

# Chapter 3: Visualization

*Chris Holden*

03/24/2015

R's default and **raster** plotting utilities make it very easy to explore your data. One of the earliest exploratory efforts in remote sensing using Landsat imagery was the development of the "Tasseled Cap" transformation by Kauth and Thomas. By applying data reduction technique (PCA, basically) for agricultural training sites they hand picked, Kauth and Thomas came up with a series of linear transformations that turn Landsat bands into physically meaningful indices. In more modern times, these "Tasseled Cap" transformations have been referred to as the "Brightness", "Greenness", and "Wetness" transforms for the physical meanings described.

Read in our data as normal:

```
library(raster)

## Loading required package: sp

if (file.exists('LE70220492002106EDC00_stack.gtif') == F) {
  download.file(url='https://raw.githubusercontent.com/ceholden/open-geo-tutorial/master/example/LE70220492002106EDC00_stack.gtif',
                destfile='LE70220492002106EDC00_stack.gtif', method='curl')
}

le7 <- brick('LE70220492002106EDC00_stack.gtif')
le7

## class       : RasterBrick
## dimensions : 250, 250, 62500, 8  (nrow, ncol, ncell, nlayers)
## resolution  : 30, 30  (x, y)
## extent      : 462405, 469905, 1734315, 1741815  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=15 +datum=WGS84 +units=m +no_defs
## data source : /home/ceholden/Documents/open-geo-tutorial/R/LE70220492002106EDC00_stack.gtif
## names       : band.1.reflectance, band.2.reflectance, band.3.reflectance, band.4.reflectance, band.5.reflectance, band.6.reflectance, band.7.reflectance, band.8.reflectance
## min values  : -32768, -32768, -32768, -32768,
## max values  : 32767, 32767, 32767, 32767,
```

## Coefficients

The coefficients required to transform Landsat surface reflectance data into these "Tasseled Cap" indices are given by Crist (1985) and Huang *et al* (2002):

```
refB = c(0.2043, 0.4158, 0.5524, 0.5741, 0.3124, 0.2330) # for Landsat 4/5
refB = c(0.3561, 0.3972, 0.3904, 0.6966, 0.2286, 0.1596) # for Landsat 7

refG = c(-0.1603, -0.2189, -0.4934, 0.7940, -0.0002, -0.1446) # for Landsat 4/5
refG = c(-0.3344, -0.3544, -0.4556, 0.6966, -0.0242, -0.2630) # for Landsat 7

refW = c(0.0315, 0.2021, 0.3102, 0.1954, -0.6806, -0.6109) # for Landsat 4/5
refW = c(0.2626, 0.2141, 0.0926, 0.0656, -0.7629, -0.5388) # for Landsat 7
```

We will ignore the last two bands in this Landsat dataset. The second to last band is the thermal band which measures brightness temperature. The last band is ignored because it is a mask band that separates clear observations of land and water from clouds, cloud shadows, and snow.

```
le7_data <- stack(le7, layers=names(le7)[seq(1, 6)])
```

