govhack: construction of token system using datasets

21/08/2021

We created a token system for 2 areas: Council and Type of Public space.

For the Type of Public space we used the data from the engagement report on public spaces and for the Council we used the time series data and national data on disaster relief.

The total rewards is the sum of the tokens collected travelling to a particular Council and the amount from going to a specific type of public space within the council area. For example, 15 tokens for going to Albury + 6 tokens for going to the park.

**Creating a token system for the Council Areas**

In order to create the token system, we used time series data to measure the gradient (using a linear regression) of positive or negative growth of different council areas over the years. Here were our consideraitons.

* We prioritised Council who were “suffering” i.e experiencing the most negative growth - to do this we ranked each variable based on magnitude of decline or increase.
* We added all the rank indices of the variables together to create a weighted score. The variables included: "Classification", "Population Density per capita/km2", "Socio-Economic", "Unemployment Rate" ,"Number of Active Businesses in LGA", "Recreational & Cultural Expenditure per capita", "Open Public Space", "Number of Public Swimming Pools", "Number of Public Halls", "Number of Public Libraries", from the original time series data. We also included data from the disaster relief (national data set) to promote travel and tourism in the council areas which had suffered disaster (in hopes to boost the economy in those areas). We merged these two datasets by lga and Council in the datasets.
* We ranked the weighted score and then applied tokens to sections of the rankings i.e. top 20 would have 15 tokens allocated, top 20-50 would get 20 tokens, top 50-80 would get 25 tokens (increased in 5 token increments)

**Creating a token system for the Public Space Areas**

Using the engagement report, we looked at the areas that the public engaged with during COVID, areas that can be improved and areas that the public loved. Then we manipulated the data as such:

* We looked at the types of public spaces that need a boost (with a lower percentage in the report) and then added more tokens to places that have not had as much attention previously.

#### Importing and cleaning the datasets

* Imported the datsets
* Chose the variables needed to calculate the score
* Cleaned the data: deleted the unneccessary terms such as: " Council", "The Council of the City of ", "The Council of the Shire of ", "The Council of the Municipality of ", " City", " Shire", " Municipal", "Council of the City of "

#state dataset  
  
library(readxl)

## Warning: package 'readxl' was built under R version 3.6.3

library(tm)

## Warning: package 'tm' was built under R version 3.6.3

## Loading required package: NLP

## Warning: package 'NLP' was built under R version 3.6.3

library(dplyr)

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

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

data2014\_2015<- read\_excel("OLG-TIME-SERIES-DATA-2014\_15.xlsm")  
data2015\_2016<- read\_excel("OLG-TIME-SERIES-DATA-2015\_16.xlsm")  
data2016\_2017<- read\_excel("OLG-TIME-SERIES-DATA-2016-17.xlsm")

## New names:  
## \* `Population aged >60 \r\n(%)` -> `Population aged >60 \r\n(%)...16`  
## \* `Population aged >60 \r\n(%)` -> `Population aged >60 \r\n(%)...19`

data2017\_2018<- read\_excel("Time-Series-Data-2017-2018.xlsm")

## New names:  
## \* `Population aged >60 \r\n(%)` -> `Population aged >60 \r\n(%)...16`  
## \* `Population aged >60 \r\n(%)` -> `Population aged >60 \r\n(%)...19`

data2018\_2019<- read\_excel("Time-Series-Data-2018-2019.xlsx")  
disaster\_2019\_2020 <- read.csv("elafent-ler-project-data-meta.csv")  
bushfires\_2019\_2020<- read\_excel("data.gov.au-extracted-on-12-may-2021-data-as-at-31-march-2021.xlsx")  
  
spaces\_love <- c("Parks", "Cycleways", "Footpaths", "Trees/greenery", "Street amenity", "Natural environment", "Local businesses/cafes", "Water", "Built environment", "Community/arts and culture")  
number <- c(26, 20, 15, 14, 13, 12, 9, 7, 5, 4)  
  
space\_love\_data <- cbind(area=spaces\_love, percent=number)%>% as.data.frame()  
  
