Demand for health care in Australia

Final Project for Statistical Methods

Group O: Buscema Andrea, Cusma Fait Omar, Derin Tanja, Špringer Christian, Živanović Uroš

2024-03-01

ABSTRACT

In this final project for the course “344SM - Statistical Methods” we will analyze the dataset based on the Australian health survey from 1977 to 1978. The dataset contains information of 5190 individuals and 19 variables, including the response variable ‘doctorco’, which is the number of consulations with a doctor or specialist in the past 2 weeks.

For the purpose of this project, after the initial data exploration analysis, including the data cleaning and the data visualization, we will focus on the prediction of the number of consulations with a doctor or specialist in the past 2 weeks. We will use:

- Negative Binomial regression
- Zero-inflated Poisson regression

- Zero-inflated Negative Binomial regression
- Hurdle Poisson regression
- Hurdle Negative Binomial regression
- Logistic regression
- Random Forest
- Naive Bayes
- Neural Networks
- K-Nearest Neighbors

The whole project was implemented with two programming languages, R and Python, thereby harnessing the specialized capabilities of each to create a robust, efficient, and versatile analytical workflow. This dual-language approach facilitated the leveraging of R's advanced statistical analysis and visualization tools alongside Python's superior data manipulation, machine learning, and integration capabilities. As a result, the project benefited from the rich and diverse libraries of both ecosystems, ensuring that each stage of the project, from data cleaning to complex statistical modeling and even deployment, was handled with the most effective tools available. This method not only enhanced the overall quality and depth of the analysis but also fostered a flexible and innovative environment for tackling the multifaceted challenges of the project.

##Dataset

The dataset contains information divided into these 19 variables:

- **Categorical (Binary) variables:**
  - 'sex': gender of the individual (1 if female, 0 if male);
  - 'levyplus': 1 if covered by private health insurance fund for private patient in public hospital (with doctor of choice), 0 otherwise;
  - 'freepoor': 1 if covered by government because of low income, recent immigrant, unemployed, 0 otherwise;
  - 'freepera': 1 if covered by government because of old-age or disability pension, or because invalid veteran or family of deceased veteran, 0 otherwise;
  - 'chcond1': 1 if chronic condition(s) but not limited in activity, 0 otherwise;
  - 'chcond2': 1 if chronic condition(s) and limited in activity, 0 otherwise;
- **Categorical (Ordinal) variables:**
  - 'age': age in years divided by 100 (measured as mid-point of the age groups from 15-19 years to 65-69 years with the last group being 70 years and over treated as 72);
  - 'agesq': 'age' squared;
  - 'income': Annual income in Australian dollars divided by 1000 (measured as mid-point of coded ranges Nil, < 200, 200-1000, 1001-, 2001-, 3001-, 4001-, 5001-, 6001-, 7001-, 8001-10000, 10001-12000, 12001-14000, with 14001- treated as 15000);
  - 'hscore': General health questionnaire score using Goldberg's method. High score indicates bad health;
- **Discrete variables:**
  - 'illness': Number of illnesses in the past 2 weeks with 5 or more coded as 5;
  - 'actdays': Number of days in the past 2 weeks with activity limitation due to illness or injury;
  - 'doctorco': (Response variable) Number of consultations with a doctor or specialist in the past 2 weeks;
  - 'nondocco': Number of consultations with non-doctor health professional (chemist, optician, physiotherapist, social worker, district community nurse, chiropodist or chiropractor, etc.) in the past 2 weeks;
  - 'hospadmi': Number of admissions to a hospital, psychiatric hospital, nursing or convalescent home in the past 12 months (up to 5 or more admissions which is coded as 5);
  - 'hospdays': Number of nights in a hospital, etc. during most recent admission: taken, where appropriate, s the mid-point of the intervals 1, 2, 3, 4, 5, 6, 7, 8-14, 15-30, 31-60, 61-79 with 80 or more admissions coded as 80. If no admission in past 12 months the coded as 0;
  - 'medicine': Total number of prescribed and nonprescribed medications used in past 2 weeks;
  - 'prescrib': Number of prescribed medications used in past 2 weeks;
  - 'nonpresc': Number of nonprescribed medications used in past 2 weeks;

