LandCover_Analysis

Anna Wolford

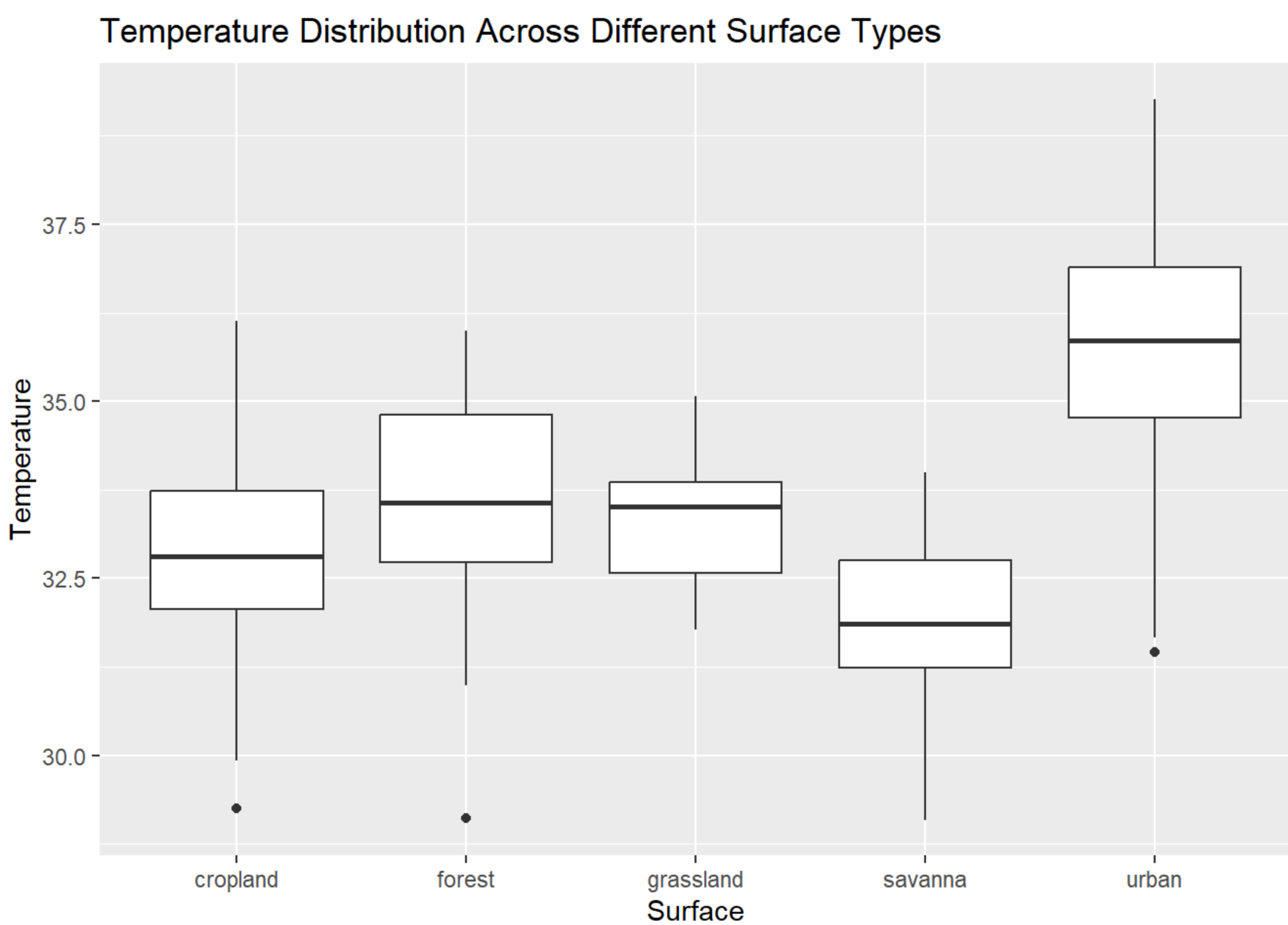
2024-03-27

Exploratory Data Analysis

1.

```
ggplot(temps, aes(x = Surface, y = Temp)) +
  geom_boxplot() +
  labs(x = "Surface", y = "Temperature") +
  ggtitle("Temperature Distribution Across Different Surface Types")
```

```
## Warning: Removed 126 rows containing non-finite values (`stat_boxplot()`).
```

