Perlombongan Data Reruang (Spatial)

Data yang melibatkan ruang, lokasi dan geografi.

Objek Reruang : spatial objects

Contoh : set data vektor mungkin menerangkan maklumat tentang sempadan negara di dunia bersama sama dengan maklumat tentang saiz populasi

Medan Reruang : spatial fields

Contoh : aplikasi analisis reruang yang melibatkan data meteorologi, sains bumi, analisis imej dan lain lain. Jika data reruang dicerap bersama dengan maklumat masa, ianya dikenali sebagai analisis ruang masa

- Data Meteorologi
- Data objek bergerak
- Data Sains Bumi
- Data Wabak Penyakit
- Data diagnostik penyakit
- Data demografi

Spatial Points / Data Titik

```
load("G:/My Drive/Master-Data-Science/Semester_1/Data_Mining/Data/wst.RData")
wst
```

```
##    longitude latitude name precip
## 1     -116.7     45.3    A    721
## 2     -120.4     42.6    B     19
## 3     -116.7     38.9    C     52
## 4     -113.5     42.1    D    188
## 5     -115.5     35.7    E    749
## 6     -120.8     38.9    F      8
## 7     -119.5     36.2    G    725
## 8     -113.7     39.0    H    843
## 9     -113.7     41.6    I    289
## 10    -110.7     36.9    J    249
```

```
attach(wst)
```

```
## The following object is masked from package:datasets:
##
##     precip
```

```r
library(sp)
library(raster)
names(wst)
```

```
## [1] "longitude" "latitude"  "name"      "precip"
```

Takrifkan data reruang

```r
lonlat = cbind(longitude, latitude)
pts = SpatialPoints(lonlat)
pts
```

```
## class       : SpatialPoints
## features    : 10
## extent      : -120.8, -110.7, 35.7, 45.3  (xmin, xmax, ymin, ymax)
## crs         : NA
```

Takrifkan CRS dalam data reruang

```r
crdref = CRS('+proj=longlat +datum=WGS84')

pts =SpatialPoints(lonlat, proj4string=crdref)
pts
```

```
## class       : SpatialPoints
## features    : 10
## extent      : -120.8, -110.7, 35.7, 45.3  (xmin, xmax, ymin, ymax)
## crs         : +proj=longlat +datum=WGS84 +no_defs
```

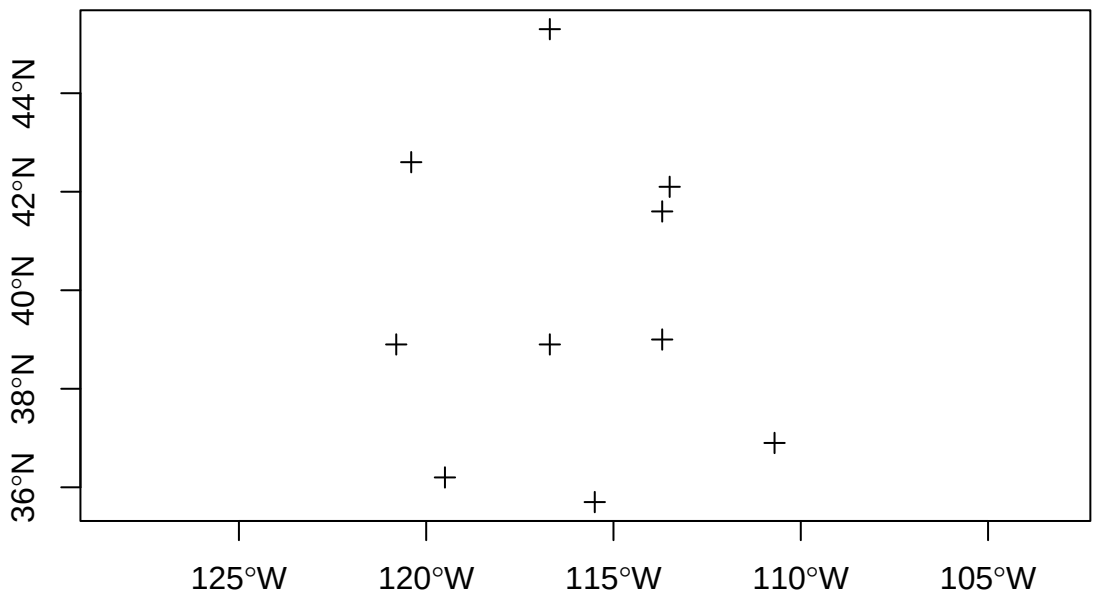
Masukkan maklumat p/ubah data yang diminati

```r
df = data.frame(ID = name, precip)
ptsdf = SpatialPointsDataFrame(pts, data=df)
ptsdf
```

```
## class       : SpatialPointsDataFrame
## features    : 10
## extent      : -120.8, -110.7, 35.7, 45.3  (xmin, xmax, ymin, ymax)
## crs         : +proj=longlat +datum=WGS84 +no_defs
## variables   : 2
## names       : ID, precip
## min values  :  A,      8
## max values  :  J,    843
```

Plot Data

```
plot(pts, axes=T)
```



Lihat details data

```
showDefault(ptsdf)
```

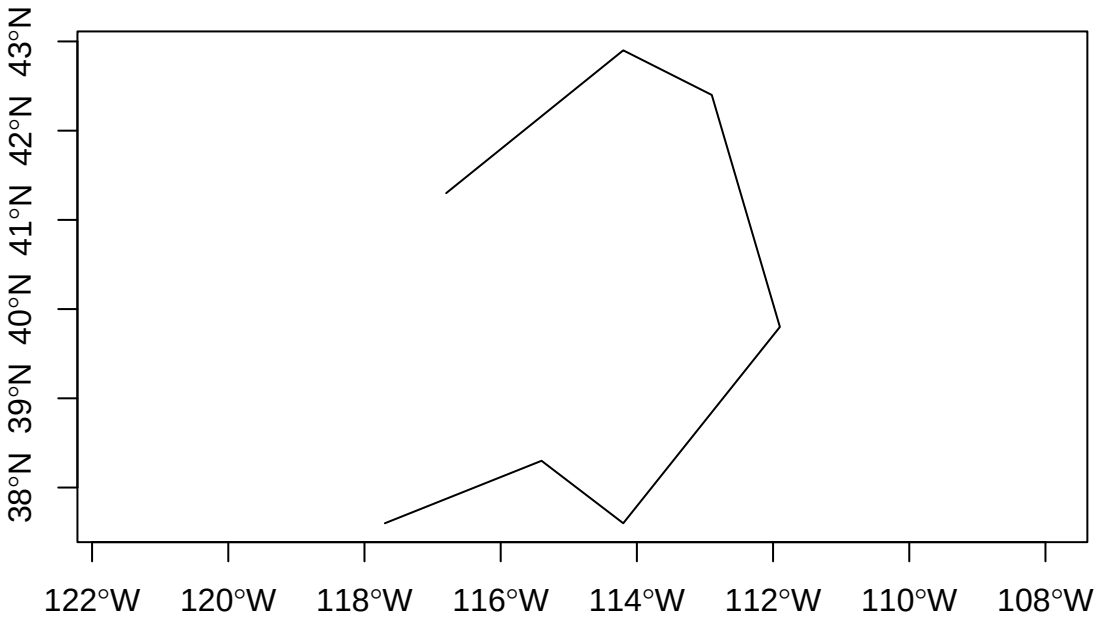
```
## An object of class "SpatialPointsDataFrame"
## Slot "data":
##    ID precip
## 1   A    721
## 2   B     19
## 3   C     52
## 4   D    188
## 5   E    749
## 6   F      8
## 7   G    725
## 8   H    843
## 9   I    289
## 10  J    249
##
```

```
## Slot "coords.nrs":
## numeric(0)
##
## Slot "coords":
##       longitude latitude
##  [1,]    -116.7     45.3
##  [2,]    -120.4     42.6
##  [3,]    -116.7     38.9
##  [4,]    -113.5     42.1
##  [5,]    -115.5     35.7
##  [6,]    -120.8     38.9
##  [7,]    -119.5     36.2
##  [8,]    -113.7     39.0
##  [9,]    -113.7     41.6
## [10,]    -110.7     36.9
##
## Slot "bbox":
##              min    max
## longitude -120.8 -110.7
## latitude    35.7   45.3
##
## Slot "proj4string":
## Coordinate Reference System:
## Deprecated Proj.4 representation: +proj=longlat +datum=WGS84 +no_defs
## WKT2 2019 representation:
## GEOGCRS["unknown",
##     DATUM["World Geodetic System 1984",
##         ELLIPSOID["WGS 84",6378137,298.257223563,
##             LENGTHUNIT["metre",1]],
##         ID["EPSG",6326]],
##     PRIMEM["Greenwich",0,
##         ANGLEUNIT["degree",0.0174532925199433],
##         ID["EPSG",8901]],
##     CS[ellipsoidal,2],
##         AXIS["longitude",east,
##             ORDER[1],
##             ANGLEUNIT["degree",0.0174532925199433,
##                 ID["EPSG",9122]]],
##         AXIS["latitude",north,
##             ORDER[2],
##             ANGLEUNIT["degree",0.0174532925199433,
##                 ID["EPSG",9122]]]]
```

Spatial Lines / Data Garis

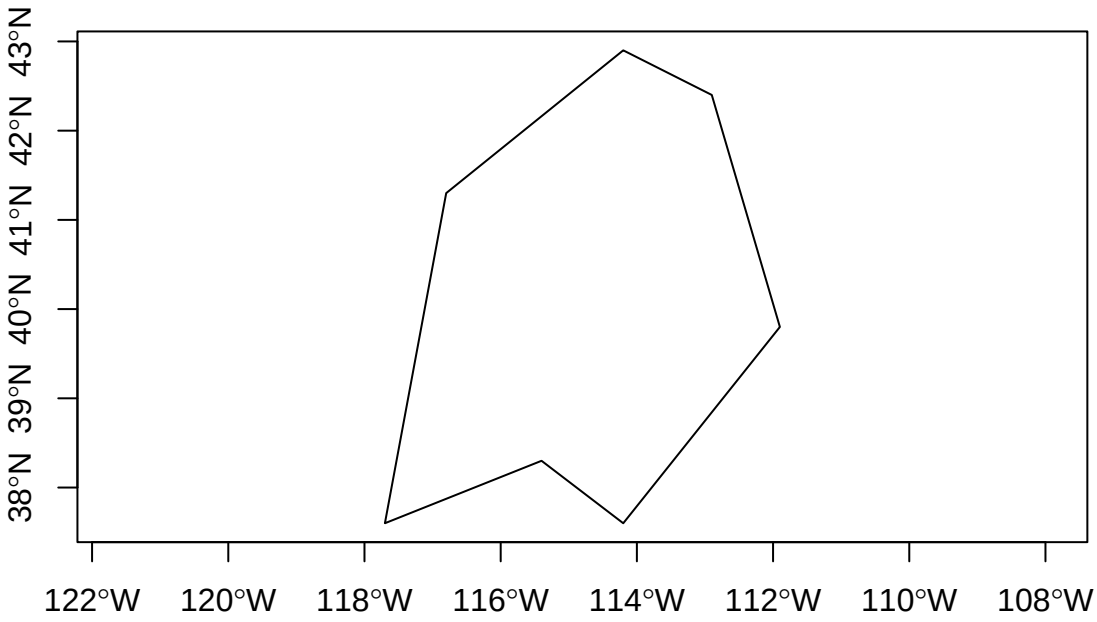
```
lon = c(-116.8, -114.2, -112.9, -111.9,-114.2,-115.4,-117.7)
lat = c(41.3,42.9,42.4,39.8,37.6,38.3,37.6)
```

```
lonlat = cbind(lon,lat)
lns = spLines(lonlat, crs=crdref)
plot(lns,axes=T)
```



