Q6. Error Calculation

* Assumptions:

1. The restaurants that contain geographical information features can be accurately categorized by continent.
2. The list of countries and continents is a complete list.
3. The restaurants that do not match the list of countries and continents cannot be categorized by continents and generate errors.

* Results:

The total number of restaurants: 4160.

The number of restaurants that contain geographical features: 3758.

Error: (4160 - 3758) / 4160 \* 100% = 9.66%.

* Perl code to get the number of total restaurants and the number of restaurants that contain geographical features.

use strict;

use warnings;

my $features = 'C:/Users/yhan/Documents/CS773/q6/features.txt';

open(my $featurefile, '<', $features) or die "Could not open file!";

my %feature;

while (<$featurefile>) {

my $line = $\_;

my ($i, $j) = split(/\t/, $line);

$feature{$i} = $j;

#print "$i is $feature{$i}\n";

}

my @countries;

my $country = 'C:/Users/yhan/Documents/CS773/q6/countries.txt';

open(my $file, '<', $country) or die "Could not open file!";

while(<$file>) {

chomp;

my $line = $\_;

push(@countries, $line);

#print $\_;

}

#my %country\_elements;

#@country\_elements{@countries} = ();

my $countrys = "African Afro-Eurasian Antarctic Americana Asian Middle Eastern Australasian Australianb Eurasian European North American Central American Oceanian South American Abkhaz Abkhazian Afghan Åland Island Albanian Algerian American Samoan Andorran Angolan Anguillan Antarctic Antiguan or Barbudan Argentine Armenian Aruban Australian Austrian Azerbaijani Azeri Bahamian Bahraini Bangladeshi Barbadian Belarusian Belgian Belizean Beninese Beninois Bermudian Bermudan Bhutanese Bolivian Bonaire Bosnian or Herzegovinian Motswana Botswanan Bouvet Island Brazilian BIOT Bruneian Bulgarian Burkinabé Burmese Burundian Cabo Verdean Cambodian Cameroonian Canadian Cabo Verdean Caymanian Central African Chadian Chilean Chinese Christmas Island Cocos Island Colombian Comoran Comorian Congolese Cook Island Costa Rican Ivorian Croatian Cuban Curaçaoan Cypriot Czech Danish Djiboutian Dominican Dominican Timorese Ecuadorian Egyptian Salvadoran English British Equatorial Guinean Equatoguinean Eritrean Estonian Ethiopian European Falkland Island Faroese Fijian Finnish French French Guianese French Polynesian French Southern Territories Gabonese Gambian Georgian German Ghanaian Gibraltar British UK Greek Hellenic Greenlandic Grenadian Guadeloupe Guamanian Guambat Guatemalan Channel Island Guinean Bissau-Guinean Guyanese Haitian Heard Island or McDonald Islands Honduran Hong Kong Hong Kongese Hungarian Magyar Icelandic Indian Indonesian Iranian Persian Iraqi Irish Manx Israeli Italian Ivorian Jamaican Jan Mayen Japanese Channel Island Jordanian Kazakhstani Kazakh Kenyan I-Kiribati North Korean South Korean Kosovar Kosovan Kuwaiti Kyrgyzstani Kyrgyz Kirgiz Kirghiz Lao Laotian Latvian Lebanese Basotho Liberian Libyan Liechtenstein Lithuanian Luxembourg Luxembourgish Macanese Chinese Macedonian Malagasy Malawian Malaysian Maldivian Malian Malinese Maltese Marshallese Martiniquais Martinican Mauritanian Mauritian Mahoran Mexican Micronesian Moldovan Monégasque Monacan Mongolian Montenegrin Montserratian Moroccan Mozambican Burmese Namibian Nauruan Nepali Nepalese Dutch Netherlandic New Caledonian New Zealand NZ Nicaraguan Nigerien Nigerian Niuean Norfolk Island Northern Irish British Northern Marianan Norwegian Omani Pakistani Palauan Palestinian Panamanian Papua New Guinean Papuan Paraguayan Peruvian Philippine Filipino Pitcairn Island Polish Portuguese Puerto Rican Qatari Réunionese Réunionnais Romanian Russian Rwandan Saba Barthélemois Saint Helenian Kittitian or Nevisian Saint Lucian Saint-Martinoise Saint-Pierrais or Miquelonnais Saint Vincentian Vincentian Samoan Sammarinese São Toméan Saudi Saudi Arabian Scots Scottish British Senegalese Serbian Seychellois Sierra Leonean Singapore Singaporean Sint Eustatius Statian Sint Maarten Slovak Slovenian Slovene Solomon Island Somali South African South Georgia or South Sandwich Islands South Ossetian South Sudanese Spanish Sri Lankan Sudanese Surinamese Svalbard Swazi Swedish Swiss Syrian Taiwanese Tajikistani Tanzanian Thai Timorese Togolese Tokelauan Tongan Trinidadian or Tobagonian Tunisian Turkish Turkmen Turks and Caicos Island Tuvaluan Ugandan Ukrainian Emirati Emirian Emiri British UK United States US American Uruguayan Uzbekistani Uzbek Ni-Vanuatu Vanuatuan Vatican Venezuelan Vietnamese British Virgin Island U.S. Virgin Island Welsh British Wallis and Futuna Wallisian or Futunan Sahrawi Sahrawian Sahraouian Yemeni Zambian Zimbabwean";

my $total = 0;

my $geo = 0;

my $dir = 'C:/Users/yhan/Documents/data';

foreach my $fp (glob("$dir/\*.txt")) {

open my $fh, "<", $fp or die "can't read open '$fp'";

while (<$fh>) {

chomp;

my $line = $\_;

$total = $total + 1;

my $flag = 0;

my @resfeatures = split /[\s,\t]/, $\_;

foreach my $ele (@resfeatures)

{

if(exists $feature{$ele}){

my $cty = $feature{$ele};

$cty =~ tr/A-Za-z//cd;

if ($countrys =~ /$cty/)

{

$flag = 1

#print $feature{$ele};

}

#print $feature{$ele};

}

}

$geo = $geo + $flag;

}

close $fh or die "can't read close";

}

print "$total, $geo \n";