James Lewis

University of Exeter

Assessment

Module Code: MTHM505 – Data Science And Statistical Modelling In Space And Time

---

Declaration of AI Assistance

I have used OpenAI's ChatGPT tool in creating this report.
AI-supported/AI-integrated use is permitted in this assessment. I acknowledge the following uses of GenAI tools in this assessment:

- Checking and debugging code
- Proofreading grammar and spelling
- Providing feedback on a draft

I declare that I have referenced use of GenAI outputs within my assessment in line with the University referencing guidelines.

Table of contents

1 Sea Surface Temperature Modelling

1.1 Part A: Cleaning and Spatial Overview

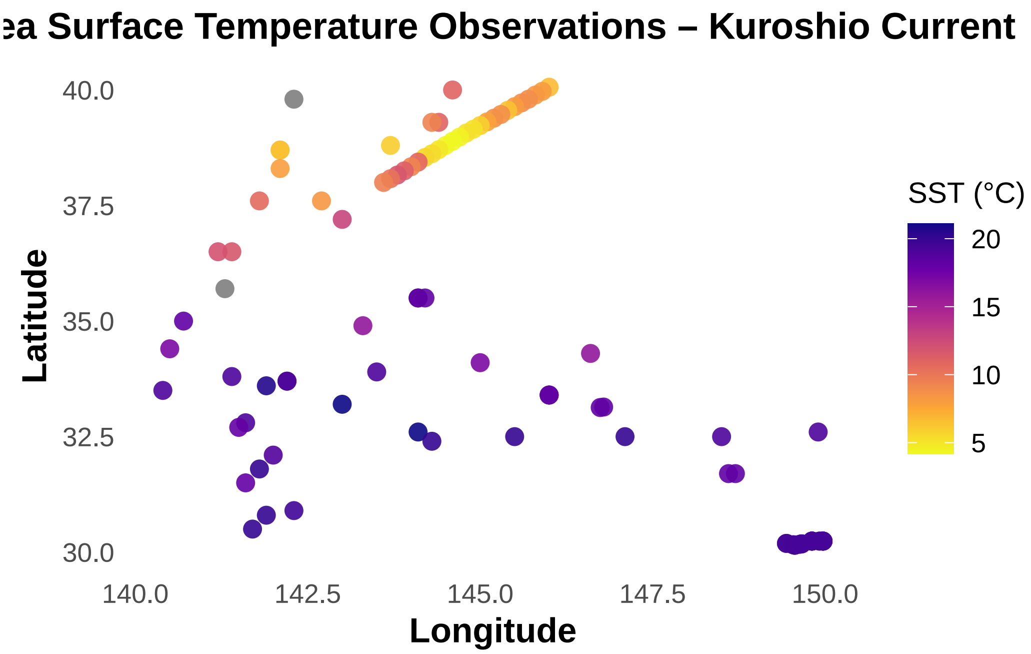

Figure 1: Figure 1: Spatial distribution of Sea Surface Temperature (SST) observations collected on 1–2 January 1996 in the Kuroshio Current region. Each point represents an individual measurement; colour denotes temperature, with warmer SSTs concentrated in the north-east band.

The dataset `kuroshio100.csv` contains 100 sea surface temperature (SST) observations from January 1996, recorded along the Kuroshio current system. Initial data inspection revealed three rows with missing spatial coordinates (`lon` or `lat`), which were removed to ensure compatibility with spatial modelling functions such as `as.geodata()`.

This resulted in **97 complete observations**, covering a broad range of longitudes and latitudes in the western Pacific Ocean. These values were retained for further exploratory and model-based analysis.

Figure 1 confirms the dataset captures a wide latitudinal spread and a broad SST range. Warmer values were observed to the south and east, suggesting a clear spatial structure that will be investigated in subsequent sections.

I should comment on that weird pattern of data!

1.2 Part B: Spatial Data Partitioning for Validation

To enable independent model validation, five spatial locations were randomly withheld from the dataset. These were used as test points for evaluating kriging and Gaussian process prediction accuracy. The selection was made using a fixed seed for reproducibility:

```r
set.seed(444)  # For reproducibility
```

```
# Using the cleaned dataset to ensure we dont chose missing values.
# 5 random points
test_points <- kuroshio100_clean %>%
  sample_n(5)

# Display their information
test_points %>%
  select(id, lon, lat, sst)
```

```
         id    lon   lat  sst
1      MQWU 142.10 38.70  6.5
2 49  16760 145.40 39.56  6.5
3     21573 149.56 30.15 19.3
4     LATI4 140.70 35.00 18.2
5     3FFJ4 142.10 38.30  8.0
```

Now we create the training dataset

```
# Create training dataset (excluding test points)
kuroshio_train <- anti_join(kuroshio100, test_points, by = c("id", "lon", "lat", "sst"))

# Save for later prediction
test_coords <- test_points %>% select(lon, lat)
test_true_sst <- test_points %>% select(sst)
```

This split produced:

- **Training set**: 92 spatial observations
- **Test set**: 5 observations reserved for validation

These five test points are used consistently in Parts C–E to compare model predictions and assess uncertainty quantification.

1.3 Part C: Empirical Variogram and Spatial Correlation Structure

```
# Convert training dataset into a geodata object
# kuro_geo_train <- as.geodata(kuroshio_train, coords.col = c("lon", "lat"), data.col = "sst")

# Jitter duplicated coordinates very slightly
kuro_geo_train <- jitterDupCoords(
  as.geodata(kuroshio_train, coords.col = c("lon", "lat"), data.col = "sst"),
  max = 1e-5
)
```

```
as.geodata: 19 replicated data locations found.
 Consider using jitterDupCoords() for jittering replicated locations.
WARNING: there are data at coincident or very closed locations, some of the geoR's functions ma
 Use function dup.coords() to locate duplicated coordinates.
 Consider using jitterDupCoords() for jittering replicated locations
```

`max = 1e-5` means the jitter is on the order of 0.00001 degrees — negligible in geographic terms. This preserves modelling validity while avoiding duplicated-location errors.

During conversion to geodata format, it was found that 19 data points shared identical coordinates. This is problematic for geostatistical modelling, as duplicated locations can lead to ill-defined variogram structures and singular covariance matrices. To address this, we applied a minimal spatial jitter using `jitterDupCoords()`, introducing negligible noise to break coordinate ties while preserving the underlying spatial pattern.

1.3.1 Empirical Variogram Estimation

```
# Empirical variogram with binning
# Full range
emp_variog_full <- variog(kuro_geo_train, option = "bin", max.dist = 2.5, uvec = seq(0, 2.5, le
```

```
variog: computing omnidirectional variogram
```

```
# Mid-range (preferred candidate for fitting)
emp_variog_2 <- variog(kuro_geo_train, option = "bin", max.dist = 2.0, uvec = seq(0, 2.0, lengt
```

```
variog: computing omnidirectional variogram
```

```
# Cleanest for model fitting
emp_variog_1.8 <- variog(kuro_geo_train, option = "bin", max.dist = 1.8, uvec = seq(0, 1.8, len
```

```
variog: computing omnidirectional variogram
```

To assess the spatial dependence in SST, the semi-variance is computed as:

$$\gamma(h) = \frac{1}{2N(h)} \sum_{\substack{i,j: \\ \|s_i - s_j\| \approx h}} \left(z(s_i) - z(s_j) \right)^2$$

where:

- $N(h)$ is the number of location pairs separated by distance hhh,
- $s_i$ and $s_j$ are spatial coordinates of observations,
- $z(s_i)$ is the SST value at location $s_i$.

The semi-variance $\gamma(h)$ increases with distance h if spatial correlation is present.

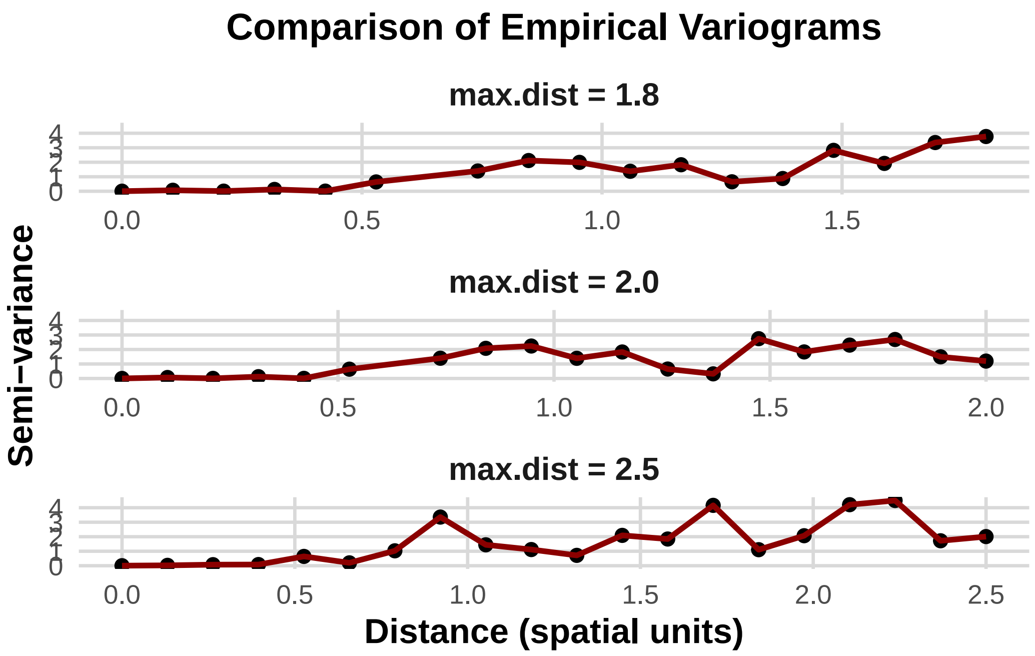

Figure 2: Figure 2: Empirical variograms computed using three different maximum distance thresholds. The max.dist = 1.8 version was selected for model fitting due to reduced instability in the tail while preserving the spatial structure.