Considerations and before starting the analysis - Levy in Australia health care in 70-80s

Health care in Australia during 1977-1978 was is a transitional phase, with major focus on a universal health insurance system called Medibank. The Medibank scheme was introduced by the Whitlam Government in 1975 through the Health Insurance Act 1973. The scheme was intended to provide universal health insurance to all Australians and to provide free treatment in public hospitals. The scheme was to be funded by a 2.5% levy on taxable incomes, an additional levy on high-income earners, as well as government funding. The Medibank scheme was later abolished by the Fraser Government in 1981, and replaced by a government-subsidised private insurance scheme, called Medibank Private, which exists to this day. The dataset we are using is from the period when the Medibank scheme was in place, and it is interesting to see how the health care system was functioning at that time.

The Australian health care system has a federal structure, with responsibilities split between federal and state/territory governments. The federal government was primarly responsible for funding and policy, while the state/territory governments were responsible for the delivery of health care services. The cost of running Medibank was a significant concerns for the government during this period. There were debates aboyt whether Medibank provided truly equitable access for all Australians and concerns about longer waiting times in the public system.

Explanatory Data Analysis

Despite the oldness of the dataset, it doesn't contain any missing value and the data is clean. Firstly, we will explore the dataset by looking at the distribution of the response variable 'doctorco' and the distribution of the other variables. Then, we will look at the correlation between the variables and the response variable.

For this analysis, we excluded 'agesq' (age squared) from data frame while maintaining 'age' in this analysis to simplify models and make it more interpretable, and avoid potential issues like redundancy and multicollinearity, without significantly compromising the accuracy of analysis.

```
## Rows: 5190 Columns: 20
## ── Column specification ────────────────────────────────────────────────────
## Delimiter: ","
## dbl (20): sex, age, agesq, income, levyplus, freepoor, freepera, illness, ac...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Dataset Analysis

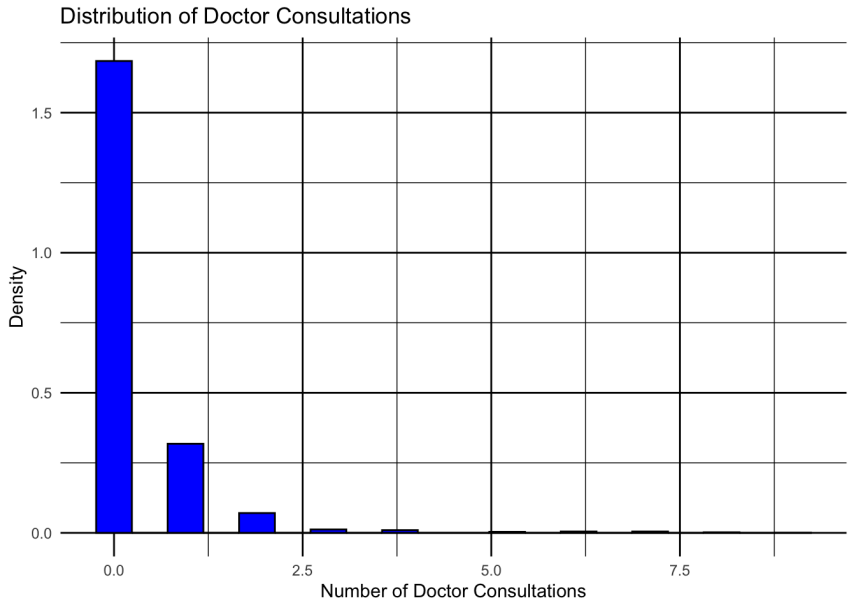
We start to analyse response variable and independent variables.

Response variable: 'doctorco'

