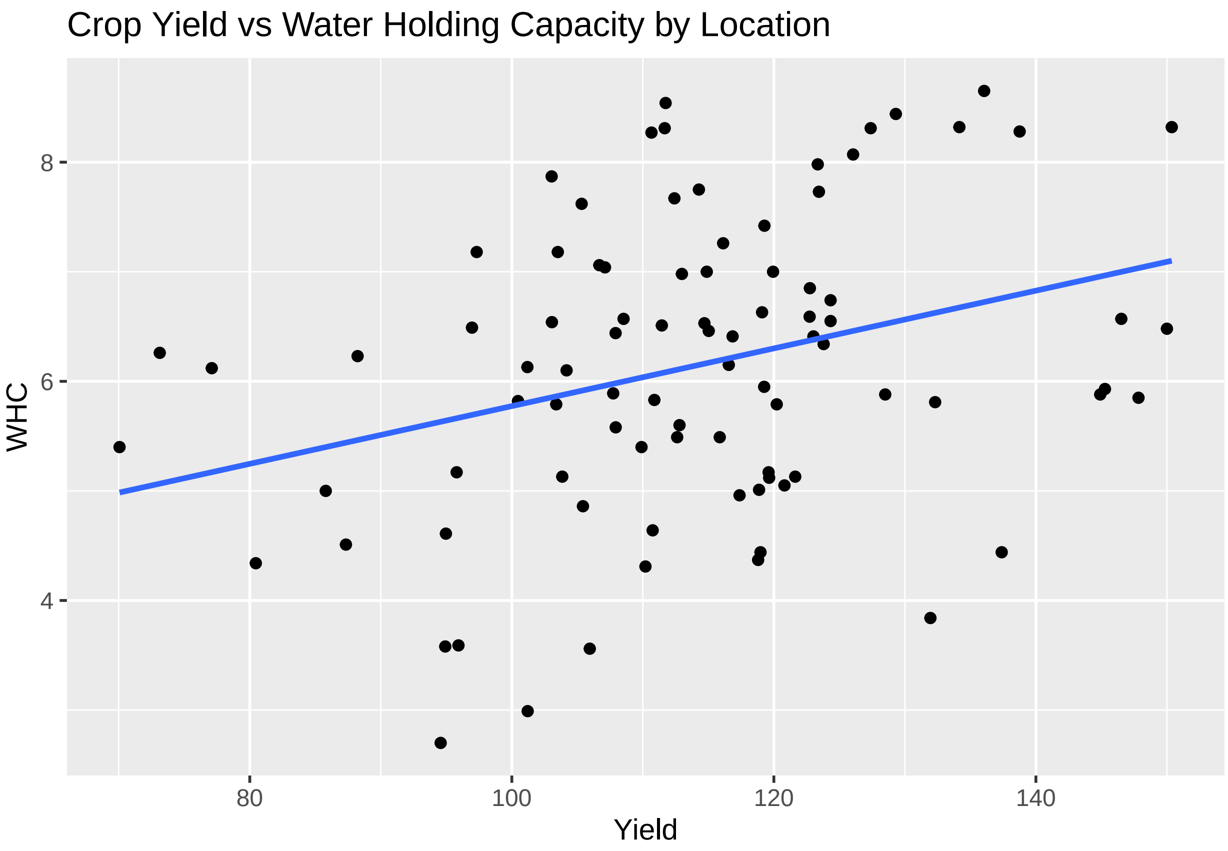
Irrigated Agriculture Homework

Mary Ebbert and Jillian Warburton

2023-04-07

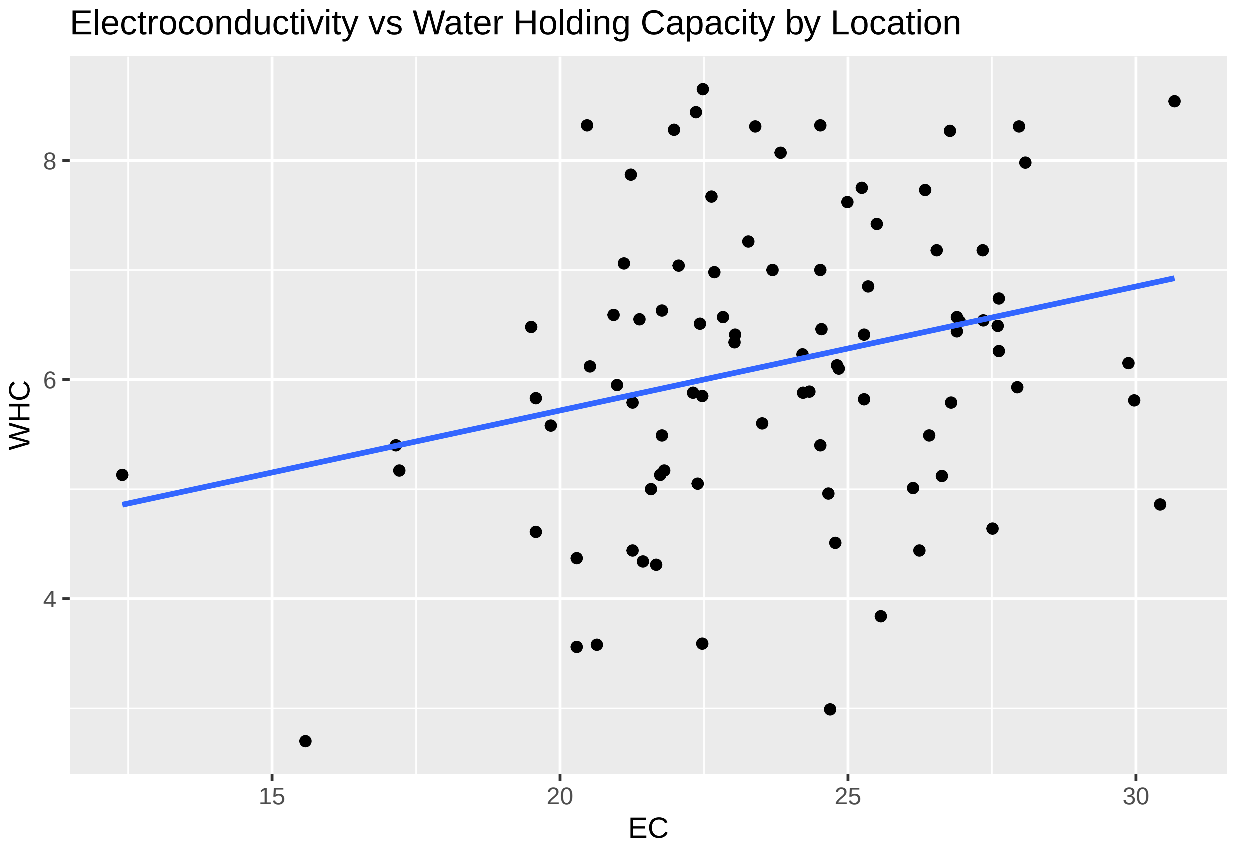
1. Create exploratory plots of the data by looking at the relationship between `WHC` (the response variable) and `Yield` and `EC`. Comment on any general relationships you see from the data.

```
data %>%
  ggplot(mapping = aes(x = Yield, y = WHC)) +
  geom_point() +
  geom_smooth(se = FALSE, method = 'lm') +
  labs(title = "Crop Yield vs Water Holding Capacity by Location")
```