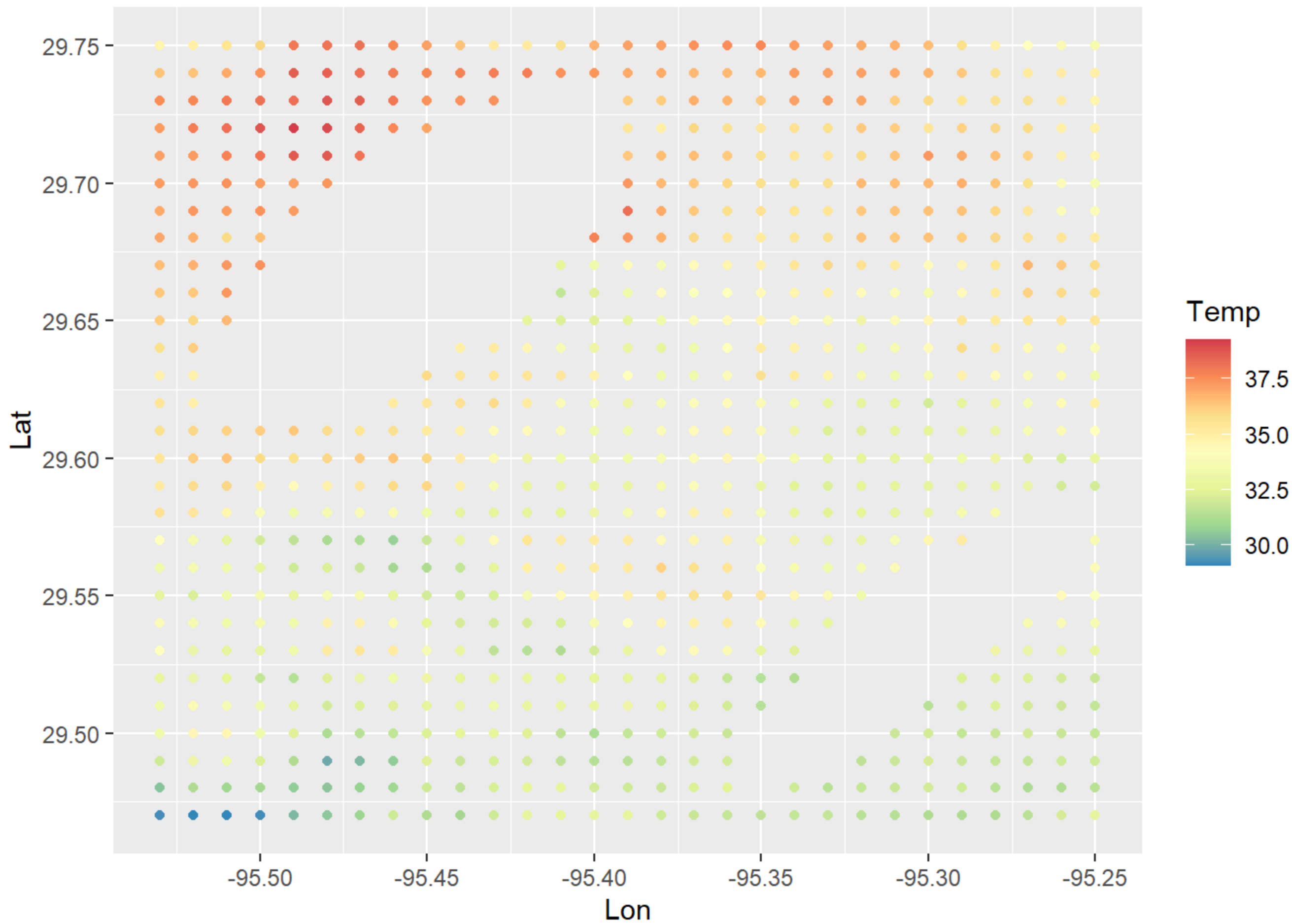


###

2.

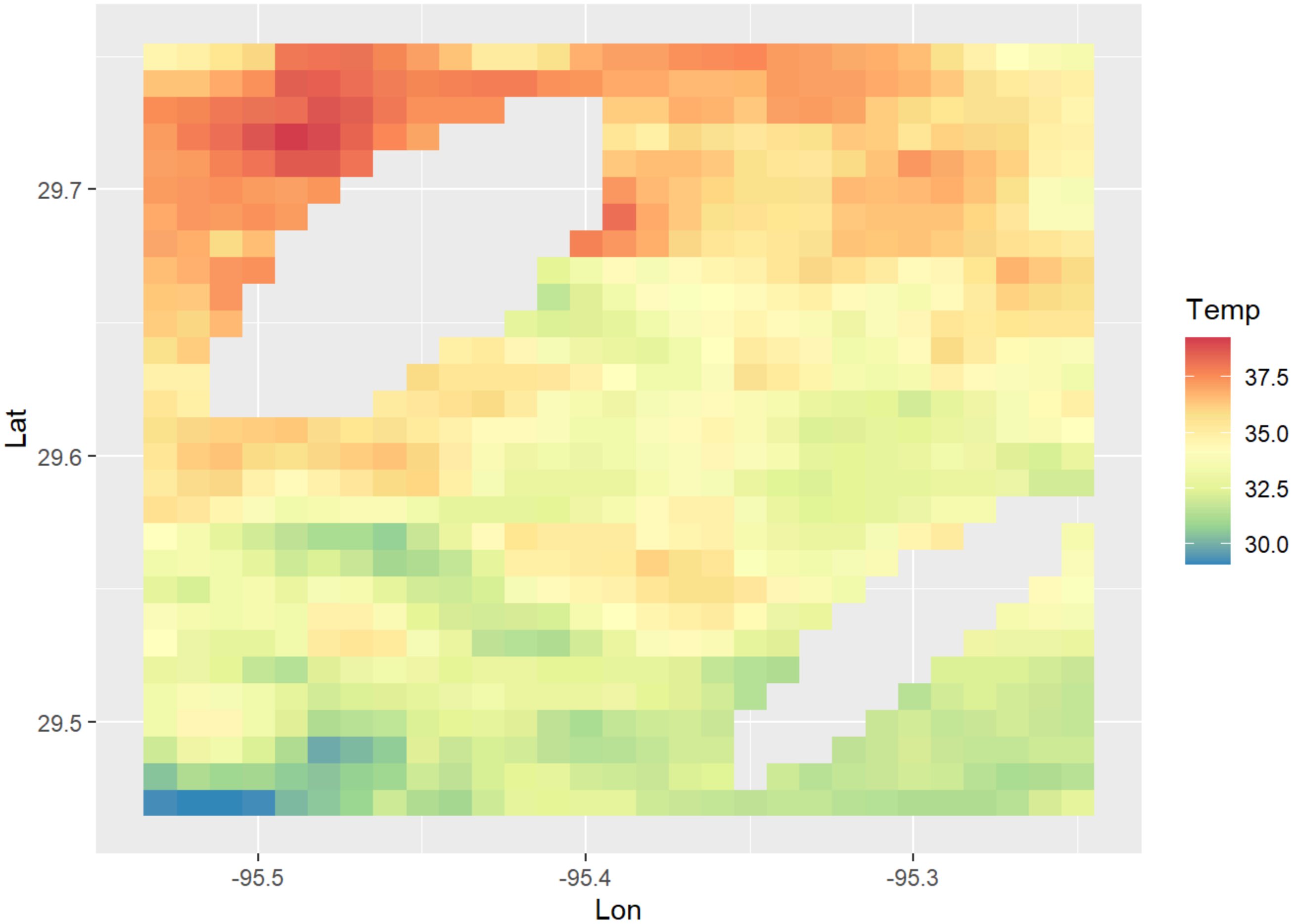
```
# ggplot(data=temps ,mapping=aes(x=Lon, y=Lat, fill=Temp)) + geom_tile()

# ggplot(data=temps ,mapping=aes(x=Lon, y=Lat, color=Temp)) + geom_point() #or
# ggplot(data=temps ,mapping=aes(x=Lon, y=Lat, fill=Temp)) + geom_raster()

ggplot(data= temps,mapping=aes(x=Lon, y=Lat, color=Temp)) + geom_point() + scale_color_distiller
(palette="Spectral",na.value=NA)
```

```
## Warning: Removed 126 rows containing missing values (`geom_point()`).
```




