

data_analysis

2023-05-13

Import data

```
pmd <- read.csv("../data/pmd-en.csv")
head(pmd, 3)

##          DBUID DBPOP     DAUID DAPOP    CSDUID    CSDNAME CSDTYPE   CSDPOP CMAUID
## 1 10010165001    160 10010165    506 1001519 St. John's      CY 108,860      1
## 2 10010165002     25 10010165    506 1001519 St. John's      CY 108,860      1
## 3 10010165006    268 10010165    506 1001519 St. John's      CY 108,860      1
##          CMAPUID CMANAME CMATYPE CMAPOP PRUID
## 1    10001 St. John's       B 205,955     10
## 2    10001 St. John's       B 205,955     10
## 3    10001 St. John's       B 205,955     10
##                               PRNAME   PRPOP      lon      lat
## 1 Newfoundland and Labrador / Terre-Neuve-et-Labrador 519,716 -52.7765 47.5300
## 2 Newfoundland and Labrador / Terre-Neuve-et-Labrador 519,716 -52.7793 47.5290
## 3 Newfoundland and Labrador / Terre-Neuve-et-Labrador 519,716 -52.7768 47.5265
##          in_db_emp prox_idx_emp in_db_pharma prox_idx_pharma in_db_childcare
## 1            1        0.0202          0        0.0121          0
## 2            1        0.0193          0        0.014          0
## 3            1        0.0199          1        0.0205          1
##          prox_idx_childcare in_db_health prox_idx_health in_db_grocery
## 1            0.0402          0        0.0069          0
## 2            0.0257          0        0.0028          0
## 3            0.0395          1        0.007          0
##          prox_idx_grocery in_db_educpri prox_idx_educpri in_db_educsec
## 1            ..          0        0.0384          0
## 2            ..          0        0.0562          0
## 3            ..          0        0.0734          0
##          prox_idx_educsec in_db_lib prox_idx_lib in_db_parks prox_idx_parks
## 1            0.0495          0        0.0486          0        0.0141
## 2            0.0375          0          ..          0          ..
## 3            0.0436          0        0.0545          0          ..
##          in_db_transit prox_idx_transit transit_na amenity_dense suppressed
## 1            1        0.0058          0          0          0
## 2            0        0.0046          0          0          0
## 3            1        0.0101          0          0          0
```

DATA Summary

```

# no. of rows and columns
dim(pmd)

## [1] 489676      41

489676 rows and 41 columns

# dataset summary
str(pmd)

## 'data.frame': 489676 obs. of 41 variables:
## $ DBUID           : num  1e+10 1e+10 1e+10 1e+10 1e+10 ...
## $ DBPOP           : chr  "160" "25" "268" "53" ...
## $ DAUID           : int  10010165 10010165 10010165 10010165 10010166 10010166 10010166 10010167 ...
## $ DAPOP           : chr  "506" "506" "506" "506" ...
## $ CSDUID          : int  1001519 1001519 1001519 1001519 1001519 1001519 1001519 1001519 ...
## $ CSDNAME         : chr  "St. John's" "St. John's" "St. John's" "St. John's" ...
## $ CSCTYPE          : chr  "CY" "CY" "CY" "CY" ...
## $ CSDPOP          : chr  "108,860" "108,860" "108,860" "108,860" ...
## $ CMAUID          : int  1 1 1 1 1 1 1 1 1 ...
## $ CMAPUID          : int  10001 10001 10001 10001 10001 10001 10001 10001 ...
## $ CMANAME          : chr  "St. John's" "St. John's" "St. John's" "St. John's" ...
## $ CMATYPE          : chr  "B" "B" "B" "B" ...
## $ CMAPOP           : chr  "205,955" "205,955" "205,955" "205,955" ...
## $ PRUID            : int  10 10 10 10 10 10 10 10 10 ...
## $ PRNAME           : chr  "Newfoundland and Labrador / Terre-Neuve-et-Labrador" "Newfoundland and Labrador / Terre-Neuve-et-Labrador" ...
## $ PRPOP             : chr  "519,716" "519,716" "519,716" "519,716" ...
## $ lon               : num  -52.8 -52.8 -52.8 -52.8 -52.8 ...
## $ lat               : num  47.5 47.5 47.5 47.5 47.5 ...
## $ in_db_emp         : chr  "1" "1" "1" "1" ...
## $ prox_idx_emp       : chr  "0.0202" "0.0193" "0.0199" "0.0204" ...
## $ in_db_pharma       : chr  "0" "0" "1" "0" ...
## $ prox_idx_pharma     : chr  "0.0121" "0.014" "0.0205" "0.0238" ...
## $ in_db_childcare     : chr  "0" "0" "1" "0" ...
## $ prox_idx_childcare  : chr  "0.0402" "0.0257" "0.0395" "0.0425" ...
## $ in_db_health        : chr  "0" "0" "1" "0" ...
## $ prox_idx_health      : chr  "0.0069" "0.0028" "0.007" "0.0074" ...
## $ in_db_grocery        : chr  "0" "0" "0" "0" ...
## $ prox_idx_grocery      : chr  ".." ".." ".." ".." ...
## $ in_db_educpri        : chr  "0" "0" "0" "0" ...
## $ prox_idx_educpri      : chr  "0.0384" "0.0562" "0.0734" "0.0733" ...
## $ in_db_educsec        : chr  "0" "0" "0" "0" ...
## $ prox_idx_educsec      : chr  "0.0495" "0.0375" "0.0436" "0.0548" ...
## $ in_db_lib             : chr  "0" "0" "0" "0" ...
## $ prox_idx_lib           : chr  "0.0486" ".." "0.0545" "0.0796" ...
## $ in_db_parks           : chr  "0" "0" "0" "0" ...
## $ prox_idx_parks         : chr  "0.0141" ".." ".." "0.013" ...
## $ in_db_transit         : chr  "1" "0" "1" "0" ...
## $ prox_idx_transit       : chr  "0.0058" "0.0046" "0.0101" "0.0098" ...
## $ transit_na             : int  0 0 0 0 0 0 0 0 0 ...
## $ amenity_dense          : chr  "0" "0" "0" "0" ...
## $ suppressed            : int  0 0 0 0 0 0 0 0 0 ...

```

```

unique(pmd$PRNAME)

## [1] "Newfoundland and Labrador / Terre-Neuve-et-Labrador"
## [2] "Prince Edward Island / Île-du-Prince-Édouard"
## [3] "Nova Scotia / Nouvelle-Écosse"
## [4] "New Brunswick / Nouveau-Brunswick"
## [5] "Quebec / Québec"
## [6] "Ontario"
## [7] "Manitoba"
## [8] "Saskatchewan"
## [9] "Alberta"
## [10] "British Columbia / Colombie-Britannique"
## [11] "Yukon"
## [12] "Northwest Territories / Territoires du Nord-Ouest"
## [13] "Nunavut"

pmd$PRNAME <- gsub("Newfoundland and Labrador / Terre-Neuve-et-Labrador", "Newfoundland and Labrador", pmd$PRNAME)
pmd$PRNAME <- gsub("Prince Edward Island / Île-du-Prince-Édouard", "Prince Edward Island", pmd$PRNAME)
pmd$PRNAME <- gsub("Nova Scotia / Nouvelle-Écosse", "Nova Scotia", pmd$PRNAME)
pmd$PRNAME <- gsub("New Brunswick / Nouveau-Brunswick", "New Brunswick", pmd$PRNAME)
pmd$PRNAME <- gsub("Quebec / Québec", "Quebec", pmd$PRNAME)
pmd$PRNAME <- gsub("British Columbia / Colombie-Britannique", "British Columbia", pmd$PRNAME)
pmd$PRNAME <- gsub("Northwest Territories / Territoires du Nord-Ouest", "Northwest Territories", pmd$PRNAME)

# percentage of missing values in each column in dataset
p <- function(x) {sum(is.na(x))/length(x)*100}
sort(apply(pmd, 2, p), decreasing = TRUE) # marmin = 2 means function will be applied in each column

##          CMAUID        CMAPUID        DBUID        DBPOP
##        43.48059      43.48059      0.00000      0.00000
##          DAUID        DAPOP        CSDUID        CSDNAME
##        0.00000      0.00000      0.00000      0.00000
##          CSCTYPE       CSDPOP       CMANAME       CMATYPE
##        0.00000      0.00000      0.00000      0.00000
##          CMAPOP       PRUID        PRNAME        PRPOP
##        0.00000      0.00000      0.00000      0.00000
##          lon          lat        in_db_emp    prox_idx_emp
##        0.00000      0.00000      0.00000      0.00000
##          in_db_pharma prox_idx_pharma in_db_childcare prox_idx_childcare
##        0.00000      0.00000      0.00000      0.00000
##          in_db_health prox_idx_health in_db_grocery   prox_idx_grocery
##        0.00000      0.00000      0.00000      0.00000
##          in_db_educpri prox_idx_educpri in_db_educsec   prox_idx_educsec
##        0.00000      0.00000      0.00000      0.00000
##          in_db_lib     prox_idx_lib     in_db_parks   prox_idx_parks
##        0.00000      0.00000      0.00000      0.00000
##          in_db_transit prox_idx_transit transit_na    amenity_dense
##        0.00000      0.00000      0.00000      0.00000
##          suppressed
##        0.00000

```

In head of dataset we saw there were missing values in `prox_idx_lib` but the above output suggest there is no missing values. Because Statistics Canada use some specific notation for missing values. The following

standard symbols are used in Statistics Canada publications:

..> not available for a specific reference period

F-> to unreliable to be published

```
# percentage of missing values in each column
p <- function(x) {sum(x == ".." | x == "F")/length(x)*100}
sort(apply(pmd, 2, p), decreasing = TRUE)
```

```
##      prox_idx_lib    prox_idx_grocery    prox_idx_educsec    prox_idx_pharma
##      76.993972        71.192584        71.161952        63.543037
##      prox_idx_transit    prox_idx_educpri    prox_idx_parks    prox_idx_childcare
##      62.974497        53.977936        52.199413        50.178485
##      prox_idx_health    prox_idx_emp       in_db_emp       in_db_pharma
##      38.640040        13.493412        1.096031        1.096031
##      in_db_childcare    in_db_health    in_db_grocery    in_db_educpri
##      1.096031         1.096031         1.096031         1.096031
##      in_db_educsec     in_db_lib       in_db_parks     in_db_transit
##      1.096031         1.096031         1.096031         1.096031
##      amenity_dense      DBUID          DBPOP          DAUID
##      1.096031         0.000000         0.000000         0.000000
##      DAPOP            CSDUID          CSDNAME        CSDTYPE
##      0.000000         0.000000         0.000000         0.000000
##      CSDPOP            CMANAME          CMATYPE        CMAPOP
##      0.000000         0.000000         0.000000         0.000000
##      PRUID             PRNAME           PRPOP          lon
##      0.000000         0.000000         0.000000         0.000000
##      lat                transit_na    suppressed
##      0.000000         0.000000         0.000000
```