Calculate brightness and wetness:

```
calc_bgw <- function(r) {
  b <- r[[1]] * refB[1] + r[[2]] * refB[2] + r[[3]] * refB[3] + r[[4]] * refB[4] + r[[5]] * refB[5] +
  g <- r[[1]] * refG[1] + r[[2]] * refG[2] + r[[3]] * refG[3] + r[[4]] * refG[4] + r[[5]] * refG[5] +
  w <- r[[1]] * refW[1] + r[[2]] * refW[2] + r[[3]] * refW[3] + r[[4]] * refW[4] + r[[5]] * refW[5] +
  bgw <- stack(b, g, w)
  names(bgw) <- c('brightness', 'greenness', 'wetness')

  return(bgw)
}

bgw <- calc_bgw(le7_data)
bgw

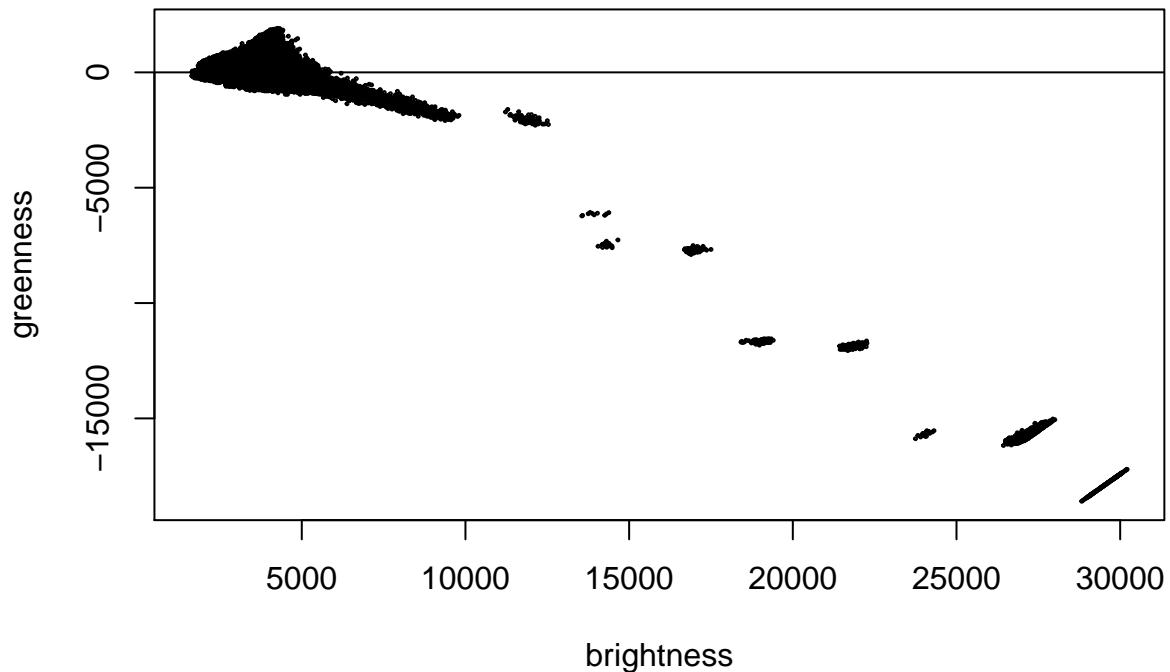
## class      : RasterStack
## dimensions : 250, 250, 62500, 3  (nrow, ncol, ncell, nlayers)
## resolution : 30, 30  (x, y)
## extent     : 462405, 469905, 1734315, 1741815  (xmin, xmax, ymin, ymax)
## coord. ref. : +proj=utm +zone=15 +datum=WGS84 +units=m +no_defs
## names       : brightness, greenness, wetness
## min values  : 1641.928, -18590.860, -12476.826
## max values  : 30207.195, 1907.556, 3490.413
```

## Tasseled Cap

We can now try to reproduce some of the “Tasseled Cap” plots:

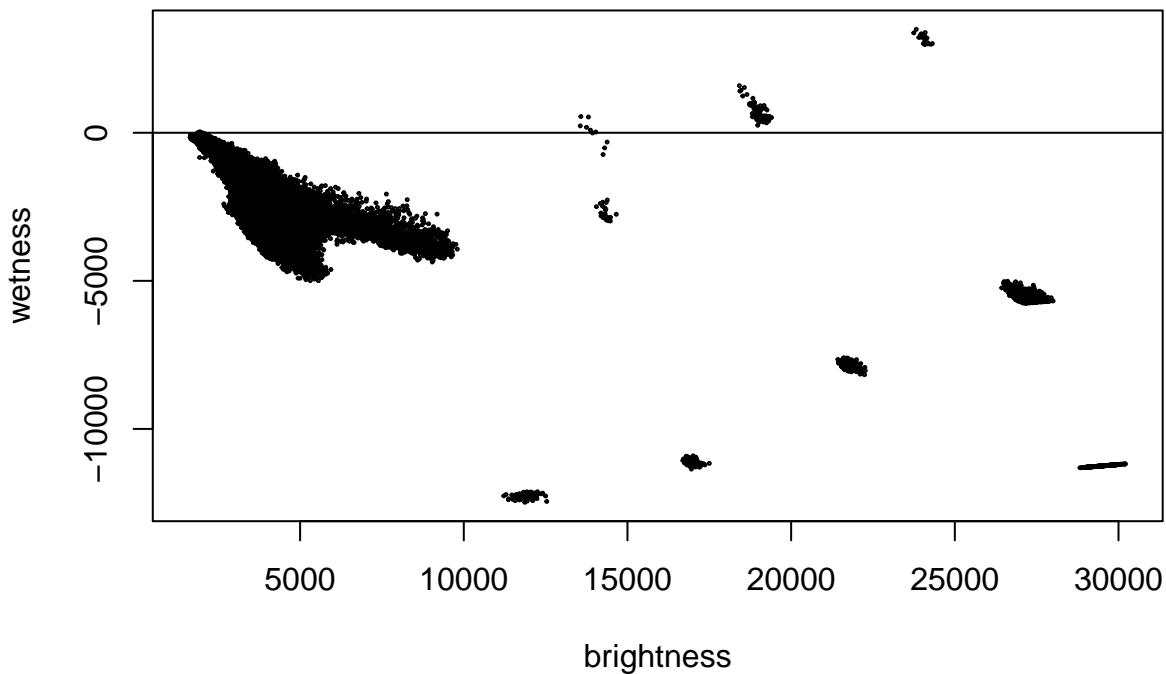
```
plot(bgw$brightness, bgw$greenness, main="Figure 1: Greenness ~ Brightness")
abline(v=0)
abline(h=0)
```

