Homework 2

Carson Keeter, Graduate Student: Statistics

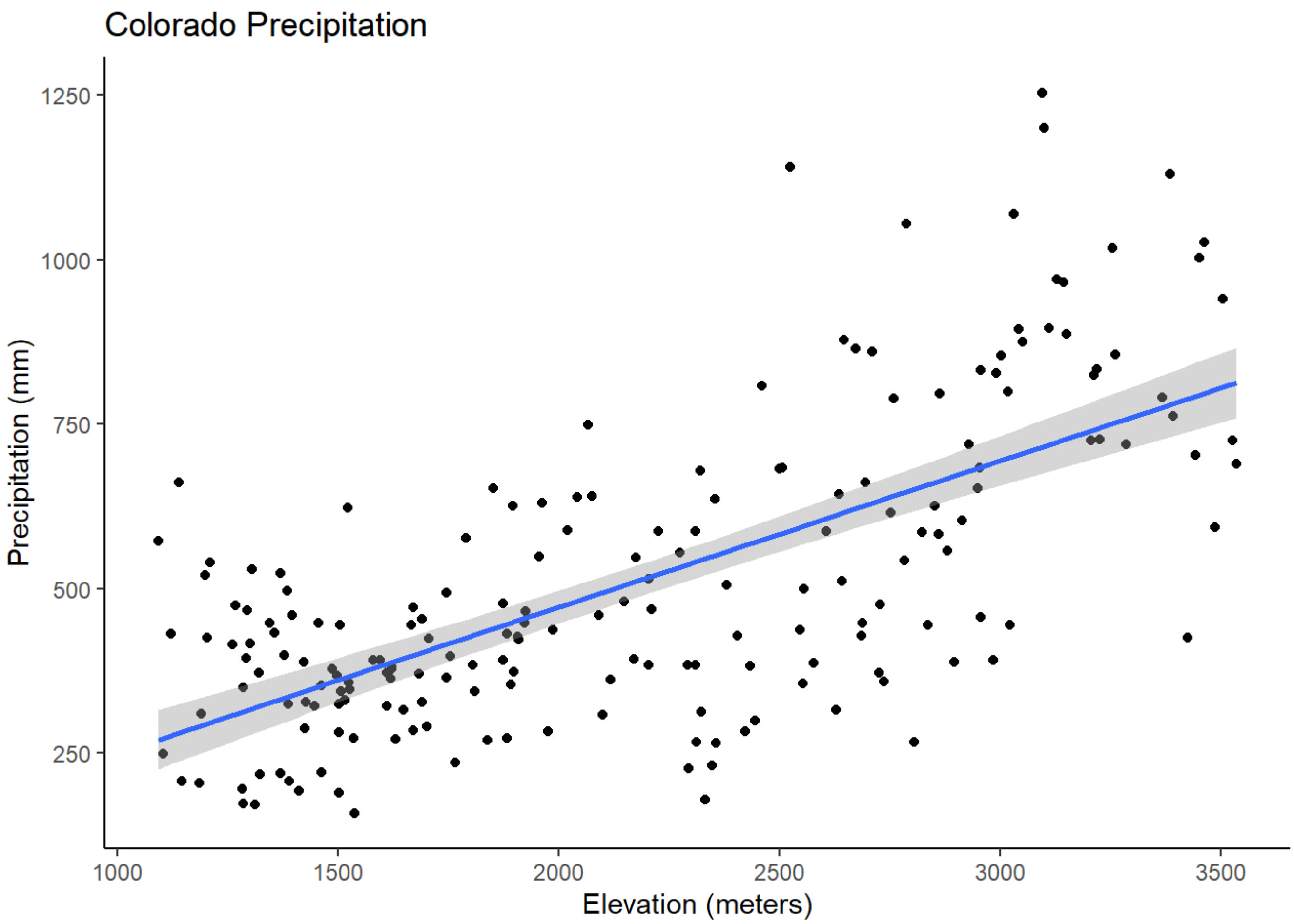
11/12/2019

Question 1: Colorado Precipitation

1a) Provide an R function that computes the log-likelihood of a spatial model.

The following scatterplot is used to investigate if there is a relationship between *precipitation* and *elevation*:

```
ggplot(
  data = colo,
  aes(
    x = Elevation,
    y = Precip
  )
) +
  geom_point(

  ) +
  geom_smooth(
    method = "lm"
  ) +
  theme_classic(

  ) +
  labs(
    title = "Colorado Precipitation",
    x = "Elevation (meters)",
    y = "Precipitation (mm)"
  )
```



Disregarding any spatial effects, it seems that elevation and precipitation are positively correlated. Now, recall that the Colorado precip/temp data contained fairly clear evidence that there is spatial correlation with the predictor variables (presumably in the Rocky Mountain Range). Thankfully, we are able to determine the likelihood of a spatial model.

To "hand" calculate the log-likelihood of a spatial model, a few matrices must be constructed:

$$Y(s) = X'(s)\beta + \omega(s) + \epsilon(s)$$

Where $w(s)$ contains variance-covariance matrix (NxN),

$$w(s) \sim MVN(0, \Sigma(\theta))$$

and

$$\Sigma(\theta) = \alpha^2[spatial\,correlation\,matrix]$$

and

$$\epsilon(s) \sim MVN(0, \tau^2[nonspatial\,correlation\,matrix])$$

Where,

- $Y(s)$: Matrix of the outcome variable (logPrecip)
- $X(s)$: Scaled matrix of the predictor variables (Zelevation) centered at -0.02985014.
- H^*: Distance matrix of the coordinates from the Colorado precipitation dataset.