```
describe(df$doctorco)
```

```
## df$doctorco
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     5190        0       10    0.489   0.3017   0.5154        0        0
##      .25      .50      .75      .90      .95
##        0        0        0        1        2
##
## Value          0     1     2     3     4     5     6     7     8     9
## Frequency   4141   782   174    30    24     9    12    12     5     1
## Proportion 0.798 0.151 0.034 0.006 0.005 0.002 0.002 0.002 0.001 0.000
##
## For the frequency table, variable is rounded to the nearest 0
```

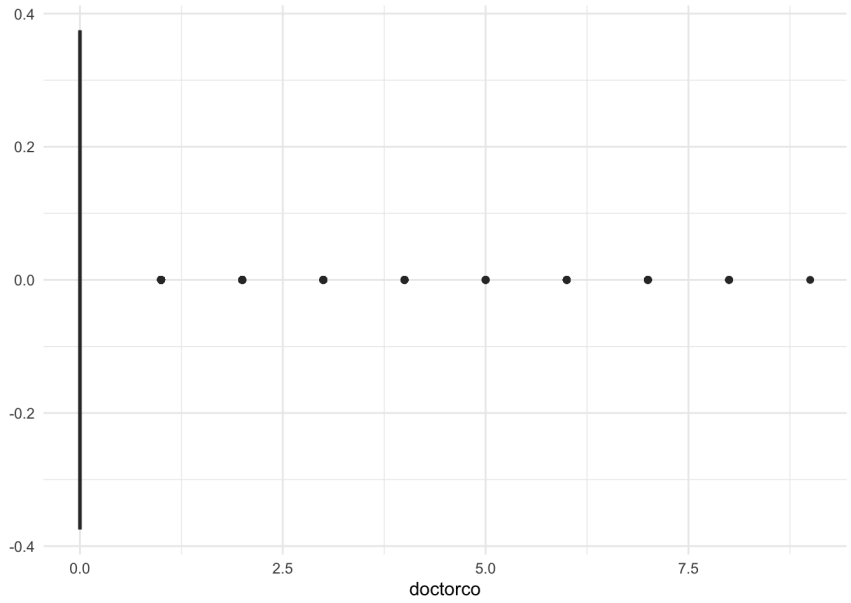
```
df %>%
  ggplot(aes(x = doctorco)) +
  geom_histogram(aes(y = after_stat(density)), color = "black", fill = "blue", bins = 20) +
  labs(title = "Distribution of Doctor Consultations",
       x = "Number of Doctor Consultations",
       y = "Density") +
  theme_minimal() +
  theme(panel.grid = element_line(colour = "black"))
```




```
# Summary Statistics
summary(df$doctorco)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3017  0.0000  9.0000
```

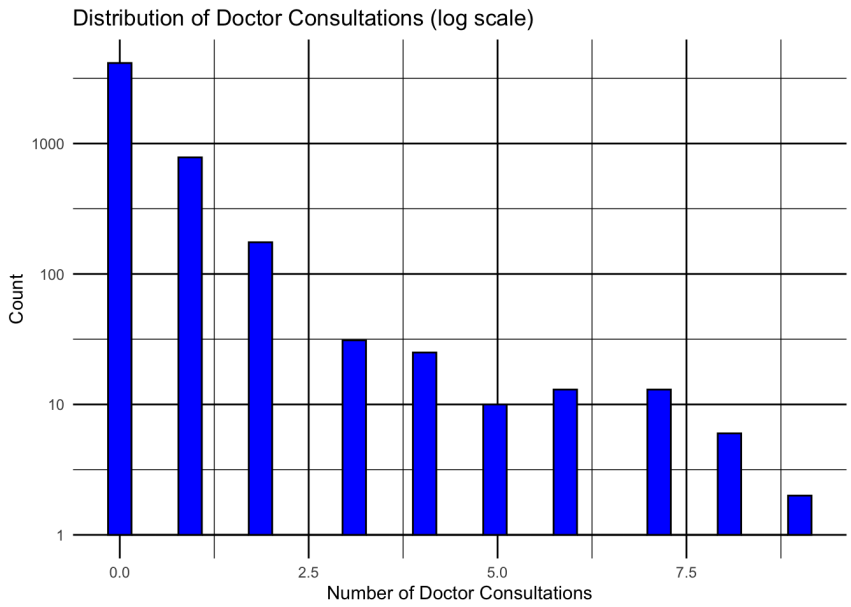
```
# Box plot for outlier visualization
df %>%
  ggplot(aes(x = doctorco)) +
  geom_boxplot() +
  theme_minimal()
```




```
# Value counts
table(df$doctorco)
```

```
##
##    0    1    2    3    4    5    6    7    8    9
## 4141  782  174   30   24    9   12   12    5    1
```

