# NDVI time series analysis

*CIAT*

*06 November, 2016, 20:41*

### 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.

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:

```r
#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.

```r
#load packages
.packages = c("rgdal","raster","ggplot2","dplyr","lubridate","grid","gridExtra","dygraphs")
.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: 1.1-10, (SVN revision 622)
##  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/3.3/rgdal/gdal
##  Loaded PROJ.4 runtime: Rel. 4.9.2, 08 September 2015, [PJ_VERSION: 492]
##  Path to PROJ.4 shared files: C:/Users/jymutua/Documents/R/win-library/3.3/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

## Loading required package: dygraphs

## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] TRUE
##
## [[5]]
## [1] TRUE
##
## [[6]]
## [1] TRUE
##
## [[7]]
## [1] TRUE
##
## [[8]]
## [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:

```
#this is how you read more on functions
help(calc)
?calc
```

Set your working directory as follows:

```
#set your working directory
setwd("C:/LDN_Workshop/Sample_dataset/NDVI_data")
```

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

```
#import rasters
r.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

```
#insert temporal ID in the dataframe
oldnames<-names(r.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

```
#calculate mean NDVI
avg.NDVI <- as.data.frame(cellStats(r.stack,mean))
```

Rename the NDVI column.

```
#rename NDVI column
names(avg.NDVI) <- "NDVI"
```

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

```
#add julina day as a column
avg.NDVI$julianDay <- julianDay
```

Let’s check out what ‘class’ in the new column

```
#check out the new column
class(avg.NDVI$julianDay)
```

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

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

```r
#create time vector and convert to date
tVector<-paste(year,julianDay,sep="-")
timeNDVI<-as.Date(tVector,format = "%Y-%j")
```

Add date values as a column in the data frame

```r
#add date values as a column in the data frame
avg.NDVI$date <- timeNDVI
```

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

```r
#check out 'class' of the two columns now
class(avg.NDVI$date)
```

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

```r
class(avg.NDVI$julianDay)
```

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

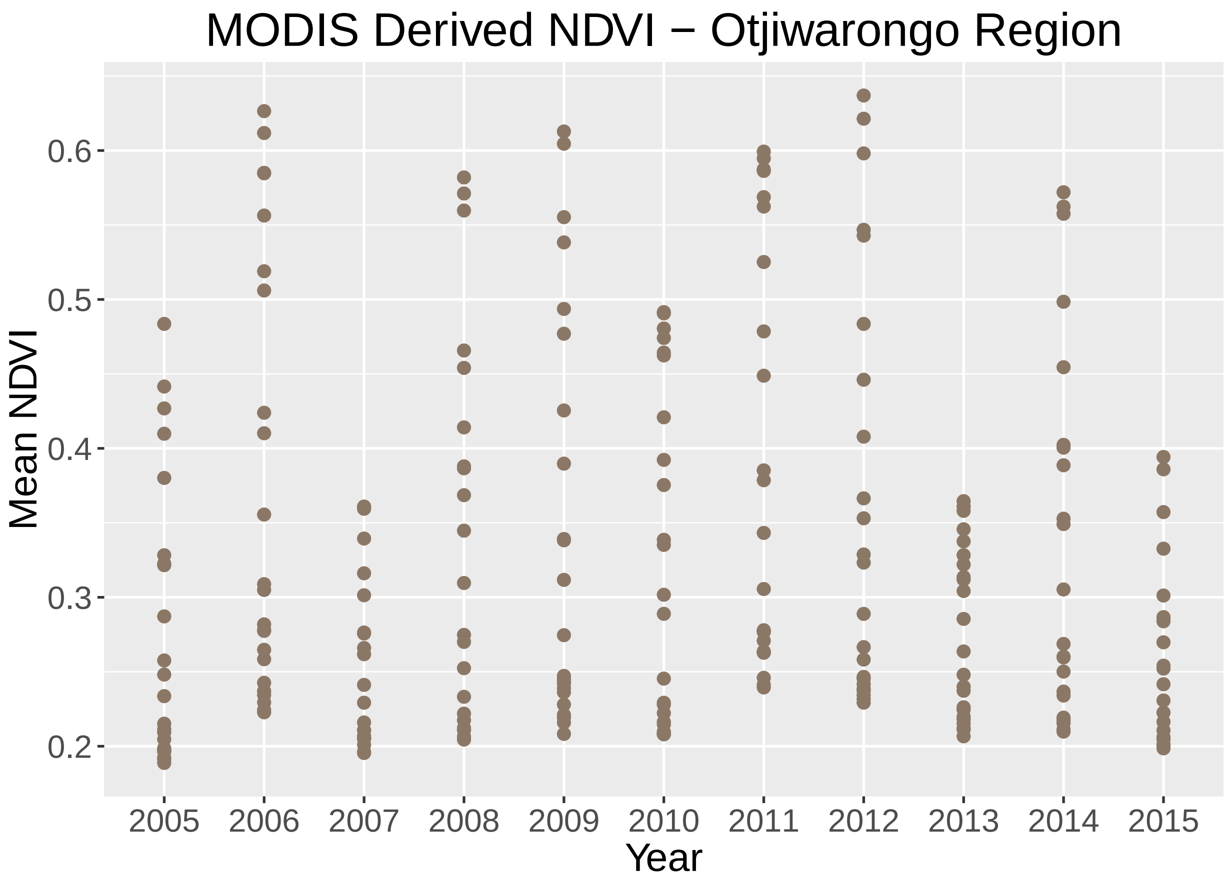
Add site and year columns to the data frame

```r
#add site and year columns to the data frame
avg.NDVI$site <- "Otjiwarongo Region"
avg.NDVI$year <- year
```

