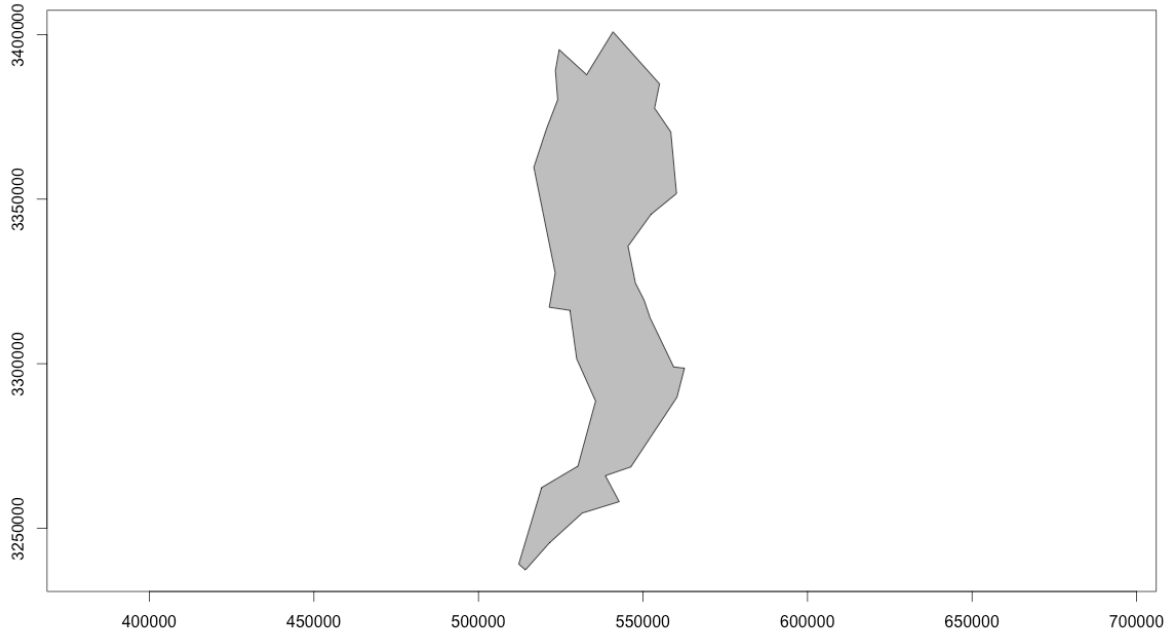
 # después de este símbolo son comentarios
= ó <- ambos quieren decir "igual a"

