Irrigated Agriculture Data Analysis

Tetsuya Chau

2022-03-30

Problem Background

In addition to drought, increased competition for water resources, aquifer depletion, and climate change increase water scarcity for irrigated agriculture. Society's ability to deal with water scarcity while still maintaining sufficient agriculture to support life is dependent upon the efficient use of water. That is, farmers need to efficiently manage their limited water resources by using only the necessary amount of water to grow their crops and allocating additional water for urban use.

Water holding capacity (WHC) describes how much plant available water a specific soil can store. Knowing the variation of WHC for the soils within an irrigated field provides information to the farmer about how much water can be supplied to plants from the soil in a specific area of the field and how much irrigation water is required to replenish a depleted soil without leaching. Obtaining measurements of WHC, however, is an expensive and time consuming process. Water holding capacity can be estimated in the laboratory from soil cores collected from multiple depths and field positions or in the field by measuring water content over a time period long enough to observe a typical range of soil water conditions. In the latter case, permanent tubes (that reach a depth of 1.5 meters) must be installed at each location in the field and at regular (e.g. weekly) time intervals, farmers manually insert a neutron probe into each tube to measure, via reflectometry, soil water content at various depths. Thus, the cost and time requirements limit the utility of both of these methods for precision irrigation applications.

In order to help farmers understand the WHC of soil across their agriculture field, this analysis seeks to use sparse WHC data on a field to predict WHC capacity at locations where measurements are not available. Specifically, the dataset we consider for this analysis has the following variables:

Variable Name and Description

*Lon - Longitude location on the field

*Lat - Latitude location on the field

*Yield - Crop Yield at the location

*EC - Electroconductivity at the location

*WHC - Water Holding Capacity at the location

To analyze this dataset, do the following:

Analysis Questions:

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.
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.

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.

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

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

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.

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.

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

```
library(ggplot2)
library(gstat)
library(nlme)
library(car)
```

```
## Loading required package: carData
```

```
library(multcomp)
```

```
## Loading required package: mvtnorm
```

```
## Loading required package: survival
```

```
## Loading required package: TH.data
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'TH.data'
```

```
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──
## ✓ tibble  3.1.6     ✓ dplyr   1.0.8
## ✓ tidyr   1.2.0     ✓ stringr 1.4.0
## ✓ readr   2.1.2     ✓ forcats 0.5.1
## ✓ purrr   0.3.4
```

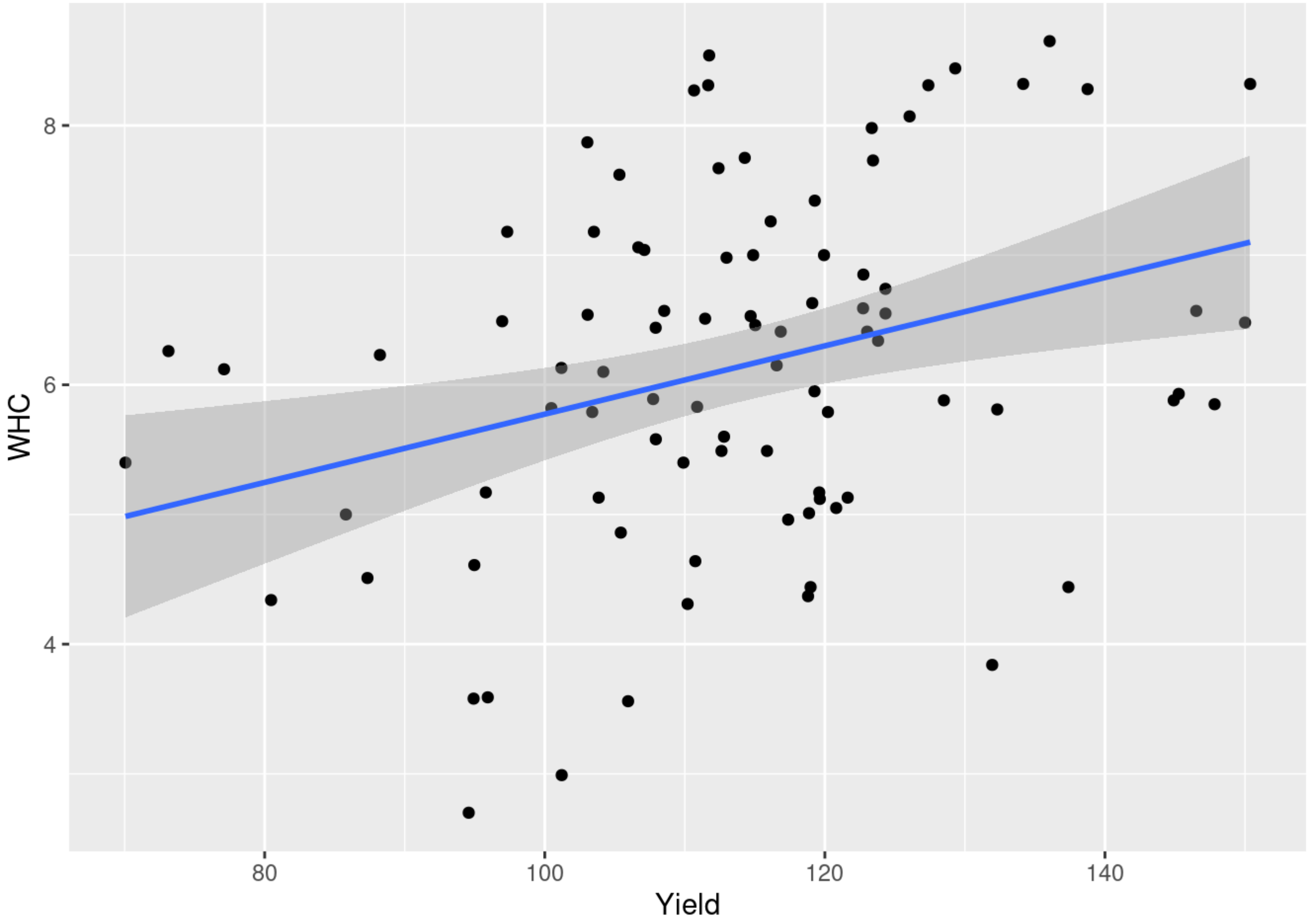
```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::collapse() masks nlme::collapse()
## x dplyr::filter()   masks stats::filter()
## x dplyr::lag()      masks stats::lag()
## x dplyr::recode()   masks car::recode()
## x dplyr::select()   masks MASS::select()
## x purrr::some()     masks car::some()
```

```
library(dplyr)
```

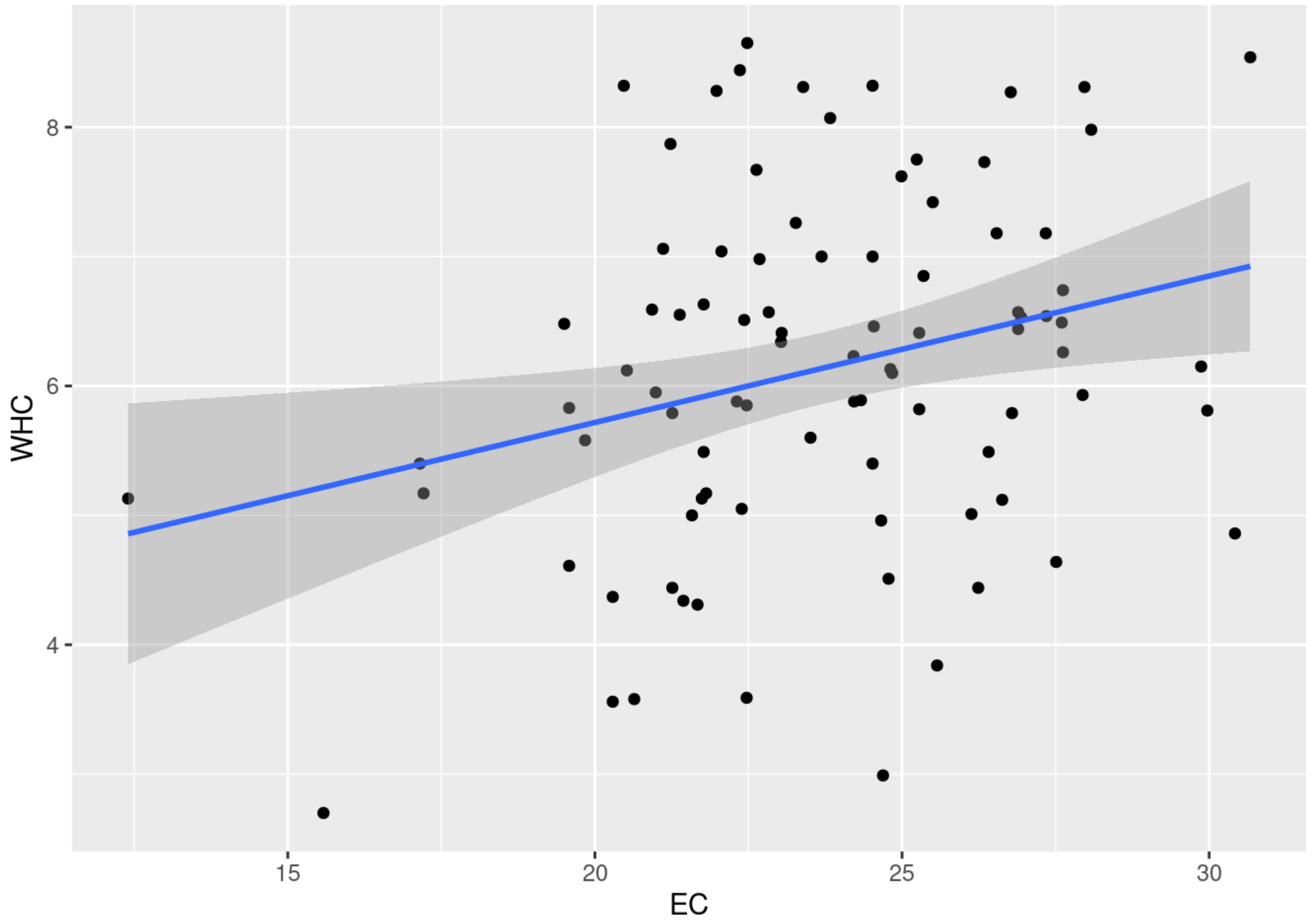
```
df <- read.csv("/cloud/project/WaterHoldingCapacity.txt", sep="")
dfr <- na.omit(df)
```

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.

```
ggplot(data=dfr,mapping=aes(x=Yield, y=WHC)) + geom_point() + geom_smooth(method="lm",formula='y
~x')
```



```
ggplot(data=dfr,mapping=aes(x=EC, y=WHC)) + geom_point() + geom_smooth(method="lm",formula='y~
x')
```



There is a positive linear relationship for WHC vs Yield and WHC vs EC although their relationships are not very strong.

