Spatio-Temporal Analysis of Sea Surface Temperatures: A Tutorial

J. Scott Berdahl and Marc G. Genton[1]

April 9, 2016

1 Getting Started

West-blowing trade winds in the Indian Ocean push warm surface waters against the eastern coast of Africa. These waters move south along the coastline, eventually spilling out along the boundary of the Indian and Atlantic Oceans. This jet of warm water, known as the Agulhas Current, collides with the cold, west to east flowing Antarctic Circumpolar Current, producing a dynamic series of meanders and eddies as the two waters mix. The result makes for an interesting target for spatio-temporal analysis.

The data_SST.mat file contains sea surface temperature data collected by satellite for the Agulhas and surrounding areas off the coast of South Africa from January 1 to November 26, 2004, a period of 331 days. The main variable, SST.zone.period, is a three dimensional 72 x 240 x 331 matrix (latitude, longitude, day) of sea surface temperatures given in degrees Celsius. Spatial resolution for the data set is roughly 25 kilometers, though exact values depend on latitude. Longitude and latitude values are respectively stored in lon_zone and lat_zone. Temporal resolution is 1 day.

As the data are in Matlab format, they must be converted for analysis in R. This can be done using the function `readMat` from the package *R.matlab*.

2 Exploratory Data Analysis

To familiarize ourselves with the geography of the dataset, we will initially ignore the temporal component of the dataset and examine the spatial distribution of temperatures on a single day.

[1] CEMSE Division, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia. E-mail: james.berdahl@kaust.edu.sa, marc.genton@kaust.edu.sa

Figure 1, generated using the `image.plot` command in the *fields* package, shows the sea surface temperatures recorded on day 10.

Numerous gaps are present in the data, corresponding to three main causes: 1) land: specifically South Africa and Lesotho, visible in the left-center of the top of the plot, as well as two small islands towards the southern boundary; 2) clipping: the two large wedge-shaped voids cutting N-S across the picture resulting from the satellite's orbital path; and 3) cloud cover: all or most of the remaining swirls and dots present in the image. Various forms of Kriging can be used to attempt to fill those gaps caused by orbital clipping and cloud cover. Of course it does not make sense to estimate sea surface temperatures for gaps caused by the presence of land.

A pronounced temperature gradient is visible from highs of over 25°C in the north of the study area to a low of 3.5°C towards the southern boundary. This is not only indicative of spatial correlation in the dataset, but it also shows that the data are not stationary, as the mean temperature must vary strongly with latitude. It also follows that variance has some dependence on latitude, as the plumes and eddies (including the Agulhas Current, mostly obscured here by

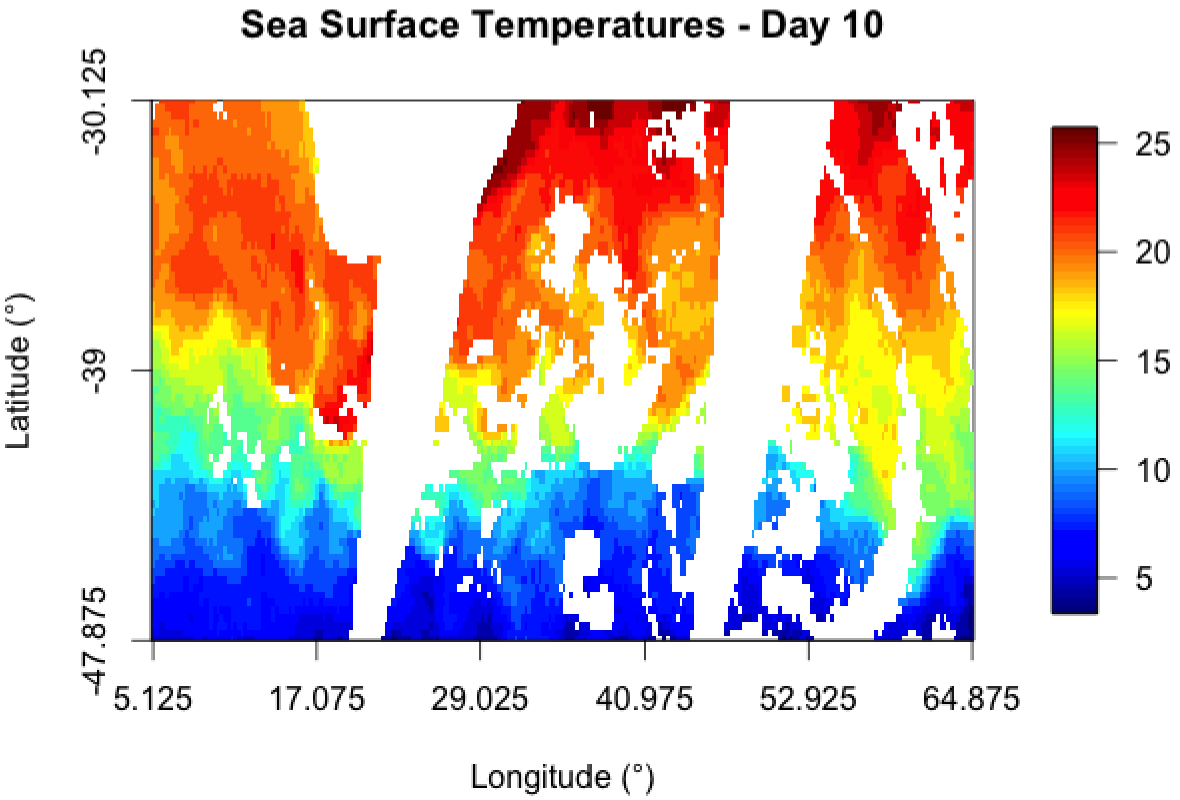

Figure 1: Sea surface temperature data in °C observed on January 10, 2004 (day 10 of the "data_SST.mat" file) in the southern Atlantic and Indian oceans off the coast of South Africa. As a map, the plot uses Mercator projection, wherein lines of latitude and longitude form a regular grid.

satellite clipping) in the transitional zone between warm and cold increase variance there.

The `hist` command in the *graphics* package can be used to generate a histogram of sea surface temperatures for Day 10, as shown in Figure 2. The temperatures have a bimodal distribution, with a warm peak at 20.5°C and a smaller cool peak at 7.5°C, and fewer values in between. The peaks correspond to the large regions of warm and cool waters at the north and south of the study area, whereas the trough is a reflection of the relatively sharp transition from warm to cold in the zone of mixing.

