

TSAR Project Part 3

Michèle Fille

May 14, 2024

Table of contents

List of Figures	1
Introduction, Preliminaries, and Loading Time Series from Disk	3
Task 1: Extract the Training Set	6
1st Benchmark Model using the mean method (Task 2-5)	7
Task 2: Train the Benchmark Model	7
Task 3: Evaluate the Model Fit of the Benchmark Model	8
Task 3a: Create a time plot 'Actual vs. Fitted'	8
Task 3b: Create a scatter plot 'Actual vs. Fitted'	9
Task 3c: Perform residual diagnostics to inspect the model fit	10
Task 3d: Double-check your results from (c)	12
Task 4: Evaluate the Point Forecast Accuracy of the Benchmark Model	13
Task 5: Evaluate the Point Forecast Uncertainty of the Benchmark Model	14
2nd Benchmark Model using the naive methos (Task 6)	16
Task 2: Train the Benchmark Model	16
Task 3: Evaluate the Model Fit of the Benchmark Model	17
Task 3a: Create a time plot 'Actual vs. Fitted'	17
Task 3b: Create a scatter plot 'Actual vs. Fitted'	18
Task 3c: Perform residual diagnostics to inspect the model fit	19
Task 3d: Double-check your results from (c)	21
Task 4: Evaluate the Point Forecast Accuracy of the Benchmark Model	22
Task 5: Evaluate the Point Forecast Uncertainty of the Benchmark Model	23
Task 7: Compare the Evaluation Results of the Two Models: Mean vs. Naive	25

List of Figures

1	Time plot fitted and true values for monthly US Employment (Mean Model)	8
2	Scatter plot of fitted and true values for monthly US Employment with an identity line (Mean Model)	9
3	Residual plots of the mean model trained on monthly US Employment	10
4	Histogram of residual for the mean model trained on monthly US Employment	11
5	Time plot of Actual vs. Forecasted Values for the mean model trained on monthly US Employment .	13
6	Residual plots of the mean model trained on monthly US Employment	14
7	Time plot fitted and true values for monthly US Employment (Naive Model)	17
8	Scatter plot of fitted and true values for monthly US Employment with an identity line (Naive Model)	18
9	Residual plots of the naive model trained on monthly US Employment	19
10	Histogram of residual for the naive model trained on monthly US Employment	20
11	Time plot of Actual vs. Forecasted Values for the naive model trained on monthly US Employment .	22
12	Residual plots of the naive model trained on monthly US Employment	23

Introduction, Preliminaries, and Loading Time Series from Disk

Project part 3 is about comparing two benchmark models and choosing the better one. Therefore the `us_employment` dataset was filtered on the `Series_ID` 'CEU1021000001' to get the US monthly employment data for 'Mining and Logging: Mining'. As nearly 24% of the months, spanning from January 1939 to June 2019, had no data (NA), these months, respective these rows were deleted. The filtered and cleaned data, was then saved as `.rds` file. These preliminary steps are included in the following R Code Chunk but are commented out, as they only need to be done once. To uncomment the below code highlight it and press `Ctrl + Shift + C`.

```

# #----- 1) Information from Moodle-----
# # Series_ID = CEU1021000001
# # Title = Mining and Logging: Mining
#
# #----- 2) Access and Inspect dataset -----
# # Install (if necessary) and Load the 'fpp3' package
# # install.packages('fpp3')
# library(fpp3)
#
# # View the 'us_employment' dataset for further inspection
# view(us_employment)
#
# #----- 3) Filter Data -----
# # Load the dplyr package
#
# # Filter the 'us_employment' dataset based on my Series_ID
# filtered_data <- us_employment %>%
#   filter(Series_ID == "CEU1021000001")
#
# # View the filtered dataset
# view(filtered_data)
#
# # Plot the filtered dataset using autoplot
# autoplot(filtered_data)
#
# #----- 4) Check and Remove NAs -----
# # Calculate the total number of rows in the dataset
# total_rows <- nrow(filtered_data)
#
# # Calculate and print the NA values in each column
# na_counts <- colSums(is.na(filtered_data))
# print(na_counts)
#
# # Calculate and print the percentage of NA values in each column
# na_percentages <- paste0(round((na_counts / total_rows) * 100, 2), '%')
# print(na_percentages)
#
# # Remove rows with NA values
# filtered_data_clean <- na.omit(filtered_data)
#
# # Check the dimensions of the cleaned dataset
# print(dim(filtered_data_clean))
#
# # Calculate and print the NA values in each column in the cleaned dataset
# na_counts_1 <- colSums(is.na(filtered_data_clean))
# print(na_counts_1)
#
# #----- 5) Save the Time Series on disk -----
# # Save the resulting time series on disk
# saveRDS(filtered_data_clean, file = "filtered_data_clean.rds")

```

To train, evaluate, and compare 2 different benchmark models (out of the 5 listed below) on my series, the prepared time series from the R Code Chunk above, gets below loaded into from the disk.

5 Possible methods for the benchmark models:

1. the mean method (see Part 2, Lecture 3),
2. the naive method (see Part 2, Lecture 3),
3. the seasonal naive method (see Part 2, Lecture 3),
4. the drift method (see Part 2, Lecture 3),
5. the seasonal naive method with drift (see Part 2, Self-Study 3, Task 2)

```
# Load the saved time series from disk
mySeries <- readRDS("filtered_data_clean.rds")

head(mySeries)
```

	Month	Series_ID	Title	Employed
1	-4383	CEU1021000001	Mining and Logging: Mining	756.6
2	-4352	CEU1021000001	Mining and Logging: Mining	736.7
3	-4324	CEU1021000001	Mining and Logging: Mining	721.0
4	-4293	CEU1021000001	Mining and Logging: Mining	706.2
5	-4263	CEU1021000001	Mining and Logging: Mining	701.9
6	-4232	CEU1021000001	Mining and Logging: Mining	708.3

```
# Load the necessary libraries
library(dplyr)
library(fpp3)
library(ggplot2)
```

All necessary libraries are loaded here to prevent errors during rendering. For example, when loading the **fpp3** library in the same code chunk where the mean model is trained, the following error is thrown:

```
==> quarto preview P3-04-Fille.qmd -to pdf -no-watch-inputs -no-browse
```

```
processing file: P3-04-Fille.qmd |..... | 46% [unnamed-chunk-3] Fehler: Auswertung zu tief verschachtelt:
unendliche Rekursion / options(expressions=)?
```

Ausführung angehalten

Note: Tibbels are not labeled, and no captions are created, as they disrupt the format of the quarto file and PDF document after rendering.