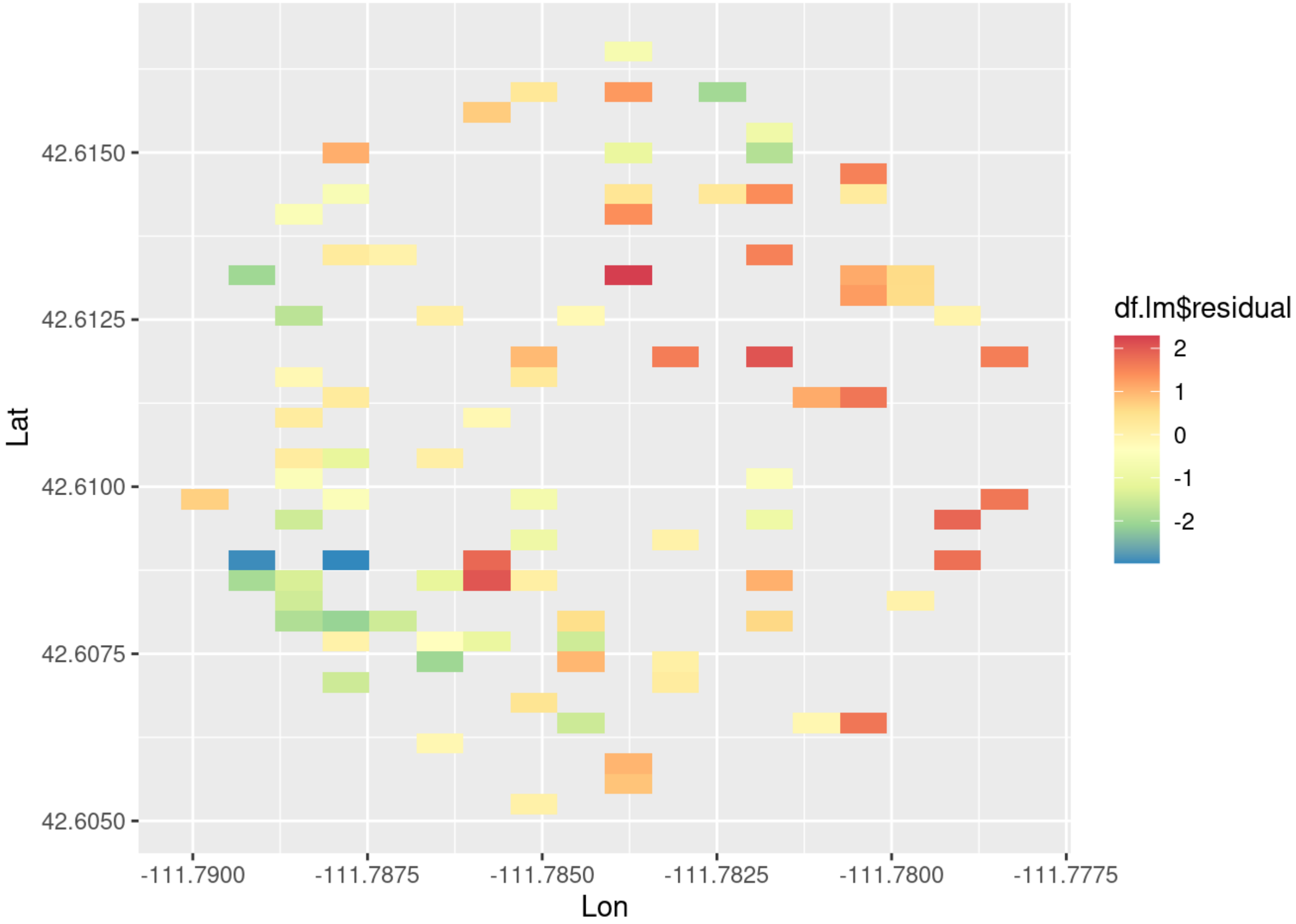
2. Fit an independent MLR model with a linear effect between Yield, EC and the response variable WHC. Explore the residuals to see if there is evidence of spatial correlation by mapping the residuals and plotting the variogram of the residuals.

```
df.lm <- lm(formula = WHC ~ Yield + EC, data = dfr)
```

```
ggplot(data=dfr ,mapping=aes(x=Lon, y=Lat, fill=df.lm$residual)) + geom_raster() + scale_fill_distiller(palette="Spectral",na.value=NA)
```

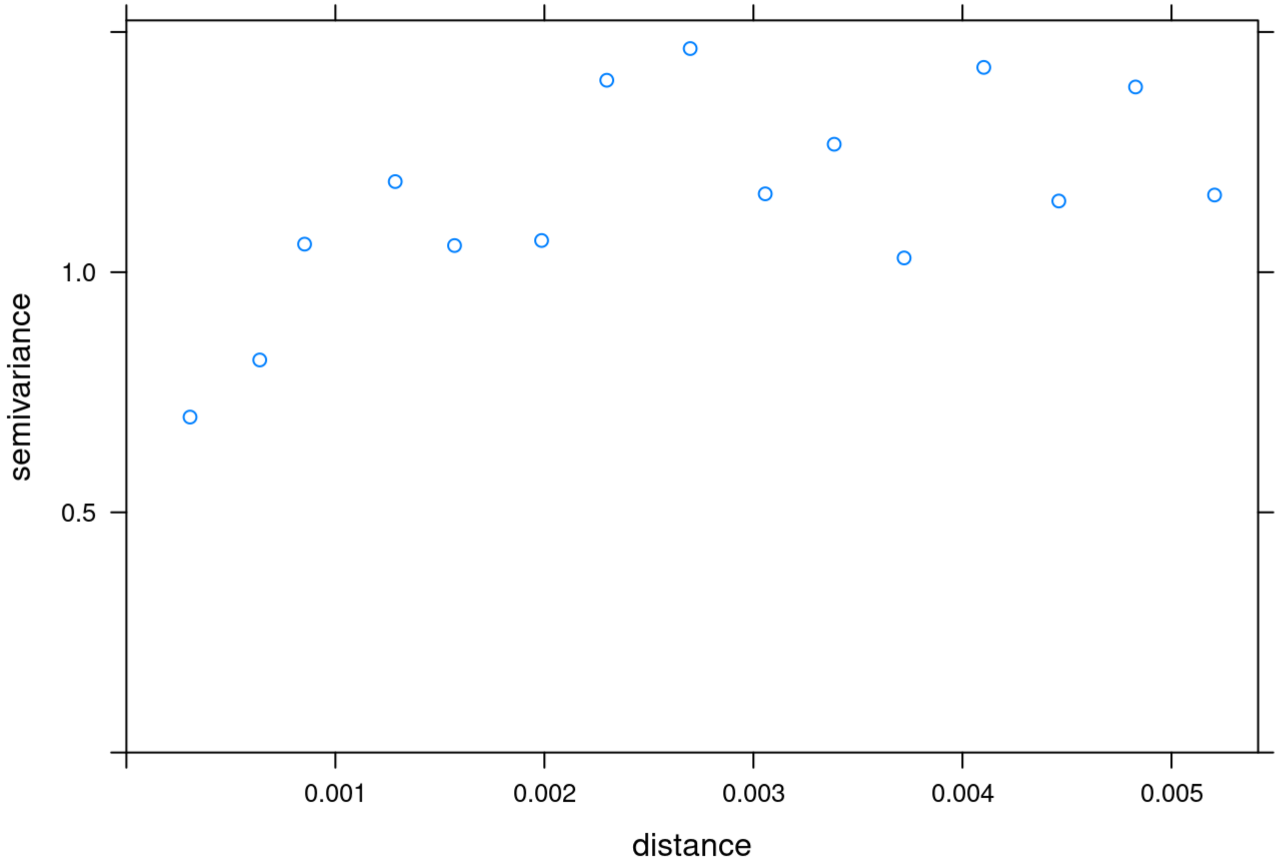
```
## Warning: Raster pixels are placed at uneven horizontal intervals and will be
## shifted. Consider using geom_tile() instead.
```

```
## Warning: Raster pixels are placed at uneven vertical intervals and will be
## shifted. Consider using geom_tile() instead.
```



```
myVario <- variogram(object=WHC~Yield+EC, locations=~Lon+Lat, data=dfr)

plot(myVario)
```



There is a sign of correlation based on looking at the variogram ans seeing that the data points don't follow a reatlively straight pattern.

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.

```
e1 <- gls(model=WHC~Yield+EC, data=dfr, correlation=corExp(form=~Lon+Lat, nugget=TRUE), method
="ML")

s1 <- gls(model=WHC~Yield+EC, data=dfr, correlation=corSpher(form=~Lon+Lat, nugget=TRUE), method
="ML")

g1 <- gls(model=WHC~Yield+EC, data=dfr, correlation=corGaus(form=~Lon+Lat, nugget=TRUE), method
="ML")
```

```
AIC(e1)
```

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

```
AIC(s1)
```

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

```
AIC(g1)
```

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

We will use the exponential correlation function because it gave us the lowest AIC value.

