

Applications of satellite remote sensing for archaeological survey

A case study from the Sinis Archaeological Project, Sardinia

10/10/2019

This document details the methodology utilized in “Applications of satellite remote sensing for archaeological survey: A case study from the Sinis Archaeological Project, Sardinia” for visibility assesment through remote sensing. Due to restrictions in sharing Planet imagery and to the unpublished nature of the SAP survey data, we do not provide the actual data used in our study. Instead, the code here allows users to input their own data and download their own imagery, and in so doing reproduce our methodology rather than our particular results.

The first step is to input spatial data that contains all the survey units the user wishes to evaluate. This will most commonly be a shapefile, which ideally should be in the [UTM projection](#). This code assumes that each survey unit has an attribute for visibility, date of survey, land cover type, and density of finds.

Loading Data

```
library(rgdal)
su <- readOGR("file path to data") #read file
su$Date <- format(as.Date(su$Date), "%Y%m%d") #format date field as class date
su$Visibility <- as.numeric(su$Visibility) #format visibility field as class numeric
su$Count <- as.numeric(su$Count) #format count field as class numeric
su$LC <- factor(su$LC, levels = c("P","V", "G", "H", "F"),
               labels = c("Plowed", "Vineyard", "Grove", "Harvested", "Fallow"))
#format land cover field as class factor
su$Density = as.numeric(su$Density) #format density field as class numeric
```

```
summary(su)
```

```
## Object of class SpatialPolygonsDataFrame
## Coordinates:
##      min      max
## x 461807.3 465881.4
## y 4428216.1 4430942.0
## Is projected: TRUE
## proj4string :
## [+proj=utm +zone=32 +datum=WGS84 +units=m +no_defs +ellps=WGS84
## +towgs84=0,0,0]
## Data attributes:
##   SurveyUnit      Date      Visibility      Count
## Min.   : 1.0   Length:223   Min.   : 5.00   Min.   : 0.00
## 1st Qu.: 56.5   Class :character 1st Qu.: 37.50   1st Qu.: 0.00
## Median :112.0   Mode  :character  Median : 50.00   Median : 3.00
## Mean   :112.0                      Mean   : 54.01   Mean   : 26.92
## 3rd Qu.:167.5                      3rd Qu.: 70.00   3rd Qu.: 17.50
```

```
## Max.      :223.0          Max.      :100.00   Max.      :613.00
##          LC           Area           Density
## Plowed   :43   Min.      : 313.6   Min.      :0.0000000
## Vineyard :12   1st Qu.: 3172.0   1st Qu.:0.0000000
## Grove    :10   Median   : 5235.8   Median   :0.0008146
## Harvested:97   Mean     : 5907.0   Mean     :0.0058836
## Fallow   :61   3rd Qu.: 7821.5   3rd Qu.:0.0042475
##          Max.      :25556.8   Max.      :0.1151562
```

