

Korea Map WGS84

coop711

2017 9 20

Map Data

```
library(maptools) #> `readShapePoly()`, `rgdal::readOGR()`로 대체
```

```
## Loading required package: sp
```

```
## Checking rgeos availability: TRUE
```

```
library(ggmap) #> `geocode()`, `ggmap()`, `qmap()`, `revgeocode()``
```

```
## Loading required package: ggplot2
```

```
library(ggplot2) #> `ggplot()``  
library(rgdal) #> `CRS`, `ogrInfo()`, `ogrListLayers()`, `readOGR()`, `spTransform()``
```

```
## rgdal: version: 1.2-12, (SVN revision 681)  
## Geospatial Data Abstraction Library extensions to R successfully loaded  
## Loaded GDAL runtime: GDAL 2.1.3, released 2017/20/01  
## Path to GDAL shared files: /Library/Frameworks/R.framework/Versions/3.4/Resources/library/rgdal/gdal  
## Loaded PROJ.4 runtime: Rel. 4.9.3, 15 August 2016, [PJ_VERSION: 493]  
## Path to PROJ.4 shared files: /Library/Frameworks/R.framework/Versions/3.4/Resources/library/rgdal/proj  
## Linking to sp version: 1.2-5
```

```
library(dplyr) #> `arrange()`, `filter()`, `group_by()`, `left_join()`, `mutate()`,
```

```
##  
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':  
##  
## filter, lag
```

```
## The following objects are masked from 'package:base':  
##  
## intersect, setdiff, setequal, union
```

```
library(broom) #> `tidy()``  
library(extrafont) #> "HCR Dotum LVT"
```

```
## Registering fonts with R
```

```
# getwd()  
#> Data source Name  
dsn <- "../data/CTPRVN_201703"  
#> 시도 단위 지도 파일 읽어들이기. OGR : OpenGis simple features Reference implementation  
#> `stringsAsFactors = FALSE`를 설정하지 않으면 시도 이름이 `factor`로 자동 변환되면서 가나다 순으로  
수준(`level`)이 매겨짐에 따라 `강원도, 경기도, ...`, `순이 됨.  
#> `encoding = "CP949"`는 한글 인코딩 작업이 윈도우 시스템에서 이루어졌기 때문에 취해진 조치임. 윈도우에  
서 읽어들이 때는 ``"``로 변경하거나 설정하지 않아도 됨.  
shp1 <- readOGR(dsn,  
                layer = "TL_SCCO_CTPRVN",  
                stringsAsFactors = FALSE,  
                encoding = "CP949")
```

```
## OGR data source with driver: ESRI Shapefile  
## Source: "../data/CTPRVN_201703", layer: "TL_SCCO_CTPRVN"  
## with 17 features  
## It has 3 fields
```

```
#> 아래와 같이 `dsn`과 `layer`를 통합하여 읽어들이일 수도 있으나 이때는 `.shp`를 붙여주어야 함.  
# shp1 <- readOGR("../data/CTPRVN_201703/TL_SCCO_CTPRVN.shp",  
#                 stringsAsFactors = FALSE,  
#                 encoding = "CP949")  
# map1 <- fortify(shp1)  
dsn2 <- "../data/SIG_201703"  
shp2 <- readOGR(dsn2,  
                layer = "TL_SCCO_SIG",  
                stringsAsFactors = FALSE,  
                encoding = "CP949")
```

```
## OGR data source with driver: ESRI Shapefile  
## Source: "../data/SIG_201703", layer: "TL_SCCO_SIG"  
## with 250 features  
## It has 3 fields
```

```
# map2 <- fortify(shp2)  
shp2@data
```

##	SIG_CD	SIG_ENG_NM	SIG_KOR_NM
## 0	11110	Jongno-gu	종로구
## 1	11140	Jung-gu	중구
## 2	11170	Yongsan-gu	용산구
## 3	11200	Seongdong-gu	성동구
## 4	11215	Gwangjin-gu	광진구
## 5	11230	Dongdaemun-gu	동대문구
## 6	11260	Jungnang-gu	종량구
## 7	11290	Seongbuk-gu	성북구
## 8	11305	Gangbuk-gu	강북구
## 9	11320	Dobong-gu	도봉구
## 10	11350	Nowon-gu	노원구
## 11	11380	Eunpyeong-gu	은평구
## 12	11410	Seodaemun-gu	서대문구
## 13	11440	Mapo-gu	마포구
## 14	11470	Yangcheon-gu	양천구
## 15	11500	Gangseo-gu	강서구
## 16	11530	Guro-gu	구로구
## 17	11545	Geumcheon-gu	금천구
## 18	11560	Yeongdeungpo-gu	영등포구
## 19	11590	Dongjak-gu	동작구
## 20	11620	Gwanak-gu	관악구
## 21	11650	Seocho-gu	서초구
## 22	11680	Gangnam-gu	강남구
## 23	11710	Songpa-gu	송파구
## 24	11740	Gangdong-gu	강동구
## 25	26110	Jung-gu	중구
## 26	26140	Seo-gu	서구
## 27	26170	Dong-gu	동구
## 28	26200	Yeongdo-gu	영도구
## 29	26230	Busanjin-gu	부산진구
## 30	26260	Dongnae-gu	동래구
## 31	26290	Nam-gu	남구
## 32	26320	Buk-gu	북구
## 33	26350	Haeundae-gu	해운대구
## 34	26380	Saha-gu	사하구
## 35	26410	Geumjeong-gu	금정구
## 36	26440	Gangseo-gu	강서구
## 37	26470	Yeonje-gu	연제구
## 38	26500	Suyeong-gu	수영구
## 39	26530	Sasang-gu	사상구
## 40	26710	Gijang-gun	기장군
## 41	27110	Jung-gu	중구
## 42	27140	Dong-gu	동구
## 43	27170	Seo-gu	서구
## 44	27200	Nam-gu	남구
## 45	27230	Buk-gu	북구
## 46	27260	Suseong-gu	수성구
## 47	27290	Dalseo-gu	달서구
## 48	27710	Dalseong-gun	달성군
## 49	28110	Jung-gu	중구
## 50	28140	Dong-gu	동구
## 51	28170	Nam-gu	남구
## 52	28185	Yeonsu-gu	연수구
## 53	28200	Namdong-gu	남동구
## 54	28237	Bupyeong-gu	부평구
## 55	28245	Gyeyang-gu	계양구

