

14조_2차발표자료

```
# 패키지 설치
#install.packages("ggplot2")
#install.packages("readr")
#install.packages("RColorBrewer")
#install.packages("data.table")
```

```
library(ggplot2)
library(readr)
library(RColorBrewer)
library(data.table)
```

< 프랑스 이민자 현황 파악 및 분석 >

1. 대륙별 이민자 수 추이

1.1. 2019년 이민자 데이터

(1) 데이터 불러오기

참고) 2019년 이민자의 수는 272,400명

```
immigrant2019 <- read.csv("demo-etran-origine-geo-immig2019.csv", encoding = "UTF-8")
```

(2) 데이터 전처리

데이터의 열이름 정리 및 필요없는 행과 열 삭제하여 아프리카, 유럽, 아시아, 아메리카, 오세아니아 다섯 대륙의 정보만 추출하였다.

```
names(immigrant2019) <- c(immigrant2019[2,1],immigrant2019[3,2],immigrant2019[3,3])
immigrant2019 <- immigrant2019[c(-1,-2,-3),]
```

```
#
# 대륙별 정보만 추출
immi19 <- immigrant2019[, c(1:3)]
immi19 <- immigrant2019[c(1, 13, 25, 37), c(1:3) ]

names(immi19) <- c("continent", "immigrant", "proportions")
```

- immi19 데이터 프레임 확인

	continent	immigrant	proportions
4	Afrique	111,700	41
16	Europe	87,000	31.9
28	Asie	44,000	16.2
40	Amerique, Oceanie	29,700	10.9

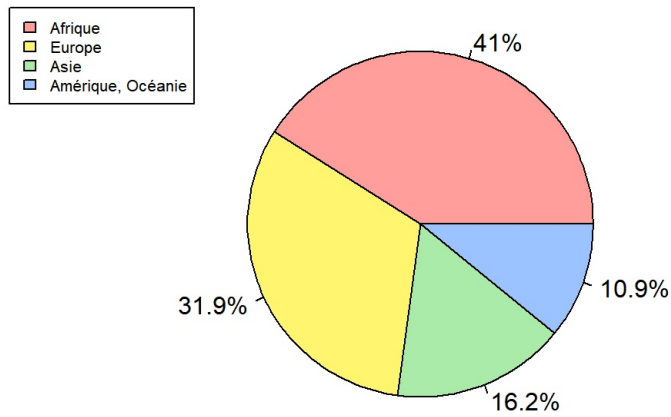
(3) 데이터 시각화하기

pie chart를 이용하여 2019년 출신대륙별 이민자의 비율 시각화하였다. 2019년도의 경우 아프리카 대륙 출신의 이민자(41%)가 가장 많았으며, 아프리카 대륙다음으로는 유럽대륙(31.9%)이 그 다음으로 많았다.

```
col1 <-c("#FF9E9B", "#FF56E", "#AAEBAA", "#9BC3FF")
immi19$proportions<- as.numeric(immi19$proportions) # 숫자형 변환
i19_labels <- paste(immi19$proportions, "%", sep="") # 퍼센트 추가

pie(immi19$proportions, labels=i19_labels, col=col1,
    main="<2019 proportion d'immigrant>" #파이차트 생성
    legend(-1.9,1,"topleft", legend=immi19$continent, # 범례추가
    fill = col1, cex= 0.65)
```

<2019 proportion d'immigrant>



1.2. 2020년 이민자 데이터

(1) 데이터 불러오기

참고) 2020년 이민자의 수는 215,200명

```
immigrant2020 <- read.csv("Donnees_complementaires2020.csv", encoding = "UTF-8", header = T)
```

(2) 데이터 전처리

위 과정은 2019년 데이터 정제방법과 동일하다

```
names(immigrant2020) <- c(immigrant2020[2,1],immigrant2020[3,2],immigrant2020[3,3])
immigrant2020 <- immigrant2020[c(-1,-2,-3),c(1,2,3)]
```

```
immi20 <- immigrant2020[, c(1:3)]
immi20 <- immigrant2020[c(1, 13, 25, 37), c(1:3) ]
names(immi20) <- c("continent", "immigrant", "proportions")
```

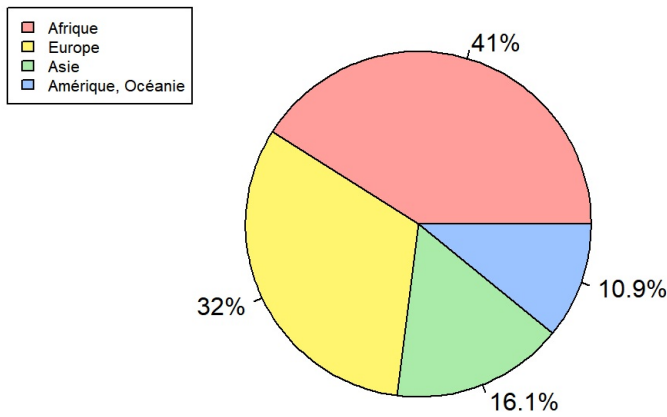
(3) 데이터 시각화하기

pie chart를 이용하여 시각화를 진행하였다. 그 결과 앞서 2019년과 마찬가지로 2020년도에도 아프리카 대륙 출신 이민자가 가장 많았으며 유럽대륙 출신의 이민자가 그 뒤를 이었다.

```
immi20$proportions<- as.numeric(immi20$proportions) # 숫자형 변환
i20_labels <- paste(immi20$proportions, "%",sep="") # 퍼센트 추가

pie(immi20$proportions, labels=i20_labels, col=col1, main="<2020 proportion d'immigrant>") #파이차트 생성
legend(-1.9,1,"topleft", legend=immi20$continent, # 범례추가
      fill = col1, cex= 0.65)
```

<2020 proportion d'immigrant>



1.3. 2021년 이민자 데이터 읽어오기

(1) 2021년 데이터 불러오기

참고) 2021년 이민자의 수는 6964(en milliers)명

(2) 데이터 전처리

```
# 데이터의 열이름 정리 및 필요없는 행열 삭제
immi21 <- c(immi21[c(3,55,98,145),c(1,2,3)])
names(immi21) <- c("continent","2018","2017")

immi21<-as.data.frame(immi21)

for(i in 2:3){
  immi21[,i] <-gsub("","",immi21[,i]) # 쉼표 삭제
  immi21[,i] <-as.numeric(immi21[,i]) # 숫자형으로 변환
}

immi21$immigrant <- immi21$X2018 - immi21$X2017
# 2018년도 총 이민자에서 2017년도 총 이민자수를 2018년도에 프랑스에 유입된 이민자 인구를 추출

immi21 <- immi21[, -c(2,3)]

immi21$proportions <- c(round(immi21$immigrant[1]/sum(immi21$immigrant)*100,1), # 비율 데이터 추가
  round(immi21$immigrant[2]/sum(immi21$immigrant)*100,1),
  round(immi21$immigrant[3]/sum(immi21$immigrant)*100,1),
  round(immi21$immigrant[4]/sum(immi21$immigrant)*100,1))
```

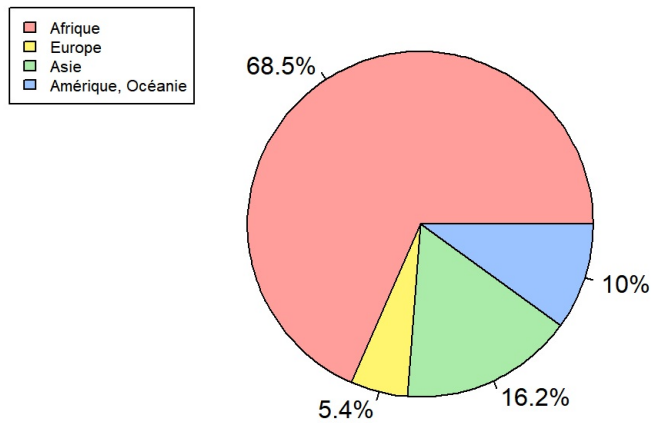
(3) 데이터 시각화하기

마지막으로 2021년에는 아프리카 대륙 출신 이민자가 가장 많았으며 그 다음으로는 유럽 대륙의 이민자가 가장 많은 것으로 확인되었다. 2021년에는 다른 연도보다 아프리카 출신의 이민자 비율이 상승했음을 확인할 수 있었다.

```
i21_labels <- paste(immi21$proportions, "%", sep=" ") # 퍼센트 추가
pie(immi21$proportions, labels=i21_labels,col=col1,
  main="2021 proportion d'immigrant") # 파이차트 생성

legend(-1.9,1,"topleft", legend=immi21$continent, # 범례추가
  fill = col1, cex= 0.65)
```

<2021 proportion d'immigrant>



1.4. 대륙별 2019-2021년 이민자 수 추이

마지막으로 대륙별 전체적인 추이를 한 눈에 알아보도록 하겠다 먼저 그래프를 그리기 위한 전처리 과정이다

(1)데이터 전처리

2019년부터 2021년의 데이터를 하나의 data frame으로 합친 후 세부 대륙별로 각각 데이터 프레임을 생성하였다.

```
immi19_21 <- rbind(immi19, immi20, immi21)
immi19_21$annee <- c("2019","2019","2019","2019",
                     "2020","2020","2020","2020",
                     "2021","2021","2021","2021")
immi19_21$immigrant <- gsub(",", "", immi19_21$immigrant)
immi19_21$immigrant <- as.numeric(immi19_21$immigrant)
immi19_21$annee <- as.numeric(immi19_21$annee)
immi19_21$continent[c(4,8,12)] <- "Amérique, Océanie" # 세부 데이터 수정
```

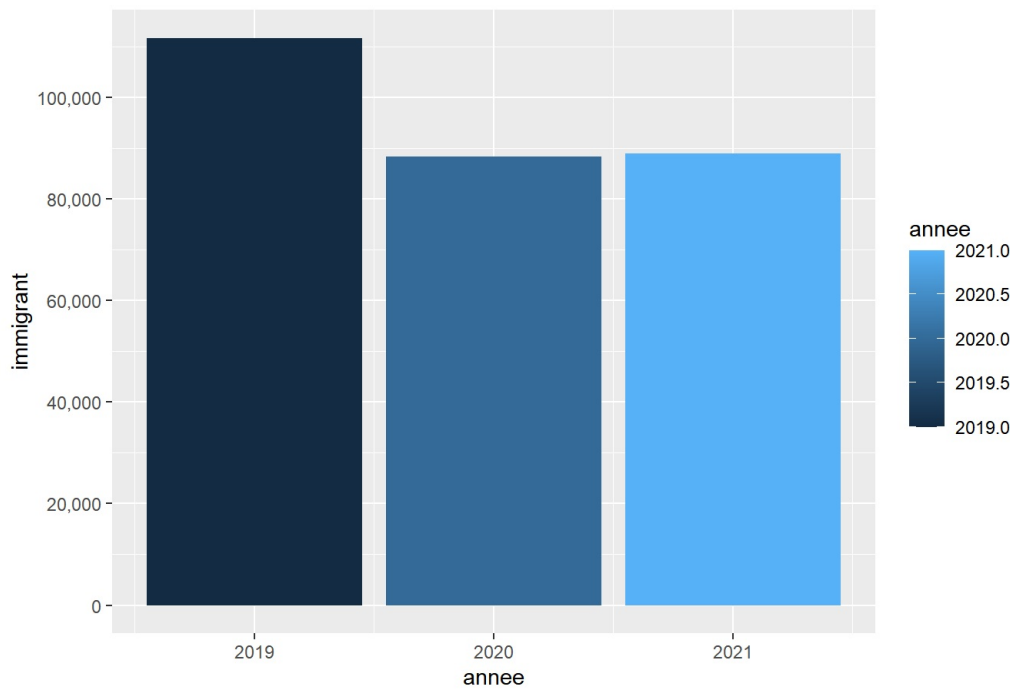
```
afri_immi <- immi19_21[c(1,5,9),] # 아프리카 대륙
Eur_immi <- immi19_21[c(2,6,10),] # 유럽 대륙
Asie_immi <- immi19_21[c(3,7,11),] # 아시아 대륙
Amer_immi <- immi19_21[c(4,8,12),] # 아메리카 오세아니아 대륙
```

(1) 아프리카 연도별 추이

그래프를 통해 확인해본 결과 3개년 중 19년도의 아프리카 대륙출신 이민자가 가장 많았으며 2021년에 전년대비 소폭 상승한것을 확인할 수 있다.

```
ggplot(afri_immi, aes(x=annee, y=immigrant, fill=annee))+
  geom_col() +
  labs(title="Nombre d'immigrés Afrique(2019-2021)") +
  scale_y_continuous(labels = scales::comma, breaks = seq(0,100000,20000))
```