Spatial Polygon / Data Poligon

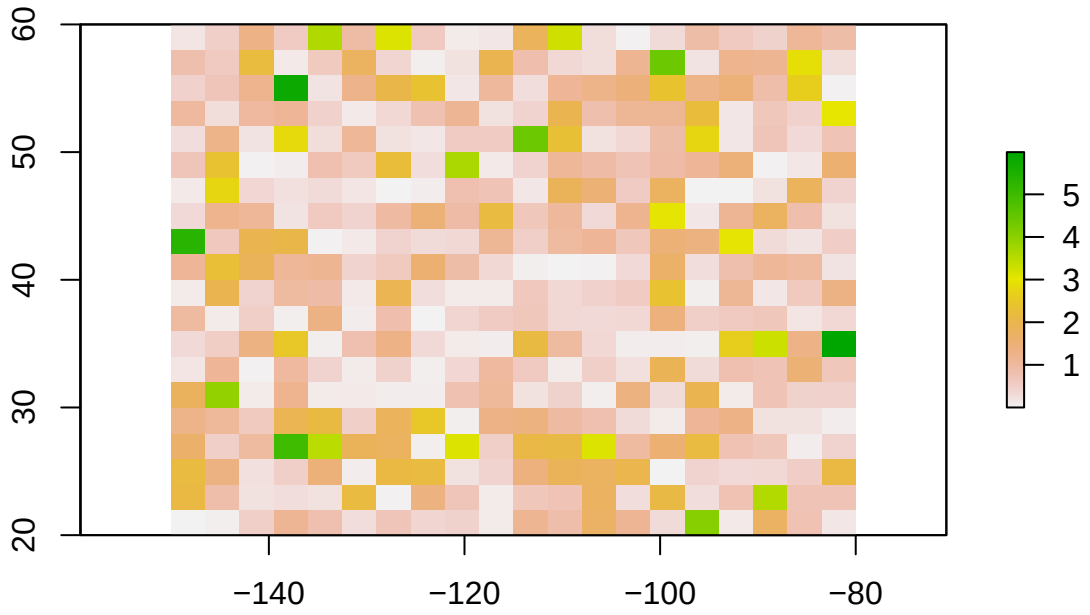
```
pols = spPolygons(lonlat, crs=crdref)
plot(pols, axes=T)
```



Data Raster

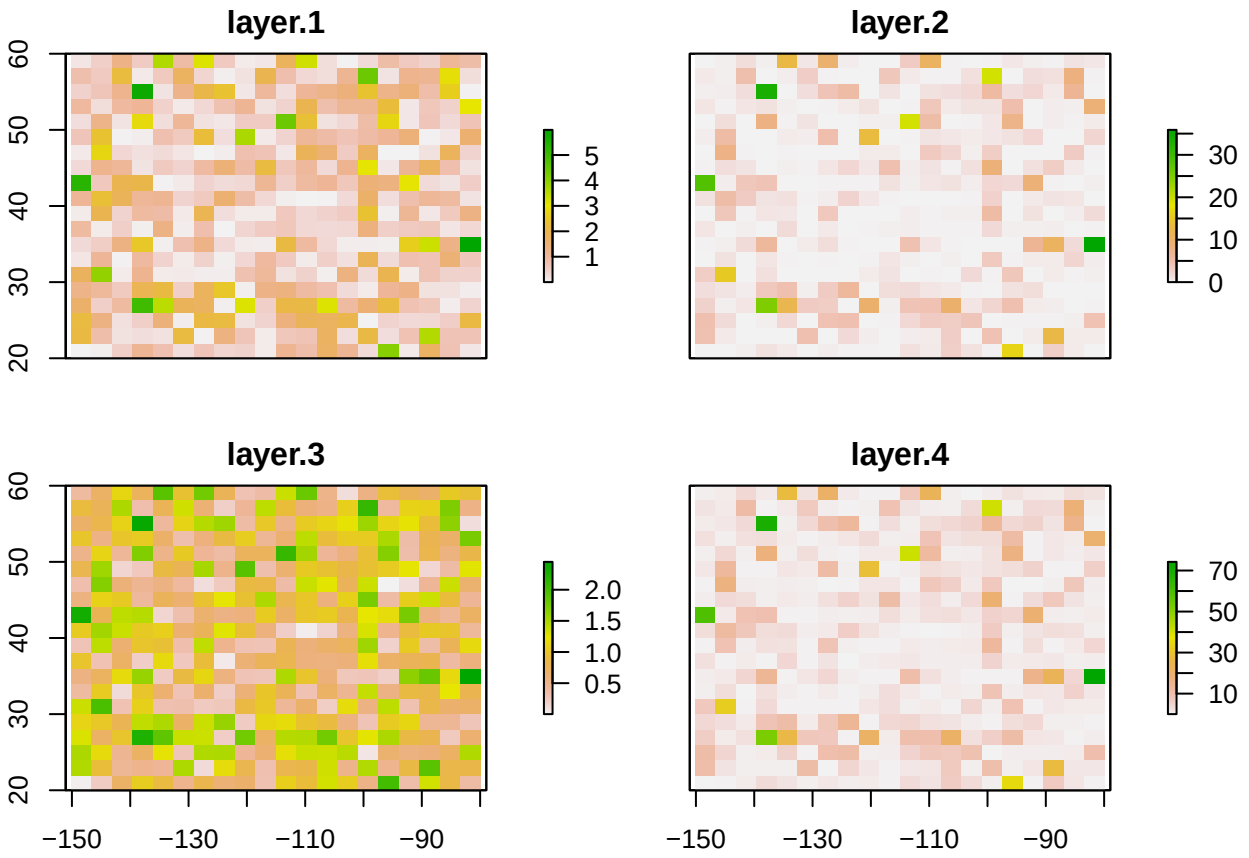
```
r=raster(ncol=20,nrow=20, xmx=-80,xmn=-150, ymn=20, ymx=60)

x = rexp(ncell(r), rate=1)
values(r) = x
plot(r)
```




```
r2 = r*r
r3 = sqrt(r)
r4 = 2*r2+r3

s = stack(r,r2,r3,r4)
plot(s)
```



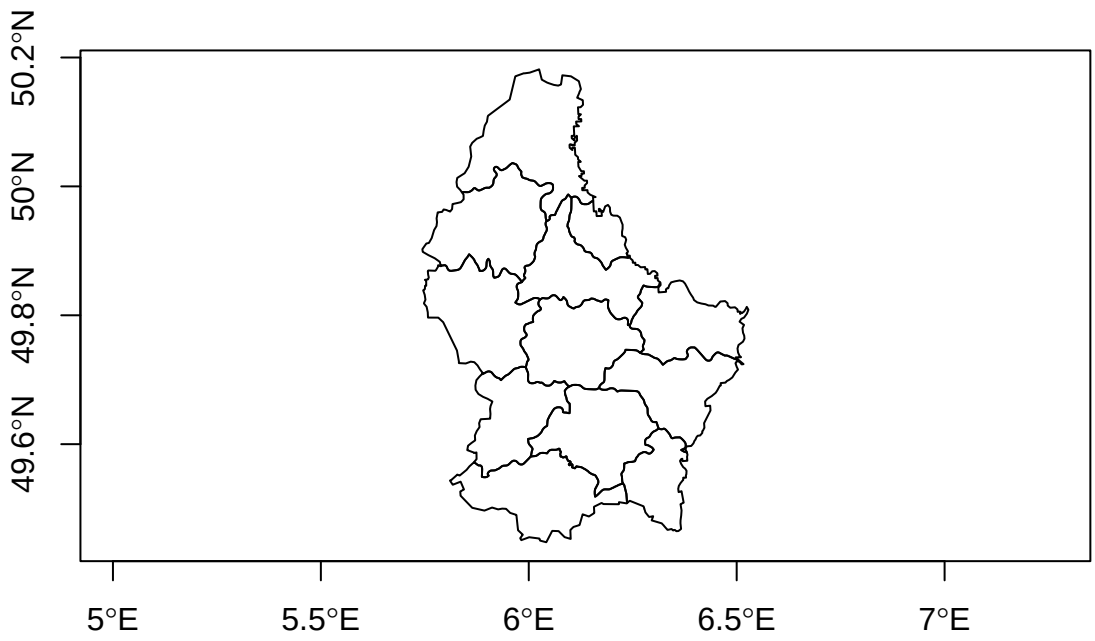
#Beberapa teknik manipulasi data vektor reruang

Mempersembahkan data reruang sebagai format data frame

```
library(terra)
```

```
## terra 1.8.5
```

```
f = system.file('external/lux.shp', package='raster')
p = shapefile(f)
plot(p, axes=T)
```




```
d = data.frame(p)
d
```

```
##    ID_1       NAME_1 ID_2           NAME_2 AREA
## 1     1     Diekirch    1         Clervaux  312
## 2     1     Diekirch    2         Diekirch  218
## 3     1     Diekirch    3          Redange  259
## 4     1     Diekirch    4          Vianden   76
## 5     1     Diekirch    5            Wiltz  263
## 6     2 Grevenmacher    6       Echternach  188
## 7     2 Grevenmacher    7           Remich  129
## 8     2 Grevenmacher   12     Grevenmacher  210
## 9     3   Luxembourg    8         Capellen  185
## 10    3   Luxembourg    9 Esch-sur-Alzette  251
## 11    3   Luxembourg   10       Luxembourg  237
## 12    3   Luxembourg   11           Mersch  233
```

Mengekstrak atribut tertentu

```
p$NAME_2
```

```
##  [1] "Clervaux"         "Diekirch"         "Redange"          "Vianden"
##  [5] "Wiltz"            "Echternach"       "Remich"           "Grevenmacher"
##  [9] "Capellen"         "Esch-sur-Alzette" "Luxembourg"       "Mersch"
```

```
p$AREA
```

```
##  [1] 312 218 259  76 263 188 129 210 185 251 237 233
```

```
p2 = p[,'NAME_2']
data.frame(p2)
```

```
##              NAME_2
## 1          Clervaux
## 2          Diekirch
## 3           Redange
## 4           Vianden
## 5             Wiltz
## 6        Echternach
## 7            Remich
## 8      Grevenmacher
## 9          Capellen
## 10 Esch-sur-Alzette
## 11       Luxembourg
## 12           Mersch
```

