

test assignR.20190812

Chao Ma

August 12, 2019

Install assignR package from Github

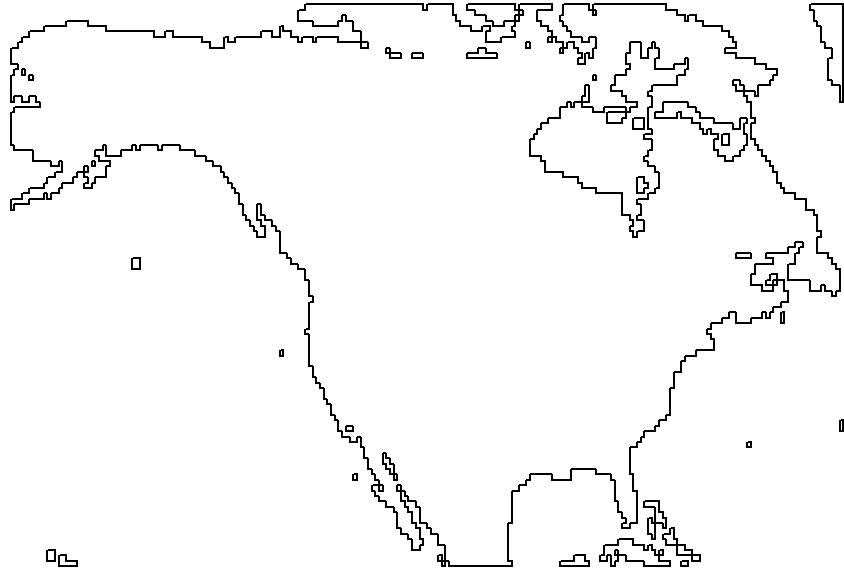
```
devtools::install_github("SPATIAL-Lab/assignR", force=T)
```

Load library

```
library(assignR)
```

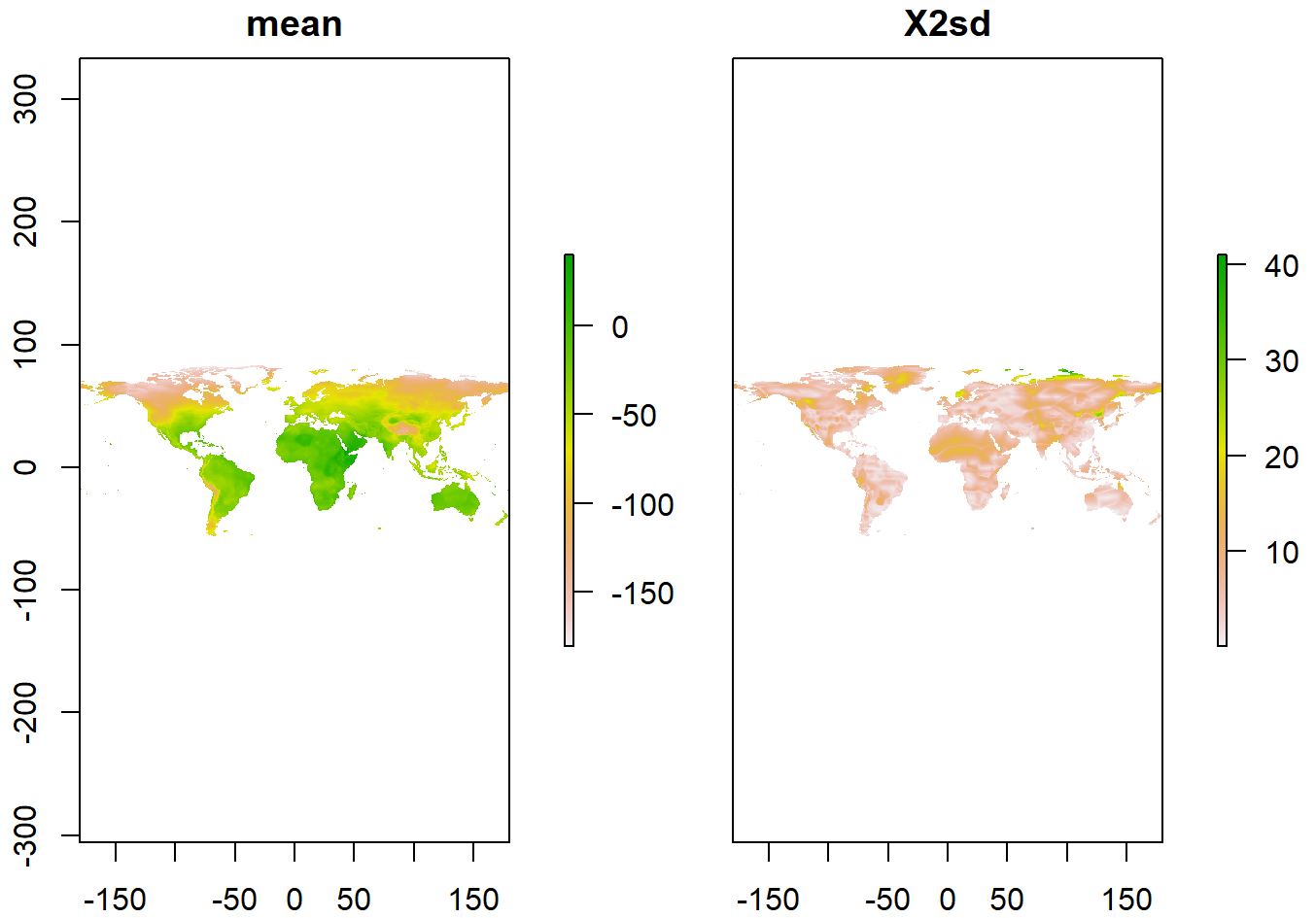
Load North America mask

```
data("naMap")  
plot(naMap)
```



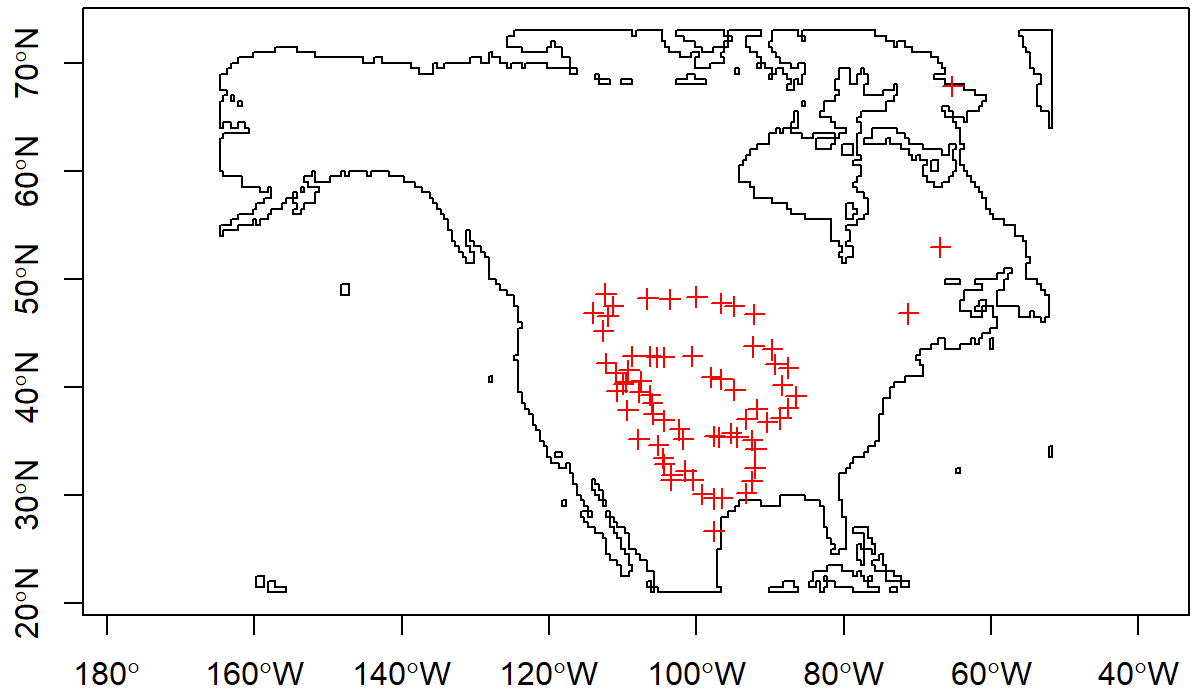
Load world precipitation hydrogen isoscape

```
data("d2h_world")  
plot(d2h_world)
```



Load hydrogen isotope for human hair in North America

```
d = subOrigData(taxon = c("Homo sapiens"), mask = naMap)
```

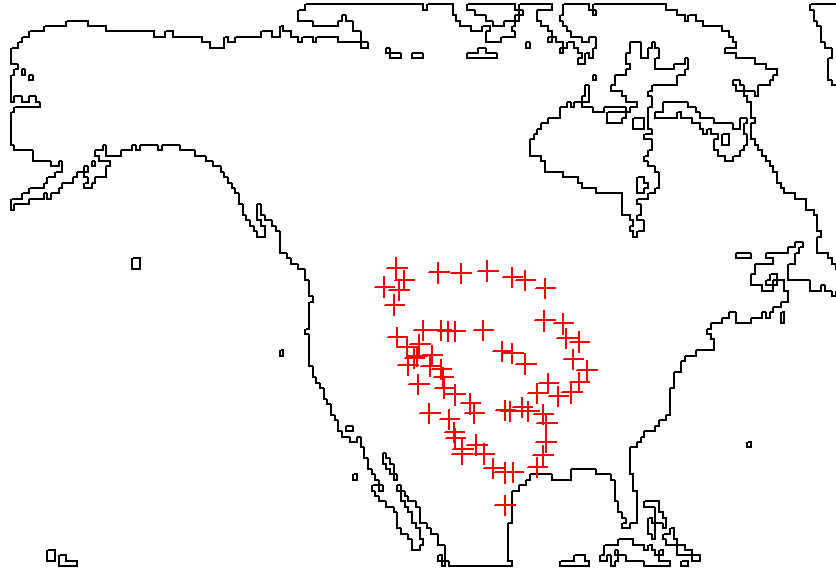


```
## 233 data points are found
```

Exclude some outliers. This step is optional, which depends on your data quality

```
d <- as.data.frame(d)
dd = d[d$coords.x1 < (-80),]
dd <- SpatialPointsDataFrame(dd[,2:3], as.data.frame(dd[,1]))
crs(dd) <- "+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
+towgs84=0,0,0"
```

```
plot(naMap)
plot(dd, add=T, col=2)
```