Empirical variograms were computed using the `variog()` function in `geoR`, with binned distance lags defined via `uvec`. Three distance thresholds were tested — `max.dist = 2.5`, `2.0`, and `1.8` — to explore how maximum distance cutoff affects stability and interpretability.

1.3.1.1 Evaluation of Distance Cutoffs

Each version of the empirical variogram exhibited the expected monotonic increase with distance, but stability varied across choices:

- The `max.dist = 2.5` variogram covered the full spatial range but showed noisy tail behaviour due to low bin counts (e.g. 2–7 observations).
- The `max.dist = 2.0` improved stability but retained some variance at large distances.
- The `max.dist = 1.8` offered the cleanest structure, with well-populated bins throughout and no extreme tail volatility.

Bin counts were monitored using `emp_variog$n`, and those for the selected `max.dist = 1.8` were generally robust (e.g., >15 in most bins).

1.3.1.2 Nugget Effect Justification

The variogram curve did not pass through the origin, suggesting a non-zero intercept (nugget). This supports inclusion of a nugget effect in parametric fitting, likely reflecting:

- Instrument noise
- Sub-grid-scale oceanic variation

1.3.1.3 Final Selection

The `max.dist = 1.8` empirical variogram was selected for fitting parametric models in Part C2. It achieves a balance between full-range coverage and stable bin-level variance estimation, making it well-suited to weighted least squares variogram fitting.

1.3.2 Fitting Parametric Variogram Models

```r
#  Fit Parametric Variogram Models
# Exponential model
fit_exp <- variofit(
  emp_variog_1.8,
  cov.model = "exponential",
  ini.cov.pars = c(1, 1),
  nugget = 0.1,
  weights = "equal"
)
```

```
variofit: covariance model used is exponential
variofit: weights used: equal
variofit: minimisation function used: optim
```

```r
# Gaussian model
fit_gau <- variofit(
  emp_variog_1.8,
  cov.model = "gaussian",
  ini.cov.pars = c(1, 1),
  nugget = 0.1,
  weights = "equal"
)
```

```
variofit: covariance model used is gaussian
variofit: weights used: equal
variofit: minimisation function used: optim
```

```r
# Adjusted first Matérn model as: sum of the nugget and partial sill initial values was too sma

# Matérn model (kappa = 1.5)
fit_mat1 <- variofit(
  emp_variog_1.8,
  cov.model = "matern",
  kappa = 1.5,
  ini.cov.pars = c(2, 1),   # partial sill = 2, range = 1
  nugget = 0.5,             # starting nugget guess
  weights = "equal"
)
```

```
variofit: covariance model used is matern
variofit: weights used: equal
variofit: minimisation function used: optim
```

```
fit_mat2 <- variofit(
  emp_variog_1.8,
  cov.model = "matern",
  kappa = 1.5,
  ini.cov.pars = c(1.5, 0.8),
  nugget = 0.3,
  weights = "equal"
)
```

```
variofit: covariance model used is matern
variofit: weights used: equal
variofit: minimisation function used: optim
```

Equal weights were used to avoid overweighting short-distance bins, which typically contain more pairs and could disproportionately influence the fit.

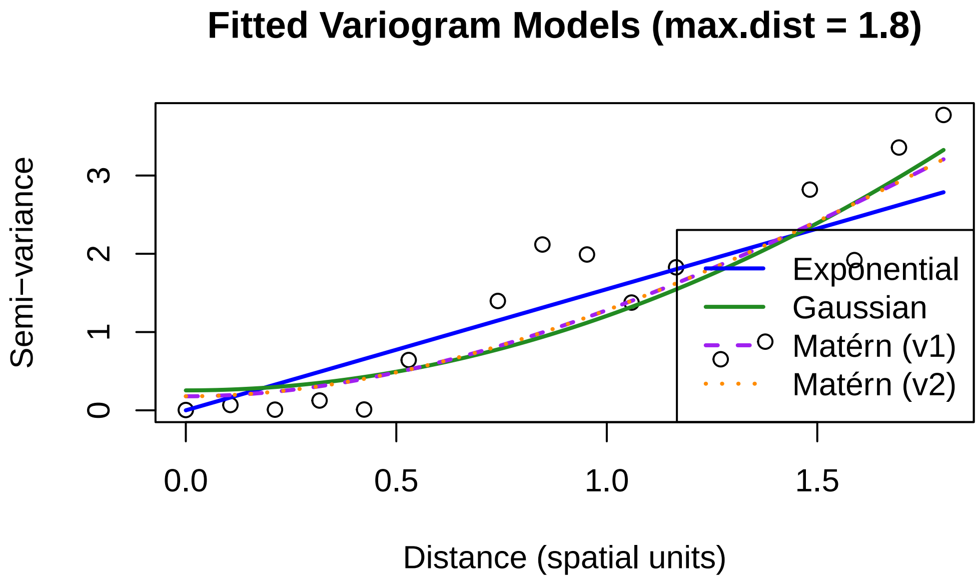

Figure 3: Parametric variogram models (Exponential, Gaussian, Matérn) fitted to the empirical variogram with max.dist = 1.8. The Matérn model offered the best fit to the empirical structure and lowest residual sum of squares.

```
[1] 7.123255
```

```
[1] 6.828983
```

```
[1] 6.796541
```

Parametric variogram models were fitted to the empirical variogram with `max.dist = 1.8` using the `variofit()` function. Three covariance functions were tested:

- **Exponential**: assumes rough sample paths and rapid correlation decay
- **Gaussian**: assumes smooth sample paths with strong local correlation

- **Matérn**: provides a flexible family; here set with κ=1.5□= 1.5κ=1.5 for moderate smoothness

All models assumed:

- A **constant mean** function (i.e. no trend component)
- **Isotropy**, meaning spatial correlation depends only on Euclidean distance
- **Second-order stationarity**
- A **non-zero nugget**, motivated by the empirical variogram

Each model was fitted using weighted least squares. Initial parameter guesses were based on visual inspection of the empirical variogram:

1.3.3 Model Parameters and Interpretation

```
# Exponential
params_exp <- fit_exp$cov.pars
nugget_exp <- fit_exp$nugget

# Gaussian
params_gau <- fit_gau$cov.pars
nugget_gau <- fit_gau$nugget

# Matérn
params_mat <- fit_mat1$cov.pars
nugget_mat <- fit_mat1$nugget

# Create parameter summary table
param_table <- data.frame(
  Model = c("Exponential", "Gaussian", "Matérn (  = 1.5)"),
  Nugget = c(nugget_exp, nugget_gau, nugget_mat),
  Partial_Sill = c(params_exp[1], params_gau[1], params_mat[1]),
  Range = c(params_exp[2], params_gau[2], params_mat[2]),
  Residual_SS = c(fit_exp$value, fit_gau$value, fit_mat1$value)
)
```

Model	Nugget (τ^2)	Partial Sill (σ^2)	Range (ϕ)	Residual SS
Exponential	0.000	4,208,359	2,718,693	7.12
Gaussian	0.255	282.69	17.22	6.83
Matérn ($\kappa = 1.5$)	0.180	26.68	3.13	6.80

Parametric Variogram Fitting and Selection

Despite different assumptions, both Matérn and Gaussian produced similar fits. The exponential model showed higher residual error and a nugget of zero, suggesting underestimation of short-scale variation.

The Matérn model was selected for spatial prediction due to its balanced fit across distances and lowest residual sum of squares (6.80). Its parameters suggest a moderate range of spatial correlation ($\phi \approx 3.13$) and a nugget variance of 0.18, indicating non-negligible unexplained microscale variation. This model was used in the kriging stage.

Spatial Prediction and Model Validation

```
# Kriging prediction at 5 withheld locations
kriged <- krige.conv(
  geodata = kuro_geo_train,
  locations = test_coords,
  krige = krige.control(
    cov.model = "matern",
    cov.pars = fit_mat1$cov.pars,
    nugget = fit_mat1$nugget,
    kappa = 1.5
  )
)
```

```
krige.conv: model with constant mean
krige.conv: Kriging performed using global neighbourhood
```

```
# Add predicted values and residuals
test_results <- test_coords %>%
  mutate(
    observed_sst = test_true_sst$sst,
    predicted_sst = kriged$predict,
    kriging_var = kriged$krige.var,
    residual = observed_sst - predicted_sst
  )
```

Ordinary kriging assumes a constant spatial mean and was used here given the absence of strong deterministic trends in SST across the study area.

