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 further understand the data, I will compare parametric methods to investigate to what extent the rate of GNP growth is linear over time. 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 compare the prophet prediction with a non-parametric forecasting method.

Understanding the data:

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

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

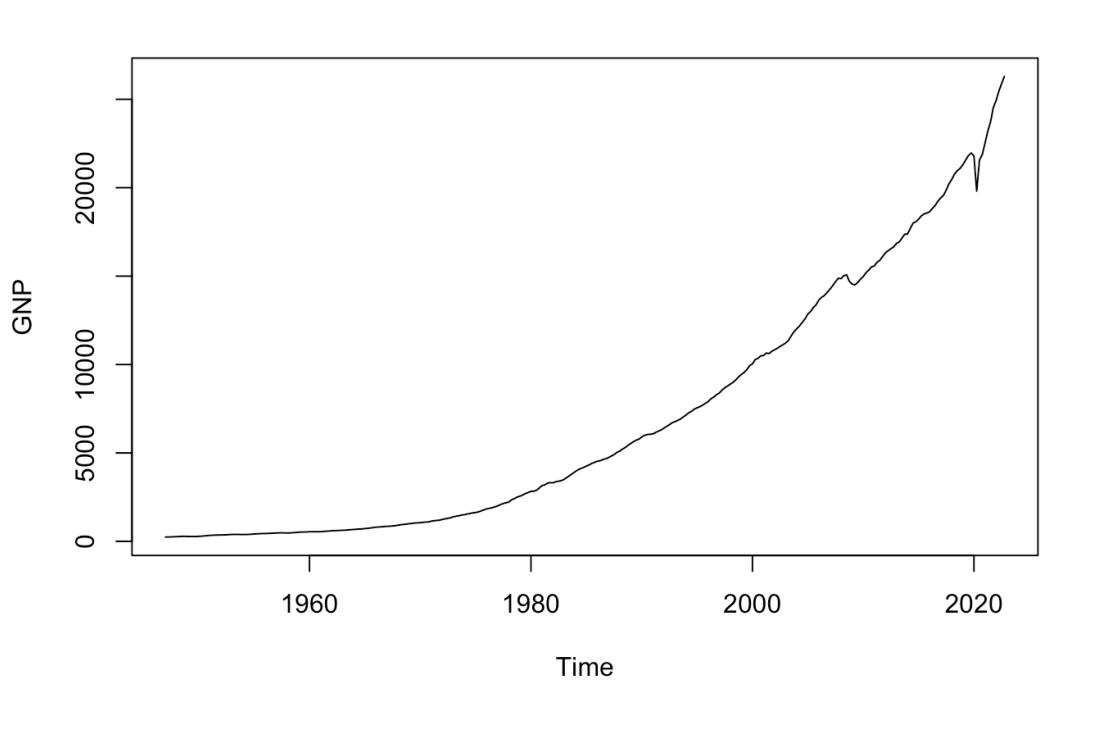
```
##  Time-Series [1:304] from 1947 to 2023: 244 247 251 261 267 ...
```

```
## [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 U.S. post-war to modern times in units of billions of dollars. Insight into source and structure I had confirmed quarterly Time-series data.

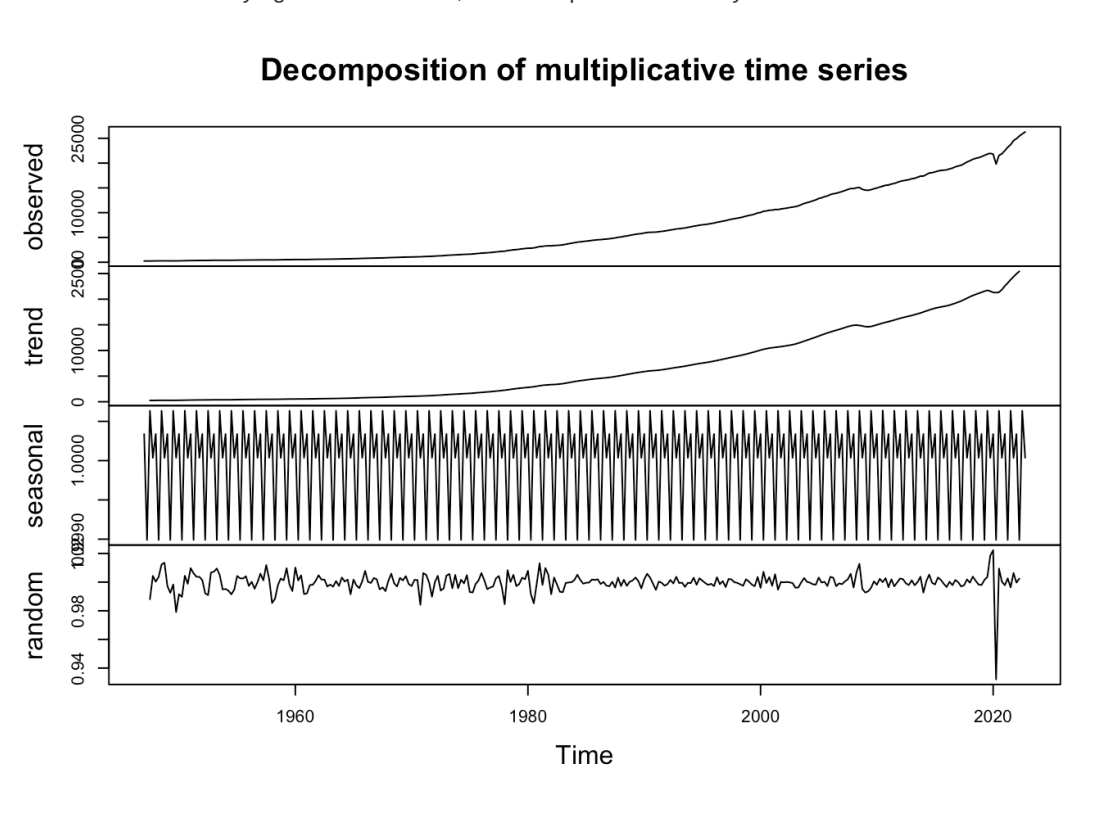
Plot of GNP Time-series:



From the apparent exponential growth in initial plot, I thought it was worth proceeding with a multiplicative model of Time series: $x_t = m_t S_t Y_t$.

Decomposition of multiplicative Time-series model:

I can continue in identifying historical trends, seasonal patterns and any measurable randomness.

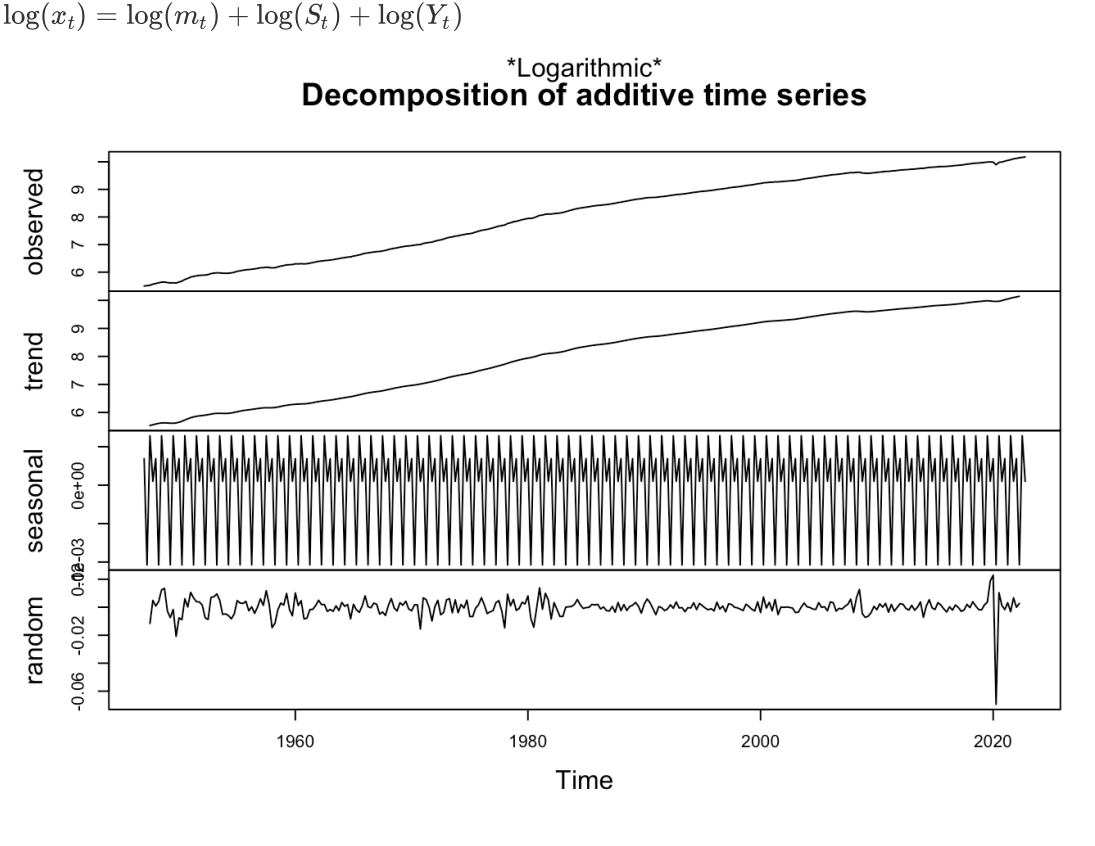


The decomposition above provides an insight in behaviour of components. Further analysis will refine these insights later on. But from the graph shown, the multiplicative model: $x_t = m_t S_t Y_t$ is typically associated with the exponential trend $m_t = \beta_0 exp(\beta_1 t)$ shown (here I'm implying a functional form on m_t as an example). The seasonal component in the multiplicative model sharply oscillates around one, acting as a small scaling factor for