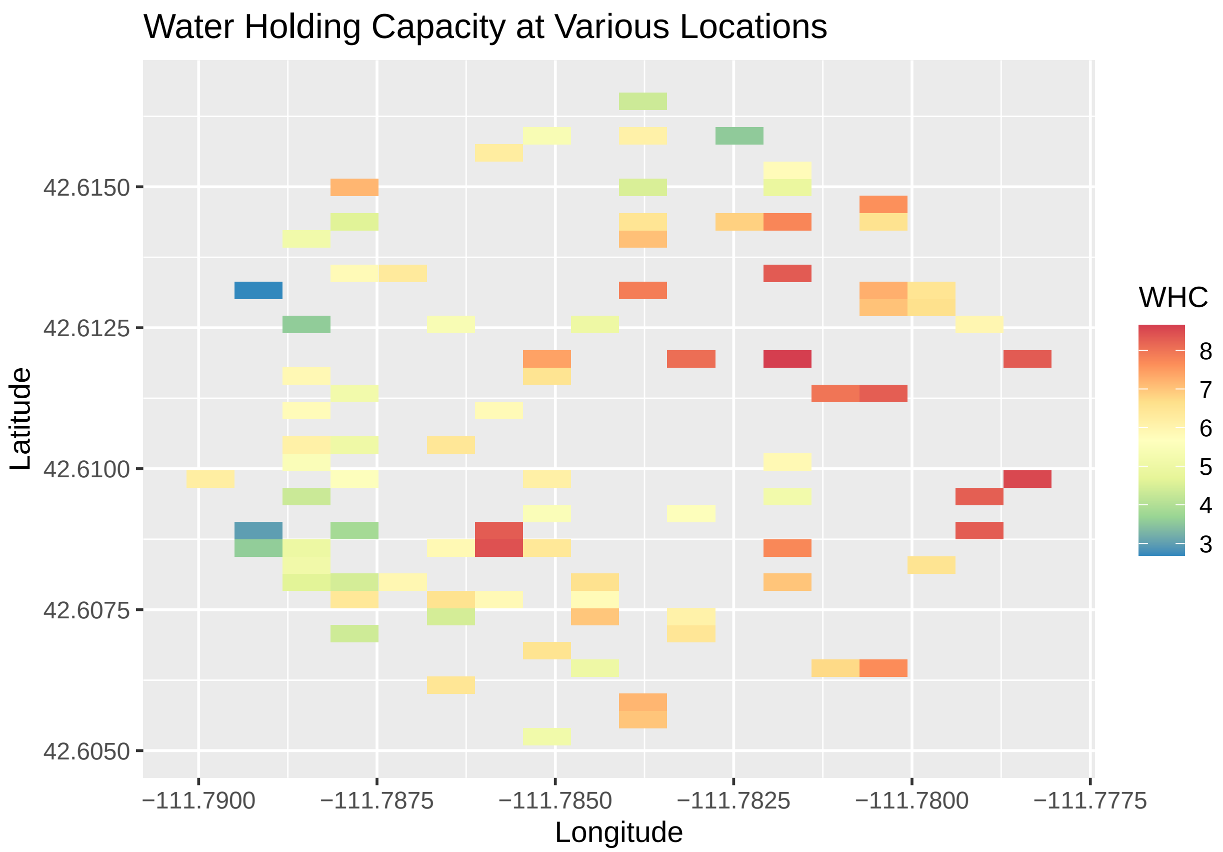
There appears to be a moderately strong positive linear relationship between crop yield and WHC (water holding capacity), implying that higher crop yield is generally correlated with higher water holding capacity at that location.

```
data %>%
  ggplot(mapping = aes(x = EC, y = WHC)) +
  geom_point() +
  geom_smooth(se = FALSE, method = 'lm') +
  labs(title = "Electroconductivity vs Water Holding Capacity by Location")
```



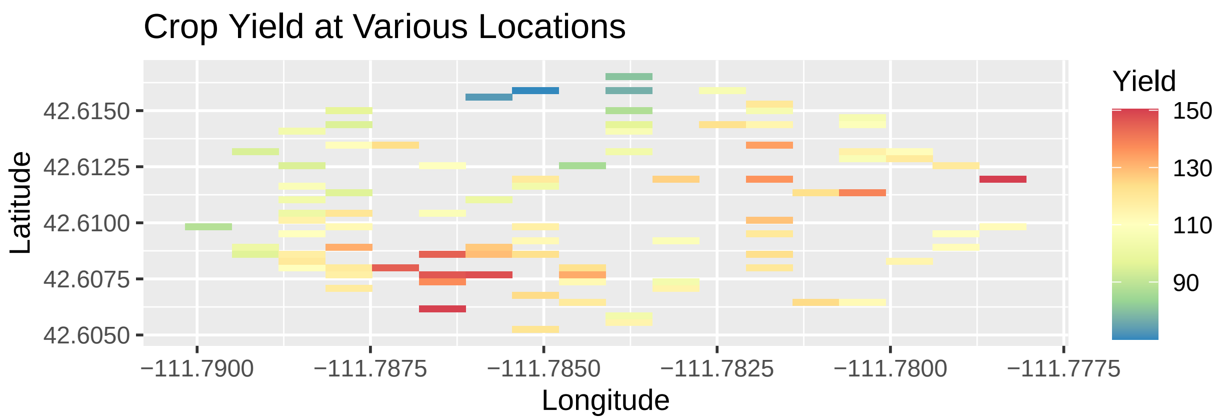
There appears to be a weaker, positive, linear relationship between a soil's electroconductivity and its water holding capacity.

