NDVI time series analysis

CIAT

27 October, 2016, 22:45

Objectives

This manual will help you construct a time series using MODIS data. At the end of this session, you will be able to:

1. Import NDVI data into R software
2. Insert a temporal ID into the data
3. Create a time series
4. Plot and export graphs

For more details on source of MODIS data see http://pekko.geog.umd.edu/usda/beta/data_new.php. Otherwise, download the sample dataset we will use in this session from this link https://drive.google.com/open?id=0B_Gkb_0tNKkQUVc5NUhKRzlhZTg

Before you start this session, it is important you have (i) the latest R software and (ii) Rstudio installed in your computer.

You can now start the session but first clear your work space:

```
rm(list = ls(all = TRUE))
```

Preliminaries

You need to first install all the required packages. Type (within R) install packages("name of package") (This needs to be done just once.). You then load the packages using the 'library' function.

The code below lists down the packages to be used in this session and installs if not already installed then loads into the session.

```
.packages = c("rgdal","raster","ggplot2","dplyr","lubridate","grid","gridExtra")
.inst <- .packages %in% installed.packages()
if(length(.packages[!.inst]) > 0) install.packages(.packages[!.inst])
lapply(.packages, require, character.only=TRUE)
```

```
## Loading required package: rgdal
```

```
## Loading required package: sp
```

```
## rgdal: version: 1.1-8, (SVN revision 616)
##  Geospatial Data Abstraction Library extensions to R successfully loaded
##  Loaded GDAL runtime: GDAL 2.0.1, released 2015/09/15
```

```
##  Path to GDAL shared files: C:/Users/jymutua/Documents/R/win-library/3.1/rgdal/gdal
##  GDAL does not use iconv for recoding strings.
##  Loaded PROJ.4 runtime: Rel. 4.9.1, 04 March 2015, [PJ_VERSION: 491]
##  Path to PROJ.4 shared files: C:/Users/jymutua/Documents/R/win-library/3.1/rgdal/proj
##  Linking to sp version: 1.2-3

## Loading required package: raster

## Loading required package: ggplot2

## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:raster':
##
##     intersect, select, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: lubridate

##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

## Loading required package: grid

## Loading required package: gridExtra

##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine

## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] TRUE
##
## [[5]]
## [1] TRUE
##
## [[6]]
## [1] TRUE
##
## [[7]]
## [1] TRUE
```

To get help on the functions and data sets in R, use `help()` or `?`. For example, to view the help file for the `calc` function, type one of the following:

```
help(calc)
?calc
```

Set your working directory as follows:

```
setwd("C:/LDN_Workshop/Sample_dataset/NDVI_data")
```

Import all NDVI raster datasets in your working directory and create a raster stack

```
NDVI_Otji_stack <- stack(list.files(getwd(), full.names = TRUE,
                                    pattern = ".tif$"))
```

Insert a Temporal id to the data frame. Characters corresponding to the year are located between the 20 and 23 position, while characters corresponding to the date are located between the 24 and 26 position

```
oldnames<-names(NDVI_Otji_stack)
head(oldnames)
```

```
## [1] "Otjozondjupa_NDVI_.2005001.C05.NDVI.MOD44.D16.R000250.MODAPS.v2_wm2"
## [2] "Otjozondjupa_NDVI_.2005017.C05.NDVI.MOD44.D16.R000250.MODAPS.v2_wm2"
## [3] "Otjozondjupa_NDVI_.2005033.C05.NDVI.MOD44.D16.R000250.MODAPS.v2_wm2"
## [4] "Otjozondjupa_NDVI_.2005049.C05.NDVI.MOD44.D16.R000250.MODAPS.v2_wm2"
## [5] "Otjozondjupa_NDVI_.2005065.C05.NDVI.MOD44.D16.R000250.MODAPS.v2_wm2"
## [6] "Otjozondjupa_NDVI_.2005081.C05.NDVI.MOD44.D16.R000250.MODAPS.v2_wm2"
```

```
year<-substr(oldnames,20,23)
table(year)
```

```
## year
## 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015
##   23   23   23   22   23   23   23   23   23   23   23
```

```
julianDay<-substr(oldnames,24,26)
```

Calculate mean NDVI for each raster and convert output array to a data frame

```
avg_NDVI_Otji <- as.data.frame(cellStats(NDVI_Otji_stack,mean))
```

Rename the NDVI column to 'meanNDVI'.

```
names(avg_NDVI_Otji) <- "meanNDVI"
```

