LING 518 Final Project

Mitchell Xiao Xiao Li

09/12/2020

Introduction

There should be only one ingredient in a jar of honey, but sometimes, manufacturers like to get *creative*. Therefore, in 2018-2019, the Canadian Food Inspection Agency (CFIA) sampled 240 random jars of honey from across the country, and checked whether they exceed the expected threshold for non-honey sugar. The publicly-available results comprise dataset I will be using for this project, which includes information about the province in which the sample was taken, the country of origin for the product, the floral source of the honey, the amount of non-honey sugars, and whether or not the sample was judged as satisfactory, according to the CFIA (https://open.canada.ca/data/en/dataset/7ecfee4c-b6f5-45f8-9fd4-16976f55f8d9).

This will be an exploratory analysis with the goal of determining if changes in province of sampling, country of origin, or floral source increase the likelihood of an unsatisfactory rating. The ideal model will be a logistic regression with the rating as the outcome variable and province of sampling, country of origin, and floral source as predictor variables, but the specific structure will depend on what the data look like.

```r
# in case the countrycode package isn't already installed
# install.packages("countrycode")


# installing necessary packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------------------- tidyverse 1.3.0

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ----------------------------------------------------------------- tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(effects)
```

```
## Loading required package: carData

## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
library(ggplot2)
library(countrycode)
```

```
## Warning: package ’countrycode’ was built under R version 4.0.3
```

```
# reading the data
honey <- read_csv("honey.csv")
```

```
## Parsed with column specification:
## cols(
##   ‘Sample # - No d’échantillon‘ = col_character(),
##   ‘Sampled From - Lieu d’échantillonnage‘ = col_character(),
##   ‘Declared Country of Origin‘ = col_character(),
##   ‘Pays d’origine déclaré‘ = col_character(),
##   ‘Declared Floral Source‘ = col_character(),
##   ‘Source florale déclarée‘ = col_character(),
##   ‘Calculated C4 Sugar Addition - Addition de sucre C4 calculé‘ = col_character(),
##   ‘SIRA Assessment - Évaluation d’ARIS‘ = col_character(),
##   ‘NMR Assessment - Évaluation de RMN‘ = col_character(),
##   ‘Sample Assessment - Évaluation d’échantillon‘ = col_character()
## )
```

Cleaning and processing the data

Reading the data already shows that some irrelevant columns need to be deleted and the remaining columns need to be renamed to be shorter and friendlier to computers.

```
# printing the column names of the dataset in a list
colnames(honey)
```

```
##  [1] "Sample # - No d’échantillon"
##  [2] "Sampled From - Lieu d’échantillonnage"
##  [3] "Declared Country of Origin"
##  [4] "Pays d’origine déclaré"
##  [5] "Declared Floral Source"
##  [6] "Source florale déclarée"
##  [7] "Calculated C4 Sugar Addition - Addition de sucre C4 calculé"
##  [8] "SIRA Assessment - Évaluation d’ARIS"
##  [9] "NMR Assessment - Évaluation de RMN"
## [10] "Sample Assessment - Évaluation d’échantillon"
```

```
# removing two redundant columns for country of origin and floral source in
# French, as well as the SIRA and NMR assessment columns (this analysis
# will only be concerned with the sample assessment column)
honey <- honey[,-c(4,6,8,9)]

# renaming the columns
honey <- honey %>%
  rename(
    sample_num = "Sample # - No d’échantillon",
    sample_prov = "Sampled From - Lieu d’échantillonnage",
    origin_country = "Declared Country of Origin",
    floral_source = "Declared Floral Source",
    C4_content = "Calculated C4 Sugar Addition - Addition de sucre C4 calculé",
    rejected = "Sample Assessment - Évaluation d’échantillon"
  )
```

The analysis will be a logistic regression model with “rejected” as the outcome variable, so the values need to be changed to TRUE and FALSE and the vector needs to be changed to a logical format.

```
# checking that there are only two types of values in the column, and counting
# how many rows include each type of value, this also shows that this will be
# a useful outcome variable, as roughly 1/5 of samples are rejected
table(honey$rejected)
```

```
##
##   S U-I
## 188  52
```

```
# replacing "S" (satisfactory) with FALSE and "U-I" (unsatisfactory) with TRUE
honey$rejected <- honey$rejected %>%
  replace(honey$rejected=="S", FALSE) %>%
  replace(honey$rejected=="U-I", TRUE)

# changing the format of rejected in the honey dataset
honey <- honey %>%
  mutate(rejected = as.logical(rejected))
```

The next parts will examine the predictors and make them suitable for use in a logistic model. First, the sample_prov predictor.

```
# counting the values of sample_prov
unique(honey$sample_prov)
```

```
## [1] "AB" "ON" "QC" "BC" "MB" "SK"
```

For sample province, there were only 6 provinces (from British Columbia to Quebec) that were included in the samples, so this column will be changed to a factor so that the data can be analyzed by province. Before deeming it ready for use, since will be a categorical predictor in a logistic regression model, it needs one final check for complete or quasi-complete separation, which is when all values of a predictor correspond to a single value of the outcome variable, rather than being dispersed across the two levels of the outcome variable. Since a logistic regression model will treat each level of the factor as a separate parameter to estimate, this means that all levels of all factors will need to fully cross the outcome variable. This is checked by counting the values of the predictor variable for sample_prov.

```
# checking for separation
table(honey$rejected, honey$sample_prov)
```

```
##
##         AB BC MB ON QC SK
##   FALSE 11 23  3 97 52  2
##   TRUE   1  1  0 37 13  0
```

