Feedback on Stillbirths Maps

Ernest Guevarra

16/08/2018

File naming

Always make sure that you don't use filenames with spaces. Too much use of Microsoft Word has trained us that the filename of a document can be as long and can have spaces. Unfortunately, it is very hard to work with filenames with spaces especially when working with code. Remember what we discussed in the workshop about file naming conventions. either use an underscore if you really want to have a space in the filename or use the camel case naming syntax (see https://en.wikipedia.org/wiki/Camel_case)

I woudl suggest renaming the `Stillbirths maps.R` file differently. With camel case syntax, you can opt for `stillbirthsMaps.R`. With underscore, it can be `stillbirths_maps.R`.

Line by line feedback

Line 4

In line 4 of the script, you wrote:

```
write.csv(provincedata, "districtdata.csv", row.names = FALSE)
```

I think here you meant to write the object named `districtdata` into a CSV file called `districtdata.csv`. So the script should read:

```
write.csv(district, "districtdata.csv", row.names = FALSE)
```

Lines 25, 26, 31

```
#
# stillbirths
#
still2015p <- pdata2015$still
still2016p <- pdata2016$still

#
# create variable females in province
#
femprov <- pop_adm1$FEMALES
```

You don't always have to create variables/objects for single values that you want to use from a data frame. If you do so, you will end up with so many objects in your workspace that you will only probably use once. You can simply make a call to these variables by indexing (using square brackets) or using the `$` syntax whenever you need them.

Lines 31, 35-36

```
#
# create variable females in province
#
femprov <- pop_adm1$FEMALES

# use femprov to normalise maternal deaths per province

still2015p <- (pdata2015$still/femprov)*100000
still2016p <- (pdata2016$still/femprov)*100000
```

Here, I think you are trying to normalise the stillbirths count using the number of female population per province. Specifically, you are normalising it to a rate of number of stillbirths per 100,000 females. Is that correct?

First comment here is that it might be better to use the number of women of reproductive age as your normalising value. This value is available in `pop_adm1` in the column named `WRA`.

Second, you will notice that the population data in `pop_adm1` has a different order of provinces as compared to `pdata2015`.

ADM1_EN	ADM1_PCODE	ADM0_EN	ADM0_PCODE	WARDS	UNITS	HOUSEHOLDS	PERSONS	MALES	FEMALES	WRA
Western Province	PG01	Papua New Guinea	PG	307	872	36800	234745	120970	113775	59618
Gulf Province	PG02	Papua New Guinea	PG	144	531	25819	158197	81814	76383	40025
Central Province	PG03	Papua New Guinea	PG	209	1109	41309	269756	142231	127525	66823
National Capital District	PG04	Papua New Guinea	PG	9	520	47559	364125	194834	169291	88708
Milne Bay Province	PG05	Papua New Guinea	PG	391	1169	55262	276512	143791	132721	69546
Northern (Oro) Province	PG06	Papua New Guinea	PG	158	948	34117	186309	97188	89121	46699
Southern Highlands Province	PG07	Papua New Guinea	PG	465	2628	88041	510245	263523	246722	129282
Hela Province	PG21	Papua New Guinea	PG	260	1616	65271	249449	128895	120554	63170
Enga Province	PG08	Papua New Guinea	PG	365	1015	76421	432045	224663	207382	108668
Jiwaka Province	PG22	Papua New Guinea	PG	190	1191	49298	243886	125458	118628	62161
Western Highlands Province	PG09	Papua New Guinea	PG	298	1145	85101	362850	183791	179059	93827
Chimbu (Simbu) Province	PG10	Papua New Guinea	PG	315	1494	81668	407567	214219	193348	101314
Eastern Highlands Province	PG11	Papua New Guinea	PG	259	3393	136992	579825	301048	278777	146079
Morobe Province	PG12	Papua New Guinea	PG	521	2429	130109	674810	350902	323908	169728
Madang Province	PG13	Papua New Guinea	PG	442	1708	86140	493906	257581	236325	123834
East Sepik Province	PG14	Papua New Guinea	PG	625	1481	87465	450343	225933	224468	117621
West Sepik (Sandaun) Province	PG15	Papua New Guinea	PG	336	926	44934	248411	127771	120640	63215
Manus Province	PG16	Papua New Guinea	PG	122	224	10360	60485	31161	29324	15366
New Ireland Province	PG17	Papua New Guinea	PG	138	631	29634	194067	102494	91573	47984
East New Britain Province	PG18	Papua New Guinea	PG	370	790	58458	328369	168760	159609	83635
West New Britain Province	PG19	Papua New Guinea	PG	107	734	50744	264264	138942	125322	65669
Autonomous Region of Bougainville	PG20	Papua New Guinea	PG	81	1069	48233	249358	127990	121368	63597

