

Time Series Analysis of Final Jeopardy Questions

Introduction

It is rare that a television program not only stretches beyond three decades, but continues to enthral audiences of all ages with various backgrounds, interests, and intellects. Since 1984 Alex Trebek has entered our hearts and living rooms to praise contestants on correct responses while smugly fixing their errors.

Many interesting Jeopardy time series could be modelled. One that easily comes to mind would be for the expected nightly winnings since each episode's champion remains and can last weeks before being eliminated—a clear correlation between neighbouring payouts.

However, my interest in fitting Jeopardy data to autoregressive integrated moving average (ARIMA) models is more selfish. With the high-level of game theory involved, I believe the most interesting part of the game is Final Jeopardy when contestants wager any amount of their daily purse to surpass their two opponents and be crowned champion. Since I began regularly watching during university, I have always felt confident at the unveiling of certain Final Jeopardy categories, while other categories guarantee my struggle through a clueless reasoning process as the theme song whistles away.

It is well known by the Jeopardy faithful that certain categories have higher repetition rates than others, and I think it will be interesting to explore ARIMA(p,d,q) models for the annual percentages of Final Jeopardy questions falling in popular categories. If a suitable model exists, I will be able to predict the successive years' percentage for that Final Jeopardy category.

Data

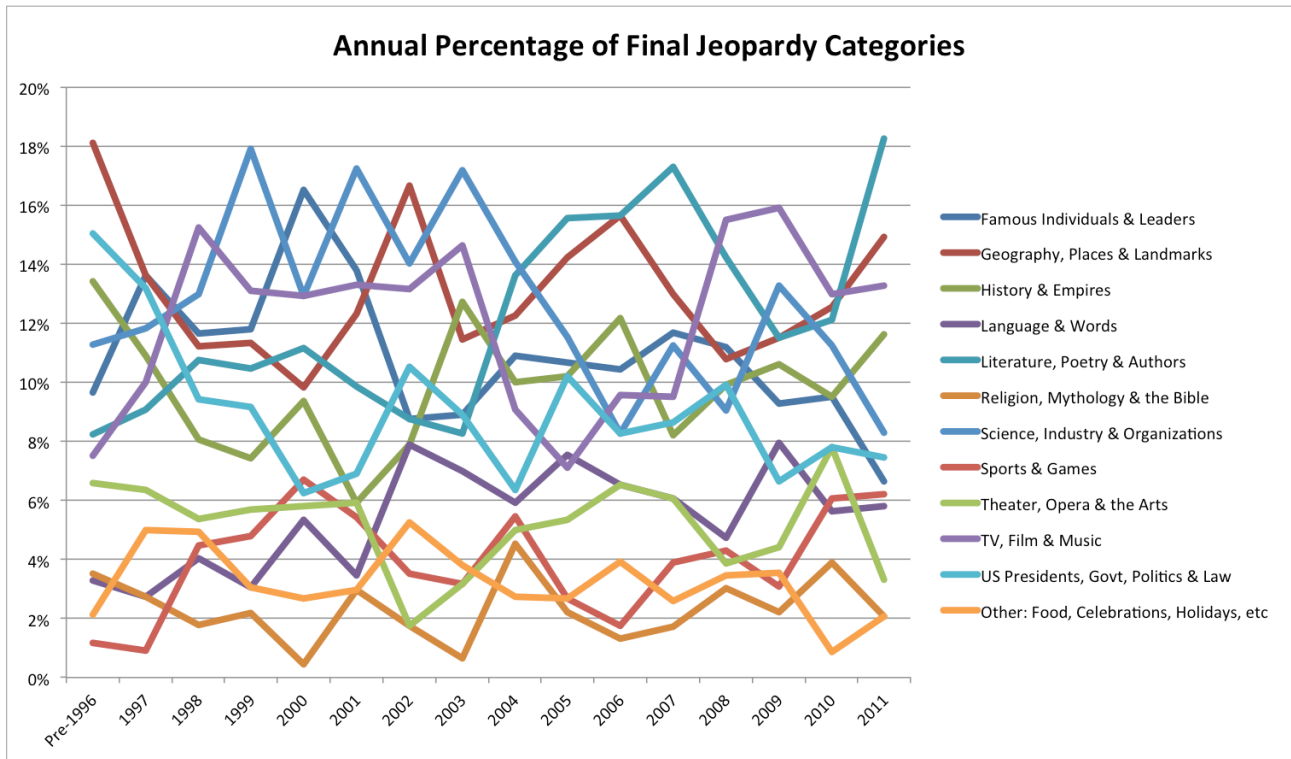
I obtained the database, filled with 216,931 lines from Jeopardy's history, on Reddit's web site (http://www.reddit.com/r/datasets/comments/1uyd0t/200000_jeopardy_questions_in_a_json_file/). With the help of archived episodes the available data includes the show number, date the show aired, round of the clue (Jeopardy, Double Jeopardy, or Final Jeopardy), category name, value of the clue, question/clue, and the correct answer.