**Figure 1: Greenness ~ Brightness**



```
plot(bgw$brightness, bgw$wetness, main="Figure 2: Wetness ~ Brightness")
abline(v=0)
abline(h=0)
```

**Figure 2: Wetness ~ Brightness**



If you squint you might be able to make out the shape of a tasseled cap...

## Fmask

One of the possible sources of error in these plots comes from the fact that our dataset very well might contain clouds or cloud shadows. There is also a river in the image that we probably do not want to include. One of the newly available standard products from the USGS is a cloud mask from the Fmask (or CFmask, referring to the USGS C-code port of Fmask) by Zhe Zhu (2012) (some of you may remember him from BU).

The values in Fmask are as follows:

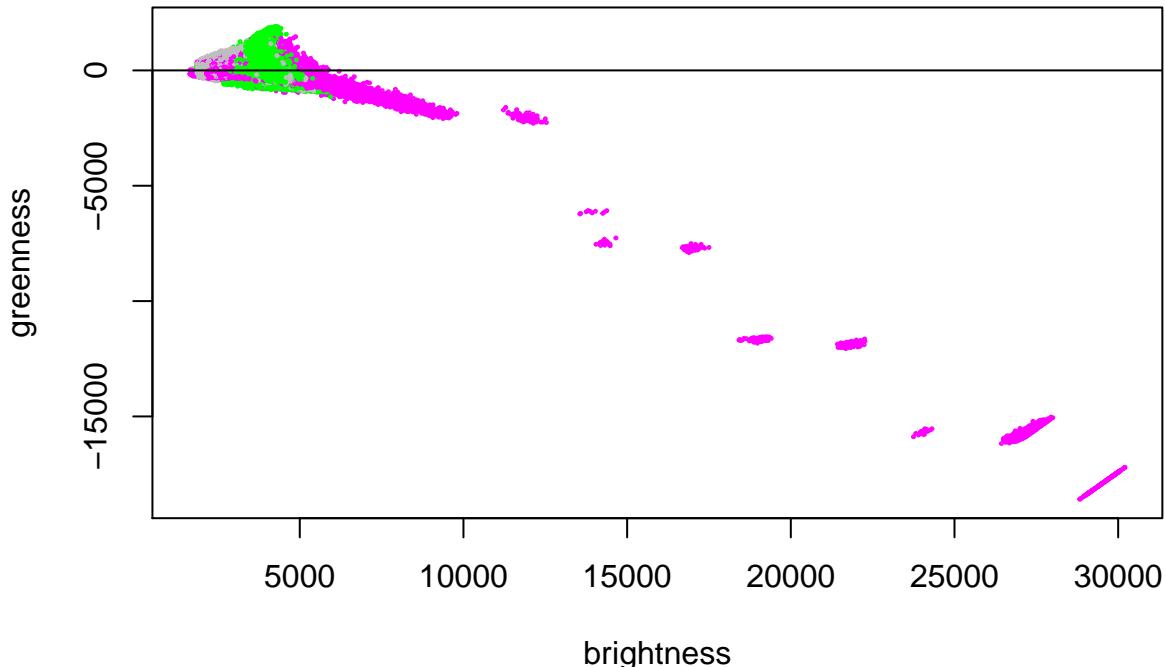
Value	Description
0	Clear land
1	Clear water
2	Shadow
3	Snow
4	Cloud
255	Fill

