STA 5377, Homework 4

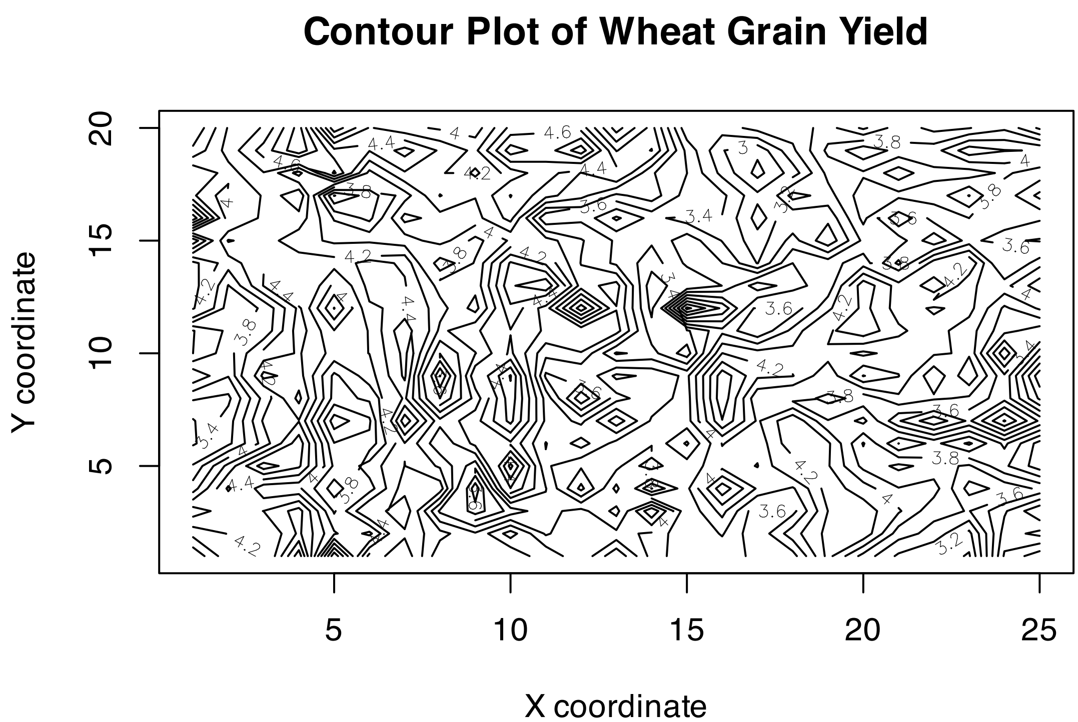
Carson Slater *Baylor University*

1.

The data set `wheat.txt` (on our web page) lists yields of grain and straw from 500 plots of wheat laid out in 25 x 20 lattice covering approximately 1 acre in total area. The data were collected by Mercer and Hall in 1910 at at the Rothamsted Experimental Station in England.

(a)

Make a contour plot of the grain measurements (use function `contour` in R).



(b)

Calculate Moran's I for these data based on a "rook's case" and on a "queen's case". Interpret your results. Are they consistent with your findings of part (a)?

	Rook	Queen	Rook.with.Permutation	Queen.with.Permutation
Moran's I Statistic	0.0947	0.0740	0.0947	0.074
p-value	0.0014	0.0005	0.0010	0.001

The results from these tests indicate that there is random spatial patterns in these data. The contour plot does appear to agree with these findings, as there is no prevalent pattern observed in the graphic.

2.

Consider the following grid of experimental plots:

1	7	13	19	25
2	8	14	20	26
3	9	15	21	27
4	10	16	22	28
5	11	17	23	29
6	12	18	24	30

The data in the file `corn.txt` consist of organic phosphorus in corn (y), organic phosphorus in soil (x_1), and inorganic phosphorus in soil (x_2) for each of the plots in the grid. Consider the following models:

$$\text{I } y = \beta_0 + \beta_1 x_1 + \epsilon$$
$$\text{II } y = \beta_0 + \beta_2 x_2 + \epsilon$$
$$\text{III } y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

(a)

