HW2 Merging Data Frames

Ryan Richardson

26/09/2020

In-Class Portion

```
gdp <- read.csv('data/gdp.csv')
cca <- read.csv('data/countrycharsA.csv')
ccb <- read.csv('data/countrycharsB.csv')
```

```
str(gdp)
```

```
## 'data.frame':    1677 obs. of  1 variable:
##  $ gdp: num  779 821 853 836 740 ...
```

```
str(cca)
```

```
## 'data.frame':    756 obs. of  5 variables:
##  $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ continent: chr  "Asia" "Asia" "Asia" "Asia" ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ pop      : int  8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 1
6317921 22227415 ...
```

```
str(ccb)
```

```
## 'data.frame':    921 obs. of  5 variables:
##  $ country  : chr  "Algeria" "Algeria" "Algeria" "Algeria" ...
##  $ continent: chr  "Africa" "Africa" "Africa" "Africa" ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  43.1 45.7 48.3 51.4 54.5 ...
##  $ pop      : int  9279525 10270856 11000948 12760499 14760787 17152804 20033753 23254956
26298373 29072015 ...
```

```
ccs <- rbind(cca, ccb)
ccgs <- cbind(ccs, gdp)

str(gdp)
```

```
## 'data.frame':    1677 obs. of  1 variable:
##  $ gdp: num  779 821 853 836 740 ...
```

```
str(cca)
```

```
## 'data.frame':    756 obs. of  5 variables:
##  $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ continent: chr  "Asia" "Asia" "Asia" "Asia" ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ pop      : int  8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 1
6317921 22227415 ...
```

```
str(ccs)
```

```
## 'data.frame':    1677 obs. of  5 variables:
##  $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ continent: chr  "Asia" "Asia" "Asia" "Asia" ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ pop      : int  8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 1
6317921 22227415 ...
```

```
str(ccgs)
```

```
## 'data.frame':    1677 obs. of  6 variables:
##  $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ continent: chr  "Asia" "Asia" "Asia" "Asia" ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ pop      : int  8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 1
6317921 22227415 ...
##  $ gdp      : num  779 821 853 836 740 ...
```

```
aboveThreshold = ccgs[which(ccgs$year > 1980 & ccgs$gdp < 20000),]
str(aboveThreshold)
```

```
## 'data.frame':    687 obs. of  6 variables:
##  $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ continent: chr  "Asia" "Asia" "Asia" "Asia" ...
##  $ year     : int  1982 1987 1992 1997 2002 2007 1982 1987 1992 1997 ...
##  $ lifeExp  : num  39.9 40.8 41.7 41.8 42.1 ...
##  $ pop      : int  12881816 13867957 16317921 22227415 25268405 31889923 2780097 3075321 3
326498 3428038 ...
##  $ gdp      : num  978 852 649 635 727 ...
```

Median By Year

```
medianByYear = data.frame("Year" = c(), "MedianGdp" = c(), "ClosestRelativeGdp" = c())
for (i in unique(ccgs$year)) {

  yearResults = ccgs[which(ccgs$year == i),]

  gdpMedian = median(yearResults$gdp)
  gdpDifference = abs(yearResults$gdp - gdpMedian)

  minCountry = yearResults[which.min(gdpDifference),]
  medianByYear = rbind(medianByYear, list(i, gdpMedian, minCountry$country))

}
```

```
## 'data.frame':    12 obs. of  3 variables:
##  $ X1952L       : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ X1968.5283435: num  1969 2173 2335 2678 3339 ...
##  $ X.Swaziland. : chr  "Swaziland" "Czech Republic" "Swaziland" "Israel" ...
```

Year	MedianGDP	Closest Relative GDP
1952	1968.528	Swaziland
1957	2173.220	Czech Republic
1962	2335.440	Swaziland
1967	2678.335	Israel
1972	3339.129	Peru
1977	3798.609	Israel
1982	4241.356	Swaziland
1987	4280.300	Denmark
1992	4386.086	Sudan
1997	4781.825	Austria
2002	5073.194	Austria
2007	6124.371	Myanmar

