Assignment 3

for INF 412 (J. Wang)
March 26, 2023

Oliver Daniel

Contents

Poisson Regression

1. Dataset

The dataset, *Bike Share Toronto Ridership Data*[1], was accessed through the Open Data Toronto portal. In particular, the 2021 ridership data were selected due to an anomaly in the more recent 2022 set's file structure.

The content of the dataset is a zipped series of CSVs, one per calendar month of 2021, with each row representing a single completed journey of a Toronto Bike Share unit (henceforth "city bike"), from its release at a particular station, at a particular time, to its deposit at another – potentially different – station, at a particular time. Also included are UUIDs for the trip and city bike unit. In total, there are 3,575,182 unique trip events across the 12 months.

Columns

Of the 10 columns included with the data, only two were retained:

- **Start Time**: A timestamp representing the date and time at which the trip began (i.e., when the city bike was released from its holder).
- **User Type**: One of `Annual Member` or `Casual Member`, representing whether the purchaser of the trip was an annual pass bearer, or merely paid for the singular trip.

[1] Contains information licensed under the Open Government Licence – Toronto.

2. Variables

Independent Variables

Start Time was further divided into two columns, representing each timestamp's constituent date and time, respectively. To that end, for the purposes of this paper, two particular facets will be of focus as two of our three independent variables: the **month** in which the trip took place, and the **hour** of day during which the trip started. As expected, **month** and **hour** are ordered categorical variables, enumerating from 1 (`Jan`) to 12 (`Dec`), and from `0` (midnight) to `23` (11 PM), respectively.

The third variable shall be **User Type**, as previously described.

3. Dependent Variable & Regression Type

For the purposes of this assignment, a **Poisson regression** shall be used, to predict the total number of unique city bike trips that might be expected at the intersection of the three independent variables. For example, the regression model should be able to predict how many total trips will be started by pass-bearing members at noon in July. With this level of specificity, we are also able to get estimates over ranges by summing predictions over the ranges of elided variables; e.g., predicting all trips by pass-bearing in July, regardless of time of day, by summing all predicted trips from midnight to 11pm.

Poissonian models are useful for predicting the total **count of independent events** that might occur as of a particular point in time. We assume that all bike trips taken throughout Toronto are independent of one another, or at least insignificantly co-dependent. Here, our dependent variable is indeed count data – a total number of trips taken – so a Poisson regression is a useful model for our prediction.

