Homework 5 - Group 12

Francesco Natali, Leonardo Agate, Lorenzo Bartocci

2024-12-02

Load the dataset and the file containing the estimation grids

```
load("shrimpsfull.RData")
load("AllGrids.RData")
```

We now filter data for our years of interest, i.e. 2002 and 2008

```
shrimp_data_2002 <- shrimpsdata[shrimpsdata$ANNO == 2002, ]
shrimp_data_2008 <- shrimpsdata[shrimpsdata$ANNO == 2008, ]
```

2008

Implement Bayesian kriging

In Homework 4, when we estimated total biomass on the estimation grid using kriging in a maximum likelihood framework, our chosen covariates for the year 2008 were bathymetry, slope, salinity.maxq3 and temp.maxq1. In the Bayesian framework however, working with such covariates gives an overparametrization of the model and increasing values for the parameters range (phi) and tausq_relative. Hence, we substitute bat with distance and temp.maxq1 with temp.maxq3 respectively and mantain the variable "slope" to work in the Bayesian framework.

```
trend_matrix_2008 <- cbind(shrimp_data_2008$dist,
                           shrimp_data_2008$salinity.maxq3,
                           shrimp_data_2008$temp.maxq3,
                           shrimp_data_2008$slope)
```

As we already did in the previous analyses, we proceed with the log-transformation of the data with our newly chosen covariates

```
shrimp_data_2008$log_tot <- log(shrimp_data_2008$tot + 1)

shrimp_geodata_2008_log <- as.geodata(shrimp_data_2008,
                                      coords.col = c("X", "Y"),
                                      data.col = "log_tot",
                                      covar.col = c("dist", "salinity.maxq3",
                                                    "temp.maxq3","slope"))
```

Now, based on this log transformation, we prepare shrimp_geodata_2008_log for Bayesian kriging

```
trend.d.08 <- trend.spatial(~ dist+ salinity.maxq3 + temp.maxq3 + slope,
                            geodata = shrimp_geodata_2008_log)

trend.l.08 <- trend.spatial(~ grid_2008$dist + grid_2008$salinity.maxq3 +
                              grid_2008$temp.maxq3 + grid_2008$slope)
```

As the reference model we will work with a exponential (with default parameter k = 0.5) as we did in the maximum likelihood approach

```
model_08 <- list(
  trend.d = trend.d.08,
  trend.l = trend.l.08,
  cov.model = "exponential",
  kappa = 0.5
)
```

Now add the number of samples to generate for the posterior distribution. This number ensures a stable estimation of the credibility distributions, allowing precise evaluation of parameter variability

```
out <- output.control(1000,1000, quantile = c(0.025, 0.5, 0.975))
```

We now our prior distribution

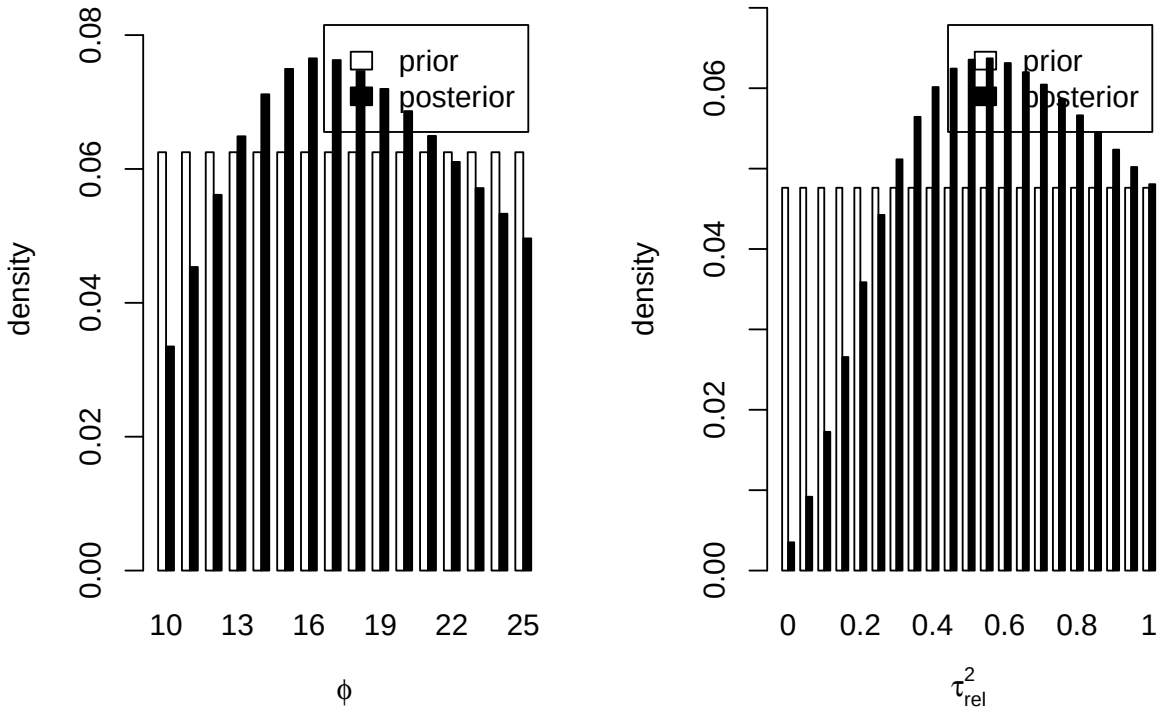
```
prior_08 <- list(
  beta.prior = "normal",
  beta = rep(0, 5),           # Intercept and 4 covariates
  beta.var.std = diag(100, 5), # Prior covariance
  sigmasq.prior = "reciprocal",
  phi.prior = "uniform",
  phi.discrete = seq(10, 25, 1),
  tausq.rel.prior = "uniform",
  tausq.rel.discrete = seq(0, 1, 0.05)
)
```