We can identify these values in our plot and recreate it:

```
fmask <- values(1e7[[8]])
fmask_col <- rep('green', length(fmask))
fmask_col[which(fmask == 1)] <- 'blue'
fmask_col[which(fmask == 2)] <- 'grey'
fmask_col[which(fmask == 4)] <- 'magenta'

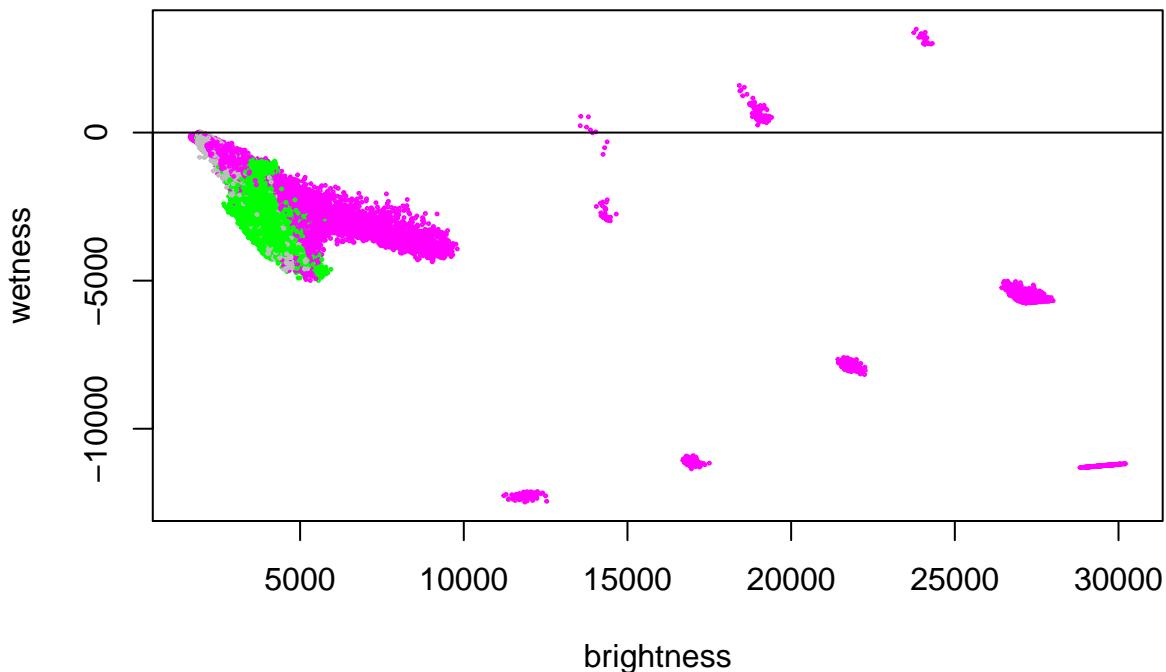
plot(bgw$brightness, bgw$greenness, main="Figure 1: Greenness ~ Brightness", col=fmask_col)
abline(v=0)
abline(h=0)
```

**Figure 1: Greenness ~ Brightness**



```
plot(bgw$brightness, bgw$wetness, main="Figure 2: Wetness ~ Brightness", col=fmask_col)
abline(v=0)
abline(h=0)
```