Task 1: Extract the Training Set

```
# Calculate 80% of the total rows and round it to an integer value
num_train <- floor(nrow(mySeries) * 0.8)
num_train # 592 are about 80% out of 741
```

```
[1] 592
```

```
# Extract the first 80% of the rows using slice()
train <- mySeries %>%
  slice(1:num_train)
```

```
head(train)
```

```
# A tsibble: 6 x 4 [1M]
# Key:      Series_ID [1]
   Month Series_ID      Title      Employed
   <mth> <chr>      <chr>      <dbl>
1 1958 Jan CEU1021000001 Mining and Logging: Mining 757.
2 1958 Feb CEU1021000001 Mining and Logging: Mining 737.
3 1958 Mär CEU1021000001 Mining and Logging: Mining 721
4 1958 Apr CEU1021000001 Mining and Logging: Mining 706.
5 1958 Mai CEU1021000001 Mining and Logging: Mining 702.
6 1958 Jun CEU1021000001 Mining and Logging: Mining 708.
```

```
# Check the length of the "test set" to forecast for the same length
test_length <- nrow(mySeries) - num_train
test_length # 741 - 592 = 149, what is the remaining 20% (ca.)
```

```
[1] 149
```

1st Benchmark Model using the mean method (Task 2-5)

Task 2: Train the Benchmark Model

```
# Fit the model to the column Employed
mean_model <- train |>
  model(
    Mean = MEAN(Employed)
  )

# Using augment() to get a first impression of the fitted values
mean_fitted_values <- augment(mean_model)

head(mean_fitted_values)
```

```
# A tsibble: 6 x 7 [1M]
# Key:   Series_ID, .model [1]
  Series_ID      .model   Month Employed .fitted .resid .innov
  <chr>          <chr>    <mth>   <dbl>   <dbl>   <dbl>   <dbl>
1 CEU1021000001 Mean    1958 Jan     757.    673.    83.2    83.2
2 CEU1021000001 Mean    1958 Feb     737.    673.    63.3    63.3
3 CEU1021000001 Mean    1958 Mär     721.    673.    47.6    47.6
4 CEU1021000001 Mean    1958 Apr     706.    673.    32.8    32.8
5 CEU1021000001 Mean    1958 Mai     702.    673.    28.5    28.5
6 CEU1021000001 Mean    1958 Jun     708.    673.    34.9    34.9
```


Task 3: Evaluate the Model Fit of the Benchmark Model

Task 3a: Create a time plot 'Actual vs. Fitted'

```
# Time plot of actual vs. fitted values for the mean method
augment(mean_model) %>%
  autoplot(.fitted, colour = "purple", linetype = "dashed") +
  autolayer(train, Employed) +
  labs(y = "Employment",
       title = "Time plot of Actual vs. Fitted")
```

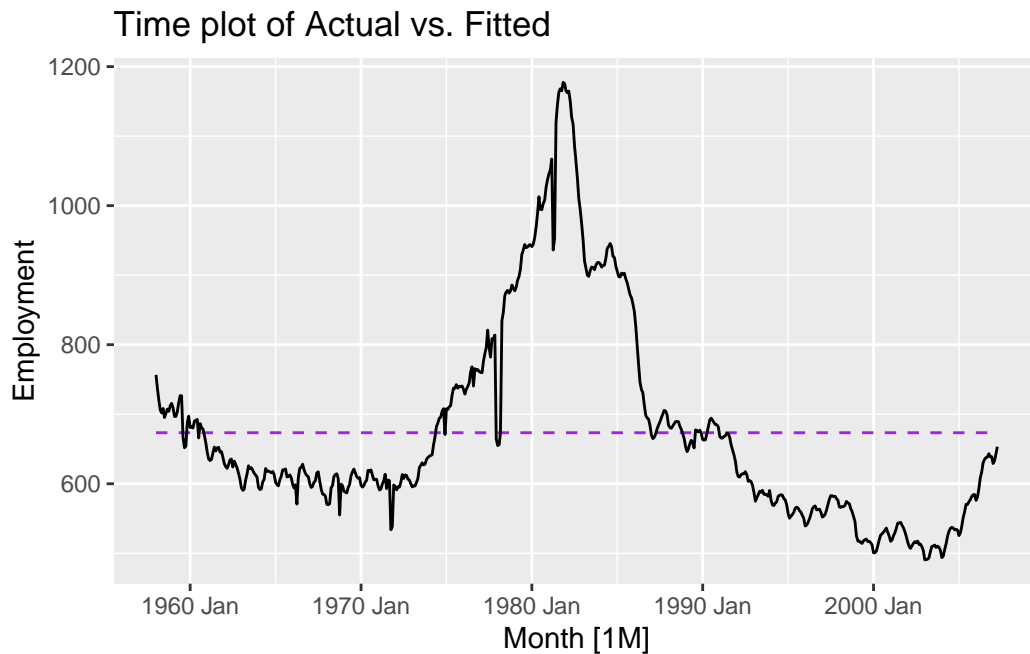


Figure 1: Time plot fitted and true values for monthly US Employment (Mean Model)

The following questions are answered based on Figure 1.

- **Are the fitted values close to the actual values?**

No, the mean-fitted values are not close to the actual values, as they do not capture the fluctuating pattern observed in the actual data due to their constant nature.

- **Are they systematically too high or too low?**

No, the mean-fitted values are not systematically too high or too low, as they remain constant, representing the mean of the data.

- **If yes, where does it come from? (Consider how your benchmark model works!)**

NA, as they are not systematically too high or too low.

Task 3b: Create a scatter plot 'Actual vs. Fitted'

```
# Scatter plot of actual vs. fitted values for the mean method
ggplot(data = mean_fitted_values, aes(x = .fitted, y = Employed)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = 2) +
  labs(x = "Fitted Values", y = "Actual Values") +
  ggtitle("Scatter plot of Actual vs. Fitted")
```