Rescale from environmental isoscape to tissue isoscape

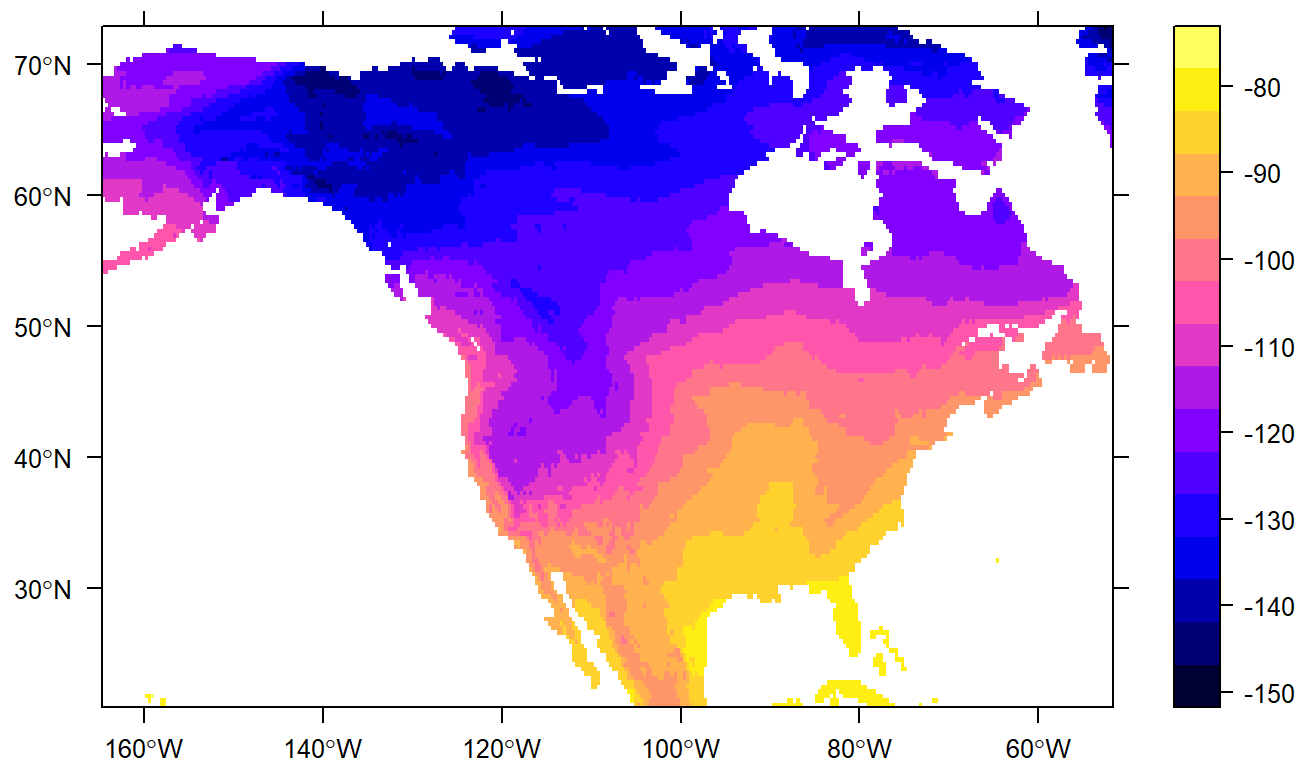
```
r = calRaster(known = dd, isoscape = d2h_world, mask = naMap)
##
##
## -----
##
## rescale function uses linear regression model, the summary of this
## model is:
## -----
##
##
## Call:
## lm(formula = tissue.iso ~ isoscape.iso[, 1])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -38.407  -4.035   0.233   4.419  22.834
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -75.06067     1.32160  -56.80  <2e-16 ***
## isoscape.iso[, 1]  0.41389     0.01996   20.74  <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.078 on 213 degrees of freedom
## Multiple R-squared:  0.6687, Adjusted R-squared:  0.6672
## F-statistic:    430 on 1 and 213 DF,  p-value: < 2.2e-16
```

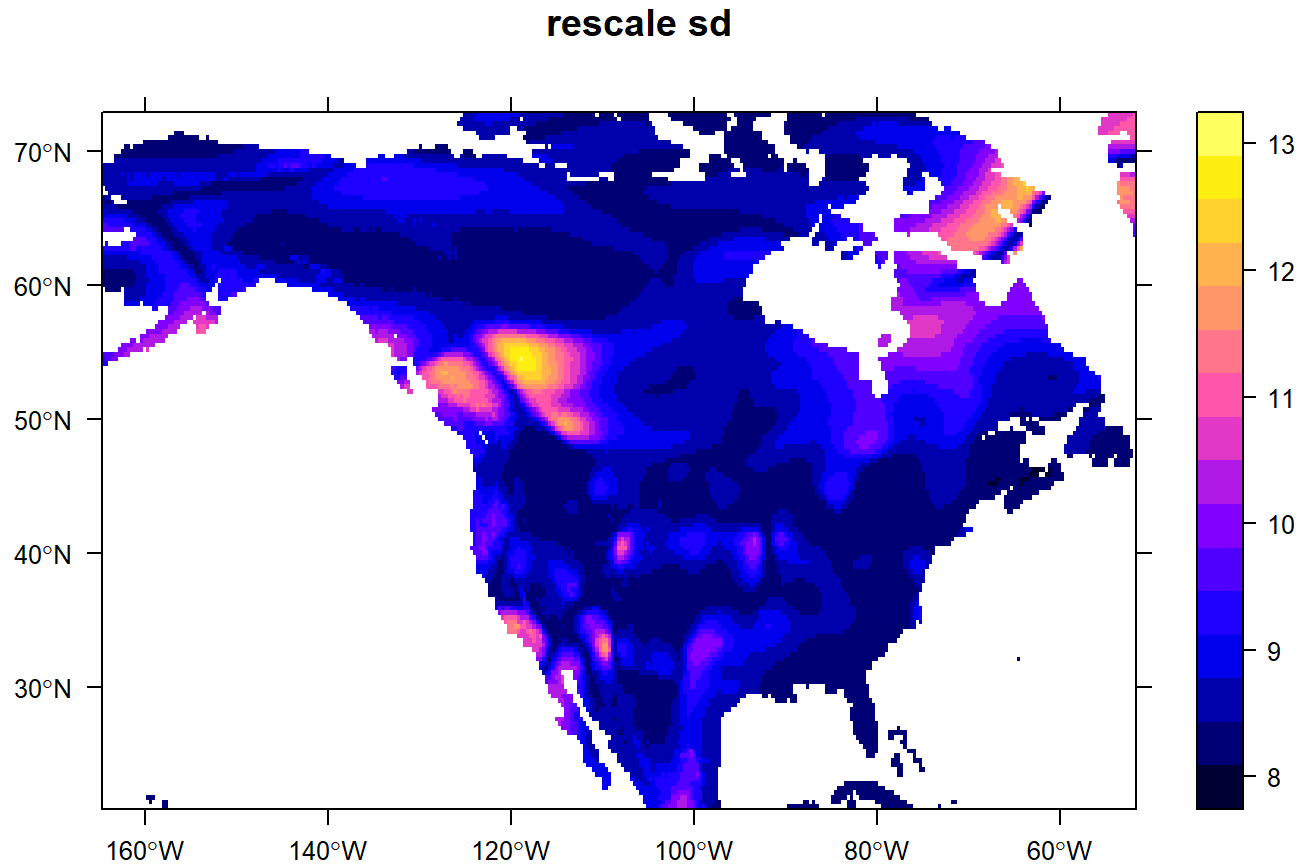
Rescale regression model



rescale mean



```
## Warning in dir.create("output"): 'output' already exists
```

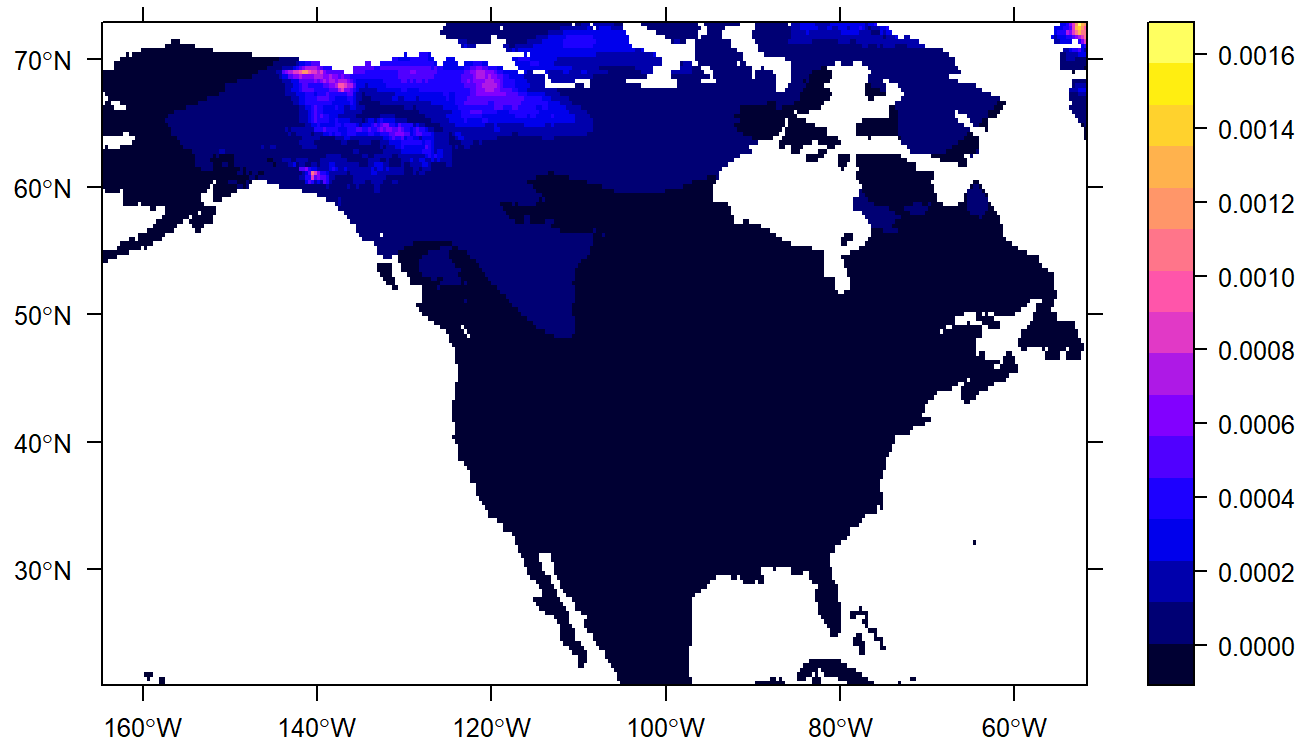
Four unknown-origin examples

```
id = letters[1:6]
d2H = seq(-160, -80, by=80/5)
un = data.frame(id,d2H)
```