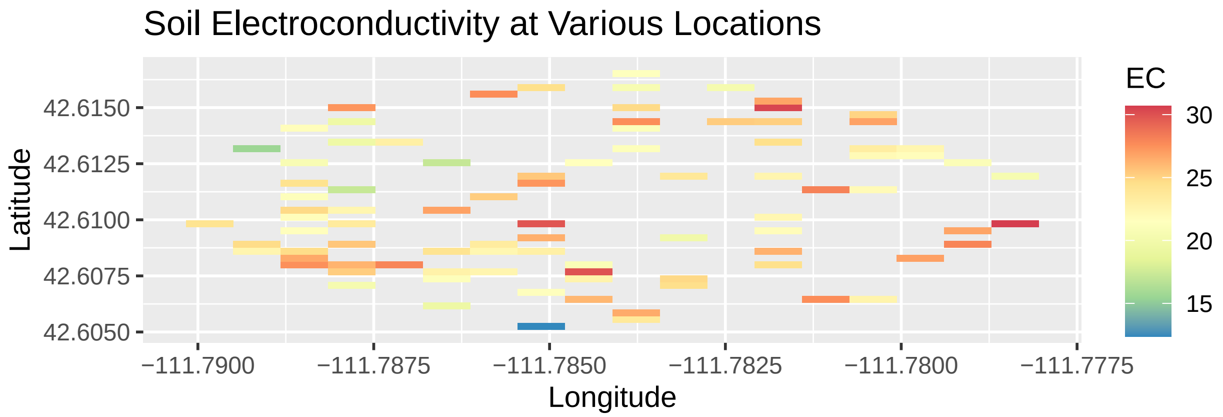
```
ggplot(data = data, mapping = aes(x = Lon, y = Lat, fill = WHC)) +
  geom_tile() +
scale_fill_distiller(palette="Spectral",na.value=NA) +
  labs(title = "Water Holding Capacity at Various Locations",
       x = "Longitude",
       y = "Latitude")
```



This plot shows that there is a clear spatial correlation between locations and their associated WHC values. The same can be said of their crop yields and soil electroconductivity (see plots below).

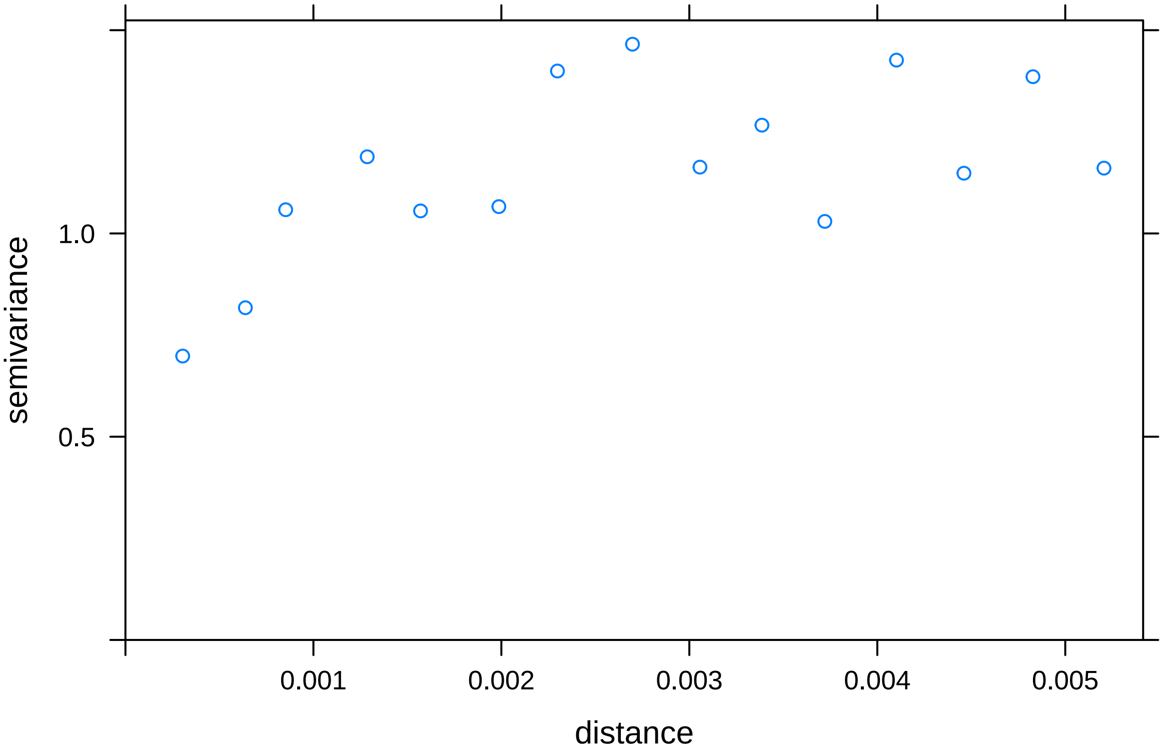
```
yield_spatial <- ggplot(data = data, mapping = aes(x = Lon, y = Lat, fill = Yield)) +
  geom_tile() +
scale_fill_distiller(palette="Spectral",na.value=NA) +
  labs(title = "Crop Yield at Various Locations",
       x = "Longitude",
       y = "Latitude")

ec_spatial <- ggplot(data = data, mapping = aes(x = Lon, y = Lat, fill = EC)) +
  geom_tile() +
scale_fill_distiller(palette="Spectral",na.value=NA) +
  labs(title = "Soil Electroconductivity at Various Locations",
       x = "Longitude",
       y = "Latitude")

grid.arrange(yield_spatial, ec_spatial)
```





2. Fit an independent MLR model with a linear effect between `Yield`, `EC` and the response variable `WHC`. Explore the residuals to see if there is evidence of spatial correlation by mapping the residuals and plotting the variogram of the residuals.

```
data_lm <- lm(data = data,
              formula = WHC ~ Yield + EC)

variogram(object=WHC~Yield + EC, locations=~Lon+Lat, data=data) %>%
  plot()
```



This variogram clearly indicates that there is spatial correlation as it is increasing until reaching a saturation point.