Figure 3 illustrates the non-stationarity of the sea surface temperatures, which can be seen to be primarily a function of latitude. As latitudes approach the equator, the mean temperature rises. Variances, on the other hand, are highest in the zone of mixing between -44N and -38N. Both mean and variance are relatively constant across longitudes, with large deviations on both plots caused by the missing N-S bands of data. Variances by longitude are an order of magnitude higher because of the N-S temperature gradient. Note that the plots of mean and variance across longitudes are rougher than those across latitudes; this could simply be due to the fact that with only 72 or fewer samples at any given longitude vs. 240 at any latitude, there is more room for random variations to affect the longitudinal values.

Exercise 1: Load the SST data into R and reproduce the exploratory data analysis for Day 10. Do the same for another day of the dataset.

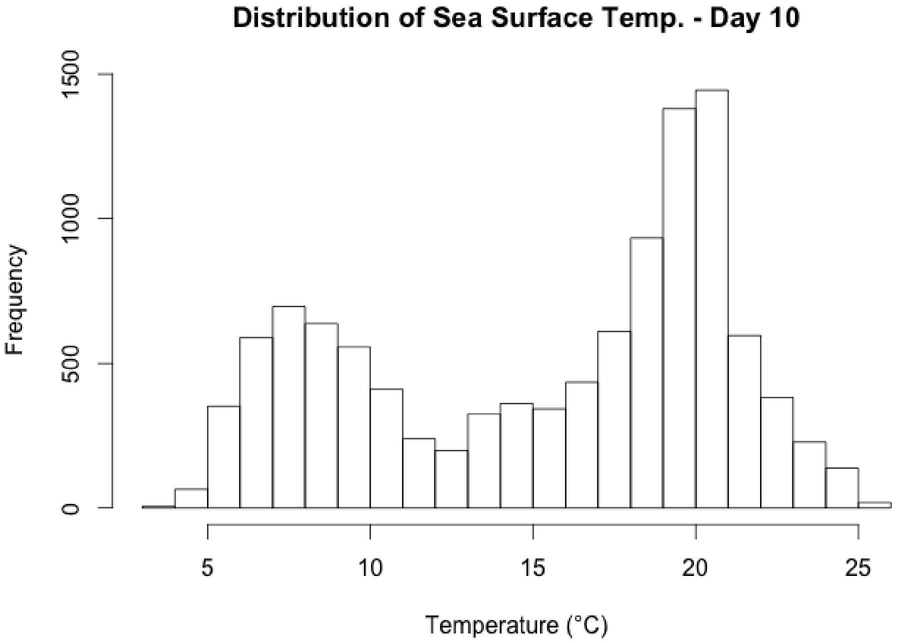

Figure 2: Histogram of sea surface temperature data in °C observed on January 10, 2004.

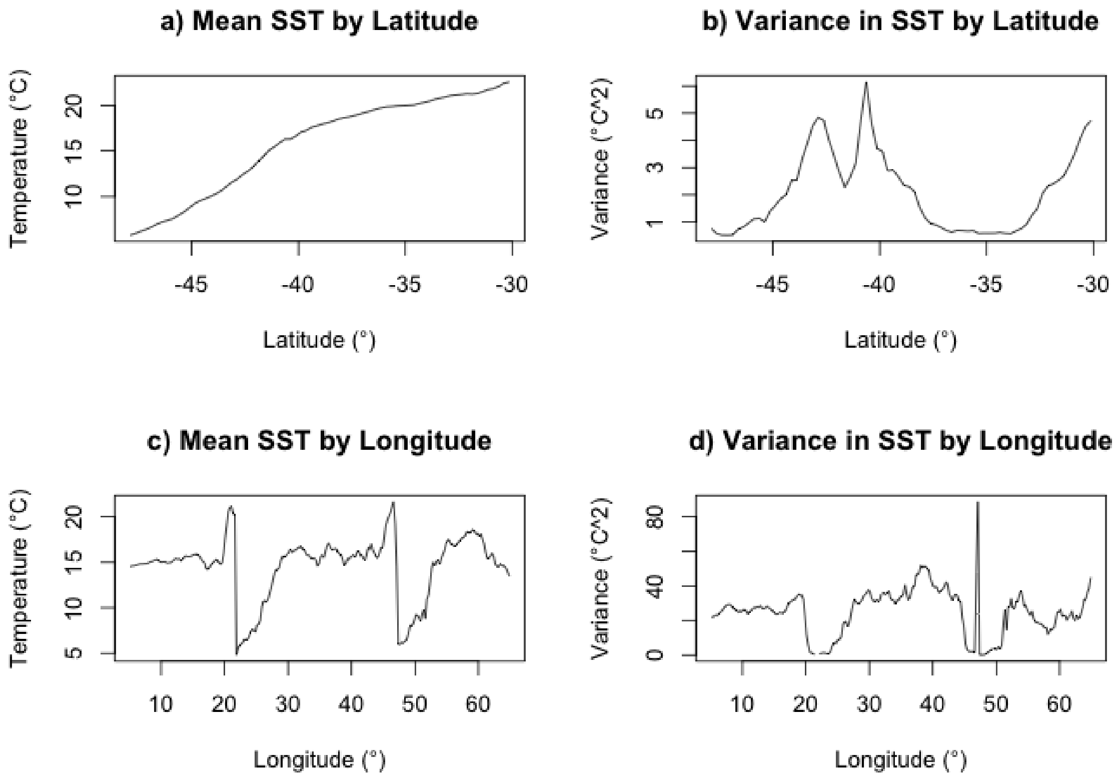

Figure 3: a), b) Mean and variance of sea surface temperature across all longitudes for each latitude value. c), d) Mean and variance of sea surface temperature across all latitudes for each longitude value.

3 Spatial Data Analysis

Note: For simplicity, in variogram modeling and kriging throughout this tutorial, we will treat the latitude and longitude coordinates as if they are Cartesian. Given the limited domain of the SST data, the biggest effect this will have on the models is not due to the differences between arc length and Cartesian distances, nor between global and Cartesian angles between points, but rather potential inaccuracies due to the E-W extension of the domain brought about by considering spacings between longitudes as equal to spacings between latitudes, when in fact at 30°- 47°S, longitudinal spacing is roughly 70 to 90% of latitudinal spacings (which are constant at all points on the globe). In a more rigorous treatment of the data, lat/long coordinates should be accounted for.

We can confirm that the data are non-stationary by plotting directional variograms. Using the `variog` command in the package *geoR* will produce a plot similar to Figure 4. As can be seen in the figure, sea surface temperatures are distinctly non-stationary in the N-S direction. The semivariogram for these relationships approximates a linear trend and does not reach a sill.