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

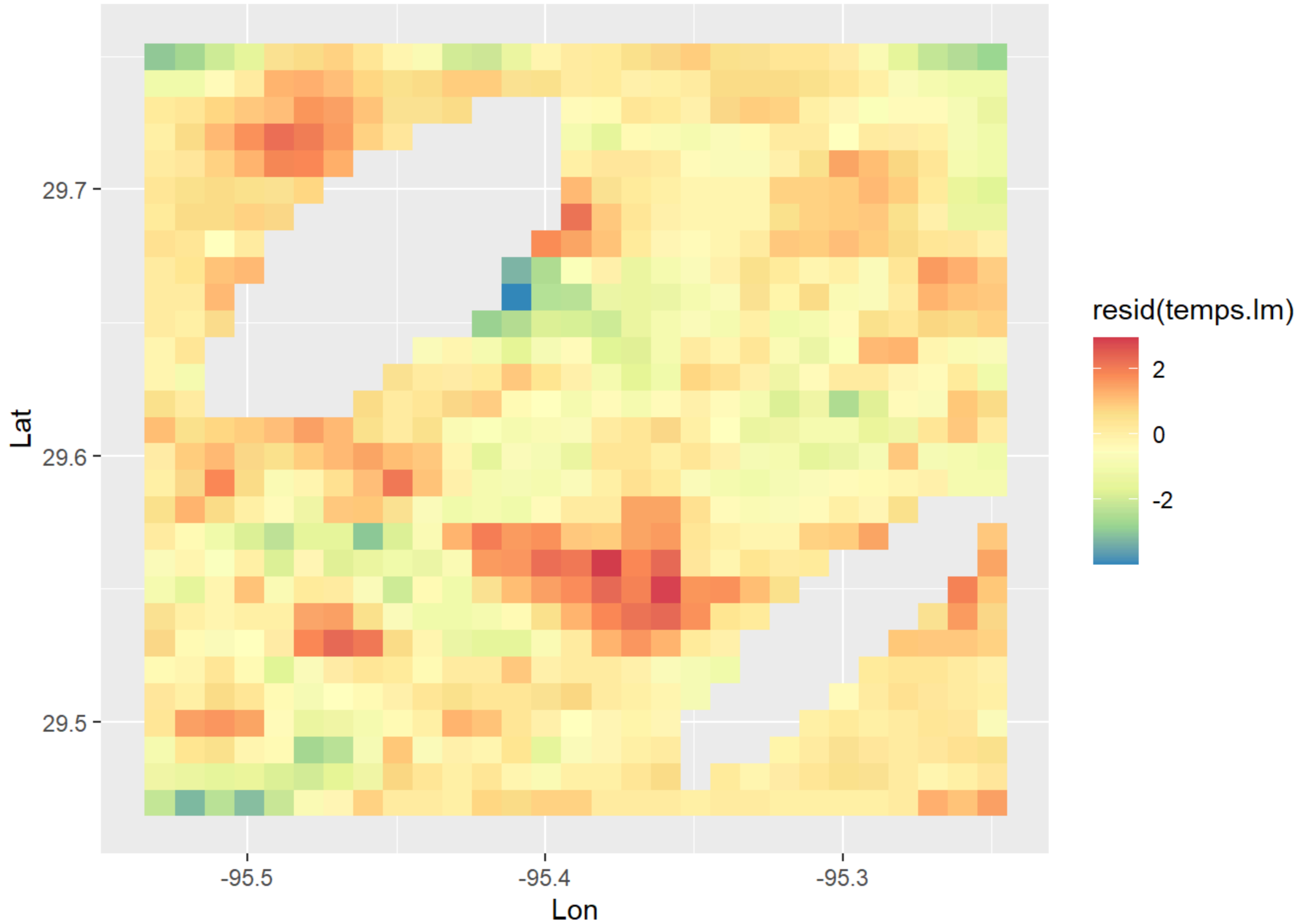
```
## Warning: Removed 126 rows containing missing values (`geom_raster()`).
```



###

3.

```
temps.lm <- lm(Temp ~ Surface + Lon + Lat, data=temps)
# summary(temps.lm)
# plot(fitted(temps.lm), resid(temps.lm),
#      xlab = "Fitted Values", ylab = "Residuals",
#      main = "Residuals vs. Fitted Values")
# abline(h = 0, col = "red")
temps_clean <- na.omit(temps)
ggplot(data= temps_clean,mapping=aes(x=Lon, y=Lat, fill=resid(temps.lm))) + geom_raster() + scal
e_fill_distiller(palette="Spectral",na.value=NA)
```