Visibility

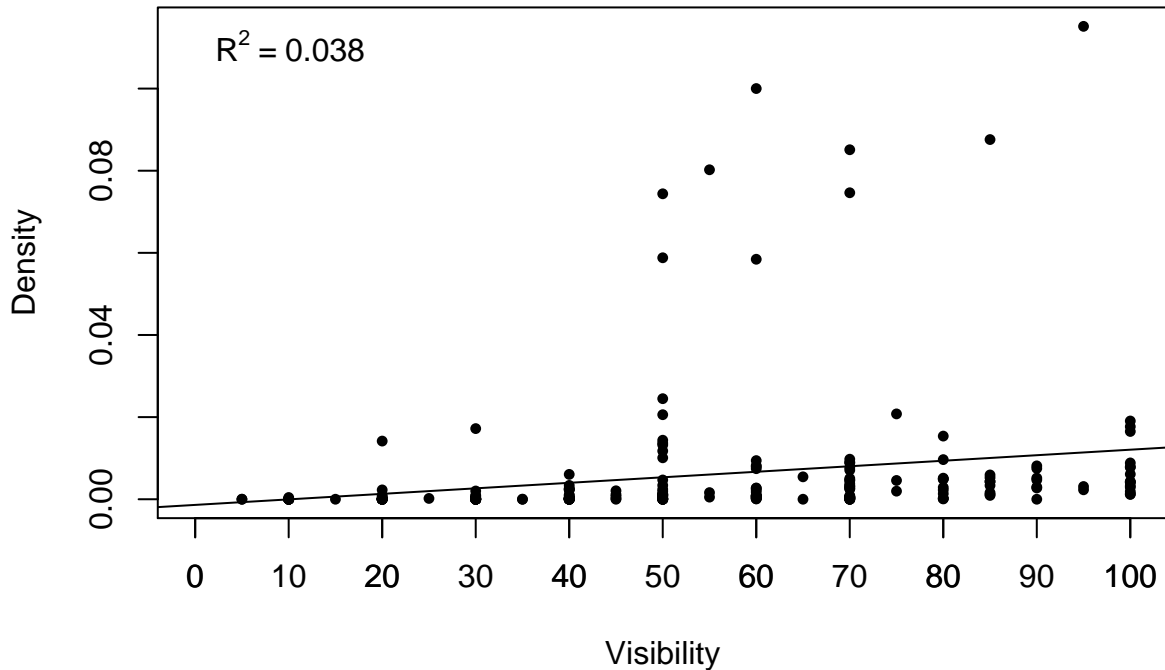
With these data loaded, right away we can see the relationship between field reported visibility and ceramic densities (Figure 2).

```
plot(su$Density~su$Visibility,
     xlab = "Visibility",
     ylab = "Density",
     main = "Density vs. Visibility",
     pch = 16,
     cex = 0.75,
     xlim = c(0,100))
axis(side = 1, at=seq(0,100,10))

#create linear regression model for Density and Visibility
lm1 <- lm(su$Density~su$Visibility)
r2.1 <- format(summary(lm1)$adj.r.squared, digits = 2)

abline(lm1)
legend("topleft", legend = substitute(paste(R^2~"="~x), list(x = r2.1)), bty = "n")
```

Density vs. Visibility



Getting Planet Data

Next, using the extent of these survey units, users can either upload or draw the boundaries of their survey area in [Planet's Explorer tool](#), or they can access the Planet API in order to query for images. The former is significantly easier to do and highly recommended for most users. If you do not have an account, one can be created through Planet's [Education and Research Program](#).

Calculating NDVI

Having downloaded the properly filtered imagery, we proceed to extract the desired assets ([surface reflectance product](#)) and merge and crop as necessary.

```
library(raster)

d <- "file path to where planet_order is saved"
bbox1 <- extent(su)

# will obtain file paths for all surface reflectance assets
images <- list.files(d, recursive = T, pattern = "MS_SR.tif$", full.names = T)

# will determine how many unique dates are present,
# since some dates will have multiple images that need to be merged
datesPlanet <- unique(substr(basename(images), 1, 8))

# will merge as necessary based on date
stk <- list()
```

```

for (i in 1:length(datesPlanet)){
  r1 <- images[grep(datesPlanet[i], images)]
  if (length(r1)>1){
    r2 <- lapply(r1, stack)
    r3 <- do.call("merge",r2)
  } else{
    r3 <- stack(r1)
  }
  r4 <- crop(r3, bbox1)
  stk[i] <- r4
}

```

Once a time series stack is created of merged multispectral imagery, we calculate NDVI for each image.

```

ndvi <- function(x){
  (x[[4]]-x[[3]])/(x[[4]]+x[[3]])
}

ndvi.stk <- lapply(stk, ndvi)
ndvi.stk <- do.call("stack",ndvi.stk)

names(ndvi.stk) <- datesPlanet

```

Now we can iterate through each SU and select the image whose date is closest to the date the unit was surveyed, extract NDVI values for all cells within the bounds of the unit, and average them. We write these values to the polygon file in a new column named NDVI. Additionally, we create another column named Datediff to record the number of days between the date the unit was surveyed and the closest available image.

```

#extract mean NDVI values per tract
su$NDVI <- NA
su$Datediff <- NA

for (i in 1:nrow(su)){
  dateRaster <-
    which(abs(as.numeric(datesPlanet)-as.numeric(su$Date[i])) ==
           min(abs(as.numeric(datesPlanet)-as.numeric(su$Date[i]))))
  su$NDVI[i] <- mean(unlist(raster::extract(ndvi.stk[[dateRaster[1]]], su[i,])))
  su$Dateddiff[i] <- min(abs(as.numeric(datesPlanet)-as.numeric(su$Date[i])))
}

