Appendix A

Owino Raymond

Load Required Libraries

```r
library(tidyverse)        # Data manipulation
library(ggplot2)      # Plotting
library(viridis)      # Color scales for plots
library(sf)           # Simple Features for spatial data
library(spmodel)      # Spatial linear model (for spatial prediction)
library(gstat)        # For semivariogram fitting and spatial analysis
library(tidyr)        # For reshaping and cleaning data (if necessary)
library(tigris)       # To load U.S. Census data
library(patchwork)
library(AICcmodavg)
```

Data Loading and Filtering

```r
# Load the data
load("data/wfigs_az_sf_EPSG32612.RData")

# convert pop.den as numeric
wfigs_az_sf$pop.density <- as.numeric(wfigs_az_sf$pop.density)

# Clean and select relevant columns, then rename for clarity
fire_size <- wfigs_az_sf %>%
  select(OBJECTID, FireDiscoveryDateTime, IncidentSize, FireCause,
         IncidentTypeCategory, POOCounty, x, y, tmax, tmin, prcp, mean_slope,
         mean_grass, mean_forest, mean_shrub, Temp_Max_Buffered,
         Temp_Min_Buffered, Precipitation_Buffered, Elevation,
         pop.density, pop., distance_rd_primary, distance_rd_min_all,
         distance_rd_secondary, distance_rd_4wd) %>%
  rename(
    ID = OBJECTID,
    Date = FireDiscoveryDateTime,
    Size = IncidentSize,
    Cause = FireCause,
    Category = IncidentTypeCategory,
    County = POOCounty,
    Long = x,
    Lat = y,
    Max_day_temp = tmax,
    Min_day_temp = tmin,
    Prcp = prcp,
    pSlope = mean_slope,
    Grass_p = mean_grass,
    Forest_p = mean_forest,
    Shrub_p = mean_shrub,
    Max_ann_temp = Temp_Max_Buffered,
    Min_ann_temp = Temp_Min_Buffered,
    Prcp_ann = Precipitation_Buffered,
    Elevation = Elevation,
    pop_density = pop.density,
    Population = pop.,
    Pri_rd = distance_rd_primary,
    All_rd = distance_rd_min_all,
    Sec_rd = distance_rd_secondary,
    Dist_4WD = distance_rd_4wd
  )

# Filter for wildfires and focus on human or natural causes
wild_fires <- fire_size %>%
  filter(Category == "WF") %>%
  filter(Cause == "Human" | Cause == "Natural")

# Filter fires in Coconino county with a size greater than or equal to 50 acres
coconino <- wild_fires %>%
  filter(County == "Coconino") %>%
  filter(Size >= 1000)
```

Visualize data in relation to Arizona

```
# Get Arizona state outline
az_outline <- states(cb = TRUE) %>%
  filter(NAME == "Arizona") %>%
  st_transform(crs = 26912)
```

```
##   |                                                                              |
```

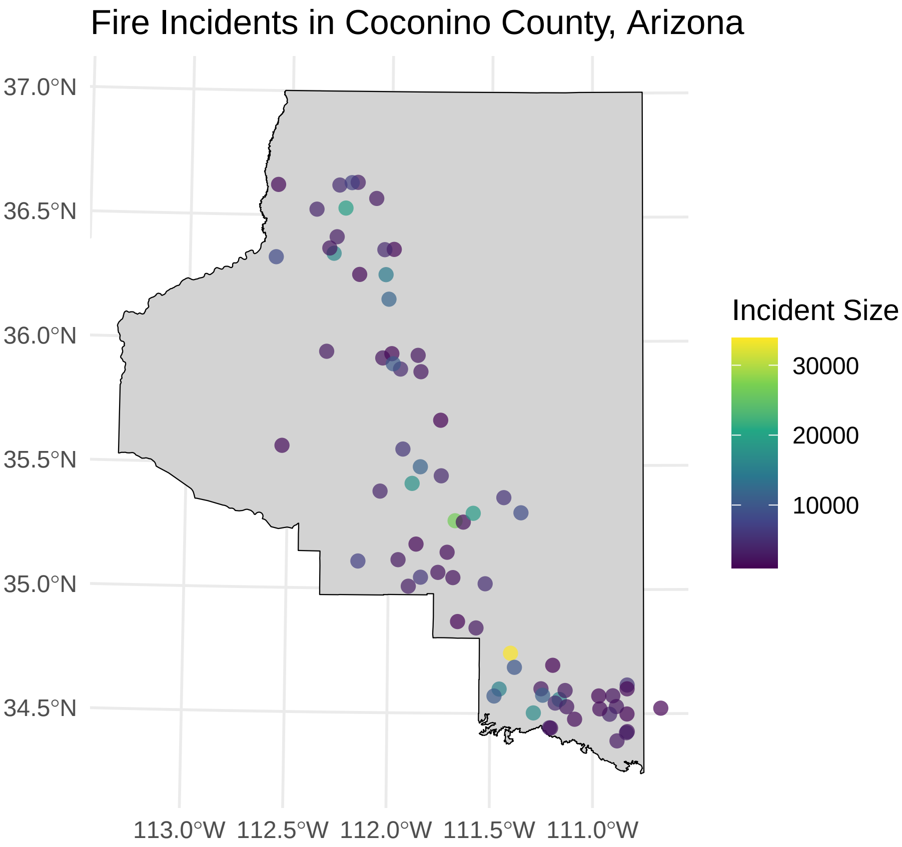
```
# Get Coconino County
coconino_county <- counties("AZ", cb = TRUE) %>%
  filter(NAME == "Coconino") %>%
  st_transform(crs = 26912)
```

```
##   |                                                                              |
```

```
# Load Census Tracts for Coconino County (2020 data)
census_tracts_sf <- tracts(state = "AZ", county = "Coconino", year = 2020, cb = TRUE) %>%
  st_transform(crs = 26912)
```

```
##   |                                                                              |
```

