HW1-1_201823869_조성우

조성우

2020 4 30

데이터 프레임 전처리

1.데이터 불러오기

*구글드라이브로부터 csv 불러들이기

```
customerURL <- "http://bitly.kr/Mv61KM6A5"
Customer <- read.csv(header=T,customerURL,stringsAsFactors = F) #구글드라이브의 파일을 R이 string을 Factor로 변환하지 않는 조건으로 불러들인다
str(Customer)
```

```
## 'data.frame':    22400 obs. of  18 variables:
##  $ 고객ID        : int  1 2 3 4 5 6 7 8 9 12 ...
##  $ 데이터셋.구분 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ 보험사기자여부: chr  "N" "N" "N" "N" ...
##  $ 성별          : int  2 1 1 2 2 1 2 1 1 1 ...
##  $ 연령          : int  47 53 60 64 54 62 60 57 54 58 ...
##  $ 주택가격      : int  21111 40000 0 12861 0 6218 11388 86527 22638 37222 ...
##  $ 거주TYPE      : int  20 20 NA 40 NA 99 30 20 20 20 ...
##  $ FP경력        : chr  "N" "N" "N" "Y" ...
##  $ 고객등록년월  : int  199910 199910 199910 199910 199910 199910 199910 199910 199910 199910 ...
##  $ 시도구분      : chr  "충북" "서울" "서울" "경기" ...
##  $ 직업그룹코드1 : chr  "사무직" "사무직" "서비스" "자영업" ...
##  $ 직업그룹코드2 : chr  "사무직" "사무직" "2차산업 종사자" "3차산업 종사자" ...
##  $ 추정가구소득1 : int  10094 9143 0 4270 0 0 0 12219 7553 10466 ...
##  $ 추정가구소득2 : int  11337 6509 4180 5914 8885 6449 3611 12063 9821 13858 ...
##  $ X             : chr  "" "" "" "" ...
##  $ X.1           : chr  "" "" "" "" ...
##  $ X.2           : chr  "" "" "" "" ...
##  $ X.3           : chr  "" "" "" "" ...
```

2. 데이터 전처리

2.1 불필요한 변수 삭제

*해당 Customer 데이터프레임의 구조를 살펴보면 X X.1 X.2 X.3 에 해당하는 변수 네 가지는 유의미한 value를 가지고 있지 않으므로 지워줄 필요가 있습니다.

```
#해당 Customer 데이터프레임의 구조를 살펴보면 X X.1 X.2 X.3 에 해당하는 변수 네 가지는 유의미한 value를 가지고 있지 않으므로 지워줄 필요가 있습니다.

Customer <- Customer[,-c(15:18)] #15,16,17행을 지운 Customer을 Customer에 재할당 (x,x1,x2,x3 변수 삭제)

str(Customer)
```

```
## 'data.frame':    22400 obs. of  14 variables:
##  $ 고객ID        : int  1 2 3 4 5 6 7 8 9 12 ...
##  $ 데이터셋.구분 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ 보험사기자여부: chr  "N" "N" "N" "N" ...
##  $ 성별          : int  2 1 1 2 2 1 2 1 1 1 ...
##  $ 연령          : int  47 53 60 64 54 62 60 57 54 58 ...
##  $ 주택가격      : int  21111 40000 0 12861 0 6218 11388 86527 22638 37222 ...
##  $ 거주TYPE      : int  20 20 NA 40 NA 99 30 20 20 20 ...
##  $ FP경력        : chr  "N" "N" "N" "Y" ...
##  $ 고객등록년월  : int  199910 199910 199910 199910 199910 199910 199910 199910 199910 19991
0 ...
##  $ 시도구분      : chr  "충북" "서울" "서울" "경기" ...
##  $ 직업그룹코드1 : chr  "사무직" "사무직" "서비스" "자영업" ...
##  $ 직업그룹코드2 : chr  "사무직" "사무직" "2차산업 종사자" "3차산업 종사자" ...
##  $ 추정가구소득1 : int  10094 9143 0 4270 0 0 0 12219 7553 10466 ...
##  $ 추정가구소득2 : int  11337 6509 4180 5914 8885 6449 3611 12063 9821 13858 ...
```