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

We will be using the exponential correlation structure for the analysis of this data.

y~N(X*B,(sigma^2)*R)

-y is the vector of the response variable, water holding capacity at the location (WHC)

-X is the matrix of all the explanatory variables in the model which are crop yield at the location and electroconductivity at the location.

-B is the coefficients of the explanatory variables in the model which are crop yield at the location and electroconductivity at the location.

-sigma^2 is the the constant that contributes to the variance.

-R is the correlation matrix of the exponential correlation function.

p(s_i,s_j)=exp(-(||s_i-s_j||)/phi)

-s is the location of each data point.

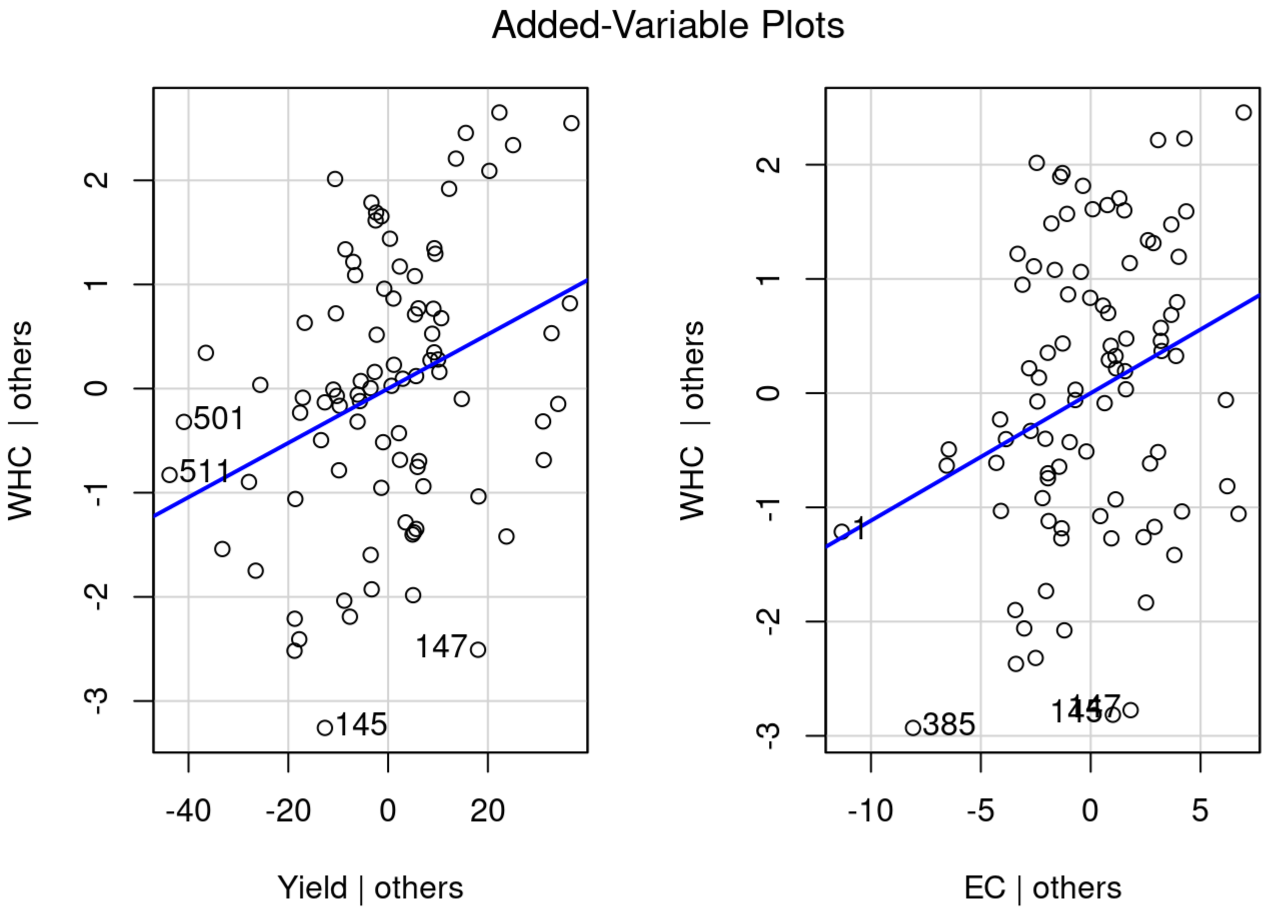
-phi is another parameter we want to estimate from the data. As phi goes up, the range of the spatial correlation goes up with it.

-omega is the same location variation.

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

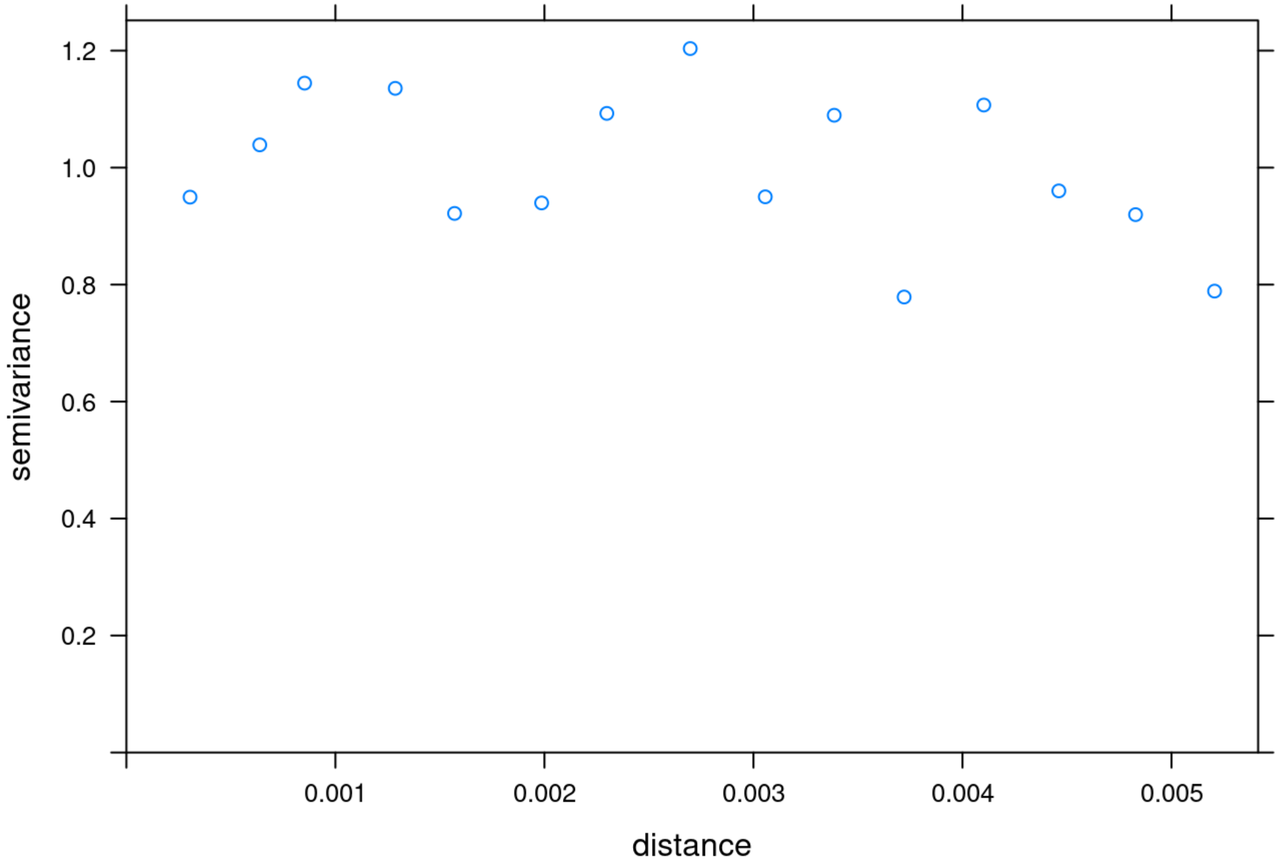
We already fit the model in step 3 of the homework.

```
avPlots(df.lm)
```