3. To determine an appropriate correlation structure to use, fit a spatial model using exponential, spherical and Gaussian correlation functions with a nugget effect (don't forget to filter out the missing observations). Compare the model fits using AIC and use the best fit model for the remainder of the analysis.

```
gls_exp = gls(model=WHC~Yield + EC, data=data,
              correlation=corExp(form=~Lon+Lat, nugget=TRUE), method="ML")

gls_sphere = gls(model=WHC~Yield + EC, data=data,
                 correlation=corSpher(form=~Lon+Lat, nugget=TRUE), method="ML")

gls_gaus = gls(model=WHC~Yield + EC, data=data,
               correlation=corGaus(form=~Lon+Lat, nugget=TRUE), method="ML")

AIC(gls_exp) # best one
```

```
## [1] 272.3653
```

```
AIC(gls_sphere)
```

```
## [1] 272.9623
```

```
AIC(gls_gaus)
```

```
## [1] 273.4355
```

```
coef(gls_exp$modelStruct$corStruct, unconstrained=FALSE)
```

```
##      range     nugget
## 0.00223684 0.36755325
```

```
summary(gls_exp)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: WHC ~ Yield + EC
##   Data: data
##        AIC      BIC    logLik
##   272.3653 287.2971 -130.1827
##
## Correlation Structure: Exponential spatial correlation
##  Formula: ~Lon + Lat
##  Parameter estimate(s):
##      range     nugget
## 0.00223684 0.36755325
##
## Coefficients:
##                 Value Std.Error  t-value p-value
## (Intercept) 1.5663358 1.3950884 1.122750  0.2647
## Yield       0.0257820 0.0094601 2.725329  0.0078
## EC          0.0739945 0.0362603 2.040648  0.0444
##
##  Correlation:
##       (Intr) Yield
## Yield -0.745
## EC    -0.605 -0.003
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.46170693 -0.76133851 -0.03222967  0.72664078  1.69219614
##
## Residual standard error: 1.226917
## Degrees of freedom: 89 total; 86 residual
```

The best fit model is the exponential correlated general linear model, as proven by it having the lowest AIC of 272.3653.

4. Write out your model for analyzing the agriculture data in terms of parameters. Explain and interpret any parameters associated with the model.

The model for this data set is $\mathbf{y} \sim \mathcal{N}(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{R}(\phi, \omega))$, with those variables expanded and explained below.

We expand the matrix $\mathbf{y}$ to the matrix $\begin{bmatrix} \text{WHC}_1 \\ \text{WHC}_2 \\ \vdots \\ \text{WHC}_{89} \end{bmatrix}$, where WHC_i is the `WHC` (water holding capacity) of a location i. Locations are determined by the longitude and latitude of a measurement for water holding capacity.

We expand the matrix $\mathbf{X}$ to include $\begin{bmatrix} 1 & x_{1,1} & x_{1,2} \\ 1 & x_{2,1} & x_{2,2} \\ \vdots & \ddots & \vdots \\ 1 & x_{89,1} & x_{89,2} \end{bmatrix}$, where $x_{i,1} =$ the electroconductivity of location i and where $x_{i,2} =$ the crop yield of location i.

We expand $\boldsymbol{\beta}$ to include $\begin{bmatrix} \beta_0 \\ \beta_{\text{Yield}} \\ \beta_{\text{EC}} \end{bmatrix}$. The β_{EC} is the change in a location's water holding capacity for one unit increase in the location's electroconductivity measurement, holding all else constant. The β_{Yield} is the change in a location's water holding capacity for one unit increase in the location's crop yield, holding all else constant.

Finally, the residuals of the model, or ϵ as written in most models, are determined by the correlation structure calculated by $\sigma^2\mathbf{R}(\phi, \omega)$). The matrix $\mathbf{R}$ is expanded to include

$$\begin{bmatrix} 1 & \rho(\text{WHC}_1, \text{WHC}_2) & \dots & \rho(\text{WHC}_1, \text{WHC}_{89}) \\ \rho(\text{WHC}_2, \text{WHC}_1) & 1 & \dots & \rho(\text{WHC}_2, \text{WHC}_{89}) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(\text{WHC}_{89}, \text{WHC}_1) & \rho(\text{WHC}_{89}, \text{WHC}_2) & \dots & 1 \end{bmatrix}$$

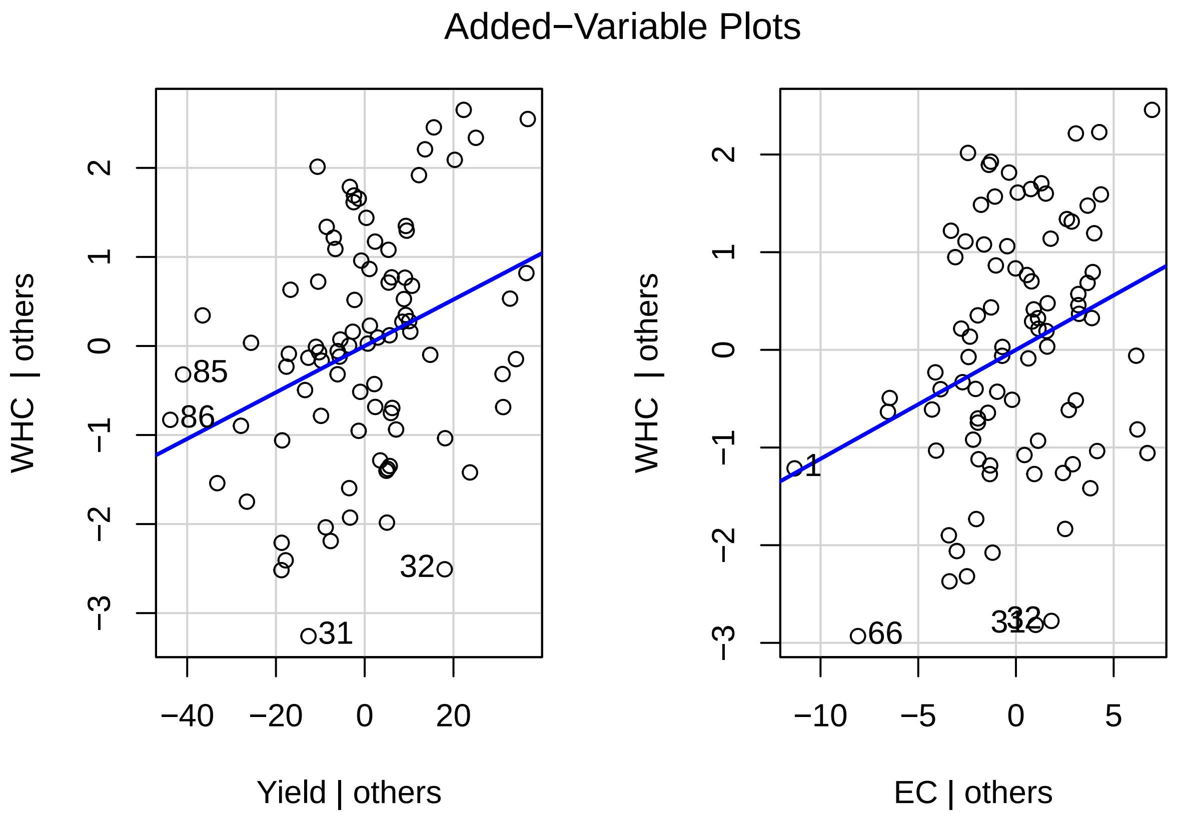
.

