Inference

Jason Luo

2025-01-02

```r
rm(list = ls())

setwd('/Users/jasonluo/Documents/R_crime_analysis')
library(tidyverse)
library(faraway)

# Importing cleaned data
df <- read.csv('RMS_Crime_Incidents_Cleaned.csv')

unique(df$year)
```

```
## [1] 2021 2022 2023 2024
```

χ^2 test

We run a χ^2 test for Homogeneity, testing if there is a difference in the amount of crimes that occur in each zipcode per year

```r
get_num_crimes_by_zip_year <- function(data, yr) {


  print(paste0("Year: ", yr))
  crimes <- data %>%
    select(year, zip_code) %>%
    filter(year == yr) %>%
    group_by(zip_code) %>%
    summarise(num_crimes = n())

  return(crimes)
}


num_crimes_by_zip_year <- list()
years = c(2021,2022,2023)

for (i in seq_along(years)) {
  crimes <- get_num_crimes_by_zip_year(df, years[i])
  num_crimes_by_zip_year[[i]] <- crimes$num_crimes
}
```

```
## [1] "Year: 2021"
## [1] "Year: 2022"
## [1] "Year: 2023"
```

```r
zip_codes <- crimes$zip_code # all 3 years have the same zip codes
contingency_table <- rbind(num_crimes_by_zip_year[[1]], num_crimes_by_zip_year[[2]], num_crimes_by_zip_
dimnames(contingency_table) <- list(Year = years, Zipcode = zip_codes)

contingency_table
```

```
##       Zipcode
## Year   48201 48202 48203 48204 48205 48206 48207 48208 48209 48210 48211 48212
##   2021  1682  1592  1986  2547  4795  1732  2583  1172  1884  1914   475  1267
##   2022  2323  1764  2375  2576  4744  1685  2904  1180  1893  1917   495  1286
##   2023  2895  2119  2630  2997  5264  1912  3199  1242  1987  2048   492  1317
##       Zipcode
## Year   48213 48214 48215 48216 48217 48219 48221 48223 48224 48226 48227 48228
##   2021  2513  1758  1342   664   615  4822  3204  2420  4633  1766  5038  6010
##   2022  2700  1989  1472   771   517  4964  3584  2418  4866  2469  5142  5985
##   2023  2768  2010  1603   835   540  5140  3626  2610  5142  2835  5655  6091
##       Zipcode
## Year   48234 48235 48236 48238 48239 48243
##   2021  3823  4689   262  3315   410    53
##   2022  4092  5031   280  3443   400    50
##   2023  4078  5231   326  3839   408    52
```

Displaying the χ^2 test results

```r
Xsq <- chisq.test(contingency_table)
Xsq
```

```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 564.47, df = 58, p-value < 2.2e-16
```

