CARLETON UNIVERSITY

SCHOOL OF MATHEMATICS AND STATISTICS

HONOURS PROJECT

TITLE: Analysis of COVID-19 Confirmed Cases based on Poisson Loglinear Regression Model

AUTHOR: Junpu Xie

SUPERVISOR: Patrick Farrell

DATE: Dec.28th , 2020

Table of Contents

Abstract

Nowadays, COVID-19 is a severe health threat for individuals all over the world. Its appearance has dramatically changed people's daily lifestyle, and its extended social problems are also evolving daily.

This paper aims to analyze a COVID-19 confirmed case dataset released by Ontario Open Data Portal; this governmental website has updated several physical variables, such as gender, age group, the city lived, and spread method of every new daily confirmed patient to the dataset. Because these explanatory variables are all categorial data and we can summarize the case counts based on different physical variable combinations, the paper chooses the Poisson Log-Linear regression model from the Generalized Linear Model to investigate, identify, and evaluate those significant physical variables for the coronavirus disease diagnosis. Then, we apply several statistical techniques to improve the fitted regression and use it to estimate the COVID-19 confirmed case counts under different physical variable combinations.

The following paper will introduce the GLM distribution, Poisson Log-Linear Regression Model, and its overdispersion. Then we use R software to visualize the original data resource, further develop the fitted Poisson regression model to estimate the confirmed patient count data. At the end of the paper, we conclude the findings to empower people understanding what physical variables have the greater probabilities of diagnosing COVID-19.

Chapter1. Introduction to GLMs with Poisson Log-Linear Model

1.1 GLMs' Basic Model Structure

There are many methods to investigate the relationship and dependence among different variables in a dataset in statistics. We usually use linear predictors and multiple regression models to analyze those continuous data. There are also diverse methods to manipulate categorical data, such as the two-way contingency table and the Generalized linear models (GLMs), a standard statistical regression model.

The most remarkable feature of GLMs is investigating a definite relationship between the response variables' mean values and the continuous or categorical explanatory independent variables via a link function. Based on the above description, it is easy to see that the three main components outline GLMs. They are the probability distribution function of response variables, linear predictor based on explanatory variables, and the link function between independent variables' linear combinations and dependent variables' expectations. Because GLMs only has three main components, it is a very flexible and generalized regression model so that many standard distributions, such as ANOVA, regression, logistic regression, Poisson distribution, and the binomial distribution, all belong to GLMs. We can see that it is easy to convert their probability density distributions into exponential forms and also GLM's probability distribution functions in the next paragraph.

The probability distributions of the response variable y in GLMs describes the response variables' features; we often select different distribution types based on the original dataset resources' conditions and features. Two standard statistical distributions from GLMs are Poisson distribution and Binomial distribution. When the response variables are binary, we can quickly assign it as the binomial distribution because its output only contains two outcomes. While the response variables are the count of a specific event with the same probability in a particular time, we classify them as Poisson distribution. Two functions in the below block's represent the general GLM's distribution expressions in the exponential forms for one or two unknown parameters separately：

$$f(y_i, \theta_i) = a(\theta_i)b(y_i)\exp\,[y_i Q(\theta_i)]$$
$$f(y_i; \theta_i, \Phi) = \exp\,[\frac{y_i\theta_i - b(\theta_i)}{a(\Phi)} + c(y_i, \theta)]$$

where i = 1…, N, N is the total number of observations, yi is the ith response variable, θ_i is the ith natural parameter, and Φ is the dispersion parameter, representing the under-dispersion or over-dispersion situation for the mean-response relationship in the fitted model.

Secondly, the linear predictor function is a general linear combination of all significant independent variables and their corresponding estimated coefficients values which can predict the dependent variables' outcome. Its expression function can be written as: $\eta_i = \sum_{j=0}^{p} \beta_j x_{ij}$, where i=1..., N, j = 1…, p, N is the total number of observations, p is the total number of variables, xij is the jth variable with respect to (w.r.t.) its ith observation, βj is the regression parameter based on its jth observed explanatory variable xij, and ηi refers to its ith linear predictor w.r.t. its jth observation.

Link function indicates the association between the overall mean of dependent variables and the linear predictor of independent variables; its expression can be written as $g(\mu_i) = \eta_i = \sum_{j=0}^{p} \beta_j x_{ij}$. Like the first component in GLMs of response variables' distribution expression, link function expression also has many different expression forms. The most well-known link function would be an identity link function $\mu_i = \eta_i = \sum_{j=0}^{p} \beta_j x_{ij}$; it implies that the response variable's expected value is precisely equal to the linear predictor about independent variables. The canonical link function of $g(\mu_i) = \mathrm{Q}(\theta_i) = \eta_i = \sum_{j=0}^{p} \beta_j x_{ij}$ suggests a transformation between the mean response function $g(\mu_i)$ and the natural parameter function $Q(\theta_i)$. Based on various format changes among different statistical distributions, the link function can also be quickly turned to inverse squared, logarithm, logistic and other forms to present the relationships between dependent variables' mean response μ_i and the estimated linear predictors η_i of the GLM.

After introducing GLM's basic structure, the following paper will show how to interpret the most probable GLM. The first step of demonstrating fitted GLM is to identify the specific model of observed response based on the dataset's feature; then, we will use two general techniques to discover the most probable estimators in the next subsection. One is the Maximum Likelihood Estimation (MLE), the standard method to find the feasible parameter estimators. Other statistical techniques are the iterative reweighted least square of Newton-Raphson and Fisher Scoring Method, which are often used to solve nonlinear likelihood equations. After examining the estimated parameters, our next interest is to investigate the deviance value between the predicted and the observed response variable w.r.t. degree of freedom. In the end, we can strongly define whether the fitted model's adaptation to the original data based on hypothesis testing and test statistic of observed deviance. After interpreting the most probable GLMs, the following paper indicates the Poisson Log-linear regression model and its overdispersion situation due to unequal mean-response relationship.

1.2 Estimate Unknown Parameters – Maximum Likelihood Estimation

The parameter estimation based on the sample data can demonstrate a corresponding fitted model. It shows the relationship between dependent variables and independent variables and infers the population data further predicts the future outcome. So, the process of estimating the unknown parameters is one of the most fundamental but critical aspects of data modeling.

The maximum likelihood estimation (MLE) statistical technique is commonly employed to interpret the parameter estimation based on sample data. The basic idea of finding MLE is to calculate its log-likelihood functions' derivatives w.r.t. the corresponding parameter, then apply two MLE properties to it so that we can interpret the best MLE.

In the following block, we have investigated GLM's likelihood function L, a joint probability distribution of a sample of N observations. This expression shows the likelihood function of GLM's probability distribution with two unknown parameters.

$$L = \prod_{i=1}^{N} L_i = \prod_{i=1}^{N} f(y_i; \theta_i, \Phi) = \prod_{i=1}^{N} [\frac{y_i\theta_i - b(\theta_i)}{a(\Phi)} + c(y_i, \Phi)]$$

where i represents the ith observation, θi is the ith natural parameter for ith observation, and Φ is the dispersion parameter.

By looking at the likelihood function expression, our first objective of demonstrating GLMs is to find MLE for natural parameter θi, located at its exponential term. The dispersion parameter is unnecessary to estimate as it only represents the mean-variance relationship.

Because the likelihood function is in the cross-product form, it is hard to interpret its MLE because of its derivative w.r.t. the unknown parameter θi is in mixed-parameters form. The most convenient way to get the function's derivative in the additive form is to convert the likelihood functions to their transformed forms, such as a log, square roots, reciprocal, etc. As we all know, the log-likelihood function is always continuously differentiable and monotone increasing; so, it can be used to interpret the unknown parameter and examine the global ML estimator. Thus, the log-likelihood function is a standard transformed method to investigate the most probable ML estimates. The latter equation shows the log-likelihood function's general expression w.r.t. the exponential family with two unknown parameters.

$$lnL = \ln (\prod_{i=1}^{N} L_i) = \sum_{i=1}^{N} \ln \left[f(y_i; \theta_i, \Phi)\right] = \sum_{i=1}^{N} [\frac{y_i\theta_i - b(\theta_i)}{a(\Phi)} + c(y_i, \Phi)]$$

After converting the function into the log-transformed form, it is easy to find the most probable parameter estimation by calculating its new-transformed derivative and further applying two MLE properties. The first property is that the expected value of its first derivative is equal to zero; another one is that the second derivative of a log-likelihood function is the same as the negative double square of its first derivative. So, we can use these two properties to investigate the most probable parameter estimation.

In summary, using the log-likelihood derivative function can evaluate all possible solutions for parameter estimation. Moreover, applying MLE principles can effectively maximize the log-likelihood function at a boundary of parameter space to conclude the most probable estimates by using the MLE technique.

The following function expression block explains the method of investigating the MLE w.r.t. the natural parameter θ_i, which includes interpreting the first and second derivatives, two expected value equations according to MLE properties and the parameter estimation results.

$$\frac{\partial lnL_i}{\partial \theta_i} = \frac{\partial ln[f(y_i; \theta_i, \Phi)]}{\partial \theta_i} = \frac{\partial ln[\frac{y_i\theta_i - b(\theta_i)}{a(\Phi)} + c(y_i, \Phi)]}{\partial \theta_i} = \frac{y_i - b'(\theta_i)}{a(\Phi)}$$

$$\frac{\partial^2 lnL_i}{\partial \theta_i^2} = \frac{\partial [\frac{y_i - b'(\theta_i)}{a(\Phi)}]}{\partial \theta_i} = -\frac{b''(\theta_i)}{a(\Phi)}$$

$$E\left(\frac{\partial lnL_i}{\partial \theta_i}\right) = E\left(\frac{y_i - b'(\theta_i)}{a(\Phi)}\right) = E\big(y_i - b'(\theta_i)\big) = 0$$

$$-E\left(\frac{\partial^2 lnL_i}{\partial \theta_i^2}\right) = E\left(\frac{b''(\theta_i)}{a(\Phi)}\right) = \frac{Var_\theta(Y_i)}{[a(\Phi)^2]} = E[(\frac{Y_i - b'(\theta_i)}{a(\Phi)})^2] = \frac{Var(Y_i)}{[a(\Phi)]^2} E[(\frac{\partial L}{\partial \theta_i})^2]$$

$$\mu_i = E(Y_i) = b'(\theta_i), \qquad Var(Y_i) = b''(\theta_i)a(\Phi)$$

After getting the natural parameter estimation, we can demonstrate the fitted distribution function w.r.t. response variable y; however, the complete GLM requires three components. Except for the probability distribution about y, we also need to interpret the link function $g(\mu_i) = \eta_i$ and linear predictor $\eta_i = \sum_{j=0}^{p} \beta_j x_{ij}$, which leads us to estimate another important unknown coefficient parameter β_j in both link function and linear predictor.

Because the best estimation for natural parameter Θ is already determined, the following block shows several functions about the unknown parameter β. They are the derivative of log-linear regression function w.r.t. parameter β based on known parameter θ_i, the linear predictor and the link function. The last two functions are the expressions which follow the MLE properties. We can use these equations to investigate the most probable estimates of the second parameter β w.r.t. the explanatory variable.

$$\frac{\partial L_i}{\partial \beta_j} = \frac{\partial L_i}{\partial \theta_j}\frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \beta_j} = \left(\frac{y_i - \mu_i}{a(\Phi)}\right)\left(\frac{a(\Phi)}{Var_\theta(Y_i)}\right)\left(\frac{\partial \mu_i}{\partial \eta_i}\right)\mathrm{x}_{ij} = \frac{\mathrm{x}_{ij}(y_i - \mu_i)}{Var_\theta(Y_i)}\frac{\partial \mu_i}{\partial \eta_i}$$

$$E\left(\frac{\partial lnL(\beta)}{\partial \beta_j}\right) = E(\sum_{i=1}^{N}\frac{\partial L_i}{\partial \beta_j}) = \sum_{i=1}^{N}(\frac{\mathrm{x}_{ij}(y_i - \mu_i)}{Var_\theta(Y_i)}\frac{\partial \mu_i}{\partial \eta_i}) = 0$$

$$\mathrm{E}\left(\frac{\partial^2 lnL(\beta)}{\partial \beta_j \partial \beta_h}\right) = -E\left[\frac{\partial lnL}{\partial \beta_j}\frac{\partial lnL}{\partial \beta_h}\right] = -E\left[\frac{\mathrm{x}_{ij}(y_i - \mu_i)}{Var_\theta(Y_i)}\frac{\partial \mu_i}{\partial \eta_i}\frac{\mathrm{x}_{ih}(y_i - \mu_i)}{Var_\theta(Y_i)}\frac{\partial \mu_i}{\partial \eta_i}\right] = -\frac{\mathrm{x}_{ih}\mathrm{x}_{ij}}{Var_\theta(Y_i)}(\frac{\partial \mu_i}{\partial \eta_i})^2$$

It is easy to see that the likelihood equations are based on corresponding response variables' variance. Different distribution functional form results in different variances, such as Poisson's variance is its mean response. Thus, it is hard to interpret the coefficient parameter β_j based on different link functions and different distribution functions. To solve the nonlinear likelihood function and interpret the best parameter, we will introduce two general iterative algorithms: The Newton-Raphson and Fisher Scoring methods.

1.3 Determine the Best Maximum Likelihood Estimates – Newton-Raphson Iterative Method & Fisher Scoring Method

To solve the non-linear likelihood function, it implies that there might exist many probable estimates. So, we are going to use the vector and matrix forms to present the likelihood function.

According to the definition of MLE, the first derivative of the log-likelihood function w.r.t. its first unknown parameter θ is named as score function U(θ), where θ is the vector of an ordered parameter for the selected distribution. When the function estimates more than one unknown parameter, the information matrix J(β) based on the estimated score functions U(Θ), another critical statistical syntax for MLE, is introduced. Its expression is

$$J(\beta) = -\frac{\partial^2 lnL}{\partial\beta\partial\theta} = -\frac{U(\theta)}{\partial\beta} = \frac{\partial U(\theta)}{\partial\mu}\frac{\partial\mu}{\partial\beta}$$

The information matrix J(β) represents the negative expected value of the second derivatives of the log-likelihood function and the desired information matrix I(β). So, two MLE properties can also apply to the information matrix J(β), which are the score vector's expected value is zero, and the variance of the score function U(Θ) is equal to the expected value of the score times its inverse matrix.

After understanding the vector forms of likelihood functions and their parameter estimations, our next step is to apply these two iterative techniques to further interpret the most probable parameters. The Newton-Raphson iterative method, which is based on linear approximation, is one of the most simple but powerful techniques to investigate the globally optimized maxima. It takes the initial assumption of the unknown parameter estimator and applies the iterative function to generate the most reasonable parameter, which is the closest to the real root r. At the beginning of the iteration, the distance between the initial assumption for the parameter estimator x0 and its real root r, $h = r - x_0$, is significantly closed but not equal. Since the value of h between two values is remarkably small so that h can also be written as $-\frac{f(x_0)}{f\prime(x_0)}$. Thus, we can conclude that the related functions about the root r as

$$0 = f(x) = f(x_0 + h) = f(x_0) + hf'(x),$$
$$r = x_0 + h \approx x_0 - \frac{f(x_0)}{f'(x_0)}$$

Based on the initial assumption of x_0 and the iterative function of $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$, it is easy to set the iterative function about the next closed estimate to the root of x_{n+1} according to the former variable x_n. The only thing needs to notice is that L(β) has many local maxima at which the derivative of L(β) equals to 0, so the assumption of initial guess should be careful.

Thus, all the above functions can also be written in the matrix form, which are shown in the following block to better review the results. Iterations will stop until the changes in $L(\beta^{(t)})$ between steps are sufficiently small.

$$u' = [\frac{\partial L(\beta)}{\partial \beta_0}, \frac{\partial L(\beta)}{\partial \beta_1}, \dots, \frac{\partial L(\beta)}{\partial \beta_p}],\ h_{ab} = \frac{\partial^2 L(\beta)}{\partial \beta_a \partial \beta_b}$$
$$L(\beta) \approx L(\beta^{(t)}) + u'^{(t)}(\beta - \beta^{(t)}) + \frac{1}{2}(\beta - \beta^{(t)})' H^{(t)} (\beta - \beta^{(t)})$$
$$\frac{\partial L(\beta)}{\partial \beta} \approx u^{(t)} + H^t(\beta - \beta^{(t)}) = 0$$
$$\rightarrow \beta^{(t+1)} = \beta^{(t)} - (H^{(t)})^{-1} u^{(t)}$$

Fisher Scoring is another iterative method for solving likelihood equations which has a similar idea foundation as the Newton-Raphson method. The only difference between them is the Hessian matrix term. Fisher scoring method uses the expected value of the Hessian matrix, which is also called the expected information J(β), as the iterative distance measure, and the Newton-Raphson method applies the Hessian matrix itself. The reason that the Fisher Scoring method uses the expected information rather than the practical information because the expected (Fisher) information matrix J(β) has a more straightforward expression form than the experimental information matrix. Moreover, it contains sufficient information about β, so J(β) is usually applied for the general iteration.