It looks like for samples taken in Manitoba and Saskatchewan, there were no rejected samples. This will be handled by grouping the Prairie provinces (Alberta, Manitoba, and Saskatchewan) in a new level of sample_prov, so that this new level fully crosses the outcome variable. While this reduces specificity in the results of the model, it still allows for the initial hypotheses to be answered, and increases the stability of the model (explanation of separation and possible solutions from https://stats.idre.ucla.edu/other/mult-pkg/faq/general/faqwhat-is-complete-or-quasi-complete-separation-in-logisticprobit-regression-and-how-do-we-deal-with-them/).

```
# replacing all values of "AB", "SK", and "MB" with "Prairies"
honey$sample_prov <- honey$sample_prov %>%
  str_replace_all(c("AB" = "Prairies",
                    "SK" = "Prairies",
                    "MB" = "Prairies"))

# recoding sample_prov to a factor
honey <- honey %>% mutate(sample_prov = as.factor(sample_prov))

# checking for separation again
table(honey$rejected, honey$sample_prov)
```

```
##
##         BC ON Prairies QC
##   FALSE 23 97       16 52
##   TRUE   1 37        1 13
```

Great, the sample_prov predictor is now ready. Next, the origin_country predictor.

```
# printing all the unique values of origin_country by their spread across the
# outcome variable, to check the number of levels and for separation
table(honey$rejected, honey$origin_country)
```

```
##
##         Argentina Australia Australia, Brazil Austria Brasil, Australia
##   FALSE         2         7                 2       3                 3
##   TRUE          0         1                 0       0                 0
##
##         Brasil, Uruguay, India Brazil Brazil, Mexico Canada Croatia
##   FALSE                      0     19              2     19       2
##   TRUE                       1      1              0      0       0
##
##         Cuba, Mexico, Argentina France Germany Greece India India, Thailand
##   FALSE                       1     11       2      1    12               1
##   TRUE                        0      0       3      8    14               1
##
##         India, Uruguay, Viet Nam India, US Iran Israel Italy Mexico
##   FALSE                        1         1    0      3     7      1
##   TRUE                         0         0    1      3     0      0
##
##         Mexico, Brasil Mexico, Cuba, Argentina, Uruguay Mexico, Thailand
##   FALSE              2                                2                1
##   TRUE               0                                0                0
##
##         Moldova Myanmar, Thailand Netherlands New Zealand Pakistan Portugal
##   FALSE       1                 0           2          21         0        1
##   TRUE        0                 1           0           1         4        0
##
##         Romania Saudi Arabia Spain
##   FALSE       4            7     2
##   TRUE        0            0     0
##
##         Spain, Mexico, Guatemala, Ukraine, Bulgaria, Thailand
##   FALSE                                                     4
##   TRUE                                                      0
##
##         Taiwan, prov. China Thailand Turkey Ukraine USA USA, Argentina
##   FALSE                   0        6      3       1  19              1
##   TRUE                    2        1      1       0   3              0
##
##         USA, Viet Nam, India, Uruguay Viet Nam Zambia
##   FALSE                             3        6      2
##   TRUE                              1        5      0
```

It looks like there are a lot unique values, and it would be too many levels if this was used as a factor, so the data need to be grouped. Going by high-frequency levels will not work here, because the biggest groups combined (Brazil, Canada, New Zealand, and the USA) only account for less than 50% of the values, and there are some samples that have a mixture of origins. This indicates that the original idea of using country of origin as an outcome variable needs to be modified so that a broader grouping variable, like continent, can be used as a predictor instead.

But first, it looks like there are two spellings of “Brazil”, namely “Brasil” and “Brazil”. The function that will be used to match values in the country of origin varialbe to a continent uses a specific spelling, so first, all the values of “Brasil” need to be changed to “Brazil”, for consistency.

```r
# replacing all instances of "Brasil" with "Brazil" in the honey dataset
honey$origin_country <- honey$origin_country %>%
    str_replace("Brasil", "Brazil")
```

Now the countrycode package can be used to generate values for continent of origin, based on the country of origin values. However, “continent” as defined by the package merges North and South America, which would merge three of the largest groups in country of origin (this is also why latitude was not used to group the data, the distinctions between South America, south Asia, and Oceania would be blurred). Therefore the data will be grouped by “region”, of which there are seven, as defined by the World Bank Development Indicators (https://datahelpdesk.worldbank.org/knowledgebase/articles/906519).

```r
# Using the countrycode function from the countrycode package to match the
# origin_country values from the honey dataset to a region. This function takes
# a vector (specified by sourcevar), defines its values (specified by origin)
# and matches it to its corresponding value in the destination specification

regions <- countrycode(sourcevar = honey$origin_country,
                       origin = "country.name",
                       destination = "region")
```

```
## Warning in countrycode(sourcevar = honey$origin_country, origin = "country.name", : Some values were
```

```
## Warning in countrycode(sourcevar = honey$origin_country, origin = "country.name", : Some strings wer
```

The warning shows that after matching the values that could be matched, some values couldn't be matched and were returned as NA, which separates the single-origin honey samples (the ones that could be matched) from the mixed origin samples (that generated an NA). In the next section, the NA values are replaced with "mixed origin" and then the values from the region vector are added to the honey dataset in a new column called "origin_region".

```
# sanity check, to see if there are a reasonable number of NA values (27 seems
# appropriate, given the table() output for origin_country)
sum(is.na(regions))
```

```
## [1] 27
```

```
# replacing the NA values with "Mixed origin"
regions <- regions %>% replace_na("Mixed origin")

# checking if any NA values remain
sum(is.na(regions))
```

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

```
# adding an origin_region column to the honey dataset using add_column, with
# the contents of the regions variable
honey <- add_column(honey, origin_region=regions)
```

One last step, which is to again check for separation.

```
# checking for separation
table(honey$rejected, honey$origin_region)
```

```
##
##         East Asia & Pacific Europe & Central Asia Latin America & Caribbean
##   FALSE                  40                    40                        22
##   TRUE                   10                    12                         1
##
##         Middle East & North Africa Mixed origin North America South Asia
##   FALSE                         10           23            38         13
##   TRUE                           4            4             3         18
##
##         Sub-Saharan Africa
##   FALSE                  2
##   TRUE                   0
```