trend. Whilst the random/residual component is mostly centered around one, the clear downward spike represents an anomaly in the effect of COVID on GNP (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, with seasonal and residual components now centered around zero. The continuation of using a log-additive model 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 highlight the linearity of GNP growth over time. 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%. I 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 rate of GNP growth is linear. Therefore, the approximate 11% increase in the log-transformed model reflects a better fit of the the data. Concluding the log transformation provides a stabilisation of variance and linearises exponential growth, making the data easier to interpret and improves the predictiveness.

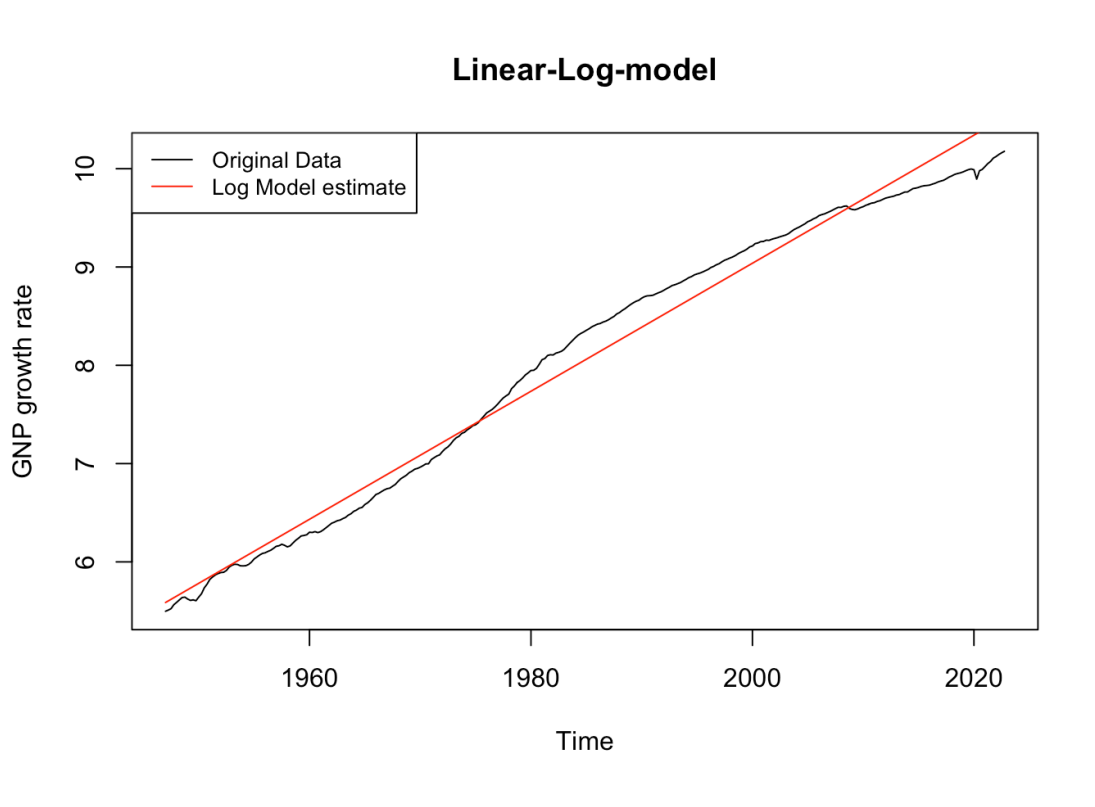
Parametric analysis:

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.

Considering a log-additive model with no seasonality: $\log(x_t) = \log(m_t) + \log(Y_t)$, where in context of regression analysis, the expected value of residual error is 0 ~ $[E[Y_t] = 0]$ and all Y_t are uncorrelated. Note that while the expected value equates to zero it doesn't necessarily rule out residual error, the model is still stochastic i.e. randomness inclusive.

In line with regression analysis, I assume $\log(x_t) = \log(m_t)$ where m_t has functional form: $m_t = \beta_0 exp(\beta_1 t) \rightarrow \log(m_t) = \log(\beta_0) + \beta_1 t$. Producing estimates of parameters $\hat{\beta}_0$ and $\hat{\beta}_1$ will be pivotal in fitting estimate of trend $\hat{m}_t$.