Add Julian day values as a column in the data frame.

```
avg_NDVI_Otji$julianDay <- julianDay
```

Let's check out what 'class' in the new column

```
class(avg_NDVI_Otji$julianDay)
```

```
## [1] "character"
```

Now we can create a time vector and convert it to a date

```
tVector<-paste(year,julianDay,sep="-")
timeNDVI<-as.Date(tVector,format = "%Y-%j")
```

Add date values as a column in the data frame

```
avg_NDVI_Otji$date <- timeNDVI
```

Let's check out 'class' of the two columns now

```
class(avg_NDVI_Otji$date)
```

```
## [1] "Date"
```

```
class(avg_NDVI_Otji$julianDay)
```

```
## [1] "character"
```

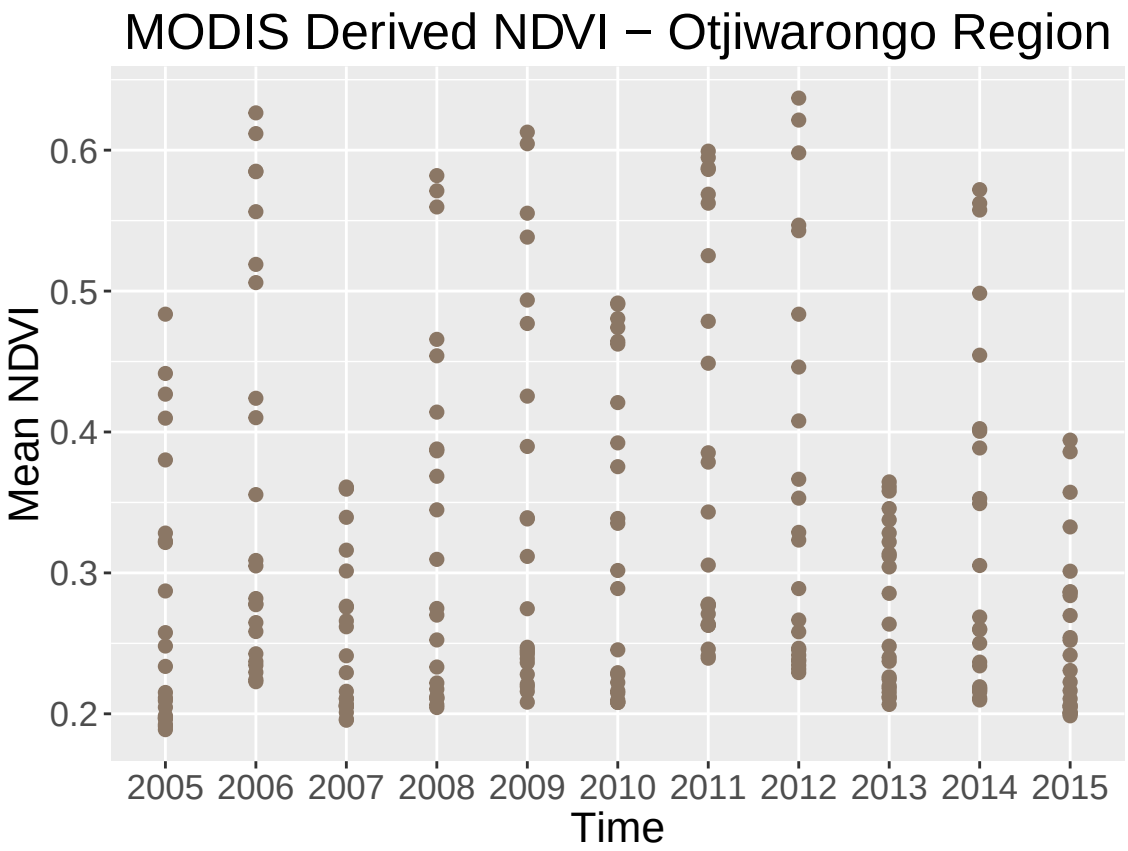
Add site and year columns to the data frame

```
avg_NDVI_Otji$site <- "Otjiwarongo Region"
avg_NDVI_Otji$year <- year
```

Plot time series

Plot NDVI by year although this doesn't make sense

```
ggplot(avg_NDVI_Otji, aes(year, meanNDVI), na.rm=TRUE) +
  geom_point(size=2,colour = "PeachPuff4") +
  ggtitle("MODIS Derived NDVI - Otjiwarongo Region") +
  xlab("Time") + ylab("Mean NDVI") +
  theme(text = element_text(size=16))
```