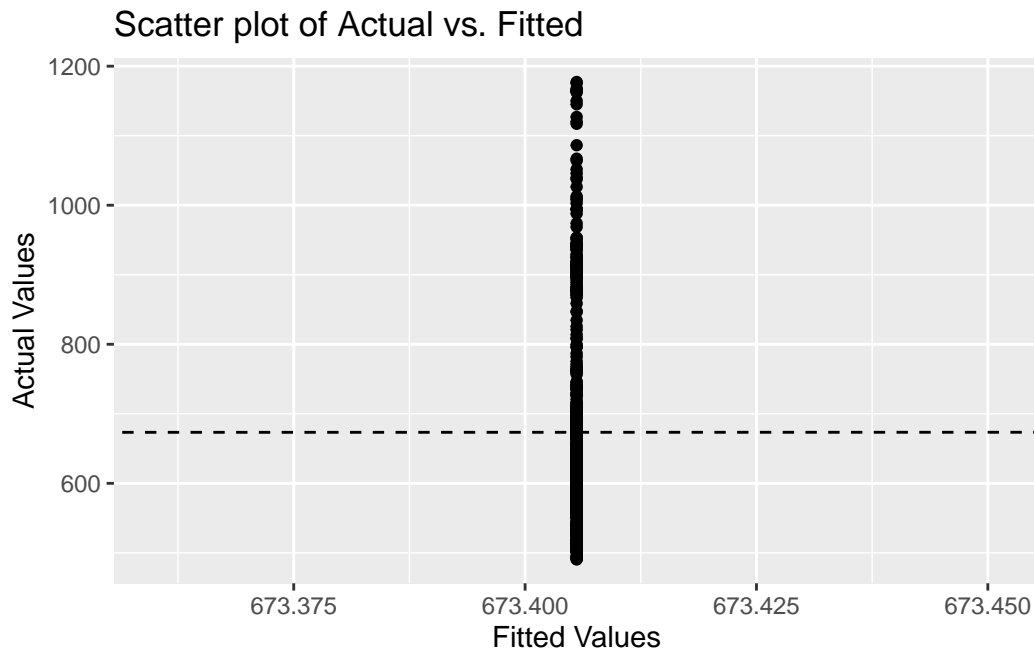


Figure 2: Scatter plot of fitted and true values for monthly US Employment with an identity line (Mean Model)

The following questions are answered based on Figure 2.

- **Are the points close to the identity line?**

No, the points are not close to the identity line, instead, they intersect it vertically due to the constant nature of the mean-fitted values.

- **Are they systematically too high or too low?**

No, the points are not systematically too high or too low, as their vertical intersection with the identity line reflects the constant mean-fitted values.

- **If yes, where does it come from? (Relate your answer to what you saw in the time plot.)**

NA, as they are not systematically too high or too low.

Task 3c: Perform residual diagnostics to inspect the model fit

```
# Residual plots of the mean method
gg_tsresiduals(mean_model) +
  labs(title = "Residual Plots of the Mean Model")
```

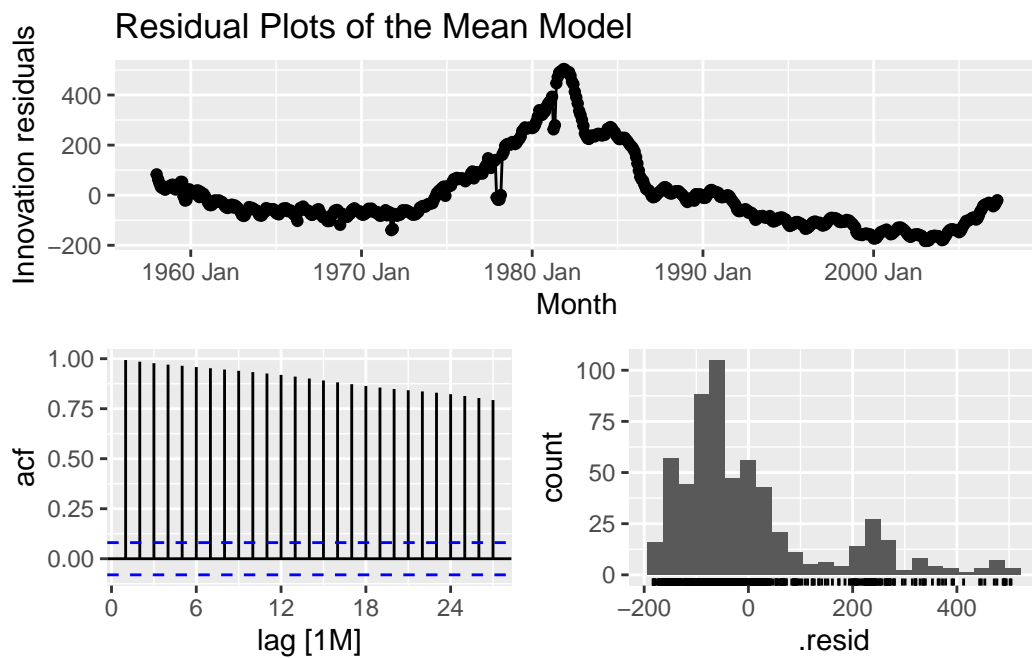


Figure 3: Residual plots of the mean model trained on monthly US Employment

```
# If no transformation has been applied the innovation residuals, column '.innov' are identical to
# the residuals, column '.resid'. Therefore the column '.innov' is used for the following histogram.

# Residual histogram with mean line for the mean model
ggplot(augment(mean_model), aes(x = .innov)) +
  geom_histogram() +
  geom_vline(aes(xintercept = mean(.innov, na.rm= TRUE)), color = "red", linetype = "dashed") +
  labs(title = "Histogram of Residuals with Mean Line (Mean Model)")
```

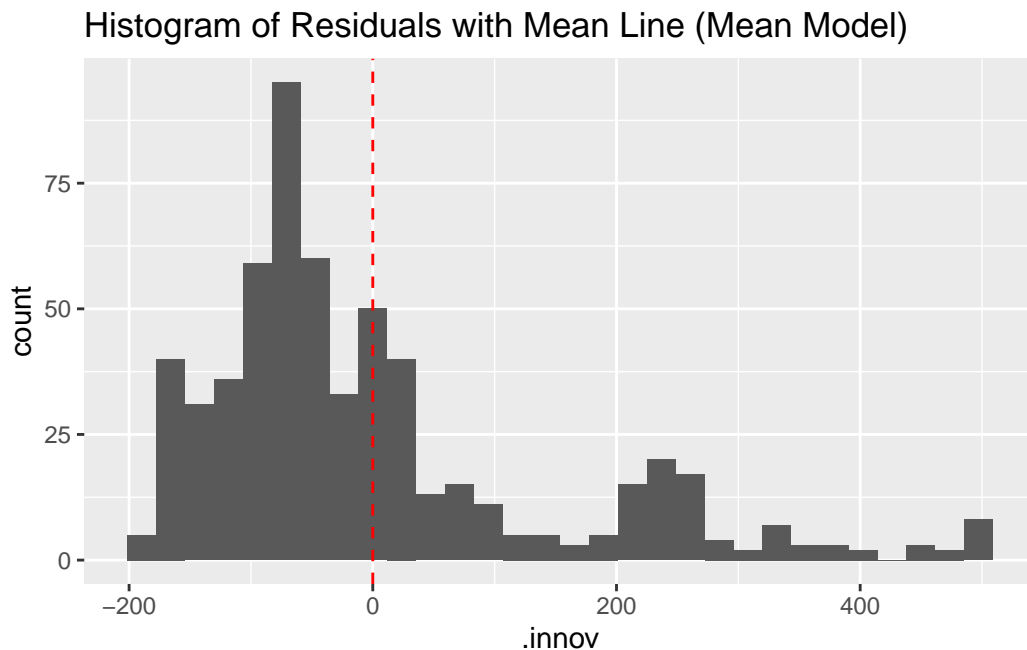


Figure 4: Histogram of residual for the mean model trained on monthly US Employment

The following questions are answered based on Figure 3.