Other related quantities:

```r
Xsq$observed   # observed counts
```

```
##       Zipcode
## Year   48201 48202 48203 48204 48205 48206 48207 48208 48209 48210 48211 48212
##   2021  1682  1592  1986  2547  4795  1732  2583  1172  1884  1914   475  1267
##   2022  2323  1764  2375  2576  4744  1685  2904  1180  1893  1917   495  1286
##   2023  2895  2119  2630  2997  5264  1912  3199  1242  1987  2048   492  1317
##       Zipcode
## Year   48213 48214 48215 48216 48217 48219 48221 48223 48224 48226 48227 48228
##   2021  2513  1758  1342   664   615  4822  3204  2420  4633  1766  5038  6010
##   2022  2700  1989  1472   771   517  4964  3584  2418  4866  2469  5142  5985
##   2023  2768  2010  1603   835   540  5140  3626  2610  5142  2835  5655  6091
##       Zipcode
## Year   48234 48235 48236 48238 48239 48243
##   2021  3823  4689   262  3315   410    53
##   2022  4092  5031   280  3443   400    50
##   2023  4078  5231   326  3839   408    52
```

```r
Xsq$expected   # expected counts under the null
```

```
##       Zipcode
## Year       48201     48202     48203     48204     48205     48206     48207     48208
##   2021 2155.483 1710.329 2183.910 2536.597 4624.292 1664.720 2713.410 1122.726
##   2022 2287.577 1815.143 2317.747 2692.047 4907.682 1766.739 2879.695 1191.529
##   2023 2456.940 1949.528 2489.343 2891.355 5271.026 1897.541 3092.895 1279.745
##       Zipcode
## Year      48209    48210    48211    48212    48213    48214    48215    48216
##   2021 1800.609 1836.534 456.7125 1208.945 2493.175 1798.423 1379.822 709.1227
##   2022 1910.956 1949.082 484.7012 1283.032 2645.964 1908.635 1464.381 752.5798
##   2023 2052.435 2093.384 520.5863 1378.023 2841.860 2049.942 1572.797 808.2975
##       Zipcode
## Year      48217    48219    48221    48223    48224    48226    48227    48228
##   2021 522.3142 4662.716 3253.217 2326.672 4573.685 2208.589 4946.677 5649.865
##   2022 554.3231 4948.461 3452.584 2469.257 4853.974 2343.938 5249.824 5996.105
##   2023 595.3628 5314.823 3708.199 2652.071 5213.341 2517.473 5638.499 6440.031
##       Zipcode
## Year      48234    48235    48236    48238    48239    48243
##   2021 3746.479 4670.526 271.1535 3310.385 380.4896 48.42027
##   2022 3976.074 4956.749 287.7706 3513.255 403.8071 51.38760
##   2023 4270.446 5323.725 309.0759 3773.361 433.7033 55.19212
```

```
Xsq$residuals   # Pearson residuals
```

```
##       Zipcode
## Year         48201     48202     48203      48204       48205      48206
##   2021 -10.1984025 -2.861220 -4.234980  0.2065443  2.51033015  1.6489753
##   2022   0.7406181 -1.200409  1.189234 -2.2366302 -2.33648409 -1.9446578
##   2023   8.8376475  3.838244  2.819159  1.9647072 -0.09677166  0.3319316
##       Zipcode
## Year        48207      48208      48209      48210      48211       48212
##   2021 -2.5035239  1.4705682  1.9652042  1.8076375  0.8557224  1.66969631
##   2022  0.4529195 -0.3340052 -0.4107527 -0.7266882  0.4677908  0.08284719
##   2023  1.9078830 -1.0551127 -1.4443560 -0.9919199 -1.2528881 -1.64385364
##       Zipcode
## Year        48213      48214      48215      48216     48217      48219
##   2021  0.3970339 -0.9531879 -1.0181878 -1.6944716  4.055529  2.3326668
##   2022  1.0504813  1.8395217  0.1991005  0.6714581 -1.585243  0.2209019
##   2023 -1.3855087 -0.8821897  0.7615649  0.9392162 -2.268961 -2.3980332
##       Zipcode
## Year        48221      48223      48224     48226      48227      48228
##   2021 -0.8629049  1.9348336  0.8770618 -9.417667  1.2984380  4.7912254
##   2022  2.2365373 -1.0315079  0.1726168  2.583169 -1.4881380 -0.1434072
##   2023 -1.3498407 -0.8169313 -0.9880568  6.328464  0.2197563 -4.3493014
##       Zipcode
## Year       48234      48235      48236       48238      48239      48243
##   2021  1.250162  0.2703242 -0.5558795  0.08021679  1.5128762  0.6581518
##   2022  1.838450  1.0546402 -0.4580688 -1.18527587 -0.1894563 -0.1935694
##   2023 -2.944912 -1.2708397  0.9626608  1.06855996 -1.2342182 -0.4296760
```

```
Xsq$stdres      # standardized residual
```

```
##       Zipcode
## Year         48201     48202     48203      48204      48205      48206
##   2021 -12.4898852 -3.492828 -5.187610  0.2536563  3.1310588  2.0123209
##   2022   0.9199243 -1.486233  1.477456 -2.7858516 -2.9556613 -2.4068982
##   2023  11.1845333  4.841875  3.568540  2.4933604 -0.1247279  0.4185879
##       Zipcode
## Year        48207      48208      48209      48210      48211      48212
##   2021 -3.0785477  1.7876257  2.4005844  2.2086833  1.0352934  2.0309402
##   2022  0.5648675 -0.4117902 -0.5088867 -0.9005372  0.5740023  0.1022042
##   2023  2.4243805 -1.3253945 -1.8232173 -1.2524304 -1.5663822 -2.0662253
##       Zipcode
## Year        48213     48214      48215      48216     48217      48219
##   2021  0.4874412 -1.164343 -1.2399955 -2.0537311  4.908855  2.9103076
##   2022  1.3080199  2.278971  0.2459212  0.8253909 -1.946077  0.2795226
##   2023 -1.7577567 -1.113575  0.9584159  1.1763315 -2.838014 -3.0916924
##       Zipcode
## Year       48221     48223      48224      48226     48227      48228     48234
##   2021 -1.065323  2.372525  1.0935157 -11.538179  1.623452  6.0226839  1.549074
##   2022  2.800438 -1.282836  0.2182777   3.209802 -1.887091 -0.1828294  2.310411
##   2023 -1.722090 -1.035160 -1.2730094   8.012114  0.283932 -5.6496023 -3.770798
##       Zipcode
## Year        48235      48236        48238      48239      48243
##   2021  0.3372847 -0.6716460  0.09907568  1.8293606  0.7939677
##   2022  1.3345890 -0.5613348 -1.48474774 -0.2323467 -0.2368344
##   2023 -1.6385415  1.2019545  1.36381536 -1.5422073 -0.5356397
```

