(fig.width=6, fig.height=4)
library(forecast)
library(tseries)
library(tsoutliers)
# Read the CSV file while skipping the first row
data <- read.csv('/Users/tee._.thy/Desktop/ucd/davis y2/wq25/sta 137/sta 137 project/Popul
# Summary Statistics
fivenum(data$philosophy...United.States.)
mean(data$philosophy...United.States.)
sd(data$philosophy...United.States.)
nrow(data)
# Rename columns properly
colnames(data) <- c("Month", "Value")

# Convert to time series object
ts_data <- ts(data$Value, start = c(2004, 1), frequency = 12)

# Print and plot
print(ts_data)
plot(ts_data, main = "Time Series Data", ylab = "Value", xlab = "Year")
plot(ts_data, type = "p",
     main = "Scatter Plot of Time Series Data",
     xlab = "Time", ylab = "Value",
     col = "blue", pch = 16)


# Find optimal lambda for Box-Cox
lambda <- BoxCox.lambda(ts_data)

# Apply Log Transformation
data$Transformed_Value <- log(data$Value)

# Convert transformed data to time series
ts_data_transformed <- ts(data$Transformed_Value, start = c(2004,1), frequency = 12)

# Plot the transformed time series
plot(ts_data_transformed, main = "Log Transformed Time Series",
     ylab = "Log(Value)", xlab = "Year", col="blue", lwd=2)
```

```r
# QQ Plot for Original Data
qqnorm(ts_data, main = "QQ Plot of Original Data")
qqline(ts_data, col = "red", lwd = 2)

# QQ Plot for Transformed Data
qqnorm(ts_data_transformed, main = "QQ Plot of Log-Transformed Data")
qqline(ts_data_transformed, col = "red", lwd = 2)


# Before transformation
plot(ts_data, main = "Original Time Series", ylab = "Value", xlab = "Year", col="blue", lwd=2)
```

```r
# After transformation
plot(ts_data_transformed, main = "Log Transformed Time Series", ylab = "Transformed Value"


# Decompose the log-transformed time series (additive model)
decomposition <- decompose(ts_data_transformed, type = "additive")

# Extract and plot the seasonal component
s_hat <- decomposition$seasonal
plot(s_hat,
     main = "Seasonal Component (s_t)",
     ylab = "Seasonality",
     xlab = "Time", col = "blue", lwd = 2)

window_size <- 12

# Apply moving average smoothing
m_hat <- filter(ts_data_transformed, rep(1/window_size, window_size), sides = 2)

plot(ts_data_transformed, main = "Time Series with Smoothed Trend", ylab = "Value", xlab =
lines(m_hat, col = "blue", lwd = 2)  # Add trend line
legend("topleft", legend = c("Original Data", "Smoothed Trend"), col = c("gray", "blue"),


# Deseasonalize and detrend the data by removing seasonality
deseasonalized_data <- ts_data_transformed - s_hat
detrended_data <- ts_data_transformed - m_hat
# Residuals:
residuals <- ts_data_transformed - m_hat - s_hat
```

```r
# Plot the deseasonalized and detrended data and residuals
plot(deseasonalized_data,
     main = "Deseasonalized Data",
     ylab = "Deseasonalized Values",
     xlab = "Time", col = "red", lwd = 2)
plot(detrended_data,
     main = "Detrended Data",
     ylab = "Detrended Values",
     xlab = "Time", col = "red", lwd = 2)
plot(residuals,
     main = "Residuals",
     ylab = "Residual Values",
     xlab = "Time", col = "red", lwd = 2)

residuals_clean <- na.omit(residuals)
# Perform ADF test on the residual component
adf_test_res <- adf.test(residuals_clean, alternative = "stationary")
print(adf_test_res)


# Check normality
hist(residuals_clean, breaks = 20, main = "Histogram of Residuals", col = "lightblue")
qqnorm(residuals_clean)
```

```r
qqline(residuals_clean, col = "red", lwd = 2)

# Check Homoscedasticity (Constant Variance)
plot(residuals_clean, main = "Residuals Over Time", ylab = "Residuals", xlab = "Time")
abline(h = 0, col = "red", lwd = 2)