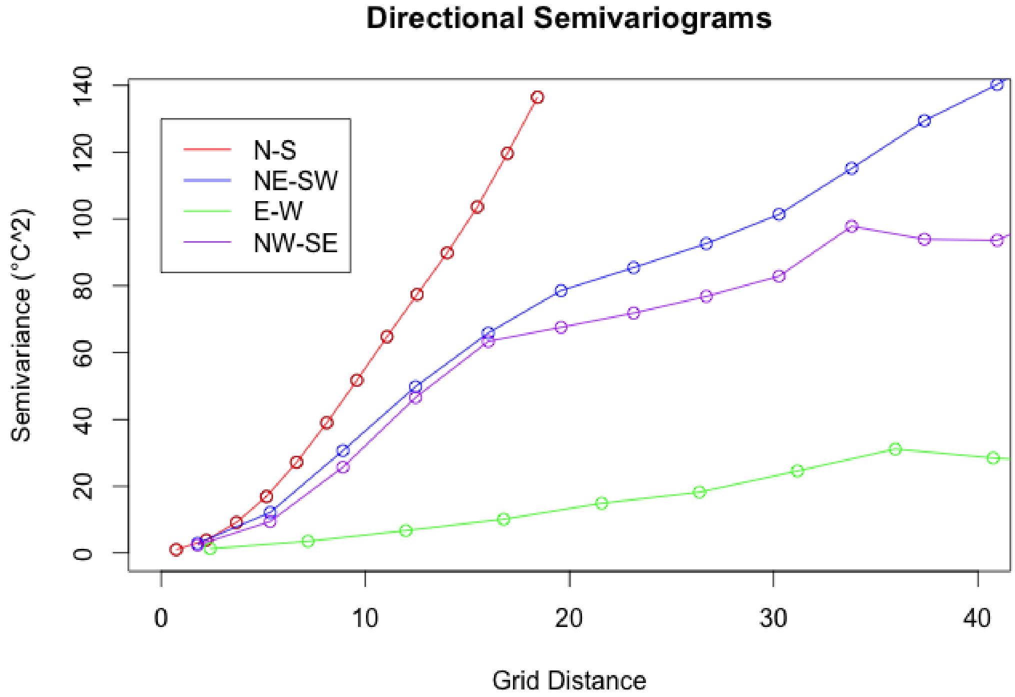

Figure 4: Directional semivariances as observed on January 10, 2004 in the sea surface temperature dataset.

NW-SE and NE-SW variances behave similarly, though the effect is not as pronounced and the range is increased. Their symmetry with one another highlights the overall N-S direction of sea surface temperature non-stationarity. Only the E-W variances form something like a stationary variogram, reaching a rough sill of around 30 $°\mathrm{C}^2$.

Our ultimate goal is to predict the missing values present in the Day 10 data, and thus, once we find an appropriate variogram, we will use kriging to make these estimates. Universal kriging models the data as a function of a spatially-dependent mean plus some Gaussian “error”:

$$Z(\mathbf{s}) = \mathbf{m}(\mathbf{s})^T\boldsymbol{\beta} + \epsilon(\mathbf{s})$$

As we saw in Figure 3a) there is a clear trend between mean temperature and latitude. Instead of modeling this trend, since we have a mean temperature value for each latitude, we can simply subtract these latitudinal means from the Day 10 data before generating the semivariogram. Thus, if the residuals are stationary, we can simply perform ordinary kriging.

A plot of the residuals is shown in Figure 5a). There are still regions of high and low temperatures and evidence of structure, particularly the Agulhas Current, which is bisected by a gap of missing satellite data. But as seen in Figure 5b), the residuals have a nearly Gaussian distribution (with mean = 0 and var = 1.96).

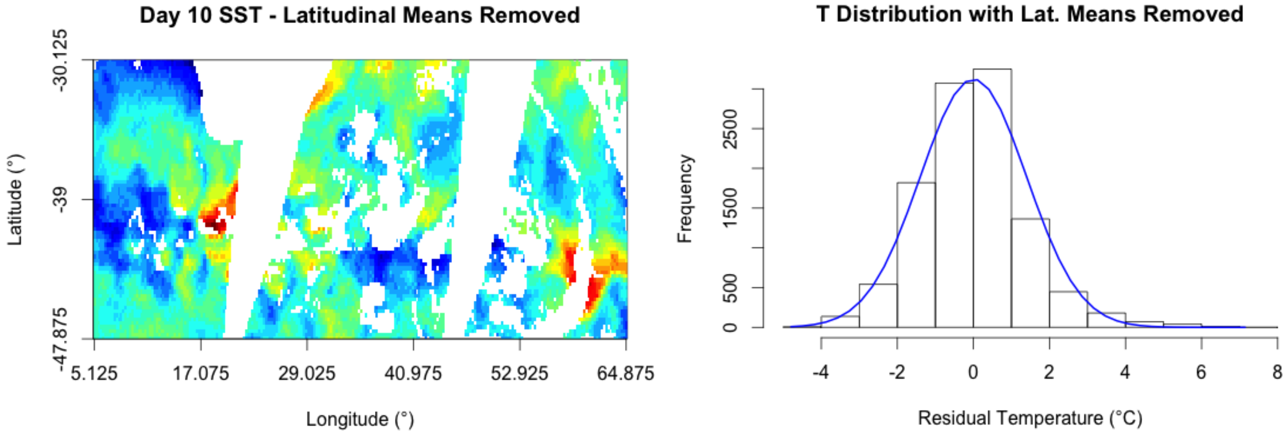

Figure 5: Directional semivariances as observed on January 10, 2004 in the sea surface temperature dataset.

Once again using the `variog` command in the *geoR* package, we can plot empirical directional variograms for the residual data, as shown in Figure 6a). While the removal of means does not create a perfect anisotropic variogram, the sills and ranges are much closer than before and importantly, they are much closer to stationary. The N-S direction still has a smaller range than E-W, and thus N-S directions should not be as closely correlated with nearby neighbors as E-W directions (though this may be corrected somewhat by the lat/lon spacing discrepancy discussed earlier). It is also worth noting that the NE-SW direction has a lower sill than other directions, suggesting some correlation in that direction.

In any case, the directional variograms of the residual will be much better represented by an omnivariogram for ordinary kriging that would the directional variograms of the initial data. Figure 6b) shows a Matérn variance function (red) fitted to a distance of 50 grid spaces in the data matrix, using the function `variofit` in *geoR*. The choice of nugget in this case should correspond to the variance at 1 grid position's difference, as no smaller distances will be considered in kriging for missing values.