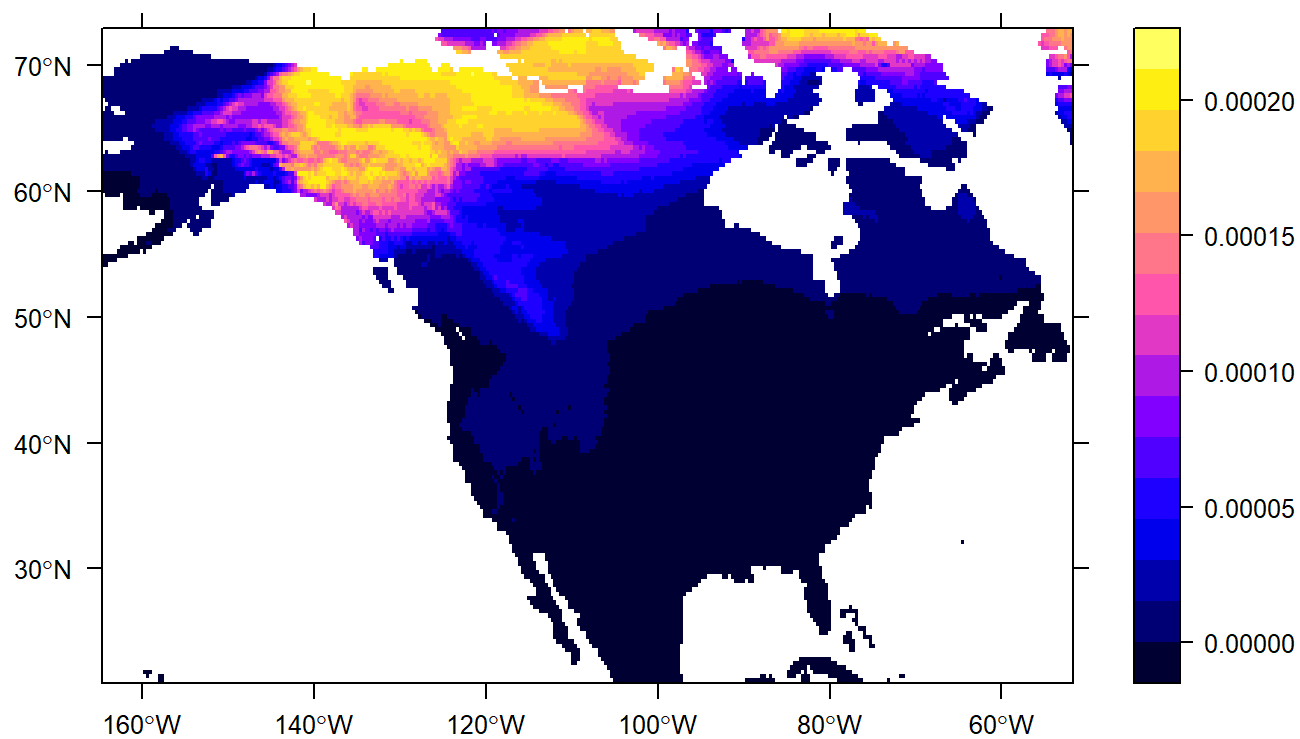
Assignment for unknown-origin examples

```
asn = pdRaster(r,unknown=un,mask=naMap)
## Warning in dir.create("output"): 'output' already exists
## Warning in dir.create("output/pdRaster_Gtif"):
'output\pdRaster_Gtif'
## already exists
```

