

KMLとR

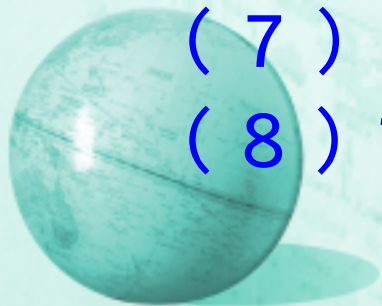
RでKMLを自由自在に操る

データキューブ
牧山 文彦



A g e n d a

- (1) K M L とGoogleEarth・・・おさらい
- (2) デジタルアースに必要な技術
- (3) K M L の立ち位置
- (4) K M L の 特 徴
- (5) R からのアプローチ
- (6) R パッケージの機能との関連
- (7) デ モ
- (8) ま と め



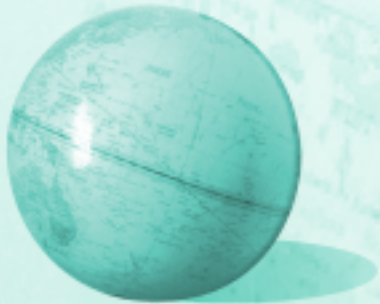
(1) K M L とGoogleEarth・・・おさらい

- K e y h o l e M a r k u p L a n g u a g e の 略、XMLの一種
- K e y h o l e とは、Googleに買収される前のGoogleEarthの名前から来ている
- 2004年10月に米Keyhole社がGoogleに買収された。米Keyhole社はSGI社(シリコングラフィックス社)を源流とする
- 2005年7月GoogleEarth公開
- GoogleEarthは「デジタルアース」というコンセプトに基づいて作成されたブラウザーの一種で、従来のGISとは概念を異にする
- GoogleEarthは、Googleが無料で配布するという暴挙(?)に出たため、世界に衝撃が走った
- そのため2004～2005年がひとつの出発点となっている
- 2008年4月、KMLはOGC標準に認定された
- 海・宇宙・月・火星・・・地中はまだない
- iPhoneに対応

(2) デジタルアースに必要な技術

1. コンピュータ科学
2. 大規模ストレージ
3. 衛星画像(特に高解像度の)
4. ブロードバンドネットワーク
5. 相互運用性
6. メタデータ

(アル・ゴア :2000年)



