

Mantel Test Script

Grace Saville

16/05/2022

1. Loading the data

```
data <- read.csv("./data/RStudio/ratsSNPs_clean.csv")
gen.matrix <- as.matrix(
  read.delim("./data/SplitsTree/geneticdist_SplitsTree_output.txt",
  sep = "\t", header = FALSE))
# Make sure the text file is the distances only, remove any extras e.g.
# column vector at bottom of file

dim(data)

## [1] 370 298
```

2. Creating geographical distance matrix

ONLY NEED TO RUN THE FOLLOWING CODE CHUNK ONCE AND THEN SAVE

Using the mapview funtion in the last code chunk of this script I noticed that Tahanea is plotted in Australia, and found out the longitude number is missing a negative sign, and that Reiono and Honuea have the same coordinates even though they're different islands. Fixing that here:

```
x <- grep("Tahanea", data$island.1, value = FALSE) # finding rows of Tahanea
names(data) # finding column number for "geo_long"
data[x, 12]
data[x, 12] <- -144.97 # replacing the number

# the coordinates given for Reiono and Honuea point to an island in
# French Polynesia called Moorea-Maiao
# Reino should be approx. -17.046, -149.546
# Honuea should be approx -17.009, -149.585
# Checking other islands close by:
# Rimatu'u is -17, -149.57 (over the sea) but should be approx -17.03, -149.558

x <- grep("Reiono", data$island.1, value = FALSE)
data[x, c(11, 12)]
data[x, 11] <- -17.046 # replcing geo_lat
data[x, 12] <- -149.546 # replacing geo_long
```

```

x <- grep("Honuea", data$island.1, value = FALSE)
data[x, c(11, 12)]
data[x, 11] <- -17.009 # replcing geo_lat
data[x, 12] <- -149.585 # replacing geo_long

x <- grep("Rimatuu", data$island.1, value = FALSE)
data[x, c(11, 12)]
data[x, 11] <- -17.03 # replcing geo_lat
data[x, 12] <- -149.558 # replacing geo_long

rm(x)

# write.csv(data, "./data/RStudio/ratsSNPs_clean.csv", row.names = FALSE)

```

(only need to do the above code once since it saves the edited df to file)

3. Setting up longitudes and latitudes

Different distance functions:

- distHaversine() assumes earth is a sphere
- distm() makes distance matrix
- distGeo() assumes earth is elliptical (ish), can choose specific model

```

head(names(data), n = 20) # need "geo_lat" "geo_long"

## [1] "island"           "registration.number" "genus"
## [4] "species"          "sex"                  "country"
## [7] "state_province"   "island.1"             "locality"
## [10] "site"              "geo_lat"              "geo_long"
## [13] "collector"         "collecting.date"      "field.number"
## [16] "Populatie"         "X299_CHR1_114679736" "X13_CHR1_116614092"
## [19] "X14_CHR1_124857905" "X15_CHR1_134869867"

longlat <- data[,c(1,11,12)]
head(longlat) # checking correct columns are used

##      island geo_lat geo_long
## 1 Borneo_002 -0.5102 117.0912
## 2 Borneo_003 -0.5102 117.0912
## 3 Borneo_004 -0.5102 117.0912
## 4 Borneo_005 -0.5102 117.0912
## 5 Borneo_006 -0.5102 117.0912
## 6 Borneo_008 -0.5102 117.0912

longlat <- as.matrix(longlat[,c(3,2)]) # distGeo function needs a matrix
# with 2 columns, col 1 longitude and col 2 latitude

geo.matrix <- distm(longlat, fun = distGeo) # converting to pairwise distance matrix
dim(geo.matrix) # 370 370

```

```
## [1] 370 370

geo.dist <- as.dist(geo.matrix, diag = TRUE, upper = TRUE) # converting to dist object
# diag = TRUE #includes diagonal zeros
# upper = TRUE #includes upper triangle
```

4. Creating genetic distance matrix

```
dim(gen.matrix) # 370 370

## [1] 370 370

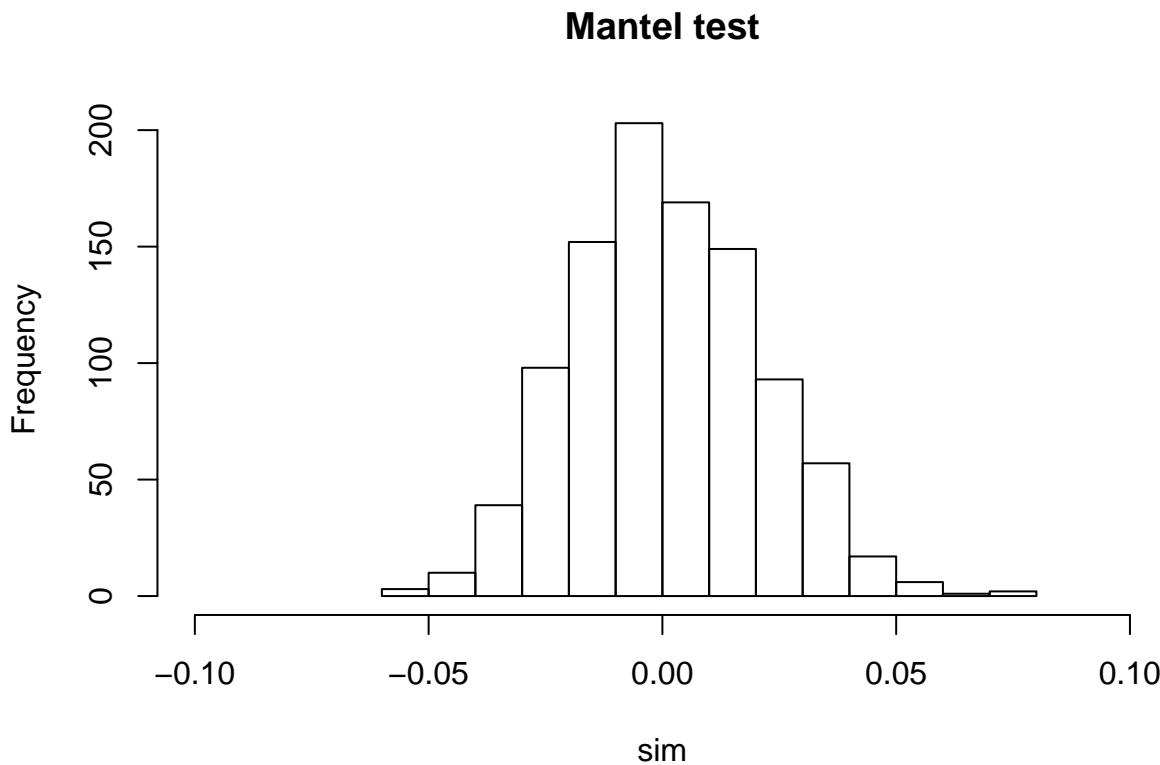
gen.dist <- as.dist(gen.matrix, diag = TRUE, upper = TRUE) # converting to dist object
```

5. Mantel test

```
r1 <- mantel.rtest(gen.dist, geo.dist, nrepet = 999)
r1

## Monte-Carlo test
## Call: mantelnoneuclid(m1 = m1, m2 = m2, nrepet = nrepet)
##
## Observation: 0.4987612
##
## Based on 999 replicates
## Simulated p-value: 0.001
## Alternative hypothesis: greater
##
##      Std.Obs   Expectation    Variance
## 2.485686e+01 9.793296e-04 4.010383e-04
```

```
plot(r1$plot$hist, main = "Mantel test", xlim = c(-0.1, 0.1)) # plotting
```



- -1 suggests strong negative correlation, e.g. closer islands mean further genetically or further islands means closer genetically
- 0 suggests no correlation, e.g. genetic difference is not correlated to island distance
- 1 suggests strong positive correlation e.g. closer islands mean closer genetically

Therefore the observed correlation of 0.4987612 suggests that there is a positive correlation between genetic distance and geographic distance (and the null hypothesis of no correlation is rejected).