Based on the interest of this time series model (and time), I first extracted only the lines associated with the Final Jeopardy round. I then proceeded to sort the remaining 3,631 lines into broader groupings than what the database listed in the category column. Although not from the exact same-named category, certain clues should generally be grouped together: 'WORLD CITIES' and 'THE 50 STATES', 'AMERICAN LITERATURE' and 'FICTIONAL CHARACTERS', 'PUNCTUATION' and 'ANAGRAMS', 'ROMAN EMPERORS' and 'HISTORIC NAMES', etc. The

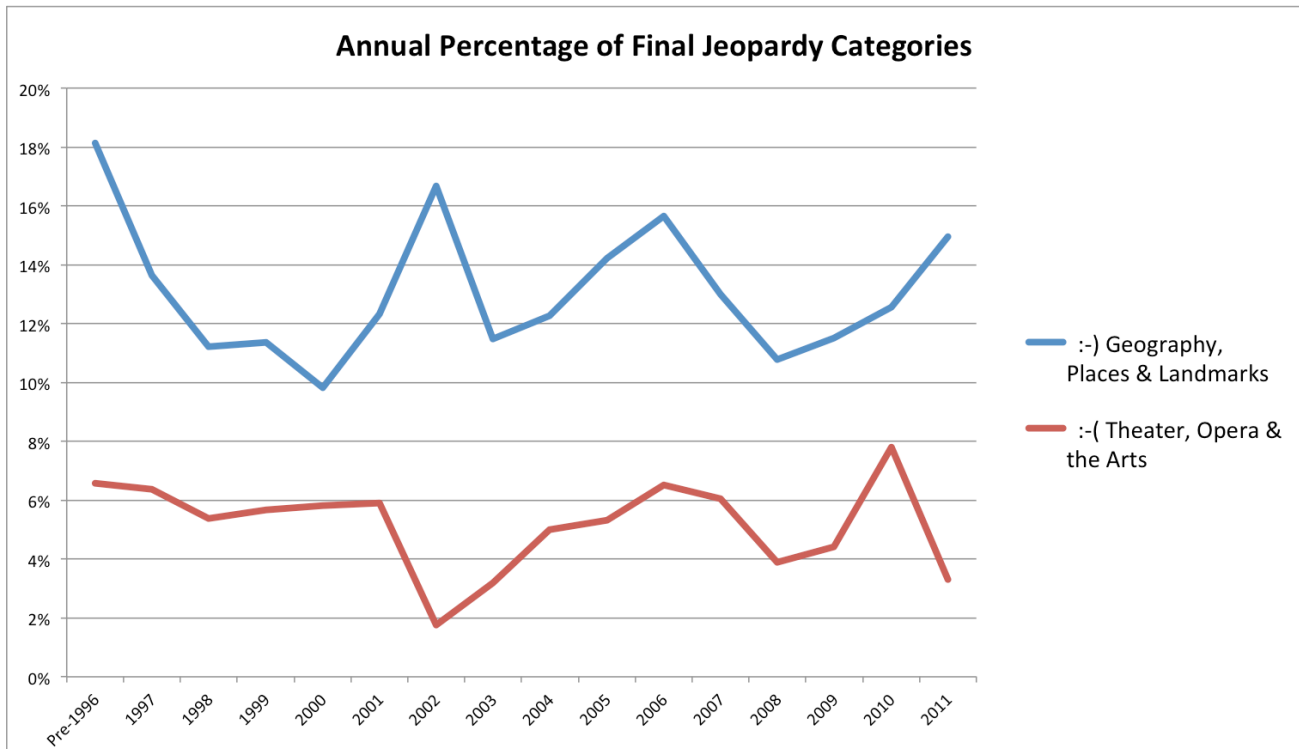
categories, question/clues, and answers from the original database were taken into consideration during this further grouping process. This process sifted the 1,948 unique Final Jeopardy categories into the 12 broader categories listed here:

- Famous Individuals & Leaders
- Geography, Places & Landmarks
- History & Empires
- Language & Words
- Literature, Poetry & Authors
- Religion, Mythology & the Bible
- Science, Industry & Organizations
- Sports & Games
- Theater, Opera & the Arts
- TV, Film & Music
- U.S. Presidents, Govt, Politics & Law
- Other: Food, Celebrations, Holidays, etc.