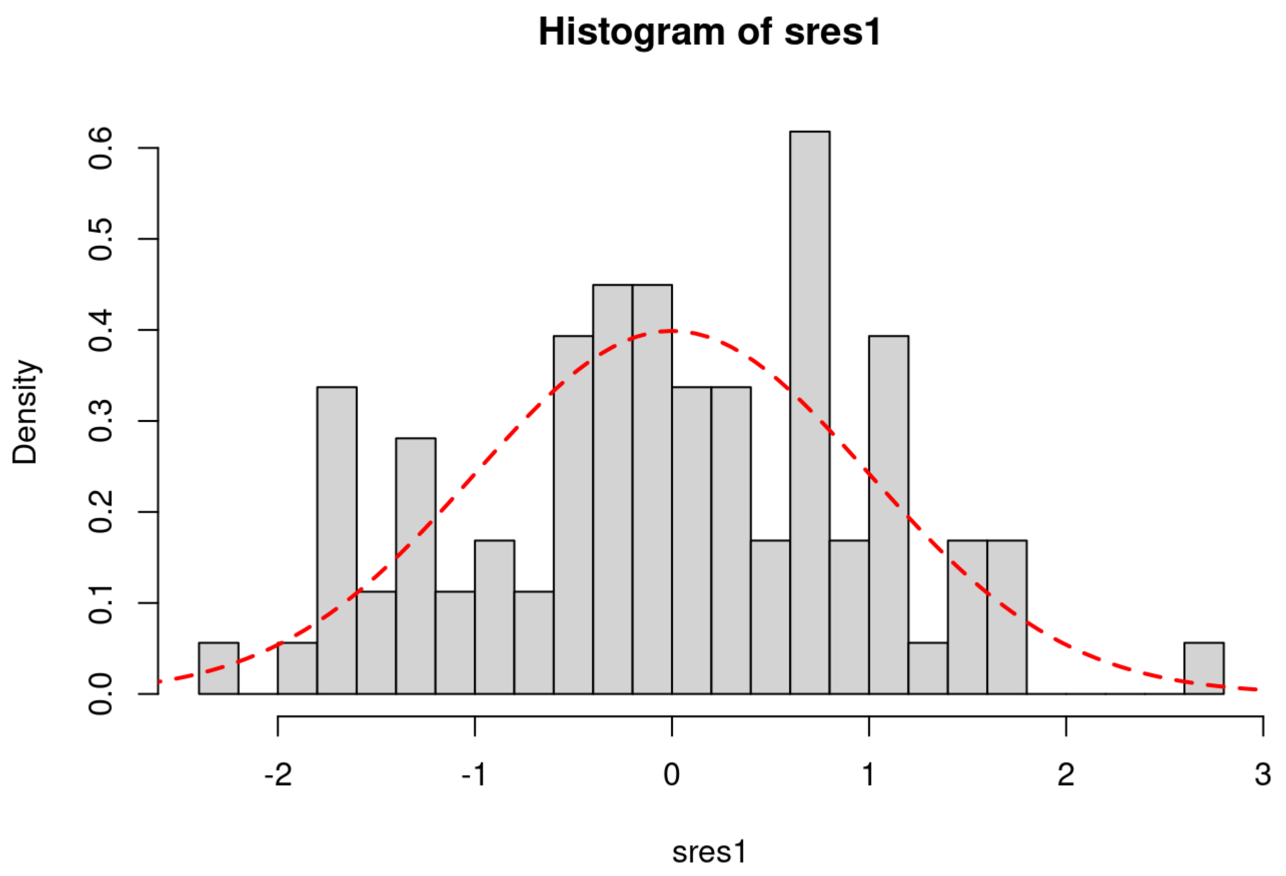
The line of best fit for the scatterplot of the dependent variable and the numeric independent variables look linear so we can assume linearity.

```
source("stdres.R")
sres1 <- stdres.gls(e1)
residDF1 <- data.frame(Lon=dfr$Lon, Lat=dfr$Lat, decorrResid=sres1)
residVariogram1 <- variogram(object=decorrResid~1, locations=~Lon+Lat, data=residDF1)
plot(residVariogram1)
```



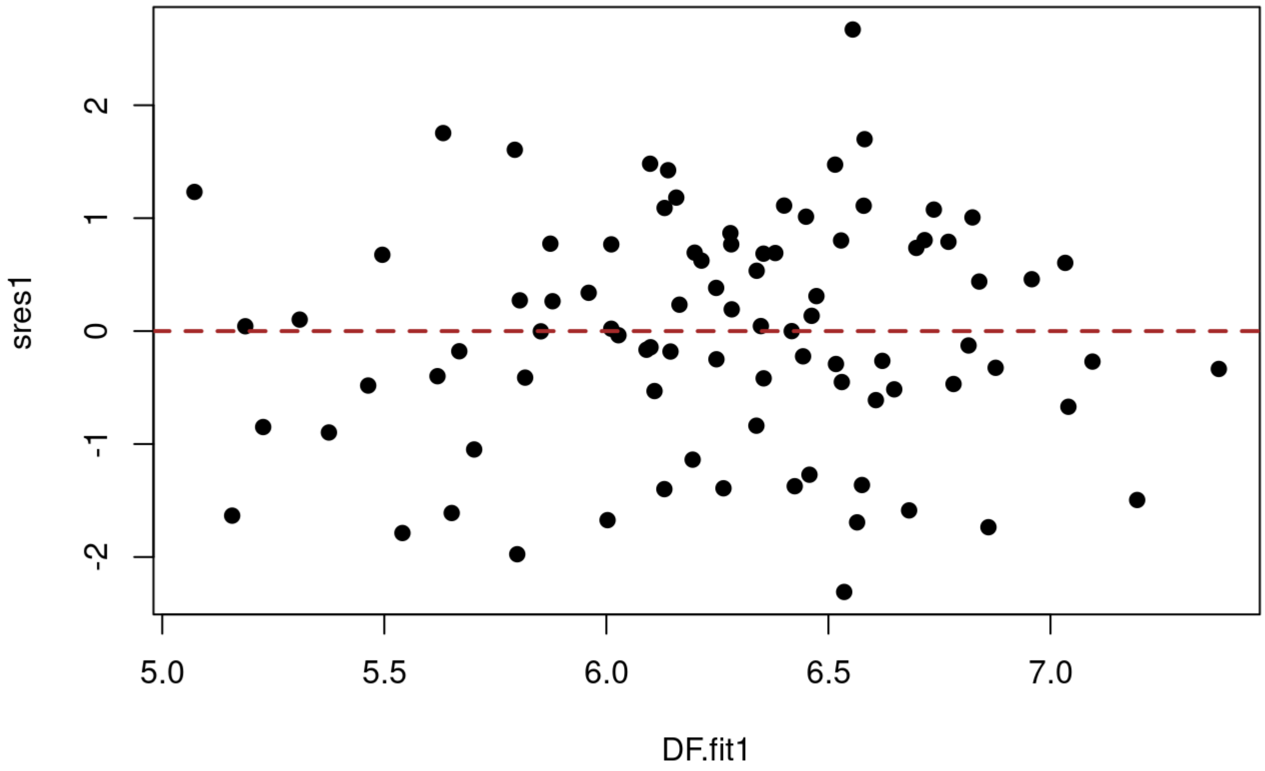
The data points in the variogram looks like they follow a pretty flat pattern so we can assume independence here.

```
hist(sres1,freq = FALSE,breaks = 20)
curve(dnorm,from = -3,to = 3,col = "red",lwd = 2,
lty = 2,add = TRUE)
```



The histogram follows a gaussian aka normal curve so we can assume normality here.

```
DF.fit1 <- fitted(e1)
plot(DF.fit1,sres1,pch=19)
abline(0,0,col = "brown",lwd = 2,lty = 2)
```



The data points in the fitted vs residual scatter plot looks evenly scattered so we can assume equal variance.

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.

```
source("predgls.R")

n.cv <- 50

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

```
##
  |
  |                                                                      |   0%
```

```
#n.cv <- 2 #Number of CV studies to run
n.test <-  floor(.2*nrow(dfr))#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)
rpmse.lm <- rep(x=NA, times=n.cv)
bias.lm <- rep(x=NA, times=n.cv)
wid.lm <- rep(x=NA, times=n.cv)
cvg.lm <- rep(x=NA, times=n.cv)
for(cv in 1:n.cv){
  ## Select test observations
  test.obs <- sample(x=1:nrow(dfr), size=n.test)

  ## Split into test and training sets
  test.set <- dfr[test.obs,]
  train.set <- dfr[-test.obs,]

  ## Fit a lm() using the training data

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

  train.lm <- lm(formula=WHC~Yield+EC,data=train.set)

  ## Generate predictions for the test set
  my.preds.gls <- predictgls(glsobj=train.gls,newdframe=test.set)

  my.preds.lm <- predict.lm(train.lm, newdata=test.set, interval="prediction")

  ## Calculate bias

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

  bias.lm[cv] <- mean(my.preds.lm[,'fit']-test.set[['WHC']])

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

  rpmse.lm[cv] <- (test.set[['WHC']]-my.preds.lm[,'fit'])^2 %>% mean() %>% sqrt()

  ## Calculate Coverage


  cvg[cv] <- ((test.set[['WHC']] > my.preds.gls[,'lwr']) & (test.set[['WHC']] < my.preds.gls[,'u
pr'])) %>% mean()

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

  ## Calculate Width
```

```r
  wid[cv] <- (my.preds.gls[,'upr'] - my.preds.gls[,'lwr']) %>% mean()

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

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

```
##
  |
  |=                                                                     |   2%
  |
  |===                                                                   |   4%
  |
  |====                                                                  |   6%
  |
  |======                                                                |   8%
  |
  |=======                                                               |  10%
  |
  |========                                                              |  12%
  |
  |==========                                                            |  14%
  |
  |===========                                                           |  16%
  |
  |=============                                                         |  18%
  |
  |==============                                                        |  20%
  |
  |===============                                                       |  22%
  |
  |=================                                                     |  24%
  |
  |==================                                                    |  26%
  |
  |====================                                                  |  28%
  |
  |=====================                                                 |  30%
  |
  |======================                                                |  32%
  |
  |========================                                              |  34%
  |
  |=========================                                             |  36%
  |
  |===========================                                           |  38%
  |
  |============================                                          |  40%
  |
  |=============================                                         |  42%
  |
  |===============================                                       |  44%
  |
  |================================                                      |  46%
  |
  |==================================                                    |  48%
  |
  |===================================                                   |  50%
  |
  |====================================                                  |  52%
  |
  |======================================                                |  54%
  |
  |=======================================                               |  56%
  |
  |=========================================                             |  58%
  |
  |==========================================                            |  60%
  |
  |===========================================                           |  62%
  |
  |=============================================                         |  64%
  |
  |==============================================                        |  66%
  |
  |================================================                      |  68%
  |
  |=================================================                     |  70%
  |
  |==================================================                    |  72%
  |
  |====================================================                  |  74%
  |
  |=====================================================                 |  76%
  |
  |=======================================================               |  78%
  |
  |========================================================              |  80%
  |
  |=========================================================             |  82%
  |
  |===========================================================           |  84%
  |
  |============================================================          |  86%
  |
  |==============================================================        |  88%
  |
  |===============================================================       |  90%
  |
  |================================================================      |  92%
  |
  |==================================================================    |  94%
  |
  |===================================================================   |  96%
  |
  |===================================================================== |  98%
  |
  |======================================================================| 100%
```

```
close(pb)
```

```
print(mean(bias))
```

```
## [1] 0.01384124
```

```
print(mean(rpmse))
```

```
## [1] 1.036747
```

