220306122

Coursework 1 220306122

Ehsan I Ghani

2025-02-24

U.S. Gross National Product Time-Series analysis.

Preface:

The motivation of this analysis is to understand the future behaviour of the time series data 'Quarterly U.S. GNP' (Gross National Product) from 1947-2023. To this end, I will use Meta's Prophet forecasting system to generate a prediction of up to this year's (2025) values. To verify the robustness of the forecast, I will further analyse non-parametric forecasting and parametric methods to understand to what extent the dependence of GNP growth is linear over time.

Understanding the data:

In R from library 'astsa', I have imported the quarterly US GNP data spanning from 1947-2023.

```
## [1] "ts"
```

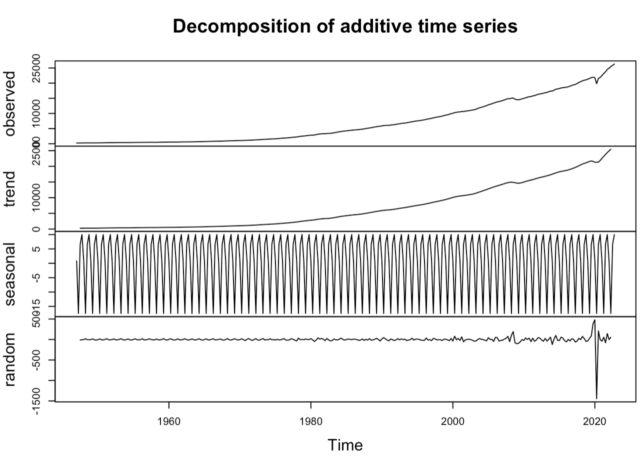
```
## [1] 244.142 247.063 250.716 260.981 267.133 274.046
```

```
## [1] 23718.26 24530.59 24929.18 25456.41 25885.43 26289.49
```

The dataset provides an insight into the economic performance of US post-war to modern times in units of billions of dollars. I can check the source and structure of data:

```
##  Time-Series [1:304] from 1947 to 2023: 244 247 251 261 267 ...
```

In understanding this data is a Time-Series from source and structure. I can continue in identifying historical trends, seasonal patterns and any measurable randomness.

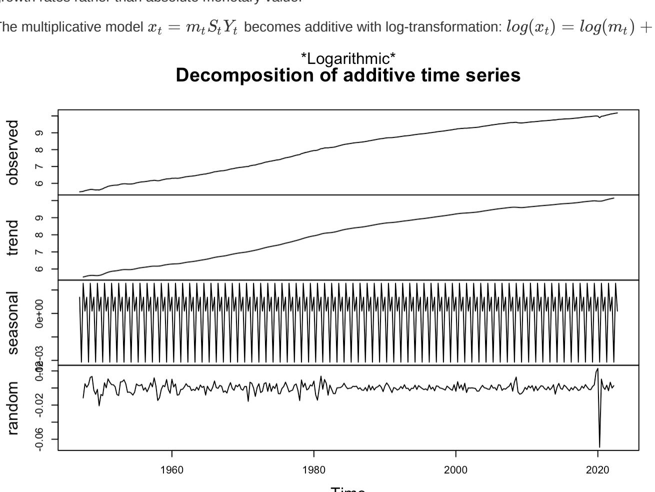


The decomposition above provides an insight in behaviour of components. Further analysis will refine these insights later on. But from graph, we can see an exponential trend, suggesting a multiplicative model: $x_t = m_t S_t Y_t$ and a clear seasonal pattern centered around zero. Whilst the random component is mostly centered around zero, the clear downward spike represents an anomaly in the effect on GNP of COVID (circa 2020). Suggesting the data correctly reflects the economic downturn during the COVID period. I will account for this deviation in the prediction.

Logarithmic Decomposition:

From the source: https://fred.stlouisfed.org/series/GNP I had gathered that the data was given in units of billions of dollars. The logarithm of this financial data will reflect the linearity of the apparent exponential growth, making data easier to interpret, as well as reflect the relevant increment of growth rates rather than absolute monetary value.

