参考

- 理論
  http://cogpsy.educ.kyoto-u.ac.jp/personal/Kusumi/datasem11/osanai.pdf (http://cogpsy.educ.kyoto-u.ac.jp/personal/Kusumi/datasem11/osanai.pdf)
- 実装
  https://www.juen.ac.jp/lab/okumura/handout/160201R%E3%81%AB%E3%82%88%E3%82%8B%E5%9B%A0%E5%AD%90%E5%88%86%E6%9E%90.pages.pdf (https://www.juen.ac.jp/lab/okumura/handout/160201R%E3%81%AB%E3%82%88%E3%82%8B%E5%9B%A0%E5%AD%90%E5%88%86%E6%9E%90.pages.pdf)

In [1]:
```r
library(openxlsx)
library(corrplot)
```

```
corrplot 0.84 loaded
```

In [6]:
```r
df <- read.xlsx("../data/data_covid_fix_name.xlsx")
```

In [3]:
```r
df <- na.omit(df)
# index reset
rownames(df) <- 1:nrow(df)
head(df)
```

A data.frame: 6 × 13

	country	pop	urb	gdp	dist	hf	pf	ef	date_first	detection	status	cumul	air
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	Albania	2866376	60.319	13364.155	6996524	7.84	8.005411	7.67	70	74.3	1	108641	303.14
2	Algeria	42228429	72.629	15481.788	9108277	4.99	5.201489	4.77	58	12.0	1	80272	6442.44
3	Angola	30809762	65.514	6452.355	10490120	5.40	5.979936	4.83	83	17.9	1	303691	76.94
4	Argentina	44494502	91.870	20610.569	19025624	6.86	8.044600	5.67	65	74.9	1	92122	1516.63
5	Australia	24992369	86.012	51663.365	7608913	8.62	9.160533	8.07	26	97.3	1	1347	75667.65
6	Austria	8847037	58.297	55454.689	7103537	8.48	9.245685	7.71	58	73.2	1	80272	12935.50

In [4]:
```r
data = df[,c(2,3,4,5,6,9,10,13)]

# log
data = log(data)
# 標準化
data= scale(data)

head(data)
```

A matrix: 6 × 8 of type dbl

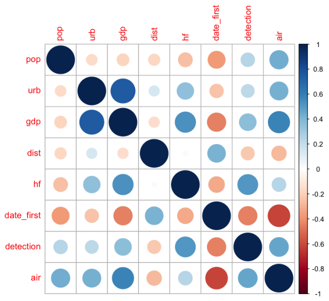
	pop	urb	gdp	dist	hf	date_first	detection	air
1	-0.8304081	0.14522960	0.00803967	-0.10000665	0.83937737	0.40068236	0.9784892	-0.7001144
2	0.8179593	0.55943437	0.13185308	0.37723785	-1.86965358	-0.09390578	-2.1891023	0.4308812
3	0.6247775	0.32948958	-0.60487411	0.63280474	-1.39618757	0.84870044	-1.4943360	-1.2074928
4	0.8499898	1.08357748	0.37271488	1.70998847	0.03871705	0.20577415	0.9924628	-0.1043414
5	0.4965495	0.93662818	1.14624886	0.05180777	1.40807981	-2.20412299	1.4470387	1.3424378
6	-0.1398002	0.06918419	1.20586029	-0.07254229	1.30989675	-0.09390578	0.9525753	0.6888197

In [84]:
```r
# 相関行列
cor_matrix = cor(data)
round(cor_matrix, 2)
```

A matrix: 8 × 8 of type dbl

	pop	urb	gdp	dist	hf	date_first	detection	air
pop	1.00	-0.15	-0.19	-0.17	-0.26	-0.38	0.24	0.44
urb	-0.15	1.00	0.75	0.14	0.36	-0.24	0.22	0.42
gdp	-0.19	0.75	1.00	-0.15	0.54	-0.44	0.37	0.59
dist	-0.17	0.14	-0.15	1.00	0.03	0.41	-0.23	-0.28
hf	-0.26	0.36	0.54	0.03	1.00	-0.32	0.52	0.25
date_first	-0.38	-0.24	-0.44	0.41	-0.32	1.00	-0.44	-0.61
detection	0.24	0.22	0.37	-0.23	0.52	-0.44	1.00	0.46
air	0.44	0.42	0.59	-0.28	0.25	-0.61	0.46	1.00