(3) K M Lの立ち位置

ArcGIS他

プロフェッショナル

スタンドアロン

Google Earth Pro

GoogleEarth

K M L

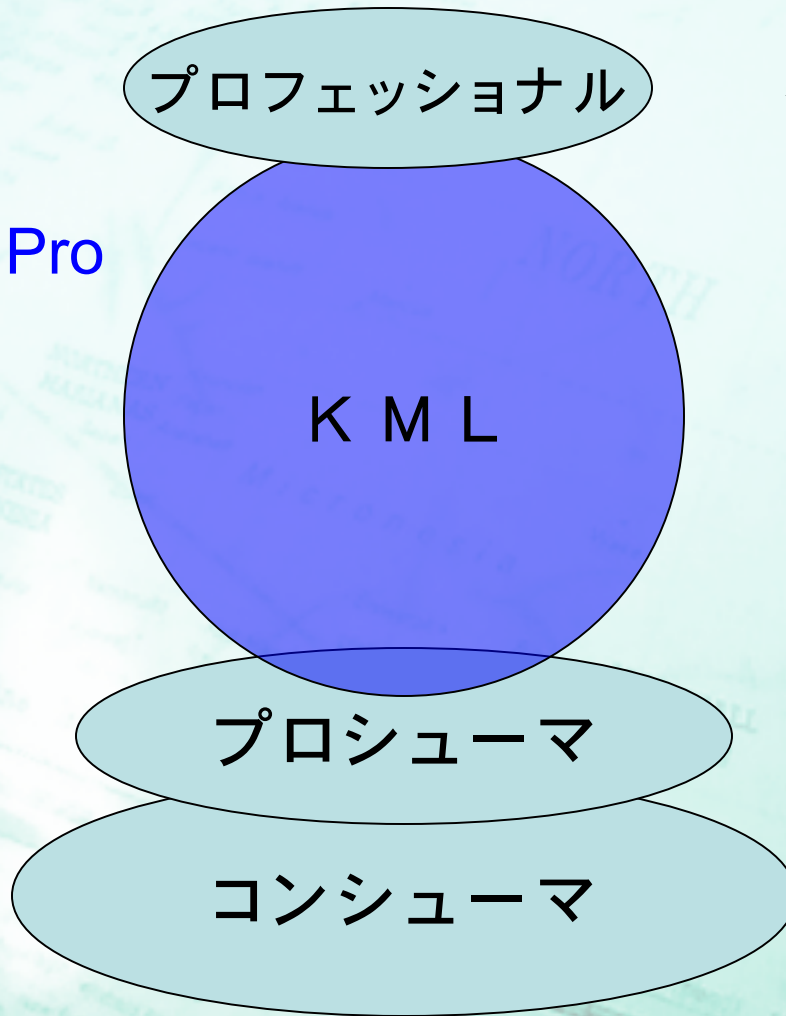
Earth Plug-In

プロシューマ

GoogleMap

コンシューマ

Webベース

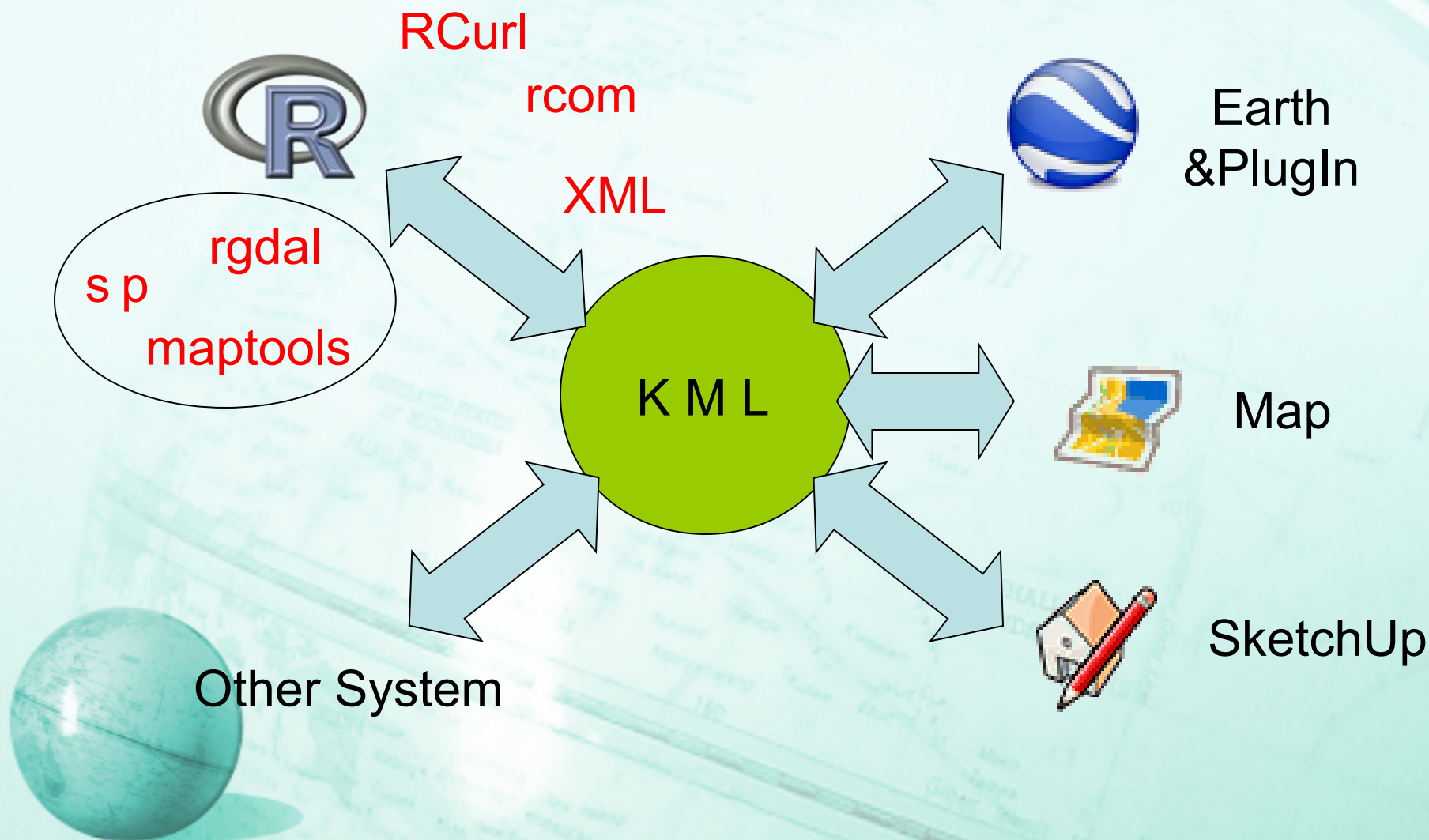


(4) K M L の 特 徴

- O G C K M Lとして標準化 現在ver 2 . 2
- X M L
- Z I P 圧 縮 可 能 (K M Z)
- ネットワークリンクに対応・・・分散型
- ベクターデータ(点・線・ポリゴン等)
- ラスターデータのオーバーレイ
- W M Sのマッピング
- 時間データ
- 写真データ(ストリートビュー含む)
- テクスチャ付き3DオブジェクトはColladaデータ
- ブラウザー機能(JavaScript&Webkit利用)
- Flash可能
- 動画再生可能
- サウンド再生可能



(5) Rからのアプローチ



(6) Rパッケージの機能との関連

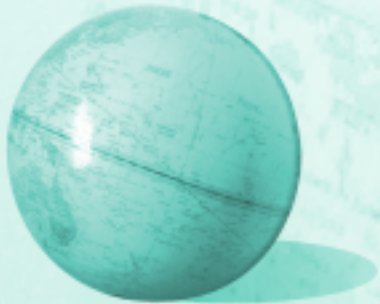
Rパッケージ	基本機能	Earth	Map	SketchUp
rgdal	データ変換	KML変換	KML変換	
maptools	GIS機能	KML出力	KML出力	
XML	XML操作	KML操作	XML操作	Collada操作
RCurl	Web操作		WebAPI 操作	
rcom	DCOM操作	Earth操作		