The multiplicative model $x_t = m_t S_t Y_t$ becomes additive with log-transformation: $log(x_t) = log(m_t) + log(S_t) + log(Y_t)$



The log-transformation reflects a more linear form in trend which will be useful moving forward in parametric analysis.

More on Logarithmic transformation:

Building on the findings of the logarithmic decomposition, I wanted definitive proof that a log transformation of the data would reflect an accurate representation of the linearity. I would continue in defining a log-data-frame and log-linear model:

```
## 
## Call:
## lm(formula = y ~ ds, data = USGNP.df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.46330 -0.14035 -0.03449  0.18382  0.29836 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.212e+02  9.795e-01  -123.7   <2e-16 ***
## ds           6.512e-02  4.934e-04   132.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1888 on 302 degrees of freedom
## Multiple R-squared:  0.983,  Adjusted R-squared:  0.9829 
## F-statistic: 1.742e+04 on 1 and 302 DF,  p-value: < 2.2e-16
```

From the summary of the log-transformed model, we can see an adjusted R^2 value of 98.29%. We can compare this to the original model's value by transforming the response back to reflect the exponential growth:

```
## 
## Call:
## lm(formula = exp(y) ~ ds, data = USGNP.df)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -2860.5 -2438.9  -532.6  1882.5  7923.6 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -5.922e+05  1.314e+04  -45.08   <2e-16 ***
## ds           3.018e+02  6.618e+00   45.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2532 on 302 degrees of freedom
## Multiple R-squared:  0.8732, Adjusted R-squared:  0.8728 
## F-statistic:  2080 on 1 and 302 DF,  p-value: < 2.2e-16
```

The adjusted R^2 value of the original data is 87.28%. R^2, known as the coefficient of determination, is the percentage of total variation in the response explained by the model. A higher value suggests a large proportion of variance is accounted for, indicating a better fit of the model and to what extent the relationship between time and GNP is linear. Therefore, the approximate 11% increase in the log-transformed model reflects a better fit of the the data, suggesting greater linearity between time and GNP. The log transformation provides a stabilisation of variance and linearises exponential growth, making the data easier to interpret and improves the predictive power.

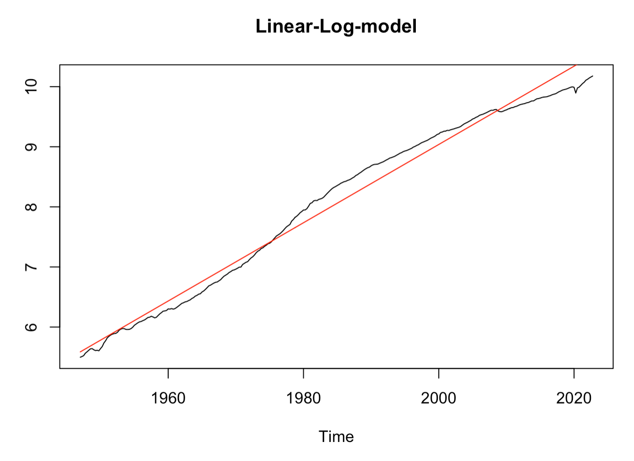
Parametric analysis:

In continuation of understanding the growth in the series, I will fit parametric models to analyse the optimal number of parameters which best explains the data. Ultimately, to summarise the trend.

I set the seasonal and residual component to zero solely for analysing the trend.