```

Having calculated these values, we may now compare field reported visibility scores to NDVI scores. The following code produces Figure 3 of the article.

```

library(formatR)

p <- c("black", "purple", "green3", "orange", "red")

lm2 <- lm(su$NDVI~su$Visibility)
r2.2 <- format(summary(lm2)$adj.r.squared, digits = 2)
r2.2 <- paste("R^2 ==", r2.2)

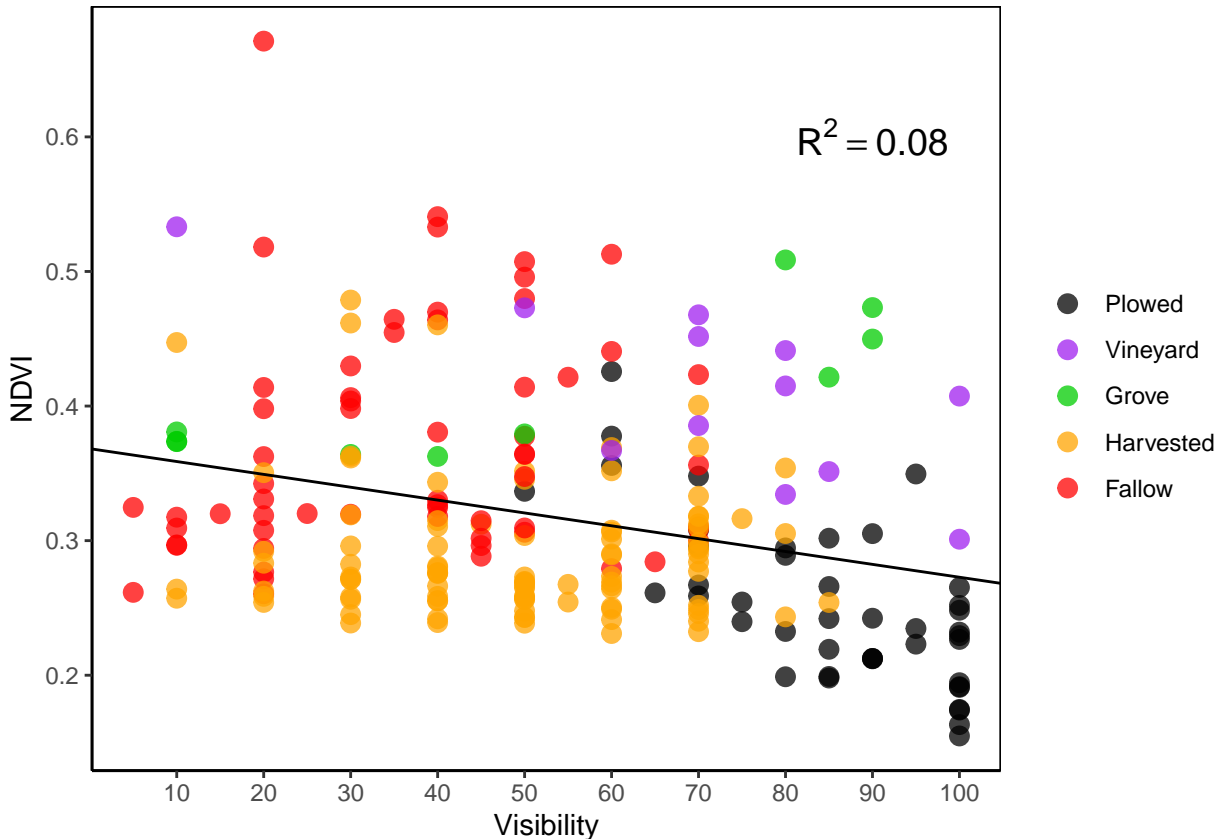
ggplot(su@data, aes(x=Visibility, y=NDVI, color = LC)) +

```

```

geom_point(size = 3, alpha = 0.75) +
geom_abline(slope = lm2$coefficients[2], intercept = lm2$coefficients[1]) +
scale_color_manual(values=p) +
theme_classic() +
theme(panel.border = element_rect(colour = "black", fill=NA, size=0.5)) +
theme(legend.title = element_blank()) +
scale_x_continuous("Visibility", seq(0,100,10)) +
scale_y_continuous("NDVI") +
annotate("text", x = 90, y = 0.6, label = r2.2, parse = T, size = 5)