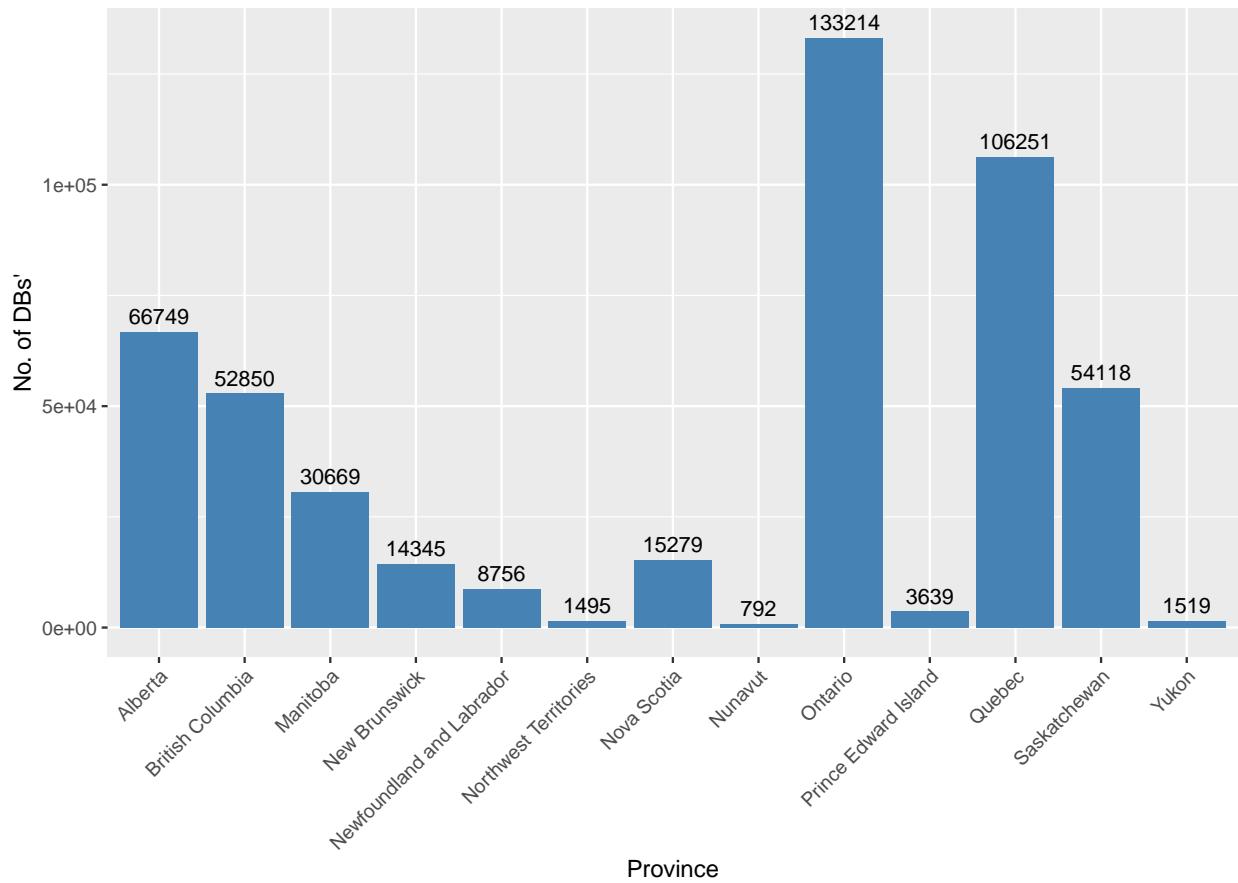
```
pmd[pmd == ".." | pmd == "F"] <- NA
```

Count the dissemination block per province.

```
# Count the number dbs' that fall under each province
db_counts <- pmd %>% count(prov = pmd$PRNAME)

# Create bar chart
ggplot(db_counts, aes(x=prov, y=n)) +
  geom_bar(stat="identity", fill="steelblue") +
  geom_text(aes(label=n), vjust=-0.5, color="black", size=3.5) +
  labs(title = "Number of DBs' by Province",
       x = "Province",
       y = "No. of DBs'") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Number of DBs' by Province



```
# summary of the dataset
# sapply(pmd, function(x) if(is.numeric(x)) summary(x))
sapply(Filter(is.numeric, pmd), summary)
```

```
## $DBPOP
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.    NA's
##      0.00   5.00  29.00    71.83  81.00 7607.00      315
##
## $DAPOP
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.    NA's
##      0.0  431.0  523.0    727.8  675.0 22077.0      315
##
## $CSDPOP
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.    NA's
##      0  2559  13678  235937  134413 2731571      315
##
## $CMAPOP
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.    NA's
##  10741 103811 747545 1670003 2463431 5928040 212914
##
## $PRPOP
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 35874 1278365 4648055 6748994 13448494 13448494
```

```

## 
## $lon
##   Min. 1st Qu. Median   Mean 3rd Qu. Max.
## -140.91 -110.03 -80.53 -90.30 -73.78 -52.66
##
## $lat
##   Min. 1st Qu. Median   Mean 3rd Qu. Max.
## 41.75 45.29 47.35 48.00 50.49 82.50
##
## $prox_idx_emp
##   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## 0.00 0.00 0.01 0.03 0.03 1.00 66074
##
## $prox_idx_pharma
##   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## 0.00 0.01 0.03 0.04 0.05 1.00 311155
##
## $prox_idx_childcare
##   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## 0.00 0.02 0.05 0.08 0.10 1.00 245712
##
## $prox_idx_health
##   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## 0.00 0.00 0.00 0.01 0.01 1.00 189211
##
## $prox_idx_grocery
##   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## 0.0 0.0 0.0 0.1 0.1 1.0 348613
##
## $prox_idx_educpri
##   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## 0.00 0.05 0.09 0.12 0.15 1.00 264317
##
## $prox_idx_educsec
##   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## 0.0 0.0 0.1 0.1 0.1 1.0 348463
##
## $prox_idx_lib
##   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## 0.0 0.1 0.1 0.1 0.1 1.0 377021
##
## $prox_idx_parks
##   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## 0.00 0.02 0.05 0.07 0.09 1.00 255608
##
## $prox_idx_transit
##   Min. 1st Qu. Median   Mean 3rd Qu. Max. NA's
## 0.00 0.00 0.01 0.02 0.02 1.00 308371

# Subset columns that start with "prox_idx"
prox_cols <- colnames(pmd)[grep("prox_idx", colnames(pmd))]

# Check if there are any dbs where all proximity measures are missing
all_prox_na <- pmd[rowSums(is.na(pmd[,prox_cols])) == length(prox_cols), ]

```

```

nrow(all_prox_na)

## [1] 64764

head(all_prox_na[prox_cols], 3)

##      prox_idx_emp prox_idx_pharma prox_idx_childcare prox_idx_health
## 300          NA            NA            NA            NA
## 768          NA            NA            NA            NA
## 779          NA            NA            NA            NA
##      prox_idx_grocery prox_idx_educpri prox_idx_educsec prox_idx_lib
## 300           NA            NA            NA            NA
## 768           NA            NA            NA            NA
## 779           NA            NA            NA            NA
##      prox_idx_parks prox_idx_transit
## 300           NA            NA
## 768           NA            NA
## 779           NA            NA

```

So, there are 64764 dissemination blocks where none of the proximity measures are available. Let's check the population of those dissemination blocks.

```

unique(all_prox_na$DBPOP) [1:5]

## [1] 0 33 78 30 19

sort(unique(all_prox_na$DBPOP), decreasing = TRUE) [1:5]

## [1] 1858 1522 1501 1404 1265

```

When the DBPOP is 0, it is understandable that there are no proximity measures because they might not have been calculated. However, even for DB populations with large values, there are missing values for proximity measures, which is not expected.

```

unique(all_prox_na$PRNAME) [1:5]

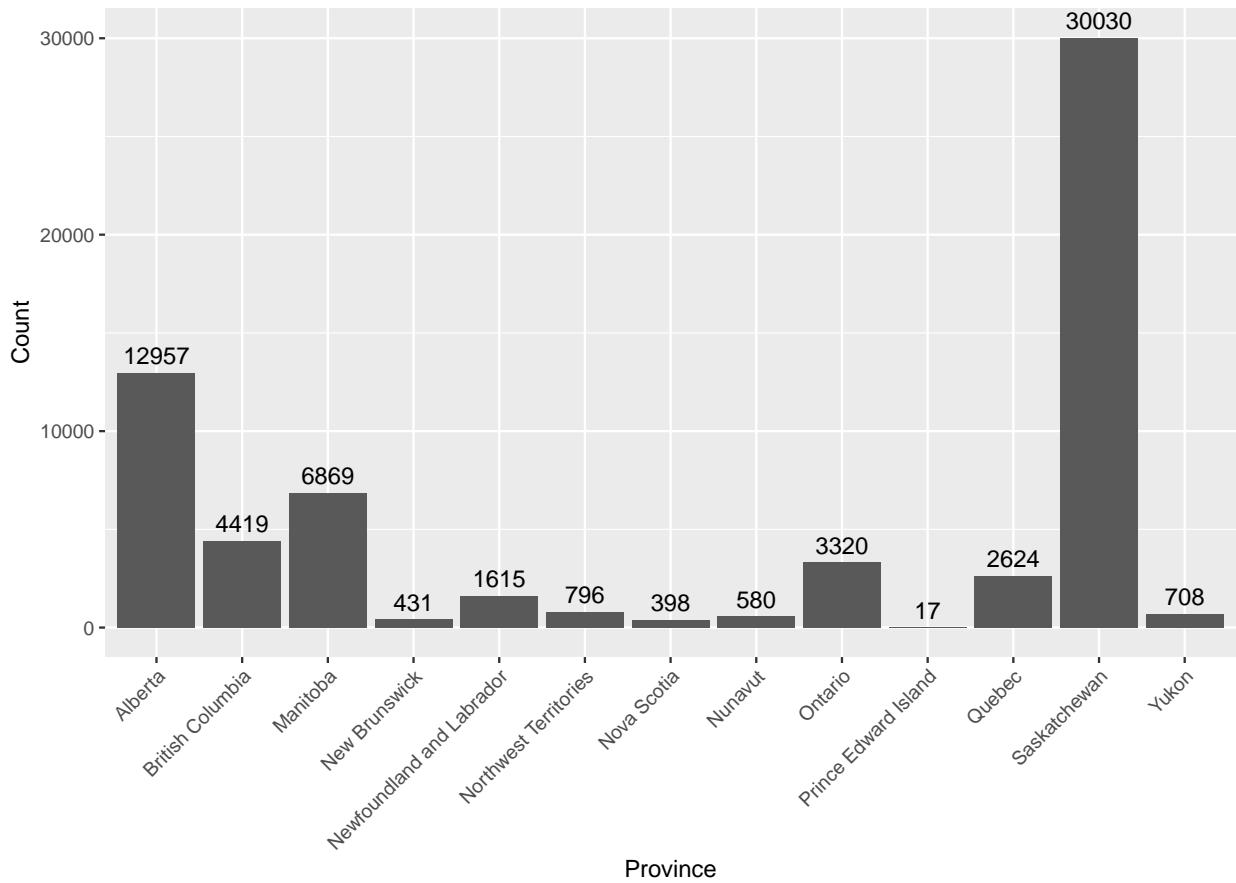
## [1] "Newfoundland and Labrador" "Prince Edward Island"
## [3] "Nova Scotia"                 "New Brunswick"
## [5] "Quebec"