It looks like most levels are okay, except for Sub-Saharan Africa, which had no rejected samples. To solve this, those values will be grouped with Middle East & North Africa to create a new level called "Middle East & Africa". Again, this is not ideal, but it will allow the model to run smoothly.

```
# replacing all values of "Middle East & North Africa" and "Sub-Saharan Africa"
  # with "Middle East & Africa"
honey$origin_region <- honey$origin_region %>%
  str_replace_all(c("Middle East & North Africa" = "Middle East & Africa",
                    "Sub-Saharan Africa" = "Middle East & Africa"))
```

```r
# recoding origin_region to a factor
honey <- honey %>% mutate(origin_region = as.factor(origin_region))

# checking for separation again
table(honey$rejected, honey$origin_region)
```

```
##
##         East Asia & Pacific Europe & Central Asia Latin America & Caribbean
##   FALSE                  40                    40                        22
##   TRUE                   10                    12                         1
##
##         Middle East & Africa Mixed origin North America South Asia
##   FALSE                   12           23            38         13
##   TRUE                     4            4             3         18
```

With the country of origin (now region of origin) predictor ready, the next sections deal with the floral_source predictor.

```r
# printing all the unique values of floral_source by their spread across the
# outcome variable, to check the number of levels and for separation
table(honey$rejected, honey$floral_source)
```

```
##
##         acacia alfalfa avocado blossom beechwood blossom canola canola, alfalfa
##   FALSE     14       1               1                 1      8               1
##   TRUE       1       0               0                 0      0               0
##
##         chestnut citrus blossom clover flower and conifer forest forest blossom
##   FALSE        2              1      5                  1      1              1
##   TRUE         0              0      0                  0      0              0
##
##         forest honey lavender linden Linden Manuka multiflower Neem
##   FALSE            1        2      1      1     20           1    1
##   TRUE             0        0      0      0      1           0    3
##
##         orange blossom rata sidr starthistle blossom strawberry sunflower
##   FALSE              4    1    0                   0          1         1
##   TRUE               0    0    1                   1          0         0
##
##         undeclared White Himalayan wild berry wildflower ziziphus blossom
##   FALSE         95               2          1         17                1
##   TRUE          42               0          0          2                1
```

It looks like there are fewer levels of floral_source (but still too many to use as a factor) and the majority of samples do not have a declared floral source, but among the named ones, wildflower and Manuka are the most common. To use this data in the analysis, a new grouping column (floral_source_named) in the honey dataframe will be created that determines whether the sample has a named floral source (separating by undeclared versus other values in floral_source), to see if naming a floral source has an influence on the likelihood of pure honey.

```r
# using the ifelse() function to go through the floral_source column and assign
# values based on whether the value matches "undeclared" (assigns FALSE if it
# matches, assigns TRUE in all other conditions)
named <- ifelse(honey$floral_source=="undeclared", "no", "yes")

# adding a floral_source_named column to the honey dataframe, containing the
# contents of the variable "named"
honey <- add_column(honey, floral_source_named=named)

# checking for separation in the new column
table(honey$rejected, honey$floral_source_named)
```

```
##
##         no yes
##   FALSE 95  93
##   TRUE  42  10
```

```r
# making the new column a factor
honey <- honey %>%  mutate(floral_source_named = as.factor(floral_source_named))
```

With the data processing done, the next section will build the model.

Modelling

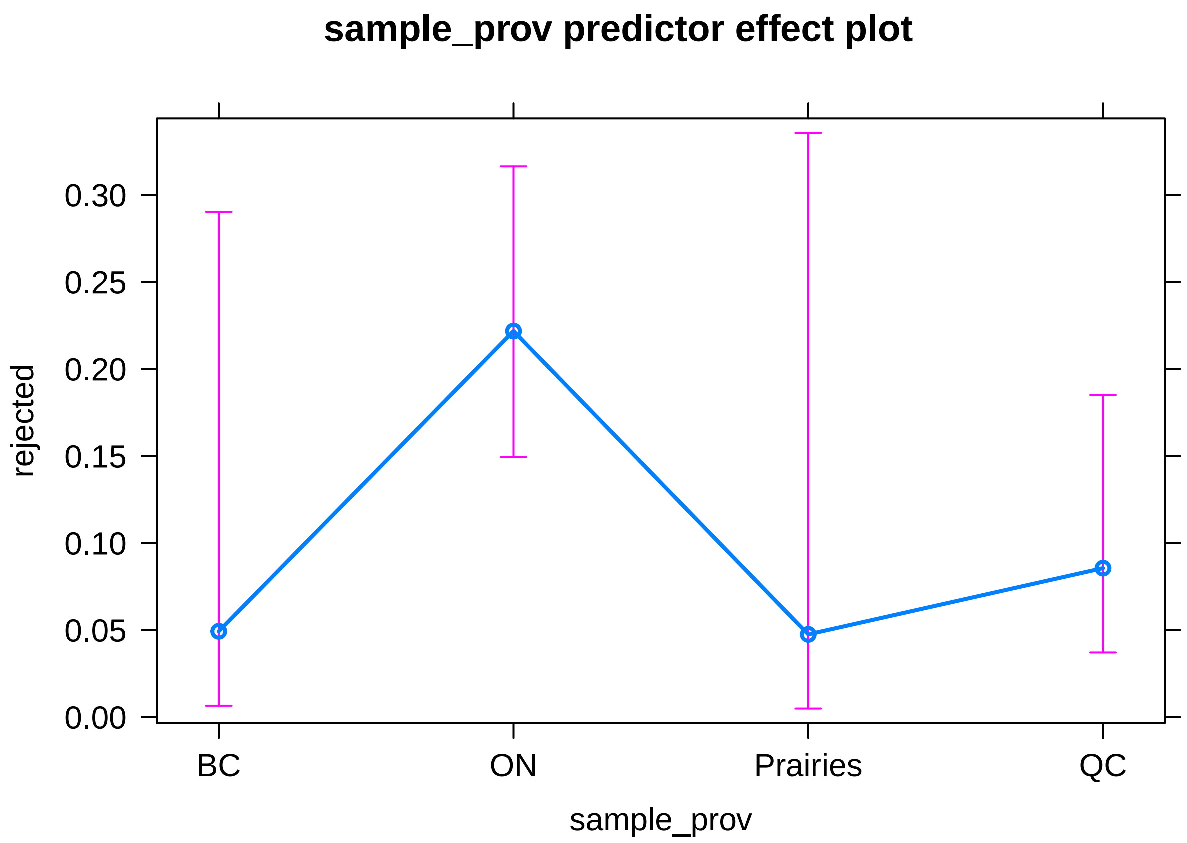
The first model is a logistic regression model with sample province, region of origin, and whether the floral source was named as outcome variables, and whether the sample failed to pass inspection as the outcome. Ideally, the different levels of country of origin and floral source would have been incorporated as random factors, but the way that the data are structured does not allow for the fixed effects to cross all levels of the (would-be) random effect, so they were not included. In addition, if the data had allowed for it, interactions between the predictors would have been included to check if the effects would change at different levels, but there are not enough samples per cell for the model to run if interactions had been included.

```r
mod_full <- glm(rejected ~ sample_prov + origin_region + floral_source_named,
            data=honey,
            family="binomial")
summary(mod_full)
```

```
##
## Call:
## glm(formula = rejected ~ sample_prov + origin_region + floral_source_named,
##     family = "binomial", data = honey)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8879  -0.6823  -0.3013  -0.1237   2.6309
##
## Coefficients:
##                                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)                             -1.92945    1.10164  -1.751  0.07987 .
## sample_provON                            1.70407    1.07688   1.582  0.11356
## sample_provPrairies                     -0.03873    1.58747  -0.024  0.98054
## sample_provQC                          0.59046    1.14049   0.518  0.60465
## origin_regionEurope & Central Asia      0.28640    0.53397   0.536  0.59171
## origin_regionLatin America & Caribbean -2.08998    1.10502  -1.891  0.05858 .
## origin_regionMiddle East & Africa      -0.15941    0.74313  -0.215  0.83015
## origin_regionMixed origin              -1.02571    0.69986  -1.466  0.14276
## origin_regionNorth America             -1.10183    0.75137  -1.466  0.14253
## origin_regionSouth Asia                 1.82328    0.56542   3.225  0.00126 **
## floral_source_namedyes                 -1.89438    0.46148  -4.105 4.04e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 250.88  on 239  degrees of freedom
## Residual deviance: 188.69  on 229  degrees of freedom
## AIC: 210.69
##
## Number of Fisher Scoring iterations: 6
```