So, in Fisher Scoring method, we let $J^{(t)}$ denotes the expected information matrix at the t-th iteration and its element, the formula of Hessian matrix, can be written as $h_{ab} = \frac{\partial^2 lnL(\beta)}{\partial\beta_a \partial\beta_b}$. Its expression is similar as the Newton-Raphson iterative method, which is $\beta^{(t+1)} = \beta^{(t)} + (J^{(t)})^{-1}u^{(t)}$, where $\beta^{(t)}$ implies the t-th iteration about β estimator, $J^{(t)}$ is the t-th iteration about the expected information matrix, and $u^{(t)}$ denotes the vector about the t-th likelihood derivatives. Hence, Fisher Scoring also can be used to do iteration until the value of $\beta^{(t+1)}$ reaches the closest point to the real root r for solving the non-linear derivative equations for the likelihood function.

1.4 Examine the Mean Value – Deviance and Goodness of Fit Test

After interpreting the best ML estimators about natural parameter Θ in the response variable distribution and coefficient tetrameter β in the linear predictor, we can easily demonstrate a fitted model based on sample data; however, we are still unsure about the mean response variable μ. We examine the best estimate's position for the mean response value by calculating its residual deviance; it is the difference of response variable between the fitted model and the ideal model, which shows how closely the desired model fits the original data resource by interpreting the statistical hypothesis.

In GLM, there are two deviances between the three models. They are the residual deviance between the saturated model and the fitted model and the null deviance between the saturated model and the fitted model. For these three different types of models, the saturated model is a theoretical model with many parameters for each observation, which results in the perfect fit from the sample data. The fitted model has several estimated parameters of β and Θ, and the null model assumes that the response variable can be calculated by only one null parameter.

Since the residual deviance calculates the difference between the fitted model and the saturated one, we will apply it to examine the fitness between predicted response values and the actual dependent values. The method of interpreting the residual deviance is similar to investigating the natural parameter Θ and linear parameter β. We apply the likelihood functions in terms of corresponding outcome variables, the actual dependent variable of y and predicted mean response variable of $\hat{\mu}$ for the models, so that the residual deviance is the difference between L (y, y) and L (y, $\hat{\mu}$), its formula can be expressed as $D(y, \hat{\mu}) = 2[L(y, y) - L(y, \hat{\mu})]$.

Because the most probable estimator about $\hat{\mu}$ shows that the fitted model based on $\hat{\mu}$ s the most closed to the observed values, we try to minimize the residual deviance to obtain the better-fitted model to maximize the log-likelihood function of the fitted model. This idea tells us that we can obtain the minimum value of residual deviances while computing the most probable estimates of the parameters and calculating the fitted value $\hat{\mu}$ at each the specific iteration.

Since residual deviance measures how closely fitted model's predictions are to the observed outcomes, it presents how much variation we would expect in the observed outcomes around the predicted means. Therefore, we might consider using it as the basis for the goodness of fit test to test the statistical hypothesis that the GLM holds against the alternative that a more general model holds. According to the definition of Chi-square distribution, we understand it is a measure of the difference between the observed and expected frequencies of the outcomes. So, it can examine the variables' independences and test the goodness-of-fit between the observed distribution and the theoretical distribution. Because Chi-square distribution also depends on the model's parameter sizes, the residual deviance has approximately a Chi-square null distribution with the degree of freedom of (n – p) under regularity conditions, where n is the total number of independent variables and p is the number of estimated parameters.

After understanding that the residual deviance with the degree of freedom follows the Chi-square distribution, our next step is to investigate the goodness of fit test for the fitted model. So, we can use the residual goodness-of-fit test to examine whether the residual deviance ratio to the degree of freedom, number of unknown parameters, is equal to one. Hence, we can set the null hypothesis as the fitted model M0 does hold, which is equivalent to all estimated coefficients βh's equal to zero, where h is the difference between the fitted model and the saturated model. Zero value indicates that the fitting of the model of interest is substantially like that of the most completed model that can be built; the alternative hypothesis is that the fitted model does not hold, suggesting that the model we built might not have good fitness to the sample data.

1.5 Poisson Log-Linear Model

The Poisson distribution is one of the Generalized Linear Model and the best model to interpret the counts data. By following the Poisson identification and GLM modeling processes, we can easily set the following functions and investigate the Poisson log-linear model by applying the canonical link function.

Poisson distribution in Exponential Natural Family:

$$f(y;m) = \frac{e^{-m}m^{y}}{y!} = \exp\,(-m)(y!)^{-1}\exp\,(ylnm)$$

Poisson Loglinear model is distributed by using the canonical link function:

$$g(\mu_i) = Q(m_i) = ln(m_i) = \eta_i = \sum_{i=1}^{p} \beta_j x_{ij}$$

The likelihood function of Poisson distribution:

$$L = \sum_{i=1}^{N} L_i = \sum_{i=1}^{N} [\frac{y_i - b'(\theta_i)}{a(\Phi)} + c(y_i, \Phi)] = \sum_{i=1}^{N} [y_i - \exp(m_i) - \ln(y_i!)]$$

Two MLE equations related to Poisson distribution:

$$\mu_i = E(Y_i) = b'(\theta_i) = \exp(m_i) = m_i$$

$$\mu_i = Var(Y_i) = b''(\theta_i)a(\Phi) = \exp(m_i) = m_i$$

Deviance from the Poisson loglinear model with $y_i = \widehat{m}_i$:

$$D(y;\widehat{m}) = 2\sum_{i=1}^{N}\left[y_i ln\frac{y_i}{\widehat{m}_i} - y_i\right] = 2\sum_{i=1}^{N} y_i \ln\left(\frac{y_i}{\widehat{m_\iota}}\right)$$

The general function to connect the linear predictor of the explanatory variables with the mean response variables:

$$\ln(m_i) = \sum_{j=0}^{p}\beta_j x_{ij} = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$

$$m_i = \exp\left[\sum_{j=0}^{p}\beta_j x_{ij}\right] = \exp(\beta_0)\exp(\beta_1 x_{i1}) \ldots \exp(\beta_p x_{ip})$$

After understanding the Poisson regression distribution's basic formulas, we also need to know four fundamental assumptions. Firstly, the response variable must be the count data, which presents in a fixed time. The observations are all independent. The mean response variable must be identical to its variance, and the log of the mean rates must be the linear function.

The assumption of an equal mean-variance relationship in the Poisson regression distribution is ideal because the variation among the variables is often more significant than the mean values in the real situation, which results in overdispersion. So, there are two useful methods to avoid this situation. Firstly, we demonstrate the Quasi-likelihood regression model based on the Poisson regression model by estimating the dispersion parameter to adjust the response variables' standard error. Another method is to fit the dataset into the negative binomial regression model because it had similar fundamental assumptions about the Poisson regression distribution. The following two subsections are going to explain these two models.

1.6 Overdispersion – Quasi-likelihood Poisson Model

Except for using residual deviance based on likelihood theory to test the model's fitness, another common test statistic called Pearson Chi-squared statistics can be employed. So we can apply this parameter to the Quasi-likelihood Poisson Regression Model. According to the generalized Pearson statistics, we have its formula: $X^2 = \sum_{i=1}^{N} \frac{(y_i - \widehat{\mu}_i)^2}{\widehat{\mu_i}} = \sum_{i=1}^{N} \frac{(y_i - \widehat{m}_i)^2}{\widehat{m_i}}$

The Poisson assumption is often realistic for count data due to overdispersion, where the variance exceeds the mean. So, in the Quasi-likelihood regression model, we create the new expression form of $Var(\mu_i) = \Phi\mu_i = \Phi m_i$ to adjust the unequal mean-variance relationship. In this formula, the dispersion parameter Φ is a fixed value, and mi is the estimate for the response variable’s mean value.

After we estimate the new unknown parameter Φ, we can apply it to the standard error term and the generalized Pearson statistics formula from the Poisson regression model. So, the estimated dispersion parameter Φ can inflate standard errors, and we also obtain the new Pearson test statistics of $x^2/_{\Phi}$. It also approximately follows Chi-square distribution w.r.t. to its degree of freedom of n-(p+1). The (p+1) term appears that we have one more parameter estimation of the dispersion parameter. Then, we can easily find the estimated value under methods of moments estimation and calculate the dispersion parameter value based on the following formula: $\Phi \approx E[\frac{X^2}{n-(p+1)}] = \frac{X^2}{n-(p+1)} = \frac{\sum_{i=1}^{n}(y_i - \widehat{\mu}_i)^2}{n-(p+1)}$

After getting the dispersion parameter Φ’s estimated value from its Pearson test statistics, we can get new variance and standard error terms for the Quasi-likelihood Poisson regression model by multiplying the original terms from the Poisson regression model with $\widehat{\Phi}$ and $\sqrt{\widehat{\Phi}}$ separately. Thus, the new mean-variance relationship expression in the Quasi-likelihood model can be shown as $Var(\mu_i) = \Phi\mu_i = \Phi m_i$.

1.7 Overdispersion – Negative Binomial Distribution

Another approach to deal with overdispersion is to apply the negative binomial regression model instead of the Poisson regression model. Its distribution formula in original and exponential forms, the mean and variance expressions are shown below:

$$f(y) = \frac{\gamma(y+k)}{\gamma(k)\gamma(y+1)}(\frac{\mu}{\mu+k})^{y}(\frac{k}{\mu+k})^{k}$$

$$f(y_i;\theta_i,\Phi) = \exp\,[\frac{y_i ln\frac{\theta_i}{1-\theta_i} - \ln\,(1-\theta_i)}{{}^{1}/_{n_i}}]\binom{n_i}{n_i y_i}$$

$$E(Y) = \mu\ \ Var(Y) = \mu + \frac{\mu^2}{k} = \mu + \Phi\mu^2$$

There are two advantages that we utilize negative binomial distribution rather than Poisson distribution. Firstly, the negative binomial distribution requires more than one parameter to build a more flexible fitted model. According to the negative binomial distribution's extended properties about the mean-variance relationship, we can easily see that its variance of the response value is equal to its mean value and mean value ratio to the second parameter k. This feature strongly indicates that applying negative binomial distribution can better fit the data with the unequal mean-variance relationship.

In this situation, 1/k becomes the dispersion parameter Φ in the GLM exponential form. Hence, if k is fixed, the negative binomial distribution will belong to its natural exponential family. According to its variance formula, we can see that as the k value increases, the ratio term of its mean to k value is more closed to zero, which leads the distribution gets more closed to the Poisson distribution. Moreover, if k is unknown, we will find its estimated value by applying the maximum likelihood function and the Newton-Raphson iterative method, which is the same as interpreting the unknown parameter β in the linear predictor.

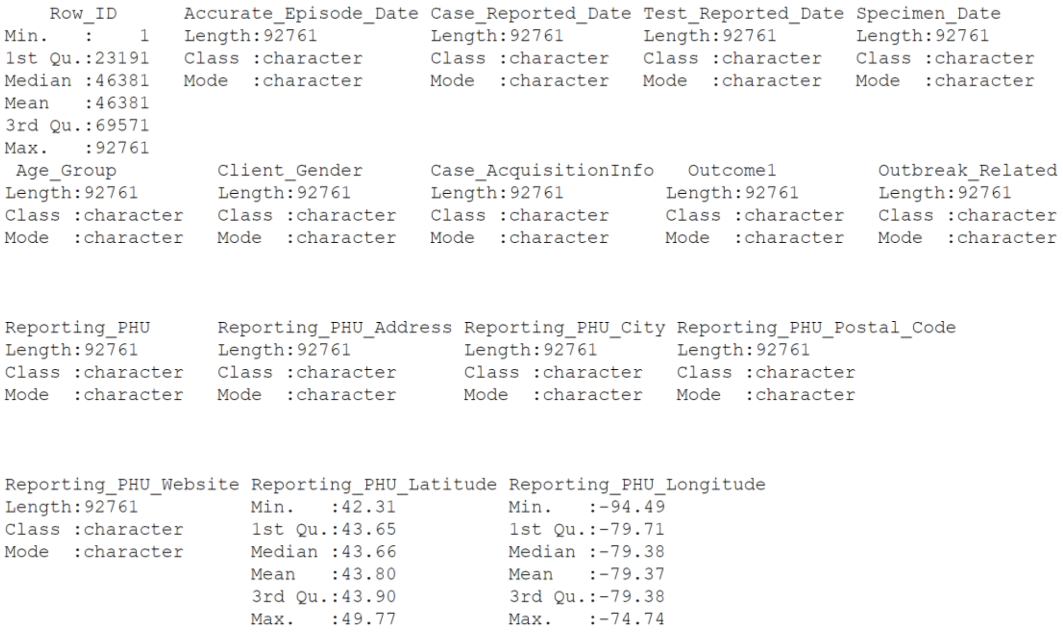
Chapter 2 – Understanding COVID-19 Data

2.1 Data Provider and Data Description

Nowadays, COVID-19 is a severe health threat for individuals, and its extended social problems are evolving worldwide. Based on the number of new daily cases in Canada, Canadians' risk is still considered high. In this situation, recorded COVID-19 data plays a vital role in managing a pandemic because data analysis and data modeling provides the public with insights about the coronavirus-related forthcoming trends.

Canada's government has put reasonable effort to make sure the public gets COVID-19 news and data updates on time. Many Canadian governmental organizations offer open data portal about coronavirus-related and diverse-subject data. So, people can easily understand the COVID-19 current situation across Canada and even the world by reviewing the original dataset, visual data gallery, and interactive data platform.

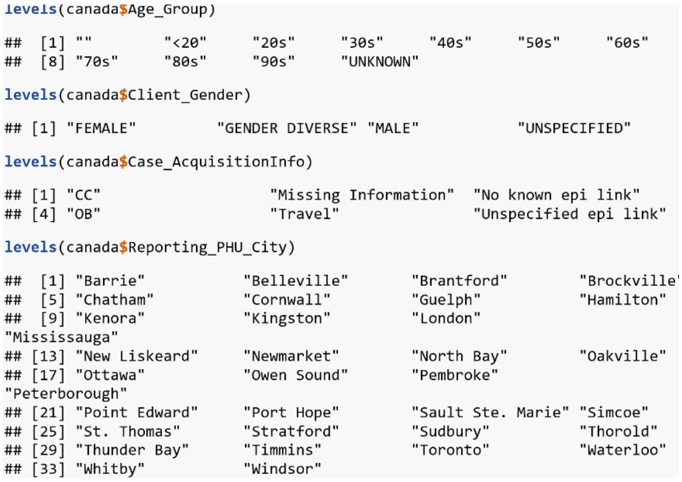
This case study chooses an open dataset about the daily COVID-19 confirmed patient cases from Ontario Data Catalogue. It records personal background information about the patients who have been tested as positive in Ontario. The dataset updates the new patient information daily; this paper only uses the dataset with the last updated date of November 15th to manipulate. The dataset variables are the test reported date, age group, gender, case acquisition information, patient's outcome, the reporting Public Health Unit (PHU), and the city where the PHU is located. The following table shows all variable information by using the summary() function in R. From the this simple summary table, we can see that this dataset contains 92,761 patient observations and 16 different measure variables. Except for variables of the reported data, PHU's websites, and their geographic coordinates, the rest of the variables are all categorical data.



```
    Row_ID      Accurate_Episode_Date Case_Reported_Date Test_Reported_Date Specimen_Date
 Min.   :    1  Length:92761          Length:92761       Length:92761       Length:92761
 1st Qu.:23191  Class :character      Class :character   Class :character   Class :character
 Median :46381  Mode  :character      Mode  :character   Mode  :character   Mode  :character
 Mean   :46381
 3rd Qu.:69571
 Max.   :92761
  Age_Group          Client_Gender      Case_AcquisitionInfo   Outcome1          Outbreak_Related
 Length:92761       Length:92761       Length:92761          Length:92761       Length:92761
 Class :character   Class :character   Class :character      Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character      Mode  :character   Mode  :character


 Reporting_PHU      Reporting_PHU_Address Reporting_PHU_City Reporting_PHU_Postal_Code
 Length:92761       Length:92761          Length:92761       Length:92761
 Class :character   Class :character      Class :character   Class :character
 Mode  :character   Mode  :character      Mode  :character   Mode  :character


 Reporting_PHU_Website Reporting_PHU_Latitude Reporting_PHU_Longitude
 Length:92761          Min.   :42.31          Min.   :-94.49
 Class :character      1st Qu.:43.65          1st Qu.:-79.71
 Mode  :character      Median :43.66          Median :-79.38
                       Mean   :43.80          Mean   :-79.37
                       3rd Qu.:43.90          3rd Qu.:-79.38
                       Max.   :49.77          Max.   :-74.74
```

By reviewing the above variable information, some variables in the dataset contain the same information, such as the PHU location expressed in different forms of the located city name, PHU's name, its postal code, longitude, and latitude. The date variables are classified into four: accurate episode date, the case reported date, test reported date, and specimen date. We only select several essential variables and form a small but sufficient-information dataset based on them; those variables are age group, gender, located city, and COVID-19 case acquisition information.