## 56	28260	Seo-gu	서구
## 57	28710	Ganghwa-gun	강화군
## 58	28720	Ongjin-gun	웅진군
## 59	29110	Dong-gu	동구
## 60	29140	Seo-gu	서구
## 61	29155	Nam-gu	남구
## 62	29170	Buk-gu	북구
## 63	29200	Gwangsan-gu	광산구
## 64	30110	Dong-gu	동구
## 65	30140	Jung-gu	중구
## 66	30170	Seo-gu	서구
## 67	30200	Yuseong-gu	유성구
## 68	30230	Daedeok-gu	대덕구
## 69	31110	Jung-gu	중구
## 70	31140	Nam-gu	남구
## 71	31170	Dong-gu	동구
## 72	31200	Buk-gu	북구
## 73	31710	Ulsju-gun	울주군
## 74	36110	Sejong-si	세종특별자치시
## 75	41111	Jangan-gu, Suwon-si	수원시 장안구
## 76	41113	Gwonseon-gu, Suwon-si	수원시 권선구
## 77	41115	Paldal-gu, Suwon-si	수원시 팔달구
## 78	41117	Yeongtong-gu, Suwon-si	수원시 영통구
## 79	41131	Sujeong-gu, Seongnam-si	성남시 수정구
## 80	41133	Jungwon-gu, Seongnam-si	성남시 중원구
## 81	41135	Bundang-gu, Seongnam-si	성남시 분당구
## 82	41150	Uijeongbu-si	의정부시
## 83	41171	Manan-gu, Anyang-si	안양시만안구
## 84	41173	Dongan-gu, Anyang-si	안양시동안구
## 85	41190	Bucheon-si	부천시
## 86	41210	Gwangmyeong-si	광명시
## 87	41220	Pyeongtaek-si	평택시
## 88	41250	Dongducheon-si	동두천시
## 89	41271	Sangnok-gu, Ansan-si	안산시상록구
## 90	41273	Danwon-gu, Ansan-si	안산시단원구
## 91	41281	Deogyang-gu, Goyang-si	고양시덕양구
## 92	41285	Ilsandong-gu, Goyang-si	고양시일산동구
## 93	41287	Ilsanseo-gu, Goyang-si	고양시일산서구
## 94	41290	Gwacheon-si	과천시
## 95	41310	Guri-si	구리시
## 96	41360	Namyangju-si	남양주시
## 97	41370	Osan-si	오산시
## 98	41390	Siheung-si	시흥시
## 99	41410	Gunpo-si	군포시
## 100	41430	Uiwang-si	의왕시
## 101	41450	Hanam-si	하남시
## 102	41461	Cheoin-gu, Yongin-si	용인시처인구
## 103	41463	Giheung-gu, Yongin-si	용인시기흥구
## 104	41465	Suji-gu, Yongin-si	용인시수지구
## 105	41480	Paju-si	파주시
## 106	41500	Icheon-si	이천시
## 107	41550	Anseong-si	안성시
## 108	41570	Gimpo-si	김포시
## 109	41590	Hwaseong-si	화성시
## 110	41610	Gwangju-si	광주시
## 111	41630	Yangju-si	양주시
## 112	41650	Pocheon-si	포천시
## 113	41670	Yeoju-si	여주시

## 114	41800	Yeoncheon-gun	연천군
## 115	41820	Gapyeong-gun	가평군
## 116	41830	Yangpyeong-gun	양평군
## 117	42110	Chuncheon-si	춘천시
## 118	42130	Wonju-si	원주시
## 119	42150	Gangneung-si	강릉시
## 120	42170	Donghae-si	동해시
## 121	42190	Taebaek-si	태백시
## 122	42210	Sokcho-si	속초시
## 123	42230	Samcheok-si	삼척시
## 124	42720	Hongcheon-gun	홍천군
## 125	42730	Hoengseong-gun	횡성군
## 126	42750	Yeongwol-gun	영월군
## 127	42760	Pyeongchang-gun	평창군
## 128	42770	Jeongseon-gun	정선군
## 129	42780	Cheorwon-gun	철원군
## 130	42790	Hwacheon-gun	화천군
## 131	42800	Yanggu-gun	양구군
## 132	42810	Inje-gun	인제군
## 133	42820	Goseong-gun	고성군
## 134	42830	Yangyang-gun	양양군
## 135	43111	Sangdang-gu, Cheongju-si	청주시 상당구
## 136	43112	Seowon-gu, Cheongju-si	청주시 서원구
## 137	43113	Heungdeok-gu, Cheongju-si	청주시 흥덕구
## 138	43114	Cheongwon-gu, Cheongju-si	청주시 청원구
## 139	43130	Chungju-si	충주시
## 140	43150	Jecheon-si	제천시
## 141	43720	Boeun-gun	보은군
## 142	43730	Okcheon-gun	옥천군
## 143	43740	Yeongdong-gun	영동군
## 144	43745	Jeungpyeong-gun	증평군
## 145	43750	Jincheon-gun	진천군
## 146	43760	Goesan-gun	괴산군
## 147	43770	Eumseong-gun	음성군
## 148	43800	Danyang-gun	단양군
## 149	44131	Dongnam-gu, Cheonan-si	천안시동남구
## 150	44133	Seobuk-gu, Cheonan-si	천안시서북구
## 151	44150	Gongju-si	공주시
## 152	44180	Boryeong-si	보령시
## 153	44200	Asan-si	아산시
## 154	44210	Seosan-si	서산시
## 155	44230	Nonsan-si	논산시
## 156	44250	Gyeryong-si	계룡시
## 157	44270	Dangjin-si	당진시
## 158	44710	Geumsan-gun	금산군
## 159	44760	Buyeo-gun	부여군
## 160	44770	Seocheon-gun	서천군
## 161	44790	Cheongyang-gun	청양군
## 162	44800	Hongseong-gun	홍성군
## 163	44810	Yesan-gun	예산군
## 164	44825	Tae'an-gun	태안군
## 165	45111	Wansan-gu, Jeonju-si	전주시완산구
## 166	45113	Deokjin-gu, Jeonju-si	전주시덕진구
## 167	45130	Gunsan-si	군산시
## 168	45140	Iksan-si	익산시
## 169	45180	Jeongeup-si	정읍시
## 170	45190	Namwon-si	남원시
## 171	45210	Gimje-si	김제시