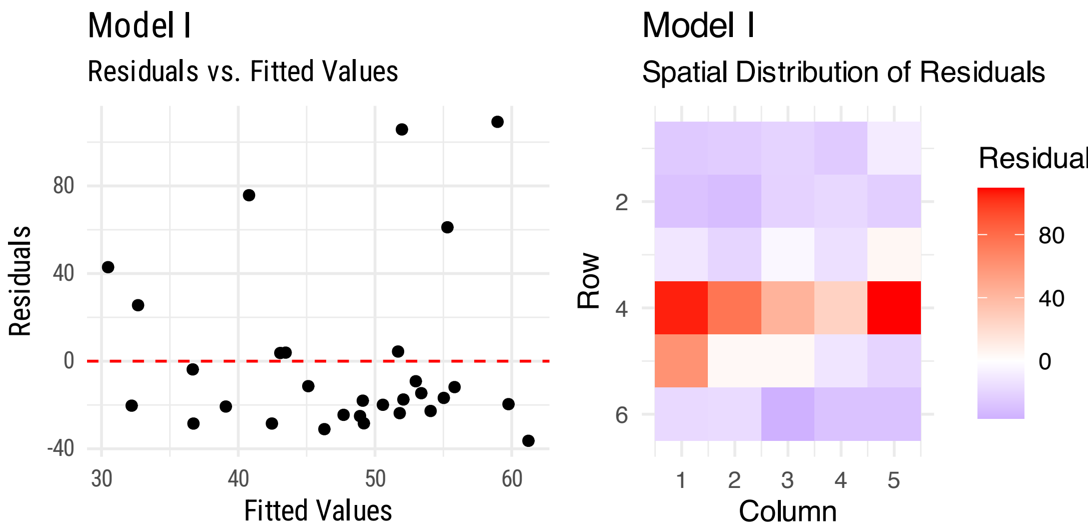
For each each model:

- **Fit the model using the usual regression approach (ignoring spatial considerations), and determine whether there is evidence of a spatial relationship in the residuals.**
- **Perform a test of significance for the slope(s) and intercept for the regression.**
- **Construct a residual plot.**