The original intention was to track how many Final Jeopardy questions fell into each of these broader categories on an annual basis dating back to 1984. However, a problem quickly became apparent. The number of episodes per year increased as the show gained in popularity, so the pure number of Final Jeopardy questions asked is naturally skewed towards the later years. Further, archived episodes are more difficult to locate from the earliest years, thus contributing to the data being heavily weighted towards the later years. The solution was to focus on the annual percentages of each category instead of the raw number of questions. Also, combining the years prior to 1997 eliminated the need to rely on the limited data (~20 data pts/year) in calculating percentages for years from 1984 to 1996.



Once again selfishly, I will focus on building a model for my favourite and least favourite categories, 'Geography, Places & Landmarks' and 'Theater, Opera & the Arts' respectively. Side note: these categories may or may not also be my most and least successful.



Test of Stationarity and Adjustments (If Necessary)

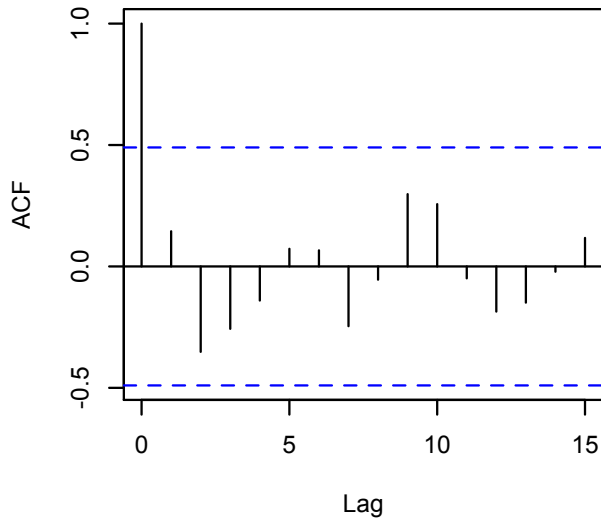
For each category, I must examine the data to see if we have a stationary time series, which is necessary to correctly model the time series. The test of stationarity will be based on autocorrelation function, which is the ratio of the sample covariance to the sample variance.

The autocorrelation formula used to estimate ρ_k can be expressed as:

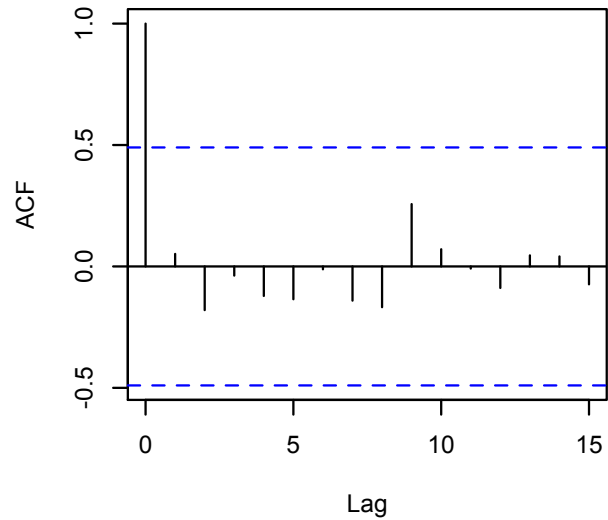
$$r_k = \frac{\sum_{t=k+1}^n (Y_t - \bar{Y})(Y_{t-k} - \bar{Y})}{\sum_{t=1}^n (Y_t - \bar{Y})^2}$$

Instead of manually calculating and plotting the "Correlogram" graph of r_k versus k , it is easily graphed using R. Below are two graphs depicting the autocorrelation function for the categories 'Geography, Places & Landmarks' (denoted X) and 'Theater, Opera & the Arts' (denoted Y).

Autocorrelation Function X



Autocorrelation Function Y

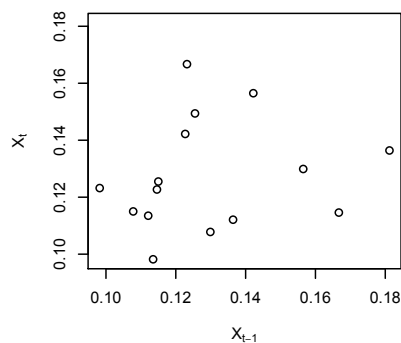


In order to suppose that after a certain amount of lag the result becomes white noise, the autocorrelation needs to be close to zero. All values are within the horizontal dashed lines placed at plus or minus two approximate standard errors of the sample autocorrelations. Since the autocorrelations oscillate around zero starting at lag 1 and continuing forever, the correlogram graphs very clearly indicate that each category can be considered a stationary process. No adjustments (differences, logarithms, diff of logs, transformations, etc.) are necessary to create a stationary process. I conclude that ARIMA(p,d,q) will have $d = 0$, and we can move ahead with estimating the p and q in our ARIMA(p,0,q) model.