## 172	45710	Wanju-gun	완주군
## 173	45720	Jinan-gun	진안군
## 174	45730	Muju-gun	무주군
## 175	45740	Jangsu-gun	장수군
## 176	45750	Imsil-gun	임실군
## 177	45770	Sunchang-gun	순창군
## 178	45790	Gochang-gun	고창군
## 179	45800	Buan-gun	부안군
## 180	46110	Mokpo-si	목포시
## 181	46130	Yeosu-si	여주시
## 182	46150	Suncheon-si	순천시
## 183	46170	Naju-si	나주시
## 184	46230	Gwangyang-si	광양시
## 185	46710	Damyang-gun	담양군
## 186	46720	Gokseong-gun	곡성군
## 187	46730	Gurye-gun	구례군
## 188	46770	Goheung-gun	고흥군
## 189	46780	Boseong-gun	보성군
## 190	46790	Hwasun-gun	화순군
## 191	46800	Jangheung-gun	장흥군
## 192	46810	Gangjin-gun	강진군
## 193	46820	Haenam-gun	해남군
## 194	46830	Yeongam-gun	영암군
## 195	46840	Muan-gun	무안군
## 196	46860	Hampyeong-gun	함평군
## 197	46870	Yeonggwang-gun	영광군
## 198	46880	Jangseong-gun	장성군
## 199	46890	Wando-gun	완도군
## 200	46900	Jindo-gun	진도군
## 201	46910	Sinan-gun	신안군
## 202	47111	Nam-gu, Pohang-si	포항시 남구
## 203	47113	Buk-gu, Pohang-si	포항시 북구
## 204	47130	Gyeongju-si	경주시
## 205	47150	Gimcheon-si	김천시
## 206	47170	Andong-si	안동시
## 207	47190	Gumi-si	구미시
## 208	47210	Yeongju-si	영주시
## 209	47230	Yeongcheon-si	영천시
## 210	47250	Sangju-si	상주시
## 211	47280	Mungyeong-si	문경시
## 212	47290	Gyeongsan-si	경산시
## 213	47720	Gunwi-gun	군위군
## 214	47730	Uiseong-gun	의성군
## 215	47750	Cheongsong-gun	청송군
## 216	47760	Yeongyang-gun	영양군
## 217	47770	Yeongdeok-gun	영덕군
## 218	47820	Cheongdo-gun	청도군
## 219	47830	Goryeong-gun	고령군
## 220	47840	Seongju-gun	성주군
## 221	47850	Chilgok-gun	칠곡군
## 222	47900	Yecheon-gun	예천군
## 223	47920	Bonghwa-gun	봉화군
## 224	47930	Ulsan-gun	울진군
## 225	47940	Ulleung-gun	울릉군
## 226	48121	Uichang-gu, Changwon-si	창원시 의창구
## 227	48123	Seongsan-gu, Changwon-si	창원시 성산구
## 228	48125	Masanhappo-gu, Changwon-si	창원시 마산합포구
## 229	48127	Masanhoewon-gu, Changwon-si	창원시 마산회원구

```
## 230 48129      Jinhae-gu, Changwon-si      창원시 진해구
## 231 48170              Jinju-si          진주시
## 232 48220      Tongyeong-si             통영시
## 233 48240      Sacheon-si              사천시
## 234 48250      Gimhae-si               김해시
## 235 48270      Miryang-si              밀양시
## 236 48310      Geoje-si                거제시
## 237 48330      Yangsan-si              양산시
## 238 48720      Uiryeong-gun            의령군
## 239 48730      Haman-gun               함안군
## 240 48740      Changnyeong-gun         창녕군
## 241 48820      Goseong-gun             고성군
## 242 48840      Namhae-gun             남해군
## 243 48850      Hadong-gun             하동군
## 244 48860      Sancheong-gun           산청군
## 245 48870      Hamyang-gun            함양군
## 246 48880      Geochang-gun           거창군
## 247 48890      Hapcheon-gun          함천군
## 248 50110              Jeju-si         제주도
## 249 50130      Seogwipo-si            서귀포시
```

```
map2 <- tidy(shp2)
```

```
## Regions defined for each Polygons
```

```
nrow(map2)
```

```
## [1] 1337823
```

```
dsn3 <- "../data/EMD_201703"
shp3 <- readOGR(dsn3,
                layer = "TL_SCCO_EMD",
                stringsAsFactors = FALSE,
                encoding = "CP949")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "../data/EMD_201703", layer: "TL_SCCO_EMD"
## with 5039 features
## It has 3 fields
```

```
# map3 <- fortify(shp3)
map3 <- tidy(shp3)
```

```
## Regions defined for each Polygons
```

```
str(map3)
```

```
## 'data.frame':      3552635 obs. of  7 variables:
## $ long : num  953700 953694 953691 953690 953688 ...
## $ lat  : num  1954605 1954591 1954583 1954577 1954569 ...
## $ order: int   1 2 3 4 5 6 7 8 9 10 ...
## $ hole : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
## $ piece: Factor w/ 238 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ group: Factor w/ 10028 levels "0.1","1.1","2.1",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ id   : chr   "0" "0" "0" "0" ...
```

```
nrow(map3)
```

```
## [1] 3552635
```

```
dsn4 <- "../data/LI_201703"
shp4 <- readOGR(dsn4,
                layer = "TL_SCCO_LI",
                stringsAsFactors = FALSE,
                encoding = "CP949")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "../data/LI_201703", layer: "TL_SCCO_LI"
## with 15178 features
## It has 3 fields
```

```
# map4 <- fortify(shp4)
map4 <- tidy(shp4)
```

```
## Regions defined for each Polygons
```

```
nrow(map4)
```

```
## [1] 6164498
```