Interpreting the numerical output from this model is difficult without transforming the estimates into probabilities, so an effect plot for each factor will be used to aid understanding. First, the sample_prov predictor plot.

```
# plotting the effect sizes for all levels of sample_prov, since type is set
# to response, the estimates are converted to probabilities and not plotted
# as log odds, which are the estimates in the model summary
plot(predictorEffects(mod_full, "sample_prov"), axes=list(y=list(type="response")))
```



And now, transforming the estimates into probabilities.

```
# saving the estimates (coefficients) from summary(mod_full) into variables for
# each province slope
intercept <- summary(mod_full)$coefficients[1,1]
slope_ON <- summary(mod_full)$coefficients[2,1]
slope_Prairies <- summary(mod_full)$coefficients[3,1]
slope_QC <- summary(mod_full)$coefficients[4,1]

# applying the plogis() function to the sum of the necessary coefficients to
# calculate a prediction for a given province; the sum is in terms of log odds
# and plogis() transforms them into probabilities
plogis(intercept)
```

```
## [1] 0.1268114
```

```
plogis(intercept+slope_ON)
```

```
## [1] 0.4438909
```

```
plogis(intercept+slope_Prairies)
```

```
## [1] 0.1225844
```

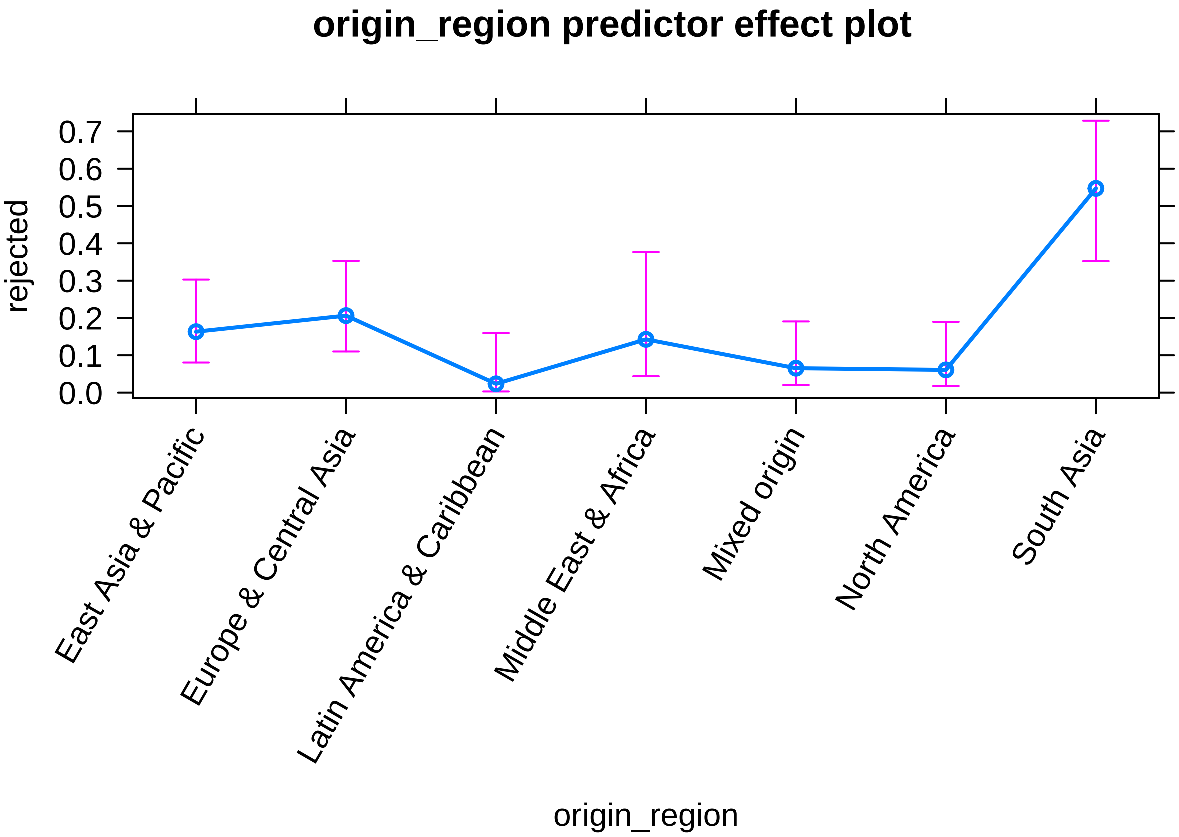
```
plogis(intercept+slope_QC)
```

```
## [1] 0.2076757
```