Model I: $y = \beta_0 + \beta_1 x_1 + \epsilon$

Characteristic	Beta	95% CI[1]	p-value
x1	1.3	-1.1, 3.8	0.3

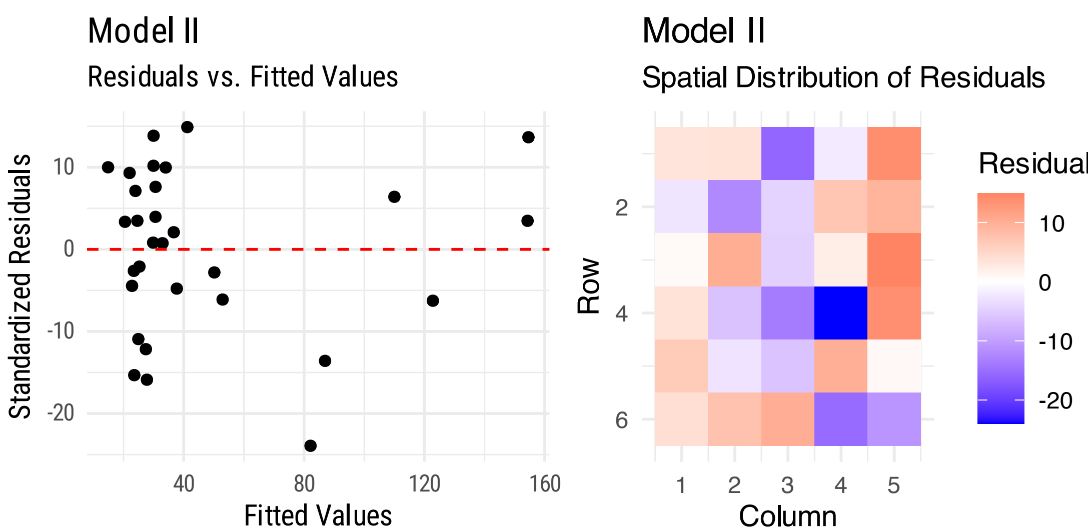
[1]CI = Confidence Interval



Model II: $y = \beta_0 + \beta_2 x_2 + \epsilon$

Characteristic	Beta	95% CI[1]	p-value
x2	1.9	1.7, 2.1	<0.001

[1]CI = Confidence Interval

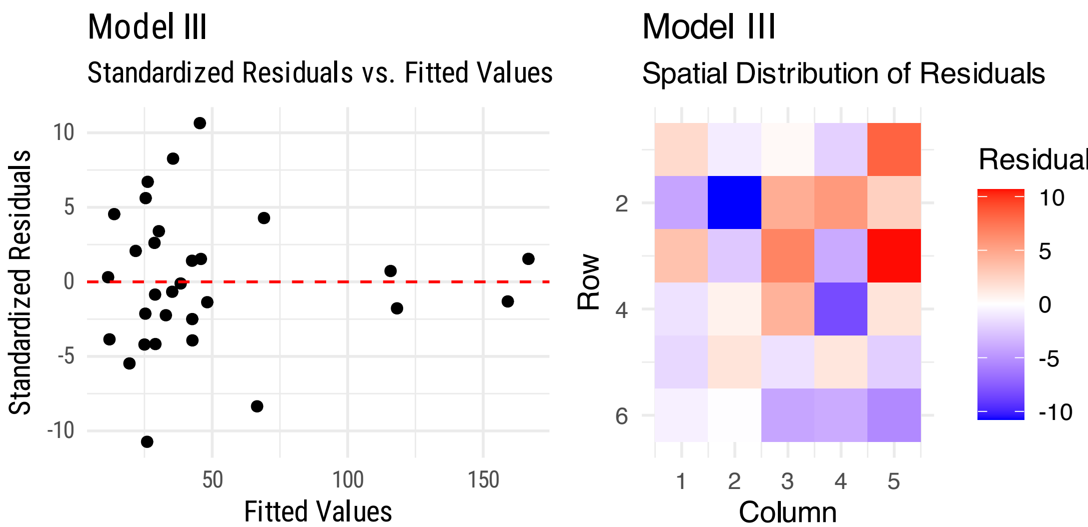


Model III: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$

Characteristic	Beta	95% CI[1]	p-value
x1	1.4	1.1, 1.7	<0.001

x2	1.9	1.9, 2.0	<0.001

[1]CI = Confidence Interval



Model 1: The estimated coefficient did not yield a statistically significant result. Additionally, the spatial distribution of the residuals yielded a spatial pattern, where there are lower residuals on the top and bottom sides of the graphic, and a spike toward the middle.

Model 2: The estimated coefficient yielded a statistically significant result. However, the spatial distribution of the residuals seems to yield a spatial pattern. There are positive residuals near the bottom left and top right of the spatial region, whereas diagonally across the center, there is a 'strip' of negative residuals.

Model 3: The estimated coefficients yielded statistically significant results. Also, the spatial distribution of the residuals does not seem to exhibit any stark patterns.

(b)

Using Moran's I for residuals from a regression is not quite correct. Explain why.

Here are a few considerations as to why Moran's I ought not be computed using residuals from a model:

- Model dependency: Residuals reflect how well a regression model is specified. Misspecified models may produce misleading spatial patterns in residuals that don't accurately represent true spatial autocorrelation in the original data.
- Induced spatial patterns: The regression process itself can create artificial spatial patterns in residuals, particularly when important spatial variables are omitted from the model.
- Violated independence assumption: Standard regression assumes independent observations. If spatial autocorrelation exists (which is what you're testing for), this fundamental assumption is already violated, potentially compromising both regression coefficients and residual analysis.

(c)

A better test is given by `lm.morantest(phosreg1, mylw)`. Perform this test, and compare it with the results of part (a).

	Model.1	Model.2	Model.3
Observed Moran I	0.1207	-0.0060	-0.0430
p.value	0.0563	0.3647	0.5123

It is difficult to compare the test of the significance of the slope to the Moran's I test, as they are not assessing the same phenomena. Alternatively, if the question is asking us to compare the Moran's I test to the graphics produced in (a), then the Moran's tests would reflect weaker conclusions than the ones deduced in (a). For example, in for part (a) with model 1, there seemed to be a spatial relationship among the residuals of the model. Here, using `lm.morantest(model1, listw)`, Model 1 has the highest Moran's I, but it is still relatively low. The other two Moran's I statistics from the models 2 and 3 are close to zero, which suggest no spatial relationship. This is not exactly congruent with the conclusions that were made in (a) for model 2, after examining the spatial distribution of the residuals.

(d)

Interpret your findings for the problem as a whole.

The analysis of three models predicting organic phosphorus in corn reveals several key insights:

- Model I ($y \sim x_1$): Non-significant coefficient with visible spatial patterns in residuals and the highest Moran's I value.
- Model II ($y \sim x_2$): Significant coefficient with diagonal patterns in residuals but low Moran's I.
- Model III ($y \sim x_1 + x_2$): Significant coefficients with no obvious spatial patterns and minimal spatial autocorrelation.