# Count the occurrences of each CSDTYPE value
pr_counts <- data.frame(table(all_prox_na$PRNAME))

# Create a bar plot
ggplot(data = pr_counts,
       aes(x = Var1, y = Freq)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = Freq), vjust = -0.5) +
  labs(title = "Number of DBs' by Province where all PMS NA") +
  xlab("Province") +
  ylab("Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

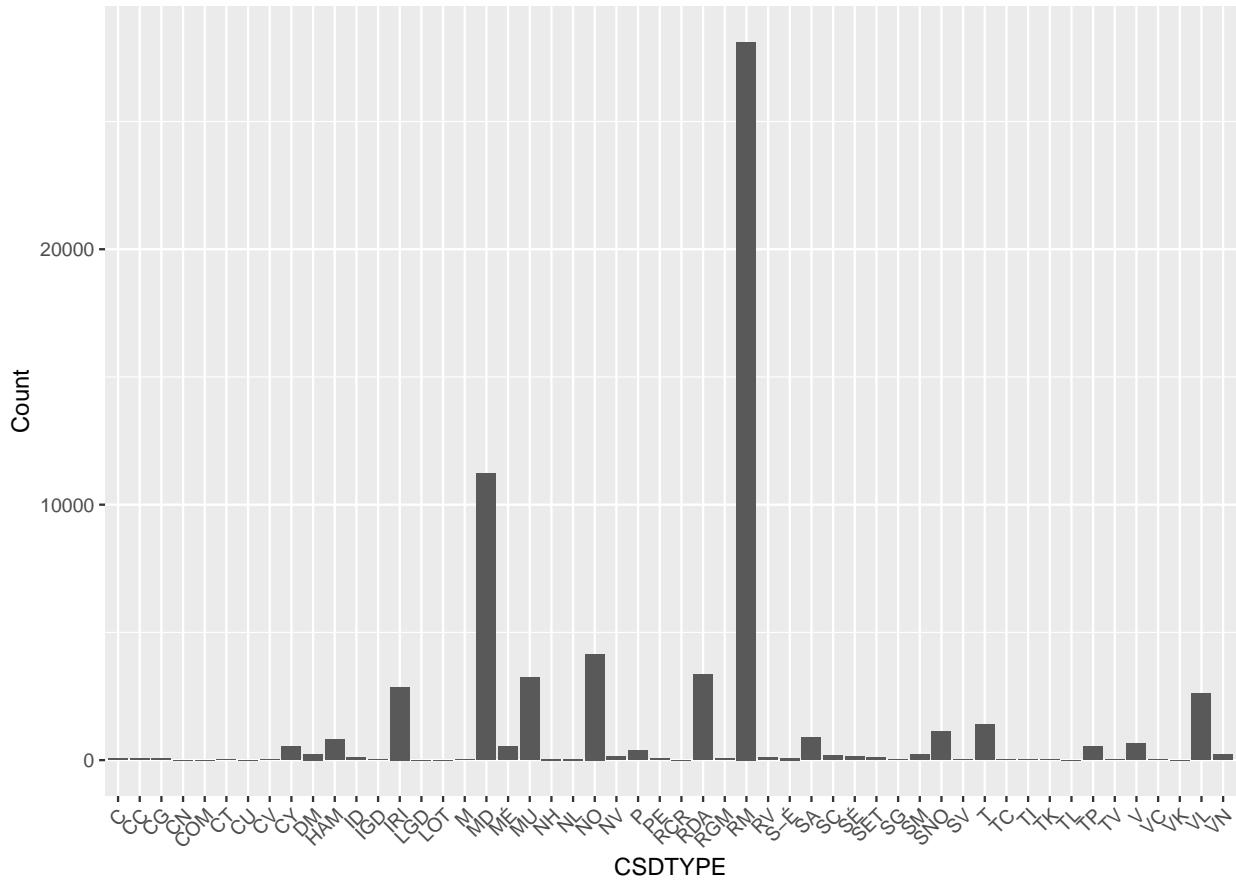
Number of DBs' by Province where all PMS NA



```
# Count the occurrences of each CSDTYPE value
csdtype_counts <- data.frame(table(all_prox_na$CSDTYPE))

# Create a bar plot
ggplot(data = csdtype_counts,
       aes(x = Var1, y = Freq)) +
  geom_bar(stat = "identity") +
  # geom_text(aes(label = Freq), vjust = -0.5) +
  labs(title = "Number of DBs' by CSDTYPE where all PMS NA") +
  xlab("CSDTYPE") +
  ylab("Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Number of DBs' by CSDTYPE where all PMS NA



We can see that These DBs are from different province and different csd. If we can plot them on map then it may make sense! Majority portions of db where all proximity measures are null from Regional municipality and Municipal district.

```
# Check if there are any dbs where all proximity measures are 0
all_prox_0 <- pmd[rowSums(pmd[, prox_cols] == 0, na.rm = TRUE) == length(prox_cols), ]
nrow(all_prox_0)
```

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

There are no dissemination blocks for which all the proximity measures of amenities are 0. So, dissemination blocks with no populations still has the proximity measures of amenities may be those dissemination blocks are close from other populated dissemination blocks, or the dissemination blocks contain parks, office buildings, industrial area etc.

Outliers

Outliers can have a significant impact on the clustering results by pulling the centroids towards themselves, creating biased clusters, and reducing the effectiveness of the clustering algorithm.

```

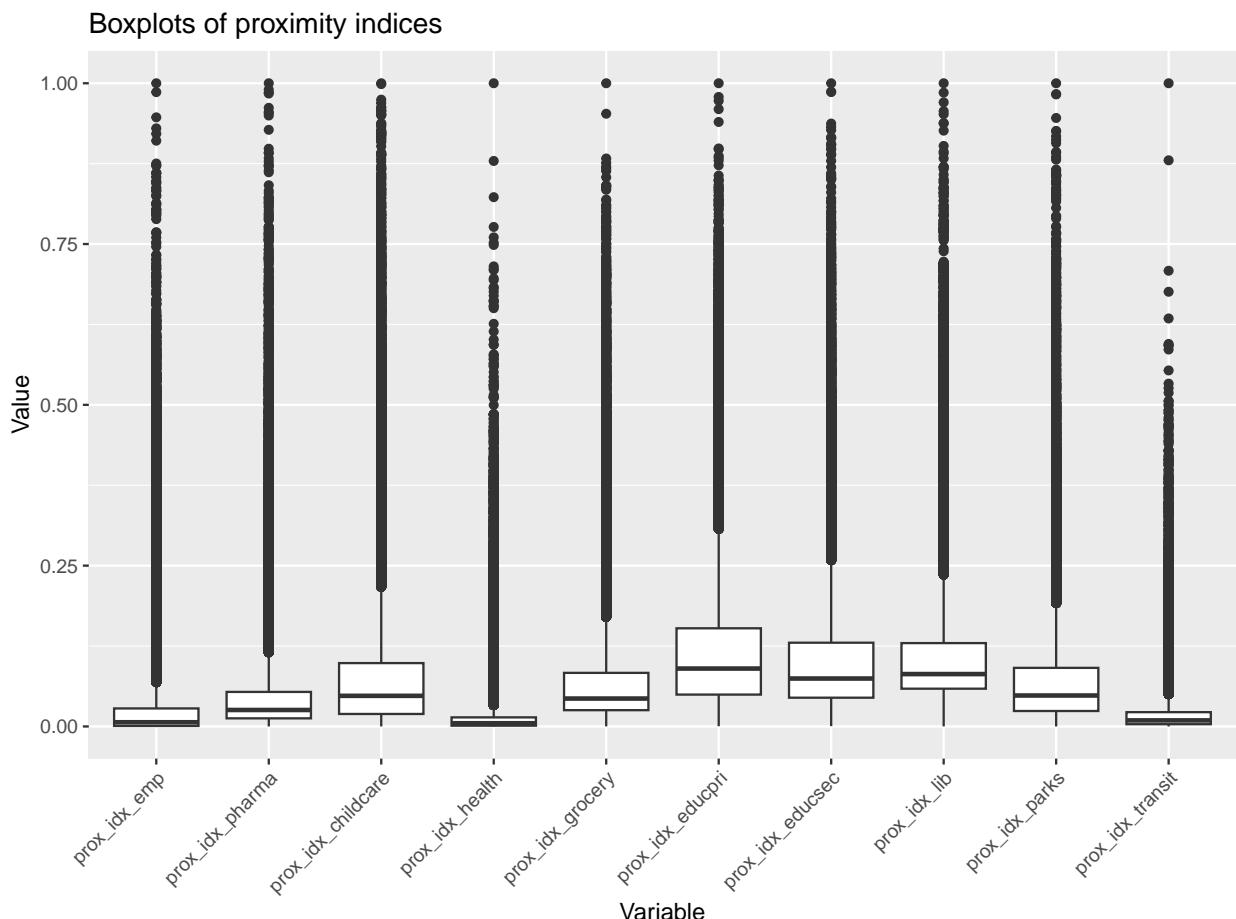
# boxplot(pmd$prox_idx_educsec,
#   ylab = "prox_idx_emp"
# )

# Create a melted data frame for the boxplot
df <- reshape2::melt(pmd[, prox_cols])

## No id variables; using all as measure variables

# Create the boxplot
ggplot(df, aes(x = variable, y = value)) +
  geom_boxplot() +
  xlab("Variable") +
  ylab("Value") +
  ggtitle("Boxplots of proximity indices") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