For this predictor, the reference level is British Columbia (represented by the intercept) which is the estimate to which all other estimates are adjustments. From both the plot and the signs and magnitudes of the estimates, samples from Ontario are the most likely to be rejected with Quebec second most likely, compared to BC, and samples from the Prairies very slightly less likely to be rejected, compared to BC. At the reference level for all the other predictors (sample from the East Asia & Pacific region, no declared floral source), the predicted probability for rejection was 0.13 for sample from BC and 0.12 for a sample from the Prairies (logit difference = -0.039, standard error = 1.587, z-score = -0.024, p = 0.98), however, the wide confidence intervals on the plot indicate that these estimates may not be reliable indicators of an effect. Comparatively, for a sample from Ontario, the predicted probability of rejection was 0.44 (logit difference = +1.704, *SE* = 1.077, *z* = 1.582, p = 0.11), and 0.21 for a sample from Quebec (logit difference = +0.590, *SE* = 1.140, *z* = 0.518 p = 0.60). The p-values from all of these estimates do not indicate that there is enough evidence to reject the null hypothesis, which is that there is no difference in likelihood for a rejected sample between BC and the province in question.

Next, the origin_region effect plot.

```
# plotting the effects for all levels of origin_region with the same syntax as
# the previous plot, except with an added rotate parameter for the x-axis that
# rotates the x-axis labels by 60 degrees for better readability
plot(predictorEffects(mod_full, "origin_region"), axes=list(y=list(type="response"),
                                                          x=list(rotate=60)))
```



Again, transforming the estimates for this predictor into probabilities.

```r
# saving the estimates (coefficients) from summary(mod_full) into variables for
# each region of origin slope, the intercept is not included here because it
# was already saved from before
slope_Europe_CentralAsia <- summary(mod_full)$coefficients[5,1]
slope_LatinAmerica_Caribbean <- summary(mod_full)$coefficients[6,1]
slope_MiddleEast_Africa <- summary(mod_full)$coefficients[7,1]
slope_Mixed_origin <- summary(mod_full)$coefficients[8,1]
slope_NorthAmerica <- summary(mod_full)$coefficients[9,1]
slope_SouthAsia <- summary(mod_full)$coefficients[10,1]

# applying the plogis() function to the sum of the necessary coefficients to
# calculate a prediction for a given region; the sum is in terms of log odds
# and plogis() transforms them into probabilities
plogis(intercept)
```

```
## [1] 0.1268114
```

```r
plogis(intercept+slope_Europe_CentralAsia)
```

```
## [1] 0.1620507
```

```r
plogis(intercept+slope_LatinAmerica_Caribbean)
```

```
## [1] 0.01764624
```

```r
plogis(intercept+slope_MiddleEast_Africa)
```

```
## [1] 0.1101847
```

```r
plogis(intercept+slope_Mixed_origin)
```

```
## [1] 0.04949319
```

```r
plogis(intercept+slope_NorthAmerica)
```

```
## [1] 0.04603264
```

```r
plogis(intercept+slope_SouthAsia)
```

```
## [1] 0.473481
```

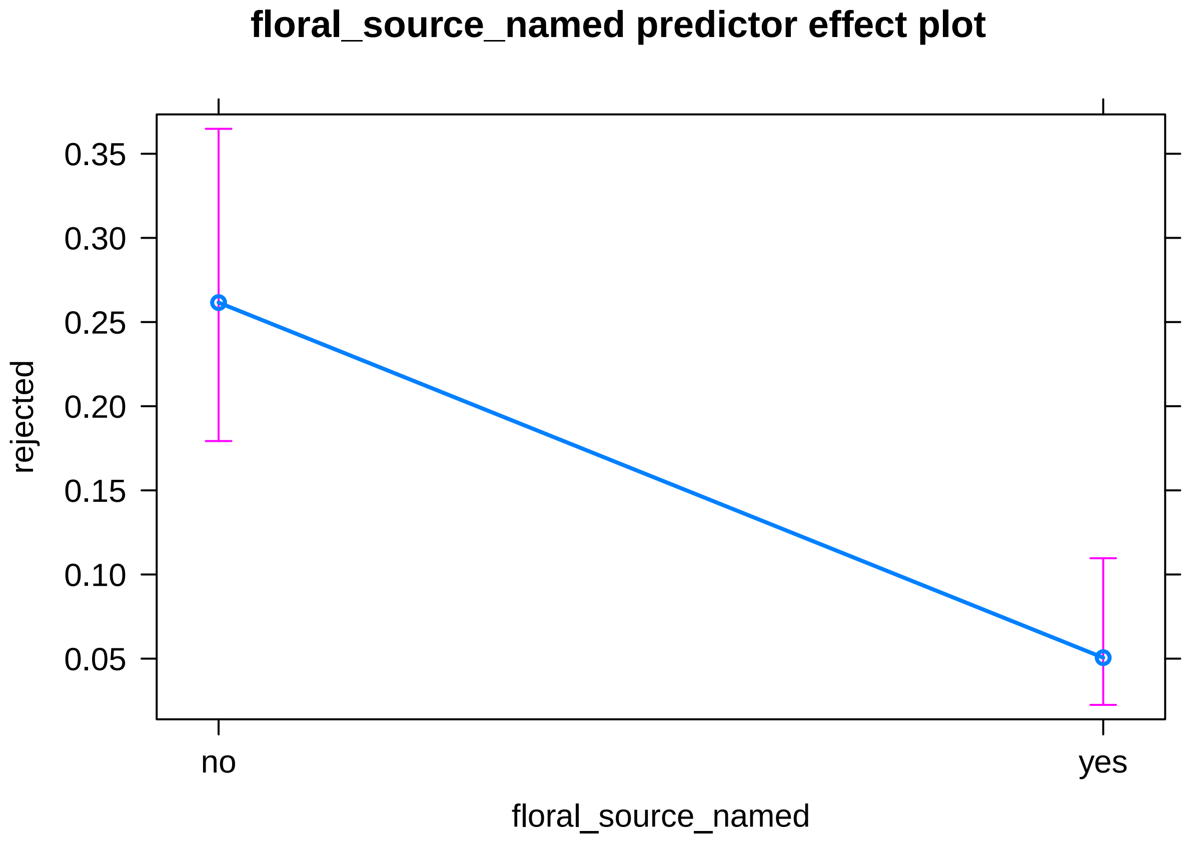
For this predictor, the reference level is East Asia & Pacific. Samples from Europe & Central Asia and South Asia are more likely to be rejected, the former of which shows a small magnitude of difference from East Asia & Pacific and the latter of which shows a much larger increase in likelihood, demonstrated on the plot and in the sign and magnitude of the estimates. At reference levels for the other predictors (sample from BC and no declared floral source), the predicted probability for a rejected sample from Europe & Central