Specifically, this correlation is a general exponential correlation structure within `WHC`, where $\rho(\text{WHC}_i, \text{WHC}_j) = -\exp\left\{\frac{\|\text{WHC}_i - \text{WHC}_j\|}{\phi}\right\}$, or the exponential correlation between locations i and j. The variables ϕ and ω are found through iterative optimization to help decorrelate the residuals in the $\mathbf{R}$ matrix, or $\rho(\text{WHC}_i, \text{WHC}_j)$ and use Maximum Likelihood Estimation to scale the variances over locations. The variable ϕ is the range of correlations between different locations. The variable ω is the correlation of a location with it's own self, and is used with $(1 - \omega)\rho(\text{WHC}_i, \text{WHC}_j)$.

5. Fit your spatial MLR model and validate any assumptions you made to fit the model.

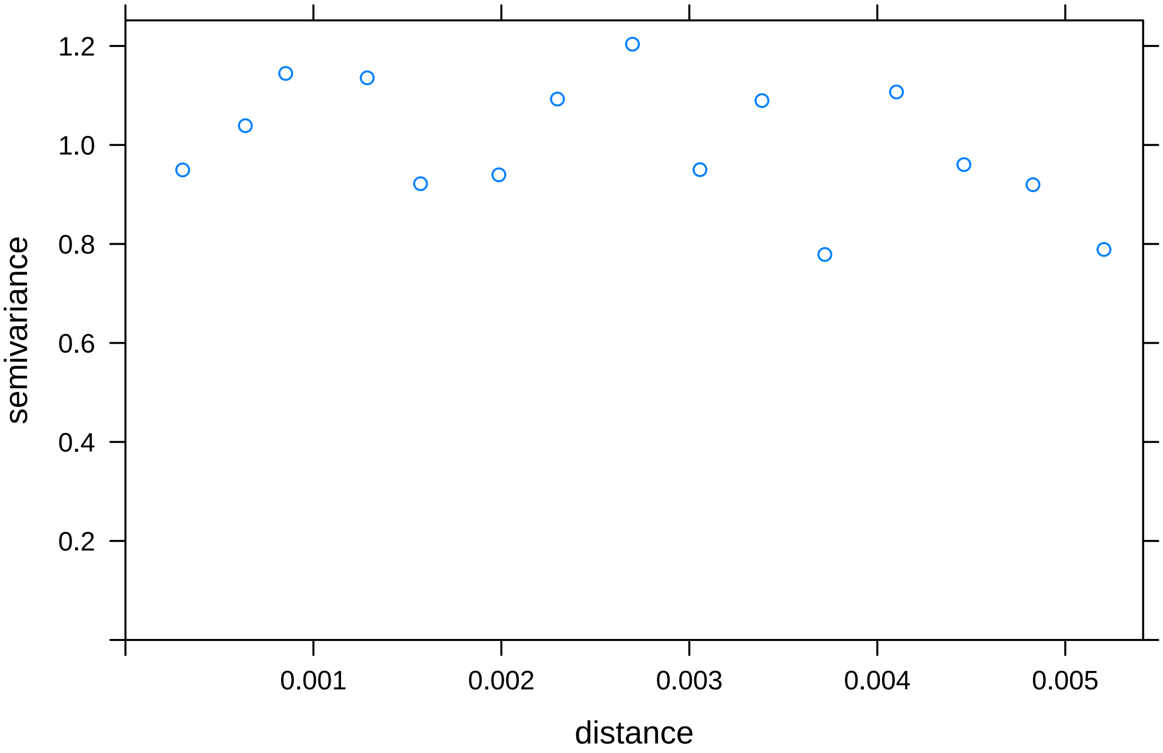
```
gls_model = gls_exp
```

```
avPlots(data_lm, ask=FALSE)
```



