Week5Lab

Frank Shen

2019/2/4

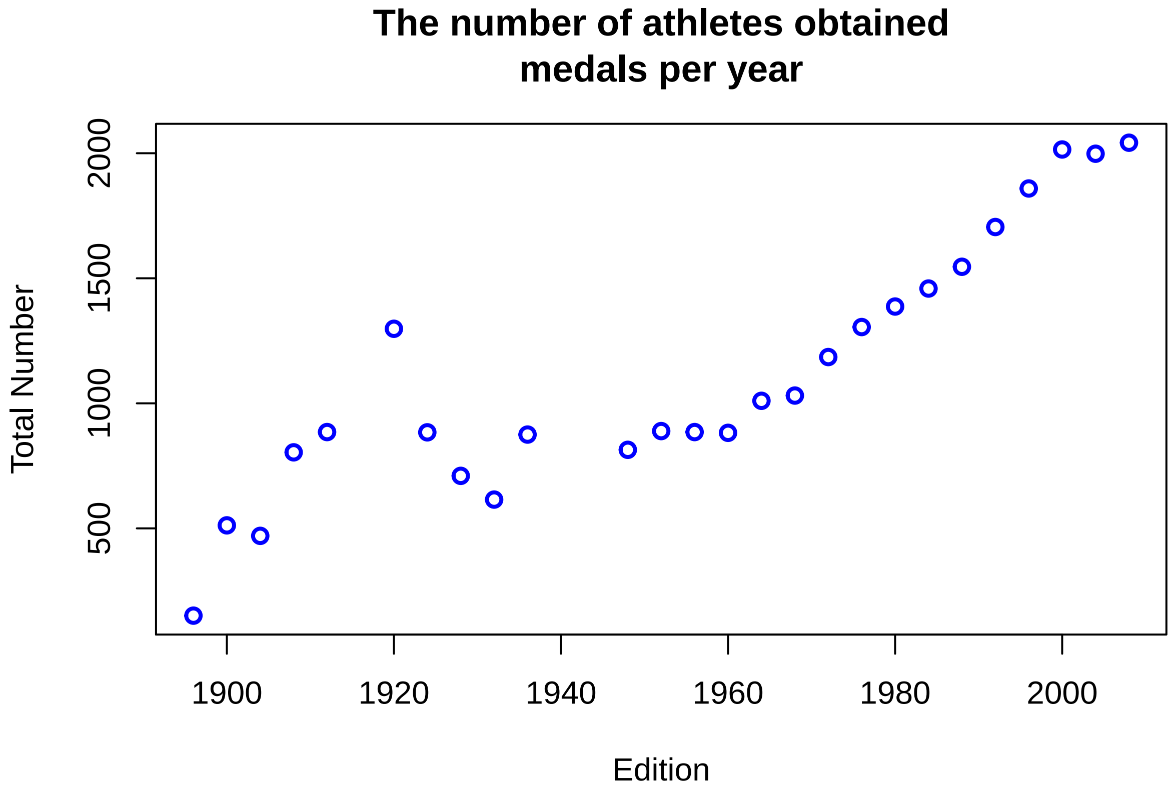
```
library(RSQLite)
library(DBI)
con = dbConnect(RSQLite::SQLite(), dbname="stat240Week5.sqlite")
exm="SELECT * FROM Olymp_meds"
omp = dbGetQuery(con, exm)
names(omp)
```

```
##  [1] "City"         "Edition"      "Sport"        "Discipline"
##  [5] "Athlete"      "NOC"          "Gender"       "Event"
##  [9] "Event_gender" "Medal"
```

```
mov_avg1 = "SELECT Edition, Count(Edition) AS TotalNumber FROM Olymp_meds GROUP BY Edition"
out = dbGetQuery(con, mov_avg1)
names(out)
```

```
## [1] "Edition"     "TotalNumber"
```

```
##Question1 a)
plot(out, type="p", main= "The number of athletes obtained
medals per year", xlab= "Edition", ylab="Total Number", col="blue", lwd = "2")
```



```
##Question1 b)
mov_avg2 = "CREATE VIEW IF NOT EXISTS tot_meds AS SELECT Edition, Count(Edition) AS TotalNumber FROM Ol
dbSendQuery(con, mov_avg2)
```

```
## <SQLiteResult>
##   SQL  CREATE VIEW IF NOT EXISTS tot_meds AS SELECT Edition, Count(Edition) AS TotalNumber FROM Olymp
##   ROWS Fetched: 0 [complete]
##        Changed: 0
```

