DSC383 PR 5

2024-07-06

The file rain.txt linked below contains a data set of the average yearly rainfall at a set of 100 locations across Switzerland. There are four variables associated with each observation: x is the x-coordinate of the location, y is the y-coordinate of the location, rainfall is the average yearly rainfall value at the location (measured in millimeters), and altitude is the altitude of the location (in feet). Convert the altitude values to miles before answering the following questions. Note: You do not need to project the data (i.e., distances between locations can be calculated directly using the x, y coordinates given in the data set).

To convert altitude in feet to miles, we divide the altitude column by 5280

```
rain <- read.table("rain.txt", header=T)

rain$altitude <- rain$altitude / 5280
head(rain)
##          x         y rainfall   altitude
## 1 29.52739  80.71854      151 0.12916667
## 2 33.77939  99.52954      255 0.15397727
## 3 46.80639 102.58454       79 0.08257576
## 4 48.71439 121.45354      191 0.15776515
## 5 49.31639 113.65554      194 0.10965909
## 6 53.21039  79.09954      334 0.09943182
```

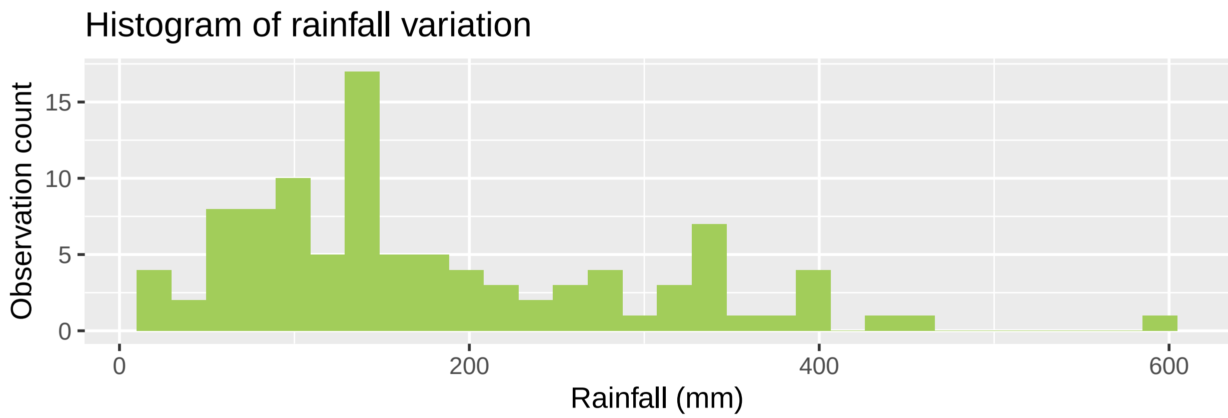
a.

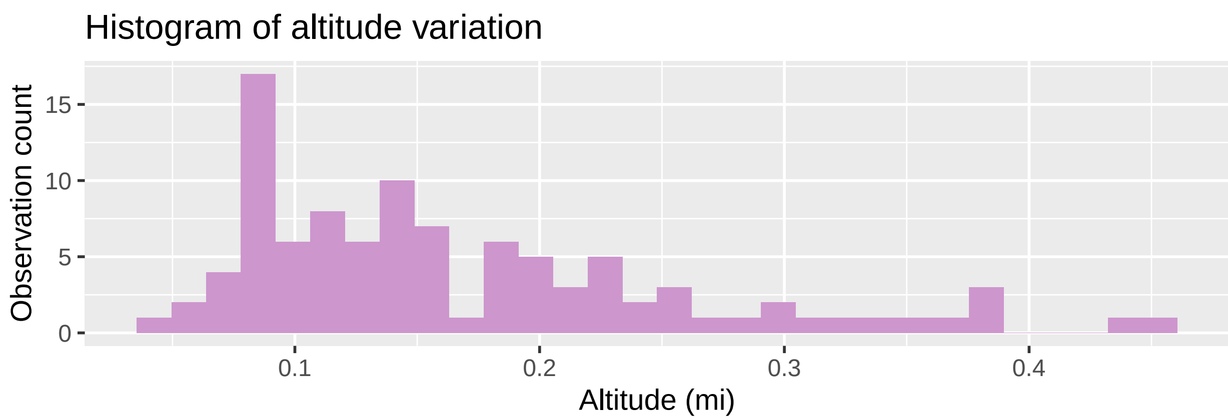
Construct exploratory plots showing the spatial variation in rainfall and altitude in the region. Briefly describe the spatial patterning (i.e., does there appear to be spatial dependence?) in average rainfall and altitude. Use x and y for axis labels.