- θ: Contains parameters of the spatial correlation function (usually effective range)
- More on this later

```
H <- as.matrix(
  dist(
    st_coordinates(
      colo_sf
    )
  )
)

X <- model.matrix(
  ~ scale(Zelevation),
  data = colo_sf
)

Y <- as.matrix(
  colo_sf$logPrecip
)
```

Once these matrices are constructed, the function to determine the log-likelihood of a *Powered Exponential spatial correlation function* is constructed. The *Powered Exponential spatial correlation function* is the following:

$$Cor(Y(s_i), Y(s_j)) = Cor(h_{ij} = ||s_i - s_j||) = exp(-(\phi h_{ij})^p))$$

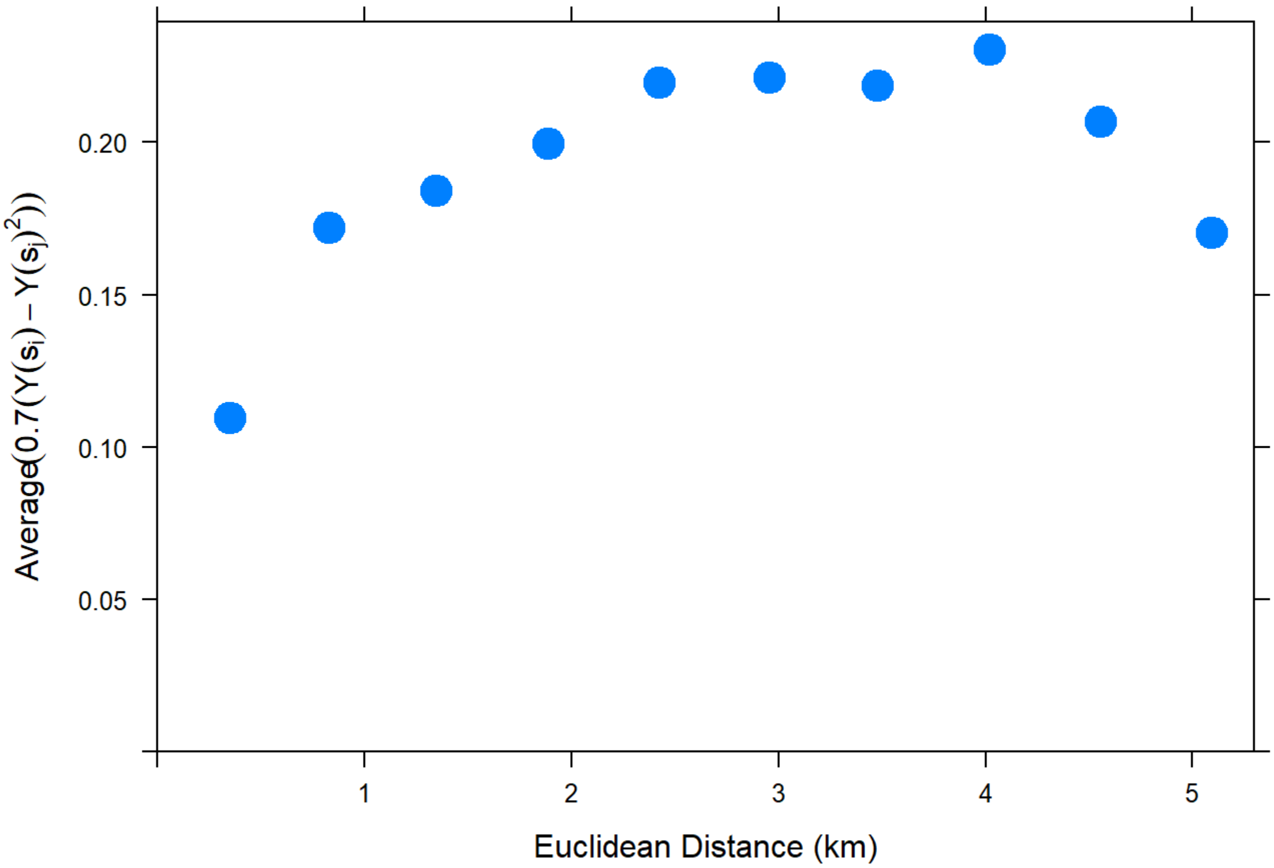
The following code replicates the *Powered Exponential spatial correlation function*:

```
MVN_fun <- function(theta, H, X, Y){

  part_sill <- theta[1]
  phi <- theta[2]
  p <- theta[3]
  nugget <- theta[4]

  Sigma <- part_sill*exp(-(phi*H)^p) + nugget*diag(length(Y))

  Sigma_inverse <- solve(Sigma)

  Beta <- solve(t(X) %*% Sigma_inverse %*% X) %*% t(X) %*% Sigma_inverse %*% Y

  log_L <- mvtnorm::dmvnorm(
    x = t(Y),
    mean = X %*% Beta,
    sigma = Sigma,
    log = TRUE
    )

  return(-1*log_L)

}
```

1b) Use *optim()* to maximize the log-likelihood using the function you wrote in part a) and provide: i) the maximum log-likelihood value; and ii) the maximum likelihood estimates of the 4 parameters.

Using the function from above and *optim()*, we can now estimate the log-likelihood of the *Powered Exponential spatial correlation function*. The starting values (partial sill and the nugget) are all estimated from the following variogram:

```
variogram <- variogram(
  logPrecip ~ 1,
  locations = ~Longitude + Latitude,
  data = colo,
  cutoff = cutoff,
  width = cutoff/bins
)

plot(
  variogram,
  cex = 2,
  pch = 19,
  ylab = expression(
    paste(
      "Average", (0.7*(Y(s[i]) - Y(s[j])^2))
    )
  ),
  xlab = "Euclidean Distance (km)"
)
```



Here, it can be estimated that the nugget and partial sill are approx. 0.10. Then, the starting values for ϕ and p are reasonably estimated.