2.2 NA값 확인

```
#통계함수를 사용하는 등의 처리를 위해 N/A값을 알맞게 변경해줘야 하는데 예를들어 추정 가구소득 등의 평균을 구하는 상황에서 N/A값을 0으로 설정해버리면 해당값이 extreme value로써 통계에 불합리하게 작용할 것이므로 굳이 0으로 조정하지 않고 통계처리를 할 때 통계함수의 매개변수를 입력할 때 na.rm=T 을 적용하여 NA를 배제하고 통계를 분석할 것입니다.

#관심있게 다룰만한 변수만 골라서 NA값 존재여부를 검사해봤습니다.

sum(is.na(Customer$연령))   #연령 변수엔 결측값이 존재하지 않음을 볼 수 있다
```

```
## [1] 0
```

```
sum(is.na(Customer$직업그룹코드1)) #직업그룹코드1 변수엔 결측값이 존재하지 않음을 볼 수 있다
```

```
## [1] 0
```

```
sum(is.na(Customer$시도구분)) #시도구분 변수엔 결측값이 존재하지 않음을 볼 수 있다.
```

```
## [1] 0
```

```
sum(is.na(Customer$FP경력)) #FP경력 변수엔 결측값이 존재하지 않음을 볼 수 있다.
```

```
## [1] 0
```

```
sum(is.na(Customer$주택가격)) #주택가격 변수엔 결측값이 존재하지 않음을 볼 수 있다.
```

```
## [1] 0
```

```
sum(is.na(Customer$추정가구소득1)) #추정가구소득1 변수엔 결측값이 존재하지 않음을 볼 수 있다
```

```
## [1] 0
```

```
sum(is.na(Customer$추정가구소득2)) #추정가구소득2 변수엔 결측값이 680개 존재함을 확인할 수 있다.
```

```
## [1] 680
```

```
#검사결과 NA가 존재하는 변수는 추정가구소득2 뿐이므로,주의해야 할 것은 추정가구소득2의 통계분석입니다. 이는 뒤에서 rm.na=T 를 활용할 것입니다.
```

3.통계분석

3.1.통계요약

```
summary(Customer) #Customer내의 각 변수에 관한 기본적 통계 요약출력
```

```
##      고객ID      데이터셋.구분  보험사기자여부          성별
##  Min.   :    1   Min.   :1.00   Length:22400       Min.   :1.000
##  1st Qu.: 5601   1st Qu.:1.00   Class :character   1st Qu.:1.000
##  Median :11200   Median :1.00   Mode  :character   Median :2.000
##  Mean   :11200   Mean   :1.08                      Mean   :1.565
##  3rd Qu.:16800   3rd Qu.:1.00                      3rd Qu.:2.000
##  Max.   :22400   Max.   :2.00                      Max.   :2.000
##
##       연령          주택가격          거주TYPE        FP경력
##  Min.   : 2.00   Min.   :     0   Min.   :11.00   Length:22400
##  1st Qu.:34.00   1st Qu.:  6733   1st Qu.:20.00   Class :character
##  Median :46.00   Median : 12222   Median :20.00   Mode  :character
##  Mean   :44.73   Mean   : 15914   Mean   :25.77
##  3rd Qu.:56.00   3rd Qu.: 20988   3rd Qu.:30.00
##  Max.   :89.00   Max.   :305555   Max.   :99.00
##                                   NA's   :1254
##   고객등록년월       시도구분          직업그룹코드1      직업그룹코드2
##  Min.   :   101   Length:22400       Length:22400       Length:22400
##  1st Qu.:200306   Class :character   Class :character   Class :character
##  Median :200306   Mode  :character   Mode  :character   Mode  :character
##  Mean   :198924
##  3rd Qu.:200402
##  Max.   :201602
##  NA's   :456
##  추정가구소득1   추정가구소득2
##  Min.   :    0   Min.   :    0
##  1st Qu.: 2995   1st Qu.: 3558
##  Median : 4807   Median : 4681
##  Mean   : 4769   Mean   : 5198
##  3rd Qu.: 6607   3rd Qu.: 6840
##  Max.   :19829   Max.   :25872
##                  NA's   :680
```

*위의 요약된 통계분석 중 제가 관심있는 몇가지 변수만 조금 더 구체적으로 살펴보겠습니다.