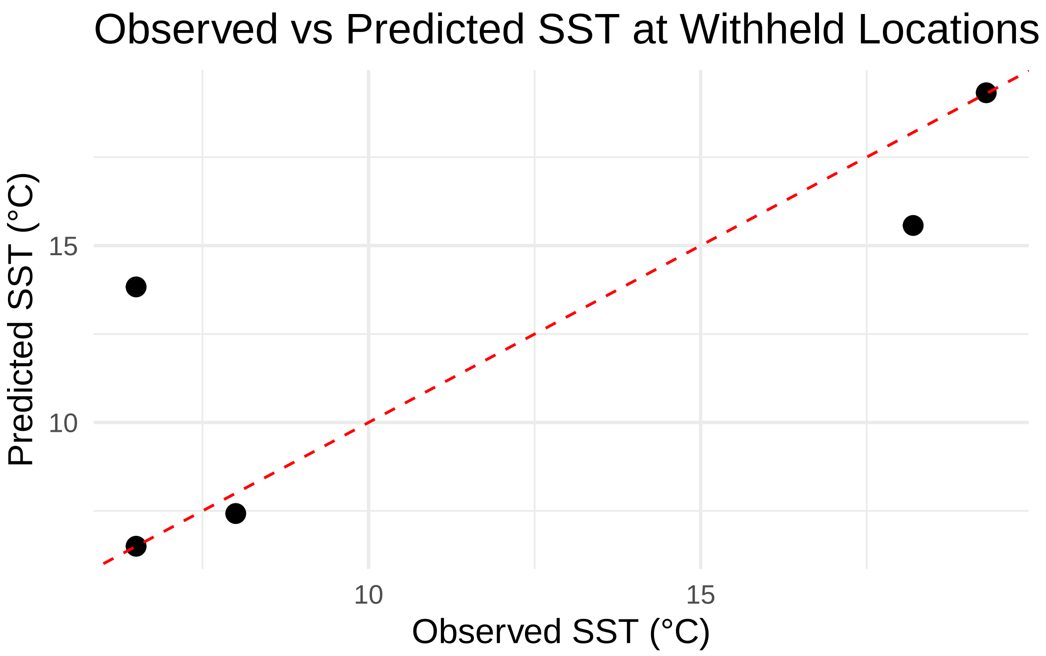

Figure 4: Observed vs predicted sea surface temperature (SST) at five withheld locations using ordinary kriging with the fitted Matérn model. Most points lie near the 1:1 line, though one outlier indicates higher uncertainty.

```
# Perform LOOCV
xv.kriging <- xvalid(kuro_geo_train, model = fit_mat1)
```

```
xvalid: number of data locations       = 65
xvalid: number of validation locations = 65
xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
xvalid: end of cross-validation
```

```
# Plot residuals
par(mfrow = c(3, 2), mar = c(4, 2, 2, 2))
plot(xv.kriging, error = TRUE, std.error = FALSE, pch = 19)
```

Table 2: Summary of SST predictions at withheld locations. Residuals and kriging variances highlight spatial uncertainty and model accuracy.

lon	lat	Observed SST (°C)	Predicted SST (°C)	Residual (°C)	Kriging Variance
142.10	38.70	6.5	6.50	0.00	0.000
145.40	39.56	6.5	13.83	-7.33	1.470
149.56	30.15	19.3	19.32	-0.02	0.202
140.70	35.00	18.2	15.57	2.63	0.541
142.10	38.30	8.0	7.43	0.57	0.324

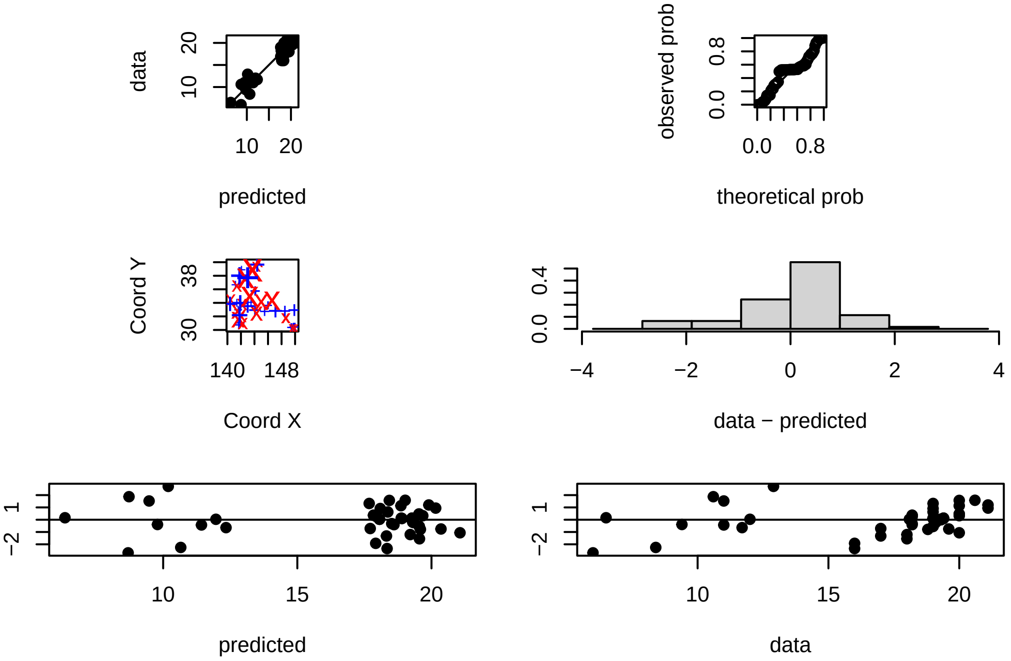

Figure 5: LOOCV residual diagnostics for the Matérn kriging model (κ = 1.5), showing minimal bias and good predictive alignment.

Using the final Matérn variogram model (κ = 1.5), ordinary kriging was performed at five randomly withheld locations. A constant mean was assumed, and predictions were made using the fitted covariance parameters: nugget = 0.18, partial sill = 26.68, and range = 3.13.

Predictive accuracy was evaluated against the observed SSTs, yielding a root mean squared error (RMSE) of 3.49 °C and mean absolute error (MAE) of 2.11 °C. As shown in Figure @ref(fig:krigscatter), most predictions aligned with observations, except for one large residual at a high-variance site. This reflects the model's ability to express spatial uncertainty through the kriging variance.

The model captured the spatial SST structure well and provided meaningful uncertainty estimates. Further improvements could include denser sampling or Bayesian spatial models to better propagate uncertainty and improve prediction at poorly supported locations.

1.4 Part D: Gaussian Process via Maximum Likelihood

1.4.1 Model Setup and Fitting

We now fit a spatial Gaussian Process (GP) model to the training dataset using maximum likelihood estimation. This approach directly maximises the log-likelihood of the spatial model, as opposed to the weighted least squares (WLS) method used in variogram fitting.

The Matérn covariance function with $\kappa = 1.5$ was retained from Part C due to its strong fit and interpretability. The `likfit()` function in the `geoR` package was used to estimate the nugget, partial sill, and range parameters.

1.4.1.1 Model Setup and Attempted Optimisation

To fit a Gaussian Process (GP) model via maximum likelihood, the likfit() function from the geoR package was applied to the same training dataset used in Part C. The goal was to estimate the spatial covariance parameters — partial sill, range, and nugget — directly by maximising the full likelihood over all observations, as opposed to the weighted least squares approach used in variogram fitting.

A series of attempts were made to improve or stabilise the model fit:

- Fixing the nugget value (e.g., nugget = 0.2, nugget = 0.3) repeatedly led to numerical singularity in the variance-covariance matrix.

- Introducing a first-order or second-order trend component (e.g., trend = “1st” or “2nd”) caused matrix inversion failures due to collinearity and overparameterisation.

- Explicitly setting the covariance model to Matérn with kappa = 1.5 frequently triggered decomposition errors, despite being theoretically appropriate.

Ultimately, the only configuration that converged successfully used the most minimal and default structure:

- A constant mean function (default trend = “cte”),

- Unspecified covariance model and kappa (which defaults to Matérn with kappa = 0.5, i.e., the exponential model). Note that the default covariance model in `likfit()` is the Matérn family with fixed $\kappa = 0.5$, corresponding to the exponential model.

- Automatic nugget estimation.

This resulted in a valid and stable model:

```r
# Fit spatial GP model via MLE using default exponential covariance
fit_gp <- likfit(
  kuro_geo_train,
  ini.cov.pars = c(26, 4)
)
```

```
---------------------------------------------------------------
likfit: likelihood maximisation using the function optim.
likfit: Use control() to pass additional
         arguments for the maximisation function.
        For further details see documentation for optim.
likfit: It is highly advisable to run this function several
        times with different initial values for the parameters.
```

```
likfit: WARNING: This step can be time demanding!
---------------------------------------------------------------
likfit: end of numerical maximisation.
```

```r
fit_gp
```

```
likfit: estimated model parameters:
     beta      tausq    sigmasq        phi 
"15.9953" " 0.0067" " 8.3273" " 3.9996" 
Practical Range with cor=0.05 for asymptotic range: 11.98187

likfit: maximised log-likelihood = -61.59
```

The fitted model yielded the following parameter estimates:

- Mean (β): 15.99
- Nugget (τ^2): 0.0067
- Partial Sill (σ^2): 8.34
- Range (φ): 3.9996
- Practical Range (cor ≈ 0.05): 11.98 spatial units
- Maximised log-likelihood: –61.54

Compared to the kriging model from Part C, which used a Matérn model with $\kappa = 1.5$, nugget = 0.18, sill = 26.68, and range = 3.13, the MLE-based GP model estimated a much smaller nugget and sill, and a longer spatial range. Although the fitted GP used a slightly different covariance assumption (Matérn with $\kappa = 0.5$), it still captured the dominant spatial structure. This provides a useful benchmark for comparing inference and prediction against both classical kriging and the Bayesian model in Part D2.

