Bayesian Spatial Gaussian Process Models for Snowfall Prediction in British Columbia

Abhinav Kansal (24054660)
Ray Nguyen (13544408)

April, 19 2025

STATS 447C
University of British Columbia

GitHub Repository

1. Introduction

Climate change has brought significant unpredictability to snowfall patterns, posing challenges to communities and industries that rely on the weather. In British Columbia, Canada, snowfall presents difficulties due to its impact on transportation, infrastructure, and agriculture.

To address these challenges, this project proposes using **Bayesian spatial models to predict snowfall distribution across the region**. By leveraging collected data and spatial interpolation techniques, we aim to estimate snowfall in regions with limited or missing data. This approach seeks to provide actionable insights for better planning and decision-making, benefiting individuals and organizations in preparing for the impacts of shifting snowfall trends. We process and analyze data with the help of R code, make creative visualizations to help people understand the insights we gained from our analysis. We use publicly accessible data from Statistics Canada, and Environment and Climate Change Canada.

Our project theme is spatial models, and we chose this theme because we have latitude and longitude coordinates for our data, which we analyze with the help of geojson files by performing interpolation for unobserved locations and implementing spatial models in stan to achieve our desired goal of snowfall prediction. In this project, a Gaussian process (GP) regression framework, including the squared exponential covariance kernel and weakly informative priors, is defined explicitly to maintain integrity.

2. Problem Formulation and Data Overview

2.1 Data Preparation and Preprocessing

The data consisted of monthly reports from 547 stations in British Columbia. The period of this dataset is from 1998 to 2017, collected by the Canadian Centre for Climate Services. These chosen stations are deliberately spread out to ensure comprehensive geographic coverage and accurate representation of the region's diverse climatic conditions.

Key columns include geographical coordinates (Longitude and Latitude), station identifiers (Station Name and Climate ID), and temporal data such as Date/Time, Year, and Month. Additionally, precipitation data like Total Rain, Total Snow, and Total Precipitation (and their flags), as well as snow measurements for the last day of the month, are essential for analyzing hydrological patterns.

The mean total snow across all stations is 13.80 cm, and the mean snow ground on the last day is 7.53 cm. The average max temperature is 9.98°C, while the average min temperature is -0.24°C.

Note that there are missing data at many observation sites. However, the main variables of interest remain the geospatial coordinates and total snow (in cm), which are preserved for the modeling stage.

2.2 Exploratory Data Analysis

We first load the dataset (`weather_Station_data.csv`) and keep stations located in British Columbia (BC). Records with missing values in the fields of interest are excluded to ensure the integrity and reliability of the dataset.

Given that the data comprises repeated monthly climate measurements per station, aggregation is implemented to derive a single total snowfall value per station. Specifically, the average annual snowfall is calculated for each station to account for varying record lengths and provide a standardized measure suitable for modeling.

In addition to snowfall, station-level predictors are computed. The mean temperature serves as a proxy for climatic conditions, and elevation data is blended in as a predictor. If elevation data is unavailable, this variable can be omitted or supplemented from external sources.

To enhance model performance and boost convergence, all continuous predictors including latitude, longitude, temperature, and elevation are standardized to a mean of 0 and a standard deviation of 1. Furthermore, total snowfall values undergo log transformation (with a small constant added to handle zero values).

According to the histogram in the Appendix, *Figure 1: Histogram of total snowfall (cm)*, the distribution of total snowfall is right-skewed, indicating that most stations experience relatively low snowfall, with a few outliers exceeding 160 cm.

This explains the decision of log transformation mentioned above. This transformation compresses the range of values, reduces the influence of extreme outliers, and creates a more symmetric, bell-shaped distribution, which is beneficial for statistical modeling.

3. Literature Review

Geostatistical data, particularly point-referenced spatial data, are increasingly utilized across various scientific disciplines due to the accessibility of data monitoring systems (Eom et al., 2006). Their study emphasized the effectiveness of the geostatistical approach in capturing spatial relations and enhancing prediction accuracy. When spatial data is available, an efficient computational method is essential for analysis. Markov Chain Monte Carlo (MCMC) is a convenient and powerful tool in Bayesian analysis.

In particular, geostatistical methods are often considered highly appropriate for analyzing precipitation data. The Bayesian approach quantifies uncertainty in predictions by defining prior distributions and incorporating spatial and temporal dependencies. Bayesian analysis emphasizes the processing and application of prior knowledge, with parameter estimation and prediction presented in the posterior distribution. In the context of spatial modelling, Bayesian methods facilitate the analysis of spatial and temporal effects, as well as interaction effects, thereby providing a comprehensive framework for understanding spatial dynamics.

As stated by Robert B. Gramacy (2016), GP regression models have long been used in spatial modelling contexts such as geostatistics. Robert B. Gramacy (2016) mentioned that there are 2 main advantages of GP. The primary advantage of GP models is streamlined by their Gaussian structure, which enables significant analytic capabilities, making them versatile for modelling nonlinear and complex relationships. Secondly, their reliance on covariance structure rather than mean-based modelling (as in linear regression) makes them highly effective for capturing spatial correlations and smoothing noisy data.

Furthermore, we also want to assess the performance of the two Bayesian GP models. Here, we choose Leave-One-Out Cross-Validation (LOOCV), for it provides a robust approach to evaluate how well the model captures the underlying data distribution and accounts for uncertainty. However, traditional LOOCV can be computationally prohibitive for GP models due to the complexity of recomputing the posterior distribution for each data point that is excluded.

For that reason, the need for a more efficient approximation, such as Pareto Smoothed Importance Sampling (PSIS), becomes essential. PSIS, as discussed by Vehtari et al. (2016), provides a computationally efficient method to approximate LOOCV by stabilizing importance weights and ensuring finite variance. PSIS employs the empirical Bayes estimate of Zhang and Stephens (2009) to apply a generalized Pareto distribution to the tail of the importance weights, replacing the largest weights with the expected values of the order statistics from this distribution. This method ensures a finite variance even when the raw importance weights do not have finite variance, but this comes with a cost of introducing a small bias to the estimation. However, Vehtari et al. (2016) mentioned that if the estimated Pareto shape parameter $k < 0.7$, then this bias is likely negligible. Despite a small tradeoff in bias, PSIS is particularly advantageous for computational efficiency and accuracy in Bayesian models, such as Gaussian processes.