spaces\_improved <- c("Cycleways", "Road infrastructure", "Footpaths", "Health and safety—pedestrian" ,"Health and safety—cycleways", "Accessibility", "Road space allocation", "Parks", "Parking allocation", "Trees/greenery")  
number\_1 <- c(39, 23, 20, 17, 13, 8, 7, 6, 5, 5)  
  
space\_improved\_data <- cbind(area=spaces\_improved,percent= number\_1)%>% as.data.frame()  
  
most\_COVID <- c("Local street in your neighbourhood", "Beaches and foreshores", "Parks outside your local area", "Off-leash dog areas", "Temporarily closed streets", "Major parkland", "Parks in your local area", "Bushland or national parks", "Local main or high street", "Walking track", "Plazas or paved public spaces", "I have not left my home", "Temporarily closed car parks")  
covid\_num <- c(76, 34, 22, 16, 2, 20, 71, 39, 36, 53, 15, 3, 4)  
  
covid\_data <- cbind(area=most\_COVID, percent=covid\_num) %>% as.data.frame()  
  
  
variables\_interest <- c("Classification", "Population Density per capita/km2", "Socio-Economic", "Unemployment Rate" ,"Number of Active Businesses in LGA", "Recreational & Cultural Expenditure per capita", "Open Public Space", "Number of Public Swimming Pools", "Number of Public Halls", "Number of Public Libraries")  
  
  
sub\_2014<- select(data2014\_2015,contains(variables\_interest))  
sub\_2015<- select(data2015\_2016,contains(variables\_interest))  
sub\_2016<- select(data2016\_2017,contains(variables\_interest))  
sub\_2017<- select(data2017\_2018,contains(variables\_interest))  
sub\_2018<- select(data2018\_2019,contains(variables\_interest))  
  
sub\_2014<- cbind(data2014\_2015[,c(1,2)],sub\_2014)   
colnames(sub\_2014) <- c( "Council", "year",variables\_interest)  
sub\_2015<- cbind(data2015\_2016[,c(1,2)],sub\_2015)   
colnames(sub\_2015) <- c( "Council", "year",variables\_interest)  
sub\_2016<- cbind(data2016\_2017[,c(1,2)],sub\_2016)   
colnames(sub\_2016) <- c( "Council", "year",variables\_interest)  
sub\_2017<- cbind(data2017\_2018[,c(1,2)],sub\_2017)  
colnames(sub\_2017) <- c( "Council", "year",variables\_interest)  
sub\_2018<- cbind(data2018\_2019[,c(1,2)],sub\_2018)   
colnames(sub\_2018) <- c( "Council", "year",variables\_interest)  
  
timeseries\_data <- rbind(sub\_2014,   
 sub\_2015,   
 sub\_2016,   
 sub\_2017,   
 sub\_2018) %>% na.omit()  
  
class(timeseries\_data$year)

## [1] "numeric"

timeseries\_data$Council <- removeWords(as.character(timeseries\_data$Council), c(" Council", "The Council of the City of ", "The Council of the Shire of ", "The Council of the Municipality of ", " City", " Shire", " Municipal", "Council of the City of " ))  
timeseries\_data$Council <- gsub("\\s\*\\([^\\)]+\\)","",as.character(timeseries\_data$Council ))  
  
timeseries\_data$Council <- trimws(timeseries\_data$Council, which = c("both"), whitespace = "[ \t\r\n]")  
  
timeseries\_data$`Number of Public Libraries`[timeseries\_data$`Number of Public Libraries` %in% c("Data not yet available", "\*", "#", "No Data", "No data")] <- NA  
timeseries\_data$`Recreational & Cultural Expenditure per capita`[timeseries\_data$`Recreational & Cultural Expenditure per capita` %in% c("Data not yet available", "\*", "#", "No Data", "No data")] <- NA  
  
timeseries\_data$PublicSpaces <-   
as.numeric(timeseries\_data$`Open Public Space`) + as.numeric(timeseries\_data$`Number of Public Swimming Pools` )+ as.numeric(timeseries\_data$`Number of Public Halls`) + as.numeric(timeseries\_data$`Number of Public Libraries`)