Coordinates

```
proj4string(shp1) #> Information regarding projection,
```

```
## [1] "+proj=tmerc +lat_0=38 +lon_0=127.5 +k=0.9996 +x_0=1000000 +y_0=2000000 +ellps
=GRS80 +units=m +no_defs"
```

```
coordinates(shp1) #> 좌표계 확인. 변환 필요
```

```
##      [,1]      [,2]
## 0    955111.8 1950406
## 1    1142809.0 1691681
## 2    1096215.2 1759774
## 3     927633.4 1944269
## 4     939472.0 1684701
## 5     990489.4 1815820
## 6    1157492.6 1730077
## 7     978414.3 1840349
## 8     972089.8 1948336
## 9    1070837.3 1968985
## 10   1029502.4 1860032
## 11   942752.1 1837017
## 12   966028.2 1747116
## 13   950297.4 1660758
## 14   1111294.8 1816892
## 15   1068755.2 1708502
## 16   911965.9 1488516
```

```
shp1_84 <- spTransform(shp1, CRS("+init=epsg:4326")) #> WGS84 좌표 체계로 변환
coordinates(shp1_84) #> 익숙한 경도, 위도 체계로 전환
```

```
##      [,1]      [,2]
## 0    126.9918 37.55192
## 1    129.0691 35.21034
## 2    128.5654 35.82981
## 3    126.6814 37.49489
## 4    126.8354 35.15573
## 5    127.3940 36.33976
## 6    129.2376 35.55423
## 7    127.2587 36.56073
## 8    127.1838 37.53505
## 9    128.3013 37.71855
## 10   127.8313 36.73878
## 11   126.8596 36.52969
## 12   127.1246 35.71979
## 13   126.9560 34.94062
## 14   128.7404 36.34407
## 15   128.2566 35.36974
## 16   126.5537 33.38458
```

```
map1 <- tidy(shp1_84) #> tidy 파일로 변환
```

```
## Regions defined for each Polygons
```

```
shp1_84@data #> data slot 확인
```

```
##      CTPRVN_CD      CTP_ENG_NM      CTP_KOR_NM
## 0             11             Seoul 서울특별시
## 1             26             Busan 부산광역시
## 2             27             Daegu 대구광역시
## 3             28             Incheon 인천광역시
## 4             29             Gwangju 광주광역시
## 5             30             Daejeon 대전광역시
## 6             31             Ulsan 울산광역시
## 7             36             Sejong-si 세종특별자치시
## 8             41             Gyeonggi-do 경기도
## 9             42             Gangwon-do 강원도
## 10            43 Chungcheongbuk-do 충청북도
## 11            44 Chungcheongnam-do 충청남도
## 12            45 Jeollabuk-do 전라북도
## 13            46 Jellanam-do 전라남도
## 14            47 Gyeongsangbuk-do 경상북도
## 15            48 Gyeongsangnam-do 경상남도
## 16            50             Jeju-do 제주특별자치도
```

```
# shp1_84$AB_name <- substr(shp1_84$CTP_KOR_NM, 1, 2) #> "충청", "충청" 등 duplicate labels
shp1_84$AB_name <- c("서울", "부산", "대구", "인천", "광주", "대전", "울산", "세종", "경기", "강원", "충북", "충남", "전북", "전남", "경북", "경남", "제주")
shp1_84@data
```

```
##      CTPRVN_CD      CTP_ENG_NM      CTP_KOR_NM AB_name
## 0             11             Seoul 서울특별시 서울
## 1             26             Busan 부산광역시 부산
## 2             27             Daegu 대구광역시 대구
## 3             28             Incheon 인천광역시 인천
## 4             29             Gwangju 광주광역시 광주
## 5             30             Daejeon 대전광역시 대전
## 6             31             Ulsan 울산광역시 울산
## 7             36             Sejong-si 세종특별자치시 세종
## 8             41             Gyeonggi-do 경기도 경기
## 9             42             Gangwon-do 강원도 강원
## 10            43 Chungcheongbuk-do 충청북도 충북
## 11            44 Chungcheongnam-do 충청남도 충남
## 12            45 Jeollabuk-do 전라북도 전북
## 13            46 Jellanam-do 전라남도 전남
## 14            47 Gyeongsangbuk-do 경상북도 경북
## 15            48 Gyeongsangnam-do 경상남도 경남
## 16            50             Jeju-do 제주특별자치도 제주
```

```
str(map1)
```

```
## 'data.frame': 798830 obs. of 7 variables:
## $ long : num 127 127 127 127 127 ...
## $ lat : num 37.6 37.6 37.6 37.6 37.6 ...
## $ order: int 1 2 3 4 5 6 7 8 9 10 ...
## $ hole : logi FALSE FALSE FALSE FALSE FALSE ...
## $ piece: Factor w/ 2226 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ group: Factor w/ 4550 levels "0.1","0.2","0.3",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ id : chr "0" "0" "0" "0" ...
```