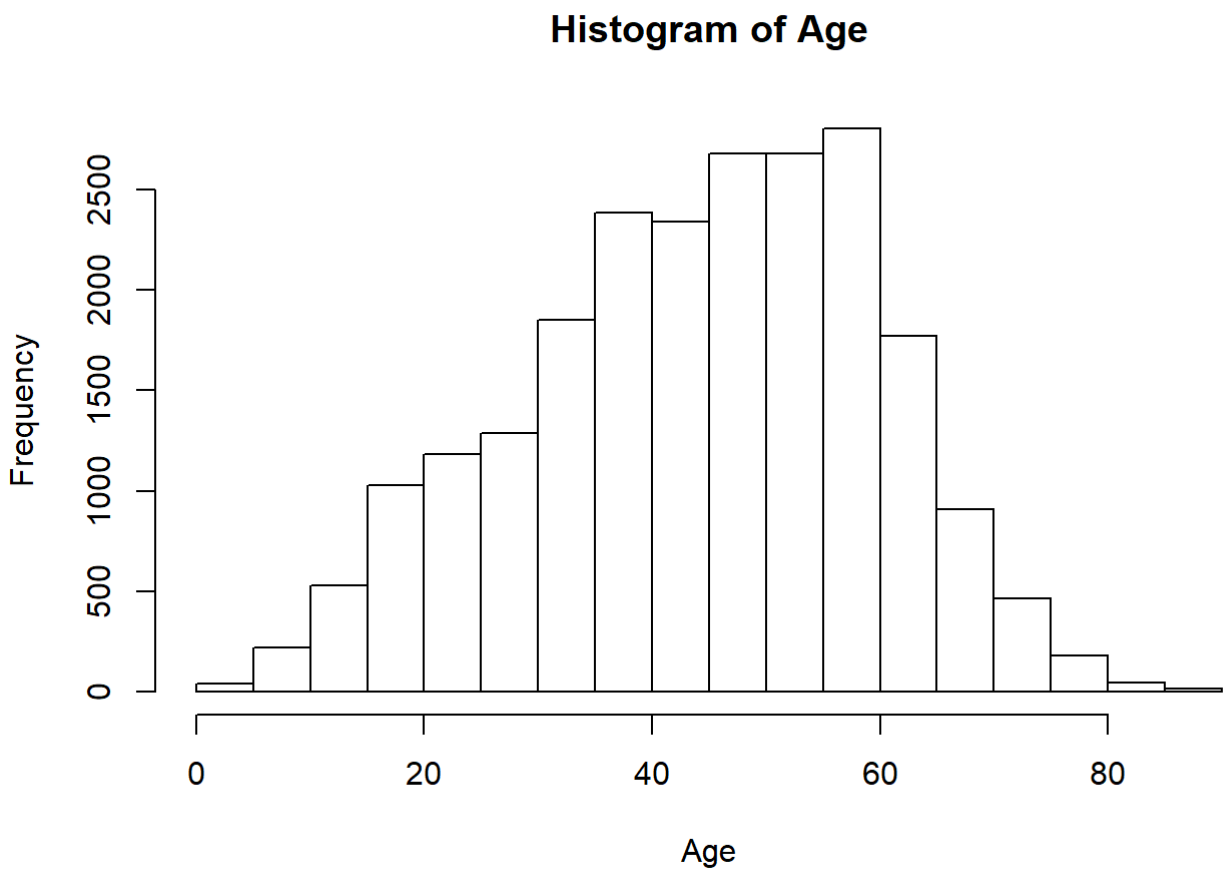
3.2 연령통계

3.2.1 Histogram

```
#먼저 연령입니다 , 고객을 연령대별로 통계하겠습니다.


Age <- Customer$연령 #연령 열만 Age로 선언

hist(Age) #먼저 hist()함수로 해당 변수의 histogram에 관해 시각적으로 살펴보고 가겠습니다.
```




```
#이를 보고 40~60세 사이의 고객수가 가장 많음을 대략적으로 파악할 수 있습니다.
```

3.2.2 각종 measures

```
#위의 Age 히스토그램에서 살펴본 내용을 조금 더 자세히 알아보겠습니다.

min(Age)   #최연소 고객은 2세
```

```
## [1] 2
```

```
max(Age)   #최고령 고객은 89세
```

```
## [1] 89
```

```
mean(Age)  #평균 나이 44.73세
```

```
## [1] 44.73487
```

```
median(Age) #중앙값 46세
```

```
## [1] 46
```