```
dbListTables(con)
```

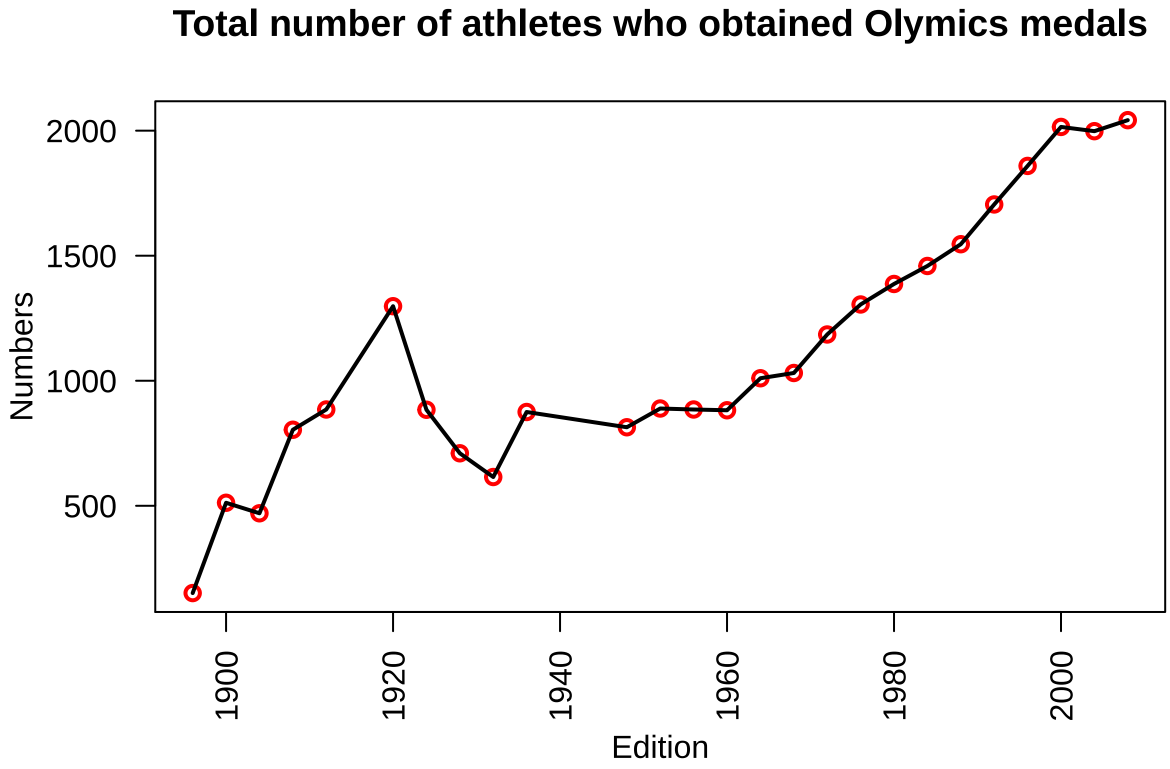
```
## Warning: Closing open result set, pending rows
```

```
##  [1] "CA"                  "Can_tot_meds"        "Olymp_meds"
##  [4] "POP2006"             "POP2011"             "Pokem"
##  [7] "SummerMedalPerYear"  "SummerMedalavgyears" "Vanpoke"
## [10] "WinterO"             "tickets"             "tot_meds"
## [13] "zip"
```

```
##Question1 c)
mov_avg3 = "SELECT Edition, TotalNumber FROM tot_meds"
out = dbGetQuery(con, mov_avg3)

plot(out$Edition, out$TotalNumber, xlab = "Edition", ylab = "Numbers",main = "Total number of athletes w

lines(out$Edition, out$TotalNumber, col= 1 , lwd= 2 )
```

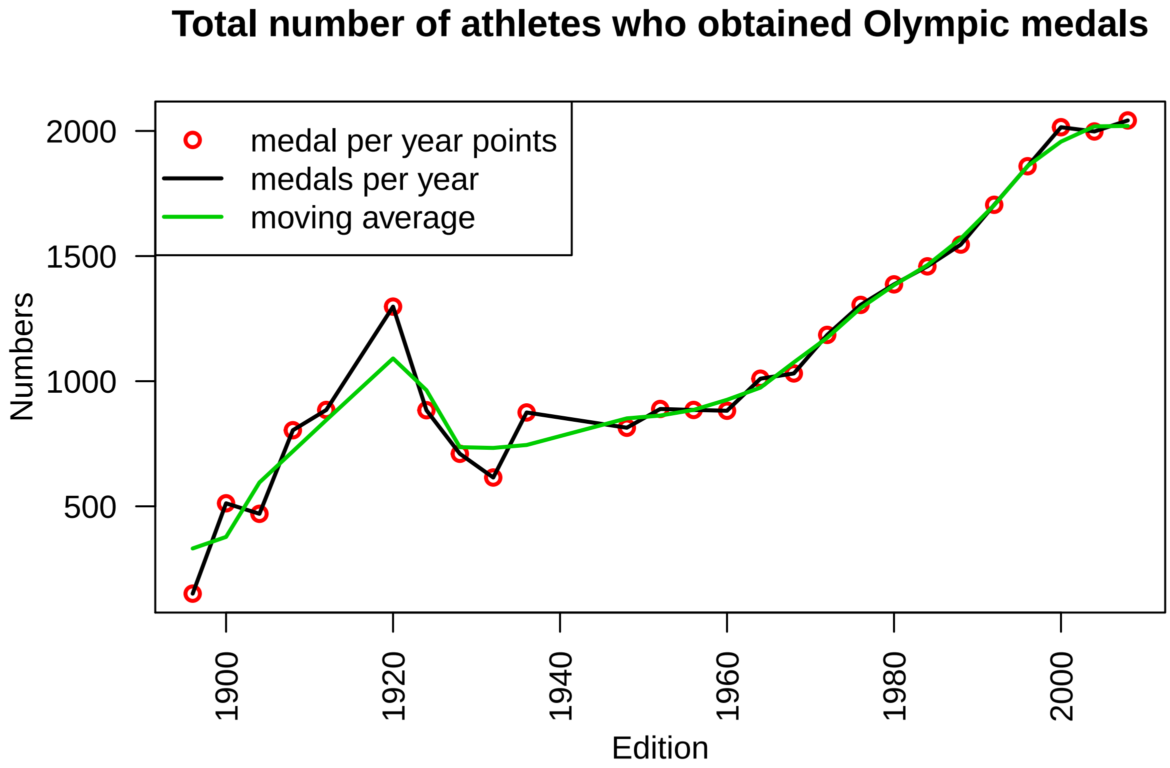


```
check = "SELECT * FROM tot_meds AS t, (SELECT t1.Edition, AVG(t2.TotalNumber) AS mavg FROM tot_meds AS t
movingAvg = dbGetQuery(con, check)
##Question1 d)
names(out)
```

```
## [1] "Edition"     "TotalNumber"
```

```
plot(out$Edition , out$TotalNumber, xlab="Edition",ylab="Numbers", main="Total number of athletes who ol
lines(out[[1]], out[[2]], col=1,lwd=2)
lines(movingAvg$Edition, movingAvg$mavg,type="l",col=3,lwd=2)
```

```r
legend("topleft",lwd=2,lty=c(NA,1,1),pch=c(1,NA,NA),col=c(2,1,3), c("medal per year points","medals per
```



```r
dbSendQuery(con, "drop view tot_meds")
```

```
## <SQLiteResult>
##   SQL  drop view tot_meds
##   ROWS Fetched: 0 [complete]
##        Changed: 0
```

```r
##Question 2 a)

meds ="SELECT Year, Count(Year) AS TotalNumber FROM WinterO GROUP BY Country, Year HAVING Country =='Ca
out = dbGetQuery(con, meds)
```

```
## Warning: Closing open result set, pending rows
```

```r
list(out)
```

```
## [[1]]
##   year TotalNumber
## 1 2010         102
```

```r
##becasue the table has different count of Athlete between in the stat240 Week5 table and wikipedia tabl
##Question 2 b)
meds = "SELECT Event,Country, Medal FROM WinterO WHERE Year== '2010' AND Country =='Canada' AND Medal !