Answer
Histograms of rainfall and altitude

```
rainfall_hist <- ggplot(data = rain,
                        aes(x = rainfall)) +
  geom_histogram(fill = "darkolivegreen3") +
  labs(title = "Histogram of rainfall variation",
       x = "Rainfall (mm)",
       y = "Observation count")


altitude_hist <- ggplot(data = rain,
                        aes(x = altitude)) +
  geom_histogram(fill = "plum3") +
  labs(title = "Histogram of altitude variation",
       x = "Altitude (mi)",
```

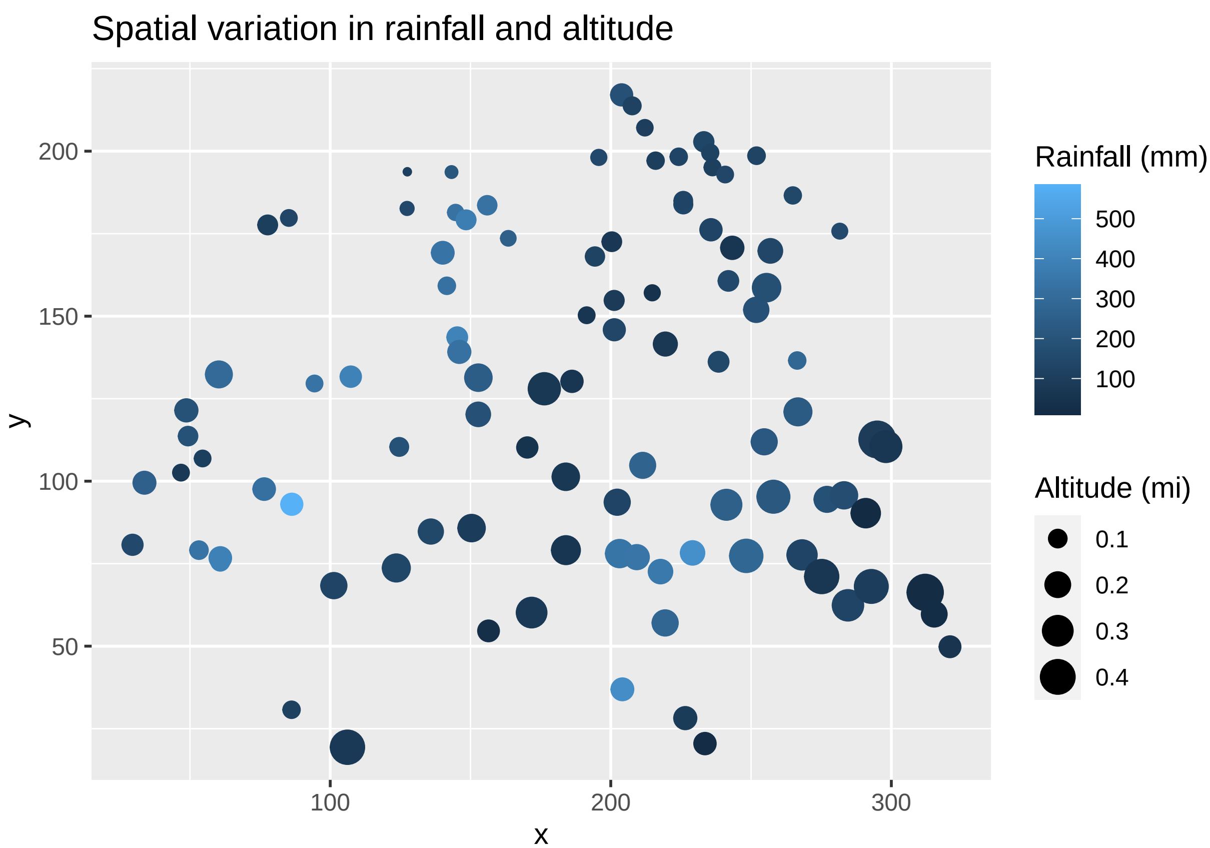
```
       y = "Observation count")

grid.arrange(rainfall_hist, altitude_hist,
             nrow = 2)
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





Plot of variation with coordindate points