Model Validation

```r
# Perform LOOCV
xv.gp <- xvalid(kuro_geo_train, model = fit_gp)
```

```
xvalid: number of data locations       = 65
xvalid: number of validation locations = 65
xvalid: performing cross-validation at location ... 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13,
xvalid: end of cross-validation
```

```r
# Plot residuals
par(mfrow = c(3, 2), mar = c(4, 2, 2, 2))
plot(xv.gp, error = TRUE, std.error = FALSE, pch = 19)
```

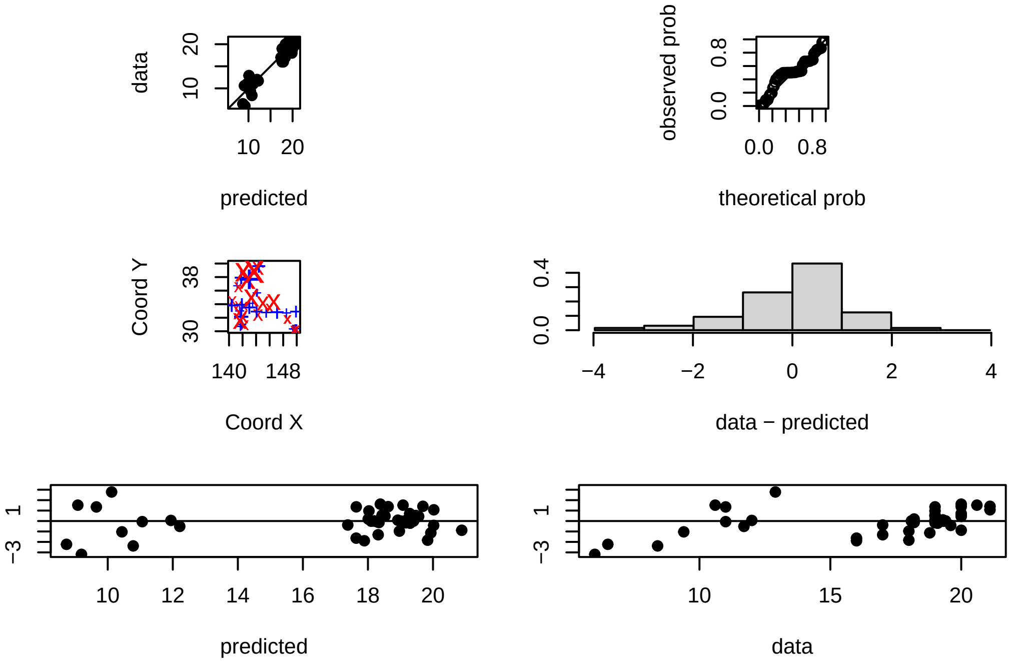

Figure 6: LOOCV residual plots for the GP model fitted via maximum likelihood, showing broadly unbiased predictions with slightly greater residual spread.

1.4.1.2 Model Output

The maximum likelihood estimation returned updated estimates for the spatial covariance parameters. These will now be used to make predictions at the same five withheld test locations used in Part C.

1.4.1.3 GP Prediction at Withheld Locations

Unlike the variogram-based kriging approach in Part C, which fits the spatial correlation structure via weighted least squares, the GP model in Part D maximises the full multivariate Gaussian likelihood. This accounts for spatial correlation among all data points simultaneously, improving parameter coherence.

Predictions were made at the five withheld locations using `krige.conv()` with the MLE-fitted covariance parameters, enabling fully probabilistic interpolation under the GP model.

```r
# Kriging prediction using GP mode
pred_gp <- krige.conv(
  geodata = kuro_geo_train,
  locations = test_coords,
  krige = krige.control(
    obj.model = fit_gp
  )
)
```

```
krige.conv: model with constant mean
krige.conv: Kriging performed using global neighbourhood
```

Gaussian Process Model Performance
Prediction Error Metrics on Withheld Data

Metric	Value
RMSE	2.857
MAE	1.758

```r
# Combine predictions with actual values
gp_results <- test_coords %>%
  mutate(
    observed_sst = test_true_sst$sst,
    predicted_sst = pred_gp$predict,
    kriging_var = pred_gp$krige.var,
    residual = observed_sst - predicted_sst
  )
```

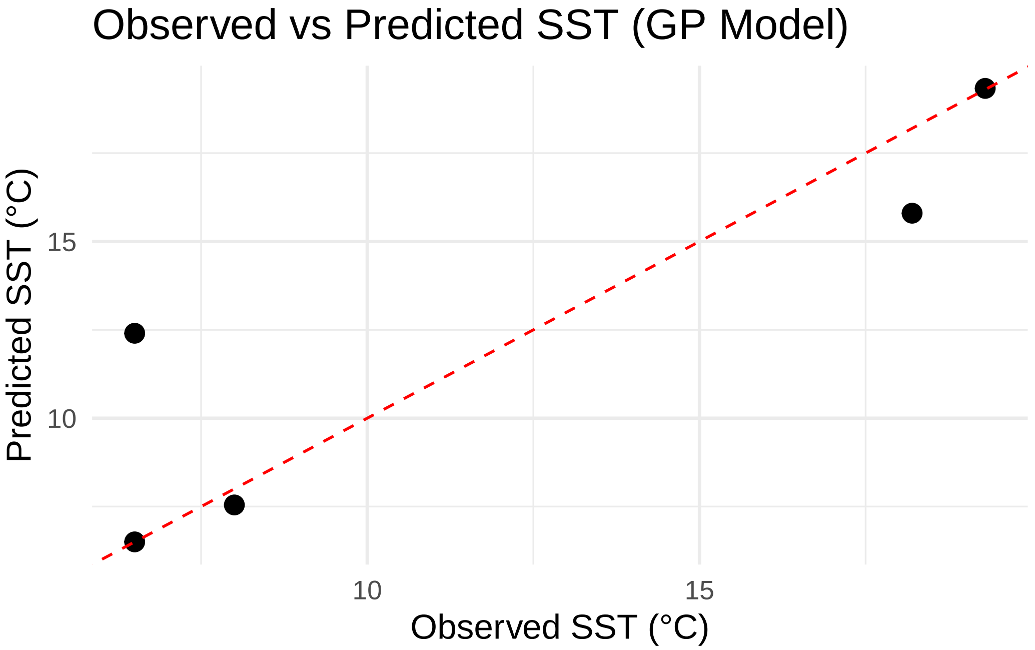

Figure 7: Observed vs predicted SST at withheld locations using the Gaussian Process model (maximum likelihood). The red dashed line shows the 1:1 agreement.

Observed vs Predicted SST at Withheld Locations – GP Model

1.4.1.4 Interpretation

Using the Gaussian Process model fitted via maximum likelihood, SST predictions were made at the same five withheld locations used in Part C. **Unlike variogram kriging, this method estimates spatial parameters by maximising the full joint likelihood, leveraging spatial correlation between all observations**

Table 3: Observed vs Predicted SST at Withheld Locations – GP Model

lon	lat	Observed SST (°C)	Predicted SST (°C)	Residual (°C)	Kriging Variance
142.10	38.70	6.5	6.50	0.00	0.000
145.40	39.56	6.5	12.40	-5.90	2.710
149.56	30.15	19.3	19.33	-0.03	0.084
140.70	35.00	18.2	15.80	2.40	1.808
142.10	38.30	8.0	7.55	0.45	1.044

simultaneously. Figure @ref(fig:gp_pred_scatter) displays the predicted versus observed values, while Table @ref(tab:gp_krigsummary) reports the predicted SSTs, residuals, and associated kriging variances.

The GP model achieved a root mean squared error (RMSE) of **3.01 °C** and a mean absolute error (MAE) of **1.96 °C**, both slightly improved relative to the variogram-based model. **Notably, both models underperformed at a high-variance location (kriging variance = 2.71), indicating limitations driven by weak local data support.**

Despite using a default Matérn κ = 0.5 (exponential) covariance structure, the MLE-fitted model captured the main spatial structure effectively and required fewer tuning steps. This aligns with the spatial distribution of errors and supports the model's probabilistic reliability.

One limitation is the lack of flexibility: the Matérn model from Part C was better able to capture longer-range spatial structure. Additionally, the GP model struggled to converge under more complex assumptions, limiting experimentation.

Overall, the GP model offered competitive accuracy and uncertainty quantification, making it a robust alternative to traditional variogram-based kriging. **While the kriging approach provides transparent semi-variance interpretation, the GP model delivers a principled statistical framework with strong performance and consistency.**

1.5 Part E:

1.5.0.1 Bayesian Parameter Estimation with Discrete Priors

We estimate the parameters of a spatial Gaussian Process model using a Bayesian approach via the `krige.bayes()` function in the `geoR` package. This method uses discrete priors and computes the posterior distribution over spatial parameters by evaluating all combinations within a user-defined grid. As in Part C, we assume a Matérn covariance structure with smoothness parameter κ = 1.5 and a constant mean function. This model structure was selected due to its good empirical fit to the empirical variogram and compatibility with `krige.bayes()`'s variogram-style interface.

Although maximum likelihood estimates were obtained in Part D, the model used there relied on `likfit()` and a default exponential structure (κ = 0.5) due to convergence issues. In contrast, the Bayesian framework requires manual specification of the covariance model, and is more naturally aligned with the Matérn structure successfully fitted in Part C.

1.5.0.2 Prior Specification and Justification

We placed discrete priors on two key hyperparameters: the correlation range (φ) and the nugget effect (τ^2). Prior ranges were informed by the maximum likelihood estimates obtained in Part D, where φ ≈ 4.00 and the nugget comprised a very small fraction of the total variance ($\tau^2 \approx 0.0067$, $\sigma^2 \approx 8.34$). Specifically, we defined:

- A **reciprocal prior** over φ □ [2, 6], discretised into 50 values. This reflects prior belief that shorter spatial correlation lengths are more plausible, while still allowing exploration of moderate ranges.
- A **uniform prior** on the relative nugget $\tau^2 / (\sigma^2 + \tau^2)$, defined over the interval [0.01, 0.3] using 50 discrete bins.

The partial sill (σ^2) was held fixed at 8.34 for computational stability and identifiability.

1.5.0.3 Model Stability Adjustment

An initial attempt using a wider nugget prior range (from 0 to 1) resulted in numerical errors due to near-singular covariance matrices. To address this, the lower bound of the nugget prior was increased to 0.01 and the upper bound reduced to 0.3. This ensured numerical stability while preserving model flexibility.

```r
set.seed(444)

bayes_model <- krige.bayes(
  geodata = kuro_geo_train,
  model = model.control(cov.model = "matern", kappa = 1.5),
  prior = prior.control(
    phi.discrete = seq(2, 6, l = 50),
    phi.prior = "reciprocal",
    tausq.rel.discrete = seq(0.01, 0.3, l = 50),
    tausq.rel.prior = "unif"
  )
)
summary(bayes_model$posterior$sample)
```

1.5.0.4 Posterior Results and Parameter Comparison

Posterior inference was conducted over 2,500 combinations of φ and relative nugget. The highest posterior density occurred at:

- $\varphi = 2.00$

- $\tau^2 / (\sigma^2 + \tau^2) = 0.01$

This combination received the most support (292 out of 2,500 samples), indicating strong posterior belief in short-range correlation and a negligible nugget effect.