$log(x_t) = log(m_t)$ where m_t has functional form: $m_t = \beta_0 exp(\beta_1 t)$

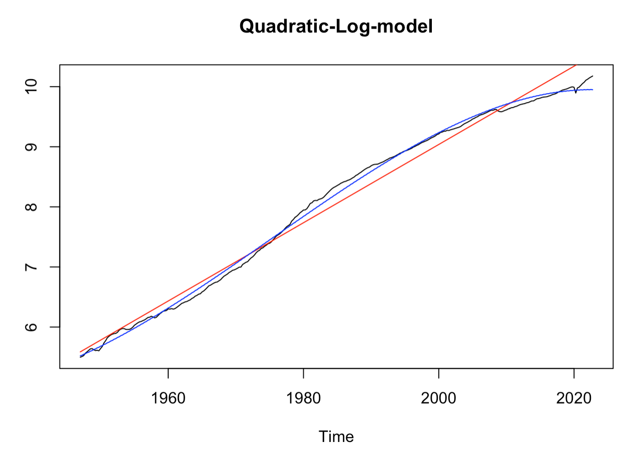
```
## 
## Call:
## lm(formula = y ~ ds, data = USGNP.df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.46330 -0.14035 -0.03449  0.18382  0.29836 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.212e+02  9.795e-01  -123.7   <2e-16 ***
## ds           6.512e-02  4.934e-04   132.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1888 on 302 degrees of freedom
## Multiple R-squared:  0.983,  Adjusted R-squared:  0.9829 
## F-statistic: 1.742e+04 on 1 and 302 DF,  p-value: < 2.2e-16
```



$log(x_t) = log(\beta_0) + \beta_1 t$

Fitting a log-linear model provides an understanding of growth of GNP over time. The fitted values/estimates are represented through the red line. Whilst it captures the general trend, we can further capture the fluctuations of the data by increasing the number of parameters to fit a quadratic regression:

```
## 
## Call:
## lm(formula = y ~ ds + I(ds^2), data = USGNP.df)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## [illegible]
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) [illegible]
## ds          [illegible]
## I(ds^2)     [illegible]
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: [illegible] on 301 degrees of freedom
## Multiple R-squared:  [illegible],  Adjusted R-squared:  [illegible]
## F-statistic: [illegible] on 2 and 301 DF,  p-value: < 2.2e-16
```



$log(x_t) = log(\beta_0) + \beta_1 t + \beta_2 t^2$

The quadratic-log model is more effective in capturing the exponential growth in the series. This also reflected by an increased adjusted R^2 value, suggesting a better fit and improved explanatory power., but I do not wish to over-fit the data, which could lead to a sub-optimal performance with new data. I will restrict from delving further in.

Having provided a basic understanding of the data in question, I will now continue in providing a forecast of values for the years 2023-2025 via Meta's Prophet forecasting system.

Forecast via Meta's Prophet package:

Importing the 'prophet' package from the library will provide the necessary functions to forecast future values of the log-transformed data. I also implemented the zoo package for confirming a quarterly time index.

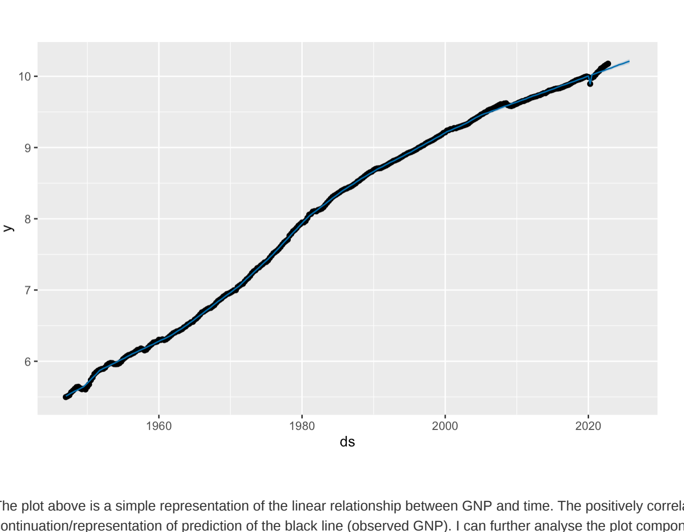
After loading the necessary packages I created a data-frame with the log-transformation of GNP data. Fitted the prophet function to model the data-frame. Then the last step of initialisation was to create the future data-frame for 12 quarterly periods, Q1 of 2023 to Q4 of 2025, for which values I will forecast. I've also accounted for the impact of COVID so that the prophet function will take quarter 2 of 2020 as an anomaly or an otherwise positively correlated trend. Accounting for such events will also improve the accuracy of the forecast.

```
##              ds
## [illegible]
```