```
ggplot(rain, aes(x, y)) +
  geom_point(aes(size = altitude,
                 color = rainfall)) +
  labs(title = "Spatial variation in rainfall and altitude",
       size = "Altitude (mi)",
       color = "Rainfall (mm)")
```



Spatial patterning
There seems to be some spatial dependence for altitude based on the plot above. As we move from the top left of the plot to the bottom right coordinates, the points increase in size, indicating a gradual growth in altitude at higher x and lower y coordinates. Rainfall, on the other hand, showcases less spatial dependence, if any at all. The spatial variance in rainfall shows more clusters of similar rainfall around x=50, 200, and 300, but no obvious trend.

b.

Fit a linear regression model of the square root of rainfall on altitude, and summarize the fitted model by reporting the estimated regression equation and the estimated error variance. What proportion of variation in the square root of rainfall is explained by altitude?

Answer
Fit linear regression model

```
reg_fit <- lm(sqrt(rainfall) ~ altitude, data = rain)
summary(reg_fit)
##
## Call:
## lm(formula = sqrt(rainfall) ~ altitude, data = rain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9070 -2.9051 -0.9749  3.4518 11.2529
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  14.1273     0.9001  15.695   <2e-16 ***
## altitude     -8.3681     4.6910  -1.784   0.0775 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.275 on 98 degrees of freedom
## Multiple R-squared:  0.03145,    Adjusted R-squared:  0.02157
## F-statistic: 3.182 on 1 and 98 DF,  p-value: 0.07754
```

Extract values from regression summary

```
# Coefficients for equation
reg_fit$coef
## (Intercept)    altitude
##   14.127308   -8.368113
# Standard error -> Error variance
reg_se <- summary(reg_fit)$sigma
err_variance <- reg_se^2
err_variance
## [1] 18.27963
# R squared
summary(reg_fit)$r.squared
## [1] 0.03145022
```

Estimated values
Regression equation: $\sqrt{\textbf{rainfall}} = \mathbf{14.127} - (\mathbf{8.368} * \textbf{altitude})$

Error variance: **18.280**

Proportion of variation in the square root of rainfall explained by altitude: $R^2 = \mathbf{0.0315}$

c.

Calculate the Euclidean distance between all pairs of observation locations and make a relative frequency (probability) histogram of these distances. Use a binwidth of 20 miles. Do not include the distances between individual points with themselves.

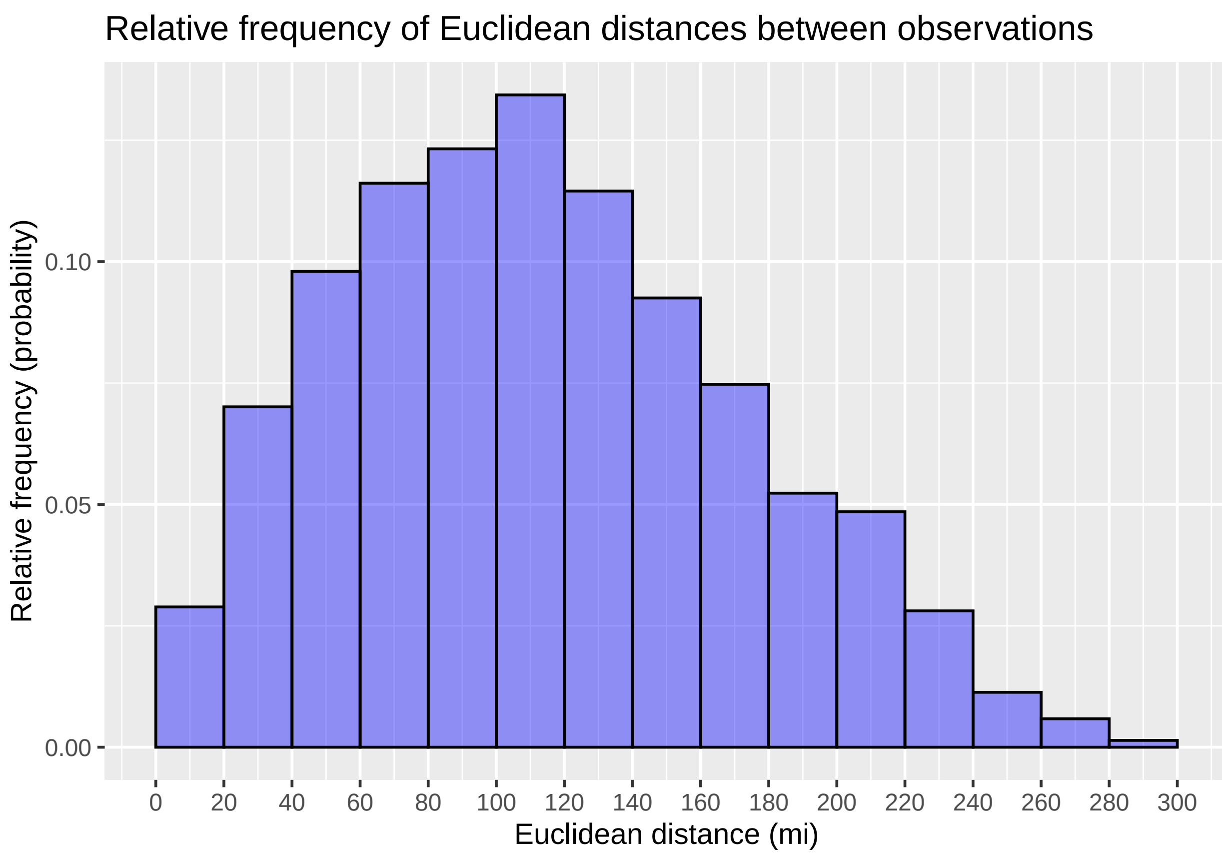
Answer
We first create a data frame with pairs of points and their distances.