```



To determine how land cover types affect the correlation between NDVI and visibility, we can also calculate the root mean square error (RMSE) of residuals according to land cover types. In our case, these types are Fallow ($n = \text{nrow}(\text{su}[\text{su}\$LC == "F",])$), Plowed ($n = 0$), Vineyard ($n = 0$), Grove ($n = 0$), and Harvested ($n = 0$). This code produces data for Table 2 and Figure 4.

```

resid = tibble("SurveyUnit" = seq(1, nrow(su), 1), "resid" = lm2$residuals)
su@data <- left_join(su@data, resid, by = "SurveyUnit")

rmse <- function(x){
  x^2 %>%
  mean() %>%
  sqrt()
}

LC_resid <- su@data %>%

```

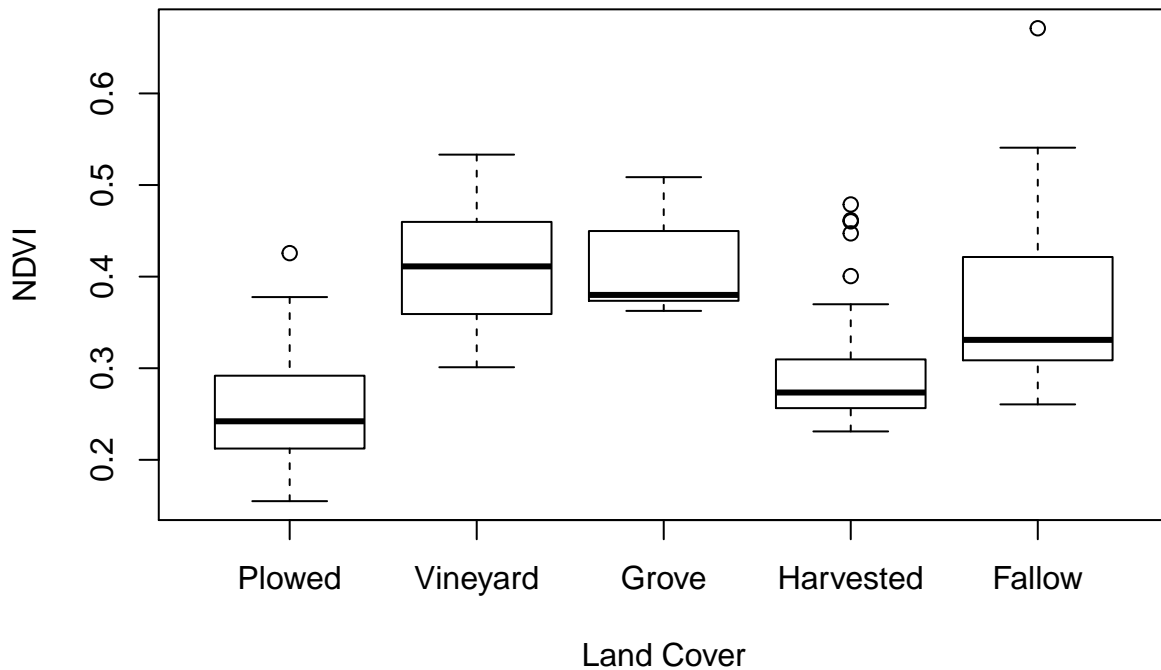
```
group_by(LC) %>%
  summarise(n = n(), rmse = rmse(resid))
```

LC_resid

	LC	n	rmse
1	Plowed	43	0.06
2	Vineyard	12	0.13
3	Grove	10	0.08
4	Harvested	97	0.07
5	Fallow	61	0.09