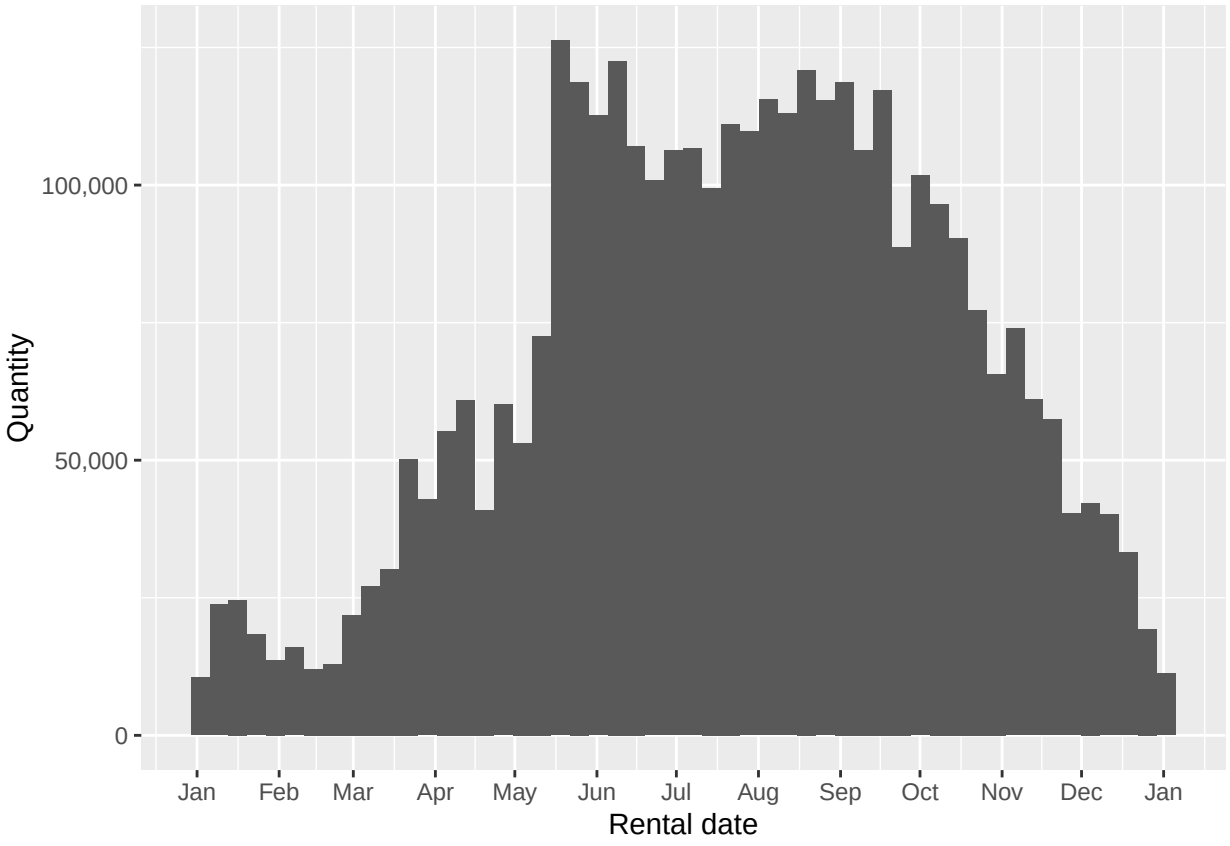


Figure 1: Distribution of 2021 City Bike rentals by week

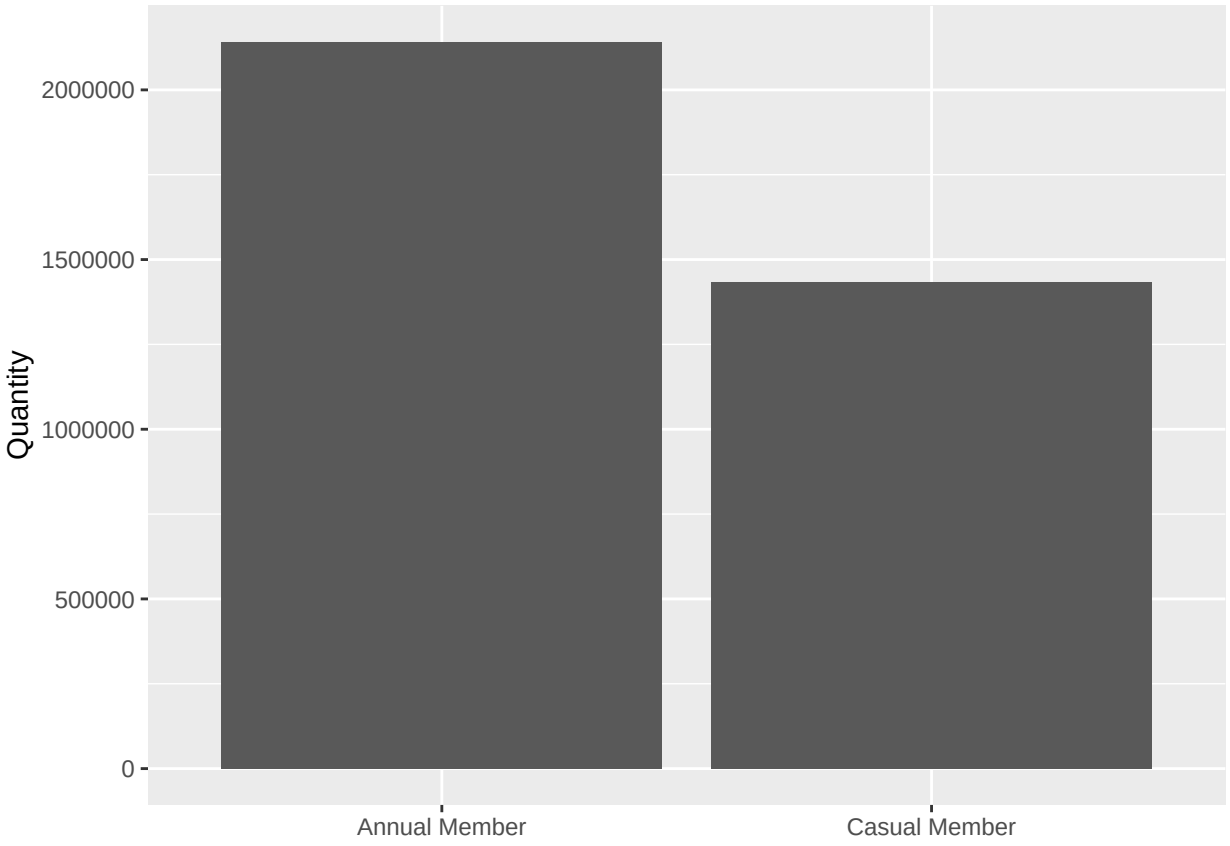
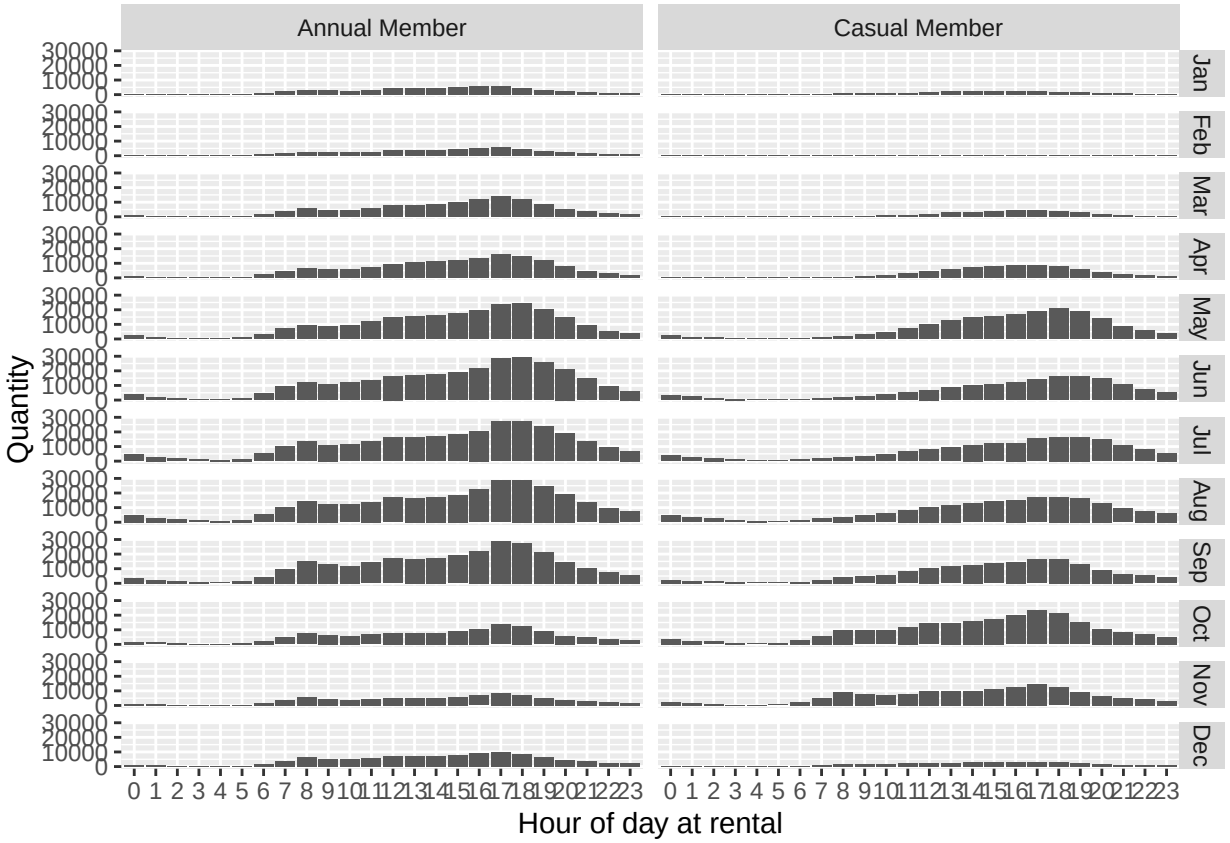


Figure 2: Distribution of user type in 2021 City Bike rentals

4. Variable Distributions

5. Correlational Diagrams

The correlation of our three independent variables to city bike count can be succinctly visualized as follows:



A number of interesting patterns can be seen here. With both types of members, there is a peak in monthly use in the warmer months (May through August), tapering off through the remainder of the year. Although annual members account for many more rentals in most of the colder months, casual members surprisingly dominate in October and November.