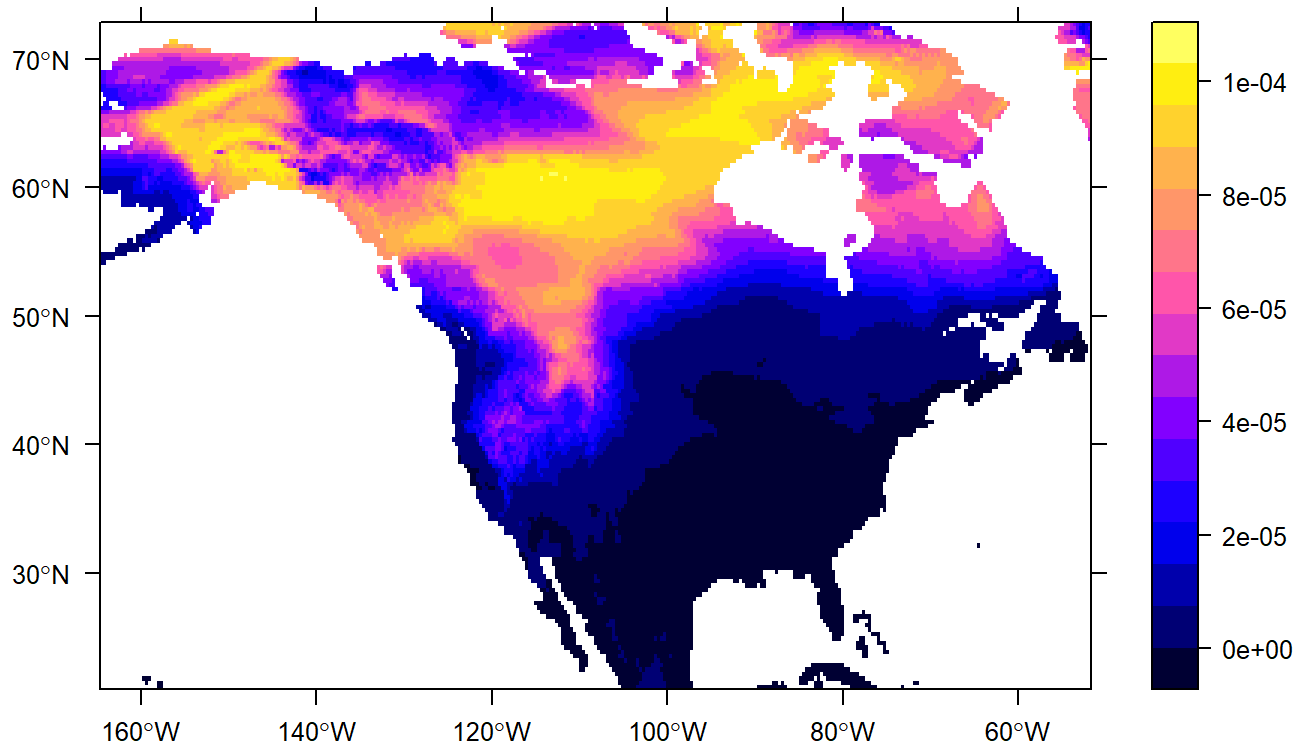
Probability Density Surface for a



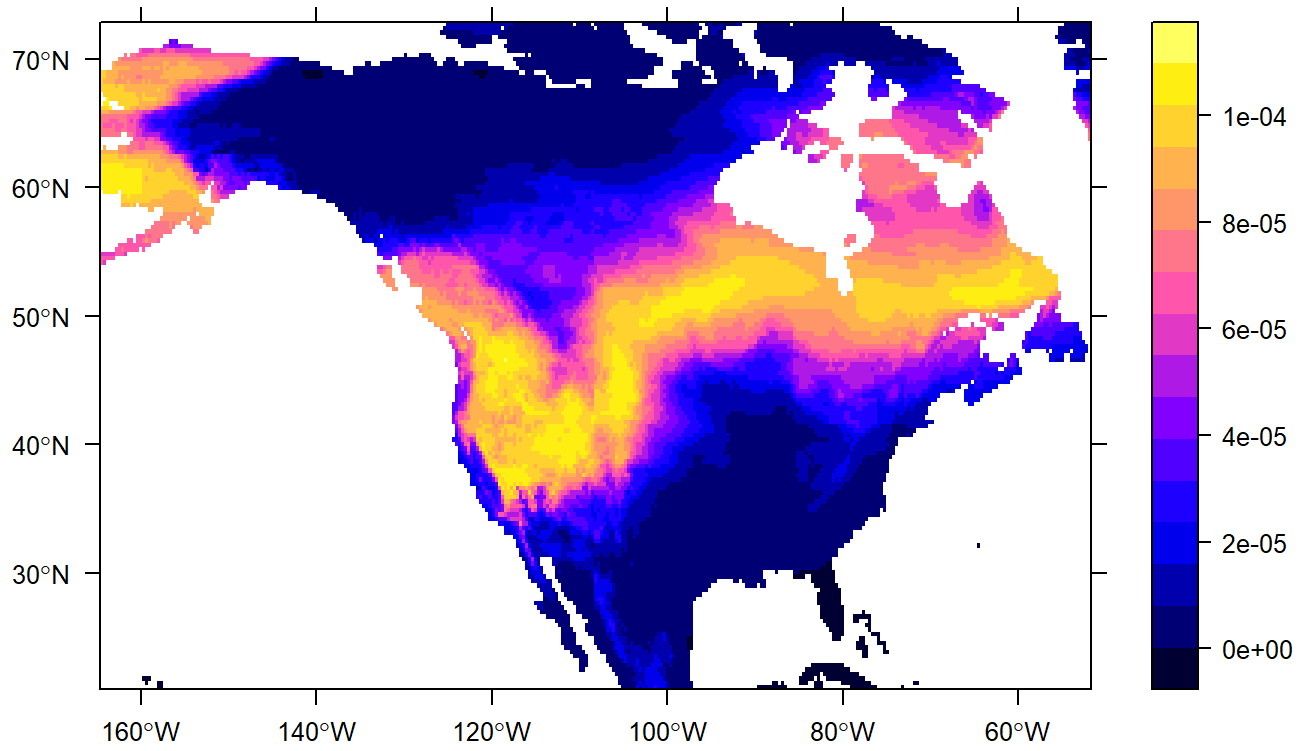
Probability Density Surface for b



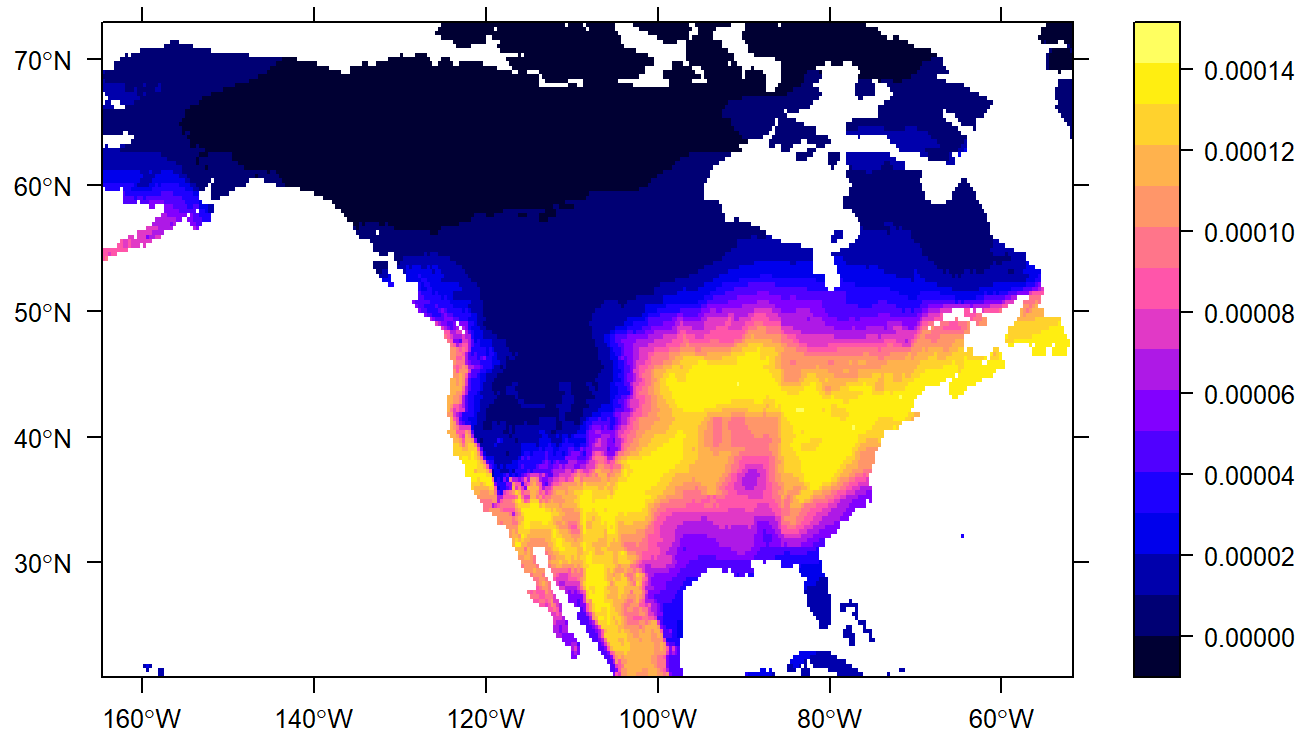
Probability Density Surface for c



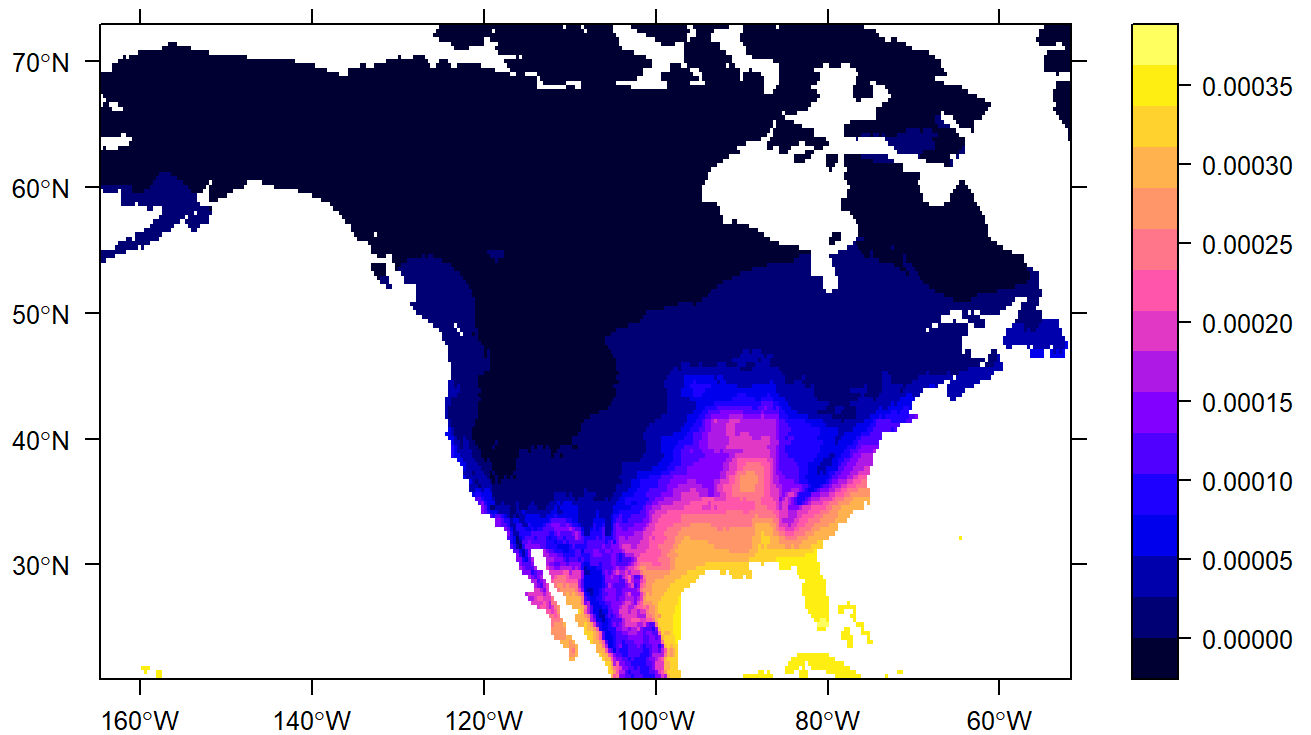
Probability Density Surface for d



Probability Density Surface for e



Probability Density Surface for f



Create SpatialPolygons with two polygons

```
p1 <- c(-100,60,-100,65,-110,65,-110,60,-100,60)
p1 <- matrix(p1, 5,2, byrow = T)
p1 <- Polygon(p1)
p1 <- Polygons(list(p1), "p1")
p2 <- c(-100,40,-100,45,-110,45,-110,40,-100,40)
p2 <- matrix(p2, 5,2, byrow = T)
p2 <- Polygon(p2)
p2 <- Polygons(list(p2), "p2")
p12 <- SpatialPolygons(list(p1,p2),1:2)
plot(p12)
```



Create data.frame with two points

```
pp1 <- c(-100,45)
pp2 <- c(-100,60)
pp12 <- as.data.frame(rbind(pp1,pp2))
```

Caculate odds ratio for the two polygons created above

```
oddsRatio(asn, p12)
## $`P1/P2_odds_ratio`
##           a           b           c           d           e
## 4.088128e+03 2.344395e+02 6.480108e+00 9.690121e-02 6.121049e-04
##           f
## 1.919561e-06
##
## $`ratio of numbers of cells in two polygons`
## [1] 1
```