In [81]:
```r
# urb - gdp (0.75)
# detection - hf (0.52)
# air - date_first (-0.61)
corrplot(cor_matrix)
```



In [85]:
```r
# 固有値
eigenvalue = eigen(cor_matrix)$values
print(eigenvalue)
```

```
[1] 3.2940895 1.8139078 0.9550529 0.8506228 0.4465154 0.2992735 0.2103101
[8] 0.1302280
```

In [7]:

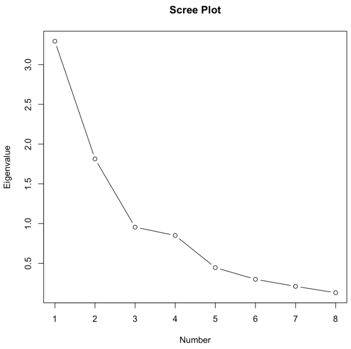
```
# 固有ベクトル
eigenvector = eigen(cor_matrix)$vectors
eigenvector
```

A matrix: 8 × 8 of type dbl

```
-0.1101945  0.6021132  0.37923281  0.33290730  0.01672323 -0.03692528 -0.52614993  0.30363663
-0.3457896 -0.3890415  0.44260838 -0.15011569  0.20900561 -0.46377227 -0.32851405 -0.37940713
-0.4510701 -0.3017049  0.10691483 -0.28538511  0.06413042  0.13021587  0.08004694  0.76495625
 0.1786327 -0.3906701  0.48211330  0.63503899 -0.26090395  0.02074917  0.31625210  0.10603141
-0.3388959 -0.3337128 -0.44309998  0.35230334 -0.32181234  0.29466419 -0.50109385 -0.10883428
 0.4146963 -0.2794451  0.04856175  0.08118728  0.69186572  0.41580075 -0.29301114  0.05969747
-0.3819424  0.1044106 -0.33972915  0.49939228  0.54213713 -0.28663592  0.31929542  0.01968484
-0.4474476  0.2051878  0.31985083 -0.04262807  0.08774613  0.65137370  0.26442717 -0.38943306
```

In [8]:

```
# スクリープロット
# 固有値の1以上の数を観察(この場合2)
# 固有値の1以上の数が共通因子の数に関係する
plot(eigenvalue, type="b", main="Scree Plot", xlab="Number", ylab="Eigenvalue")
```



因子分析

In [92]:

```
# 因子の回転法を“varimax”もしくは“promax”から選択する。回転させない場合は“none”と指定。
# factor:共通因子の数を指定する。
# scores:回帰方法 (regression) とバートレット法 (Bartlett)
fit0<- factanal(x=data, factors=2, rotation="none",scores="Bartlett")
fit1<- factanal(x=data, factors=2, rotation="varimax",scores="Bartlett")
fit2<- factanal(x=data, factors=2, rotation="promax",scores="Bartlett")
```

結果の読み取り

- loadings　因子負荷量
- correlation　相関係数
- factors　求めた因子数
- STATISTIC　カイ2乗値
- dof　カイ2乗決定の自由度
- PVAL　カイ2乗統計量のP値(元のデータの分散と指定した共通因子のモデルに基づいて求めたデータの分散との間の有意差に関する検定統計量である。この検定統計量は、探索的に因子分析を行う際の因子の数を決める際の1つの参考材料となる)