Nombre d'immigres Afrique(2019-2021)

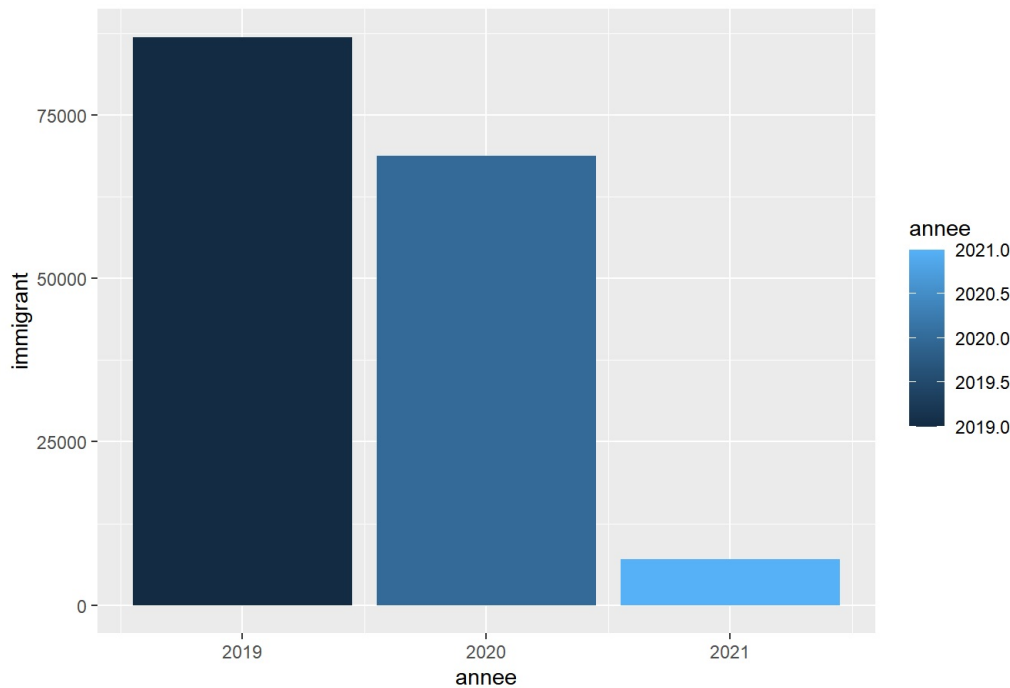


(2) 유럽 대륙의 추이

다음으로는 유럽 대륙 출신 이민자 수를 그래프를 통해 시각화해 보았다. 아프리카 대륙과 마찬가지로 19년도에 가장 많았던 이민자수는 21년 가파르게 하락하는 양상을 보였다.

```
ggplot(Eur_immi, aes(x=annee, y=immigrant, fill=annee))+
  geom_col() +
  labs(title="Nombre d'immigrés Europe(2019-2021)")
```

Nombre d'immigres Europe(2019-2021)

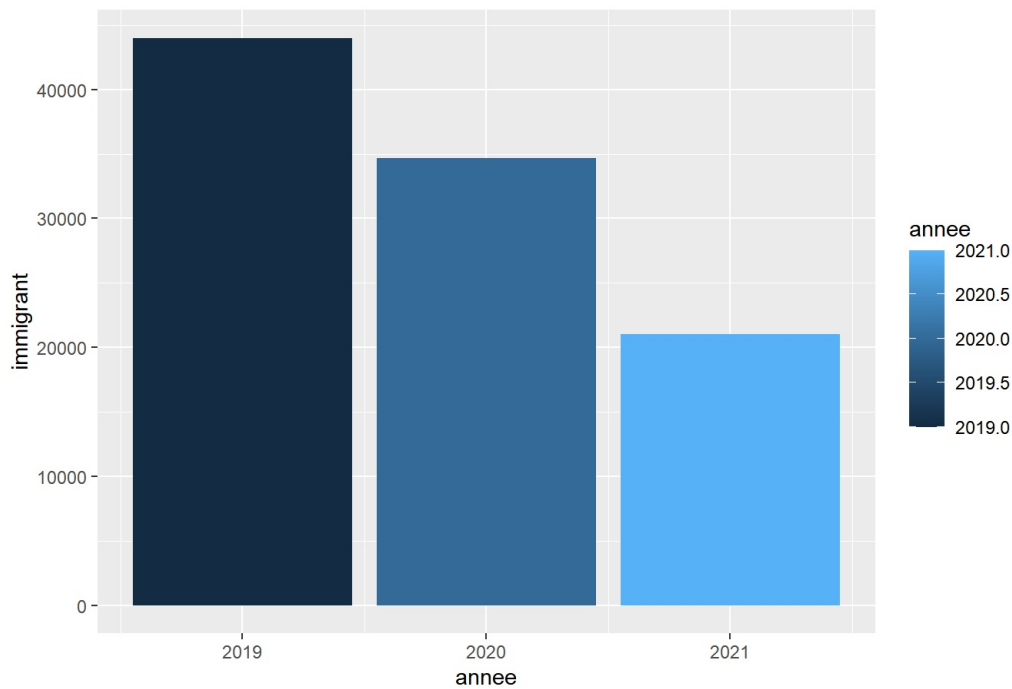


(3) 아시아 연도별 추이

아시아 대륙 또한 점차 하락하는 양상을 보였다.

```
ggplot(Asie_immi, aes(x=annee, y=immigrant, fill=annee))+
  geom_col() +
  labs(title="Nombre d'immigrés Asie(2019-2021)")
```

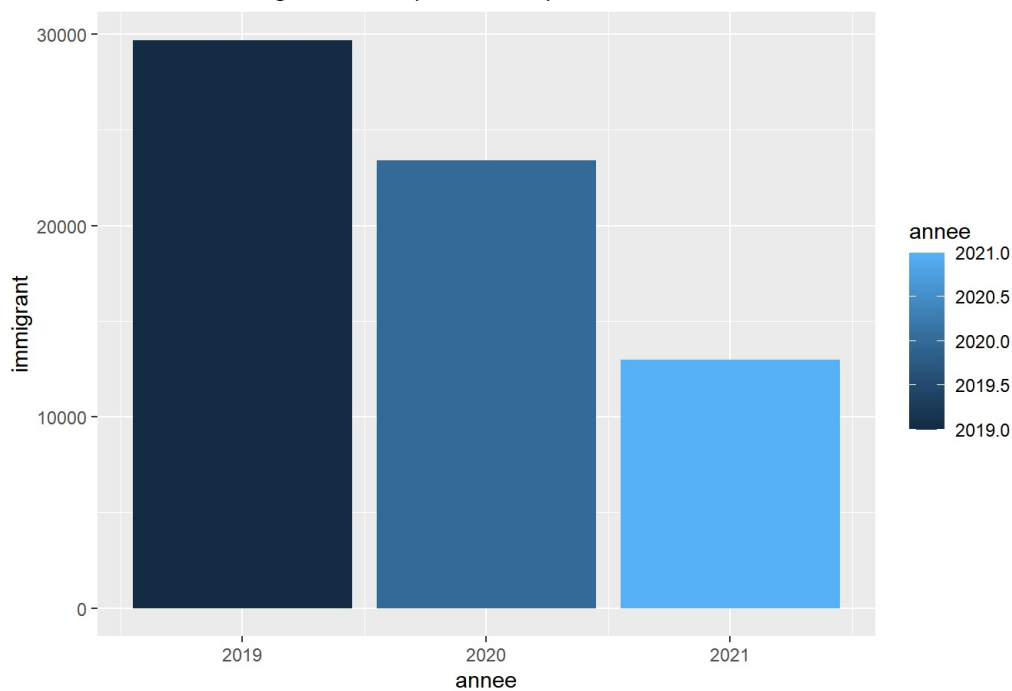
Nombre d'immigres Asie(2019-2021)



(4) 아메리카, 오세아니아 대륙 연도별 추이

```
ggplot(Amer_immi, aes(x=annee, y=immigrant, fill=annee))+
  geom_col() +
  labs(title="Nombre d'immigrés Autres(2019-2021)") +
  scale_y_continuous(breaks = seq(0,100000,10000))
```

Nombre d'immigres Autres(2019-2021)

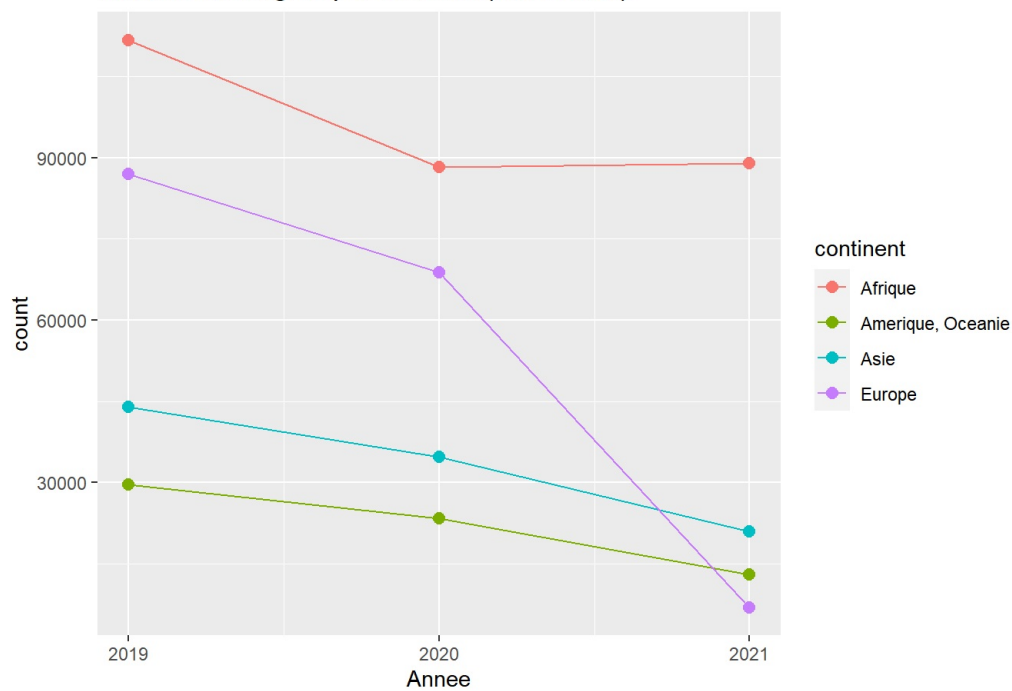


(5) 모든 대륙 비교

모든 대륙의 연도별 추이를 하나의 line chart로 시각해보았다. 그래프를 살펴본 결과 아프리카 대륙 출신을 제외하면 모든 대륙이 하락하는 양상을 띄었고 특히 유럽의 경우 그 하락세가 가파른 것으로 확인되었다.

```
ggplot(imm19_21, aes(x=annee, colour=continent))+
  geom_line(aes(y=immigrant))+
  geom_point(aes(y=immigrant), size=2.5)+
  xlab('Annee')+
  ylab('count')+
  labs(title="Nombre d'immigrés par continent(2019-2021)") +
  scale_x_continuous(breaks = seq(2019,2021,1))
```

Nombre d'immigres par continent(2019-2021)



2. 이민자와 비이민자의 비율 비교

2.1. 데이터 불러오기

```
#새로운 데이터 프레임 형성
immi_total <- read.csv("Donnees_complementaires_total.csv")
```

2.2.데이터 전처리

```
immi_total <- immi_total[,c(1,2,4,6)] #전체 인구 수, 이민자 수 추출
immi_total <- immi_total[-c(1,2,3),]

#열이름을 통해 "연도", "외국인수", "이민자수", "전체인구"로 설정
colnames(immi_total) <- c("연도", "외국인수", "이민자수", "전체인구")

# 문자형을 숫자형으로 변환
for(i in 2:4){
  immi_total[,i] <- gsub(",", "", immi_total[,i])
  immi_total[,i] <- as.numeric(immi_total[,i])
}

immi_total$연도 <- gsub("(p)", "", immi_total$연도) # 연도의 추정치 표시 제거
```