Write NDVI data to a comma separated file in your local drive, Drop the row names column

```
Otji_NDVI_Values<-avg_NDVI_Otji
row.names(Otji_NDVI_Values)<-NULL
write.csv(Otji_NDVI_Values, file="meanNDVI_Otji_2005-2015.csv")
```

Add month to data frame

```
avg_NDVI_Otji$month  <- month(avg_NDVI_Otji$date)
```

Subset data by season by creating a new categorical variable called season by grouping months together

```
avg_NDVI_Otji_Seasons <- avg_NDVI_Otji %>%
  mutate(season =
           ifelse(month %in% c(12, 1, 2, 3, 4, 5), "Hot-Wet",
                  ifelse(month %in% c(6, 7, 8), "Cool-Dry",
                         ifelse(month %in% c(9, 10, 11), "Hot-Dry", "Error"))))
```

Let's check to see if that worked

```r
head(avg_NDVI_Otji_Seasons$month)
```

```
## [1] 1 1 2 2 3 3
```

```r
head(avg_NDVI_Otji_Seasons$season)
```

```
## [1] "Hot-Wet" "Hot-Wet" "Hot-Wet" "Hot-Wet" "Hot-Wet" "Hot-Wet"
```

```r
tail(avg_NDVI_Otji_Seasons$month)
```

```
## [1]  9 10 11 11 12 12
```

```r
tail(avg_NDVI_Otji_Seasons$season)
```

```
## [1] "Hot-Dry" "Hot-Dry" "Hot-Dry" "Hot-Dry" "Hot-Wet" "Hot-Wet"
```

Start of monthly plots

Aggregate data by month

```r
monthNDVI<-avg_NDVI_Otji_Seasons %>%
  group_by(month) %>%
  summarise(monthNDVI=mean(meanNDVI, na.rm=TRUE))
```

Convert month numeric to month abbreviation

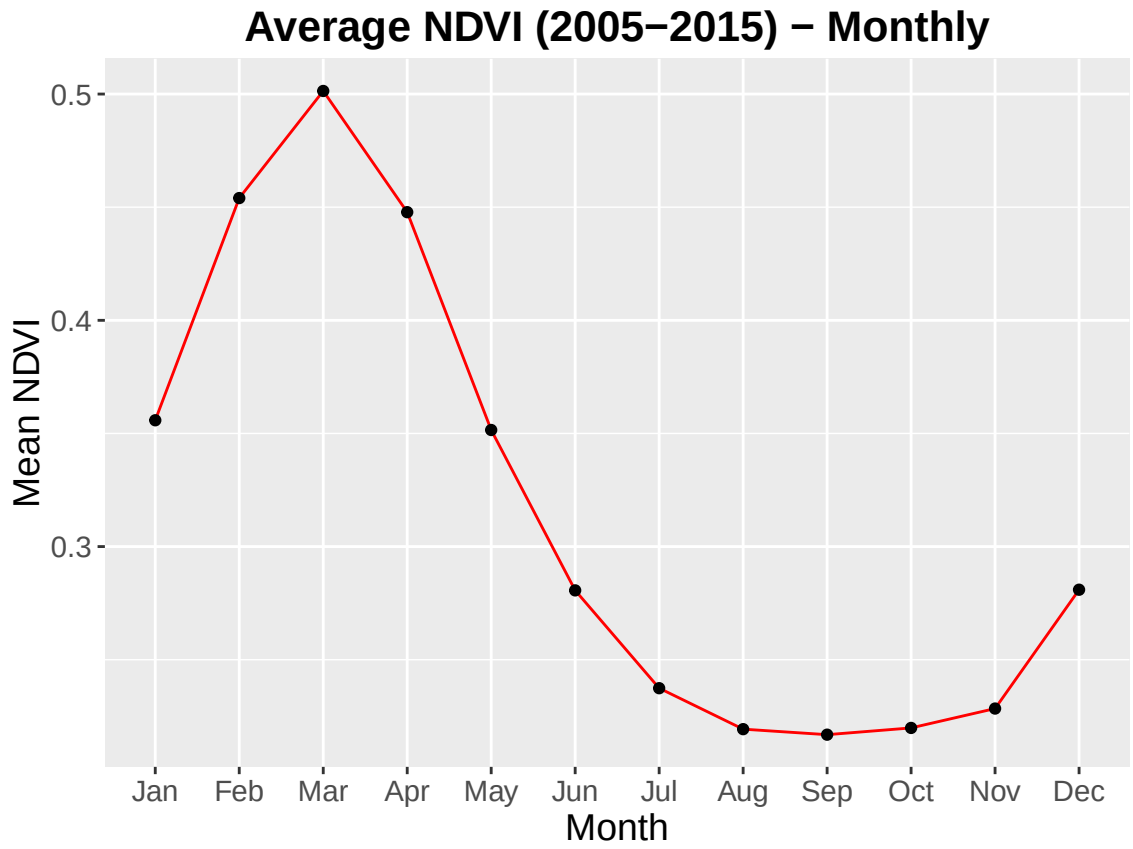
```r
monthNDVI$month_name <- month.abb[monthNDVI$month]
```