RでKMLを扱うためのクラスはspの3つのクラスを利用する

SpatialPointsDataFrame-class

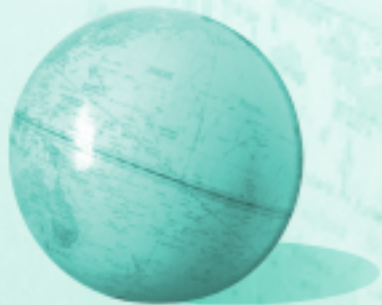
SpatialLinesDataFrame-class

SpatialPolygonsDataFrame-class



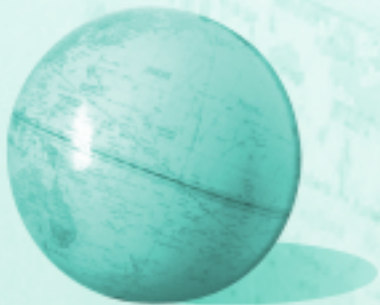
(7) デモ

- ①ShapeFile-KML変換(rgdal)
- ②ラスターデータのKMLオーバーレイ(maptools)
- ③ベクターデータのKMLオーバーレイ(rgdal)
- ④ポリゴンデータのKMLオーバーレイ(maptools)
- ⑤複数ポリゴンデータのKMLオーバーレイ1(maptools)
- ⑥KMLデータの修正(XML:Mac)
- ⑦Colladaデータの修正(XML:Mac)



デモ①ShapeFile-KML変換(rgdal)

```
#demo1
#rgdal WriteOGR
library(rgdal)
x<-readOGR("Z:/test/japan_ver62/gunma.shp",layer="gunma")
x@data$PREF<-iconv(x@data$PREF,"","utf-8")
x@data$CITY1<-iconv(x@data$CITY1,"","utf-8")
x@data$CITY2<-iconv(x@data$CITY2,"","utf-8")
x@data$TOWN1<-iconv(x@data$TOWN1,"","utf-8")
x@data$TOWN2<-iconv(x@data$TOWN2,"","utf-8")
writeOGR(x,"Z:/test/gunma.kml", "gunma", "KML")
```



Apple logo グラブ ファイル 編集 取り込み ウィンドウ ヘルプ

Google Earth

▼ 検索

ジャンプ ビジネスを検索 ルート

ジャンプ 例: サンフランシスコ

▼ 場所 コンテンツを追加

▼ お気に入り

▶ 観光
ツアーを開始するには、このフォルダを選択して下の [再生] ボタン

▼ 保留

▶ R image

▶ gunma.kml

▼ レイア

▼ プライマリ データベース

▶ ジオグラフィック ウェブ

▶ 道路

▶ 建物の 3D 表示

▶ ストリートビュー

▶ 境界線や地名

▶ 渋滞状況

▶ 天気

▶ ギャラリー

▶ 海

▶ グローバル アウェアネス

▶ スポット

▶ その他

▶ 地形

36°31'20.00" N 139°01'59.53" E 標高 0 メートル 高度 142.91 キロメートル

Image © 2009 DigitalGlobe
© 2009 Europa Technologies
Image © 2009 TerraMetrics
© 2009 ZENRIN