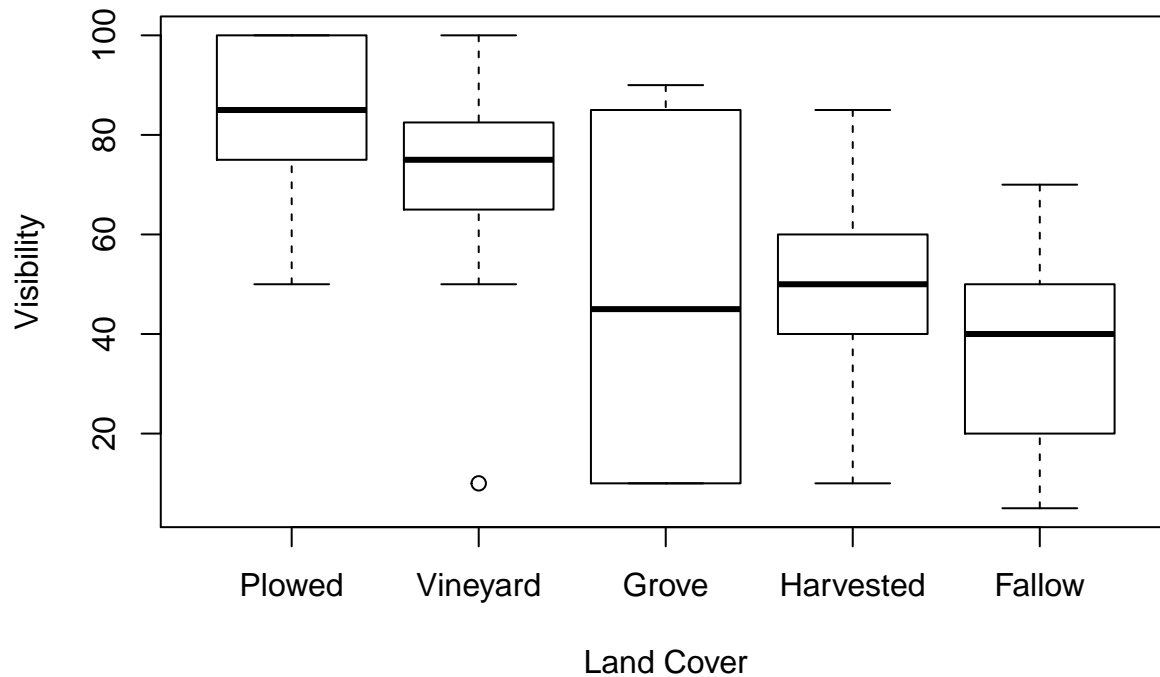
```
boxplot(su$NDVI~su$LC,
  main = "NDVI by Land Cover",
  ylab = "NDVI",
  xlab = "Land Cover")
```

NDVI by Land Cover



```
boxplot(su$Visibility~su$LC,
  main = "Visibility by Land Cover",
  ylab = "Visibility",
  xlab = "Land Cover")
```

Visibility by Land Cover



To explore further how land cover types affect the relationship between visibility scores and NDVI values, we create linear regression models for each of these types. This data are used to create Table 2.

```
LC_lm <- su@data %>%
  group_by(LC) %>%
  summarise(n = n(), Rsquared = summary(lm(NDVI~Visibility))$adj.r.squared,
            p = summary(lm(NDVI~Visibility))$coefficients[,4][2])
```

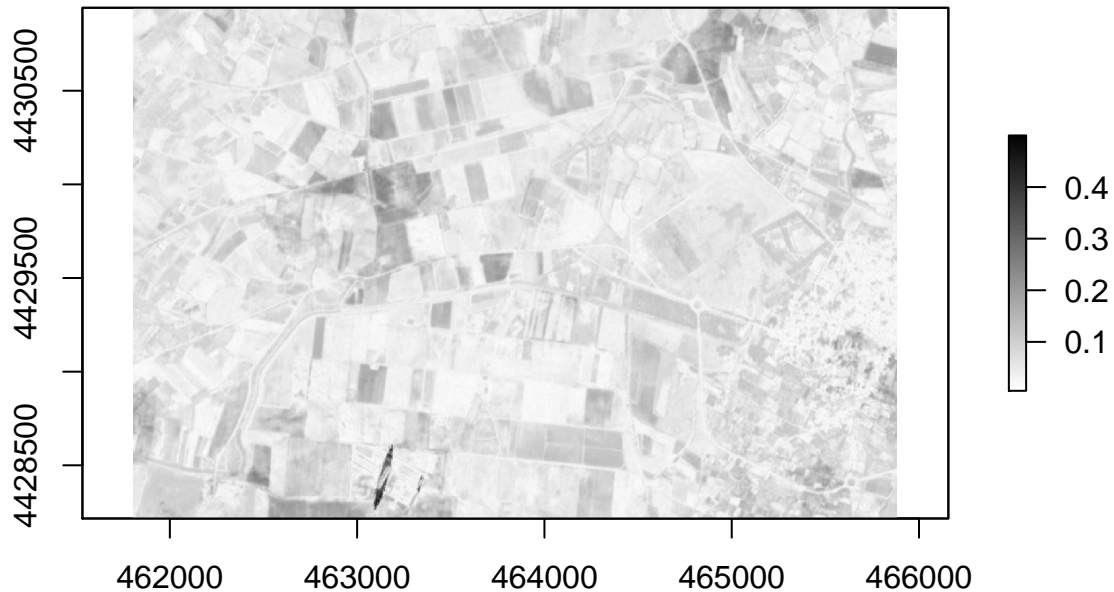
```
LC_lm
```

```
print(xtable(LC_lm), comment = F)
```