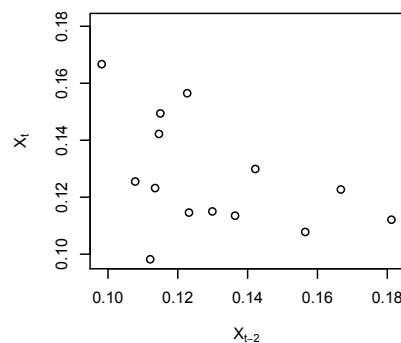
Model Estimation

Now that the stationarity of these time series have been determined, the question becomes which model is a best fit. The graphs of X_t vs. X_{t-1} , X_t vs. X_{t-2} , and X_t vs. X_{t-3} (Similarly for the Ys) give a general idea if AR(1), AR(2), or AR(3) models would be a good fit for the data. A correlation in these graphs would suggest an autoregressive model is appropriate, where as a general scatter with little correlation might insinuate a moving average model.

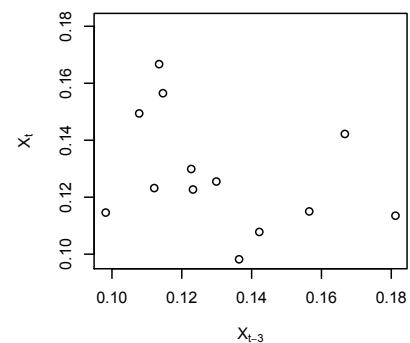
Exploring Lag 1 Correlation of X

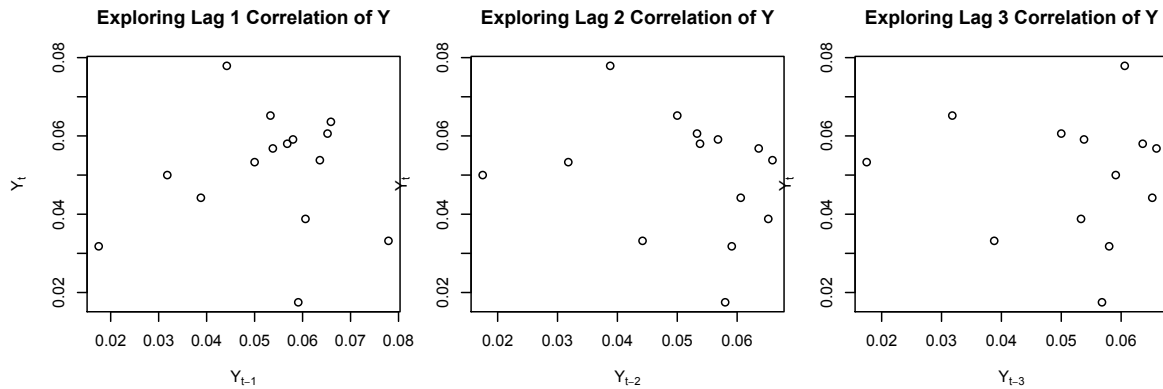


Exploring Lag 2 Correlation of X



Exploring Lag 3 Correlation of X





For 'Geography, Places & Landmarks' (X) one can see a slight positive correlation in the Lag 1 graph and slight negative correlation in the Lag 2 graph, however neither are strong. The graphs for 'Theater, Opera & the Arts' (Y) are more randomly scattered with very little, if any, trend. I will proceed by fitting and testing AR(1), AR(2), AR(3) and MA(1) models for these data using R.

Model/Parameter Fitting

A p^{th} -order autoregressive process and q^{th} -order moving average model satisfy the equations:

Autoregressive [AR(p)]

Moving Average [MA(q)]

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \delta + e_t$$

$$Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \dots - \theta_q e_{t-q} + \delta$$

Note: δ is simply a constant corresponding to the intercept of the model.

Using R's built-in ARIMA(p,d,q) function I was able to find the values of the unknown parameters for the AR(1), AR(2), AR(3) and MA(1) models of X ('Geography, Places & Landmarks') and Y ('Theater, Opera & the Arts'). Please refer to the Appendix: R output.

X: Geography, Places & Landmarks								
	R^2	Adj R^2	ϕ_1	ϕ_2	ϕ_3	θ_1	δ	$\Sigma\phi$
AR(1)	0.0299	-0.0447	0.2179	-	-	-	0.1321	0.2179
AR(2)	0.2551	0.1196	0.3053	-0.5283	-	-	0.1301	-0.2230
AR(3)	0.2562	0.0083	0.2050	-0.4759	-0.1944	-	0.1297	-0.4653
MA(1)	-	-	-	-	-	0.5483	0.1330	-

Y: Theater, Opera & the Arts								
	R^2	Adj R^2	ϕ_1	ϕ_2	ϕ_3	θ_1	δ	$\Sigma\phi$
AR(1)	0.0038	-0.0728	0.0568	-	-	-	0.0518	0.0568
AR(2)	0.0578	-0.1135	0.1028	-0.2702	-	-	0.0513	-0.1674
AR(3)	0.0781	-0.2292	0.1244	-0.2898	0.0594	-	0.0514	-0.1060
MA(1)	-	-	-	-	-	0.1002	0.0518	-