2.3.데이터 시각화

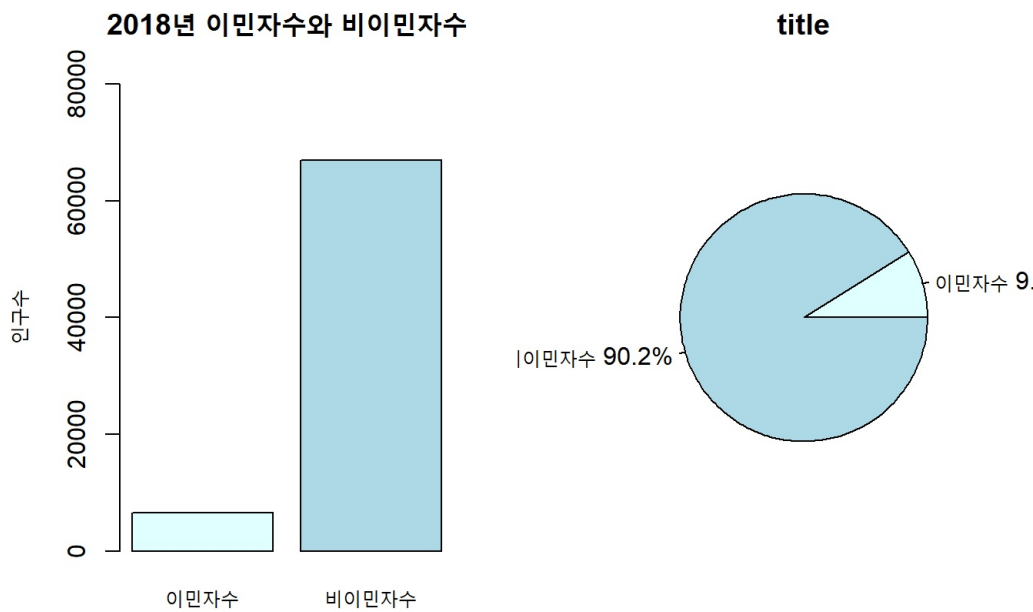
```
#2018 이민자수와 비이민자수 비교
comp_immi <-subset(immi_total,연도==2018)

unlist_df <-comp_immi[,c(3,4)]
unlist_df<-unlist(unlist_df)

opar = par(mfrow=c(1,2))
barplot(unlist_df, names=c("이민자수", "비이민자수"), ylim=c(0,80000),
        col = c("lightcyan","lightblue"))

title(main="2018년 이민자수와 비이민자수", ylab="인구수")
lab <- c("이민자수","비이민자수")
lab <- paste(lab,c(round(comp_immi$이민자수*100/comp_immi$전체인구,1),
                    round(100 - (comp_immi$비이민자수*100/comp_immi$전체인구),1)))
lab <- paste(lab, "%",sep="")

pie(unlist_df, labels=lab , main ="title",
    col = c("lightcyan","lightblue"))
```



3. 2018년 출신대륙별 이민자의 비율

3.1. 데이터 불러오기

```
#새로운 데이터 프레임 형성
#df1 <- read.csv("pays_naissance_detaille_1968_2018.txt")

library(readr)
df1 <- read_delim("pays_naissance_detaille_1968_2018.txt",
  delim = "\t", escape_double = FALSE,
  trim_ws = TRUE)
```

```
## New names:
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * `` -> ...6
## * ...
```

```
## Rows: 182 Columns: 1024
## -- Column specification -----
## Delimiter: "\t"
## chr (9): R?partition des immigr?s par pays de naissance d?taill? depuis 2...
## lgl (1010): ...15, ...16, ...17, ...18, ...19, ...20, ...21, ...22, ...23, ....
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```


3.2. 데이터 전처리

```
df1 <- df1[,1:2]
df1 <- df1[-c(1,2),]

#열이름을 통해 "pays", "population"로 설정
colnames(df1) <- c("pays", "population")
df1$population <- gsub(",", "", df1$population)
df1$population <- as.integer(df1$population)

#"continent_population"라는 데이터 프레임 형성
continent_population <- df1[c(1,53,96,143),]
colnames(continent_population) <- c("pays","population")
head(df1)
```

```
## # A tibble: 6 x 2
##   pays      population
##   <chr>      <int>
## 1 Algérie      838800
## 2 Maroc        784900
## 3 Tunisie      290900
## 4 Comores      125200
## 5 Soudan      110500
## 6 Côte d'Ivoire 101200
```

```
#character에서 integer로변환
class(continent_population$population)
```

```
## [1] "integer"
```

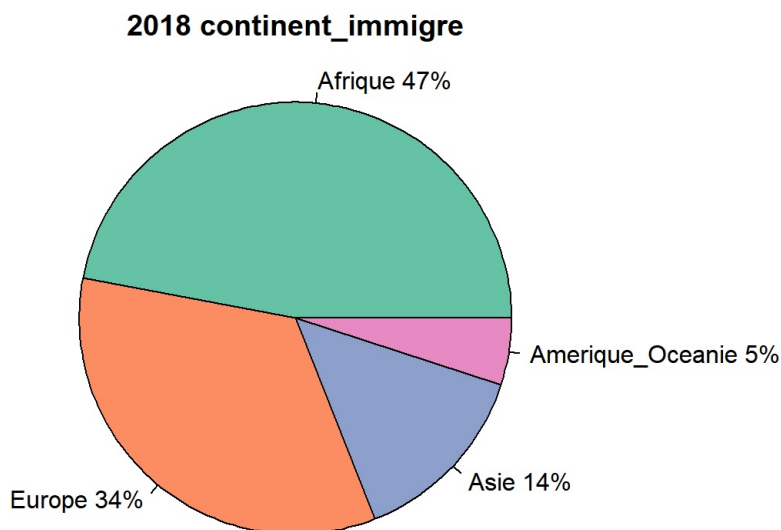
```
class(continent_population$population)
```

```
## [1] "integer"
```

```
continent_population_proportion <- round(100*prop.table(continent_population$population))
```

3.3. 데이터 시각화

```
#대륙별 파이차트 만들기(lbl을 사용해 새롭게 벡터 형태로 저장. paste0를 사용하여 "lbl"과 "대륙별 인구수 비율"을 합쳐줌.)
lbl <- c("Afrique","Europe","Asie","Amérique_Océanie")
lbl_continent_population_proportion <- paste(lbl, continent_population_proportion)
lbl_continent_population_proportion <- paste0(lbl_continent_population_proportion,"%")
pie(continent_population_proportion, labels =lbl_continent_population_proportion, col=brewer.pal(4,"Set2"), main=
"2018 continent_immigré", init.angle=0, radius=1.0)
```



```
#대륙별 변수 설정
Afrique <- df1[2:52,]
Europe <- df1[54:95,]
Asie <- df1[97:142,]
Amérique_Océanie <- df1[144:177,]
```

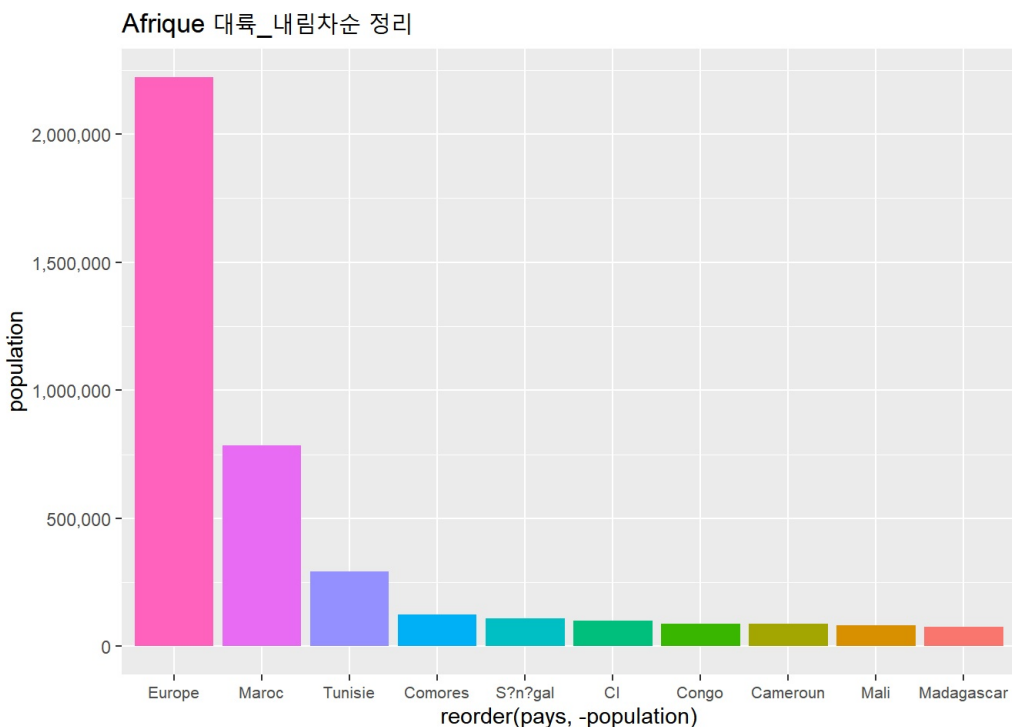
4. 이민자의 출신 국가 시각화

```
Afrique <- Afrique[order(Afrique$population, decreasing = TRUE),]
Europe <- Europe[order(Europe$population, decreasing = TRUE),]
Asie <- Asie[order(Asie$population, decreasing = TRUE),]
Amérique_Océanie <- Amérique_Océanie[order(Amérique_Océanie$population, decreasing = TRUE),]
Afrique <- Afrique[1:10,]
Europe <- Europe[1:10,]
Asie <- Asie[1:10,]
Amérique_Océanie <- Amérique_Océanie[1:10,]
```

```
#변수명 축약 (그래프 글자 겹침 현상 제거 위함. 약어 사용.)
Afrique$pays[7]="Congo"
Afrique$pays[6]="CI"
```

4.1. Afrique 대륙 출신 이민자의 출신 국가 상위 10개국

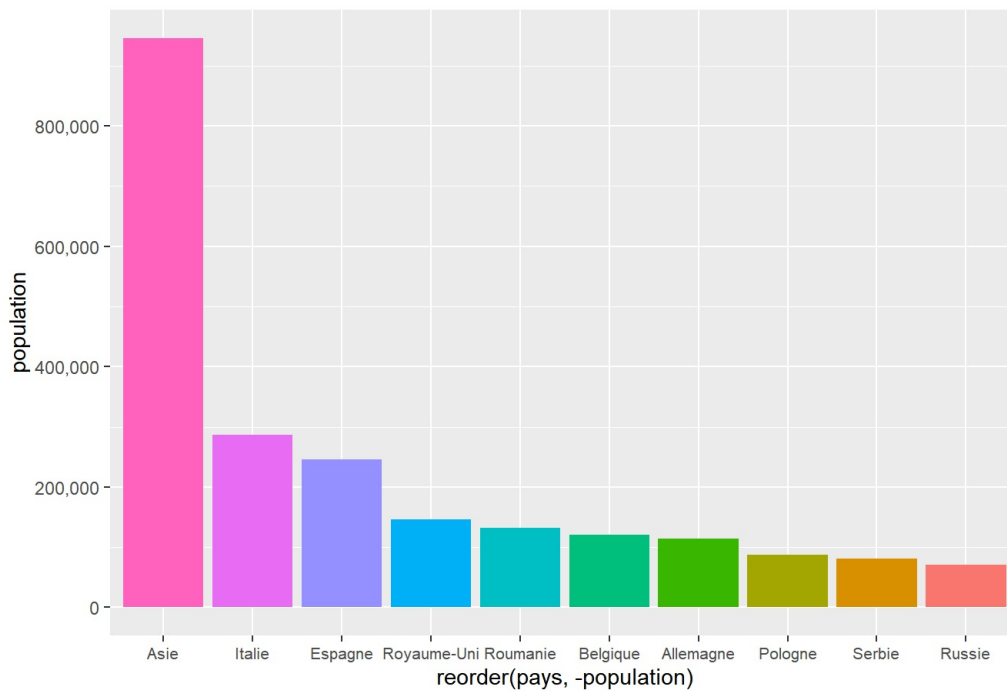
```
ggplot(Afrique, aes(x = reorder(pays, -population), y = population ,fill = factor(population)))+
  geom_bar(stat = "identity")+
  ggtitle("Afrique 대륙_내림차순 정리")+
  theme(axis.text.x = element_text(size=8))+ theme(legend.position='none')+
  scale_y_continuous(labels = scales::comma,
    breaks=seq(0, 3000000, 500000))
```



4.2. Europe 대륙 출신 이민자의 출신 국가 상위 10개국