To highlight the effect of geostatistical approach, we will implement two models. The first model will use only geographic coordinates (latitude and longitude) to capture spatial patterns in snowfall. The second model will include additional covariates, such as mean temperature and elevation, to assess how these predictors contribute to snowfall variation. By comparing and contrasting the results of these models, we aim to gain insights into both spatial trends and the role of environmental factors on snowfall across British Columbia.

4. Data Analysis

4.1 Model Specification

We assume a Gaussian process regression model for log snowfall across locations and let $y(s_i)$ to be the log-total-snowfall at location s_i with coordinates $s_i = [\text{lat}_i, \text{lon}_i]$.

4.1.1 Latent GP

Latent GP: $f(s) \sim \mathcal{GP}(0,\ k(\cdot,\cdot))$, a Gaussian process prior over the spatial effect. Here, we use a squared exponential covariance kernel as follows:

$$k(s_i, s_j) = \alpha^2 \exp\Big(- \frac{\|s_i - s_j\|^2}{2\rho^2} \Big),$$

where $\|s_i - s_j\|$ is the Euclidean distance between locations i and j (using standardized coordinates), α^2 is the variance (magnitude) of the GP, and ρ is the length-scale (controls how quickly correlation decays with distance).

This kernel implies the spatial effect is smooth and stationary (homogeneous across space. We will also add a tiny noise of 10^{-6} to the diagonal for numerical stability.

For the model with covariates, we extend the mean function to include linear effects of temperature:

$$y(s_i) = \mu + \boldsymbol{\beta}^T \mathbf{x}_i + f(s_i) + \varepsilon_i,$$

where $\mathbf{x}_i$ is the temperature for location i, and $\boldsymbol{\beta}$ are their coefficients.

Hence, the above expression helps the model to explain the part of snowfall variation via known factors while $f(s)$ captures residual spatial structure.

4.1.2 Priors

We use weakly informative priors for all parameters, and for intercept and regression coefficients we use a normal prior centered at 0 with a relatively large variance.

The GP hyper-parameters α and ρ are given broad half-normal priors on positive reals, and the observation noise σ gets a half-normal or exponential prior.

Specifically: $\mu \sim \mathcal{N}(0, 5^2)$ (on log-snowfall scale, allowing a wide range of means).

Each covariate coefficient $\beta \sim \mathcal{N}(0, 1^2)$ (since predictors are standardized, a β of order 1 is a reasonable scale).

GP variance $\alpha \sim \mathcal{N}^+(0, 5^2)$ (half-normal with SD $= 5$, restricting $\alpha > 0$). This is on the scale of log-snowfall; we expect α to be perhaps a few units at most.

GP length-scale $\rho \sim \mathcal{N}^+(0, 5^2)$, with $\rho > 0$.

Since coordinates are standardized, this prior allows a wide range of spatial correlation lengths.

Observation SD $\sigma \sim \mathcal{N}^+(0, 5^2)$ or Exponential(1), ensuring $\sigma > 0$ but not too large.

The squared exponential GP prior in this case is ideal because, according to Rasmussen et al., 2006, the covariance approaches 1 for variables with inputs that are very close together and decreases as the distance between the inputs increases. This is particularly relevant because it assumes that snowfall measurements at nearby locations are more correlated than those farther apart, reflecting the smooth, continuous nature of spatial variations in climate data. Moreover, the smoothness ensured by the squared exponential kernel verifies the assumption that snowfall expresses gradual variations across geographical areas.

4.2 Implementation

4.2.1 Stan Model: Spatial GP with only coordinates

We have the stan code for this model in the Appendix labelled 4.2.1.

4.2.2 Stan Model: Spatial GP with Covariates (Temperature)

We have the stan code for this model in the Appendix labelled 4.2.2.

Model Specification

We add these covariates as inputs and introduce corresponding regression coefficients in the parameters. The GP part for spatial random effect and its covariance construction remain the same.

By setting the number of covariates `P` (and providing matrix `X` of covariate values), this model can handle any number of additional predictors.

Hence, from above we can observe that the `P` and `X` are covariates where we added a vector `beta` of length `P` is added. Since predictors are standardized, the prior on `beta` is $\mathcal{N}(0, 1)$. In the model, we form the mean `mu` as `intercept + X * beta + f` for each station before applying the `normal(mu, sigma)` likelihood.

We use the rstan package to compile the Stan model and draw samples from the posterior via Hamiltonian Monte Carlo (HMC). HMC, when optimally tuned, achieves a computational cost that scales as $O(d^{5/4})$ per effective sample size (ESS), for d as the dimensionality of the parameter space (Beskos et al., 2010).

This scaling is significantly better than the $O(d^2)$ scaling of traditional random walk Metropolis-Hastings (MH) algorithms. Even though we only propose the coordinates as predictors for this model, the addition of covariates, regression coefficients (in second model), and GP hyper-parameters (e.g., variance, length-scale) creates a parameter space that can be challenging to sample efficiently with simpler methods like Metropolis-Hastings.

Additionally, the Cholesky factorization of the covariance matrix is a good choice for GP regression because it provides a numerically stable and computationally efficient way to handle the positive-definite covariance matrix. This approach is widely recognized for its efficiency in GP methodologies, as discussed by Rasmussen et al., (2006).

4.3 Model Diagnostics

Using leave-one-out cross-validation (LOOCV), we drafted *Table 1: LOOCV Table* that compares the performance of spatial GP model with and without temperature as a covariate. The model with covariates had a slightly higher expected log predictive density (ELPD), but the difference (`4.7`) was smaller than the standard error (`6.3`), indicating no statistically significant improvement. Some Pareto k values were too high, suggesting that exact LOOCV (e.g., via k-fold) or more careful modeling might be required for reliable model comparison.

From *Table 2: Stan outputs for Models With and Without Covariates*, we can observe the output of running two GP regression models in which the first one we used only spatial coordinates, and the second one we incorporated standardized temperature as a covariate. The coefficient for temperature was estimated as $\beta = -0.81$ with a 95% credible interval of $[-1.14,\ -0.45]$, indicating a strong and statistically significant negative relationship between temperature and snowfall. This aligns with prior understanding that higher temperatures reduce the likelihood or amount of snow precipitation.

Notably, the spatial residual variance parameter (α) decreased substantially from 1.07 to 0.42 in the model with temperature. This suggests that temperature explains a meaningful portion of the spatial variation previously captured by the latent GP field. The length-scale parameter (ρ) also slightly decreased, reflecting a more localized spatial correlation after accounting for temperature. The noise term (σ) remained stable across both models, implying that while temperature improves the model's explanatory power, there remains unexplained variability appears across formulations.