Reassign the 'month_name' field to a factor

```r
monthNDVI$month_name = factor(monthNDVI$month_name,
                              levels=c('Jan','Feb','Mar',
                                       'Apr','May','Jun','Jul',
                                       'Aug','Sep','Oct',
                                       'Nov','Dec'))
```

Plot data by month

```r
Monthly_NDVI_Plot<-ggplot(monthNDVI, aes(month_name, monthNDVI, group=4)) +
  geom_line(colour="red") +
  ggtitle("Average NDVI (2005-2015) - Monthly") +
  xlab("Month") + ylab("Mean NDVI") +
  theme(plot.title = element_text(lineheight=.8, face="bold",
                                  size = 16)) +
  theme(text = element_text(size=14)) + geom_point()
Monthly_NDVI_Plot
```



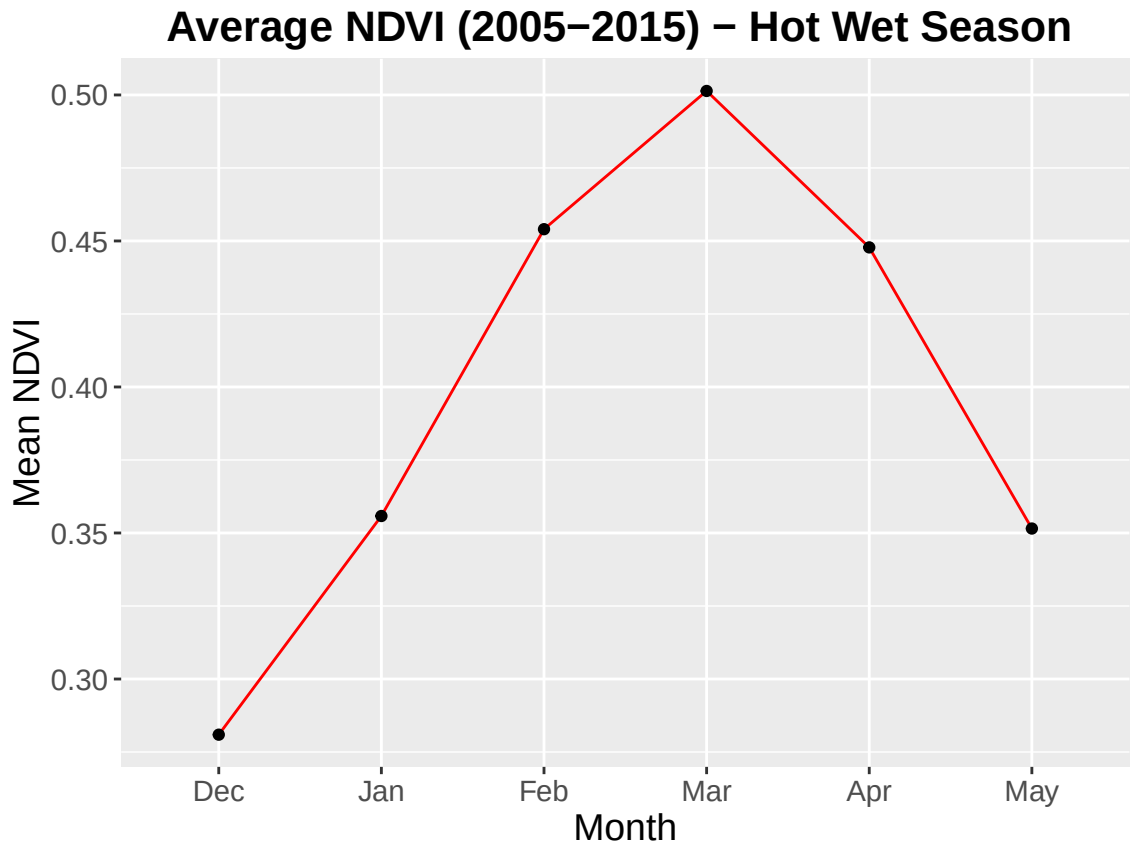
Plot the same plot as before but with one plot per season, save as .pdf

```
ggsave(file="Otjiwarongo_Monthly_NDVI.pdf", width = 297, height = 210, units =
         "mm")
```