```
##
## 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
```



$\log(\hat{m}_t) = \log(\hat{\beta}_0) + \hat{\beta}_1 t$

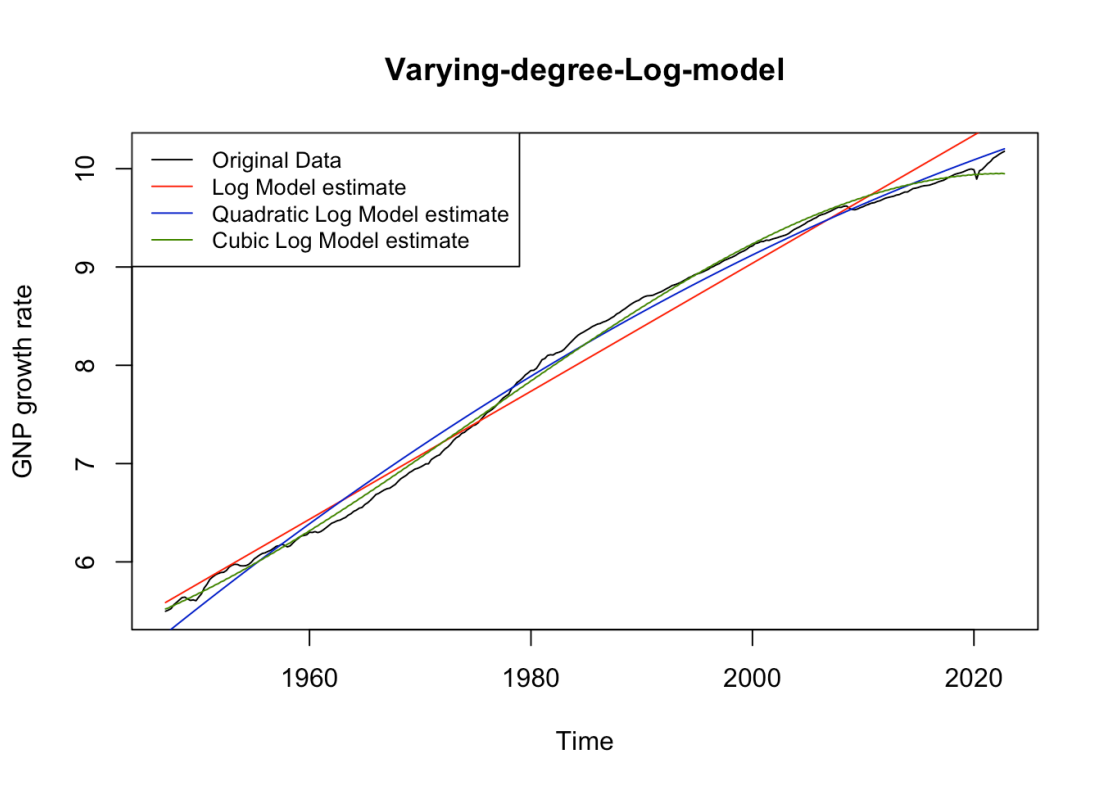
Fitting a log-linear model provides an understanding of growth of GNP over time. The fitted values/estimates are represented through the red line. Whilst it captures the general trend, I can further capture the fluctuations of the data by increasing the number of parameters via polynomial regression:

$\log(\hat{m}_t) = \log(\hat{\beta}_0) + \hat{\beta}_1 t + \hat{\beta}_2 t^2$ ~ ***Quadratic-log-model estimate***

```
##
## Call:
## lm(formula = y ~ +ds + I(ds^2), data = USGNP.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22928 -0.08675  0.04068  0.09457  0.23354
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.435e+03  6.410e+01  -22.39   <2e-16 ***
## ds           1.389e+00  6.460e-02   21.51   <2e-16 ***
## I(ds^2)     -3.336e-04  1.627e-05  -20.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1221 on 301 degrees of freedom
## Multiple R-squared:  0.9929, Adjusted R-squared:  0.9928
## F-statistic: 2.101e+04 on 2 and 301 DF,  p-value: < 2.2e-16
```

$\log(\hat{m}_t) = \log(\hat{\beta}_0) + \hat{\beta}_1 t + \hat{\beta}_2 t^2 + \hat{\beta}_3 t^3$ ~ ***Cubic-log-model estimate***