6. Creating pairwise distances dataframe

```
specimens <- read.delim(  
  "./data/SplitsTree/geneticdist_SplitsTree_output_taxa_only.txt", header = FALSE)  
specimens <- as.vector(specimens[,2])  
  
colnames(gen.matrix) <- specimens  
rownames(gen.matrix) <- specimens # naming the rows and columns by the order  
# given in the SplitsTree output  
  
colnames(geo.matrix) <- data[,1]  
rownames(geo.matrix) <- data[,1] # naming the rows and columns here by the
```

```

# order of lat/long, which came from the data df

gen.matrix[lower.tri(gen.matrix)] <- NA # keeping only the upper triangle of each matrix
geo.matrix[lower.tri(geo.matrix)] <- NA

dist.summary <- data.frame(
  col = colnames(gen.matrix)[col(gen.matrix)],
  row = rownames(gen.matrix)[row(gen.matrix)],
  gen.dist = c(gen.matrix)
) # converting the genetic matrix into a df with columns describing which
# combos result in the distance

x <- data.frame(
  col = colnames(geo.matrix)[col(geo.matrix)],
  row = rownames(geo.matrix)[row(geo.matrix)],
  geo.dist = c(geo.matrix)
) # doing the same with the geographic matrix

dist.summary <- merge(dist.summary, x, by = 1:2, all = TRUE) # merging the 2 dfs
rm(x)

```

The above df “dist.summary” is not ideal since it’s not space efficient (~12MB), however it provides and easy way to link the row and column specimens that generated the distances, therefore making points on the graph labelable.

```

dist.summary <- tidyr::unite(dist.summary, specimens.combo, 1:2, sep = ":", remove = TRUE)
# combining the first 2 columns (the names of the matrices columns and rows)
# to give a label to each pairwise distance

dist.summary <- na.omit(dist.summary) # removing NA's that were in the bottom triangle

# creating a column of island combination labels, rather than specimen combination:
islands.combo <- dist.summary$specimens.combo
islands.combo <- gsub("[0-9]+", "", islands.combo) # removing specimen ID numbers
islands.combo <- gsub("_", "", islands.combo) # removing underscores
# unique(islands.combo)
islands.combo <- gsub("Thailand", "Mainland", islands.combo) # replacing 3
# countries with mainland
islands.combo <- gsub("Laos", "Mainland", islands.combo)
islands.combo <- gsub("Cambodia", "Mainland", islands.combo)
islands.combo <- gsub("Halmahera", "Halmaher", islands.combo) # replacing some double ups
islands.combo <- gsub("NewBritai", "NewBrita", islands.combo)
islands.combo <- gsub("NewGuinea", "NewGuine", islands.combo)
islands.combo <- gsub("Motukawan", "Motukawa", islands.combo)
dist.summary <- cbind(dist.summary, islands.combo) # adding this new column
rm(islands.combo)

# removing self-self distances, since I'm looking at inter-island not intra-island
intra <- as.character(c(
  "Aotea:Aotea",
  "Borneo:Borneo",
  "GrtMercury:GrtMercury",
  "Halmaher:Halmaher",

```

```

"Hatutaa:Hatutaa",
"Honuea:Honuea",
"Kaikura:Kaikura",
"Kamaka:Kamaka",
"Kayangel:Kayangel",
"LateIs:LateIs",
"Luzon:Luzon",
"Mainland:Mainland",
"Malenge:Malenge",
"Mohotani:Mohotani",
"Motukawa:Motukawa",
"NewBrita:NewBrita",
"NewGuine:NewGuine",
"Normanby:Normanby",
"Rakiura:Rakiura",
"Reiono:Reiono",
"Rimatuu:Rimatuu",
"Slipper:Slipper",
"Southland:Southland",
"Sulawesi:Sulawesi",
"Tahanea:Tahanea",
"WakeIs:WakeIs"
))
dist.summary[which(dist.summary$islands.combo %in% intra),4] <- NA # removing
# same-same island combos
dist.summary[dist.summary == 0] <- NA # turning zeros to NAs
dist.summary <- na.omit(dist.summary) # removing rows with NAs
rm(intra, specimens)

dist.summary$geo.dist <- dist.summary$geo.dist/1000 # going from metres to km

```

Rows which have either or both gen and geo dist at zero (although there should never be a gen.dist at zero) will sit on the axis and potentially pull on the regression line. Geo distances at 0 will always be specimens on the same island, which I think it best to remove since they don't contribute to whether there is a correlation between islands distance (since they're actually the same island).

```

# getwd()
write.csv(dist.summary, "./data/RStudio/gen_geo_distance_matrices_df.csv", row.names = FALSE)