### Plot time series

Plot NDVI by year although this doesn't make sense

```r
#plot NDVI by year although this doesn't make sense
ggplot(avg.NDVI, aes(year, NDVI), na.rm=TRUE) +
  geom_point(size=2,colour = "PeachPuff4") +
  ggtitle("MODIS Derived NDVI - Otjiwarongo Region") +
  xlab("Year") + 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

```
#write NDVI data to .csv
NDVI.Values<-avg.NDVI
row.names(NDVI.Values)<-NULL
write.csv(NDVI.Values, file="NDVI_2005-2015.csv")
```

Add month to data frame

```
#add month to data frame
avg.NDVI$month  <- month(avg.NDVI$date)
```

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

```
#subset data by season
avg.NDVI <- avg.NDVI %>%
  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

```
#did the above work?
head(avg.NDVI$month)
```

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

```
head(avg.NDVI$season)
```

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

```
tail(avg.NDVI$month)
```

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

```
tail(avg.NDVI$season)
```

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

### Start of monthly plots

Aggregate data by month and calculate standard deviation and error

```
#aggregate data by month and calculate standard deviation and error
month.summary<-avg.NDVI %>%
  group_by(month) %>%
  summarise(mean.NDVI=mean(NDVI, na.rm=TRUE),
            sd_NDVI = sd(NDVI), #standard deviation of each group
            n_NDVI = n(),  #sample size per group
            SE_NDVI = sd(NDVI)/sqrt(n())) #standard error of each group
```