```
# Create function to calcualte Euclidian distance
euclid_dist <- function(coord_1, coord_2) {
  sqrt(sum((coord_2 - coord_1)^2))
}

# Create empty data frame to store coordinates and distances
distances <- data.frame(
  index_1 = integer(),
  x1 = double(),
  y1 = double(),
  index_2 = integer(),
```

```r
  x2 = double(),
  y2 = double(),
  distance = double()
)

# Loop through all points,
# avoiding distances of 0 and repeated calculations
for (i in 1:(nrow(rain) - 1)) {
  for (j in (i+1):nrow(rain)) {
    x1 <- rain$x[i]
    y1 <- rain$y[i]
    x2 <- rain$x[j]
    y2 <- rain$y[j]
    distance <- euclid_dist(c(x1, y1),
                            c(x2, y2))
    # Add to distances data frame
    distances[nrow(distances)+1,] <- c(i, x1, y1,
                                       j, x2, y2,
                                       distance)
  }
}

head(distances)
##   index_1       x1       y1 index_2       x2        y2 distance
## 1       1 29.52739 80.71854       2 33.77939  99.52954 19.28557
## 2       1 29.52739 80.71854       3 46.80639 102.58454 27.86908
## 3       1 29.52739 80.71854       4 48.71439 121.45354 45.02756
## 4       1 29.52739 80.71854       5 49.31639 113.65554 38.42461
## 5       1 29.52739 80.71854       6 53.21039  79.09954 23.73827
## 6       1 29.52739 80.71854       7 54.51039 106.87954 36.17386
```

Histogram of distances

```r
relative_freq_hist <- ggplot(data = distances,
                             aes(x = distance,
                                 y = after_stat(count)/sum(after_stat(count)))) +
  geom_histogram(binwidth = 20,
                 col = "black",
                 fill = "blue",
                 alpha = 0.4,
                 boundary = 0) +
  scale_x_continuous(breaks = seq(0, 300, 20)) +
  labs(title = "Relative frequency of Euclidean distances between observations",
       x = "Euclidean distance (mi)",
       y = "Relative frequency (probability)")