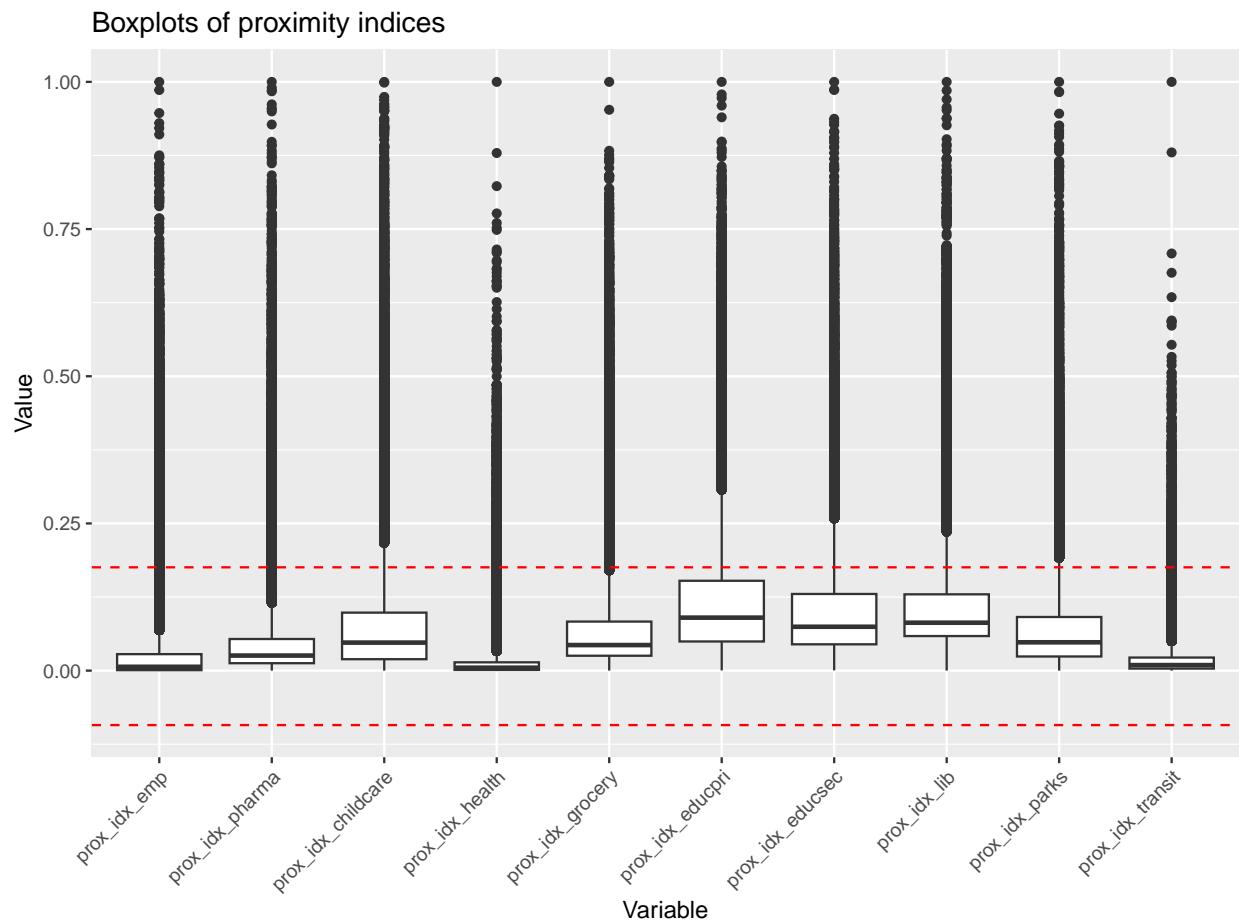


```

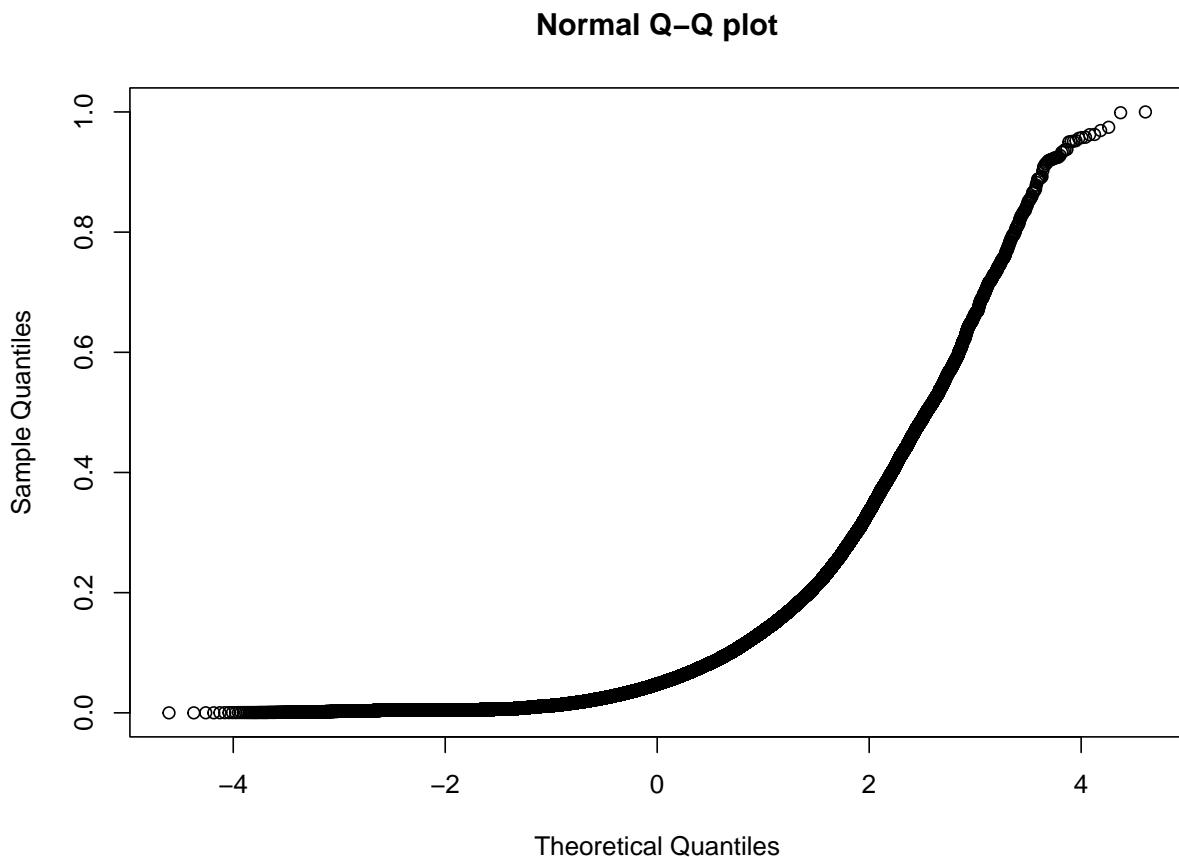
# Calculate Q1 and IQR
Q1 <- quantile(df$value, 0.25, na.rm = TRUE)
Q3 <- quantile(df$value, 0.75, na.rm = TRUE)
IQR <- IQR(df$value, na.rm = TRUE)

```

```
# Create the boxplot with the horizontal line
ggplot(df, aes(x = variable, y = value)) +
  geom_boxplot() +
  geom_hline(yintercept = Q1 - 1.5*IQR, linetype = "dashed", color = "red") +
  geom_hline(yintercept = Q3 + 1.5*IQR, linetype = "dashed", color = "red") +
  xlab("Variable") +
  ylab("Value") +
  ggtitle("Boxplots of proximity indices") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
qqnorm(pmd$prox_idx_childcare, main = "Normal Q-Q plot")
```



It's hard to detect the outliers from Normal Q-Q plot.

Rosner's test for Outliers Let's do the Rosner's test.

Rosner's test for multiple outliers is used by VSP to detect up to 10 outliers among the selected data values. This test will detect outliers that are either much smaller or much larger than the rest of the data. Rosner's approach is designed to avoid the problem of masking, where an outlier that is close in value to another outlier can go undetected.

Rosner's test is appropriate only when the data, excluding the suspected outliers, are approximately normally distributed, and when the sample size is greater than or equal to 25. (https://vsp.pnnl.gov/help/vsample/rosners_outlier_test.htm)

```
# Rosner's test
library(EnvStats)

##
## Attaching package: 'EnvStats'

## The following object is masked from 'package:terra':
##      predict
```

```

## The following objects are masked from 'package:stats':
##
##     predict, predict.lm

## The following object is masked from 'package:base':
##
##     print.default

test <- rosnerTest(pmd$prox_idx_emp, k = 5000, warn = TRUE)

```

```
test$all.stats
```

```
length(test$all.stats$Outlier == TRUE)
```

```
## [1] 5000
```

```

library(outliers)
chisq.out.test(pmd$prox_idx_emp)

```

Chi-squared test for Outliers

```

##
## chi-squared test for outlier
##
## data: pmd$prox_idx_emp
## X-squared = 394.08, p-value < 2.2e-16
## alternative hypothesis: highest value 1 is an outlier

```

So the max proximity measure of grocery is an outlier.

```
chisq.out.test(pmd$prox_idx_emp, opposite = TRUE)
```

```

##
## chi-squared test for outlier
##
## data: pmd$prox_idx_emp
## X-squared = 0.26787, p-value = 0.6048
## alternative hypothesis: lowest value 0 is an outlier

```

So the min proximity measure of grocery is an outlier.

The proximity measures are already normalized but we can still see outliers in these. So, we should use clustering algorithms that can handle outliers.

For example DBSCAN clustering is robust against outliers when we choose minimum number of points (minPts) - (a threshold) large enough.

Ordering points to identify the clustering structure (OPTICS) is an algorithm for finding density-based clusters in spatial data which is also robust against outliers.

https://en.wikipedia.org/wiki/OPTICS_algorithm#cite_note-1 But we can't use general k-means: the

squared error approach is sensitive to outliers. But there are variants such as k-medians for handling outliers.

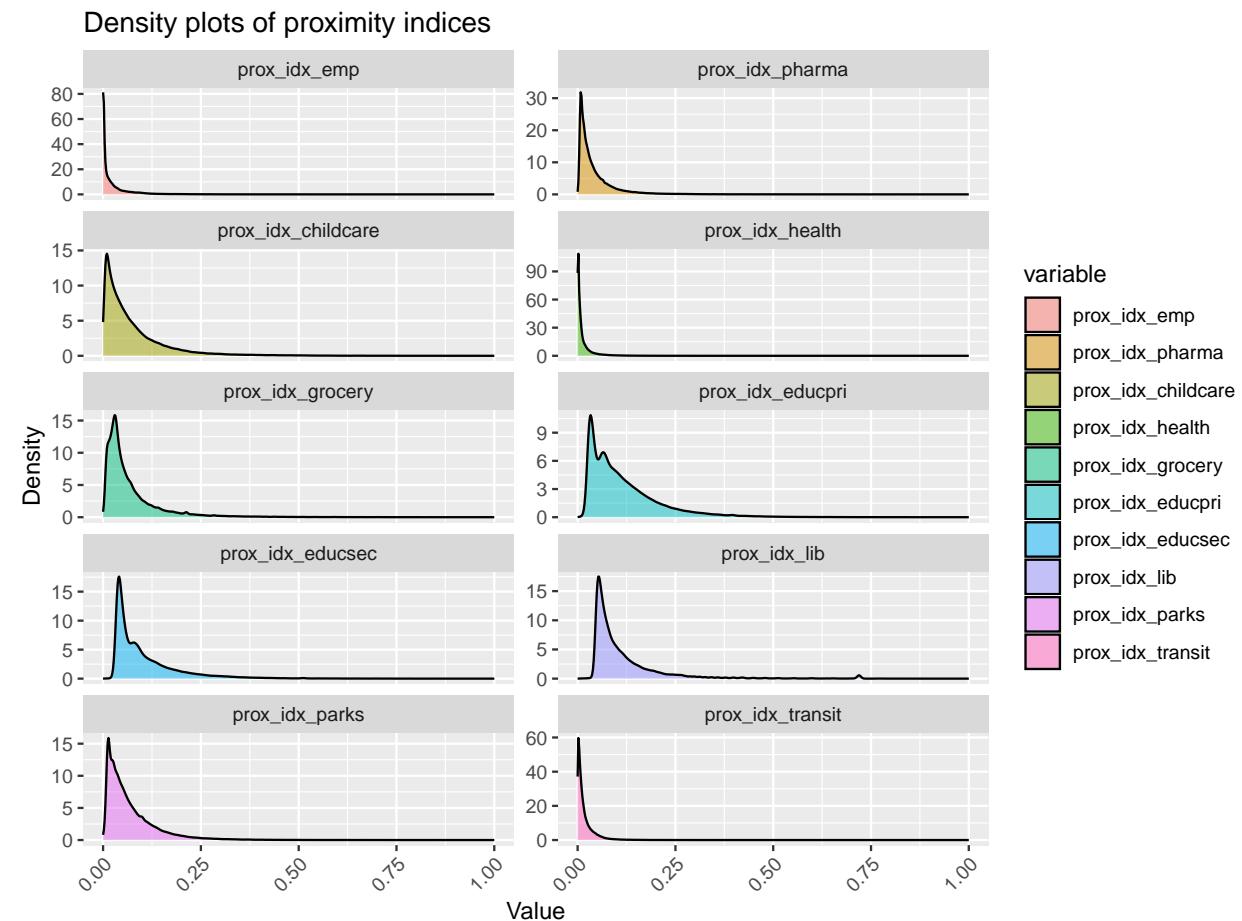
(https://www.researchgate.net/publication/220490566_A_review_of_robust_clustering_methods)

Another approach is to apply a transformation to the data that can reduce the impact of outliers. For example, we could apply a log transformation or a power transformation to the data. These transformations can help to reduce the influence of extreme values and make the data more symmetric.