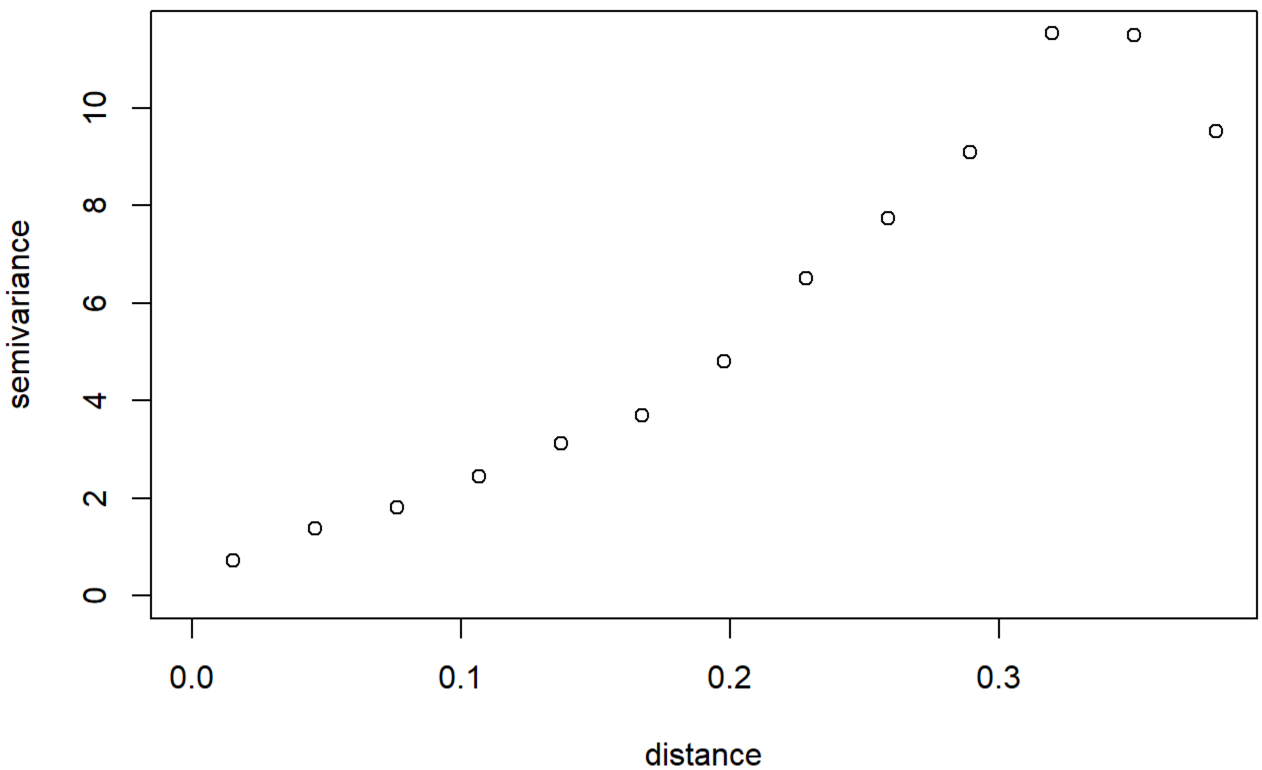


4.

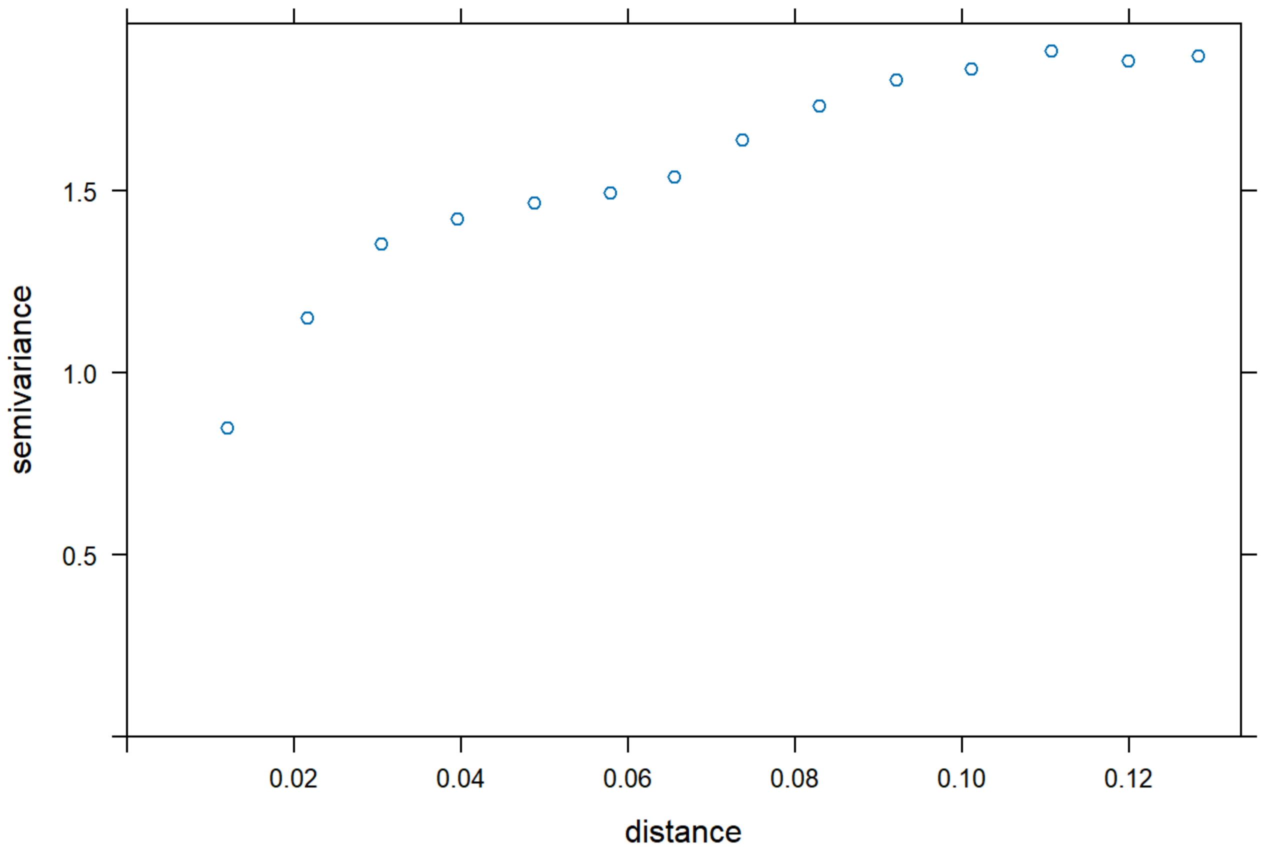
```
temps_clean <- na.omit(temps)
coords <- matrix(c(temps_clean$Lon, temps_clean$Lat), ncol=2, byrow=FALSE)
myVariogram <- variog(coords=coords, data=temps_clean$Temp)
```

```
## variog: computing omnidirectional variogram
```

```
plot(myVariogram)
```




```
myVariogram <- variogram(object=Temp~Surface, locations=~Lon+Lat, data=temps_clean)
plot(myVariogram)
```