```
# Distribution of 'doctorco' (log scale for better visualization)
df %>%
  ggplot(aes(x = doctorco)) +
  geom_histogram(aes(y = after_stat(count + 1)), color = "black", fill = "blue", bins = 30) +
  labs(title = "Distribution of Doctor Consultations (log scale)",
       x = "Number of Doctor Consultations",
       y = "Count") +
  scale_y_continuous(trans = "log10") +  # Use log scale for y-axis
  theme_minimal() +
  theme(panel.grid = element_line(colour = "black"))
```




```
# Summary Statistics
print(summary(df$doctorco))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3017  0.0000  9.0000
```

```
# Value counts
print(table(df$doctorco))
```

```
##
##    0    1    2    3    4    5    6    7    8    9
## 4141  782  174   30   24    9   12   12    5    1
```

The variable ‘doctorco’ (doctor consultations) is the response variable in the dataset, reflecting the number of consultations with a doctor or specialist in the past 2 weeks. The distribution is highly skewed to the right, with the majority of individuals (79.8%) reporting zero consultations. This indicates a low overall rate of doctor consultations within the given timeframe. A small proportion of individuals have one or more consultations, but these are much less common.

The plots provided likely include a histogram showing the frequency distribution of consultations, a density plot highlighting the probability density of the different consultation counts, and a log-transformed histogram to better visualize the distribution of the non-zero consultation counts. The log scale is useful for distributions like this where there’s a large concentration of zero values and a long tail for higher values.

The first histogram shows a very high bar at 0 consultations, which greatly surpasses the frequency of any other number of consultations. This reflects the skewness towards lower numbers of doctor visits within the two-week period.

The second plot, likely a rug plot or a variation thereof, indicates the individual data points along the axis, emphasizing the concentration at 0 consultations and the sparsity of data points as the number of consultations increases.

The third histogram, presented on a log scale, adjusts the y-axis to better display the distribution of counts above 0. The height of the bars for counts of 1 or more consultations is more discernible on this scale, highlighting the long-tail nature of the distribution.

The distribution and mean of this variable are critical for healthcare planning and resource allocation, as they provide insight into the utilization of medical professional services.

Variables

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
##
## Attaching package: 'rlang'
```

```
## The following objects are masked from 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

Principal statistics:

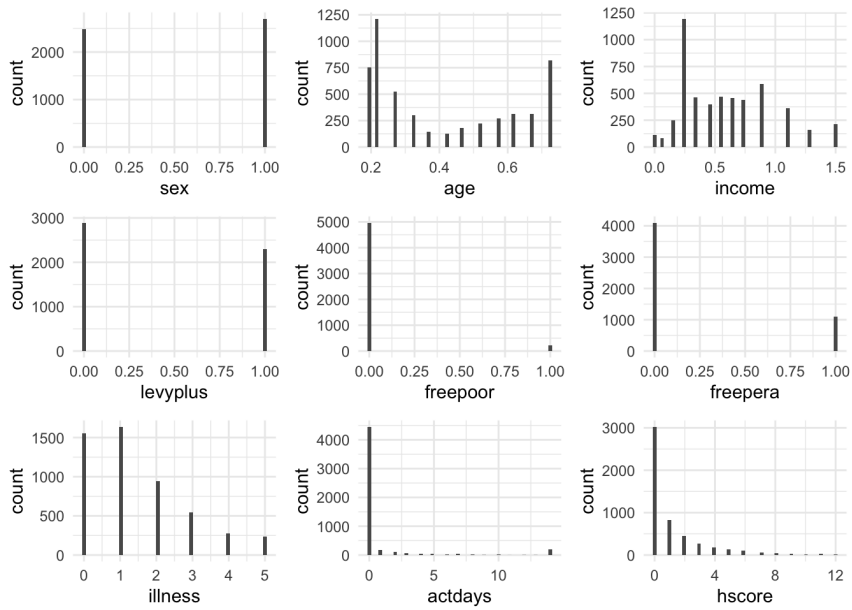
```
##describe(df_d)
```