Note that as the dataset is quite large, even for just the Day 10 data, depending on computational resources it can be a good idea to fit the semivariance function to a random subset of points. For the Matérn function in Figure 6b) and the kriging estimates shown in Figure 7 a subset of 500 data points selected randomly from the Day 10 data was used.

With an omnivariogram model established, we can employ kriging to estimate temperature values at missing locations. As it does not make sense to estimate sea surface temperatures where land is present, these values do not need to be estimated. (One can establish which locations are land by looking at the rest of the dataset; clouds and satellite gaps move, but land is always

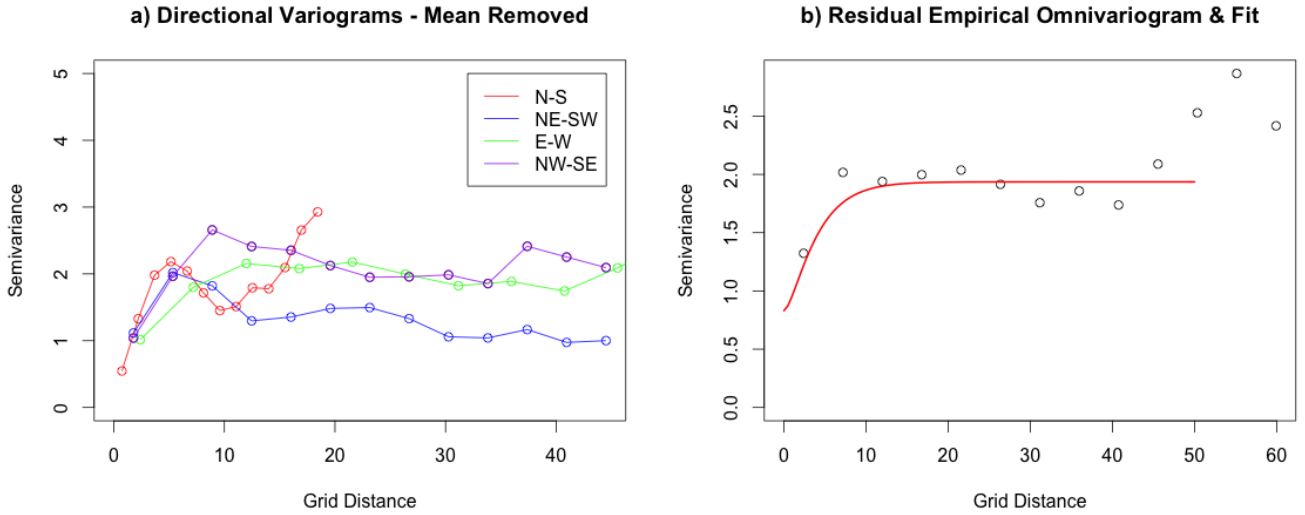

Figure 6: a) Directional semivariograms for January 10, 2004 sea surface temperature with latitudinal means removed. b) Omnivariogram for residual temperatures after latitudinal means removed and Matérn fit.

land. Thus wherever the average across 331 days returns a null value, it is very likely that that location represents land.)

Using the `krige.conv` function, also in the package *geoR*, one can estimate the residuals for all locations missing sea surface temperatures on Day 10. Once this calculation is performed, adding back in the latitudinal means gives a complete estimate, as shown in Figure 7.

Unfortunately, it is easy to see that the estimates produced in Figure 7 are unlikely to reflect reality. The Agulhas Current, for instance, does not cross the orbital gap as it might be expected to, connecting the warmest waters off the east coast of Africa with the plume of warm water off Africa's southern tip. Instead the gap is filled in with a broad average of regional values, plus the latitudinal means. Other large gaps are filled in similarly. Small details, or estimates thereof, cannot be accurately recovered in this case when only spatial data is used to make an estimate.

Exercise 2: Reproduce the spatial data analysis for Day 10. Do the same for another day of the SST dataset.

4 Spatio-Temporal Analysis

4.1 Exploratory Temporal Data Analysis

A quick comparison of mean sea surface temperatures across the entire study area on each day (Figure 8) shows a clear seasonal curve, with temperatures highest around Day 60 in early March,

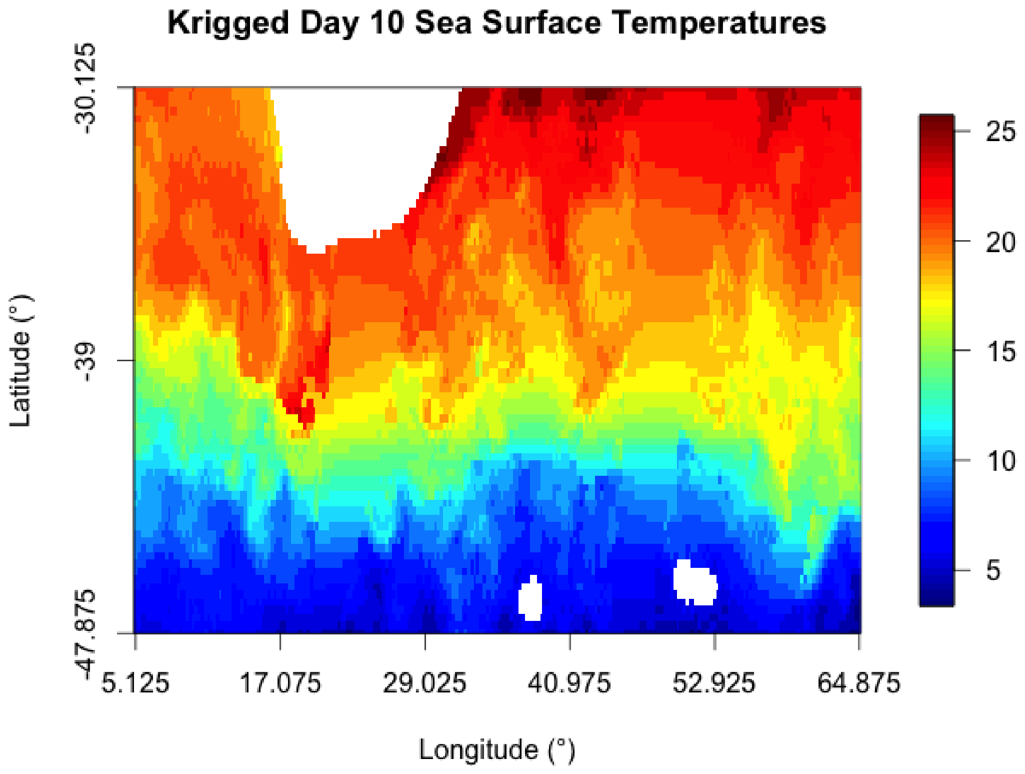

Figure 7: Sea surface temperature data in °C for January 10, 2004, with missing values caused by cloud cover and satellite gaps estimated through spatial kriging.

and lowest around Day 240 in early September. This makes sense given the timing of the southern hemisphere seasons, plus some lag.