Tambah maklumat atribut baharu

```
Temp = round(10*rexp(12),3)

p$Temperature = Temp
data.frame(p)
```

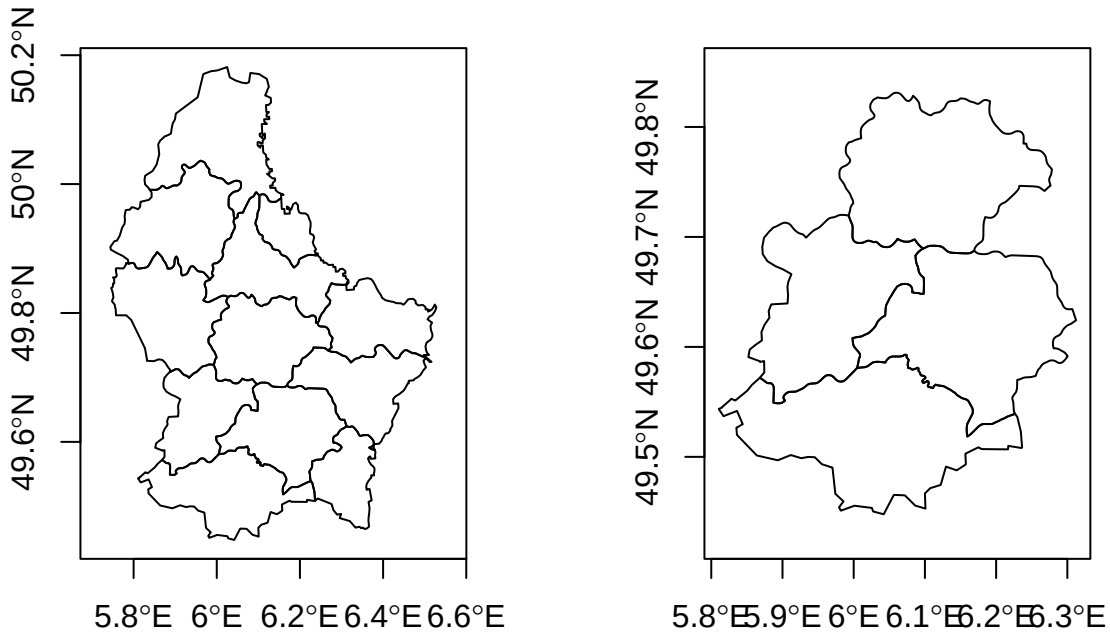
```
##    ID_1       NAME_1 ID_2           NAME_2 AREA Temperature
## 1     1     Diekirch    1         Clervaux  312      11.115
## 2     1     Diekirch    2         Diekirch  218       5.181
## 3     1     Diekirch    3          Redange  259       5.501
## 4     1     Diekirch    4          Vianden   76      16.091
## 5     1     Diekirch    5            Wiltz  263       3.438
## 6     2 Grevenmacher    6       Echternach  188      14.013
## 7     2 Grevenmacher    7           Remich  129       1.261
## 8     2 Grevenmacher   12     Grevenmacher  210       4.997
## 9     3   Luxembourg    8         Capellen  185       7.575
## 10    3   Luxembourg    9 Esch-sur-Alzette  251       3.238
## 11    3   Luxembourg   10       Luxembourg  237      11.959
## 12    3   Luxembourg   11           Mersch  233       9.250
```

Pilih subset data

```
i = which(p$NAME_1 == 'Luxembourg')

g = p[i,]
par(mfrow=c(1,2))
```

```
plot(p,axes=T)
plot(g, axes=T)
```




```
data.frame(g)
```

```
##    ID_1     NAME_1 ID_2           NAME_2 AREA Temperature
## 9     3 Luxembourg    8         Capellen  185       7.575
## 10    3 Luxembourg    9 Esch-sur-Alzette  251       3.238
## 11    3 Luxembourg   10       Luxembourg  237      11.959
## 12    3 Luxembourg   11           Mersch  233       9.250
```

Integrasi Data

Data simulasi

```
dfr = data.frame(District = p$NAME_1, Canton = p$NAME_2,
                 Precipitation = round(10*rexp(12),3))
dfr = dfr[order(dfr$Canton),]
```

```
data2 = merge(dfr,p, by.y=c('NAME_1','NAME_2'), by.x=c('District','Canton'))
data.frame(data2)
```

```
##        District           Canton Precipitation ID_1 ID_2 AREA Temperature
```

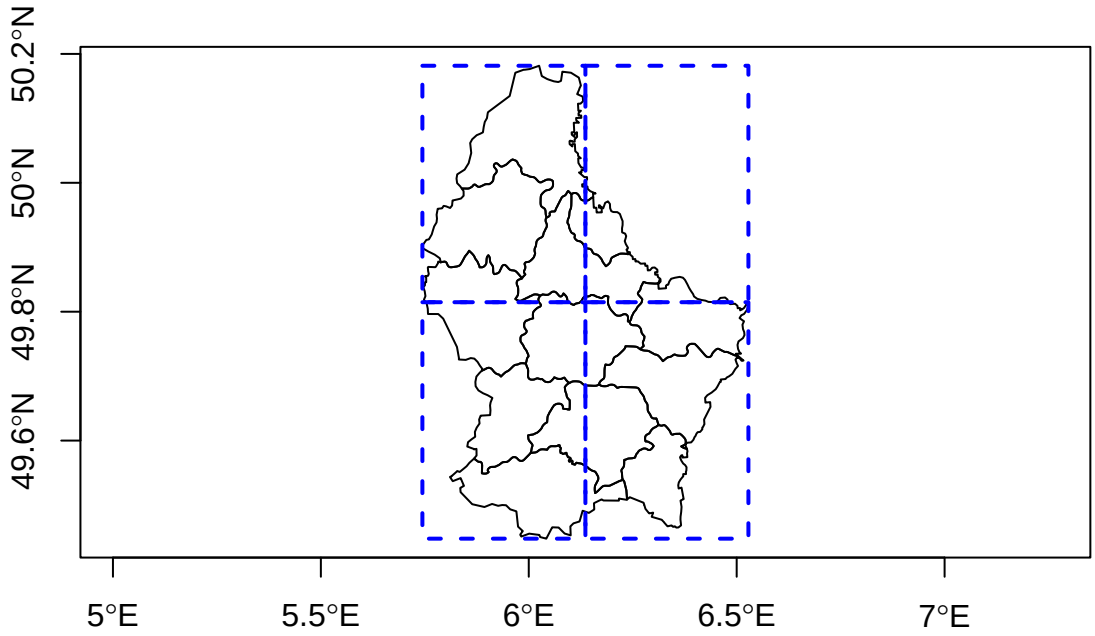
```
## 1      Diekirch         Clervaux         1.416    1    1  312       11.115
## 2      Diekirch         Diekirch         4.597    1    2  218        5.181
## 3      Diekirch          Redange         4.063    1    3  259        5.501
## 4      Diekirch          Vianden         4.133    1    4   76       16.091
## 5      Diekirch           Wiltz          2.222    1    5  263        3.438
## 6  Grevenmacher       Echternach        15.805    2    6  188       14.013
## 7  Grevenmacher     Grevenmacher         6.966    2   12  210        4.997
## 8  Grevenmacher           Remich         3.056    2    7  129        1.261
## 9    Luxembourg         Capellen        10.246    3    8  185        7.575
## 10   Luxembourg Esch-sur-Alzette         1.948    3    9  251        3.238
## 11   Luxembourg       Luxembourg        15.063    3   10  237       11.959
## 12   Luxembourg           Mersch        14.093    3   11  233        9.250
```

Manipulasi peta

Bahagikan peta kepada 4 zon

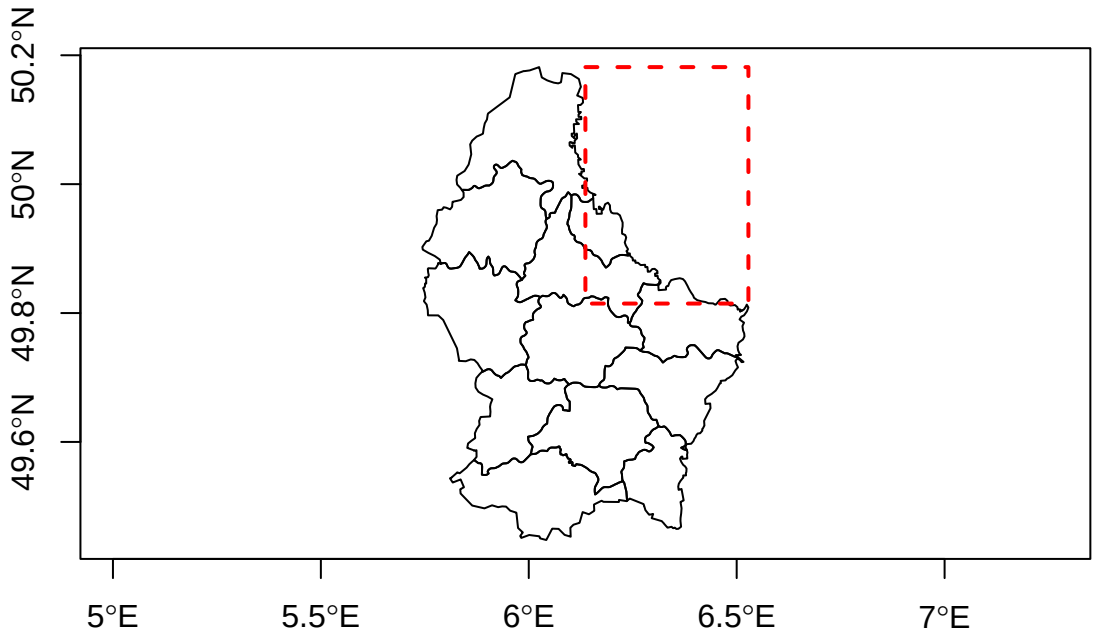
```
zon = raster(p,nrow=2,ncol=2, vals=1:4)
names(zon) = "Zone"
```

```
zon = as(zon, 'SpatialPolygonsDataFrame')
plot(p, axes=T)
plot(zon, add=T, border='blue', lwd=2, lty=2)
```