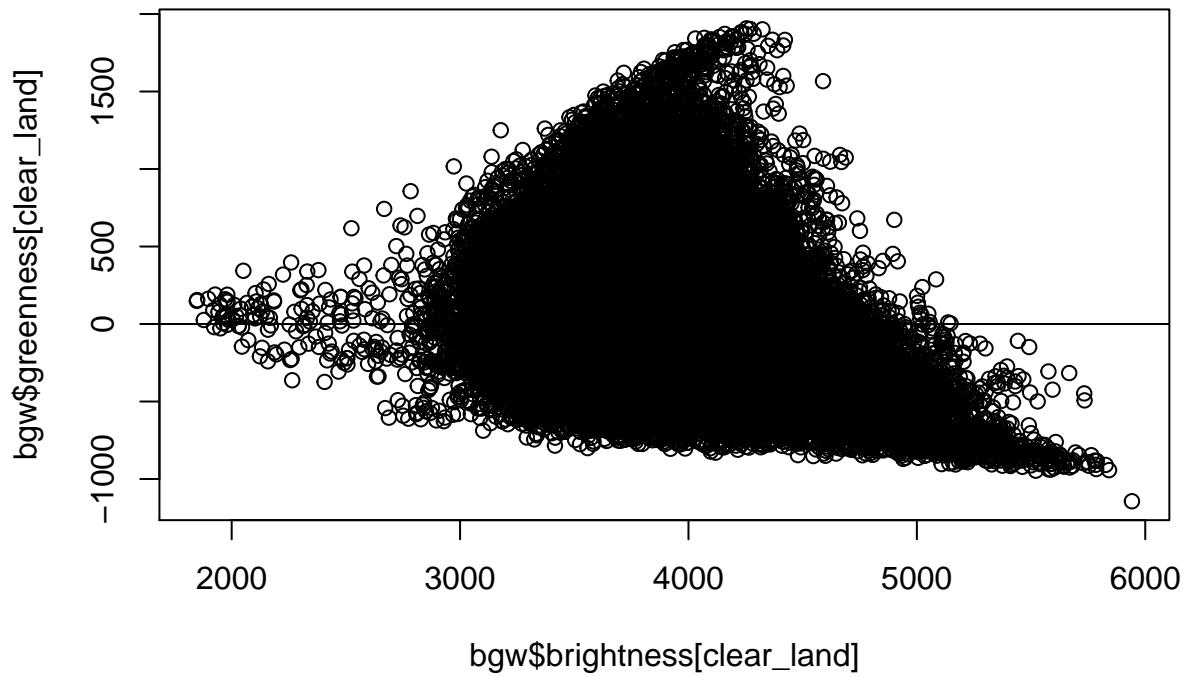
**Figure 2: Wetness ~ Brightness**



Looks like identified the problem. Now, to remove it:

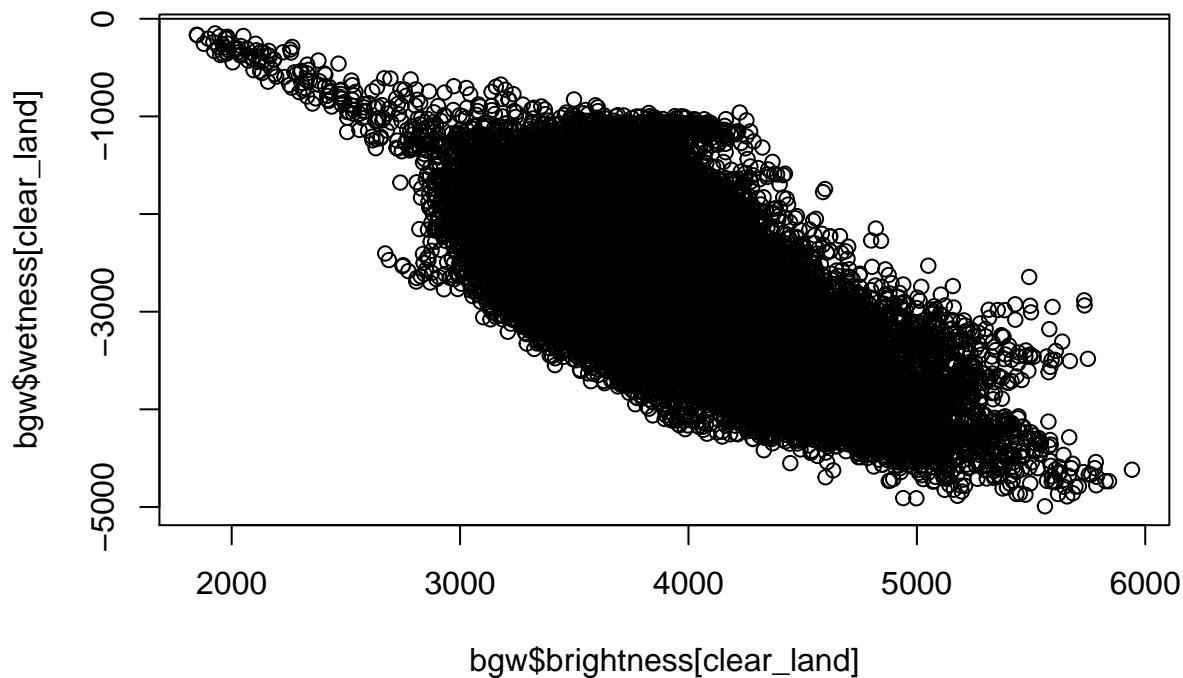
```
clear_land <- which(fmask == 0)
plot(bgw$brightness[clear_land], bgw$greenness[clear_land], main="Figure 1: Greenness ~ Brightness")
abline(v=0)
abline(h=0)
```

**Figure 1: Greenness ~ Brightness**



```
plot(bgw$brightness[clear_land], bgw$wetness[clear_land], main="Figure 2: Wetness ~ Brightness")
abline(v=0)
abline(h=0)
```

**Figure 2: Wetness ~ Brightness**



Wow! Clouds, and especially cloud shadows, are a dreadfully persistent fact of life with remote sensing.

Fortunately, some poor soul spent many many years creating the wonderful community tool – Fmask. It's not perfect – it does consistently miss some very difficult types of clouds and the shadow matching doesn't always work – but it easily removes 90%+ of the bad observations. More advanced methods based on timeseries (Zhu *et al* 2015) have also been developed to further screen residual noise. Another alternative is to use an image composite approach to retain only the best available observations, usually at a yearly interval. Examples of this include LandTrendr (Kennedy *et al* 2010) and the recently published global forest change map (Hansen *et al* 2013).