Google

群馬県

栃木県

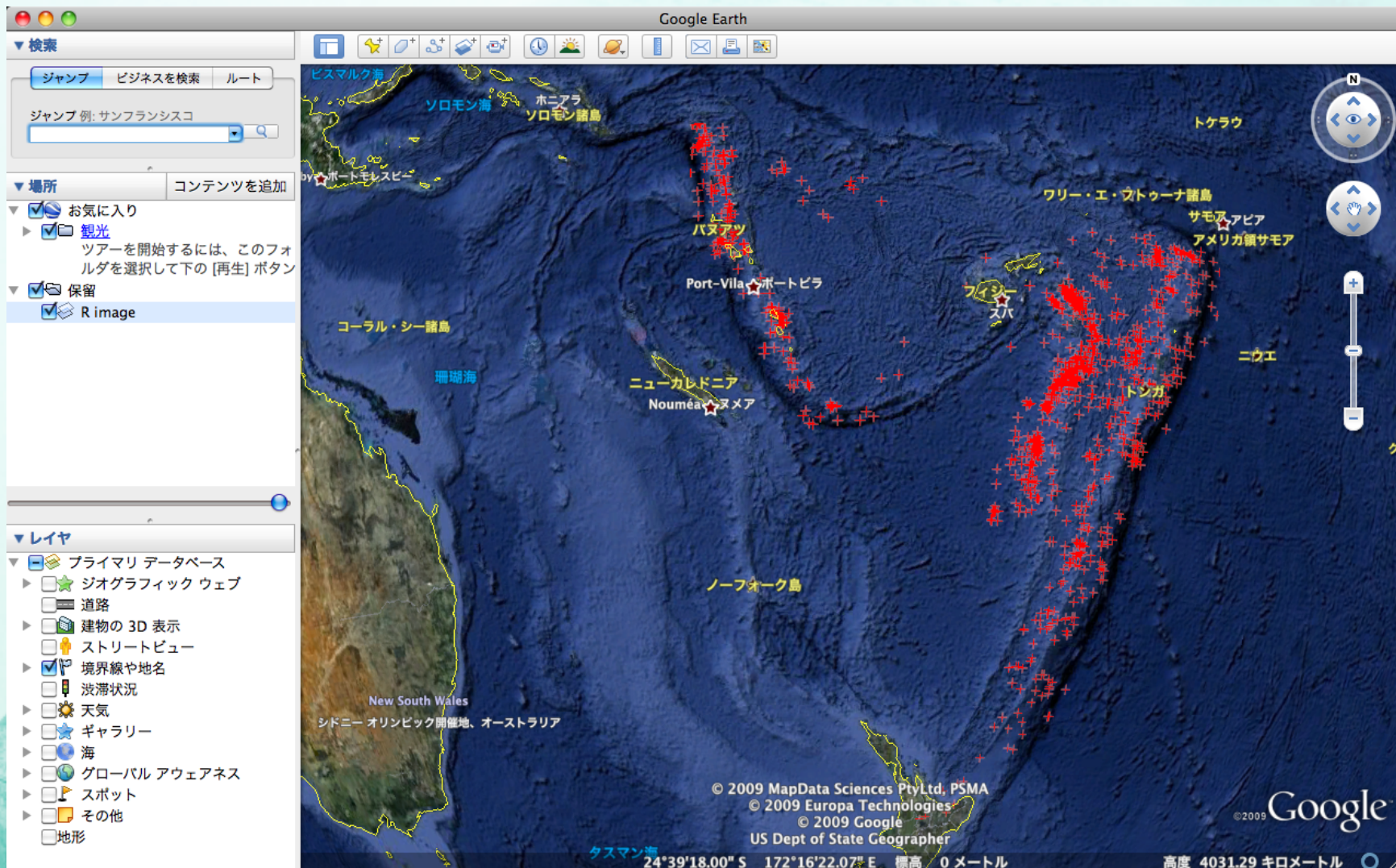
埼玉県

OBJECTID	567
PREF	群馬県
CITY1	川場村
CITY2	NA
TOWN1	利根郡
TOWN2	川場村
JCODE	10444
P_NUM	4203
H_NUM	967
FLAG1	1
FLAG2	0
Shape_Leng	0.475182895445
Shape_Area	0.0085295044025

②ラスターデータのKMLオーバーレイ(maptools)

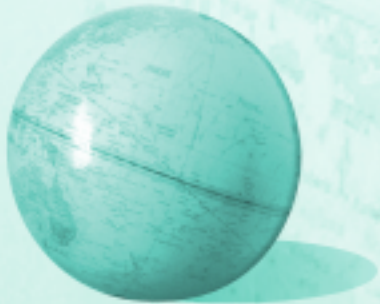
```
#demo2
#maptools kmlOverlay
#
library(maptools)
qk <- SpatialPointsDataFrame(quakes[, c(2:1)], quakes)
proj4string(qk) <- CRS("+proj=longlat")
tf<-"Z:/test/"
SGqk <- GE_SpatialGrid(qk)
png(file=paste(tf, "qk.png", sep=""), width=SGqk$width,
height=SGqk$height,bg="transparent")
par(mar=c(0,0,0,0), xaxs="i", yaxs="i")
plot(qk, xlim=SGqk$xlim, ylim=SGqk$ylim, setParUsrBB=TRUE,col="red")
dev.off()
kmlOverlay(SGqk, paste(tf, "qk.kml", sep=""), paste("qk.png", sep=""))
```

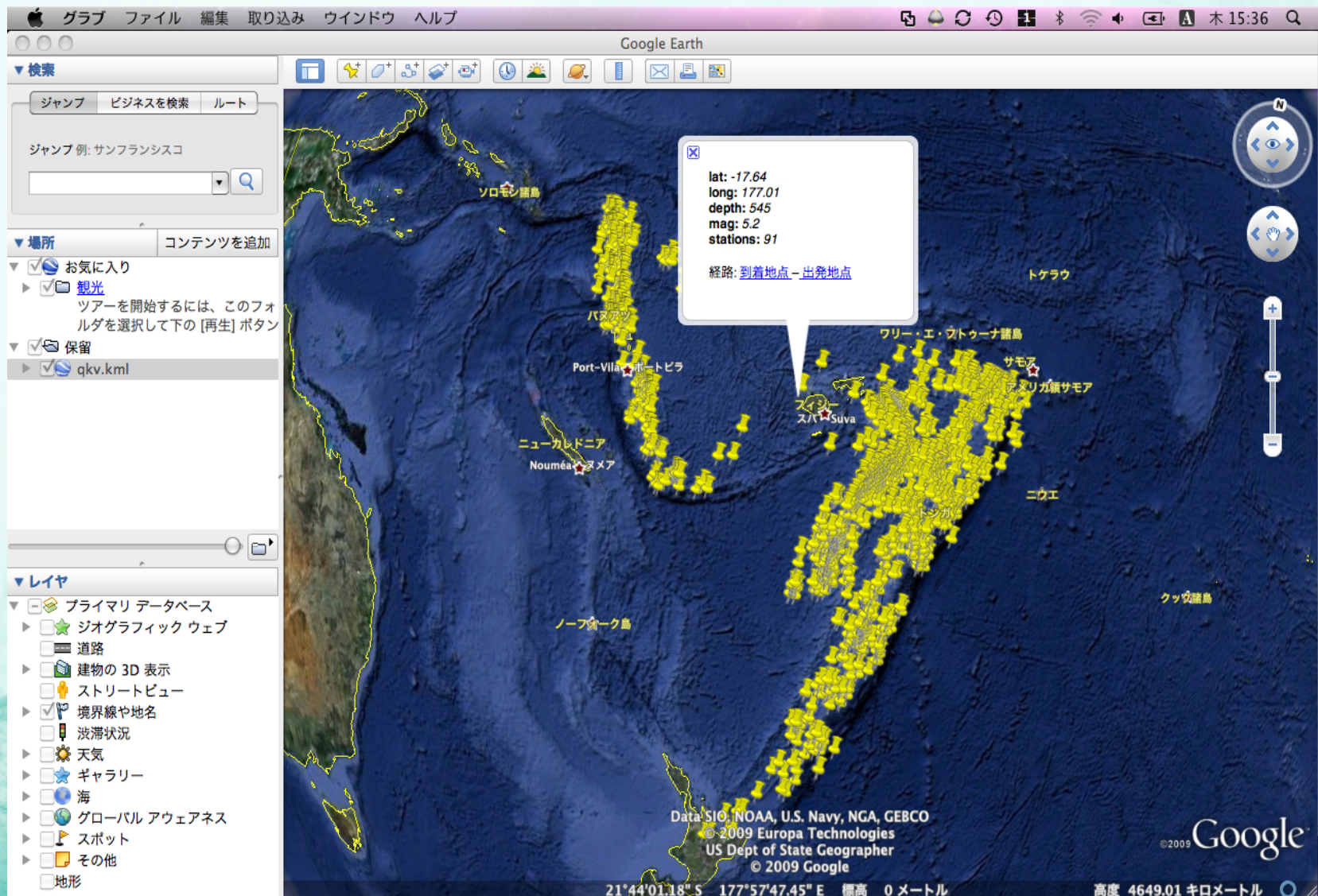




③ベクターデータのKMLオーバーレイ(rgdal)