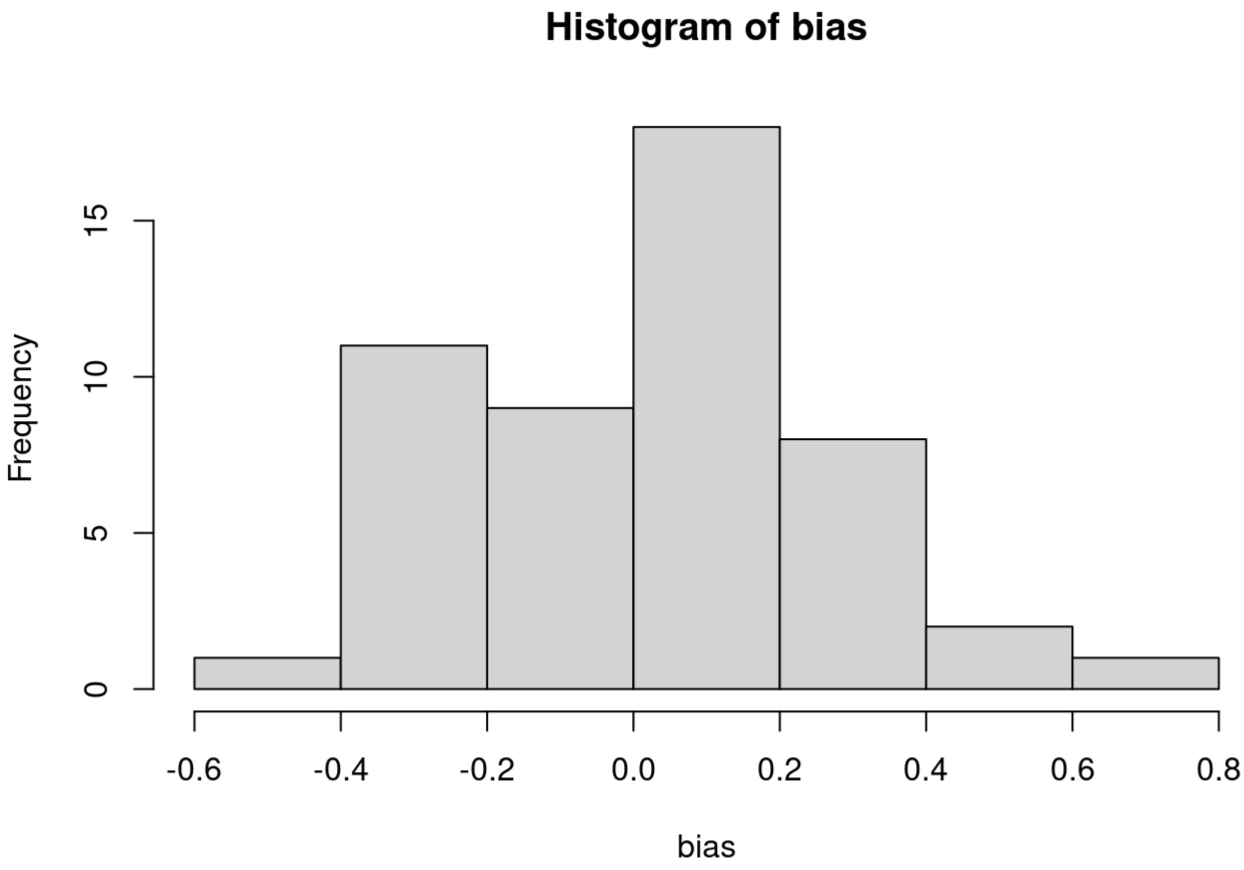
```
print(mean(cvg))
```

```
## [1] 0.98
```

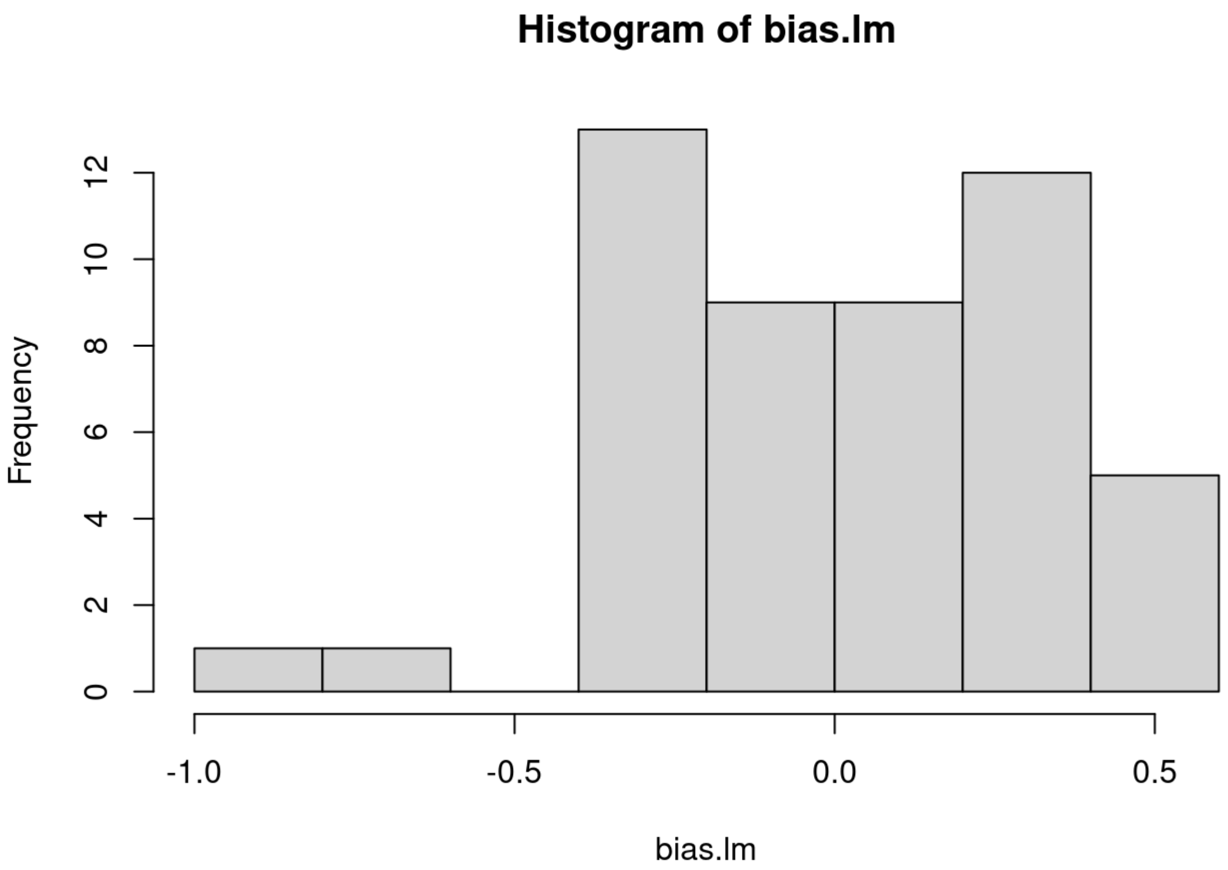
```
print(mean(wid))
```

```
## [1] 4.366123
```