Spatial MLR Model Fitting

1. Choose with AIC

```
model_exp <- gls(Temp ~ Surface, data=temps_clean,
                 correlation=corExp(form=~Lon+Lat, nugget=TRUE), method="ML")

model_spher <- gls(Temp ~ Surface, data=temps_clean,
                   correlation=corSpher(form=~Lon+Lat, nugget=TRUE), method="ML")

model_gaus <- gls(Temp ~ Surface, data=temps_clean,
                  correlation=corGaus(form=~Lon+Lat, nugget=TRUE), method="ML")

summary(model_exp)$AIC
```

```
## [1] 1191.253
```

```
summary(model_spher)$AIC
```

```
## [1] 1189.086
```

```
summary(model_gaus)$AIC ## Gaussian has the lowest AIC value
```

```
## [1] 1188.856
```

```
coefs <- coef(model_gaus)
cors <- coef(model_gaus$modelStruct$corStruct, unconstrained=FALSE)

coefs
```

```
##      (Intercept)    Surfaceforest Surfacegrassland   Surfacesavanna
##      34.06175985      -0.08052132      -0.14724355      -0.13270423
##     Surfaceurban
##       0.18835542
```

```
cors
```

```
##      range     nugget
## 0.02863235 0.04134444
```

```
(summary(model_gaus)$sigma)^2
```

```
## [1] 2.719901
```

Validating Spatial MLR Model Assumptions and Predictions

Linearity

```
library(car)
```

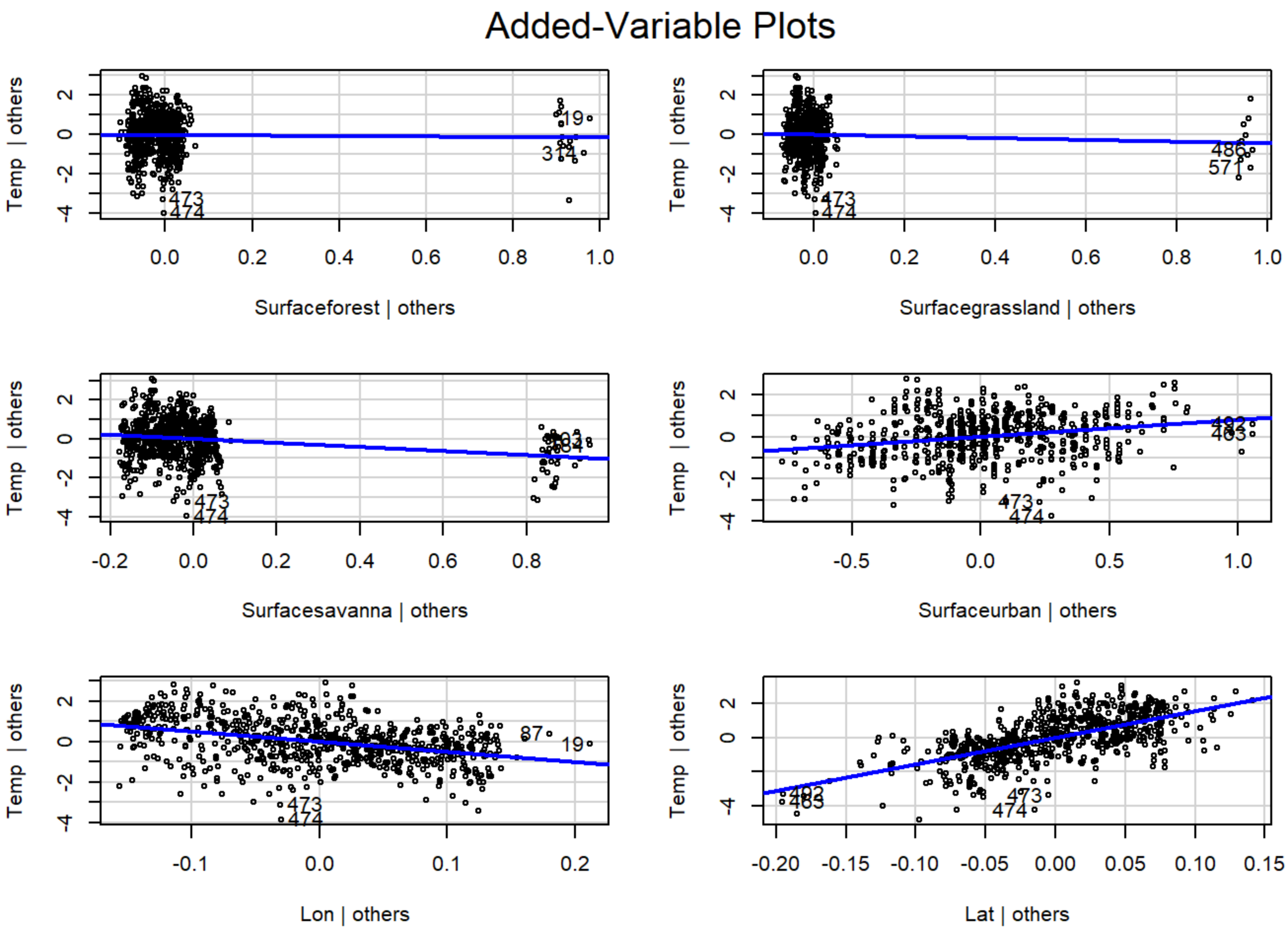
```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

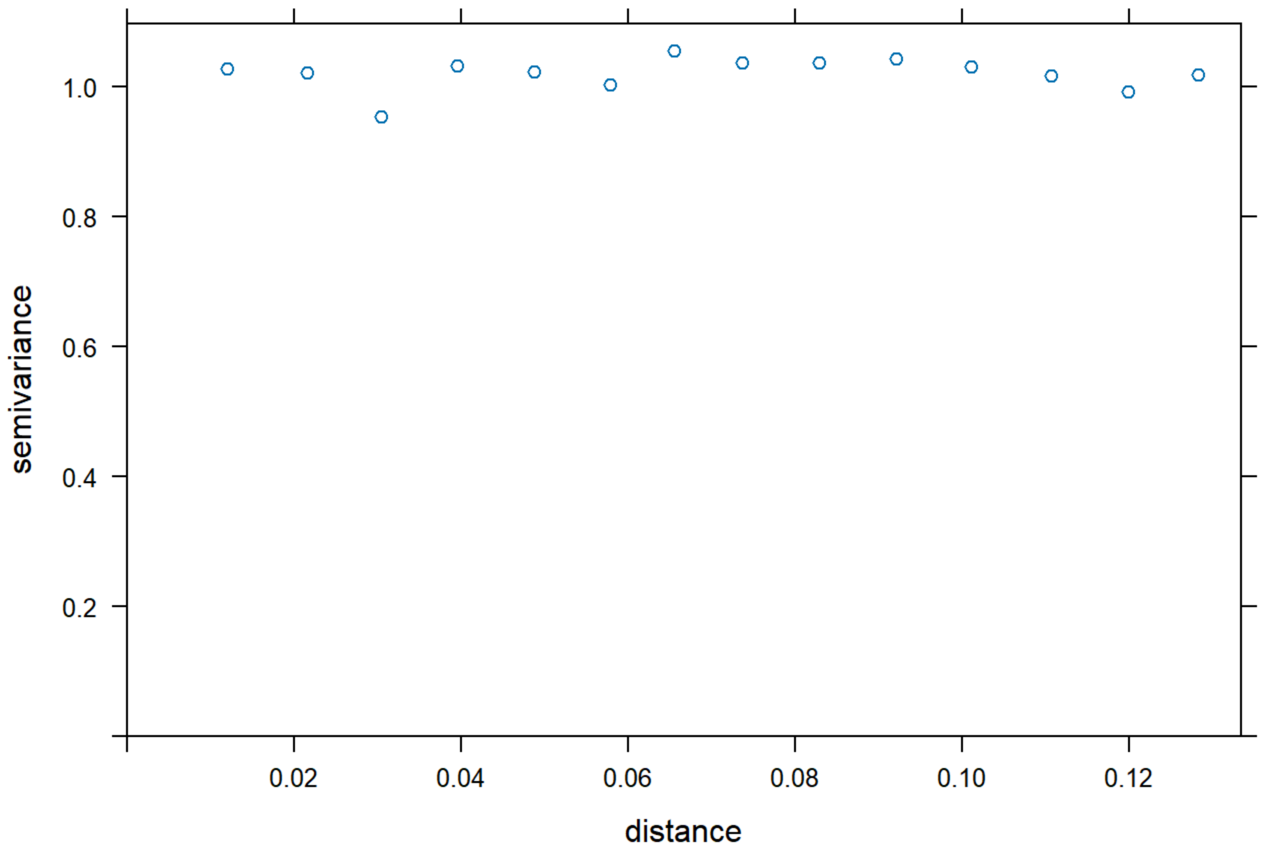
```
## The following object is masked from 'package:purrr':
##
##     some
```