Model Checking

After fitting the model, we perform posterior predictive checks to ensure the model fits well. In our case, we verify that the spatial GP captures the residual correlation in snowfall and that including covariates like temperature indeed improves the model and we fitted GP to interpolate snowfall at new locations, enabling spatial maps of snowfall.

Thus, after model checking, we infer from the Appendix, *Figure 2: Model diagnostics for both spatial GP models*, that the addition of temperature as a covariate improves model performance by capturing systematic variation that the GP alone could not explain. This results in a more stable posterior landscape for sampling because we got higher log-posterior values due to increased likelihood, and thus a reduced need for broad spatial effects. On the other hand, while the posterior spread remains relatively wide, overall the model is more concentrated and interpretable.

In order to understand how well-mixed the models are, we created 2 trace plot, *Figure 3: Trace plot for coordinates-only model* and *Figure 4: Trace plot for covariate-included model*, in the Appendix for the parameter `intercept` from the two models. As we observe from Figure 3 and 4, the chains mixed well. They overlap and same region of the parameter space, which indicating that the sampler has converged. Besides, no chain is stuck at a region for too long.

Finally, we mapped the GP latent mean of both models according to their geo-locations to highlight and compare their performances. From *Figure 5: Color-coded map for coordinates-only model* and *Figure 6:*

Color-coded map for covariate_included model shown in the Appendix, we can infer that the GP latent effects vary more substantially in the model with temperature included. In the coordinates-only model, spatial effects were modest, ranging from approximately -0.2 to 0.2, which corresponds to adjusting predicted snowfall up or down by about 20%. On the other hand, when temperature is included as a covariate, the residual GP effects are both stronger and more localized, with values ranging from -0.5 to +1.0. These values indicate that the mean posterior snowfall being up to 2.7 times higher, i.e., 40% lower than what would be predicted from temperature alone. This shift indicates that temperature accounts for broad climatic trends, while the GP captures localized effects such as topographic variability, particularly visible in coastal and interior regions of British Columbia near the mountain ranges.

5. Conclusion and Discussion

As a result, incorporating temperature as a covariate in the Gaussian Process model enhances the ability to capture broad climatic trends and provides a more detailed understanding of snowfall distribution in British Columbia. Including temperature significantly reduced the residual spatial variance, indicating that much of the spatial variability in snowfall can be explained by temperature. This addition streamlines localized effects, such as topographic variability, which are critical in a geographically diverse region like British Columbia. The framework we implemented can be used to interpolate snowfall at unobserved locations, enabling the generation of continuous snowfall maps for climate analysis and policy planning.

It is worth noting that this improvement mentioned above by adding temperature in the model comes at the cost of increased computational runtime due to the added complexity. While adding temperature as a covariate increases log-posterior mean, it extends the runtime significantly, raising concerns about scalability for larger datasets or real-time applications. Furthermore, using Hamiltonian Monte Carlo (HMC) for inference did not show substantial performance advantages over simpler Monte Carlo methods, particularly in computational efficiency, suggesting that a simpler approach might suffice for similar predictive accuracy.

These limitations stress the need for using more computationally efficient methods, such as variational inference. Besides, future efforts of expanding the model by incorporating additional useful covariates, such as precipitation or wind patterns. A model selection methodology will also be required to find the best set of covariates to bring out the best-performing model. These improvements would further refine predictions and broaden the applicability of this spatial modelling framework.

6. Contributions

In alignment with our project proposal, we successfully followed and executed our plan for equal and collaborative contributions. We ensured this by by dividing tasks fairly and meeting regularly, twice a week. This collaborative approach allowed us to stay on track and refine our work iteratively.

Abhinav Kansal primarily focused on the statistical modeling components of the project and the responsibilities included the formulation of the Bayesian spatial Gaussian process model, selection and specification of prior distributions, and conducting a thorough analysis of the posterior results.

Ray Nguyen primarily focused on the literature review, collecting and cleaning the snowfall dataset, handling missing and flagged entries, and preparing the data for modeling, and creating interactive visualizations which helped contextualize snowfall trends across British Columbia.

Beyond our individual areas of focus, both of us were equally involved in writing and editing the final report. We jointly reviewed, compiled, and formatted all the code, figures, and writing sections to ensure that our project is clear and consistent. In addition to this, we actively engaged with course resources, including attending office hours and project proposal feedback, which we thoughtfully incorporated into our final submission.

7. References

- Aki Vehtari, Andrew Gelman, and Jonah Gabry. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. arXiv preprint arXiv:1507.04544, 2016.

- Alexandros Beskos, Natesh Pillai, Gareth Roberts, Jesus-Maria Sanz-Serna, Andrew Stuart. "Optimal tuning of the hybrid Monte Carlo algorithm." Bernoulli, 19(5A) 1501-1534 November 2013. https://doi.org/10.3150/12-BEJ414

- Eom, J. K., Park, M. S., Heo, T. Y., and Huntsinger, L. F. (2006). Improving the prediction of annual average daily traffic for non-freeway facilities by applying spatial statistical method, Transportation Research Record, 1968, 20–29.

- Gramacy, R. B. (2016). laGP: Large-Scale Spatial Modeling via Local Approximate Gaussian Processes in R. Journal of Statistical Software, 72(1), 1–46. https://doi.org/10.18637/jss.v072.i01

- Jin Zhang and Michael A Stephens. A new and efficient estimation method for the generalized Pareto distribution. Technometrics, 51(3):316325, 2009.