```
summary(df_d)
```

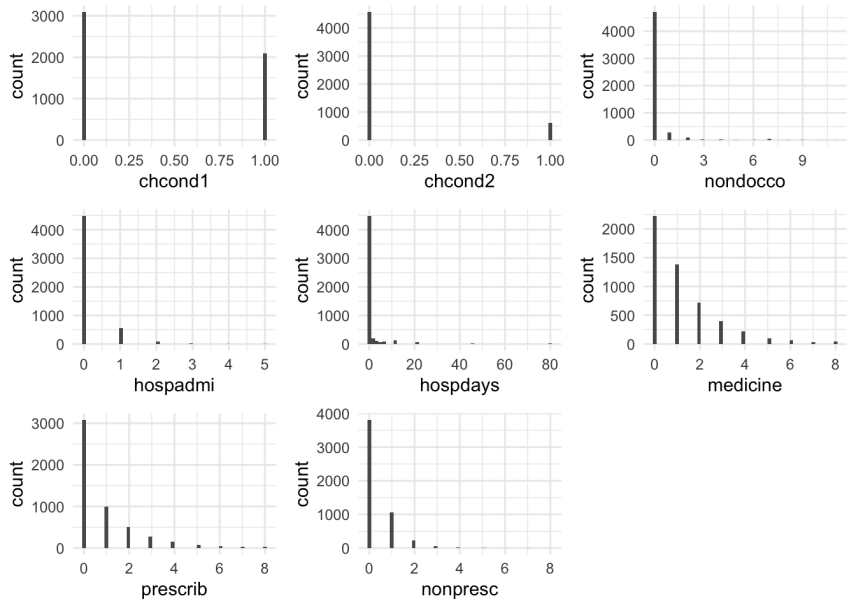
```
##       sex              age             income          levyplus
##  Min.   :0.0000   Min.   :0.1900   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.2200   1st Qu.:0.2500   1st Qu.:0.0000
##  Median :1.0000   Median :0.3200   Median :0.5500   Median :0.0000
##  Mean   :0.5206   Mean   :0.4064   Mean   :0.5832   Mean   :0.4428
##  3rd Qu.:1.0000   3rd Qu.:0.6200   3rd Qu.:0.9000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :0.7200   Max.   :1.5000   Max.   :1.0000
##     freepoor          freepera         illness         actdays
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.000   Min.   : 0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.: 0.0000
##  Median :0.00000   Median :0.0000   Median :1.000   Median : 0.0000
##  Mean   :0.04277   Mean   :0.2102   Mean   :1.432   Mean   : 0.8619
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:2.000   3rd Qu.: 0.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :5.000   Max.   :14.0000
##      hscore          chcond1          chcond2          nondocco
##  Min.   : 0.000   Min.   :0.0000   Min.   :0.0000   Min.   : 0.0000
##  1st Qu.: 0.000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.0000
##  Median : 0.000   Median :0.0000   Median :0.0000   Median : 0.0000
##  Mean   : 1.218   Mean   :0.4031   Mean   :0.1166   Mean   : 0.2146
##  3rd Qu.: 2.000   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.: 0.0000
##  Max.   :12.000   Max.   :1.0000   Max.   :1.0000   Max.   :11.0000
##     hospadmi         hospdays         medicine        prescrib
##  Min.   :0.0000   Min.   : 0.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.0000   Median : 0.000   Median :1.000   Median :0.0000
##  Mean   :0.1736   Mean   : 1.334   Mean   :1.218   Mean   :0.8626
##  3rd Qu.:0.0000   3rd Qu.: 0.000   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :5.0000   Max.   :80.000   Max.   :8.000   Max.   :8.0000
##     nonpresc
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.3557
##  3rd Qu.:1.0000
##  Max.   :8.0000
```

Plots:

```
# Arrange the plots in the first list in a grid
do.call(grid.arrange, plot_list1)
```