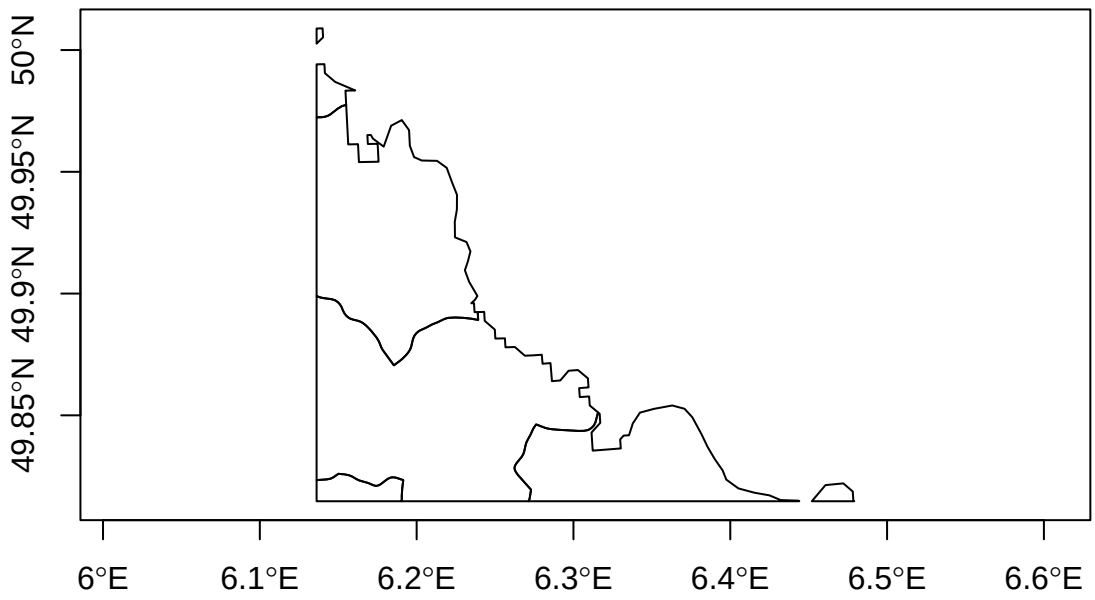
Pilih zon tertentu

```
zon2 = zon[2,]
plot(p, axes=T)
plot(zon2, add=T, border='red', lwd=2, lty=2)
```



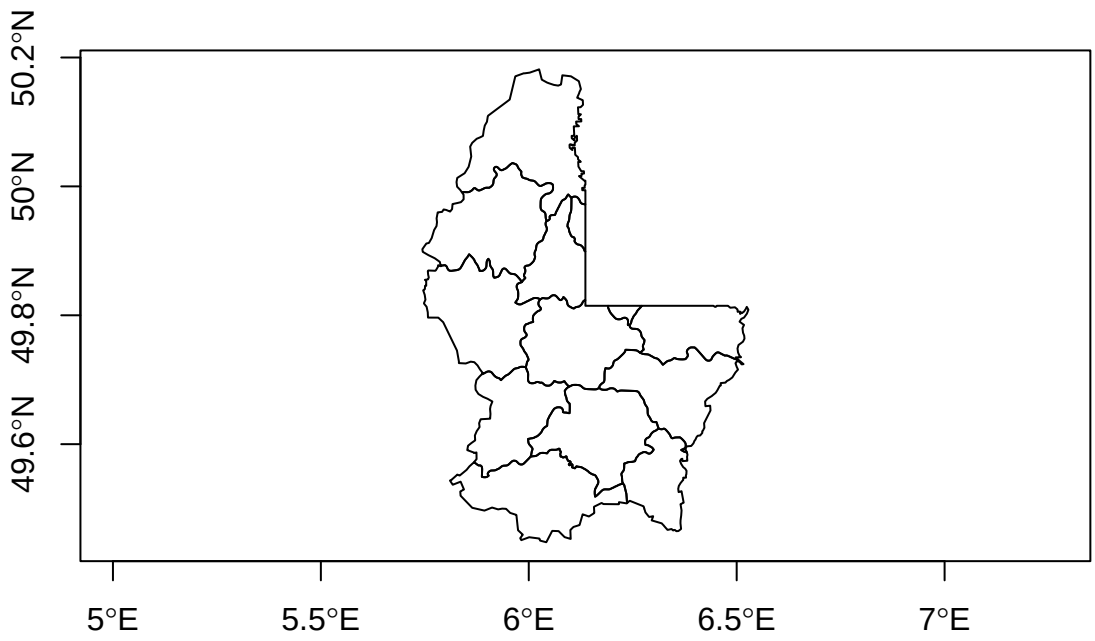
Zoom Zon

```
e = intersect(p, zon2)
plot(e,axes=T)
```



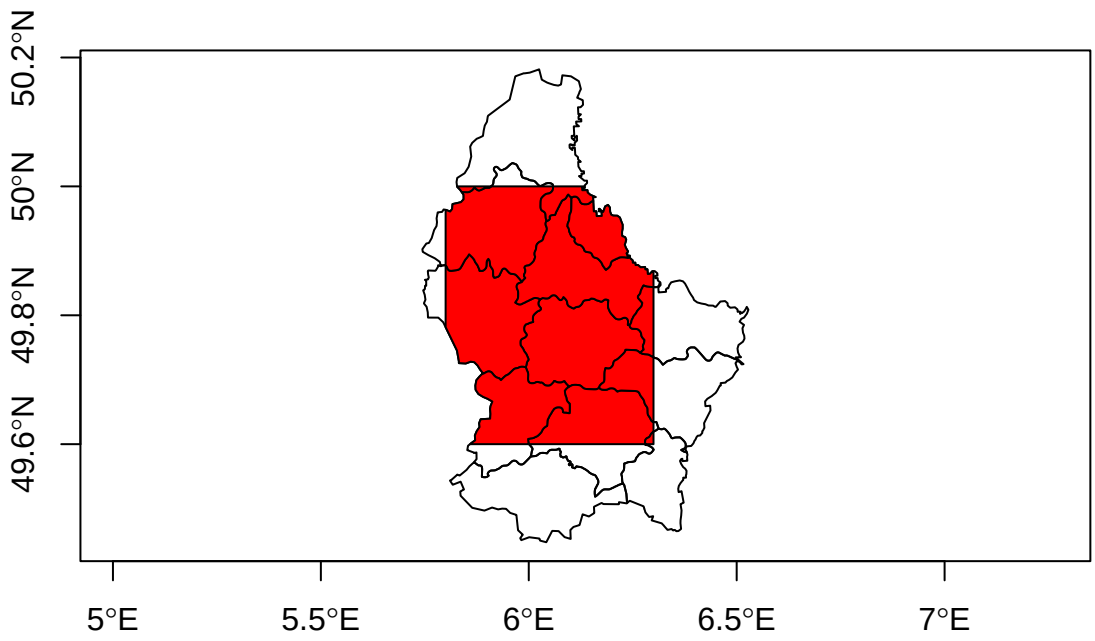
Buang Zon

```
e2 = erase(p, zon2)
plot(e2,axes=T)
```



Takrik lokasi yang nak pilih

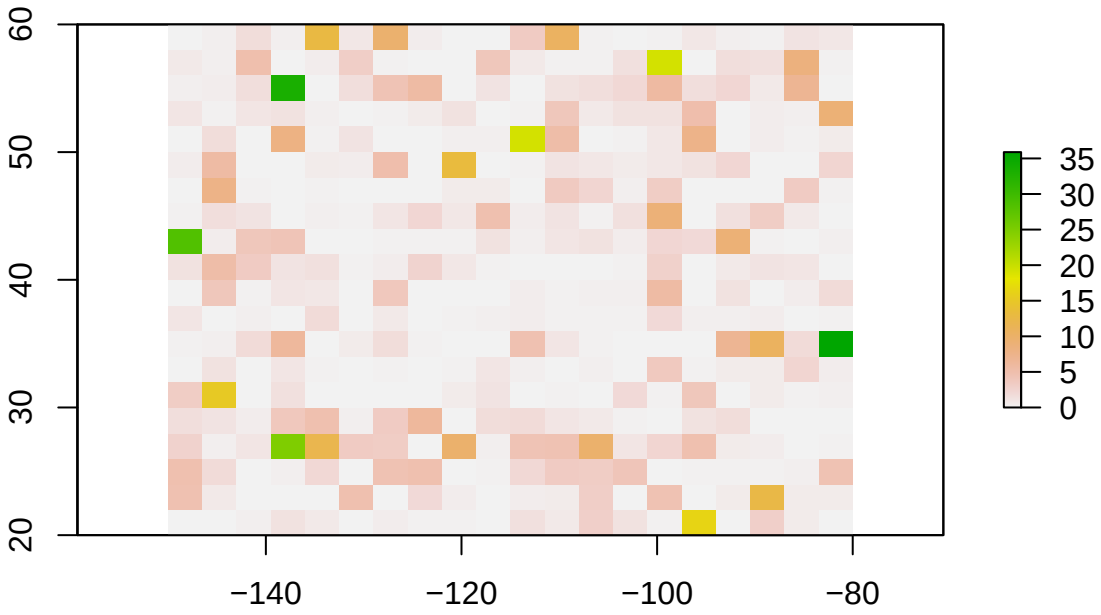
```
e3 = extent(5.8,6.3,49.6,50)
pe = crop(p, e3)
plot(p, axes=T)
plot(pe, axes=T, add=T, col='red')
```



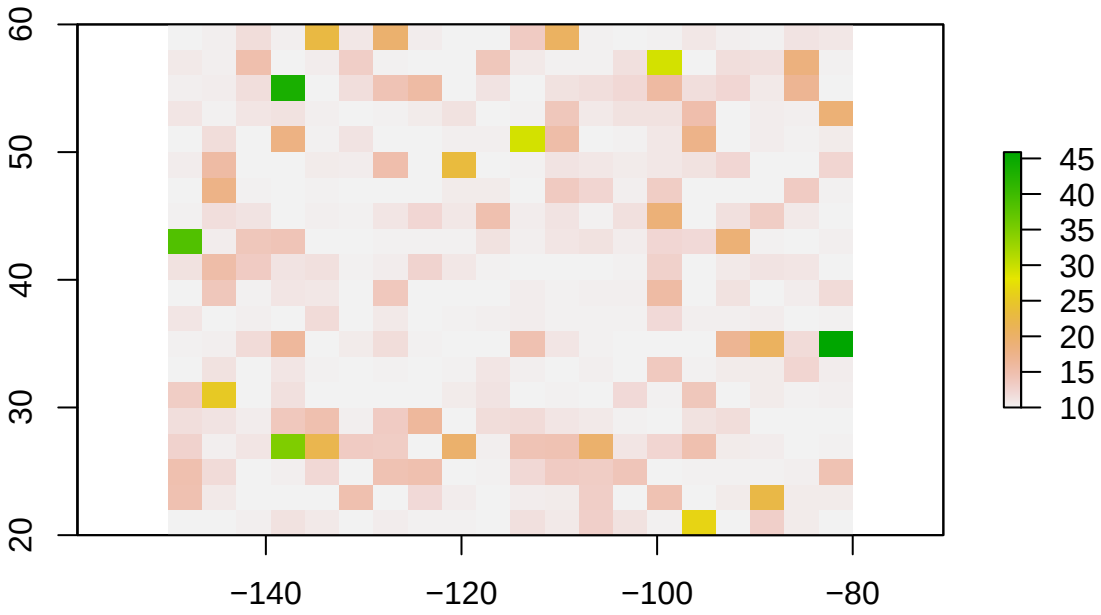
Manipulasi Data Raster

Mengekstrak objek tunggal RasterLayer daripada objek RasterBrick atau RasterStack

```
r5 = raster(s, layer=2)
plot(r5)
```