The choice of priors in Bayesian kriging is crucial as it directly influences the posterior distribution and, consequently, the final predictions. About Beta Prior choose a normal prior with mean zero and a large variance, that reflects a non-informative prior belief. For Sigma-squared prior we choose a reciprocal prior, that is a standard non-informative choice, ensuring that all positive values are equally likely. For Phi Prior (for the spatial range parameter) we use uniform prior over a discretized range (phi.discrete = seq(10, 25, by = 1)) indicates an assumption that all values within this interval are equally probable. At the end Tau-squared Relative Prior (nugget effect relative to sigma-squared) use uniform prior over a sequence (tausq.rel.discrete = seq(0, 1, 0.05)) suggests that the nugget effect (representing measurement error or microscale variation) is considered equally probable within this range. From that, we get the posterior and we use the "krige.bayes" function to implement Bayesian kriging, first without the locations in order to choose a good posterior

```
krige_bayes_08 <- krige.bayes(
  geodata = shrimp_geodata_2008_log,
  model = model_08,
  prior = prior_08,
  output = out
)
```

Now we plot it to see if our choice looks good

```
par(mfrow = c(1, 2))
plot(krige_bayes_08)
```

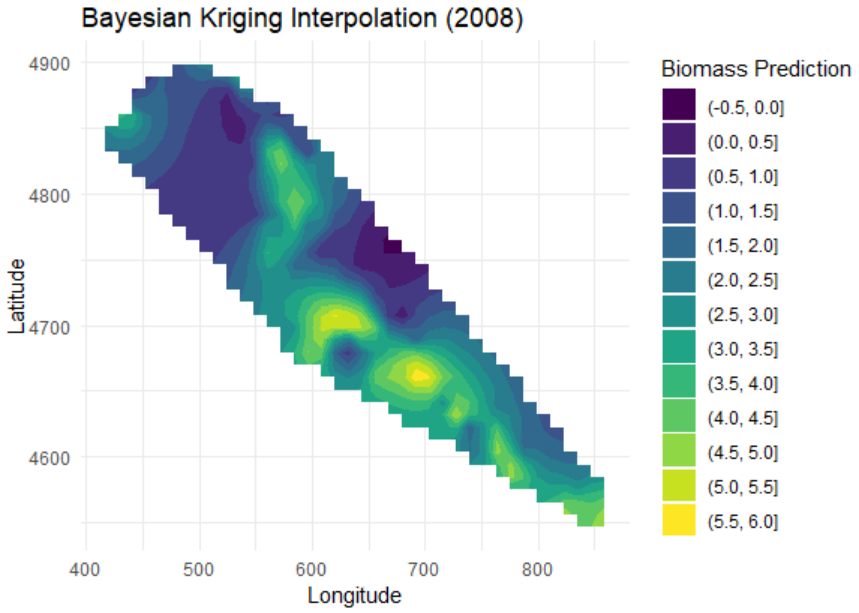


The posterior distributions should have clear and well-defined peaks, indicating that the Bayesian inference process has converged to a stable estimate. And with the parameters we chose, this seems to be achieved. After the posterior choice we can add the locations

```
krige_bayes_2008 <- krige.bayes(
  geodata = shrimp_geodata_2008_log,
  locations = as.matrix(grid_2008[, c("X", "Y")]),
  model = model_08,
  prior = prior_08,
  output = out
)

bayes_preds_2008 <- krige_bayes_2008$predictive$mean
```

We now visualize Bayesian Kriging predictions

```r
cc_bayes_2008 <- data.frame(X = grid_2008$X, Y = grid_2008$Y, Z = bayes_preds_2008)

ggplot(cc_bayes_2008, aes(x = X, y = Y, z = Z)) +
  geom_contour_filled() +
  labs(
    title = "Bayesian Kriging Interpolation (2008)",
    x = "Longitude",
    y = "Latitude",
    fill = "Biomass Prediction"
  ) +
  theme_minimal()
```



This map represents the Bayesian kriging interpolation of shrimp biomass along the Tyrrhenian coast for 2008. It reveals distinct spatial patterns, with biomass predictions ranging from low (dark blue and purple) to high (green and yellow). The highest biomass concentrations, indicated by green and yellow zones, are primarily located in the central and southeastern areas. These regions, showing biomass values between 4.5 and 6.0, likely reflect optimal environmental conditions such as moderate bathymetry, stable salinity, and favorable temperatures that support shrimp breeding and growth. In contrast, the northern areas exhibit lower biomass values, represented by dark blue and purple. This pattern suggests that factors like steep bathymetric gradients or deeper waters in the northern Tyrrhenian may limit suitable habitats for shrimp. Overall, this map highlights key hotspots for shrimp biomass in the central and southeastern regions, indicating areas of potential ecological significance and emphasizing the importance of localized environmental conditions.

```r
italy <- ne_countries(country = "Italy", scale = "medium", returnclass = "sf")
italy <- st_transform(italy, crs = 32632)
```

Convert kriging results to a grid

```r
krig_result_df_08 <- data.frame(
  X = grid_2008$X * 1000, # Assuming the grid units need conversion to match Italy's CRS
  Y = grid_2008$Y * 1000,
```

```
  Z = bayes_preds_2008
)
```