Convert month numeric to month abbreviation

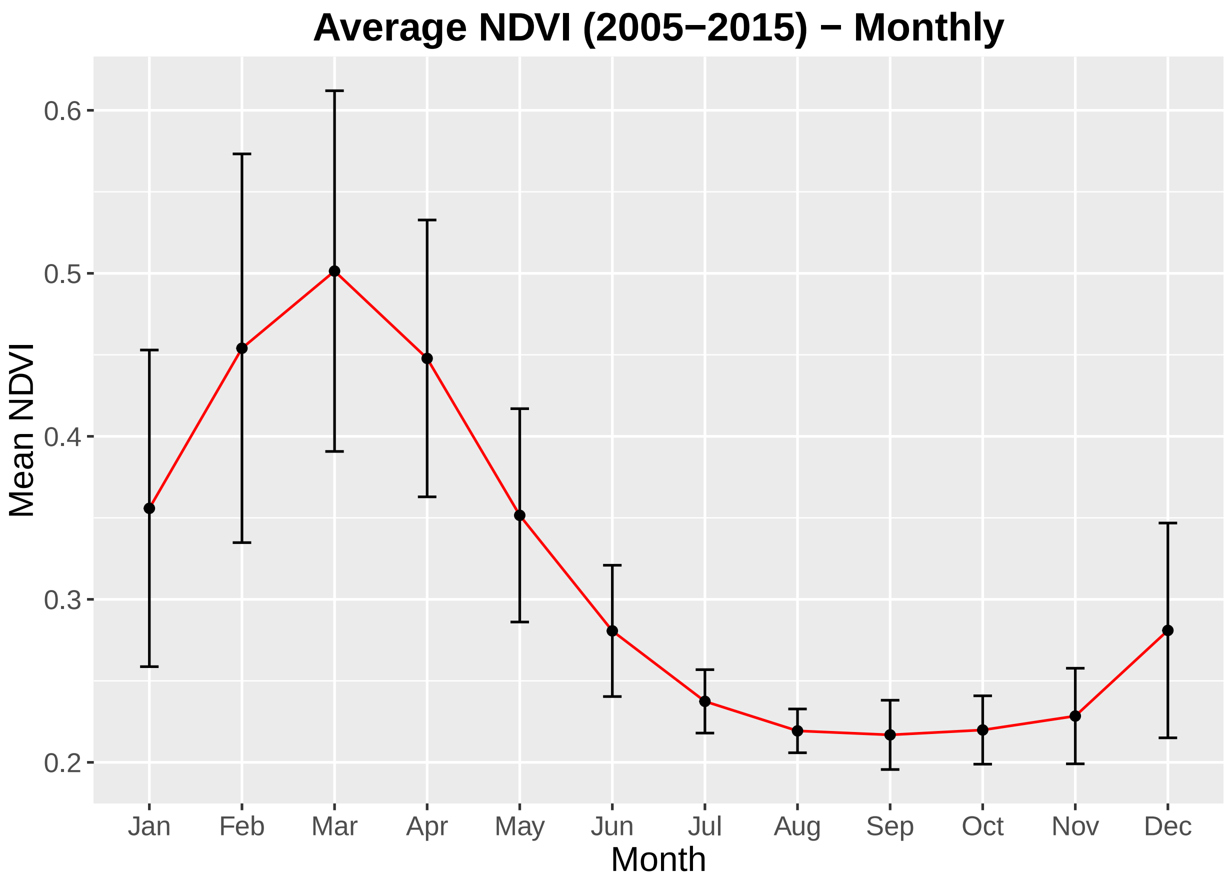
```
#convert month numeric to month abbreviation
month.summary$month_name <- month.abb[month.summary$month]
```

Reassign the 'month_name' field to a factor

```
#reassign the 'month_name' field to a factor
month.summary$month_name = factor(month.summary$month_name,
                              levels=c('Jan','Feb','Mar',
                                       'Apr','May','Jun','Jul',
                                       'Aug','Sep','Oct',
                                       'Nov','Dec'))
```

Plot data by month and save as .pdf

```
#plot data by month and save as .pdf
monthly.NDVI.plot<-ggplot(month.summary, aes(month_name, mean.NDVI, group=4)) +
  geom_line(colour="red") +
  geom_errorbar(aes(ymin=mean.NDVI-sd_NDVI, ymax=mean.NDVI+sd_NDVI),width=0.2) +
  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
```