Median By Year By Continent

```
medianByYearByContinent = data.frame("Year" = c(), "MedianGdp" =  c(), "ClosestRelativeGdp" =
c(), "Continent" = c())
for (i in unique(ccgs$year)) {

  yearResults = ccgs[which(ccgs$year == i),]

  for (j in unique(yearResults$continent)) {

    continentResults = yearResults[which(yearResults$continent == j),]

    gdpMedian = median(continentResults$gdp)
    gdpDifference = abs(continentResults$gdp - gdpMedian)

    minCountry = continentResults[which.min(gdpDifference),]
    medianByYearByContinent = rbind(medianByYearByContinent, list(i, gdpMedian, minCountry$co
untry, minCountry$continent))

    }

}
```

```
## 'data.frame':    58 obs. of  4 variables:
##  $ X1952L             : int  1952 1952 1952 1952 1952 1957 1957 1957 1957 1957 ...
##  $ X2627.009471       : num  2627 2152 2142 1273 3716 ...
##  $ X.Korea..Dem..Rep..: chr  "Korea, Dem. Rep." "France" "Swaziland" "Jamaica" ...
##  $ X.Asia.            : chr  "Asia" "Europe" "Africa" "Americas" ...
```

Year	MedianGDP	Closest Relative GDP	Continent
1952	2627.009	Korea, Dem. Rep.	Asia
1952	2151.935	France	Europe
1952	2142.120	Swaziland	Africa
1952	1272.881	Jamaica	Americas
1952	3715.999	New Zealand	Oceania
1957	2990.011	Korea, Dem. Rep.	Asia
1957	2174.087	Czech Republic	Europe
1957	2420.101	Egypt	Africa
1957	1547.945	Jamaica	Americas
1957	5089.985	Australia	Oceania
1962	3460.937	Korea, Dem. Rep.	Asia
1962	2302.023	Switzerland	Europe
1962	2851.142	Congo, Rep.	Africa
1962	1856.182	Peru	Americas
1962	6989.476	Australia	Oceania
1967	4161.728	Korea, Dem. Rep.	Asia

Year	MedianGDP	Closest Relative GDP	Continent
1967	2673.541	Albania	Europe
1967	3310.026	Swaziland	Africa
1967	2029.228	Brazil	Americas
1967	9626.639	Australia	Oceania
1972	5118.147	Korea, Dem. Rep.	Asia
1972	3146.877	Albania	Europe
1972	3904.302	Egypt	Africa
1972	2523.338	Haiti	Americas
1972	13581.136	New Zealand	Oceania
1977	3815.808	Israel	Asia
1977	3530.743	Albania	Europe
1977	4294.463	Libya	Africa
1977	3248.373	Haiti	Americas
1977	15943.704	Australia	Oceania
1982	4879.508	Jordan	Asia
1982	3864.612	Albania	Europe
1982	4216.228	Swaziland	Africa
1982	18625.414	Australia	Oceania
1987	4903.219	Israel	Asia
1987	4012.951	Norway	Europe
1987	4936.315	Congo, Rep.	Africa
1987	3984.840	Peru	Americas
1987	20527.238	New Zealand	Oceania
1992	4016.240	Jordan	Asia
1992	3438.225	Switzerland	Europe
1992	5868.301	Swaziland	Africa
1992	3553.022	Peru	Americas
1992	20519.806	Australia	Oceania
1997	3614.101	Myanmar	Asia
1997	4428.748	Netherlands	Europe
1997	6336.686	Swaziland	Africa
1997	3876.768	Peru	Americas

Year	MedianGDP	Closest Relative GDP	Continent
1997	22988.846	New Zealand	Oceania
2002	5755.260	Israel	Asia
2002	4806.476	Netherlands	Europe
2002	6412.125	Swaziland	Africa
2002	3783.674	Haiti	Americas
2007	6873.262	Nepal	Asia
2007	5654.767	Netherlands	Europe
2007	8064.199	Libya	Africa
2007	3970.095	Panama	Americas
2007	28712.163	New Zealand	Oceania

Merge Comparisons

Left Join

Dimensions; 1667 by 8, 2 new cols added.

The number of obersevations would always need to be at least the same as the number of rows in our first dataframe.