Plots

Simplified Versions (from mapshaper (<https://mapshaper.org>))

시도

```
shp_simp <- readOGR(dsn = "../data/TL_SCCO_CTPRVN",  
  layer = "TL_SCCO_CTPRVN")
```

```
## OGR data source with driver: ESRI Shapefile  
## Source: "../data/TL_SCCO_CTPRVN", layer: "TL_SCCO_CTPRVN"  
## with 17 features  
## It has 1 fields
```

```
shp_simp@data #> data file 이 빈약한 것 확인.
```

```
##      FID  
## 0      0  
## 1      1  
## 2      2  
## 3      3  
## 4      4  
## 5      5  
## 6      6  
## 7      7  
## 8      8  
## 9      9  
## 10     10  
## 11     11  
## 12     12  
## 13     13  
## 14     14  
## 15     15  
## 16     16
```

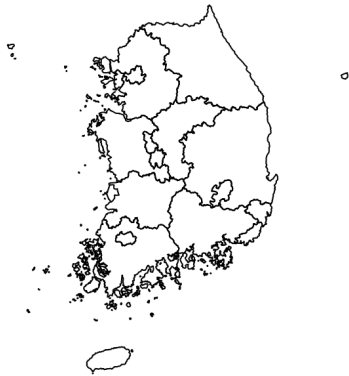
```
shp_simp@data <- shp1_84@data #> 원 데이터의 data 슬롯 복사  
shp_simp@data #> `data` 슬롯 확인
```