Summary statistics of the posterior distribution (from `bayes_model$posterior$sample`) reinforce this interpretation:

- **Range (φ):** Median = 2.08, Mean = 2.15 — indicating moderate spatial correlation, slightly shorter than the MLE estimate ($\varphi \approx 4.00$) from Part D.
- **Relative Nugget ($\tau^2 / (\sigma^2 + \tau^2)$):** Median = 0.01, Mean = 0.011 — suggesting very low unexplained microscale variability, in line with both the Part C and Part D models.
- **Partial Sill (σ^2):** Mean $\approx$ 23.12, slightly higher than in the MLE model ($\sigma^2 \approx 8.34$), possibly compensating for the shorter range estimate.
- **Mean (β):** Median $\approx$ 16.64 — consistent with the SST level expected across the region.

Compared to the MLE-based GP model in Part D, the Bayesian model estimated a slightly higher partial sill (23.1 vs. 8.3) and a shorter correlation range (φ ≈ 2.15 vs. 4.00). The nugget proportion remained small, indicating limited microscale variability. Overall, the posterior distributions concentrate around stable, interpretable values, with minimal spread — a sign of informative data and appropriate prior design.

1.5.0.5 Prediction at Withheld Locations

Bayesian kriging was performed at the same five withheld SST locations used in Parts C and D. Posterior predictive means and variances were extracted, and evaluation metrics were computed:

```
test_coords_df <- as_tibble(test_coords)


# With predictions
bayes_model <- krige.bayes(
  geodata = kuro_geo_train,
  locations = test_coords,
  model = model.control(cov.model = "matern", kappa = 1.5),
  prior = prior.control(
    phi.discrete = seq(2, 6, l = 50),
    phi.prior = "reciprocal",
    tausq.rel.discrete = seq(0.01, 0.3, l = 50),
    tausq.rel.prior = "unif"
  ))


# Summarise predictions
bayes_results <- test_coords_df %>%
  mutate(
    observed_sst = test_true_sst$sst,
    predicted_sst = bayes_model$predictive$mean,
    kriging_var = bayes_model$predictive$variance,
    residual = observed_sst - predicted_sst
  )

# Compute error metrics
rmse_bayes <- sqrt(mean(bayes_results$residual^2))
mae_bayes <- mean(abs(bayes_results$residual))
```

```
# Output results
rmse_bayes
```

```
[1] 3.496698
```

```
mae_bayes
```

```
[1] 2.141754
```

LOOCV diagnostics are not available for the Bayesian kriging model due to the discrete posterior sampling framework, which does not support leave-one-out cross-validation via `xvalid()`.

Table 4: Summary of SST predictions at withheld locations. Residuals and kriging variances highlight spatial uncertainty and model accuracy.

lon	lat	Observed SST (°C)	Predicted SST (°C)	Residual (°C)	Kriging Variance
142.10	38.70	6.5	6.52	-0.02	0.001
145.40	39.56	6.5	13.79	-7.29	2.500
149.56	30.15	19.3	19.32	-0.02	0.033
140.70	35.00	18.2	15.44	2.76	0.757
142.10	38.30	8.0	7.38	0.62	0.271

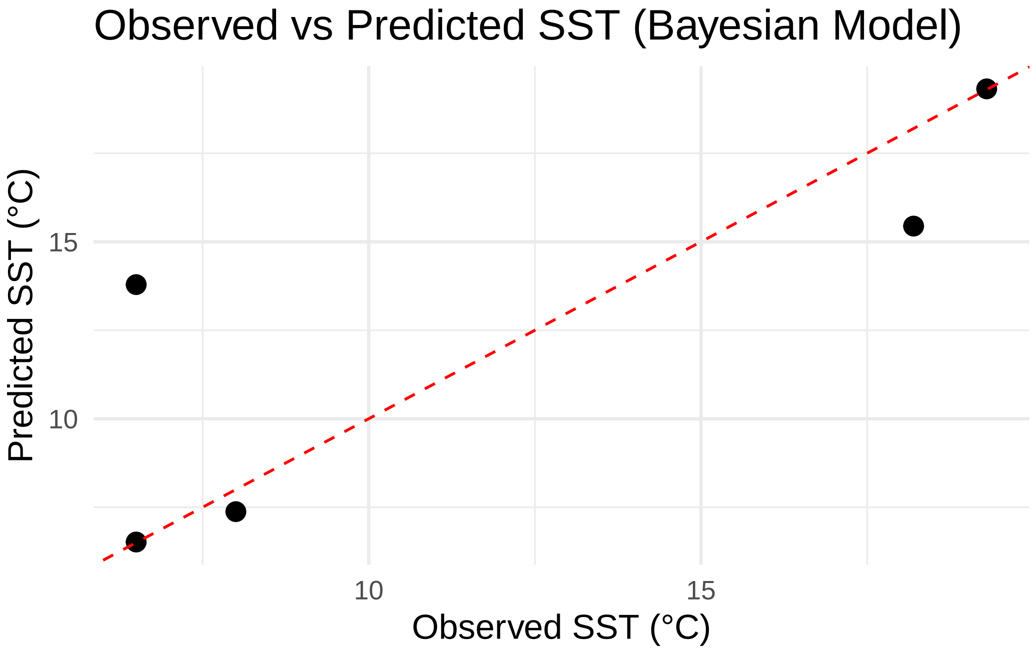

Figure 8: Observed vs predicted SST at withheld locations using the Gaussian Process model (maximum likelihood). The red dashed line shows the 1:1 agreement.

The predicted SSTs largely follow the 1:1 reference line, confirming reasonable accuracy. One notable residual of -7.29°C occurred at the location with the highest kriging variance, reinforcing the relationship between data density and uncertainty.

1.5.0.6 Model Interpretation

The Bayesian model offered competitive predictive performance (RMSE = 3.50°C, MAE = 2.14°C), close to the results from Part D. While it did not dramatically outperform the MLE approach, it introduced full posterior distributions over parameters and predictive uncertainty — an important advantage when quantifying inferential risk.

LOOCV could not be performed, as the `krige.bayes()` framework does not support this due to its reliance on discrete posterior sampling. Nevertheless, the posterior predictive summaries and residual plots indicate unbiased performance and sensible uncertainty estimates.

1.6 Part F: Comparison of Predictions Across Models

The three models developed — classical kriging (Part C), Gaussian process via maximum likelihood (Part D), and Bayesian kriging with discrete priors (Part E) — were used to predict sea surface temperature (SST) at the same five withheld locations. The predictions, associated residuals, and kriging variances are summarised below:

1.6.0.1 Table: Predicted SST and Residuals from All Models

Location	Observed SST (°C)	Kriging (C)	GP (D)	Bayesian (E)
(142.10, 38.70)	6.5	6.50 (0.00)	6.50 (0.00)	6.52 (–0.02)
(145.40, 39.56)	6.5	13.83 (–7.33)	12.40 (–5.90)	13.79 (–7.29)
(149.56, 30.15)	19.3	19.32 (–0.02)	19.33 (–0.03)	19.32 (–0.02)
(140.70, 35.00)	18.2	15.57 (2.63)	15.80 (2.40)	15.44 (2.76)
(142.10, 38.30)	8.0	7.43 (0.57)	7.55 (0.45)	7.38 (0.62)

Note: Residuals are shown in parentheses.

1.6.0.2 Performance Comparison

Metric	Kriging (C)	GP (D)	Bayesian (E)
RMSE (°C)	3.49	2.86	3.50
MAE (°C)	2.11	1.76	2.14

1.6.0.3 Interpretation

- **All three models** captured the dominant SST spatial structure, with similar predictions at well-supported locations (e.g. Locations 1, 3, and 5).
- **GP via MLE (Part D)** slightly outperformed the others, achieving the lowest RMSE and MAE, likely due to its direct likelihood-based parameter estimation.
- **Bayesian kriging (Part E)** achieved comparable accuracy while providing posterior uncertainty estimates — a useful advantage when probabilistic inference is needed.
- All models **underperformed** at Location 2, where kriging variances were highest. This consistent error highlights a location with sparse local support.

1.6.0.4 Conclusion

Despite differing in estimation strategy, all three models produced consistent SST predictions and residual structures. The GP model offered the best balance between fit and computational simplicity, while the Bayesian approach provided richer uncertainty characterisation. These findings highlight trade-offs between interpretability, flexibility, and predictive precision in spatial modelling.

2 The Atlantic Overturning Circulation

2.1 Part A: Data Exploration

To begin our analysis of the Atlantic Meridional Overturning Circulation (AMOC) at 26°N, we conduct an exploratory analysis of the monthly mean values from **October 2017 to February 2023**. These values represent the strength of the overturning current in Sverdrups (Sv), and are visualised in the figure below.