The following code uses the function created above and estimates the log-likelihood of a *Powered Exponential correlation function*:

```
MLE <- optim(
  c(0.1, 1, 0.5, 0.1),
  fn = MVN_fun,
  H = H,
  X = X,
  Y = Y,
  hessian = TRUE
  )
```

```
MLE
```

```
## $par
## [1] 0.22183220 0.62437503 0.91519970 0.00551723
##
## $value
## [1] 1.066688
##
## $counts
## function gradient
##      463       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
##
## $hessian
##           [,1]      [,2]       [,3]       [,4]
## [1,] 1486.4722  420.4558  -504.3666   8183.087
## [2,]  420.4558  127.9996  -159.4623   2552.197
## [3,] -504.3666 -159.4623   261.3692  -4276.478
## [4,] 8183.0866 2552.1965 -4276.4775 158431.474
```

Here, the following parameters are estimated from this function:

- Log-likelihood: 1.067
- Partial Sill: 0.2218
- p: 0.9152
- ϕ: 0.6244
- Nugget: 0.0055

1c) Estimate an identical model using geoR. Please provide all relevant code and output.

Now, the same estimates are made with a real R package. Using *geoR*, we are able to create and estimate the log-likelihood of a *Powered Exponential correlation function* with the following:

```r
d <- data.frame(
  st_coordinates(
    colo_sf
  ),
  colo[, -1:-2]
)

d_geo <- as.geodata(
  d,
  coords.col = 1:2,
  data.col = 4,
  covar.col = 6
)

fit_geoR <- geoR::likfit(
  d_geo,
  ini.cov.pars = c(0.2, 1),
  nugget = .1,
  fix.nugget = FALSE,
  trend = ~ scale(Zelevation),
  cov.model = "powered.exponential",
  kappa = 1,
  fix.kappa = FALSE,
  lik.method = "ML"
)
```

```r
summary(fit_geoR)
```

```
## Summary of the parameter estimation
## ----------------------------------
## Estimation method: maximum likelihood
##
## Parameters of the mean component (trend):
##  beta0  beta1
## 6.3960 0.5352
##
## Parameters of the spatial component:
##    correlation function: powered.exponential
##       (estimated) variance parameter sigmasq (partial sill) =  0.2219
##       (estimated) cor. fct. parameter phi (range parameter)  =  1.602
##       (estimated) extra parameter kappa = 0.9153
##    anisotropy parameters:
##       (fixed) anisotropy angle = 0  ( 0 degrees )
##       (fixed) anisotropy ratio = 1
##
## Parameter of the error component:
##       (estimated) nugget =  0.0055
##
## Transformation parameter:
##       (fixed) Box-Cox parameter = 1 (no transformation)
##
## Practical Range with cor=0.05 for asymptotic range: 5.311941
##
## Maximised Likelihood:
##    log.L n.params      AIC      BIC
## "-1.067"      "6"  "14.13"  "33.77"
##
## non spatial model:
##    log.L n.params      AIC      BIC
## "-65.32"      "3"  "136.6"  "146.5"
##
## Call:
## geoR::likfit(geodata = d_geo, trend = ~scale(Zelevation), ini.cov.pars = c(0.09,
##     1), fix.nugget = FALSE, nugget = 0.1, fix.kappa = FALSE,
##     kappa = 1, cov.model = "powered.exponential", lik.method = "ML")
```

Here, we see the following estimates:

- Log-likelihood: -1.067
- Partial Sill: 0.2219
- p: 0.9153
- ϕ: 1/1.602 = 0.6242
- Nugget: 0.0055

It is important to note that the *anisotropy* parameters are all consistent with an *isotropic* model (confirming that I did not mix up the code).

1d) Demonstrate that the estimates you obtained in part b) match the estimates out of geoR for the 4 parameters. Note that these must match to at least the second decimal point. Hint: recall that geoR parameterizes its spatial correlation functions in terms of $1/\phi$ and not in terms of ϕ.

The following table compares the values from the hand calculated log-likelihood model and the geoR model:

```
q1_model_comp <- data.frame(
  model = c("By Hand", "geoR"),
  log.likelihood = c(MLE$value, fit_geoR$loglik),
  nugget = c(MLE$par[4], fit_geoR$nugget),
  partial.sill = c(MLE$par[1], fit_geoR$sigmasq),
  range.parm = c(MLE$par[2], 1/fit_geoR$phi),          # Notice 1/phi for phi in geoR
  power = c(MLE$par[3], fit_geoR$kappa)
  )

q1_model_comp
```

```
##     model log.likelihood      nugget partial.sill range.parm     power
## 1 By Hand       1.066688 0.005517230    0.2218322  0.6243750 0.9151997
## 2    geoR      -1.066688 0.005520452    0.2219004  0.6242309 0.9153002
```

As per the requirements for this question, the estimates are identical to (at least) the second decimal.

1e) Derive the mathematical expression for the effective range of the powered exponential.

First, λ is assumed to be a arbitrarily small correlation (usually around 0.05). For now, it will be kept in general terms.

Then, the correlation is state in terms of ϕ (range parameter), *h* (effective range), *p* (exponent), and λ (spatial correlation).

$$\lambda = exp(-\phi h^*)^p$$

To find *h*

Take the *log* of both sides,

$$log(\lambda) = -(\phi h^*)^p$$

Divide by -1,

$$-log(\lambda) = (\phi h^*)^p$$

Divide by ϕ,

$$\frac{-log(\lambda)}{\phi} = (h^*)^p$$

Raise each side to the power of *1/p*,

$$\left(\frac{-log(\lambda)}{\phi}\right)^{\frac{1}{p}} = h^*$$

##—

Question 2: Fukushima Daiichi Nuclear Power Plant Disaster

2a) Using a non-spatial model, decide how to specify the mean function. Be sure to explore linear and/or quadratic distance variables and possible interactions with rain. You don't have to provide every model you considered, but label/describe your final answer and describe how you arrived there.