- **Are the residuals auto-correlated? How do you decide that based on the plots?**

Yes, based on the residual ACF plot, the residuals are auto-correlated, as all autocorrelation coefficients (black spikes) clearly exceeding the confidence intervals (blue dotted lines) for all lag values.

- **Do the residuals have zero mean? How do you decide that based on the plots?**

Yes, based on the histogram of residuals from Figure 3 and Figure 4, the residuals have a zero mean, or at least their mean is very close to 0.

Task 3d: Double-check your results from (c)

```
# Ljung-Box test for autocorrelation in residuals
augment(mean_model) |>
  features(.innov, ljung_box, lag = 27) # 27 lags, as acf plot shows 27 autocorrelation coefficients
```

```
# A tibble: 1 x 4
  Series_ID      .model lb_stat lb_pvalue
  <chr>         <chr>    <dbl>    <dbl>
1 CEU1021000001 Mean      13267.        0
```

- Does the test result support your conclusions from (c)? How do you conclude that from the test result?

Yes, the Ljung-Box test result supports the conclusion from (c) that there is autocorrelation in the residuals, as indicated by the p-value of 0, suggesting statistically significant autocorrelation.

```
# Calculate the residuals
mean_residuals <- train$Employed - mean_fitted_values$.fitted
```

```
# Calculate the residual mean
mean(mean_residuals, na.rm = TRUE)
```

```
[1] -9.47581e-15
```

```
# or directly calculate the residual mean
mean(mean_fitted_values$.innov, na.rm = TRUE)
```

```
[1] -9.47581e-15
```

- Does the result support your conclusions from (c)?

The result from the residual mean supports the conclusion from (c), as the residual mean is -9.47581e-15 (= -0.00000000000000947581), what is really close to 0.

Task 4: Evaluate the Point Forecast Accuracy of the Benchmark Model

```
# Feed the trained model into the forecast() function
forecast_mean <- forecast(mean_model, h = test_length) #test_length = 149

# Plot actual vs. forecast values
forecast_mean %>% autoplot(mySeries)
```

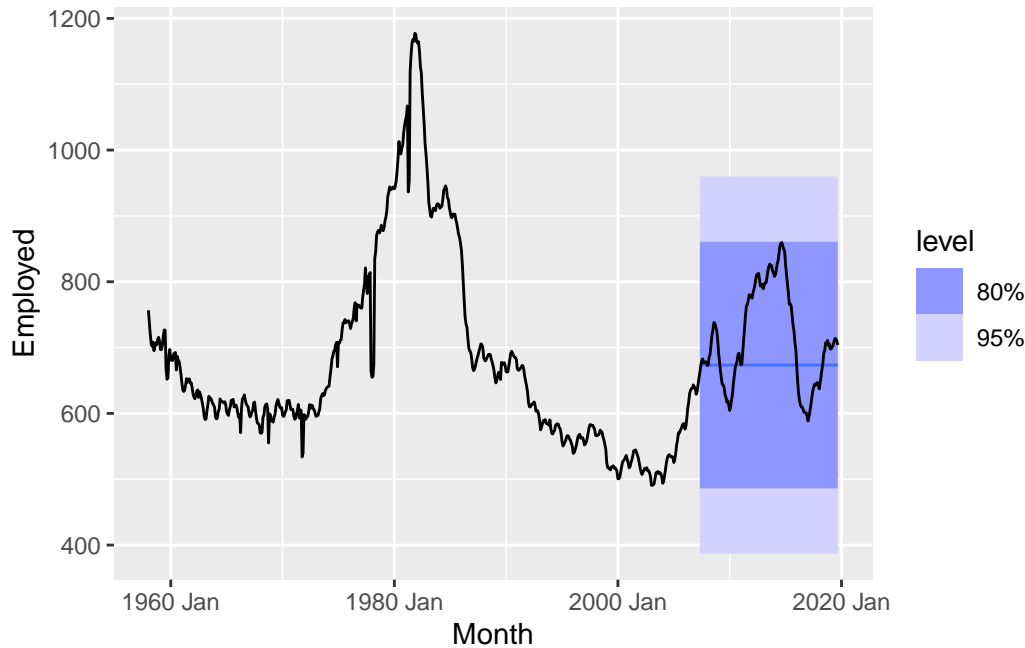


Figure 5: Time plot of Actual vs. Forecasted Values for the mean model trained on monthly US Employment

The time plot in Figure 5 shows that the data exhibits cyclic behavior and potentially some seasonality, but there is no clear trend evident.

```
# Calculate the forecast accuracy metrics
forecast_mean %>% accuracy(mySeries)
```

```
# A tibble: 1 x 11
  .model Series_ID .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1
  <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Mean CEU1021000001 Test 40.2 84.9 67.2 4.60 8.96 1.79 1.45 0.991
```

The Root Mean Squared Error (RMSE) is scale-dependent and measures the average magnitude of the errors between the actual and forecasted values. A value of 84.91 suggests that, on average, the forecasted values differ from the actual values by approximately 84.91 units. Considering the significant fluctuations in the data as the time plot in Figure 5 shows, the RMSE can be considered relatively high, indicating that the model's forecasts deviate considerably from the actual values.

The Mean Absolute Percentage Error (MAPE) measures the average percentage difference between the forecasted and actual values. A value of 8.96 indicates that, on average, the model's forecasts deviate from the actual values by approximately 8.96% of the actual values. Given the variability evident in the time plot in Figure 5 the MAPE suggests that the model's forecasts have a moderate level of accuracy, as the error rate is less than 10%.