##	CTPRVN_CD	CTP_ENG_NM	CTP_KOR_NM	AB_name
## 0	11	Seoul	서울특별시	서울
## 1	26	Busan	부산광역시	부산
## 2	27	Daegu	대구광역시	대구
## 3	28	Incheon	인천광역시	인천
## 4	29	Gwangju	광주광역시	광주
## 5	30	Daejeon	대전광역시	대전
## 6	31	Ulsan	울산광역시	울산
## 7	36	Sejong-si	세종특별자치시	세종
## 8	41	Gyeonggi-do	경기도	경기
## 9	42	Gangwon-do	강원도	강원
## 10	43	Chungcheongbuk-do	충청북도	충북
## 11	44	Chungcheongnam-do	충청남도	충남
## 12	45	Jeollabuk-do	전라북도	전북
## 13	46	Jeollanam-do	전라남도	전남
## 14	47	Gyeongsangbuk-do	경상북도	경북
## 15	48	Gyeongsangnam-do	경상남도	경남
## 16	50	Jeju-do	제주특별자치도	제주

```
summary(shp_simp) #> 요약
```

```
## Object of class SpatialPolygonsDataFrame  
## Coordinates:  
##      min      max  
## x  746110.3 1302026  
## y 1464332.9 2065895  
## Is projected: TRUE  
## proj4string :  
## [+proj=tmerc +lat_0=38 +lon_0=127.5 +k=0.9996 +x_0=1000000  
## +y_0=2000000 +ellps=GRS80 +units=m +no_defs]  
## Data attributes:  
##      CTPRVN_CD      CTP_ENG_NM      CTP_KOR_NM  
## Length:17      Length:17      Length:17  
## Class :character Class :character Class :character  
## Mode :character Mode :character Mode :character  
##      AB_name  
## Length:17  
## Class :character  
## Mode :character
```

```
plot(shp_simp)
```



```
ogrInfo(dsn = "../data/TL_SCCO_CTPRVN", #> `ogrInfo()`  
layer = "TL_SCCO_CTPRVN")
```

```
## Source: "../data/TL_SCCO_CTPRVN", layer: "TL_SCCO_CTPRVN"  
## Driver: ESRI Shapefile; number of rows: 17  
## Feature type: wkbPolygon with 2 dimensions  
## Extent: (746110.3 1464333) - (1302026 2065895)  
## CRS: +proj=tmerc +lat_0=38 +lon_0=127.5 +k=0.9996 +x_0=1000000 +y_0=2000000 +ellps  
=GRS80 +units=m +no_defs  
## LDID: 0  
## Number of fields: 1  
##   name type length typeName  
## 1  FID    0       2 Integer
```

```
proj4string(shp_simp) #> .shp 파일을 mapshaper.org 에서 간략히 하면 .prj 는 생성되지 않음.
```

```
## [1] "+proj=tmerc +lat_0=38 +lon_0=127.5 +k=0.9996 +x_0=1000000 +y_0=2000000 +ellps  
=GRS80 +units=m +no_defs"
```

```
#> 원시 layer에 제공되어 있는 .prj 파일을 옮겨와야 함.  
coordinates(shp_simp) #> 좌표 확인. 변환 필요한 좌표계.
```

```
##           [,1]      [,2]  
## 0    955105.3 1950416  
## 1   1142766.5 1691632  
## 2   1096232.3 1759777  
## 3    927630.9 1944237  
## 4    939495.1 1684721  
## 5    990490.9 1815810  
## 6   1157518.7 1730063  
## 7    978425.2 1840334  
## 8    972068.5 1948316  
## 9   1070838.0 1968967  
## 10  1029500.1 1860033  
## 11   942798.0 1836970  
## 12   966036.2 1747105  
## 13   950318.4 1660785  
## 14  1111289.1 1816889  
## 15  1068775.8 1708454  
## 16   911955.6 1488494
```

```
shp_simp84 <- shp_simp %>%  
  spTransform(CRS("+init=epsg:4326")) #> 통상의 WGS84 좌표계로 변환  
coordinates(shp_simp84) #> 통상의 좌표계로 변환된 것을 확인함.
```

```
##           [,1]      [,2]  
## 0   126.9917 37.55201  
## 1   129.0686 35.20991  
## 2   128.5655 35.82984  
## 3   126.6813 37.49461  
## 4   126.8357 35.15591  
## 5   127.3941 36.33968  
## 6   129.2379 35.55410  
## 7   127.2588 36.56059  
## 8   127.1836 37.53487  
## 9   128.3013 37.71839  
## 10  127.8313 36.73878  
## 11  126.8601 36.52927  
## 12  127.1247 35.71969  
## 13  126.9562 34.94087  
## 14  128.7403 36.34404  
## 15  128.2568 35.36931  
## 16  126.5536 33.38438
```

```
names(shp_simp84@data) <- c("code", "name_eng", "name_kor", "AB_name") #> 변수명 바꿈.  
str(shp_simp84@data) #> 변수명 등 확인.
```

```
## 'data.frame':   17 obs. of  4 variables:  
##  $ code   : chr  "11" "26" "27" "28" ...  
##  $ name_eng: chr  "Seoul" "Busan" "Daegu" "Incheon" ...  
##  $ name_kor: chr  "서울특별시" "부산광역시" "대구광역시" "인천광역시" ...  
##  $ AB_name : chr  "서울" "부산" "대구" "인천" ...
```

```
# geocode(shp_simp$name_eng)
# map_simp <- tidy(shp_simp84) #> `tidy` 데이터로 변환
# str(map_simp) #> 데이터 구조 확인
shp_simp84$id <- rownames(shp_simp84@data) #> `id` 변수 생성. `left_join()`에 반드시 필요
함. `tidy()`의 결과로 나오는 데이터 프레임에는 `@data`의 `code`, `region`의 정보가 누락됨.
shp_simp84@data #> 데이터 슬롯 확인
```

```
##      code      name_eng      name_kor AB_name id
## 0      11         Seoul      서울특별시 서울    0
## 1      26         Busan      부산광역시 부산    1
## 2      27         Daegu      대구광역시 대구    2
## 3      28         Incheon    인천광역시 인천    3
## 4      29         Gwangju     광주광역시 광주    4
## 5      30         Daejeon     대전광역시 대전    5
## 6      31         Ulsan       울산광역시 울산    6
## 7      36         Sejong-si   세종특별자치시 세종    7
## 8      41         Gyeonggi-do  경기도      경기    8
## 9      42         Gangwon-do   강원도      강원    9
## 10     43 Chungcheongbuk-do   충청북도    충북    10
## 11     44 Chungcheongnam-do   충청남도    충남    11
## 12     45         Jeollabuk-do  전라북도    전북    12
## 13     46         Jellanam-do   전라남도    전남    13
## 14     47 Gyeongsangbuk-do    경상북도    경북    14
## 15     48 Gyeongsangnam-do    경상남도    경남    15
## 16     50         Jeju-do      제주특별자치도 제주    16
```

```
map_coords <- coordinates(shp_simp84) #> `geom_text()`에 필요한 좌표 복사
colnames(map_coords) <- c("long", "lat") #> 좌표계 이름 명칭.
map_coords # 좌표 확인
```

```
##      long      lat
## 0  126.9917  37.55201
## 1  129.0686  35.20991
## 2  128.5655  35.82984
## 3  126.6813  37.49461
## 4  126.8357  35.15591
## 5  127.3941  36.33968
## 6  129.2379  35.55410
## 7  127.2588  36.56059
## 8  127.1836  37.53487
## 9  128.3013  37.71839
## 10 127.8313  36.73878
## 11 126.8601  36.52927
## 12 127.1247  35.71969
## 13 126.9562  34.94087
## 14 128.7403  36.34404
## 15 128.2568  35.36931
## 16 126.5536  33.38438
```

```
map_data <- cbind(map_coords, shp_simp84@data) # shp_simp@data 에 좌표 추가
# map_data
# map_df <- left_join(map_simp, shp_simp84@data, c("id" = "id")) #> `map_simp` 에 `shp
_sim@data` 추가
# str(map_df)
shp_simp84$name_kor
```

```
## [1] "서울특별시"      "부산광역시"      "대구광역시"      "인천광역시"
## [5] "광주광역시"      "대전광역시"      "울산광역시"      "세종특별자치시"
## [9] "경기도"          "강원도"          "충청북도"        "충청남도"
## [13] "전라북도"        "전라남도"        "경상북도"        "경상남도"
## [17] "제주특별자치도"
```

```
shp_simp84$AB_name
```

```
## [1] "서울" "부산" "대구" "인천" "광주" "대전" "울산" "세종" "경기" "강원"
## [11] "충북" "충남" "전북" "전남" "경북" "경남" "제주"
```

```
shp_simp84$name_kor_f <- factor(shp_simp84$name_kor,
                                levels = shp_simp84$name_kor,
                                labels = shp_simp84$AB_name)
shp_simp84$region <- shp_simp84$code
shp_simp84_df <- tidy(shp_simp84)
```

```
## Regions defined for each Polygons
```

```
# map_df$name_kor_f <- factor(map_df$name_kor,
#                             levels = shp_simp84$name_kor,
#                             labels = shp_simp84$AB_name)
# str(map_df)
str(shp_simp84_df)
```

```
## 'data.frame':    11660 obs. of  7 variables:
## $ long : num  127 127 127 127 127 ...
## $ lat : num  37.6 37.6 37.7 37.7 37.7 ...
## $ order: int  1 2 3 4 5 6 7 8 9 10 ...
## $ hole : logi FALSE FALSE FALSE FALSE FALSE ...
## $ piece: Factor w/ 117 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ group: Factor w/ 215 levels "0.1","1.1","2.1",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ id : chr  "0" "0" "0" "0" ...
```

```
str(shp_simp84@data)
```



```
## 'data.frame':   17 obs. of  7 variables:
## $ code       : chr  "11" "26" "27" "28" ...
## $ name_eng    : chr  "Seoul" "Busan" "Daegu" "Incheon" ...
## $ name_kor    : chr  "서울특별시" "부산광역시" "대구광역시" "인천광역시" ...
## $ AB_name     : chr  "서울" "부산" "대구" "인천" ...
## $ id          : chr  "0" "1" "2" "3" ...
## $ name_kor_f  : Factor w/ 17 levels "서울","부산",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ region     : chr  "11" "26" "27" "28" ...
```

```
map_df <- left_join(shp_simp84_df, shp_simp84@data, by = c("id" = "id"))
str(map_df)
```

```
## 'data.frame':   11660 obs. of  13 variables:
## $ long       : num  127 127 127 127 127 ...
## $ lat        : num  37.6 37.6 37.7 37.7 37.7 ...
## $ order      : int  1 2 3 4 5 6 7 8 9 10 ...
## $ hole       : logi  FALSE FALSE FALSE FALSE FALSE ...
## $ piece      : Factor w/ 117 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ group      : Factor w/ 215 levels "0.1","1.1","2.1",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ id         : chr  "0" "0" "0" "0" ...
## $ code       : chr  "11" "11" "11" "11" ...
## $ name_eng    : chr  "Seoul" "Seoul" "Seoul" "Seoul" ...
## $ name_kor    : chr  "서울특별시" "서울특별시" "서울특별시" "서울특별시" ...
## $ AB_name     : chr  "서울" "서울" "서울" "서울" ...
## $ name_kor_f  : Factor w/ 17 levels "서울","부산",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ region     : chr  "11" "11" "11" "11" ...
```

```
Dokdo_coord <- geocode("Dokdo")
```