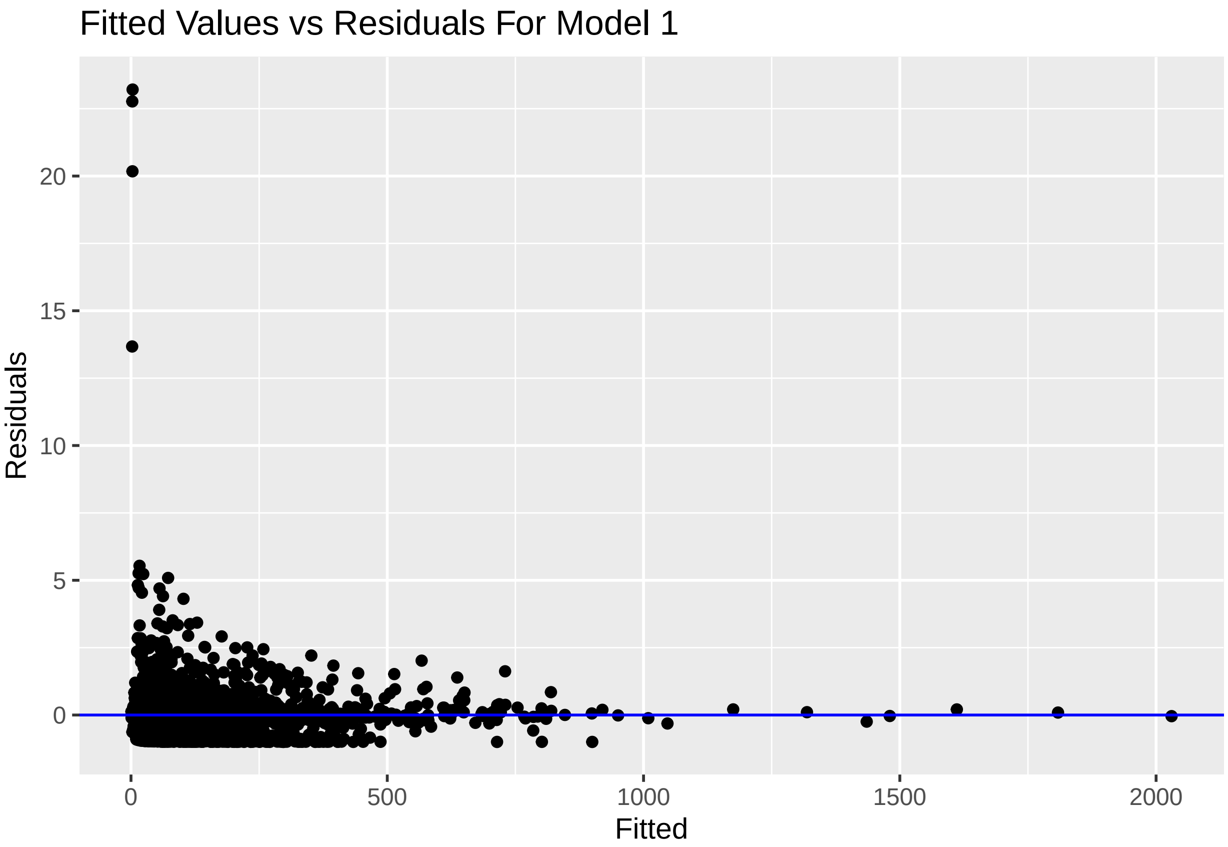
Poisson Regression

Using poisson regression to model number (count) of crime incidents based off of neighborhood, council district, year of occurence, and zip code