```
ggplot(Europe, aes(x = reorder(pays, -population), y = population ,fill = factor(population)))+
  geom_bar(stat = "identity")+
  ggtitle("Europe 대륙_내림차순 정리")+
  theme(axis.text.x = element_text(size=8))+ theme(legend.position='none') +
  scale_y_continuous(labels = scales::comma,
    breaks=seq(0, 1000000, 200000))
```

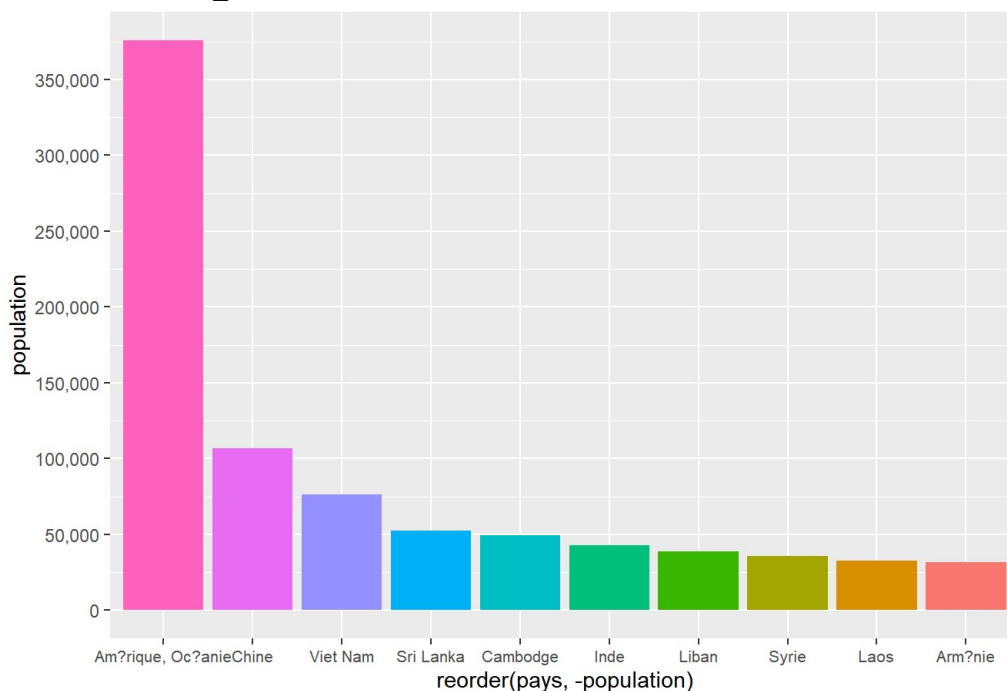
Europe 대륙_내림차순 정리



4.3. Asie 대륙 출신 이민자의 출신 국가 상위 10개국

```
ggplot(Asie, aes(x = reorder(pays, -population), y = population ,fill = factor(population)))+
  geom_bar(stat = "identity")+
  ggtitle("Asie 대륙_내림차순 정리")+
  theme(axis.text.x = element_text(size=8))+
  theme(legend.position='none') +
  scale_y_continuous(labels = scales::comma,
    breaks=seq(0, 4000000, 500000))
```

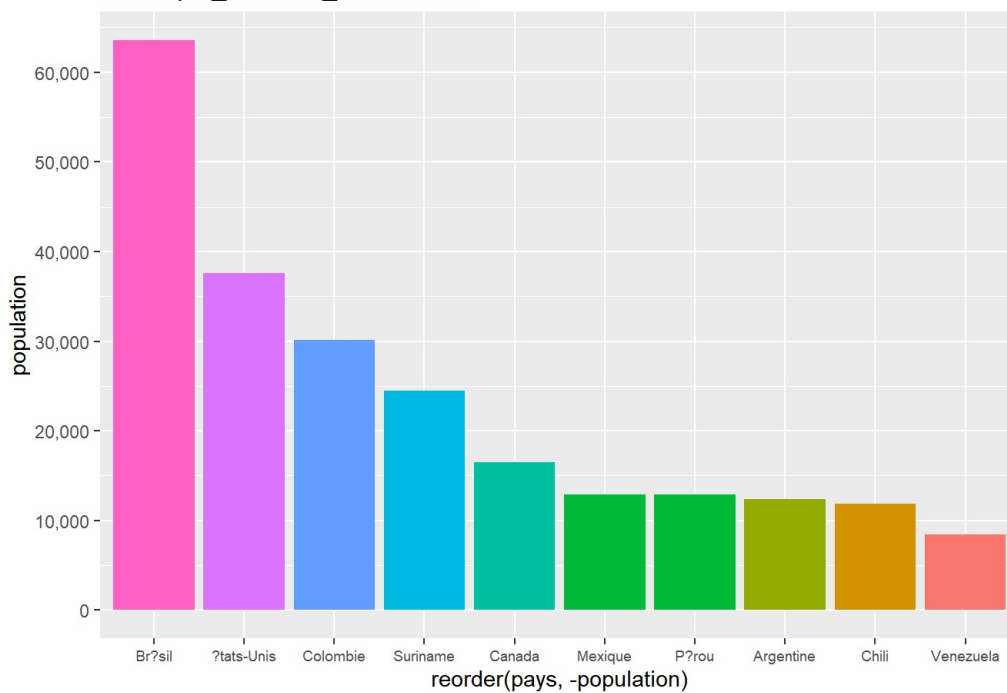
Asie 대륙_내림차순 정리



4.4. Amérique,Océanie 대륙 출신 이민자의 출신 국가 상위 10개국

```
ggplot(Am?rique_Oc?anie, aes(x = reorder(pays, -population), y = population ,fill = factor(population)))+
  geom_bar(stat="identity")+
  ggtitle("Amérique_Océanie 내림차순 정리")+
  theme(axis.text.x = element_text(size=7))+ theme(legend.position='none')+
  scale_y_continuous(labels = scales::comma,
    breaks=seq(0, 2500000, 100000))
```

Amerique_Oceanie_내림차순 정리



5. 프랑스 내 이민자의 거주지역

5.1. Paris, Marseilles, Lyon 3개 도시 거주 이민자의 출신 대륙

```
library(readr)
df_dep <- read_delim("TCRD_012.txt", delim = "\t",
  escape_double = FALSE, trim_ws = TRUE)
```

```
## New names:
## * `` -> ...2
## * `` -> ...3
## * `` -> ...4
## * `` -> ...5
## * `` -> ...6
## * ...
```

```
## Rows: 100 Columns: 9
## -- Column specification -----
## Delimiter: "\t"
## chr (9): Population immigr?e selon les principaux pays de naissance en 2018 ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_dep = df_dep[4:100,]
labels = c("Europe", "Algerie", "Maroc", "Tunisie", "Turquie")
```

(1) Paris의 이민자 출신 국가(%) 파이차트

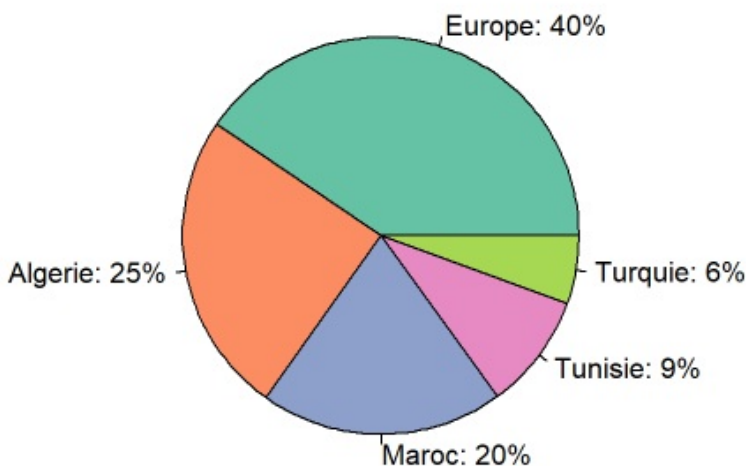
```

Paris = df_dep[76,]
Seine_et_Marne = df_dep[78,]
Yvelines = df_dep[79,]
Paris_deps = df_dep[92:96,]
Paris_region = data.frame(rbind(Paris, Seine_et_Marne, Yvelines, Paris_deps))
Paris_region_Europe_avg = round(mean(x=as.numeric(gsub(",", ".", Paris_region$X.3)), na.rm = TRUE),2)
Paris_region_Algerie_avg = round(mean(x=as.numeric(gsub(",", ".", Paris_region$X.4)), na.rm = TRUE),2)
Paris_region_Maroc_avg = round(mean(x=as.numeric(gsub(",", ".", Paris_region$X.5)), na.rm = TRUE),2)
Paris_region_Tunisie_avg = round(mean(x=as.numeric(gsub(",", ".", Paris_region$X.6)), na.rm = TRUE),2)
Paris_region_Turquie_avg = round(mean(x=as.numeric(gsub(",", ".", Paris_region$X.7)), na.rm = TRUE),2)
Paris_region_perc = c(Paris_region_Europe_avg, Paris_region_Algerie_avg, Paris_region_Maroc_avg, Paris_region_Tunisie_avg, Paris_region_Turquie_avg)
Paris_region_sum = sum(Paris_region_perc)
Paris_region_Europe_avg = round(Paris_region_Europe_avg/Paris_region_sum,2)*100
Paris_region_Algerie_avg = round(Paris_region_Algerie_avg/Paris_region_sum,2)*100
Paris_region_Maroc_avg = round(Paris_region_Maroc_avg/Paris_region_sum,2)*100
Paris_region_Tunisie_avg = round(Paris_region_Tunisie_avg/Paris_region_sum,2)*100
Paris_region_Turquie_avg = round(Paris_region_Turquie_avg/Paris_region_sum,2)*100
Paris_region_perc_2 = c(Paris_region_Europe_avg, Paris_region_Algerie_avg, Paris_region_Maroc_avg, Paris_region_Tunisie_avg, Paris_region_Turquie_avg)
Paris_region_labels = c(paste(c("Europe: ", Paris_region_Europe_avg, "%"),collapse = ""),
                        paste(c("Algerie: ", Paris_region_Algerie_avg, "%"),collapse = ""),
                        paste(c("Maroc: ", Paris_region_Maroc_avg, "%"),collapse = ""),
                        paste(c("Tunisie: ", Paris_region_Tunisie_avg, "%"),collapse = ""),
                        paste(c("Turquie: ", Paris_region_Turquie_avg, "%"),collapse = ""))

#pie(Paris_region_perc, Paris_region_labels, main="Paris의 이민자 출신 국가(%)", #col=brewer.pal(5,"Set2"))

```

Paris의 이민자 출신 국가(%)



paris지역의 이민자의 출신 국가(%)

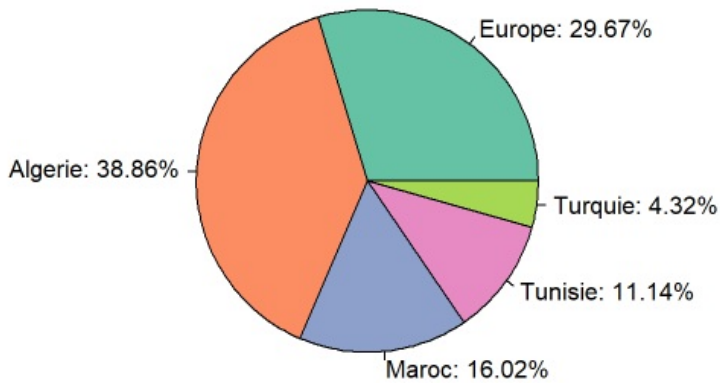
(2) Marseilles의 이민자 출신 국가(%) 파이차트