relative_freq_hist
```



d.

Consider distance bins [0,20), [20,40), [40,60), [60,80), [80, 100), [100,120), [120,140), [140,160), [160,180), [180, 200), [200,220), [220,240), [240,260), [260,280), [280, 300). For each distance bin, calculate the correlation between all pairs of residuals from your fitted model corresponding to locations whose distance falls within the bin's limits. Make a scatter plot of the correlation between residuals and the center of the bins. Use color or point size to indicate the number of pairs of locations whose distance falls into each bin.

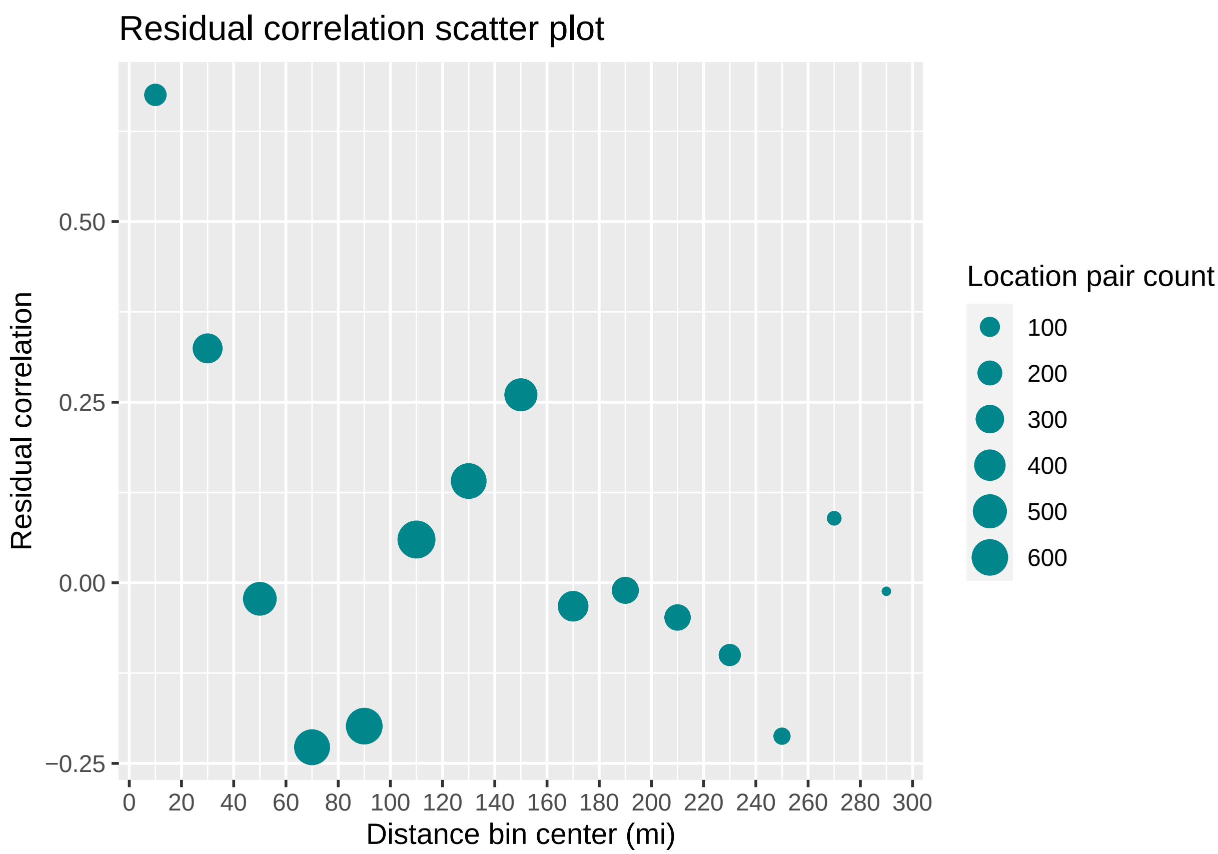
Answer
We create a data frame with rows corresponding to a bin and add columns for pair count within the bin and residual correlation.

```
residuals <- reg_fit$residuals

resid_correlations <- data.frame(
  bin_center = integer(),
  loc_pair_count = integer(), # Number of pairs of locations in bin
  residual_correlation = double()
)

bins <- seq(0, 300, 20)  # 0, 20, 40, ..., 300
total_bins <- 300/20  # 15

for (b in 1:total_bins) {
  min_dist <- bins[b]
```

```r
  max_dist <- bins[b + 1]
  bin_center <- mean(c(min_dist, max_dist))

  # Filter to all points in a bin range
  points_in_bin <- distances[distances$distance >= min_dist & distances$distance < max_dist, ]
  loc_pair_count <- nrow(points_in_bin)

  # Locate residuals at those points
  resid_1 <- residuals[points_in_bin$index_1]
  resid_2 <- residuals[points_in_bin$index_2]

  # Correlation
  resid_cor <- cor(resid_1, resid_2)

  # Add to resid_correlations data frame
  resid_correlations[nrow(resid_correlations)+1,] <- c(bin_center,
                                                       loc_pair_count,
                                                       resid_cor)
}

resid_correlations
##    bin_center loc_pair_count residual_correlation
## 1          10            143           0.67543865
## 2          30            347           0.32452161
## 3          50            485          -0.02217726
## 4          70            575          -0.22768508
## 5          90            610          -0.19846987
## 6         110            665           0.05989266
## 7         130            567           0.14086698
## 8         150            458           0.26025208
## 9         170            370          -0.03246737
## 10        190            259          -0.01046816
## 11        210            240          -0.04803162
## 12        230            139          -0.10000850
## 13        250             56          -0.21237153
## 14        270             29           0.08931016
## 15        290              7          -0.01179919
```

Scatter plot of residual correlations