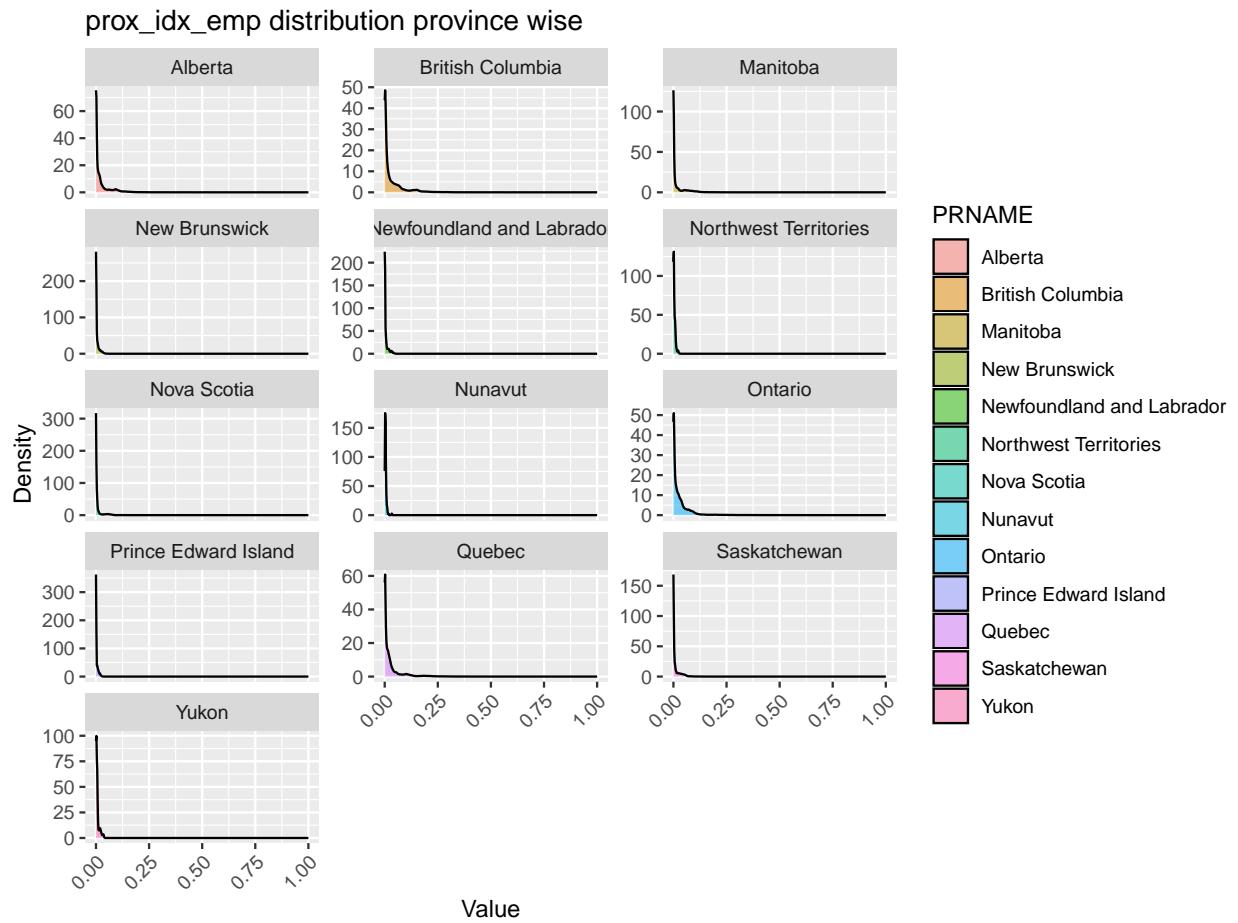
Distributions

```
# plot(density(pmd$prox_idx_health, na.rm = TRUE))

# Create the density plot
ggplot(df, aes(x = value, fill = variable)) +
  geom_density(alpha = 0.5) +
  xlab("Value") +
  ylab("Density") +
  ggtitle("Density plots of proximity indices") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~variable, scales = 'free_y', nrow = 5)
```

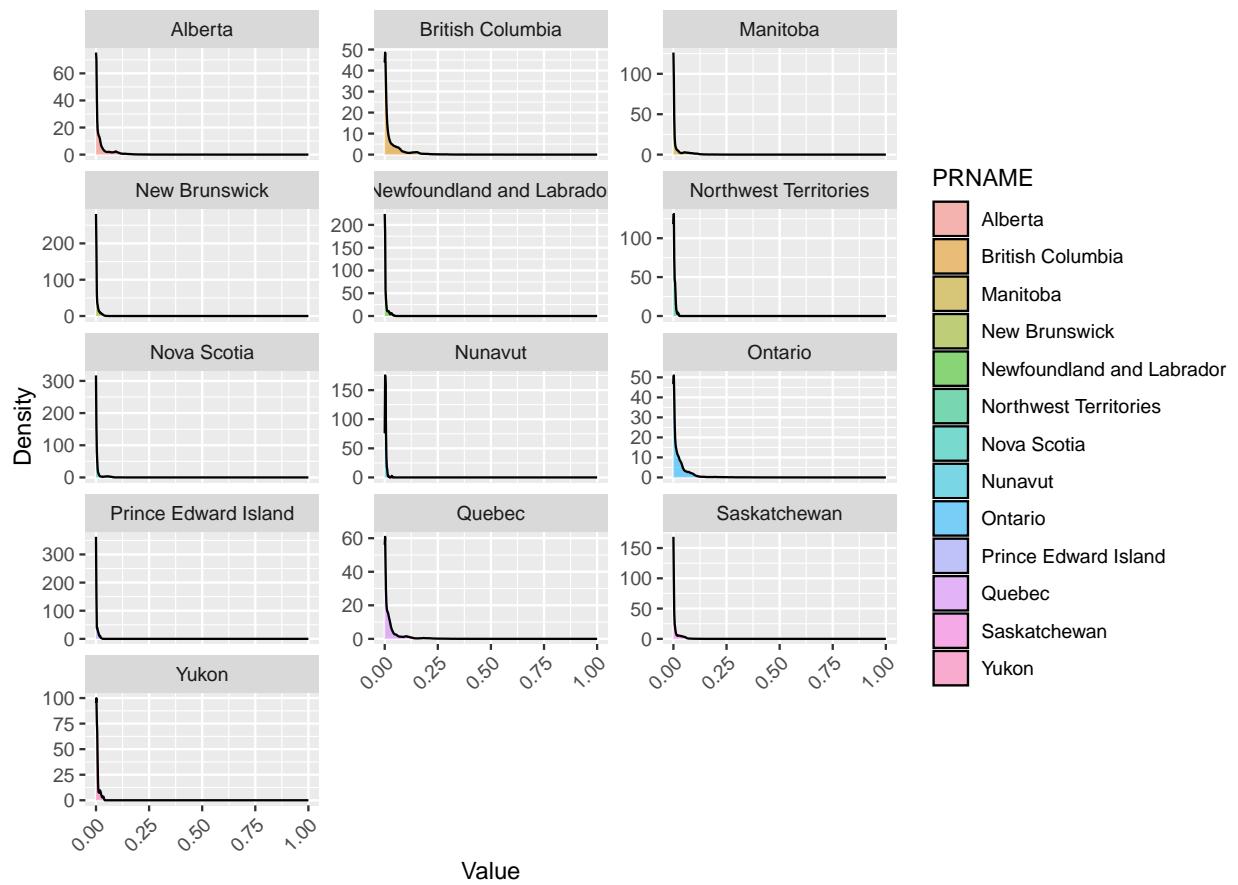


```
# prox_idx_emp distribution province wise
ggplot(pmd, aes(x = prox_idx_emp, fill = PRNAME)) +
  geom_density(alpha = 0.5) +
  xlab("Value") +
  ylab("Density") +
  ggtitle("prox_idx_emp distribution province wise") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~PRNAME, scales = 'free_y', nrow = 5)
```