After creating the new dataset from the original one and converting those selected variables into the factorial data type, we can use levels() to see each variable's factor levels.



```
levels(canada$Age_Group)

##  [1] ""        "<20"     "20s"     "30s"     "40s"     "50s"     "60s"
##  [8] "70s"     "80s"     "90s"     "UNKNOWN"

levels(canada$Client_Gender)

## [1] "FEMALE"          "GENDER DIVERSE" "MALE"           "UNSPECIFIED"

levels(canada$Case_AcquisitionInfo)

## [1] "CC"                    "Missing Information"  "No known epi link"
## [4] "OB"                    "Travel"               "Unspecified epi link"

levels(canada$Reporting_PHU_City)

##  [1] "Barrie"           "Belleville"       "Brantford"        "Brockville'
##  [5] "Chatham"          "Cornwall"         "Guelph"           "Hamilton"
##  [9] "Kenora"           "Kingston"         "London"
"Mississauga"
## [13] "New Liskeard"     "Newmarket"        "North Bay"        "Oakville"
## [17] "Ottawa"           "Owen Sound"       "Pembroke"
"Peterborough"
## [21] "Point Edward"     "Port Hope"        "Sault Ste. Marie" "Simcoe"
## [25] "St. Thomas"       "Stratford"        "Sudbury"          "Thorold"
## [29] "Thunder Bay"      "Timmins"          "Toronto"          "Waterloo"
## [33] "Whitby"           "Windsor"
```

For these four variables, the age groups cluster the patient in 10-year intervals, which are the 20s, 30s, 40s, etc. Patients who are less than 20 years-old, who is recorded as "< 20s", and patients with ages not known or not recorded, are listed as "Unknown".

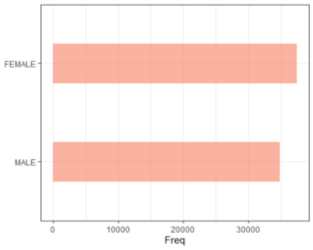
For Case acquisition information, the value of "CC" refers to the people who test positive by his or her closed contact, "No known Epi-link" represents for no epidemiological link, like community spread, and "OB" refers to the outbreak.

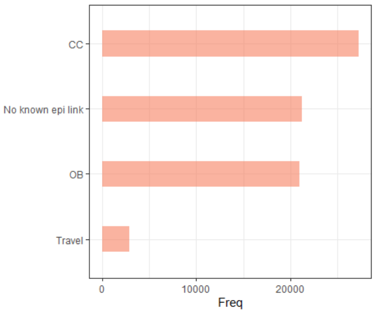
There are some inadequate factor levels in the following table, such as "missing information" and "Unspecified epi link" for case acquisition, the blank cell and "unknown" factor levels for the age group, "gender diverse" and "unspecified" for the gender group. To create a better dataset, we remove those observations that contain those unimportant variables.

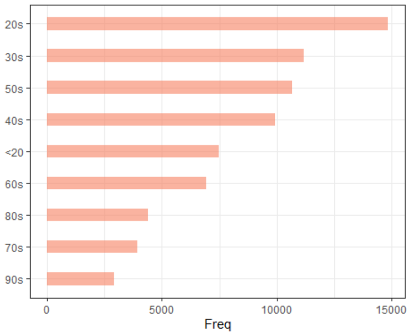
After finishing the data cleaning steps, we can start plotting simple statistical charts about the COVID-19 patients' background information to perform Exploratory data analysis (EDA); it is often used to summarize the variables' main characteristics by interpreting data information based on simple charts.

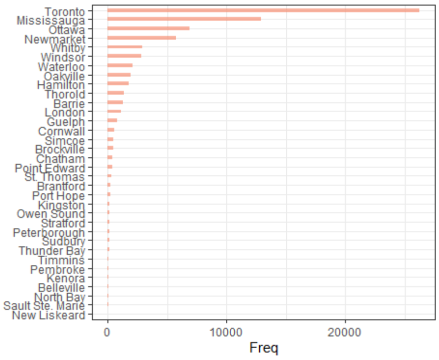
2.2 Exploratory Data Analysis

Because Exploratory data analysis (EDA) can investigate the variables' main characteristics by using statistical visual methods, the findings from EDA are the foundation of data analysis and data modelling all the time. After the process of data cleaning, we create some charts by using ggplot() function to investigate the variables' essential features. The following four charts are the frequency bar plots about different variables: age group, gender, spread method, and city location. From those plots, we can find the patients aged between the 20s and 30s occupies the greatest COVID-19 confirmed cases, and the confirmed proportion of senior people is small. Moreover, people who test positive are mainly from his/her closed contact; other significant spread methods are outbreak and new epidemiological link. The proportion of positive cases grouped by gender is considered almost identical; this implies that the gender variable might not significantly impact COVID-19 cases. For the variable of the located city, the top three cities which contain a large number of confirmed cases are Toronto, Mississauga, and Ottawa, all three of them are the cities which have the large population capicities.









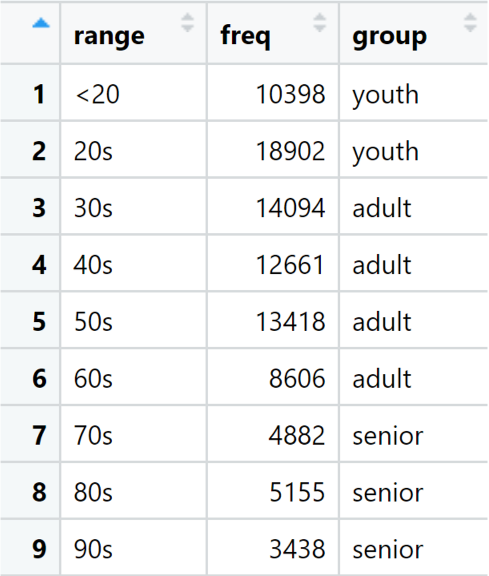
2.3 Create the Disjunctive Table with Dummy Variables

Binary variables, which are also called dummy variables, are commonly used in descriptive statistics. In a dummy column of the dataset, each cell represents the occurrence of a specific event. A zero value implies the event does not occur, and the amount of one show that it happens. Therefore, we can use the dummy_cols() function to shift the categorical factor levels to several dummy variables so that each dummy variable indicates the absence or presence of a specific factor level.

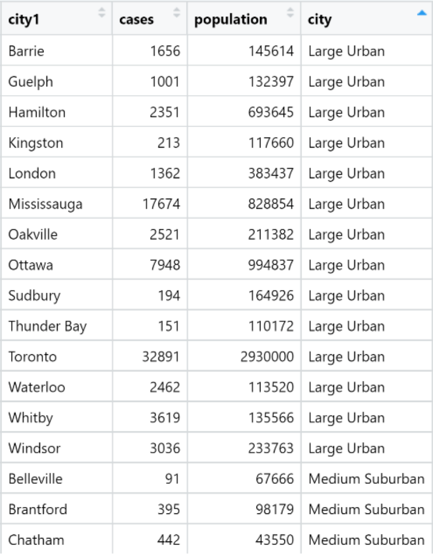
The new-formed dataset, based on the dummy variables, is called the disjunctive table. It displays the multivariate indicator matrix, where each row represents an individual patient observation, and each column is a dummy variable which stands for a specific factor level from a categorical variable; the value in each cell only shows 0 or 1, which refers to that whether a patient possesses a specific recorded factor level.

By reviewing the above factor levels from the new-formed dataset, we find that the variables of age group and city have 10-factor levels and 34-factor levels separately. Dozens of factor levels in a variable might result in hundreds of dummy variables and their corresponding variable combinations. So, before creating the disjunctive table, we need to group those two variables further to minimize their factor levels.

The best method to classify categorical data is grouped by its characteristics, such as the city size based on the population capacity and age groups based on the proportion of different age groups who test positive. Hence, we grouped people under 30 as a youth-aged group; people aged between 30 and 60 are adult, and people whose age is greater than 70 are classified as a senior. According to the list of the city population in Ontario which we found in Wikipedia (2016), we classify the city with a population size greater than 100,000 as large urban, city with a population between 30,000 and 100,000 are classified as medium suburban, and city that population less than 30,000 are small rural.



	range	freq	group
1	<20	10398	youth
2	20s	18902	youth
3	30s	14094	adult
4	40s	12661	adult
5	50s	13418	adult
6	60s	8606	adult
7	70s	4882	senior
8	80s	5155	senior
9	90s	3438	senior

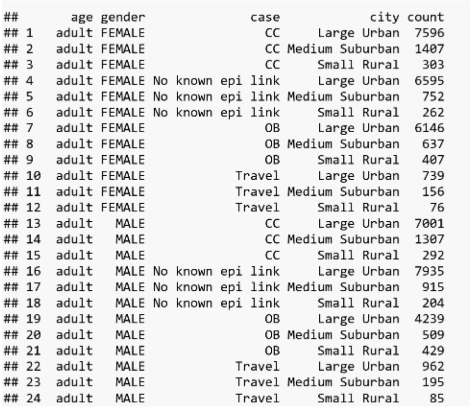


city1	cases	population	city
Barrie	1656	145614	Large Urban
Guelph	1001	132397	Large Urban
Hamilton	2351	693645	Large Urban
Kingston	213	117660	Large Urban
London	1362	383437	Large Urban
Mississauga	17674	828854	Large Urban
Oakville	2521	211382	Large Urban
Ottawa	7948	994837	Large Urban
Sudbury	194	164926	Large Urban
Thunder Bay	151	110172	Large Urban
Toronto	32891	2930000	Large Urban
Waterloo	2462	113520	Large Urban
Whitby	3619	135566	Large Urban
Windsor	3036	233763	Large Urban
Belleville	91	67666	Medium Suburban
Brantford	395	98179	Medium Suburban
Chatham	442	43550	Medium Suburban

The new variable table is shown below:

```
> levels(as.factor(canada$age))
[1] "adult"  "senior" "youth"
> levels(as.factor(canada$gender))
[1] "FEMALE" "MALE"
> levels(as.factor(canada$case))
[1] "CC"                "No known epi link" "OB"
[4] "Travel"
> levels(as.factor(canada$city))
[1] "Large Urban"     "Medium Suburban" "Small Rural"
```

After simplifying all factor levels based on four categorical variables, we can use the dummy_cols() function to create a disjunctive table that contains sixteen different permutations based on four factorial variables along with their frequency counts. The disjunctive dataset is shown below, and we will keep using it to perform data modeling for our further data analysis.



```
##         age gender              case            city count
## 1   adult FEMALE                CC     Large Urban  7596
## 2   adult FEMALE                CC Medium Suburban  1407
## 3   adult FEMALE                CC     Small Rural   303
## 4   adult FEMALE No known epi link     Large Urban  6595
## 5   adult FEMALE No known epi link Medium Suburban   752
## 6   adult FEMALE No known epi link     Small Rural   262
## 7   adult FEMALE                OB     Large Urban  6146
## 8   adult FEMALE                OB Medium Suburban   637
## 9   adult FEMALE                OB     Small Rural   407
## 10  adult FEMALE            Travel     Large Urban   739
## 11  adult FEMALE            Travel Medium Suburban   156
## 12  adult FEMALE            Travel     Small Rural    76
## 13  adult   MALE                CC     Large Urban  7001
## 14  adult   MALE                CC Medium Suburban  1307
## 15  adult   MALE                CC     Small Rural   292
## 16  adult   MALE No known epi link     Large Urban  7935
## 17  adult   MALE No known epi link Medium Suburban   915
## 18  adult   MALE No known epi link     Small Rural   204
## 19  adult   MALE                OB     Large Urban  4239
## 20  adult   MALE                OB Medium Suburban   509
## 21  adult   MALE                OB     Small Rural   429
## 22  adult   MALE            Travel     Large Urban   962
## 23  adult   MALE            Travel Medium Suburban   195
## 24  adult   MALE            Travel     Small Rural    85
```

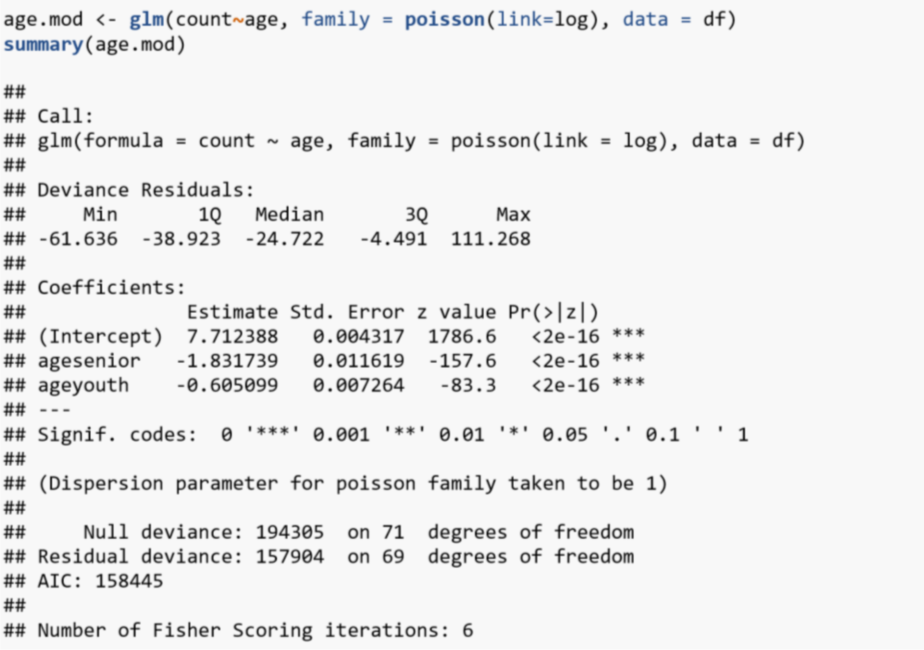
Chapter 3 – Interpreting Poisson Log-Linear Model with COVID-19 Data

3.1 Fit the Poisson Log-Linear Model with COVID-19 Dataset

The case study's main objective is to fit the Poisson log-linear model to the COVID-19 confirmed patient cases dataset to better estimate the confirmed case counts based on different physical variables' combinations and compare different physical variables' significances for diagnosing the coronavirus. We can develop the fitted model by applying several statistical techniques to use the final developed model to estimate the patient counts based on the developed model. The developing processes might include

- testing the factor variable's hypothesis to determine the essential factor level combinations.
- modifying the overdispersion when the lack-of-fit exists in the model.
- reweighting the observations to adjust the fitted model.

The first step of our data modeling would be generating the general Poisson Loglinear regression model by using the glm() function, and we analyze the model by applying the summary() function at the same time. The following R code tests whether the age groups' three-factor levels significantly affect the COVID-19 case counts using the Poisson Log-linear Model.



```
age.mod <- glm(count~age, family = poisson(link=log), data = df)
summary(age.mod)

##
## Call:
## glm(formula = count ~ age, family = poisson(link = log), data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -61.636  -38.923  -24.722   -4.491  111.268
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  7.712388   0.004317  1786.6   <2e-16 ***
## agesenior   -1.831739   0.011619  -157.6   <2e-16 ***
## ageyouth    -0.605099   0.007264   -83.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 194305  on 71  degrees of freedom
## Residual deviance: 157904  on 69  degrees of freedom
## AIC: 158445
##
## Number of Fisher Scoring iterations: 6
```

The estimated regression coefficients for three-factor levels of age group variable, along with their standard errors, z-values, and p-values, are all shown in the above summary table. We can test their significances by setting the hypothesis and looking at their p-values, where H0: At least one age group is not a significant factor; H1: All three age groups are significant factors.