```r
resid_cor_scatter <- ggplot() +
  geom_point(data = resid_correlations,
             aes(x = bin_center, y = residual_correlation,
                 size = loc_pair_count),
             color = "turquoise4") +
  scale_x_continuous(breaks = seq(0, 300, 20)) +
  scale_size_continuous(breaks = seq(100, 1000, by = 100)) +
  labs(title = "Residual correlation scatter plot",
       x = "Distance bin center (mi)",
       y = "Residual correlation",
       size = "Location pair count")
resid_cor_scatter
```



e.

Explain why there are fewer pairs of locations in the longer-distance bins. (Hint: Why do you expect more pairs of locations in the [80, 100) than in the [280,300) bin even without looking at the histogram of pairwise distances?)

Answer
Especially since the data is taken within a single country, we expect most of the observations to be closer together. Very far points would most likely be from one end of the country to the opposite side of the country, whereas we would expect to encounter more observations within the borders of Switzerland.

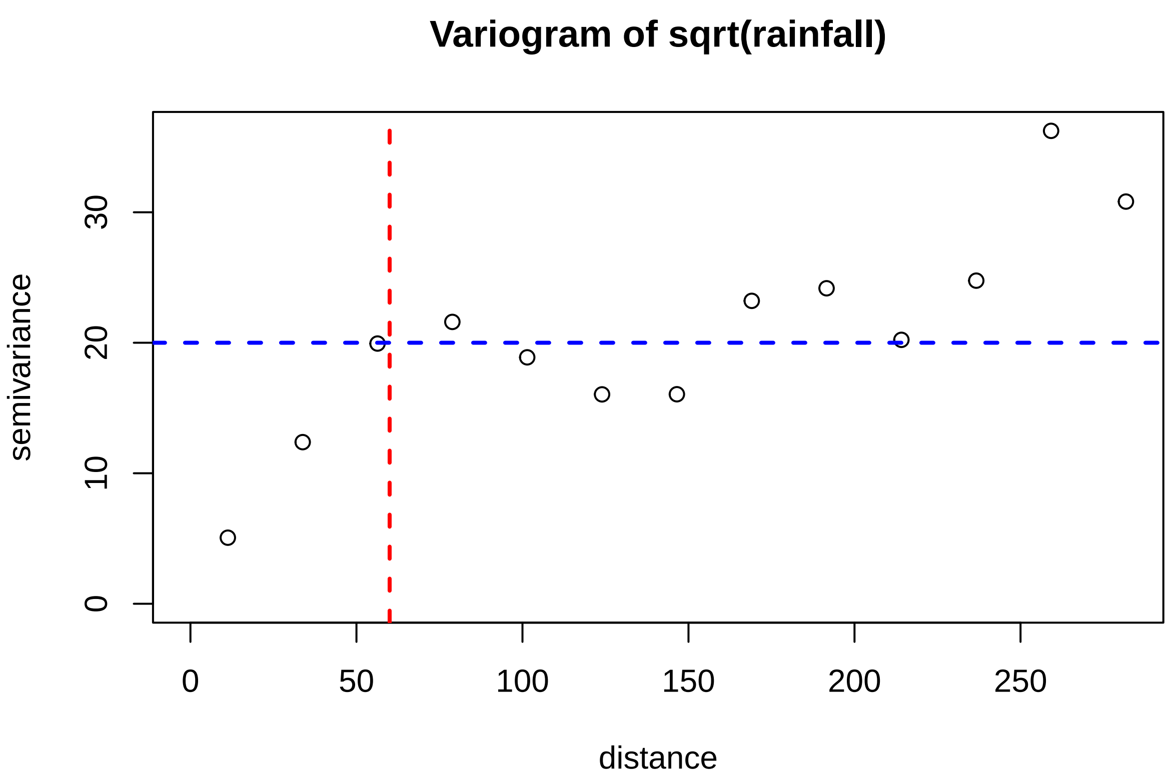
f.

Use the likfit() function in geoR to fit a normal spatial linear regression model with an exponential covariance structure of the square-root of rainfall on altitude. Assume that the nugget effect is zero (i.e., $\sigma_\epsilon^2 = 0$). Provide the numerical value of the estimated intercept (β_0), slope (β_1), and covariance parameters (σ^2, ϕ).

Answer
First we make a geodata object out of the rainfall data, then plot a variogram to estimate initial values for the covariance parameters using sill and range.