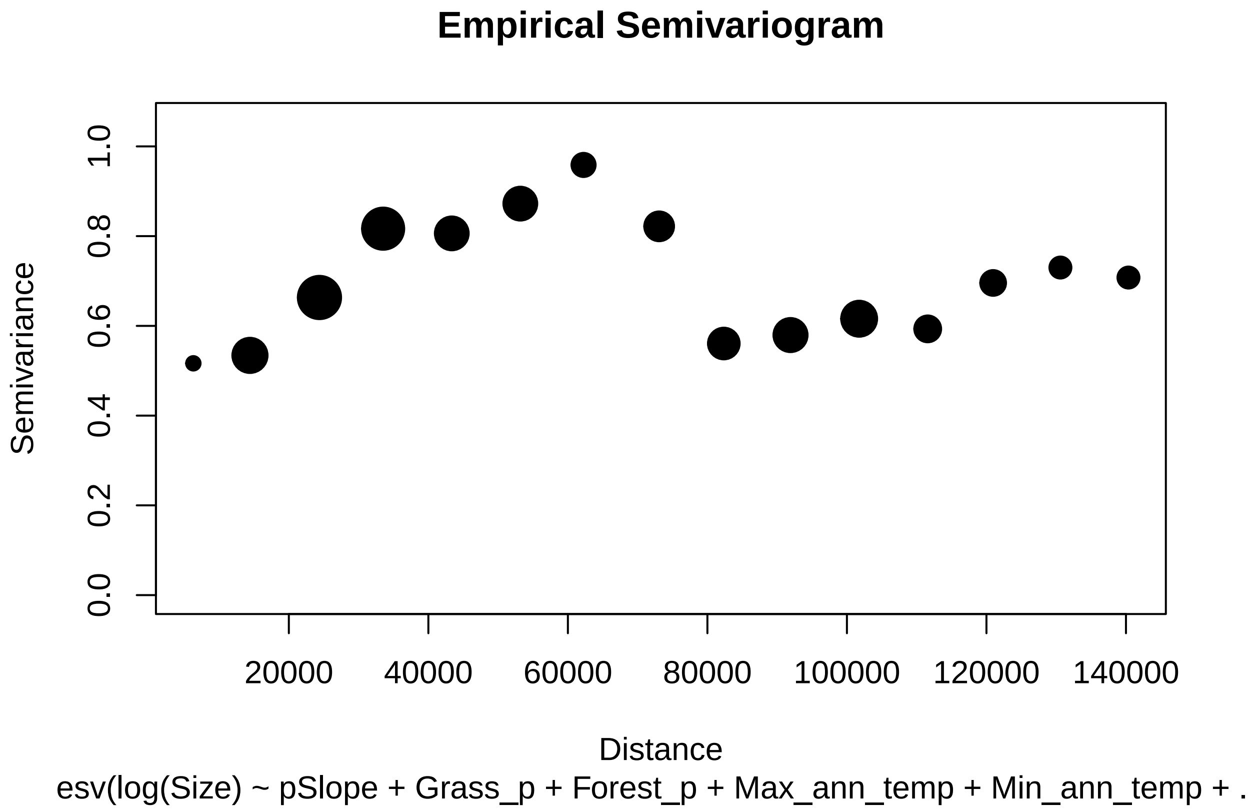
```
# Assuming coconino_sf contains the fire incident points
# Transform to the same CRS if needed
coconino_sf <- st_transform(coconino, crs = 26912)

# Plot
original_plot <- ggplot() +
  geom_sf(data = coconino_county, fill = "lightgray", color = "black", size = 1) +
  geom_sf(data = coconino_sf, aes(color = Size), size = 2, alpha = 0.7) +
  scale_color_viridis_c(option = "YlOrRd") +
  labs(title = "Fire Incidents in Coconino County, Arizona", color = "Incident Size") +
  theme_minimal()

original_plot
```



```
# save
# ggsave("coconino_fire_map.png", width = 10, height = 8, dpi = 300)
```