```
# Arrange the plots in the second list in a grid
do.call(grid.arrange, plot_list2)
```



- Starting with the variable ‘sex’, the histogram would typically display two bars reflecting the count of each gender category in the dataset, with 0 representing male and 1 representing female participants. The statistical summary indicates a slight majority of females over males, as evidenced by the mean of 0.52, wich suggests that approximately 52% of individuals are female.
- The ‘age’ variable is normalized by dividing by 100, thus the range in the dataset is from 0.19 (representing 15-19 years old) to 0.72 (representing 72 years or older). The median age is 0.32 and the mean age is slightly higher at 0.40, suggesting a distribution that is slightly skewed towards older individuals. The histogram’s shape would likely show a decline in frequency as age increases, which is typical for a population with fewer older individuals.
- The variable ‘income’ represent annual income in Australian dollars, divided by 1000 and coded into ranges. The highest frequency is observed in values 0.25. This data provides insights into the economic diversity and distribution within the sample population.
- The variable ‘levyplus’ indicate whether individuals are covered by a private health insurance fund for a private patient in a public hospital. The total sum of ’1’s in the dataset is 2,298, which means that 2298 individuals out of the 5190 have this specific type of private health insurance coverage.
- The variable ‘freepoor’ is another binary categorical variable and it indicates whether individuals are covered by government healthcare due to low income, recent immigration status, or unemployment. The total sum of 1’s is 222, which suggests that 222 individuals out of the total 5190 (approximately 4.2%) in the dataset are covered by the government for the reasons mentioned.
- The variable ‘freepera’, binary categoricak variable, that indicates whether individuals are covered by the government for health care due to old-age or disability pension, or because they are invalid veterans or family members of deceased veterans. Out of the total dataset, 1,091 individuals are indicated as covered under this category, as shown by the sum of ’1’s (approximately 21.02%).
- The variable ‘illness’ is a discrete variable that represents the number of illnesses an individual had in the past two weeks. The mean number of illnesses reported is 1.432, indicating that on average, individuals reported having about one to two illnesses in the past two weeks. The distribution of this variable shows the most common values are 0 and 1, with 1,554 and 1,638 occurrences respectively, suggesting a significant portion of the population either had no illness or just one illness in the past two weeks. The number of reported illnesses decreases as the count increases, with the least common being 5 illnesses, reported by 236 individuals.
- The variable ‘actdays’ represents the number of days of reduced activity in the past two weeks due to illness or injury. The mean of 0.86 indicates the average number of days of reduced activity, suggesting that, on average, individuals experienced less than one day of reduced activity. In fact, most of the individuals (85.8%) reported no days of reduced activity, with the highest frequency (3.6%) observed for 14 days (the maximum value reported), so right skewed distributed.
- the variable ‘hscore’ is a measure of general health status using Goldberg’s method, where a higher score indicates poorer health. The mean score is 1.218, suggesting a moderately low average health score in the population, indicating generally good health.Most individuals

(58.3%) have a score of 0, indicating no health issues according to the questionnaire. The distribution is right skewed.
- The variable ‘chcond1’ indicates the presence of chronic conditions without limiting activity. The mean of 0.4031 indicates that approximately 40.31% of individuals have a chronic condition without activity limitation.
- The variable ‘chcond2’ is a binary categorical variable that indicates the presence of chronic conditions with activity limitation. The total sum of ’1’s in the dataset is 605, meaning that 605 individuals reported having chronic conditions that limit their activity.The mean of 0.1166 shows that about 11.66% of the dataset’s individuals have chronic conditions that limit their activities.
- The variable ‘nondocco’ measures the number of consultations with non-doctor health professionals. The mean of 0.2146 suggests that, on average, there are very few consultations with non-doctor health professionals.The data is heavily skewed towards 0, with 90.9% of the sample reporting no consultations, and very few individuals reporting more than one consultation.
- The variable ‘hospadmi’ represents the number of hospital admissions within the past 12 months. The majority of individuals (86.5%) reported no hospital admissions in the past year. A smaller proportion, 10.8%, had one hospital admission, and even fewer had two or more admissions.