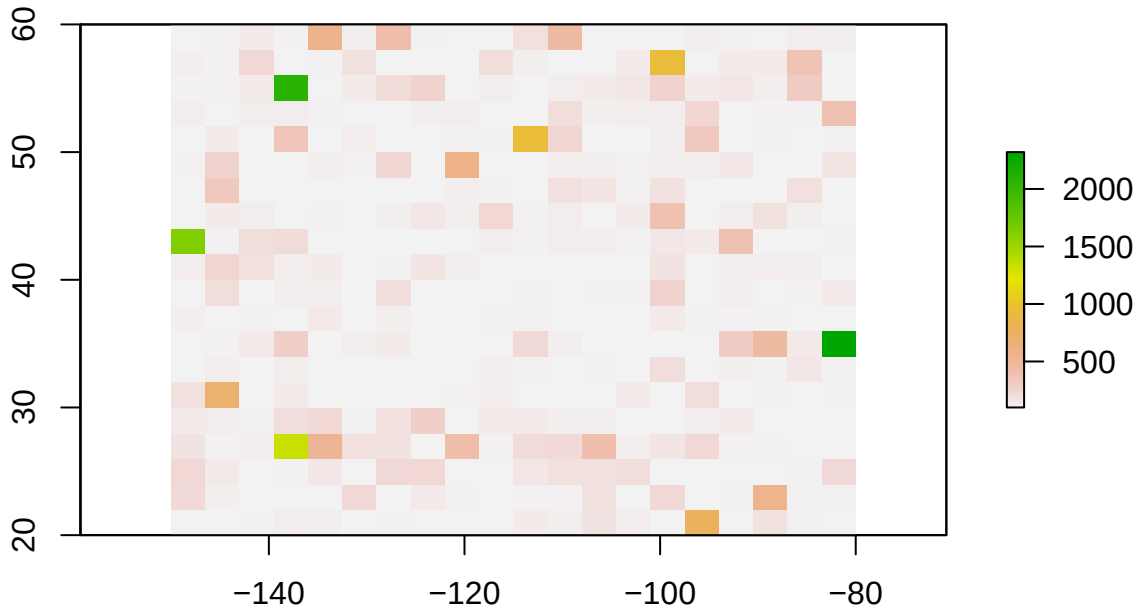



```
r6 = r2+10
plot(r6)
```



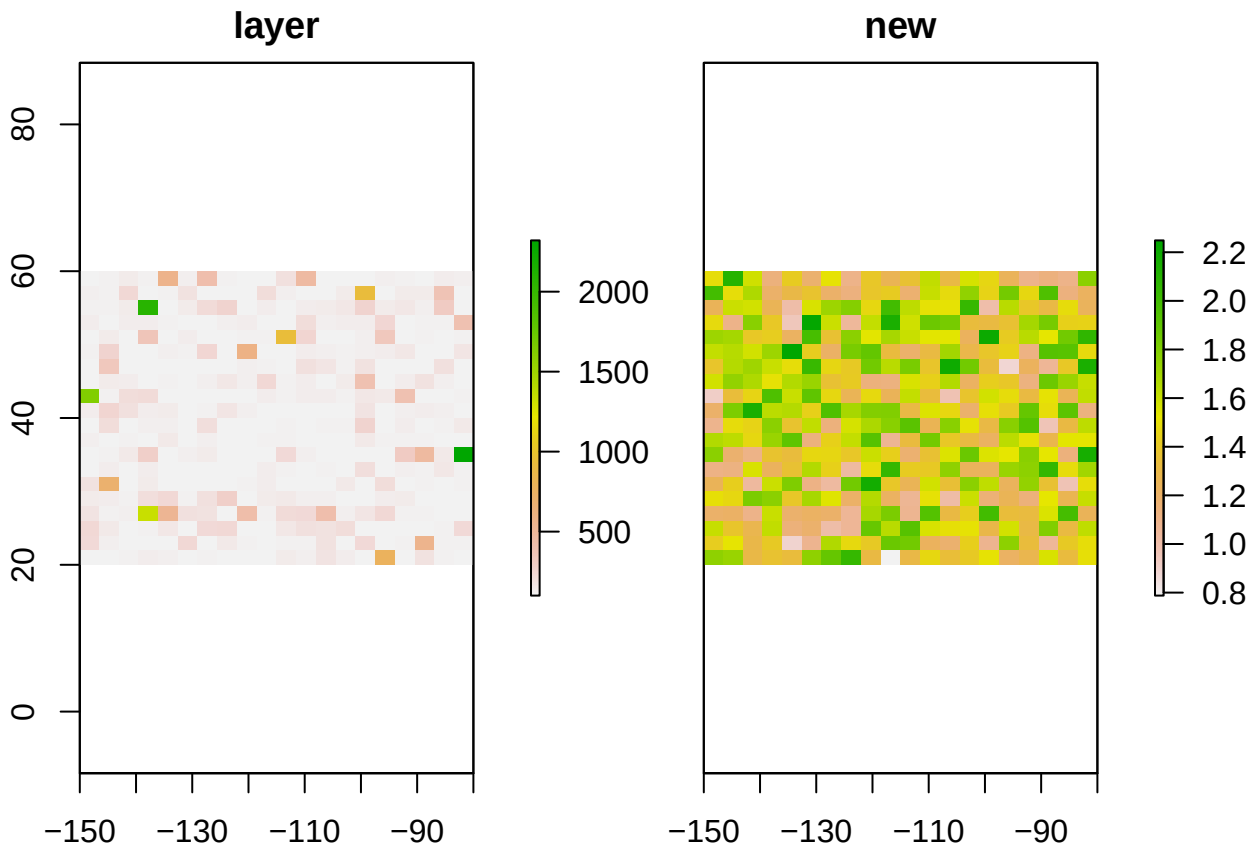
Algebra dalam data raster

```
r7=r6^2
r8 = r*r2+r7
plot(r8)
```



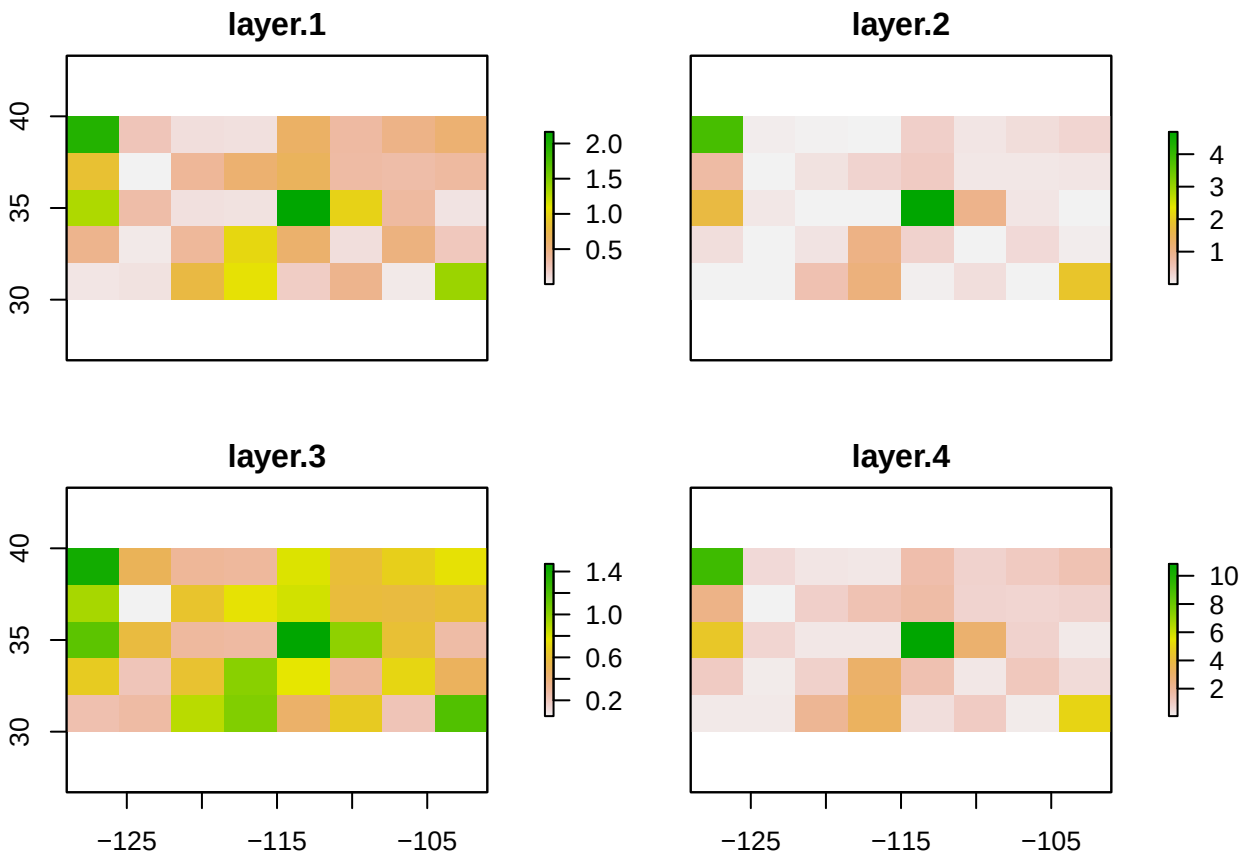
Tambah nilai baharu dalam sel

```
pi = rgamma(400,30,20)
r8$new = pi
plot(r8)
```