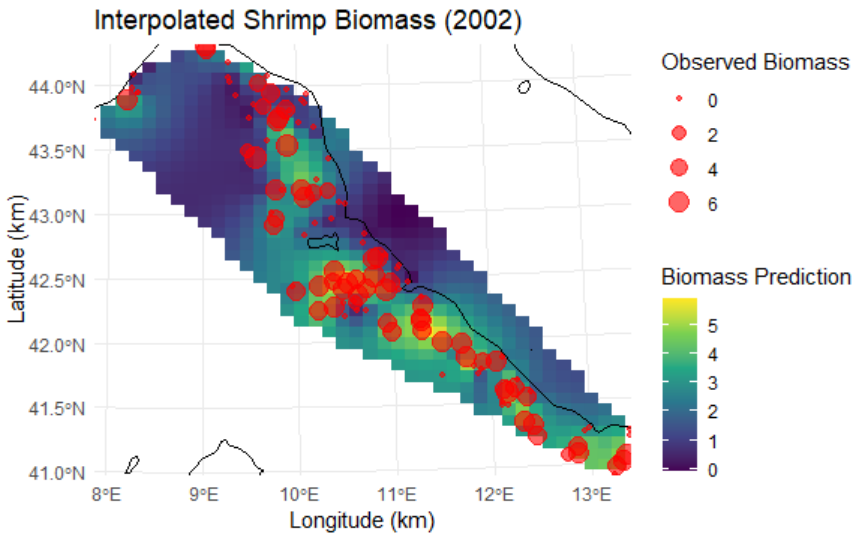
Define the bounds for the area of interest based on your prediction data

```
x_min <- min(krig_result_df_08$X) - 10000
x_max <- max(krig_result_df_08$X) + 10000
y_min <- min(krig_result_df_08$Y) - 10000
y_max <- max(krig_result_df_08$Y) + 10000
```

```
ggplot() +
  geom_raster( data = krig_result_df_08, aes(x = X, y = Y, fill = Z) ) +
  scale_fill_viridis_c( option ="viridis", name = "Biomass Prediction" ) +
  geom_sf( data = italy, fill = NA, color = "black",lwd = 0.7 ) +
  coord_sf( xlim = c(x_min, x_max), ylim = c(y_min, y_max), expand =FALSE ) +
  labs( title = "Interpolated Shrimp Biomass (2008)", x = "Longitude (km)", y ="Latitude (km)" ) +
  theme_minimal()
```

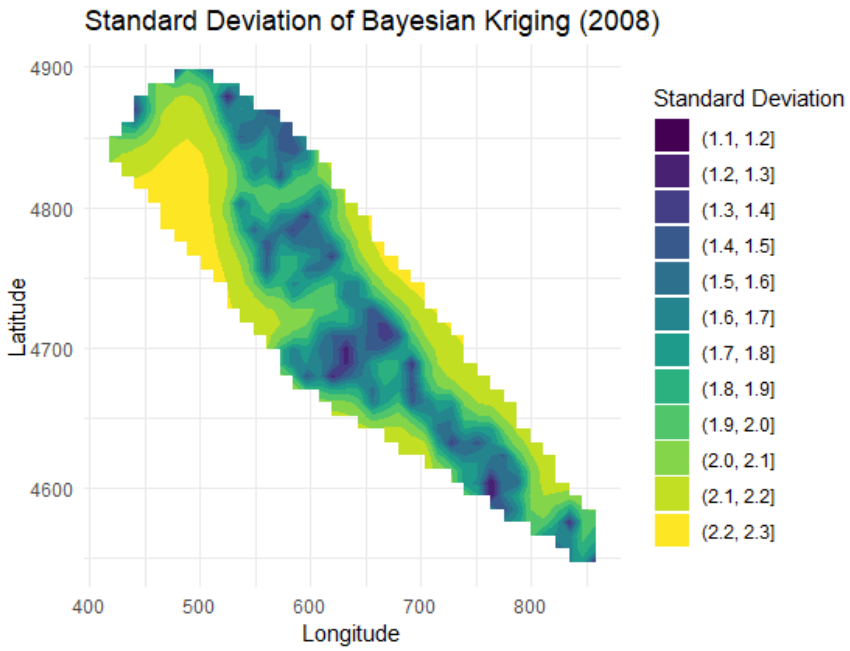
Add the observed biomass

```
observed_points_df08 <- data.frame(
  X = shrimp_geodata_2008_log$coords[, 1] * 1000,
  Y = shrimp_geodata_2008_log$coords[, 2] * 1000,
  Biomass = shrimp_geodata_2008_log$data
)

# Plot with Italy map, kriging results, and observed points
ggplot() +
  geom_raster(data = krig_result_df_08, aes(x = X, y = Y, fill = Z)) +
  scale_fill_viridis_c(option = "viridis", name = "Biomass Prediction") +
  geom_sf(data = italy, fill = NA, color = "black", lwd = 0.7) +
  geom_point(data = observed_points_df02, aes(x = X, y = Y, size = Biomass),
             color = "red", alpha = 0.6) +
  scale_size_continuous(name = "Observed Biomass", range = c(1, 5)) +
  coord_sf(xlim = c(x_min, x_max), ylim = c(y_min, y_max), expand = FALSE) +
  labs(title = "Interpolated Shrimp Biomass (2008)", x = "Longitude (km)", y = "Latitude (km)") +
  theme_minimal()
```



This plot integrates Bayesian kriging interpolation with observed shrimp biomass data points along the Tyrrhenian coast. The predicted biomass distribution (from low in purple to high in yellow) is visually compared with the observed biomass values, represented by red circles of varying sizes. There is a general alignment between areas with high predicted biomass (green and yellow zones) and the observed high biomass concentrations, suggesting that the Bayesian kriging model captures the spatial variability of shrimp distribution effectively. The central and southern sections, with observed high biomass values, align well with the model's predictions. However, discrepancies are noticeable in some areas where observed biomass values do not perfectly correspond with the predicted values. These deviations may stem from factors such as data limitations or unaccounted environmental variables. The lower biomass predictions in the northern regions (dark blue) are consistent with fewer observed shrimp populations, likely due to the less favorable conditions in deeper waters. In summary, this map demonstrates the Bayesian kriging model's effectiveness in predicting shrimp biomass distribution, with some deviations highlighting the need for continuous model validation against observed data. Also we can plot the spatial distribution of uncertainty

```
std_dev_bayes_08 <- sqrt(krige_bayes_2008$predictive$variance)

cc_bayes_2008$std_dev <- std_dev_bayes_08

ggplot(cc_bayes_2008, aes(x = X, y = Y, z = std_dev)) +
  geom_contour_filled() +
  labs(
    title = "Standard Deviation of Bayesian Kriging (2008)",
    x = "Longitude",
    y = "Latitude",
    fill = "Standard Deviation"
  ) +
  theme_minimal()
```