After much model selection, I believe that *log(Dose)* as a function of *Distance (km)* and *Distance^2 (km)* is the most appropriate in this scenario. The residuals for this model meet the requirements of linear regression fairly well and the overall model is fairly interpretable. It also turns out that *rain* is insignificant when considered with the aforementioned variables. The OLS regression equation, $log(Dose) = 0.8296 - 0.1348 distance + 0.0032 distance^2 + \epsilon(s)$, is constructed from the following:

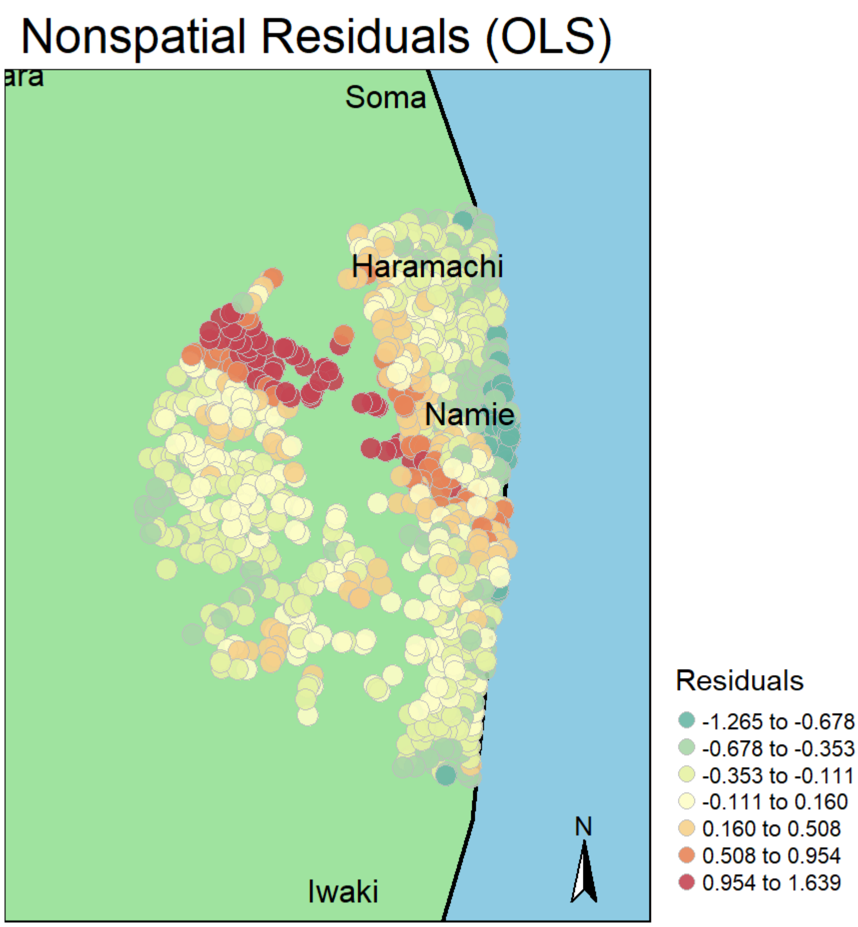
```
model <- lm(
  log_Dose ~ Dist_km + Dist_km_sq,
  data = rad_df
)

summary.lm(model)
```

```
##
## Call:
## lm(formula = log_Dose ~ Dist_km + Dist_km_sq, data = rad_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26490 -0.30182 -0.09353  0.19383  1.63930
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.8296154  0.0833376   9.955   <2e-16 ***
## Dist_km     -0.1347750  0.0106554 -12.649   <2e-16 ***
## Dist_km_sq   0.0031717  0.0003046  10.412   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4884 on 808 degrees of freedom
## Multiple R-squared:  0.2353, Adjusted R-squared:  0.2335
## F-statistic: 124.3 on 2 and 808 DF,  p-value: < 2.2e-16
```

2b/c) Make a choropleth map of the residuals from the non-spatial model you decided on in part a). Please provide the coast of Japan as a reference. Make sensible choices to make your map as readable as possible. Comment on whether you observe spatial correlation in the residuals. Investigate the residuals for evidence of anisotropy. Provide the apparent anisotropy angle.

```
max_lon <- max(rad_df$LONG)
min_lon <- min(rad_df$LONG)

max_lat <- max(rad_df$LAT)
min_lat <- min(rad_df$LAT)

correction <- .12

tm_shape(
  japan_fill,
  proj4string = CRS(
    "+proj=utm +zone=54"
  ),
  ylim = c(
    min_lat - correction,
    max_lat + correction
  ),
  xlim = c(
    min_lon - correction,
    max_lon + correction
    )
  ) +
  tm_fill(
    col = "lightgreen"
  ) +
  tm_borders(
    lwd = 2,
    col = "black"
  ) +
  tm_shape(
    rad_df_points
  ) +
  tm_symbols(
    col = "resid",
    palette = "-Spectral",
    n = 7,
    style = "jenks",
    border.lwd = 0.1,
    border.col = 'gray',
    alpha = 0.9,
    scale = .75,
    title.col = "Residuals"
  ) +
  tm_shape(
    japan_cities
  ) +
  tm_text(
    "name",
    textNA = "",
    remove.overlap = T,
    shadow = T
  ) +
  tm_legend(
```

```
    position = c(
      "left",
      "bottom"
    ),
    legend.outside = TRUE,
    frame = F,
    main.title = 'Nonspatial Residuals (OLS)'
  ) +
  tm_layout(
    bg.color = "skyblue",
    saturation = .85
  ) +
  tm_compass(
    north = 0,
    type = "arrow"
  )
```



Based on this map, there seems to be some *surprisingly* evident spatial correlation along the NW direction. Since the residuals are positive along this direction, the fitted values are underestimated. Furthermore, it is apparent that there is spatial correlation around the 125° diagonal (where 0° is completely horizontal and 90° is completely vertical moving counter-clockwise).

2d) Using sensible notation, write out all parts of a likelihood-based anisotropic spatial model. Be sure to specify the following: i) likelihood of observed data; ii) specification of the mean function; iii) specification of the variance-covariance matrix. Your notation can be specific to this exact dataset, or general for any spatial dataset.