```
##
## Call:
## lm(formula = y ~ +ds + I(ds^2) + I(ds^3), data = USGNP.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.11734 -0.06032 -0.01876  0.05347  0.22792
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.967e+04  4.004e+03   22.39   <2e-16 ***
## ds          -1.363e+02  6.052e+00  -22.52   <2e-16 ***
## I(ds^2)      6.905e-02  3.049e-03   22.64   <2e-16 ***
## I(ds^3)     -1.165e-05  5.121e-07  -22.75   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0741 on 300 degrees of freedom
## Multiple R-squared:  0.9974, Adjusted R-squared:  0.9974
## F-statistic: 3.822e+04 on 3 and 300 DF,  p-value: < 2.2e-16
```



Both quadratic and cubic log models increasingly capture the fluctuations of growth in the series graphically. However, from the summaries of models, there is a 0.01% decrease in adjusted R^2 from linear to quadratic, suggesting the quadratic term lacks the explanatory power to justify the increase in model complexity. This in fact leads me to believe that the increase seen in the summary of the cubic variation is not an increase in explanatory power but an instance of having over-fit the model, which could lead to a sub-optimal performance with new data/predictions. Hence I will restrict from further analysis of increasing the number of parameters.

Residual error in Time series:

Consider again the log-additive model with no seasonality: $\log(x_t) = \log(m_t) + \log(Y_t)$, then setting log residual error to $\log(x_t) - \log(m_t)$.

If $\log(x_t)$ exceeds trend $\log(m_t)$ for particular t, this could increase the likelihood of the next iteration, say t+1, following with an increase. Suggesting that unlike in regression analysis, residual errors of Time-series has the potential to affect predictions through its correlated behaviour.

Initially, my aim was to keep the residual error as minimal as possible; to avoid randomness affecting the predictive values. In using the log-transformation, I had changed residual error from being a scaled factor of the multiplicative model, to being centered around zero in the log-additive model. This hadn't necessarily reduced the residual error but rather adjusted according to the model. Consequently, my aim had changed from trying to minimise residual error, to explaining and adjusting the component relative to producing predictions.

As a side note, to isolate the residual component I could've used a high-pass filter such as differencing to produce Y_t without having to of calculated $\hat{m}_t$, or the Brockwell and Davies algorithm etc. But the decompositions was sufficient in isolating residual component for both log-additive and multiplicative models.

Having provided an in-depth 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 had provided 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 again defined the data-frame with the log-transformation of GNP data. Then, fitted the prophet function to model the data-frame. 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.

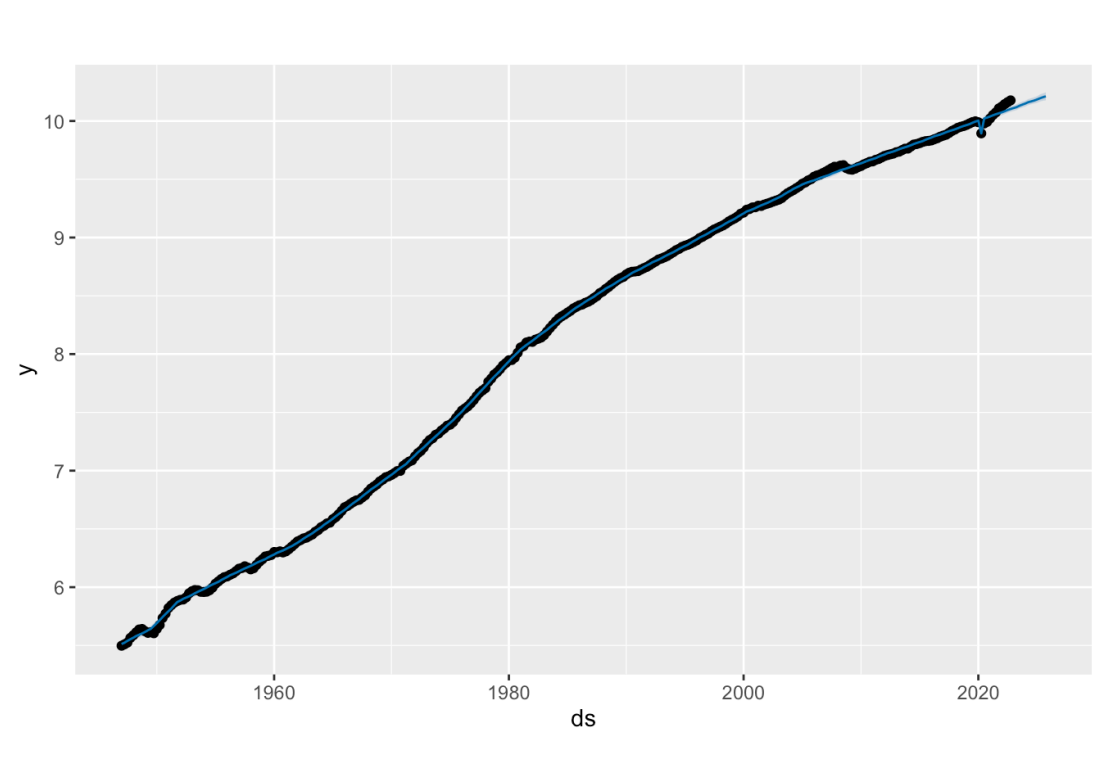
I can confirm the last 6 quarters of future dates:

```
##             ds
## 311 2024-07-01
## 312 2024-10-01
## 313 2025-01-01
## 314 2025-04-01
## 315 2025-07-01
## 316 2025-10-01
```

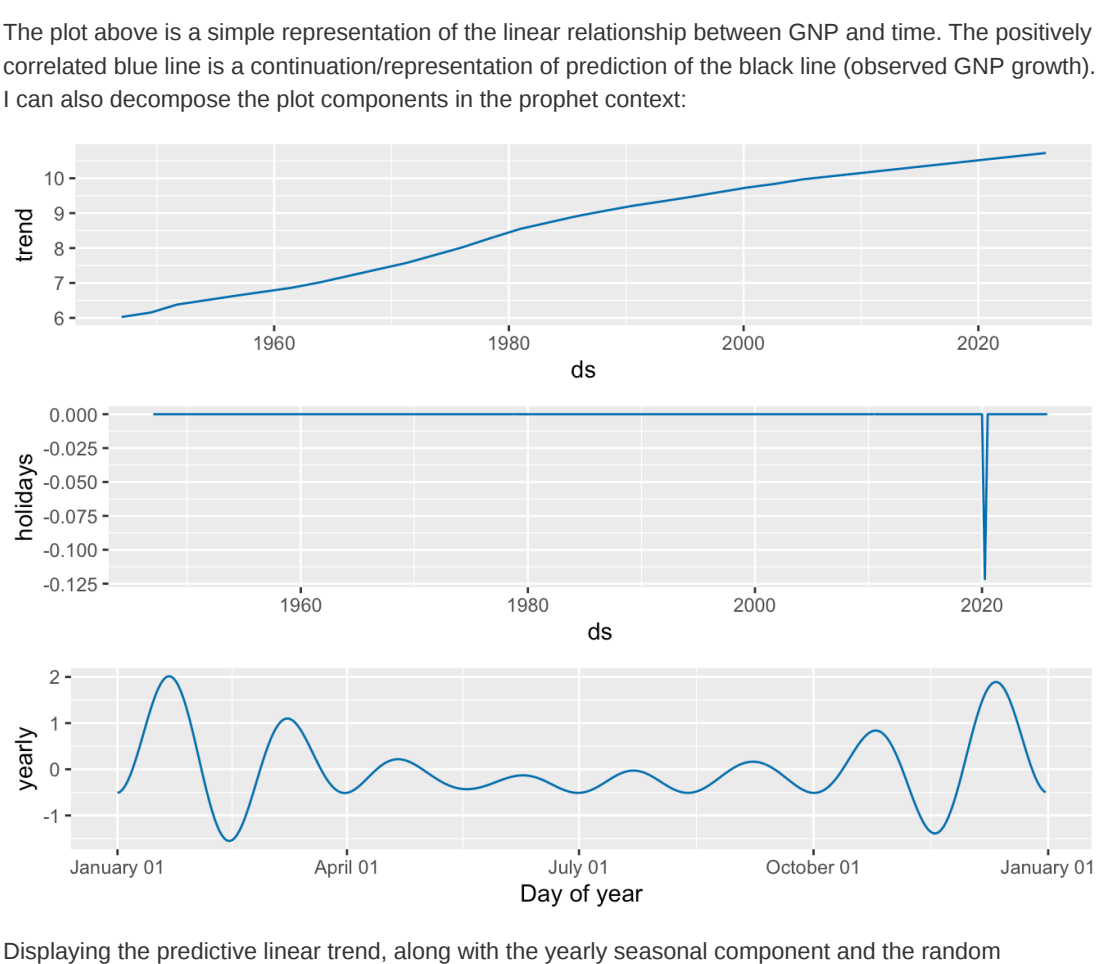
I also implemented intervals for the predictions in the last 6 quarters, the numerical representation was purely for informative purposes of what to expect from the upcoming plots:

```
##             ds yhat_lower yhat_upper
## 311 2024-07-01   10.14422   10.19458
## 312 2024-10-01   10.15057   10.20165
## 313 2025-01-01   10.15612   10.21140
## 314 2025-04-01   10.16754   10.22661
## 315 2025-07-01   10.17373   10.23594
## 316 2025-10-01   10.17794   10.24450
```

Prophet forecast plot:

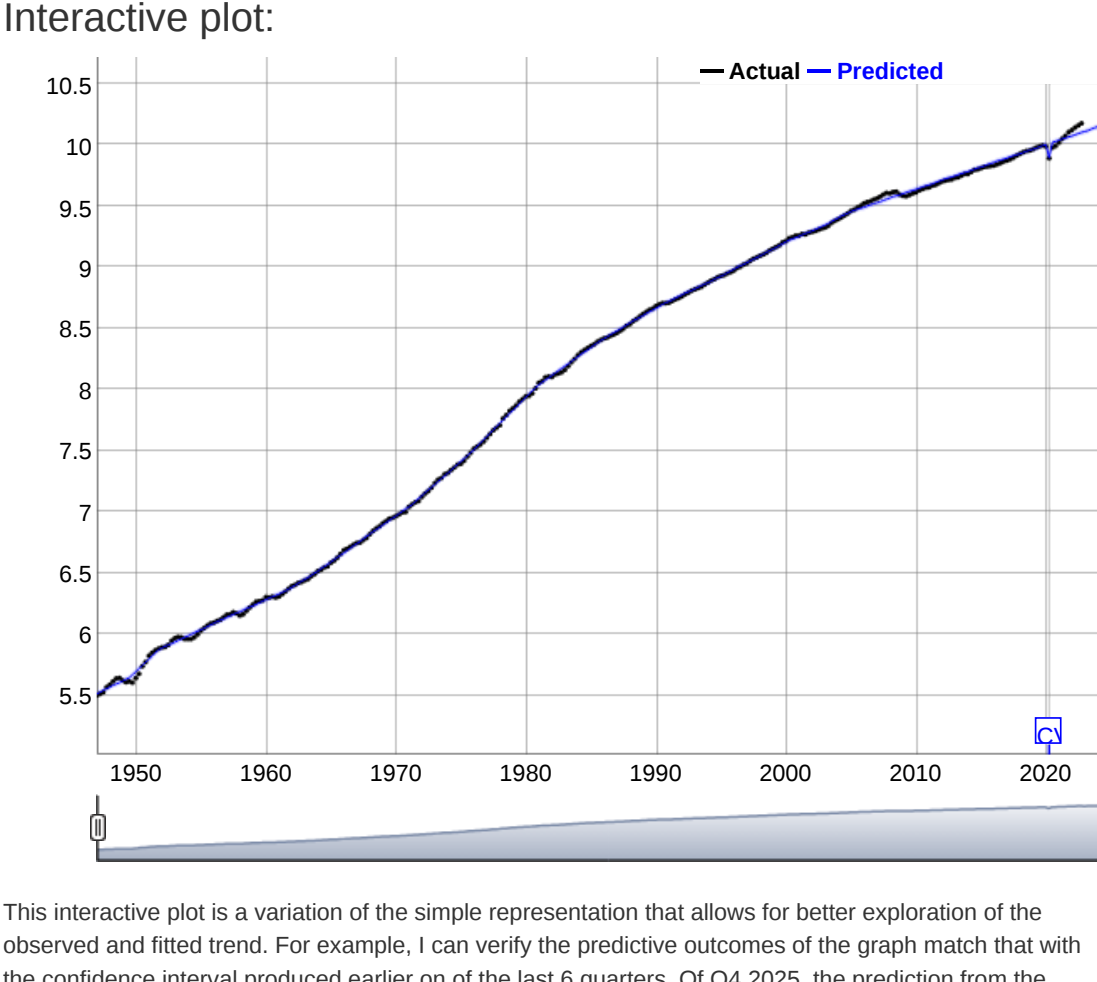


The plot above is a simple representation of the linear relationship between GNP and time. The positively correlated blue line is a continuation/representation of prediction of the black line (observed GNP growth). I can also decompose the plot components in the prophet context:



Displaying the predictive linear trend, along with the yearly seasonal component and the random component centered around zero. Accounting for the spike in the forecast initialisation was important, in that I've accounted for a significant deviation from the usual pattern, in an attempt to increase accuracy of the forecast.

Interactive plot:



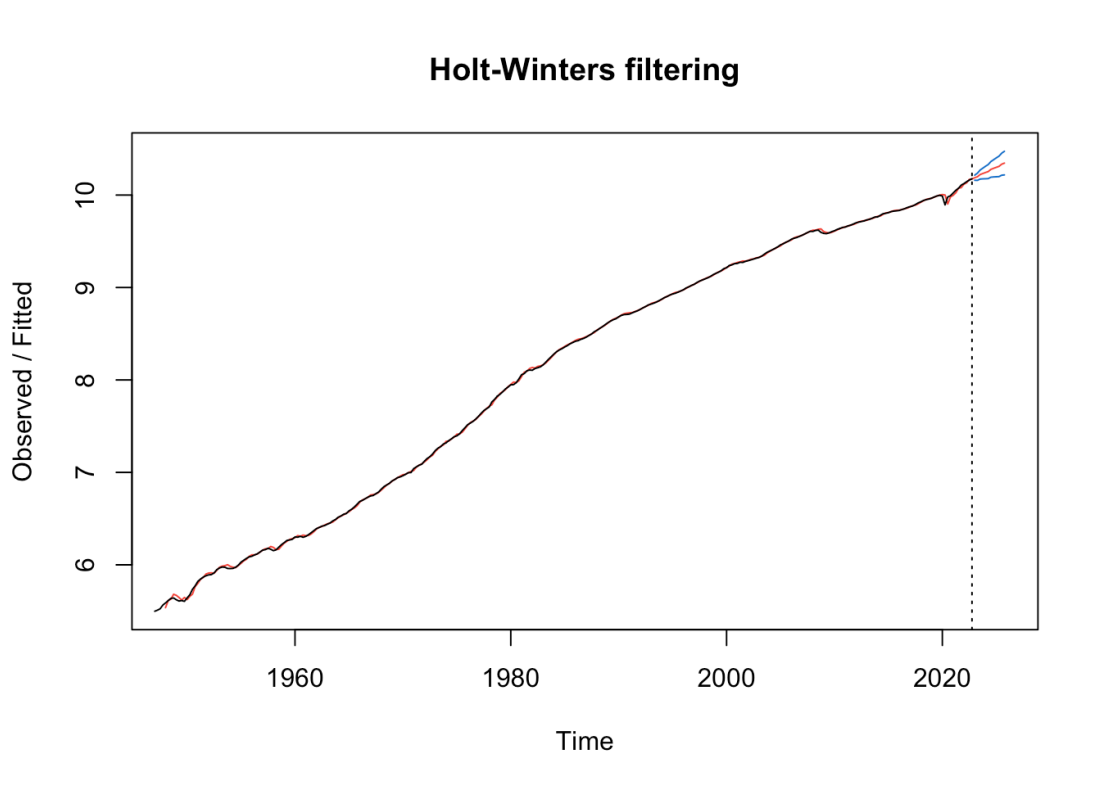
This interactive plot is a variation of the simple representation that allows for better exploration of the observed and fitted trend. For example, I can verify the predictive outcomes of the graph match that with the confidence interval produced earlier on of the last 6 quarters. Of Q4 2025, the prediction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

Non-parametric forecasting via Holt-Winters model:

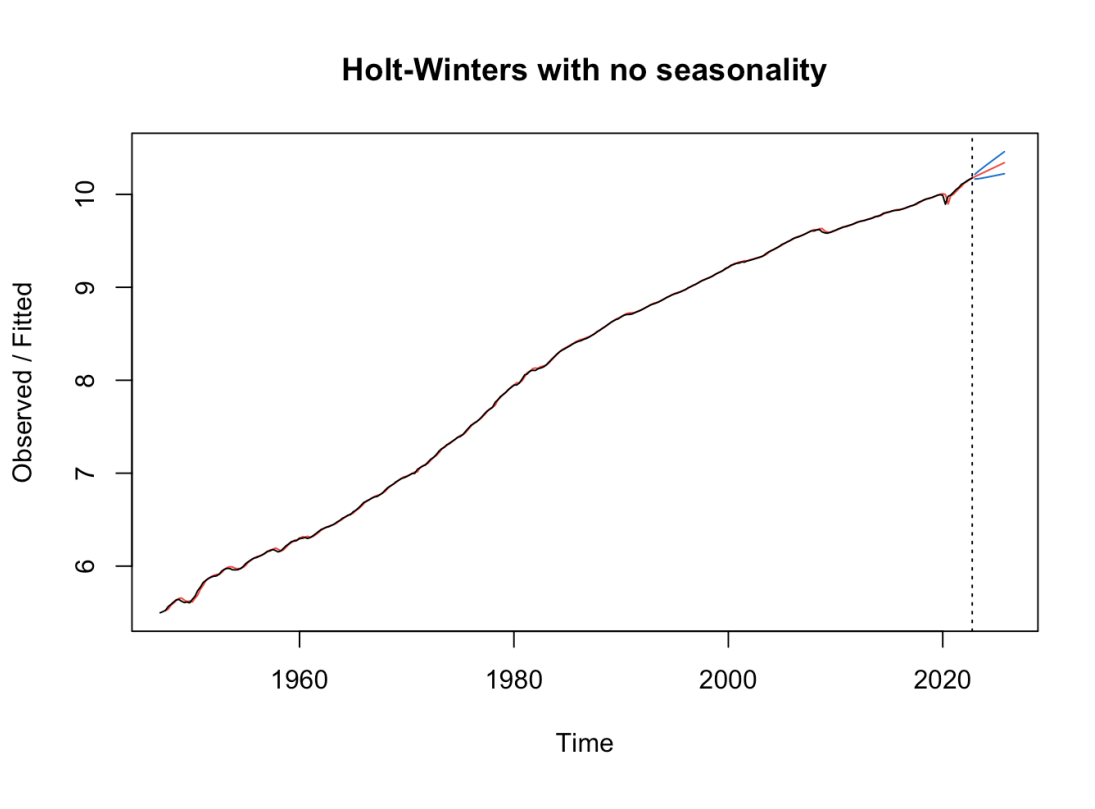
My choice to include a comparison in the form of a non-parametric forecast is that unlike parametric methods, it does not impose a functional form on m_t with unknown parameters. This in turn, offers a more data driven shape for m_t.

Understanding the seasonal component:

The seasonal component of the series refers to the regular, predictable patterns that repeat over specific periods. In either of the multiplicative or log additive models, It's important that the seasonal component effects the trend as minimal as possible. By minimising seasonal influence, the predictions become more stable and reflect the underlying pattern. Here, through triple exponential smoothing/Holt-Winters forecasting, I can show that the exclusion of the seasonal component does not significantly alter the predictions.



The graph above is the instance in which the model applies triple exponential smoothing, a process of averaging past values, to the series. Considering trend, level and seasonal components, Parameters α, β and γ control the degree of smoothing, setting parameters to zero or false effectively kills the component in the model:



Here, having set γ to false, I have produced a forecast of no seasonal effect, so the model had not accounted for seasonality.

Triple exponential smoothing prediction with prediction interval:

```
##            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
```

Disabling seasonal component:

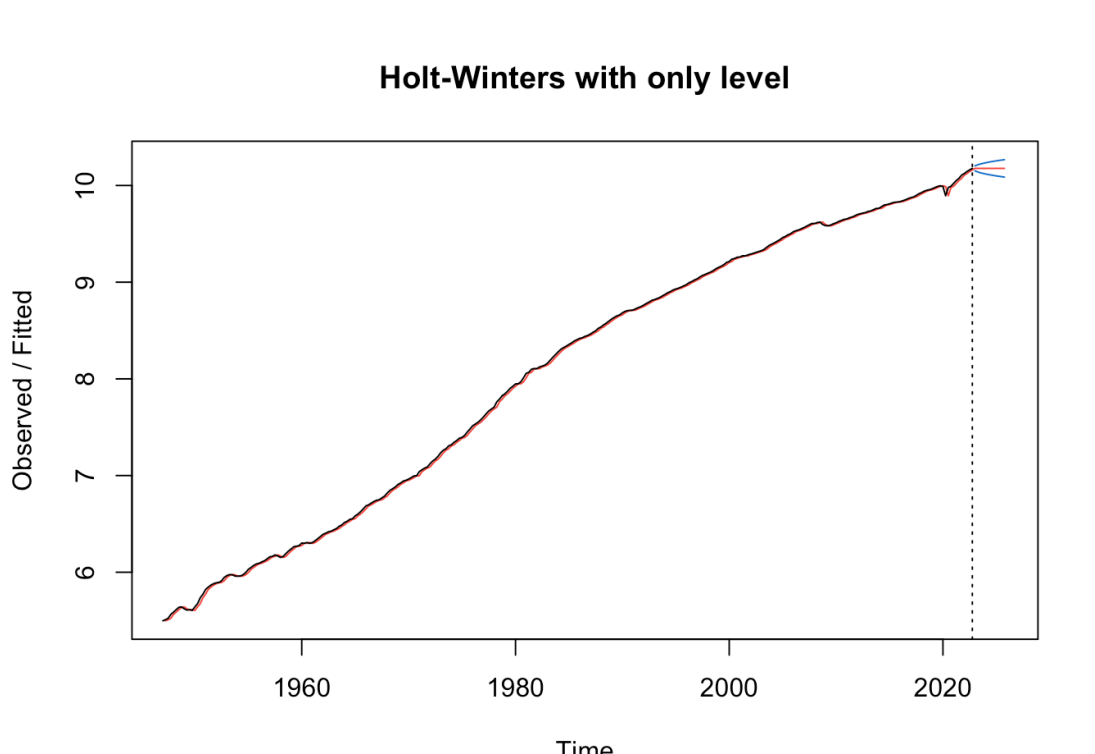
```
##            fit      upr      lwr
##  [7,] 10.27239 10.35200 10.19278
##  [8,] 10.28602 10.37347 10.19858
##  [9,] 10.29966 10.39490 10.20442
## [10,] 10.31330 10.41633 10.21026
## [11,] 10.32694 10.43778 10.21609
## [12,] 10.34057 10.45926 10.22189
```

Ultimately seeming indifferent to the full triple method. The notion is also satisfied through numerical representation, without seasonal components the predictions are not significantly altered. Evidently supporting that seasonality, the repeatable predictable patterns of GNP, is not a deciding factor in the determining of future values.

Importance of Holt-Winters trend b_t:

Terminology of Holt-Winters refers to regular trend (m_t) as the level component in which it is a necessary in providing a baseline for forecast. Trend, say now b_t, is understood to be the measurement of how much m_t and time series x_t are expected to increase.

First I considered the baseline of the level component by removing trend and seasonal effects:



The model above only considers level component. Graphically, the intervals appear wider and the estimate itself has seemingly leveled to one value.

This is supported numerically:

```
##            fit      upr      lwr
##  [7,] 10.17692 10.24566 10.10819
##  [8,] 10.17692 10.25040 10.10344
##  [9,] 10.17692 10.25486 10.09899
## [10,] 10.17692 10.25908 10.09477
## [11,] 10.17692 10.26309 10.09076
## [12,] 10.17692 10.26692 10.08693
```

The constant estimate of 10.17692 represents the baseline value in which the data with trend and seasonal effect fluctuates.

In comparing the differences of intervals from subtracting from the initial forecast, I wanted evidence in what decides the predictions of the triple forecast:

Difference of Triple model and disabled seasonal model with trend effect intact:

```
##                fit         upr           lwr
##  [7,]  0.005184006 0.010067993  0.0003000187
##  [8,]  0.003267487 0.008756670 -0.0022216953
##  [9,]  0.000327175 0.007209682 -0.0065553316
## [10,] -0.003232984 0.004322669 -0.0107886369
## [11,]  0.006817750 0.015107299 -0.0014717994
## [12,]  0.004901231 0.013979890 -0.0041774276
```

Consistent in that the seasonal component has minimal effect.

Difference of Triple model and disabled seasonal and disabled trend effect model:

```
##             fit       upr        lwr
##  [7,] 0.1006470 0.1164081 0.08488593
##  [8,] 0.1123680 0.1318239 0.09291208
##  [9,] 0.1230651 0.1472543 0.09887590
## [10,] 0.1331424 0.1615806 0.10470425
## [11,] 0.1568306 0.1898021 0.12385916
## [12,] 0.1685516 0.2063247 0.13077845
```

Shows the wider interval may stem from removing the trend component.

Now defining a model without trend :

```
##            fit      upr      lwr
##  [7,] 10.17952 10.25277 10.10627
##  [8,] 10.17692 10.25512 10.09873
##  [9,] 10.17432 10.25753 10.09111
## [10,] 10.17338 10.26098 10.08579
## [11,] 10.17952 10.27130 10.08775
## [12,] 10.17692 10.27270 10.08115
```

Difference of No trend model and level model:

```
##                 fit         upr          lwr
##  [7,]  2.599845e-03 0.007115224 -0.001915534
##  [8,]  7.797967e-07 0.004719834 -0.004718274
##  [9,] -2.602503e-03 0.002670290 -0.007875295
## [10,] -3.538999e-03 0.001906410 -0.008984409
## [11,]  2.599845e-03 0.008214300 -0.003014611
## [12,]  7.797967e-07 0.005780606 -0.005779046
```

It is evident there is minimal difference between level model and no trend model

The Model without trend behaved in a similar manner to the model with only level component. The similar behaviour in effecting estimates to remain relatively constant and wider intervals is evidence that trend, with level as a baseline, ultimately decides the values of the triple forecast model, with seasonal effect playing a minimal role.

Naturally, by definition of trend b_t, one could expect it play a role in the deliverance of predictions. But in having compared it in this manner I hope to of provided some definitive numerical representation in the minimal effect of seasonal patterns, and intuitively the importance of the underlying growth in GNP.

Comparing procedures of forecast:

Difference of Holt-Winters triple forecast and Prophet forecast intervals:

```
##                      ds yhat_lower  yhat_upper
## 311 2024-06-30 23:59:49 -0.2178448 0.001502060
## 312 2024-09-30 23:59:49 -0.2316553 0.005288619
## 313 2024-12-31 23:59:49 -0.2459929 0.013533392
## 314 2025-03-31 23:59:49 -0.2531173 0.027138119
## 315 2025-06-30 23:59:49 -0.2791589 0.021321653
## 316 2025-09-30 23:59:49 -0.2953027 0.026791093
```

Comparing characteristics of procedures:

The difference in prediction intervals stems from varying procedure features. Such as Prophet being able to adapt to shifts in trend, Holt-winters being the less adaptive and working better with a consistent pattern.

Based on my own findings and reading between articles and websites about the procedures (listed in references):

Both procedures use tuning parameters; seasonality and holiday priors of prophet and α, β and γ of Holt-Winters. Values of parameters for the Holt-Winters being pivotal to the forecast initialisation and predictions. Whereas with prophet, tuning parameters are not necessary in initialisation but can further enhance the accuracy of the forecast.

Prophet's procedure is also based on an additive Time-series model in which capabilities range from fitting non-linear trends and working best with strong seasonal effects. Holt-Winters is a generalised method that does not impose an additive or multiplicative model.

In Relation to the GNP dataset, I believe both procedures had effectively showcased their respective features. Despite for example, the GNP dataset being minimal in seasonal effect; contrasting to the benefit of the Prophet forecast.

Conclusion:

Prophet's flexibility stems in accounting for deviations/impacts; as evidently shown in accounting for COVID. Additionally the forecasting procedure, in my use, having the more straightforward and cleaner approach.

Holt-Winters I had found to of offered a complex but robust approach in accessing data-driven predictions. It's utilisation was especially emphasised in being able to deduce the relative importance of components. The main example being it provided numerical proof in seasonal effect (the regular predictive patterns of GNP growth) being the lesser of important components.

Concluding that the choice of either method is dependent on the complexity of the model and the necessity of insight in relative importance of components.

References:

. Source of GNP data

. Forecast using Prophet package tutorial in R

. Additional prophet package information

. 'Forecasting at Scale' article

. More on Holt-Winters method (application and theory)

. QMUL 'Time-Series' lecture notes

. QMUL 'Statistical Modelling 1' lecture notes (2021 and 2023\24 versions)