```

7. Linear Modelling

```

dist.summary <- read.csv("./data/RStudio/gen_geo_distance_matrices_df.csv")

```

7a. Test Model

```

testLM <- lm(gen.dist ~ geo.dist, data = dist.summary) # model
summary(testLM) # model results

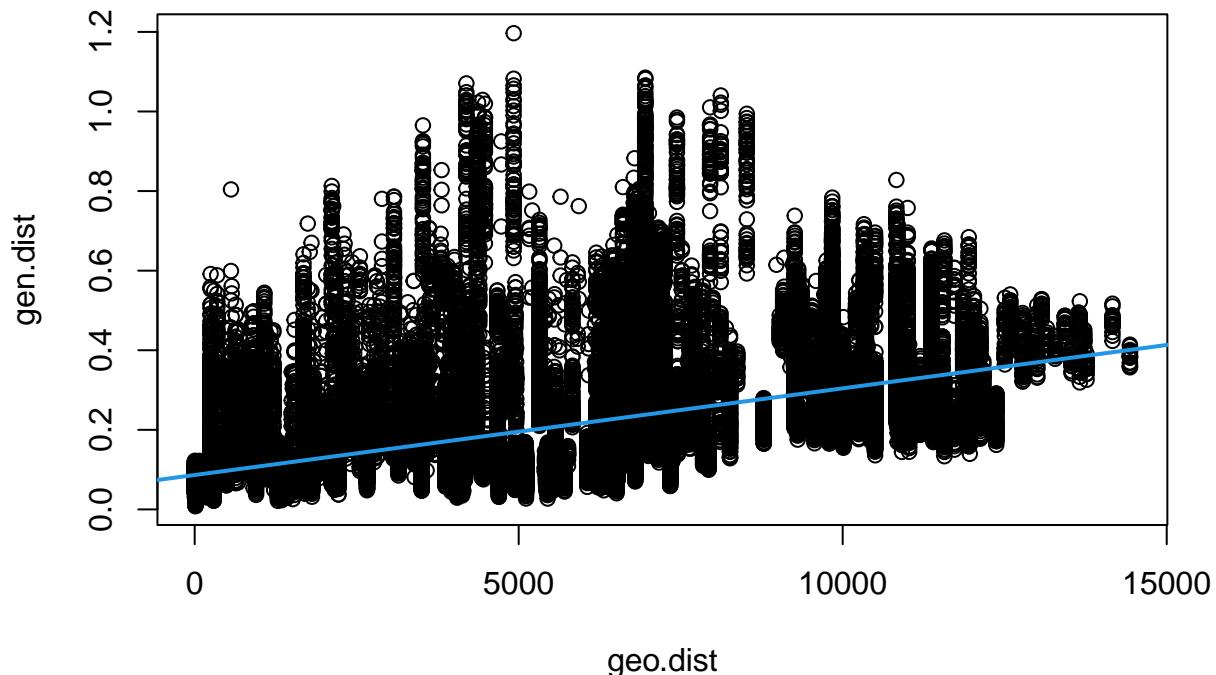
```

```

## 
## Call:
## lm(formula = gen.dist ~ geo.dist, data = dist.summary)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.20733 -0.08651 -0.03480  0.03001  1.00306 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 8.644e-02  1.014e-03  85.23   <2e-16 ***
## geo.dist    2.178e-05  1.619e-07 134.54   <2e-16 ***  
## ---        
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1346 on 65067 degrees of freedom
## Multiple R-squared:  0.2176, Adjusted R-squared:  0.2176 
## F-statistic: 1.81e+04 on 1 and 65067 DF,  p-value: < 2.2e-16

plot(gen.dist ~ geo.dist, data = dist.summary)
abline(coef = coef(testLM), col = 4, lwd = 2)

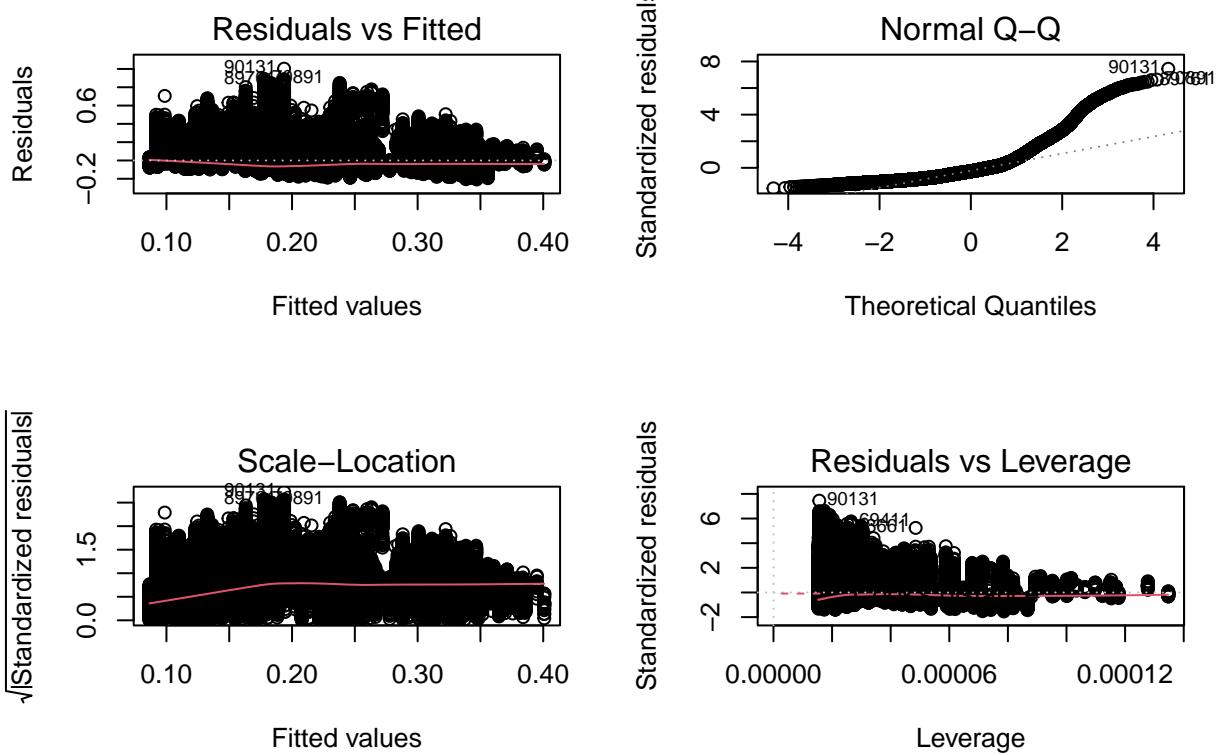
```



```

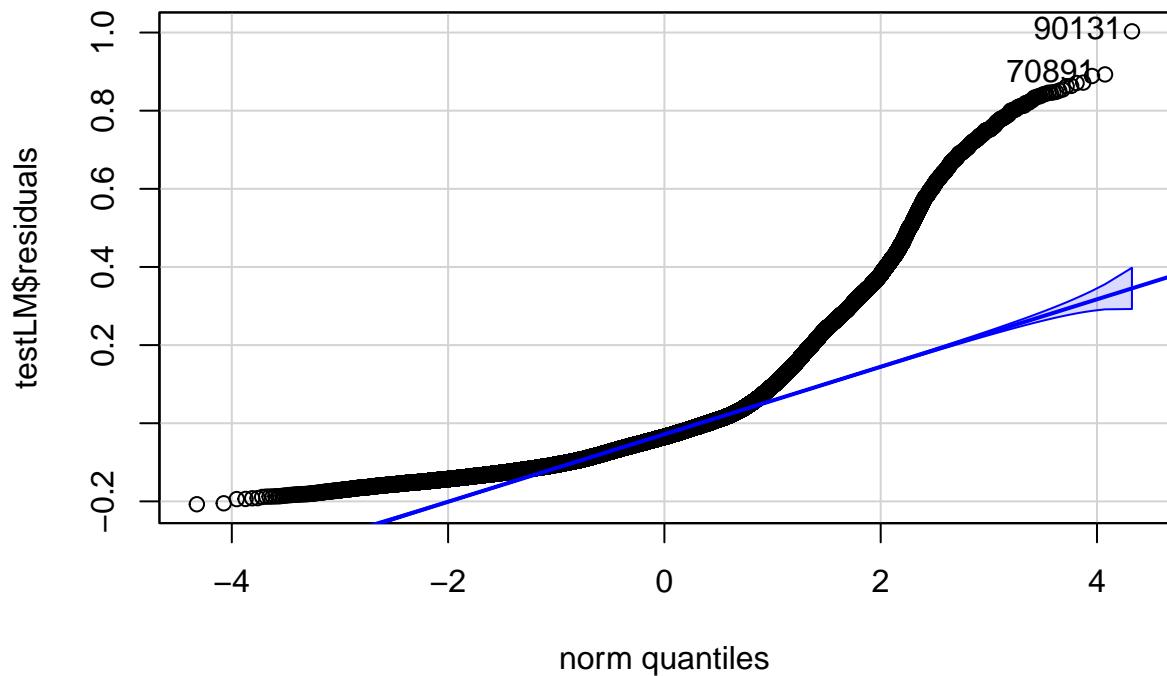
par(mfrow = c(2, 2)) # changes the number of plots visible at once
plot(testLM) # # diagnostic plots, not normal, distribution may be skewed

```



```
# towards the left side ("Right skew" or "positive skew")
par(mfrow = c(1, 1))

qqPlot(testLM$residuals, line = "quartiles") # non-normal dist
```



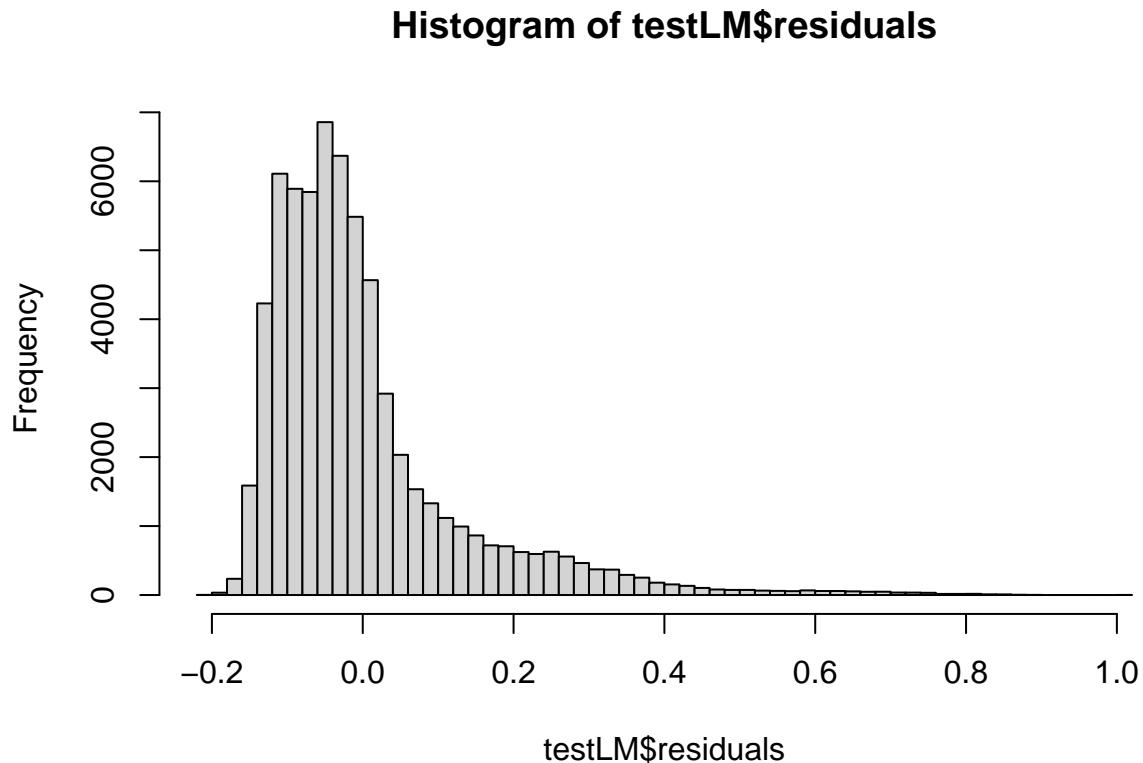
```
## 90131 70891
## 42506 32209
```

```
ks.test(testLM$residuals, 'pnorm') # non-normal
```

```
## Warning in ks.test(testLM$residuals, "pnorm"): ties should not be present for
## the Kolmogorov-Smirnov test
```

```
##
## One-sample Kolmogorov-Smirnov test
##
## data: testLM$residuals
## D = 0.43238, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