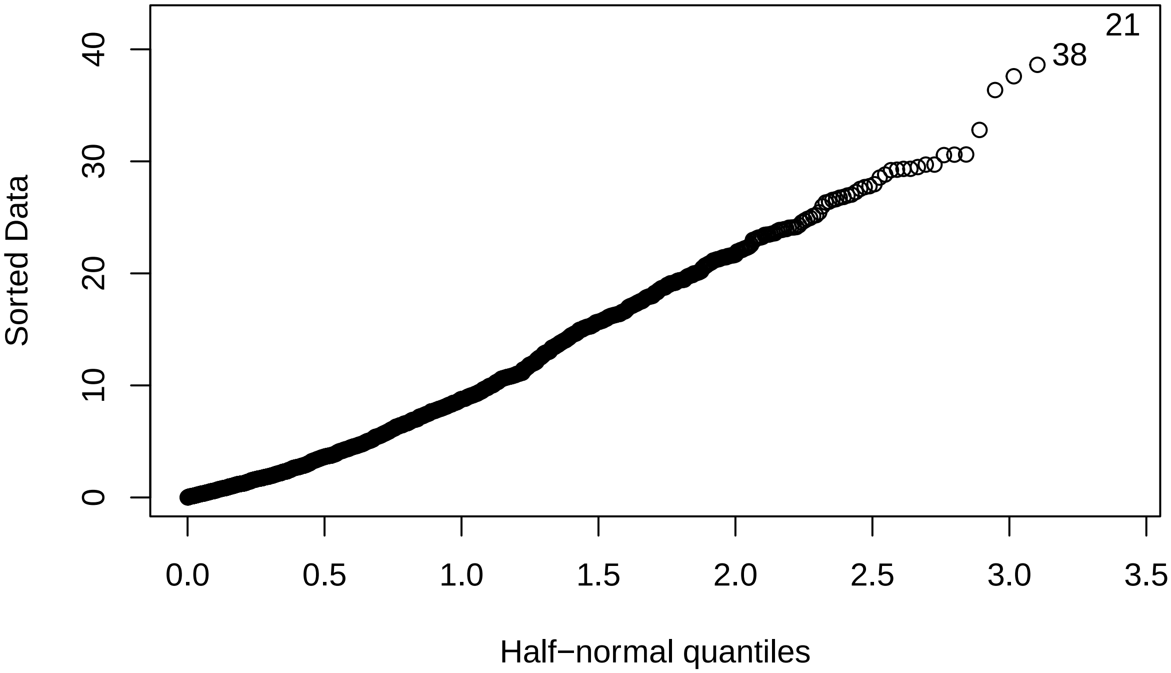
```
df_agg <- df %>%
  group_by(zip_code, year, council_district, neighborhood) %>%
  summarise(num_crimes = n()) %>%
  mutate(zip_code = as.factor(zip_code))
```

```
## `summarise()` has grouped output by 'zip_code', 'year', 'council_district'. You
## can override using the `.groups` argument.
```

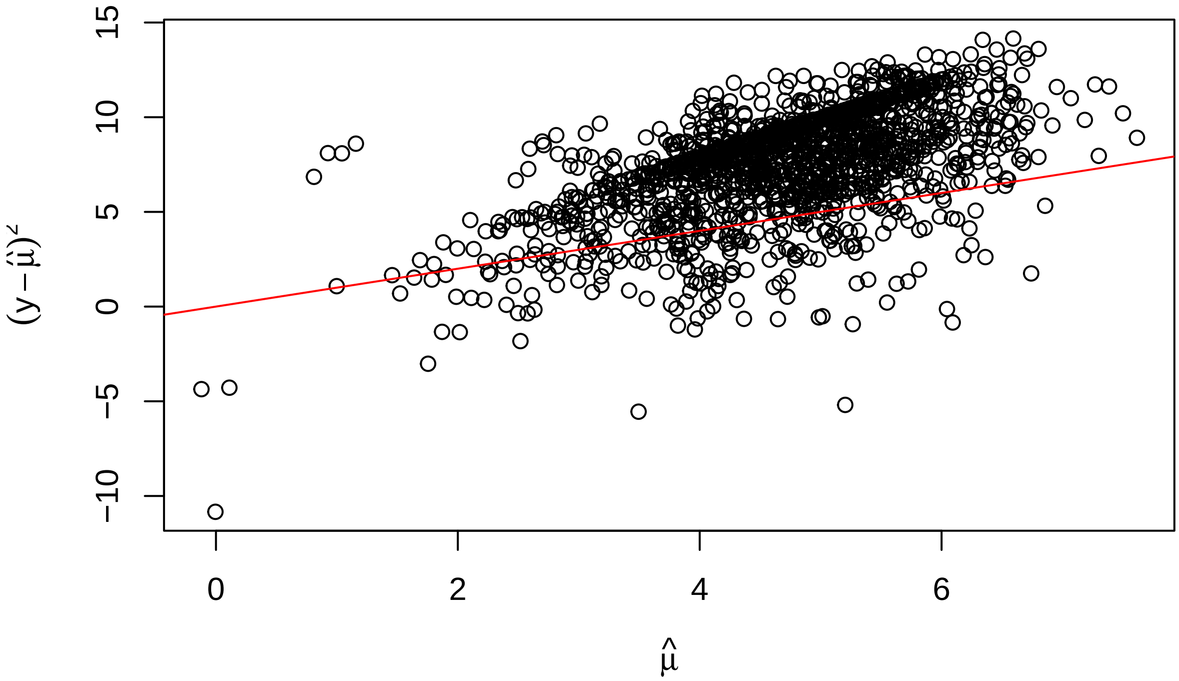
```
model1 <- glm(formula = num_crimes ~ neighborhood + council_district + year + zip_code,
              data = df_agg, family = poisson(link = 'log'))

# Residuals
#plot(model1$fitted.values, model1$residuals, xlab="Fitted",ylab="Residuals")
#abline(h=0, col = 'red')
ggplot() +
  geom_point(aes(x = model1$fitted.values, y = model1$residuals)) +
  geom_abline(intercept = 0, slope = 0, color = 'blue') +
  xlab('Fitted') +
  ylab('Residuals') +
  ggtitle('Fitted Values vs Residuals For Model 1')
```



```r
# Half-Norm plot of residuals for checking outliers
halfnorm(residuals(model1))
```



```r
# Checking relationship between mean and variance
plot(log(fitted(model1)),log((df_agg$num_crimes-fitted(model1))^2), xlab=expression(hat(mu)),ylab=expre
abline(0,1, col = 'red')
```