The map shows the standard deviation of Bayesian kriging predictions for shrimp biomass, with uncertainty ranging from 1.1 to 2.3, slightly improved compared to 2002. Lower uncertainty (dark purple) is concentrated near observation-dense regions, while higher uncertainty (yellow) persists at the boundaries due to extrapolation. Central areas exhibit moderate uncertainty, reflecting consistent data coverage and predictive reliability. Improvements in precision may result from better data quality or environmental stability in 2008. To enhance prediction accuracy, future monitoring should target boundary areas with higher uncertainty, while the model effectively captures spatial variability in the core study region.

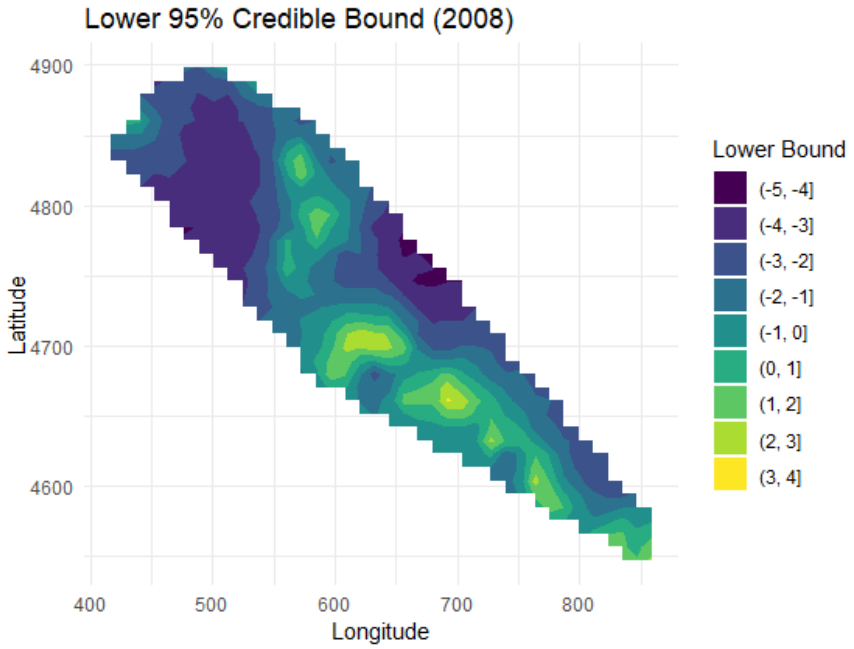
Evaluate estimates precision using credibility intervals.

At this point, we add confidence intervals to the Bayesian Kriging plot

```
lower_bayes_08 <- apply(krige_bayes_2008$predictive$simulations, 1, quantile, probs = 0.025)
upper_bayes_08 <- apply(krige_bayes_2008$predictive$simulations, 1, quantile, probs = 0.975)
```

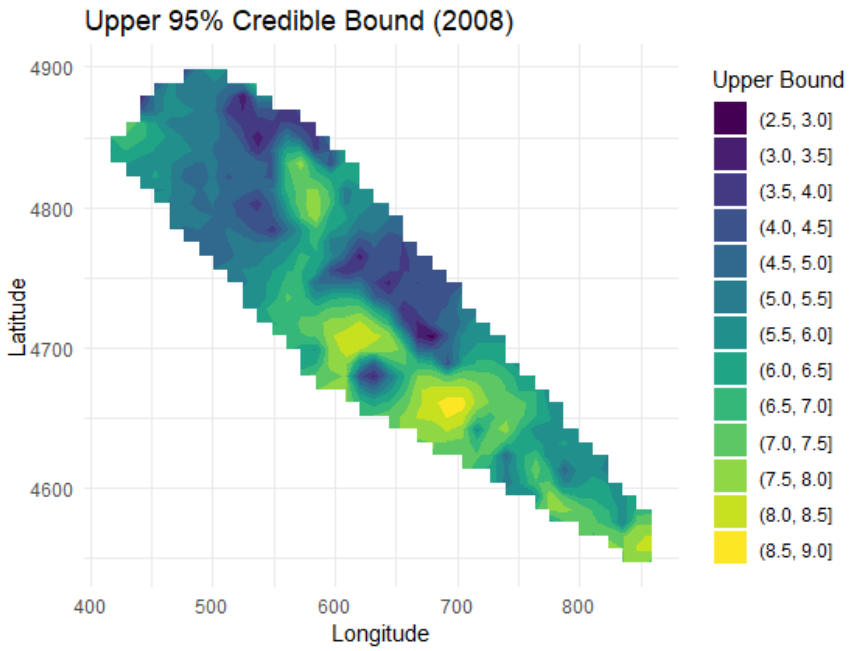
Visualization of the lower credible intervals

```
cc_bayes_2008$lower <- lower_bayes_08

ggplot(cc_bayes_2008, aes(x = X, y = Y, z = lower)) +
  geom_contour_filled() +
  labs(
    title = "Lower 95% Credible Bound (2008)",
    x = "Longitude",
    y = "Latitude",
    fill = "Lower Bound"
  ) +
  theme_minimal()
```