```
hist(testLM$residuals, breaks = 50) # clear positive skew
```

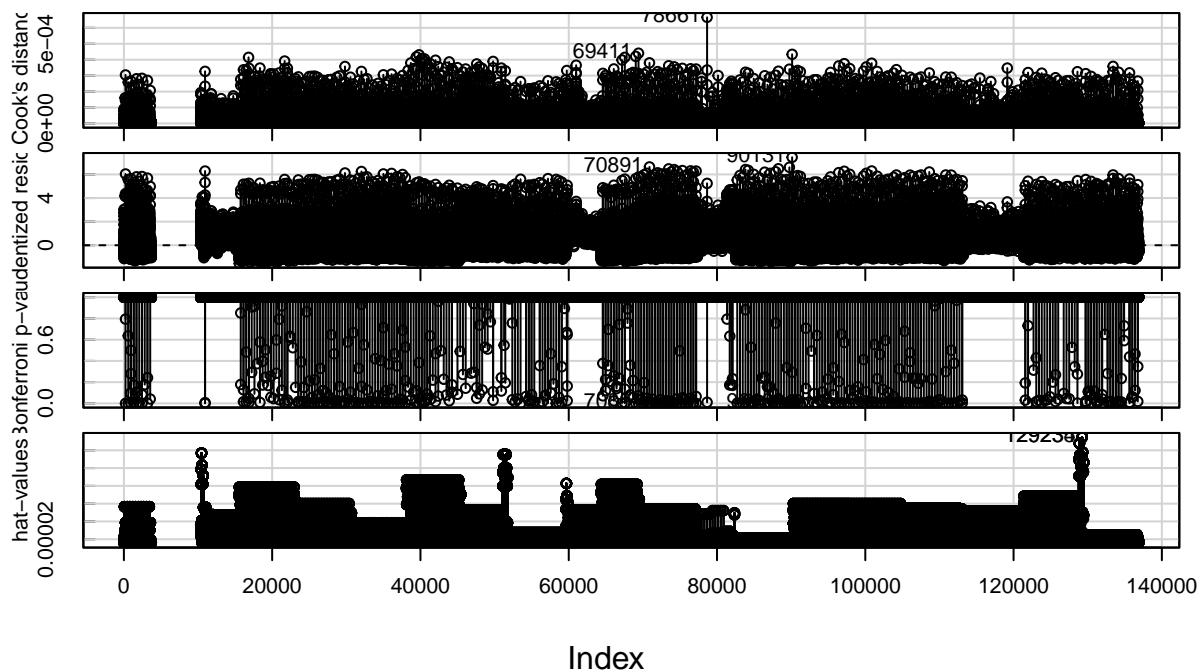


```
ncvTest(testLM) # homoscedasticity test: H0 of constant variance is rejected
```

```
## Non-constant Variance Score Test  
## Variance formula: ~ fitted.values  
## Chisquare = 950.7684, Df = 1, p = < 2.22e-16
```

```
influenceIndexPlot(testLM) # outliers, hard to judge since there seem to be many,
```

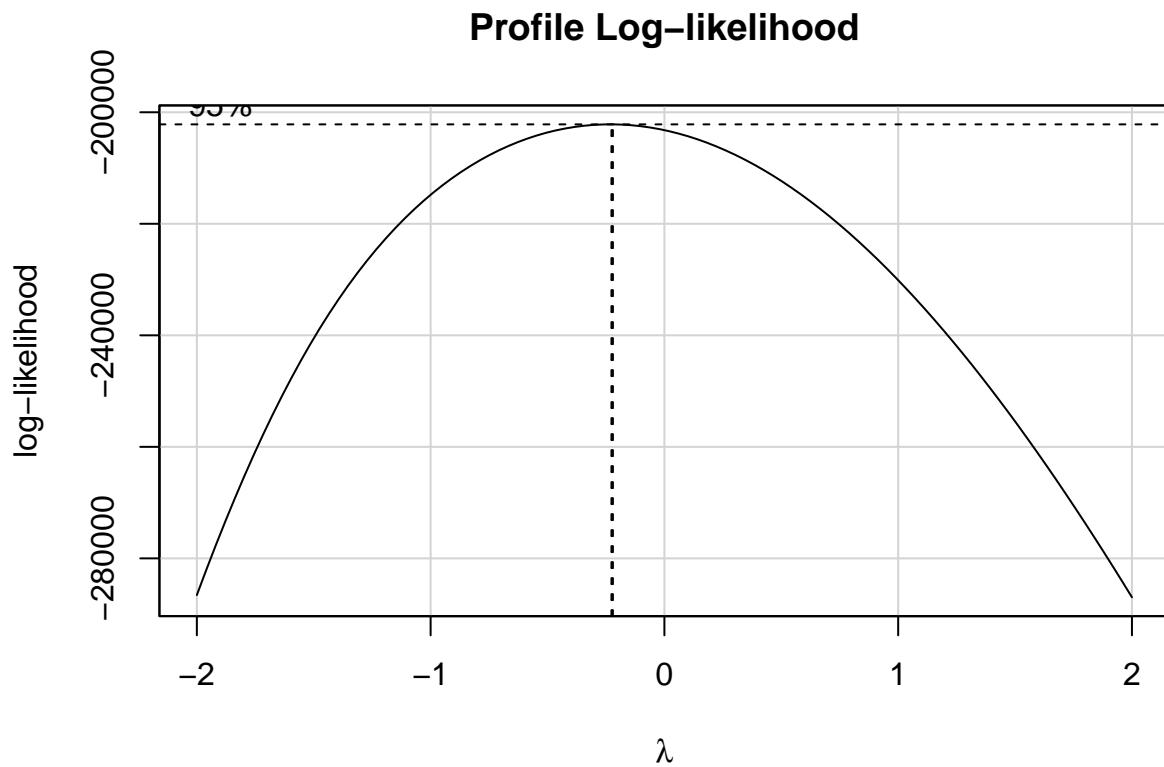
Diagnostic Plots



```
# or none which are an issue
outlierTest(testLM) # lists 10 different points which could be an issue
```

```
##          rstudent unadjusted p-value Bonferroni p
## 90131    7.453189    9.2235e-14   6.0017e-09
## 70891    6.633691    3.2995e-11   2.1469e-06
## 89761    6.603267    4.0528e-11   2.6371e-06
## 88281    6.477534    9.3896e-11   6.1097e-06
## 73481    6.469914    9.8752e-11   6.4257e-06
## 83841    6.414796    1.4198e-10   9.2383e-06
## 74221    6.409933    1.4658e-10   9.5376e-06
## 84211    6.347093    2.2086e-10   1.4371e-05
## 75331    6.325434    2.5416e-10   1.6538e-05
## 76441    6.301214    2.9721e-10   1.9339e-05
```

```
# (all over 3 rstudent and signif.)
boxCox(testLM) # recommended log transformation
```



I decided to do a brief check of observed correlation to compare with the Mantel test results, using the above dataframe;

```
cor.test(dist.summary$gen.dist, dist.summary$geo.dist, method = "pearson") #

##
## Pearson's product-moment correlation
##
## data: dist.summary$gen.dist and dist.summary$geo.dist
## t = 134.54, df = 65067, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4604831 0.4725059
## sample estimates:
##      cor
## 0.466516