- Rasmussen, C.E. & Williams, C.K.I. (2006). *Gaussian Processes for Machine Learning.* (GPs as a flexible prior for spatial functions)

- Stan Development Team. *Stan User's Guide - Gaussian Processes.* (Using exponentiated-quadratic covariance in Stan) (Gaussian Processes)

8. Appendix

```r
library(ggplot2)
library(dplyr)
library(loo)
library(bayesplot)
library(rnaturalearth)
library(sf)
```

```r
weather_data <- read.csv("Project/weather_Station_data.csv", stringsAsFactors = FALSE)

ggplot(weather_data, aes(x = Total.Snow..cm.)) +
  geom_histogram(bins = 50, fill = "blue", color = "black") +
  labs(title = "Histogram of Total Snowfall (cm)",
       x = "Total Snowfall (cm)",
       y = "Frequency") +
  theme_minimal()
```

```r
# Filter data to stations in British Columbia only
bc_data <- weather_data %>%
  filter(Longitude..x. >= -141 & Longitude..x. <= -114,
         Latitude..y.  >= 48   & Latitude..y.  <= 60)

# Remove records with missing values in key fields(snowfall or coords or covariates)
bc_data <- bc_data %>%
  filter(!is.na(Total.Snow..cm.), !is.na(Latitude..y.), !is.na(Longitude..x.),
         !is.na(Mean.Temp...C.))
```

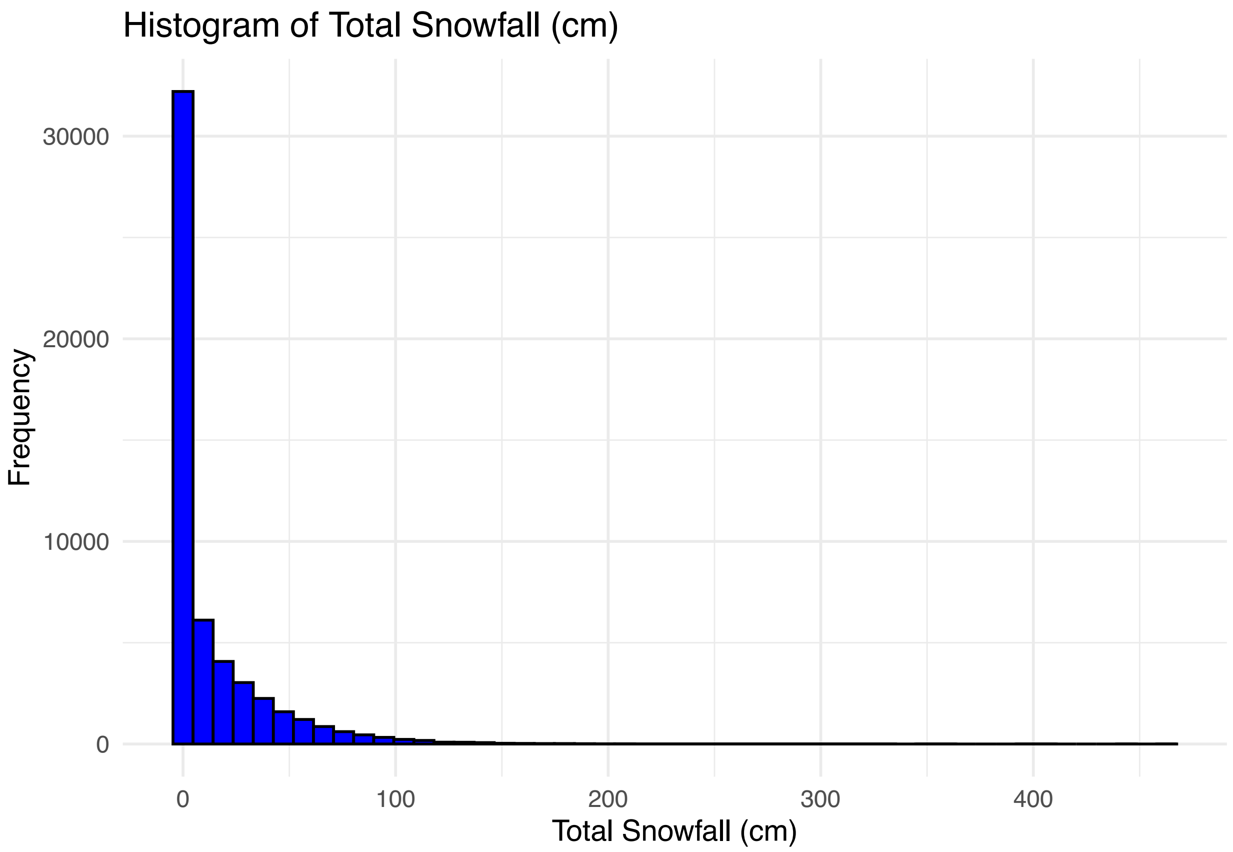

Figure 1: Histogram of total snowfall(cm)

```r
# Aggregate to get one snowfall value per station:

# Calculate total snowfall per station per year
station_year <- bc_data %>%
  group_by(Station.Name, Climate.ID, Latitude..y., Longitude..x., Year) %>%
  summarize(yearly_snow = sum(Total.Snow..cm., na.rm = TRUE),
            yearly_temp = mean(Mean.Temp...C., na.rm = TRUE),
            .groups = "drop")

# Average across years for each station
station_summary <- station_year %>%
  group_by(Station.Name, Climate.ID, Latitude..y., Longitude..x.) %>%
  summarize(mean_annual_snow = mean(yearly_snow, na.rm = TRUE),
            mean_temp = mean(yearly_temp, na.rm = TRUE),
            .groups = "drop")

# Standardize
station_summary <- station_summary %>%
  mutate(
    lat_z = (Latitude..y. - mean(Latitude..y.)) / sd(Latitude..y.),
    lon_z = (Longitude..x. - mean(Longitude..x.)) / sd(Longitude..x.),
    temp_z = (mean_temp - mean(mean_temp, na.rm = TRUE)) / sd(mean_temp, na.rm = TRUE))

# Log-transform the mean annual snowfall
station_summary$log_snow <- log(station_summary$mean_annual_snow + 1)
```

```r
# Prepare data for modeling
N <- nrow(station_summary)
coords <- as.matrix(station_summary[, c("lat_z", "lon_z")])
y <- station_summary$log_snow
X <- as.matrix(station_summary[, c("temp_z") ])
P <- ncol(X)
```