Table 1: Monthly comparison of casual vs. annual members in counts of bike rentals

Month	Casual rentals	Annual rentals	Difference
Jan	24,903	61,468	-36,565
Feb	5,822	53,057	-47,235
Mar	35,183	122,214	-87,031
Apr	73,633	150,666	-77,033
May	183,922	240,101	-56,179
Jun	155,494	297,153	-141,659
Jul	171,814	292,096	-120,282
Aug	190,770	306,173	-115,403
Sep	166,920	287,868	-120,948
Oct	233,206	133,942	99,264
Nov	155,089	87,839	67,250
Dec	37,696	108,153	-70,457

See Figure @ref(fig:heatmaps) in the Appendix for a density map correlating start month and hour among casual and annual membership.

6. Pre-regression analysis

Looking at the above figures, it seems visually that the date and time of rental bear a *non*-linear relationship to the total number of bike rentals in any given month. Indeed, Figure @ref(fig:distribCount) seems to exhibit a rough normal curve over the year with a peak in the summer months, and Figure @ref(fig:distribStartTime) shows a similar, 'normal-ish' curve over the time of day, regardless of month.

However, the month and user type appear to apply something of a static multiplier to each of these distributions. Although they are not likely to be perfectly linear across their enumeration, the impact that, say, being a casual vs. annual member has on total rentals in April appears more linear.

7. Regression analysis

```
combined_model <- df |>
    mutate(month = factor(month, ordered = FALSE)) |>
    count(month, hour, user_type) |>
    glm(
        formula = n ~ month + hour + user_type,
        family = poisson
    )

summary(combined_model)
```

```
##
## Call:
## glm(formula = n ~ month + hour + user_type, family = poisson,
##     data = count(mutate(df, month = factor(month, ordered = FALSE)),
##         month, hour, user_type))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -74.942  -17.155   -3.017   15.318   76.110
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           6.582349   0.005614 1172.53   <2e-16 ***
## monthFeb             -0.383167   0.005344  -71.70   <2e-16 ***
## monthMar              0.600119   0.004235  141.72   <2e-16 ***
## monthApr              0.954328   0.004005  238.31   <2e-16 ***
## monthMay              1.591136   0.003733  426.22   <2e-16 ***
## monthJun              1.656461   0.003713  446.11   <2e-16 ***
## monthJul              1.681039   0.003706  453.61   <2e-16 ***
## monthAug              1.749823   0.003686  474.66   <2e-16 ***
## monthSep              1.661179   0.003712  447.55   <2e-16 ***
## monthOct              1.447113   0.003782  382.66   <2e-16 ***
## monthNov              1.034113   0.003962  261.03   <2e-16 ***
## monthDec              0.523920   0.004294  122.03   <2e-16 ***
## hour1                -0.441196   0.007153  -61.68   <2e-16 ***
## hour2                -0.824643   0.008107 -101.73   <2e-16 ***
## hour3                -1.389665   0.010021 -138.68   <2e-16 ***
## hour4                -1.763715   0.011700 -150.75   <2e-16 ***
```

```
## hour5                  -1.245996   0.009469 -131.59   <2e-16 ***
## hour6                  -0.100658   0.006495  -15.50   <2e-16 ***
## hour7                   0.622555   0.005548  112.22   <2e-16 ***
## hour8                   1.032010   0.005212  198.01   <2e-16 ***
## hour9                   0.931272   0.005284  176.24   <2e-16 ***
## hour10                  0.980444   0.005248  186.82   <2e-16 ***
## hour11                  1.198715   0.005106  234.78   <2e-16 ***
## hour12                  1.419956   0.004987  284.73   <2e-16 ***
## hour13                  1.484980   0.004956  299.61   <2e-16 ***
## hour14                  1.558258   0.004924  316.47   <2e-16 ***
## hour15                  1.643807   0.004889  336.25   <2e-16 ***
## hour16                  1.767185   0.004843  364.93   <2e-16 ***
## hour17                  1.945203   0.004785  406.56   <2e-16 ***
## hour18                  1.914170   0.004794  399.29   <2e-16 ***
## hour19                  1.733443   0.004855  357.07   <2e-16 ***
## hour20                  1.450278   0.004972  291.66   <2e-16 ***
## hour21                  1.099366   0.005167  212.76   <2e-16 ***
## hour22                  0.761443   0.005421  140.47   <2e-16 ***
## hour23                  0.436374   0.005742   75.99   <2e-16 ***
## user_typeCasual Member -0.400364   0.001079 -371.05   <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: 3599484  on 575  degrees of freedom
## Residual deviance:  380700  on 540  degrees of freedom
## AIC: 386396
##
## Number of Fisher Scoring iterations: 5
```