# should be taken with a grain of salt since the residual variance is not constant
```

7b. Adjusted Model

I attempted to transform the data (e.g. sqrt, log10) to achieve normal residual distribution however it was not successful so I will build a GLM instead.

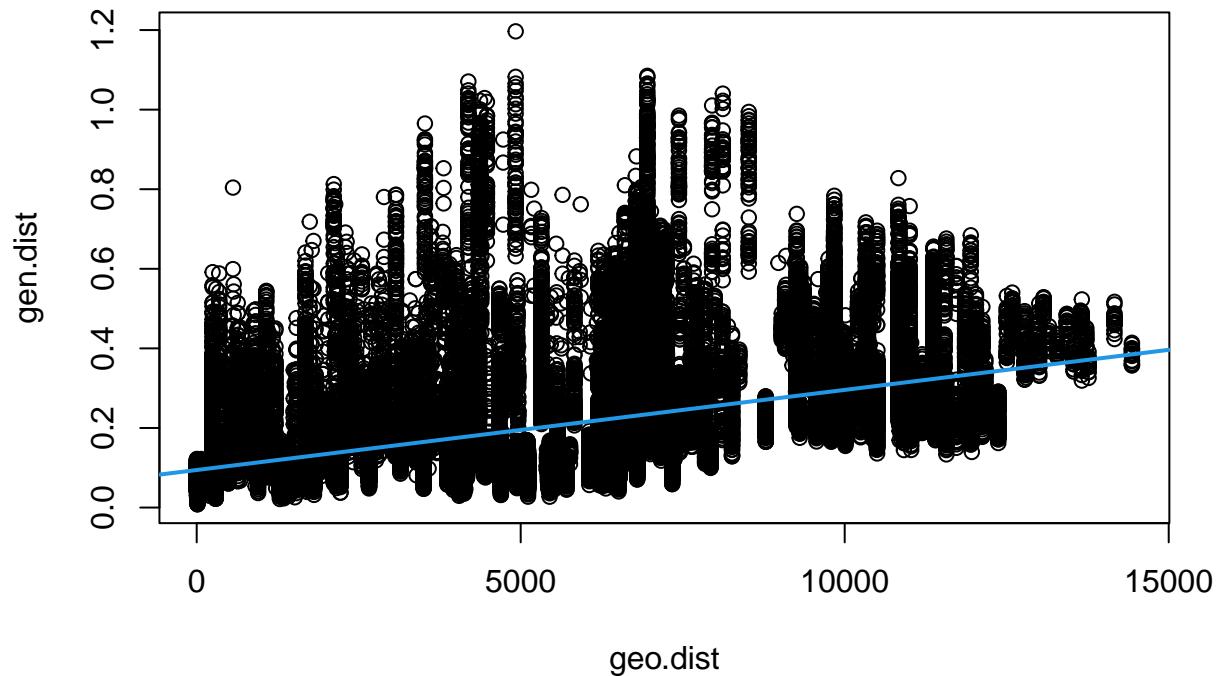
```

GLM <- glm(gen.dist ~ geo.dist, data = dist.summary, family = Gamma(link = "identity"))
# Gamma distribution family used for right-skewed data.
# I tried the link functions "log" and "inverse" first but the resulting coefficients
# were very peculiar (e.g. intercepts at -2 or 7)
summary(GLM) # model results

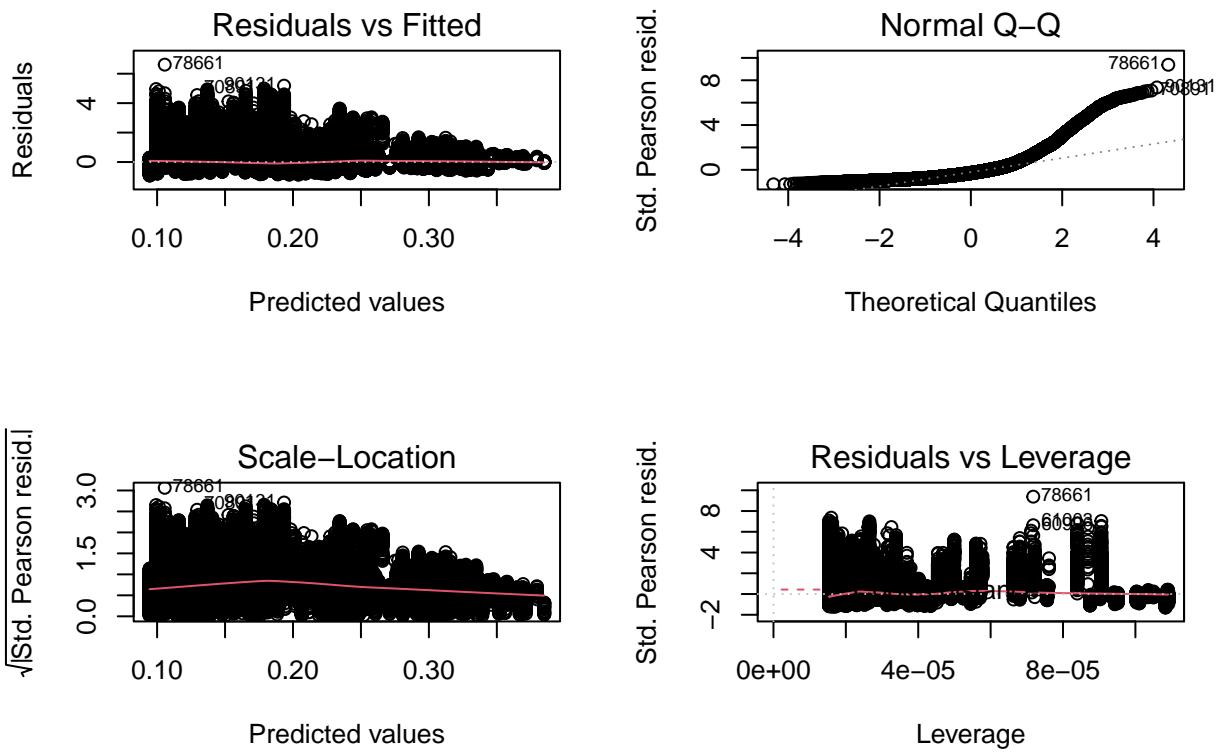
## 
## Call:
## glm(formula = gen.dist ~ geo.dist, family = Gamma(link = "identity"),
##      data = dist.summary)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.7423 -0.5323 -0.2198  0.1474  3.0278
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9.434e-02 6.950e-04 135.7   <2e-16 ***
## geo.dist    2.013e-05 1.621e-07 124.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.4967076)
##
## Null deviance: 30821  on 65068  degrees of freedom
## Residual deviance: 22856  on 65067  degrees of freedom
## AIC: -117995
##
## Number of Fisher Scoring iterations: 5

par(mfrow = c(1, 1))
plot(gen.dist ~ geo.dist, data = dist.summary)
abline(coef = coef(GLM), col = 4, lwd = 2)

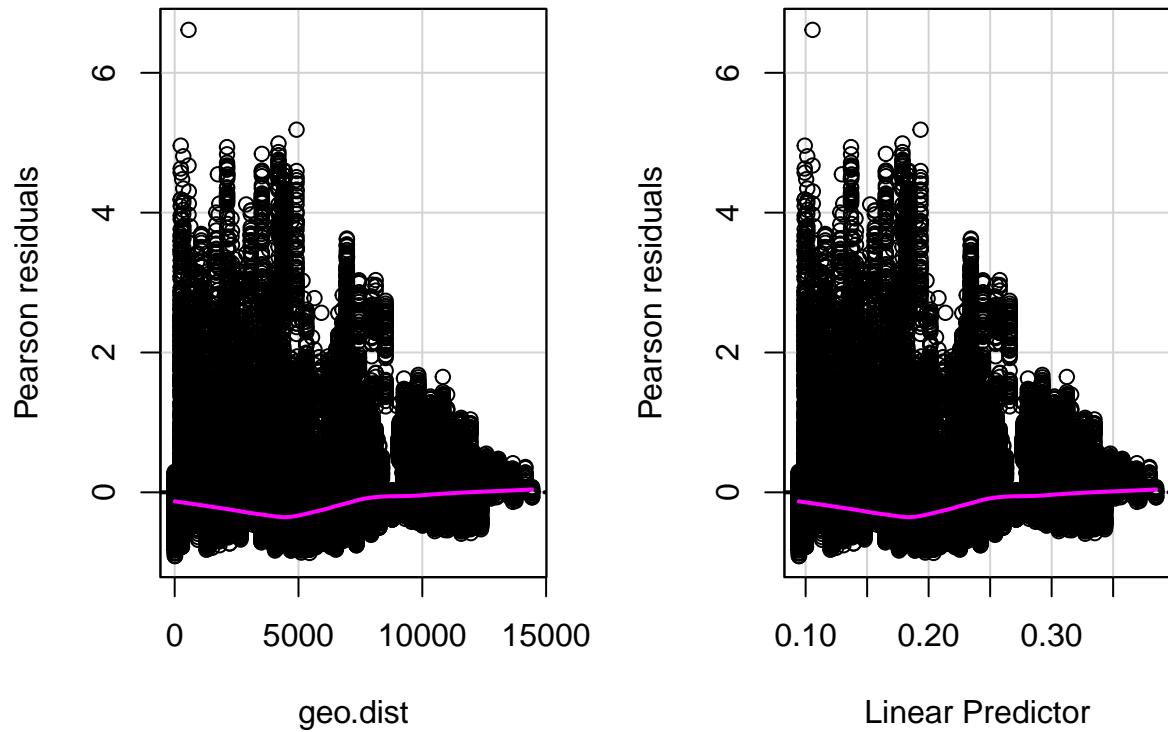
```



```
par(mfrow = c(2, 2)) # changing the number of plots visible at once  
plot(GLM) # diagnostic plots
```