Caculate odds ratio for the two points created above


```

oddsRatio(asn, ppl2)
## $`P1/P2_odds_ratio`
##      a          b          c          d          e
## 1.146839e-08 8.417012e-06 4.205954e-03 1.431251e+00 3.316179e+02
##      f
## 5.230446e+04
##
## $`odds of a pixel to the odds of the max/min pixel`
##      ratioToMax.a ratioToMax.b ratioToMax.c ratioToMax.d ratioToMax.e
## 1 1.626156e-11 2.111854e-06 0.002773832 0.02499529 1.166192978
## 2 1.080405e-02 2.359802e-01 0.257425866 0.13306854 0.003307999
##      ratioToMax.f ratioToMin.a ratioToMin.b ratioToMin.c ratioToMin.d
## 1 1.241220e-01 6.034369e+10 313.3168 1.113673e+04 9.275299e+19
## 2 9.263921e-07 1.793020e+12 1439.8675 3.025310e+11 2.208380e+13
##      ratioToMin.e ratioToMin.f
## 1 1.730175e+08 498340.6
## 2 2.018424e+01 1088710.9

```

Binary reclassification

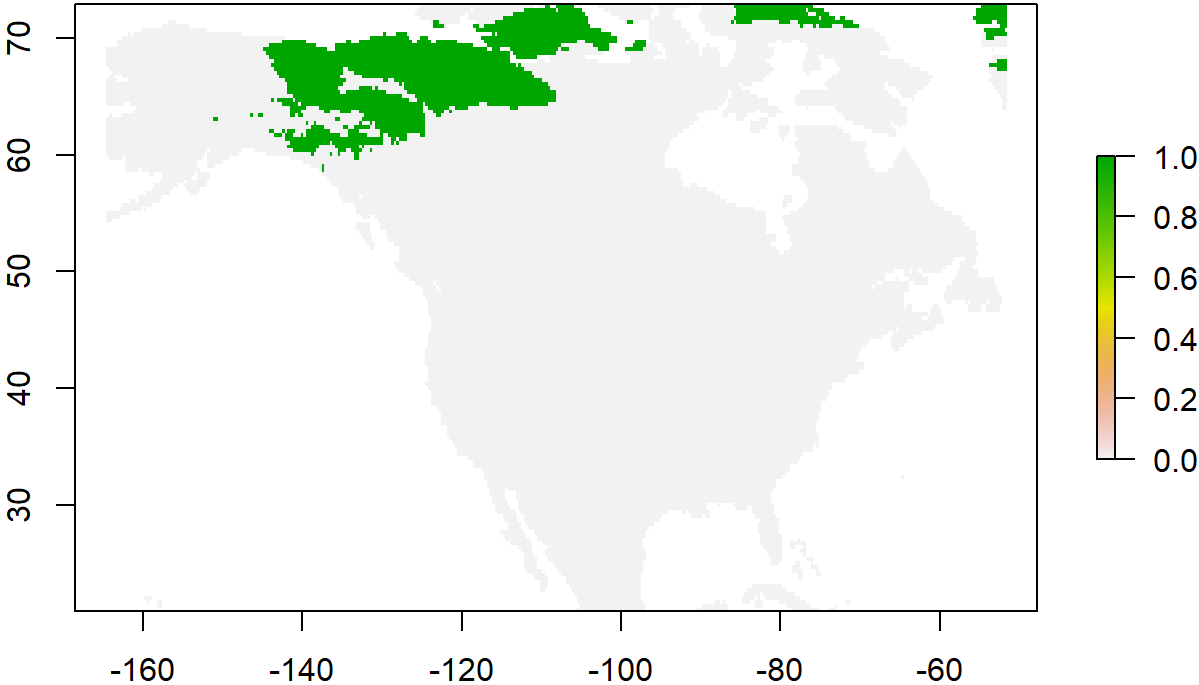
Top 10% of probability surface (defined by % area)

```

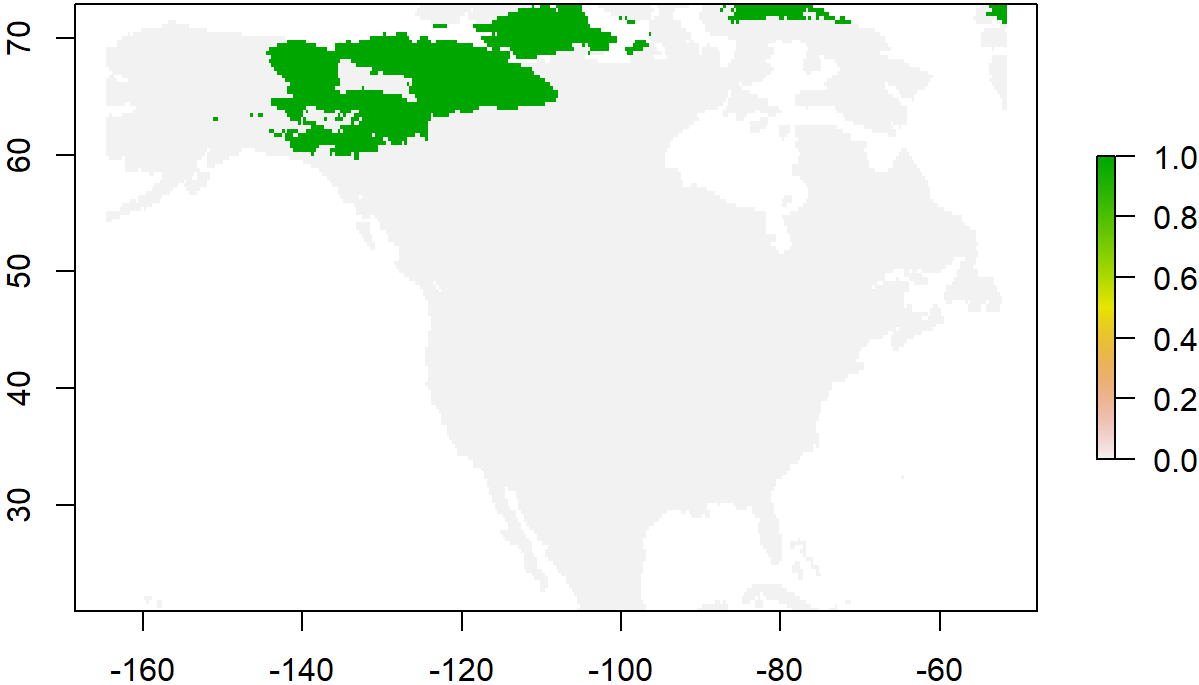
qtlRaster(asn, threshold = 0.1, thresholdType = 2)