The multiple regression model (Model III) best captures the relationship between soil phosphorus variables and corn phosphorus content. The inclusion of both predictors adequately accounts for spatial structure in the data, eliminating patterns observed in the single-predictor models. This suggests that both organic and inorganic soil phosphorus forms contribute to organic phosphorus content in corn. The discrepancy between visual assessment and formal testing highlights the importance of proper spatial statistical methods in agricultural field trials.

Code Appendix

Below is all of the code used for this assignment.

```
# Setup

knitr::opts_chunk$set(dev = "cairo_pdf",
                      fig.width = 6,
                      fig.height = 3,
                      fig.align = 'center',
                      echo = FALSE,
                      message = FALSE,
                      warning = TRUE,
                      error = FALSE)
library("tidyverse"); library("patchwork"); library("glue")
library("scales", warn.conflicts = FALSE); library("extrafont")
library("tinytex"); library("rjags"); library("coda")
library("bayesplot"); library("patchwork"); library("geoR")
library("gridExtra"); library("tidyr"); library("latex2exp")
library("gstat"); library("furrr"); library("knitr")
library("gtsummary")
theme_set(theme_minimal(base_family = "Roboto Condensed"))

conflicted::conflicts_prefer(
  readr::col_factor(),
  purrr::discard(),
  rstan::extract(),
  dplyr::lag(),
  rstan::traceplot(),
  viridis::viridis_pal(),
  readr::parse_date
)
# Question 1a ###########################################

wheat <- read.delim("wheat.txt", sep = "", header = TRUE)
# Create a matrix for the grain data (25x20 lattice)
grain_matrix <- matrix(wheat$grain, nrow=25, byrow=TRUE)
par(mfrow = c(1,1))
contour(1:25, 1:20, grain_matrix,
        main="Contour Plot of Wheat Grain Yield",
        xlab="X coordinate",
        ylab="Y coordinate")
# Question 1b ###########################################
# Install and load required packages
if (!require(spdep)) {
  install.packages("spdep")
  library(spdep)
```

```r
}

# Set up the spatial grid
coords <- expand.grid(x=1:25, y=1:20)
coords$grain <- as.vector(grain_matrix)

# Create a neighbors list using rook's case (4 neighbors: up, down, left, rig
nb_rook <- cell2nb(25, 20, type="rook")

# Create a neighbors list using queen's case (8 neighbors: including diagonal
nb_queen <- cell2nb(25, 20, type="queen")

# Create spatial weights
w_rook <- nb2listw(nb_rook, style="W")
w_queen <- nb2listw(nb_queen, style="W")

# Calculate Moran's I for rook's case
moran_rook <- moran.test(coords$grain, w_rook)

# Calculate Moran's I for queen's case
moran_queen <- moran.test(coords$grain, w_queen)

# Calculate Moran's I permutation test to assess significance
moran_rook_perm <- moran.mc(coords$grain, w_rook, nsim=999)
moran_queen_perm <- moran.mc(coords$grain, w_queen, nsim=999)
#
# cat("\nMoran's I permutation test for rook's case:\n")
# print(moran_rook_perm)
#
# cat("\nMoran's I permutation test for queen's case:\n")
# print(moran_queen_perm)

# Visualize the spatial autocorrelation with a Moran plot
# par(mfrow=c(1,2))
# moran.plot(coords$grain, w_rook, labels=FALSE,
#           xlab="Grain Yield",
#           ylab="Spatially Lagged Grain Yield",
#           main="Moran Plot (Rook's Case)")
#
# moran.plot(coords$grain, w_queen, labels=FALSE,
#           xlab="Grain Yield",
#           ylab="Spatially Lagged Grain Yield",
#           main="Moran Plot (Queen's Case)")

extract <- \(obj) {
  c(obj[3] |> unname() |> unlist() |> pluck(1), obj[2] |> unname() |> unlist(
}
```

```r
extract2 <- \(obj) {
  c(obj[1] |> unname() |> unlist(), obj[3] |> unname() |> unlist())
}


data.frame(
 "Rook" = extract(moran_rook),
 "Queen" = extract(moran_queen),
 "Rook with Permutation" = extract2(moran_rook_perm),
 "Queen with Permutation" = extract2(moran_queen_perm)
) |>
  `row.names<-.data.frame`(c("Moran's I Statistic", "p-value")) |>
  kable(digits = 4)