Start of indivindual seasonal plots

Plot HOT-WET Season

```
Hot_Wet <- subset(monthNDVI, month >= 12 | month <= 5)
target <- c("12", "5", "4", "3", "2", "1")
Hot_Wet<-Hot_Wet[match(target, Hot_Wet$month),]
Hot_Wet$month_name <- factor(Hot_Wet$month_name, c("Dec", "Jan", "Feb", "Mar",
                                                   "Apr", "May"))
Hot_Wet_NDVI_Plot<-ggplot(Hot_Wet, aes(month_name, monthNDVI, group=1)) +
  geom_line(colour="red") +
  ggtitle("Average NDVI (2005-2015) - Hot Wet Season ") +
  xlab("Month") + ylab("Mean NDVI") +
  theme(plot.title = element_text(lineheight=.8, face="bold",
                                  size = 16)) +
  theme(text = element_text(size=14)) + geom_point()
Hot_Wet_NDVI_Plot
```

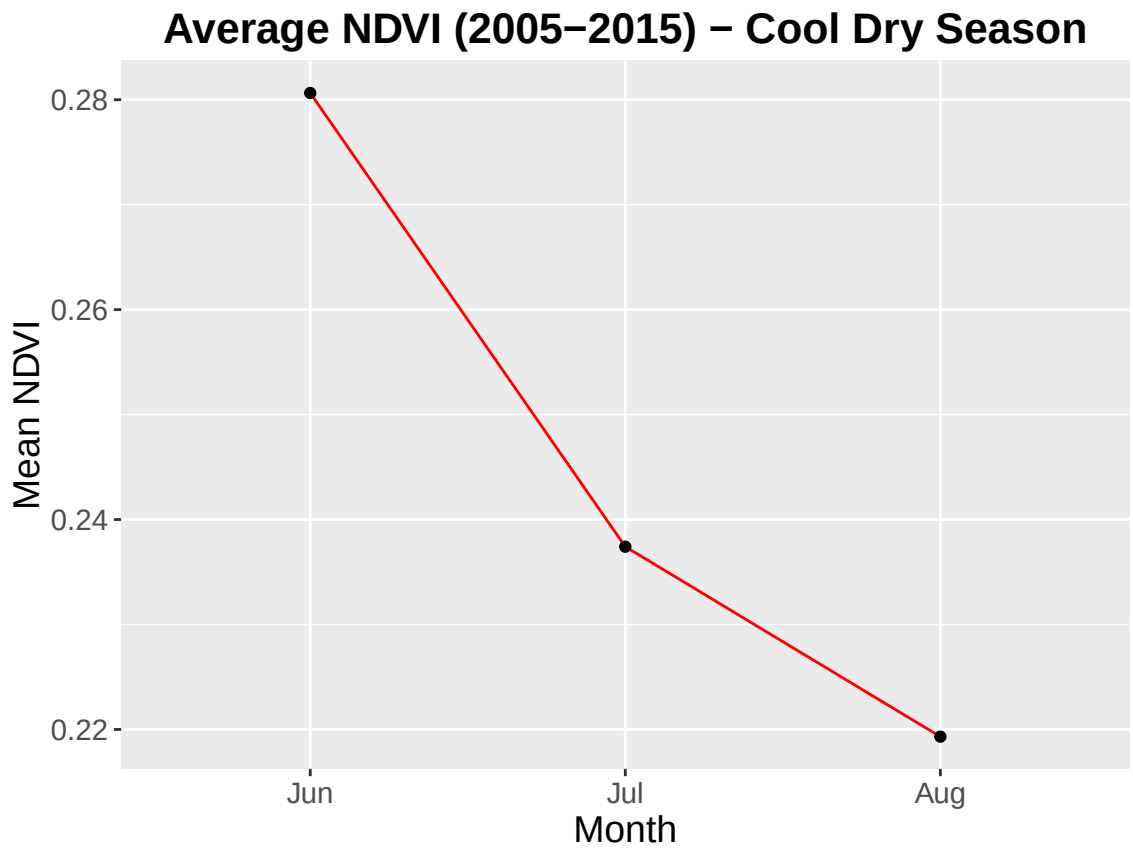


Plot the same plot as before but with one plot per season, save as .pdf

```
ggsave(file="Otjiwarongo_Hot_Wet_NDVI.pdf", width = 297, height = 210, units =
        "mm")
```