out = dbGetQuery(con, meds)
head(out)
```

```
##          Event Country  Medal
## 1 moguls women  Canada Silver
## 2 four-man men  Canada Bronze
## 3 four-man men  Canada Bronze
## 4 four-man men  Canada Bronze
## 5 four-man men  Canada Bronze
## 6 four-man men  Canada Bronze
```

```r
dim(out)
```

```
## [1] 102   3
```

```r
dbSendQuery(con, 'drop view SummerMedalavgyears')
```

```
## <SQLiteResult>
##   SQL  drop view SummerMedalavgyears
##   ROWS Fetched: 0 [complete]
##        Changed: 0
```

```r
##Question 3 a)

summer= "SELECT Edition, Count(Edition) AS TotalNumber FROM Olymp_meds WHERE NOC == 'FRA' GROUP BY Editi
summermedals= dbGetQuery(con, summer)
```

```
## Warning: Closing open result set, pending rows
```

```r
head(summermedals)
```

```
##   Edition TotalNumber
## 1    1896          11
## 2    1900         185
## 3    1908          35
## 4    1912          25
## 5    1920         141
## 6    1924         122
```

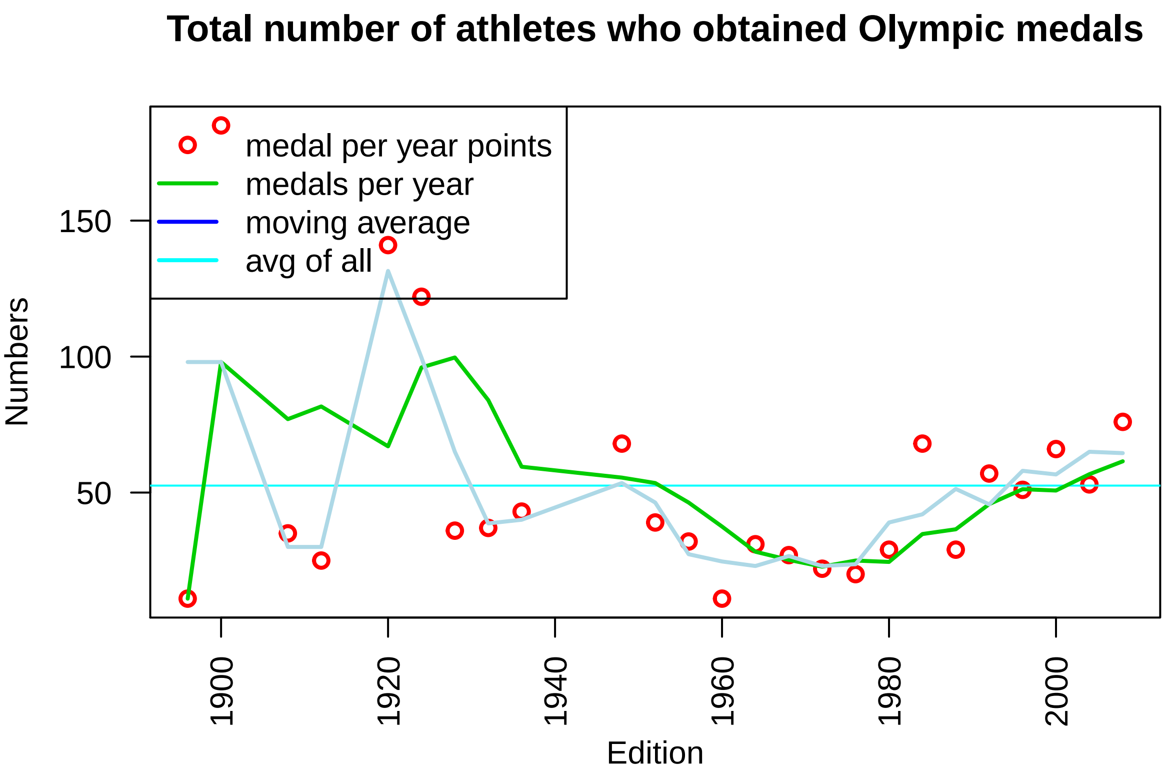
```r
Summermeds = paste("CREATE VIEW SummerMedalavgyears AS",summer)
dbSendQuery(con, Summermeds)
```

```
## <SQLiteResult>
##   SQL  CREATE VIEW SummerMedalavgyears AS SELECT Edition, Count(Edition) AS TotalNumber FROM Olymp_me
##   ROWS Fetched: 0 [complete]
##        Changed: 0
```

```r
plot(summermedals$Edition , summermedals$TotalNumber, xlab="Edition",ylab="Numbers", main="Total number
abline(h=mean(summermedals$TotalNumber),col=5)
check = "SELECT * FROM SummerMedalavgyears AS t, (SELECT t1.Edition, AVG(t2.TotalNumber) AS mavg FROM S
movingAvg1= dbGetQuery(con, check)
```

```
## Warning: Closing open result set, pending rows
```

```r
lines(movingAvg1[[1]], movingAvg1[[4]],type="l",col=3,lwd=2)
check = "SELECT * FROM SummerMedalavgyears AS t, (SELECT t1.Edition, AVG(t2.TotalNumber) AS mavg FROM Su
movingAvg1= dbGetQuery(con, check)
lines(movingAvg1[[1]], movingAvg1[[4]],type="l",col="lightblue",lwd=2)
legend("topleft",lwd=2,lty=c(NA,1,1,1),pch=c(1,NA,NA,NA),col=c(2,3,4,5), c("medal per year points","meda
```



```r