Semivariogram

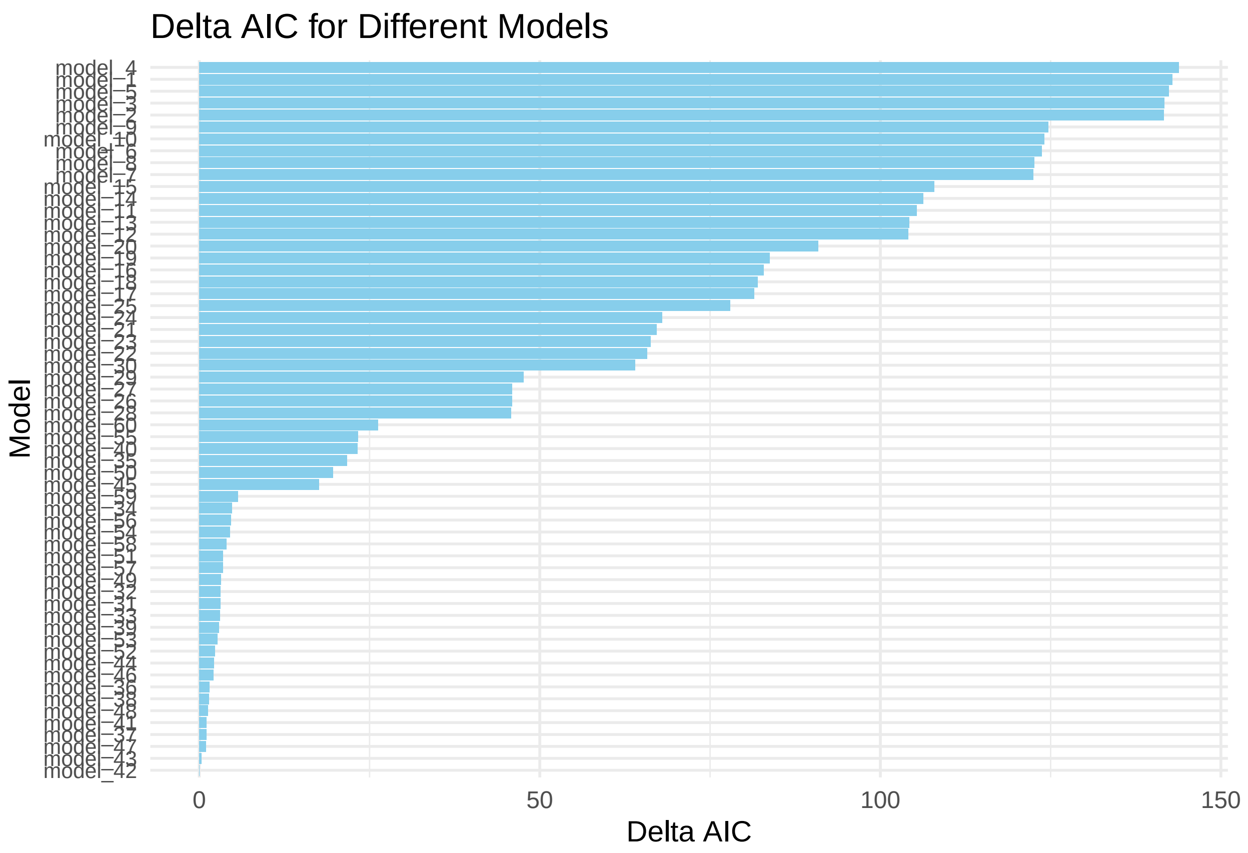
```
# Fit the model
fire_esv <- esv(
  log(Size) ~ pSlope + Grass_p + Forest_p + Max_ann_temp + Min_ann_temp + Prcp_ann + pop_density,
  data = coconino_sf
)


# Plot the semivariogram to assess spatial autocorrelation
plot(fire_esv)
```