Asia was 0.16 (logit difference = +0.286, *SE* = 0.534, *z* = 0.536, p = 0.59) and it was 0.47 for a sample from South Asia (logit difference = +1.823, *SE* = 0.565, *z* = 3.225, p = 0.001).

All other origin regions were less likely than East Asia & Pacific to have a sample rejected, demonstrated both on the plot and by the sign of the estimate. The smallest reduction in likelihood was for samples Middle East & Africa with a predicted probability of 0.11 (logit difference = -0.159, *SE* = 0743, *z* = -0.215, p = 0.83), followed by samples of mixed origin with a predicted probability of 0.049 (logit difference = -1.026, *SE* = 0.700, *z* = -1.466, p = 0.14), and samples from North America with a predicted probability of 0.046 (logit difference = -1.102, *SE* = 0.751, *z* = -1.466, p = 0.14). The largest reduction in likelihood was for samples from Latin America & the Caribbean with a predicted probability of 0.02 (logit difference = -2.090, *SE* = 1.105, *z* = -1.891, p = 0.059). No estimates reached significance at the alpha = 0.05 level, with the exception of the estimate for South Asia (logit difference = +1.823, *SE* = 0.565, *z* = 3.225, p = 0.001), indicating that for all the other regions, there is not enough evidence to reject the null hypothesis of no differences in likelihood between any given origin region and the East Asia & Pacific region. The estimate for Latin America & the Caribbean is approaching significance (logit difference = -2.090, *SE* = 1.105, *z* = -1.891, p = 0.059), but without more data, the reliability of this effect cannot be determined.

Finally, the floral_source_declared effect plot.

```r
# plotting the effect for floral_source_named with the same syntax as
# the first plot
plot(predictorEffects(mod_full, "floral_source_named"), axes=list(y=list(type="response")))
```



And transforming the estimate into a probability.

```r
# saving the estimate for floral_source_namedyes into a variable
slope_floral_named_yes <- summary(mod_full)$coefficients[11,1]
```

```
# using the plogis() function on the intercept and the estimate for named
# floral source to get the probability of rejection
plogis(intercept+slope_floral_named_yes)
```

```
## [1] 0.02137702
```

This was the most significant effect, with a large negative estimate indicating that honeys that had declared a floral source were significantly less likely to be rejected. At the reference level of all the other predictors (samples from BC, with an origin region of East Asia & Pacific), the estimated probability is 0.02 for a sample with a named floral source (logit difference = -1.894, *SE* = 0.461, *z* = -4.105, p < 0.0001). This is indicates that under the null hypothesis (no difference between samples with and without a declared floral source), these results are very unlikely.

Given that this predictor seems to be very strong, compared to the others, an AIC model comparison between the full model detailed above and a model with only the named floral source predictor was used to determine which was a better fit.

```
# creating a model with just floral_source_named as a predictor
mod_floral <- glm(rejected ~ floral_source_named,
                  data=honey,
                  family="binomial")

AIC(mod_full, mod_floral)
```

```
##            df      AIC
## mod_full   11 210.6894
## mod_floral  2 238.5129
```

The lower AIC of the full model shows that the other factors are contributing towards making the model more representative of the data.