pcode	year	bfpills1	combpills1	inj1	uno1	vasectomy	iud1	ovulation1	condom1	bfpills2	combpills2	inj2	iud2	ovulation2
1	2015	844	544	2187	433	3	196	80	390	2995	4255	15682	82	128
2	2015	315	177	721	3	0	46	5	1216	475	795	4154	3	6
3	2015	141	171	855	72	10	93	9	110	218	568	4438	10	56
4	2015	411	417	1863	31	0	77	133	50	466	4104	18634	19	174
5	2015	547	506	2021	93	17	53	200	485	972	3490	9394	4	1529
6	2015	312	300	1061	118	0	27	123	543	534	1316	6158	43	70
7	2015	1316	1115	2818	43	7	29	133	1438	4004	6593	11454	64	231
8	2015	458	728	1233	938	644	104	263	898	622	2689	4753	103	213
9	2015	330	606	2797	69	2	85	263	642	291	4507	16185	40	92
10	2015	829	927	2398	190	166	428	131	3027	855	1845	6932	51	130
11	2015	1782	1547	3284	145	427	1310	51	1608	3058	10347	14809	350	120
12	2015	1397	1365	3329	87	84	284	112	3195	2228	10355	21457	7	175
13	2015	1035	1312	4584	190	48	123	61	825	1505	3066	10910	148	120
14	2015	495	475	1226	16	4	5	81	623	808	4061	6901	0	93
15	2015	272	383	1245	6	16	293	36	41	412	1625	6966	28	12
16	2015	27	20	280	15	7	39	36	87	22	69	1526	38	26
17	2015	9	35	415	32	0	88	88	161	5	75	1902	64	154
18	2015	199	186	1649	80	52	170	1556	524	123	700	4076	3	1084
19	2015	518	432	1129	448	40	686	80	638	854	2450	4924	191	77
20	2015	125	168	383	49	2	6	5	780	214	381	1910	4	313
21	2015	1127	1058	1577	0	0	1	1069	16645	2494	3148	4478	0	213
22	2015	717	615	1375	14	13	10	65	3099	867	2106	6588	17	37

The province data goes province with code 1 to code 22 while the population data is a little bit mixed up in that the `ADM1_PCODE` values go from `PG01` to `PG07` and then `PG21`, `PG08`, `PG22` and then `PG09` to `PG20`. This means that simply doing a straight division as you did in lines 35 and 36 would mean dividing some provinces with code 8 onwards with the wrong normalising population value.

To address this, one solution is to sort `pop_adm1` data frame in such a way that it goes from 1-22 province code order. This can be done as follows:

```
pop_adm1 <- pop_adm1[order(pop_adm1$ADM1_PCODE), ]
```

What this command does is that it orders the variable `ADM1_PCODE` from `PG01` to `PG22` and then sorts `pop_adm1` data frame based on this new ordering.