Select best linear model

```
# List of all variables
variables <- c("pSlope", "Grass_p", "Forest_p", "Max_ann_temp", "Min_ann_temp",
               "Prcp_ann", "pop_density","Population", "Pri_rd", "All_rd", "Sec_rd", "Dist_4WD")

# List of spatial covariance types
spcov_types <- c("exponential", "gaussian", "spherical", "matern","none")

# Initialize an empty list to store models
models <- list()

# Counter for model names
model_counter <- 1

# Loop through variables
for (i in 1:length(variables)) {
  # Create formula with current set of variables and log-transformed Size
  formula <- as.formula(paste("Size ~", paste(variables[1:i], collapse = " + ")))

  # Loop through spatial covariance types
  for (spcov in spcov_types) {
    # Fit model
    model <- try(splm(formula, data = coconino_sf, spcov_type = spcov), silent = TRUE)
```

```r
    # If model fitting was successful, add to list
    if (!inherits(model, "try-error")) {
      model_name <- paste0("model_", model_counter)
      models[[model_name]] <- model
      model_counter <- model_counter + 1
    }
  }
}

# Calculate AIC for all models
aic_values <- sapply(models, AIC)

# Find the model with the lowest AIC
best_model <- models[[which.min(aic_values)]]

# Calculate delta AICs
min_aic <- min(aic_values)
delta_aic <- aic_values - min_aic

# Create a data frame for plotting
plot_data <- data.frame(
  Model = names(delta_aic),
  DeltaAIC = delta_aic
)

# Sort the data frame by DeltaAIC
plot_data <- plot_data[order(plot_data$DeltaAIC), ]

# Create the plot
ggplot(plot_data, aes(x = reorder(Model, DeltaAIC), y = DeltaAIC)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  coord_flip() +
  labs(title = "Delta AIC for Different Models",
       x = "Model",
       y = "Delta AIC") +
  theme_minimal() +
  theme(axis.text.y = element_text(size = 8))
```



```r
# Print the best model
cat("Best model:\n")
```

```
## Best model:
```

```r
print(summary(best_model))
```

```
##
## Call:
## splm(formula = formula, data = coconino_sf, spcov_type = spcov)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -13436  -3485   -939   1843  21910
##
## Coefficients (fixed):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   9.253e+04  2.387e+04   3.876 0.000106 ***
## pSlope        1.183e+01  8.533e+01   0.139 0.889706
## Grass_p       1.186e+03  4.087e+03   0.290 0.771747
## Forest_p     -5.391e+03  3.178e+03  -1.696 0.089793 .
## Max_ann_temp -4.064e+03  1.242e+03  -3.272 0.001067 **
## Min_ann_temp  1.654e+03  1.140e+03   1.450 0.147039
## Prcp_ann     -7.421e+03  2.749e+03  -2.699 0.006953 **
## pop_density  -3.286e+08  7.405e+08  -0.444 0.657203
## Population    3.877e-03  6.232e-01   0.006 0.995037
## Pri_rd       -1.066e-01  3.604e-02  -2.957 0.003110 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pseudo R-squared: 0.2678
##
## Coefficients (gaussian spatial covariance):
##       de       ie    range
##  5790546 29326176    32436
```

```
# Print the formula of the best model
cat("\nBest model formula:\n")
```

```
##
## Best model formula:
```

```
print(formula(best_model))
```

```
## Size ~ pSlope + Grass_p + Forest_p + Max_ann_temp + Min_ann_temp +
##     Prcp_ann + pop_density + Population + Pri_rd
```