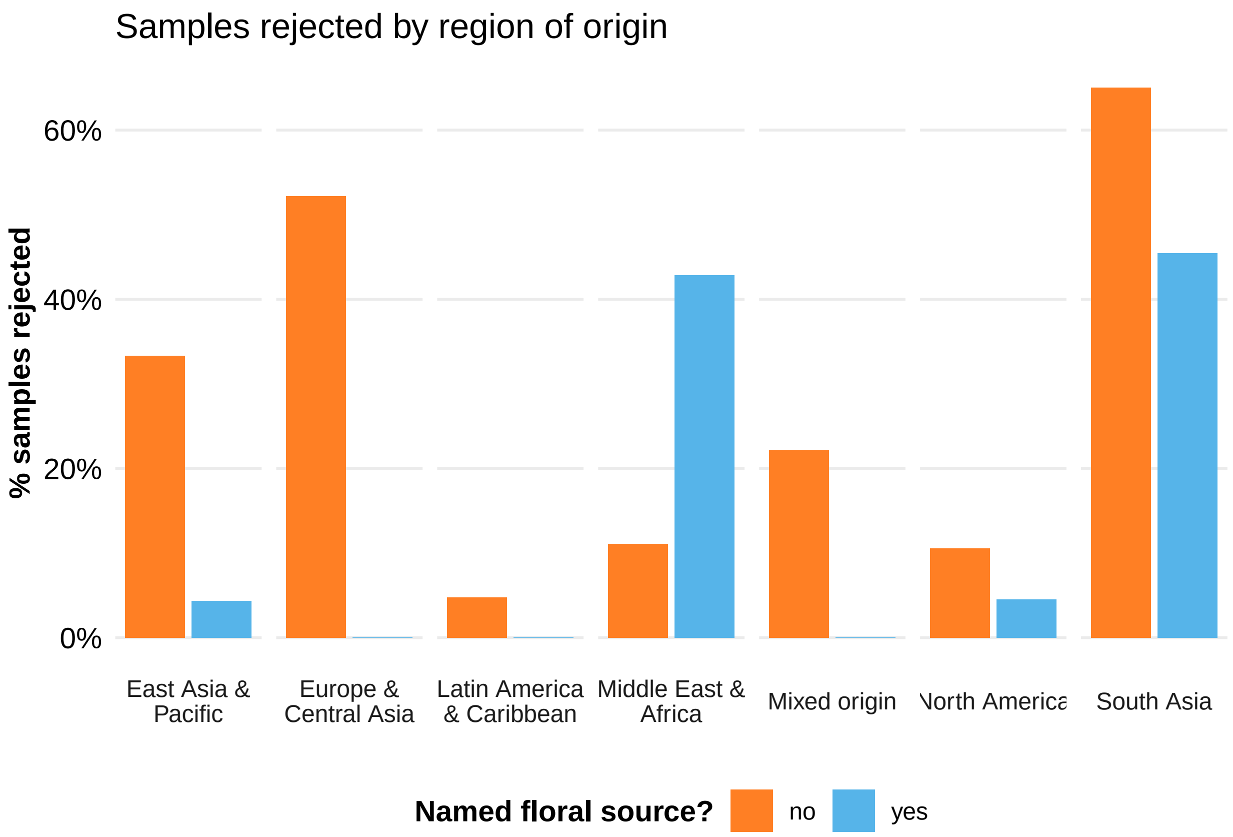
Plotting

The initial questions asked were very broad, and the general idea was to try and determine how each factor (of origin country, sample province, and floral source) influenced the likelihood of a sample being rejected. After looking at the model, seems like there are some general trends, and some statistically significant effects: The main goal of the plots is to demonstrate the named floral source effect, across both region of origin and province of sample, and the secondary goal is to indicate general trends for rejection rates across region of origin and province of sample.

```
# creating a summarized version of the honey dataset that contains the
# proportion of samples that were rejected by origin_region and whether they
# had a named floral source
region_plot <- honey %>%
  group_by(floral_source_named, origin_region) %>%
  summarize(prop = (sum(rejected)/length(rejected))) %>%
  ungroup()
```

```
## `summarise()` regrouping output by 'floral_source_named' (override with `.groups` argument)
```