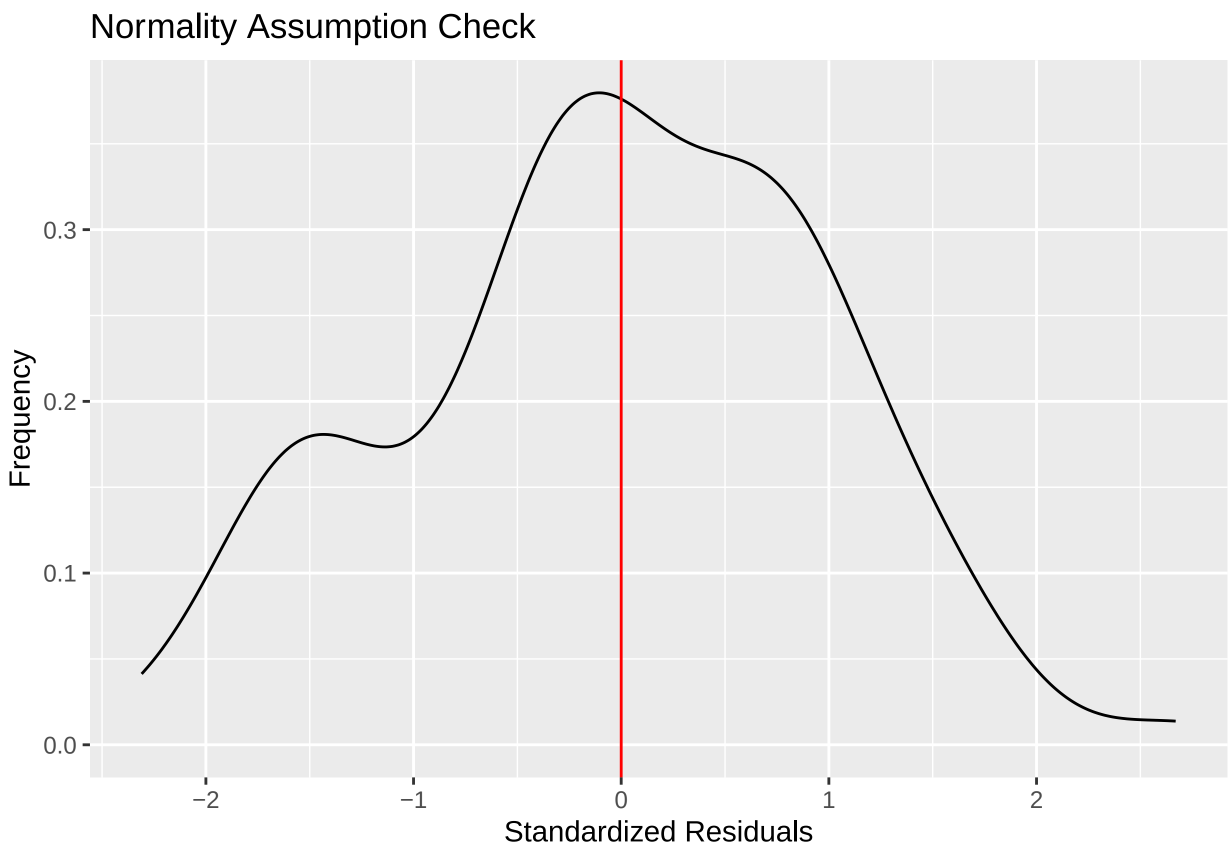
These added variable plots do not indicate any deviation from linearity. Thus, we can say that this assumption is met.

```
sres = stdres.gls(gls_model)

residDF <- data.frame(Lon=data$Lon, Lat=data$Lat, decorrResid=sres)
residVariogram <- variogram(object=decorrResid~1, locations=~Lon+Lat, data=residDF)
plot(residVariogram)
```



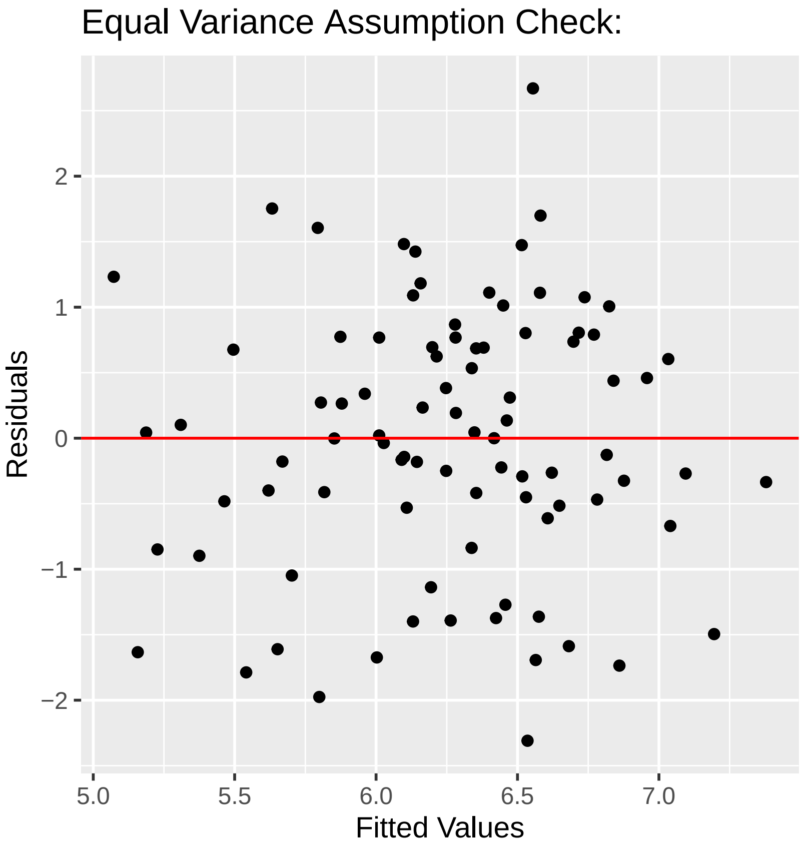
This variogram shows that the independence assumption is now met as it is no longer increasing and is now fairly flat.

```
ggplot() +
  geom_density(mapping = aes(x = sres)) +
  xlab('Standardized Residuals') +
  ylab('Frequency') +
  ggtitle('Normality Assumption Check') +
  geom_vline(xintercept = 0, col = "red")
```



This density plot shows that the residuals are distributed mostly normally distributed. Thus, we can say that the normality assumption is met.

```r
# fitted values vs decorrelated residuals
ggplot(mapping = aes(x=gls_model$fitted, y=sres)) +
  geom_point() +
  xlab('Fitted Values') +
  ylab('Residuals') +
  ggtitle('Equal Variance Assumption Check:') +
  geom_hline(yintercept = 0, col = "red") +
  theme(aspect.ratio = 1)
```