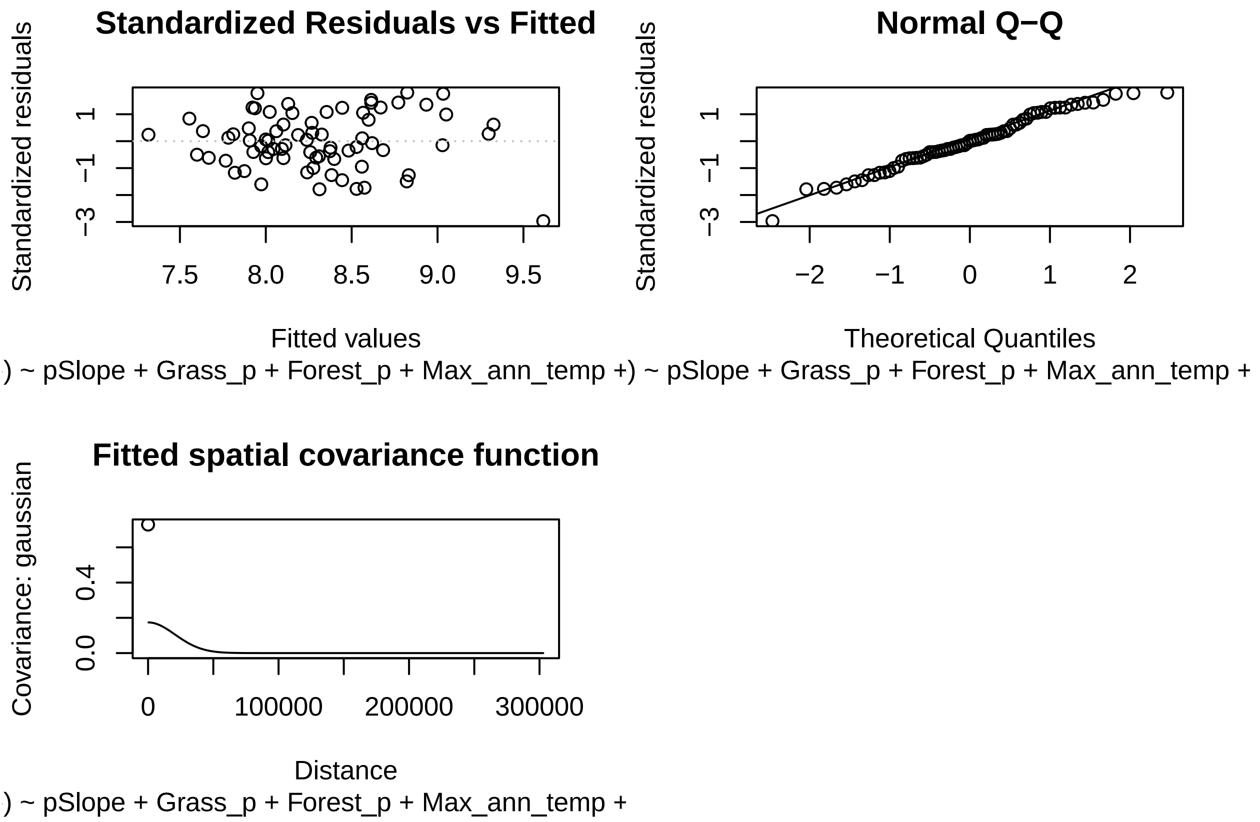
Fit spatial linear model

```
# Spatial linear model
spmod <- splm(log(Size) ~ pSlope + Grass_p + Forest_p + Max_ann_temp + Min_ann_temp + Prcp_ann + pop_de

# Display summary and diagnostics of the fitted spatial model
summary(spmod)
```

```
##
## Call:
## splm(formula = log(Size) ~ pSlope + Grass_p + Forest_p + Max_ann_temp +
##     Min_ann_temp + Prcp_ann + pop_density + Population + Pri_rd,
##     data = coconino_sf, spcov_type = "gaussian")
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0343 -0.4827 -0.1032  0.5320  1.6863
##
## Coefficients (fixed):
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.826e+01  3.389e+00   5.387 7.16e-08 ***
## pSlope        9.238e-03  1.223e-02   0.755   0.4500
## Grass_p       2.773e-01  5.822e-01   0.476   0.6338
## Forest_p     -5.344e-01  4.479e-01  -1.193   0.2328
## Max_ann_temp -5.156e-01  1.771e-01  -2.911   0.0036 **
## Min_ann_temp  2.264e-01  1.648e-01   1.374   0.1695
## Prcp_ann     -8.095e-01  3.881e-01  -2.086   0.0370 *
## pop_density   1.640e+04  1.083e+05   0.151   0.8796
## Population    6.920e-05  9.247e-05   0.748   0.4542
## Pri_rd       -1.013e-05  5.304e-06  -1.911   0.0561 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Pseudo R-squared: 0.2248
##
```

```
## Coefficients (gaussian spatial covariance):
##        de        ie     range
##     0.174     0.555 29119.613
```