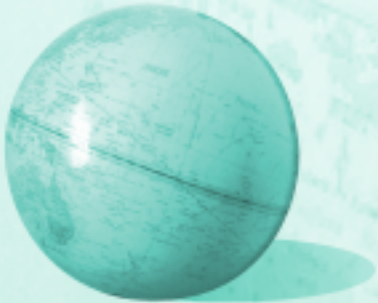
```
#demo3
#rgdal point overlay
library(rgdal)
qk0 <- quakes
qk0$long <- ifelse(qk0$long <= 180, qk0$long, qk0$long-360)
qk0a <- SpatialPointsDataFrame(qk0[, c(2:1)], qk0)
proj4string(qk0a) <- CRS("+proj=longlat")
writeOGR(qk0a, paste(tf, "v.kml", sep=""), "Quakes", "KML")
```

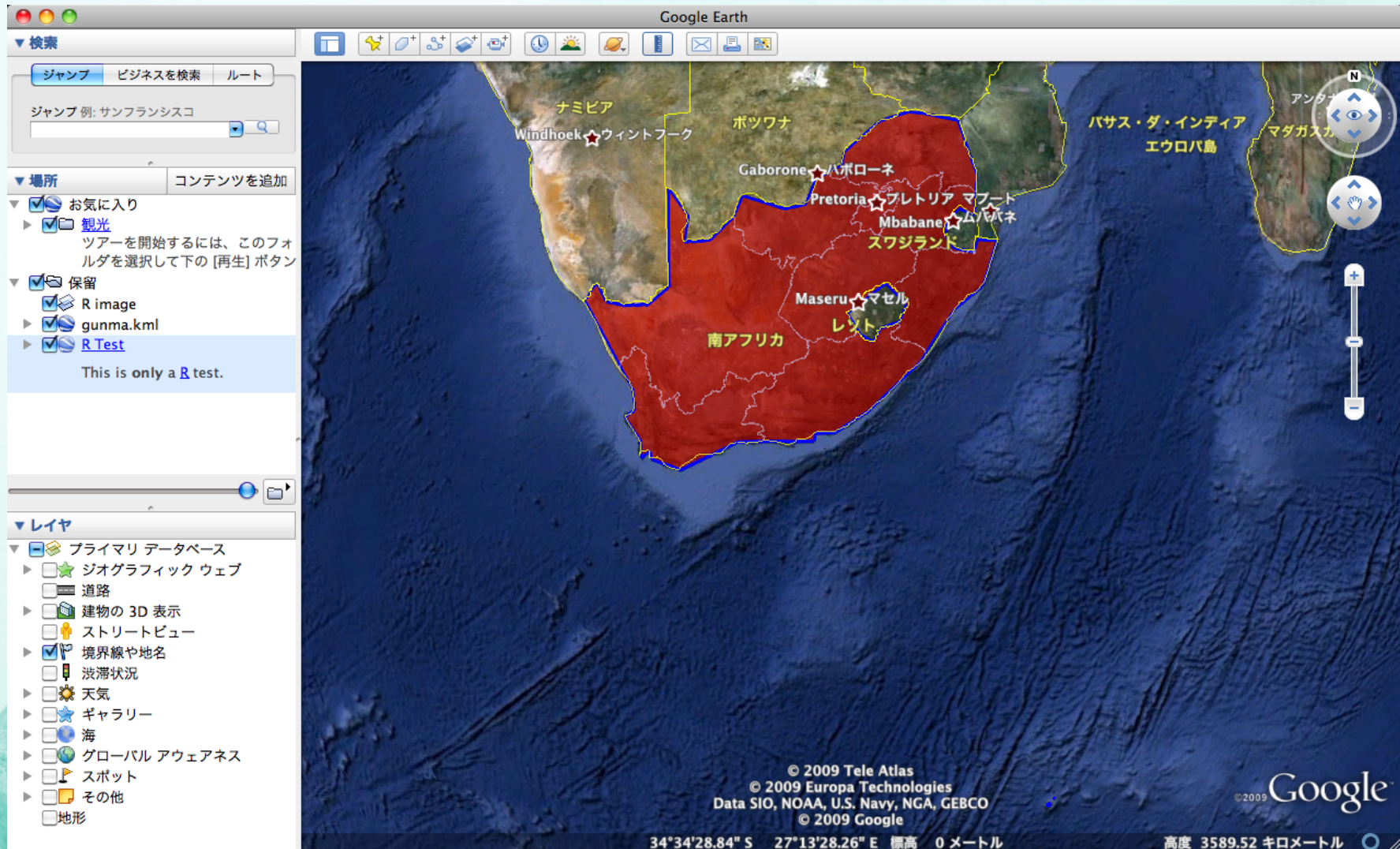




④ポリゴンデータのKMLオーバーレイ(maptools)