Since all three p-values for the age groups are less than 0.05, we can reject the null hypothesis and conclude that all three-factor levels in the age groups have critical effects for the COVID-19 confirmed case counts. Moreover, we can substitute factors' coefficients into the link formula (The equation formula between the mean response and the explanatory variables' linear predictor) and discover different factor levels' association from the model expression:

$$\ln\,[\hat{\mu}(x_i)] = \sum_{i=0}^{2} \hat{\beta}_i x_i = 7.712 - 1.832x_{senior} - 0.6051x_{youth}$$

The above model contains two independent dummy factor levels: senior and youth groups; the adult group is included in the intercept term as a base category. The model shows three different scenarios based on different values from the dummy factor levels:

1. The mean confirmed COVID-19 case counts of a four-factor group which contains the adult-aged group is $\mu(\widehat{x_i}) = \exp(7.7124) \approx 2236$, where $x_{senior} = x_{youth} = 0$;
2. The mean confirmed COVID-19 case counts of a four-factor group which contains the senior-aged group is $\mu(\widehat{x_i}) = \exp(7.7124 - 1.8317) \approx 358$, where $x_{senior} = 1,\ x_{youth} = 0$;
3. The mean confirmed COVID-19 case counts of a four-factor group which contains the young-aged group is $\mu(\widehat{x_i}) = \exp(7.7124 - 0.6051) \approx 1221$, where $x_{senior} = 0,\ x_{youth} = 1$

The adult age-group factor does not appear in the coefficient summary table and the fitted model because it has been made to the base category. If R calculates the estimated coefficients for all three-factor levels of the age group variable while generating the model, we will create a 3*4 design matrix with an intercept term and three variable terms; this form causes the variable's resulting set linearly dependent and impossible to calculate the coefficients for all factor level. Moreover, investigating coefficients for all factor levels might result in the all-zero situation as dummy variables. The equation of mean counts value is equivalent to the intercept, which is unintelligible. So, it is necessary to select one factor as the base category.

Because the adult group is set as the baseline intercept, the intercept coefficient represents the adult group's superimposed value subset from the original dataset. Moreover, the corresponding coefficients for youth and senior groups refer to the differences between age groups. Thus, we can calculate the mean COVID-19 confirmed patient counts for each age-group and discover the factor level's difference by only looking at their coefficient values. The mean number of confirmed cases in the senior group is $e^{-1.8317} = 0.1601$, which is smaller than the adult group; The averaged confirmed case in the youth group is almost half of the adult group, the difference is $e^{-0.6051} = 0.5215$. Based on the estimated mean counts and the comparisons among different age groups, we can conclude that the adult groups have greater COVID-19 confirmed cases than other aged groups.

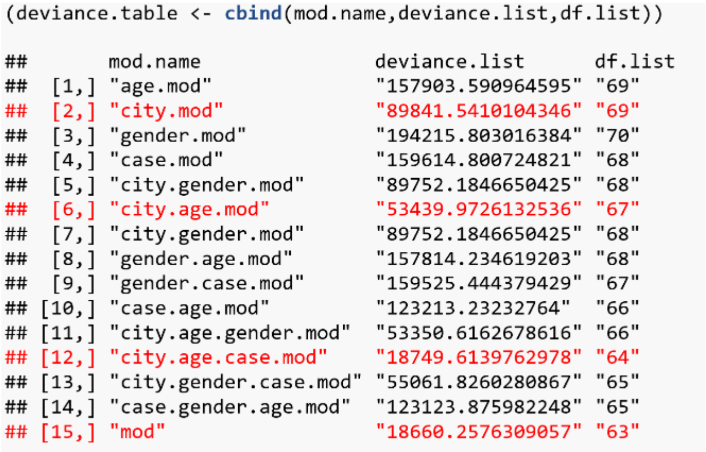
The above summary table also shows other information, such as residual deviance with respect to its degree of freedom, the number of Fisher iteration, and AIC value. AIC (Akaike Information Criteria) evaluates the model's fit by considering its maximum log-likelihood function and the number of parameters. Zajic (2019) mentioned that the AIC value is often used for examining the time-series models, and its formula is AIC=2k-2ln(L). According to its formula expression, it is easy to see that a large likelihood function score in MLE can receive a lower AIC score as it shows a good-of-fit model. Moreover, a penalty is applied for over-fitting with a large number of parameters. So, a smaller AIC value indicates that the model is closer to the ideal one. From the summary table for the age group model, an immense AIC value of 146,261 is shown; this might indicate that the model does not fit well. Because AIC values are only useful for comparisons among different models, we save this score and compare its value with other models later.

By reviewing the residual deviance value of 145,723 and its corresponding degree of freedom value of 69 from the summary table, we can use the goodness of fit test to conclude that the model does not fit data well. In other words, the three-factor levels of the age group in the fitted model are significant, but the disappearance of other significant variables in the model results in the lack-of-fits. So, we consider other reasons like the covariates' associations, extreme observations' effects, or model overdispersion to explain this situation.

There are a few ways to investigate a better-fitted model based on Poisson log-linear regression. Firstly, we can examine covariate effects by comparing their p-values to investigate the significant covariables. We could then validate different factor levels' interaction effects to remove those extreme observations that contain insignificant factor combinations. If the updated model still has an unfitted residual deviance value with respect to its degree of freedom, it will imply more variation in the response than the fitted model predicted. Then, there are two methods to adjust the model. The first technique is that we can apply other models like the quasi-Poisson regression model and negative binomial distribution by adding an estimated dispersion factor to inflate standard errors; Secondly, we can add the extra weights for all variables according to their residuals, which is based on robust regression. The next few subsections of this paper will improve the fitted model by following these processes.

3.3 Improve the Model –Covariate

The first step of developing the fitted model is to investigate the factor levels' significances based on coefficient comparisons. We generate the Poisson log-linear regression models with factorial variables and covariate combinations at first, then summarize the residual deviances of the model and correlated degree of freedom values together as a table.



```
(deviance.table <- cbind(mod.name,deviance.list,df.list))

##       mod.name                 deviance.list      df.list
##  [1,] "age.mod"                "157903.590964595" "69"
##  [2,] "city.mod"               "89841.5410104346" "69"
##  [3,] "gender.mod"             "194215.803016384" "70"
##  [4,] "case.mod"               "159614.800724821" "68"
##  [5,] "city.gender.mod"        "89752.1846650425" "68"
##  [6,] "city.age.mod"           "53439.9726132536" "67"
##  [7,] "city.gender.mod"        "89752.1846650425" "68"
##  [8,] "gender.age.mod"         "157814.234619203" "68"
##  [9,] "gender.case.mod"        "159525.444379429" "67"
## [10,] "case.age.mod"           "123213.23232764"  "66"
## [11,] "city.age.gender.mod"    "53350.6162678616" "66"
## [12,] "city.age.case.mod"      "18749.6139762978" "64"
## [13,] "city.gender.case.mod"   "55061.8260280867" "65"
## [14,] "case.gender.age.mod"    "123123.875982248" "65"
## [15,] "mod"                    "18660.2576309057" "63"
```

By reviewing the above table, we can quickly see different covariate combinations' residual deviances and corresponding Degree of freedoms. After comparing the models with the same number of covariates, we colored the models with the best fitness in red. It is easy to see that the model with covariates about the city, age, and case (spread method) have better fitness than others (Residual deviance is 18,749.61, Degree of freedom is 64). We also conclude that gender is not a significant factor in examining positive COVID-19 cases because the models containing gender variables generally receive larger parameter values.

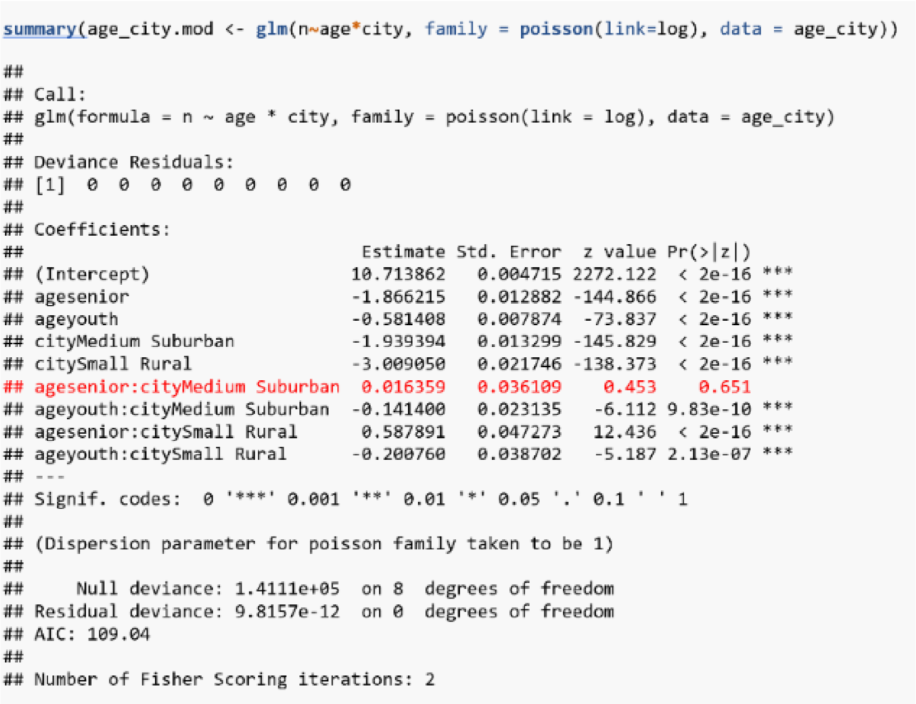
3.3 Improve the Model – Interactions

Although comparing the covariate model's fitness helps us investigate the crucial variables, the ratios between residual deviance and degree of freedom are still considerably large; this suggests that the model might contain other extreme observations. Therefore, we will investigate the models' variable interactions' effects by following the same statistical analysis technique, such as the significance test and goodness-of-fit test from subsection 3.1.

According to the conclusion in the 3.2 subsection, we understand that the gender variable is not essential for this model due to its large correlated residual deviance value. We remove this variable from the conjunctive dataset and only examine the models with the other three variables' combinations. The rest variables are

- the age group with three-factor levels (youth, adult, senior),
- city size with three-factor levels (small rural, medium suburban, large urban), and
- the spread method with four-factor levels (closed contact, outbreak, travel, no known epidemiological link).

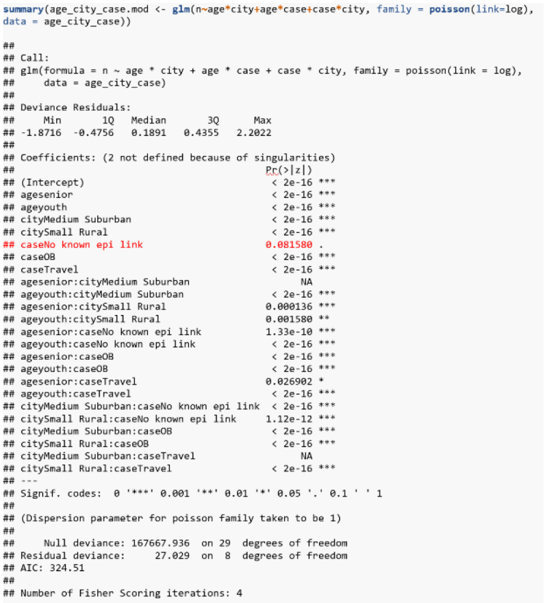
To interpret the variables' interaction effect using the Poisson log-linear model, we test the models with two-factor interactions at first. The following chart demonstrates the fitted model's summary results with two factorial variables of age group and PHC city, and their factor levels' interactions. The significance tests for all variables can be easily performed by checking their p-values. From the results, we can quickly find that the combined factor level of the senior-aged people living in a medium suburban city does not significantly affect getting COVID-19. The reason for this investigation is that its p-value is 0.651, which is reasonably more significant than the default confidence level of 0.05. So, we remove those patient observations whose independent variables include the age group of seniors and the city of medium suburban.



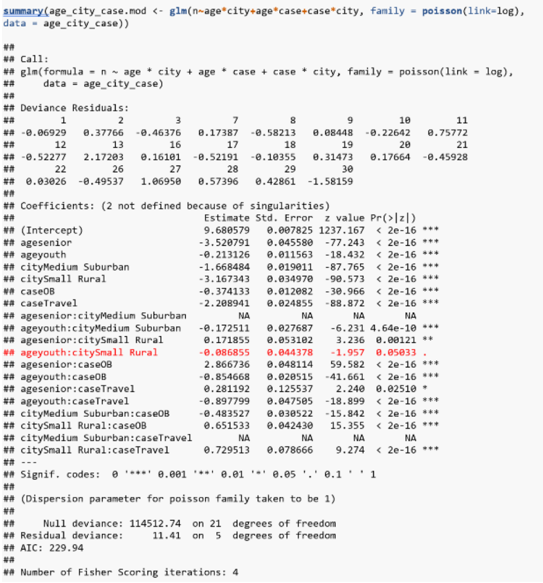
```
summary(age_city.mod <- glm(n~age*city, family = poisson(link=log), data = age_city))

##
## Call:
## glm(formula = n ~ age * city, family = poisson(link = log), data = age_city)
##
## Deviance Residuals:
## [1]  0  0  0  0  0  0  0  0  0
##
## Coefficients:
##                              Estimate Std. Error  z value Pr(>|z|)
## (Intercept)                 10.713862   0.004715 2272.122  < 2e-16 ***
## agesenior                   -1.866215   0.012882 -144.866  < 2e-16 ***
## ageyouth                    -0.581408   0.007874  -73.837  < 2e-16 ***
## cityMedium Suburban         -1.939394   0.013299 -145.829  < 2e-16 ***
## citySmall Rural             -3.009050   0.021746 -138.373  < 2e-16 ***
## agesenior:cityMedium Suburban  0.016359   0.036109    0.453    0.651
## ageyouth:cityMedium Suburban  -0.141400   0.023135   -6.112 9.83e-10 ***
## agesenior:citySmall Rural     0.587891   0.047273   12.436  < 2e-16 ***
## ageyouth:citySmall Rural     -0.200760   0.038702   -5.187 2.13e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1.4111e+05  on 8  degrees of freedom
## Residual deviance: 9.8157e-12  on 0  degrees of freedom
## AIC: 109.04
##
## Number of Fisher Scoring iterations: 2
```

After generating the models with all possibilities of variable interactions, we find three other uncorrelated variable combinations to the dependent variables. Two extreme values appear in the model with three two-factor interaction do not have significant effects, they are the spread method of no known epidemiological link and the factor combination of youth people who live in small rural cities. The reason for their insignificances is that their p-values are 0.08158 and 0.05033 separately, which are all greater than the default significant levels of 0.05. Moreover, the patients who live in the medium suburban cities test positive by traveling based on the covariate model with the city and spread method variables do not have a significant effect due to its large p-value of 0.121. Along with their p-values, three insignificant variables are all marked in red, shown in the following charts. Therefore, we can drop those patient observations which contain these two-factor combinations and one variable.