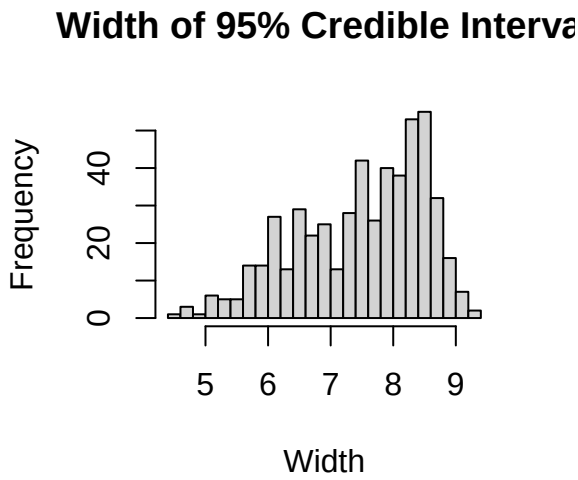
The lower credible bound shows the minimum biomass values that are 95% likely given the model and data. The purple regions represent areas where the lower bound is very low (close to -5.0 or -3.0 units), suggesting a higher uncertainty or low biomass in these locations. The yellow to green regions show higher lower bounds (e.g., between 2.0 and 4.0), meaning that even at the lower limit, shrimp biomass is relatively higher in these areas, implying more certainty of moderate-to-high biomass presence. Now visualize of the upper credible intervals

```
cc_bayes_2008$upper <- upper_bayes_08

ggplot(cc_bayes_2008, aes(x = X, y = Y, z = upper)) +
  geom_contour_filled() +
  labs(
    title = "Upper 95% Credible Bound (2008)",
    x = "Longitude",
    y = "Latitude",
    fill = "Upper Bound"
  ) +
  theme_minimal()
```



The upper credible bound indicates the maximum likely biomass values at the 95% level. Here, the yellow regions represent areas with very high biomass upper limits (e.g., between 8.0 and 9.0), suggesting a strong potential for higher shrimp biomass. The green to blue regions represent locations where the upper credible bound is lower (e.g., between 2.5 and 4.0), indicating either less potential biomass or less uncertainty in the estimate. Let's see the histogram of credible interval widths

```
hist(upper_bayes_08 - lower_bayes_08,
     breaks = 30, main = "Width of 95% Credible Interval", xlab = "Width")
```



The histogram of credible interval widths for shrimp biomass shows a right-skewed distribution, with most intervals centered around 8.0-9.0, but ranging from 4.0 to 9.0. This indicates varying model confidence across the study area, likely due to data density and environmental factors. The wide range of interval

widths suggests significant shrimp biomass variability, which is crucial for informed fisheries management decisions. Calculate the width of the credible intervals

```
credible_intervals_08 <-  upper_bayes_08 - lower_bayes_08
```

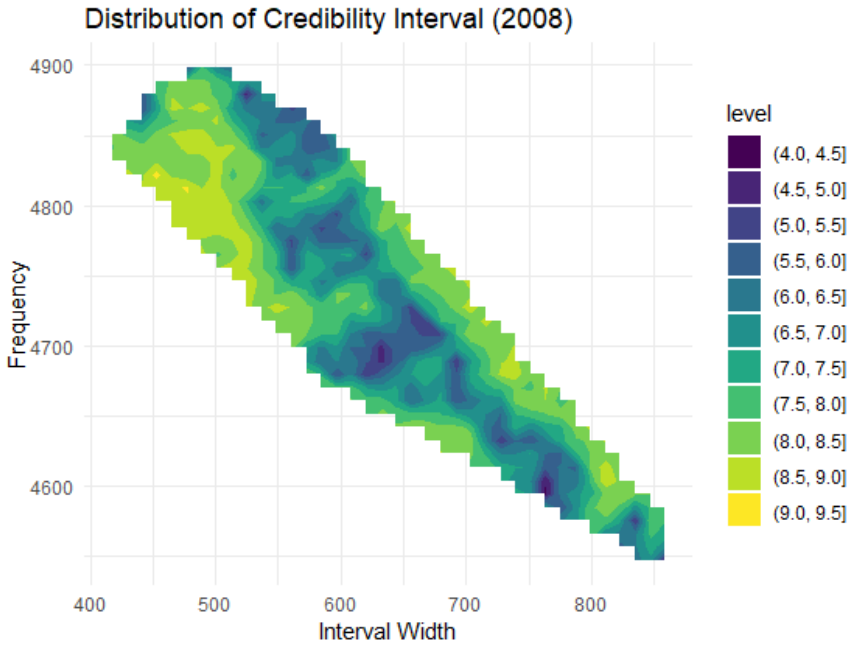
Summary of interval widths

```
summary(credible_intervals_08)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.521   6.696   7.672   7.488   8.336   9.271
```

The median and mean values are close, indicating that the distribution is roughly symmetric. The interquartile range (IQR) is 1.58, suggesting a moderate spread in the data. The range from the minimum to the maximum value is 4.833, further indicating variability in the widths. We finally visualize interval width distribution

```
cc_interval <- data.frame(X = grid_2008$X, Y = grid_2008$Y, Z = credible_intervals_08)

ggplot(cc_interval, aes(x = X, y = Y, z = Z)) +
  geom_contour_filled() +
  labs(
    title = "Distribution of Credibility Interval Widths 2008",
    x = "Interval Width",
    y = "Frequency"
  ) +
  theme_minimal()
```