8. Coefficient exponentiation

```
exponentiated_coefficients <- exp(summary(combined_model)$coefficients)
exponentiated_coefficients |> knitr::kable()
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	722.2336225	1.005630	Inf	1
monthFeb	0.6816987	1.005359	0.000000e+00	1
monthMar	1.8223362	1.004244	3.533925e+61	1
monthApr	2.5969249	1.004013	3.144312e+103	1
monthMay	4.9093214	1.003740	1.272616e+185	1
monthJun	5.2407289	1.003720	5.542402e+193	1
monthJul	5.3711315	1.003713	1.004556e+197	1
monthAug	5.7535863	1.003693	1.383877e+206	1
monthSep	5.2655174	1.003719	2.338545e+194	1
monthOct	4.2508249	1.003789	1.533475e+166	1
monthNov	2.8126107	1.003969	2.318687e+113	1
monthDec	1.6886339	1.004303	9.886017e+52	1
hour1	0.6432663	1.007179	0.000000e+00	1
hour2	0.4383913	1.008139	0.000000e+00	1
hour3	0.2491588	1.010071	0.000000e+00	1
hour4	0.1714068	1.011768	0.000000e+00	1
hour5	0.2876542	1.009514	0.000000e+00	1
hour6	0.9042421	1.006516	2.000000e-07	1
hour7	1.8636837	1.005563	5.456057e+48	1
hour8	2.8067017	1.005226	9.844189e+85	1
hour9	2.5377343	1.005298	3.474142e+76	1
hour10	2.6656385	1.005262	1.362584e+81	1
hour11	3.3158548	1.005119	9.154765e+101	1
hour12	4.1369372	1.005000	4.538891e+123	1
hour13	4.4148774	1.004969	1.316418e+130	1
hour14	4.7505408	1.004936	2.754078e+137	1
hour15	5.1748318	1.004901	1.072054e+146	1
hour16	5.8543503	1.004854	3.071545e+158	1
hour17	6.9950529	1.004796	3.678248e+176	1
hour18	6.7813051	1.004806	2.554111e+173	1
hour19	5.6601106	1.004866	1.184616e+155	1
hour20	4.2643006	1.004985	4.638265e+126	1
hour21	3.0022633	1.005181	2.509985e+92	1
hour22	2.1413636	1.005435	1.016288e+61	1
hour23	1.5470878	1.005759	1.006526e+33	1
user_typeCasual Member	0.6700761	1.001080	0.000000e+00	1

Analysis

9. Coefficients

Our first exponentiated coefficient is the intercept rate, which in this case is the expected total city bike trips taken by annual members, beginning at midnight, through the month of January. In other words, our models predicts that 722 annual members would rent a city bike between midnight and 1 AM. The remaining coefficients represent a ratio by which, if all other values are held constant, this rate would be expected to change. For example, the coefficient for casual member rentals is 0.67, so we would expect something like $722 \times 0.67 \approx 484$ total rentals from casual members from midnight to 1AM in January. These coefficients can be multiplied together, one per categorical independent variable, to specify additional parameters.

The patterns of these coefficients across different levels of each variable resemble the visual properties of Figure @ref(fig:distribStartTime). For example, `user_typeCasual Member` being a negative logarithmic value makes sense, as there appeared to be something of a general diminishing effect on rentals regardless of time of day or month. The highest multiplicative values for month are found in the summer months (June – September), and February – often the coldest month – has a *chilling* effect on rentals. As for hours, the late afternoon and early evening (4PM – 7PM) are correlated with larger increases in rental rate, whereas the small hours of the morning (1AM – 5AM) are correlated with large reductions in rate, before more than doubling several times over into 8AM.

10. Significance

The summary table for the regression in 7. Regression Analysis calculates that the p-value for every row is less than 2×10^{-16}, meaning that each value of each variable is a highly significant predictor of rental rates. Calculating a simple McFadden's pseudo-R^2 for a rough approximation of goodness of fit yields:

```
mcfadden_r2 <- with(summary(combined_model), 1 - deviance/null.deviance)
mcfadden_r2
```

```
## [1] 0.8942349
```

So, our model is able to account for roughly 89% of variance in city bike rentals over hours, months, and types of renting members.

Extra Credit: Predictions

Because all of our independent variables are categorical and finite, we can actually use `expand.grid` to calculate model predictions for all 576 possible values that it can account for. Figure @ref(fig:heatmaps) in the Appendix shows density maps for bike rentals in the real data, as well as our Poissonian model. The two are visually similar, but the model predictions look more 'de-noised' or regular compared to the Open Data Toronto plot. But, it misses some interesting details, such as the higher density of casual member rentals in October compared to annual members.

Appendix

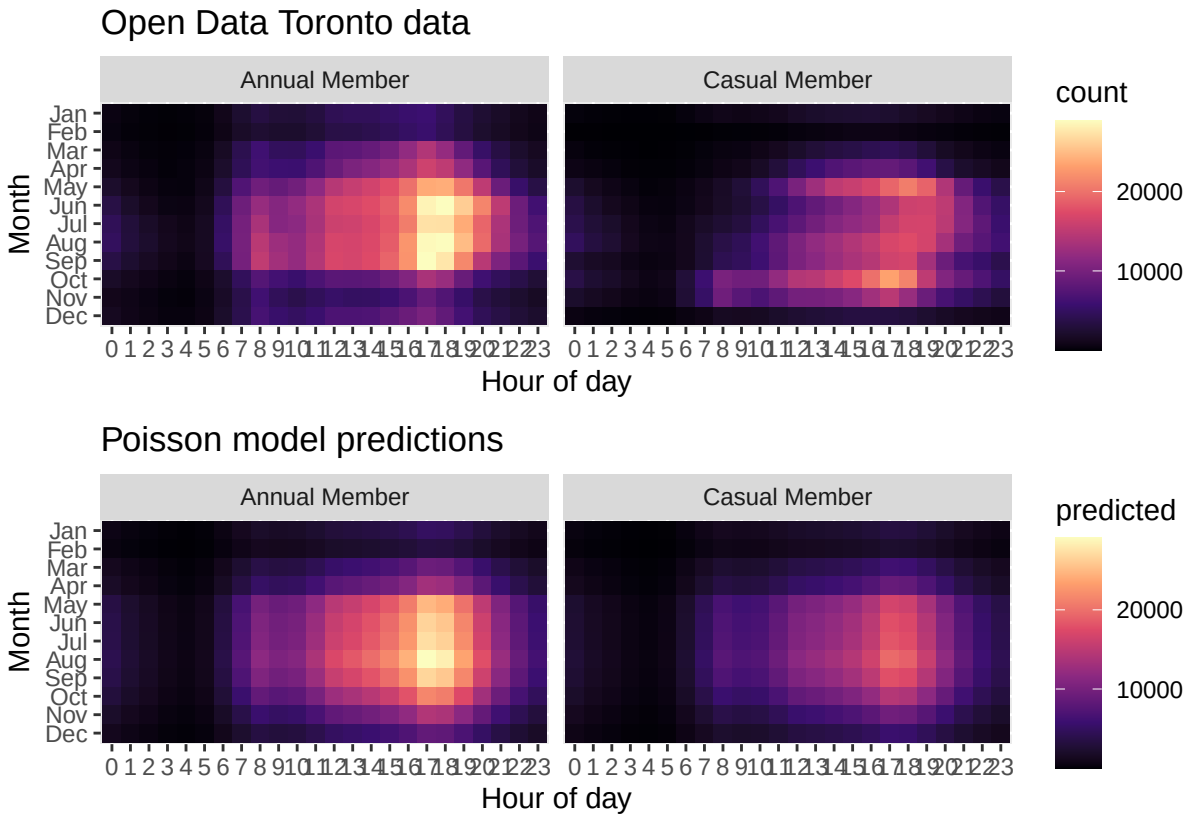


Figure 3: Comparison of real vs. predicted data for 2021 City Bike rentals