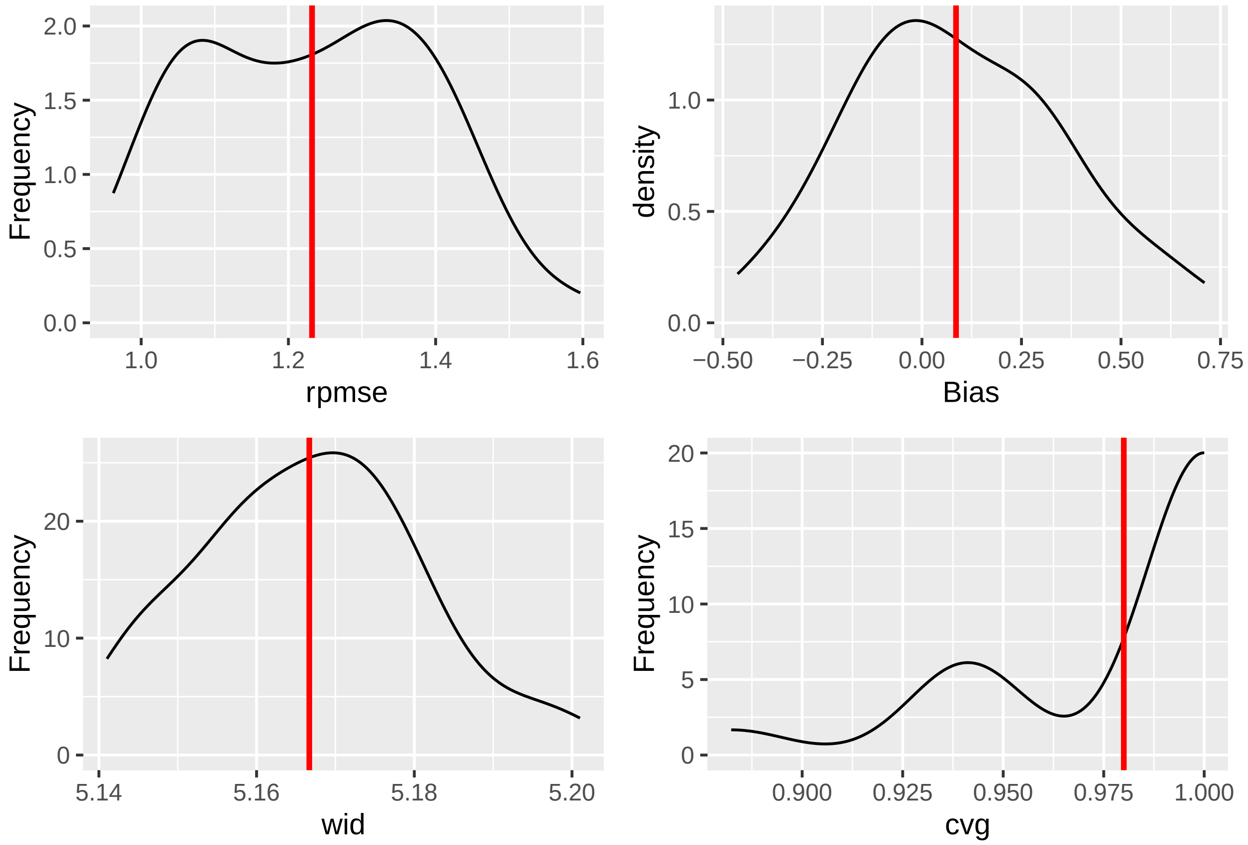


The residuals appear to be mostly equally spread around the horizontal line, indicating that the assumption of equal variance is met.

6. Determine the predictive accuracy of your model in terms of RPMSE, coverage and width of prediction intervals and interpret each of these predictive diagnostics.

```
## Run the CV code
set.seed(59)
n.cv <- 50 #Number of CV studies to run

# pb <- txtProgressBar(min = 0, max = n.cv, style = 3)

n.test <- dim(data)[1]*0.2 #Number of observations in a test set
rpmse <- rep(x=NA, times=n.cv)
bias <- rep(x=NA, times=n.cv)
wid <- rep(x=NA, times=n.cv)
cvg <- rep(x=NA, times=n.cv)

n = dim(data)[1]

for(cv in 1:n.cv){
  ## Select test observations
  test.obs <- sample(x=1:n, size=n.test)
  ## Split into test and training sets
  test.set <- data[test.obs,]
  train.set <- data[-test.obs,]

  ## Fit a lm() using the training data
  train.lm <- gls(model=WHC ~ Yield + EC, data=data,
                  correlation=corExp(form=~Lon+Lat, nugget=TRUE), method="ML")

  ## Generate predictions for the test set
  my.preds <- predictgls(train.lm, newdframe = test.set)
    # predict.lm(train.lm, newdata=test.set, interval="prediction")

  ## Calculate bias
  bias[cv] <- mean(my.preds[,'Prediction']-test.set[['WHC']])

  ## Calculate RPMSE
  rpmse[cv] <- (test.set[['WHC']]-my.preds[,'Prediction'])^2 %>% mean() %>% sqrt()

  ## Calculate Coverage
  cvg[cv] <- ((test.set[['WHC']] > my.preds[,'lwr']) & (test.set[['WHC']] <
                                                          my.preds[,'upr'])) %>% mean()

  ## Calculate Width
  wid[cv] <- (my.preds[,'upr'] - my.preds[,'lwr']) %>% mean()

  ## Update the progress bar
  # setTxtProgressBar(pb, cv)
}

# close(pb)

cv_bias <- ggplot() +
  geom_density(mapping = aes(x=bias)) +
  xlab("Bias") +
  geom_vline(xintercept = mean(bias), col = "red", lwd = 1)

cv_rpmse <- ggplot() +
  geom_density(mapping = aes(x=rpmse)) +
  xlab("rpmse") +
  ylab("Frequency") +
  geom_vline(xintercept = mean(rpmse), col = "red", lwd = 1)

cv_wid <- ggplot() +
  geom_density(mapping = aes(x=wid)) +
  xlab("wid") +
  ylab("Frequency") +
  geom_vline(xintercept = mean(wid), col = "red", lwd = 1)

cv_cvg <- ggplot() +
  geom_density(mapping = aes(x=cvg)) +
  xlab("cvg") +
  ylab("Frequency") +
  geom_vline(xintercept = mean(cvg), col = "red", lwd = 1)
```

```
grid.arrange(cv_rpmse, cv_bias, cv_wid, cv_cvg)
```



```
paste("Mean coverage =",  round(mean(cvg), 3))
```

```
## [1] "Mean coverage = 0.98"
```

```
# paste("Mean bias =", round(mean(bias), 3))
paste("Mean width =", round(mean(wid), 3))
```

```
## [1] "Mean width = 5.167"
```

```
paste("Mean RPMSE =", round(mean(rpmse), 3))
```

```
## [1] "Mean RPMSE = 1.232"
```

After performing cross-validation, we obtained the mean RPMSE, coverage, and width of prediction intervals. The mean RPMSE is 1.232, meaning that the root predictive mean squared error of the residuals is 1.232. Comparing this to the standard error of `WHC`, 1.356. The mean RPMSE is lower than this, indicating that there is some predictive value to the model. Additionally, the mean coverage of the prediction intervals is 0.98, which means that 98% of the time, the prediction intervals captured the true mean. This is a good sign as we should expect a coverage of approximately 95% for 95% confidence intervals. Finally, the width of the prediction intervals is 5.167. This means that, on average, the prediction intervals are about 5 units of Water Holding Capacity wide.