Canadian = "CREATE VIEW IF NOT EXISTS Can_tot_meds AS SELECT Edition AS Year,
Count(Edition) AS TotalNumber FROM  Olymp_meds GROUP BY NOC,
Edition HAVING NOC =='CAN'"
dbGetQuery(con, Canadian)
```

```
## Warning in result_fetch(res@ptr, n = n): Don't need to call dbFetch() for
## statements, only for queries

## data frame with 0 columns and 0 rows
```

```r
summaries = "SELECT COUNT(Year) AS YearsInOlympics,
AVG(TotalNumber) AS AVGmedalcount , MIN(TotalNumber) AS MINmedalcount ,
MAX(TotalNumber) AS MAXmedalcount  FROM Can_tot_meds"
(out = dbGetQuery(con, summaries))
```

```
##   YearsInOlympics AVGmedalcount MINmedalcount MAXmedalcount
## 1              24      24.66667             2            86
```

```r
getmedian = "SELECT TotalNumber AS Median FROM Can_tot_meds
ORDER BY TotalNumber LIMIT 1 OFFSET (SELECT COUNT(TotalNumber)
        FROM Can_tot_meds) /2"
(out = dbGetQuery(con, getmedian))
```

```
##   Median
## 1     20
```

```r
##Question 4:
dbSendQuery(con,"CREATE VIEW IF NOT EXISTS LakePlacid AS SELECT year,
Count(Country) AS TotalNumber, Country FROM  WinterO GROUP BY Country, year HAVING place == 'Lake Placi
```

```
## <SQLiteResult>
##   SQL  CREATE VIEW IF NOT EXISTS LakePlacid AS SELECT year,
## Count(Country) AS TotalNumber, Country FROM  WinterO GROUP BY Country, year HAVING place == 'Lake Pl
##   ROWS Fetched: 0 [complete]
##        Changed: 0
```

```
pick10 = "SELECT TotalNumber AS percentile10 FROM LakePlacid ORDER BY TotalNumber LIMIT 1 OFFSET (SELEC
percent10 = dbGetQuery(con, pick10)
```

```
## Warning: Closing open result set, pending rows
```

```
pick30 = "SELECT TotalNumber AS percentile30 FROM LakePlacid ORDER BY TotalNumber LIMIT 1 OFFSET (SELEC
percent30 = dbGetQuery(con, pick30)
pick50 = "SELECT TotalNumber AS percentile50 FROM LakePlacid ORDER BY TotalNumber LIMIT 1 OFFSET (SELEC
percent50 = dbGetQuery(con, pick50)
pick70 = "SELECT TotalNumber AS percentile70 FROM LakePlacid ORDER BY TotalNumber LIMIT 1 OFFSET (SELEC
percent70 = dbGetQuery(con, pick70)
pick90 = "SELECT TotalNumber AS percentile90 FROM LakePlacid ORDER BY TotalNumber LIMIT 1 OFFSET (SELEC