Obvious outliers, with average temperatures around 22°C, occur just prior to Day 50. These could be from days with abnormally prevalent southern cloud cover masking cool southern ocean temperatures, or they may result from some error in the data collection. By checking these dates using the `image` command, we can see that the latter is the case - values in the southern regions are missing completely. Thus these outliers do not reflect the true nature of the dataset, and should be removed for any relevant model fitting or analysis (or one should use robust statistical procedures).

Some broad behavior can be deduced from the way that the distributions of means in Figure 9a) deviates from a simple N-S gradient. For example, the Agulhas Current can clearly be seen extending SW from eastern Africa, and prominent eddy locations can be seen extending east from Agulhas. This in itself divulges the direction of the Antarctic Circumpolar Current: west to east, as the Agulhas waters are pushed eastward when they move south. Similarly, cooler waters on the eastern edge of the figure hint at northward flow and thus the counter-clockwise

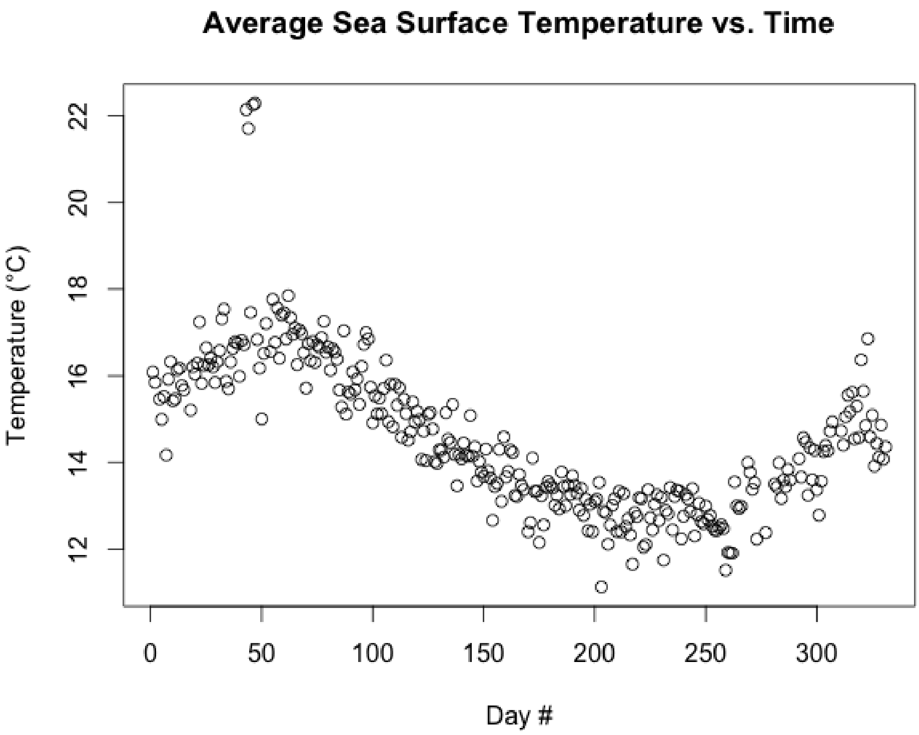

Figure 8: Average sea surface temperature across the entire survey area as recorded on each of the 331 days from January 1, 2004 through November 26, 2004.

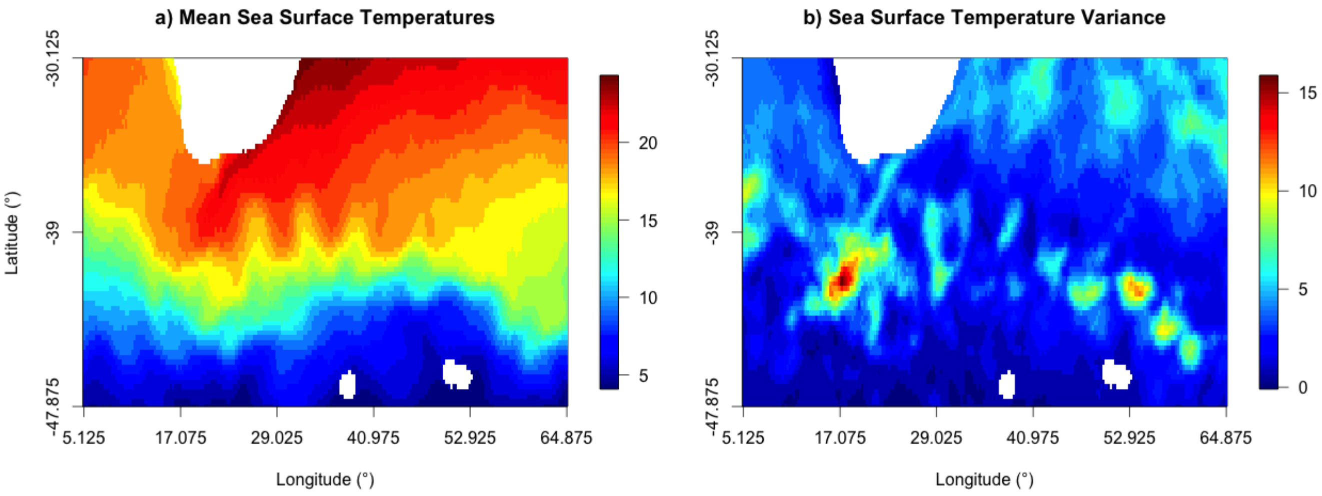

Figure 9: a) Mean and b) variance across the full 331 day period for each location.

direction of surface water circulation in the southern Indian Ocean - a large-scale feature known as the Indian Ocean Gyre.

Additional useful information can be gleaned from examination of the variance in temperatures recorded at each point throughout the 331 day period, as shown in Figure 9b). This map shows that the Agulhas Current itself, despite being a dynamic feature, is fairly steady, forming only a light cyan line (variance $\approx 6(°C)^2$) extending SW from South Africa. More strikingly, however, it shows zones where temperatures can fluctuate most dramatically due to eddies mixing warm and cold waters, particularly in the zone from 39°S to 44°S fed by the Agulhas Current. To the east of this, once again, the path of the eddy train can be seen as a series of zones of high variance. Note that these zones occur on the edges of the warm eddies revealed in the temporal mean plot. In the southeast corner of the domain, the eddy train may explain why the Day 10 temperatures in that region are unusually high (Figures 1 and 7); they are remnants of an influx of warm surface waters from the Agulhas Current transported east and south by the Antarctic Circumpolar Current.