In [93]:

```
# cutoff=0 は(値の小さいものも含め)全ての因子負荷の値を表示させる指示
# Uniquenesses:独自性 (=1-共通性)
# 因子の回転によって因子負荷と寄与の値は変化するが、独自性の値は変化しない(ということは共通性の値も変化しない)
# 共通性の値が1以上,あるいは0に近い値でないか注意
#

# 因子負荷量は相関係数なので0から±1の値をとる
print(fit0, cutoff=0)
```

```
Call:
factanal(x = data, factors = 2, scores = "Bartlett", rotation = "none")

Uniquenesses:
       pop        urb        gdp       dist         hf date_first  detection
     0.286      0.427      0.005      0.873      0.702      0.476      0.712
       air
     0.226

Loadings:
           Factor1 Factor2
pop        -0.180   0.825
urb         0.754  -0.068
gdp         0.997  -0.009
dist       -0.149  -0.324
hf          0.540  -0.081
date_first -0.447  -0.569
detection   0.372   0.387
air         0.596   0.646

               Factor1 Factor2
SS loadings      2.603   1.690
Proportion Var   0.325   0.211
Cumulative Var   0.325   0.537

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 104.92 on 13 degrees of freedom.
The p-value is 1.83e-16
```

In [77]:

```
tmp = promax(fit0$loadings)$rotmat
solve(t(tmp)  %*% tmp)
```

A matrix: 2 × 2 of type dbl

```
1.0000000 0.2854474
0.2854474 1.0000000
```

varimax(直行回転)

In [64]:

```
# 相関が出ない
# distとhf、detectionに関しては独自因子が0.5を超えているので、これらの変数は共通因子で説明できない情報が半分を超えていると言える。
print(fit1, cutoff=0)
```

```
Call:
factanal(x = data, factors = 2, scores = "Bartlett", rotation = "varimax")

Uniquenesses:
      pop       urb       gdp      dist        hf date_first  detection
    0.286     0.427     0.005     0.873     0.702     0.476     0.712
      air
    0.226

Loadings:
           Factor1 Factor2
pop        -0.402   0.743
urb         0.743   0.143
gdp         0.961   0.268
dist       -0.053  -0.353
hf          0.541   0.072
date_first -0.272  -0.671
detection   0.250   0.475
air         0.394   0.786

               Factor1 Factor2
SS loadings      2.225   2.068
Proportion Var   0.278   0.259
Cumulative Var   0.278   0.537

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 104.92 on 13 degrees of freedom.
The p-value is 1.83e-16
```

In [78]:

```
tmp = promax(fit1$loadings)$rotmat
solve(t(tmp)  %*% tmp)
```

A matrix: 2 × 2 of type dbl

1.0000000	0.2854632
0.2854632	1.0000000

promax(斜交回転)

In [63]:

```
# 因子間の相関を仮定
# but, 相関はなかった
print(fit2, cutoff=0)
```

```
Call:
factanal(x = data, factors = 2, scores = "Bartlett", rotation = "promax")

Uniquenesses:
      pop       urb       gdp      dist        hf date_first  detection
    0.286     0.427     0.005     0.873     0.702     0.476     0.712
      air
    0.226

Loadings:
           Factor1 Factor2
pop        -0.487   0.843
urb         0.759  -0.008
gdp         0.974   0.074
dist       -0.023  -0.350
hf          0.556  -0.039
date_first -0.220  -0.630
detection   0.216   0.434
air         0.336   0.722

               Factor1 Factor2
SS loadings      2.279   1.947
Proportion Var   0.285   0.243
Cumulative Var   0.285   0.528

Factor Correlations:
        Factor1 Factor2
Factor1   1.000  -0.285
Factor2  -0.285   1.000

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 104.92 on 13 degrees of freedom.
The p-value is 1.83e-16
```

In [1]:

```
tmp = promax(fit2$loadings)$rotmat
solve(t(tmp)  %*% tmp)
```

```
Error in ncol(x):  オブジェクト 'fit2' がありません
Traceback:

1. promax(fit2$loadings)
2. ncol(x)
```

結果の解釈

- fit2では因子間の相関を仮定してpromax(斜交回転)で因子分析をした。
  しかし、相関は得られなかった。(相関係数-0.285)
  よって相関をないと仮定する
- fi1では因子間の相関はないと仮定してvarimax(直行回転)で因子分析をした。
  結果、因子間相関はなかった。(相関係数0.285)