```
residualPlots(GLM)
```

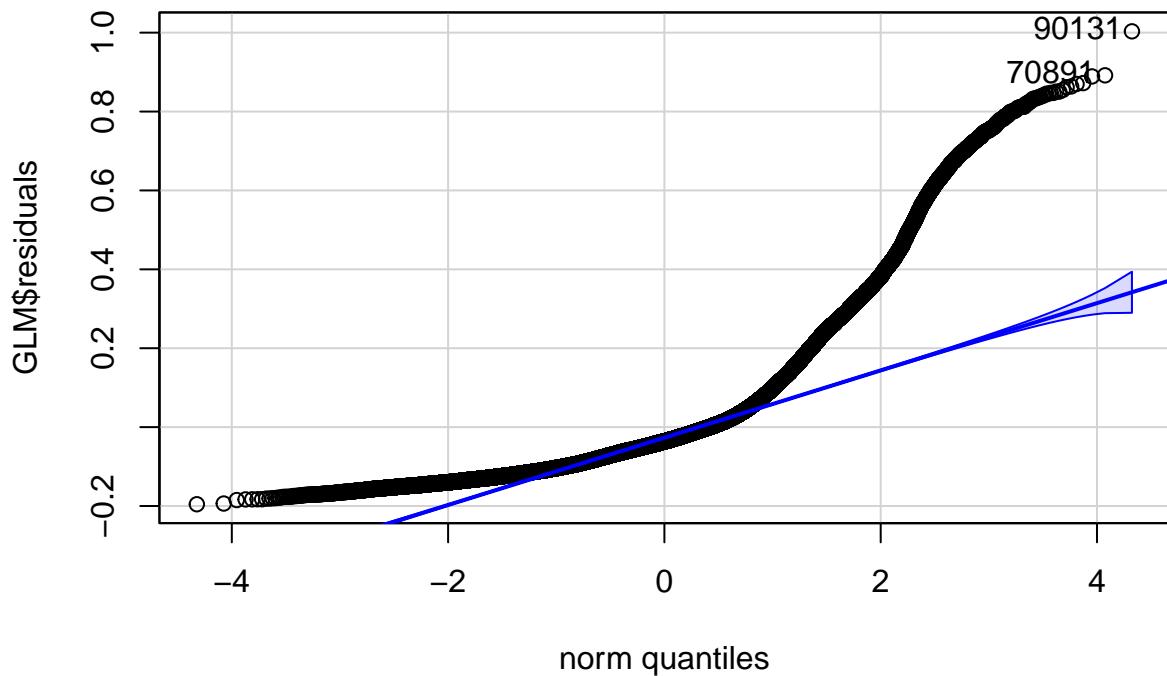


```

##           Test stat Pr(>|Test stat|)
## geo.dist      121.28      < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

par(mfrow = c(1, 1))
qqPlot(GLM$residuals, line = "quartiles") #

```



```

## 90131 70891
## 42506 32209

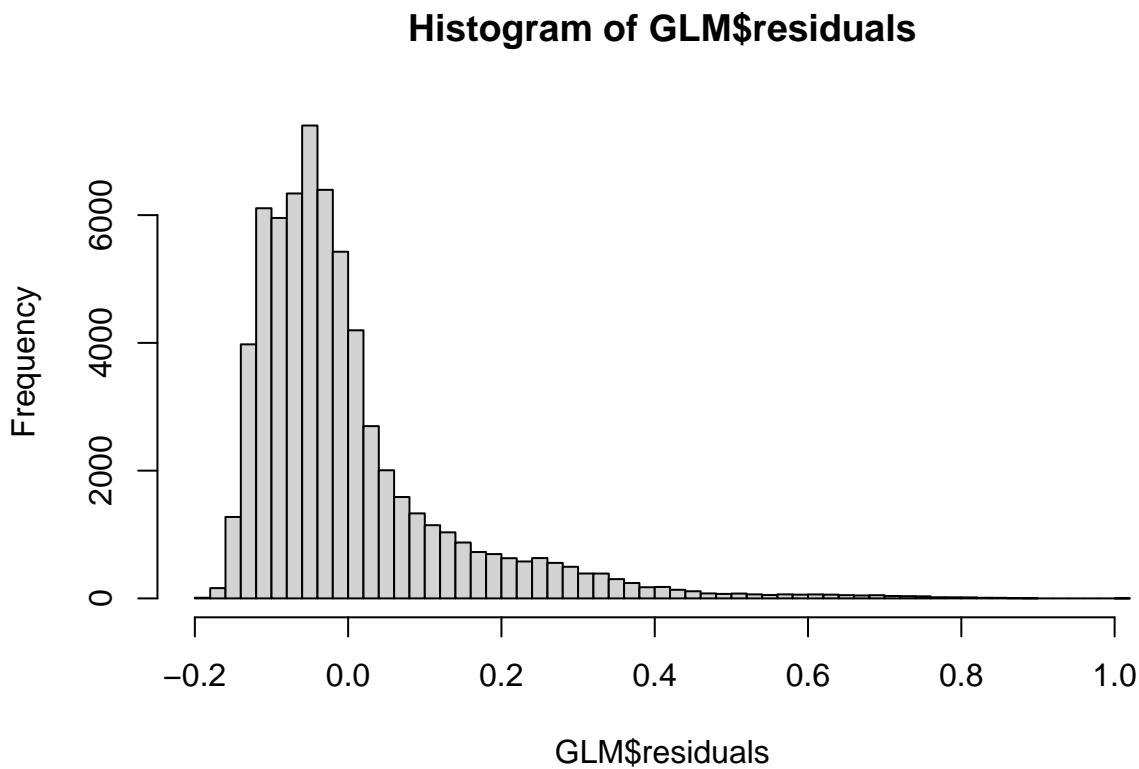
ks.test(GLM$residuals, 'pnorm') #

## Warning in ks.test(GLM$residuals, "pnorm"): ties should not be present for the
## Kolmogorov-Smirnov test

##
## One-sample Kolmogorov-Smirnov test
##
## data: GLM$residuals
## D = 0.43417, p-value < 2.2e-16
## alternative hypothesis: two-sided

```

```
hist(GLM$residuals, breaks = 50)
```



```
with(summary(GLM), 1 - deviance/null.deviance) # R^2 value of 0.2584382.
```

```
## [1] 0.2584382
```

```
# This is not high therefore there may be additional variables to add to  
# the model to explain the variance.
```

8. Looking at the genetically closest island pairs (based on specimens)

```
# top 10 genetically closest islands  
# sorting by geo dist, increasing in size  
dist.summary <- dist.summary[order(dist.summary$geo.dist, decreasing = FALSE),]  
  
# sorting by gen dist,  
dist.summary <- dist.summary[order(dist.summary$gen.dist, decreasing = FALSE),]  
  
x <- dist.summary[!duplicated(dist.summary$islands.combo),]
```

```
x <- x[order(x$gen.dist, decreasing = FALSE),]
kable(head(x, n = 20))
```

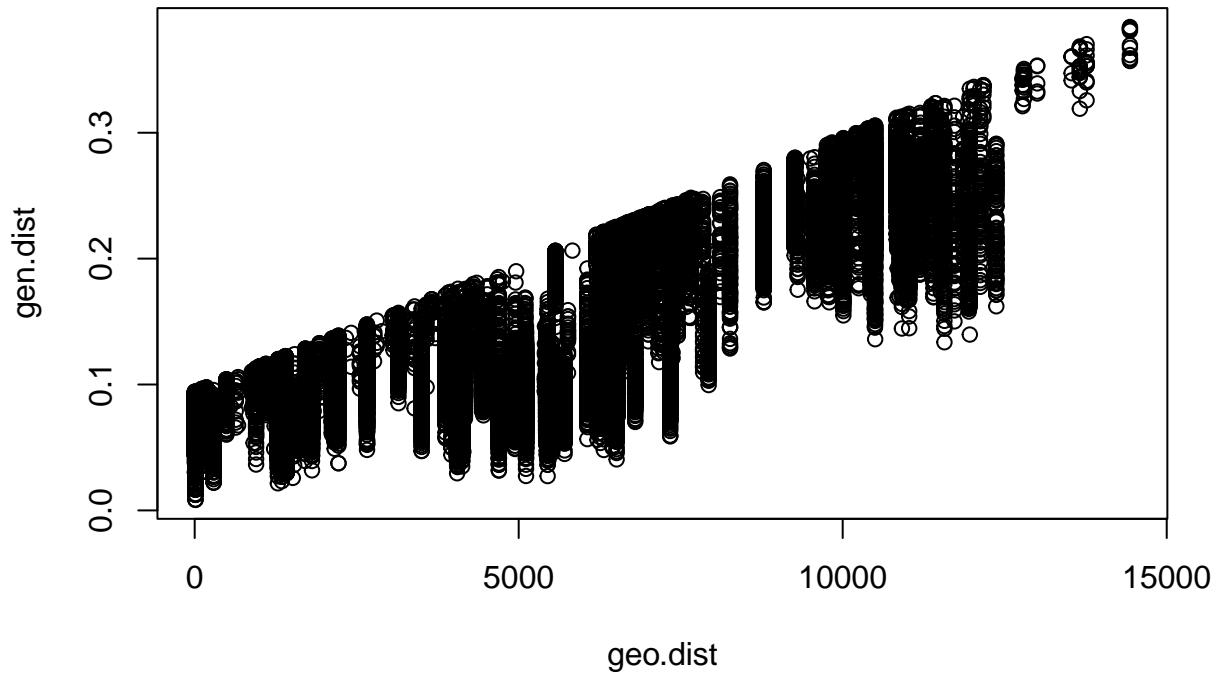
9. Taking a closer look at islands with high distance/close genetics