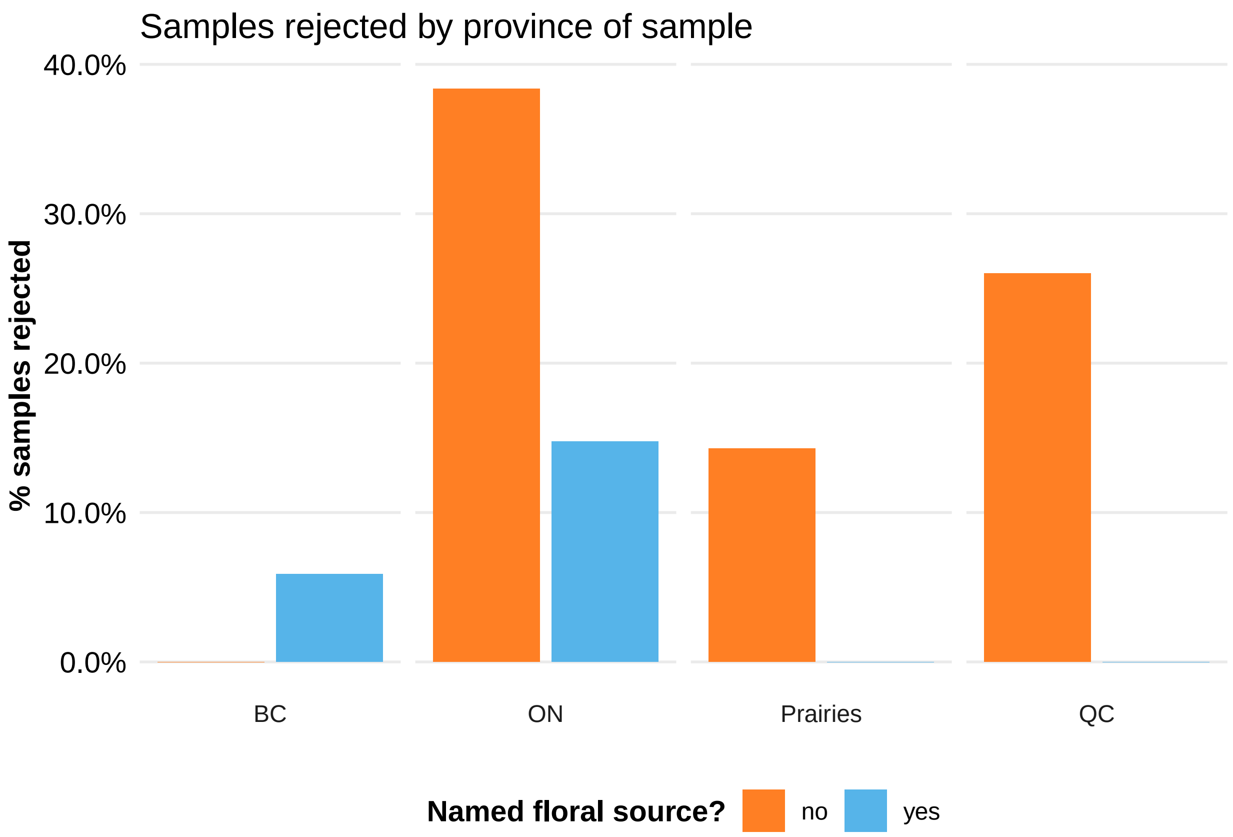
```r
# initializing ggplot, specifying the data source (hon_plot) and aesthetic
# mappings (different levels of floral_source_named on the x axis and the
# proportion of samples that were rejected within those levels)
ggplot(data=region_plot, aes(x=floral_source_named, y=prop)) +
  # adding a column geom, and specifying that the colours should correspond to
  # the levels of floral_source_named
  geom_col(aes(fill=floral_source_named)) +
  # labelling the y axis in terms of percentages
  scale_y_continuous(labels=scales::percent) +
  # creating a grid of plots
  facet_grid(
    # creating a plot for each level of origin_region
    cols = vars(origin_region),
    # moving the origin_region labels to the bottom of the plot
    switch="x",
    # specifying that the origin_region labels have a set width of 14 and
    # that they should wrap onto multiple lines
    labeller=label_wrap_gen(width=14,multi_line=TRUE)) +
  # manually setting the colours of the columns, specifically with a fill
  # command, because the fill aesthetic for geom_col() was specified earlier
  scale_fill_manual(values=c("chocolate1", "#56B4E9")) +
  # specifying the minimal theme
  theme_minimal() +
  # a command for various aesthetic changes to the plot
  theme(
    # removes x-axis grid lines
    panel.grid.major.x = element_blank(),
    # removes major y-axis grid lines
    panel.grid.minor.y = element_blank(),
    # specifying the font size and colour for the axis labels
    axis.text = element_text(size=11, colour="black"),
    # specifying the font size, face, and colour for the axis titles
    axis.title = element_text(size=11, face="bold", colour="black"),
    # removing the x-axis title
    axis.title.x = element_blank(),
    # removing the x-axis labels
    axis.text.x=element_blank(),
    # removing the x-axis tick marks
    axis.ticks.x=element_blank(),
    # specifying the font face for the legend title
    legend.title = element_text(face="bold"),
    # moving the legend to the bottom
    legend.position = "bottom"
  ) +
  # adding a title
  labs(title = "Samples rejected by region of origin") +
  # adding a y-axis label
  ylab("% samples rejected") +
  # changing the title of the legend
  guides(fill=guide_legend(title="Named floral source?"))
```



```
# creating a summarized version of the honey dataset that contains the
# proportion of samples that were rejected by sample_prov and whether they
# had a named floral source
prov_plot <- honey %>%
  group_by(floral_source_named, sample_prov) %>%
  summarize(prop = (sum(rejected)/length(rejected))) %>%
  ungroup()
```

```
## 'summarise()' regrouping output by 'floral_source_named' (override with '.groups' argument)
```

```
# creating the same plot as above, but sorted by province of sampling
ggplot(data=prov_plot, aes(x=floral_source_named, y=prop)) +
  # adding a column geom, and specifying that the colours should correspond to
  # the levels of floral_source_named
  geom_col(aes(fill=floral_source_named)) +
  # labelling the y axis in terms of percentages
  scale_y_continuous(labels=scales::percent) +
  # creating a grid of plots
  facet_grid(
    # creating a plot for each level of origin_region
    cols = vars(sample_prov),
    # moving the origin_region labels to the bottom of the plot
    switch="x") +
# manually setting the colours of the columns, specifically with a fill
  # command, because the fill aesthetic for geom_col() was specified earlier
  scale_fill_manual(values=c("chocolate1", "#56B4E9")) +
  # specifying the minimal theme
  theme_minimal() +
  # a command for various aesthetic changes to the plot
  theme(
    # removes x-axis grid lines
    panel.grid.major.x = element_blank(),
    # removes major y-axis grid lines
    panel.grid.minor.y = element_blank(),
    # specifying the font size and colour for the axis labels
    axis.text = element_text(size=11, colour="black"),
    # specifying the font size, face, and colour for the axis titles
    axis.title = element_text(size=11, face="bold", colour="black"),
    # removing the x-axis title
    axis.title.x = element_blank(),
    # removing the x-axis labels
    axis.text.x=element_blank(),
    # removing the x-axis tick marks
    axis.ticks.x=element_blank(),
    # specifying the font face for the legend title
    legend.title = element_text(face="bold"),
    # moving the legend to the bottom
    legend.position = "bottom"
  ) +
  # adding a title
  labs(title = "Samples rejected by province of sample") +
  # adding a y-axis label
  ylab("% samples rejected") +
  # changing the title of the legend
  guides(fill=guide_legend(title="Named floral source?"))
```



Summary

Samples with a declared floral source (such as wildflower or Manuka) was significantly more likely to be pure honey than ones that did not declare a floral source. This is demonstrated in the plots where in each region or province, the highest percentage of samples rejected were for samples without a declared floral source, with the exception of samples from the Middle East & Africa and samples drawn in BC. There was no significant difference in rejection rates between samples drawn from different provinces, however, the plots indicate that there are trends, where samples from Ontario and Quebec are more likely to be rejected. Given that samples from Ontario made up half of the data, more samples from the other provinces are needed to determine whether this is a statistically significant trend or due to chance. As well, samples from South Asia were significantly more likely to be rejected, shown by their higher overall percentages, but more data could determine whether the higher rejection rates for samples Europe & Central Asia are significant or due to chance.

These conclusions were drawn from a small dataset with observational data, and it is possible that more data with equal samples in all cells will reveal more results: For example, more data could confirm or deny the pattern observed in the plot where samples that had a named floral source originating from the Middle East & Africa or drawn in BC were more likely to be rejected than samples without a named floral source, but because the data were too few and not structured to answer these more complex questions, there is no statistically reasoned answer.