In [13]:

```
unclass(fit1$loadings)
```

A matrix: 8 × 2 of type dbl

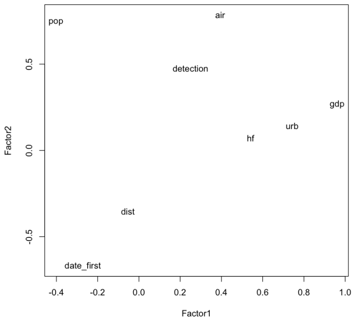
	Factor1	Factor2
pop	-0.4020358	0.74297671
urb	0.7430598	0.14301895
gdp	0.9608995	0.26771818
dist	-0.0534007	-0.35297926
hf	0.5413911	0.07163641
date_first	-0.2719973	-0.67102067
detection	0.2500562	0.47489143
air	0.3939073	0.78635833

因子負荷量の可視化

- factor1・・・gdp, urb, hfが影響(先進国)
- factor2・・・pop, air, data_firstが影響()

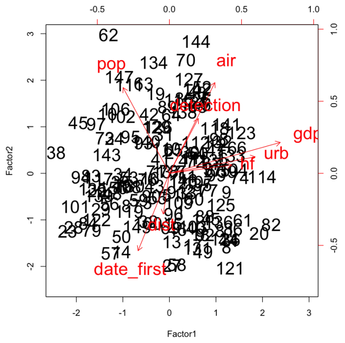
In [96]:

```
# 因子負荷量の散布図(値は相関係数)
plot(fit1$loadings[,1:2],type="n")
text(fit1$loadings[,1:2],colnames(data))
```



In [14]:

```
# 因子得点の散布図と因子負荷量の散布図
biplot(fit1$scores,fit1$loading,cex=2)
```



In [15]:

```
# factor2(第二共通因子)までの累積寄与率は、0.537(0.5を超えておけばok?)
fit1
```

```
Call:
factanal(x = data, factors = 2, scores = "Bartlett", rotation = "varimax")

Uniquenesses:
      pop       urb       gdp      dist        hf date_first  detection
    0.286     0.427     0.005     0.873     0.702     0.476     0.712
      air
    0.226

Loadings:
          Factor1 Factor2
pop       -0.402   0.743
urb        0.743   0.143
gdp        0.961   0.268
dist              -0.353
hf         0.541
date_first -0.272  -0.671
detection  0.250   0.475
air        0.394   0.786

               Factor1 Factor2
SS loadings      2.225   2.068
Proportion Var   0.278   0.259
Cumulative Var   0.278   0.537

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 104.92 on 13 degrees of freedom.
The p-value is 1.83e-16
```

In [61]:

```
score_df = as.data.frame(fit1$score)
head(score_df)
```

A data.frame: 6 × 2

	Factor1	Factor2
	<dbl>	<dbl>
1	0.239466982	-0.8067330
2	0.001249695	0.4462947
3	-0.466127050	-0.6306339
4	0.306802239	0.2783279
5	0.792748876	1.4713779
6	1.161208313	0.3134312

In [55]:

```
# Factor1の因子スコアが多い国の確認

#tmp = subset(score_df, (Factor1 >1.5) | (Factor1 < -1.5))
tmp = subset(score_df, Factor1 >1.5)
index_f1 = as.integer(rownames(tmp))

df[index_f1,]
```

A data.frame: 6 × 13

	country	pop	urb	gdp	dist	hf	pf	ef	date_first	detection	status	cumul	air
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
20	Brunei	428962	77.629	80920.05	3719566	5.69	4.760902	6.62	71	30.5	1	113067	1234.45
61	Iceland	353574	93.813	57303.06	7695516	8.41	9.084634	7.74	61	37.2	1	84456	7819.74
74	Kuwait	4137309	100.000	72897.56	5244993	6.19	5.968881	6.41	56	47.5	1	78620	6464.85
82	Luxembourg	607728	90.981	113337.42	7510071	8.56	9.263800	7.86	62	41.7	1	86276	2099.10
114	Qatar	2781677	99.135	126898.43	5117748	6.15	5.238267	7.07	62	32.7	1	86276	29178.92
123	Singapore	5638676	100.000	101531.63	3897578	8.11	7.502452	8.71	25	64.5	1	896	40401.51