percent90 = dbGetQuery(con, pick90)
c(percent10,percent30,percent50,percent70,percent90)
```

```
## $percentile10
## [1] 1
##
## $percentile30
## [1] 3
##
## $percentile50
## [1] 4
##
## $percentile70
## [1] 13
##
## $percentile90
## [1] 47
```

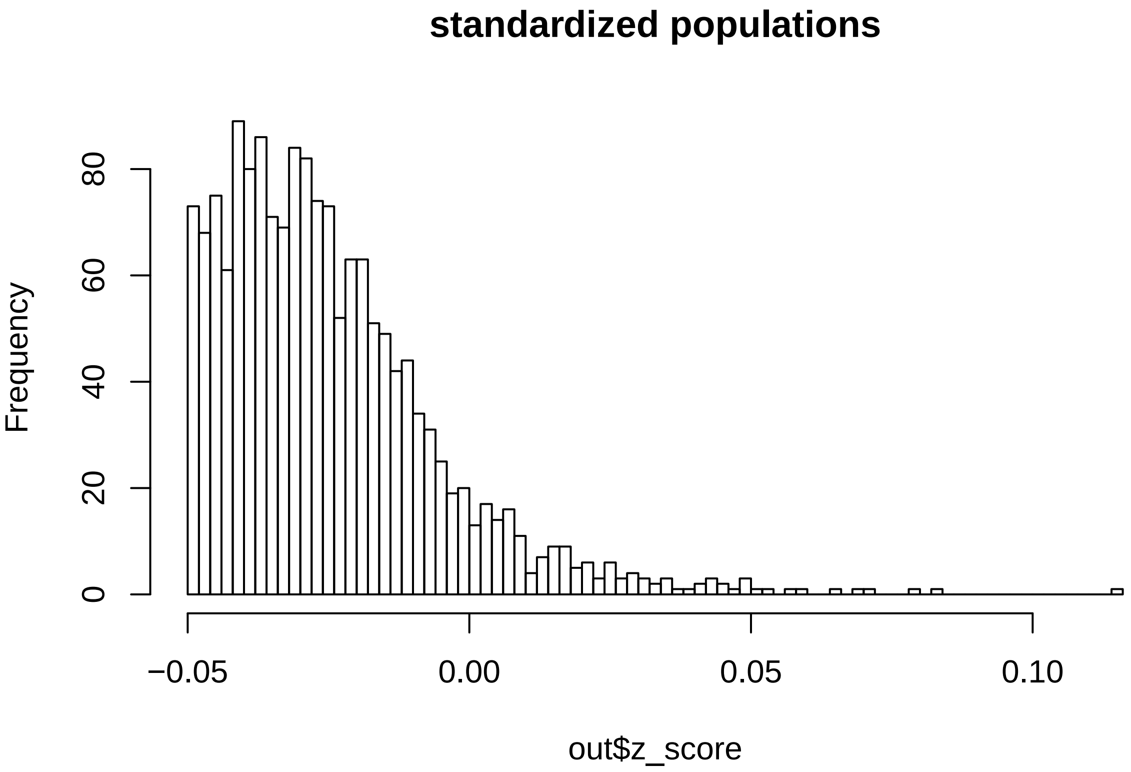
```
check = "SELECT * FROM POP2011"
get= dbGetQuery(con,check)
head(get)
```

```
##   Geographic_name
## 1          Canada
## 2             A0A
## 3             A0B
## 4             A0C
## 5             A0E
## 6             A0G
##   Incompletely_enumerated_Indian_reserves_and_Indian_settlements__2011
## 1                                                                    1
## 2                                                                   NA
## 3                                                                   NA
## 4                                                                   NA
## 5                                                                   NA
## 6                                                                   NA
##   Population__2011 Total_private_dwellings__2011
## 1         33476688                      14569633
## 2            46297                         23950
## 3            20985                         12585
## 4            12834                          8272
## 5           23384                          12733
## 6           36264                          21153
##   Private_dwellings_occupied_by_usual_residents__2011
## 1                                            13320614
## 2                                               18701
## 3                                                8854
## 4                                                5482
## 5                                                9659
## 6                                               14967
```

```
initExtension(con)
dbGetQuery(con, "SELECT  STDEV(TotalNumber) FROM Can_tot_meds")
```

```
##   STDEV(TotalNumber)
## 1           19.63065
```

```