```

Marseilles = df_dep[13,]
Marseilles_perc = as.numeric(gsub(",", ".", Marseilles[,5:9]))
Marseilles_sum = sum(Marseilles_perc)
Marseilles_Europe_perc = round(Marseilles_perc[1]/Marseilles_sum*100, 2)
Marseilles_Algerie_perc = round(Marseilles_perc[2]/Marseilles_sum*100, 2)
Marseilles_Maroc_perc = round(Marseilles_perc[3]/Marseilles_sum*100, 2)
Marseilles_Tunisie_perc = round(Marseilles_perc[4]/Marseilles_sum*100, 2)
Marseilles_Turquie_perc = round(Marseilles_perc[5]/Marseilles_sum*100, 2)
Marseilles_perc_2 = c(Marseilles_Europe_perc, Marseilles_Algerie_perc, Marseilles_Maroc_perc, Marseilles_Tunisie_perc, Marseilles_Turquie_perc)
Marseilles_labels = c(paste(c("Europe: ", Marseilles_Europe_perc, "%"),collapse = ""),
                    paste(c("Algerie: ", Marseilles_Algerie_perc, "%"),collapse = ""),
                    paste(c("Maroc: ", Marseilles_Maroc_perc, "%"),collapse = ""),
                    paste(c("Tunisie: ", Marseilles_Tunisie_perc, "%"),collapse = ""),
                    paste(c("Turquie: ", Marseilles_Turquie_perc, "%"),collapse = ""))
#pie(Marseilles_perc_2, Marseilles_labels, main="Marseilles의 이민자 출신 국가(%)", col=brewer.pal(5,"Set2"))

```

Marseilles의 이민자 출신 국가(%)



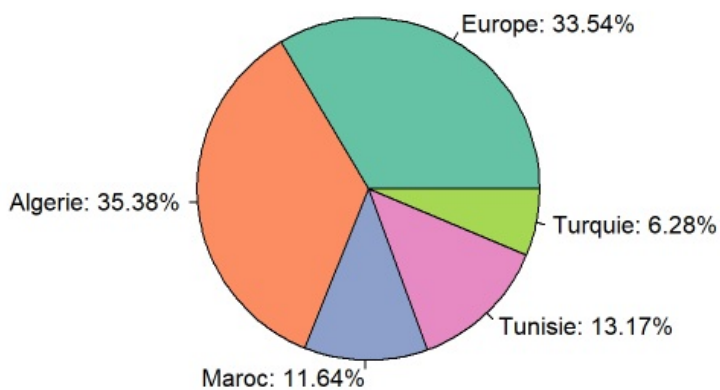
Marseilles지역의 이민자의 출신 국가(%)

(3) Lyon의 이민자 출신 국가(%) 파이차트

```

Lyon = df_dep[70,]
Lyon_perc = as.numeric(gsub(",", ".", Lyon[,5:9]))
Lyon_sum = sum(Lyon_perc)
Lyon_Europe_perc = round(Lyon_perc[1]/Lyon_sum*100, 2)
Lyon_Algerie_perc = round(Lyon_perc[2]/Lyon_sum*100, 2)
Lyon_Maroc_perc = round(Lyon_perc[3]/Lyon_sum*100, 2)
Lyon_Tunisie_perc = round(Lyon_perc[4]/Lyon_sum*100, 2)
Lyon_Turquie_perc = round(Lyon_perc[5]/Lyon_sum*100, 2)
Lyon_perc_2 = c(Lyon_Europe_perc, Lyon_Algerie_perc, Lyon_Maroc_perc, Lyon_Tunisie_perc, Lyon_Turquie_perc)
Lyon_labels = c(paste(c("Europe: ", Lyon_Europe_perc, "%"),collapse = ""),
                paste(c("Algerie: ", Lyon_Algerie_perc, "%"),collapse = ""),
                paste(c("Maroc: ", Lyon_Maroc_perc, "%"),collapse = ""),
                paste(c("Tunisie: ", Lyon_Tunisie_perc, "%"),collapse = ""),
                paste(c("Turquie: ", Lyon_Turquie_perc, "%"),collapse = ""))
#pie(Lyon_perc_2, Lyon_labels, main="Lyon의 이민자 출신 국가(%)", col=brewer.pal(5,"Set2"))
    
```

Lyon의 이민자 출신 국가(%)

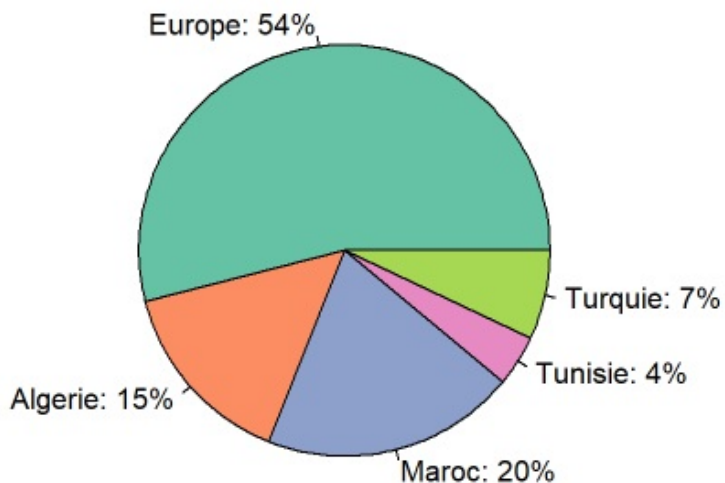


Lyon지역의 이민자의 출신 국가(%)

(4) 기타 지역의 이민자 출신 국가(%) 파이차트

```
France = data.frame(rbind(df_dep[1:12, 1:9], df_dep[14:69, 1:9], df_dep[71:75, 1:9],df_dep[77, 1:9],df_dep[80:91, 1:9]))
France_Europe_avg = round(mean(x=as.numeric(gsub(",", ".", France$X.3)), na.rm = TRUE),2)
France_Algerie_avg = round(mean(x=as.numeric(gsub(",", ".", France$X.4)), na.rm = TRUE),2)
France_Maroc_avg = round(mean(x=as.numeric(gsub(",", ".", France$X.5)), na.rm = TRUE),2)
France_Tunisie_avg = round(mean(x=as.numeric(gsub(",", ".", France$X.6)), na.rm = TRUE),2)
France_Turquie_avg = round(mean(x=as.numeric(gsub(",", ".", France$X.7)), na.rm = TRUE),2)
France_perc = c(France_Europe_avg, France_Algerie_avg, France_Maroc_avg, France_Tunisie_avg, France_Turquie_avg)
France_sum = sum(France_perc)
France_Europe_avg = round(France_Europe_avg/France_sum,2)*100
France_Algerie_avg = round(France_Algerie_avg/France_sum,2)*100
France_Maroc_avg = round(France_Maroc_avg/France_sum,2)*100
France_Tunisie_avg = round(France_Tunisie_avg/France_sum,2)*100
France_Turquie_avg = round(France_Turquie_avg/France_sum,2)*100
France_perc_2 = c(France_Europe_avg, France_Algerie_avg, France_Maroc_avg, France_Tunisie_avg, France_Turquie_avg)
France_labels = c(paste(c("Europe: ", France_Europe_avg, "%"),collapse = ""),
  paste(c("Algerie: ", France_Algerie_avg, "%"),collapse = ""),
  paste(c("Maroc: ", France_Maroc_avg, "%"),collapse = ""),
  paste(c("Tunisie: ", France_Tunisie_avg, "%"),collapse = ""),
  paste(c("Turquie: ", France_Turquie_avg, "%"),collapse = ""))
#pie(France_perc_2, France_labels, main="기타 지역의 이민자 출신 국가(%)", col=brewer.pal(5,"Set2"))
```

기타 지역의 이민자 출신 국가(%)



기타 지역의 이민자의 출신 국가(%)

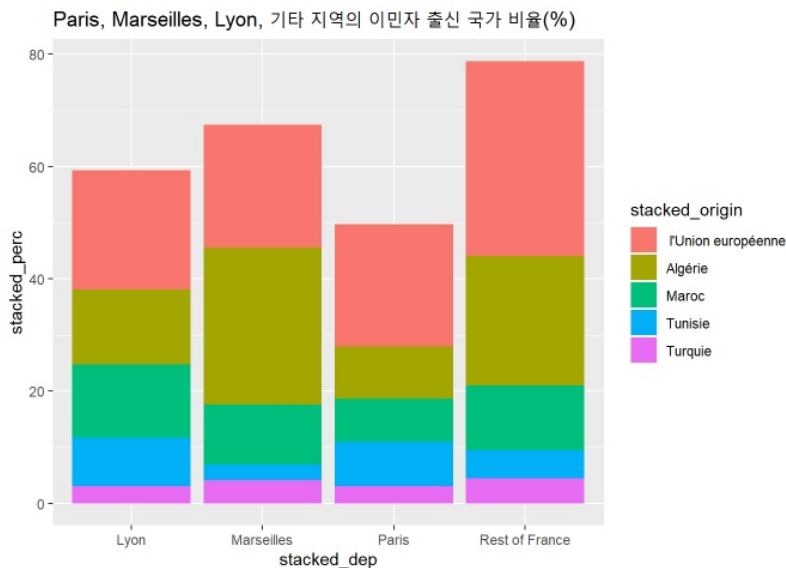
(5) 거주지지역의 이민자 출신 국가 비율

```
stacked_dep = c("Paris", "Lyon", "Marseilles", "Rest of France")
stacked_origin = rep(c(" l'Union européenne", "Algérie", "Maroc", "Tunisie", "Turquie"), 4)
#stacked_perc = c(Paris_region_perc, Marseilles_perc, Lyon_perc, France_perc)

#avg = (Paris_region_perc + Lyon_perc + Marseilles_perc)/3

#stacked_data = data.frame(stacked_dep, stacked_origin, stacked_perc)

#ggplot(stacked_data, aes(fill=stacked_origin, y=stacked_perc, x=stacked_dep)) +
  #geom_bar(position="stack", stat="identity") +
  #ggtitle("Paris, Marseilles, Lyon, 기타 지역의 이민자 출신 국가 비율(%))")
```



이민자 출신 국가 비율(%)_누적막대그래프

(6) 3개도시와 그외지역의 거주 이민자수 비교

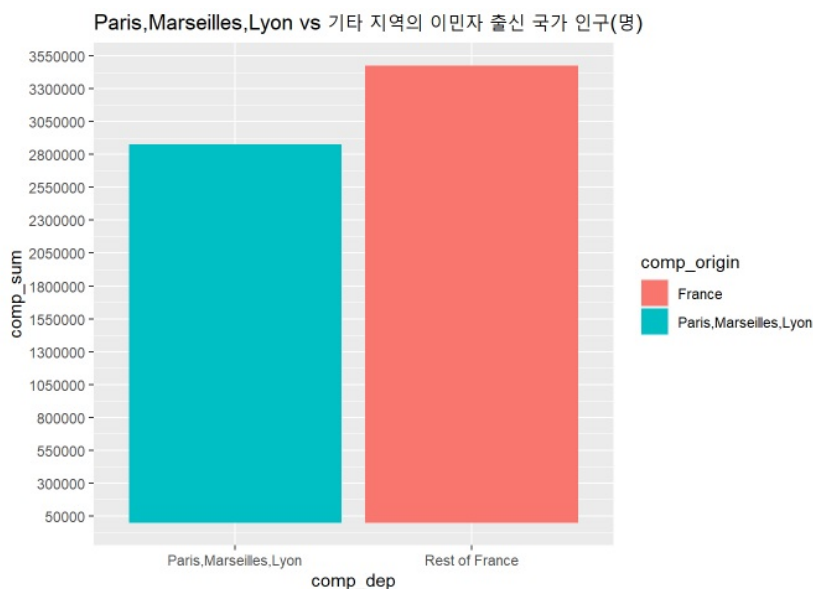
```
#comp_Lyon = as.numeric(Lyon$X.1)*1000
#comp_Marseilles = as.numeric(Marseilles$X.1)*1000
#comp_Paris_region = sum(as.numeric(Paris_region$X.1))*1000

#comp_France = sum(as.numeric(France$X.1))*1000

#comp_sum = c(comp_Lyon + comp_Marseilles + comp_Paris_region, comp_France)

#comp_dep = c("Paris,Marseilles,Lyon", "Rest of France")
#comp_origin = c("Paris,Marseilles,Lyon", "France")
#comp_data = data.frame(comp_dep, comp_origin, comp_sum)

#ggplot(comp_data, aes(fill=comp_origin, y=comp_sum, x=comp_dep)) +
#  #geom_bar(position="dodge", stat="identity") +
#  #scale_y_continuous(breaks=seq(50000, 500000000, 250000))+
#  #ggtitle("Paris,Marseilles,Lyon vs 기타 지역의 이민자 출신 국가 인구(명)")
```



3개 도시와 기타지역의 이민자 인구수 비교

6. 이민자와 비이민자의 직업 분포 비교

이민자의 노동시장 현황에 관련한 통계분석에 사용할 데이터는 2018년도의 데이터이다. 때문에 2018년 프랑스의 이민자와 비이민자가 주로 종사하였던 직종이 어떤 것인지 현황을 파악해보고 시각화해보고자한다.

6.1. 2018년도 데이터 불러오기

```
domaines2018 <- read.csv("immigres_secteurs_metiers_2018.csv", encoding = "UTF-8", header = T)
```

6.2. 데이터 정제하기

(1) 비이민자의 상위 5개 직업 추출 및 정제

먼저 비이민자가 주로 종사하는 직업 상위 5부문을 추출하여 데이터를 정제하였다.