In [56]:

```
# Factor2の因子スコアが多い国の確認
tmp = subset(score_df, Factor2 >1.5)
index_f2 = as.integer(rownames(tmp))

df[index_f2,]
```

A data.frame: 16 × 13

	country	pop	urb	gdp	dist	hf	pf	ef	date_first	detection	status	cumul	air
	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
19	Brazil	209469333	86.569	16096.401	16335606	6.48	6.739923	6.23	58	82.4	1	80272	102109.98
26	Canada	37058856	81.411	48130.256	8965622	8.65	9.222964	8.08	27	96.4	1	2020	89380.00
48	France	66987244	80.444	45342.396	8550121	8.02	8.693716	7.35	26	75.3	1	1347	70188.03
52	Germany	82927922	77.312	53074.540	7170071	8.53	9.247491	7.82	29	84.6	1	4580	109796.20
62	India	1352617328	34.030	7762.882	2777278	6.64	6.368844	6.91	31	47.4	1	7815	164035.64
63	Indonesia	267663435	55.325	13079.619	4514121	6.83	6.384634	7.27	63	68.1	1	88313	115154.10
70	Japan	126529100	91.616	42797.459	3024190	8.28	8.699179	7.86	16	70.1	1	61	126387.53
85	Malaysia	31528585	76.036	31782.153	3678054	6.52	5.698542	7.34	26	73.2	1	1347	60481.77
111	Philippines	106651922	46.907	8951.086	3345565	6.88	6.442997	7.32	31	63.6	1	7815	43080.12
116	Russia	144478050	74.433	27588.125	2870579	6.34	5.899881	6.78	64	34.1	1	89882	99327.31
127	South Korea	51635256	81.459	40111.776	2146566	8.20	8.814165	7.59	21	92.1	1	239	88157.88
128	Spain	46723749	80.321	39715.439	8741360	8.12	8.686116	7.55	33	83.0	1	11936	80672.10
134	Thailand	69428524	49.949	19051.333	2392656	6.55	6.247297	6.86	14	81.0	1	60	76053.04
142	UK	66488991	83.398	45973.574	7765919	8.47	8.849797	8.09	32	87.3	1	9817	165388.61
144	USA	327167434	82.256	62794.586	10196715	8.46	8.720049	8.19	22	98.2	1	391	889022.00
147	Vietnam	95540395	35.919	7447.814	2220676	6.29	6.310973	6.27	25	57.4	1	896	47049.67

可視化(各変数の大小の確認)

In [184]:

```
library(classInt)
```

In [205]:

```
library(classInt)
#b5 n等分する、style=によって分け方変わるので注意
b5<-classIntervals(as.integer(df$air),n=5)

#c5はb5で分けした数(length(b5$brks)の色を用意
c5<-rev(heat.colors(length(b5$brks)-1))
#b.colで区分の色を決める
b.col<-findColours(b5,c5)
```

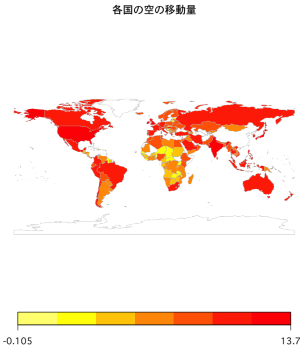
In [229]:

```
tmp = df
tmp["air"] = log(tmp["air"])

map_obj <- joinCountryData2Map(tmp, joinCode="NAME", nameJoinColumn="country")
par(family="Osaka")
mapCountryData(map_obj, nameColumnToPlot="air"
            , catMethod="fixedWidth"
            , mapTitle = "各国の空の移動量"
            #, colourPalette=heat.colors(7)
            #,colourPalette=c("#A6CEE3", "#B2DF8A", "#FB9A99") # 赤、緑、青
            #,colourPalette=b.col
            , addLegend = TRUE)
```

```
149 codes from your data successfully matched countries in the map
1 codes from your data failed to match with a country code in the map
94 codes from the map weren't represented in your data

Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(-0.105360515657826, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(-0.105360515657826, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(-0.105360515657826, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(-0.105360515657826, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(-0.105360515657826, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(-0.105360515657826, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in title(mapTitle):
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in title(mapTitle):
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
```