Exercise 3: Reproduce the exploratory temporal data analysis of the SST dataset.

4.2 Spatio-Temporal Kriging

In Section 3, we used spatial data alone to estimate missing sea surface temperatures for Day 10. Of course, it is possible to get a better kriging estimate of sea surface temperatures by considering the temporal correlations between temperatures observed at a given location at different times - assuming that the data are temporally correlated. As can be seen in Figure 10, which compares temperatures on two subsequent days and on two days separated by six months, a clear temporal correlation exists. Thus it makes sense to include observations from additional days in making our estimates of the missing values on Day 10.

However, though the data are temporally correlated, temporal variograms are not necessarily spatially stationary across the survey area. In other words, the way temperatures vary in time at one location may not be the same as how they vary in time at another location. Figure 11, which plots empirical variograms for a central and a northern point, shows that this is the case. Fortunately, the variograms are reasonably well-behaved within periods of 50 days, and thus if longer time scales are ignored or down-weighted, the additional data provided by incorporating surrounding dates into our estimates will prove useful in reconstructing the missing data on Day 10.

Exercise 4: Reproduce the spatio-temporal analysis above of the SST dataset. Explore a spatio-temporal covariance model fit and kriging.

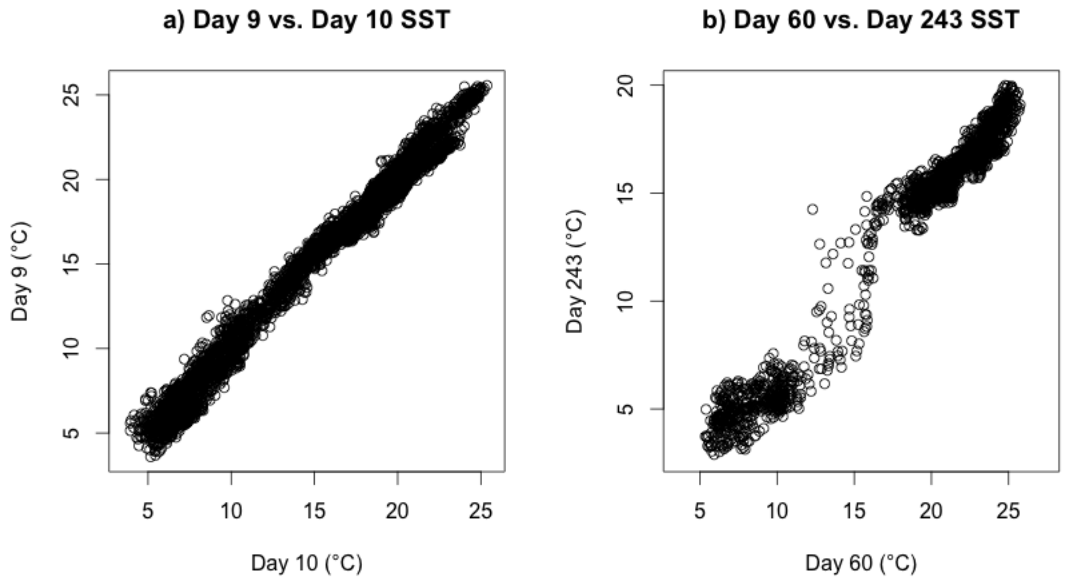

Figure 10: The scatterplots above compare temperatures at each single site temporally spaced by steps of a) 1 day and b)183 days (0.5 years). January 10, 2004 sea surface temperatures closely resemble those of January 9, 2004, as do other relationships from day to day. Even comparing two observation sets 6 months apart, from the maximum average temperatures observed on February 29, 2004 (Day 60) to the minimum temperatures on August 30th, 2004 (Day 243), a clear correlation is present, though the data are notably less correlated in the zone of mixing, where moderate temperatures exist.

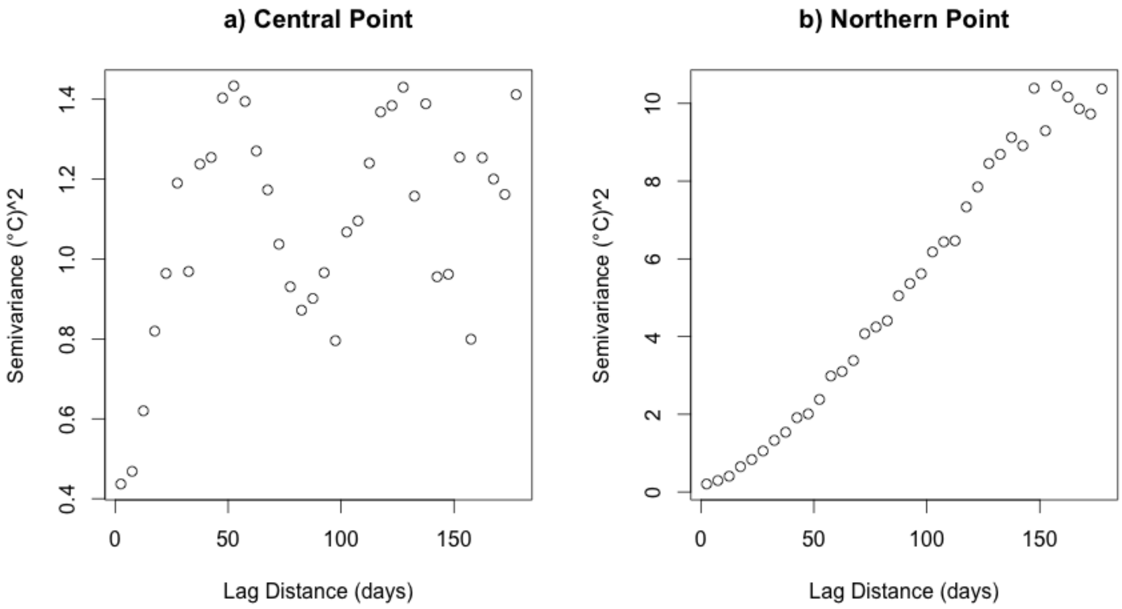

Figure 11: Temporal semivariograms over six months at two different locations in the study area, demonstrating spatial non-stationarity of temporal variograms. a) At 38.875°S, 34.875°E, in the center of the survey area, the semivariogram shows a potentially periodic signal, whereas b) at 32.375°S, 49.875°E, towards the northern boundary of the survey area, semivariance rises through to a peak at around 150+ days.