## Warning: NAs introduced by coercion

## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion

timeseries\_data$class\_score[timeseries\_data$Classification == "Metropolitan"] <- 50  
timeseries\_data$class\_score[timeseries\_data$Classification == "Metropolitan Fringe"] <- 40  
timeseries\_data$class\_score[timeseries\_data$Classification == "Regional Town/City"] <- 30  
timeseries\_data$class\_score[timeseries\_data$Classification == "Rural"] <- 20  
timeseries\_data$class\_score[timeseries\_data$Classification == "Large Rural"] <- 10  
  
timeseries\_data <-timeseries\_data[,c(1,2,4:8,13,14)]

#### Matching Councils: Merging the NSW time series dataset with the National disaster relief + bushfires datasets

* Merge the NSW time series dataset for public spaces with the national disaster relief data set

#bushfires\_2019\_2020 <- bushfires\_2019\_2020%>%mutate(across(where(is.character), as.factor))  
#summary(bushfires\_2019\_2020)  
#colnames(bushfires\_2019\_2020)  
bushfires\_2019\_2020$location\_name<- gsub("\\s\*\\([^\\)]+\\)","",as.character(bushfires\_2019\_2020$location\_name))  
  
  
#disaster\_2019\_2020 <- disaster\_2019\_2020%>%mutate(across(where(is.character), as.factor))  
#summary(disaster\_2019\_2020)  
disaster\_2019\_2020$lga<- removeWords(as.character(disaster\_2019\_2020$lga),"Council")  
#summary(as.factor(disaster\_2019\_2020$lga))  
  
  
#Match the councils in the disaster relief dataset with the councils in the time series dataset  
index<- which(as.character(disaster\_2019\_2020$lga) %in% as.character(timeseries\_data$Council))  
  
disaster\_NSW<- disaster\_2019\_2020[index,] %>% dplyr::select(lga, cwlth\_funding)  
  
#Match the councils in the bushfire dataset with the councils in the time series dataset  
index<- which(as.character(bushfires\_2019\_2020$location\_name) %in% as.character(timeseries\_data$Council))  
  
bushfire\_NSW<- bushfires\_2019\_2020[index,]  
  
   
bush\_data <- bushfire\_NSW %>%   
 group\_by(location\_name) %>%   
 summarise(n=n())

## `summarise()` ungrouping output (override with `.groups` argument)

#### Analysis on the time series datasets: percentage change

Used a linear regression model to calculate the gradient of the longitudinal data.

timeseries\_data <- timeseries\_data %>%   
 na.omit()   
summary(timeseries\_data$`Socio-Economic`)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 35.00 70.00 70.53 105.00 153.00

summary(timeseries\_data$year)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2014 2014 2015 2016 2018 2018

summary(timeseries\_data$year)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2014 2014 2015 2016 2018 2018

#getting the percentage change  
# load scales to format dates on x-axis  
  
#(timeseries\_data$year,timeseries\_data$`Socio-Economic` )  
  
result\_matrix <- matrix(ncol=3)  
for (i in timeseries\_data$Council){  
 dat <- timeseries\_data[which(timeseries\_data$Council %in% i),]  
 class\_score <- c(i, "class\_Score", dat[,9])  
 result\_matrix <- rbind(result\_matrix, class\_score)  
  
 for (j in colnames(dat[,c(3:8)])){  
 #print(j)  
 indx <- which(colnames(dat) %in% j)  
 #print(indx)  
 mod <- lm(dat[,indx]~year, data = dat)  
 res <- c(i, j, coef(mod)[2])  
 result\_matrix <- rbind(result\_matrix, res)  
   
 }  
}  
  
result\_matrix <- as.data.frame(result\_matrix) %>% na.omit()  
result\_matrix<- result\_matrix %>%   
 group\_by(V1) %>%   
 filter(n()>1) %>%   
 as.data.frame() %>%   
 mutate(year = as.numeric(as.character(year)))%>%  
 mutate(V1 = (as.character(V1)))%>%  
 mutate(year = round(year,3))%>%  
 rename("grad" = "year")