3.2.3 Defining function / Frequencies

```
#연령대로 class를 나누어 정확한 빈도수의 값을 알고싶습니다. 최연소 2세,최고령 89세 고객을 확인했으므로 0~90까지로 Range를 설정하겠습니다.
#7주차에 배운 function의 Define을 활용하여 과정을 함수화시키겠습니다.

Age.Range.Freq <-function(x,a,b){   # 매개변수를 x,a,b로 X에 vector를받아 a보다 크고 b보다 작은 값의 빈도를 간편하게 세기 위해 함수를 만들겠습니다
  c <- length(which(x>a & x<=b))
  return(c)}                        # 결과로 c를 반환하도록 설정합니다

Age.Range.Freq(Age,0,10)  #0~10살 사이의 고객 수    #call
```

```
## [1] 259
```

```
Age.Range.Freq(Age,10,20) #10~20살 사이의 고객 수   #call
```

```
## [1] 1556
```

```
Age.Range.Freq(Age,20,30) #20~30살 사이의 고객 수   #call
```

```
## [1] 2470
```

```
Age.Range.Freq(Age,30,40) #30~40살 사이의 고객 수   #call
```

```
## [1] 4235
```

```
Age.Range.Freq(Age,40,50) #40~50살 사이의 고객 수   #call
```

```
## [1] 5018
```

```
Age.Range.Freq(Age,50,60) #50~60살 사이의 고객 수   #call
```

```
## [1] 5479
```

```
Age.Range.Freq(Age,60,70) #60~70살 사이의 고객 수   #call
```

```
## [1] 2676
```

```
Age.Range.Freq(Age,70,80) #70~80살 사이의 고객 수   #call
```

```
## [1] 644
```

```
Age.Range.Freq(Age,80,90) #80~90살 사이의 고객 수   #call
```

```
## [1] 63
```

#50~60의 클래스가 가장 큽니다. 보험사기자여부의 항목이 있는것으로 보아 해당 데이터셋은 증권사 또는 보험사의 고객데이터셋으로 추정되는데, 50~60세의 장년층은 일반적인 라이프사이클에 비춰 봤을때 저축해둔 자산이 비교적 많거나, 노후에 대비해 어느정도의 투자를 해뒀을 확률이 높을 나이입니다. 해당 통계결과는 이러한 사실을 반영하는것으로 보입니다.

3.3 FP경력 여부와 타변수간 관계분석

3.3.1 FP경력 여부와 주택가격 간 관계분석

```
#추정가구소득의 통계를 분석해보겠습니다. 추정가구소득1을 할 수도 있겠지만 , 해당 데이터셋에서 NA값을 가진  유일한 변수인 추정가구소득2를 다루도록 하겠습니다.

#은행 FP경력이 가구소득 및 거주주택의 가격에 영향이 있는지 알아보고싶습니다. FP경력과 가구소득, 주택가격 의 관계를 유추하고자 합니다.

#FP경력과 주택가격의 관계 분석
CustFPY <- Customer[Customer$FP경력=="Y",] #Customer중 FP경력이 있는 고객만 서브셋 하겠습니다.
CustFPN <- Customer[Customer$FP경력=="N",] #Customer중 Fp경력이 없는 고객만 서브셋 하겠습니다.

FPYHouseP<- mean(CustFPY$주택가격) #FP경력이 있는 고객들의 주택가격의 평균을 선언
FPNHouseP<- mean(CustFPN$주택가격) #FP경력이 없는 고객들의 주택가격의 평균을 선언

#이 둘을 비교해 본 결과 오히려 FP경력이 없는 고객의 주택가격이 근소하게 높습니다. FP경력과 주택가격간 관계는 없거나 약한(weak),부정적(negative) 관계입니다
FPYHouseP>FPNHouseP
```

```
## [1] FALSE
```

```
FPYHouseP
```

```
## [1] 15514.63
```

```
FPNHouseP
```

```
## [1] 15937.4
```

3.3.2 FP경력 여부와 추정가구소득2 간 관계분석

```
#FP경력과 추정가구소득2의 상관관계
FPYIncome <- mean(CustFPY$추정가구소득2,na.rm=T) #FP경력이 있는 고객들의 평균 추정가구소득2, 추정가구소득2엔 NA값이 포함돼있는것을 잊지말고 na.rm=T 로 설정하여 분석
FPNIncome <- mean(CustFPN$추정가구소득2,na.rm=T) #FP경력이 없는 고객들의 평균 추정가구소득2

FPYIncome>FPNIncome ; FPYIncome;FPNIncome
```

```
## [1] TRUE
```

```
## [1] 6157.298
```

```
## [1] 5142.401
```