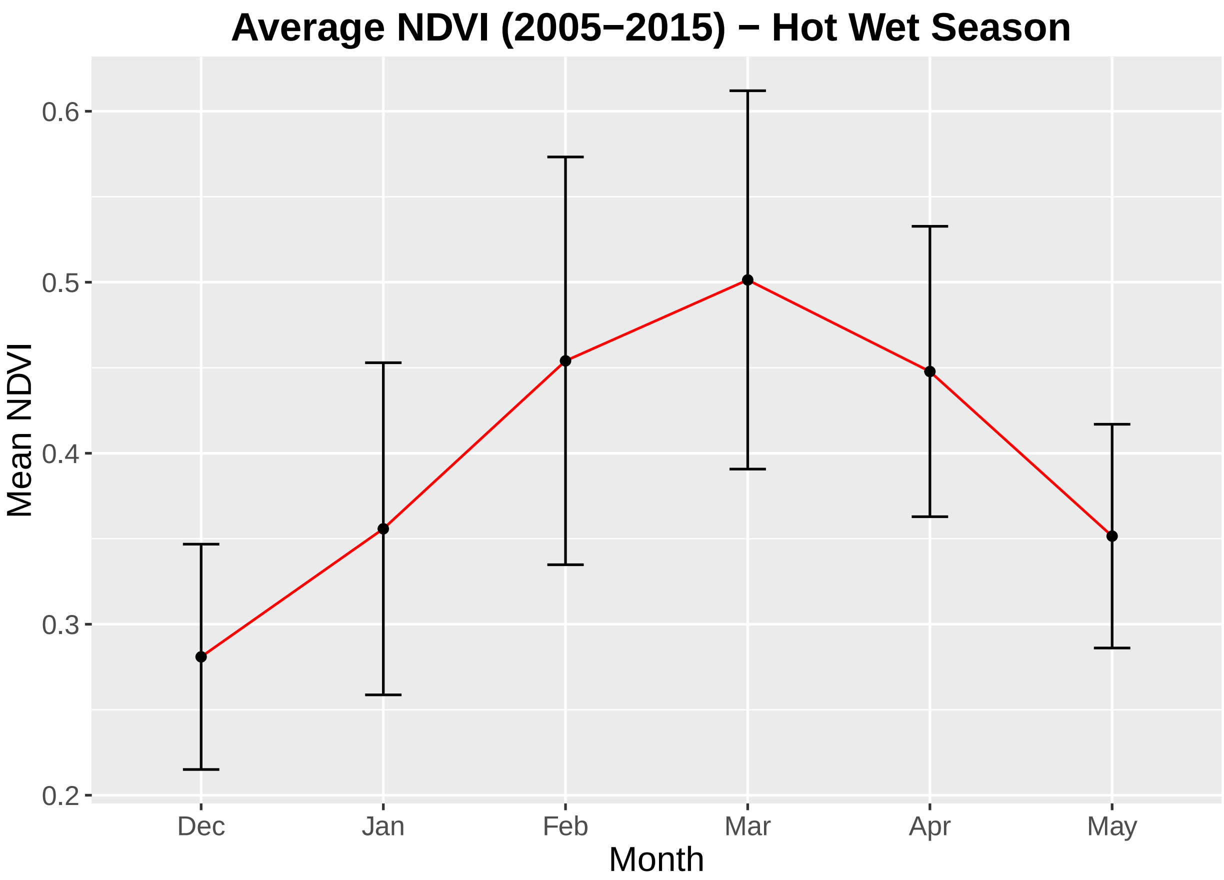


```
ggsave(file="Otjiwarongo_Monthly_NDVI.pdf", width = 297, height = 210, units =
         "mm")
```