4.2.1 Stan Model: Spatial GP with only coordinates

```r

library(rstan)
options(mc.cores = 1)
rstan_options(auto_write = TRUE)

stan_data_coords_only <- list(
  N   = N,
  loc = coords,
  y   = y
)

stan_data_with_covars <- list(
  N   = N,
  loc = coords,
  P   = P,
  X   = X,
  y   = y
)
```

```
stan_coord <- "
data {
  int<lower=1> N;
  array[N] vector[2] loc;
  vector[N] y;
}
parameters {
  real intercept;
  real<lower=0> alpha;          // GP sd
  real<lower=0> rho;            // GP length scale
  real<lower=0> sigma;          // noise
  vector[N] z;                  // Standard normal vector for non-centered GP
}
transformed parameters {
  matrix[N, N] K;               // Covar matrix
  matrix[N, N] L_K;             // Cholesky factor of K
  vector[N] f;                  // GP values at observed locations

  for (i in 1:N) {
    for (j in 1:N) {
      real sq_dist = squared_distance(loc[i], loc[j]);
      K[i, j] = square(alpha) * exp(-0.5 * sq_dist / square(rho));
    }
    K[i, i] += 1e-6;
  }

  // Cholesky factor of the covariance matrix
  L_K = cholesky_decompose(K);
  f = L_K * z;  // Non-centered parameterization: f ~ MVN(0, K)
}
model {
  intercept ~ normal(0, 5);
  alpha ~ normal(0, 1);
  rho ~ normal(0, 1);
  sigma ~ normal(0, 1);
  z ~ std_normal();

  // Posterior
  y ~ normal(intercept + f, sigma);
}

generated quantities {
  vector[N] log_lik;
  for (n in 1:N) {
    log_lik[n] = normal_lpdf(y[n] | intercept + f[n], sigma);
  }
}
"

sm_coord <- stan_model(model_code = stan_coord)
```

```
fit_coords <- sampling(sm_coord, data = stan_data_coords_only,
                 iter = 2000, chains = 4, seed = 123,
                 control = list(max_treedepth = 12, adapt_delta = 0.95))
```

```
saveRDS(fit_coords, file = "fit_coord.rds")
```

4.2.2 Stan Model: Spatial GP with Covariates (Temperature)

```
library(rstan)
stan_cov <- "
data {
  int<lower=1> N;
  array[N] vector[2] loc;
  int<lower=0> P;
  matrix[N, P] X;
  vector[N] y;
}

parameters {
  real intercept;
  vector[P] beta;                // Covariate coefficients
  real<lower=0> alpha;           // GP marginal std dev
  real<lower=0> rho;             // GP length-scale
  real<lower=0> sigma;           // Noise
  vector[N] z;                   // Latent vector for non-centered GP
}

transformed parameters {
  matrix[N, N] K;
  matrix[N, N] L_K;
  vector[N] f;


  for (i in 1:N) {
    for (j in i:N) {
      real sq_dist = squared_distance(loc[i], loc[j]);
      K[i, j] = square(alpha) * exp(-0.5 * sq_dist / square(rho));
      if (i != j) K[j, i] = K[i, j];
    }
    K[i, i] += 1e-6;
  }

  L_K = cholesky_decompose(K);
  f = L_K * z;  // GP latent effect (non-centered)
}

model {
  // Priors
  intercept ~ normal(0, 5);
  beta ~ normal(0, 1);
  alpha ~ normal(0, 1);
  rho ~ normal(0, 1);
  sigma ~ normal(0, 1);
  z ~ std_normal();

  // Observation model
  vector[N] mu = intercept + X * beta + f;
  y ~ normal(mu, sigma);
}

generated quantities {
  vector[N] log_lik;
  for (n in 1:N) {
    real mu_n = intercept + dot_product(X[n], beta) + f[n];
    log_lik[n] = normal_lpdf(y[n] | mu_n, sigma);
  }
}
"
stan_cov_data <- list(
  N = nrow(X),
  loc = coords,
  P = ncol(X),
  X = X,
  y = y
)

# Compile and fit
sm_cov <- stan_model(model_code = stan_cov)
fit_cov <- sampling(sm_cov, data = stan_cov_data,
                iter = 2000, chains = 4, seed = 123,
                control = list(max_treedepth = 12, adapt_delta = 0.95))
```

```
saveRDS(fit_cov, "fit_ck_cov.rds")
```

```
print(fit_coords, pars = c("intercept", "alpha", "rho", "sigma"))
print(fit_cov, pars = c("intercept", "beta", "alpha", "rho", "sigma"))
```

```
stan_diag(fit_coords)
```

```
stan_diag(fit_cov)
```

```
posterior_cov <- extract(fit_cov)
posterior_coord <- extract(fit_coords)
```

