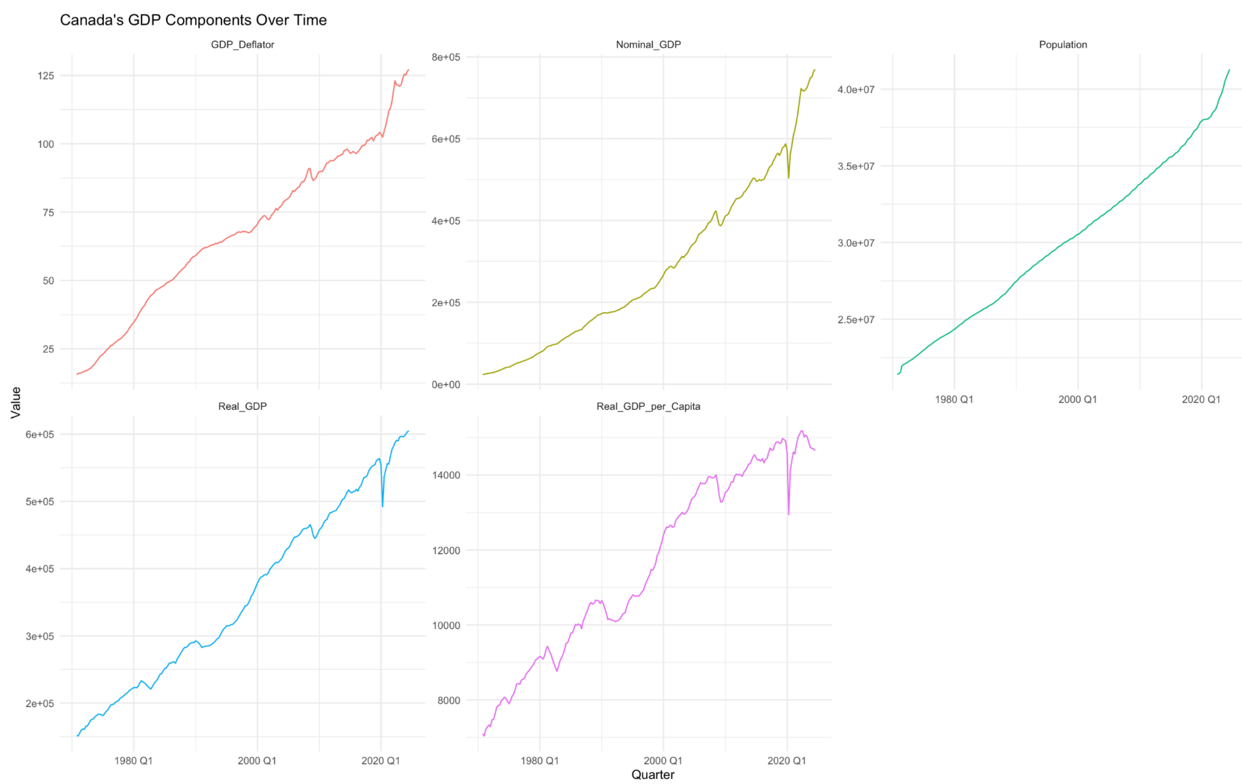
1.1

The Data was collected using the FRED API in R and Government of Canada Website.

Variable	Source	Source Link	Units	Frequency	Start Date	End Date
Population	Government of Canada	Gov Canada	People	Quarterly	1970-Q1	2024-Q3
Nominal GDP	FRED (NGDPSAXDCCAQ)	FRED API	Million CAD	Quarterly	1970-Q1	2024-Q3
Real GDP per Capita	FRED (NGDPRSAXDCCAQ)	FRED API	Million CAD	Quarterly	1970-Q1	2024-Q3