The function for the likelihood of an anisotropic model is the following:

$$Y(s) \sim MVN(X(s)\beta,\ \alpha^2\rho(H^*,\theta)_{ij} + \tau^2 I)$$

Where,

- $Y(s)$: Outcome variable, log(dose), distributed in a *multi-variate normal* distribution (MVN)
- $X(s)$: Design matrix of predictor variables, distance (km) and distance^2 (km), scaled by matrix coefficients, β
- $\alpha\ ^2$: Spatial variance, partial sill
- $\rho(H^*,\theta)_{ij}$: $\rho(h^*_{ij} = ||G(s_i - s_j)||, \theta)$ estimated by $\frac{n(n-1)}{2}$, with

$$G = \begin{pmatrix} 1 & 0 \\ 0 & \lambda \end{pmatrix} \begin{pmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{pmatrix}$$

- τ^2: Non-spatial variance, nugget
- I: Identity matrix where,

$$I = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2e) Estimate: i) isotropic *and* ii) anisotropic likelihood-based models for these data. Make sensible decisions regardings he spatial correlation function. Compare these two models to each other and also to the non-spatial model using information criteria. Which of the three models is best for predicting radiation dose-rates?

So far, only a non-spatial model has been constructed (OLS). Now, two models are constructed: one that considers both distance *and* orientation/direction (anisotropic) and one that only consideres distance (isotropic).

First, the model that only considers distance is created (isotropic):

```
rad_geostat_iso <- lgm(
  log_Dose ~ Dist_km + Dist_km_sq,
  data = rad_sp,
  grid = 500,
  shape = 0.5,
  fixShape = TRUE,
  fixNugget = FALSE,
  aniso = FALSE,
  reml = TRUE
)
```

```
summary(rad_geostat_iso)
```

```
##                        estimate      stdErr      ci0.005      ci0.995
## (Intercept)         0.777909762 1.016960309 -1.841606403  3.397425927
## Dist_km            -0.136524889 0.060663715 -0.292784263  0.019734485
## Dist_km_sq          0.002774876 0.001501043 -0.001091555  0.006641307
## sdNugget            0.144174821          NA  0.125294816  0.165899753
## sdSpatial           0.771695708          NA  0.174594759  3.410837003
## range               0.616262339          NA  0.029721116 12.778096030
## shape               0.500000000          NA           NA           NA
## anisoRatio          1.000000000          NA           NA           NA
## anisoAngleRadians   0.000000000          NA           NA           NA
## anisoAngleDegrees   0.000000000          NA           NA           NA
## boxcox              1.000000000          NA           NA           NA
##                         ci0.025      ci0.975        ci0.05       ci0.95
## (Intercept)        -1.2152958181  2.771115342 -0.8948410914  2.450660615
## Dist_km            -0.2554235847 -0.017626193 -0.2363078199 -0.036741958
## Dist_km_sq         -0.0001671145  0.005716866  0.0003058797  0.005243872
## sdNugget            0.1295708753  0.160424779  0.1318148670  0.157693738
## sdSpatial           0.2490841115  2.390815943  0.2987458789  1.993380686
## range               0.0613592858  6.189434338  0.0889112984  4.271439935
## shape                         NA           NA            NA           NA
## anisoRatio                    NA           NA            NA           NA
## anisoAngleRadians             NA           NA            NA           NA
## anisoAngleDegrees             NA           NA            NA           NA
## boxcox                        NA           NA            NA           NA
##                           ci0.1       ci0.9       pval Estimated
## (Intercept)        -0.5253773146  2.08119684 0.44430953      TRUE
## Dist_km            -0.2142685673 -0.05878121 0.02441597      TRUE
## Dist_km_sq          0.0008512118  0.00469854 0.06451101      TRUE
## sdNugget            0.1344503208  0.15460267         NA      TRUE
## sdSpatial           0.3684113624  1.61643838         NA      TRUE
## range               0.1363540436  2.78524392         NA      TRUE
## shape                         NA          NA         NA     FALSE
## anisoRatio                    NA          NA         NA     FALSE
## anisoAngleRadians             NA          NA         NA     FALSE
## anisoAngleDegrees             NA          NA         NA     FALSE
## boxcox                        NA          NA         NA     FALSE
```

The output is clunky but packed with information. Fist, it is important to note (again) that the anisotropy measures are the opposite of anisotropic. The “shape” of the function is *exponential* with a 0° angle. We also see that the data become independent around 0.616 kilometers (*effective range*). We also see the relevant estimates for the beta coefficients and the respective p-values. Unlike OLS regression, these beta coefficients are not as useful in this spatial context.