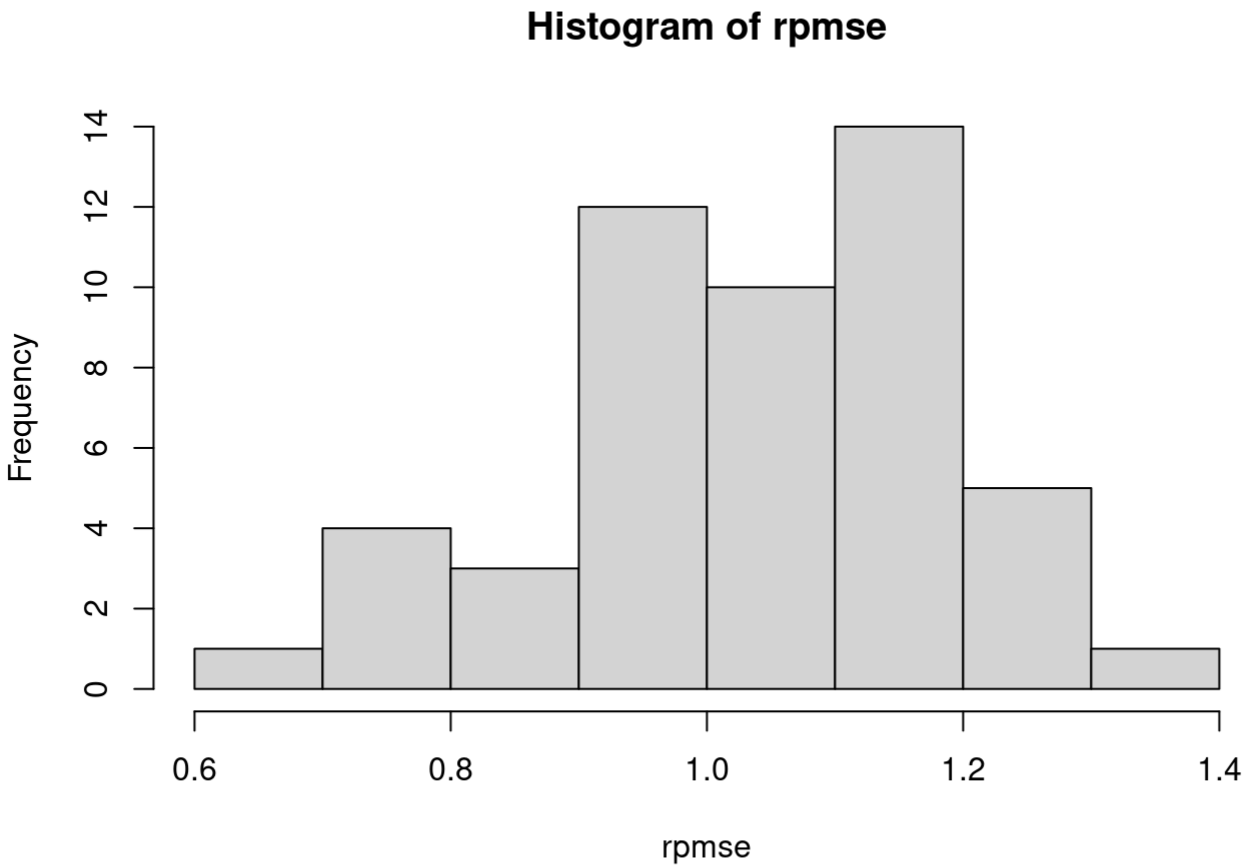
```
hist(bias)
```



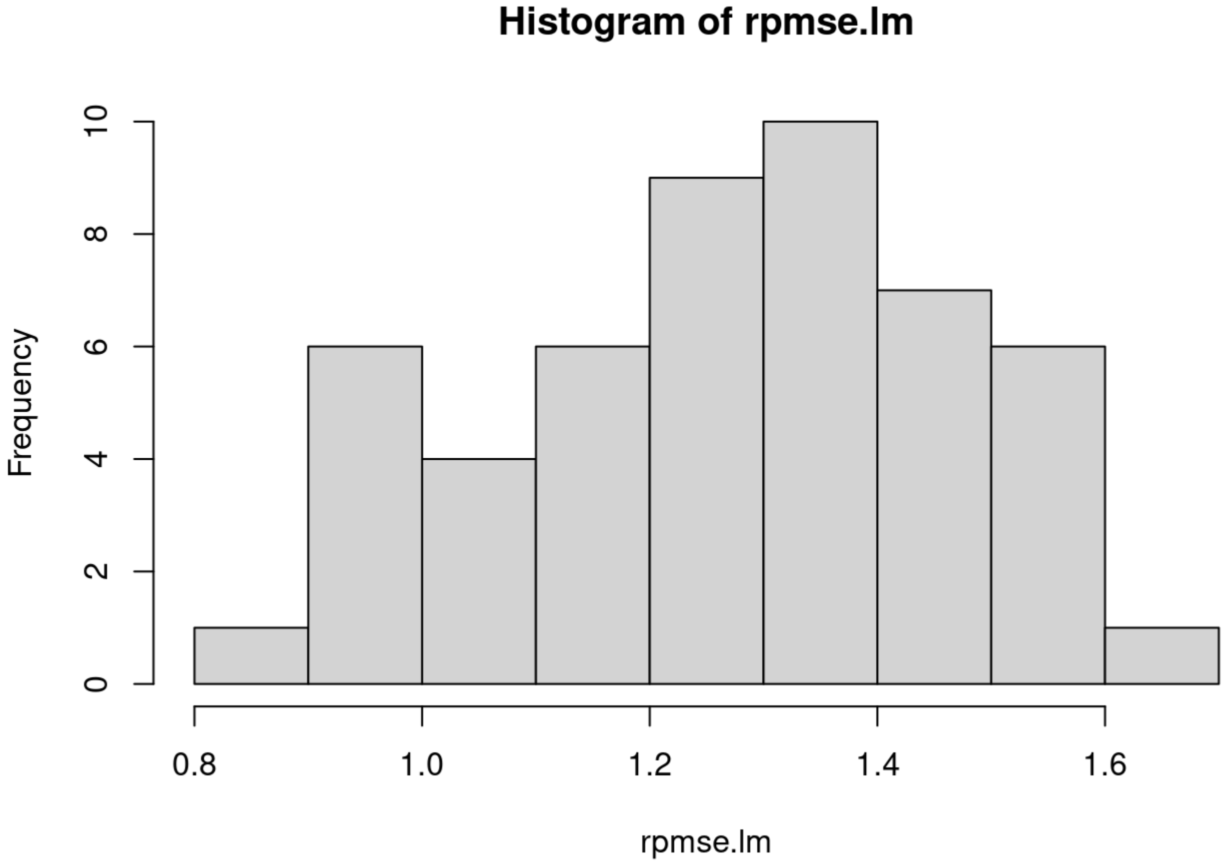
```
hist(bias.lm)
```



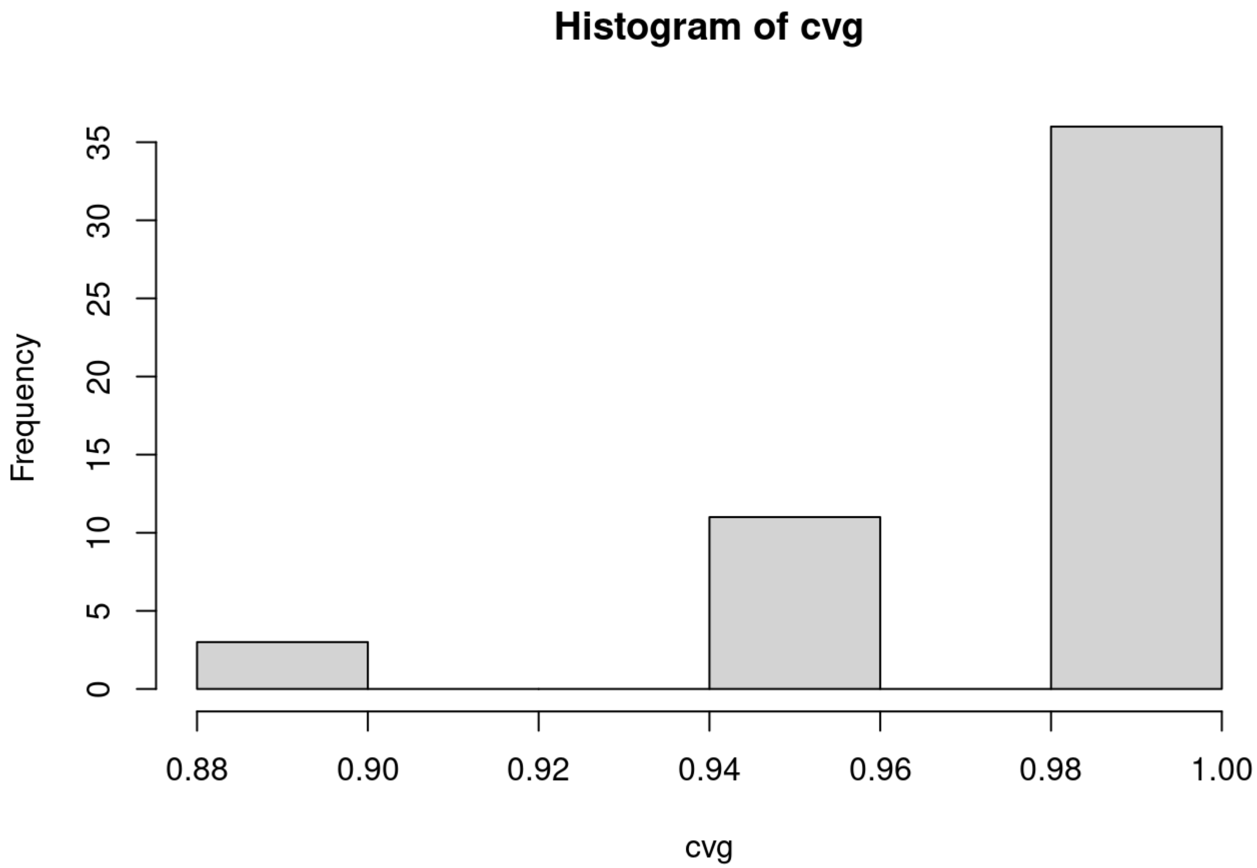
```
hist(rpmse)
```



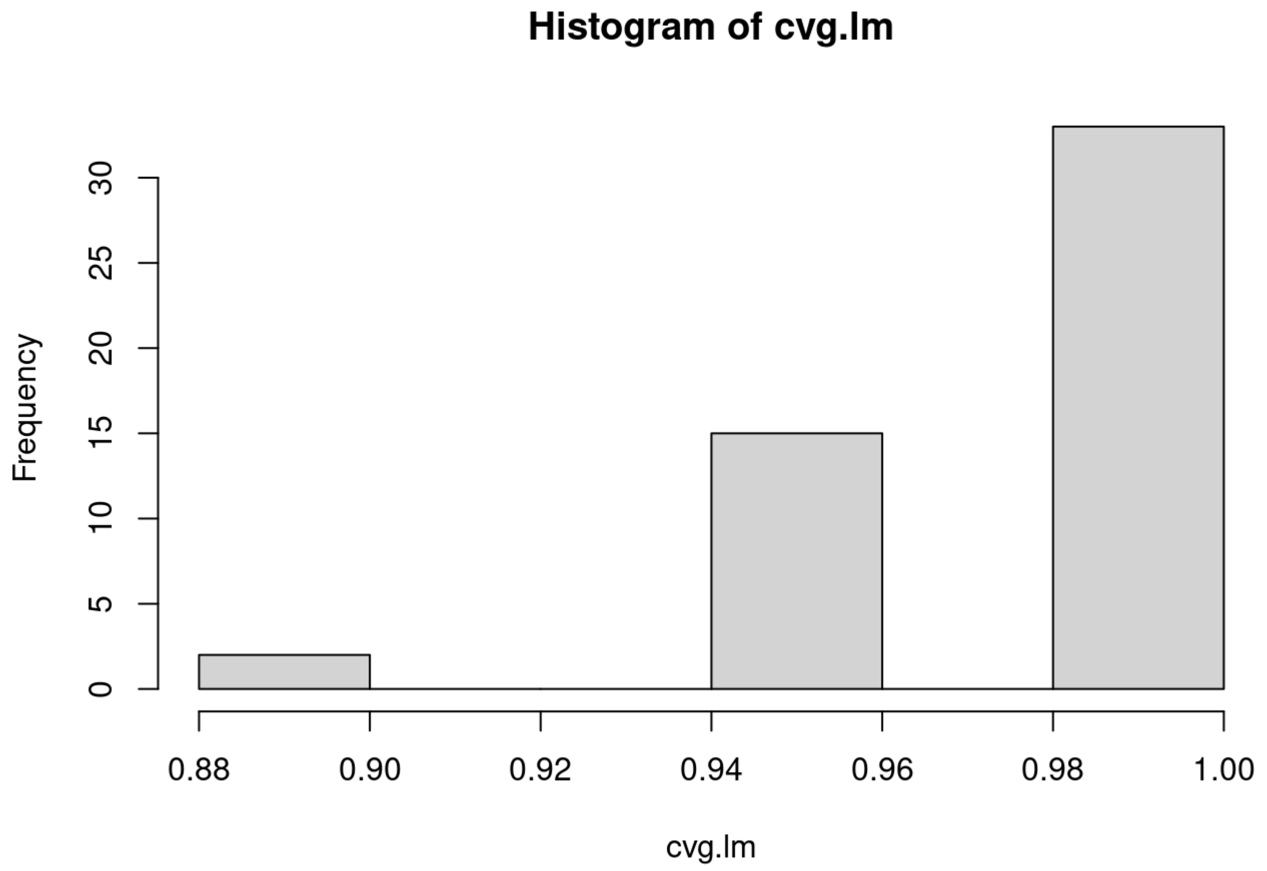
```
hist(rpmse.lm)
```


Histogram of rpmse.lm
Frequency
0
2
4
6
8
10
0.8
1.0
1.2
1.4
1.6
rpmse.lm

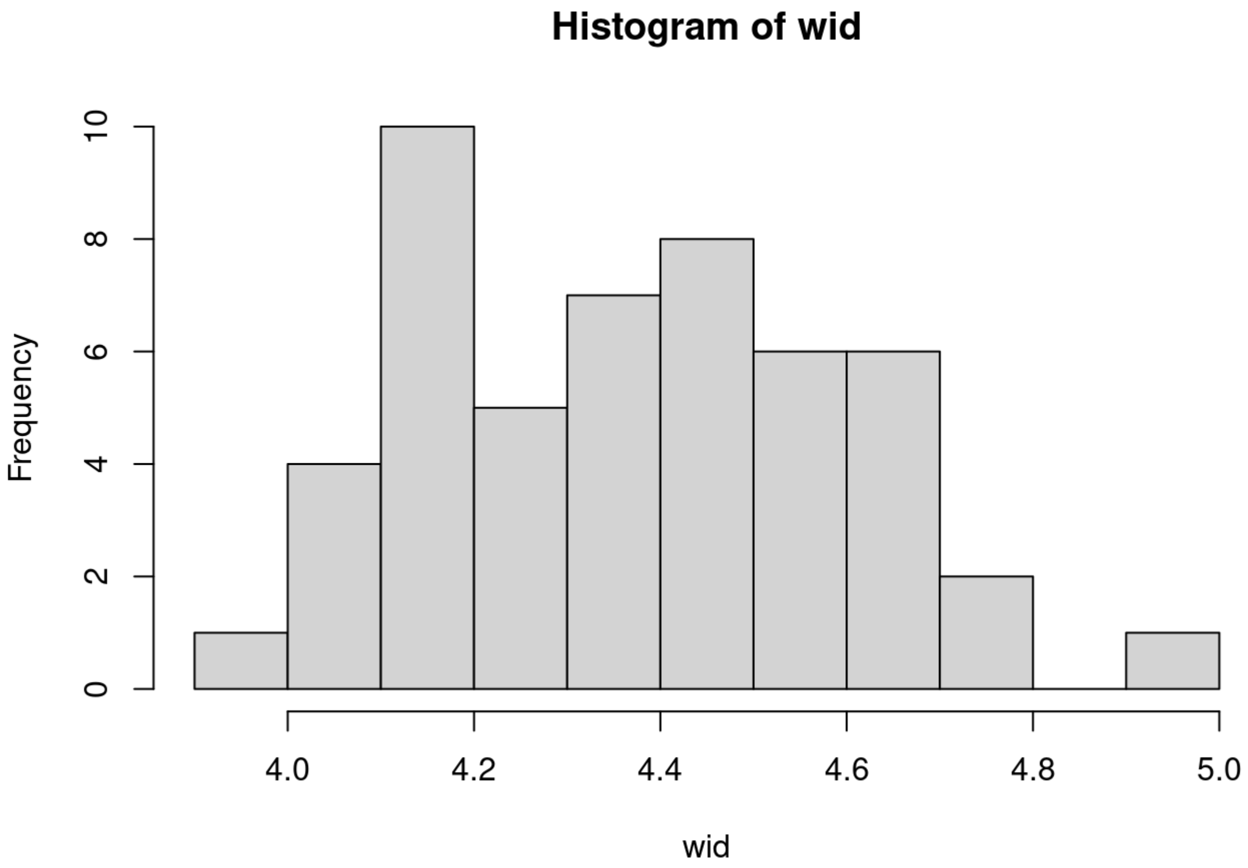
```
hist(cvg)
```


Histogram of cvg
Frequency
0
5
10
15
20
25
30
35
0.88
0.90
0.92
0.94
0.96
0.98
1.00
cvg

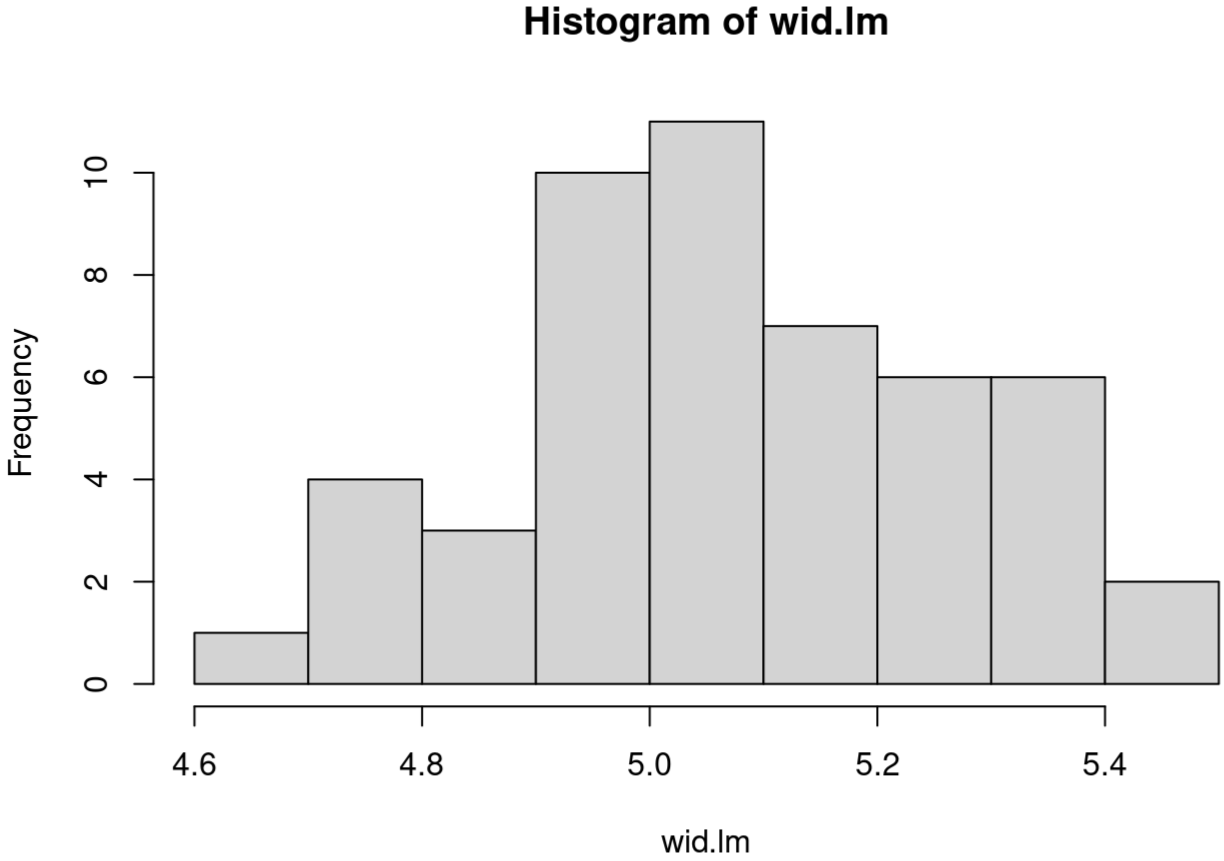
```
hist(cvg.lm)
```


Histogram of cvg.lm
Frequency
0
5
10
15
20
25
30
0.88
0.90
0.92
0.94
0.96
0.98
1.00
cvg.lm

```
hist(wid)
```



```
hist(wid.lm)
```



The bias is -0.01966238 which means that the model fits the data well since its a very small bias.

The RPMSE of 1.0177 is smaller than the standard deviation of the y-variable so we can say that the model fits the data well.

The width is 4.38 which means that there is uncertainty and we want it to be as small as possible. It's not horrible but it's not great either given that the max is 8.65 and the min is 2.7.

The coverage is 0.9835294 which means that 98% of values are in the 95% prediction interval.

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.

Ho: Locations with higher yield do not have higher WHC (beta_yield=0). Ha: Locations with higher yield have higher WHC (beta_yield>0).

The p-value is 0.0031 which means that we can reject the null and conclude that locations with higher yield have higher WHC (with alpha=0.05).

Because the confidence of (0.00724 0.04432) does not include 0, we can conlude that the test is significant.