Task 5: Evaluate the Point Forecast Uncertainty of the Benchmark Model

```
# Residual plots of the mean method
gg_tsresiduals(mean_model) +
  labs(title = "Residual Plots of the Mean Model")
```

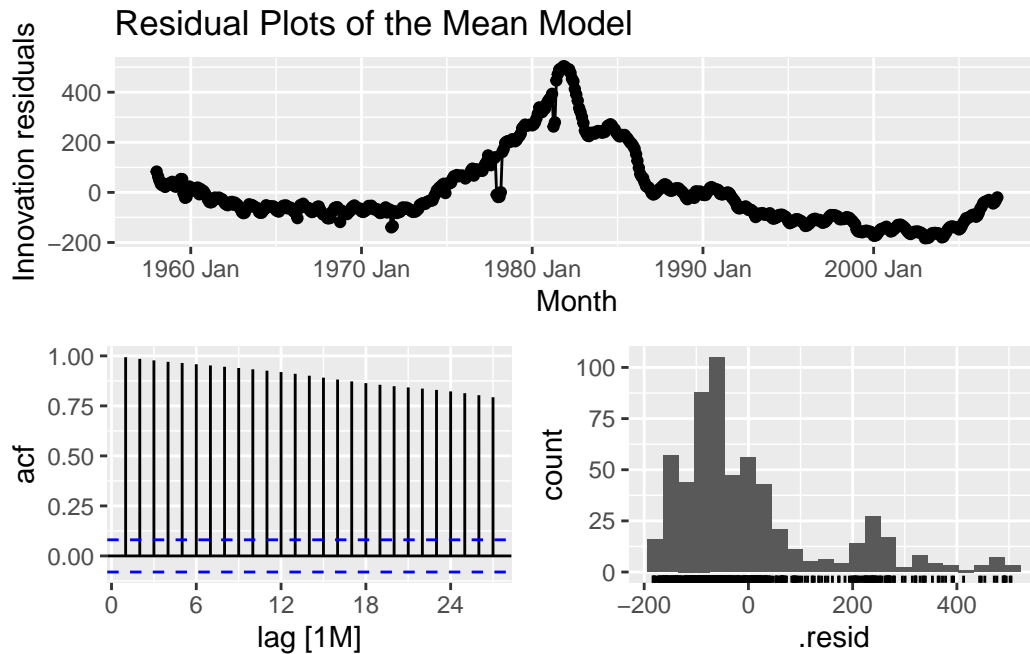


Figure 6: Residual plots of the mean model trained on monthly US Employment

The following questions are answered based on Figure 6.

- **Are the residuals homoscedastic? (That is, do they have constant variance?)**

Looking at the time plot of the residual, it seems that the variance is not constant, thus the series seems to be not homoscedastic (i.e., it is “heteroscedastic”).

- **Are the residuals normally distributed?**

The histogram shows that the distribution is not normal, as it has more than one peak, and therefore is multimodal instead of unimodal and, as discussed above, its mean is zero, or nearly zero with $-9.47581e-15$.

```
# Display the prediction intervals' boundaries
forecast_mean_intervals <- hilo(forecast_mean)

head(forecast_mean_intervals)
```

```
# A tsibble: 6 x 7 [1M]
# Key:      Series_ID, .model [1]
  Series_ID      .model    Month      Employed .mean      `80%`
  <chr>          <chr>    <mth>      <dist> <dbl>      <hilo>
1 CEU1021000001 Mean    2007 Mai N(673, 21356) 673. [486.1218, 860.6894] 80
2 CEU1021000001 Mean    2007 Jun N(673, 21356) 673. [486.1218, 860.6894] 80
3 CEU1021000001 Mean    2007 Jul N(673, 21356) 673. [486.1218, 860.6894] 80
4 CEU1021000001 Mean    2007 Aug N(673, 21356) 673. [486.1218, 860.6894] 80
5 CEU1021000001 Mean    2007 Sep N(673, 21356) 673. [486.1218, 860.6894] 80
6 CEU1021000001 Mean    2007 Okt N(673, 21356) 673. [486.1218, 860.6894] 80
# i 1 more variable: `95%` <hilo>
```

- Does it make sense to include these prediction intervals in your model evaluation? Why/why not?

Yes, it makes sense to include these prediction intervals, in the model evaluation, as they provide valuable information about the uncertainty associated with the predictions and thus help to understand the range of possible outcomes.

2nd Benchmark Model using the naive methos (Task 6)

Task 2: Train the Benchmark Model

```
# Fit the model to the column Employed
naive_model <- train |>
  model(
    'Naive' = NAIVE(Employed)
  )

# Using augment() to get a first impression of the fitted values
naive_fitted_values <- augment(naive_model)

head(naive_fitted_values)
```

```
# A tsibble: 6 x 7 [1M]
# Key:   Series_ID, .model [1]
  Series_ID      .model   Month Employed .fitted .resid .innov
  <chr>          <chr>    <mth>   <dbl>   <dbl>   <dbl>   <dbl>
1 CEU1021000001 Naive   1958 Jan    757.    NA     NA     NA
2 CEU1021000001 Naive   1958 Feb    737.   757.  -19.9  -19.9
3 CEU1021000001 Naive   1958 Mär    721.   737.  -15.7  -15.7
4 CEU1021000001 Naive   1958 Apr    706.   721.  -14.8  -14.8
5 CEU1021000001 Naive   1958 Mai    702.   706.   -4.30  -4.30
6 CEU1021000001 Naive   1958 Jun    708.   702.    6.40   6.40
```

Task 3: Evaluate the Model Fit of the Benchmark Model

Task 3a: Create a time plot 'Actual vs. Fitted'

```
# Time plot of actual vs. fitted values for the naive method
augment(naive_model) %>%
  autoplot(.fitted, colour = "purple", linetype = "dashed") +
  autolayer(train, Employed) +
  labs(y = "Employment",
       title = "Time plot of Actual vs. Fitted")
```

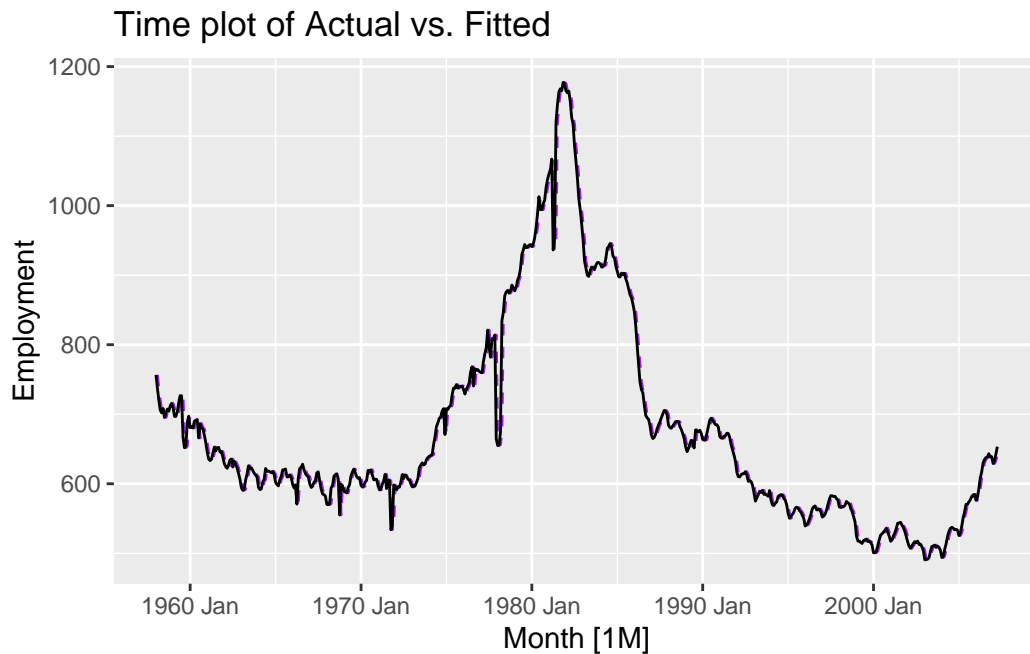


Figure 7: Time plot fitted and true values for monthly US Employment (Naive Model)

The following questions are answered based on Figure 7.