Plot COOL-DRY Season

```
Cool_Dry <- subset(monthNDVI, month <= 8 & month >= 6)
Cool_Dry_NDVI_Plot<-ggplot(Cool_Dry, aes(month_name, monthNDVI, group=1)) +
  geom_line(colour="red") +
  ggtitle("Average NDVI (2005-2015) - Cool Dry Season") +
  xlab("Month") + ylab("Mean NDVI") +
  theme(plot.title = element_text(lineheight=.8, face="bold",
                                  size = 16)) +
  theme(text = element_text(size=14)) + geom_point()
Cool_Dry_NDVI_Plot
```

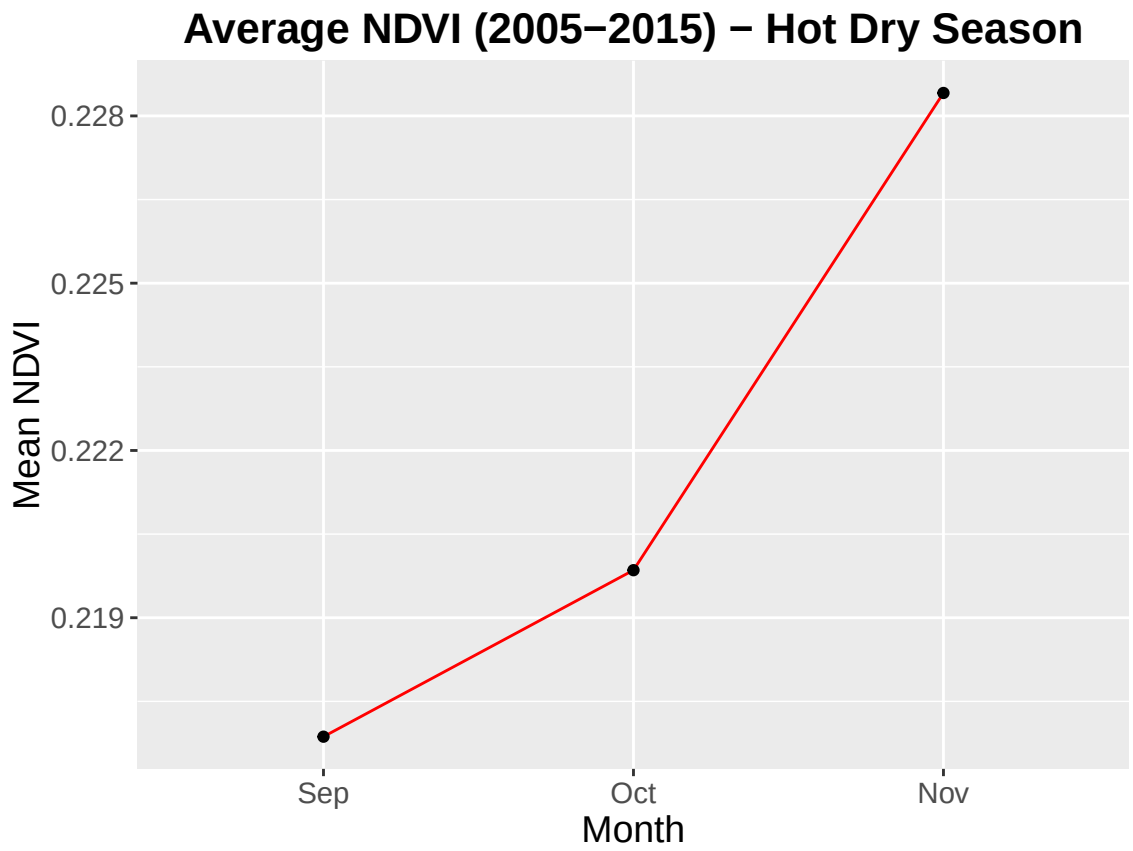


Plot the same plot as before but with one plot per season, save as .pdf

```
ggsave(file="Otjiwarongo_Cool_Dry_NDVI.pdf", width = 297, height = 210, units =
        "mm")
```