```
# create geodata (geoR) object
sqrt_rainfall_geo <- as.geodata(
  cbind(sqrt(rain$rainfall),
        rain$altitude,
```

```r
        rain$x, rain$y),
              data.col = 1,
        covar.col = 2,
                coords.col = 3:4)

# Determine sill and range
variog_sqrt_rainfall <- variog(sqrt_rainfall_geo)
## variog: computing omnidirectional variogram
plot(variog_sqrt_rainfall,
     main = "Variogram of sqrt(rainfall)")
abline(h = 20, v = 60,
       col=c("blue", "red"),
       lty = 2, lwd = 2)
```



From the plot above, we will use a sill of 20 (where the scatter plot flattens) and a range of 60 (the distance at which the scatter plot flattens). We now fit the data using the estimated covariance parameters

```r
ini_sill <- 20
ini_range <- 60

spat_reg_fit <- likfit(
  sqrt_rainfall_geo,
  trend = ~ rain$altitude,
  cov.model = "exponential",
  ini.cov.pars = c(ini_sill, ini_range),
```

```r
  nugget = 0,
  fix.nugget = TRUE)
## kappa not used for the exponential correlation function
## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optimize.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optimize.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
summary(spat_reg_fit)
## Summary of the parameter estimation
## -----------------------------------
## Estimation method: maximum likelihood
##
## Parameters of the mean component (trend):
##   beta0   beta1
## 11.5970  0.0288
##
## Parameters of the spatial component:
##    correlation function: exponential
##       (estimated) variance parameter sigmasq (partial sill) =  20.97
##       (estimated) cor. fct. parameter phi (range parameter)  =  42.41
##    anisotropy parameters:
##       (fixed) anisotropy angle = 0  ( 0 degrees )
##       (fixed) anisotropy ratio = 1
##
## Parameter of the error component:
##       (fixed) nugget = 0
##
## Transformation parameter:
##       (fixed) Box-Cox parameter = 1 (no transformation)
##
## Practical Range with cor=0.05 for asymptotic range: 127.0387
##
## Maximised Likelihood:
##    log.L n.params      AIC      BIC
## "-247.7"      "4"  "503.5"  "513.9"
##
## non spatial model:
##    log.L n.params      AIC      BIC
## "-286.2"      "3"  "578.3"  "586.2"
##
## Call:
## likfit(geodata = sqrt_rainfall_geo, trend = ~rain$altitude, ini.cov.pars = c(ini_sill,
##     ini_range), fix.nugget = TRUE, nugget = 0, cov.model = "exponential")
```

Extract values from fit summary