Clearly none of the AR(p) models are stellar. In regards to the ‘Geography, Places & Landmarks’ category, one might conclude the AR(2) model is the best autoregressive option due to it having the highest coefficient of determination (R^2) in combination with the principle of parsimony, which aims to simplify the model with the least amount of terms. That said, a model that only explains 25.5% of variation in a data set means it is not a good model for the data. None of the autoregressive models have any clout for the ‘Theater, Opera & the Arts’ category. Looking even further into the associated p-values of each AR(p) models’ coefficients, it becomes clear to select the MA(1) models since the p-values are almost all considered insignificant (>0.05).

AR(p)	X : Geography, Places & Landmarks			Y : Theater, Opera & the Arts		
p	p-value ϕ_1	p-value ϕ_2	p-value ϕ_3	p-value ϕ_1	p-value ϕ_2	p-value ϕ_3
1	0.4664	-	-	0.8282	-	-
2	0.2662	0.0284	-	0.6947	0.3457	-
3	0.4995	0.0542	0.4771	0.6624	0.3386	0.8458

Note: calculation of p-value ϕ_1 for X: $z = 0.2179/0.2992_{(s.e.)} = 0.7283 \rightarrow$ p-value = 0.4664

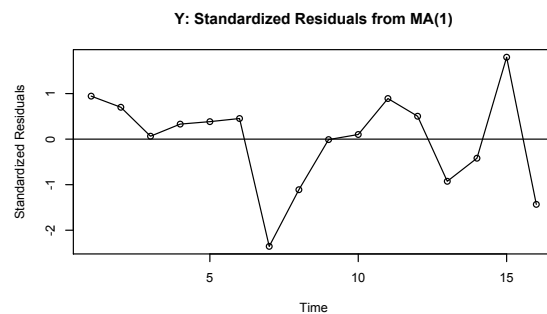
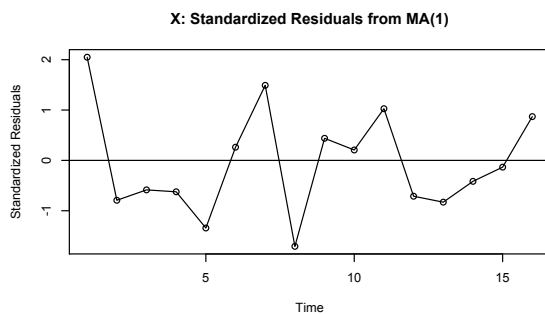
Residuals analysis

Since the Moving Average Model is selected for this data, the Durbin Watson and Box-Pierce Q test for the white noise of the residuals are unnecessary.

$$DWS = \frac{\text{Summation of Residuals Difference Squared}}{\text{Summation of Residuals Squared}} = \frac{\sum(\varepsilon_t - \varepsilon_{t-1})^2}{\sum \varepsilon_t^2}$$

$$\text{Box - Pierce } Q = T * \sum_{k=1}^K \hat{r}_k^2$$

Looking at the graphs of the standardized residuals of the MA(1) models below, there is no apparent trends and hence the plots generally support the models. The residuals have the near properties of a stochastic white noise process.



Forecasting and Conclusion

Autoregressive models are best for time series hinting towards long-term dependencies. Since the annual percentage of Final Jeopardy questions asked within a certain category is completely arbitrary (subject to the show’s question developers), it is reasonable that no autoregressive

model fits this data well. These conclusions make me wish that reliable historical data was readily available for the nightly Jeopardy payouts (contestant-winnings/show-losses) as discussed in the introduction. This data could lead to a more interesting ARIMA model.