#FP경력이 있는 고객과 없는 고객의 평균소득을 분석해 본 결과,주택가격과의 관계와 다르게 FP경력고객이 무경력고객에 비해 소득평균이 대략 1000정도 앞서는 것으로 FP경력과 가구소득이 긍정의 관계임을 보여주며,이 차이값이 유의미한 수준입니다. 이로부터 FP경력이 추정가구소득2에 긍정적인(Positive) 영향력을 가진다고 유추해 볼 수 있습니다.(물론 이는 0값과 ,여타 다른 변수를 무시하여 과장된 값일 수 있습니다)

3.4 새 통계를 포함한 변수를 기존 데이터프레임에 추가

```
#추정가구소득1의 평균과 각 고객 별 평균 추정가구소득1 과의 차이 계산

IncomeDiff <- Customer$추정가구소득1 - mean(Customer$추정가구소득1)

Customer$IncomeDiff <- IncomeDiff #IncomeDiff를  Customer 데이터프레임의 새로운 변수로 추가

head(Customer) #해당변수의 앞 5개값만 보여줍니다.
```

	고객ID <int>	데이터셋.구분 <int>	보험사기자여부 <chr>	성별 <int>	연령 <int>	주택가격 <int>	거주TYPE <int>	FP경력 <chr>	고객등록년월 <int>
1	1	1	N	2	47	21111	20	N	199910
2	2	1	N	1	53	40000	20	N	199910
3	3	1	N	1	60	0	*NA*	N	199910
4	4	1	N	2	64	12861	40	Y	199910
5	5	1	N	2	54	0	*NA*	Y	199910
6	6	1	N	1	62	6218	99	N	199910

6 rows | 1-10 of 16 columns

지도 시각화

```
library(dplyr) # 파이프함수를 사용하기 위해 dplyr 패키지를 불러옴
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## 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
```

```
#install.packages("leaflet") #지도 제작을 위해 leaflet 패키지를 불러옴
library(leaflet)
```

```
## Warning: package 'leaflet' was built under R version 3.6.3
```

```
covid_case <- read.csv("http://bitly.kr/C5ykr25ql", stringsAsFactors = F)  # 문자열을 팩터로 변환하지 않는다는 조건으로 해당 링크에서 다루고자 하는 데이터프레임을 가져옵니다.
str(covid_case)
```

```
## 'data.frame':    102 obs. of  8 variables:
##  $ case_id       : int  1000001 1000002 1000003 1000004 1000005 1000006 1000007 1000008 1000009 1000010 ...
##  $ province      : chr  "Seoul" "Seoul" "Seoul" "Seoul" ...
##  $ city          : chr  "Guro-gu" "Dongdaemun-gu" "Guro-gu" "Eunpyeong-gu" ...
##  $ group         : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ infection_case: chr  "Guro-gu Call Center" "Dongan Church" "Manmin Central Church" "Eunpyeong St. Mary's Hospital" ...
##  $ confirmed     : int  96 20 20 14 13 10 7 6 109 32 ...
##  $ latitude      : chr  "37.508163" "37.592888" "37.481059" "37.63369" ...
##  $ longitude     : chr  "126.884387" "127.056766" "126.894343" "126.9165" ...
```

```
#전처리----------
covid_case$longitude <- as.numeric(covid_case$longitude)
```

```
## Warning: 강제형변환에 의해 생성된 NA 입니다
```

```
covid_case$latitude <- as.numeric(covid_case$latitude)  ##str(covid_case)로 해당 데이터프레임의 구조를 확인해 본 결과 latitude(위도)/longitude(경도) 가 character로 설정되어 있어 차후 지도에 발생지를 표시하고자 할 때 문제가 될것이므로 ,밑의 as.numeric 과정으로 숫자형 데이터로 변환 시켜줍니다.
```

```
## Warning: 강제형변환에 의해 생성된 NA 입니다
```

```
str(covid_case) #위도,경도가 numeric data로 변환된 것을 볼 수 있습니다.
```

```
## 'data.frame':    102 obs. of  8 variables:
##  $ case_id       : int  1000001 1000002 1000003 1000004 1000005 1000006 1000007 1000008 1000
009 1000010 ...
##  $ province      : chr  "Seoul" "Seoul" "Seoul" "Seoul" ...
##  $ city          : chr  "Guro-gu" "Dongdaemun-gu" "Guro-gu" "Eunpyeong-gu" ...
##  $ group         : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ infection_case: chr  "Guro-gu Call Center" "Dongan Church" "Manmin Central Church" "Eunpy
eong St. Mary's Hospital" ...
##  $ confirmed     : int  96 20 20 14 13 10 7 6 109 32 ...
##  $ latitude      : num  37.5 37.6 37.5 37.6 37.6 ...
##  $ longitude     : num  127 127 127 127 127 ...
```

```
names(covid_case)[c(5,7,8)] <- c("name","lat","long") #편의를 위해 발생지 , 위도 , 경도의 이름을 바꿔줍니다
str(covid_case)
```

```
## 'data.frame':    102 obs. of  8 variables:
##  $ case_id  : int  1000001 1000002 1000003 1000004 1000005 1000006 1000007 1000008 1000009 1
000010 ...
##  $ province : chr  "Seoul" "Seoul" "Seoul" "Seoul" ...
##  $ city     : chr  "Guro-gu" "Dongdaemun-gu" "Guro-gu" "Eunpyeong-gu" ...
##  $ group    : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
##  $ name     : chr  "Guro-gu Call Center" "Dongan Church" "Manmin Central Church" "Eunpyeong
St. Mary's Hospital" ...
##  $ confirmed: int  96 20 20 14 13 10 7 6 109 32 ...
##  $ lat      : num  37.5 37.6 37.5 37.6 37.6 ...
##  $ long     : num  127 127 127 127 127 ...
```

```
covid_case
```