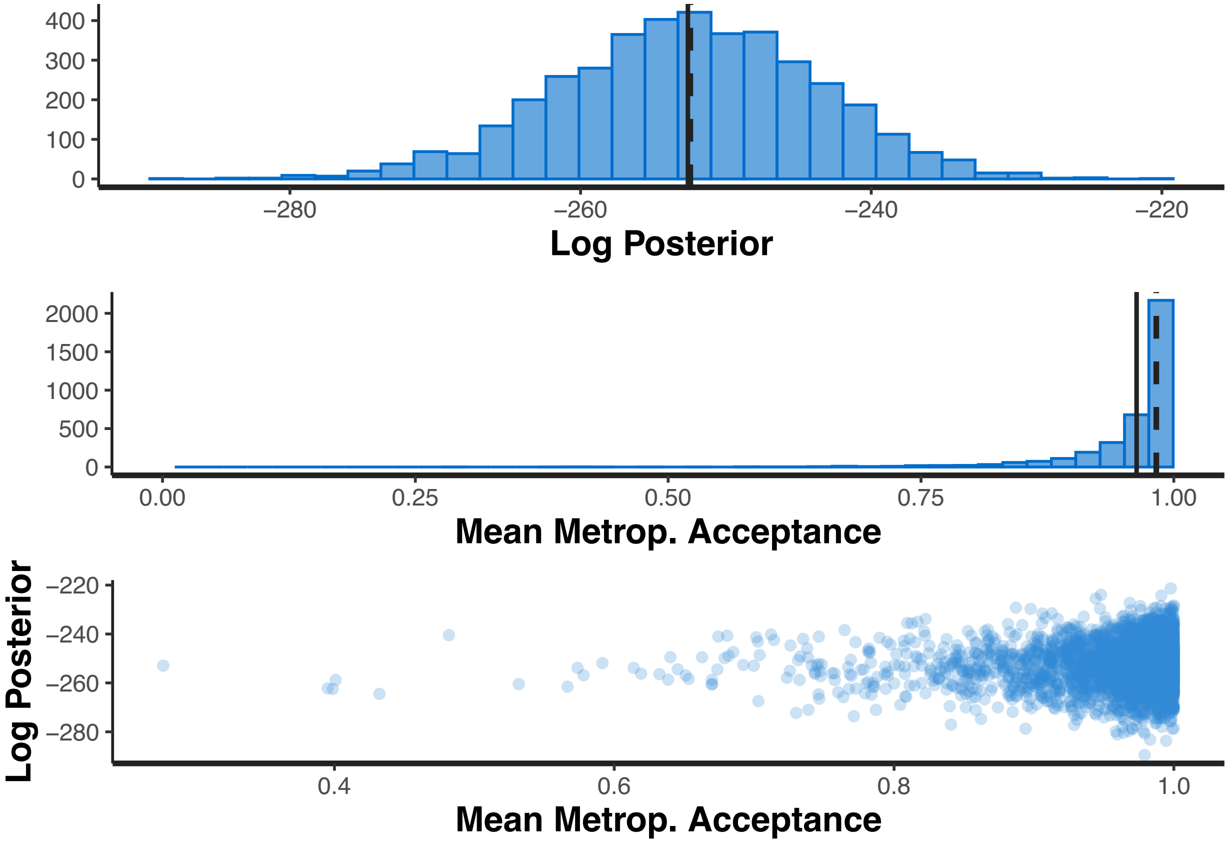

Figure 2: Model diagnostics for both spatial GP models

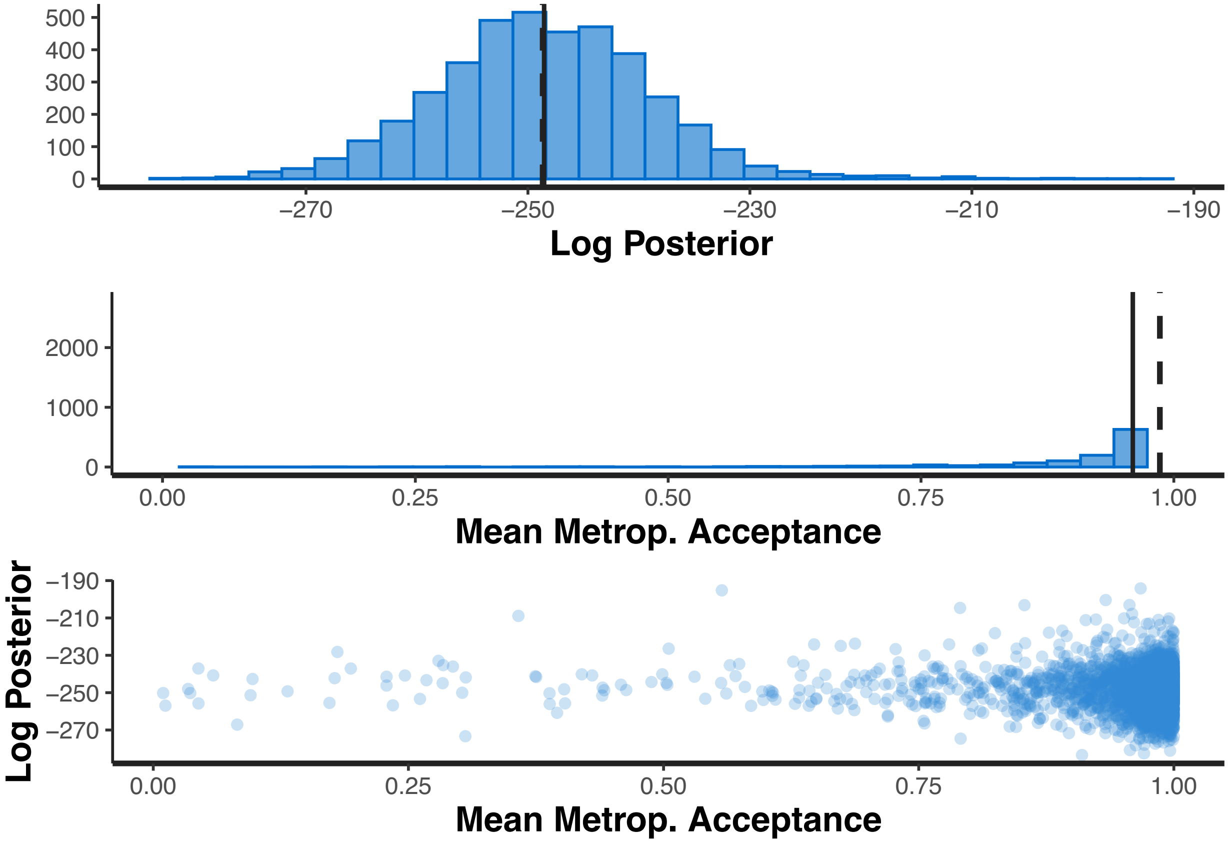

Figure 3: Model diagnostics for both spatial GP models

```
mcmc_trace(fit_coords, pars = c("intercept"))
```