If there were missing values on the merge then the original row data will still exist with empty values in the newly added columns in the joined set.

```
## 'data.frame':    1677 obs. of  8 variables:
##  $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ continent: chr  "Asia" "Asia" "Asia" "Asia" ...
##  $ year     : int  2007 2002 1967 1952 1957 1987 1962 1992 1997 1982 ...
##  $ lifeExp  : num  43.8 42.1 34 28.8 30.3 ...
##  $ pop      : int  31889923 25268405 11537966 8425333 9240934 13867957 10267083 16317921 2
2227415 12881816 ...
##  $ gdp      : num  975 727 836 779 821 ...
##  $ long     : num  74.9 74.9 74.9 74.9 74.9 ...
##  $ lat      : num  37.2 37.2 37.2 37.2 37.2 ...
```

Right Join

Dimensions: 1657 by 8, 2 new cols and 10 fewer records.

Number of observations changes because the coords dataset’s country col does not have a match for those 10 records.

There may still be rows with empty values, as in the left join, if they were present in the original coords dataset but not the ccgs dataset

```
## 'data.frame':    1657 obs. of  8 variables:
##  $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ continent: chr  "Asia" "Asia" "Asia" "Asia" ...
##  $ year     : int  2007 2002 1967 1952 1957 1987 1962 1992 1997 1982 ...
##  $ lifeExp  : num  43.8 42.1 34 28.8 30.3 ...
##  $ pop      : int  31889923 25268405 11537966 8425333 9240934 13867957 10267083 16317921 2
2227415 12881816 ...
##  $ gdp      : num  975 727 836 779 821 ...
##  $ long     : num  74.9 74.9 74.9 74.9 74.9 ...
##  $ lat      : num  37.2 37.2 37.2 37.2 37.2 ...
```

Inner Join

Dimensions: 1535 by 8, 2 new cols with 122 fewer rows.

Only data matched by a country that exists in both datasets will be present in this merged set, leading to the reduction of rows present.

```
innerJoin <-  merge(x = ccgs, y = coords, by = "country")
str(innerJoin)
```

```
## 'data.frame':    1535 obs. of  8 variables:
##  $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ continent: chr  "Asia" "Asia" "Asia" "Asia" ...
##  $ year     : int  2007 2002 1967 1952 1957 1987 1962 1992 1997 1982 ...
##  $ lifeExp  : num  43.8 42.1 34 28.8 30.3 ...
##  $ pop      : int  31889923 25268405 11537966 8425333 9240934 13867957 10267083 16317921 2
2227415 12881816 ...
##  $ gdp      : num  975 727 836 779 821 ...
##  $ long     : num  74.9 74.9 74.9 74.9 74.9 ...
##  $ lat      : num  37.2 37.2 37.2 37.2 37.2 ...
```