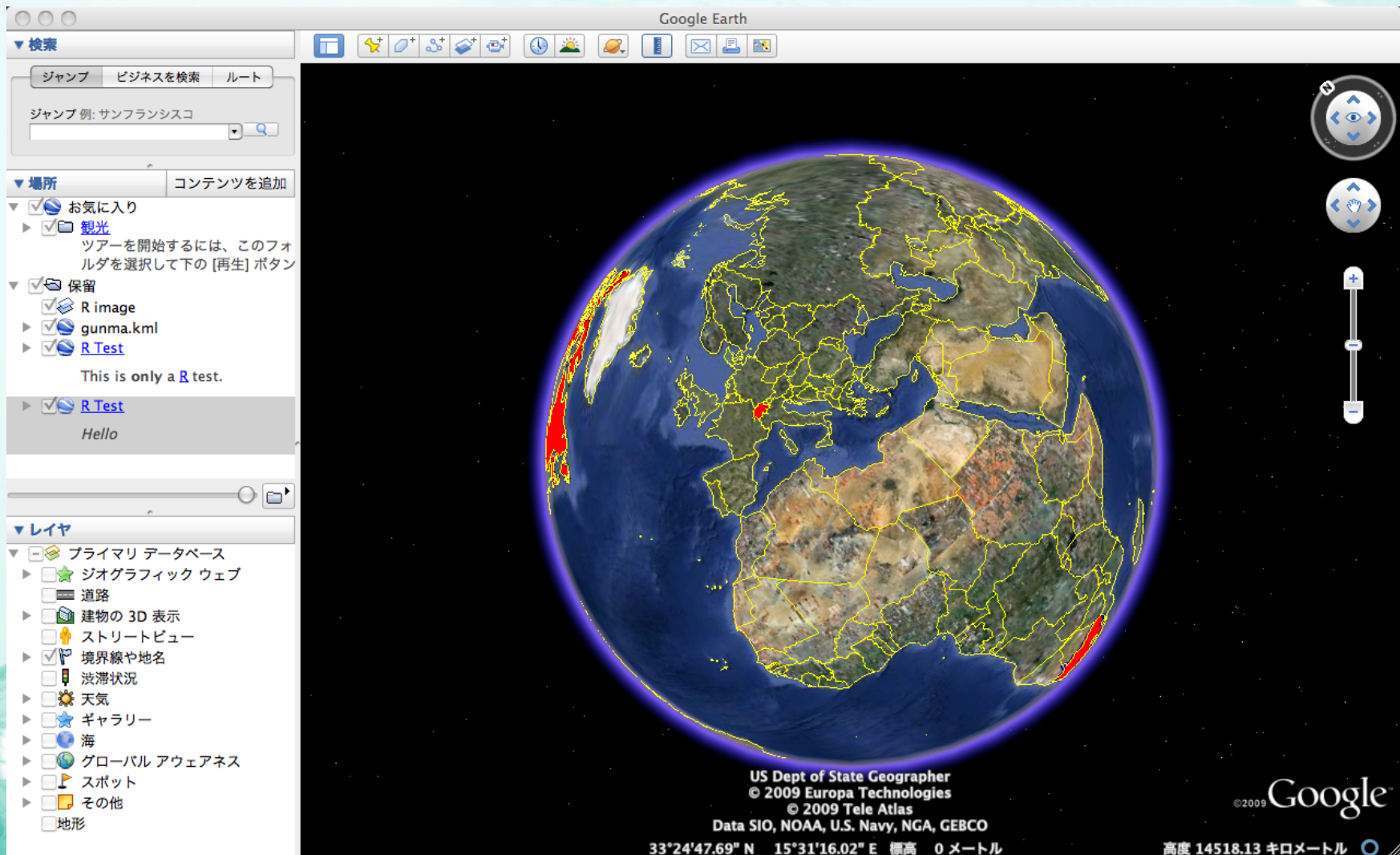
```
#demo4
library(maptools)
data(wrld_simpl)
## creates a KML file containing the polygons of South Africa (plus hole)
sw <- slot(wrld_simpl[wrld_simpl$NAME=="South Africa",], "polygons")[[1]]
tf <- "Z:/test/SA.kml"
kmlPolygon(sw, kmlfile=tf, name="South Africa", col="#df0000aa", lwd=5,
  border=4, kmlname="R Test",
  kmldescription="This is <b>only</b>
  a <a href='http://www.r-project.org'>R</a> test.")
tf
```





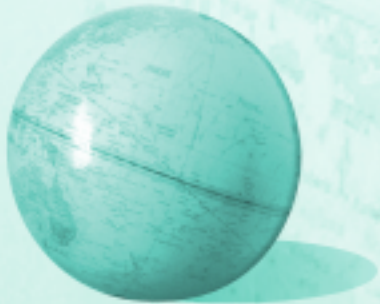
⑤複数ポリゴンデータのKMLオーバーレイ1(maptools)

```
#demo5
## creates a KML file containing the polygons
##           of South Africa, Switzerland, and Canada
sw  <- wrld_simpl[wrld_simpl$NAME %in% c("South Africa",
    "Switzerland", "Canada"),]
out <- sapply(slot(sw, "polygons"), function(x) { kmlPolygon(x,
    name=as(sw, "data.frame")[slot(x, "ID"), "NAME"],
    col="red", lwd=1.5, border='black',
    description=paste("ISO3:", slot(x, "ID"))) })
tf <- "Z:/test/SA2.kml"
kmlFile <- file(tf, "w")
tf
cat(kmlPolygon(kmlname="R Test", kmldescription="<i>Hello</i>")$header,
    file=kmlFile, sep="\n")
cat(unlist(out["style",]), file=kmlFile, sep="\n")
cat(unlist(out["content",]), file=kmlFile, sep="\n")
cat(kmlPolygon()$footer, file=kmlFile, sep="\n")
close(kmlFile)
```