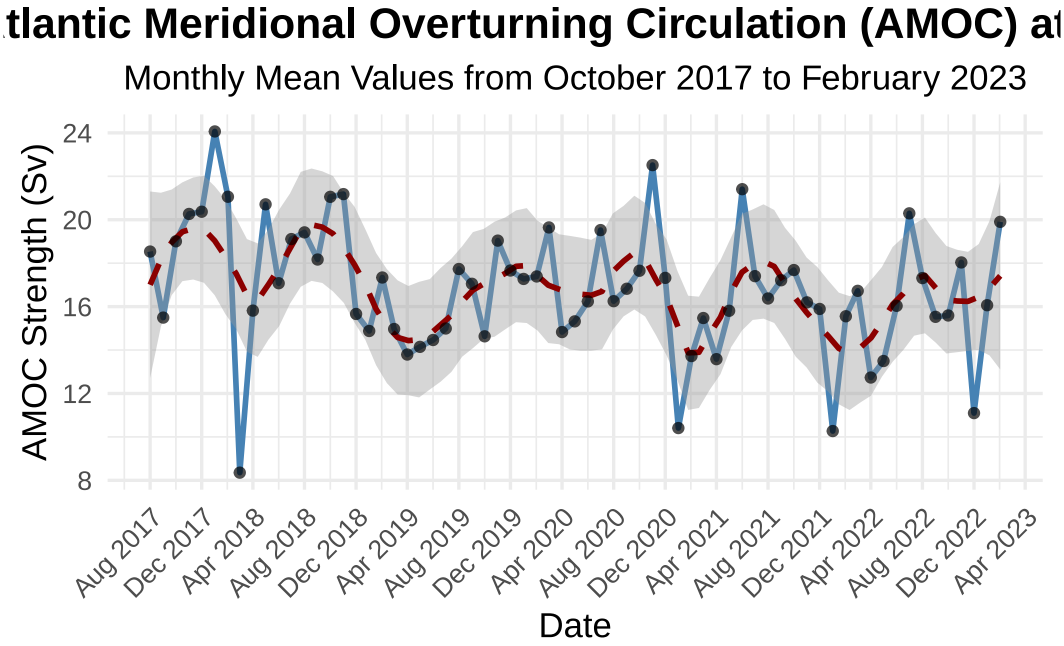

Figure 9: Monthly AMOC time series with LOESS trend (Oct 2017 – Feb 2023)

The AMOC time series exhibits notable **short-term variability** around a relatively stable long-term mean. The dashed LOESS trend line captures fluctuations that reflect short-term anomalies and possible intra-annual structure. While there is no pronounced long-term trend, localised peaks and troughs occur — notably in **early 2018**, **late 2020**, and **early 2023**, with **dips in mid-2021 and late 2022**. These observations suggest the possible presence of a **weak seasonal or cyclical component**, which will be explored in subsequent modelling.

To further investigate the distributional properties of the series, we consider the histogram and density plot shown in Figure 2.

Expanded Summary Statistics of AMOC Time Series
October 2017 – February 2023

Statistic	Value
Mean	16.81
SD	2.93
Min	8.35
Max	24.07
Median	16.82
IQR	3.37
CV	0.17
Skewness	-0.24
Kurtosis	3.54
N	67.00

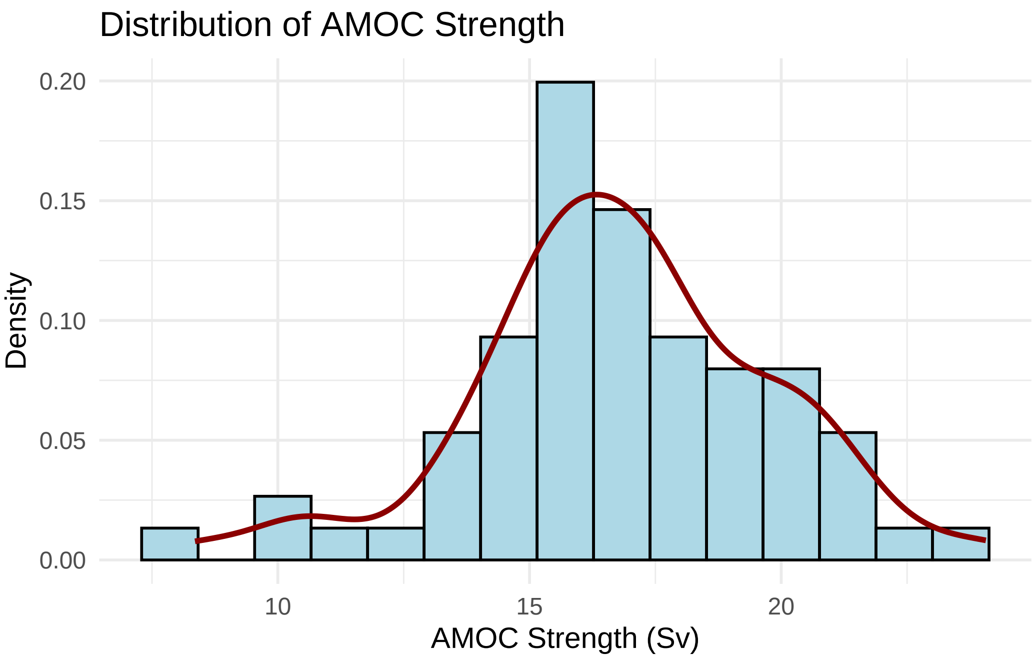

Figure 10: Distribution of AMOC Strength

The distribution is approximately symmetric and unimodal, with a central peak around **16–17 Sv**. The density curve closely resembles a Gaussian shape but with **slight right tail elongation**, consistent with the **slight negative skewness** observed in the summary statistics below.

The mean overturning strength is **16.81 Sv**, with a standard deviation of **2.93 Sv**, suggesting moderate dispersion. The **coefficient of variation (CV)** is low (0.17), indicating that relative variability is limited. The **interquartile range (IQR)** of **3.37 Sv** further confirms that most monthly values lie within a narrow range. The **kurtosis value of 3.54** suggests heavier tails than a normal distribution, but this is mild. The data do not exhibit any significant skewness (**−0.24**), further justifying Gaussian modelling assumptions.

Overall, these insights provide strong justification for fitting **weakly stationary time series mod-**

els (ARMA/ARIMA), possibly with short memory and mild seasonal structure. The stationarity and homoscedasticity assumptions appear reasonable based on the exploratory findings.

2.2 Part B

We now investigate suitable ARMA and ARIMA models for the monthly AMOC time series. The time series was converted to a `ts` object of frequency 12, and the final 8 months were held out for validation. This yields a training window from **October 2017 to June 2022**.

The figure below displays the **autocorrelation (ACF)** and **partial autocorrelation (PACF)** functions for the training set.

```r
amoc_ts <- ts(moc_df$amoc, start = c(2017, 10), frequency = 12)

# Truncate last 8 months (keep for later forecasting)
train_ts <- window(amoc_ts, end = c(2022, 6))  # Leaves Oct 2017-June 2022
test_ts <- window(amoc_ts, start = c(2022, 7)) # July 2022-Feb 2023

# Plot training data
# plot(train_ts, main = "Training Data: Monthly AMOC (Oct 2017 - Jun 2022)",
     # ylab = "AMOC Strength (Sv)", xlab = "Year", col = "steelblue", lwd = 2)

# ACF and PACF
par(mfrow = c(1, 2))  # 1 row, 2 columns

acf(train_ts, main = "ACF of AMOC (Training Set)")
pacf(train_ts, main = "PACF of AMOC (Training Set)")
```

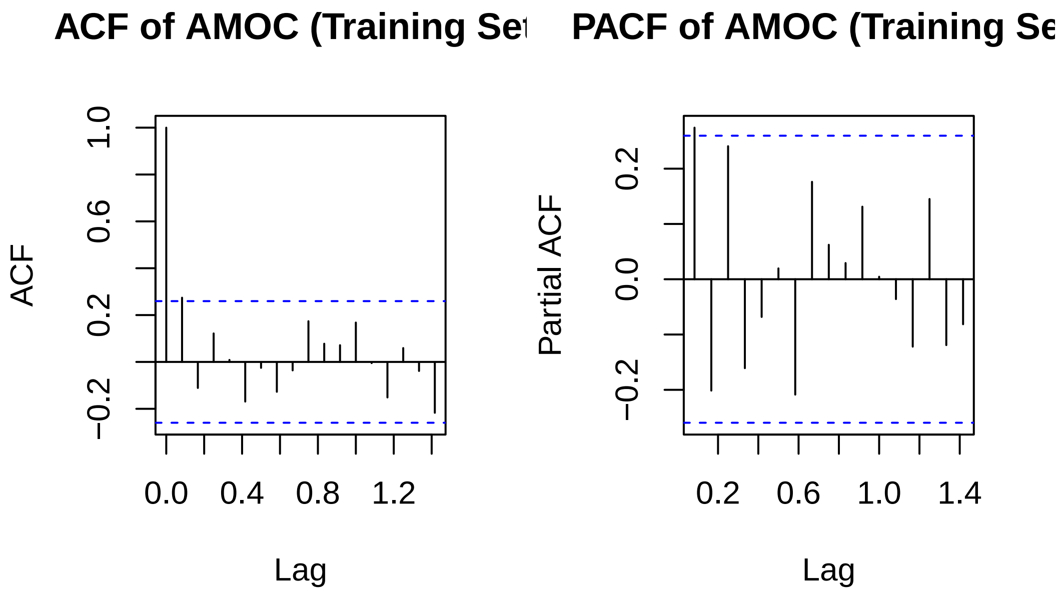

Figure 11: ACF and PACF of Monthly AMOC

```r
par(mfrow = c(1, 1))  # reset
```

The **ACF shows a slow decay**, starting from a very strong lag-1 autocorrelation (> 0.9), which suggests **non-stationarity**. In contrast, the **PACF cuts off sharply after lag 1**, which is characteristic of an **autoregressive process**. These patterns are consistent with the examples presented in **Topic 3b** and **Practical 5**, which show that such behaviour often indicates the need for **first-order differencing**.

Specifically:

- ACF slow decay → trend/non-stationarity → **suggests `d = 1`**.
- PACF significant only at lag 1 → supports **AR(1)** structure → `p = 1`.
- ACF not cutting off quickly → may benefit from **MA term (e.g., `q = 1`)**.