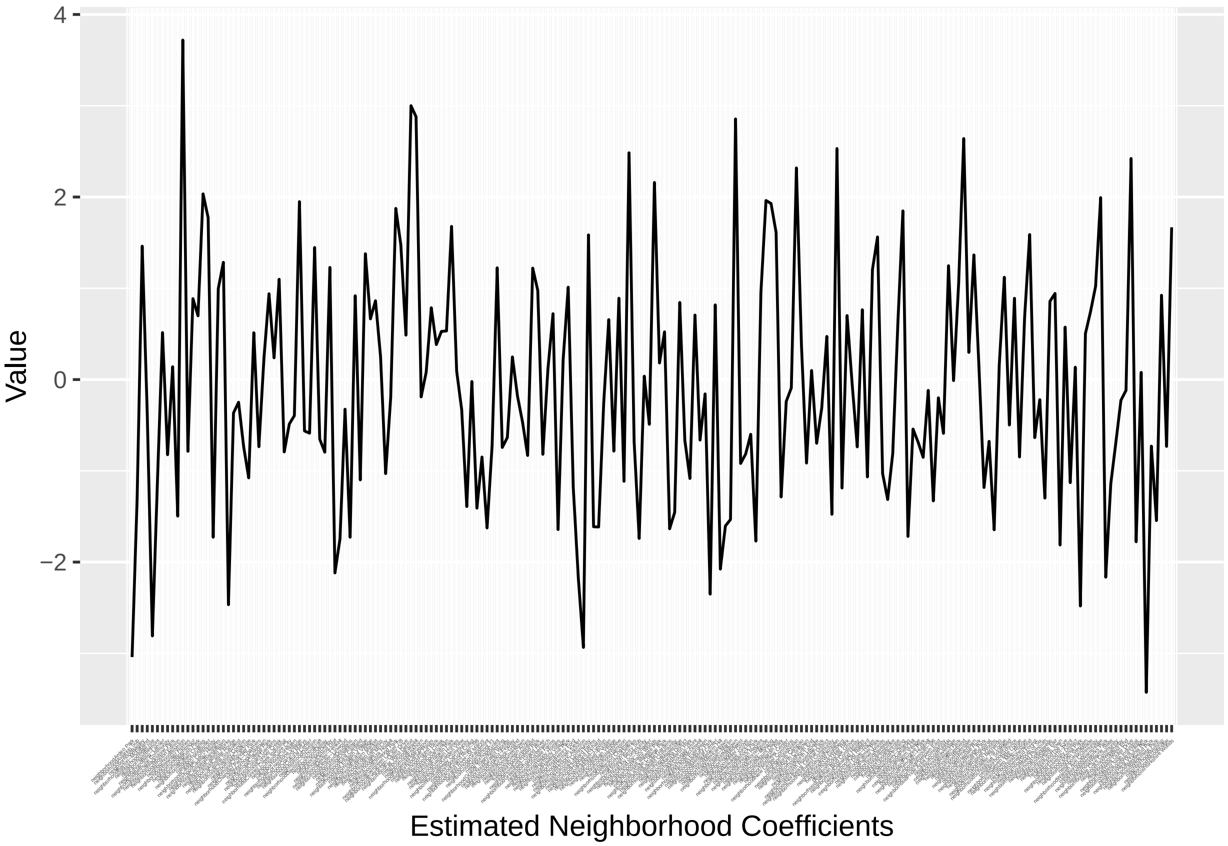


Plotting values of coefficients for the various qualitative variables:

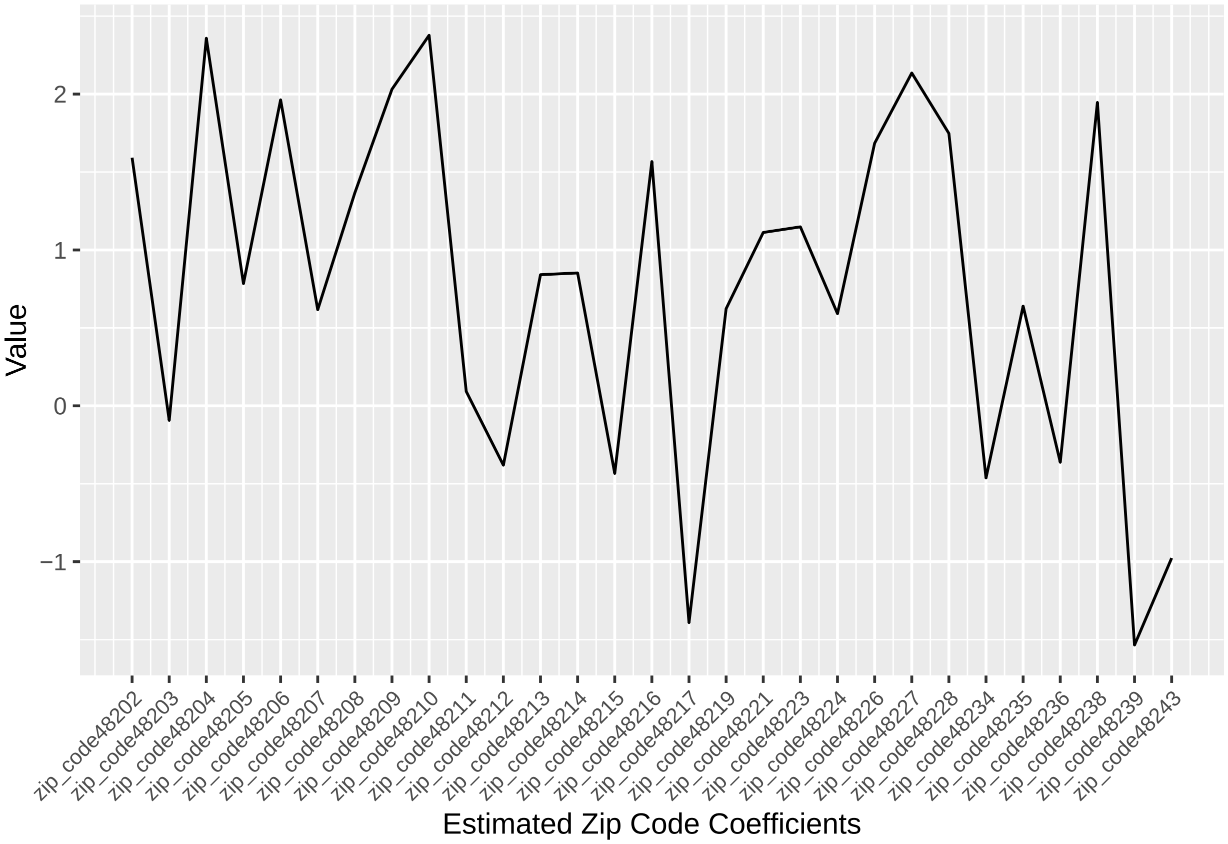
```r
length(model1$coefficients)
```

```
## [1] 238
```