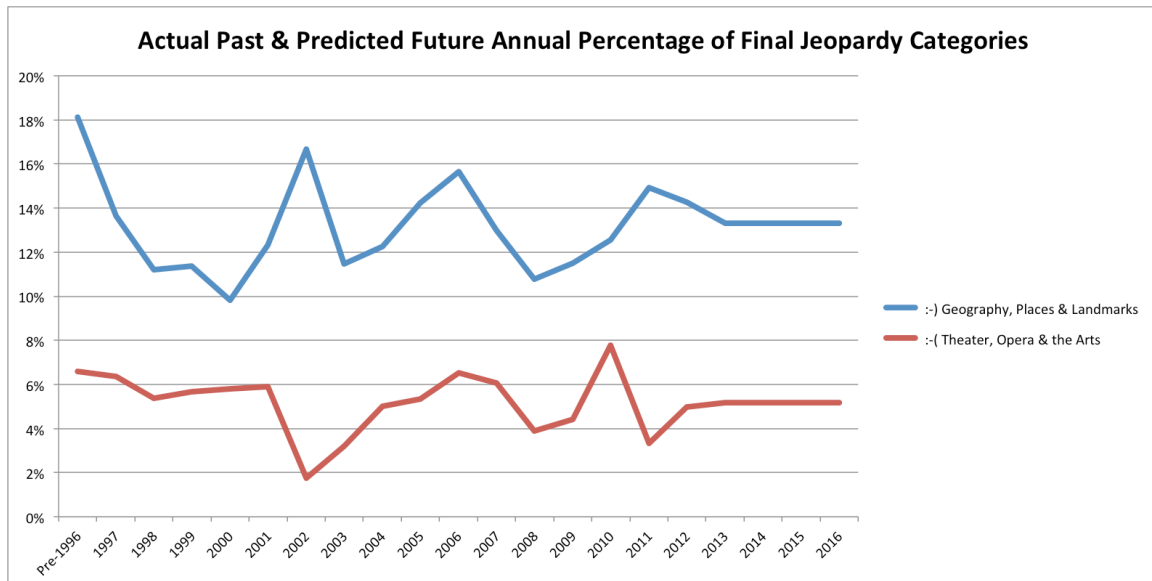
Although the percentages seem to be arbitrary from year to year, I will still predict them for the next 5 years from the final data in 2011. To compute I used the following MA(1) formulas:

$$X_t = e_t - 0.5483e_{t-1} + 0.1330 \quad \text{and} \quad Y_t = e_t - 0.1002e_{t-1} + 0.0518$$

The predicted future percentages are italicized in the chart below and can also be found in the Appendix: R output.

Year	X: Geography, Places & Landmarks	Y: Theater, Opera & the Arts
Pre-1996	18.12%	6.59%
1997	13.63%	6.36%
...
2011	14.94%	3.32%
<i>2012</i>	<i>14.28%</i>	<i>4.97%</i>
<i>2013</i>	<i>13.30%</i>	<i>5.18%</i>
<i>2014</i>	<i>13.30%</i>	<i>5.18%</i>
<i>2015</i>	<i>13.30%</i>	<i>5.18%</i>
<i>2016</i>	<i>13.30%</i>	<i>5.18%</i>

Below is a repeat graph now containing these predictions. Unfortunately it does not look like Jeopardy will eliminate questions regarding the 'Theater, Opera & the Arts'. On the bright side, there will continue to be a higher percentage of questions from the 'Geography, Places & Landmarks' associated categories.



Appendix: R output

```
> library(TSA)
> getwd()
[1] "/Users/Owner"
> setwd("~/Desktop/VEE Time Series/Projects")
> data <- read.table("JEOPboth%.csv", header=TRUE, sep=",")
> data
      X      Y
1 0.1812 0.0659
2 0.1364 0.0636
3 0.1121 0.0538
4 0.1135 0.0568
5 0.0982 0.0580
6 0.1232 0.0591
7 0.1667 0.0175
8 0.1146 0.0318
9 0.1227 0.0500
10 0.1422 0.0533
11 0.1565 0.0652
12 0.1299 0.0606
13 0.1078 0.0388
14 0.1150 0.0442
15 0.1255 0.0779
16 0.1494 0.0332
> attach(data)
>
> dev.new(width=3, height=3, pointsize=8)
> plot(acf(X,lag.max= 15), xlab="Lag", ylab=NULL, main =
"Autocorrelation Function X")
>
> dev.new(width=3, height=3, pointsize=8)
> plot(acf(Y,lag.max= 15), xlab="Lag", ylab=NULL, main =
"Autocorrelation Function Y")
>
> dev.new(width=3,height=3,pointsize=8)
> plot(y=X, x=zlag(X,1), ylab=expression(X[t]), xlab=expression(X[t-
1]), type='p', main="Exploring Lag 1 Correlation of X")
>
> dev.new(width=3,height=3,pointsize=8)
> plot(y=X, x=zlag(X,2), ylab=expression(X[t]), xlab=expression(X[t-
2]), type='p', main="Exploring Lag 2 Correlation of X")
>
> dev.new(width=3,height=3,pointsize=8)
> plot(y=X, x=zlag(X,3), ylab=expression(X[t]), xlab=expression(X[t-
3]), type='p', main="Exploring Lag 3 Correlation of X")
>
> dev.new(width=3,height=3,pointsize=8)
> plot(y=Y, x=zlag(Y,1), ylab=expression(Y[t]), xlab=expression(Y[t-
1]), type='p', main="Exploring Lag 1 Correlation of Y")
> dev.new(width=3,height=3,pointsize=8)
> plot(y=Y, x=zlag(Y,2), ylab=expression(Y[t]), xlab=expression(Y[t-
2]), type='p', main="Exploring Lag 2 Correlation of Y")
> dev.new(width=3,height=3,pointsize=8)
> plot(y=Y, x=zlag(Y,3), ylab=expression(Y[t]), xlab=expression(Y[t-
3]), type='p', main="Exploring Lag 3 Correlation of Y")
```



```

> arima(X, order=c(1,0,0))

Call:
arima(x = X, order = c(1, 0, 0))

Coefficients:
      ar1  intercept
      0.2179  0.1321
s.e.  0.2992  0.0071

sigma^2 estimated as 0.0004767:  log likelihood = 38.46,  aic = -72.92
>
>
>
> arima(X, order=c(2,0,0))

Call:
arima(x = X, order = c(2, 0, 0))

Coefficients:
      ar1      ar2  intercept
      0.3053 -0.5283  0.1301
s.e.  0.2746  0.2411  0.0042

sigma^2 estimated as 0.0003621:  log likelihood = 40.34,  aic = -74.68
>
>
>
> arima(X, order=c(3,0,0))

Call:
arima(x = X, order = c(3, 0, 0))

Coefficients:
      ar1      ar2      ar3  intercept
      0.2050 -0.4759 -0.1944  0.1297
s.e.  0.3036  0.2472  0.2734  0.0035

sigma^2 estimated as 0.0003482:  log likelihood = 40.58,  aic = -73.16
>
>
>
> arima(X, order=c(0,0,1))

Call:
arima(x = X, order = c(0, 0, 1))

Coefficients:
      ma1  intercept
      0.5483  0.1330
s.e.  0.2483  0.0079

sigma^2 estimated as 0.0004258:  log likelihood = 39.21,  aic = -74.42
>
>
>
>