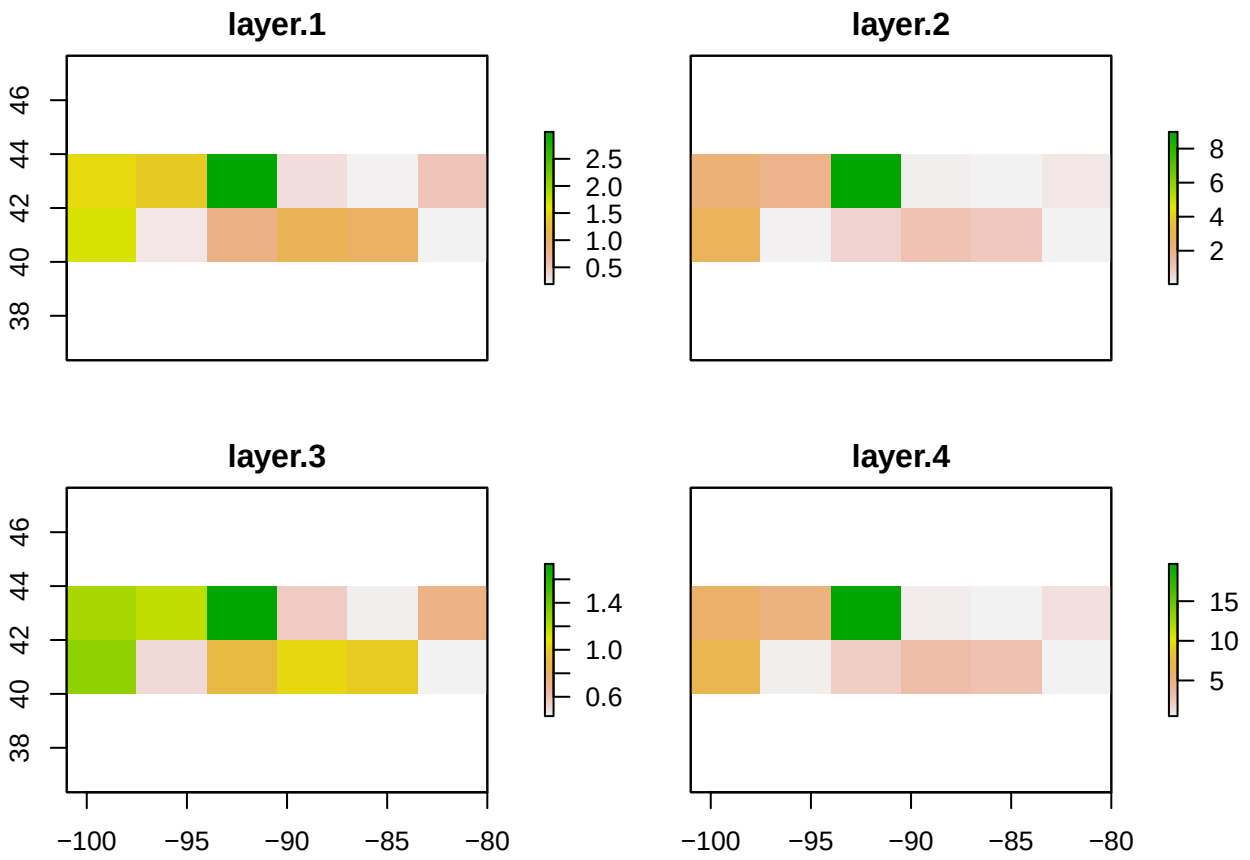


Pangkas dan gabungkan data raster

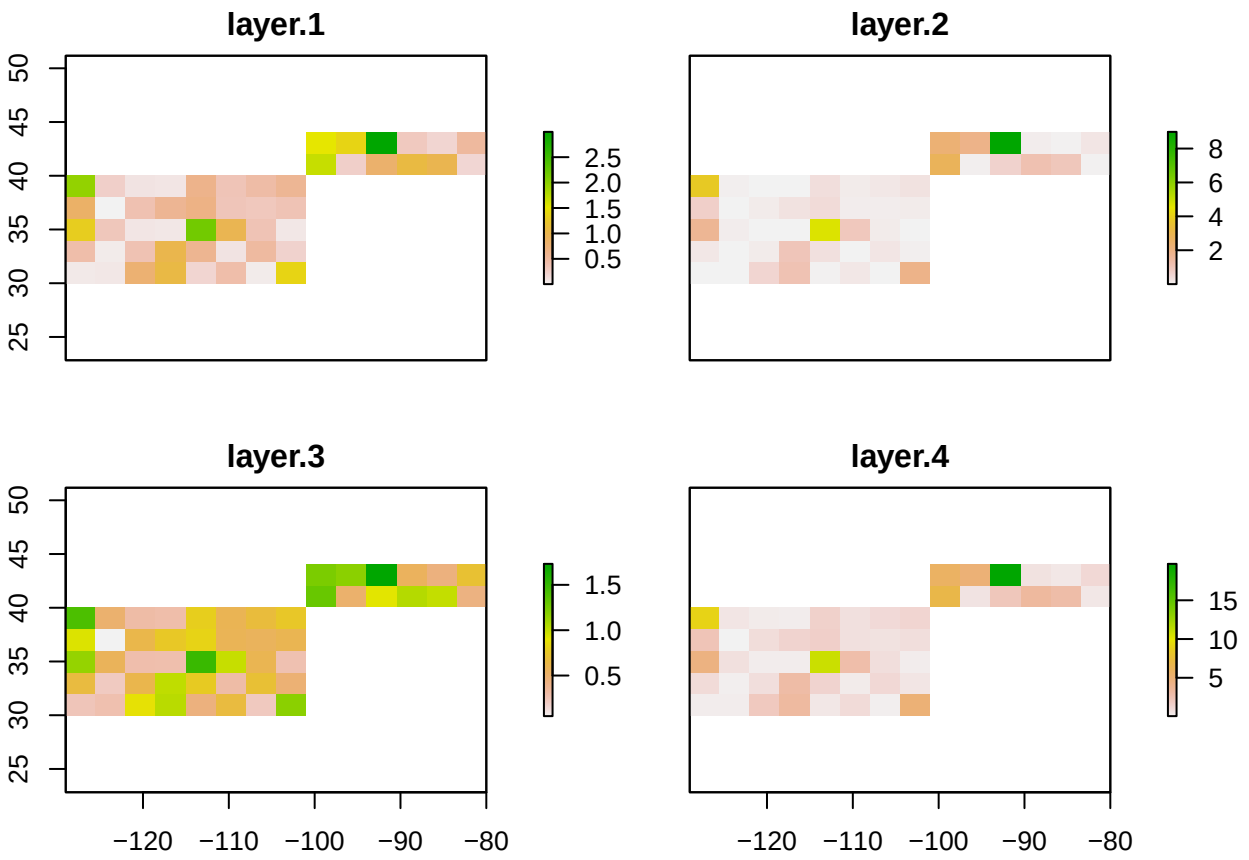
```
l2 = crop(s,extent(-130,-100,30,40))
plot(l2)
```




```
l3 = crop(s, extent(-100,-80,40,45))
plot(l3)
```




```
m = merge(l2,l3)
plot(m)
```



Fungsi deskriptif

```
cellStats(s, mean)
```

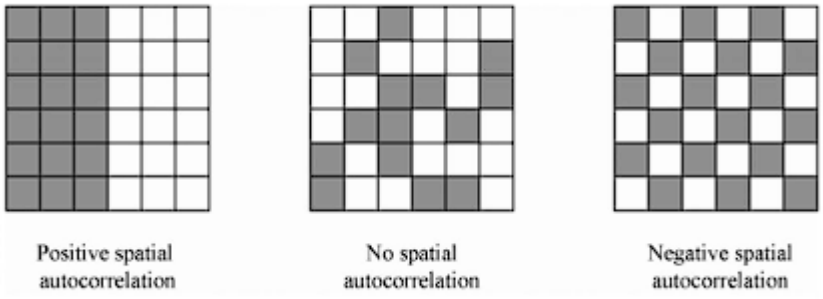
```
##   layer.1   layer.2   layer.3   layer.4
## 0.9706341 1.8967340 0.8678554 4.6613234
```

```
cellStats(s, sd)
```

```
##   layer.1   layer.2   layer.3   layer.4
## 0.9782617 4.0613618 0.4669113 8.4851117
```

Autokorelasi Reruang (Statistik Moran-ii)

measure of similarity between nearby observations




```
p <- shapefile(system.file("external/lux.shp", package="raster"))
library(spdep)
```

```
## Loading required package: spData
```

```
## To access larger datasets in this package, install the spDataLarge
## package with: `install.packages('spDataLarge',
## repos='https://nowosad.github.io/drat/', type='source')`
```

```
## Loading required package: sf
```

```
## Linking to GEOS 3.12.2, GDAL 3.9.3, PROJ 9.4.1; sf_use_s2() is TRUE
```

```
w <- poly2nb(p)
ww <- nb2listw(w)
moran.test(p$AREA, ww)
```

```
##
##  Moran I test under randomisation
##
## data:  p$AREA
## weights: ww
##
## Moran I statistic standard deviate = -0.93397, p-value = 0.8248
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic       Expectation          Variance
##       -0.24476153       -0.09090909        0.02713563
```

Interpolasi RUang

```
library(devtools)
```

```
## Loading required package: usethis
```

```
#install_github('rspatial/rspat')
library(rspat)
```

Precipitation in California

```
d <- spat_data('precipitation')
head(d)
```

```
##      ID                  NAME   LAT    LONG ALT  JAN FEB MAR APR MAY JUN JUL
## 1 ID741         DEATH VALLEY 36.47 -116.87 -59  7.4 9.5 7.5 3.4 1.7 1.0 3.7
## 2 ID743  THERMAL/FAA AIRPORT 33.63 -116.17 -34  9.2 6.9 7.9 1.8 1.6 0.4 1.9
## 3 ID744          BRAWLEY 2SW 32.96 -115.55 -31 11.3 8.3 7.6 2.0 0.8 0.1 1.9
## 4 ID753 IMPERIAL/FAA AIRPORT 32.83 -115.57 -18 10.6 7.0 6.1 2.5 0.2 0.0 2.4
## 5 ID754               NILAND 33.28 -115.51 -18  9.0 8.0 9.0 3.0 0.0 1.0 8.0
## 6 ID758        EL CENTRO/NAF 32.82 -115.67 -13  9.8 1.6 3.7 3.0 0.4 0.0 3.0
##    AUG SEP OCT NOV DEC
## 1  2.8 4.3 2.2 4.7 3.9
## 2  3.4 5.3 2.0 6.3 5.5
## 3  9.2 6.5 5.0 4.8 9.7
## 4  2.6 8.3 5.4 7.7 7.3
## 5  9.0 7.0 8.0 7.0 9.0
## 6 10.8 0.2 0.0 3.3 1.4
```

```
str(d)
```

```
## 'data.frame':    456 obs. of  17 variables:
##  $ ID  : chr  "ID741" "ID743" "ID744" "ID753" ...
##  $ NAME: chr  "DEATH VALLEY" "THERMAL/FAA AIRPORT" "BRAWLEY 2SW" "IMPERIAL/FAA AIRPORT" ...
##  $ LAT : num  36.5 33.6 33 32.8 33.3 ...
##  $ LONG: num  -117 -116 -116 -116 -116 ...
##  $ ALT : int  -59 -34 -31 -18 -18 -13 -9 -6 2 2 ...
##  $ JAN : num  7.4 9.2 11.3 10.6 9 ...
##  $ FEB : num  9.5 6.9 8.3 7 8 ...
##  $ MAR : num  7.5 7.9 7.6 6.1 9 3.7 5 9.2 72.9 72.4 ...
##  $ APR : num  3.4 1.8 2 2.5 3 3 1 2.2 32.1 30.1 ...
##  $ MAY : num  1.7 1.6 0.8 0.2 0 0.4 1 1.3 7.6 2 ...
##  $ JUN : num  1 0.4 0.1 0 1 0 0 0.2 2.2 1.1 ...
##  $ JUL : num  3.7 1.9 1.9 2.4 8 3 2 3.5 0.6 0.6 ...
##  $ AUG : num  2.8 3.4 9.2 2.6 9 10.8 9 6.5 0.6 0.5 ...
##  $ SEP : num  4.3 5.3 6.5 8.3 7 0.2 8 6.4 5.3 1.4 ...
##  $ OCT : num  2.2 2 5 5.4 8 0 8 3.8 11.4 11.6 ...
##  $ NOV : num  4.7 6.3 4.8 7.7 7 3.3 7 7.3 47.8 45.3 ...
##  $ DEC : num  3.9 5.5 9.7 7.3 9 1.4 11 7.4 63.7 58 ...
```