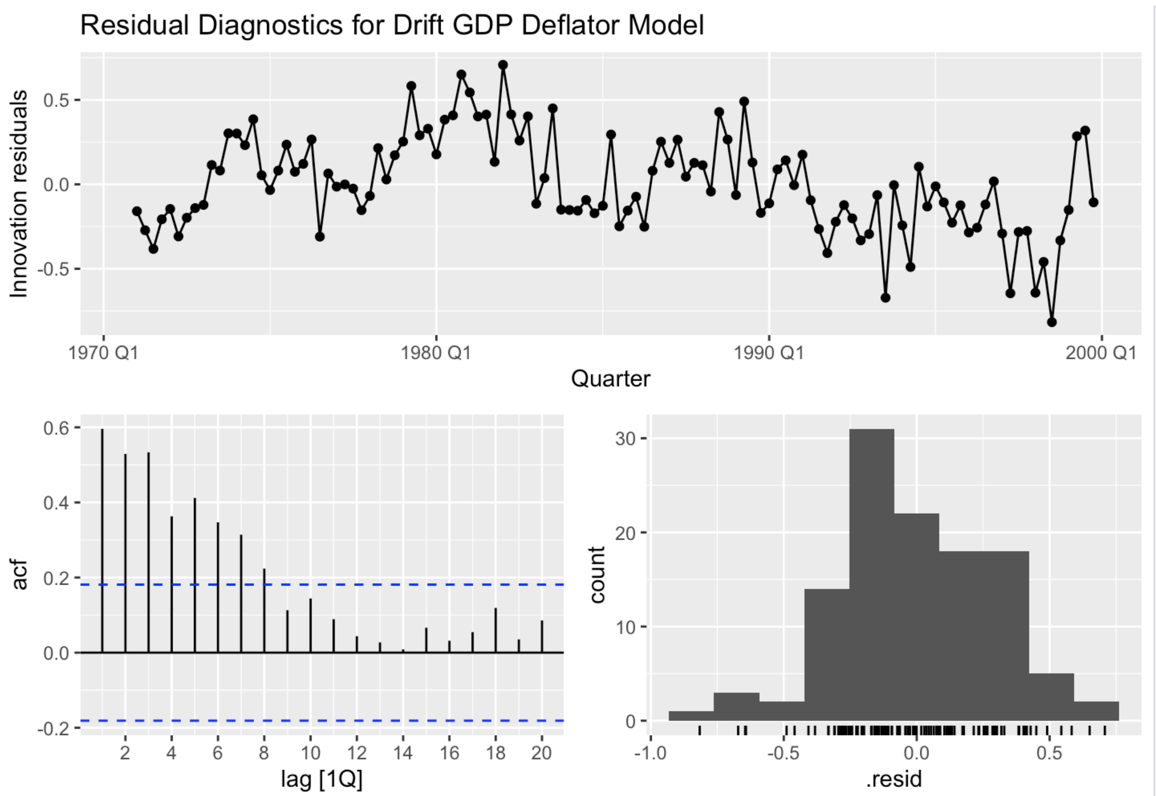
1.2



The figure above shows each of the components of GDP. We can see that all of them have an upward trend. GDP, Real GDP and Real GDP per Capita all have a sudden decrease in value in 2020 Q1, likely due to the coronavirus pandemic. We also see that there is a sudden increase in the population right after 2020 Q1. This is because of policies introduced by the Canadian Government at the time to attract a large number of immigrants to address labor shortages. The increase in population is also reflected in nominal GDP and GDP deflator. I tried generating summary statistics, but was unable to as I could not aggregate across the quarters.