7. Carry out a hypothesis test that locations with higher yield had higher WHC (which would make sense because more water would be available for the plant to use). Include a confidence interval for the effect of `Yield` on `WHC` and interpret this interval.

```
Yield_matrix = c(1, 1, 0)
NoYield_matrix = c(1, 0, 0)
glht_matrix <- c(0,1,0)

mytest <- glht(gls_model, linfct = t(glht_matrix), rhs = 0, alternative="greater")
summary(mytest)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: gls(model = WHC ~ Yield + EC, data = data, correlation = corExp(form = ~Lon +
##     Lat, nugget = TRUE), method = "ML")
##
## Linear Hypotheses:
##        Estimate Std. Error z value  Pr(>z)
## 1 <= 0  0.02578    0.00946   2.725 0.00321 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
mytest2 <- glht(gls_model, linfct = t(glht_matrix), rhs = 0, alternative="two.sided")
confint(mytest2)
```

```
##
##   Simultaneous Confidence Intervals
##
## Fit: gls(model = WHC ~ Yield + EC, data = data, correlation = corExp(form = ~Lon +
##     Lat, nugget = TRUE), method = "ML")
##
## Quantile = 1.96
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##        Estimate lwr     upr
## 1 == 0 0.02578  0.00724 0.04432
```

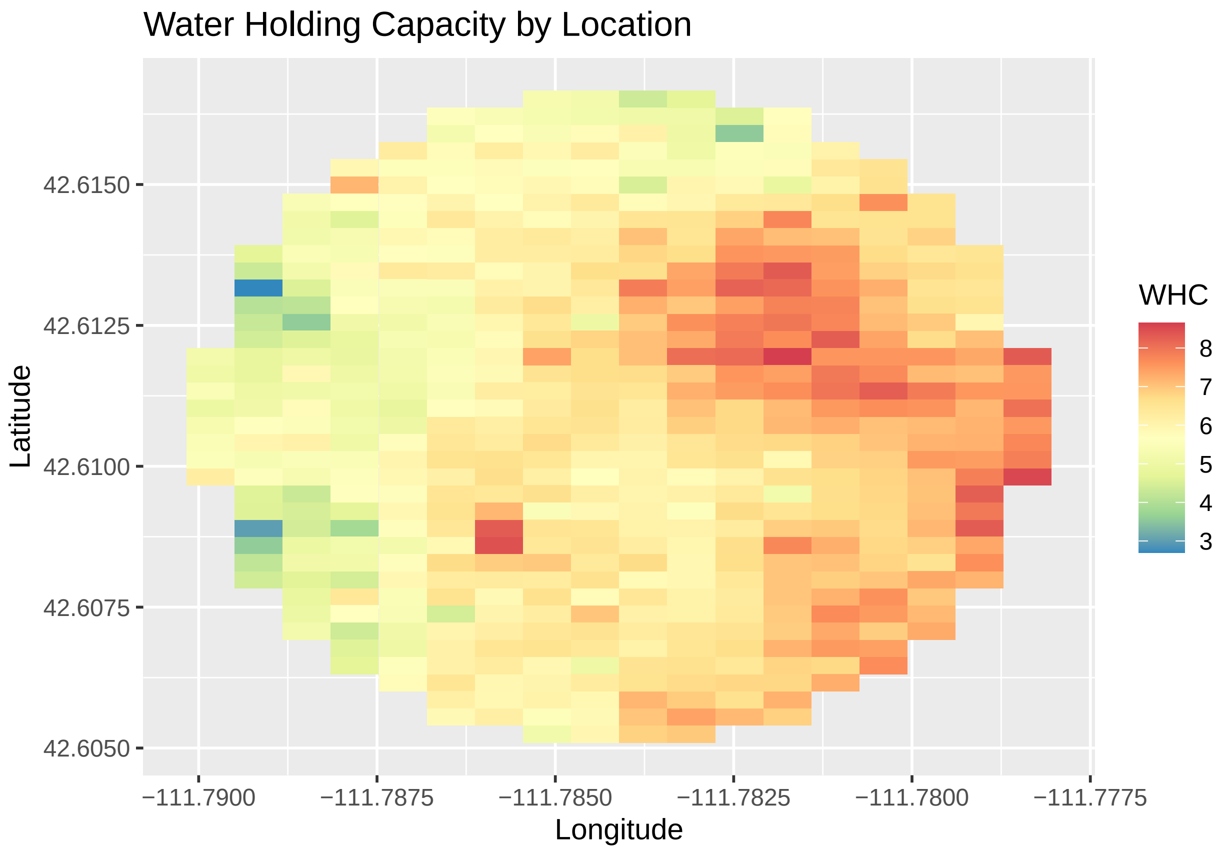
For a hypothesis test, we set H_0 : higher yield has no effect on water holding capacity and set H_A : higher yield means higher water holding capacity. Mathematically, this would be $H_0 : \beta_{\text{Yield}} = 0$ and $H_0 : \beta_{\text{Yield}} > 0$. A general linear hypothesis test results in a p-value of 0.00321, so reject the null hypothesis and conclude there is evidence that β_{Yield} has an effect on water holding capacity.

For a confidence interval to determine the effect of `Yield` on `WHC`, we are 95% confident that the effect of `Yield` on `WHC` is between 0.00724 and 0.04432.

8. Predict WHC at all the locations where WHC is missing. Provide a plot of your predictions.

```
data_na = read_table("WaterHoldingCapacity.txt", show_col_types = FALSE) %>%
  filter(is.na(WHC))


data_pred = predictgls(gls_model, data_na[c(1,2,3,4)]) %>%
  mutate(
    WHC = Prediction
  )

data_pred = rbind(data, tibble(data_pred[c(1, 2, 3, 4, 9)]))
```

```
ggplot(data = data_pred, mapping = aes(x = Lon, y = Lat, fill = WHC)) +
  geom_tile() +
  scale_fill_distiller(palette="Spectral",na.value=NA) +
  labs(title = "Water Holding Capacity by Location",
       x = "Longitude",
       y = "Latitude")
```


Water Holding Capacity by Location
Latitude
Longitude
WHC