```
idp <- domaines2018[, c(2:4)]
idp <- domaines2018[c(4, 9, 18, 22, 29, 33, 40, 44, 46, 53, 55, 63, 67, 69, 75, 79, 85, 90, 98, 101, 108, 111), c(2:4)]

names(idp) <- c("domaine", "nonimmigrés","immigrés")

# 상위 5개 직업 분야 추출
NI2018 <- idp[,c(1,2)]
NI2018 <- idp[c(11,14,16,18,20), c(1,2)]

# 자료 타입 변환
NI2018$nonimmigres <- gsub(",","",NI2018$nonimmigres) # 숫자 쉼표 제거
NI2018$nonimmigres <- as.numeric(NI2018$nonimmigres) # 숫자형 자료로 변환

# 이름 간소화
domaine1 <- c("Gestion","Administration","Commerce","Services","Santé")
NI2018$domaine <- domaine1
```

(2) 이민자의 상위 5개 직업 추출 및 정제

다음으로는 이민자가 주로 종사하는 직종 상위 5개 부문을 추출하여 데이터를 정제하였다.

```
# 상위 5개 직업 분야 추출
I2018 <- idp[,c(1,3)]
I2018 <- idp[c(2,9,11,16,18),c(1,3)]

I2018$immigres <- gsub(",","",I2018$immigres) # 숫자 쉼표 제거
I2018$immigres <- as.numeric(I2018$immigres) # 숫자형 자료로 변환
str(I2018)
```

```
## 'data.frame':   5 obs. of  2 variables:
## $ domaine : chr  "B : Batiment, travaux publics" "J : Transports, logistique et tourisme" "L : Gestion, administration des entreprises" "R : Commerce" ...
## $ immigres: num  333000 226100 234000 240900 486600
```

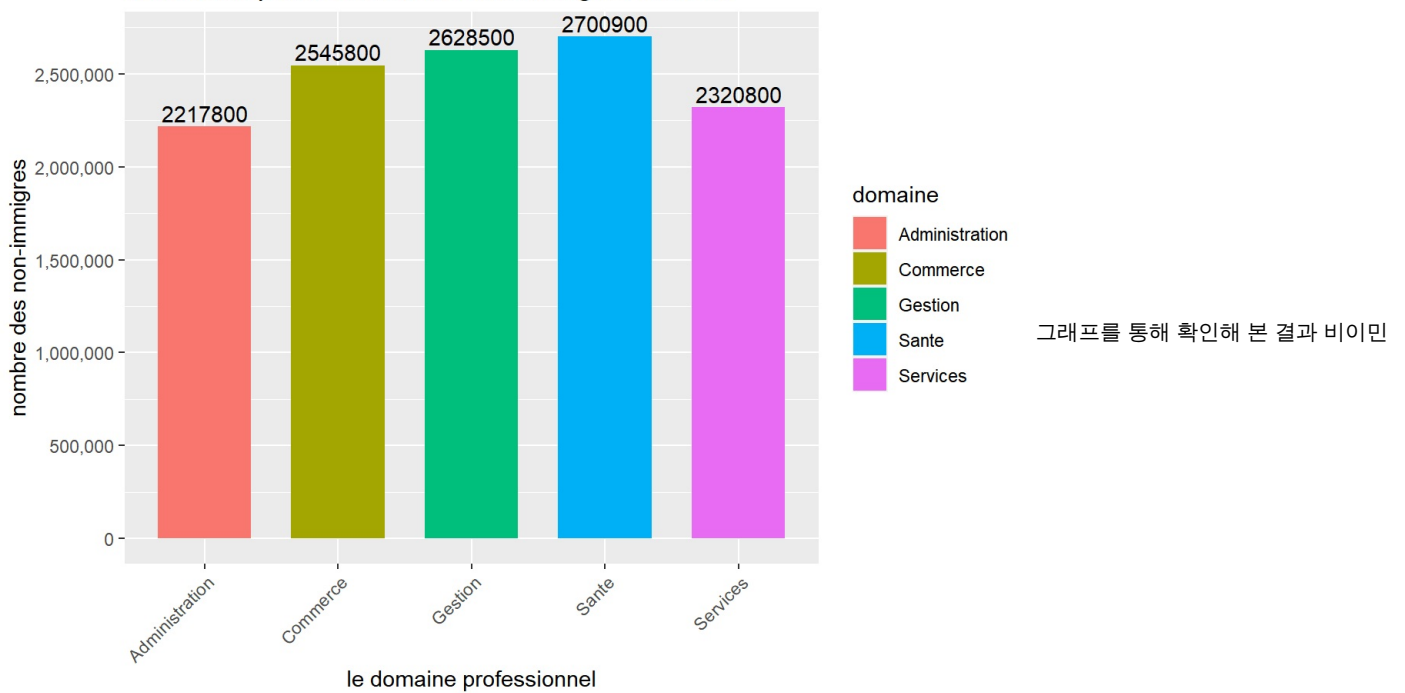
```
# 이름 간소화
domaine2 <- c("Bâtiment","Transports","Gestion","Commerce","Services")
I2018$domaine <- domaine2
```

(3) 시각화 하기

(3)-1. 비이민자 상위 5개 직업 분야

```
ggplot(NI2018, aes(x=domaine, y=nonimmigres, fill=domaine))+
  geom_bar(width = 0.7, stat="identity" )+
  labs (x="le domaine professionnel",
        y="nombre des non-immigrés",
        title="le domaine professionnel des non-immigrés en 2018")+
  theme(axis.text.x = element_text(angle=45, hjust=1, size=8.5))+
  geom_text(aes(label=nonimmigrés), vjust=-0.3) +
  scale_y_continuous(labels = scales::comma, breaks = seq(0,3000000,500000))
```

le domaine professionnel des non-immigrés en 2018



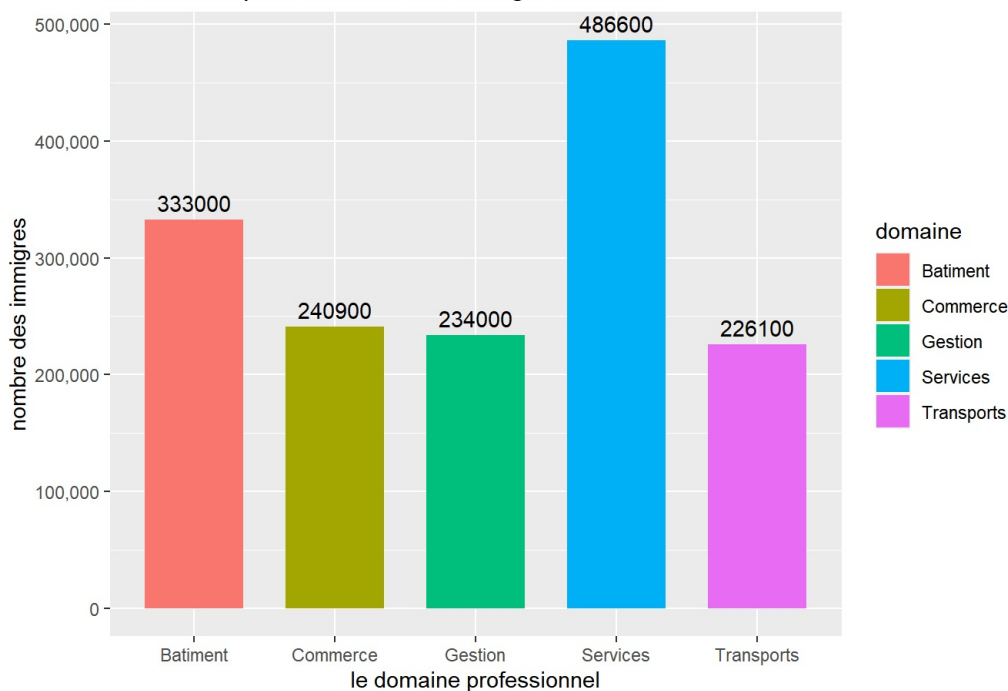
자의 경우 **Santé, action sociale, culturelle et sportive(의료, 문화 및 스포츠)** 직종에 종사하는 사람이 가장 많았다. 여기에는 보건위생분야에 종사하는 의사와 간병인, 그리고 스포츠 관련 종사자를 포함한다. 다음으로는 **Gestion, administration des entreprises(경영)** 종사하는 사람이 많았다. 여기에는 비서나, 회계업무를 맡는 직원, 비즈니스 리더 등 경영분야에 종사하는 사람들이 포함되었다. 세번째로 가장 많은 부문을 차지한 것은 **Commerce(상업)**에 종사하는 사람이었다. 여기에는 판매원, 영업담당자 등이 이에 속한다.

(3)-2.이민자 상위 5개 직업 분야

다음으로는 이민자가 종사하고 있는 직업 상위 5개 분야를 시각화하였다.

```
ggplot(I2018, aes(x=domaine, y=immigrés, fill=domaine))+
  geom_bar(width = 0.7, stat="identity")+
  labs (x="le domaine professionnel",
        y="nombre des immigrés",
        title="le domaine professionnel des immigrés en 2018")+
  geom_text(aes(label=immigrés), vjust=-0.5) +
  scale_y_continuous(labels = scales::comma, breaks = seq(0,700000,100000))
```

le domaine professionnel des immigrés en 2018



눈에 띄는 것은 비이민자의 직업 분포에서 4위를 차지하였던 **Services aux particuliers et aux collectivités (서비스업 종사자)**가 가장 많은 비율을 차지하고 있다는 점이다. 그 다음순위를 차지하는 **Bâtiment, travaux publics(건축업)** 또한 비이민자의 직업부문 5순위안에 들지 않았던 분야이다. 마지막으로 주목할 만한 점은 비이민자 직업 상위 5위에 없었던 **Transports, logistique et tourisme (교통 물류업 및 관광업)** 분야가 이민자 집단에서는 5위로 올라와있다는 점이다.

7. 성별 및 학위 데이터

2018년과 가장 가까운 해인 2019년의 이민자의 전체 성별과 학벌에 관한 데이터를 살펴보도록하겠다. ## 7.1. 데이터 불러오기

```
diplome_age <- read_delim("diplome_age.txt",
                          delim = "\t", escape_double = FALSE,
                          trim_ws = TRUE)
```

```
## Rows: 19 Columns: 6
## -- Column specification -----
## Delimiter: "\t"
## chr (1): Caractéristique
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

7.2. 데이터 추출

```
sex_immi <-diplome_age[1,] #성비 정보만 추출
age_immi <-diplome_age[c(3:7),] #연령 정보만 추출
dip_immi <-diplome_age[c(9:12),] #학위 정보만 추출
```

7.3. 데이터 전처리 및 그래프 생성

(1) 나이데이터:age_immi

(1)-1. 대륙별 이민자의 연령pie chart

```
mycol1 <-c("#FF9E9B", "#FFF56E", "#AAEBAA", "#9BC3FF", "#6E6ED7") # 사용할 색상 저장

opar = par(mfrow=c(2,2), mar=c(2,2,2,2)+0.1)

labels_age <-c("Moins de 18", "18-29", # 나이 범주 입력
              "30-44", "45-64", "65 ou plus")

labels_afri <-paste(labels_age, age_immi$Afrique) # 퍼센트 수치 입력
labels_afri <- paste(labels_afri, "%",sep="")

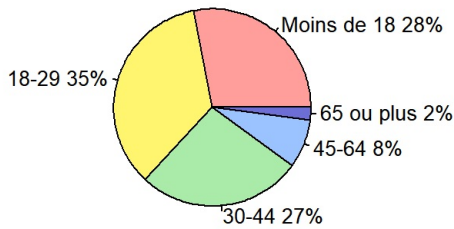
pie(age_immi$Afrique,labels=labels_afri,
    main="Afrique",col =mycol1) # 아프리카 대륙 파이차트 출력

labels_eu <-paste(labels_age, age_immi$Europe)
labels_eu <- paste(labels_eu, "%",sep="")
pie(age_immi$Europe,labels=labels_eu,main="Europe",col =mycol1) #유럽대륙 파이차트

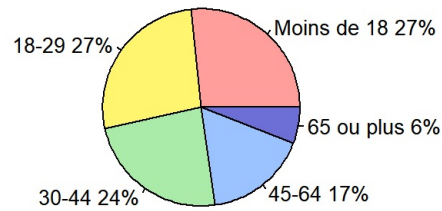
labels_asie <-paste(labels_age, age_immi$Asie)
labels_asie <- paste(labels_asie, "%",sep="")
pie(age_immi$Asie,labels=labels_asie,main="Asie",col =mycol1) #아시아대륙 파이차트

labels_am <-paste(labels_age, age_immi`Amérique, Océanie`)
labels_am <- paste(labels_am, "%",sep="")
pie(age_immi`Amérique, Océanie`,labels=labels_am,main="Amérique, Océanie",col =mycol1) #기타대륙 파이차트
```