```

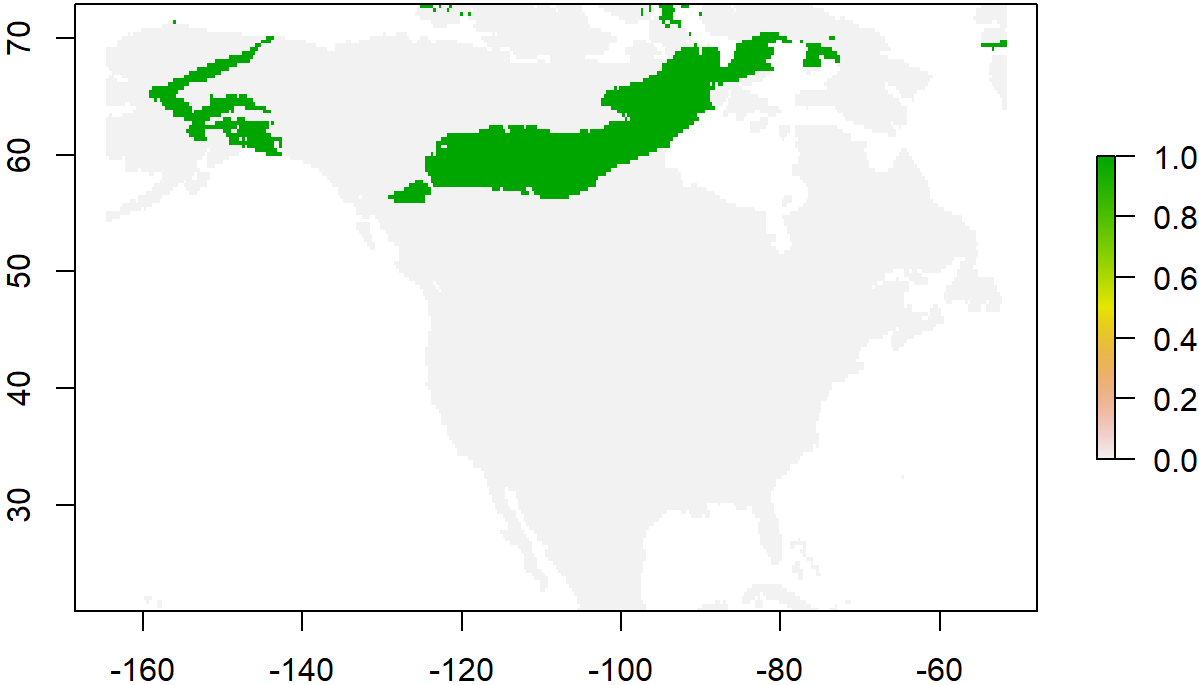
Top 10% by Area for a



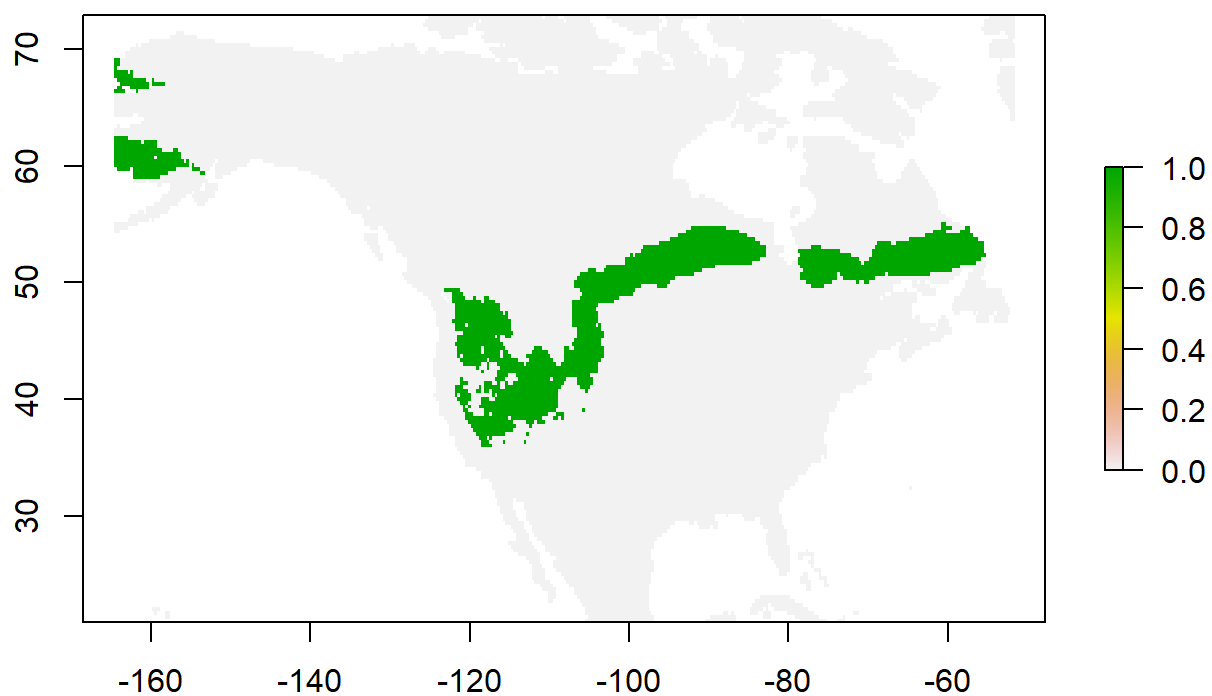
Top 10% by Area for b



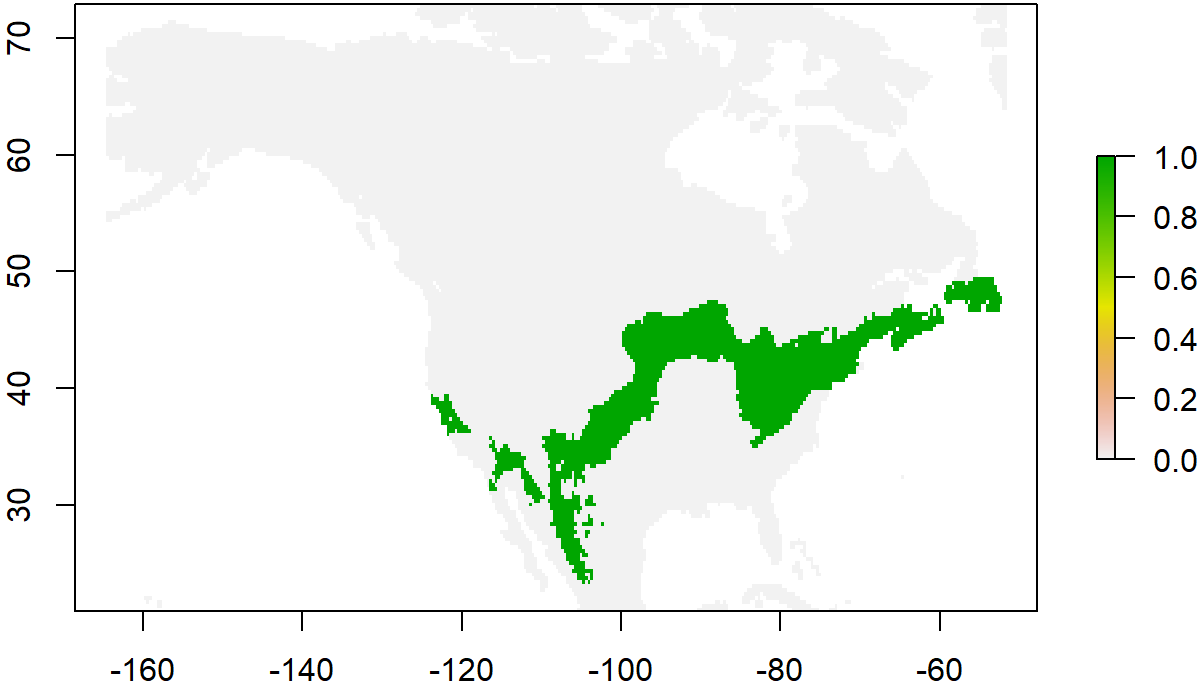
Top 10% by Area for c



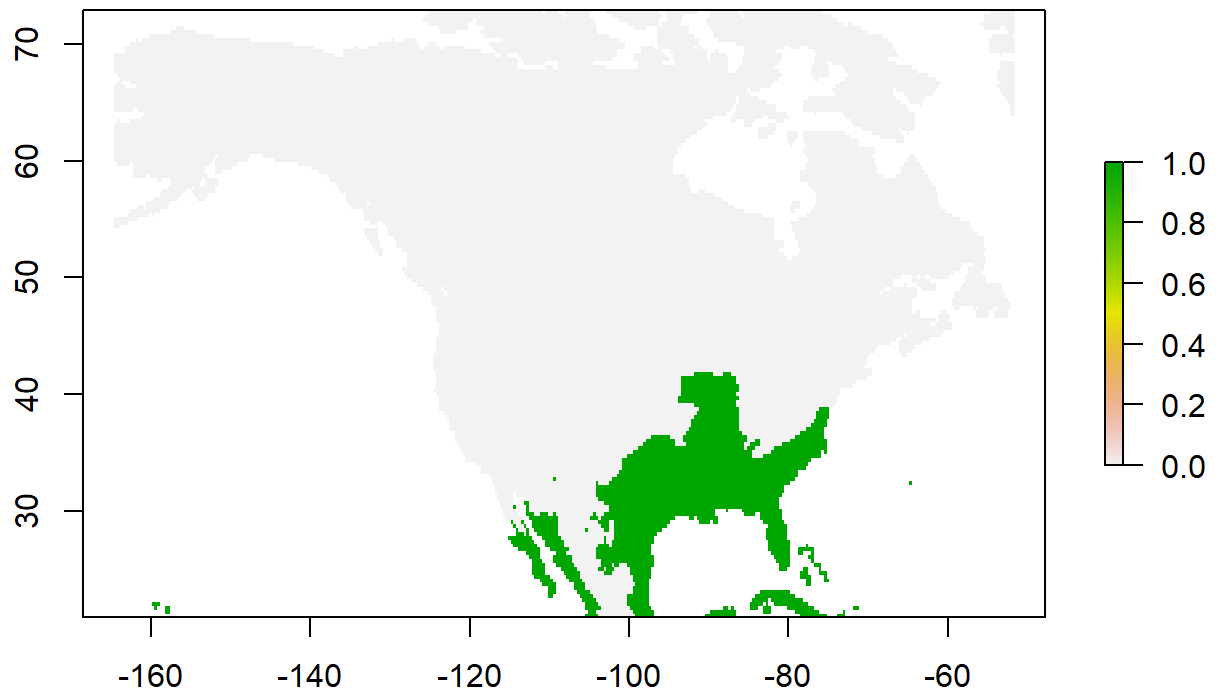
Top 10% by Area for d



Top 10% by Area for e



Top 10% by Area for f

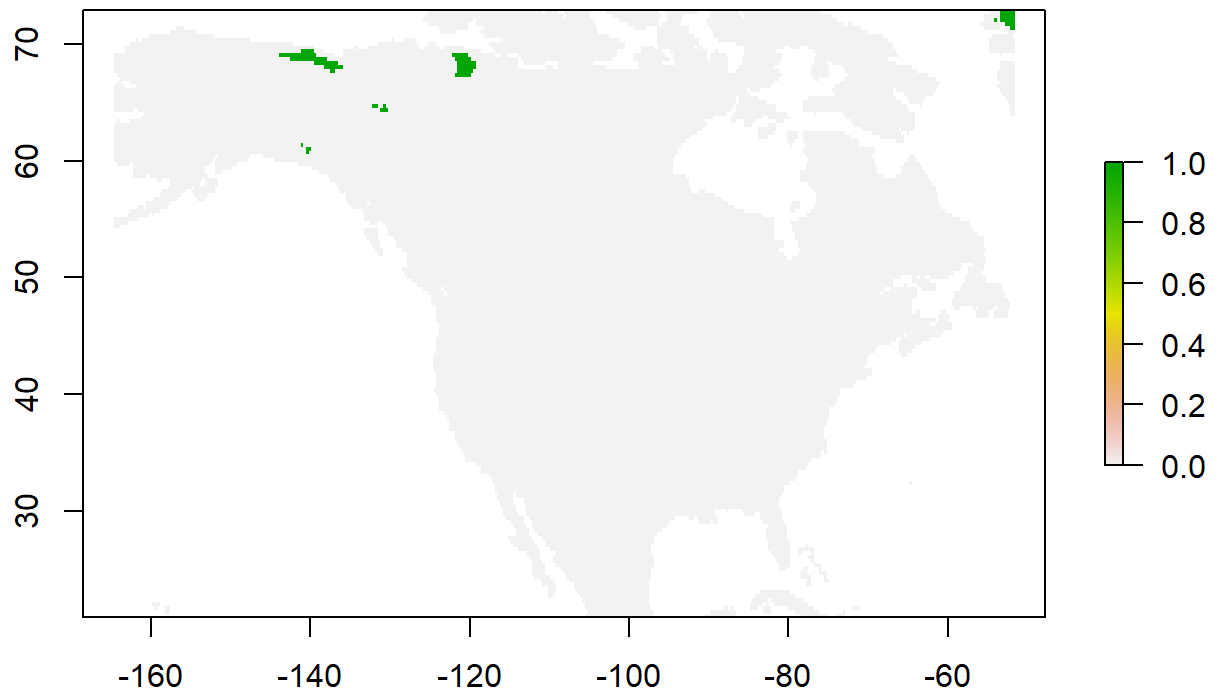


```
## class      : RasterStack
## dimensions : 156, 339, 52884, 6  (nrow, ncol, ncell, nlayers)
## resolution : 0.3333332, 0.3333332  (x, y)
## extent     : -164.6667, -51.66672, 20.91662, 72.9166  (xmin, xmax,
ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
+towgs84=0,0,0
## names      : a, b, c, d, e, f
## min values : 0, 0, 0, 0, 0, 0
## max values : 1, 1, 1, 1, 1, 1
```

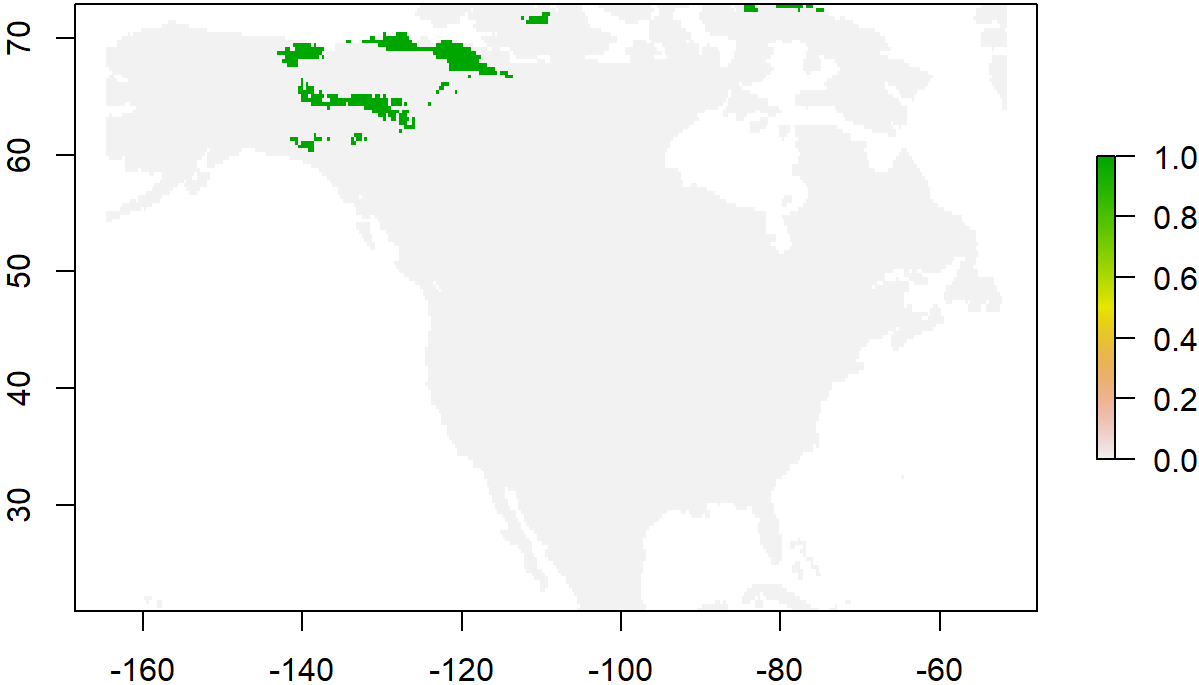
Top 10% of probability surface (defined by % cumulative probability)

```
qtlRaster(asn, threshold = 0.1, thresholdType = 1)
```