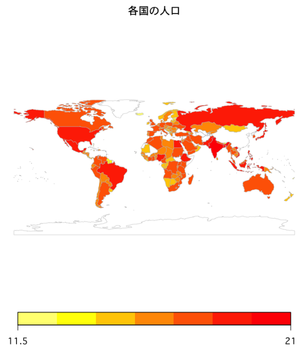
In [228]:

```
tmp = df
tmp["pop"] = log(tmp["pop"])

map_obj <- joinCountryData2Map(tmp, joinCode="NAME", nameJoinColumn="country")
par(family="Osaka")
mapCountryData(map_obj, nameColumnToPlot="pop"
            , catMethod="fixedWidth"
            , mapTitle = "各国の人口"
            #, colourPalette=heat.colors(7)
            #,colourPalette=c("#A6CEE3", "#B2DF8A", "#FB9A99") # 赤、緑、青
            #,colourPalette=b.col
            , addLegend = TRUE)
```

```
149 codes from your data successfully matched countries in the map
1 codes from your data failed to match with a country code in the map
94 codes from the map weren't represented in your data

Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(11.4800096342086, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(11.4800096342086, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(11.4800096342086, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(11.4800096342086, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(11.4800096342086, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(11.4800096342086, :
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in title(mapTitle):
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
Warning message in title(mapTitle):
“ ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません ”
```

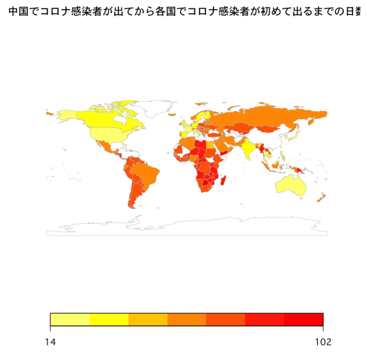


In [227]:

```
map_obj <- joinCountryData2Map(df, joinCode="NAME", nameJoinColumn="country")
par(family="Osaka")
mapCountryData(map_obj, nameColumnToPlot="date_first"
            , catMethod="fixedWidth"
            , mapTitle = "中国でコロナ感染者が出てから各国でコロナ感染者が初めて出るまでの日数"
            , colourPalette=rev(heat.colors(7))
            #,colourPalette=c("#A6CEE3", "#B2DF8A", "#FB9A99") # 赤、緑、青
            #,colourPalette=b.col
            , addLegend = TRUE)
```

```
149 codes from your data successfully matched countries in the map
1 codes from your data failed to match with a country code in the map
94 codes from the map weren't represented in your data
```

```
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(14, 102), cex.axis = 1, :
" ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません "
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(14, 102), cex.axis = 1, :
" ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません "
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(14, 102), cex.axis = 1, :
" ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません "
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(14, 102), cex.axis = 1, :
" ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません "
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(14, 102), cex.axis = 1, :
" ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません "
Warning message in axis(side = 1, mgp = c(3, 1, 0), las = 0, at = c(14, 102), cex.axis = 1, :
" ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません "
Warning message in title(mapTitle):
" ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません "
Warning message in title(mapTitle):
" ポストスクリプトフォントのデータベースにフォントファミリ 'Osaka' が見付かりません "
```


中国でコロナ感染者が出てから各国でコロナ感染者が初めて出るまでの日数
14
102

In [ ]:

```

```