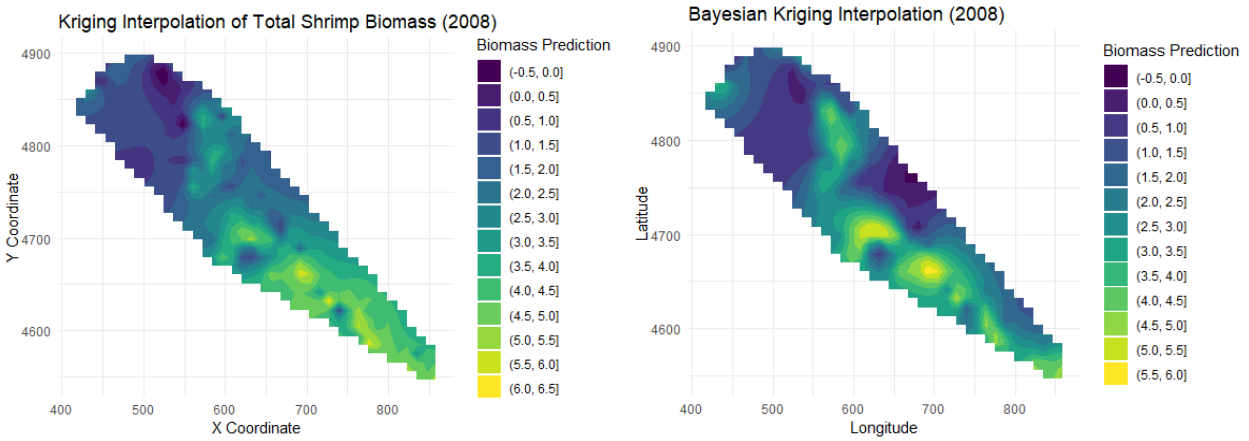


The analysis reveals greater uncertainty in shrimp biomass predictions for Liguria region, likely due to complex habitats, variable conditions, or data scarcity. In contrast, Lazio and Tuscany region shows higher confidence, possibly due to simpler environments, more data, or better understanding.

Discuss your results and compare them to those obtained in liklihood framework

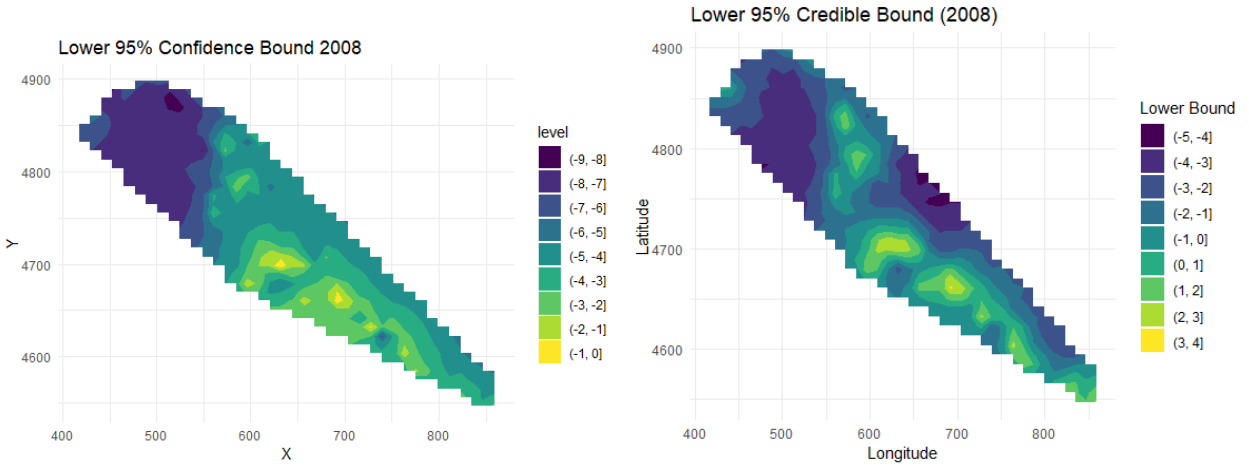
Finally, we want to compare the results obtained under the two frameworks related to the map of our kriging interpolations, start with a visual comparison

```
knitr::include_graphics(c("images/krig2008-01.png", "Bayes_08.png"))
```



Contrary to the 2002 case, the results obtained for 2008 under the Bayesian framework are much different compared to the maximum likelihood approach. Indeed, as we can now see from the obtained map, the highest concentration of biomass lies, similarly to 2002, in the central region of the area under study, that is in between the coasts of Tuscany and Lazio. However, in Homework 4 what we obtained was a kriging interpolation where the highest biomass concentration was much more spread all along the Southern regions under study (i.e. all along the coast of Lazio). For what concerns the lowest biomass values (including the zero's) instead the results are overall quite similar, with the Northern regions (near Liguria) showing darker blue areas in the kriging interpolation map for both approaches. Also we can compare the bounds

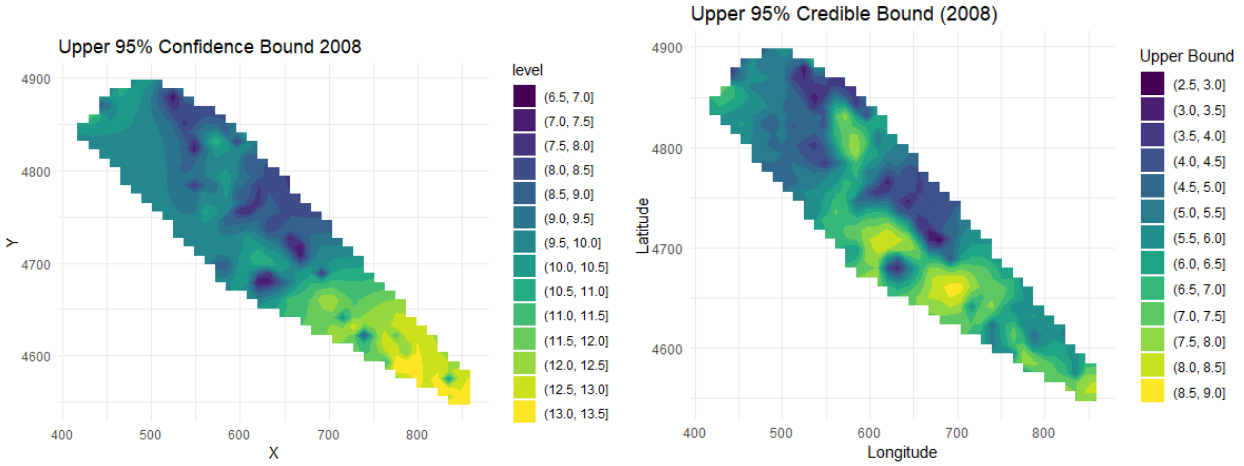
```
knitr::include_graphics(c("images/low_08-01.png", "Blow_08.png"))
```