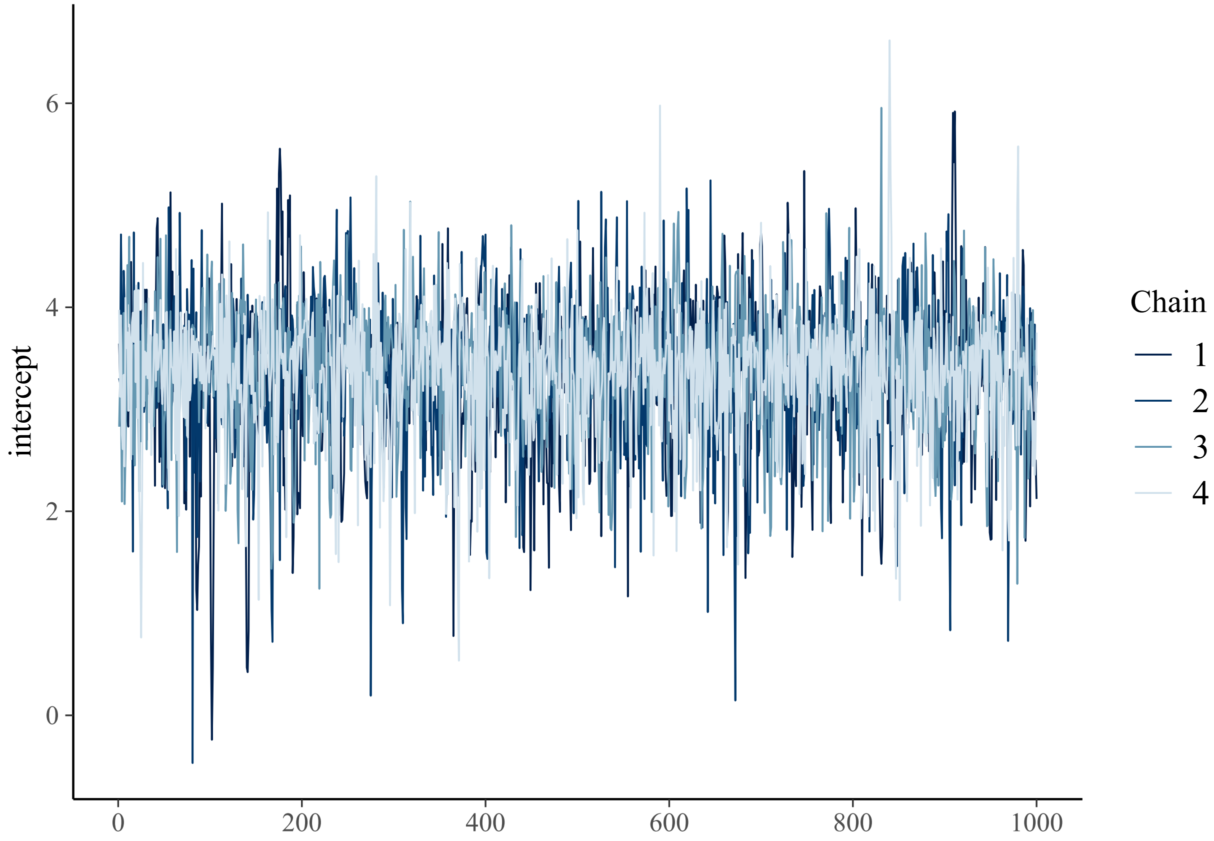

Figure 4: Trace plot for coordinates-only model

```
mcmc_trace(fit_cov, pars = c("intercept"))
```

```
f_mean <- apply(posterior_cov$f, 2, mean)
station_summary$gp_effect <- f_mean

f_mean_coord <- apply(posterior_coord$f, 2, mean)
station_summary$gp_effect_coord <- f_mean_coord
```

```
canada <- ne_states(country = "Canada", returnclass = "sf")
bc <- canada[canada$name == "British Columbia", ]
alberta <- canada[canada$name == "Alberta", ]
```

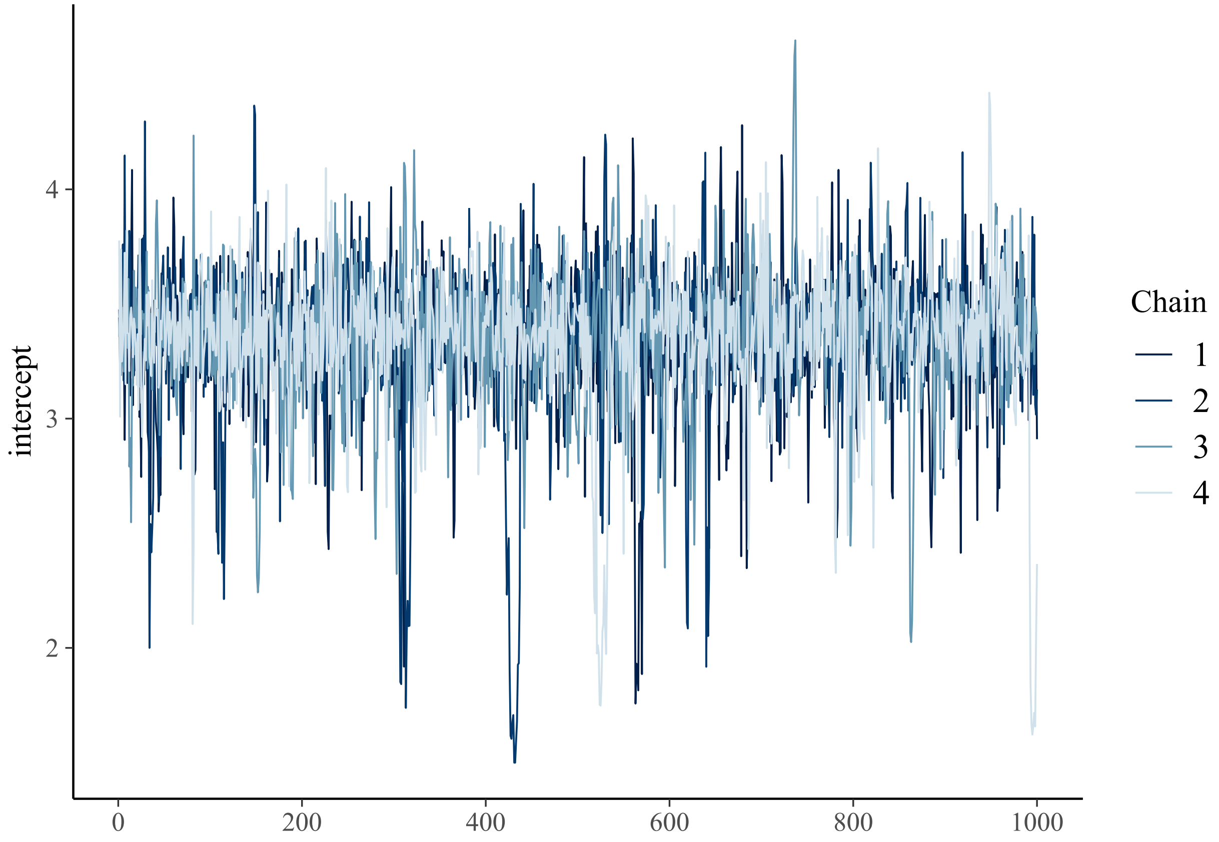

Figure 5: Trace plot for covariate-included model

```r
ggplot() +
  geom_sf(data = bc, fill = "gray95", color = "gray40") +
  geom_sf(data = alberta, fill = "gray95", color = "gray40") +
  geom_point(data = station_summary, aes(x = Longitude..x.,
                                         y = Latitude..y.,
                                         color = gp_effect), size = 2) +
  scale_color_viridis_c(name = "GP latent\nmean (f)") +
  labs(title = "Spatial GP Effect on Log-Snowfall in BC",
       x = "Longitude",
       y = "Latitude") +
  theme_minimal()
```