Plot HOT DRY Season

```
Hot_Dry <- subset(monthNDVI, month <= 11 & month >= 9)
Hot_Dry_NDVI_Plot<-ggplot(Hot_Dry, aes(month_name, monthNDVI, group=1)) +
  geom_line(colour="red") +
  ggtitle("Average NDVI (2005-2015) - Hot Dry Season") +
  xlab("Month") + ylab("Mean NDVI") +
  theme(plot.title = element_text(lineheight=.8, face="bold",
                                  size = 16)) +
  theme(text = element_text(size=14)) + geom_point()
Hot_Dry_NDVI_Plot
```



Plot the same plot as before but with one plot per season, save as .pdf

```
ggsave(file="Otjiwarongo_Hot_Dry_NDVI.pdf", width = 297, height = 210, units =
        "mm")
```

Start of merged seasonal plots

Let's aggregate data by season

```
seasonNDVI<-avg_NDVI_Otji_Seasons %>%
  group_by(year, season) %>%
  summarise(seasonNDVI=mean(meanNDVI, na.rm=TRUE))
```

Export comma separated file of seasonal NDVI values

```
write.csv(seasonNDVI, file="Seasonal_NDVI_2005-2015.csv")
```

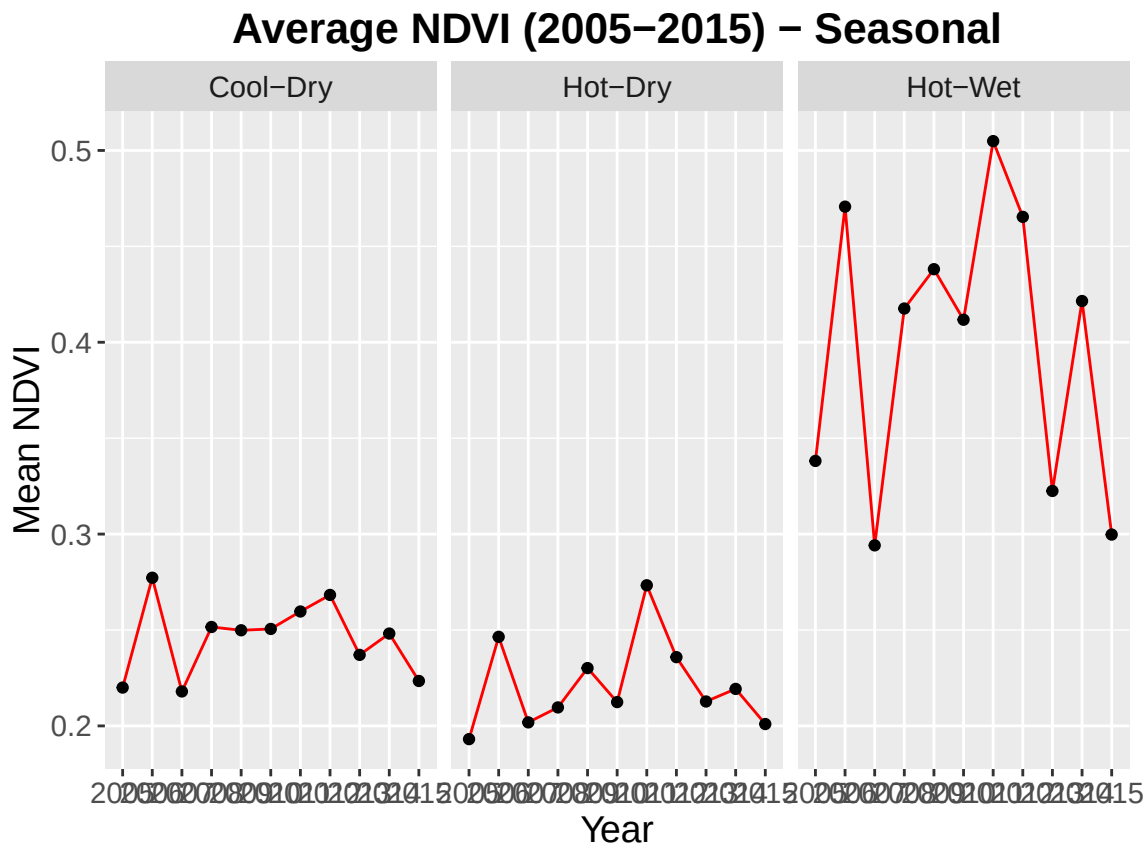
Plot data by season

```
Seasonal_NDVI_Plot<-ggplot(seasonNDVI, aes(year, seasonNDVI, group=3)) +
  geom_line(colour="red") +
  ggtitle("Average NDVI (2005-2015) - Seasonal") +
```

```
  xlab("Year") + ylab("Mean NDVI") +
  theme(plot.title = element_text(lineheight=.8, face="bold",
                                  size = 16)) +
  theme(text = element_text(size=14)) + geom_point()
```