Geographical Location vs GDP Relationship

GDP Quartiles

The functions appear to be working as expected. The quantiles have been created accordingly, with 50% appearing at the median value. Assignment to gdp.q also appears to be working as intended based on the true values of the gdp per record appearing within the bounds of the appropriate quantile. Additionally, each table has the same number of obesrvations, or about 25% of the total data per quartile.

```
##          0%         25%         50%         75%        100% 
##    241.1659   1191.0260   3614.1013   9341.5210 113523.1329
##    [1] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 3 2 2 3 2 2 2 3 3 4 4 3 4 4 3 3 4 4 3 4 4 3
##   [38] 3 4 4 3 4 4 4 4 4 4 4 3 3 3 3 3 3 2 3 3 3 3 2 4 4 4 4 4 3 3 4 4 4 2 3 3 3
##   [75] 3 2 2 2 3 3 3 3 3 3 2 3 2 2 3 2 2 3 2 2 3 3 3 4 3 4 4 3 3 3 3 4 4 4 4 4 4
##  [112] 4 4 4 4 4 4 4 2 3 2 2 2 2 3 3 2 2 3 3 1 1 2 1 2 2 2 2 1 2 2 4 3 3 4 4 4 4
##  [149] 4 4 4 4 4 3 1 2 2 3 2 3 2 2 3 3 2 3 2 2 2 4 2 3 4 1 4 4 4 4 4 4 4 4 4 4 4
##  [186] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 2 1 1 1 1 1
##  [223] 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 4 4 2 4 4 2 4 4 4 4 3 4
##  [260] 2 2 2 2 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 4 3 4 4 4 4 4 4 4 4 4 4 4 2
##  [297] 2 3 2 2 3 4 2 3 4 1 1 1 1 1 1 1 1 1 1 1 1 3 4 2 4 4 4 3 3 3 3 3 2 1 1 1 2
##  [334] 2 1 1 1 2 2 2 4 4 3 2 3 3 3 3 3 3 3 2 2 2 2 2 2 2 2 2 3 2 2 3 2 1 2 2 2 2
##  [371] 3 2 3 3 3 2 2 2 2 2 2 2 2 2 2 3 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [408] 1 1 2 3 2 2 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 3 4 4 4 4 4 4
##  [445] 4 4 4 4 4 4 4 4 4 4 4 4 3 3 2 2 2 2 2 2 3 3 2 2 2 3 2 2 3 3 1 1 4 3 1 4 3
##  [482] 3 2 3 3 3 3 2 2 3 3 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 1 1 2 2 2 2 2 2 3
##  [519] 3 3 2 3 3 3 3 4 3 3 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 2 1 1 1 1 1 1 4 4 4
##  [556] 2 2 2 3 4 4 4 4 1 1 2 1 1 1 2 2 2 1 2 2 3 3 3 3 2 2 2 4 3 3 3 3 2 2 2 2 3
##  [593] 3 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 2 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 2
##  [630] 2 2 2 2 4 4 4 4 4 4 4 4 4 4 4 4 1 2 1 1 1 1 1 1 1 1 1 1 2 2 1 1 1 1 1 1 2
##  [667] 2 1 1 3 3 3 4 3 3 3 3 4 3 4 3 2 1 1 2 3 1 2 1 1 1 2 1 2 2 3 3 2 3 3 2 3 2
##  [704] 3 3 1 2 2 2 1 2 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1
##  [741] 2 2 2 2 2 3 3 2 3 2 2 3 3 3 4 4 3 3 3 3 4 3 3 4 4 3 3 3 4 4 3 2 3 4 4 4 3
##  [778] 3 3 3 3 3 3 3 3 3 3 3 2 4 4 4 4 3 3 3 3 4 4 4 3 4 3 3 4 3 2 4 4 4 4 4 3 3
##  [815] 3 3 2 2 2 3 3 3 3 3 3 3 3 4 3 4 4 2 3 3 4 3 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4
##  [852] 3 4 3 4 4 4 4 4 4 2 2 2 2 1 1 1 2 2 2 2 2 4 4 4 4 3 3 4 4 4 4 4 4 2 2 2 2
##  [889] 2 2 2 2 2 2 2 2 2 3 3 3 4 4 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 2 2 2 2 2
##  [926] 2 2 2 2 2 2 1 1 1 2 1 2 2 2 2 1 1 1 3 3 2 4 4 2 4 3 4 4 4 4 2 3 3 2 3 2 2
##  [963] 2 2 2 2 2 3 4 4 4 3 4 3 3 4 4 4 4 3 3 3 3 2 3 3 3 3 3 3 3 3 3 2 2 2 2 2 3
## [1000] 2 3 2 3 3 4 4 4 3 4 3 4 4 4 4 1 1 2 1 1 2 2 2 1 1 2 3 3 3 3 3 3 3 3 3 3 3
## [1037] 3 3 4 3 3 4 4 3 4 4 4 4 4 3 3 3 3 3 2 3 3 3 3 2 3 1 1 1 2 1 1 4 1 1 1 3 1
## [1074] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 2 2 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 1 3 2
## [1111] 2 3 3 2 2 2 3 3 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 3 4 4 3 4 4 4 4 4 4 4 3 4 4
## [1148] 4 4 3 4 4 4 4 4 4 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 4 4 4 3
## [1185] 4 4 3 3 4 2 3 2 2 2 2 3 3 3 2 2 2 2 2 2 4 3 3 3 2 4 4 4 4 1 1 1 1 1 1 1 1
## [1222] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 4 4 4 4 4 4 4 4 4 4 4 1 1 1 3 3 3 2 2 2
## [1259] 2 2 3 1 1 1 2 1 1 1 1 1 1 1 1 3 2 3 2 2 3 2 3 3 3 3 3 2 2 2 2 1 1 1 1 1 1
## [1296] 2 1 3 3 3 3 2 3 3 3 3 4 4 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [1333] 1 2 2 2 3 2 2 3 3 2 2 3 3 2 2 3 3 3 3 2 2 2 3 3 3 2 2 2 2 2 2 2 2 2 2 2 2
## [1370] 2 2 2 2 2 2 2 2 2 2 2 2 4 3 3 4 4 4 4 4 4 3 2 3 4 4 4 3 4 4 3 4 4 4 3 3 1
## [1407] 1 1 1 1 1 1 1 1 1 1 1 4 4 4 3 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 3
## [1444] 3 3 3 3 3 3 3 3 3 4 2 2 1 1 1 1 1 2 1 1 1 1 3 4 3 4 3 4 4 4 4 4 4 4 2 2 2
## [1481] 2 2 3 3 3 3 3 2 1 1 1 1 2 2 2 2 2 2 2 2 4 2 4 3 3 4 3 3 3 3 3 1 2 2 2 2 2
## [1518] 2 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
## Levels: 1 2 3 4
```

```
## 
##   1   2   3   4 
## 384 383 384 384
```