The resulting data frame is:

ADM1_EN	ADM1_PCODE	ADM0_EN	ADM0_PCODE	WARDS	UNITS	HOUSEHOLDS	PERSONS	MALES	FEMALES	WRA
Western Province	PG01	Papua New Guinea	PG	307	872	36800	234745	120970	113775	59618
Gulf Province	PG02	Papua New Guinea	PG	144	531	25819	158197	81814	76383	40025
Central Province	PG03	Papua New Guinea	PG	209	1109	41309	269756	142231	127525	66823
National Capital District	PG04	Papua New Guinea	PG	9	520	47559	364125	194834	169291	88708
Milne Bay Province	PG05	Papua New Guinea	PG	391	1169	55262	276512	143791	132721	69546
Northern (Oro) Province	PG06	Papua New Guinea	PG	158	948	34117	186309	97188	89121	46699
Southern Highlands Province	PG07	Papua New Guinea	PG	465	2628	88041	510245	263523	246722	129282
Enga Province	PG08	Papua New Guinea	PG	365	1015	76421	432045	224663	207382	108668
Western Highlands Province	PG09	Papua New Guinea	PG	298	1145	85101	362850	183791	179059	93827
Chimbu (Simbu) Province	PG10	Papua New Guinea	PG	315	1494	81668	407567	214219	193348	101314
Eastern Highlands Province	PG11	Papua New Guinea	PG	259	3393	136992	579825	301048	278777	146079
Morobe Province	PG12	Papua New Guinea	PG	521	2429	130109	674810	350902	323908	169728
Madang Province	PG13	Papua New Guinea	PG	442	1708	86140	493906	257581	236325	123834
East Sepik Province	PG14	Papua New Guinea	PG	625	1481	87465	450343	225933	224468	117621
West Sepik (Sandaun) Province	PG15	Papua New Guinea	PG	336	926	44934	248411	127771	120640	63215
Manus Province	PG16	Papua New Guinea	PG	122	224	10360	60485	31161	29324	15366
New Ireland Province	PG17	Papua New Guinea	PG	138	631	29634	194067	102494	91573	47984
East New Britain Province	PG18	Papua New Guinea	PG	370	790	58458	328369	168760	159609	83635
West New Britain Province	PG19	Papua New Guinea	PG	107	734	50744	264264	138942	125322	65669
Autonomous Region of Bougainville	PG20	Papua New Guinea	PG	81	1069	48233	249358	127990	121368	63597
Hela Province	PG21	Papua New Guinea	PG	260	1616	65271	249449	128895	120554	63170
Jiwaka Province	PG22	Papua New Guinea	PG	190	1191	49298	243886	125458	118628	62161

You will now notice that the order of the provinces is from `PG01` to `PG22`, the same order that we have for `pdata2015` and `pdata2016`. Once it has been ordered this way, The stillbirths data can now be normalised using the women of reproductive age populations using the following commands.

```
still2015p <- (pdata2015$still/pop_adm1$WRA) * 100000
still2016p <- (pdata2016$still/pop_adm1$WRA) * 100000
```

Lines 38 - 39, 41 - 42

```
pop2015 <- data.frame("pcode" = 1:22, "pop2015" = pdata2015$still)
pop2016 <- data.frame("pcode" = 1:22, "pop2016" = pdata2016$still)

pdata2015 <- merge(pdata2015, pop2015, by = "pcode")
pdata2016 <- merge(pdata2016, pop2016, by = "pcode")
```

In these commands, what I think you are trying to do is to create a data frame for the stillbirths data.