Afrique

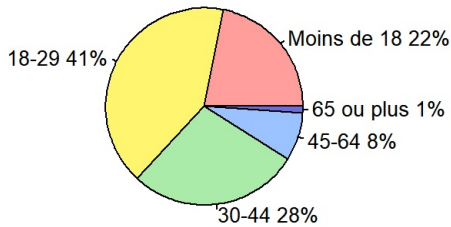


Europe

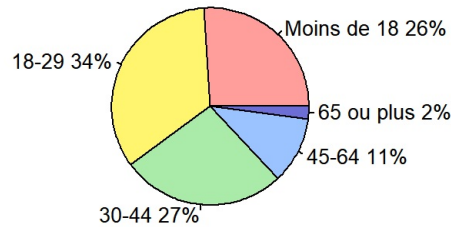


시각화 결과 모든 대륙에서 18세/이서

Asie



Amerique, Oceanie

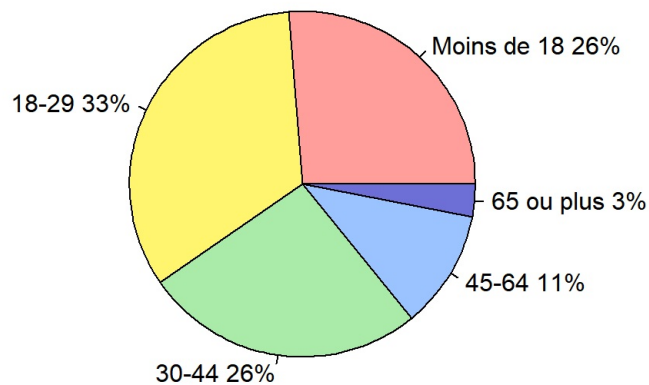


29세 사이의 이민자 비율이 가장 높은 것으로 나타났으며 65세 이상의 이민자 수가 가장 낮은 비율을 차지했다.

- 모든 대륙 이민자의 연령 분포

```
##### 모든 대륙 이민자의 연령 분포
opar = par(mfrow=c(1,1))
labels_en <- paste(labels_age, age_immi$`Ensemble des immigrants`)
labels_en <- paste(labels_en, "%", sep=" ")
pie(age_immi$`Ensemble des immigrants`, labels=labels_en, main="Ensemble", col =mycol1)
```

Ensemble



모든 대륙을 합하여 파이 차트로 만들

었을 때, 18세에서 29세 사이의 이민자가 가장 많았으며 가장 적은 비율은 65세 이상 이민자가 차지하고 있었다.

(1)-2. 대륙별 이민자의 연령 bar_plot

pie chart를 이용해 나타냈던 데이터를 barplot()을 이용하여 시각화하였다.

```

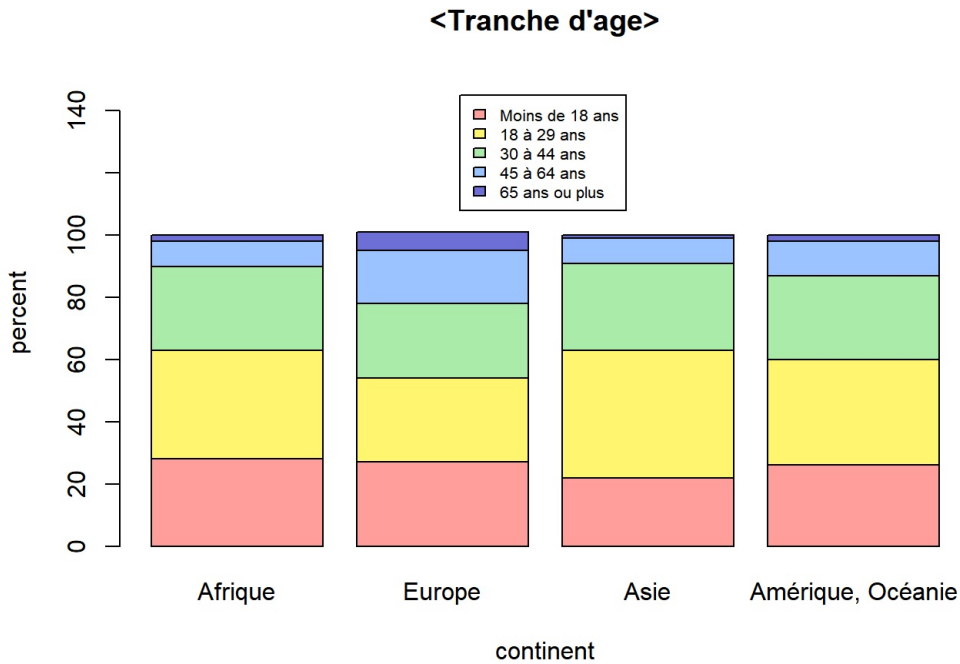
opar = par(mfrow=c(1,1))
age_count <- as.matrix(age_immi[,c(2:5)])
age_plot <- barplot(age_count,
                    main="<Tranche d'âge>",
                    width=0.5,
                    ylab="percent",
                    xlab="continent",
                    col=mycol1,
                    ylim=c(0,150))

```

```

legend(1, 145, age_immi$Caractéristique,
       fill = mycol1, cex = 0.65)

```



(2) 학력 데이터: dip_immi

전체 이민자의 학력수준을 시각화해보았다. 확인결과 모든 대륙에서 고등교육을 이수한 이민자의 비율이 가장 높은 것으로 나타났다.

```

mycol2 = c("#FF92B1", "#FFEB5A", "#AAEBAA", "#9BC3FF")
dip_count <- as.matrix(dip_immi[,c(2:5)])

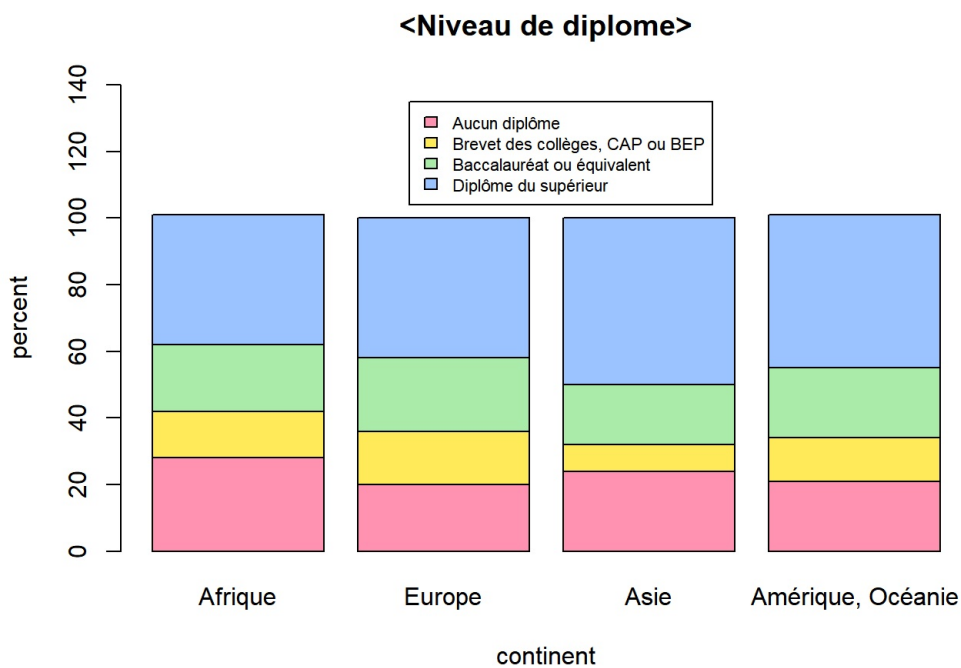
dip_plot <- barplot(dip_count,
                    main="<Niveau de diplôme>",
                    width=0.5,
                    ylab="percent",
                    xlab="continent",
                    col=mycol2,
                    ylim=c(0,140))

```

```

legend(0.85, 135, dip_immi$Caractéristique,
       fill = mycol2, cex = 0.7)

```



(3) 성비 데이터: sex_immi

전체 성별 비율을 시각화해보았다. 확인 결과 여성이 52%, 남성이 48%로 두 성별이 비슷한 비율로 나타난 것을 확인할 수 있다.

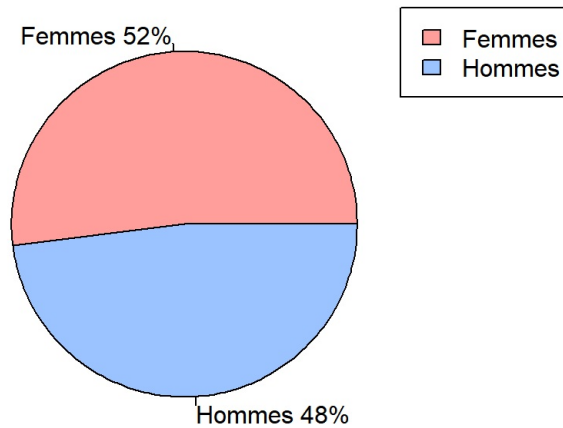
```
sex_count <- sex_immi[, c(1, 6)]
hommes <- c("Hommes", 100 - sex_immi[, 6]) # 100 - 여성비율 = 남성비율
sex_count[2,] <- hommes # 두번째 열의 남성 데이터 입력
sex_count
```

```
## # A tibble: 2 x 2
##   `Caractéristique` `Ensemble des immigrés`
##   <chr>                <dbl>
## 1 Femmes                52
## 2 Hommes                48
```

```
sexe_labels <- paste(sex_count$Caractéristique, sex_count$`Ensemble des immigrés`)
sexe_labels <- paste(sexe_labels, "%", sep="")

pie(sex_count$`Ensemble des immigrés`,
    labels=sexe_labels,
    col=c("#FF9E9B", "#9BC3FF"),)

legend(1, 1, sex_count$Caractéristique,
    fill = c("#FF9E9B", "#9BC3FF"), cex = 1)
```



8. 이민자의 출신 대륙별 경제활동 현황 시각화

8.1. 데이터 불러오기

```
Type_act2018 <- read.csv("type d'activite et pays de naissance en 2018.csv", header=T)
```

8.2. 데이터 전처리

(1) df의 column명 지정

```
Type_act2018 <- Type_act2018[c(7:18),] # 모든 성별을 포괄하는 데이터셋만 추출
names(Type_act2018) <- c("Continent", "active",
                        "chomeur", "retraite",
                        "eleves", "foyer", "AutresInact", "total") #데이터의 열 이름 지정
```

(2) 각 나라를 대륙별로 묶어주기

(2)-1. 데이터 타입 변환

```
for(i in 2:8){ # 국가명을 제외한 데이터를 숫자형으로 변환해주는 반복문
  Type_act2018[,i] <- as.numeric(Type_act2018[,i]) # i번째 열의 데이터를 숫자형으로 변환해 다시 i번째 열의 지정
}
```

(2)-2. 모든 대륙의 정보가 요약된 새로운 데이터 프레임 생성

```

Con_ens2018 <- data.frame(Continent = c("Europe","Afrique","Autres pays"), #대륙명

                           Emloi = c(sum(Type_act2018$active[1:5]), #유럽대륙의 고용인구 합계
                                       sum(Type_act2018$active[6:9]), #아프리카대륙의 고용인구 합계
                                       sum(Type_act2018$active[10:11])), #기타 대륙의 고용인구 합계

                           Chomeurs = c(sum(Type_act2018$chomeur[1:5]), #유럽대륙의 실업인구 합계
                                          sum(Type_act2018$chomeur[6:9]), #아프리카대륙의 실업인구 합계
                                          sum(Type_act2018$chomeur[10:11])), #기타대륙의 실업인구 합계

                           Retraite = c(sum(Type_act2018$retraite[1:5]), #유럽대륙의 은퇴인구 합계
                                          sum(Type_act2018$retraite[6:9]), #아프리카대륙의 은퇴인구 합계
                                          sum(Type_act2018$retraite[10:11])), #유럽기타대륙의 은퇴인구 합계

                           Inactive= c(sum(sum(Type_act2018$eleves[1:5]), #유럽대륙의 비경제활동인구 합계
                                          sum(Type_act2018$foyer[1:5]),
                                          sum(Type_act2018$AutresInact[1:5])),

                                          sum(sum(Type_act2018$eleves[6:9]), #아프리카대륙의 은퇴인구 합계
                                          sum(Type_act2018$foyer[6:9]),
                                          sum(Type_act2018$AutresInact[6:9])),

                                          sum(sum(Type_act2018$eleves[10:11]), #기타대륙의 은퇴인구 합계
                                          sum(Type_act2018$foyer[10:11]),
                                          sum(Type_act2018$AutresInact[10:11]))))

knitr::kable(head(Con_ens2018,3))