- **Are the fitted values close to the actual values?**

Yes, the naive-fitted values are close to the actual values; they capture the fluctuating pattern observed in the actual data due to their simplistic nature.

- **Are they systematically too high or too low?**

No, the naive-fitted values are not systematically too high or too low, however the fitted values are slightly shifted to the right, as the naive method sets the forecasts of the future values to the value of the last observation.

- **If yes, where does it come from? (Consider how your benchmark model works!)**

NA, as they are not systematically too high or too low.

Task 3b: Create a scatter plot 'Actual vs. Fitted'

```
# Scatter plot of actual vs. fitted values for the naive method
ggplot(data = naive_fitted_values, aes(x = .fitted, y = Employed)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = 2) +
  labs(x = "Fitted Values", y = "Actual Values") +
  ggtitle("Scatter plot of Actual vs. Fitted")
```

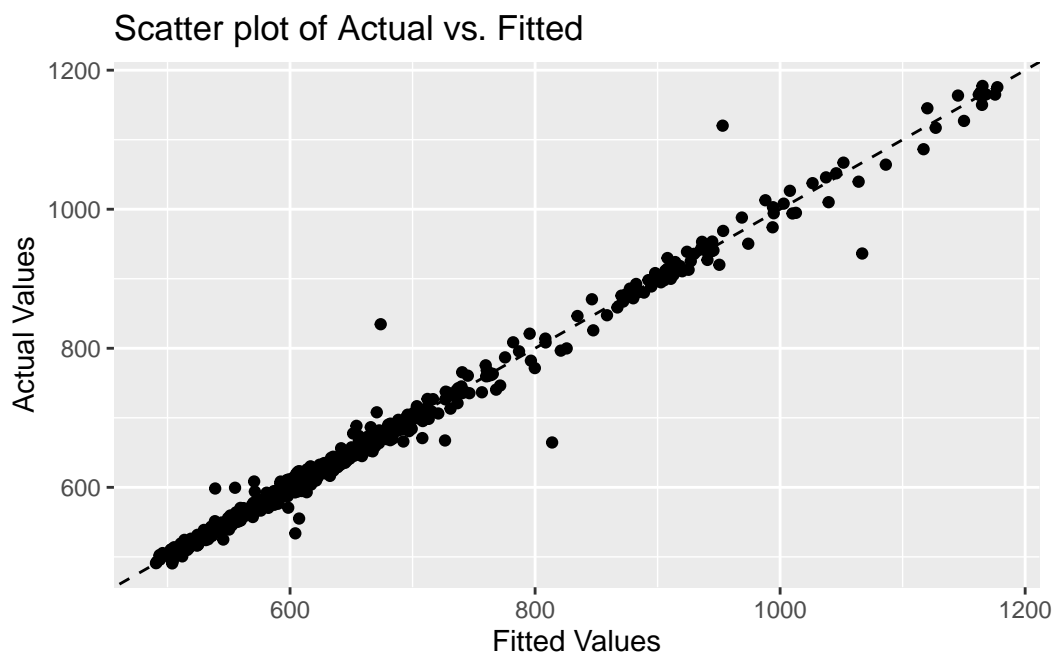


Figure 8: Scatter plot of fitted and true values for monthly US Employment with an identity line (Naive Model)

The following questions are answered based on Figure 8.

- Are the points close to the identity line?

Yes, the points are close to the identity line, with some minor outliers.

- Are they systematically too high or too low?

No, the points are not systematically too high or too low.

- If yes, where does it come from? (Relate your answer to what you saw in the time plot.)

NA, as they are not systematically too high or too low.

Task 3c: Perform residual diagnostics to inspect the model fit

```
# Residual plots of the naive method
gg_tsresiduals(naive_model) +
  labs(title = "Residual Plots of the Naive Model")
```

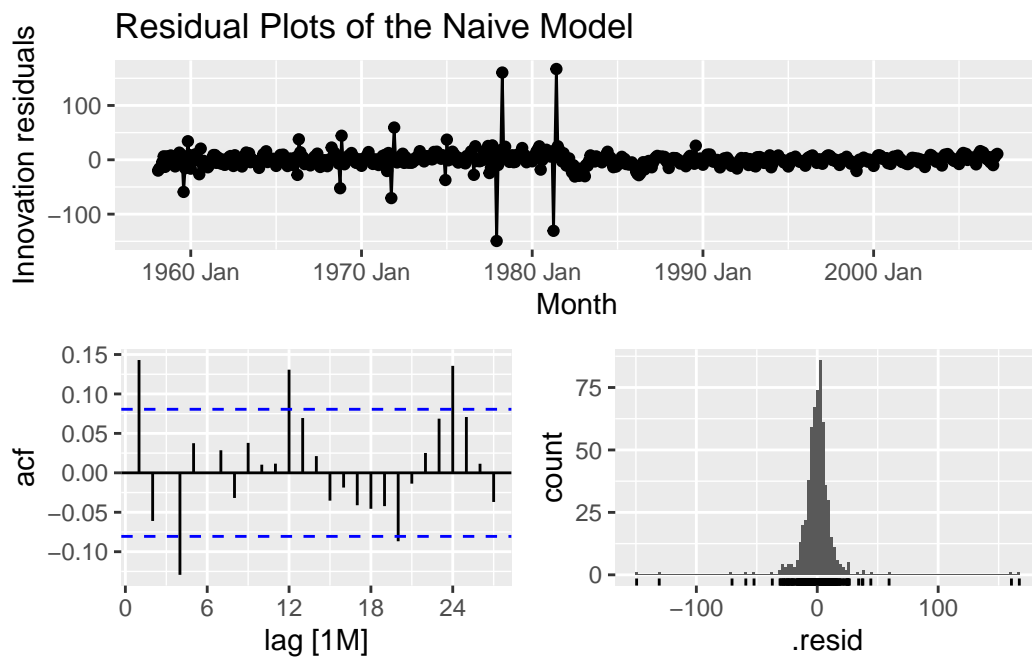


Figure 9: Residual plots of the naive model trained on monthly US Employment

```
# If no transformation has been applied the innovation residuals, column '.innov' are identical to
# the residuals, column '.resid'. Therefore the column '.innov' is used for the following histogram.

# Residual histogram with mean line for the naive model
ggplot(augment(naive_model), aes(x = .innov)) +
  geom_histogram() +
  geom_vline(aes(xintercept = mean(.innov, na.rm= TRUE)), color = "red", linetype = "dashed") +
  labs(title = "Histogram of Residuals with Mean Line (Naive Model)")
```