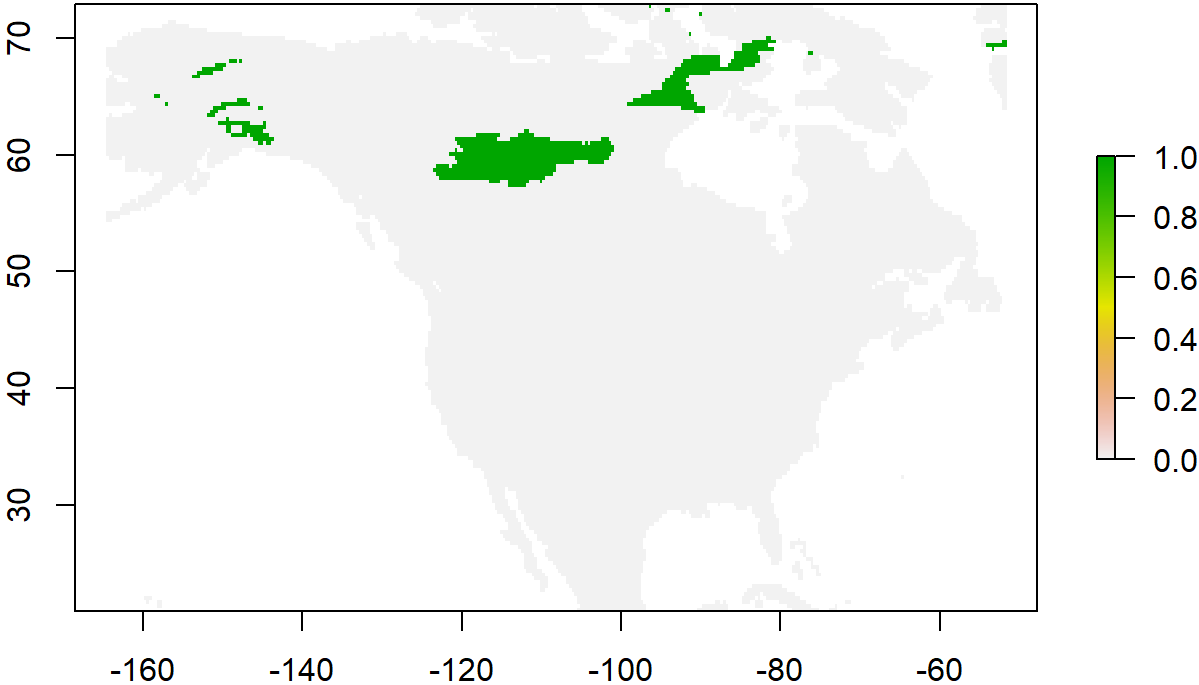
Top 10% by Cumulative Probability for a



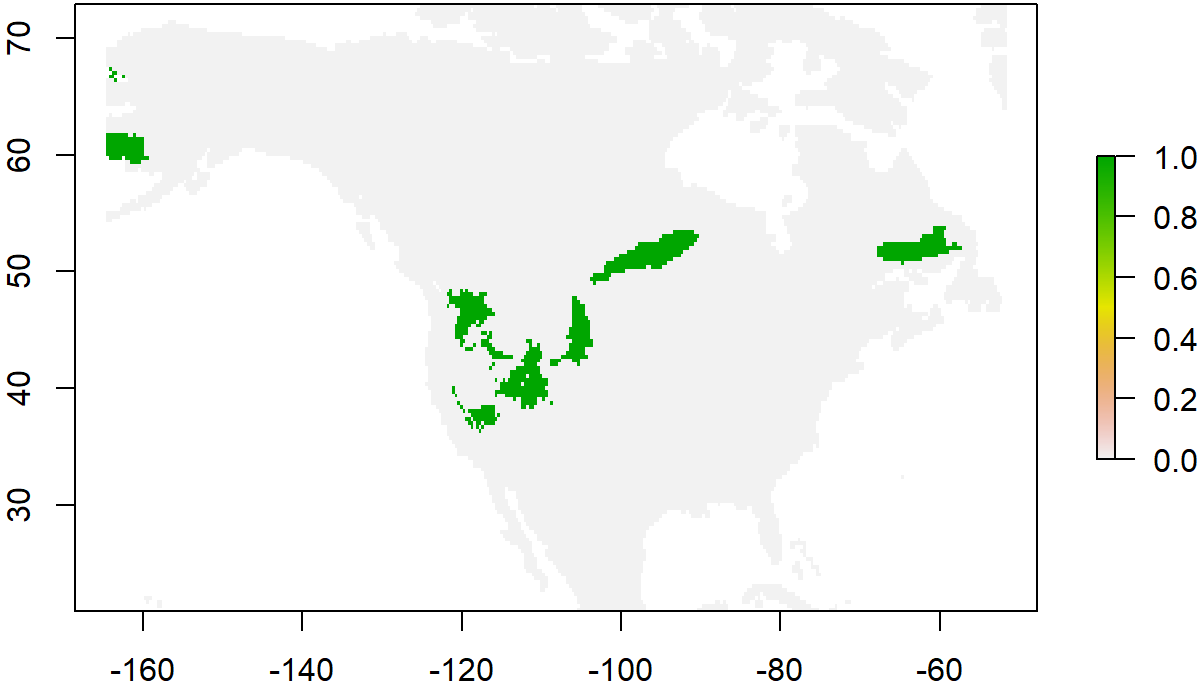
Top 10% by Cumulative Probability for b



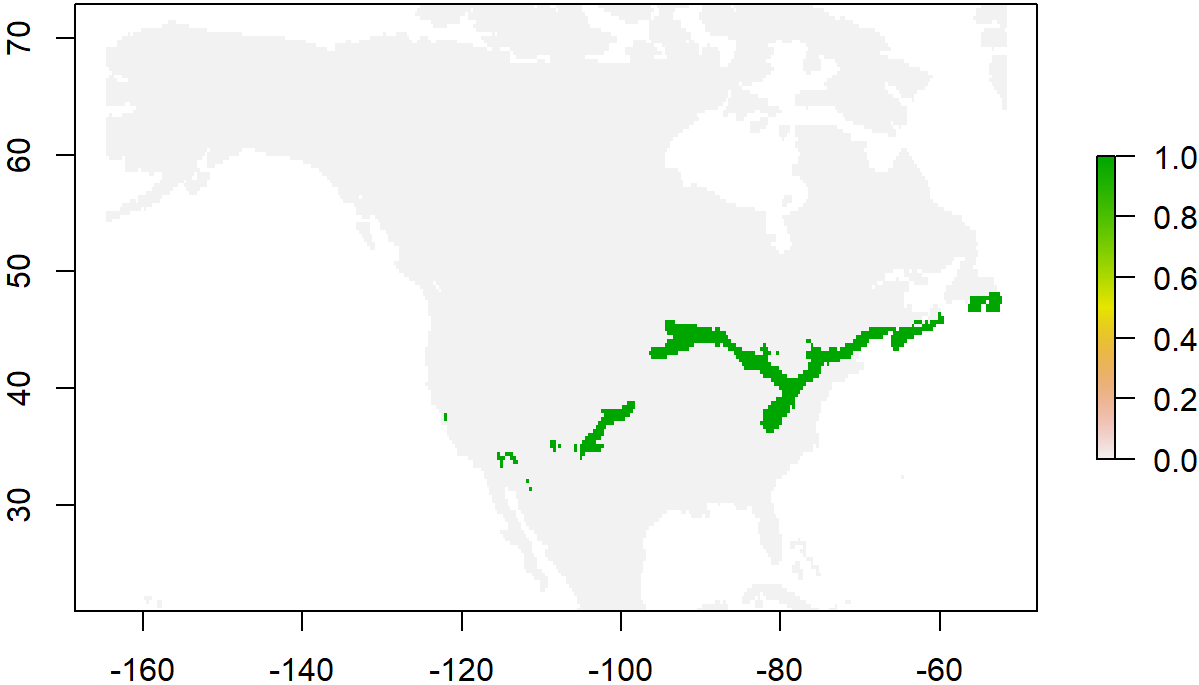
Top 10% by Cumulative Probability for c



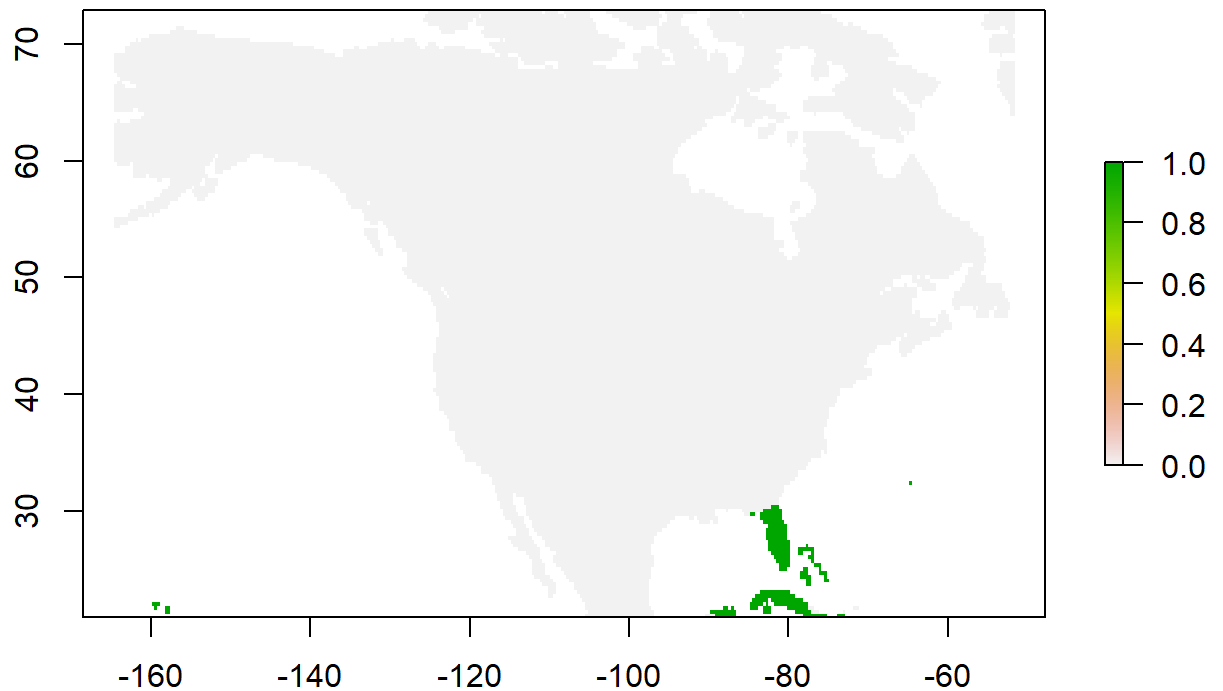
Top 10% by Cumulative Probability for d



Top 10% by Cumulative Probability for e



Top 10% by Cumulative Probability for f

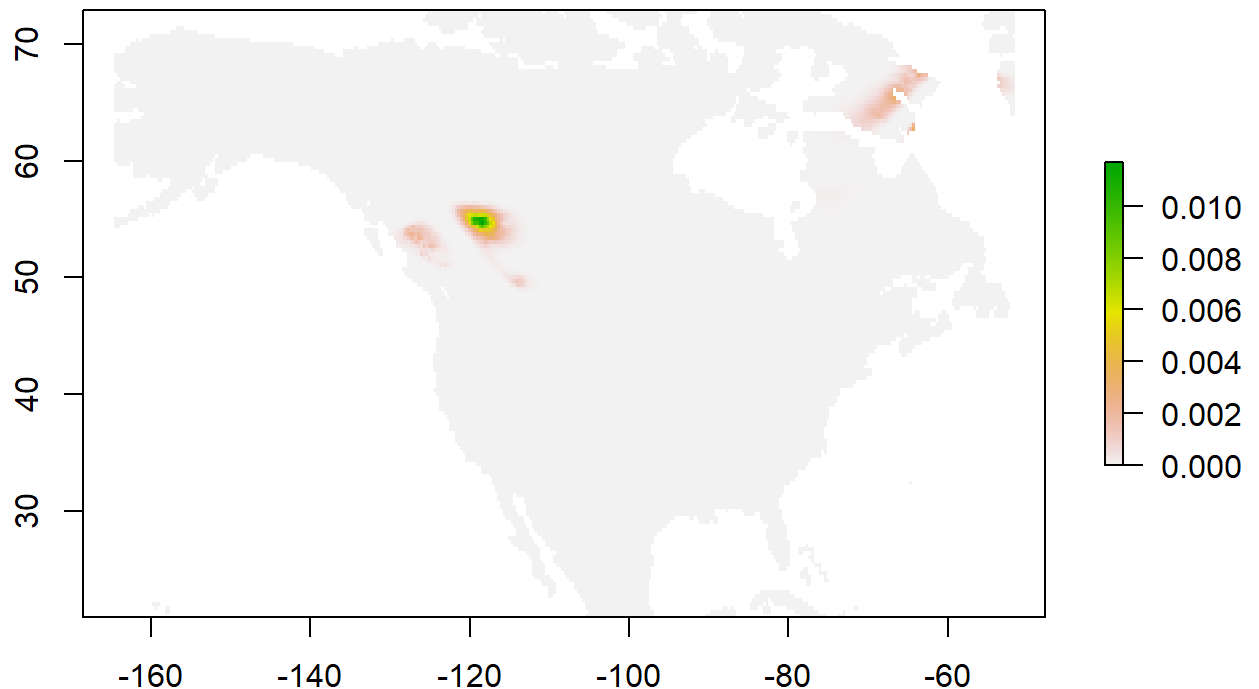


```
## class      : RasterStack
## dimensions : 156, 339, 52884, 6  (nrow, ncol, ncell, nlayers)
## resolution : 0.3333332, 0.3333332  (x, y)
## extent     : -164.6667, -51.66672, 20.91662, 72.9166  (xmin, xmax,
ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
+towgs84=0,0,0
## names      : a, b, c, d, e, f
## min values : 0, 0, 0, 0, 0, 0
## max values : 1, 1, 1, 1, 1, 1
```

Joint probability for individuals of common origin

```
jointP(asn)
```

Joint Probability

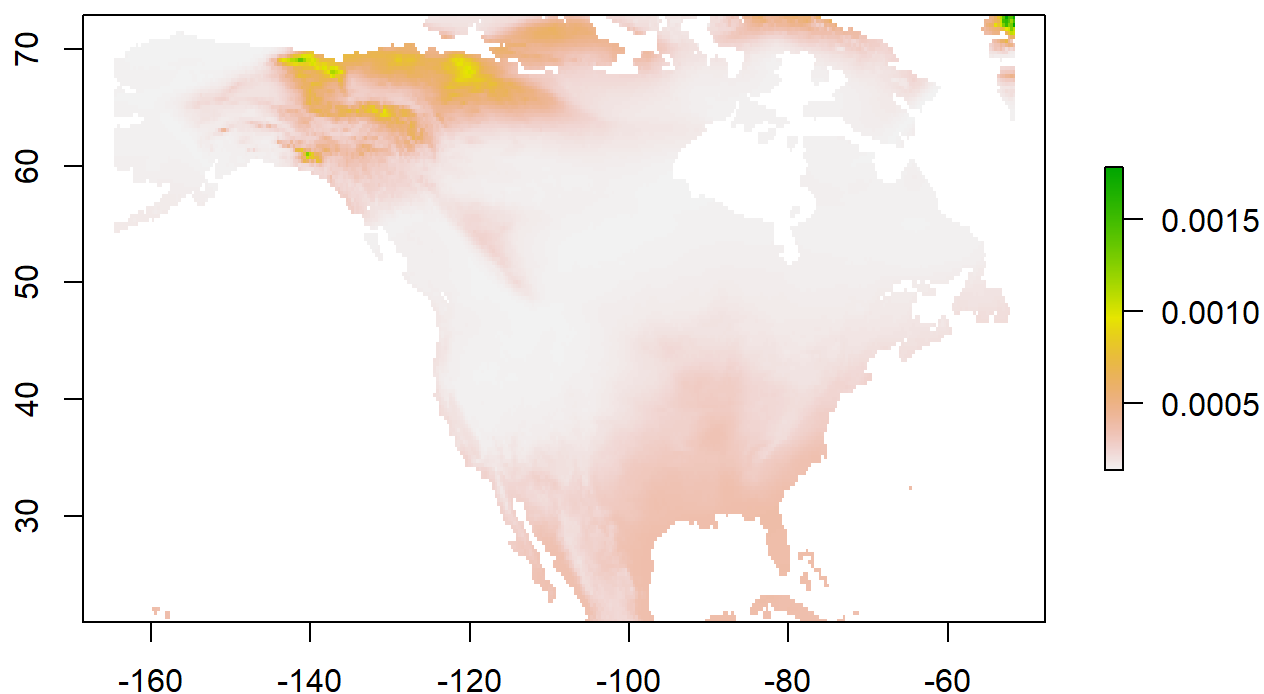


```
## class      : RasterLayer
## dimensions : 156, 339, 52884  (nrow, ncol, ncell)
## resolution : 0.3333332, 0.3333332  (x, y)
## extent     : -164.6667, -51.66672, 20.91662, 72.9166  (xmin, xmax,
ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
+towgs84=0,0,0
## source     : memory
## names      : Joint_Probability
## values     : 1.031402e-44, 0.01170466  (min, max)
```