```
car::avPlots(temps.lm)
```



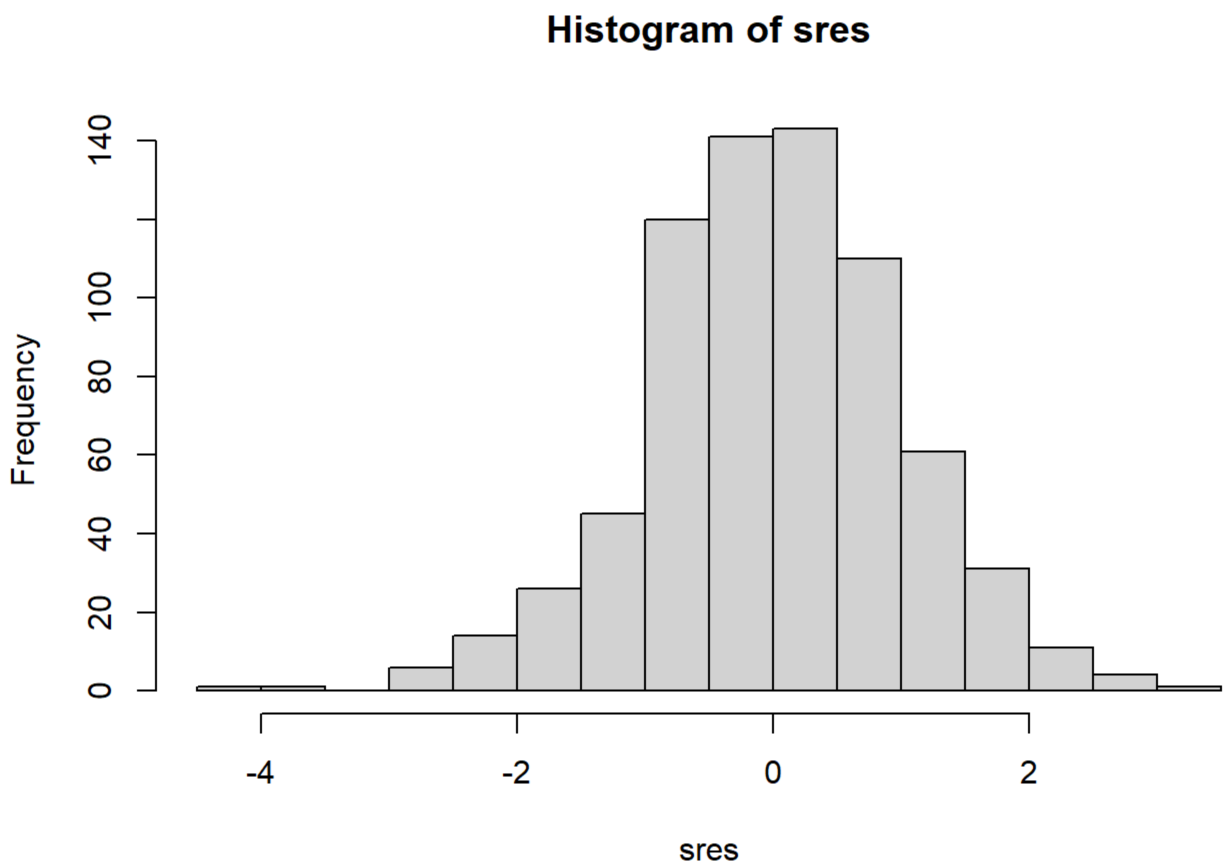
2.