The main issue with this is that you are still using the non-normalised stillbirth data instead of the normalised stillbirth data that you have just calculated. Also, it might be more informative to call this new data frame as `still2015` and `still2016` rather than `pop2015` and `pop2016` So, the more appropriate command would be:

```
still2015 <- data.frame("pcode" = 1:22, "still2015" = still2015p)
still2016 <- data.frame("pcode" = 1:22, "still2016" = still2016p)
```

And then, these can now be merged to the province data as follows:

```
pdata2015 <- merge(pdata2015, still2015, by = "pcode")
pdata2016 <- merge(pdata2016, still2016, by = "pcode")
```

Lines 44-73

```
# map stillbirths per province normalised by total women per province x 100000
#
# first create classification of stillbirths for each year
#
still2015pclass <- base::cut(x = still2015p,
                             breaks = c(0, 10, 20, 40, 60, 80, 100),
                             labels = FALSE)

still2016pclass <- base::cut(x=still2016p,
                             breaks = c(0, 10, 20, 40, 60, 80, 100),
                             labels = FALSE)

# create colourscheme and provide color for classification per province
# of mat deaths
colourscheme <- c("#eff3ff", "#c6dbef", "#9ecae1",
                  "#6baed6", "#3182bd", "#08519c")

plot (province, lwd = 1, border = "gray50",
      col = ifelse(still2015pclass == 0, colourscheme[1],
              ifelse(still2015pclass == 1, colourscheme[2],
```

```
                ifelse(still2015pclass == 2, colourscheme[3],
                  ifelse(still2015pclass == 3, colourscheme[4],
                    ifelse(still2015pclass == 4, colourscheme[5],
                      ifelse(still2015pclass == 5, colourscheme[6],
                             colourscheme[7])))))))

plot (province, lwd = 1, border = "gray50",
      col = ifelse(still2016pclass == 0, colourscheme[1],
              ifelse(still2016pclass == 1, colourscheme[2],
                ifelse(still2016pclass == 2, colourscheme[3],
                  ifelse(still2016pclass == 3, colourscheme[4],
                    ifelse(still2016pclass == 4, colourscheme[5],
                      ifelse(still2016pclass == 5, colourscheme[6],
                             colourscheme[7])))))))
```

There are a lot of issues with these lines of code.

First, in grouping/classifying the normalised stillbirths, you will need to see the range of values for the normalised stillbirths. In the way you have classified them, you assume that the maximum value is 100. However, the normalised values for stillbirhts goes beyond 100. To see the range of normalised stillbirth values, you can create a simple summary:

```
summary(still2015p)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   68.10   97.68  102.43  132.06  196.99
```

```
summary(still2016p)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   33.82   57.22   98.78  111.17  135.32  260.40
```

To address this, we will have to use an approach in which the normalised stillbirth counts are grouped into meaningful classes. A useful approach will be using quantiles. For this, we can use the R package called `classInt` which has a function called `classIntervals()`. To install `classInt`, run the following commands:

```
install.packages("classInt")
```

Once installed, you can now use the `classIntervals()` function to create 5 classes from the normalised stillbirth values:

```
#
# 2015 data
#
still2015classes <- cut(x = still2015p,
                        breaks = classIntervals(var = still2015p,
                                                n = 5,
                                                style = "quantile")$brks,
                        labels = FALSE)
#
# Recode NA class to 0
#
still2015classes <- ifelse(is.na(still2015classes), 0, still2015classes)
```

```
#
# 2016 data
#
still2016classes <- cut(x = still2016p,
                        breaks = classIntervals(var = still2016p,
                                                n = 5,
                                                style = "quantile")$brks,
                        labels = FALSE)
#
# Recode NA class to 0
#
still2016classes <- ifelse(is.na(still2016classes), 0, still2016classes)
```