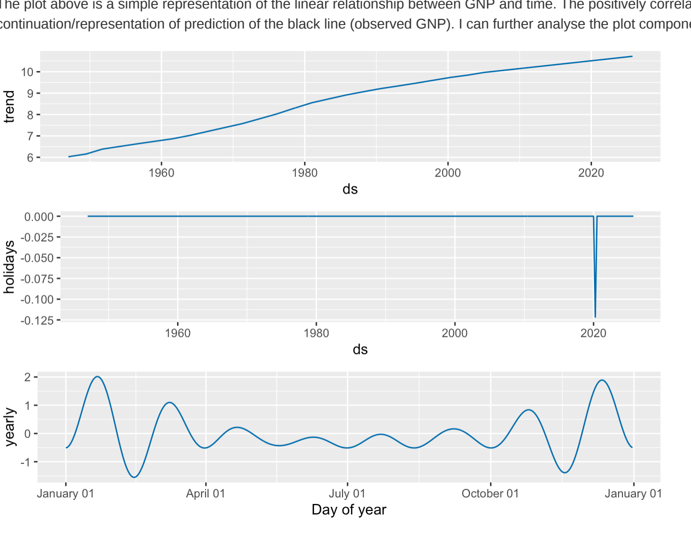
Using the tail() function in R I can confirm the last 6 quarters of future dates:

```
##              ds yhat_lower yhat_upper
## [illegible]
```

The predict() function uses the prophet model and future dates to forecast the values for the quarters in question. I also implemented a confidence interval for the predictions in the last 6 quarters, the numerical representation was purely for informative purposes of what to expect from the upcoming plots.

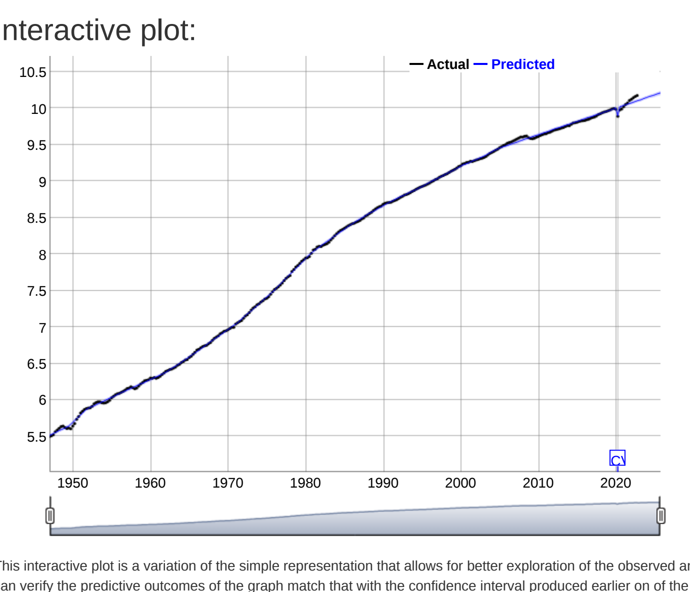


The plot above is a simple representation of the linear relationship between GNP and time. The positively correlated blue line is a continuation/representation of prediction of the black line (observed GNP). I can further analyse the plot components:



Displaying the predictive linear trend along with the yearly seasonal component and the random component centered around zero. While accounting for the spike in the forecast initialization is important, it does not imply that the randomness is fully explained. Instead, it highlights a significant deviation from the usual pattern. Accounting for this improves the accuracy of the trend.