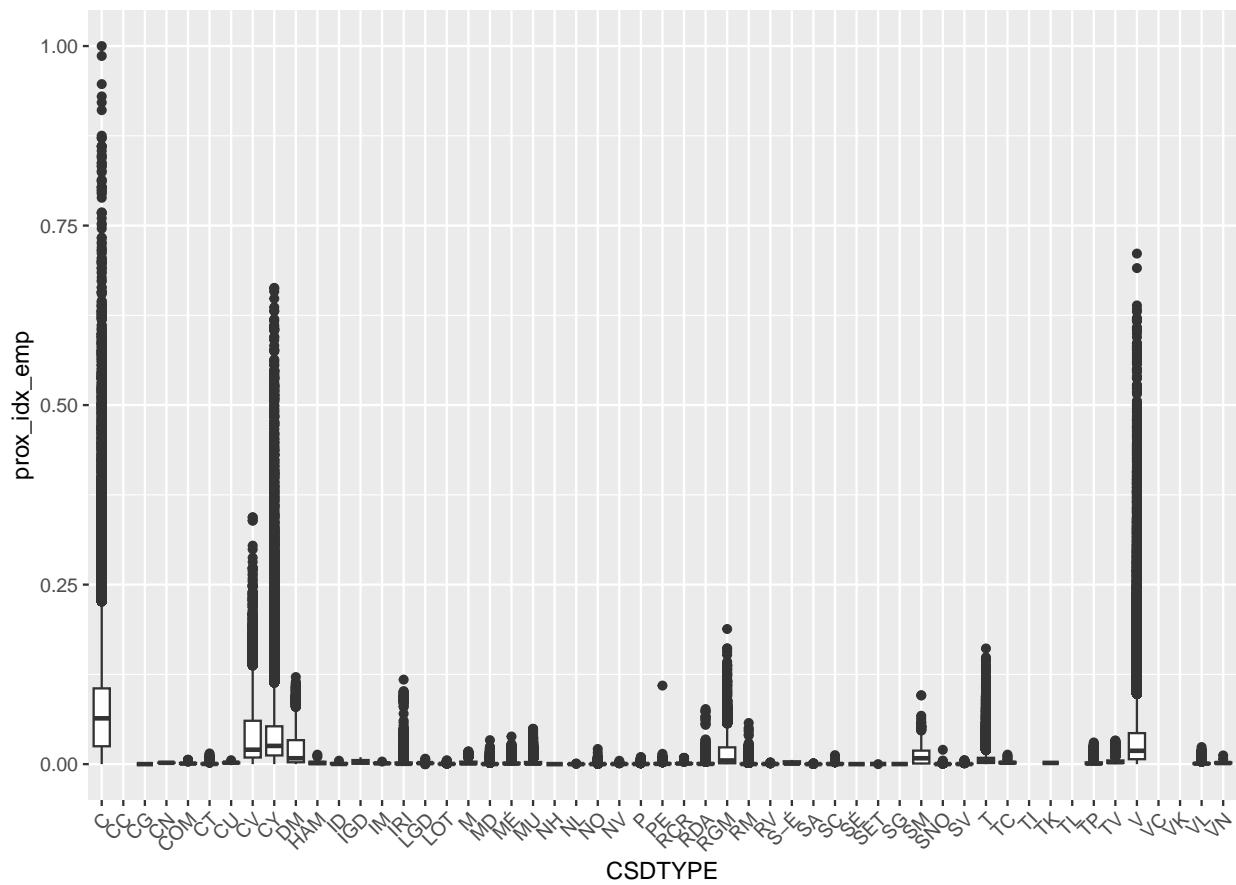
```
# prox_idx_health distribution province wise
ggplot(pmd, aes(x = prox_idx_emp, fill = PRNAME)) +
  geom_density(alpha = 0.5) +
  xlab("Value") +
  ylab("Density") +
  ggtitle("prox_idx_health distribution province wise") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  facet_wrap(~PRNAME, scales = 'free_y', nrow = 5)
```

prox_idx_health distribution province wise



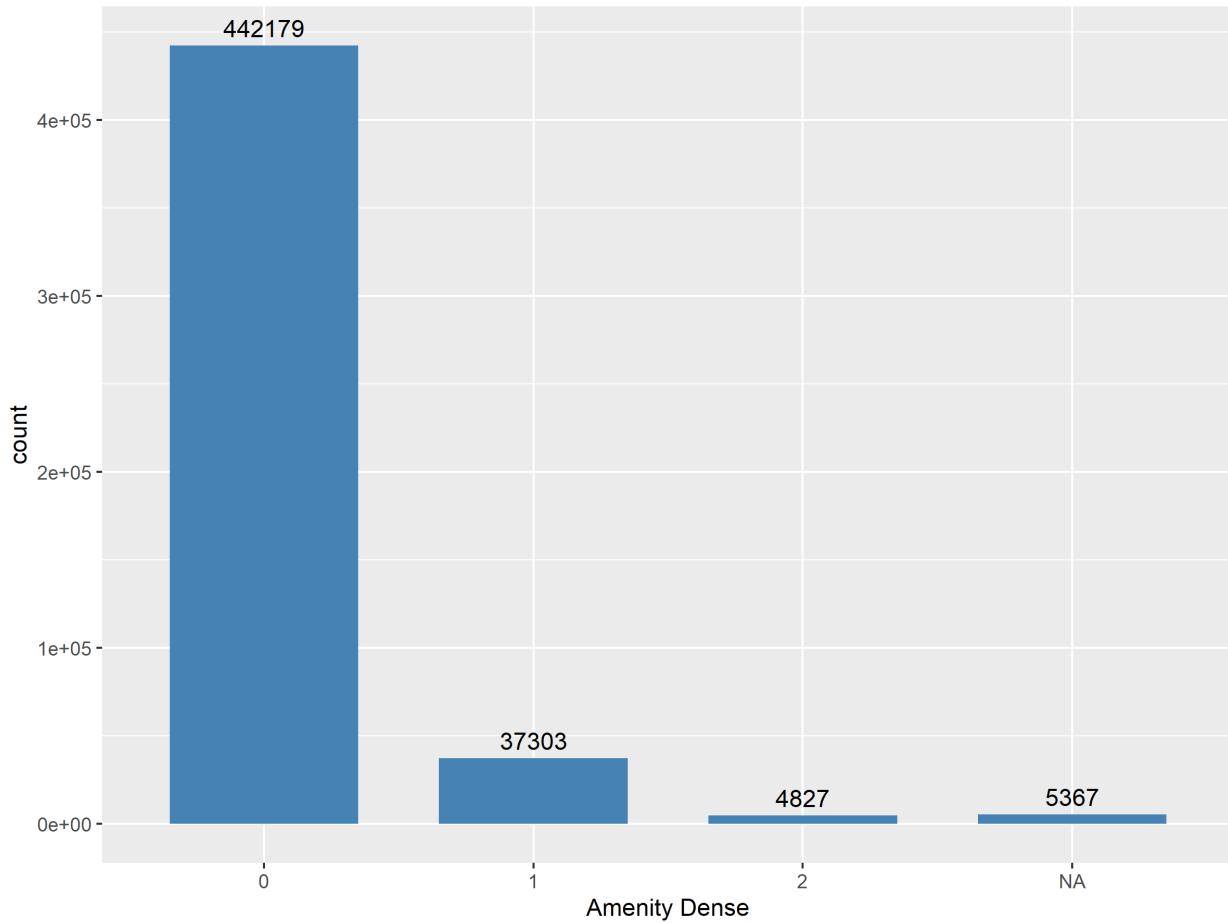
```
# prox_idx_emp distribution csdtype wise
ggplot(data = pmd, aes(x = CSDTYPE, y = prox_idx_emp)) +
  geom_boxplot() +
  xlab("CSDTYPE") +
  ylab("prox_idx_emp") +
  ggtitle("prox_idx_emp distribution csdtype wise") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

prox_idx_emp distribution csdtype wise



Density of Categorical and binary Variables

```
ggplot(pmd, aes(x=as.factor(amenity_dense) )) +
  geom_bar(width=0.7, fill="steelblue") +
  geom_text(stat="count", aes(label=..count..), vjust=-0.5) +
  labs(x = "Amenity Dense")
```



A dissemination block is an amenity dense neighbourhood, a high amenity density neighbourhood, a non-amenity dense neighbourhood.

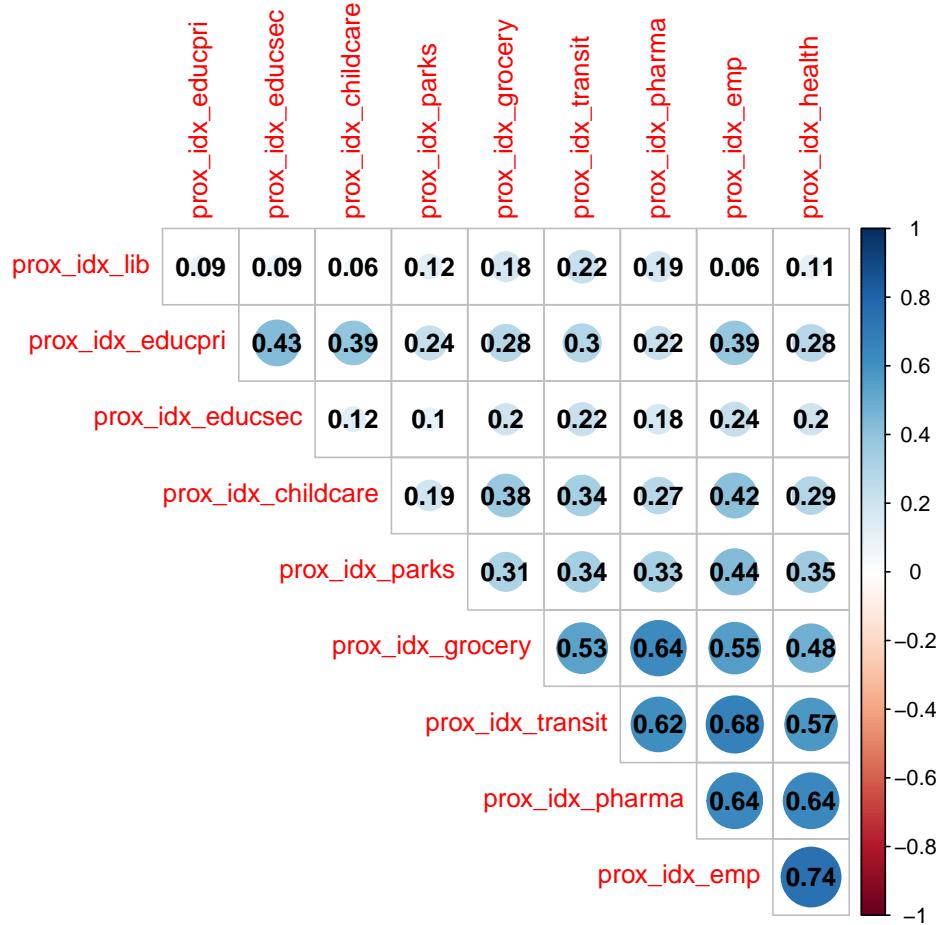
Covariate

Correlation

```
library(corrplot)