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

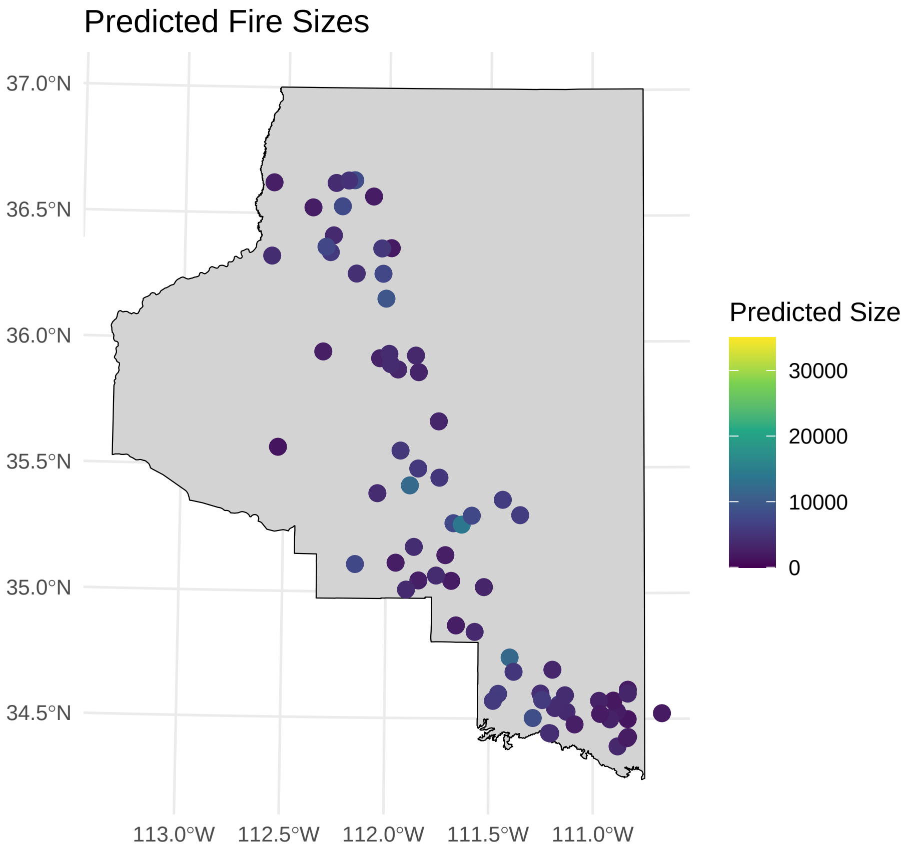


Prediction from the model

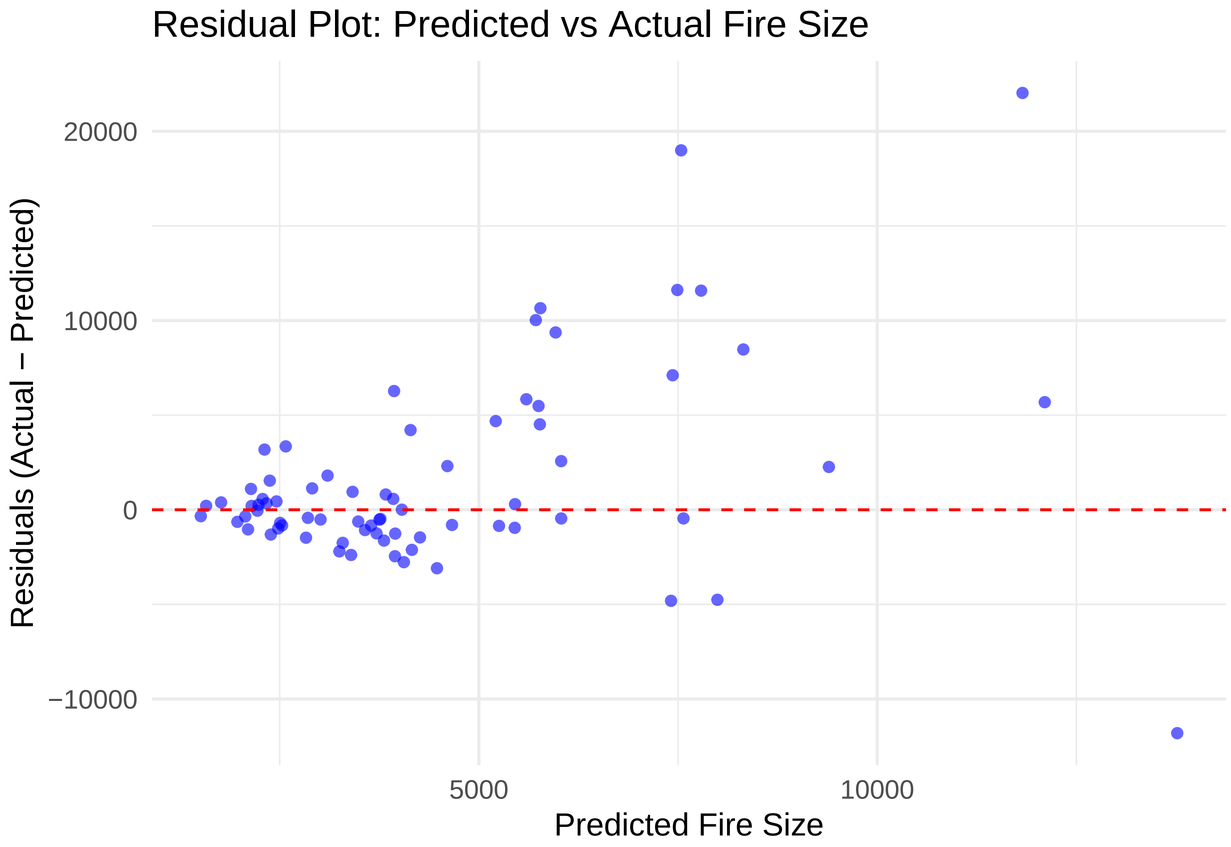
```
# Make predictions
predictions_log_actual <- predict(spmod, newdata = coconino_sf, type = "response")

# Convert log predictions back to the original scale (Size)
coconino_sf$predicted_size <- exp(predictions_log_actual)

# Plot the predicted fire sizes on the map
plot_actual <- ggplot() +
  geom_sf(data = coconino_county, fill = "lightgray", color = "black", size = 1) +  # Coconino County b
  geom_sf(data = coconino_sf, aes(color = predicted_size), size = 2.5) +
  scale_color_viridis_c(option = "YlOrRd", limits = c(0, 35000))+
  labs(title = "Predicted Fire Sizes", color = "Predicted Size") +
  theme_minimal(base_size = 10)
```

```