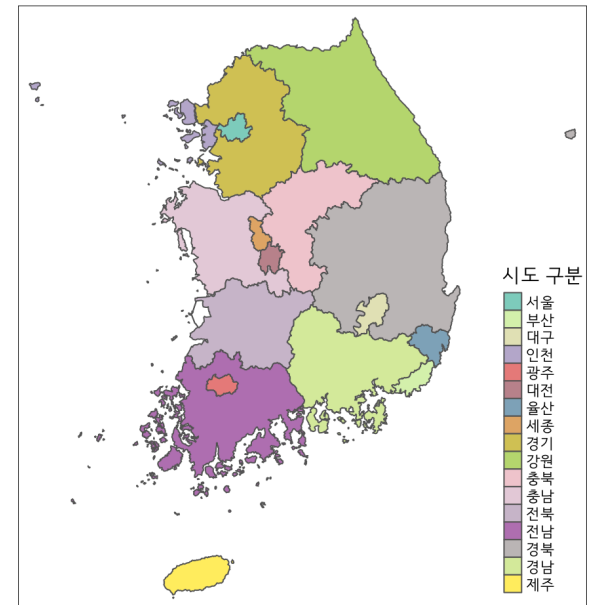
```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Dokdo&sensor=false
```

```
Ulleung_coord <- geocode("Ulleungdo")
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Ulleungdo&sensor=false
```

시도 이름 삽입

```
library(tmap)
qtm(shp_simp84, fill = "name_kor_f", fill.title = "시도 구분") +
  tm_layout(fontfamily = "HCR Dotum LVT")
```

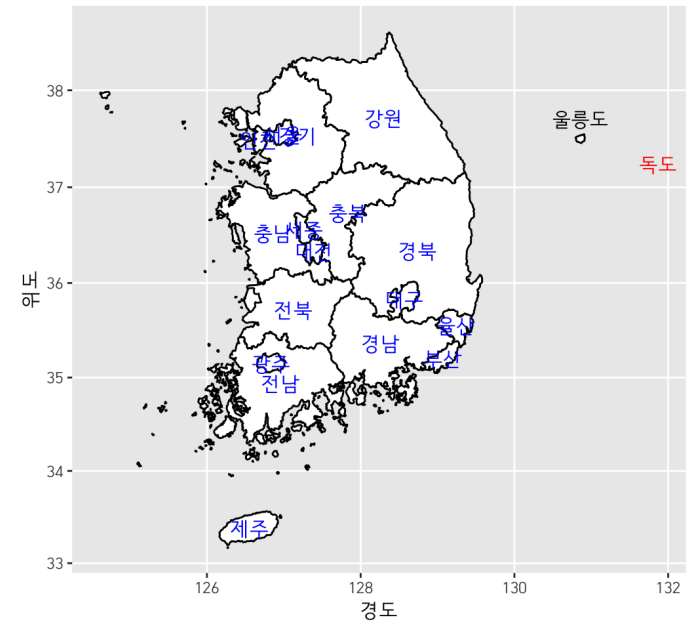


```
qtm(shp_simp84, text = "name_kor_f", text.size = 1) +
  tm_layout(fontfamily = "HCR Dotum LVT")
```