```

```

> arima(Y,order=c(1,0,0))

Call:
arima(x = Y, order = c(1, 0, 0))

Coefficients:
      ar1  intercept
      0.0568    0.0518
s.e.  0.2618    0.0039

sigma^2 estimated as 0.0002213:  log likelihood = 44.62,  aic = -85.25
>
>
>
> arima(Y,order=c(2,0,0))

Call:
arima(x = Y, order = c(2, 0, 0))

Coefficients:
      ar1      ar2  intercept
      0.1028  -0.2702    0.0513
s.e.  0.2620   0.2866    0.0032

sigma^2 estimated as 0.0002079:  log likelihood = 45.05,  aic = -84.09
>
>
>
> arima(Y,order=c(3,0,0))

Call:
arima(x = Y, order = c(3, 0, 0))

Coefficients:
      ar1      ar2      ar3  intercept
      0.1244  -0.2898   0.0594    0.0514
s.e.  0.2849   0.3029   0.3055    0.0034

sigma^2 estimated as 0.0002071:  log likelihood = 45.06,  aic = -82.13
>
>
>
> arima(Y, order=c(0,0,1))

Call:
arima(x = Y, order = c(0, 0, 1))

Coefficients:
      ma1  intercept
      0.1002    0.0518
s.e.  0.3591    0.0041

sigma^2 estimated as 0.0002207:  log likelihood = 44.64,  aic = -85.28
>
>
>
>
>
>

```

```
> modelXAR1 <- lm(X ~ zlag(X,1) +1)
> summary(modelXAR1)
```

```
Call:
lm(formula = X ~ zlag(X) + 1)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.027052 -0.013997 -0.001531  0.008523  0.040054
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.10894    0.02988   3.646  0.00296 **
zlag(X)      0.14371    0.22703   0.633  0.53772
```

```
---
```

```
Signif. codes:
```

```
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.01971 on 13 degrees of freedom
(1 observation deleted due to missingness)
Multiple R-squared:  0.0299, Adjusted R-squared:  -0.04472
F-statistic: 0.4007 on 1 and 13 DF, p-value: 0.5377
```

```
>
>
>
```

```
> modelXAR2 <- lm(X ~ zlag(X,1) + zlag(X,2) +1)
> summary(modelXAR2)
```

```
Call:
lm(formula = X ~ zlag(X, 1) + zlag(X, 2) + 1)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-0.033249 -0.007999 -0.001594  0.012569  0.027273
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.15152    0.04077   3.717  0.0034 **
zlag(X, 1)   0.23089    0.27734   0.833  0.4228
zlag(X, 2)  -0.41285    0.21918  -1.884  0.0863 .
```

```
---
```

```
Signif. codes:
```

```
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.01863 on 11 degrees of freedom
(2 observations deleted due to missingness)
Multiple R-squared:  0.2551, Adjusted R-squared:  0.1196
F-statistic: 1.883 on 2 and 11 DF, p-value: 0.198
```

```
>
>
>
```

```
> modelXAR3 <- lm(X ~ zlag(X,1) + zlag(X,2) + zlag(X,3) +1)
> summary(modelXAR3)
```