We are 95% confident that as yield goes up by 1, the WHC goes up by between 0.00724 and 0.04432.

```
coef(e1)
```

```
## (Intercept)       Yield          EC
##  1.56633579  0.02578200  0.07399449
```

```
df.yield <- matrix(c(0,1,0), nrow=1)
#urban <- matrix(c(1,0,0,0,1), nrow=1)
#diff_d <- urban - savannah
df.yield.p <- glht(e1, linfct = df.yield, alternative = "two.sided")
summary(df.yield.p)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: gls(model = WHC ~ Yield + EC, data = dfr, 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  0.00642 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```
confint(df.yield.p)
```

```
##
##   Simultaneous Confidence Intervals
##
## Fit: gls(model = WHC ~ Yield + EC, data = dfr, 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
```

```
confint(e1)
```

```
##                    2.5 %     97.5 %
## (Intercept) -1.167987207 4.30065879
## Yield        0.007240462 0.04432354
## EC           0.002925613 0.14506337
```

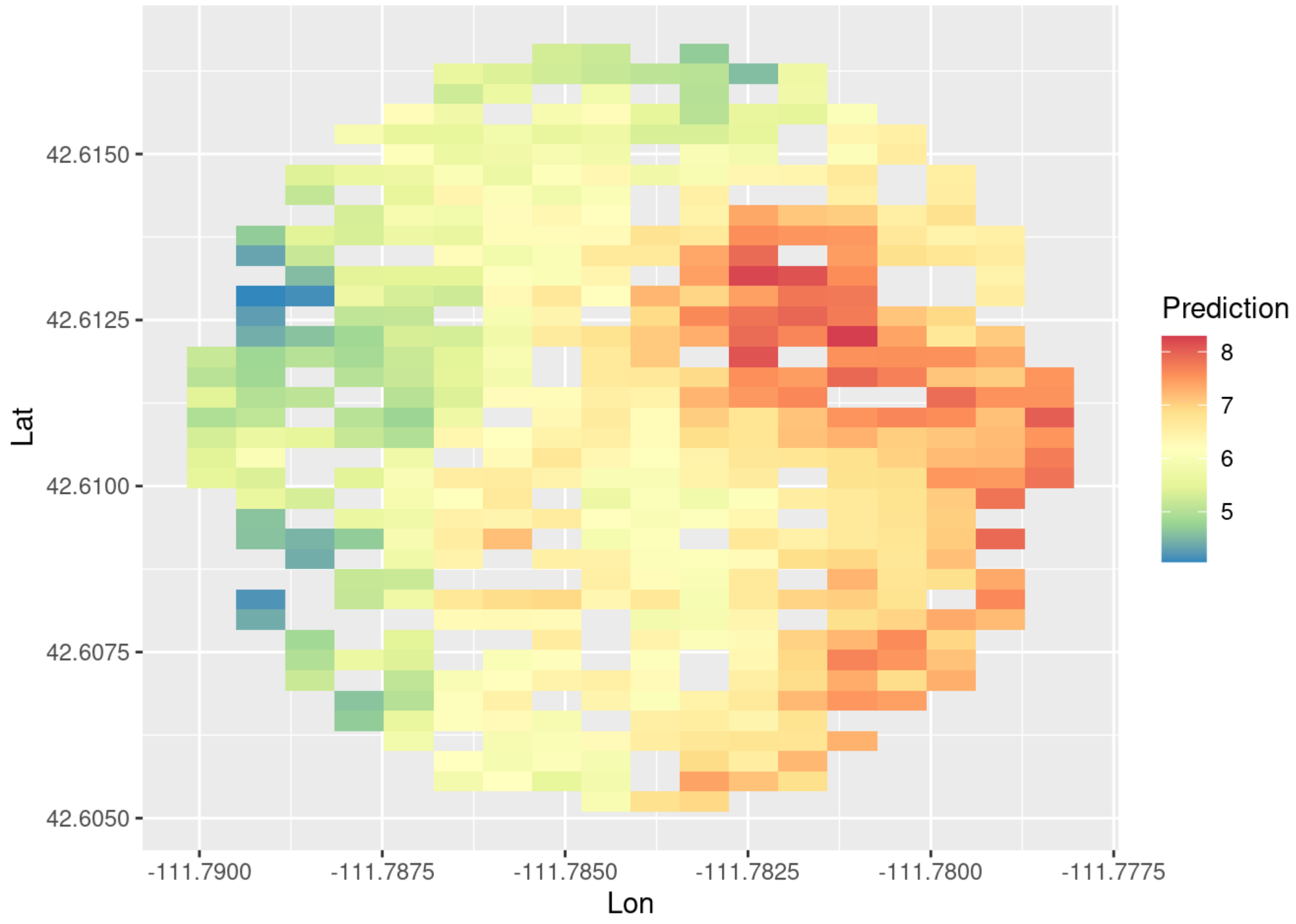
8.Predict WHC at all the locations where WHC is missing. Provide a plot of your predictions.

```
df.pred <- predictgls(glsobj=e1,newdframe=df[is.na(df$WHC),])

ggplot(data=df.pred ,mapping=aes(x=Lon, y=Lat, fill=Prediction)) + geom_raster() + scale_fill_distiller(palette="Spectral",na.value=NA)
```

```
## Warning: Raster pixels are placed at uneven horizontal intervals and will be
## shifted. Consider using geom_tile() instead.
```

```
## Warning: Raster pixels are placed at uneven vertical intervals and will be
## shifted. Consider using geom_tile() instead.
```

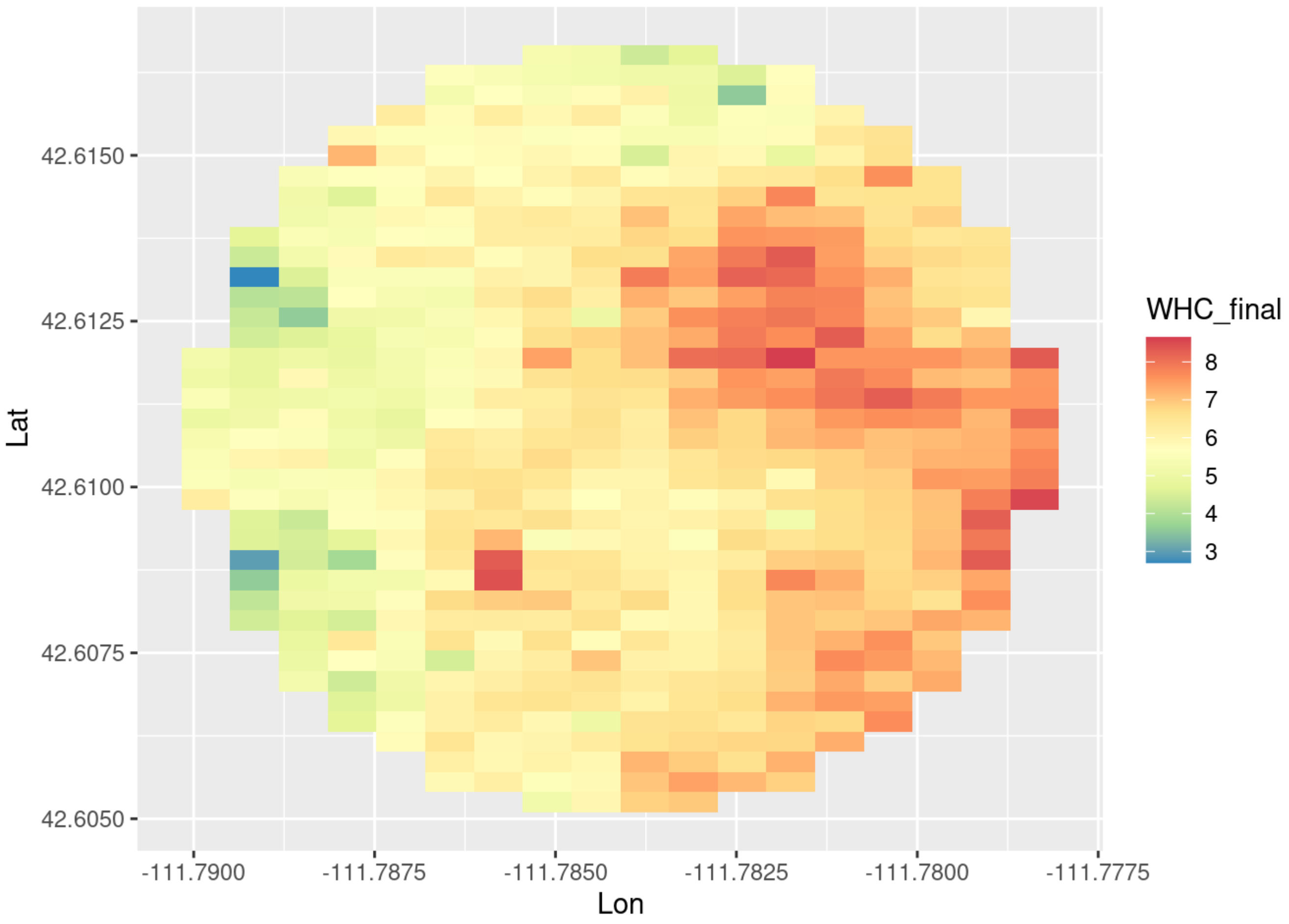


```
df$WHC_final <- df$WHC

df[rownames(df.pred),"WHC_final"] <- df.pred$Prediction

ggplot(data=df ,mapping=aes(x=Lon, y=Lat, fill=WHC_final)) + geom_raster() + scale_fill_distiller(palette="Spectral",na.value=NA)
```

```
## Warning: Raster pixels are placed at uneven horizontal intervals and will be shifted. Conside
r using geom_tile() instead.
## Raster pixels are placed at uneven vertical intervals and will be shifted. Consider using geo
m_tile() instead.
```


Lat
Lon
WHC_final