However, this model assumes that there is no directionality in the correlation. The following model does:

```
rad_fit_aniso <- lgm(
  log_Dose ~ Dist_km + Dist_km_sq,
  data = rad_sp,
  grid = 500,
  shape = 0.5,
  fixShape = FALSE,
  nugget = 0.05,
  fixNugget = FALSE,
  aniso = TRUE,
  reml = TRUE
  )
```

```
summary(rad_fit_aniso)
```

```
##                      estimate    stdErr      ci0.005      ci0.995
## (Intercept)       -0.245672617 0.8737335 -2.496261056  2.004915821
## Dist_km           -0.046506727 0.0554415 -0.189314577  0.096301123
## Dist_km_sq         0.001059555 0.0013226 -0.002347238  0.004466347
## sdNugget           0.166575927        NA  0.143628813  0.193189228
## sdSpatial          0.671884120        NA  0.241937263  1.865889800
## range              0.677367946        NA  0.101340853  4.527565341
## shape              0.709061207        NA  0.282311655  1.135810759
## anisoRatio         0.339894667        NA  0.228807612  0.504914953
## anisoAngleRadians  0.626444243        NA  0.487855414  0.765033072
## anisoAngleDegrees 35.892611224        NA 27.952056221 43.833166227
## boxcox             1.000000000        NA           NA           NA
##                       ci0.025      ci0.975       ci0.05       ci0.95
## (Intercept)       -1.958158874  1.466813640 -1.682836388  1.191491154
## Dist_km           -0.155170078  0.062156623 -0.137699886  0.044686431
## Dist_km_sq        -0.001532694  0.003651803 -0.001115929  0.003235038
## sdNugget           0.148810033  0.186462828  0.151532890  0.183112324
## sdSpatial          0.308860844  1.461591133  0.349968530  1.289911041
## range              0.159605071  2.874766640  0.201361655  2.278623170
## shape              0.384344910  1.033777504  0.436550700  0.981571713
## anisoRatio         0.251515342  0.459329374  0.263991792  0.437621123
## anisoAngleRadians  0.520991171  0.731897315  0.537945235  0.714943251
## anisoAngleDegrees 29.850595241 41.934627207 30.821991568 40.963230880
## boxcox                      NA           NA           NA           NA
##                         ci0.1        ci0.9      pval Estimated
## (Intercept)       -1.3654071946  0.874061960 0.7785757      TRUE
## Dist_km           -0.1175578734  0.024544418 0.4015573      TRUE
## Dist_km_sq        -0.0006354257  0.002754535 0.4230652      TRUE
## sdNugget           0.1547340600  0.179324058        NA      TRUE
## sdSpatial          0.4041982018  1.116848787        NA      TRUE
## range              0.2632344343  1.743036905        NA      TRUE
## shape              0.4967406329  0.921381781        NA      TRUE
## anisoRatio         0.2791463735  0.413863104        NA      TRUE
## anisoAngleRadians  0.5574921852  0.695396301        NA      TRUE
## anisoAngleDegrees 31.9419493222 39.843273126        NA      TRUE
## boxcox                       NA           NA        NA     FALSE
```

We see similar values as the isotropic model. However, the shape and anisotropic parameters are considered and estimated. The “anisoAngleDegrees” estimates the minor axis of the spatial correlation in degrees. The major axis, intuitively, is calculated by adding 90° to the the minor axis estimate. The angle and ratio will be discussed in part g.

The table below compares the three models (OLS, anisotropic, and isotropic) based on their Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). These tests effectively estimate the out-of-sample prediction error and overall reliability. In short, the comparatively smaller the number, the better.

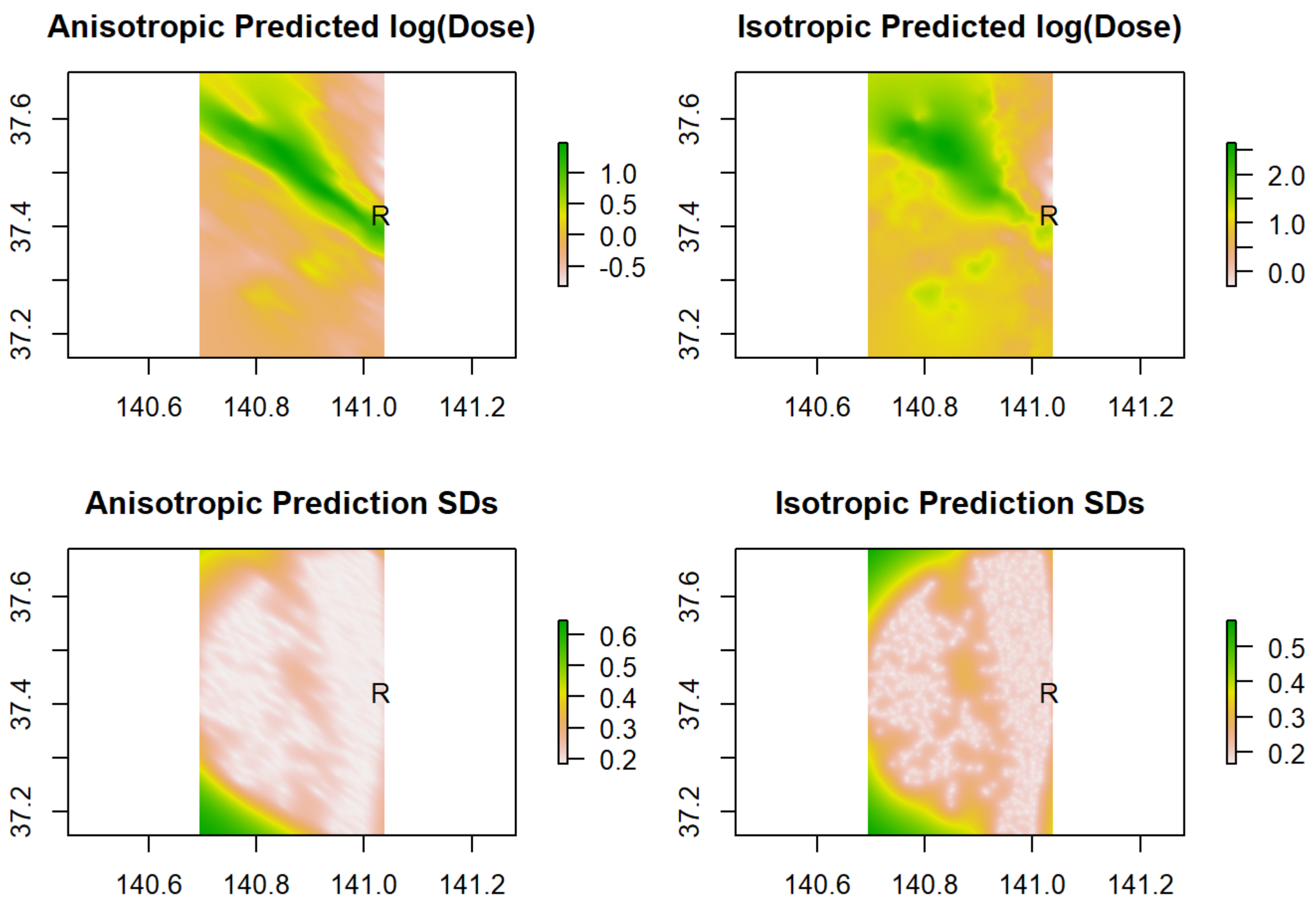
```
model_comp <- data.frame(
  model = c("OLS", "Anisotropic", "Isotropic"),
  AIC = c(AIC(model), AIC(rad_fit_aniso),AIC(rad_geostat_iso)),
  BIC = c(BIC(model), BIC(rad_fit_aniso), BIC(rad_geostat_iso))
  )

model_comp
```

```
##         model        AIC        BIC
## 1         OLS 1144.15373 1162.94680
## 2 Anisotropic -130.95484  -88.67043
## 3   Isotropic  -84.98471  -56.79510
```

Unsurprisingly, the model that accounts for the most information (spatial effects, distance, *and* orientation) is the best based on AIC and BIC.