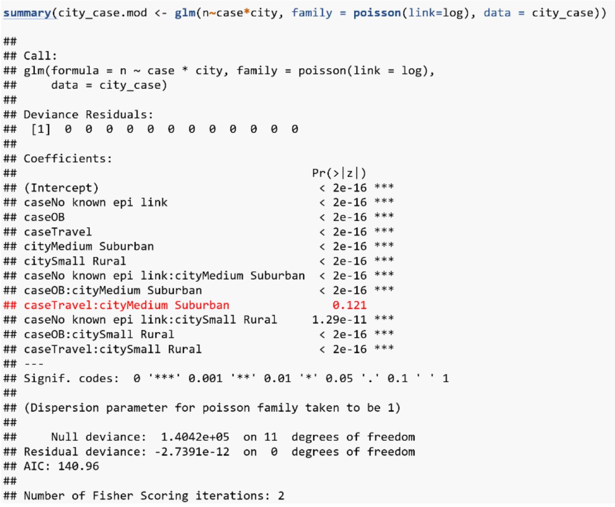
By comparing to the initial summary result about the model based on the single variable of age group, it is clear to see that our current models with the covariates and factor level combinations have smaller values in not only the residual deviances but also the AIC values, but also the ratio of deviance to the degree of freedom. Based on this observation, we may also conclude that the model with three two-factor combinations has better fitness for the data.



```
summary(age_city_case.mod <- glm(n~age*city+age*case+case*city, family = poisson(link=log),
data = age_city_case))

##
## Call:
## glm(formula = n ~ age * city + age * case + case * city, family = poisson(link = log),
##     data = age_city_case)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8716  -0.4756   0.1891   0.4355   2.2022
##
## Coefficients: (2 not defined because of singularities)
##                                           Pr(>|z|)
## (Intercept)                                < 2e-16 ***
## agesenior                                  < 2e-16 ***
## ageyouth                                   < 2e-16 ***
## cityMedium Suburban                        < 2e-16 ***
## citySmall Rural                            < 2e-16 ***
## caseNo known epi link                     0.081580 .
## caseOB                                     < 2e-16 ***
## caseTravel                                 < 2e-16 ***
## agesenior:cityMedium Suburban                   NA
## ageyouth:cityMedium Suburban               < 2e-16 ***
## agesenior:citySmall Rural                 0.000136 ***
## ageyouth:citySmall Rural                  0.001580 **
## agesenior:caseNo known epi link           1.33e-10 ***
## ageyouth:caseNo known epi link             < 2e-16 ***
## agesenior:caseOB                           < 2e-16 ***
## ageyouth:caseOB                            < 2e-16 ***
## agesenior:caseTravel                      0.026902 *
## ageyouth:caseTravel                        < 2e-16 ***
## cityMedium Suburban:caseNo known epi link  < 2e-16 ***
## citySmall Rural:caseNo known epi link     1.12e-12 ***
## cityMedium Suburban:caseOB                 < 2e-16 ***
## citySmall Rural:caseOB                     < 2e-16 ***
## cityMedium Suburban:caseTravel                  NA
## citySmall Rural:caseTravel                 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 167667.936  on 29  degrees of freedom
## Residual deviance:     27.029  on  8  degrees of freedom
## AIC: 324.51
##
## Number of Fisher Scoring iterations: 4
```



```
summary(age_city_case.mod <- glm(n~age*city+age*case+case*city, family = poisson(link=log),
data = age_city_case))

##
## Call:
## glm(formula = n ~ age * city + age * case + case * city, family = poisson(link = log),
##     data = age_city_case)
##
## Deviance Residuals:
##        1         2         3         7         8         9        10        11
## -0.06929   0.37766  -0.46376   0.17387  -0.58213   0.08448  -0.22642   0.75772
##       12        13        16        17        18        19        20        21
## -0.52277   2.17203   0.16101  -0.52191  -0.10355   0.31473   0.17664  -0.45928
##       22        26        27        28        29        30
##  0.03026  -0.49537   1.06950   0.57396   0.42861  -1.58159
##
## Coefficients: (2 not defined because of singularities)
##                                 Estimate Std. Error  z value Pr(>|z|)
## (Intercept)                     9.680579   0.007825 1237.167  < 2e-16 ***
## agesenior                      -3.520791   0.045580  -77.243  < 2e-16 ***
## ageyouth                       -0.213126   0.011563  -18.432  < 2e-16 ***
## cityMedium Suburban            -1.668484   0.019011  -87.765  < 2e-16 ***
## citySmall Rural                -3.167343   0.034970  -90.573  < 2e-16 ***
## caseOB                         -0.374133   0.012082  -30.966  < 2e-16 ***
## caseTravel                     -2.208941   0.024855  -88.872  < 2e-16 ***
## agesenior:cityMedium Suburban         NA         NA       NA       NA
## ageyouth:cityMedium Suburban   -0.172511   0.027687   -6.231 4.64e-10 ***
## agesenior:citySmall Rural       0.171855   0.053102    3.236  0.00121 **
## ageyouth:citySmall Rural       -0.086855   0.044378   -1.957  0.05033 .
## agesenior:caseOB                2.866736   0.048114   59.582  < 2e-16 ***
## ageyouth:caseOB                -0.854668   0.020515  -41.661  < 2e-16 ***
## agesenior:caseTravel            0.281192   0.125537    2.240  0.02510 *
## ageyouth:caseTravel            -0.897799   0.047505  -18.899  < 2e-16 ***
## cityMedium Suburban:caseOB     -0.483527   0.030522  -15.842  < 2e-16 ***
## citySmall Rural:caseOB          0.651533   0.042430   15.355  < 2e-16 ***
## cityMedium Suburban:caseTravel        NA         NA       NA       NA
## citySmall Rural:caseTravel      0.729513   0.078666    9.274  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 114512.74  on 21  degrees of freedom
## Residual deviance:     11.41  on  5  degrees of freedom
## AIC: 229.94
##
## Number of Fisher Scoring iterations: 4
```

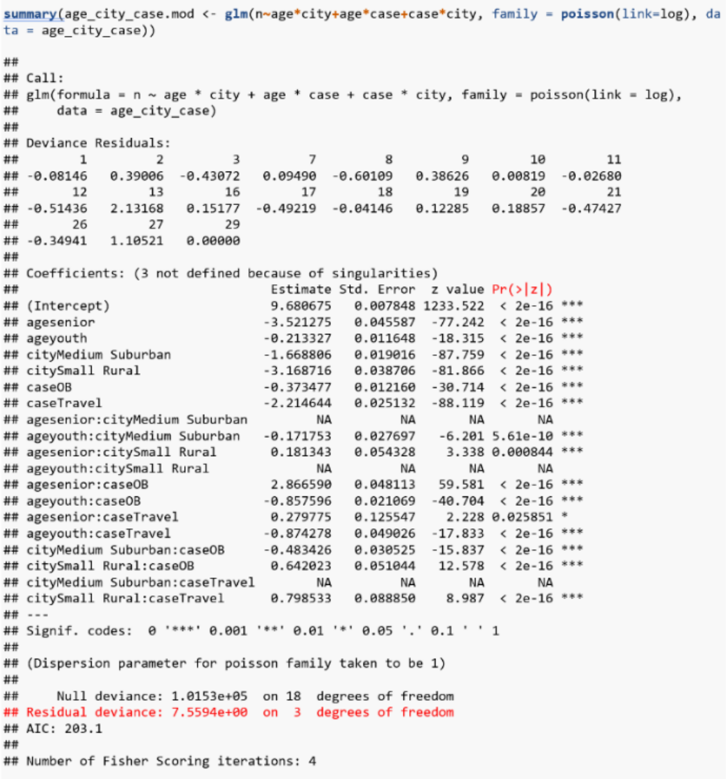


```
summary(city_case.mod <- glm(n~case*city, family = poisson(link=log), data = city_case))

##
## Call:
## glm(formula = n ~ case * city, family = poisson(link = log),
##     data = city_case)
##
## Deviance Residuals:
##  [1]  0  0  0  0  0  0  0  0  0  0  0  0
##
## Coefficients:
##                                           Pr(>|z|)
## (Intercept)                                < 2e-16 ***
## caseNo known epi link                      < 2e-16 ***
## caseOB                                     < 2e-16 ***
## caseTravel                                 < 2e-16 ***
## cityMedium Suburban                        < 2e-16 ***
## citySmall Rural                            < 2e-16 ***
## caseNo known epi link:cityMedium Suburban  < 2e-16 ***
## caseOB:cityMedium Suburban                 < 2e-16 ***
## caseTravel:cityMedium Suburban               0.121
## caseNo known epi link:citySmall Rural     1.29e-11 ***
## caseOB:citySmall Rural                     < 2e-16 ***
## caseTravel:citySmall Rural                 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance:  1.4042e+05  on 11  degrees of freedom
## Residual deviance: -2.7391e-12  on  0  degrees of freedom
## AIC: 140.96
##
## Number of Fisher Scoring iterations: 2
```

After dropping those extreme observations, we generate the Poisson log-linear model with three two-factor interactions and use the summary results to properly perform its significance test and goodness-of-fit test.

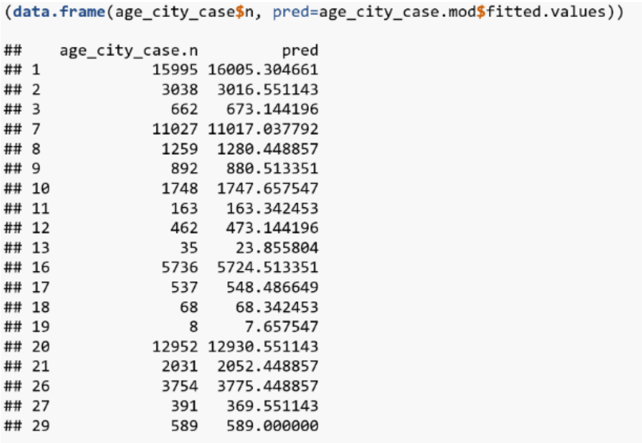
From the summary table below, we can see that all p-values for all 26 are smaller than 0.05; the variables shown in the table indicate that they all significantly affect COVID-19 confirmed patient case counts. Moreover, the ratio of residual deviance to its degree of freedom is 7.5594/3 = 2.5198, which is very closed to 1.

This measure illustrates that the Poisson log-linear model with three two-factor combinations reasonably fits the saturated model.



```
summary(age_city_case.mod <- glm(n~age*city+age*case+case*city, family = poisson(link=log), da
ta = age_city_case))

##
## Call:
## glm(formula = n ~ age * city + age * case + case * city, family = poisson(link = log),
##     data = age_city_case)
##
## Deviance Residuals:
##        1         2         3         7         8         9        10        11
## -0.08146   0.39006  -0.43072   0.09490  -0.60109   0.38626   0.00819  -0.02680
##       12        13        16        17        18        19        20        21
## -0.51436   2.13168   0.15177  -0.49219  -0.04146   0.12285   0.18857  -0.47427
##       26        27        29
## -0.34941   1.10521   0.00000
##
## Coefficients: (3 not defined because of singularities)
##                                Estimate Std. Error  z value Pr(>|z|)
## (Intercept)                    9.680675   0.007848 1233.522  < 2e-16 ***
## agesenior                     -3.521275   0.045587  -77.242  < 2e-16 ***
## ageyouth                      -0.213327   0.011648  -18.315  < 2e-16 ***
## cityMedium Suburban           -1.668806   0.019016  -87.759  < 2e-16 ***
## citySmall Rural               -3.168716   0.038706  -81.866  < 2e-16 ***
## caseOB                        -0.373477   0.012160  -30.714  < 2e-16 ***
## caseTravel                    -2.214644   0.025132  -88.119  < 2e-16 ***
## agesenior:cityMedium Suburban        NA         NA       NA       NA
## ageyouth:cityMedium Suburban  -0.171753   0.027697   -6.201 5.61e-10 ***
## agesenior:citySmall Rural      0.181343   0.054328    3.338 0.000844 ***
## ageyouth:citySmall Rural             NA         NA       NA       NA
## agesenior:caseOB               2.866590   0.048113   59.581  < 2e-16 ***
## ageyouth:caseOB               -0.857596   0.021069  -40.704  < 2e-16 ***
## agesenior:caseTravel           0.279775   0.125547    2.228 0.025851 *
## ageyouth:caseTravel           -0.874278   0.049026  -17.833  < 2e-16 ***
## cityMedium Suburban:caseOB    -0.483426   0.030525  -15.837  < 2e-16 ***
## citySmall Rural:caseOB         0.642023   0.051044   12.578  < 2e-16 ***
## cityMedium Suburban:caseTravel       NA         NA       NA       NA
## citySmall Rural:caseTravel     0.798533   0.088850    8.987  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 1.0153e+05  on 18  degrees of freedom
## Residual deviance: 7.5594e+00  on  3  degrees of freedom
## AIC: 203.1
##
## Number of Fisher Scoring iterations: 4
```

Except for examining the AIC values and the residual deviance ratio, we also can compare the predicted value from the fitted model to the actual counts to test the model fitness. We create a simple comparison table to see the difference between the actual count data from the original dataset and the predicted mean count values from the Poisson log-linear model. By reviewing the tiny differences between the two columns, we can strongly agree that our model fits the original dataset well.



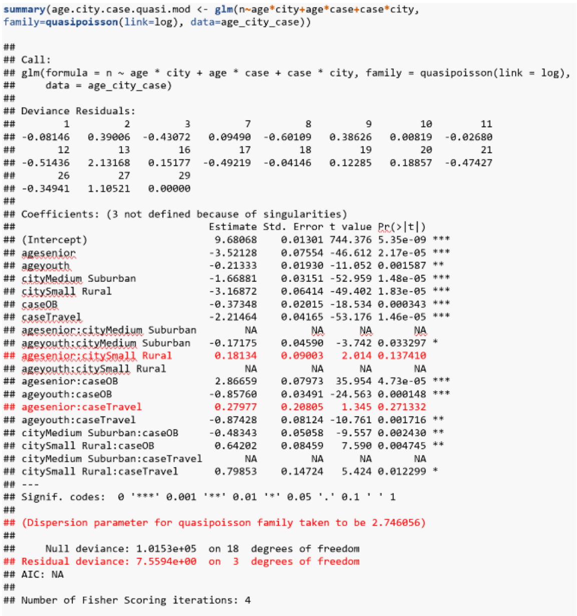
```
(data.frame(age_city_case$n, pred=age_city_case.mod$fitted.values))

##    age_city_case.n         pred
## 1            15995 16005.304661
## 2             3038  3016.551143
## 3              662   673.144196
## 7            11027 11017.037792
## 8             1259  1280.448857
## 9              892   880.513351
## 10            1748  1747.657547
## 11             163   163.342453
## 12             462   473.144196
## 13              35    23.855804
## 16            5736  5724.513351
## 17             537   548.486649
## 18              68    68.342453
## 19               8     7.657547
## 20           12952 12930.551143
## 21            2031  2052.448857
## 26            3754  3775.448857
## 27             391   369.551143
## 29             589   589.000000
```

3.4 Improve the Model - Overdispersion

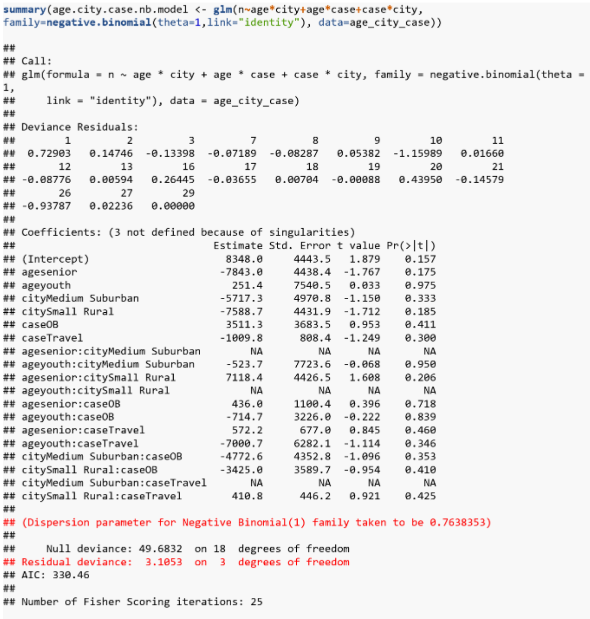
Although the fitted Poisson log-linear model's predicted values provide an inspired result by comparing with the actual data, the ratio between residual deviance and degree of freedom of 2.5198 is still greater than one. According to Legler and Roback's (2019) research, they suggest that the ratio value more extensive than one indicates that the model may contain overdispersion, so there is more variation in the response than the model expects.

Under the Poisson model, we assume that the mean value and variance of the response are the same in various groups; however, the overdispersion indicates unexpected standard errors happened in the fitted model. Our next step is to adjust the overdispersion by applying other models; they are quasi-Poisson likelihood regression and negative binomial regression model. The quasi-likelihood model is based on the Poisson regression model; its development adds the dispersion parameter Φ to inflate variables' standard errors and further redescribe the mean-variance relationship. The negative binomial regression model can also be applied to better fit the original dataset into its distribution expression instead of the Poisson regression model. After interpreting these two models with three two-factor interactions under these two developed models, we can compare their results to the Poisson log-linear model based on the factorial variables' p-values and residual deviances along with their degree of freedom. Then, the most probable model is chosen as the result based on the compared consequence.