Call:

```
lm(formula = X ~ zlag(X, 1) + zlag(X, 2) + zlag(X, 3) + 1)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.033121	-0.008847	-0.000265	0.012861	0.024652

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.18496	0.06654	2.780	0.0214 *
zlag(X, 1)	0.12887	0.33880	0.380	0.7125
zlag(X, 2)	-0.40303	0.30874	-1.305	0.2241
zlag(X, 3)	-0.16924	0.27199	-0.622	0.5492

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02009 on 9 degrees of freedom

(3 observations deleted due to missingness)

Multiple R-squared: 0.2562, Adjusted R-squared: 0.008278

F-statistic: 1.033 on 3 and 9 DF, p-value: 0.4232

```
>
>
>
```

```
> modelYAR1 <- lm(Y ~ zlag(Y,1) +1)
```

```
> summary(modelYAR1)
```

Call:

```
lm(formula = Y ~ zlag(Y, 1) + 1)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.033799	-0.009205	0.002576	0.008393	0.027542

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.04757	0.01562	3.045	0.0094 **
zlag(Y, 1)	0.06317	0.28374	0.223	0.8273

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.016 on 13 degrees of freedom

(1 observation deleted due to missingness)

Multiple R-squared: 0.003799, Adjusted R-squared: -0.07283

F-statistic: 0.04957 on 1 and 13 DF, p-value: 0.8273

```
>
>
>
```

```
> modelYAR2 <- lm(Y ~ zlag(Y,1) + zlag(Y,2) + 1)
> summary(modelYAR2)
```

Call:

```
lm(formula = Y ~ zlag(Y, 1) + zlag(Y, 2) + 1)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.031153	-0.007961	0.002519	0.010164	0.024974

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.06056	0.02111	2.869	0.0153 *
zlag(Y, 1)	0.07108	0.30993	0.229	0.8228
zlag(Y, 2)	-0.27774	0.33834	-0.821	0.4291

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01648 on 11 degrees of freedom

(2 observations deleted due to missingness)

Multiple R-squared: 0.0578, Adjusted R-squared: -0.1135

F-statistic: 0.3374 on 2 and 11 DF, p-value: 0.7208

```
>
>
>
```

```
> modelYAR3 <- lm(Y ~ zlag(Y,1) + zlag(Y,2) + zlag(Y,3) + 1)
> summary(modelYAR3)
```

Call:

```
lm(formula = Y ~ zlag(Y, 1) + zlag(Y, 2) + zlag(Y, 3) + 1)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.0305192	-0.0108302	0.0004322	0.0110347	0.0236959

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.05855	0.03115	1.880	0.0929 .
zlag(Y, 1)	0.08418	0.37316	0.226	0.8266
zlag(Y, 2)	-0.36727	0.42829	-0.858	0.4134
zlag(Y, 3)	0.10198	0.42396	0.241	0.8153

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01797 on 9 degrees of freedom

(3 observations deleted due to missingness)

Multiple R-squared: 0.07808, Adjusted R-squared: -0.2292

F-statistic: 0.2541 on 3 and 9 DF, p-value: 0.8566

```
>
>
```

```
> modelXMA1 = arima(X,order=c(0,0,1))
> dev.new(width=5,height=3,pointsize=8)
> plot(rstandard(modelXMA1), ylab='Standardized Residuals', type='o',
main="X: Standardized Residuals from MA(1)"); abline(h=0)
```

```
> modelYMA1 = arima(Y,order=c(0,0,1))
> dev.new(width=5,height=3,pointsize=8)
> plot(rstandard(modelYMA1), ylab='Standardized Residuals', type='o',
main="Y: Standardized Residuals from MA(1)"); abline(h=0)
```

```
>
```

```
>
```

```
>
```

```
> predict(arima(X,order=c(0,0,1)),n.ahead=5)
```

```
$pred
```

```
Time Series:
```

```
Start = 17
```

```
End = 21
```

```
Frequency = 1
```

```
[1] 0.1428137 0.1329763 0.1329763 0.1329763
```

```
[5] 0.1329763
```

```
$se
```

```
Time Series:
```

```
Start = 17
```

```
End = 21
```

```
Frequency = 1
```

```
[1] 0.02063584 0.02353448 0.02353448 0.02353448
```

```
[5] 0.02353448
```

```
> predict(arima(Y,order=c(0,0,1)),n.ahead=5)
```

```
$pred
```

```
Time Series:
```

```
Start = 17
```

```
End = 21
```

```
Frequency = 1
```

```
[1] 0.04967142 0.05180394 0.05180394 0.05180394
```

```
[5] 0.05180394
```

```
$se
```

```
Time Series:
```

```
Start = 17
```

```
End = 21
```

```
Frequency = 1
```

```
[1] 0.01485645 0.01493085 0.01493085 0.01493085
```

```
[5] 0.01493085
```