2f) Compare radiation dose-rate predictions for the anisotropic vs. isotropic model using a 4-panel figure. Plot the two (2) dose-rate prediction surfaces and two (2) prediction standard deviation surfaces. Add anything to the map you think will help with its readability. Comment on how the predictions differ between the two models.

The following 4-panel figure shows the predicted values for log(Dose) and standard deviations for both spatial models:



As one can see in the “Prediction SD” plots, the isotropic model fits perfectly round circles while the anisotropic model fits ellipse rotated to the same extent as listed in the anisotropic degree (~126°). Furthermore, the anisotropic model fits the predicted values for log(Dose) more tightly while the isotropic model fits the values more loosely.

2g) Examine the anisotropy parameters (ratio and angle) from your anisotropic model. Comment on how this aligns with what you found for part c).

Based on this anisotropic model, the angle in the *minor axis* is 35.89° which translates to approx. 125.89° in the major axis (90° + 35.89°). This confirms the visual inspection done in the previous plot (in residuals from part b, part c, and part f).

The anisotropic ratio estimates the amount of stretch occuring within the data and is a function of the range parameter and anisotropic angle.

2h) Comment briefly on what the latent spatial effects may be measuring in your model.

I would imagine that wind direction should have some influence on spread of radiation, since wind had such a major effect during similar disasters. Additionally, the dose of radiation experienced may be some function of proximity to water reservoir (since radiation can be spread via drinking water). Looking at pictures of Namie, Japan (located in the Fukushima Prefecture), it seems that there are areas with higher terrain. It would make sense that these areas would be more exposed to radiation, especially if the slope faces that of the reactors.