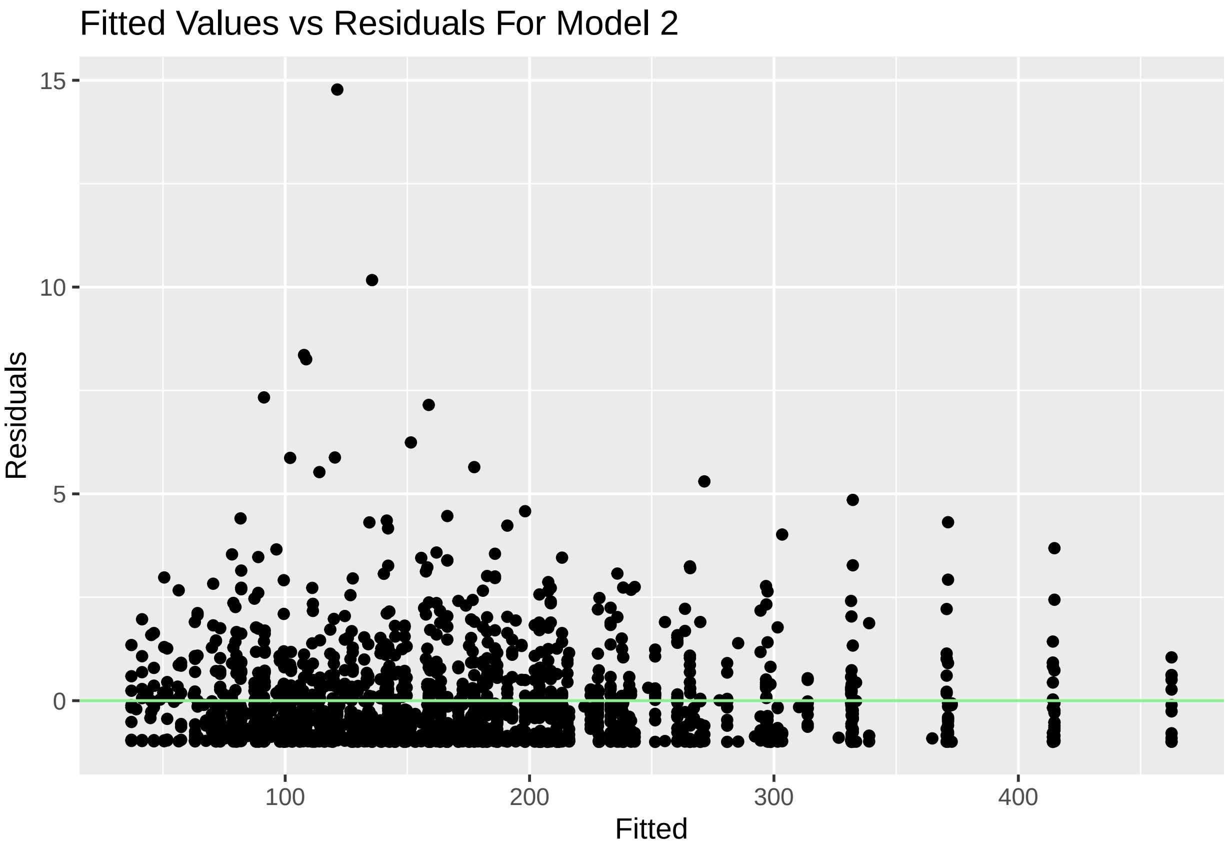
```r
ggplot() +
  geom_line(aes(x = 1:206, y = model1$coefficients[2:207])) +
  #scale_x_discrete(labels = names(model1$coefficients[2:207])) +
  scale_x_continuous(breaks = 1:206, labels = names(model1$coefficients[2:207])) +
  xlab("Estimated Neighborhood Coefficients") +
  ylab("Value") +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 2))
```



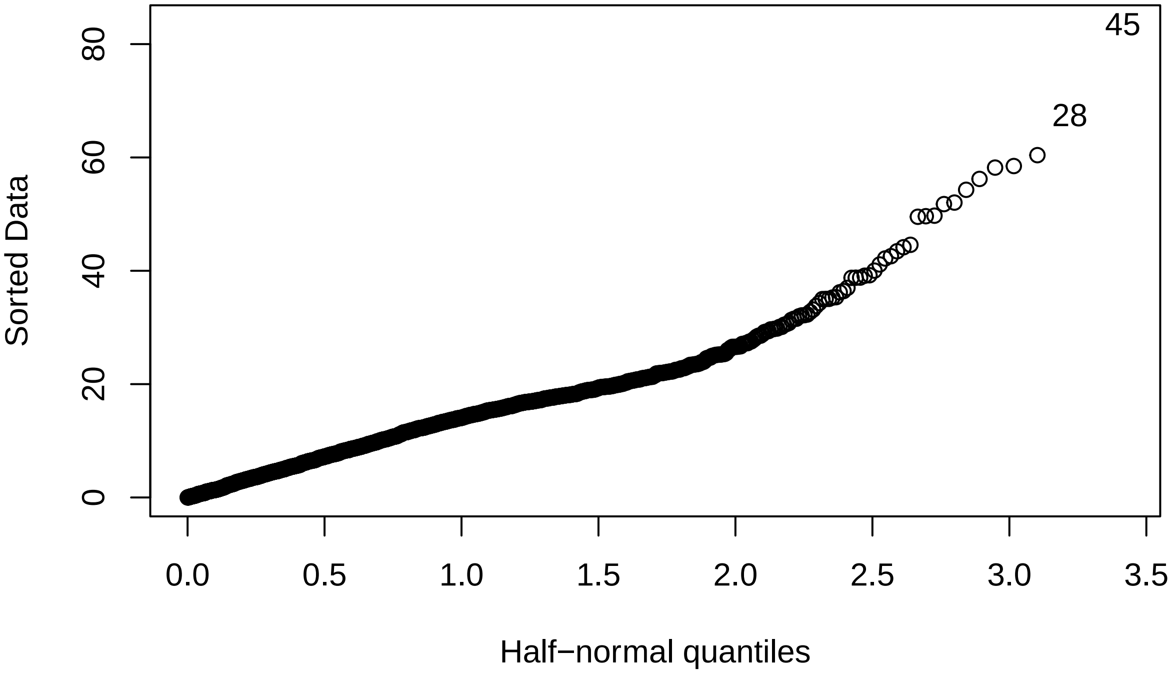
```r
ggplot() +
  geom_line(aes(x = 1:29, y = model1$coefficients[210:238])) +
  scale_x_continuous(breaks = 1:29, labels = names(model1$coefficients[210:238])) +
  xlab("Estimated Zip Code Coefficients") +
  ylab("Value") +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 8))
```