# Question 2a ###########################################

# Load the data
corn_data <- read.table("corn.txt", header = TRUE)

# Assign grid positions to the data
# Plots are numbered 1 to 30 in a grid with 6 rows and 5 columns
# Create a data frame with plot numbers and their row/column positions
plot_positions <- expand.grid(col = 1:5, row = 1:6)
plot_positions$plot <- 1:30

# Merge the positions with the original data
corn_data <- cbind(corn_data, plot_positions)
# Fit Model I
model1 <- lm(y ~ x1, data = corn_data)

# Summary of Model I
# summary(model1)

model1 |> tbl_regression()

# Add residuals to the data
corn_data$residuals_model1 <- residuals(model1)

# Plot residuals against fitted values
m1p1 <- ggplot(corn_data, aes(x = fitted.values(model1), y = residuals_model1)
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Model I",
       subtitle = "Residuals vs. Fitted Values",
       x = "Fitted Values",
       y = "Residuals")
# Create a spatial weights matrix based on the grid structure
# Define neighbors based on queen contiguity (adjacent in any direction)
```

```
nb <- cell2nb(5, 6, type = "queen")

# Create a spatial weights object
listw <- nb2listw(nb, style = "W")

# Moran's I test for spatial autocorrelation in residuals
# moran.test(corn_data$residuals_model1, listw)

# Plot the residuals on the grid
m1p2 <- ggplot(corn_data, aes(x = col, y = row, fill = residuals_model1)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = (
  labs(title = "Model I",
       subtitle = "Spatial Distribution of Residuals",
       x = "Column",
       y = "Row",
       fill = "Residual") +
  theme_minimal() +
  scale_y_reverse() + # To match the grid layout in the problem
  coord_equal()
m1p1 + m1p2
# Fit Model II
model2 <- lm(y ~ x2, data = corn_data)

# Summary of Model II
model2 |> tbl_regression()

# Add residuals to the data
corn_data$residuals_model2 <- residuals(model2)

# Plot residuals against fitted values
m2p1 <- ggplot(corn_data, aes(x = fitted.values(model2), y = residuals_model2)
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Model II",
       subtitle = "Residuals vs. Fitted Values",
       x = "Fitted Values",
       y = "Standardized Residuals")
# Moran's I test for spatial autocorrelation in residuals
# moran.test(corn_data$residuals_model2, listw)

# Plot the residuals on the grid
m2p2 <- ggplot(corn_data, aes(x = col, y = row, fill = residuals_model2)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = (
  labs(title = "Model II",
       subtitle = "Spatial Distribution of Residuals",
```

```
       x = "Column",
       y = "Row",
       fill = "Residual") +
  theme_minimal() +
  scale_y_reverse() +
  coord_equal()
m2p1 + m2p2
# Fit Model III
model3 <- lm(y ~ x1 + x2, data = corn_data)

# Summary of Model III
model3 |> tbl_regression()

# Add residuals to the data
corn_data$residuals_model3 <- residuals(model3)

# Plot residuals against fitted values
m3p1 <- ggplot(corn_data, aes(x = fitted.values(model3), y = residuals_model3)
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
  labs(title = "Model III",
       subtitle = "Standardized Residuals vs. Fitted Values",
       x = "Fitted Values",
       y = "Standardized Residuals")
# Moran's I test for spatial autocorrelation in residuals
# moran.test(corn_data$residuals_model3, listw)

# Plot the residuals on the grid
m3p2 <- ggplot(corn_data, aes(x = col, y = row, fill = residuals_model3)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", mid = "white", high = "red", midpoint = 0
  labs(title = "Model III",
       subtitle = "Spatial Distribution of Residuals",
       x = "Column",
       y = "Row",
       fill = "Residual") +
  theme_minimal() +
  scale_y_reverse() +
  coord_equal()
m3p1 + m3p2
# Question 2c ############################################

extract3 <- \(obj) {
  c(obj[[3]][1], obj[2]) |> unlist()
}

data.frame(
```

```r
  "Model 1" = lm.morantest(model1, listw) |> extract3(),
  "Model 2" = lm.morantest(model2, listw) |> extract3(),
  "Model 3" = lm.morantest(model3, listw) |> extract3()
) |> kable(digits = 4)
```