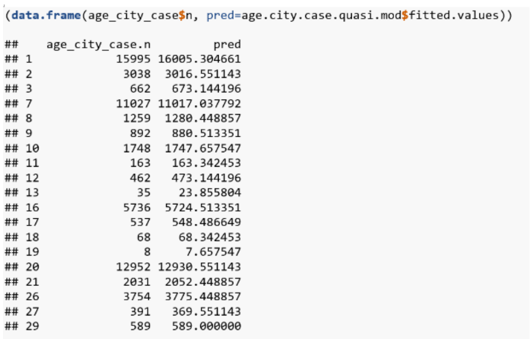
The following summary tables illustrate two models' summaries and their predicted mean responses' values compared to the actual COVID-19 patient counts. The table in the left column represents the summary of the quasi-likelihood Poisson model, and the one in the right column refers to the result of the negative binomial regression model. The information marked in red indicates the crucial parameters, such as the values of dispersion parameters, residual deviance with respect to its degree of freedoms, and the insignificant variables with their coefficient values.



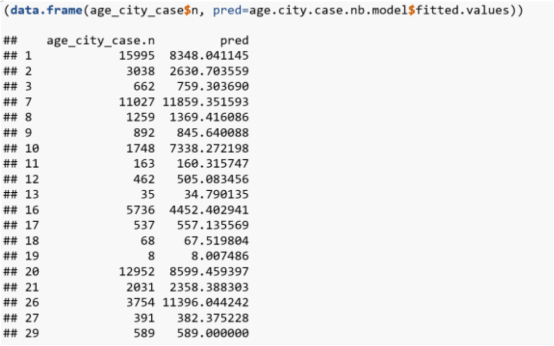
```
summary(age.city.case.quasi.mod <- glm(n~age*city+age*case+case*city,
family=quasipoisson(link=log), data=age_city_case))

##
## Call:
## glm(formula = n ~ age * city + age * case + case * city, family = quasipoisson(link = log),
##     data = age_city_case)
##
## Deviance Residuals:
##        1         2         3         7         8         9        10        11
## -0.08146   0.39006  -0.43072   0.09490  -0.60109   0.38626   0.00819  -0.02680
##       12        13        16        17        18        19        20        21
## -0.51436   2.13168   0.15177  -0.49219  -0.04146   0.12285   0.18857  -0.47427
##       26        27        29
## -0.34941   1.10521   0.00000
##
## Coefficients: (3 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     9.68068    0.01301 744.376 5.35e-09 ***
## agesenior                      -3.52128    0.07554 -46.612 2.17e-05 ***
## ageyouth                       -0.21333    0.01930 -11.052 0.001587 **
## cityMedium Suburban            -1.66881    0.03151 -52.959 1.48e-05 ***
## citySmall Rural                -3.16872    0.06414 -49.402 1.83e-05 ***
## caseOB                         -0.37348    0.02015 -18.534 0.000343 ***
## caseTravel                     -2.21464    0.04165 -53.176 1.46e-05 ***
## agesenior:cityMedium Suburban        NA         NA      NA       NA
## ageyouth:cityMedium Suburban   -0.17175    0.04590  -3.742 0.033297 *
## agesenior:citySmall Rural       0.18134    0.09003   2.014 0.137410
## ageyouth:citySmall Rural             NA         NA      NA       NA
## agesenior:caseOB                2.86659    0.07973  35.954 4.73e-05 ***
## ageyouth:caseOB                -0.85760    0.03491 -24.563 0.000148 ***
## agesenior:caseTravel            0.27977    0.20805   1.345 0.271332
## ageyouth:caseTravel            -0.87428    0.08124 -10.761 0.001716 **
## cityMedium Suburban:caseOB     -0.48343    0.05058  -9.557 0.002430 **
## citySmall Rural:caseOB          0.64202    0.08459   7.590 0.004745 **
## cityMedium Suburban:caseTravel       NA         NA      NA       NA
## citySmall Rural:caseTravel      0.79853    0.14724   5.424 0.012299 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2.746056)
##
##     Null deviance: 1.0153e+05  on 18  degrees of freedom
## Residual deviance: 7.5594e+00  on  3  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```



```
summary(age.city.case.nb.model <- glm(n~age*city+age*case+case*city,
family=negative.binomial(theta=1,link="identity"), data=age_city_case))

##
## Call:
## glm(formula = n ~ age * city + age * case + case * city, family = negative.binomial(theta =
1,
##     link = "identity"), data = age_city_case)
##
## Deviance Residuals:
##        1         2         3         7         8         9        10        11
##  0.72903   0.14746  -0.13398  -0.07189  -0.08287   0.05382  -1.15989   0.01660
##       12        13        16        17        18        19        20        21
## -0.08776   0.00594   0.26445  -0.03655   0.00704  -0.00088   0.43950  -0.14579
##       26        27        29
## -0.93787   0.02236   0.00000
##
## Coefficients: (3 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      8348.0     4443.5   1.879    0.157
## agesenior                       -7843.0     4438.4  -1.767    0.175
## ageyouth                          251.4     7540.5   0.033    0.975
## cityMedium Suburban             -5717.3     4970.8  -1.150    0.333
## citySmall Rural                 -7588.7     4431.9  -1.712    0.185
## caseOB                           3511.3     3683.5   0.953    0.411
## caseTravel                      -1009.8      808.4  -1.249    0.300
## agesenior:cityMedium Suburban        NA         NA      NA       NA
## ageyouth:cityMedium Suburban     -523.7     7723.6  -0.068    0.950
## agesenior:citySmall Rural        7118.4     4426.5   1.608    0.206
## ageyouth:citySmall Rural             NA         NA      NA       NA
## agesenior:caseOB                  436.0     1100.4   0.396    0.718
## ageyouth:caseOB                  -714.7     3226.0  -0.222    0.839
## agesenior:caseTravel              572.2      677.0   0.845    0.460
## ageyouth:caseTravel             -7000.7     6282.1  -1.114    0.346
## cityMedium Suburban:caseOB      -4772.6     4352.8  -1.096    0.353
## citySmall Rural:caseOB          -3425.0     3589.7  -0.954    0.410
## cityMedium Suburban:caseTravel       NA         NA      NA       NA
## citySmall Rural:caseTravel        410.8      446.2   0.921    0.425
##
## (Dispersion parameter for Negative Binomial(1) family taken to be 0.7638353)
##
##     Null deviance: 49.6832  on 18  degrees of freedom
## Residual deviance:  3.1053  on  3  degrees of freedom
## AIC: 330.46
##
## Number of Fisher Scoring iterations: 25
```



```
(data.frame(age_city_case$n, pred=age.city.case.quasi.mod$fitted.values))

##    age_city_case.n         pred
## 1            15995 16005.304661
## 2             3038  3016.551143
## 3              662   673.144196
## 7            11027 11017.037792
## 8             1259  1280.448857
## 9              892   880.513351
## 10            1748  1747.657547
## 11             163   163.342453
## 12             462   473.144196
## 13              35    23.855804
## 16            5736  5724.513351
## 17             537   548.486649
## 18              68    68.342453
## 19               8     7.657547
## 20           12952 12930.551143
## 21            2031  2052.448857
## 26            3754  3775.448857
## 27             391   369.551143
## 29             589   589.000000
```

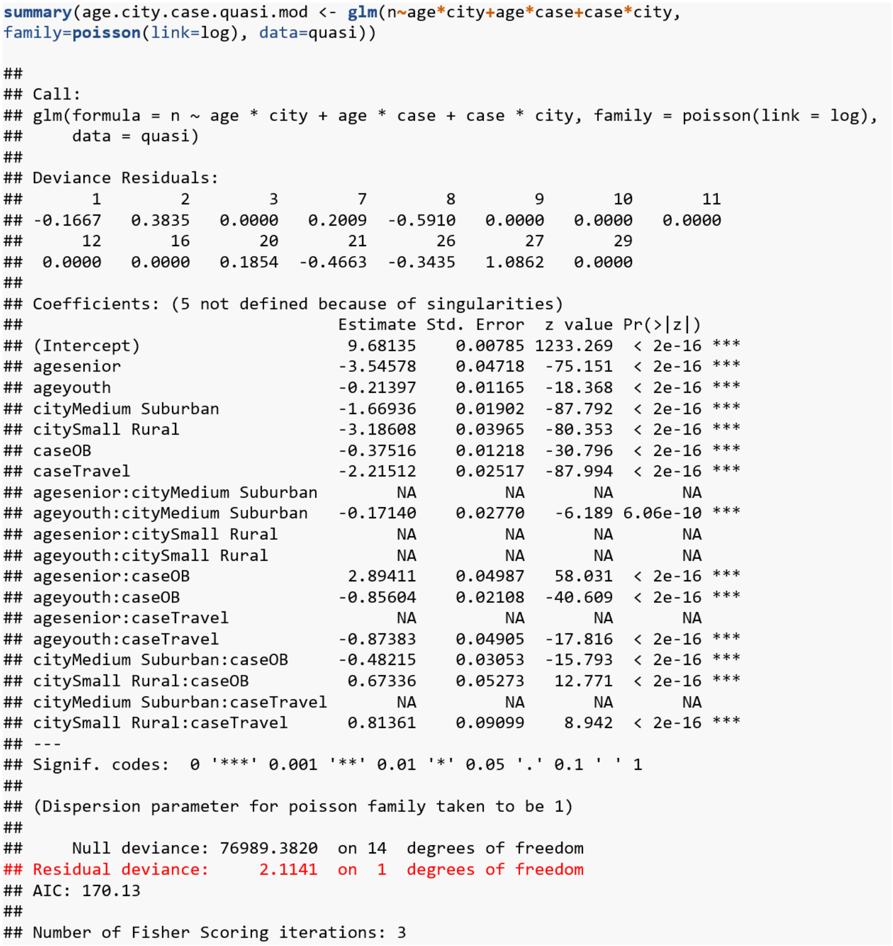


```
(data.frame(age_city_case$n, pred=age.city.case.nb.model$fitted.values))

##    age_city_case.n         pred
## 1            15995  8348.041145
## 2             3038  2630.703559
## 3              662   759.303690
## 7            11027 11859.351593
## 8             1259  1369.416086
## 9              892   845.640088
## 10            1748  7338.272198
## 11             163   160.315747
## 12             462   505.083456
## 13              35    34.790135
## 16            5736  4452.402941
## 17             537   557.135569
## 18              68    67.519804
## 19               8     8.007486
## 20           12952  8599.459397
## 21            2031  2358.388303
## 26            3754 11396.044242
## 27             391   382.375228
## 29             589   589.000000
```

Although the negative binomial regression model in the right column provides the acceptable residual deviance of 3.1053 to its degree of freedom of 3; however, the corresponding variable coefficients are all reasonably larger than the default 0.05 significance level. We can conclude that this model exhibited a remarkable lack-of-fit based on three investigations that appeared in the summary table: the large p-values for all variables, double larger AIC value than the Poisson regression model unfitted predicted response value in the comparison table. The dispersion parameter value of 0.7638353 can sufficiently explain why the negative binomial model does not fit well. According to subsection 1.7 of this paper about negative binomial distribution, we understand that the negative binomial distribution in the exponential dispersion family is looked like $f(y_i;\theta_i,\Phi) = \exp\left[\frac{y_i \ln\left(\frac{\theta_i}{1-\theta_i}\right) - \ln(1-\theta_i)}{{}^{1}/{}_{n_i}}\right] + \ln\binom{n_i}{n_i y_i}$, where $\Phi = {}^{1}/{}_{n_i} = 0.7638$ is the dispersion parameter value, so the dispersion parameter, which is less than one, shows the greater value of the log-likelihood function; this results in the more considerable AIC value. Moreover, the negative binomial regression model's variance formula can be expressed as $Var(Y) = \mu + \frac{\mu^2}{k} = \mu + \Phi\mu^2$, where k is the dispersion parameter in this formula. Its additional term of the squared mean value times dispersion parameter of 0.7618 results in the significant difference between the predicted mean response value and the corresponding actual count value, also its larger standard error and p-value for each variable.

By reviewing the Quasi-likelihood Poisson model in the left column, we can see that different results. Several variable effects shown in red color represent their insignificances at the 0.05 level. They are the factor combination of senior-aged people who lives in small rural cities (Standard Error = 0.09003, p-value = 0.137410) and the senior-aged people who tests positive by travelling (Standard Error = 0.20805, p-value = 0.271332). The p-values for all factorial variables and their combinations increase, compared to the Poisson regression values. We can also explain this observation by using the dispersion parameter value of 2.7461 in the summary table; it indicates that the dependent variables' variance is 2.7461 times larger than its mean values. By following the quasi-likelihood regression model's mean-variance relationship formula $Var(\mu_i) = 2.7461 \times \widehat{\mu_i} \approx 2.7461m_i$, we understand that the standard error for each variable increases by $\sqrt{2.7416} \approx 1.6558$ times compared to the one in the Poisson regression model. Thus, the greater standard errors result in the larger p-values for all variables. Moreover, the residual deviance stays remain because the dispersion parameter term is removed along with the process of differentiation about the maximum log-likelihood function. This investigation about the same ratio of residual deviance to the degree of freedom also can explain the same predicted values between the Quasi-likelihood Poisson model and Poisson log-linear model.

By comparing several developed models' parameters, we can easily conclude that the Quasi-Poisson regression model better fits our data. We remove those extreme observations to develop the model, which contains two insignificant factor combinations we mentioned in the last paragraph. After taking extreme observations in the Quasi-likelihood regression model summary, we can investigate the smaller deviance value and more accurate predicted values in the Poisson log-linear regression model. The results are shown in the following summary tables.



```
summary(age.city.case.quasi.mod <- glm(n~age*city+age*case+case*city,
family=poisson(link=log), data=quasi))

##
## Call:
## glm(formula = n ~ age * city + age * case + case * city, family = poisson(link = log),
##     data = quasi)
##
## Deviance Residuals:
##       1        2        3        7        8        9       10       11
## -0.1667   0.3835   0.0000   0.2009  -0.5910   0.0000   0.0000   0.0000
##      12       16       20       21       26       27       29
##  0.0000   0.0000   0.1854  -0.4663  -0.3435   1.0862   0.0000
##
## Coefficients: (5 not defined because of singularities)
##                                  Estimate Std. Error  z value Pr(>|z|)
## (Intercept)                       9.68135    0.00785 1233.269  < 2e-16 ***
## agesenior                        -3.54578    0.04718  -75.151  < 2e-16 ***
## ageyouth                         -0.21397    0.01165  -18.368  < 2e-16 ***
## cityMedium Suburban              -1.66936    0.01902  -87.792  < 2e-16 ***
## citySmall Rural                  -3.18608    0.03965  -80.353  < 2e-16 ***
## caseOB                           -0.37516    0.01218  -30.796  < 2e-16 ***
## caseTravel                       -2.21512    0.02517  -87.994  < 2e-16 ***
## agesenior:cityMedium Suburban          NA         NA       NA       NA
## ageyouth:cityMedium Suburban     -0.17140    0.02770   -6.189 6.06e-10 ***
## agesenior:citySmall Rural              NA         NA       NA       NA
## ageyouth:citySmall Rural               NA         NA       NA       NA
## agesenior:caseOB                  2.89411    0.04987   58.031  < 2e-16 ***
## ageyouth:caseOB                  -0.85604    0.02108  -40.609  < 2e-16 ***
## agesenior:caseTravel                   NA         NA       NA       NA
## ageyouth:caseTravel              -0.87383    0.04905  -17.816  < 2e-16 ***
## cityMedium Suburban:caseOB       -0.48215    0.03053  -15.793  < 2e-16 ***
## citySmall Rural:caseOB            0.67336    0.05273   12.771  < 2e-16 ***
## cityMedium Suburban:caseTravel         NA         NA       NA       NA
## citySmall Rural:caseTravel        0.81361    0.09099    8.942  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 76989.3820  on 14  degrees of freedom
## Residual deviance:     2.1141  on  1  degrees of freedom
## AIC: 170.13
##
## Number of Fisher Scoring iterations: 3
```

```
(data.frame(age_city_case$n, pred=age.city.case.nb.model$fitted.values))