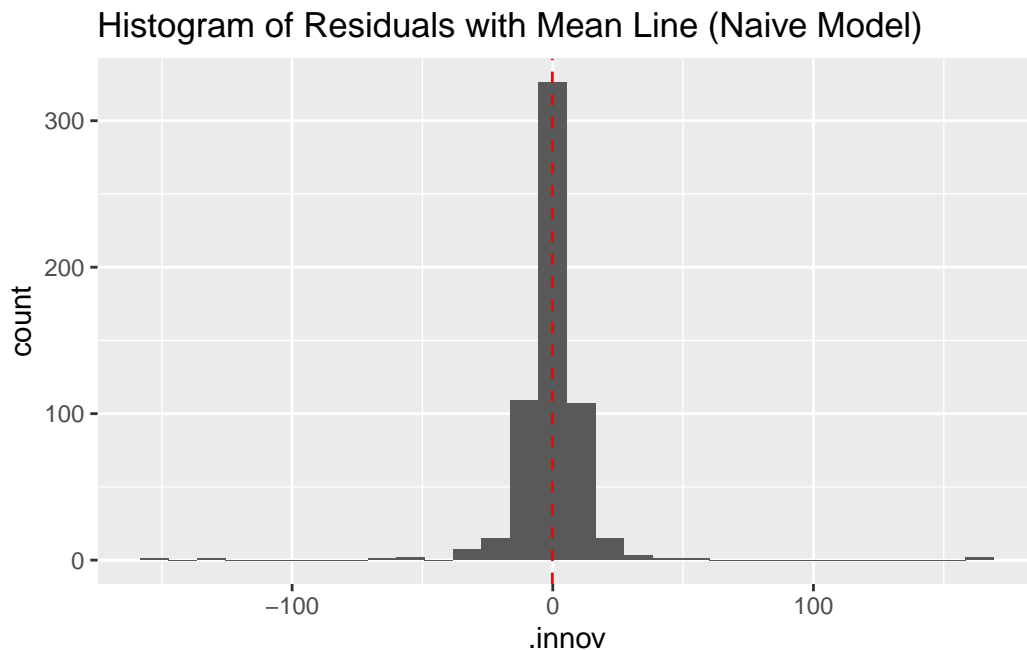


Figure 10: Histogram of residual for the naive model trained on monthly US Employment

The following questions are answered based on Figure 9.

- **Are the residuals auto-correlated? How do you decide that based on the plots?**

Yes, the residual ACF plot suggests autocorrelation, as several autocorrelation coefficients (black spikes) exceed the confidence intervals (blue dotted lines), particularly at lags 1, 4, 12, 20, and 24, where 4 spikes out of 27 are significant, totaling 14.8%, substantially exceeding the 5% threshold.

- **Do the residuals have zero mean? How do you decide that based on the plots?**

Yes, based on the histogram of residuals from Figure 9 and Figure 10, the residuals have a zero mean, or at least their mean is very close to 0.

Task 3d: Double-check your results from (c)

```
# Ljung-Box test for autocorrelation in residuals
augment(naive_model) |>
  features(.innov, ljung_box, lag = 27) # 27 lags, as acf plot shows 27 autocorrelation coefficients
```

```
# A tibble: 1 x 4
  Series_ID      .model lb_stat lb_pvalue
  <chr>         <chr>    <dbl>    <dbl>
1 CEU1021000001 Naive      68.7 0.0000174
```

- Does the test result support your conclusions from (c)? How do you conclude that from the test result?

Yes, the Ljung-Box test result supports the conclusion from (c) that there is autocorrelation in the residuals, as indicated by the very low p-value of 1.741199e-05 (=0.00001741199), suggesting statistically significant autocorrelation.

```
# Calculate the residuals
naive_residuals <- train$Employed - naive_fitted_values$.fitted
```

```
# Calculate the residual mean
mean(naive_residuals, na.rm = TRUE)
```

```
[1] -0.1747885
```

```
# or directly calculate the residual mean
mean(naive_fitted_values$.innov, na.rm = TRUE)
```

```
[1] -0.1747885
```

- Does the result support your conclusions from (c)?

The result from the residual mean supports the conclusion from (c), as the residual mean is -0.1747885, what is close to 0.

Task 4: Evaluate the Point Forecast Accuracy of the Benchmark Model

```
# Feed the trained model into the forecast() function
forecast_naive <- forecast(naive_model, h = test_length) #test_length = 149

# Plot actual vs. forecast values
forecast_naive %>% autoplot(mySeries)
```

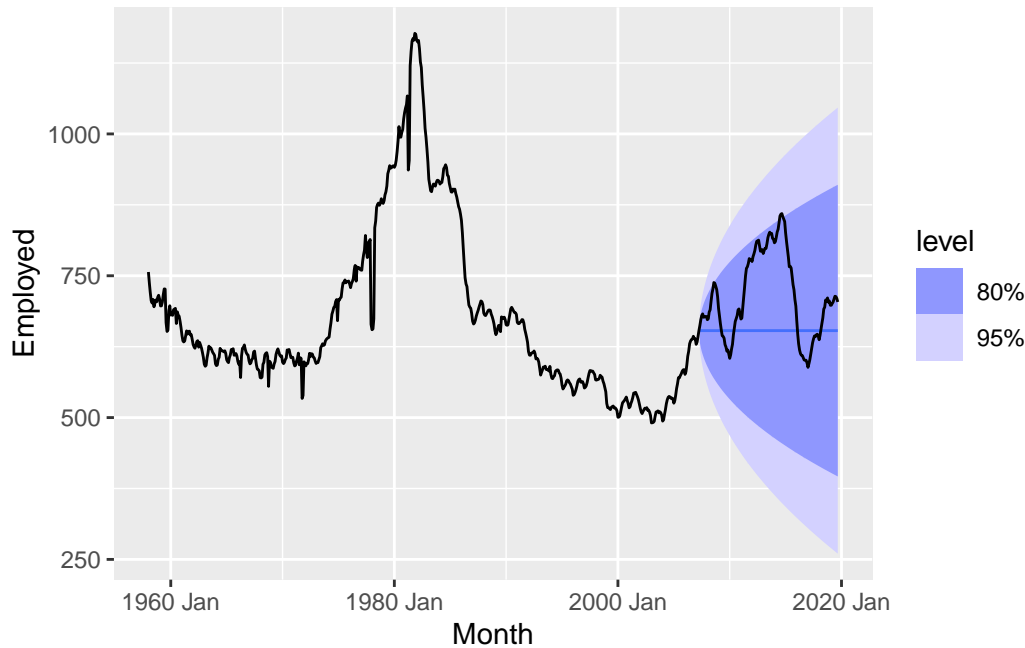


Figure 11: Time plot of Actual vs. Forecasted Values for the naive model trained on monthly US Employment