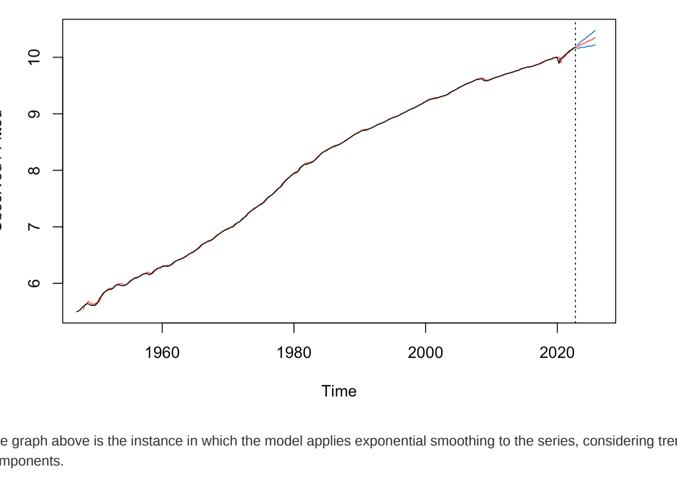
Interactive plot:



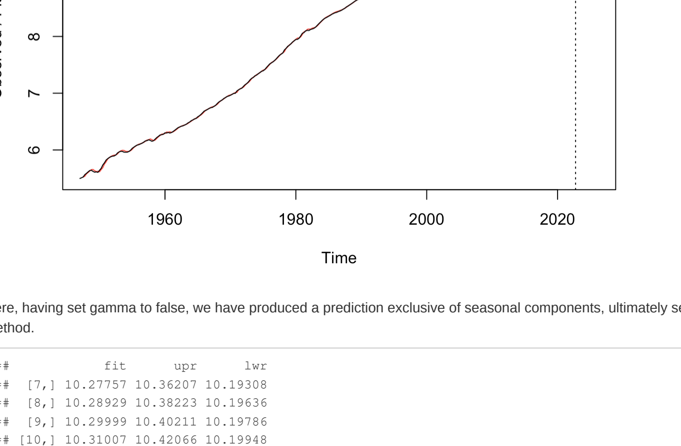
This interactive plot is a variation of the simple representation that allows for better exploration of the observed and fitted trend. For example, we can verify the predictive outcomes of the graph match that with the confidence interval produced earlier on of the last 6 quarters. Of Q4 2025, the prediction from the graph is 10.21 which coincides with the prediction interval of 10.17-10.24. Confirming this is consistent in numerical and graphical representation.

Non-parametric forecasting via Holt-Winters model:

The seasonal component of the series refers to the regular, predictable patterns that repeat over specific periods. It's important that the seasonal component is centered around zero, as if it wasn't, it would introduce bias and artificial trends. Here, through Holt-Winters filtering, I can show that the seasonal component does not significantly alter the predictions.



The graph above is the instance in which the model applies exponential smoothing to the series, considering trend, level and seasonal components.



Here, having set gamma to false, we have produced a prediction exclusive of seasonal components, ultimately seeming indifferent to the full method.

```
##              fit      upr      lwr
##   [7,] 10.27757 10.36207 10.19308
##   [8,] 10.28929 10.38223 10.19636
##   [9,] 10.29999 10.40211 10.19786
##  [10,] 10.31007 10.42066 10.19948
##  [11,] 10.33375 10.45289 10.21462
##  [12,] 10.34548 10.47324 10.21771
```

```
##              fit      upr      lwr
##   [7,] 10.27239 10.35200 10.19278
##   [8,] 10.28602 10.37347 10.19858
##   [9,] 10.29966 10.39490 10.20442
##  [10,] 10.31330 10.41633 10.21026
##  [11,] 10.32694 10.43778 10.21609
##  [12,] 10.34057 10.45926 10.22189
```

The notion is also satisfied through numerical representation, without seasonal components the predictions are not significantly altered. In comparison to the prophet forecast, there is an explanation to the slight differences in confidence intervals, it being the methods differ in flexibility and approach. Prophet has flexibility in accounting for deviations/impacts as evidently shown, but Holt-Winters offers a straightforward approach in accessing predictions. Concluding that the choice of either method is dependent on the complexity of the model.