```
sres <- stdres.gls(model_gaus)

# sresm <- matrix(sres, ncol=4, byrow=TRUE)
# round(cor(sresm),2)

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



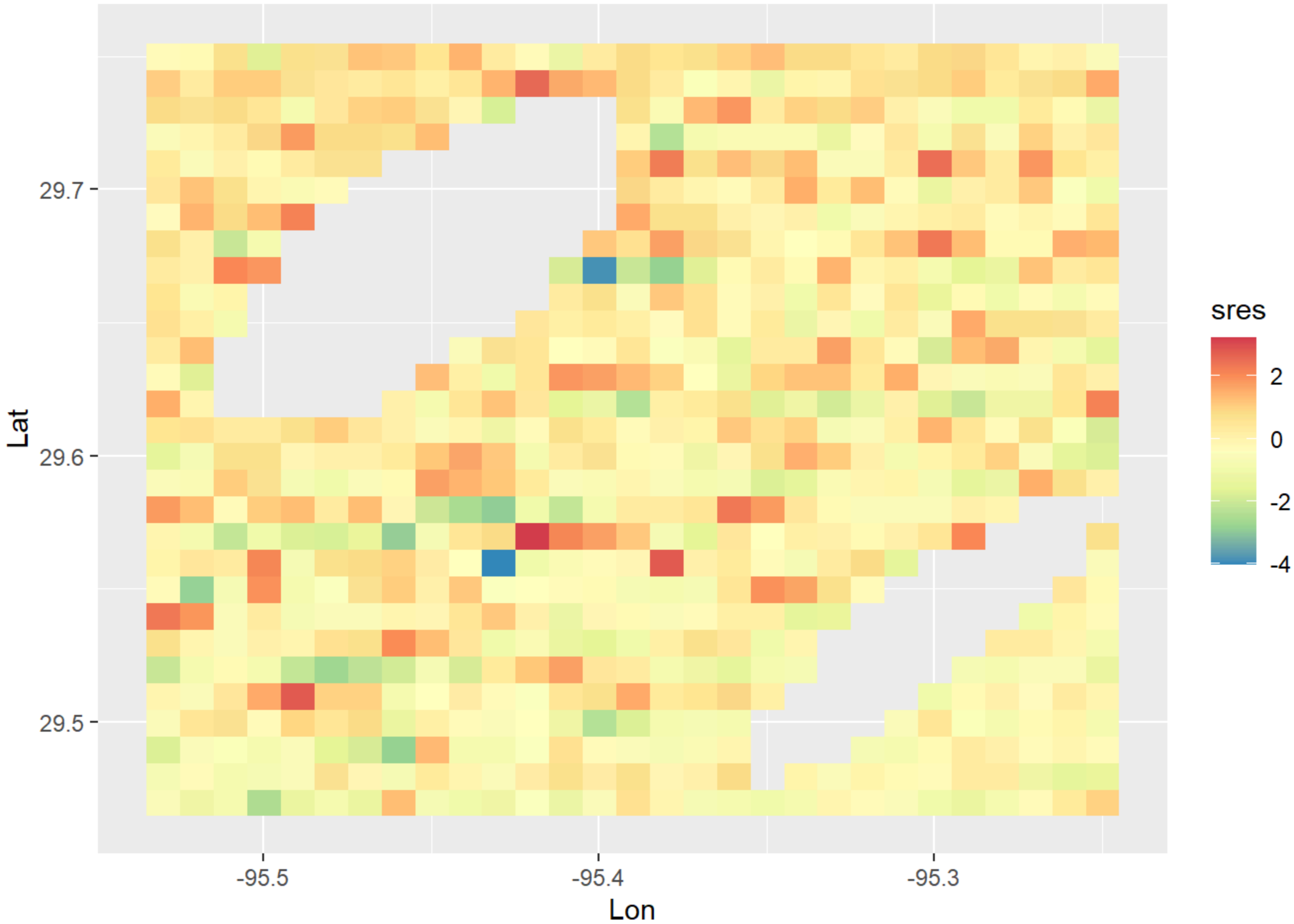
3.

```
hist(sres)
```



4.

```
ggplot(data= residDF,mapping=aes(x=Lon, y=Lat, fill=sres)) + geom_raster() + scale_fill_distille
r(palette="Spectral",na.value=NA)
```




```
# ggplot(data= temps_clean,mapping=aes(x=Lon, y=Lat, fill=resid(temps.lm))) + geom_raster() + sc
ale_fill_distiller(palette="Spectral",na.value=NA)

## Compare
```

5.

```
## CHECK THIS W DR H :)

system.time({gls(Temp ~ Surface, data=temps_clean,
                  correlation=corGaus(form=~Lon+Lat, nugget=TRUE), method="ML")})
```

```
##    user  system elapsed
##   14.84    0.23   27.21
```

```
n.cv <- 50 #Number of CV studies to run
n.test <- nrow(temps_clean)*.2 #Number of observations in a test set
rpmse <- rep(x=NA, times=n.cv)
cvg <- rep(x=NA, times=n.cv)
bias <- rep(x=NA, times=n.cv)
wid <- rep(x=NA, times=n.cv)

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

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

```
  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 <- temps_clean[test.obs,]
  train.set <- temps_clean[-test.obs,]

  ## Fit a gls() using the training data ???
  train.lm <- gls(Temp ~ Surface, data=train.set,
                  correlation=corGaus(form=~Lon+Lat, nugget=TRUE), method="ML")

  ## Generate predictions for the test set ???
  my.preds <- predictgls(train.lm, newdframe=test.set, level = .95)

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

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

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

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

```
mean(rpmse)
```

```
## [1] 0.4308505
```

```
  mean(cvg)
```

```
## [1] 0.9685315
```

```
  mean(bias)
```

```
## [1] 0.001805533
```

```
  mean(wid)
```

```
## [1] 1.947548
```

```
close(pb)
```

```
# n.cv <- 50 #Number of CV studies to run
# n.test <- nrow(temps_clean)*.2 #Number of observations in a test set
rpmse.lm <- rep(x=NA, times=n.cv)
cvg.lm  <- rep(x=NA, times=n.cv)
bias.lm  <- rep(x=NA, times=n.cv)
wid.lm <- rep(x=NA, times=n.cv)

# n = nrow(temps_clean)

  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 <- temps_clean[test.obs,]
  train.set <- temps_clean[-test.obs,]

  ## Fit a gls() using the training data ???
  train.lm <- lm(formula=Temp ~ Surface, data=train.set)

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

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

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

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

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

  mean(rpmse.lm)
```

```
## [1] 1.378659
```

```
  mean(wid.lm)
```

```
## [1] 5.403683
```

```
  mean(cvg.lm)
```

```
## [1] 0.9483916
```

```
mean(bias.lm)
```

```
## [1] 0.007357378
```

```
mean(rpmse)
```

```
## [1] 0.4308505
```

```
mean(wid)
```

```
## [1] 1.947548
```

```
mean(cvg)
```

```
## [1] 0.9685315
```

```
mean(bias)
```

```
## [1] 0.001805533
```

Statistical Inference

Use an F-test to see if temperatures are difference across any of the land-cover types.

```
anova(temps.lm)
```

```
## Analysis of Variance Table
##
## Response: Temp
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## Surface     4 1511.30  377.83 366.749 < 2.2e-16 ***
## Lon         1  102.35  102.35  99.349 < 2.2e-16 ***
## Lat         1  501.03  501.03 486.346 < 2.2e-16 ***
## Residuals 708  729.38    1.03
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Create confidence intervals for each effect of land cover and determine which land cover types result in increased temperatures.