#### Start of indivindual seasonal plots

Plot HOT-WET season and save as .pdf

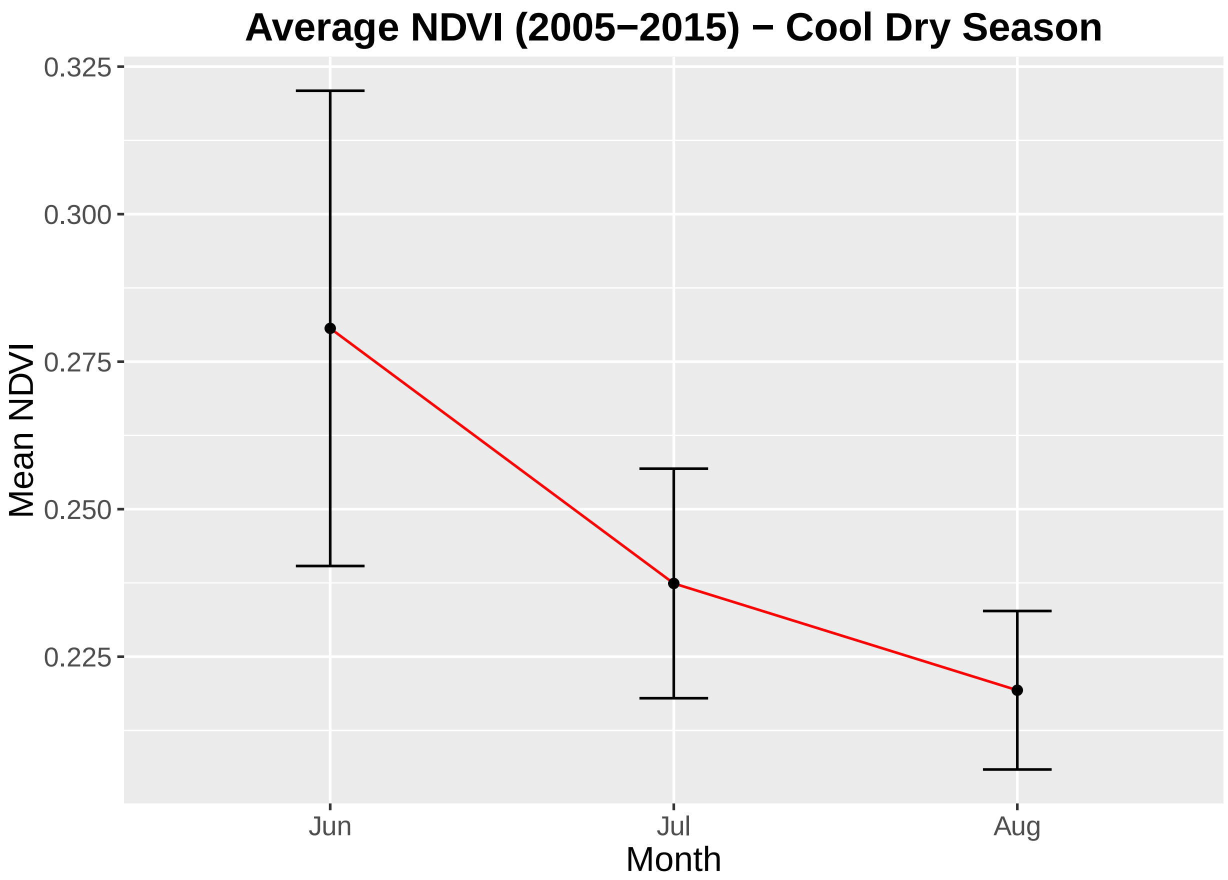
```
#plot HOT-WET season and save as .pdf
Hot.Wet <- subset(month.summary, month >= 12 | month <= 5)
target <- c("12", "5", "4", "3", "2", "1")
Hot.Wet<-Hot.Wet[match(target, Hot.Wet$month),]
Hot.Wet$month_name <- factor(Hot.Wet$month_name, c("Dec", "Jan", "Feb", "Mar",
                                                   "Apr", "May"))
Hot.Wet.NDVI.Plot<-ggplot(Hot.Wet, aes(month_name, mean.NDVI, group=1)) +
  geom_line(colour="red") +
  geom_errorbar(aes(ymin=mean.NDVI-sd_NDVI, ymax=mean.NDVI+sd_NDVI),width=0.2) +
  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
```