Reverse Order

```
##    [1] 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 2 3 3 2 3 3 3 2 2 1 1 2 1 1 2 2 1 1 2 1 1 2
##   [38] 2 1 1 2 1 1 1 1 1 1 1 2 2 2 2 2 2 3 2 2 2 2 3 1 1 1 1 1 2 2 1 1 1 3 2 2 2
##   [75] 2 3 3 3 2 2 2 2 2 2 3 2 3 3 2 3 3 2 3 3 2 2 2 1 2 1 1 2 2 2 2 1 1 1 1 1 1
##  [112] 1 1 1 1 1 1 1 3 2 3 3 3 3 2 2 3 3 2 2 4 4 3 4 3 3 3 3 4 3 3 1 2 2 1 1 1 1
##  [149] 1 1 1 1 1 2 4 3 3 2 3 2 3 3 2 2 3 2 3 3 3 1 3 2 1 4 1 1 1 1 1 1 1 1 1 1 1
##  [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 4 4 4 4 3 4 4 4 4 4
##  [223] 4 4 4 4 4 4 4 3 4 4 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 1 1 3 1 1 3 1 1 1 1 2 1
##  [260] 3 3 3 3 4 4 3 3 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4 1 2 1 1 1 1 1 1 1 1 1 1 1 3
##  [297] 3 2 3 3 2 1 3 2 1 4 4 4 4 4 4 4 4 4 4 4 4 2 1 3 1 1 1 2 2 2 2 2 3 4 4 4 3
##  [334] 3 4 4 4 3 3 3 1 1 2 3 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 2 3 3 2 3 4 3 3 3 3
##  [371] 2 3 2 2 2 3 3 3 3 3 3 3 3 3 3 2 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
##  [408] 4 4 3 2 3 3 2 2 2 2 2 2 2 2 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 2 1 1 1 1 1 1
##  [445] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 3 3 3 3 3 2 2 3 3 3 2 3 3 2 2 4 4 1 2 4 1 2
##  [482] 2 3 2 2 2 2 3 3 2 2 3 4 4 4 4 4 4 4 4 4 4 4 4 3 3 3 4 4 4 4 3 3 3 3 3 3 2
##  [519] 2 2 3 2 2 2 2 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 4 4 4 4 4 3 4 4 4 4 4 4 1 1 1
##  [556] 3 3 3 2 1 1 1 1 4 4 3 4 4 4 3 3 3 4 3 3 2 2 2 2 3 3 3 1 2 2 2 2 3 3 3 3 2
##  [593] 2 3 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 4 3 4 4 4 4 4 4 4 4 4 4 3 3 3 3 3 3 4 3
##  [630] 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 4 3 4 4 4 4 4 4 4 4 4 4 3 3 4 4 4 4 4 4 3
##  [667] 3 4 4 2 2 2 1 2 2 2 2 1 2 1 2 3 4 4 3 2 4 3 4 4 4 3 4 3 3 2 2 3 2 2 3 2 3
##  [704] 2 2 4 3 3 3 4 3 4 3 3 3 4 4 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4
##  [741] 3 3 3 3 3 2 2 3 2 3 3 2 2 2 1 1 2 2 2 2 1 2 2 1 1 2 2 2 1 1 2 3 2 1 1 1 2
##  [778] 2 2 2 2 2 2 2 2 2 2 2 3 1 1 1 1 2 2 2 2 1 1 1 2 1 2 2 1 2 3 1 1 1 1 1 2 2
##  [815] 2 2 3 3 3 2 2 2 2 2 2 2 2 1 2 1 1 3 2 2 1 2 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1
##  [852] 2 1 2 1 1 1 1 1 1 3 3 3 3 4 4 4 3 3 3 3 3 1 1 1 