The time plot in Figure 11 shows that the data exhibits cyclic behavior and potentially some seasonality, but there is no clear trend evident.

```
# Calculate the forecast accuracy metrics
forecast_naive %>% accuracy(mySeries)
```

```
# A tibble: 1 x 11
  .model Series_ID .type ME RMSE MAE MPE MAPE MASE RMSSE ACF1
  <chr> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1 Naive CEU1021000001 Test 60.3 96.1 75.8 7.44 9.99 2.02 1.64 0.991
```

The Root Mean Squared Error (RMSE) is scale-dependent and measures the average magnitude of the errors between the actual and forecasted values. A value of 96.06 suggests that, on average, the forecasted values differ from the actual values by approximately 96.06 units. Considering the significant fluctuations in the data as the time plot in Figure 11 shows, the RMSE can be considered relatively high, indicating that the model's forecasts deviate considerably from the actual values.

The Mean Absolute Percentage Error (MAPE) measures the average percentage difference between the forecasted and actual values. A value of 9.99 indicates that, on average, the model's forecasts deviate from the actual values by approximately 9.99% of the actual values. Given the variability evident in the time plot in Figure 11 the MAPE suggests that the model's forecasts have a moderate to low level of accuracy, as the error rate is slightly less than 10%.

Task 5: Evaluate the Point Forecast Uncertainty of the Benchmark Model

```
# Residual plots of the naive method
gg_tsresiduals(naive_model) +
  labs(title = "Residual Plots of the Naive Model")
```

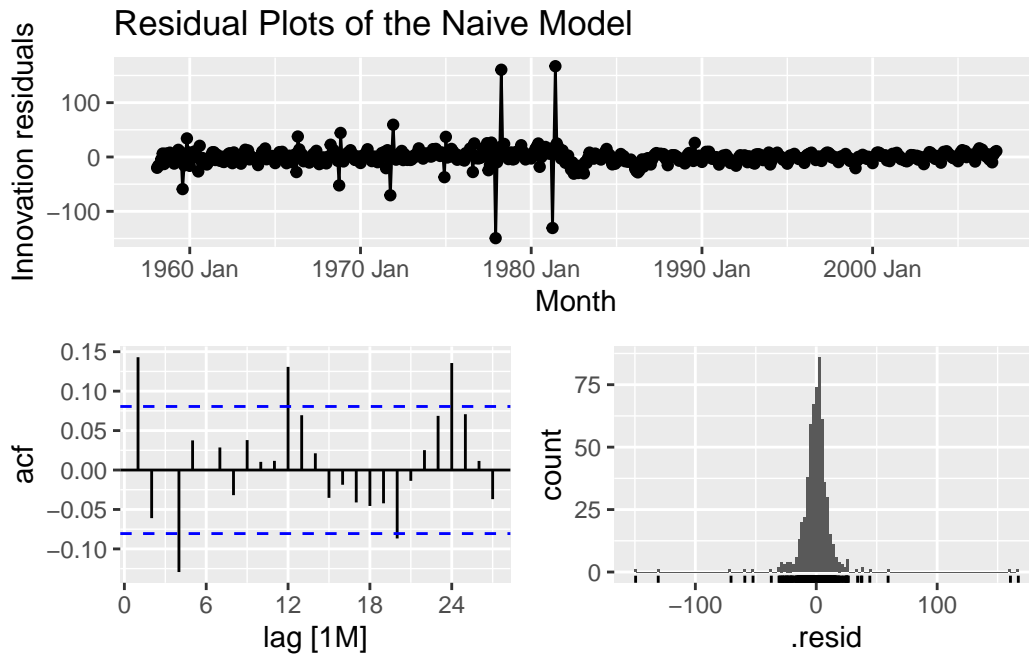


Figure 12: Residual plots of the naive model trained on monthly US Employment

The following questions are answered based on Figure 12.

- **Are the residuals homoscedastic? (That is, do they have constant variance?)**

Looking at the time plot of the residual, it seems that the variance is relatively constant, with minor fluctuations, thus the series seems to be homoscedastic.

- **Are the residuals normally distributed?**

The histogram indicates a normal distribution due to its unimodality (one peak), however, it is slightly asymmetric, and as discussed earlier the mean of -0.1747885 is nearly zero.


```
# Display the prediction intervals' boundaries
forecast_naive_intervals <- hilo(forecast_naive)

head(forecast_naive_intervals)
```

```
# A tsibble: 6 x 7 [1M]
# Key:      Series_ID, .model [1]
  Series_ID      .model   Month   Employed .mean      `80%`
  <chr>         <chr>    <mth>    <dist> <dbl>      <hilo>
1 CEU1021000001 Naive   2007 Mai   N(653, 270) 653. [632.2425, 674.3575] 80
2 CEU1021000001 Naive   2007 Jun   N(653, 540) 653. [623.5202, 683.0798] 80
3 CEU1021000001 Naive   2007 Jul   N(653, 810) 653. [616.8273, 689.7727] 80
4 CEU1021000001 Naive   2007 Aug   N(653, 1080) 653. [611.1850, 695.4150] 80
5 CEU1021000001 Naive   2007 Sep   N(653, 1350) 653. [606.2140, 700.3860] 80
6 CEU1021000001 Naive   2007 Okt   N(653, 1620) 653. [601.7198, 704.8802] 80
# i 1 more variable: `95%` <hilo>
```

- Does it make sense to include these prediction intervals in your model evaluation? Why/why not?

Yes, it makes sense to include these prediction intervals, in the model evaluation, as they provide valuable information about the growing uncertainty associated with the predictions and thus help to understand the range of possible outcomes.

Task 7: Compare the Evaluation Results of the Two Models: Mean vs. Naive

Both models demonstrate notable deviations from actual values, as indicated by their relatively high RMSE values. While both models exhibit moderate accuracy levels, the mean model's lower RMSE and MAPE suggest slightly better performance compared to the naive model. However, it is important to note that both models do not offer optimal forecasting accuracy, and there may be other models better suited to capture the underlying patterns in the data. Nonetheless, considering the available options, the mean model is preferred for its comparatively lower error rates.