```
# reloading the df so that the order is reset to original:
dist.summary <- read.csv("./data/RStudio/gen_geo_distance_matrices_df.csv")
# making sure correct fitted values are ready:
GLM <- glm(gen.dist ~ geo.dist, data = dist.summary, family = Gamma(link = "identity"))

fv <- as.vector(GLM$fitted.values)
x <- cbind(dist.summary, fv)

x <- x[x$gen.dist < x$fv,] # keeping only gen.dist values less than fitted values

plot(gen.dist ~ geo.dist, data = x)
```



```
# should be all values up to the regression line
```

```
all.combos <- count(dist.summary$islands.combo) # if doesn't work try $ instead of comma
under.fv.combos <- count(x$islands.combo)
```

```

combo.totals <- merge(all.combos, under.fv.combos, by = "x", all = TRUE)
colnames(combo.totals) <- c("combo", "all", "below.line")
rm(all.combos, under.fv.combos, x)
combo.totals$perc.under <- round(((combo.totals$below.line / combo.totals$all) * 100), 3)
combo.totals[is.na(combo.totals)] <- 0

# perc.under is the percentage of the specimens with genetic distances (between
# the two stated islands) which are less than the average genetic distance/point
# on the regression line at that geographic distance.

kable(combo.totals[combo.totals$perc.under > 99,])

```

	combo	all	below.line	perc.under
3	Aotea:Hatutaa	210	210	100.000
4	Aotea:Honuea	200	200	100.000
5	Aotea:Kaikura	200	200	100.000
6	Aotea:Kamaka	200	200	100.000
13	Aotea:Rakiura	210	210	100.000
14	Aotea:Reiono	210	210	100.000
15	Aotea:Rimatuu	190	190	100.000
17	Aotea:Tahanea	200	200	100.000
18	Aotea:WakeIs	200	200	100.000
19	GrtMercury:Aotea	10	10	100.000
22	GrtMercury:Hatutaa	21	21	100.000
23	GrtMercury:Honuea	20	20	100.000
25	GrtMercury:Kamaka	20	20	100.000
30	GrtMercury:Mohotani	14	14	100.000
31	GrtMercury:Motukawa	21	21	100.000
35	GrtMercury:Rakiura	21	21	100.000
36	GrtMercury:Reiono	21	21	100.000
37	GrtMercury:Rimatuu	19	19	100.000
40	GrtMercury:Tahanea	20	20	100.000
41	GrtMercury:WakeIs	20	20	100.000
47	Hatutaa:Kamaka	420	420	100.000
48	Hatutaa:Kayangel	315	314	99.683
56	Hatutaa:WakeIs	420	420	100.000
61	Honuea:Kayangel	300	300	100.000
67	Honuea:Rakiura	420	420	100.000
70	Honuea:WakeIs	400	400	100.000
73	Kaikura:Hatutaa	420	420	100.000
74	Kaikura:Honuea	400	400	100.000
75	Kaikura:Kamaka	400	400	100.000
82	Kaikura:Rakiura	420	420	100.000
83	Kaikura:Reiono	420	420	100.000
85	Kaikura:Tahanea	400	400	100.000
86	Kaikura:WakeIs	400	400	100.000
97	Kayangel:Kamaka	300	300	100.000
115	LateIs:WakeIs	420	419	99.762
167	Mohotani:Aotea	140	140	100.000
170	Mohotani:Hatutaa	294	294	100.000
172	Mohotani:Kaikura	280	280	100.000
173	Mohotani:Kamaka	280	280	100.000
174	Mohotani:Kayangel	210	210	100.000

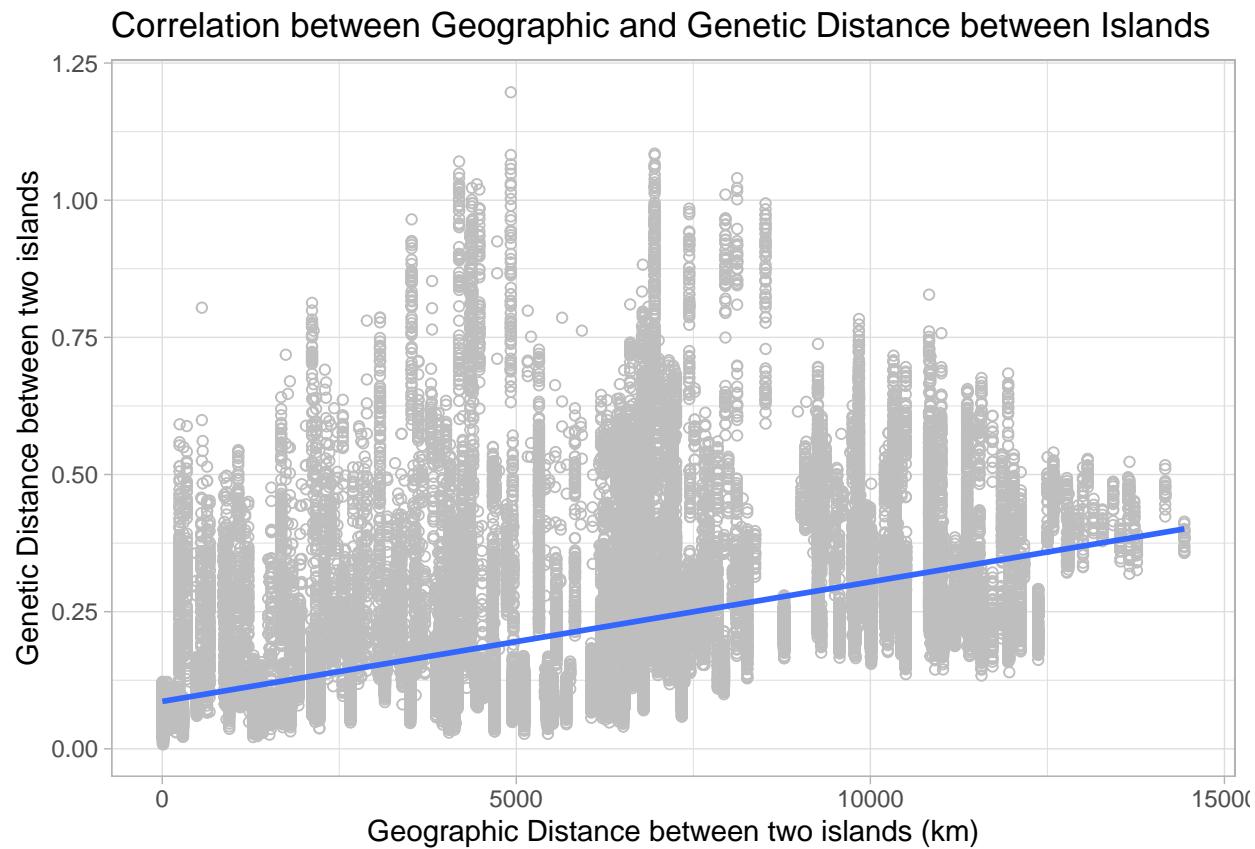
	combo	all	below.line	perc.under
177	Mohotani:Motukawa	294	294	100.000
181	Mohotani:Rakiura	294	294	100.000
184	Mohotani:Slipper	294	294	100.000
187	Mohotani:WakeIs	280	280	100.000
188	Motukawa:Aotea	210	209	99.524
191	Motukawa:Hatutaa	441	441	100.000
192	Motukawa:Honuea	420	420	100.000
194	Motukawa:Kamaka	420	420	100.000
201	Motukawa:Rakiura	441	441	100.000
202	Motukawa:Reiono	441	441	100.000
203	Motukawa:Rimattuu	399	399	100.000
205	Motukawa:Tahanea	420	420	100.000
206	Motukawa:WakeIs	420	420	100.000
224	Rakiura:Hatutaa	441	441	100.000
225	Rakiura:Kamaka	420	420	100.000
233	Rakiura:Tahanea	420	420	100.000
234	Rakiura:WakeIs	420	420	100.000
237	Reiono:Hatutaa	441	439	99.546
239	Reiono:Kamaka	420	420	100.000
240	Reiono:Kayangel	315	313	99.365
246	Reiono:Rakiura	441	441	100.000
249	Reiono:WakeIs	420	420	100.000
252	Rimatuu:Hatutaa	399	399	100.000
254	Rimatuu:Kaikura	380	380	100.000
255	Rimatuu:Kamaka	380	380	100.000
262	Rimatuu:Rakiura	399	399	100.000
266	Rimatuu:WakeIs	380	380	100.000
270	Slipper:Hatutaa	441	441	100.000
271	Slipper:Honuea	420	420	100.000
273	Slipper:Kamaka	420	420	100.000
277	Slipper:Motukawa	441	441	100.000
281	Slipper:Rakiura	441	441	100.000
282	Slipper:Reiono	441	441	100.000
283	Slipper:Rimatuu	399	399	100.000
285	Slipper:Tahanea	420	420	100.000
286	Slipper:WakeIs	420	420	100.000
287	Southland:Aotea	10	10	100.000
289	Southland:GrtMercury	1	1	100.000
291	Southland:Hatutaa	21	21	100.000
292	Southland:Honuea	20	20	100.000
294	Southland:Kamaka	20	20	100.000
300	Southland:Motukawa	14	14	100.000
301	Southland:Rakiura	21	21	100.000
306	Southland:Reiono	21	21	100.000
307	Southland:Rimatuu	19	19	100.000
310	Southland:Tahanea	20	20	100.000
311	Southland:WakeIs	20	20	100.000
322	WakeIs:Kamaka	400	400	100.000
328	WakeIs:Tahanea	400	400	100.000