	LC	n	Rsquared	p
1	Plowed	43	0.47	0.00
2	Vineyard	12	0.48	0.01
3	Grove	10	0.61	0.00
4	Harvested	97	-0.00	0.38
5	Fallow	61	-0.00	0.33

To investigate how NDVI values changed with the survey area through time, we calculate standard deviations using the stack of NDVI images already created. These data procude Figure 6.

```
ndvi.sd <- calc(ndvi.stk, sd, na.rm = T)
plot(ndvi.sd, col=grey(100:1/100))
```



To explore how NDVI values can inform us about recent changes in land-use within our survey tracts, we select a group of adjacent survey units and plot their NDVI values over the course of our survey period.

```
su_ts <- su[su$SurveyUnit==104 |
  su$SurveyUnit==105 |
  su$SurveyUnit==147 |
  su$SurveyUnit==148 |
  su$SurveyUnit==165 |
  su$SurveyUnit==166 |
  su$SurveyUnit==167,]

ndvi_ts <- as.data.frame(matrix(nrow = length(datesPlanet), ncol = nrow(su_ts)))
names(ndvi_ts)= c(104,105,147,148,165,166,167)

for (i in 1:length(datesPlanet)){
  ndvi_ts$`104`[i] <- mean(unlist(raster::extract(ndvi.stk[[i]], su[su$SurveyUnit==104,])), na.rm = T)
  ndvi_ts$`105`[i] <- mean(unlist(raster::extract(ndvi.stk[[i]], su[su$SurveyUnit==105,])), na.rm = T)
  ndvi_ts$`147`[i] <- mean(unlist(raster::extract(ndvi.stk[[i]], su[su$SurveyUnit==147,])), na.rm = T)
  ndvi_ts$`148`[i] <- mean(unlist(raster::extract(ndvi.stk[[i]], su[su$SurveyUnit==148,])), na.rm = T)
  ndvi_ts$`165`[i] <- mean(unlist(raster::extract(ndvi.stk[[i]], su[su$SurveyUnit==165,])), na.rm = T)
  ndvi_ts$`166`[i] <- mean(unlist(raster::extract(ndvi.stk[[i]], su[su$SurveyUnit==166,])), na.rm = T)
  ndvi_ts$`167`[i] <- mean(unlist(raster::extract(ndvi.stk[[i]], su[su$SurveyUnit==167,])), na.rm = T)
}
```

```
plot(ndvi_ts$`104`,
  type = "l",
  xlim = ,
  ylim = c(0.1, 0.6),
  col = "red",
  ylab = "NDVI",
  xaxt="n",
  xlab = "",
  main = "Time Series NDVI")
axis(side = 1, at = 1:length(datesPlanet), labels = F)
```