1.3

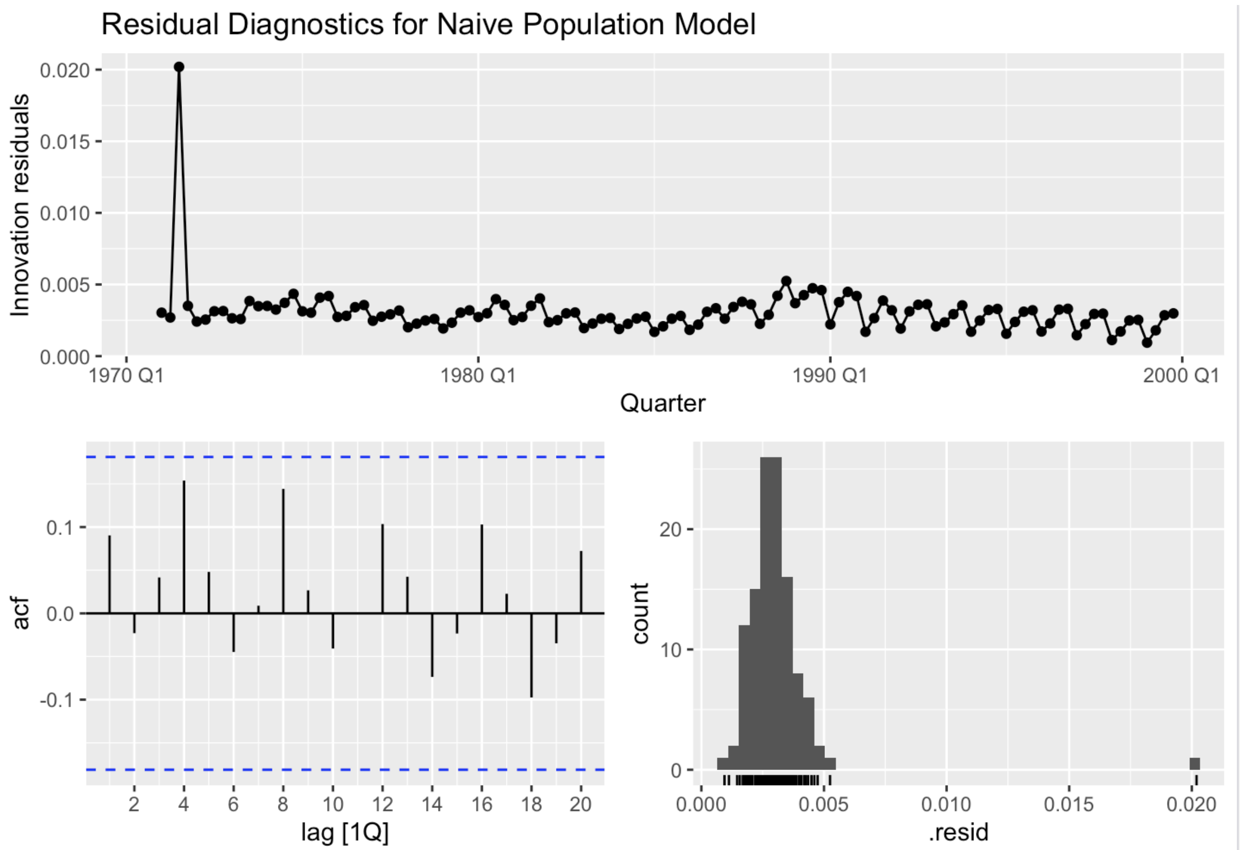
To plot the residuals, we used the 3 models discussed in class, Naive, Drift and Mean. After multiple transformations, including logs, we see that these models are not particularly good for this exercise. There is either high autocorrelation, or the distribution of the residuals is not normal or they do not average around 0. After many iterations, we find that the following models give the most acceptable results.



Here we can see that the residuals are almost scattered around 0 and the distribution is almost normal. The acf graph indicates some autocorrelation but it is the best that we have. The Ljung Box test further validates our hypothesis.

	.model	lb_stat	lb_pvalue
	<chr>	<dbl>	<dbl>
1	Drift	186.	0
2	Mean	967.	0
3	Naïve	186.	0