```

Continent	Emloi	Chomeurs	Retraite	Inactive
Europe	940348	155593	665692	297890
Afrique	1312217	458283	428913	641987
Autres pays	600157	180873	129472	339054

8.3. 데이터 시각화

(1)-1. 경제활동 현황 별 출신국가 빈도

```

#table 형태로 구성된 df을 matrix로 변환

# df를 data.table로 변환해주는 패키지

contin_table18 <- as.data.table(Con_ens2018)
class(contin_table18) # data.table로 변환된 것을 확인

```

```
## [1] "data.table" "data.frame"
```

```
number1 <- as.matrix(contin_table18[,2:5]/10000) # 매트릭스 형태로 다시 변환
```

(1)-2. barplot을 이용하여 그래프 그리기

출신 대륙별로 경제활동 현황을 시각화하였다. 고용인구 중에서는 실업인구와 고용인구 그리고 비경제활동인구에서 아프리카 대륙 출신이 가장 많았으며, 은퇴인구 비율에서는 유럽대륙 출신 이민자가 가장 많았다.


```

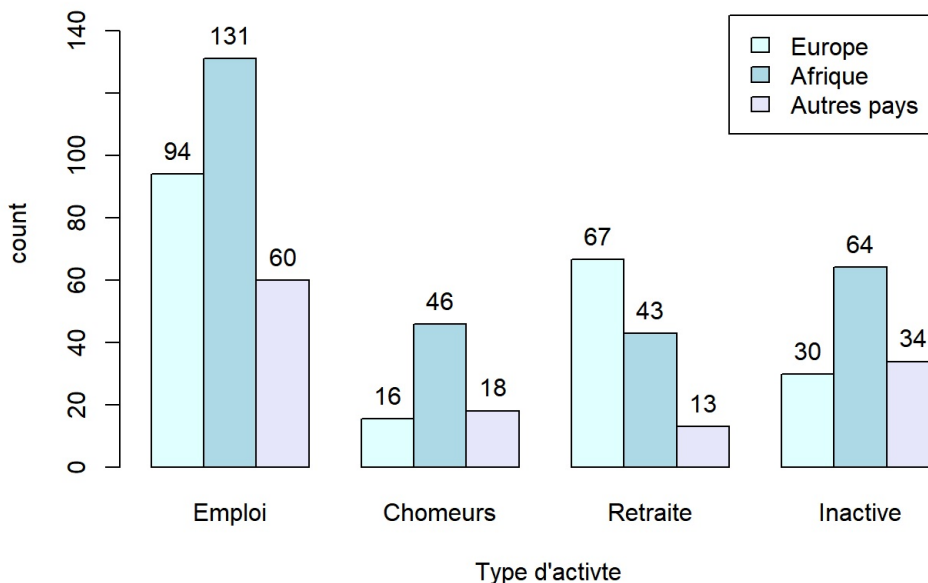
contin_plot <- barplot(number1,
  beside = T, # 가로로 펼쳐진 그래프
  xlab=" Type d'activte",
  ylab="count",
  main="<Type d'activite et pays de naissance en 2018>",
  col = c("lightcyan", "lightblue", "lavender"), ylim = c(0,150))

text(x=contin_plot, y=number1, # 그래프 위에 숫자 추가
  pos=3, labels=round(number1),
  col="black")

legend(12, 145, contin_table18$Continent,
  fill = c("lightcyan", "lightblue", "lavender"), cex = 1) # 그래프에 범례추가

```

<Type d'activite et pays de naissance en 2018>



< 2018년 자료 시각화하기 >

통계 분석에 사용할 2018년 대도시 3개에 대한 인구총조사 자료를 시각화해보았다. 해당 데이터가 어떤 특성을 지니고 있는지 간략하고 확인해보고 분석해보고자한다.

1. 데이터 불러오기

해당 데이터는 따로 전처리 과정으로 통해 1:1000 비율로 샘플링 되었다. 원본 데이터가 가지고 있는 데이터의 총 개수는 356만명에 해당하는 데이터였고, 이를 3563개의 행을 가진 데이터 프레임으로 압축하여 csv 파일로 저장하였다.

```

demo2018 <- read.csv("demo_sample_2018.csv", header=T)
demo2018 <- demo2018[, -1] # 의미없는 첫번째 열 삭제

```

2. 데이터 프레임 순서 섞기

현재 데이터 프레임은 순서대로 입력되어 정렬된 상태이다. 이렇게 정렬된 데이터를 통계분석에 사용하면 분석 결과에 영향을 미치므로 샘플링 함수를 통해 무작위로 순서를 배치해주었다.

```

random_shuffle <- sample(1:3563)
demo2018 <- demo2018[random_shuffle,]
head(demo2018)

```

```

##      ID Sexe immi      job region
## 542  542   1    1      Autres  Mar
## 968  968   2    2      Cadres  Lyon
## 1347 1347   2    2      Autres  Lyon
## 2338 2338   2    1  Employes  Paris
## 3080 3080   1    2      Autres  Paris
## 2167 2167   2    2 Professions intermediaire  Paris

```

3. 데이터 한 눈에 살펴보기

3.1. 성비

```
library(ggplot2)
demo2018$Sexe[demo2018$Sexe==1] ="Homme"
demo2018$Sexe[demo2018$Sexe==2] ="Femme"

sexe_table<-table(demo2018$Sexe)
sexe_table<-round(prop.table(sexe_table)*100,1) # 백분율로 만들어 반올림
sexe_table
```

```
##
## Femme Homme
## 54.1 45.9
```

```
as.matrix(sexe_table)
```

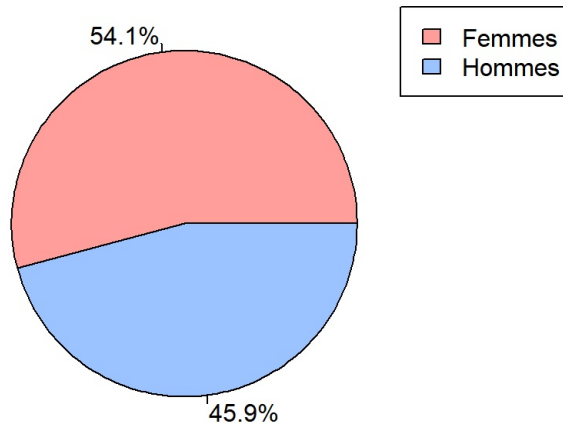
```
##      [,1]
## Femme 54.1
## Homme 45.9
```

```
sexe_labels <- paste(sexe_table, "%",sep="")

pie(sexe_table,
    labels=sexe_labels,
    main="<Immigrant Sexe proportion>",
    col=c("#FF9E9B", "#9BC3FF"))

sexe_labels=c("Femmes", "Hommes")
legend(1, 1, sexe_labels,
    fill = c("#FF9E9B", "#9BC3FF"), cex = 1)
```

<Immigrant Sexe proportion>



시각화를 확인해본 결과 전체 2018년 이민자의 성비보다 여성의 비율이 조금 더 높게 나타났으나 여성 54% 남성 약 46% 비율로 비슷한 수준인 것으로 확인되었다.

3.2.전체 인구 중 이민자의 비율

```
demo2018$immi[demo2018$immi==1] ="immigrés"
demo2018$immi[demo2018$immi==2] ="Non-immigrés"

immi_table <- table(demo2018$immi)
immi_table<-round(prop.table(immi_table)*100,1) # 백분율로 만들어 반올림
immi_table
```

```
##
##   immigres Non-immigres
##      17.9      82.1
```

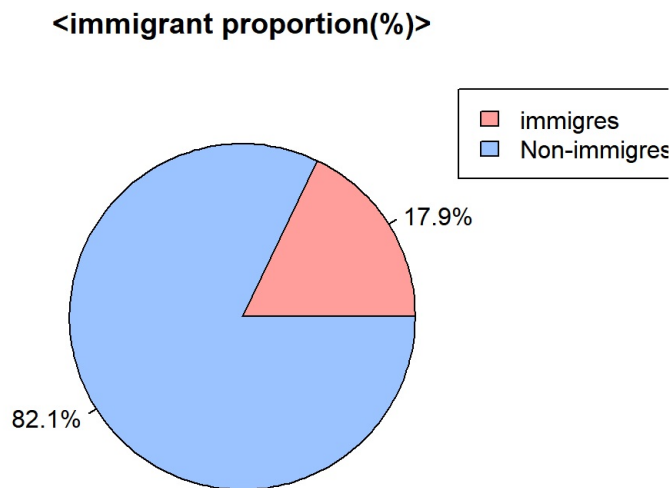
```
as.matrix(immi_table)
```

```
##           [,1]
## immigres   17.9
## Non-immigres 82.1
```

```
immi_labels <- paste(immi_table, "%", sep="")

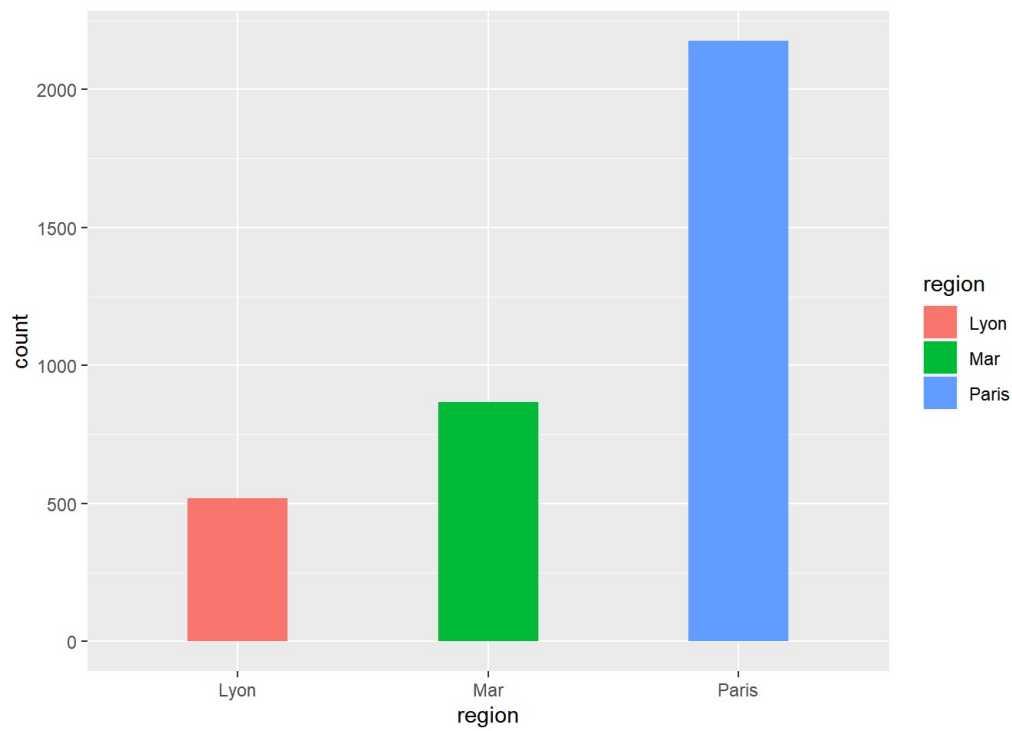
pie(immi_table,
    labels=immi_labels,
    main="<immigrant proportion(%)>",
    col=c("#FF9E9B", "#9BC3FF"))

immi_labels=c("immigrés", "Non-immigrés")
legend(1, 1.05, immi_labels,
    fill = c("#FF9E9B", "#9BC3FF"), cex = 1)
```



3.3. 지역별 데이터 개수 비교

```
ggplot(demo2018, aes(x=region, fill=region)) +
  geom_bar(width=0.4,)
```



3.4. 이민자와 비이민자의 직업 비교

```
ggplot(demo2018, aes(x=immi, fill=job)) +
  geom_bar(width=0.4, position='fill') +
  scale_fill_brewer(palette="Set3")
```