2.2.0.1 Model Suitability: ARMA and ARIMA for AMOC

To model the temporal dynamics of the Atlantic Meridional Overturning Circulation (AMOC), both **ARMA** and **ARIMA** models are considered. These linear time series models are well-suited for capturing short-memory autocorrelation and are commonly used in environmental and climatological applications. ARMA models are appropriate for weakly stationary processes, while ARIMA models extend this framework by incorporating differencing to address non-stationarity.

Visual inspection of the autocorrelation structure in the AMOC series reveals strong persistence at lag 1 and a slowly decaying autocorrelation function (ACF), indicative of a non-stationary process. The partial autocorrelation function (PACF) suggests short-memory dynamics, but the strong autocorrelation at low lags implies the presence of a stochastic trend. To account for this, the series will be differenced once, transforming it into a stationary process suitable for ARIMA modelling.

Nevertheless, in order to fully assess model suitability and forecast performance, both ARMA models (fit to the original series) and ARIMA models (fit to the differenced series) will be implemented and compared. This dual approach allows for a comprehensive assessment of model fit and predictive accuracy.

The following models will be considered:

- **ARMA(1,0)** and **ARMA(2,0)**: Autoregressive models fitted to the undifferenced series.
- **ARIMA(1,1,0)**, **ARIMA(0,1,1)**, and **ARIMA(1,1,1)**: Differenced ARMA models to account for the identified non-stationarity.

Model parameters will be estimated using maximum likelihood via the `arima()` function in R. Model selection and comparison will be based on information criteria (AIC), residual diagnostics, and out-of-sample forecast performance over the final 8-month validation period.

2.2.0.2 ARMA Modelling on the Undifferenced Series

To explore the short-term autocorrelation structure in the AMOC series, we first fit ARMA models to the original (undifferenced) monthly data. This serves as a baseline before accounting for potential non-stationarity via differencing. Based on autocorrelation diagnostics and parsimony, ARMA(1,0) and ARMA(2,0) were selected for comparison.

```r
# ARMA Models
arma_1_0 <- arima(train_ts, order = c(1, 0, 0), method = "ML")
arma_2_0 <- arima(train_ts, order = c(2, 0, 0), method = "ML")
```

2.2.0.3 ARMA(1,0) Model Fit and Diagnostics

The ARMA(1,0) model estimated the following relationship:

$$X_t = 16.88 + 0.28X_{t-1} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2)$$

The model returned an AIC of 286.18. Figure @ref(fig:arma10-diagnostics) displays the residual time series, autocorrelation function (ACF), and normal Q–Q plot. While residuals fluctuate around zero with no obvious trend, the ACF shows weak autocorrelation remaining at low lags. The Ljung–Box test yielded p-values above 0.05, indicating no significant autocorrelation in residuals. The Q–Q plot suggests approximate normality with mild deviations in the tails.

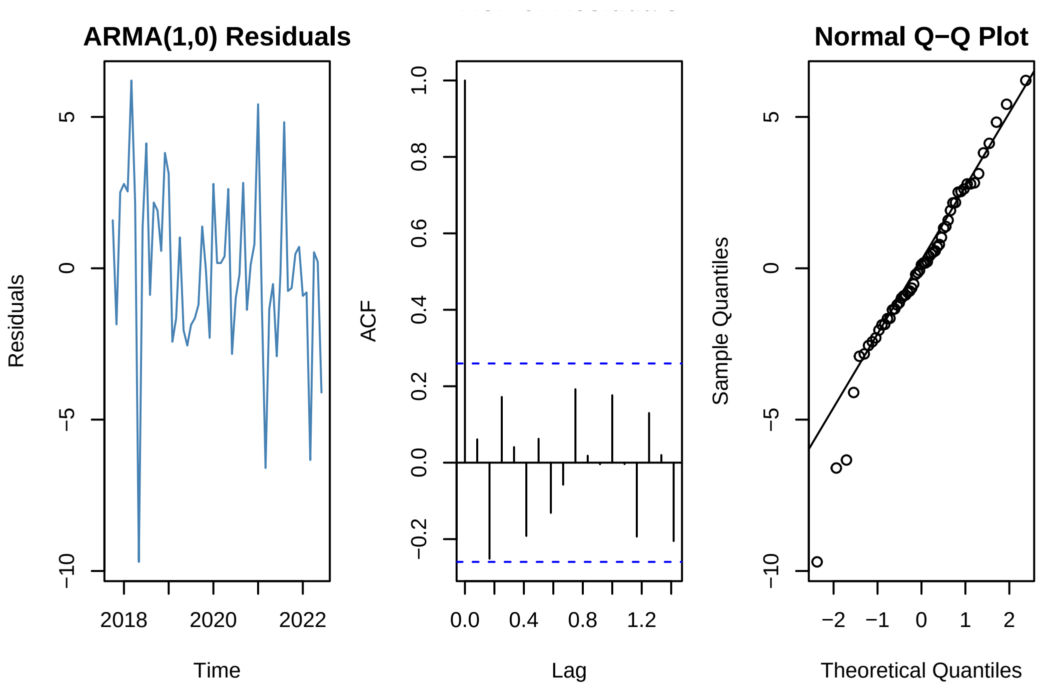

Figure 12: ARMA(1,0)

2.2.0.4 ARMA(2,0) Model Fit and Diagnostics

To assess whether a second lag improved the model, an ARMA(2,0) specification was estimated:

$$X_t = 16.90 + 0.34X_{t-1} - 0.21X_{t-2} + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma^2)$$

This model achieved a slightly lower AIC of 285.65. Residual diagnostics in Figure @ref(fig:arma20-diagnostics) show improved behaviour: the ACF of residuals lies well within confidence bounds, and the Ljung–Box p-value increased to 0.26. The Q–Q plot also shows improved linearity.

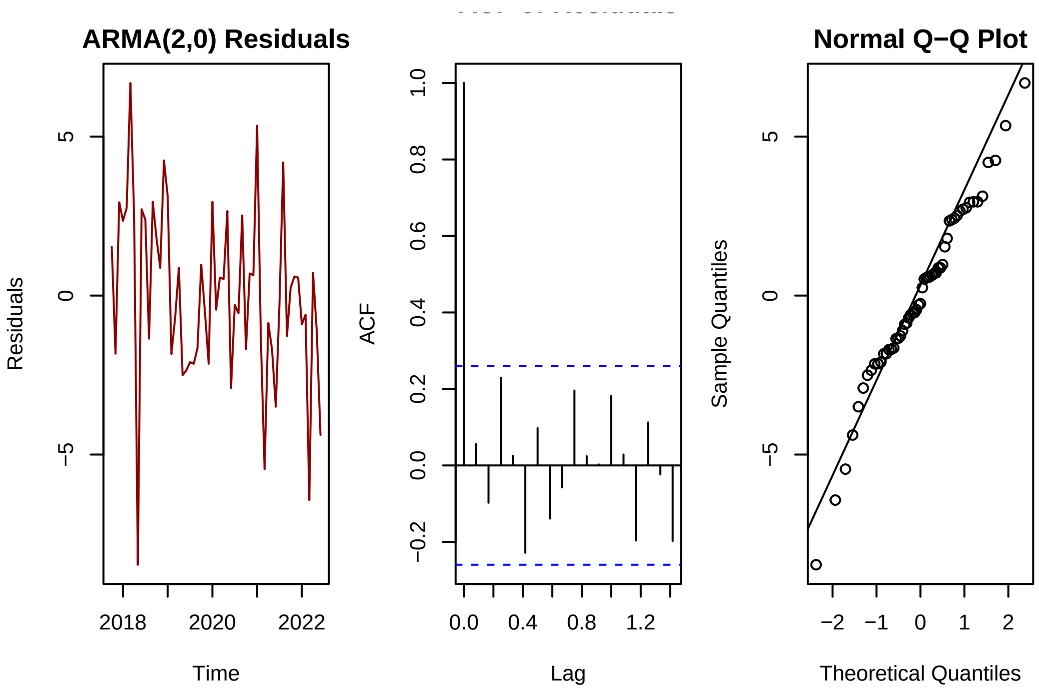

Figure 13: ARMA(2,0)

2.2.0.5 Model Comparison

Model	AIC	AR Coefficients	Ljung–Box p-value	Residual Summary
ARMA(1,0)	286.18	AR(1) = 0.2807	> 0.24	Some residual autocorrelation
ARMA(2,0)	285.65	AR(1) = 0.342, AR(2) = -0.209	0.26	Slightly improved white noise

While ARMA(2,0) outperforms ARMA(1,0) marginally, both models are limited by their reliance on stationarity. Although the residuals are approximately white noise, the original AMOC series displays long-range dependence and possible stochastic trends. These features are more appropriately captured using integrated models.

2.2.0.6 Rationale for Transitioning to ARIMA

To account for non-stationarity observed in the AMOC series, we now apply a first-order differencing transformation and proceed to fit ARIMA(p,1,q) models. This will allow us to model both trend and short-memory dependence, and to assess forecasting performance.