## Warning in viridisLite::viridis(n, alpha, begin, end, direction, option):
## Option 'YlOrRd' does not exist. Defaulting to 'viridis'.
```

```
# Show the predicted fire sizes plot
print(plot_actual)
```



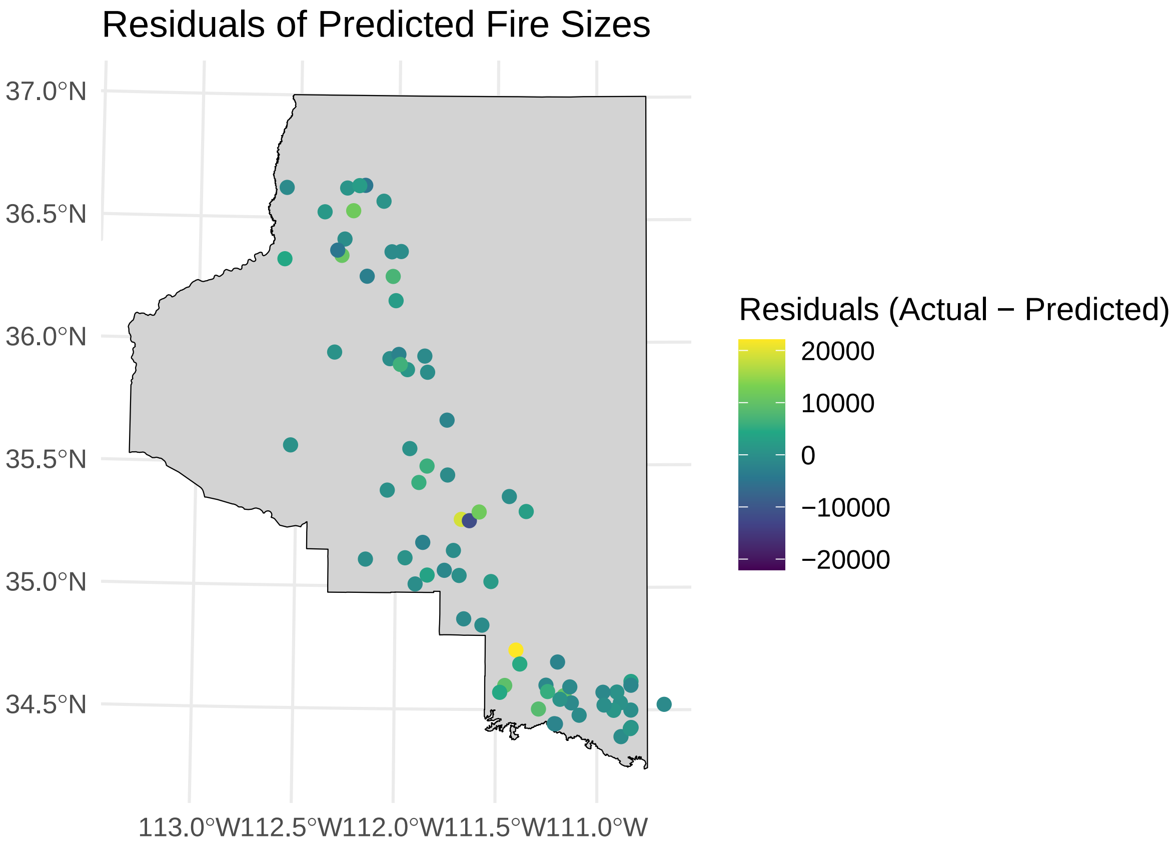
```
# Calculate residuals: Actual fire size - Predicted fire size
coconino_sf$residuals <- coconino_sf$Size - coconino_sf$predicted_size

# Create a residual plot (Predicted vs Residuals)
residual_plot <- ggplot(coconino_sf, aes(x = predicted_size, y = residuals)) +
  geom_point(color = "blue", alpha = 0.6) +  # Scatter plot of residuals
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Residual Plot: Predicted vs Actual Fire Size",
       x = "Predicted Fire Size", y = "Residuals (Actual - Predicted)") +
  theme_minimal(base_size = 12)  # Minimal theme for the plot

# Show the residual plot
print(residual_plot)
```