Curahan Hujan Tahunan

```
d$prec <- rowSums(d[,6:17])
```

peta

```
dsp<- vect(d, c("LONG", "LAT"),
           crs="+proj=longlat +datum=WSG84")
```

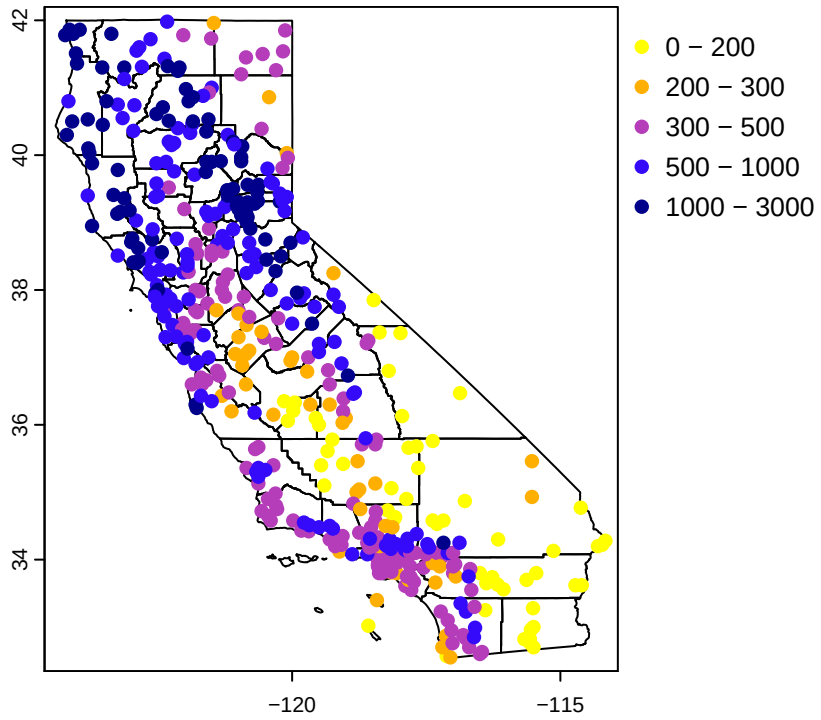
```
## Warning: PROJ: proj_create: Error 1027 (Invalid value for an argument): Unknown
## value for datum (GDAL error 1)
```

```
## Warning: [vect] Cannot set SRS to vector: empty srs
```

```
CA <- spat_data("counties")
```

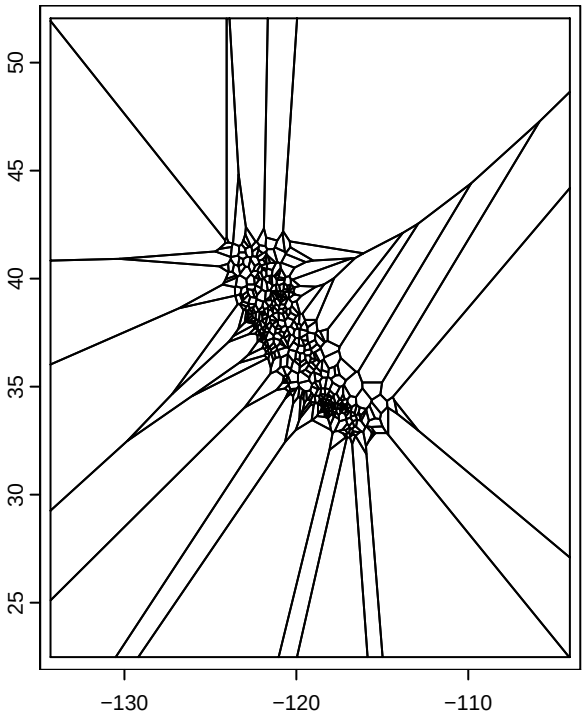
Takrifkan kategori amaun curahan hujan

```
cuts <- c(0,200,300,500,1000,3000)
library(ggplot2)
blues <- colorRampPalette(c('yellow','orange','purple','blue','darkblue'))
plot(CA)
plot(dsp, "prec", type="interval", col=blues(10),
     breaks=cuts, add=T)
```

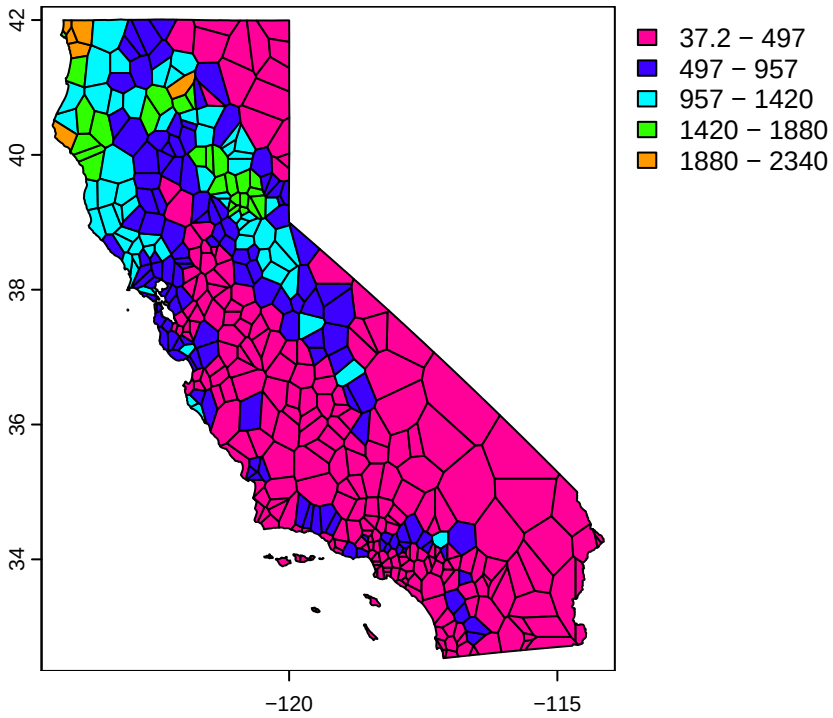


Anggarkan data curahan hujan di kawasan yang tiada cerapan data teknik Poligon Hampiran

```
v <- voronoi(dsp)
plot(v)
```




```
vca <- crop(v, CA)
plot(vca, 'prec')
```



Kaedah regressi setempat

```
houses <- read.csv("G:/My Drive/Master-Data-Science/Semester_1/Data_Mining/Data/hd.csv", sep=';', heade
str(houses)
```

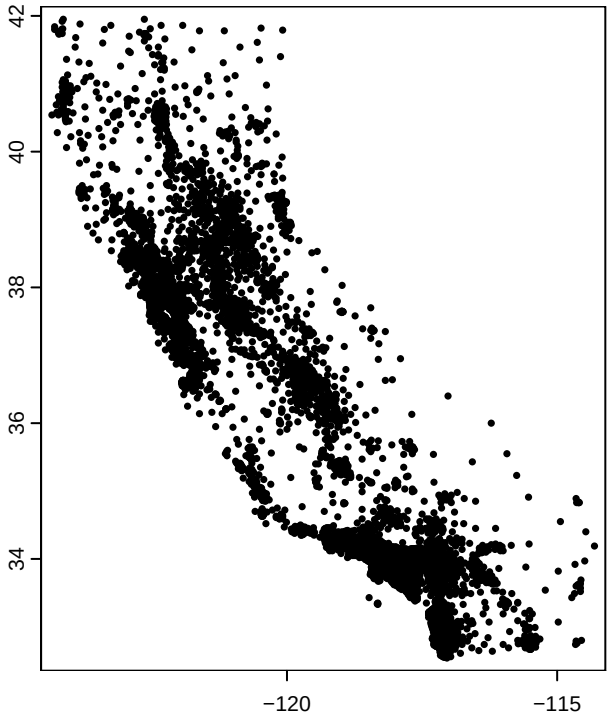
```
## 'data.frame':    20640 obs. of  19 variables:
##  $ houseValue : int  452600 358500 352100 341300 342200 269700 299200 241400 226700 261100 ...
##  $ income     : num  8.33 8.3 7.26 5.64 3.85 ...
##  $ houseAge   : int  41 21 52 52 52 52 52 52 42 52 ...
##  $ rooms      : int  880 7099 1467 1274 1627 919 2535 3104 2555 3549 ...
##  $ bedrooms   : int  129 1106 190 235 280 213 489 687 665 707 ...
##  $ population : int  322 2401 496 558 565 413 1094 1157 1206 1551 ...
##  $ households : int  126 1138 177 219 259 193 514 647 595 714 ...
##  $ latitude   : num  37.9 37.9 37.9 37.9 37.9 ...
##  $ longitude  : num  -122 -122 -122 -122 -122 ...
##  $ id.y       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ STATE      : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ COUNTY     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ NAME       : chr  "Alameda" "Alameda" "Alameda" "Alameda" ...
##  $ LSAD       : int  6 6 6 6 6 6 6 6 6 6 ...
##  $ LSAD_TRANS : chr  "County" "County" "County" "County" ...
##  $ suminc     : num  1049 9447 1285 1236 996 ...
##  $ roomhead   : num  2.73 2.96 2.96 2.28 2.88 ...
```

```
##  $ bedroomhead: num  0.401 0.461 0.383 0.421 0.496 ...
##  $ hhsize     : num  2.56 2.11 2.8 2.55 2.18 ...
```

Jelmakan data kepada kelas reruang

```
hvect <- vect(houses, c('longitude','latitude'))
```

```
plot(hvect, cex=0.5, axes=T)
```




```
countries<- spat_data('counties')
crs(hvect) <- crs(countries)
```

Regresi biasa

```
hd <- houses
model <- glm(houseValue~income+houseAge+roomhead+bedroomhead+population, data=hd)
summary(model)
```

```
##
## Call:
## glm(formula = houseValue ~ income + houseAge + roomhead + bedroomhead +
##     population, data = hd)
```

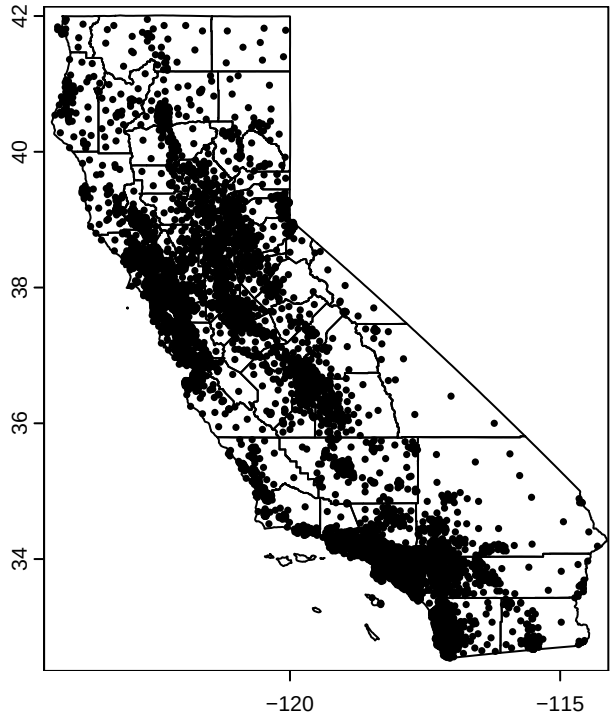
```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.508e+04  2.533e+03 -25.686  < 2e-16 ***
## income       5.179e+04  3.833e+02 135.092  < 2e-16 ***
## houseAge     1.832e+03  4.575e+01  40.039  < 2e-16 ***
## roomhead    -4.720e+04  1.489e+03 -31.688  < 2e-16 ***
## bedroomhead  2.648e+05  6.820e+03  38.823  < 2e-16 ***
## population   3.947e+00  5.081e-01   7.769 8.27e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 6022427437)
##
##     Null deviance: 2.7483e+14  on 20639  degrees of freedom
## Residual deviance: 1.2427e+14  on 20634  degrees of freedom
## AIC: 523369
##
## Number of Fisher Scoring iterations: 2
```