Now for mapping the stillbirths, we first need to relate the stillbirths data with the map. This means making sure that the map data is also ordered in the same way as the stillbirths data. We know that the stillbirthsdata is ordered from the province with code 1 to province with code 22. Now we need to inspect the map data. We can do this as follows:

```
province@data
```

```
##                               ADM1_EN ADM1_PCODE          ADM0_EN
## 0  Autonomous Region of Bougainville         20 Papua New Guinea
## 1                    Central Province         03 Papua New Guinea
## 2            Chimbu (Simbu) Province         10 Papua New Guinea
## 3          East New Britain Province         18 Papua New Guinea
## 4                East Sepik Province         14 Papua New Guinea
## 5         Eastern Highlands Province         11 Papua New Guinea
## 6                       Enga Province         08 Papua New Guinea
## 7                       Gulf Province         02 Papua New Guinea
## 8                       Hela Province         21 Papua New Guinea
## 9                     Jiwaka Province         22 Papua New Guinea
## 10                    Madang Province         13 Papua New Guinea
## 11                     Manus Province         16 Papua New Guinea
## 12                 Milne Bay Province         05 Papua New Guinea
## 13                    Morobe Province         12 Papua New Guinea
## 14         National Capital District         04 Papua New Guinea
## 15              New Ireland Province         17 Papua New Guinea
## 16            Northern (Oro) Province         06 Papua New Guinea
## 17       Southern Highlands Province         07 Papua New Guinea
## 18         West New Britain Province         19 Papua New Guinea
## 19     West Sepik (Sandaun) Province         15 Papua New Guinea
## 20        Western Highlands Province         09 Papua New Guinea
## 21                    Western Province         01 Papua New Guinea
##    ADM0_PCODE
## 0          PG
## 1          PG
## 2          PG
## 3          PG
## 4          PG
## 5          PG
## 6          PG
```

```
## 7          PG
## 8          PG
## 9          PG
## 10         PG
## 11         PG
## 12         PG
## 13         PG
## 14         PG
## 15         PG
## 16         PG
## 17         PG
## 18         PG
## 19         PG
## 20         PG
## 21         PG
```

Here we can see that the province map data is not ordered from code 1 to 22. We will therefore have to order it in a similar way that we did the ordering for `pop_adm1`. This can be done as follows:

```
province@data <- province@data[order(province@data$ADM1_PCODE), ]
```

This orders the provinces in the map data based on their code from 1 to 22. This data is now in the same order as our stillbirths data as shown below:

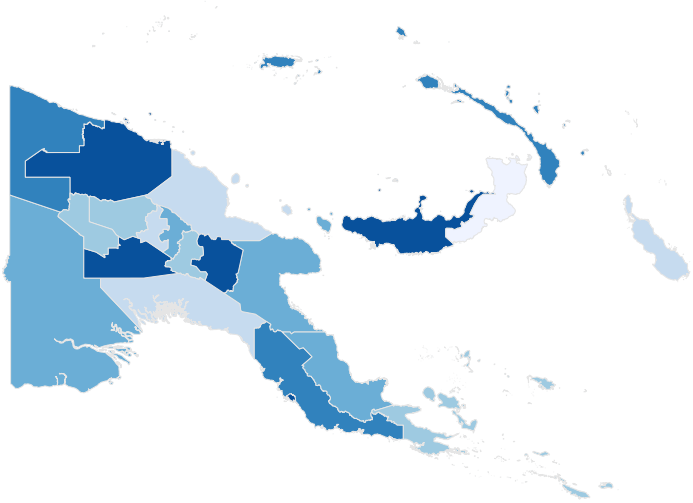
```
##                              ADM1_EN ADM1_PCODE          ADM0_EN
## 21                 Western Province         01 Papua New Guinea
## 7                     Gulf Province         02 Papua New Guinea
## 1                  Central Province         03 Papua New Guinea
## 14         National Capital District         04 Papua New Guinea
## 12                Milne Bay Province         05 Papua New Guinea
## 16           Northern (Oro) Province         06 Papua New Guinea
## 17       Southern Highlands Province         07 Papua New Guinea
## 6                     Enga Province         08 Papua New Guinea
## 20        Western Highlands Province         09 Papua New Guinea
## 2            Chimbu (Simbu) Province         10 Papua New Guinea
## 5         Eastern Highlands Province         11 Papua New Guinea
## 13                  Morobe Province         12 Papua New Guinea
## 10                  Madang Province         13 Papua New Guinea
## 4               East Sepik Province         14 Papua New Guinea
## 19     West Sepik (Sandaun) Province         15 Papua New Guinea
## 11                   Manus Province         16 Papua New Guinea
## 15              New Ireland Province         17 Papua New Guinea
## 3          East New Britain Province         18 Papua New Guinea
## 18         West New Britain Province         19 Papua New Guinea
## 0  Autonomous Region of Bougainville         20 Papua New Guinea
## 8                     Hela Province         21 Papua New Guinea
## 9                   Jiwaka Province         22 Papua New Guinea
##    ADM0_PCODE
## 21         PG
## 7          PG
## 1          PG
```

```
## 14          PG
## 12          PG
## 16          PG
## 17          PG
## 6           PG
## 20          PG
## 2           PG
## 5           PG
## 13          PG
## 10          PG
## 4           PG
## 19          PG
## 11          PG
## 15          PG
## 3           PG
## 18          PG
## 0           PG
## 8           PG
## 9           PG
```