⑥KMLデータの修正(XML:Mac)

```
#demo6
#XML MacOSX
library(XML)
source("/Users/fumihiko_maki/Desktop/test/kmlFunctions.r")
doc <- xmlTreeParse("/Users/fumihiko_maki/Desktop
/test/gunma.kml")
data<-sample(1000:10000,39)
fillcolor<-colors.kml(heat.colors(39),"EE")
linecolor<-NA
x<-kmlAltitudeNode(doc,data,fillcolor,linecolor)
saveXML(x,file="/Users/fumihiko_maki/Desktop/test/gunma2.kml",
prefix = '<?xml version="1.0" encoding="UTF-8"?>')
```



Apple logo グラブ ファイル 編集 取り込み ウィンドウ ヘルプ

Google Earth

▼ 検索

ジャンプ ビジネスを検索 ルート

ジャンプ 例: サンフランシスコ

▼ 場所 コンテンツを追加

▼ お気に入り

- 観光
 - ツアーを開始するには、このフォルダを選択して下の [再生] ボタン
- 保留
 - R image
 - gunma.kml
 - R Test
 - This is only a R test.
 - R Test
 - Hello
- gunma2.kml

▼ レイヤ

- プライマリ データベース
 - ジオグラフィック ウェブ
 - 道路
 - 建物の 3D 表示
 - ストリートビュー
 - 境界線や地名
 - 渋滞状況
 - 天気
 - ギャラリー
 - 海
 - グローバル アウェアネス
 - スポット
 - その他
 - 地形

日本

群馬県

利根郡

片品村

OBJECTID 566

PREF 群馬県

CITY1 片品村

CITY2 NA

TOWN1 利根郡

TOWN2 片品村

JCODE 10443

P_NUM 5399

H_NUM 1730

FLAG1 1

FLAG2 0

Shape_Leng 1.10415718566

Shape_Area 0.0395160182552

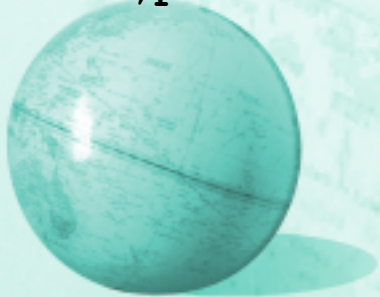
Data © 2009
Image © 2009 TerraMetrics
© 2009 ZENRIN