R Code

Loading the SST data

```
library(R.matlab)
```

Use `readMat` to locate the SST data in your working folder:

```
data.SST <- readMat("your file path", maxLength=NULL, fixNames=TRUE)
```

Exercise 1

Figure 1 - Plot Day 10 Data:

First, some initial setup:

```
library(fields)
lat <- (data.SST$lat.zone)
lon <- (data.SST$lon.zone)
```

Next, isolate Day 10:

```
SST.10 <- (data.SST$SST.zone.period[1:72, 1:240, 10])
```

Now plot Day 10 (with a legend):

```
   par(mar=c(5,5,5,5))
   image(t(SST.10), col=tim.colors(25), xaxt="n", yaxt="n", ylim=c(1, 0),
main="Sea Surface Temperatures - Day 10", xlab="Longitude (°)", ylab="Latitude (°)")
   axis(1, at = seq(0, 1, by = 1/5), labels=seq(min(lon), max(lon),
(max(lon)-min(lon))/5))
   axis(2, at = seq(0, 1, by = 1/2), labels=rev(seq(min(lat), max(lat),
(max(lat)-min(lat))/2)))
   image.plot(SST.10, legend.only=T)
```

Figure 2 - Day 10 Histogram

Use the `hist` command:

```
   hist(SST.10, sqrt(240*2), main="Distribution of Sea Surface Temp.  - Day 10",
xlab="Temperature (°C)")
```

Figure 3 - Mean and Variance by Latitude and Longitude:

To put all four figures in one plot, one can use the par command:

```
   library(graphics)
   par(mfrow=c(2,2))
```

Mean vs. Latitude:

```
   latMean = rowMeans(SST.10, na.rm=TRUE)
   plot(lat, latMean, type="n", main="Mean SST by Latitude", ylab="Temperature (°C)",
xlab="Latitude (°)")
   lines(lat, latMean)
```

Variance vs. Latitude:

```
   latVar <- NULL
   for (i in 1:72) {
   latVar[i] <- var(SST.10[i,1:240], na.rm=TRUE)
   }
   plot(lat, latVar, type="n", main="Variance in SST by Latitude", ylab="Variance
(°C^2)", xlab="Latitude (°)")
   lines(lat, latVar)
```

Mean vs. Longitude:

```
   lonMean = colMeans(SST.10, na.rm=TRUE)
   plot(lon, lonMean, type="n", main="Mean SST by Longitude", ylab="Temperature (°C)",
xlab="Longitude (°)")
   lines(lon, lonMean)
```

Variance vs. Longitude:

```
   lonVar <- NULL
   for (i in 1:240) {
   lonVar[i] <- var(SST.10[1:72,i], na.rm=TRUE)
   }
   plot(lon, lonVar, type="n", main="Variance in SST by Longitude", ylab="Variance
(°C^2)", xlab="Longitude (°)")
```

```
   lines(lon, lonVar)
```

Exercise 2

Figure 4 - Directional Semivariograms

To compute the semivariograms, we need to associate a position to each temperature point in such a way that the variog command in the *geoR* package can use the information. Assuming that latitude and longitude form a Cartesian grid:

```
   library(geoR)
   x <- rep(lon,each=72)
   y <- rep(lat,times=240)
   z <- c(SST.10)
   XYZ <- cbind(x,y,z)
   Geo1 <- as.geodata(XYZ, coords.col=1:2, data.col=3)
```

Use the command variog to find directional semivariograms at 0, 45, 90 and 135 degrees.

```
   Var0 <- variog(Geo1, direction=0)
   Var45 <- variog(Geo1, direction=0.78539816339)
   Var90 <- variog(Geo1, direction=1.57079632679)
   Var135 <- variog(Geo1, direction=2.35619449019)
```

Plot results:

```
   plot(Var0, main="Directional Semivariograms", xlab="Grid Distance",
ylab="Semivariance (°C^2)", xlim=c(0,40))
   lines(Var0, col="red")
   lines(Var45, col="blue")
   lines(Var90, col="green")
   lines(Var135, col="purple")
   legend(0,130, c("N-S","NE-SW","E-W","NW-SE" ), lty=c(1,1,1,1),
lwd=c(1,1,1,1),col=c("red","blue","green","purple"))
```

The command variog4, also in *geo.R*, can be used to a similar effect.

Figure 5 a) - Removing Latitudinal Means

The process for removing each latitudinal mean directly is quite straightforward:

```
  M <- rep(latMean,times=240)
  z.M <- z-M
  XYZM <- cbind(x,y,z.M)
  SST.10.M <- matrix(z.M, nrow = 72, ncol = 240)
```

Plot results:

```
  par(mar=c(5,5,5,5))
  image(t(SST.10.M), col=tim.colors(25), xaxt="n", yaxt="n", ylim=c(1, 0), main="Sea
Surface Temperatures - Day 10", xlab="Longitude (°)", ylab="Latitude (°)")
  axis(1, at = seq(0, 1, by = 1/5), labels=seq(min(lon), max(lon),
  (max(lon)-min(lon))/5))
  axis(2, at = seq(0, 1, by = 1/2), labels=rev(seq(min(lat), max(lat),
  (max(lat)-min(lat))/2)))
  image.plot(SST.10, legend.only=T)
```

Figure 5 b) - Histogram with Gaussian Curve

Histogram (as in Figure 2):

```
  h <- hist(z.M, main = "T Distribution with Lat.  Means Removed", xlab = "Residual
Temperature (°C)")
```

Normal curve overlay:

```
  xfit<-seq(min(z.M, na.rm=TRUE),max(z.M, na.rm=TRUE),length=40)
  yfit<-dnorm(xfit,mean=mean(z.M, na.rm=TRUE),sd=sd(z.M, na.rm=TRUE))
  yfit <- yfit*diff(h$mids[1:2])*(length(z.M)-6335)
  lines(xfit, yfit, col="blue", lwd=2)
```

Figure 6 a) - Directional Semivariograms with Means Removed

Use either method mentioned in the code for Figure 4 (`variog` or `variog4`) to produce the direction semivariograms for the residuals. Don't forget to use the command `as.geodata` on the residual temperatures (`XYZM`) in order to use them with *geoR*.