Now we can map the stillbirths for 2015 data as follows:

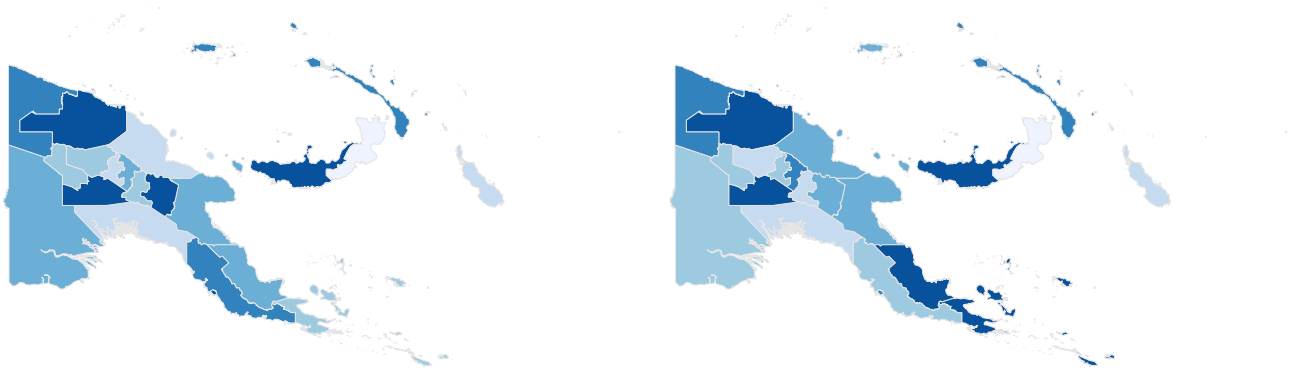
```
colourscheme <- c("#eff3ff", "#c6dbef", "#9ecae1",
                  "#6baed6", "#3182bd", "#08519c")

plot(province,
     col = colourscheme[still2015classes + 1],
     border = "gray90",
     lwd = 0.5)
```



To map both 2015 and 2016, we can plot these maps side-by-side in order to make comparisons. This can be done as follows:

```
par(mar = c(0, 0, 0, 0), mfrow = c(1, 2))
plot(province,
     col = colourscheme[still2015classes + 1],
     border = "gray90",
     lwd = 0.5)
plot(province,
     col = colourscheme[still2016classes + 1],
     border = "gray90",
     lwd = 0.5)
```