Full Code

```
setwd("H:/Desktop/Spatial/Homework 2")    # Work
setwd("C:/Users/keete/Documents/Fall 2019/Spatial Stats/Homework 2")   # Not work
### Homework 3

# Libraries

library(geoR)
library(tidyverse)
library(readxl)
library(sf)
library(sp)
library(mvtnorm)
library(gstat)
library(ggfortify)
library(maps)
library(maptools)
library(sf)
library(tmap)
library(tmaptools)
library(geostatsp)
library(RColorBrewer)


# Question 1

# Prelim

colo <- read_excel("co_precip.xls")             # Read in

ggplot(
  data = colo,
  aes(
    x = Elevation,
    y = Precip
  )
) +
  geom_point(

  ) +
  geom_smooth(
    method = "lm"
  ) +
  theme_classic(

  )

summary(colo)

colo_sf <- st_as_sf(
  colo,
  coords = c("Longitude", "Latitude"),
  crs = CRS("+proj=utm +zone=13")
)
```

```
head(colo_sf)

# Variogram for Nuggest estimation

cutoff <- .7*max(
  dist(
    cbind(
      colo$Longitude,
      colo$Latitude
    )
  )
)

bins <- 10

variogram <- variogram(
  logPrecip ~ 1,
  locations = ~Longitude + Latitude,
  data = colo,
  cutoff = cutoff,
  width = cutoff/bins
)

plot(
  variogram,
  cex = 2,
  pch = 19,
  ylab = expression(
    paste(
      "Average", (0.7*(Y(s[i]) - Y(s[j])^2))
    )
  ),
  xlab = "Euclidean Distance (km)"
)
# Maximum likelihood by hand

H <- as.matrix(
  dist(
    st_coordinates(
      colo_sf
    )
  )
)

X <- model.matrix(
  ~ scale(Zelevation),
  data = colo_sf
)

Y <- as.matrix(
  colo_sf$logPrecip
)
```

```
MVN_fun <- function(theta, H, X, Y){

  part_sill <- theta[1]
  phi <- theta[2]
  p <- theta[3]
  nugget <- theta[4]

  Sigma <- part_sill*exp(-(phi*H)^p) + nugget*diag(length(Y))

  Sigma_inverse <- solve(Sigma)

  Beta <- solve(t(X) %*% Sigma_inverse %*% X) %*% t(X) %*% Sigma_inverse %*% Y

  log_L <- mvtnorm::dmvnorm(
    x = t(Y),
    mean = X %*% Beta,
    sigma = Sigma,
    log = TRUE
    )

  return(-1*log_L)

}


MLE <- optim(
  c(0.1, 1, 0.5, 0.1),
  fn = MVN_fun,
  H = H,
  X = X,
  Y = Y,
  hessian = TRUE
  )

MLE

# Maximum Likelihood by geoR

d <- data.frame(
  st_coordinates(
    colo_sf
  ),
  colo[, -1:-2]
)

d_geo <- as.geodata(
  d,
  coords.col = 1:2,
  data.col = 4,
  covar.col = 6
)

fit_geoR <- geoR::likfit(
  d_geo,
```

```
  ini.cov.pars = c(0.09, 1),
  nugget = .1,
  fix.nugget = FALSE,
  trend = ~ scale(Zelevation),
  cov.model = "powered.exponential",
  kappa = 1,
  fix.kappa = FALSE,
  lik.method = "ML"
)

fit_geoR

summary(fit_geoR)

MLE

q1_model_comp <- data.frame(
  model = c("By Hand", "geoR"),
  log.likelihood = c(MLE$value, fit_geoR$loglik),
  nugget = c(MLE$par[4], fit_geoR$nugget),
  partial.sill = c(MLE$par[1], fit_geoR$sigmasq),
  range_parm = c(MLE$par[2], 1/fit_geoR$phi),
  power = c(MLE$par[3], fit_geoR$kappa)
  )
######### Question 2

rad_df <- read_excel("Fukushima_30km_2016.xlsx")

# Splorin'

summary(rad_df)

rad_df <- rad_df %>%                 # Change Rain to 0/1
  mutate(
    Rain_code = ifelse(
      test = Rain == "Rain",
      yes = 1,
      no = 0
      ),
    interaction = Dist_km*Rain_code
    )


model <- lm(
  log_Dose ~ Dist_km + Dist_km_sq,
  data = rad_df
)

summary.lm(model)

autoplot(
  model,
  smooth.colour = NA
) +
```

```
  theme_classic(

  )

rad_df <- rad_df %>%
  mutate(resid = resid(model))

rad_df_points <- st_as_sf(
  x = rad_df,
  coords = c(
    "LONG",
    "LAT"
  ),
  crs = CRS(
    "+proj=utm +zone=54"
  )
)

data(world.cities)
data(World, metro, rivers, land)

world.cities <- world.cities %>%
  filter(
    country.etc == "Japan"
    )

japan_cities <- st_as_sf(
  x = world.cities,
  coords = c(
    "long",
    "lat"
  ),
  crs = CRS(
    "+proj=utm +zone=54"
  )
)

summary(japan_cities)
summary(rad_df_points)

japan_map <- map(
  database = "world",
  region = "Japan",
  plot = F,
  fill = T
)

summary(japan_map)

japan_fill <- map2SpatialPolygons(
  japan_map,
  IDs = japan_map$names,
  proj4string = CRS(
    "+proj=utm +zone=54"
```

```
  )
)


max_lon <- max(rad_df$LONG)
min_lon <- min(rad_df$LONG)

max_lat <- max(rad_df$LAT)
min_lat <- min(rad_df$LAT)

correction <- .12

tm_shape(
  japan_fill,
  proj4string = CRS(
    "+proj=utm +zone=54"
  ),
  ylim = c(
    min_lat - correction,
    max_lat + correction
  ),
  xlim = c(
    min_lon - correction,
    max_lon + correction
    )
  ) +
  tm_fill(
    col = "lightgreen"
  ) +
  tm_borders(
    lwd = 2,
    col = "black"
  ) +
  tm_shape(
    rad_df_points
  ) +
  tm_symbols(
    col = "resid",
    palette = "-Spectral",
    n = 7,
    style = "jenks",
    border.lwd = 0.1,
    border.col = 'gray',
    alpha = 0.9,
    scale = .75,
    title.col = "Residuals"
  ) +
  tm_shape(
    japan_cities
  ) +
  tm_text(
    "name",
    textNA = "",
    remove.overlap = T,
```

```
    shadow = T
  ) +
  tm_legend(
    position = c(
      "left",
      "bottom"
    ),
    legend.outside = TRUE,
    frame = F,
    main.title = 'Nonspatial Residuals (OLS)'
  ) +
  tm_layout(
    bg.color = "skyblue",
    saturation = .85
  ) +
  tm_compass(
    north = 0,
    type = "arrow"
  )

cutoff_rad <- .65*max(
  dist(
    cbind(
      rad_df$LONG,
      rad_df$LAT
    )
  )
)

bins_rad <- 10

variogram <- variogram(
  log_Dose ~ 1,
  locations = ~LONG + LAT,
  data = rad_df,
  cutoff = cutoff_rad,
  width = cutoff_rad/bins_rad
)

plot(
  variogram,
  pch = 16,
  cex = 1.5
  )

######################### Isotropic Models

### geostatsp

rad_d <- data.frame(
  st_coordinates(
    rad_df_points
  ),
  rad_df[, -1:-2]
```

```
)

rad_sp_iso  <- SpatialPointsDataFrame(
  coords = rad_d[, 1:2],
  data = rad_d,
  proj4string = CRS(
    "+proj=utm +zone=54"
  )
)

rad_geostat_iso <- lgm(
  log_Dose ~ Dist_km + Dist_km_sq,
  data = rad_sp_iso,
  grid = 500,
  shape = 0.5,
  fixShape = TRUE,
  fixNugget = FALSE,
  aniso = FALSE,
  reml = TRUE
)

summary(rad_geostat_iso)

################################### Anisotropic Models

# Geostatsp

rad_fit_aniso <- lgm(
  log_Dose ~ Dist_km + Dist_km_sq,
  data = rad_sp_iso,
  grid = 500,
  shape = 0.5,
  fixShape = FALSE,
  nugget = 0.05,
  fixNugget = FALSE,
  aniso = TRUE,
  reml = TRUE
  )

summary(rad_fit_aniso)


# Plots

reactor <- c(141.0298, 37.4245)

par(mfcol=c(2,2), mai=c(0.5,0.5,0.5,0.5))

plot(rad_fit_aniso$predict[["krigeSd"]],main='Anisotropic Prediction SDs')
points(reactor[1],reactor[2],pch="R")

plot(rad_geostat_iso$predict[["krigeSd"]],main='Isotropic Prediction SDs')
points(reactor[1],reactor[2],pch="R")
```

```
plot(rad_fit_aniso$predict[["predict"]],main='Anisotropic Predicted log(Dose)')
points(reactor[1],reactor[2],pch="R")

plot(rad_geostat_iso$predict[["predict"]],main='Isotropic Predicted log(Dose)')
points(reactor[1],reactor[2],pch="R")

model_comp <- data.frame(
  model = c("OLS", "Anisotropic", "Isotropic"),
  AIC = c(AIC(model), AIC(rad_fit_aniso),AIC(rad_geostat_iso)),
  BIC = c(BIC(model), BIC(rad_fit_aniso), BIC(rad_geostat_iso))
  )

model_comp
```