Model regresi biasa tidak mencerminkan maklumat yang berbeza terhadap lokasi yang berbeza

Geographically Weighted Regression (GWR) Regresi

Regresi setempat yang mempertimbangkan maklumat reruang dalam data

```
plot(hvect, cex=0.5, axes=T)
plot(countries, add=T)
```




```
countrynames <- unique(hd$NAME)
```

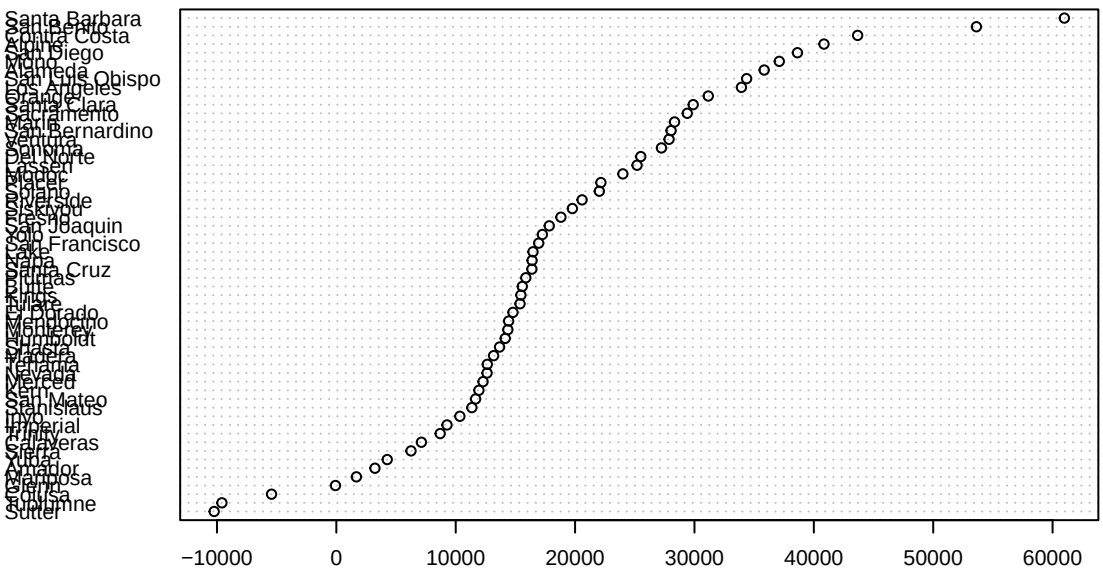
Model regresi yang berbeza terhadap kawasan berbeza

```
regfun <- function(x) {
  dat <- hd[hd$NAME == x,]
  m <- glm(houseValue~income+houseAge+roomhead+bedroomhead+population, data=dat)
  coefficients(m)
}

hd2 <- hd[!is.na(hd$NAME),]
countrynames <- unique(hd2$NAME)
res <- sapply(countrynames, regfun)
```

Kesan p/ubah income terhadap harga rumah

```
dotchart(sort(res['income',]), cex=0.65)
```

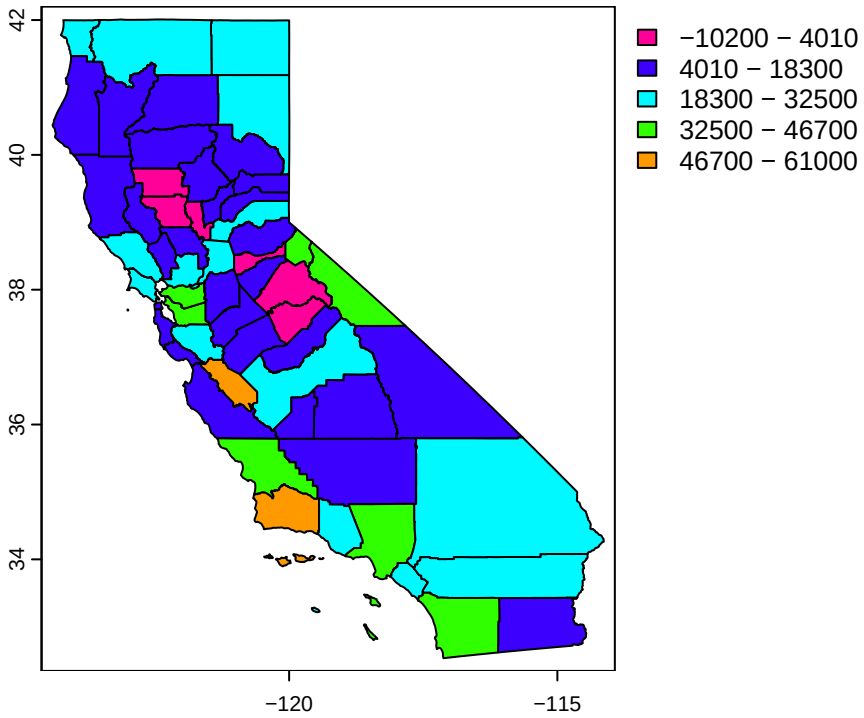


Hasilkan peta bagi variasi parameter regresi setempat

```
resdf<- data.frame(NAME=colnames(res), t(res))
dcounties <- aggregate(countries[,'NAME'], 'NAME')
cnres <- merge(dcounties, resdf, by='NAME')
```

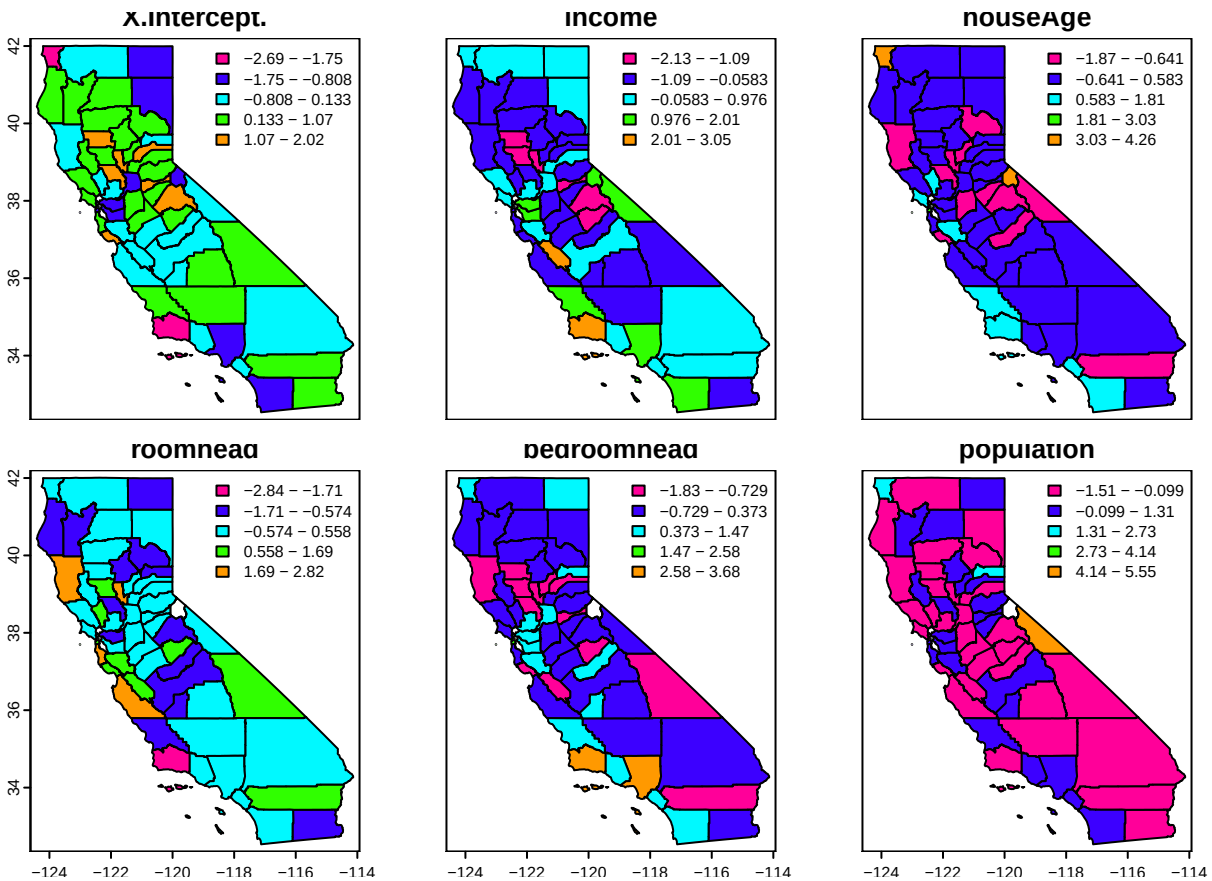
Pengaruh income terhadap nilai rumah berdasarkan lokasi berbeza

```
plot(cnres, 'income')
```



Untuk p/ubah yang lain

```
cnres2 <- cnres
values(cnres2) <- as.data.frame(scale(as.data.frame(cnres)[-1]))
plot(cnres2, 2:7, plg=list(x='topright'),
     mar=c(1,1,1,1))
```


X.Intercept.
−2.69 − −1.75
−1.75 − −0.808
−0.808 − 0.133
0.133 − 1.07
1.07 − 2.02
Income
−2.13 − −1.09
−1.09 − −0.0583
−0.0583 − 0.976
0.976 − 2.01
2.01 − 3.05
houseAge
−1.87 − −0.641
−0.641 − 0.583
0.583 − 1.81
1.81 − 3.03
3.03 − 4.26
roomnead
−2.84 − −1.71
−1.71 − −0.574
−0.574 − 0.558
0.558 − 1.69
1.69 − 2.82
bedroomnead
−1.83 − −0.729
−0.729 − 0.373
0.373 − 1.47
1.47 − 2.58
2.58 − 3.68
population
−1.51 − −0.099
−0.099 − 1.31
1.31 − 2.73
2.73 − 4.14
4.14 − 5.55
42
40
38
36
34
−124
−122
−120
−118
−116
−114