# Plot ACF and PACF of residuals
residuals_ts <- ts(residuals_clean)
acf(residuals_ts, main = "ACF of Residuals")
pacf(residuals_ts, main = "PACF of Residuals")

# Model selection
arima(residuals_ts)
arima(residuals_ts, order = (c(1,0,0)))
arima(residuals_ts, order = (c(1,0,1)))
arima(residuals_ts, order = (c(2,0,0)))
arima(residuals_ts, order = (c(2,0,1)))
arima(residuals_ts, order = (c(2,0,2)))
arima(residuals_ts, order = (c(3,0,1)))
arima(residuals_ts, order = (c(2,0,3)))
```

```r
# Fit the ARIMA(2,0,1) model
arma_model <- arima(residuals_ts, order = c(2, 0, 1))

# Extract residuals from the fitted model
arma_residuals <- residuals(arma_model)

# Plot ACF of residuals
acf(arma_residuals, main = "ACF of ARIMA(2,0,1) Residuals")

# Plot PACF of residuals
pacf(arma_residuals, main = "PACF of ARIMA(2,0,1) Residuals")

# Plot residuals of arma model
plot(arma_residuals, type = "l", main = "Residuals Over Time", ylab = "Residuals", xlab = "Time")
abline(h = 0, col = "red", lwd = 2)


forecast_horizon <- 12
# Predict the trend component
time_index <- 1:length(m_hat)
trend_model <- lm(m_hat ~ time_index, na.action = na.exclude)
future_time <- (length(m_hat) + 1):(length(m_hat) + forecast_horizon)
trend_forecast <- predict(trend_model, newdata = data.frame(time_index = future_time))
trend_forecast <- ts(trend_forecast, start = c(2016, 1), frequency = 12)

# Predict the seasonal component
seasonal_forecast <- tail(s_hat, 12)
seasonal_forecast <- ts(seasonal_forecast, start = c(2016, 1), frequency = 12)

# Predict residuals using MLE
```

```r
arma_model_mle <- arima(residuals_ts, order = c(2, 0, 1), method = "ML")
sum_model_residual <- summary(arma_model_mle)
arma_forecast <- predict(arma_model_mle, n.ahead = forecast_horizon)

residuals_forecast <- arma_forecast$pred
residuals_forecast <- ts(residuals_forecast, start = c(2016, 1), frequency = 12)

# Combine all component and predict 2016
forecast_2016 <- trend_forecast + seasonal_forecast + residuals_forecast
print(forecast_2016)
forecast_original_2016 <- exp(forecast_2016)

# Extract phi_hat and se_phi from the model
phi_hat <- arma_model_mle$coef[1:2]
se_phi <- sqrt(diag(arma_model_mle$var.coef))[1:2]
theta_hat <- arma_model_mle$coef[3]
se_theta <- sqrt(diag(arma_model_mle$var.coef))[3]
# Compute 95% confidence intervals
conf_int_phi1 <- phi_hat[1] + c(-1.96, 1.96) * se_phi[1]
conf_int_phi2 <- phi_hat[2] + c(-1.96, 1.96) * se_phi[2]
conf_int_theta <- theta_hat + c(-1.96, 1.96) * se_theta


# Predict 2025
forecast_horizon_2025 <- 12
future_time_2025 <- (length(m_hat) + 13):(length(m_hat) + 24)
trend_forecast_2025 <- predict(trend_model, newdata = data.frame(time_index = future_time_2025))
seasonal_forecast_2025 <- seasonal_forecast
arma_forecast_2025 <- predict(arma_model_mle, n.ahead = forecast_horizon_2025)
residuals_forecast_2025 <- ts(arma_forecast_2025$pred, start = c(2025, 1), frequency = 12)
trend_forecast_2025 <- ts(trend_forecast_2025, start = c(2025, 1), frequency = 12)
seasonal_forecast_2025 <- ts(seasonal_forecast_2025, start = c(2025, 1), frequency = 12)

# Final Forecast for 2025
final_forecast_2025 <- trend_forecast_2025 + seasonal_forecast_2025 + residuals_forecast_2025
print(final_forecast_2025)
final_forecast_2025 <- exp(final_forecast_2025)
```