Plot the same plot as before but with one plot per season, save as .pdf

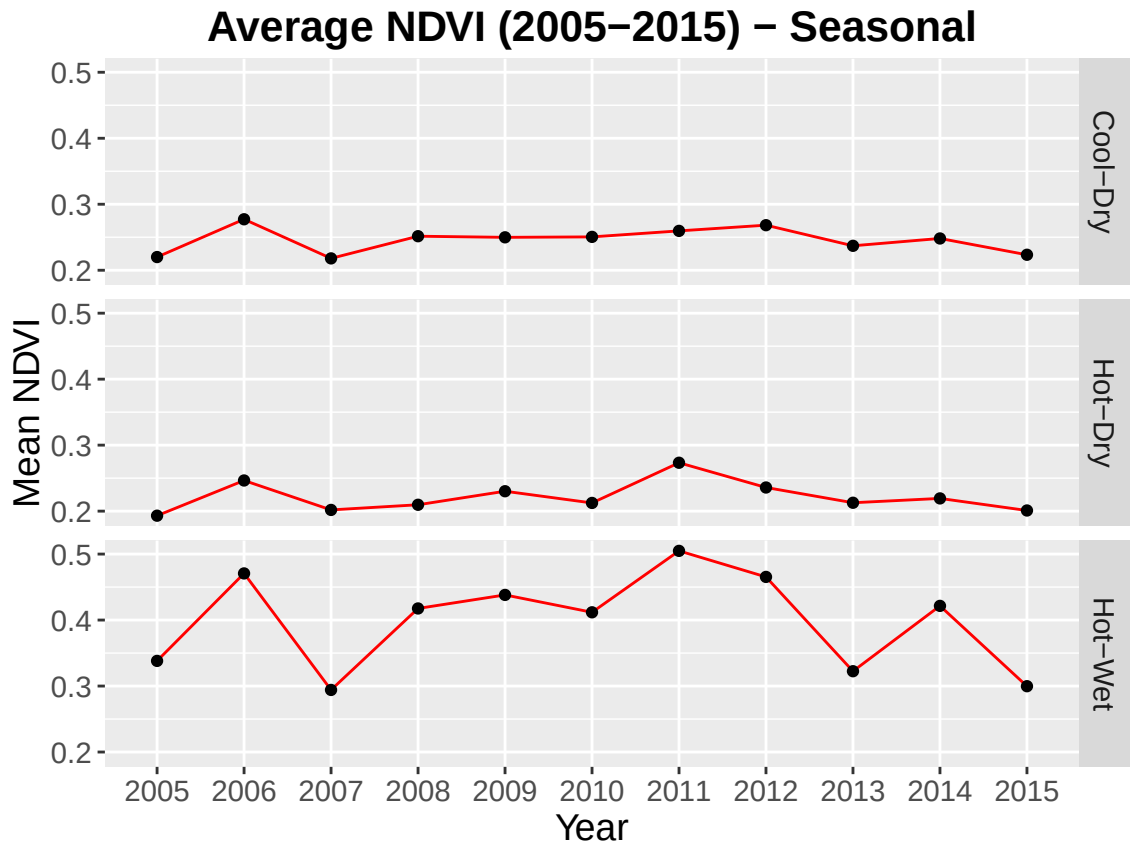
```
Seasonal_NDVI_Plot + facet_grid(. ~ season)
```




```
ggsave(file="Otjiwarongo_Seasonal_NDVI.pdf", width = 297, height = 210, units =
         "mm")
```

Plot the same plot in a landscape orientation, save as .pdf

```
Seasonal_NDVI_Plot + facet_grid(season ~ .)
```




```
ggsave(file="Otjiwarongo_Seasonal_NDVI.pdf", width = 297, height = 210, units =
        "mm")
```