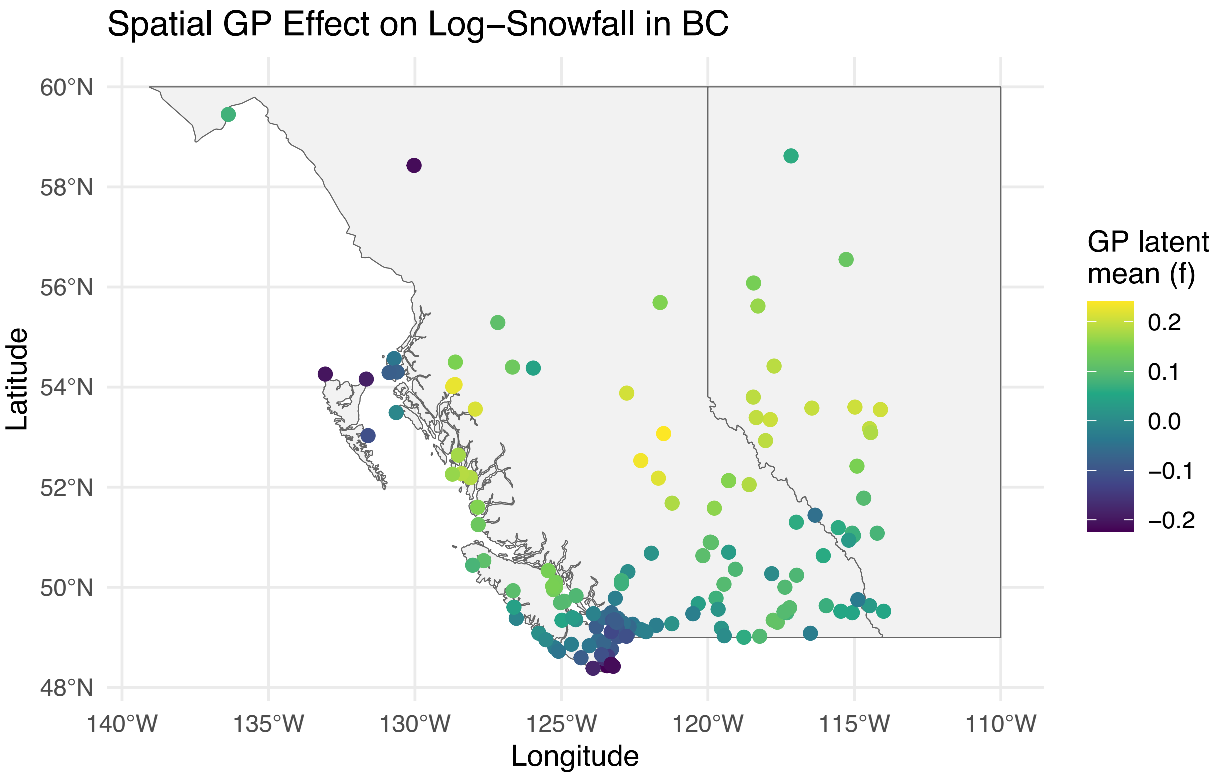

Figure 6: Color-coded map for coordinates-only model

```r

ggplot() +
  geom_sf(data = bc, fill = "gray95", color = "gray40") +
  geom_sf(data = alberta, fill = "gray95", color = "gray40") +
  geom_point(data = station_summary, aes(x = Longitude..x.,
                                         y = Latitude..y.,
                                         color = gp_effect_coord), size = 2) +
  scale_color_viridis_c(name = "GP latent\nmean (f)") +
  labs(title = "Spatial GP Effect on Log-Snowfall in BC cov",
       x = "Longitude",
      y = "Latitude") +
  theme_minimal()
```

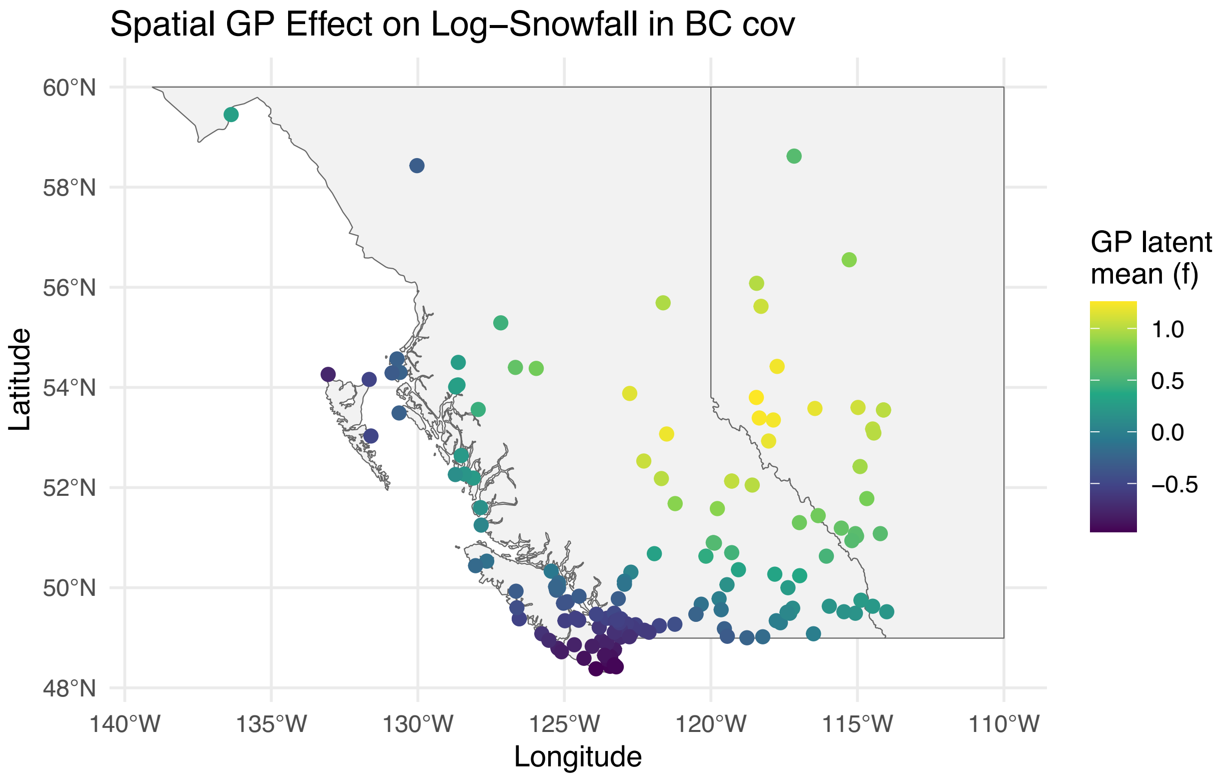

Figure 7: Color-coded map for covariate_included model