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

Plot COOL-DRY season and save as .pdf

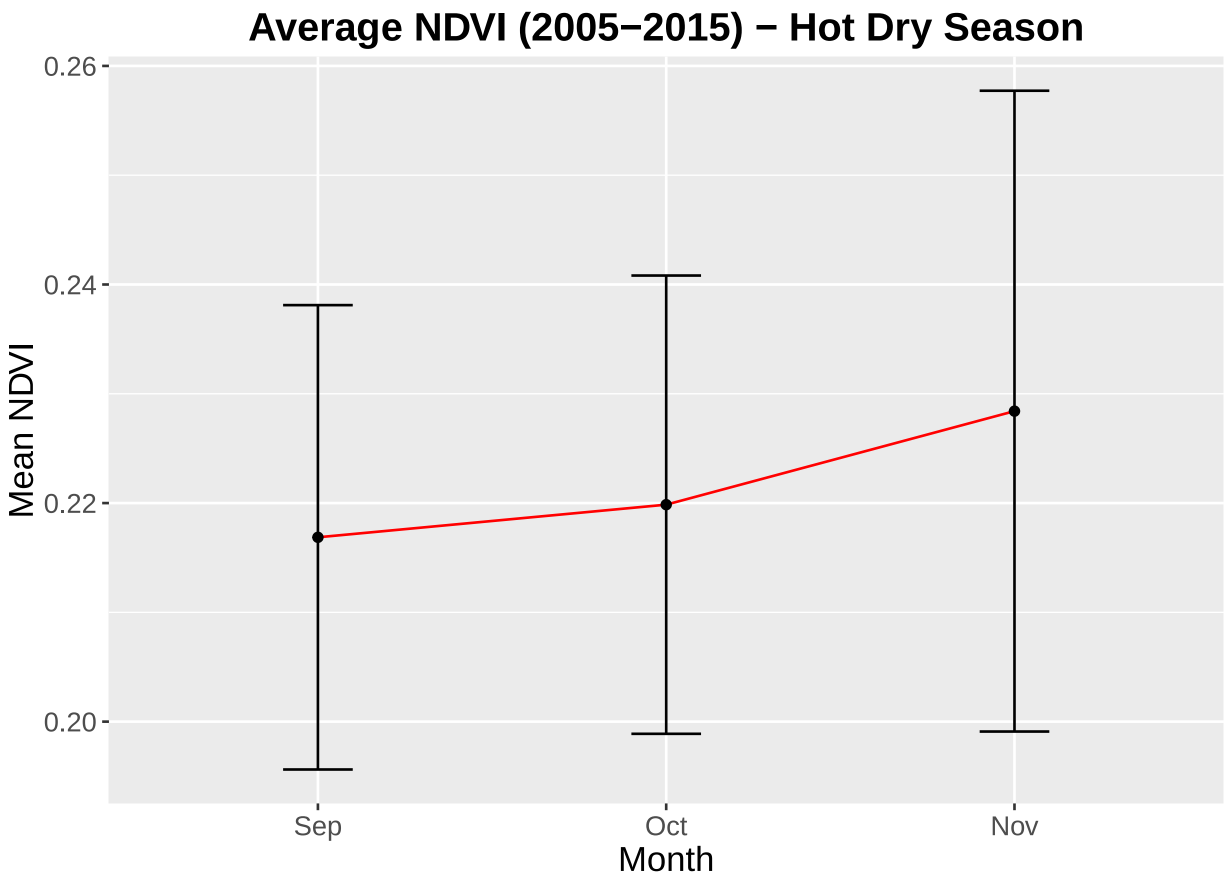
```
#plot COOL-DRY season and save as .pdf
Cool_Dry <- subset(month.summary, month <= 8 & month >= 6)
Cool_Dry_NDVI_Plot<-ggplot(Cool_Dry, aes(month_name, mean.NDVI, group=1)) +
  geom_line(colour="red") +
  geom_errorbar(aes(ymin=mean.NDVI-sd_NDVI, ymax=mean.NDVI+sd_NDVI),width=0.2) +
  ggtitle("Average 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
```



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

Plot HOT DRY Season

```
#plot HOT DRY season
Hot.Dry <- subset(month.summary, month <= 11 & month >= 9)
Hot.Dry.NDVI.Plot<-ggplot(Hot.Dry, aes(month_name, mean.NDVI, group=1)) +
  geom_line(colour="red") +
  geom_errorbar(aes(ymin=mean.NDVI-sd_NDVI, ymax=mean.NDVI+sd_NDVI),width=0.2) +
  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
```