##    age_city_case.n         pred
## 1            15995  8348.041145
## 2             3038  2630.703559
## 3              662   759.303690
## 7            11027 11859.351593
## 8             1259  1369.416086
## 9              892   845.640088
## 10            1748  7338.272198
## 11             163   160.315747
## 12             462   505.083456
## 13              35    34.790135
## 16            5736  4452.402941
## 17             537   557.135569
## 18              68    67.519804
## 19               8     8.007486
## 20           12952  8599.459397
## 21            2031  2358.388303
## 26            3754 11396.044242
## 27             391   382.375228
## 29             589   589.000000
```

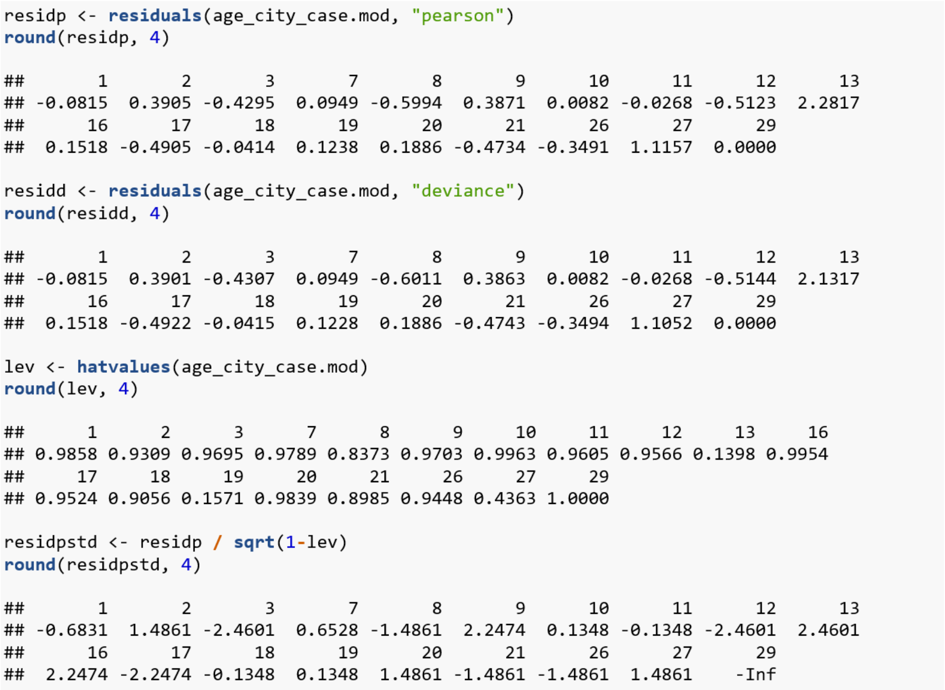
3.5 Improve the Model – Weighted GLMs

By interpreting the Quasi-likelihood Poisson model, we can sufficiently improve the Poisson regression model; however, the result of the residual deviance value is still not right. This identification suggests that we add weights for the different variables based on how well behaved these observations are so that we can create a better-fitted model. Then our next interest is to apply the reweighting methods in the model.

In statistics, there exist some residual terms in regression models. Residuals, which we used in the Chi-square fitness test before, is a measure that calculates the difference between the predicted mean count value and observed value. Outlier with respect to its residual is the observation that has a large residual value. Leverage is another measure used to compute the distance between the independent variable and the actual dependent variable, and influence based on the leverage is the observation that contains a considerable leverage value. Moreover, the cook's distance is the measure that combines the leverage and residuals. ("Robust regression | R data analysis examples", n.d.).

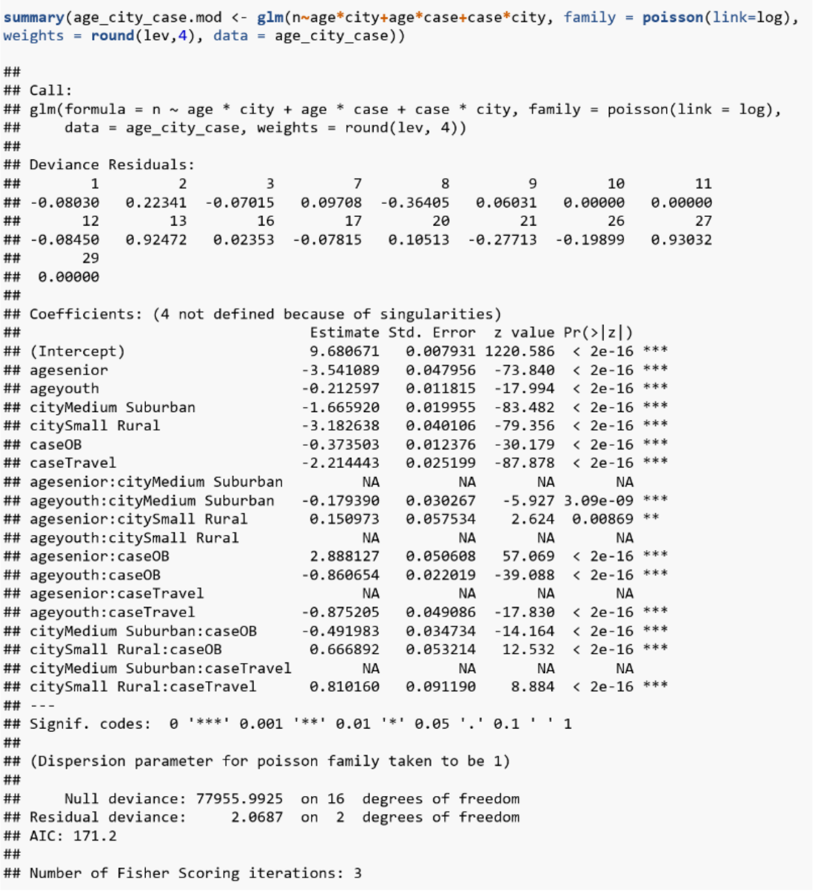
Since we have already determined the residual values, the next interest is to find the leverage values about the observation; then, we can use the findings to reweight the observations and develop the fitted model. At this time, we do not remove the unusual observations by investigating the large leverage values; as we already know, all observations contain the essential factor level combination for modeling. Moreover, sometimes removing a specific observation might cause a great difference in the final result. So, we will reweight all variables based on their performances to find a better-fitted model without losing any critical variables.

In R, there exist several functions for calculating different types of residuals. The following table shows different residual types and their corresponding functions. The function of residuals(model," pearson") is used to manipulate standardized Pearson residual values; the functions of residuals(model, "deviance") are used to present the residual deviance about the fitted model. Moreover, the hatvalues(model) can presents the observation leverages, and the last function of residpstd() can express the corresponding standardized residuals based on its hat value.

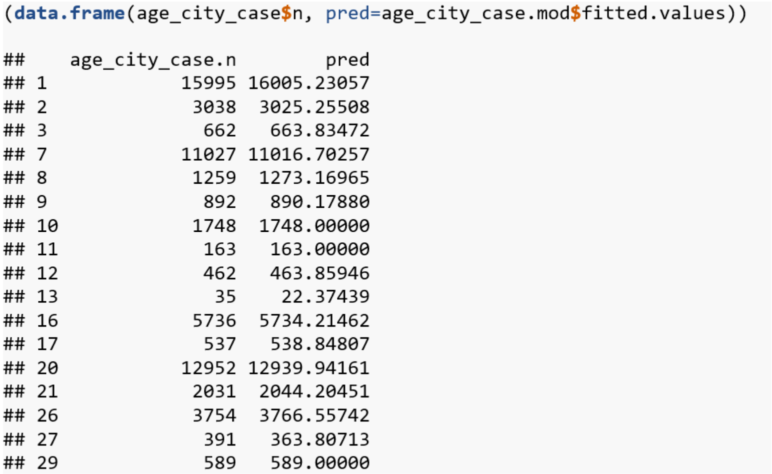


```
residp <- residuals(age_city_case.mod, "pearson")
round(residp, 4)

##       1       2       3       7       8       9      10      11      12      13 
## -0.0815  0.3905 -0.4295  0.0949 -0.5994  0.3871  0.0082 -0.0268 -0.5123  2.2817 
##      16      17      18      19      20      21      26      27      29 
##  0.1518 -0.4905 -0.0414  0.1238  0.1886 -0.4734 -0.3491  1.1157  0.0000 

residd <- residuals(age_city_case.mod, "deviance")
round(residd, 4)

##       1       2       3       7       8       9      10      11      12      13 
## -0.0815  0.3901 -0.4307  0.0949 -0.6011  0.3863  0.0082 -0.0268 -0.5144  2.1317 
##      16      17      18      19      20      21      26      27      29 
##  0.1518 -0.4922 -0.0415  0.1228  0.1886 -0.4743 -0.3494  1.1052  0.0000 

lev <- hatvalues(age_city_case.mod)
round(lev, 4)

##      1      2      3      7      8      9     10     11     12     13     16 
## 0.9858 0.9309 0.9695 0.9789 0.8373 0.9703 0.9963 0.9605 0.9566 0.1398 0.9954 
##     17     18     19     20     21     26     27     29 
## 0.9524 0.9056 0.1571 0.9839 0.8985 0.9448 0.4363 1.0000 

residpstd <- residp / sqrt(1-lev)
round(residpstd, 4)

##       1       2       3       7       8       9      10      11      12      13 
## -0.6831  1.4861 -2.4601  0.6528 -1.4861  2.2474  0.1348 -0.1348 -2.4601  2.4601 
##      16      17      18      19      20      21      26      27      29 
##  2.2474 -2.2474 -0.1348  0.1348  1.4861 -1.4861 -1.4861  1.4861    -Inf 
```

Since the weight term in the glm() function only allows positive values, we substitute the hat values, which refer to the observation's leverage, into the Poisson log-linear regression model. Then we summarize the findings and create the table of the predicted count value and observed response in the following block.



```
summary(age_city_case.mod <- glm(n~age*city+age*case+case*city, family = poisson(link=log),
weights = round(lev,4), data = age_city_case))

##
## Call:
## glm(formula = n ~ age * city + age * case + case * city, family = poisson(link = log),
##     data = age_city_case, weights = round(lev, 4))
##
## Deviance Residuals:
##        1         2         3         7         8         9        10        11
## -0.08030   0.22341  -0.07015   0.09708  -0.36405   0.06031   0.00000   0.00000
##       12        13        16        17        20        21        26        27
## -0.08450   0.92472   0.02353  -0.07815   0.10513  -0.27713  -0.19899   0.93032
##       29
##  0.00000
##
## Coefficients: (4 not defined because of singularities)
##                                Estimate Std. Error  z value Pr(>|z|)
## (Intercept)                    9.680671   0.007931 1220.586  < 2e-16 ***
## agesenior                     -3.541089   0.047956  -73.840  < 2e-16 ***
## ageyouth                      -0.212597   0.011815  -17.994  < 2e-16 ***
## cityMedium Suburban           -1.665920   0.019955  -83.482  < 2e-16 ***
## citySmall Rural               -3.182638   0.040106  -79.356  < 2e-16 ***
## caseOB                        -0.373503   0.012376  -30.179  < 2e-16 ***
## caseTravel                    -2.214443   0.025199  -87.878  < 2e-16 ***
## agesenior:cityMedium Suburban        NA         NA       NA       NA
## ageyouth:cityMedium Suburban  -0.179390   0.030267   -5.927 3.09e-09 ***
## agesenior:citySmall Rural      0.150973   0.057534    2.624  0.00869 **
## ageyouth:citySmall Rural             NA         NA       NA       NA
## agesenior:caseOB               2.888127   0.050608   57.069  < 2e-16 ***
## ageyouth:caseOB               -0.860654   0.022019  -39.088  < 2e-16 ***
## agesenior:caseTravel                 NA         NA       NA       NA
## ageyouth:caseTravel           -0.875205   0.049086  -17.830  < 2e-16 ***
## cityMedium Suburban:caseOB    -0.491983   0.034734  -14.164  < 2e-16 ***
## citySmall Rural:caseOB         0.666892   0.053214   12.532  < 2e-16 ***
## cityMedium Suburban:caseTravel       NA         NA       NA       NA
## citySmall Rural:caseTravel     0.810160   0.091190    8.884  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 77955.9925  on 16  degrees of freedom
## Residual deviance:     2.0687  on  2  degrees of freedom
## AIC: 171.2
##
## Number of Fisher Scoring iterations: 3
```



```
(data.frame(age_city_case$n, pred=age_city_case.mod$fitted.values))