## corrplot 0.92 loaded

M <- cor(pmd$prox_cols, use = "pairwise.complete.obs")
corrplot(M, type = "upper", order = "hclust",
         addCoef.col = "black",
         diag = FALSE)
```



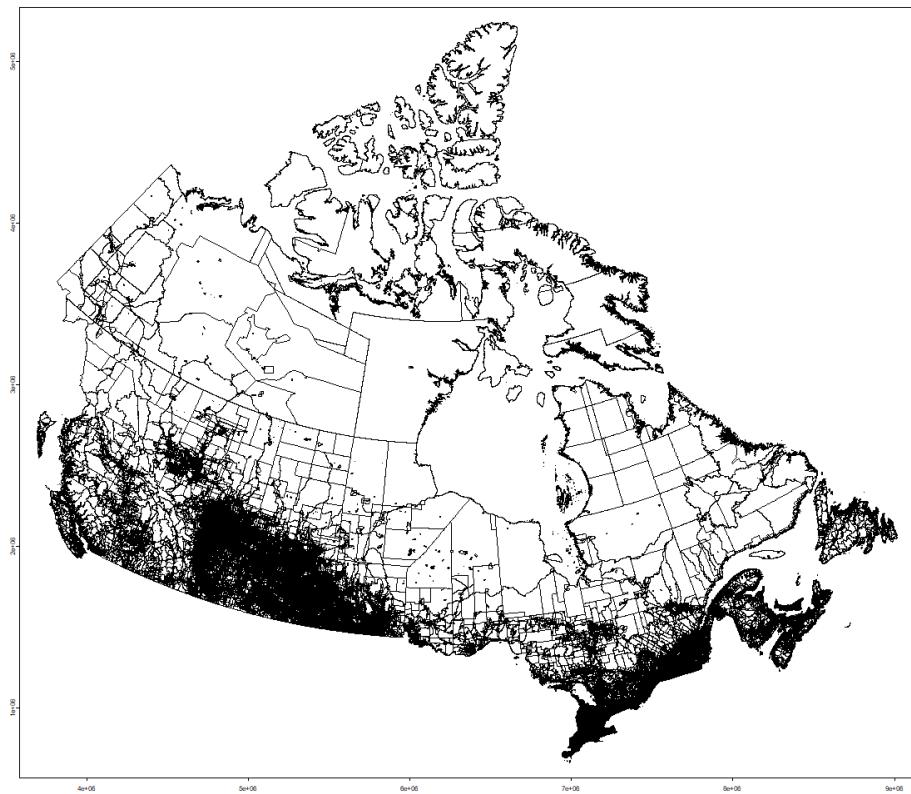
Spat

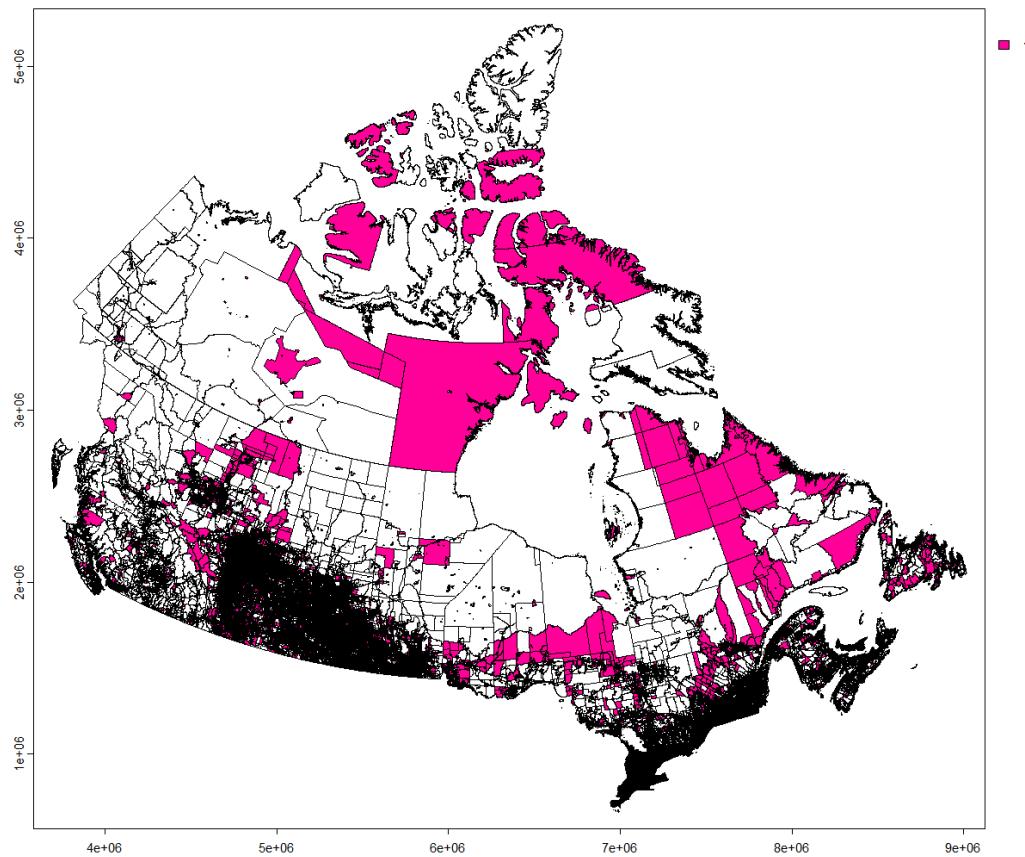
```

library(terra)
db_shp <- vect("../data/boundary/ldb_000b21a_e.shp")
db_shp

##  class       : SpatVector
##  geometry    : polygons
##  dimensions  : 498547, 6 (geometries, attributes)
##  extent      : 3689321, 9015751, 659305, 5242009 (xmin, xmax, ymin, ymax)
##  source      : ldb_000b21a_e.shp
##  coord. ref. : NAD83 / Statistics Canada Lambert (EPSG:3347)
##  names       : DBUID          DGUID   DBRPLAMX  DBRPLAMY LANDAREA PRUID
##  type        : <chr>          <chr>   <num>    <num>    <num>  <chr>
##  values      : 10010165001 2021S0513100101~ 8.978e+06 2.147e+06  0.0599   10
##                  10010165002 2021S0513100101~ 8.978e+06 2.146e+06  0.0084   10
##                  10010165006 2021S0513100101~ 8.978e+06 2.146e+06  0.3063   10