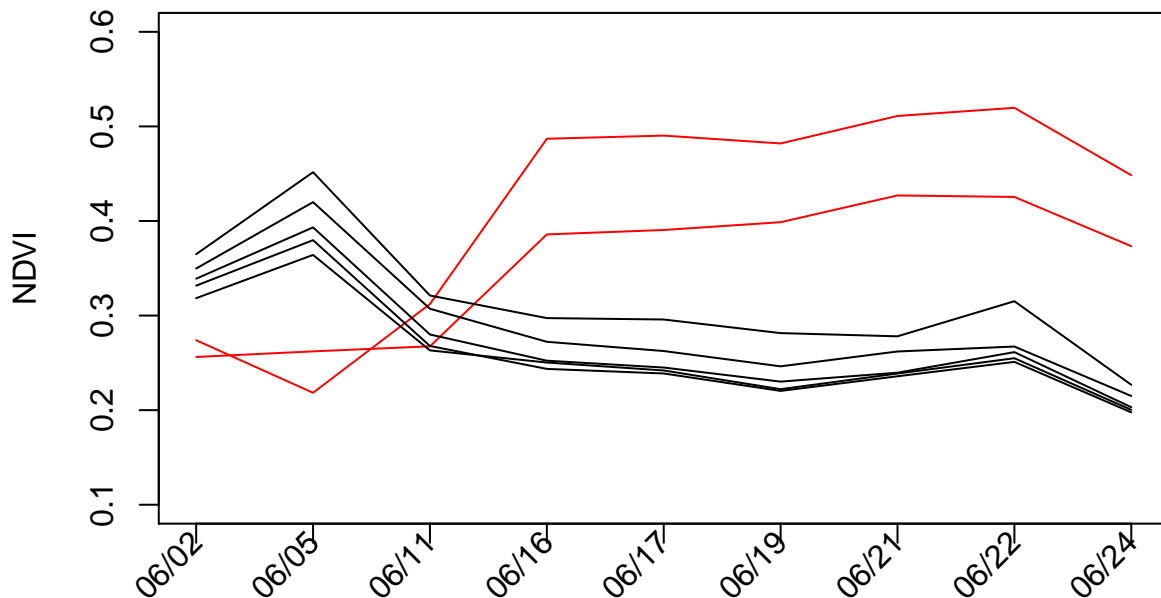
```

text(x=1:length(datesPlanet), y=par()$usr[3]-0.025*(par()$usr[4]-par()$usr[3]),
labels=format(as.Date(datesPlanet, "%Y%m%d"), "%m/%d"), srt=45, adj=1, xpd=TRUE)

lines(ndvi_ts$`105`, type = "l", col = "red")
lines(ndvi_ts$`147`, type = "l", col = "black")
lines(ndvi_ts$`148`, type = "l", col = "black")
lines(ndvi_ts$`165`, type = "l", col = "black")
lines(ndvi_ts$`166`, type = "l", col = "black")
lines(ndvi_ts$`167`, type = "l", col = "black")

```

Time Series NDVI



Finally, we test how ceramic counts per unit correlate with NDVI values and with field reported visibility scores at both the unit and transect levels.

```

lm3 <- lm(su$NDVI~su$Density)
summary(lm3)

```

```

##
## Call:
## lm(formula = su$NDVI ~ su$Density)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16200 -0.05556 -0.01617  0.04449  0.35668
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.313001   0.005861   53.4   <2e-16 ***
## su$Density   0.636241   0.334795    1.9   0.0587 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

```
## Residual standard error: 0.08243 on 221 degrees of freedom
## Multiple R-squared: 0.01608, Adjusted R-squared: 0.01163
## F-statistic: 3.611 on 1 and 221 DF, p-value: 0.05868
```

```
transects <- readOGR("file path to transect shapefile")

#extra mean NDVI values per tract
transects$tNDVI <- NA

for (i in 1:nrow(transects)){
  dateRaster <-
    which(abs(as.numeric(datesPlanet)-as.numeric(transects$Date[i]))==
          min(abs(as.numeric(datesPlanet)-as.numeric(transects$Date[i]))))
  transects$tNDVI[i] <- mean(unlist(raster::extract(ndvi.stk[[dateRaster[1]]], transects[i,])))
}

lm3 <- lm(transects$tNDVI~transects$density)
summary(lm3)
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Volumes/GoogleDrive/My Drive/Archives of Projects/completed/Publications/2020_AAP/Sardinia"
## with 1052 features
## It has 13 fields
```

```
##
## Call:
## lm(formula = transects$tNDVI ~ transects$density)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.136486 -0.033792 -0.007543  0.032054  0.187305
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.304146   0.001709  177.949 < 2e-16 ***
## transects$density 0.109663   0.015714   6.979 5.27e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05335 on 1049 degrees of freedom
## (1 observation deleted due to missingness)
## Multiple R-squared: 0.04437, Adjusted R-squared: 0.04346
## F-statistic: 48.71 on 1 and 1049 DF, p-value: 5.265e-12
```