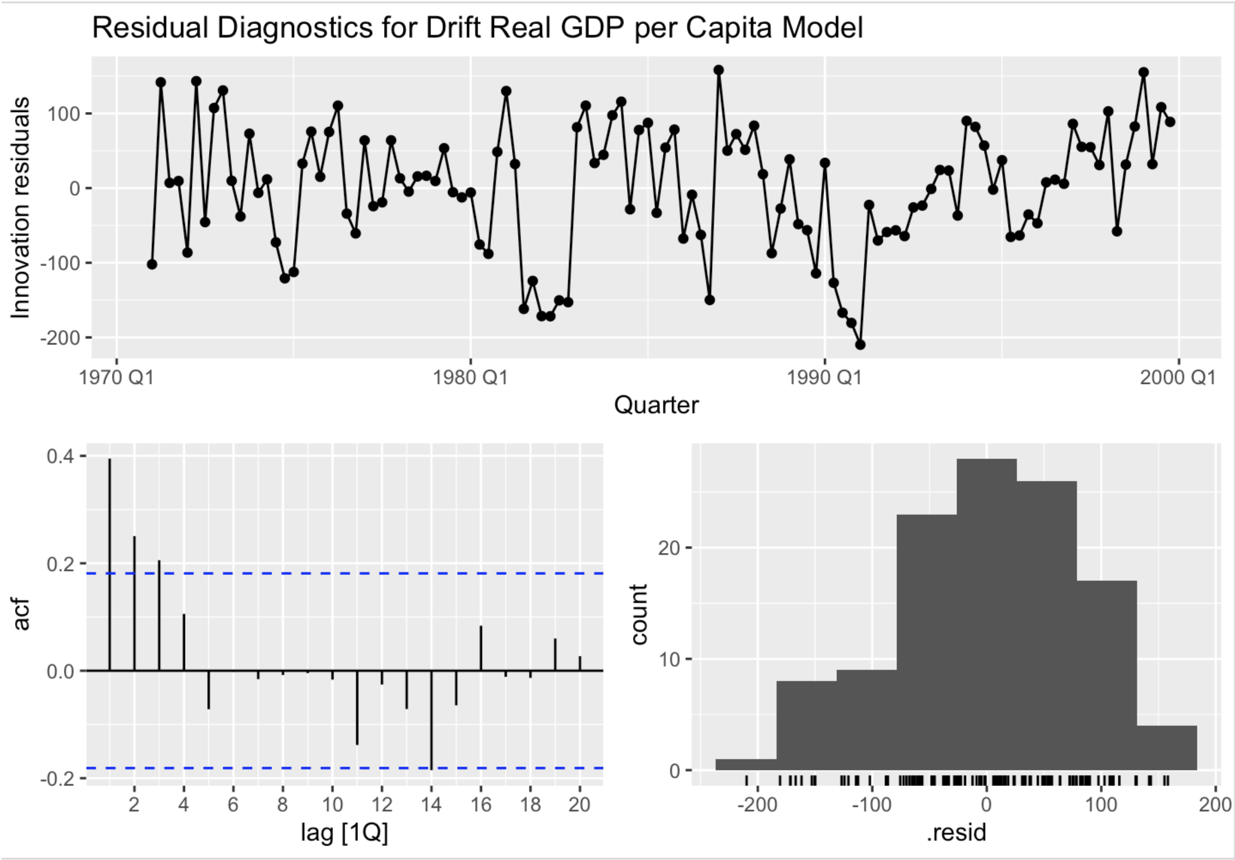
We apply the same logic to both the Population and the Real GDP per Capita.



Despite the initial jump, there is no clear trend in the residuals and the graphs suggest that there is less likely to be autocorrelation as well. Residuals are right-skewed (due to the large spike). Otherwise, the distribution is relatively normal. Also thinking about it logically, the population of Canada has been increasing the last 50 years, so using the Naïve model makes sense as we can just forecast based on the last point.

Looking at the Ljung Box Test we can further conclude that the Naive Model is a good fit.

	.model	lb_stat	lb_pvalue
	<chr>	<dbl>	<dbl>
1	Drift	7.63	0.665
2	Mean	919.	0
3	Naïve	7.63	0.665



Looking at the top panel we see fluctuations over time but no clear pattern of non-stationarity, indicating that the model captures the general trend well. However, some clusters of high residual values suggest periods of under- or over-prediction. The bottom-left panel presents the autocorrelation function (ACF) of residuals, where significant spikes at early lags indicate some level of autocorrelation, suggesting that the residuals may not be entirely random. Finally, the histogram of residuals in the bottom-right panel appears roughly normal, but slight skewness may be present.

The Ljung Box Test is given below.

```
# A tibble: 3 × 3
  .model lb_stat lb_pvalue
  <chr>    <dbl>     <dbl>
1 Drift     33.3  0.000242
2 Mean     792.   0
3 Naïve     33.3  0.000242
```

1.4

Looking at the Battery results for the Test Data, we can interpret that the Drift Model seems to be most accurate for the deflator and the real GDP forecasts and nominal forecasts. The ME values are the lowest for each of them amongst the models and the MAPE is also the smallest. There seems to be some problem with the population dataset. The higher ACF does indicate the possibility of autocorrelation.