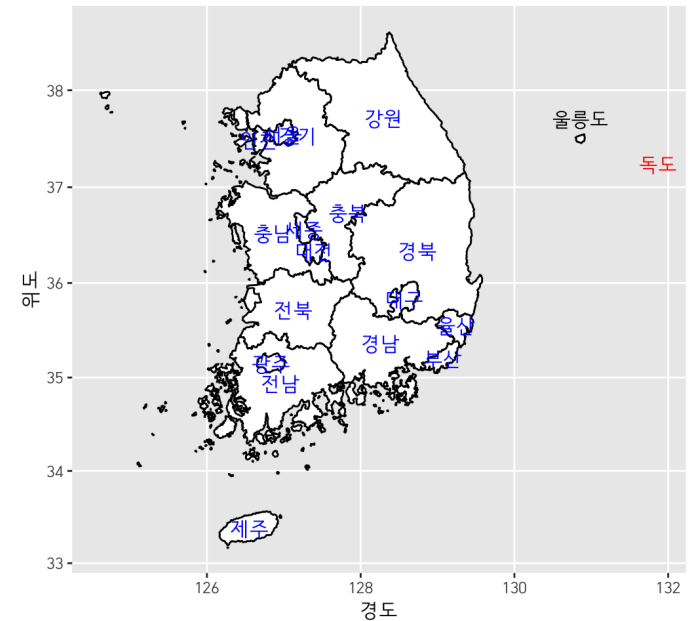


#> "discrete value supplied to continuous scale" 에러가 발생하면, rstudio 버전을 체크해 볼 것.
 특히 inline 코드로 는 에러가 발생하지 않는다면 rstudio 버전 문제일 가능성이 큼.

```
ggplot(data = map_df,
       mapping = aes(x = long,
                     y = lat,
                     group = group)) +
  geom_polygon(fill = "white",
              colour = "black") +
  coord_map() +
  geom_text(data = map_data,
            mapping = aes(x = long,
                          y = lat,
                          label = AB_name),
            inherit.aes = FALSE,
            family = "HCR Dotum LVT",
            size = 4,
            colour = "blue") +
  theme(axis.title.x = element_text(family = "HCR Dotum LVT"),
        axis.title.y = element_text(family = "HCR Dotum LVT"),
        text = element_text(family = "HCR Dotum LVT")) +
  labs(x = "경도",
       y = "위도") +
  annotate("text",
         x = Dokdo_coord$lon,
         y = Dokdo_coord$lat,
         label = "독도",
         family = "HCR Dotum LVT",
         colour = "red") +
  annotate("text",
         x = Ulleung_coord$lon,
         y = Ulleung_coord$lat + 0.2,
         label = "울릉도",
         family = "HCR Dotum LVT",
         colour = "black")
```

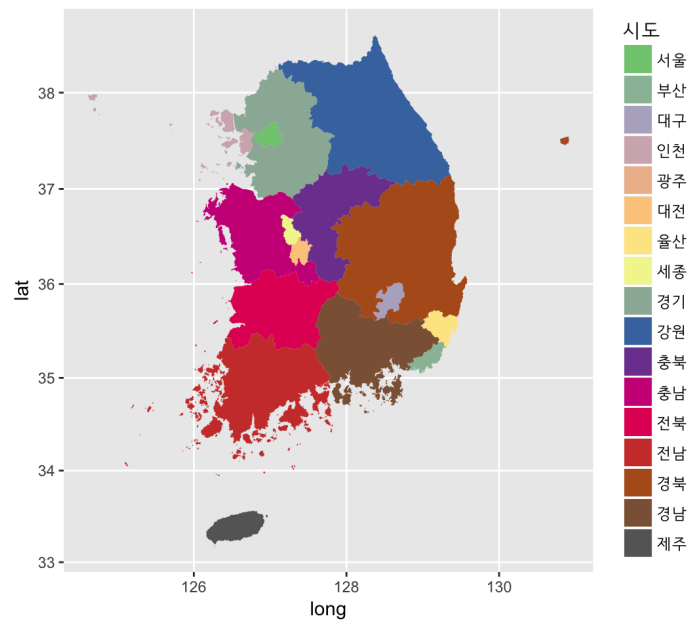


```
#> data frame으로 변환하여 그리는 전형적인 과정
ggplot(data = shp_simp84_df,
       mapping = aes(x = long,
                     y = lat,
                     group = group)) +
  geom_polygon(fill = "white",
              colour = "black") +
  coord_map() +
  geom_text(data = map_data,
            mapping = aes(x = long,
                          y = lat,
                          label = AB_name),
            inherit.aes = FALSE,
            family = "HCR Dotum LVT",
            size = 4,
            colour = "blue") +
  theme(axis.title.x = element_text(family = "HCR Dotum LVT"),
        axis.title.y = element_text(family = "HCR Dotum LVT"),
        text = element_text(family = "HCR Dotum LVT")) +
  labs(x = "경도",
       y = "위도") +
  annotate("text",
         x = Dokdo_coord$lon,
         y = Dokdo_coord$lat,
         label = "독도",
         family = "HCR Dotum LVT",
         colour = "red") +
  annotate("text",
         x = Ulleung_coord$lon,
         y = Ulleung_coord$lat + 0.2,
         label = "울릉도",
         family = "HCR Dotum LVT",
         colour = "black")
```



색으로 구분

```
library(RColorBrewer)
fill.pal <- colorRampPalette(brewer.pal(8, "Accent"))(17)
ggplot(data = map_df,
       mapping = aes(x = long,
                     y = lat,
                     group = group,
                     fill = name_kor_f)) +
  geom_polygon() +
  # geom_polygon(fill = "white",
  #             colour = "black") +
  coord_map() +
  theme(legend.text = element_text(family = "HCR Dotum LVT"),
        legend.title = element_text(family = "HCR Dotum LVT")) +
  guides(fill = guide_legend(title = "시도")) +
  # scale_fill_brewer()
  scale_fill_manual(values = fill.pal)
```



Save

```
save.image("./korea_map.RData")
```