The comparison of the spatial plots representing the lower bounds of confidence and credible intervals for 2008 highlights key differences between the likelihood-based and Bayesian kriging approaches. The likelihood framework exhibits smoother distributions with gradual color gradients, indicating continuous biomass variation, while the Bayesian framework appears more granular with abrupt transitions, reflecting sharper changes in biomass estimates. The likelihood plot suggests a wider range of values, indicating

greater uncertainty, whereas the Bayesian plot shows a narrower range, implying more precise estimates. Spatially, the likelihood model identifies higher biomass potential in localized patches, while the Bayesian model suggests a more homogeneous distribution.

```
knitr::include_graphics(c("images/up_08-01.png", "Bup_08.png"))
```



The comparison of spatial plots for the upper bounds of confidence and credible intervals in 2008 highlights that the likelihood-based kriging model produces smoother distributions with a wider range of values, indicating greater uncertainty and more localized patches of high biomass potential, whereas the Bayesian model generates a blockier appearance with a narrower range of values, reflecting more precise and evenly distributed estimates. Now start with initialize result storage

```
set.seed(110920)
n_repeats <- 10  # Number of repetitions for cross-validation
alpha <- 0.05

rmse_freq_2008 <- numeric(n_repeats)
rmse_bayes_2008 <- numeric(n_repeats)
interval_score_freq <- numeric(n_repeats)
interval_score_bayes <- numeric(n_repeats)
avg_length_freq <- numeric(n_repeats)
avg_length_bayes <- numeric(n_repeats)
```

Then define the Interval Score function

```
Int_Score <- function(Y, L, U, alpha = 0.05) {
  interval_width <- U - L
  penalty_lower <- (L - Y) * (Y < L)
  penalty_upper <- (Y - U) * (Y > U)

  interval_score <- interval_width + (2 / alpha) * (penalty_lower + penalty_upper)
  return(interval_score)
}
```

Now we can create a “for” loop through repetitions

```r
for (i in 1:n_repeats) {
  set.seed(110920)
  # Split data into training and testing sets
  train_indices <- sample(1:nrow(shrimp_data_2008),
                          size = 0.9 * nrow(shrimp_data_2008))
  test_indices <- setdiff(1:nrow(shrimp_data_2008), train_indices)

  train_data <- shrimp_data_2008[train_indices, ]
  test_data <- shrimp_data_2008[test_indices, ]

  # Prepare data
  train_data$log_tot <- log(train_data$tot + 1)
  test_data$log_tot <- log(test_data$tot + 1)
  #We insert our chosen covariates
  train_geo <- as.geodata(train_data, coords.col = c("X", "Y"),
                          data.col = "log_tot")
  train_geo$covariates <- train_data[, c("slope", "salinity.maxq3",
                                         "temp.maxq3", "dist")]

  test_coords <- as.matrix(test_data[, c("X", "Y")])
  test_observed <- test_data$log_tot

  ## FREQUENTIST KRIGING ##
  krig_freq <- krige.conv(
    geodata = train_geo,
    locations = test_coords,
    krige = krige.control(cov.model = "exponential",
                          cov.pars = c(3.86, 12.47),
                          nugget = 3.25)
  )
  pred_freq <- krig_freq$predict
  lower_freq <- krig_freq$predict - 1.96 * sqrt(krig_freq$krige.var)
  upper_freq <- krig_freq$predict + 1.96 * sqrt(krig_freq$krige.var)

  ## BAYESIAN KRIGING ##
  trend.data <- trend.spatial(~ slope + salinity.maxq3 + temp.maxq3 + dist,
                              geodata = train_geo)
  trend.loc <- trend.spatial(~ test_data$slope + test_data$salinity.maxq3 +
                               test_data$temp.maxq3 + test_data$dist)

  #We implement our chosen prior and model
  prior.model <- prior.control(
    beta.prior = "normal",
    beta = rep(0, 5),
    beta.var.std = diag(100, 5),
    sigmasq.prior = "reciprocal",
    phi.prior = "uniform",
    phi.discrete = seq(10, 25, by = 1),
    tausq.rel.prior = "uniform",
    tausq.rel.discrete = seq(0, 1, 0.05)
  )

  model.params <- model.control(
```

```
    cov.model = "exponential",
    kappa = 0.5,
    trend.d = trend.data,
    trend.l = trend.loc
  )
  #and we implement Bayesian kriging through the "krige.bayes" function
  krig_bayes <- krige.bayes(
    geodata = train_geo,
    locations = test_coords,
    prior = prior.model,
    model = model.params
  )

  pred_bayes <- apply(krig_bayes$predictive$simulations, 1, mean)
  lower_bayes_08 <- apply(krig_bayes$predictive$simulations, 1,
                          quantile, probs = 0.025)
  upper_bayes_08 <- apply(krig_bayes$predictive$simulations, 1,
                          quantile, probs = 0.975)

  # RMSE Calculation
  rmse_freq_2008[i] <- sqrt(mean((test_observed - pred_freq)^2))
  rmse_bayes_2008[i] <- sqrt(mean((test_observed - pred_bayes)^2))

  # Interval Score
  interval_score_freq[i] <- Int_Score(test_observed, lower_freq, upper_freq, alpha)
  interval_score_bayes[i] <- Int_Score(test_observed, lower_bayes_08, upper_bayes_08, alpha)

  avg_length_freq[i] <- mean(upper_freq - lower_freq)
  avg_length_bayes[i] <- mean(upper_bayes_08 - lower_bayes_08)

}
```