Figure 6 b) - Omnivariogram with Means Removed

Using a random subsample:

```
   Geo.M <- as.geodata(XYZM, coords.col=1:2, data.col=3)
   Var.M <- variog(Geo.M)
   Geo.M.sub <- sample.geodata(Geo.M, 500)
   Var.M.sub <- variog(Geo.M.sub)
```

Fit empirical semivariogram with Matern model:

```
   fitVAR <- variofit(Var.M.sub, ini=c(0.5,1), cov.model="matern", fix.nugget = FALSE,
nugget = 1, fix.kappa = FALSE, kappa = 0.5, max.dist=50)
```

Plot results:

```
   plot(Var.M, main="b) Residual Empirical Omnivariogram & Fit", ylab=
"Semivariance", xlab="Grid Distance")
   lines(fitVAR, col="red", lwd=2)
```

Figure 7 - Spatial Kriging Results

Find coordinates for all missing values (to be predicted through Kriging):

```
   predCoords <- XYZM[is.na(XYZM[,3]),]
   predCoords <- predCoords[,-3]
```

Kriging parameters (ordinary kriging):

```
   j <- krige.control(type.krige = "ok", obj.model = fitVAR)
```

Perform kriging:

```
   Krig.10 <- krige.conv(Geo.M.sub, coords=Geo.M.sub$coords, data=Geo.M.sub$data,
locations=predCoords, krige=j)
```

Assign coordinates & matrix location to kriging results:

```
   lm <- length(predCoords)/2
   X_lon <- rep(0,lm)
   Y_lat <- rep(0,lm)
```

```
KSST.10.M <- cbind(predCoords, X_lon, Y_lat, Krig.10$predict, Krig.10$krige.var)
```

Generate indices from lat/long:

```
KSST.10.M[,3] <- 4*(KSST.10.M[,1]-5.125)+1
KSST.10.M[,4] <- -4*(KSST.10.M[,2]+30.125)+1
```

Use indices to place predictions into correct locations in an empty matrix:

```
SST.10KM <- matrix(0, 72, 240)
SST.10KV <- matrix(0, 72, 240)

for (i in 1:lm) {
SST.10KM[KSST.10.M[i,4],KSST.10.M[i,3]] <- KSST.10.M[i,5]
SST.10KV[KSST.10.M[i,4],KSST.10.M[i,3]] <- KSST.10.M[i,6]
}
```

Combine observations and krigged data:

```
TSST.10.M <- SST.10.M
TSST.10.M[!is.finite(TSST.10.M)] <- 0
TSST.10.M <- TSST.10.M + SST.10KM
```

Add Latitudinal Means:

```
TSST.10 <- TSST.10.M
for (i in 1:72) {
TSST.10[i,] <- TSST.10[i,] + latMean[i]
}
```

Finally, plot your results using the `image.plot` command, as in Figure 1. Note that this gives predictions for all areas - land included. To remove land from our predictions, we must create a mask of all points which we suspect to be land (the ones that return null values for every observation date) and remove this from our prediction. For computational efficiency, this step would normally be done before kriging, so that fewer sites are predicted.

We can decipher land by taking the mean temperature throughout the 331 days at each point:

```
MeanT.Map <- matrix(0, 72, 240)

for (i in 1:72) {
for (k in 1:240) {
MeanT.Map[i,k] <- mean(data.SST$SST.zone.period[i, k, 1:331], na.rm=TRUE)
}
}
```

Now, in a similar fashion to how we generated a list of locations for which to perform kriging, we can remove these "land" sites from the prediction:

```
zmt <- c(MeanT.Map)
XYzmt <- cbind(x,y,zmt)
nullCoords <- XYzmt[is.na(XYzmt[,3]),]
nullCoords <- nullCoords[,-3]
ln <- length(nullCoords)/2
Xn_lon <- rep(0,ln)
Yn_lat <- rep(0,ln)
nullList <- cbind(nullCoords, Xn_lon, Yn_lat)
nullList[,3] <- 4*(nullList[,1]-5.125)+1
nullList[,4] <- -4*(nullList[,2]+30.125)+1

for (i in 1:ln) {
TSST.10[nullList[i,4],nullList[i,3]] <- NA
}
```

Exercise 3

Figure 8 - Average SST vs. Time

This one is quite straightforward:

```
Day <- 1:331
MeanT <- NULL

for (i in 1:331) {
   MeanT[i] <- mean(data.SST$SST.zone.period[1:72, 1:240,i], na.rm=TRUE)
   }


   plot(Day, MeanT, main="Average Sea Surface Temperature vs.  Time",
ylab="Temperature (°C)", xlab="Day #")
```

Figure 9a) - Mean SSTs

See code for Figure 7.

Figure 9b) - SST Variance

```
VarT.Map <- matrix(0, 72, 240)
   for (i in 1:72) {
   for (k in 1:240) {
   VarT.Map[i,k] <- (sd(data.SST$SST.zone.period[i, k, 1:331], na.rm=TRUE))**2
   }
   }
```

Exercise 4

Figure 10 - Compare sea surface temperatures on different dates

Adjacent dates:

```
   SST.9 <- (data.SST$SST.zone.period[1:72, 1:240, 9])
   SST.11 <- (data.SST$SST.zone.period[1:72, 1:240, 11])
```

Farthest Pair:

```
   SST.60 <- (data.SST$SST.zone.period[1:72, 1:240, 60])
   SST.243 <- (data.SST$SST.zone.period[1:72, 1:240, 243])
```

Plot results:

```
   z9 <- c(SST.9); z11 <- c(SST.11);
   z60 <- c(SST.60); z243 <- c(SST.243)
   plot(z, z9, main="a) Day 9 vs.  Day 10 SST", ylab="Day 9 (°C)", xlab="Day 10 (°C)")
plot(z60, z243, main="b) Day 60 vs.  Day 243 SST", ylab="Day 243 (°C)", xlab="Day
60 (°C)")
```

Figure 11 - Temporal semivariograms at two different points

At central point:

```
   Mid <- data.SST$SST.zone.period[36, 120, 1:331]
   TCoord <- rep(1,1,331)
   TCoord <- cbind(c(1:331),TCoord)
   TVM <- vgram(TCoord, Mid, N=50, dmax=180)
   plot(TVM$centers,TVM$stats[2,], main="Temporal Variogram (Central Point)",
xlab="Lag Distance (days)", ylab="Gamma")
```

At northern point:

```
   Nor <- data.SST$SST.zone.period[10, 180, 1:331]
   TVN <- vgram(TCoord, Nor, N=50, dmax=180)
   plot(TVN$centers,TVN$stats[2,], main="Empirical Temporal Variogram (Northern Point)",
xlab="Lag Distance (days)", ylab="Gamma")
```