sql_z = "WITH pop_cnt AS (SELECT avg(Population__2011) AS mean, stdev(Population__2011) AS sd FROM POP2
out = dbGetQuery(con, sql_z)
##Question5 a)
sql_z = "WITH pop_cnt AS (SELECT avg(Population__2011) AS mean, stdev(Population__2011) AS sd FROM POP2
out = dbGetQuery(con, sql_z)
hist(out$z_score,100,main="standardized populations")
##Question5 b)
library(sp)
```



```
library(rworldmap)
```

```
## ### Welcome to rworldmap ###
```

```
## For a short introduction type :   vignette('rworldmap')
```

```
library(rworldxtra)
worldmap = getMap(resolution = "high")
dim(worldmap)
```

```
## [1] 253  51
```

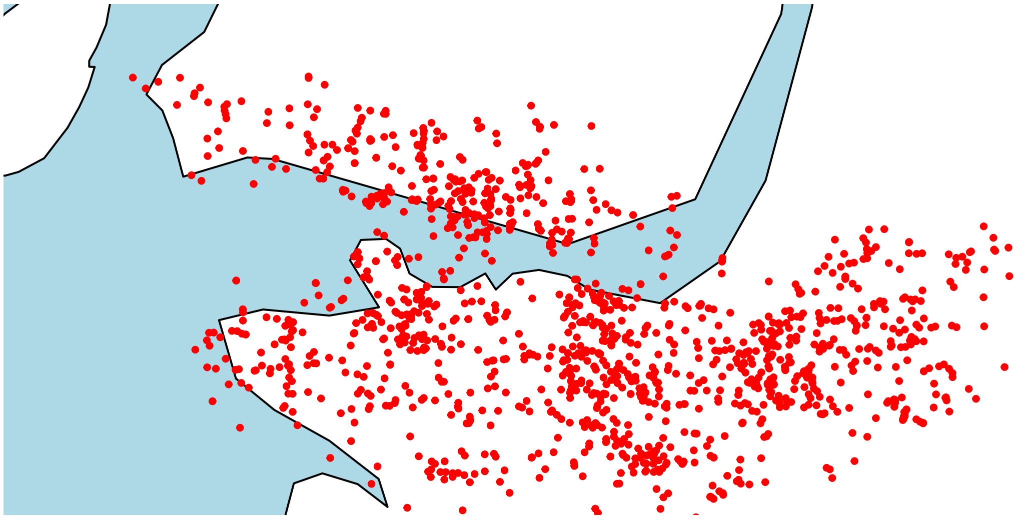
```
names(dbReadTable(con, "Vanpoke"))
```

```
##  [1] "date"       "time_until" "name"       "cp"         "level"
##  [6] "iv"         "moveset"    "gender"     "address"    "city"
## [11] "latitude"   "longitude"  "weather"
```

```
names(dbReadTable(con, "Pokem"))
```

```
##  [1] "Number"           "Name"             "Type_1"
##  [4] "Type_2"           "Total"            "HP"
##  [7] "Attack"           "Defense"          "Sp_Atk"
## [10] "Sp_Def"           "Speed"            "Generation"
## [13] "isLegendary"      "Color"            "hasGender"
## [16] "Pr_Male"          "Egg_Group_1"      "Egg_Group_2"
## [19] "hasMegaEvolution" "Height_m"         "Weight_kg"
## [22] "Catch_Rate"       "Body_Style"
```

```
sql_qry = "SELECT * FROM Pokem INNER JOIN Vanpoke ON Pokem.name=Vanpoke.name WHERE Type_1 == 'Ground' or
joindensity = dbGetQuery(con, sql_qry)
NrthAm = worldmap[which(worldmap$REGION == "North America"), ]
plot(NrthAm, col = "white", bg = "lightblue", xlim = c(-123.116226,-123), ylim = c(49.2,49.4), cex =0.8)
points(joindensity$longitude, joindensity$latitude, col="red", pch = 20, cex = 0.6)
```



```
dbSendQuery(con, "drop view LakePlacid")
```

```
## <SQLiteResult>
##   SQL  drop view LakePlacid
##   ROWS Fetched: 0 [complete]
##        Changed: 0
```

```
dbDisconnect(con)
```

```
## Warning in connection_release(conn@ptr): There are 1 result in use. The
## connection will be released when they are closed
```