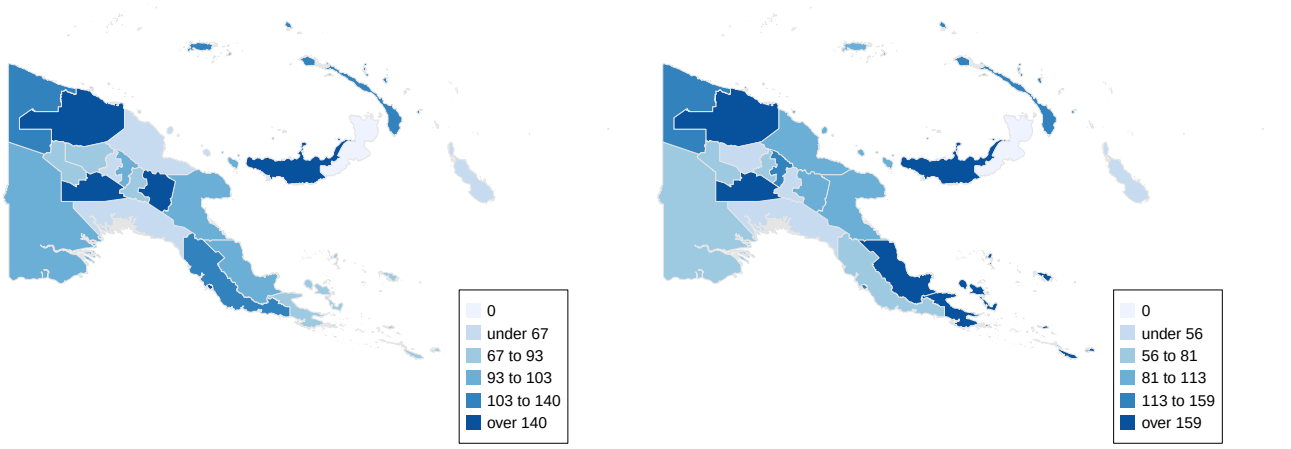
Now, it will be useful to add a title to each plot to identify which map is for which year and to add a legend to show what the colours refer to. This can be done as follows:

```
par(mar = c(0, 0, 0, 0), mfrow = c(1, 2))
plot(province,
     col = colourscheme[still2015classes + 1],
     border = "gray90",
     lwd = 0.5)
title(main = "Stillbirths 2015", line = -1, adj = 1)
legend(x = "bottomright",
       inset = 0.1,
       y.intersp = 1.2,
       legend = c("0", names(print(classIntervals(still2015p,
                                                  n = 5,
                                                  style = "quantile",
                                                  dataPrecision = 0),
                                   between = "to",
                                   cutlabels = FALSE))),
       pch = 15, pt.cex = 2,
       col = colourscheme)
```

```
## style: quantile
##   one of 5,985 possible partitions of this variable into 5 classes
##   under 67   67 to 93  93 to 103 103 to 140   over 140
##          5          4          4          4          5
```

```
plot(province,
     col = colourscheme[still2016classes + 1],
```

```
     border = "gray90",
     lwd = 0.5)
title(main = "Stillbirths 2016", line = -1, adj = 1)
legend(x = "bottomright",
       inset = 0.1,
       y.intersp = 1.2,
       legend = c("0", names(print(classIntervals(still2016p,
                                                  n = 5,
                                                  style = "quantile",
                                                  dataPrecision = 0),
                                   between = "to",
                                   cutlabels = FALSE))),
       pch = 15, pt.cex = 2,
       col = colourscheme)
```




```
## style: quantile
##   one of 5,985 possible partitions of this variable into 5 classes
##   under 56   56 to 81  81 to 113 113 to 159   over 159
##          5          4          4          4          5
```

Now, you might want to use the same classes for each map so that they can be compared with each other. To do this, we will need to create the classes using the normalised stillbirth values for 2015 and 2016. This can be done as follows:

```
#
# 2015 data
#
still2015classes <- cut(x = still2015p,
                        breaks = classIntervals(var = c(still2015p, still2016p),
                                                n = 5,
                                                style = "quantile")$brks,
                        labels = FALSE)
#
# Recode NA class to 0
#
```

```
still2015classes <- ifelse(is.na(still2015classes), 0, still2015classes)
#
# 2016 data
#
still2016classes <- cut(x = still2016p,
                        breaks = classIntervals(var = c(still2015p, still2016p),
                                                n = 5,
                                                style = "quantile")$brks,
                        labels = FALSE)
#
# Recode NA class to 0
#
still2016classes <- ifelse(is.na(still2016classes), 0, still2016classes)
```