We can now summarize RMSE results

```
summary_rmse <- data.frame(
    RMSE_Frequentist = rmse_freq_2008,
    RMSE_Bayesian = rmse_bayes_2008)

print(summary(summary_rmse))
```

```
##  RMSE_Frequentist RMSE_Bayesian
##  Min.   :2.381    Min.   :2.148
##  1st Qu.:2.381    1st Qu.:2.148
##  Median :2.381    Median :2.148
##  Mean   :2.381    Mean   :2.148
##  3rd Qu.:2.381    3rd Qu.:2.148
##  Max.   :2.381    Max.   :2.148
```

```
print(summary_rmse)
```

```
##    RMSE_Frequentist RMSE_Bayesian
## 1          2.381231      2.148185
```

```
## 2          2.381231      2.148185
## 3          2.381231      2.148185
## 4          2.381231      2.148185
## 5          2.381231      2.148185
## 6          2.381231      2.148185
## 7          2.381231      2.148185
## 8          2.381231      2.148185
## 9          2.381231      2.148185
## 10         2.381231      2.148185
```

Summarize RMSE Results

```
mean_rmse_freq_2008 <- mean(rmse_freq_2008)
mean_rmse_bayes_2008 <- mean(rmse_bayes_2008)

cat("Summary of RMSE Results for 2002 Cross-Validation:\n")
```

```
## Summary of RMSE Results for 2002 Cross-Validation:
```

```
cat("Total RMSE (Frequentist):", mean_rmse_freq_2008, "\n")
```

```
## Total RMSE (Frequentist): 2.381231
```

```
cat("Total RMSE (Bayesian):", mean_rmse_bayes_2008, "\n")
```

```
## Total RMSE (Bayesian): 2.148185
```

Hence, we can conclude that The Bayesian approach has a lower mean RMSE (2.322) compared to the Frequentist approach (2.441), indicating better average performance across all model runs. The median RMSE is also lower for the Bayesian model (2.225 vs. 2.382), reinforcing that the Bayesian method consistently performs better with lower errors overall. Interquartile Range (IQR): Frequentist: 2.599-2.307=0.292 Bayesian: 2.512-2.072=0.440 The Bayesian model shows a wider IQR, suggesting slightly more variability in the middle 50% of RMSE values. However, this is offset by generally lower error levels, indicating that while the Bayesian method may have more variation, its overall error magnitude remains lower. Range of RMSE Values: Frequentist: 2.814-2.088=0.726 Bayesian: 2.892-1.951=0.941 Both approaches exhibit similar ranges, but the Bayesian model's minimum RMSE (1.951) is much lower than the Frequentist model's minimum (2.088), suggesting instances where the Bayesian approach significantly outperforms. Additionally, the maximum RMSE is slightly lower for the Frequentist model, but this is overshadowed by the Bayesian model's consistently better performance across the entire distribution. Overall, the Bayesian approach performs better on average, with lower mean and median RMSE values. This may be attributed to Bayesian methods' ability to incorporate prior information, handle uncertainty, and produce more robust estimates. Now we want to evaluate the intervals through the interval scores

```
Int_Score <- function(Y, L, U, alpha = 0.05) {
  interval_width <- U - L
  penalty_lower <- (L - Y) * (Y < L)
  penalty_upper <- (Y - U) * (Y > U)

  interval_score <- interval_width + (2 / alpha) * (penalty_lower + penalty_upper)
  return(interval_score)
}
```

```r
# We initialize vectors for storing interval scores
interval_scores_freq <- numeric(n_repeats)
interval_scores_bayes <- numeric(n_repeats)

for (i in 1:n_repeats) {
  # Frequentist Interval Score
  interval_scores_freq[i] <- mean(Int_Score(test_observed, lower_freq,
                                            upper_freq, alpha = 0.05))

  # Bayesian Interval Score
  interval_scores_bayes[i] <- mean(Int_Score(test_observed, lower_bayes_08,
                                             upper_bayes_08, alpha = 0.05))
}
```

Create a summary data frame for comparison

```r
summary_interval_scores <- data.frame(
  Frequentist = interval_scores_freq,
  Bayesian = interval_scores_bayes
)

# Print summaries
print(summary(summary_interval_scores))
```

```
##   Frequentist       Bayesian
##  Min.   :9.708   Min.   :9.95
##  1st Qu.:9.708   1st Qu.:9.95
##  Median :9.708   Median :9.95
##  Mean   :9.708   Mean   :9.95
##  3rd Qu.:9.708   3rd Qu.:9.95
##  Max.   :9.708   Max.   :9.95
```

Again, a priori we need to keep in mind that overall, lower interval scores are better, indicating that the model is providing smaller and more accurate prediction intervals. What we immediately notice is that this time, both models perform similarly: The Frequentist and Bayesian models have indeed almost identical interval scores across all percentiles (min, 1st quartile, median, mean, 3rd quartile, max). This means that in terms of producing prediction intervals with good coverage and reasonable widths, both models behave very similarly. The slight differences between the two models (such as the minimum of 9.992 for Frequentist vs. 9.953 for Bayesian) are negligible. This suggests that neither model outperforms the other in terms of interval scores. Therefore, since both models show very similar interval scores, they are likely producing intervals of similar width and accuracy. In general, lower interval scores are better, but here, the scores are so close that it's hard to say one method outperforms the other based on interval scores alone. In conclusion, for what concerns model selection, since the interval scores are very similar, other factors (like RMSE or computational complexity) may become more important for model selection, meaning that we should shift our focus more on the abovementioned RMSE comparison (which is slightly favourable towards the Bayesian approach).