Unfortunately due to some errors I was unable to calculate the evaluation metrics for the training dataset. The results indicate that there may have been some error

```
> accuracy(fc_deflator, test_data)
# A tibble: 3 × 10
  .model .type     ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>  <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Drift  Test   0.921  3.81  2.57 0.784  2.46   NaN   NaN 0.924
2 Mean   Test  48.0   50.2  48.0  49.9   49.9    NaN   NaN 0.958
3 Naïve  Test  24.3   28.4  24.3  24.0   24.0    NaN   NaN 0.958
> accuracy(fc_population, test_data)
# A tibble: 3 × 10
  .model .type        ME      RMSE       MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>  <chr>     <dbl>     <dbl>     <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Drift  Test  34858705. 34972340. 34858705.  100.  100.   NaN   NaN 0.961
2 Mean   Test  34858706. 34972341. 34858706.  100.  100.   NaN   NaN 0.961
3 Naïve  Test  34858706. 34972340. 34858706.  100.  100.   NaN   NaN 0.961
> accuracy(fc_real_gdp, test_data)
# A tibble: 3 × 10
  .model .type    ME  RMSE   MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>  <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Drift  Test  -459.  807.  626. -3.14  4.38   NaN   NaN 0.912
2 Mean   Test  4389. 4452. 4389. 31.2  31.2    NaN   NaN 0.928
3 Naïve  Test  1753. 1905. 1753. 12.3  12.3    NaN   NaN 0.928

> accuracy(fc_nominal_gdp, test_data)
# A tibble: 3 × 10
  .model .type      ME    RMSE     MAE   MPE  MAPE  MASE RMSSE  ACF1
  <chr>  <chr>   <dbl>   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Drift  Test  107040. 132333. 107040.  20.1  20.1   NaN   NaN 0.946
2 Mean   Test  343210. 368184. 343210.  70.9  70.9   NaN   NaN 0.958
3 Naïve  Test  208988. 247875. 208988.  40.0  40.0   NaN   NaN 0.958
```

Q2

Bias Correction for Box-Cox Transformation

We have the Box-Cox transformation defined as:

$$f(x) = \begin{cases} (\lambda x + 1)^{\frac{1}{\lambda}}, & \lambda \neq 0 \\ e^x, & \lambda = 0 \end{cases}$$

We aim to derive the bias-corrected expectation of the back-transformed mean y=f(x)y = f(x).

Step 1: Setup and Motivation

Due to the nonlinearity of the transformation, simply applying the inverse transform directly at the mean, i.e., f(μ)f(\mu), introduces bias. To correct for this, we use a second-order Taylor expansion around the mean μ\mu:

$$E[f(x)] \approx f(\mu) + \frac{1}{2} f''(\mu)\sigma^2$$

Step 2: Second Derivative Calculation

We compute the second derivative explicitly. For $\lambda \neq 0$

$$f(x) = (\lambda x + 1)^{\frac{1}{\lambda}}$$

The first derivative is:

$$f'(x) = (\lambda x + 1)^{\frac{1-\lambda}{\lambda}}$$

The second derivative is:

$$f''(x) = (1 - \lambda)(\lambda x + 1)^{\frac{1-2\lambda}{\lambda}}$$

Evaluating at $x=\mu$, we have

$$f''(\mu) = (1-\lambda)(\lambda\mu+1)^{\frac{1-2\lambda}{\lambda}}$$

Step 3: Apply Taylor Approximation

Applying the delta method approximation, we obtain:

$$E[y] = E[f(x)] \approx f(\mu) + \frac{1}{2} f''(\mu)\sigma^2$$

Substituting explicitly for λ≠0

$$E[y] \approx (\lambda\mu+1)^{\frac{1}{\lambda}} + \frac{1}{2}(1-\lambda)(\lambda\mu+1)^{\frac{1-2\lambda}{\lambda}}\sigma^2$$

Step 4: Simplify the Expression

Factoring out

$$E[y] \approx (\lambda\mu+1)^{\frac{1}{\lambda}} \left[1 + \frac{\sigma^2(1-\lambda)}{2(\lambda\mu+1)^2}\right]$$

For the special case λ=0

$$E[y] \approx e^{\mu} + \frac{1}{2} e^{\mu}\sigma^2 = e^{\mu}\left[1 + \frac{\sigma^2}{2}\right]$$

Final Bias-Corrected Result

The final bias-corrected expectation for the back-transformed mean is:

$$\boxed{E[y] \approx \begin{cases} (\lambda\mu+1)^{\frac{1}{\lambda}} \left[1 + \frac{\sigma^2(1-\lambda)}{2(\lambda\mu+1)^2}\right], & \lambda \neq 0 \\ e^{\mu}\left[1 + \frac{\sigma^2}{2}\right], & \lambda = 0 \end{cases}}$$