##    age_city_case.n        pred
## 1            15995 16005.23057
## 2             3038  3025.25508
## 3              662   663.83472
## 7            11027 11016.70257
## 8             1259  1273.16965
## 9              892   890.17880
## 10            1748  1748.00000
## 11             163   163.00000
## 12             462   463.85946
## 13              35    22.37439
## 16            5736  5734.21462
## 17             537   538.84807
## 20           12952 12939.94161
## 21            2031  2044.20451
## 26            3754  3766.55742
## 27             391   363.80713
## 29             589   589.00000
```

By looking at the p-values in the updated model, we can see all variables and their corresponding covariates are statistically significant as the values are all reasonably smaller than the default significance levels. This implies that all factor combinations and the main factor are essential for the mean COVID-19 confirmed case counts.

The goodness-of-fit test about examining the residual values ratio to the degree of freedom reveals that the model is significantly fitted (Residual deviance: 2.0687 with 2 degrees of freedom). We also can double-check the model fitness by reviewing the AIC value, which is only 171.2. The above comparison table about the observed value and predicted value also shows the model's good fitness as the differences between the two values are small.

In conclusion, we can get our final model based on variable coefficients and their expression shown in the next block.

$$\begin{aligned}\ln(\hat{\mu}_i) = 9.6807 \\ &+ \left(-3.5411x_{as} - 0.2126x_{ay} - 1.6659x_{cm} - 3.1826x_{cs} \right. \\ &\left. - 0.3735x_{so} - 2.2144x_{st}\right) + (0.1794x_{ay-cm} + 0.1510x_{as-cs} \\ &+ 2.8881x_{ay-so} - 0.8752x_{ay-st} - 0.4920x_{cm-so} + 0.6669x_{cs-so} \\ &+ 0.8102x_{cs-st})\end{aligned}$$

In this model expression, 'as' represents the senior age group, 'ay' represents the youth age group, 'cm' represents the medium suburban city size, 'cs' represents the small rural city size, 'so' represents the outbreak spread method, 'st' refers to the traveling spread method. The rest terms in the second block are all factor-level combinations. The first intercept term represents the base category of the three-factor level combinations about people in the adult group who live in the large urban city and tests positive by the closed contact's spread method.

The above link function expression shows the relationship between the mean response value in the log form and the linear predictor among all factorial variables. We can easily understand that each coefficient value refers to the difference between the category base, and each variable is a dummy variable that only contains the values of zero or one. So, we can easily get the estimated confirmed case counts under a specific physical variable or their variable combinations by setting a variable as one and other variables as zero. An example would be the estimated mean confirmed COVID-19 case counts of a four-factor group which contains the traveling factor is $\hat{\mu}_i = \exp(9.6807 - 2.2144) = \exp(7.4663) = 1748.1266$ when $x_{st} = 1$ and other variables are zero, a four-factor group which contains the outbreak factor has the estimated mean confirmed case count of $\hat{\mu}_i = \exp(9.6807 - 0.3735) = \exp(9.3072) = 11{,}017.0571$, where $x_{so} = 1$ and other variables are zero. By reviewing their difference, we can see that the which is almost 7 times larger than the one with traveling factor. Thus, we can conclude that more people get COVID-19 by outbreak rather than traveling between different cities.

Also, the positive estimated coefficient in this model indicates that as the factor variable increases, the estimated mean count value of a four-factor group that contains this specific factor variable also increases; the negative coefficient represents the negative relationship between the estimated means count values for a four-factor group and this specific factor variable or a factor combination.

Therefore, we can quickly summarize some critical factors from the above model. A young patient who tests positive by the spread method of the outbreak has a reasonably significant effect on the mean COVID-19 confirmed case counts as its corresponding estimated coefficient is 2.8881, which shows that almost $\hat{\mu}_i = \exp(9.6807 + 2.8881) = \exp(12.5688) = 287448$ young patient who gets COVID-19 because of the outbreak.

Three-factor variables of the senior age group, small rural city size, and traveling spread method all negatively affect the mean confirmed case counts. This finding implies that people who age in the senior, people who lived in a small rural city, or traveling might not be considered a high-risk factor for getting COVID-19 cases. The reason that traveling is not a significant factor for coronavirus diagnosis is Canadian air transportation changes. According to Air Canada's report (2020), it shows that the number of traveling passengers by air in the third quarter of 2020 was dropped by almost 90. The reason is the traveling restrictions and coronavirus spread; however, the senior-aged group's findings have the lower confirmed case counts in this paper have the opposite conclusion to the well-known health organization. Centers for Disease Control and Prevention (2020) claimed that the risk for diagnosing coronavirus increases by age, which indicated that senior people are at the highest risk.

Chapter 4 – Summary

This project aims to learn the generalized linear model for analyzing categorical data and understanding the Poisson log-linear regression model's basic concept for interpreting counts data. After studying the model's fundamental knowledge and essential concepts, we fit the Poisson log-linear model into the publicly reported data about Ontario's COVID-19 confirmed patient cases with their several background patient information. According to this dataset, we address an essential question about patients with what kind of physical variables might significantly affect COVID-19 diagnose case counts. We then solve this question by modeling the Poisson log-linear regression model and estimating the COVID-19 confirmed case counts based on the estimated fitted model.

In the first section of this paper, we introduced the generalized linear model, which is commonly used in analyzing categorical data. We focused on learning the GLMs' three essential model components. Then, the paper found several methods for parameter estimation to demonstrate the fitted regression model. We studied the most probable parameter estimations using Maximum Likelihood Techniques of Newton-Raphson Interactive Methods and Fisher Scoring Methods. Then, we learned how to do mean response value examination by applying the goodness of fit test. After roughly understanding the basic concept of GLMs, the paper bought in the Poisson log-linear regression model for manipulating counts data along with their functions. Then the article talked about two crucial advanced models for adjusting the overdispersion situation; they are quasi-likelihood Poisson distribution and Negative Binomial regression distribution.

In the second chapter, we cleaned the dataset by converting important variables into factors and merging a few variables which contain many factor levels. Then, we used the Exploratory Data Analysis technique to analyze the variables' characteristics by reviewing those simple plots, converting those factorial variables into the dummy variables, and creating the contingency table. In the third section, we started to interpret the Poisson log-linear regression model with COVID-19 data. We performed the significance test by looking at covariates' p-values to examine their main factor and interaction effects in the beginning stage. We found that the residual variance values in the developed model, which contain the variable interactions, was still great. This observation implies an overdispersion in the Poisson regression model, so we tried different models, such as the Quasi-Poisson Log-Linear regression model and the Negative Binomial regression model applied the reweighted technique from the robust model to adjust this variation. In the end, we found the reweighted leverage of the Poisson log-linear regression model with three two-factor interactions about age groups, spread methods, and city size provided the most fitted predicted values. We summarized the result as a table and compared the expected value to the actual COVID-19 confirmed case counts in the end.

By reviewing the findings, we understand that the senior age groups occupied the least confirmed cases while identified as the most susceptible age group. We also concluded that as the city size rises, the confirmed cases also increase because of the city population. Traveling is also not a crucial factor for getting COVID-19; the reason for this observation might be the strict travel restriction in Canada. Moreover, we can see that the young patients who test positive by the outbreak have a large proportion. This might because many young people still hang out with friends and hold some parties to relax during the COVID-19 pandemic.

Reference

Centers for Disease Control and Prevention. (2020). *Older Adults at greater risk of requiring hospitalization or dying if diagnosed with COVID-19.* Retrieved from https://www.cdc.gov/coronavirus/2019-ncov/need-extra-precautions/older-adults.html.

Cision. (2020). *Air Canada Reports Third Quarter 2020 Results.* Retrieved from https://www.newswire.ca/news-releases/air-canada-reports-third-quarter-2020-results-823700314.html.

Legler J. & Roback P. (2019, January 20). *Broadening Your Statistical Horizons (BYSH): Generalized Linear Models and Multilevel Models.* Retrieved from https://bookdown.org/roback/bookdown-bysh/ch-poissonreg.html#sec-overdispPois.

List of population centres in Ontario. (2016). In *Wikipedia*. https://en.wikipedia.org/wiki/List_of_population_centres_in_Ontario

Ontario Government. (2020). *Confirmed Positive Cases of COVID-19 in Ontario.* [Data file]. Retrieved from https://data.ontario.ca/dataset/f4112442-bdc8-45d2-be3c-12efae72fb27/resource/455fd63b-603d-4608-8216-7d8647f43350/download/conposcovidloc.csv.

ROBUST REGRESSION | R DATA ANALYSIS EXAMPLES (n.d.). UCLA institute for Digital Research & Education Statistical Consulting. Retrieved from https://stats.idre.ucla.edu/r/dae/robust-regression/.

Scortchi – Reinstate Monica. (n.d.). In *StackExcahnge* [Answer page]. Retrieved November 24 2013, from https://stats.stackexchange.com/questions/77522/deviance-vs-pearson-goodness-of-fit.

Zajic. A. (2019, December 27). *Introduction to AIC – Akaike Information Criterion.* Towards Data Science. Retrieved from https://towardsdatascience.com/introduction-to-aic-akaike-information-criterion-9c9ba1c96ced.

Appendix – R Code

```
### Load libraries
library(fastDummies)
library(dplyr)
library(ggplot2)
library(forcats)
library(MASS)
library(AICcmodavg)
library(ggplot2)
library(knitr)

### Load Datasets
canada <- read.csv("C:/Users/sunni/Desktop/2020 FALL/MATH4905/canada.csv", stringsAsFactors = FALSE, header = TRUE)
summary(canada)
canada <- canada[,c(6:8,13)]
canada$Age_Group<-as.factor(canada$Age_Group)
canada$Client_Gender<-as.factor(canada$Client_Gender)
canada$Case_AcquisitionInfo<-as.factor(canada$Case_AcquisitionInfo)
canada$Reporting_PHU_City<-as.factor(canada$Reporting_PHU_City)
levels(canada$Age_Group)
levels(canada$Client_Gender)
levels(canada$Case_AcquisitionInfo)
levels(canada$Reporting_PHU_City)

### Data Cleaning
canada <- read.csv("C:/Users/sunni/Desktop/2020 FALL/MATH4905/canada.csv", stringsAsFactors = FALSE, header = TRUE)
canada <- canada[which(canada$Client_Gender=='FEMALE' |
canada$Client_Gender == "MALE"), ]
canada <- canada[-which(canada$Age_Group == "" | canada$Age_Group ==
"UNKNOWN"),]
canada <- canada[-which(canada$Case_AcquisitionInfo == "Missing Information"
| canada$Case_AcquisitionInfo == "Unspecified epi link"),]

### Explanatory Data Analysis
age <- as.data.frame(table(canada$Age_Group))
gender <- as.data.frame(table(canada$Client_Gender))
city <- as.data.frame(table(canada$Reporting_PHU_City))
case <- as.data.frame(table(canada$Case_AcquisitionInfo))

age %>%
  mutate(name = fct_reorder(Var1, Freq)) %>%
  ggplot( aes(x=name, y=Freq)) +
  geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +
  coord_flip() +
  xlab("") +
  theme_bw()

gender %>%
  mutate(name = fct_reorder(Var1, Freq)) %>%
  ggplot( aes(x=name, y=Freq)) +
  geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +
  coord_flip() +
  xlab("") +
  theme_bw()
```

```
city %>%
  mutate(name = fct_reorder(Var1, Freq)) %>%
  ggplot(aes(x=name, y=Freq)) +
  geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +
  theme(axis.text.x = element_text(color = "grey20", size = 10)) +
  coord_flip() +
  xlab("") +
  theme_bw()

case %>%
  mutate(name = fct_reorder(Var1, Freq)) %>%
  ggplot( aes(x=name, y=Freq)) +
  geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +
  coord_flip() +
  xlab("") +
  theme_bw()

### Data Cleaning
population <-
as.numeric(c("145614","67666","98179","21854","43550","45723","132397",
                     "693645", "10687", "117660", "383437", "828854", "9920",
"84224", "51553", "211382","994837","21341", "13882", "84230", "2037",
"16753", "73368", "13922", "38909", "31465", "164926", "18801", "110172",
"41788", "2930000", "113520", "135566", "233763"))
city <- cbind(city, population)
names(city) <- c("city", "cases", "population")
for (i in 1:nrow(city)){
  if (city[i,3] > 100000){
    city[i,4] = "Large Urban"
    } else if (city[i,3]<100000 & city[i,3]> 30000){
        city[i,4] = "Medium Suburban"
    } else {city[i,4] = "Small Rural"}}
names(city) <- c("city1", "cases", "population", "city")
canada$match <- match(canada$Reporting_PHU_City, city$city1)
canada$pop <- city[canada$match,3]

age$group <-c(rep("youth",2), rep("adult",5), rep("senior",2))
age$average <- c(10, 25, 35, 45, 55, 65, 75, 85, 95)
names(age) <- c("range", "freq", "group","average")

canada <- canada[,c(6:9,13,19)]
canada$match <- match(canada$Age_Group, age$range)
canada$Age_Group <- age[canada$match, 3]
canada$match <- match(canada$Reporting_PHU_City, city$city1)
canada$city <- city[canada$match,4]

canada <- canada[,-c(5:7)]
names(canada) <- c("age","gender","case","outcome","city")

df <- fastDummies::dummy_cols(canada[,-4]) %>% group_by_all() %>% count()
df <- as.data.frame(df[,c(1:4,17)])
names(df) <- c("age","gender","case","city","count")
df$age<-as.factor(df$age)
df$gender<-as.factor(df$gender)
df$case<-as.factor(df$case)
df$city<-as.factor(df$city)

### Poisson Log-linear Regression
```

```
age.mod <- glm(count~age, family = poisson(link=log), data = df)
summary(age.mod)
city.mod <- glm(count~city, family = poisson(link=log), data = df)
gender.mod <- glm(count~gender, family = poisson(link=log), data = df)
case.mod <- glm(count~case, family = poisson(link=log), data = df)
### Validate Dataset by Covariate
city.gender.mod <- glm(count~city+gender, family = poisson(link=log), data =
df)
city.age.mod <- glm(count~city+age, family = poisson(link=log), data = df)
city.gender.mod <- glm(count~city+gender, family = poisson(link=log), data =
df)
gender.age.mod <- glm(count~gender+age, family = poisson(link=log), data = df)
gender.case.mod <- glm(count~gender+case, family = poisson(link=log), data =
df)
case.age.mod <- glm(count~case+age, family = poisson(link=log), data = df)

city.age.gender.mod <- glm(count~city+age+gender, family = poisson(link=log),
data = df)
city.age.case.mod <- glm(count~city+age+case, family = poisson(link=log), data =
df)
city.gender.case.mod <- glm(count~city+gender+case, family =
poisson(link=log), data = df)
case.gender.age.mod <- glm(count~case+gender+age, family = poisson(link=log),
data = df)
mod <- glm(count~age+case+city+gender, family = poisson(link=log), data = df)

mod.list <- c(age.mod,city.mod,gender.mod,case.mod,

city.gender.mod,city.age.mod,city.gender.mod,gender.age.mod,gender.case.mod,c
ase.age.mod, city.age.gender.mod, city.age.case.mod, city.gender.case.mod,
case.gender.age.mod,mod)

deviance.list <-
c(age.mod$deviance,city.mod$deviance,gender.mod$deviance,case.mod$devianc
e, city.gender.mod$deviance, city.age.mod$deviance, city.gender.mod$deviance,
gender.age.mod$deviance, gender.case.mod$deviance, case.age.mod$deviance,
city.age.gender.mod$deviance, city.age.case.mod$deviance,
gender.case.mod$deviance, case.gender.age.mod$deviance, mod$deviance)
df.list <-
c(age.mod$df.residual,city.mod$df.residual,gender.mod$df.residual,case.mod$df.
residual, city.gender.mod$df.residual, city.age.mod$df.residual,
city.gender.mod$df.residual, gender.age.mod$df.residual,
gender.case.mod$df.residual, case.age.mod$df.residual,
city.age.gender.mod$df.residual, city.age.case.mod$df.residual,
city.gender.case.mod$df.residual, case.gender.age.mod$df.residual,
mod$df.residual)
mod.name <- c("age.mod","city.mod","gender.mod","case.mod",
                  "city.gender.mod","city.age.mod","city.gender.mod",
                  "gender.age.mod","gender.case.mod","case.age.mod",
                 "city.age.gender.mod", “city.age.case.mod","city.gender.case.mod",
                  "case.gender.age.mod","mod")
(deviance.table <- cbind(mod.name,deviance.list,df.list))
summary(city.age.case.mod)
summary(mod)

### Validate the model with Interactions
age_city <- canada[,c(1,5)] %>% group_by(age,city)
age_city <- age_city %>% summarise(n=n())
```

```
age_City <- as.data.frame(age_city)
summary(age_city.mod <- glm(n~age*city, family = poisson(link=log), data =
age_city))

age_case <- canada[,c(1,3)] %>% group_by(age,case)
age_case <- age_case %>% summarise(n=n())
age_Case <- as.data.frame(age_case)
summary(age_case.mod <- glm(n~age*case, family = poisson(link=log), data =
age_case))

city_case <- canada[,c(3,5)] %>% group_by(case,city)
city_case <- city_case %>% summarise(n=n())
city_case <- as.data.frame(city_case)
summary(city_case.mod <- glm(n~case*city, family = poisson(link=log), data =
city_case))

age_city_case <- canada[,c(1,3,5)] %>% group_by(age,case,city)
age_city_case <- age_city_case[!(age_city_case$age == "senior" &
age_city_case$city == "Medium Suburban"),]
age_city_case <- age_city_case[!(age_city_case$case == "Travel" &
age_city_case$city == "Medium Suburban"),]
age_city_case <- age_city_case %>% summarise(n=n())
age_city_case <- as.data.frame(age_city_case)

summary(age_city_case.mod <- glm(n~city+age*case, family = poisson(link=log),
data = age_city_case))
summary(age_city_case.mod <- glm(n~case+age*city, family = poisson(link=log),
data = age_city_case))
```

```
summary(age_city_case.mod <- glm(n~age*city+age*case+case*city, family =
poisson(link=log), data = age_city_case))

age_city_case <- age_city_case[!(age_city_case$case=="No known epi link"),]
summary(age_city_case.mod <- glm(n~age*city+age*case+case*city, family =
poisson(link=log), data = age_city_case))

age_city_case <- age_city_case[!(age_city_case$age == "youth" &
age_city_case$city == "Small Rural"),]
summary(age_city_case.mod <- glm(n~age*city+age*case+case*city, family =
poisson(link=log), data = age_city_case))
(data.frame(age_city_case$n, pred=age_city_case.mod$fitted.values))

### Validate the model with Overdispersion
summary(age.city.case.quasi.mod <- glm(n~age*city+age*case+case*city,
family=quasipoisson(link=log), data=age_city_case))
(data.frame(age_city_case$n, pred=age.city.case.quasi.mod$fitted.values))
summary(age.city.case.nb.model <- glm(n~age*city+age*case+case*city,
family=negative.binomial(theta=1,link="identity"), data=age_city_case))
(data.frame(age_city_case$n, pred=age.city.case.nb.model$fitted.values))

quasi <- age_city_case[!(age_city_case$age == "senior" & age_city_case$case ==
"Travel"),]
summary(age.city.case.quasi.mod <- glm(n~age*city+age*case+case*city,
family=quasipoisson(link=log), data=quasi))
quasi <- quasi[!(quasi$age == "senior" & quasi$city == "Small Rural" ),]
summary(age.city.case.quasi.mod <- glm(n~age*city+age*case+case*city,
family=quasipoisson(link=log), data=quasi))
```

```
summary(age.city.case.quasi.mod <- glm(n~age*city+age*case+case*city,
family=poisson(link=log), data=quasi))
(data.frame(age_city_case$n, pred=age.city.case.nb.model$fitted.values))

### Validate the Model with Weights
residp <- residuals(age_city_case.mod, "pearson")
round(residp, 4)
residd <- residuals(age_city_case.mod, "deviance")
round(residd, 4)
lev <- hatvalues(age_city_case.mod)
round(lev, 4)
residpstd <- residp / sqrt(1-lev)
round(residpstd, 4)

summary(age_city_case.mod <- glm(n~age*city+age*case+case*city, family =
poisson(link=log), weights = round(lev,4), data = age_city_case))
(data.frame(age_city_case$n, pred=age_city_case.mod$fitted.values))

age_city_case <- age_city_case[!(age_city_case$age == "senior" &
age_city_case$case == "Travel"),]
summary(age_city_case.mod <- glm(n~age*city+age*case+case*city, family =
poisson(link=log), data = age_city_case))
lev <- hatvalues(age_city_case.mod)
round(lev, 4)
summary(age_city_case.mod <- glm(n~age*city+age*case+case*city, family =
poisson(link=log), weights = round(lev,4), data = age_city_case))
(data.frame(age_city_case$n, pred=age_city_case.mod$fitted.values))
```