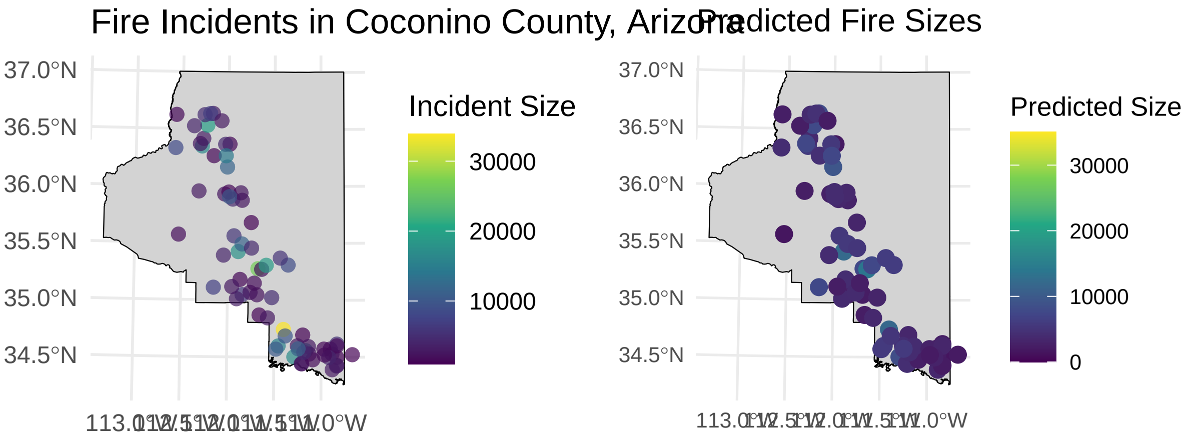
```
# Plot the residuals on the map
residual_map <- ggplot() +
  geom_sf(data = coconino_county, fill = "lightgray", color = "black", size = 1)+
  geom_sf(data = coconino_sf, aes(color = residuals), size = 2) +
  scale_color_viridis_c(option = "YlOrRd", limits = c(-max(abs(coconino_sf$residuals)), max(abs(coconin
  labs(title = "Residuals of Predicted Fire Sizes", color = "Residuals (Actual - Predicted)") +
  theme_minimal(base_size = 12)
```

```
## Warning in viridisLite::viridis(n, alpha, begin, end, direction, option):
## Option 'YlOrRd' does not exist. Defaulting to 'viridis'.
```

```
# Show the residual map
print(residual_map)
```


Residuals of Predicted Fire Sizes
37.0°N
36.5°N
36.0°N
35.5°N
35.0°N
34.5°N
113.0°W112.5°W112.0°W111.5°W111.0°W
Residuals (Actual − Predicted)
20000
10000
0
−10000
−20000

```
# compare original and predicted
original_plot + plot_actual
```



Fire Incidents in Coconino County, Arizona

Predicted Fire Sizes