1 2 2 1 1 1 1 1 1 3 3 3 3
##  [889] 3 3 3 3 3 3 3 3 3 2 2 2 1 1 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 3 3 3 3 3
##  [926] 3 3 3 3 3 3 4 4 4 3 4 3 3 3 3 4 4 4 2 2 3 1 1 3 1 2 1 1 1 1 3 2 2 3 2 3 3
##  [963] 3 3 3 3 3 2 1 1 1 2 1 2 2 1 1 1 1 2 2 2 2 3 2 2 2 2 2 2 2 2 2 3 3 3 3 3 2
## [1000] 3 2 3 2 2 1 1 1 2 1 2 1 1 1 1 4 4 3 4 4 3 3 3 4 4 3 2 2 2 2 2 2 2 2 2 2 2
## [1037] 2 2 1 2 2 1 1 2 1 1 1 1 1 2 2 2 2 2 3 2 2 2 2 3 2 4 4 4 3 4 4 1 4 4 4 2 4
## [1074] 4 4 4 4 4 4 4 4 4 4 4 4 4 3 4 4 3 3 3 4 3 3 2 3 3 3 3 3 3 3 3 3 3 3 4 2 3
## [1111] 3 2 2 3 3 3 2 2 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 1
## [1148] 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2
## [1185] 1 1 2 2 1 3 2 3 3 3 3 2 2 2 3 3 3 3 3 3 1 2 2 2 3 1 1 1 1 4 4 4 4 4 4 4 4
## [1222] 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 1 1 1 1 1 1 1 1 1 1 1 4 4 4 2 2 2 3 3 3
## [1259] 3 3 2 4 4 4 3 4 4 4 4 4 4 4 4 2 3 2 3 3 2 3 2 2 2 2 2 3 3 3 3 4 4 4 4 4 4
## [1296] 3 4 2 2 2 2 3 2 2 2 2 1 1 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
## [1333] 4 3 3 3 2 3 3 2 2 3 3 2 2 3 3 2 2 2 2 3 3 3 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3
## [1370] 3 3 3 3 3 3 3 3 3 3 3 3 1 2 2 1 1 1 1 1 1 2 3 2 1 1 1 2 1 1 2 1 1 1 2 2 4
## [1407] 4 4 4 4 4 4 4 4 4 4 4 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
## [1444] 2 2 2 2 2 2 2 2 2 1 3 3 4 4 4 4 4 3 4 4 4 4 2 1 2 1 2 1 1 1 1 1 1 1 3 3 3
## [1481] 3 3 2 2 2 2 2 3 4 4 4 4 3 3 3 3 3 3 3 3 1 3 1 2 2 1 2 2 2 2 2 4 3 3 3 3 3
## [1518] 3 3 3 4 4 3 4 4 4 4 4 4 4 4 4 4 4 4
## Levels: 4 3 2 1
```

```
## 
##   4   3   2   1 
## 384 383 384 384
```

Mean GDP By Quartile

There does not appear to be a significant difference of the average lattitudes between each quartile. With the average distance between being about 3 degrees latitude between all groups, or a ~210 miles.

```
meanLatitude = data.frame("Quartile" = c(), "Mean Latitude" = c())
for (i in unique(levels(innerJoin$gdp.q))){
   levelResults = innerJoin[which(innerJoin$gdp.q == i),]
   meanLat = mean(levelResults$lat)
   meanLatitude = rbind(meanLatitude, list(i, meanLat))
}


knitr::kable(meanLatitude, col.names = c("Quartile", "MeanLatitude"))
```

Quartile	MeanLatitude
1	13.68388
2	19.53061
3	16.54738
4	14.44658