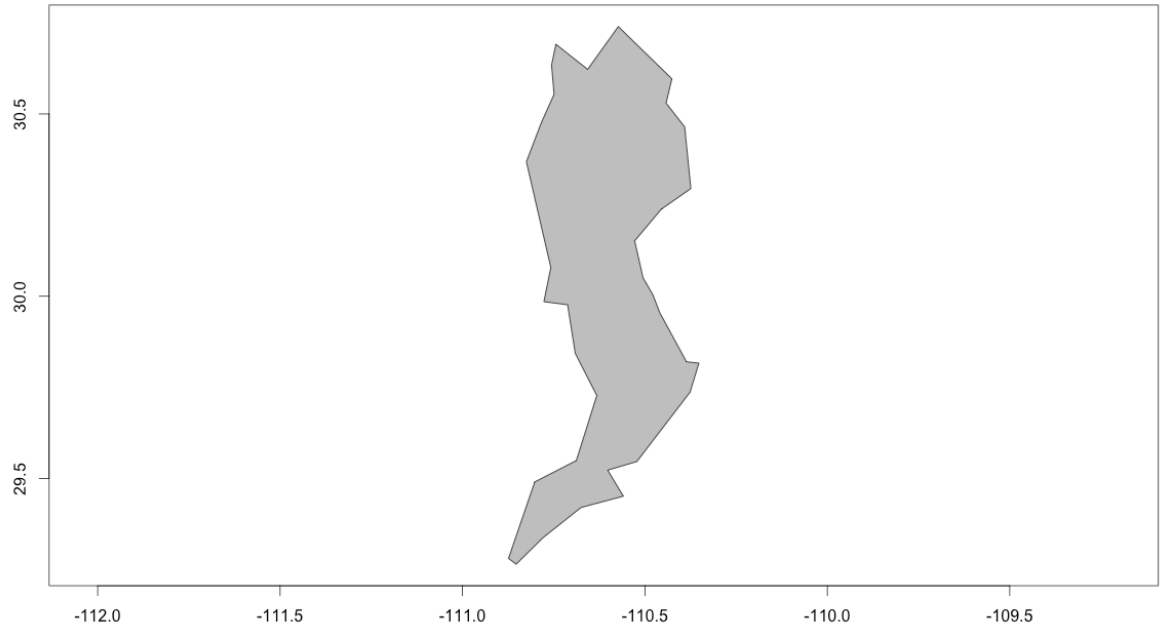
Sensible a mayúsculas por ejemplo:
"B1" no es igual a "b1"

```
# Caso de Acuifero del Rio San Miguel tengo puntos en Longitud y Latitud, cada uno subdividido en grados,
#  minutos y segundos, quiero integrarlos en un solo valor en grados con decimales.
# fijo directorio de trabajo
setwd('/Users/COLSON/Google Drive/AlanFiles/San Miguel de Horcasitas/Bases de Datos')
getwd()
list.files() # lista de archivos en el directorio
options(digits=6) # para que NO redondee y conserve 6 digitos
acuifero<- read.csv("acuif_SM.csv",header=T,na.strings= c('NA', -99), quote="\"",dec="." ) # importo datos
head(acuifero) # visualizar encabezado de tabla
acuifero$Lon<-as.numeric(0) # creo variable nueva,me aseguro que la longitud (variable Lon) sea numerica
acuifero$Lat<-as.numeric(0) # igual para Lat
names(acuifero)
# transformo a grados decimales
acuifero$Lon<-(signif(acuifero[,2] + round((acuifero[,3]+(round(acuifero[,4]/60,6))),6)/60,digits=6))
acuifero$Lat<-signif(acuifero[,5] + round((acuifero[,6]+(round(acuifero[,7]/60,6))),6)/60,digits=6)
acuifero$Lon<-acuifero$Lon*-1 # hago la Longitud negativa multiplicandola por -1
head(acuifero) #checo el objeto
acuifero$Lon[1] # checo el primer valor de la variable "Lon"
write.csv(acuifero, file = "acuif_SM2.csv") #exporto la tabla a un formato que lee Excel
################################
tail(acuifero)
dim(acuifero)
acuifero<- acuifero[-c(34), ] #elimino la ultima linea
################################
library(sp)  # vector data
library(raster)  # raster data
library(rgdal)  # input/output, projections
library(rgeos)  # geometry ops
library(spdep)  # spatial dependence
library(maptools)
###### CREO DOS VECTORES CON LAS COORDENADAS
x1=acuifero$Lon
y1= acuifero$Lat
##### Hago el poligono
c1 = cbind(x1, y1)
r1 = rbind(c1, c1[1, ])
P1 = Polygon(r1)
Ps1 = Polygons(list(P1), ID = "a")
# Spatial Polygons Data Frame
SPs = SpatialPolygons(list(Ps1))
### esta funcion ya la usamos antes, pero con puntos
poligono_1 = SpatialPolygonsDataFrame(SPs, data.frame(Nombre = c("Poligono"), row.names = c("a")))
#######################################
plot(poligono_1, axes = TRUE, col="grey")
#######################################
### Proyectamos definimos la proyeccion
WGS84_Datum_GCS=CRS("+init=epsg:4326")
projection(poligono_1) <- WGS84_Datum_GCS
print(proj4string(poligono_1))
# Fijamos el directorio de trabajo
setwd("/Users/COLSON/Documents/Datos SIG")
# write out a new shapefile (including .prj component)
writeOGR(poligono_1, ".", "Acuifero_SanMiguel_GCS_WGS84", driver="ESRI Shapefile")
###########
### pero hay un problema, mi archivo poligono_1 esta en UTM!
poligono_1UTM<-spTransform(poligono_1, CRS("+init=epsg:32612"))
#### lo visualizamos 3 busca colores en Google "R colors"
#######################################
plot(poligono_1UTM, axes = TRUE, col="grey")
#######################################
# write out a new shapefile (including .prj component)
writeOGR(poligono_1UTM, ".", "Acuifero_SanMiguel_UTM_WGS84", driver="ESRI Shapefile")
#######################################
# or for Google Earth KMZ, una vez en Google Earth lo puedes exportar como KML
KML(poligono_1, "Acuifero_SanMiguel.kmz")
```

la función setwd ("ruta")
fijas dierectorio de trabajo
para exportar o importar archivos
getwd ( ) te devuelve el directorio de trabajo actual

dataframe es una "tabla" con columnas e hileras,
la primera línea es de nombres de las variables y
las hileras serían observaciones:
la función
names(nombredeldataframe)
para ver el nombre de las variables que incluye
Para ver los datos de una columna:
nombredeldataframe$nombredelavariable
Para ver el segundo dato de una columna
nombredeldataframe$nombredelavariable[2]
Las funciones:
head(nombredeldataframe)
tail(nombredeldataframe)
te permiten ver las primeras y últimas 6 líneas
dim(nombredeldataframe)
la dimensión de la tabla en hileras y columnas


3400000
3350000
3300000
3250000
400000
450000
500000
550000
600000
650000
700000


30.5
30.0
29.5
-112.0
-111.5
-111.0
-110.5
-110.0
-109.5