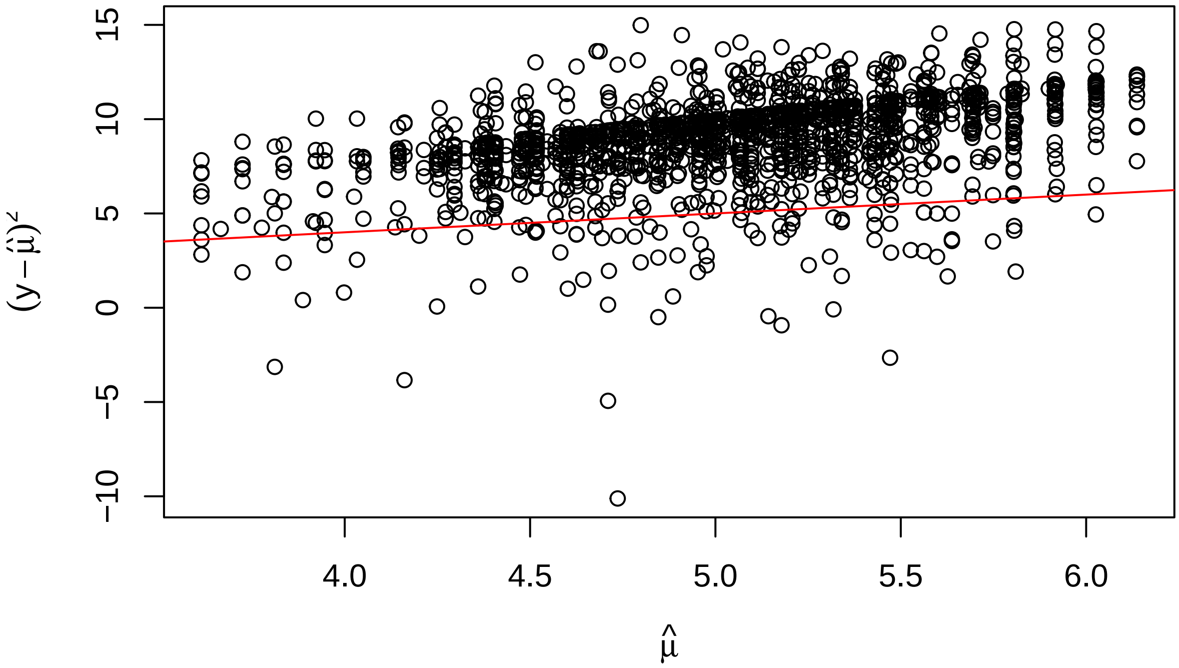
```r
model2 <- glm(formula = num_crimes ~ council_district + year + zip_code,
              data = df_agg, family = poisson(link = 'log'))

# Residuals
ggplot() +
  geom_point(aes(x = model2$fitted.values, y = model2$residuals)) +
  geom_abline(intercept = 0, slope = 0, color = 'lightgreen') +
  xlab('Fitted') +
  ylab('Residuals') +
  ggtitle('Fitted Values vs Residuals For Model 2')
```



```
# Half-Norm plot of residuals for checking outliers
halfnorm(residuals(model2))
```



```
# Checking relationship between mean and variance
plot(log(fitted(model2)),log((df_agg$num_crimes-fitted(model2))^2), xlab=expression(hat(mu)),ylab=expres
abline(0,1, col = 'red')
```



Comparing models, model 1 has a AIC of 1.6714464×10^5 vs model 2 which has 3.2361333×10^5. The model with the smaller AIC considered better performing in terms of complexity and performance