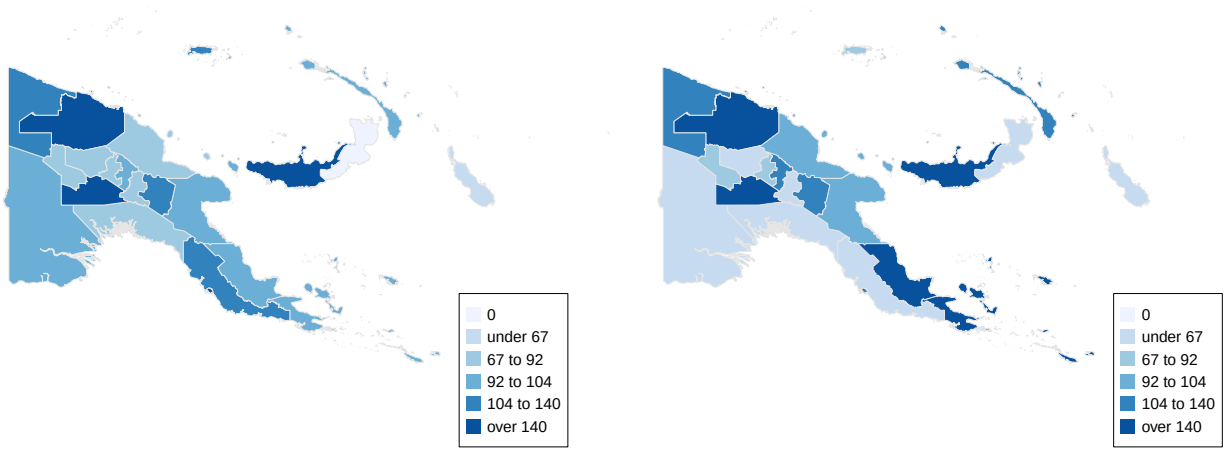
Using these new classes, the map can be mapped again with using these new classifications.

```
par(mar = c(0, 0, 0, 0), mfrow = c(1, 2))
plot(province,
     col = colourscheme[still2015classes + 1],
     border = "gray90",
     lwd = 0.5)
title(main = "Stillbirths 2015", line = -1, adj = 1)
legend(x = "bottomright",
       inset = 0.1,
       y.intersp = 1.2,
       legend = c("0", names(print(classIntervals(c(still2015p, still2015p),
                                                  n = 5,
                                                  style = "quantile",
                                                  dataPrecision = 0),
                                   between = "to",
                                   cutlabels = FALSE))),
       pch = 15, pt.cex = 2,
       col = colourscheme)
```

```
## style: quantile
##   one of 5,985 possible partitions of this variable into 5 classes
##   under 67   67 to 92  92 to 104 104 to 140   over 140
##          8         10          8          8         10
```

```
plot(province,
     col = colourscheme[still2016classes + 1],
     border = "gray90",
     lwd = 0.5)
title(main = "Stillbirths 2016", line = -1, adj = 1)
legend(x = "bottomright",
       inset = 0.1,
       y.intersp = 1.2,
       legend = c("0", names(print(classIntervals(c(still2015p, still2015p),
                                                  n = 5,
                                                  style = "quantile",
                                                  dataPrecision = 0),
```

```
                                    between = "to",
                                    cutlabels = FALSE))),
       pch = 15, pt.cex = 2,
       col = colourscheme)
```




```
## style: quantile
##   one of 5,985 possible partitions of this variable into 5 classes
##   under 67   67 to 92  92 to 104 104 to 140   over 140
##          8         10          8          8         10
```