```
# x <- combo.totals$combo[combo.totals$perc.under == 100]
# x <- as.data.frame(x)
# x <- tidyverse::separate(x, sep = ":", col = x, into = c("isl1", "isl2"))
```

11. Plots

```
ggplot(data = dist.summary, aes(x = geo.dist, y = gen.dist)) +
  geom_point(shape = 1, colour = "grey") +
  geom_smooth(method = "lm") +
  ggtitle("Correlation between Geographic and Genetic Distance between Islands") +
  xlab("Geographic Distance between two islands (km)") +
  ylab("Genetic Distance between two islands") +
  theme_light()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



11. Plotting the coordinates on a map

```

head(names(data), n = 20) # need "geo_lat" "geo_long"

## [1] "island"           "registration.number" "genus"
## [4] "species"          "sex"                  "country"
## [7] "state_province"   "island.1"            "locality"
## [10] "site"              "geo_lat"             "geo_long"
## [13] "collector"         "collecting.date"    "field.number"
## [16] "Populatie"         "X299_CHR1_114679736" "X13_CHR1_116614092"
## [19] "X14_CHR1_124857905" "X15_CHR1_134869867"

longlat <- data[,c(8,11,12)]
longlat <- longlat[!duplicated(longlat$island.1),] # keeping only 1 coordinate for each island
head(longlat) # checking correct columns are used

##      island.1 geo_lat geo_long
## 1      Borneo -0.5102 117.0912
## 19     Sulawesi -1.3252 120.1039
## 41     Malenge -0.2659 122.0439
## 53     Halmahera  1.2660 127.8565
## 65 New Guinea -6.2208 147.3689
## 68 New Britain -5.8145 150.0610

dim(longlat)

## [1] 26  3

rownames(longlat) <- 1:26

longlat$island.1

## [1] "Borneo"           "Sulawesi"
## [3] "Malenge"          "Halmahera"
## [5] "New Guinea"       "New Britain"
## [7] "Normanby Island" "Tahanea"
## [9] "Kamaka"           "Wake Island"
## [11] "Late Island"      "Kayangel"
## [13] "Hatutaa"          "Rakiura (Stewart Isl)"
## [15] "Honuea"           "Reiono"
## [17] "Kaikura Island"  "Rimatuu (Tetiaroa)"
## [19] "Aotea (Great Barrier I)" "Motukawanui"
## [21] "Slipper Island"  "Mohotani"
## [23] "Mainland"         "Great Mercury Island"
## [25] "Luzon"            "Doubtful Sound"

longlat[14,1] <- "Rakiura"
longlat[18,1] <- "Rimatuu"
longlat[19,1] <- "Aotea"

longlat[longlat$geo_long < 0,] # only longitude values up to 180 are plotting, rest

```

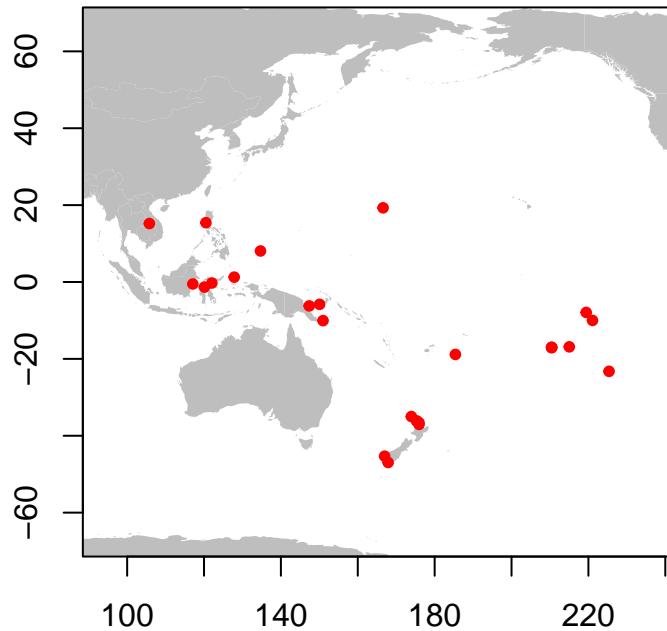
```

##      island.1 geo_lat geo_long
## 8      Tahanea -16.870 -144.970
## 9      Kamaka -23.240 -134.630
## 11 Late Island -18.850 -174.600
## 13     Hatutaa  -7.920 -140.570
## 15     Honuea -17.009 -149.585
## 16     Reiono -17.046 -149.546
## 18     Rimatuu -17.030 -149.558
## 22     Mohotani -10.000 -138.930

# is blank. Can try fixing this by adding 360 to all the negative longitude points
longlat[8,3] <- longlat[8,3] + 360
longlat[9,3] <- longlat[9,3] + 360
longlat[11,3] <- longlat[11,3] + 360
longlat[13,3] <- longlat[13,3] + 360
longlat[15,3] <- longlat[15,3] + 360
longlat[16,3] <- longlat[16,3] + 360
longlat[18,3] <- longlat[18,3] + 360
longlat[22,3] <- longlat[22,3] + 360

map("world2", # "world" over Africa, "world2" over Pacific
    xlim = c(90,240), # x axis limits
    ylim = c(-70,70),
    fill = TRUE,
    col = "grey", # fill colour
    # bg = "azure2", # background colour
    border = NA, # country borders
    wrap = TRUE # tuck map into axes
)
map.axes() # adds axes
points(longlat$geo_long, longlat$geo_lat,
       col = "red", cex = 1, pch = 20)

```



```
# trying with ggplot
ggplot(longlat, aes(x = geo_long, y = geo_lat, label = island.1)) +
  borders("world2", colour = NA, fill = "wheat1") +
  geom_point(color = "red", size = 2) +
  scale_x_continuous(limits = c(100, 240)) +
  scale_y_continuous(limits = c(-50, 35)) +
  theme(panel.background = element_rect(fill = "azure1", colour = "azure1"),
        axis.text = element_blank(),
        axis.title = element_blank(),
        axis.ticks = element_blank()) +
  geom_text_repel(size = 3, point.size = 1, min.segment.length = 0.25) +
  ggtitle("Map of Islands Sampled for Polynesian Rats")
```

Map of Islands Sampled for Polynesian Rats