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

**Start of merged seasonal plots**

Let's aggregate data by season

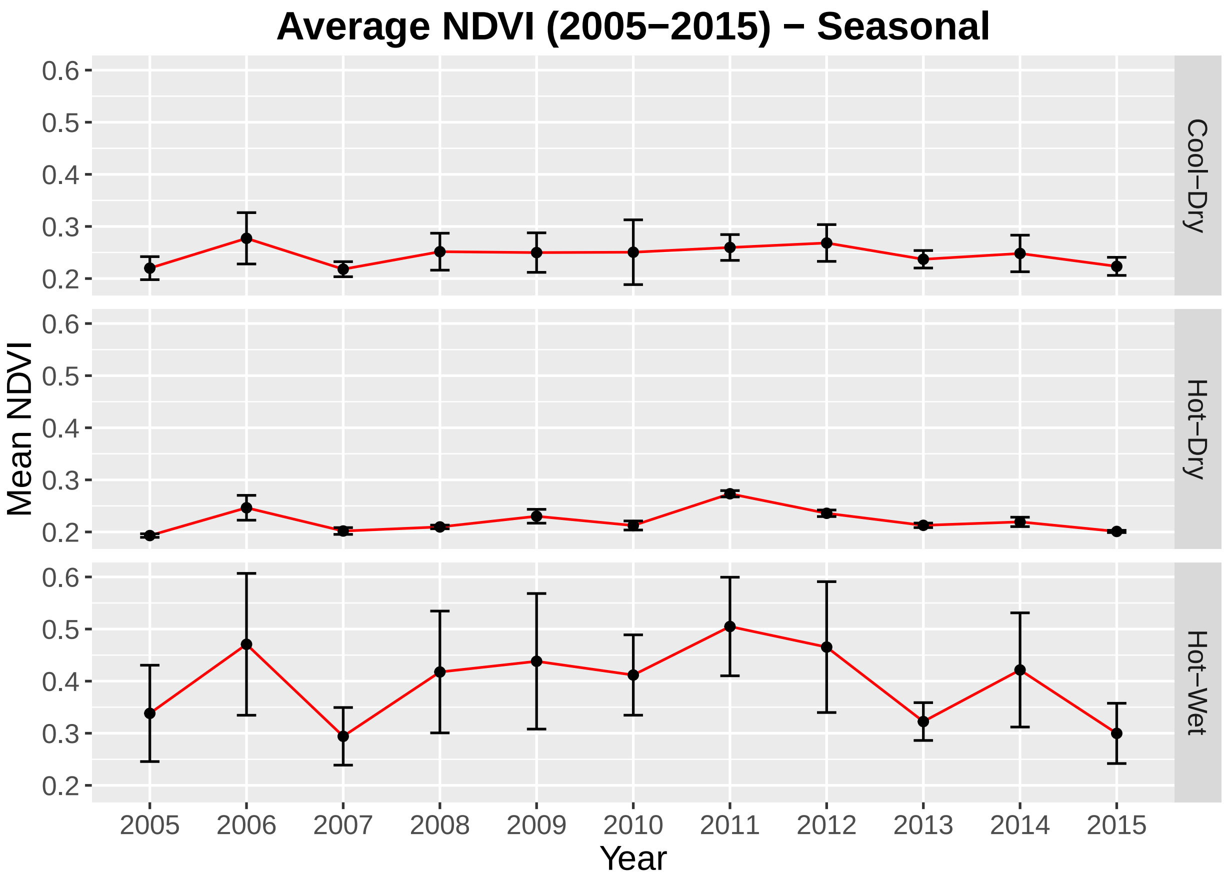
```
#aggregate data by season
season.NDVI<-avg.NDVI %>%
  group_by(year, season) %>%
  summarise(season.NDVI=mean(NDVI, na.rm=TRUE),
            sd_NDVI = sd(NDVI), #standard deviation of each group
            n_NDVI = n(),  #sample size per group
            SE_NDVI = sd(NDVI)/sqrt(n())) #standard error of each group)
```

Export comma separated file of seasonal NDVI values

```
#export .csv file of seasonal NDVI values
write.csv(season.NDVI, file="Seasonal_NDVI_2005-2015.csv")
```

Plot data by season and save as .pdf

```
#plot data by season and save as .pdf
Seasonal.NDVI.Plot<-ggplot(season.NDVI, aes(year, season.NDVI, group=3)) +
  geom_line(colour="red") +
  geom_errorbar(aes(ymin=season.NDVI-sd_NDVI, ymax=season.NDVI+sd_NDVI),width=0.2) +
  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()
Seasonal.NDVI.Plot + facet_grid(season ~ .)
```



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

End of exercise.

```
#end of exercise
```

“‘