```
confint(model_gaus)
```

```
##                        2.5 %      97.5 %
## (Intercept)      33.56477163 34.55874806
## Surfaceforest    -0.31928653  0.15824390
## Surfacegrassland -0.38883755  0.09435045
## Surfacesavanna   -0.28968774  0.02427929
## Surfaceurban      0.06982921  0.30688163
```

Only the urban surface results in increased temperatures (with 95% confidence).

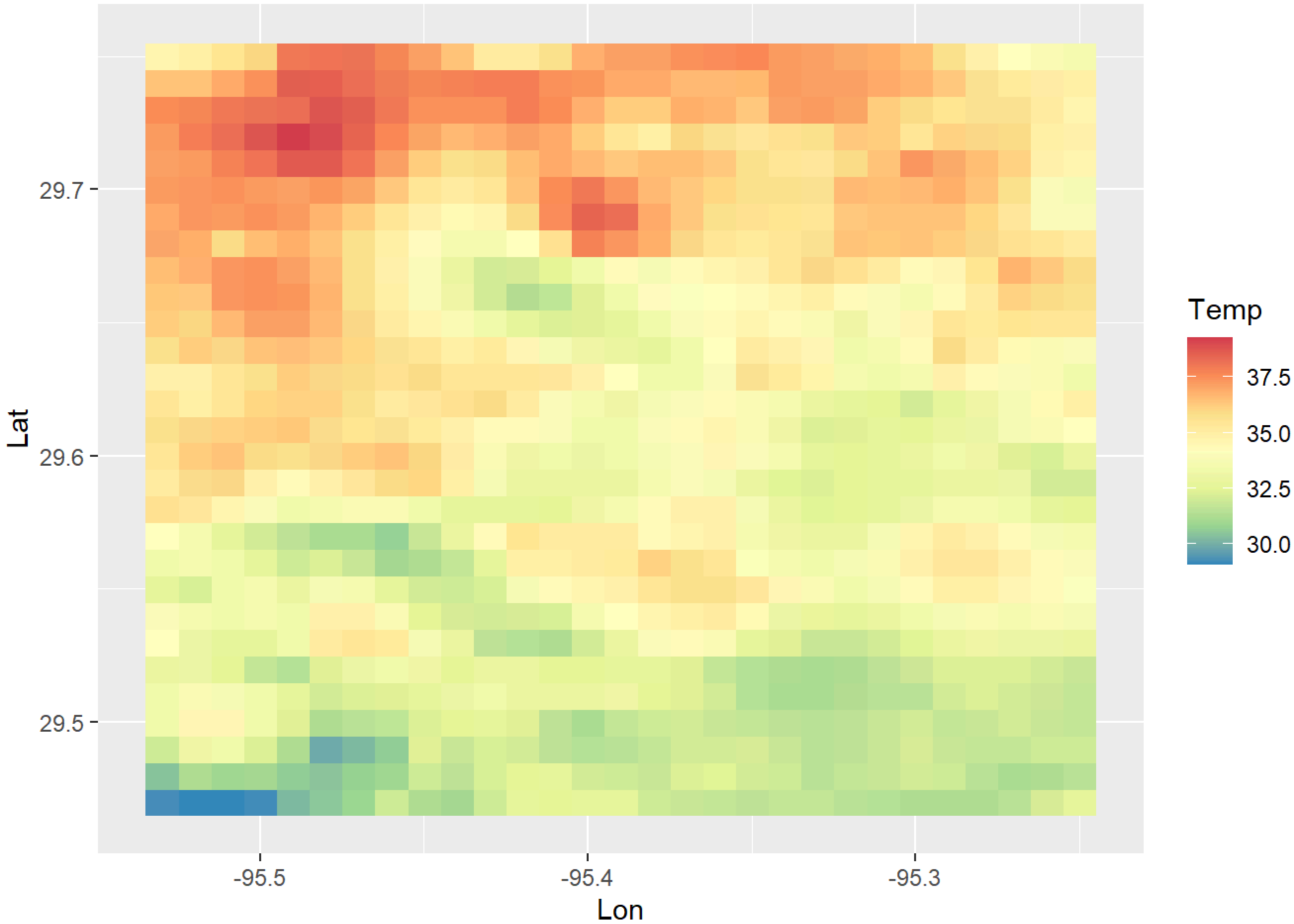
Perform a GLHT to construct a confidence interval of the difference temperature between Savannah and Urban land covers.

```
a <- c(1, 0,0,1,0)
b <- c(1,0,0,0,1)
summary_glht <- multcomp::glht(model_gaus, linfct = t(a-b), alternative="two.sided")
confint(summary_glht)
```

```
##
##   Simultaneous Confidence Intervals
##
## Fit: gls(model = Temp ~ Surface, data = temps_clean, correlation = corGaus(form = ~Lon +
##     Lat, nugget = TRUE), method = "ML")
##
## Quantile = 1.96
## 95% family-wise confidence level
##
##
## Linear Hypotheses:
##        Estimate lwr     upr
## 1 == 0 -0.3211  -0.5038 -0.1383
```

Create and map predictions of the temperature at each location that was impeded by cloud cover.

```
temp_nas <- setdiff(temps, temps_clean)
preds.na <- predictgls(model_gaus, newdframe=(temps %>% filter(is.na(Temp))), level=0.95)
preds.na <- preds.na %>%
  mutate(Temp = Prediction)
full_temp <- rbind(preds.na[,1:4], temps_clean)

ggplot(data= full_temp,mapping=aes(x=Lon, y=Lat, fill=Temp)) + geom_raster() + scale_fill_distil
ler(palette="Spectral",na.value=NA)
```


Lat
29.7
29.6
29.5
-95.5
-95.4
-95.3
Lon
Temp
37.5
35.0
32.5
30.0