case_id <int>	province <chr>	city <chr>	gr… <lgl>	name <chr>
1000001	Seoul	Guro-gu	TRUE	Guro-gu Call Center
1000002	Seoul	Dongdaemun-gu	TRUE	Dongan Church
1000003	Seoul	Guro-gu	TRUE	Manmin Central Church
1000004	Seoul	Eunpyeong-gu	TRUE	Eunpyeong St. Mary's Hospital
1000005	Seoul	Seongdong-gu	TRUE	Seongdong-gu APT
1000006	Seoul	Jongno-gu	TRUE	Jongno Community Center
1000007	Seoul	Jung-gu	TRUE	Jung-gu Fashion Company
1000008	Seoul	from other city	TRUE	Shincheonji Church
1000009	Seoul	-	FALSE	overseas inflow
1000010	Seoul	-	FALSE	contact with patient

1-10 of 102 rows | 1-6 of 8 columns Previous **1** 2 3 4 5 6 … 11 Next

```
#해당 데이터를 살펴볼 때 위,경도가 NA값으로 들어간 데이터들이 존재하는데 이는 기존에 "-" 라고 Char Type으로 저장되어있던 data가 numeric형으로 변환되며 강제로 NA값을 부여받은 것이다. 몇개의 예외를 빼고 대부분의 경우 infection_case를 살펴 볼 때 위/경도가 존재하는 다른데이터들과 달리 특정 발생지를 말하는 것이 아닌 일반화 된 케이스를 말하고 있는 것이기 때문에 의도적으로(또는 몇몇경우 알아내지못해) 위/경도를 입력하지 않은 케이스로 보입니다.
#위도/경도에 city값을 기준으로 평균을 내서 작은 등차값을 주어 NA에 할당 할 수도 있겠으나, 해당 마킹 지도의 목적이 사용자에게 코로나 발생지를 알림으로써 주의를 주기위함임이라고 고려할 때 앞에 언급한 전처리는 불필요한데 더해 해석에 혼란을 야기할 수 있다고 생각하여 하지 않겠습니다. 더불어 위도/경도를 0으로 처리한다면 지도상 Mark가 지역의 도메인인 대한민국을 벗어나므로 이또한 하지 않겠습니다.


#-----------------

#해당 데이터셋에 케이스가 102개밖에 되지 않고 그마저도 위,경도가 포함된 데이터가 얼마 되지 않으므로 따로 서브셋시키지 않겠습니다.
m <- leaflet() %>%  #파이프함수를 사용하여 leaflet,addTiles,addMarkers 함수를 한번에 실행시켜줍니다.
  addTiles() %>%
  addMarkers(lng=covid_case$long, lat=covid_case$lat, popup=covid_case$name,)
```

```
## Warning in validateCoords(lng, lat, funcName): Data contains 71 rows with either
## missing or invalid lat/lon values and will be ignored
```

```
m
```