```r
# Intercept and slope
spat_reg_fit$beta
##   intercept      covar1
## 11.59701590  0.02875611
# Covariance parameters
spat_reg_fit$cov.pars
## [1] 20.97328 42.40656
```

Estimated values
Intercept: $\beta_0 = \mathbf{11.597}$
Slope: $\beta_1 = \mathbf{0.029}$
Covariance parameters: $(\sigma^2, \phi) = (\mathbf{20.973}, \mathbf{42.407})$

g.

Add the fitted exponential correlation to the plot you made in part d.

Answer
We will use the following formula:

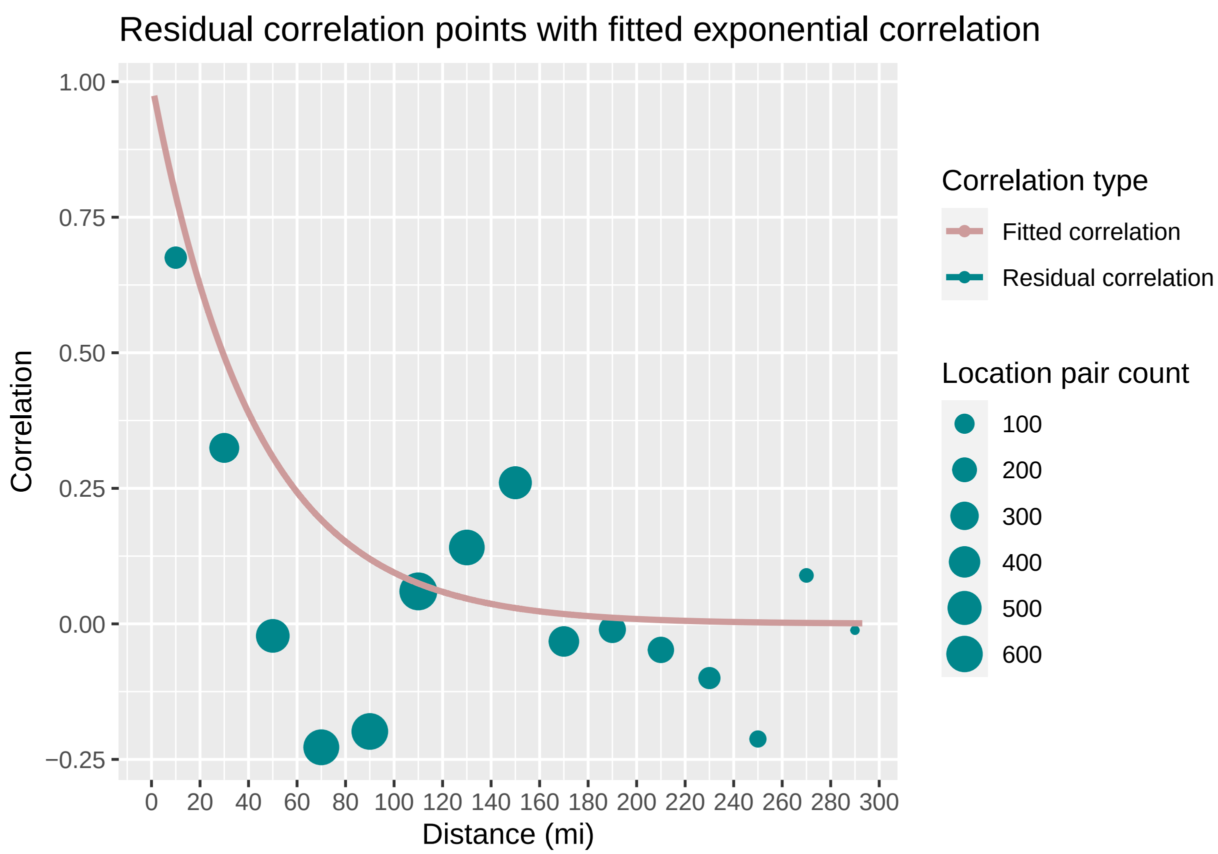
$$\rho(h) = exp(-\frac{h}{\phi})$$

where h is the distance between the two points

```r
# Fitted parameters
phi <- spat_reg_fit$cov.pars[2]

# Exponential correlation equation
exp_cor <- function(h, phi) {
  exp(-h / phi)
}

# Calculate exponential correlation based on distances
distances$fitted_correlation <- exp_cor(distances$distance,
                                        phi)
```

```r
ggplot() +
  geom_point(data = resid_correlations,
             aes(x = bin_center, y = residual_correlation,
                 size = loc_pair_count,
                 color = "Residual correlation")) +
  scale_x_continuous(breaks = seq(0, 300, 20)) +
  scale_size_continuous(breaks = seq(100, 1000, by = 100)) +
  geom_line(data = distances,
            aes(x = distance, y = fitted_correlation,
                color = "Fitted correlation"),
            linewidth = 1.1) +
  scale_color_manual(name = "Correlation type",
                     values = c("Residual correlation" = "turquoise4",
                                "Fitted correlation" = "rosybrown3")) +
  guides(size=guide_legend(override.aes=list(colour="turquoise4"))) +
  labs(title = "Residual correlation points with fitted exponential correlation",
       x = "Distance (mi)",
```

```
       y = "Correlation",
       size = "Location pair count")
```



h.

Report the AIC values for both the non-spatial and spatial regression models. Based on the AIC, which model do you believe better fits the data, the non-spatial or spatial regression model?

Answer

```
# Non-spatial
AIC(reg_fit)
## [1] 578.3462
# Spatial
 AIC(spat_reg_fit)
## [1] 503.4867
```

AIC values
Non-spatial model: **578.346**
Spatial model: **503.487**
The spatial regression model fits the data better based on its lower AIC value.

i.

What information (in addition to coordinates) would you need to predict rainfall at unmonitored locations in the study region? (Hint: Think about why you are not able to make a plot similar to the last plot in the DEMO-kriging.R example?)

Answer

In addition to coordinates, we would need covariate values and a convex hull enclosing the study region to make predictions on rainfall at unmonitoried locations, so that we have additional spatial information (such as altitude, x, and y) on both monitored and unmonitored locations.