#### Calculating the score and token system

#### Calculating the score and token system  
  
result\_data<- reshape(result\_matrix, idvar="V1", timevar="V2", v.names="grad", direction="wide")  
  
result\_data<- merge(result\_data, bush\_data, by.x ="V1", by.y = "location\_name", all.x = "TRUE")  
result\_data <- result\_data %>%   
 mutate(bush = ifelse(!is.na(n), 0, 20))  
  
#merge(result\_data, disaster\_NSW, by.x = "V1", by.y = "lga", all.x = "TRUE")  
  
result<- cbind(council = as.character(result\_data$V1),  
 classification = as.numeric(result\_data$grad.class\_Score),  
 ord\_pop\_dens = as.numeric(rank(result\_data$`grad.Population Density per capita/km2`)),  
 ord\_socio = as.numeric(rank(result\_data$`grad.Socio-Economic`)),  
 ord\_unemploy = as.numeric(rank(result\_data$`grad.Unemployment Rate`)),  
 ord\_activbus = as.numeric(rank(result\_data$`grad.Number of Active Businesses in LGA`)),  
 ord\_recexpcap = as.numeric(rank(result\_data$`grad.Recreational & Cultural Expenditure per capita`)),  
 ord\_pubspac = as.numeric(rank(result\_data$grad.PublicSpaces)),  
 bush = result\_data$bush) %>%as.data.frame()  
  
result[,c(2:9)]<- result[,c(2:9)] %>%   
 mutate\_if(is.factor, as.character)%>%   
 mutate\_if(is.character, as.numeric)  
  
str(result)

## 'data.frame': 129 obs. of 9 variables:  
## $ council : Factor w/ 129 levels "Albury","Armidale Regional",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ classification: num 30 30 30 20 30 30 10 10 50 10 ...  
## $ ord\_pop\_dens : num 100 61 97 48 78 73.5 36 68 116 40.5 ...  
## $ ord\_socio : num 13 102 63 81.5 26.5 50 128 57 118 46 ...  
## $ ord\_unemploy : num 31 13 25 62.5 76.5 120 124 34 84.5 74.5 ...  
## $ ord\_activbus : num 82 65 84 32 74 68 45 4 125 16 ...  
## $ ord\_recexpcap : num 58 2 45 4 78 89 60 15 51 101 ...  
## $ ord\_pubspac : num 103 68.5 104.5 68.5 17 ...  
## $ bush : num 20 0 0 20 20 0 0 20 20 20 ...

result$score <- NA  
for (i in 1:nrow(result)){  
 result[i,]$score <- result[i,2]+result[i,3]+result[i,4]+result[i,5]+result[i,6]+result[i,7]+result[i,8]+result[i,9]  
 result$order\_score <- rank(result$score)  
}  
  
result<-result %>% arrange(desc(order\_score))  
result$tokens\_council <- c(rep(15,38), rep(20,35), rep(25, 35), rep(30,21))  
  
write.csv(result, file = "tokens.csv")

#### Adding token system to COVID data from the engagment report

Creating the token system for the type of public space

covid\_data$cat <- "COVID"  
space\_improved\_data$cat <- "improve"  
space\_love\_data$cat <- "love"  
  
extra\_tokens <- rbind(covid\_data,  
 space\_improved\_data,  
 space\_love\_data)   
  
extra\_tokens <- extra\_tokens%>%   
 mutate(percent = as.numeric(as.character(percent)))%>%  
 arrange(desc(percent))  
extra\_tokens$order\_num <- rank(extra\_tokens$percent)  
extra\_tokens$extra\_tokens <- c(rep(1,5), rep(2, 5), rep(3, 5), rep(4,5), rep(5,5), rep(6,5), rep(7,3))  
  
write.csv(extra\_tokens, file = "extra\_tokens.csv")