Probability that at least one individual came from the location (union of probabilities)

```
unionP(asn)
```

Union Probability

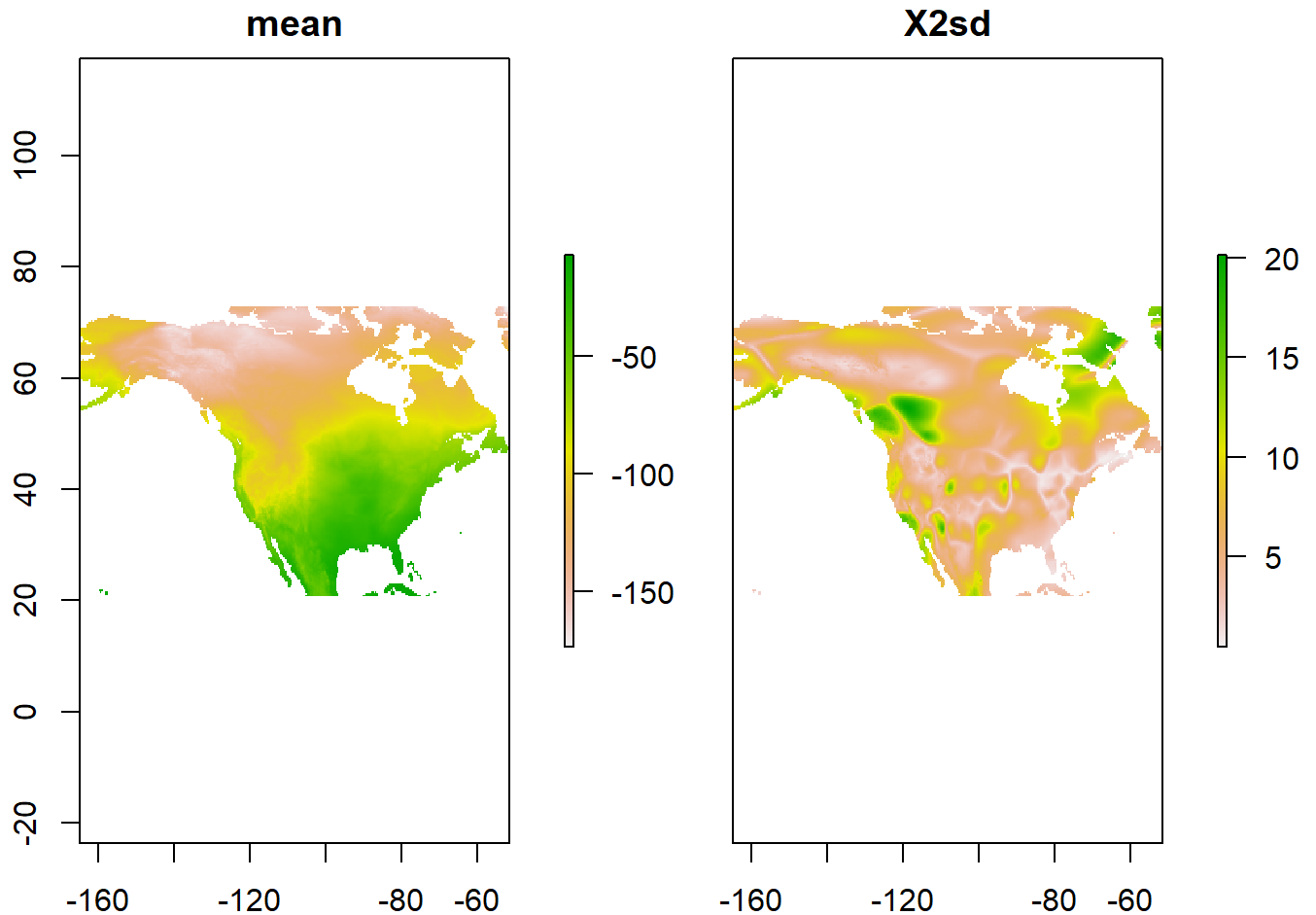


```
## class      : RasterLayer
## dimensions : 156, 339, 52884  (nrow, ncol, ncell)
## resolution : 0.3333332, 0.3333332  (x, y)
## extent     : -164.6667, -51.66672, 20.91662, 72.9166  (xmin, xmax,
ymin, ymax)
## crs        : +proj=longlat +datum=WGS84 +no_defs +ellps=WGS84
+towgs84=0,0,0
## source     : memory
## names      : layer
## values     : 0.0001361405, 0.001787688  (min, max)
```

Quality analysis of geographic assignment

```
# oxygen and hydrogen isotopes of known-origin bird
data(bird_isotope)

# crop the world hydrogen data to North America
r <- crop(d2h_world, naMap)
plot(r)
```



```
# convert 2 standard deviation from d2h_world to 1 standard deviation
r[[2]] <- r[[2]]/2
```

```
# separate the hydrogen isotope for the known-origin bird
bird_d2h <- bird_isotope[1:20,c("Longitude", "Latitude", "d2H")]
coordinates(bird_d2h) <- c(1,2)
proj4string(bird_d2h) <- proj4string(d2h_world)
```

```
# run quality assessment based hydrogen isotope from precipitation and
known-origin bird
d2h_QA <- QA(isoscape = r, known = bird_d2h, valiStation = 2,
             valiTime = 5, setSeed = T)
```

```
##
|
|
0%
|
|=====
20%
|
|=====
40%
|
|
```



```
|=====|
60%
|
|=====|
80%
|
|=====|
100%
# plot the QA result
plot(d2h_QA)
```