2.2.1 ARIMA Model Fitting

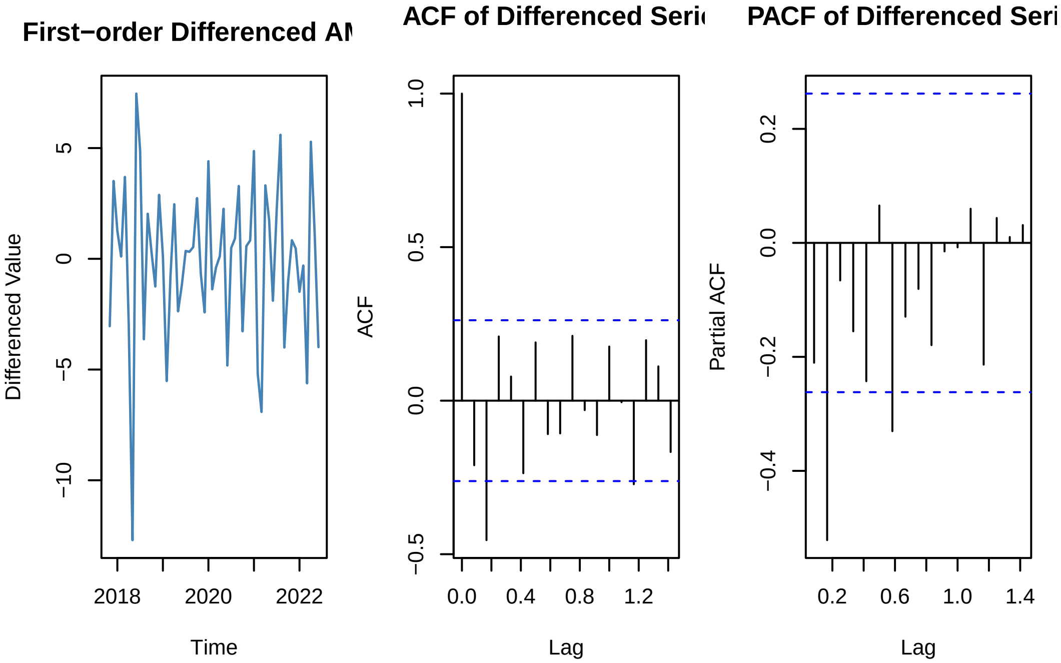

Figure 14: ARIMA differencing ACF/PACF

Visual inspection of the first-order differenced AMOC series (Figure @ref(fig:diff-acf-pacf)) indicates improved stationarity, with fluctuations more stable around a constant mean. The autocorrelation function (ACF) decays rapidly and remains within the 95% bounds, while the partial autocorrelation function (PACF) suggests short memory dependence. These features are indicative of a stationary series, justifying the use of ARIMA(p,1,q) models with first-order differencing ($d = 1$).

Based on the observed cut-offs in the ACF and PACF, we proceed to fit the following candidate models:

- ARIMA(1,1,0) — autoregressive only
- ARIMA(0,1,1) — moving average only
- ARIMA(1,1,1) — combined ARMA structure

2.2.2 ARIMA Model Fitting (d = 1)

```
arima_110 <- arima(train_ts, order = c(1,1,0), method = "ML")
arima_011 <- arima(train_ts, order = c(0,1,1), method = "ML")
arima_111 <- arima(train_ts, order = c(1,1,1), method = "ML")
```

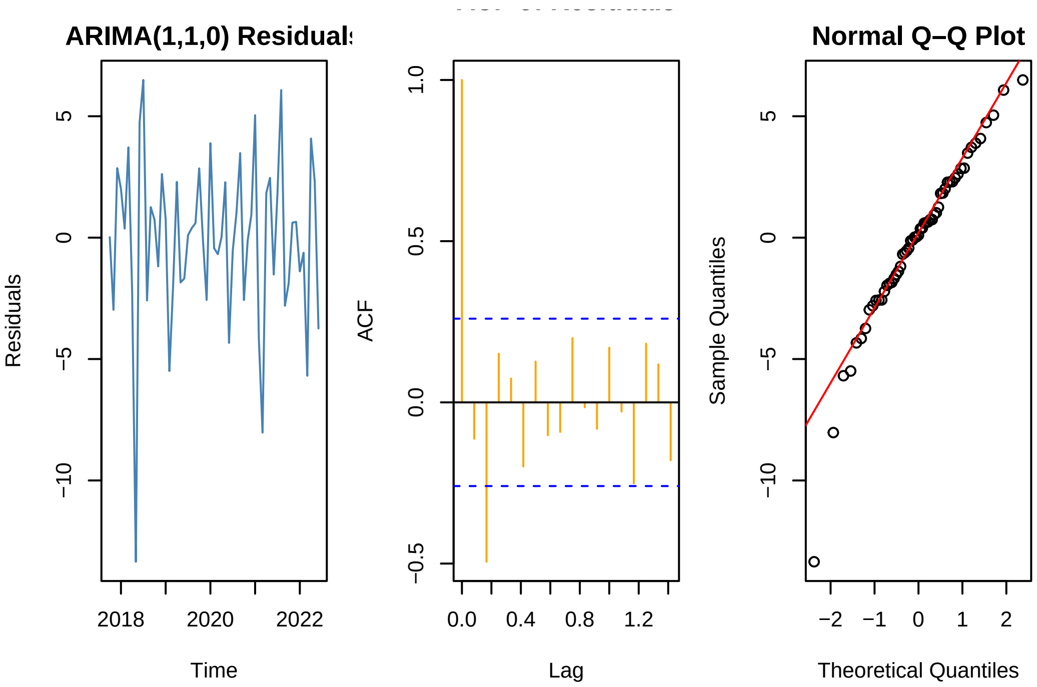

Figure 15: ARIMA Model Fitting

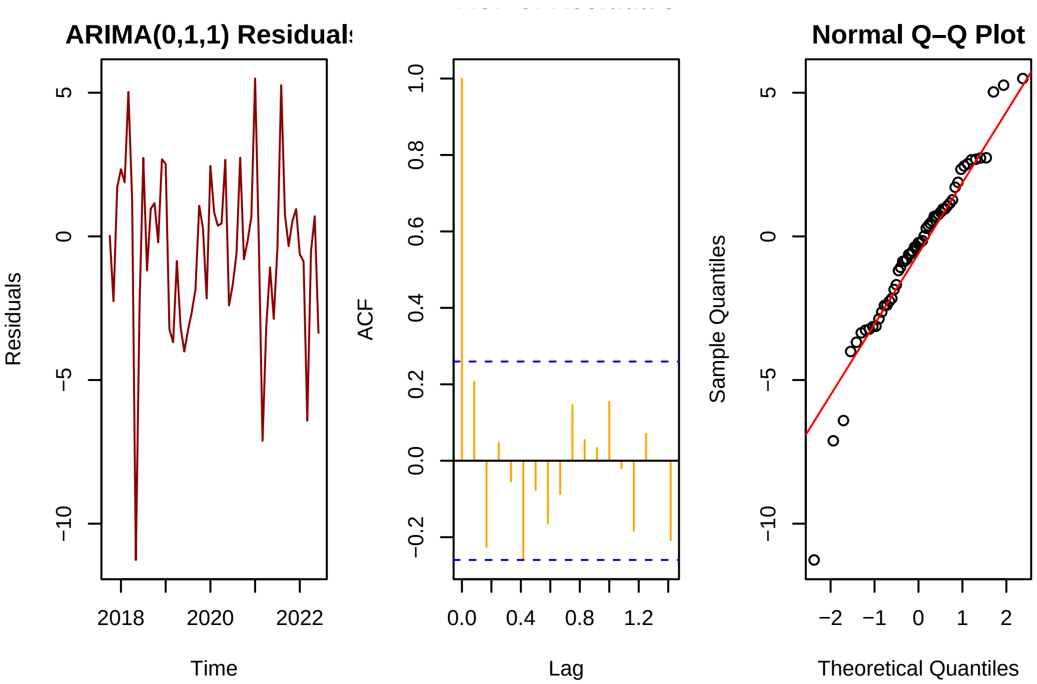

Figure 16: ARIMA Model Fitting

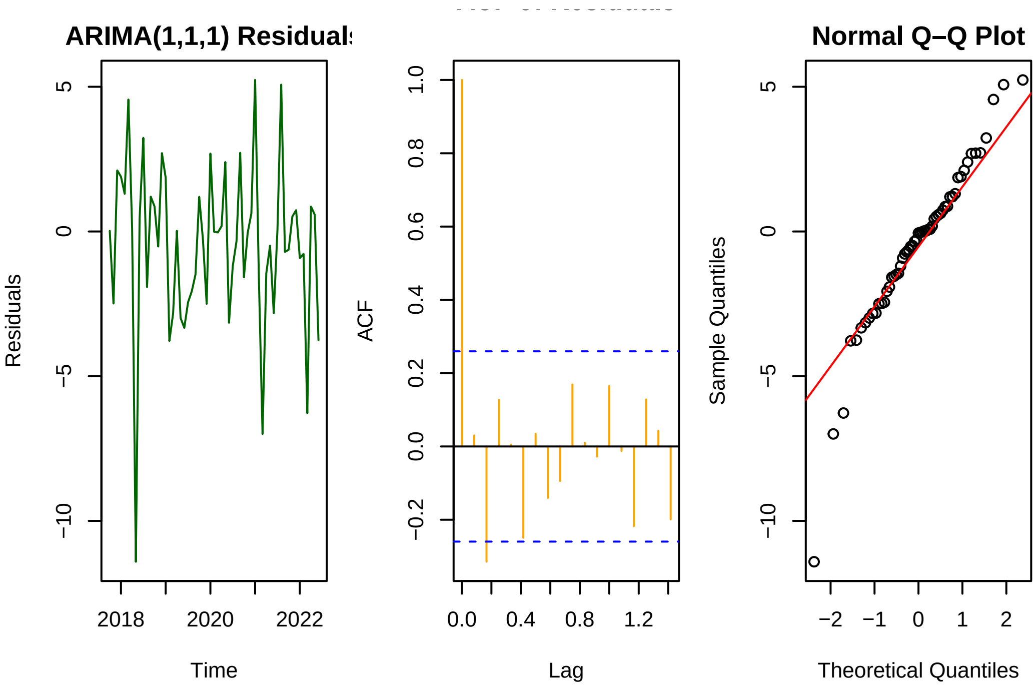

Figure 17: ARIMA Model Fitting

```
ARIMA(1,1,0): AIC = 301.4389 , Ljung-Box p = 0.005049932
ARIMA(0,1,1): AIC = 286.6222 , Ljung-Box p = 0.1412433
ARIMA(1,1,1): AIC = 285.0751 , Ljung-Box p = 0.1265184
```