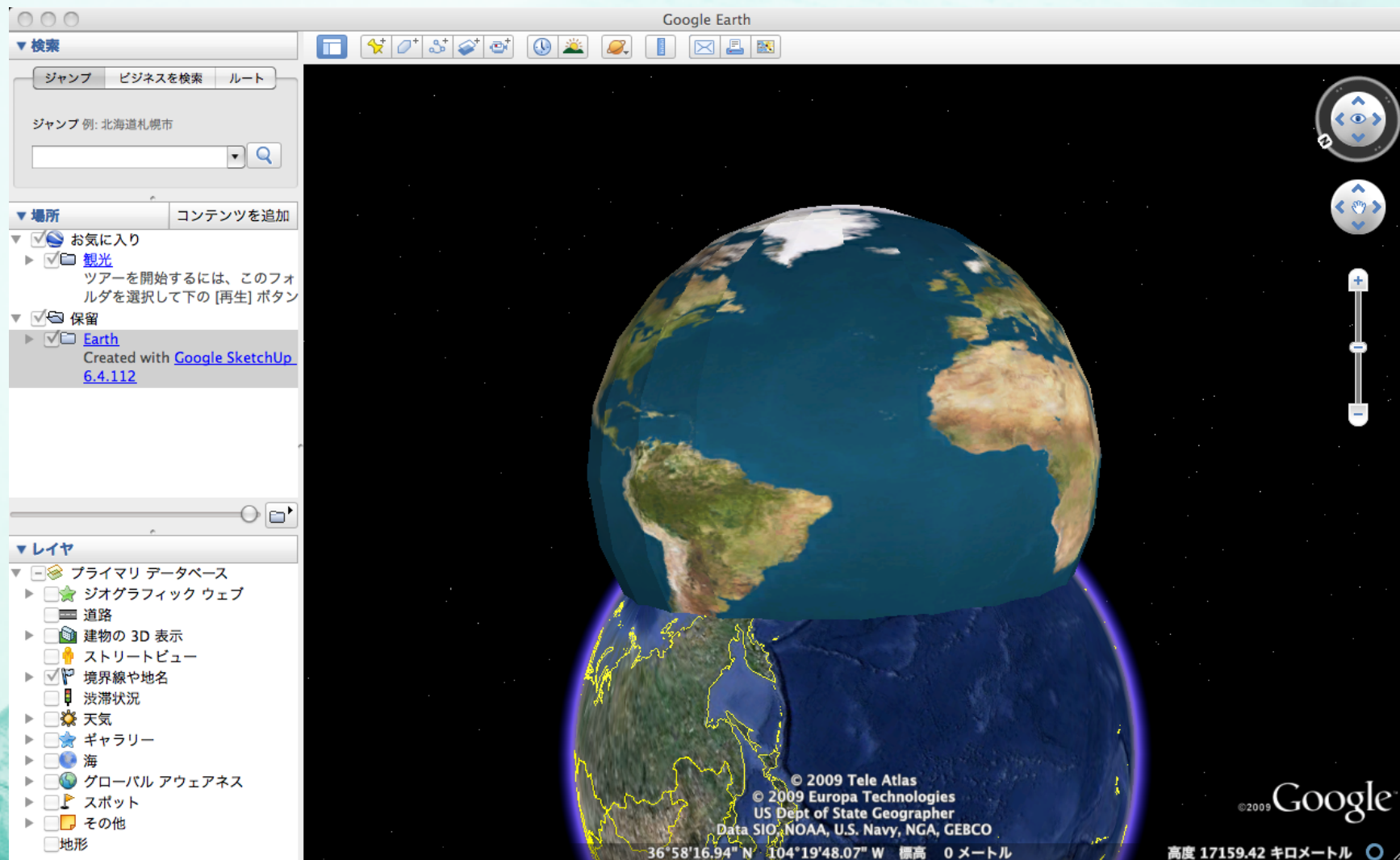
Google

36°31'36.75" N 139°01'58.26" E 標高 0 メートル 高度 122.39 キロメートル

⑦Colladaデータの修正(XML:Mac)

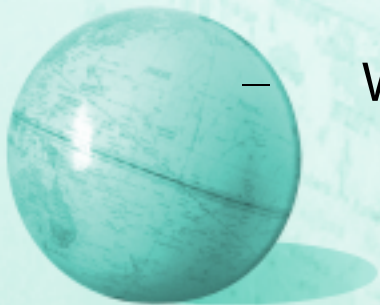
```
#demo7
#collada
library(XML)
doc<-xmlTreeParse("/Users/fumihiko_maki/Desktop/test/
Collada/doc.kml")
x<-xmlRoot(doc)
x[["Folder"]][["Placemark"]][["Model"]][["Scale"]][["x"]]<-
xmlNode("x","10000000.0")
x[["Folder"]][["Placemark"]][["Model"]][["Scale"]][["y"]]<-
xmlNode("y","10000000.0")
x[["Folder"]][["Placemark"]][["Model"]][["Scale"]][["z"]]<-
xmlNode("z","10000000.0")
saveXML(x,"/Users/fumihiko_maki/Desktop/test/Collada/doc2.kml"
,prefix = '<?xml version="1.0" encoding="UTF-8"?>')
```





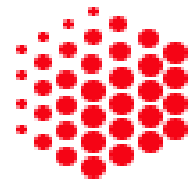
(8) ま と め

- 以前と比べてRでKMLを利用する環境が整ってきた
- 基本をSPパッケージクラスの構造にして互換性を持たせてある
- 高さ等の未実装の部分もある
- データ変換にはrgdal
- KMLオーバーレイ出力にはmaptools
- KMLの内容を操作するにはXML
- 問題点として
 - MacOSX : rgdal
 - バイナリパッケージが無い
 - 自分でmakeしろってか？
 - Windows : XML
 - UTF-8が基本の世界ではプログラムが面倒
 - インナークラス難しい。私の勉強不足？



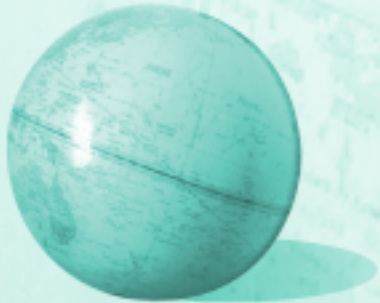
今後はGoogleMapに手を出してみたい

ご清聴ありがとうございました



dat@cube

データキューブ株式会社
<http://www.datacube.co.jp>



資料

```
#####  
# kmlFunctions.r ver 0.2.1  
# fumihiko makiyama  
#####  
  
## kmlAltitudeNode()  
kmlAltitudeNode<-function(doc,data,fillcolor,linecolor,extrude="1",tessellate="1",altitudeMode="absolute"){  
  kmlnode <- xmlRoot(doc)  
  node0<-kmlnode[[1]]  
  kmlver5<- 'http://www.opengis.net/kml/2.2' xmlns:gx="http://www.google.com/kml/ext/2.2"  
  kmlnode<-xmlNode("kml",node0,attrs=c(xmlns=kmlver5))  
  len1<-length(node0[["Folder"]])  
  for(i in 3:len1){  
    node1<-node0[["Folder"]][[i]][["MultiGeometry"]][["Polygon"]]  
    node1<-  
    addChildren(node1,xmlNode("extrude",extrude),xmlNode("tessellate",tessellate),xmlNode("gx:altitudeMode",altitu  
deMode))  
    node2<-node0[["Folder"]][[i]][["Style"]][["PolyStyle"]]  
    node2<-xmlNode("PolyStyle",xmlNode("color",fillcolor[i-2]))  
    #linecolor  
    node3<-node0[["Folder"]][[i]][["Style"]][["LineStyle"]]  
    node3<-xmlNode("LineStyle",xmlNode("color",linecolor))  
    #polygon  
    len2<-length(node0[["Folder"]][[i]][["MultiGeometry"]][["Polygon"]])  
    for(j in 1:len2){  
      x1<-xmlValue(node1[[j]][[1]][[1]])  
      x2<-rev(unlist(strsplit(x1," ")))  
      x3<-paste(x2,data[[i-2]],sep="," ,collapse=" ")  
      node1[[j]][[1]][[1]]<-xmlNode("coordinates",x3)  
    }  
    #replace node  
    node0[["Folder"]][[i]][["MultiGeometry"]][["Polygon"]]<-node1  
    node0[["Folder"]][[i]][["Style"]][["PolyStyle"]]<-node2  
    node0[["Folder"]][[i]][["Style"]][["LineStyle"]]<-node3  
    kmlnode[[1]]<-node0  
  }  
  return(kmlnode)  
}
```