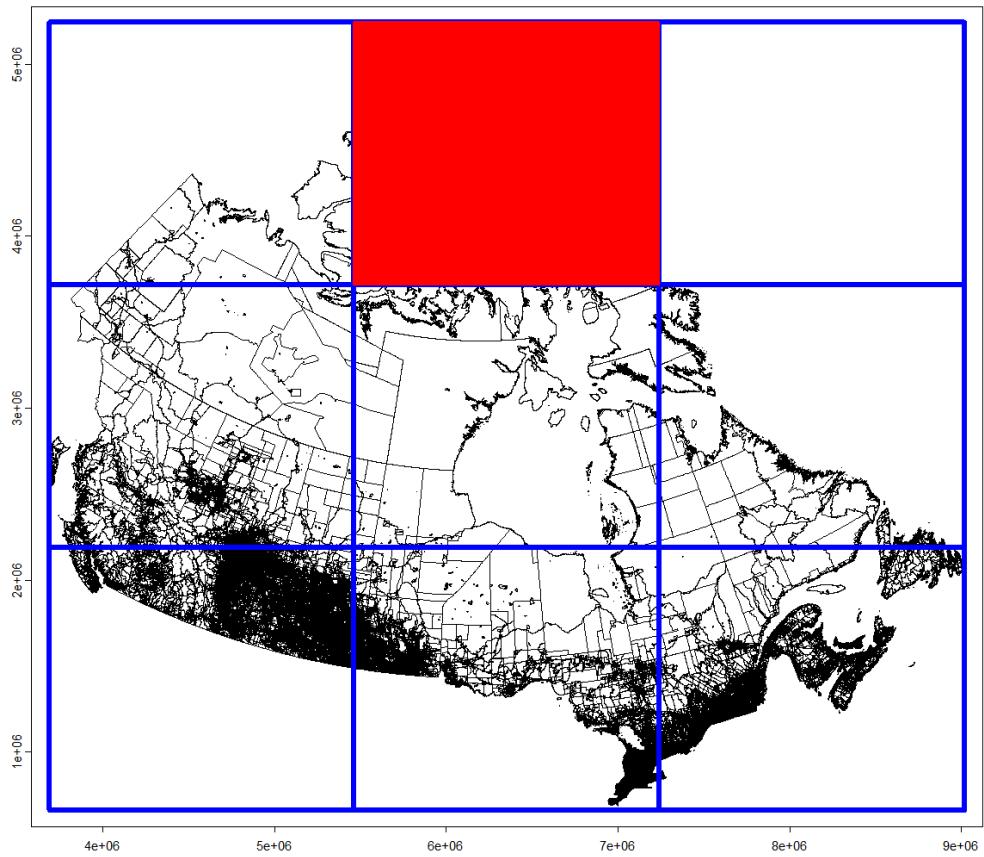
```



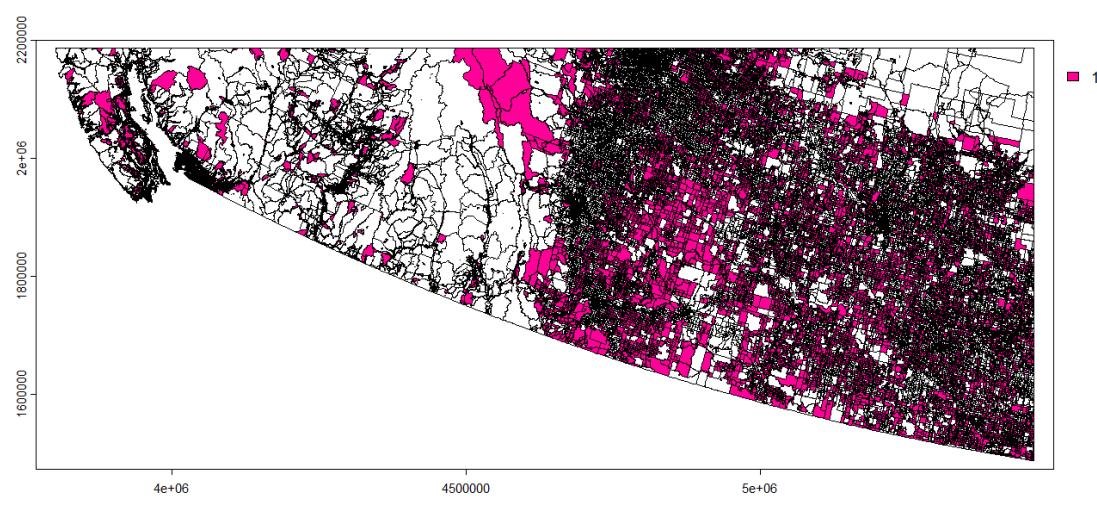


So where all proximity measures are missing are not randomly distributed. We have seen before that these areas have population. But our main target is to cluster the proximity measures. As all the proximity measures are missing here we can delete these rows from our database then do the clustering and also we can do clustering by keeping them aswell and see the difference.

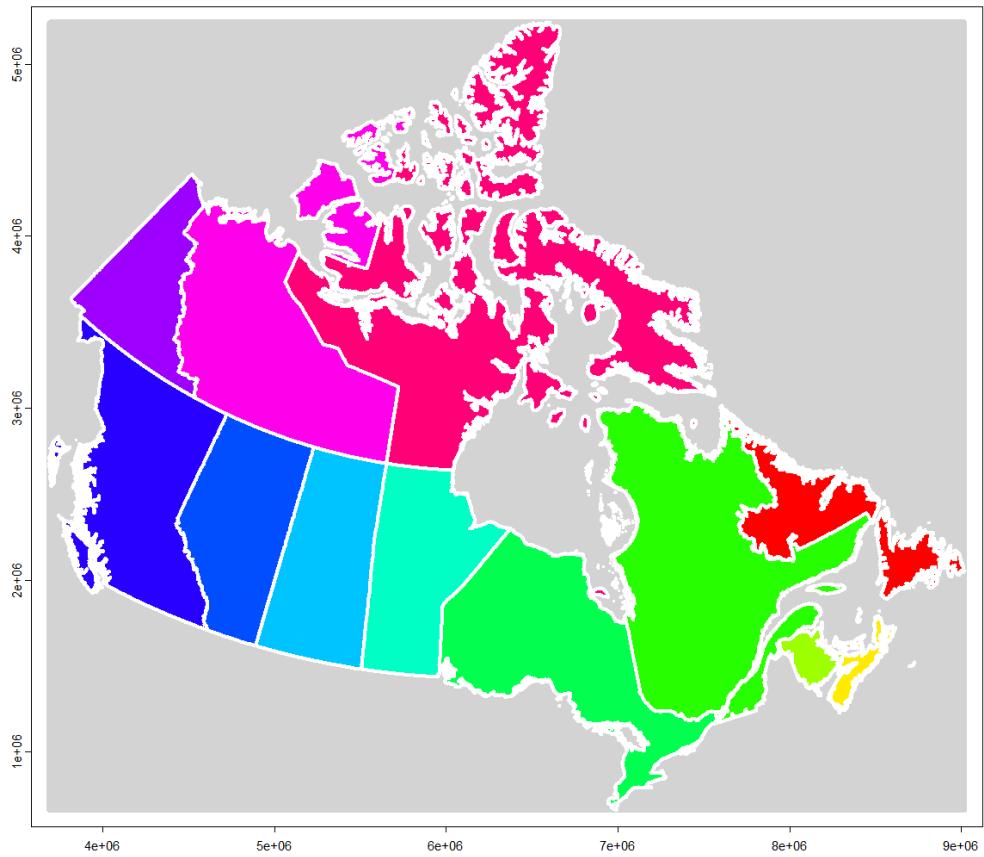
Cut into 9 zones.



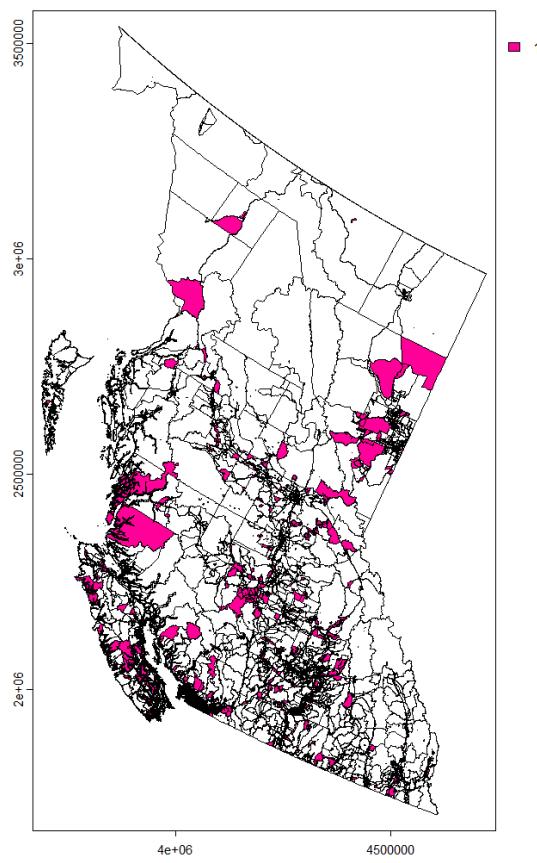
Left Lower portions of Canada



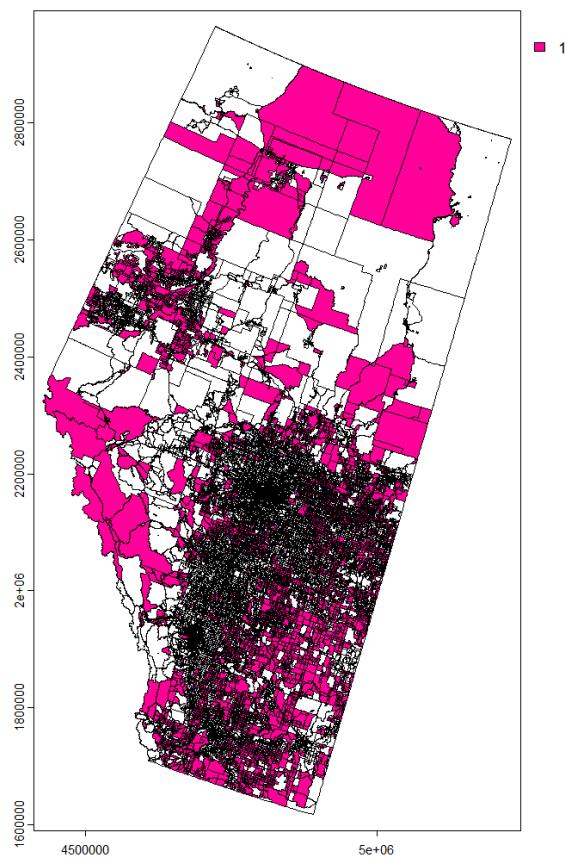
Provinces



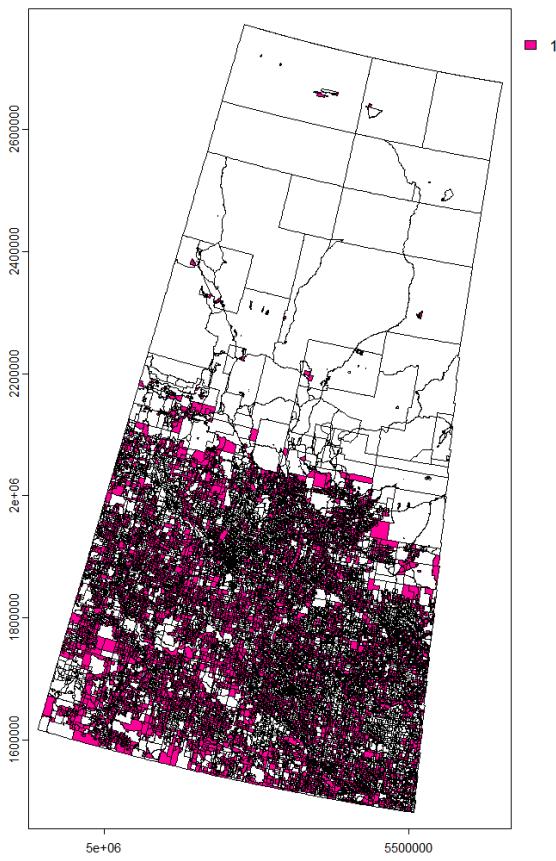
Closer look to all null proximity measures of BC.



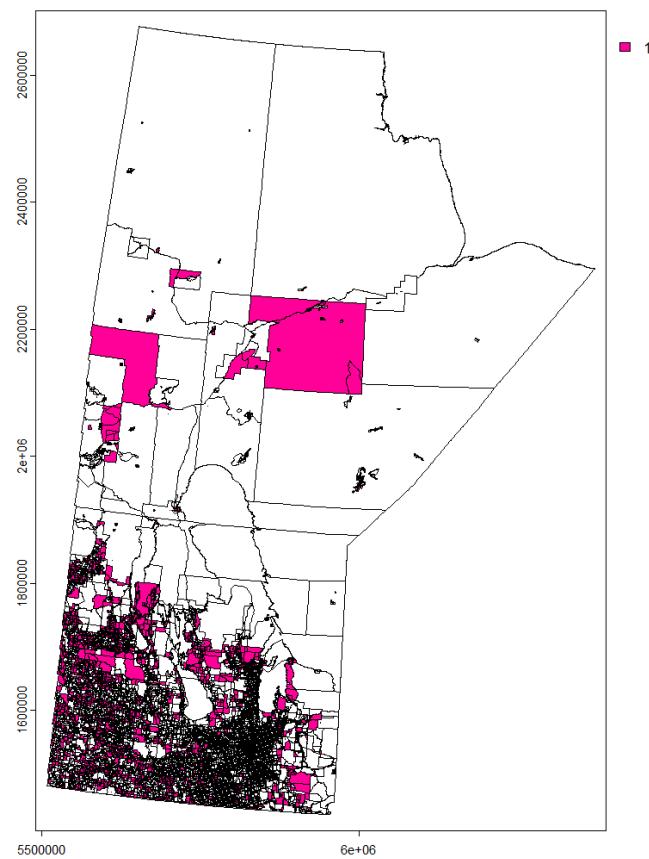
Closer look to all null proximity measures of Alberta.



Closer look to all null proximity measures of Saskatchewan

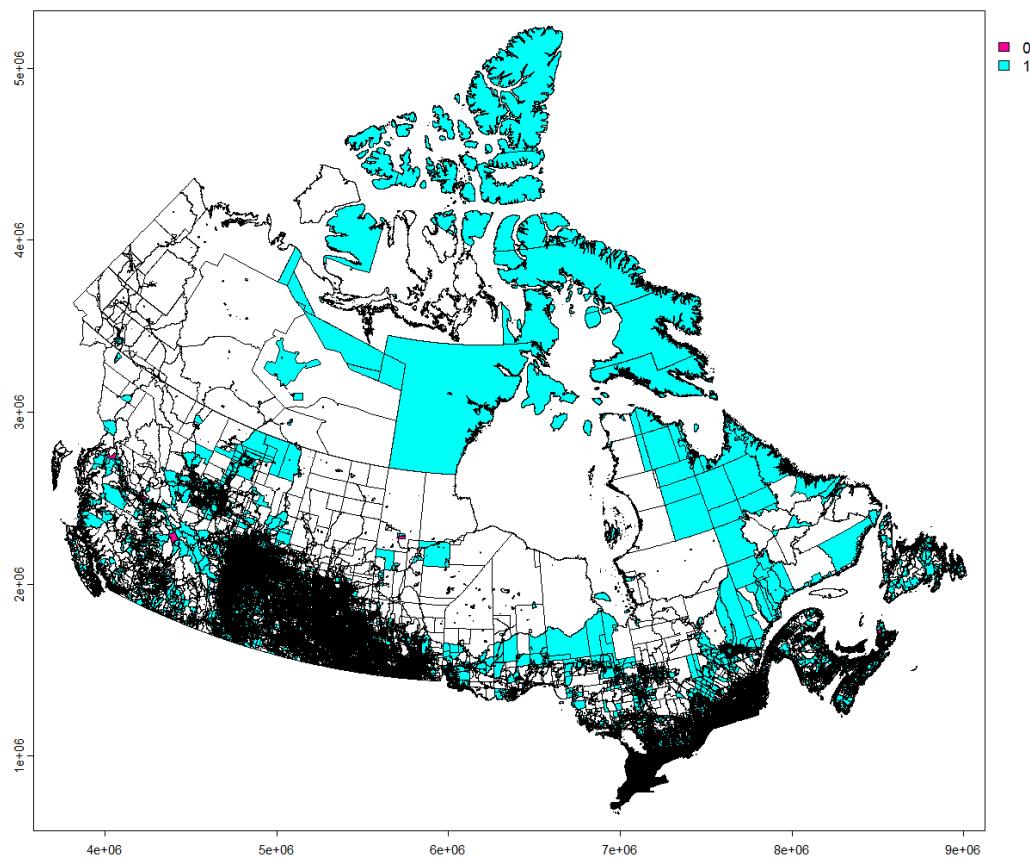


Closer look to all null proximity measures of Manitoba



Now let's look at the proximity measures na values aminity wise.

Let's look where grocery proximity measures are missing all over Canada.



Closer look to null grocery proximity measures of BC