- The variable ‘hospdays’ reflects the number of nights spent in a hospital or similar institution during the most recent admission. A significant majority of individuals (86.5%) reported no nights spent in a hospital in the past 12 months. Smaller proportions reported staying 1 to 80 nights, with the highest frequencies observed for shorter stays (1 to 2 nights).
- The variable ‘medicine’ quantifies the total number of prescribed and nonprescribed medications used in the past two days. A significant proportion of individuals (42.9%) reported not using any medication in the past two days. The frequencies gradually decrease as the number of medications increases, with 26.8% using one medication and smaller proportions using two or more. The mean of 1.218 indicates that on average, individuals used just over one medication in the past two days.
- The variable ‘prescrib’ refers to the total number of prescribed medications used in the past 2 days ranging from 0 to 8. A significant proportion of the dataset, 59.4%, reported not using any prescribed medication in the past 2 days. The frequency decreases as the number of medications increases, with fewer individuals reporting the use of a higher number of prescribed medications. The mean of 0.8626 indicates that, on average, individuals used less than one prescribed medication in the past 2 days.
- The variable nonpresc represents the total number of nonprescribed medications used in the past 2 days ranging from 0 to 8 (like ‘prescrib’). A large majority of the dataset, 73.5%, reported not using any nonprescribed medication in the past 2 days. A notable proportion (20.3%) reported using one nonprescribed medication. The frequency decreases significantly with an increase in the number of nonprescribed medications used. The mean of 0.3557 suggests that, on average, individuals used around one third of a nonprescribed medication in the past 2 days.

Considerations

The analysis of the variables and the response variable ‘doctorco’ in the dataset has revealed some key points:

The data provides a comprehensive view of healthcare utilization, with a range of variables from health status to service usage. The binary variables (like sex, levyplus, freepoor and chcond1/2) allow for clear distinctions in the population and are easy to model. The other dicrete variables like illness and actdays capture count data which can be informative for healthcare demand modelling. Also the response variable’s distribution is quite clear, showing a majority of individuals not consulting a doctor, which may reflect real-world scenarios and, for example, support planning and policy development.

But, there are also negative points on it:

The right-skewed distribution of ‘doctorco’ suggests that most of the data points are zeros, which could lead to zero-inflation issues in modelling. Also the use of discretized continuous variables like age and income may lose some information compared to truly continuous measurements. Then, the presence of long-tail distributions in variables like hospdays and medicine may require specific transformation or specific modelling techniques like Poisson or negative binomial regression to handle overdispersion. Binary and discrete variables with limited variance might not contribute as much to a predictive model’s complexity or accuracy.

In order to produce a good report, it was necessary to pay close attention to the use of variables and their modelling. The dataset is a typical use case of real data, which one may come across every day. The analysis, in all its steps, is of fundamental importance to fully understand the application context.

We can say also that other variables could be included in the analysis, such as the use of other types of health services, the use of emergency services, the use of preventive health services, among others. The inclusion of these variables could provide a more comprehensive view of the use of health services and the factors that influence this use. For example, if we mind about people who had ictus, heart attack, diabetes, among others, and how these diseases influence the use of health services. In that case, the doctor consultation in the last year could be more influenced by the presence of these diseases. In fact, not all diseases are the same, and some diseases require more frequent consultations than others, but only in rare cases individuals got specific consultations for these diseases in the last 2 weeks. Same things for the use of emergency services, preventive health services, and other types of health services, or individuals who had broken bones, or other types of accidents. In that case, is not required to have doctor consultations in the range 2 weeks, maybe the consultations are required every 3 weeks, or every month.

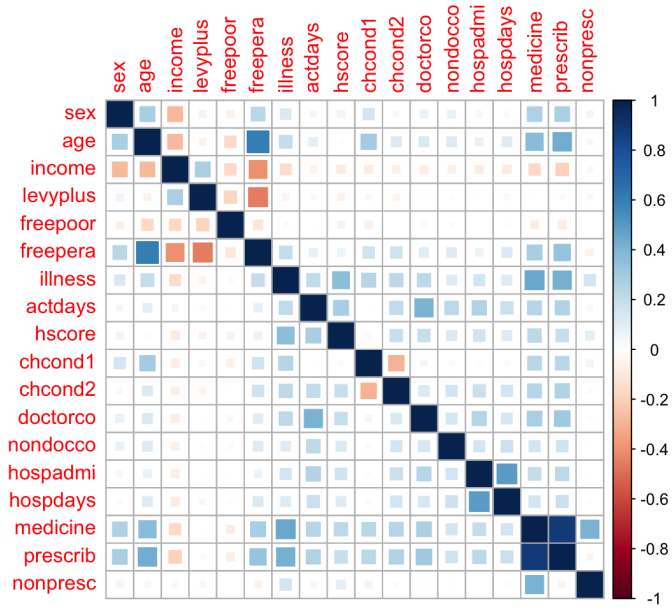
Furthermore, both R and Python were used to perform the analysis and the results were consistent. The use of both languages was important in order to show the flexibility of the analysis and the importance of the use of both languages in data analysis. Again, the use and combinations of these different methods generally reflect the know-how of the contributors to this analysis project.

Bivariate Analysis: ‘doctorco’ vs other variables

Correlation Matrix

This is the Correlation Matrix:

```
# Calculate the correlation matrix
correlation_matrix <- cor(df, method = "pearson")
#correlation_matrix

corrplot(correlation_matrix, method = "square",
         diag = TRUE)
```



From a visually interpretation of the correlation matrix:

Key Findings:

- Healthcare Utilization: Older individuals with more illnesses, hospitalizations, and medication use tend to have more frequent doctor consultations. This highlights the increased healthcare needs associated with age and health conditions.
- Income Disparity and Healthcare: Individuals with lower income see doctors less frequently, even in the presence of health problems. This suggests potential barriers to healthcare access for lower-income populations.
- Illness, Hospitalization, and Length of Stay: Older individuals with more health issues and frequent hospitalizations tend to have longer hospital stays. This points to the complexity of managing complex health conditions.
- Medication Use: Unsurprisingly, those with more health issues, doctor visits, and hospitalizations receive more prescriptions. This underscores the role of medications in managing multiple health conditions.
- Non-Prescribed Medication Use: Older individuals with more health problems and doctor consultations also use over-the-counter or non-prescribed medications more frequently.

Important Reminders:

- Correlation vs. Causation: The findings highlight associations, not direct cause-and-effect relationships. Other factors could be influencing these patterns.
- Addressing Healthcare Disparities: The income-related findings warrant further investigation into potential socioeconomic barriers that can limit access to care for lower-income individuals.

'doctorco' vs. 'sex'

Starting this bivariate analysis with variable 'sex':

```
describe(df$sex)
```

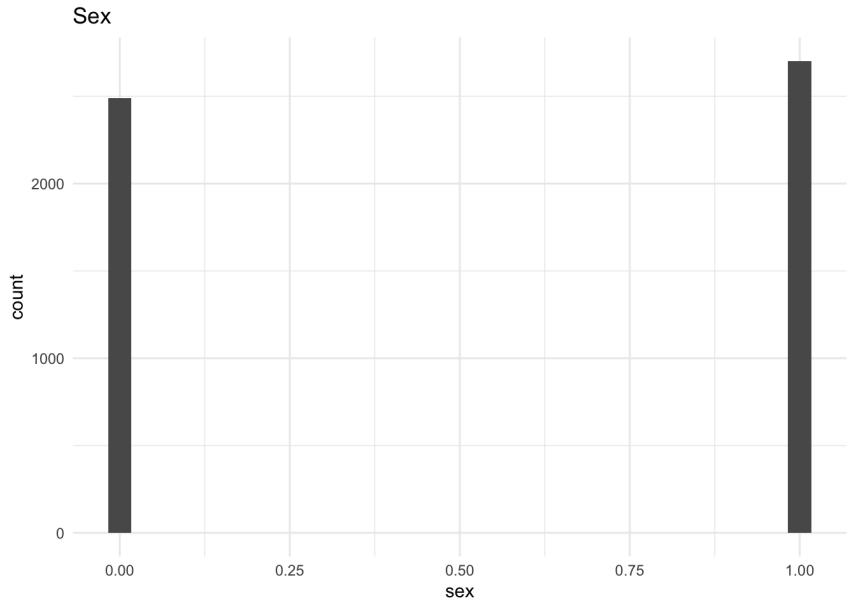
```
## df$sex
##        n  missing distinct     Info      Sum     Mean      Gmd
##     5190        0        2    0.749     2702   0.5206   0.4992
```

```
# value counts
df %>% count(sex)
```

sex <dbl>	n <int>
0	2488
1	2702

2 rows

```
# plot of sex
ggplot(df, aes(x=sex)) +
  geom_histogram(position="dodge", bins=30) +
  ggtitle("Sex") +
  theme_minimal()
```




```
# Create a cross-tabulation
sex_doctorco_table <- table(df$sex, df$doctorco)

# Chi-square test of independence
chisq.test(sex_doctorco_table)
```

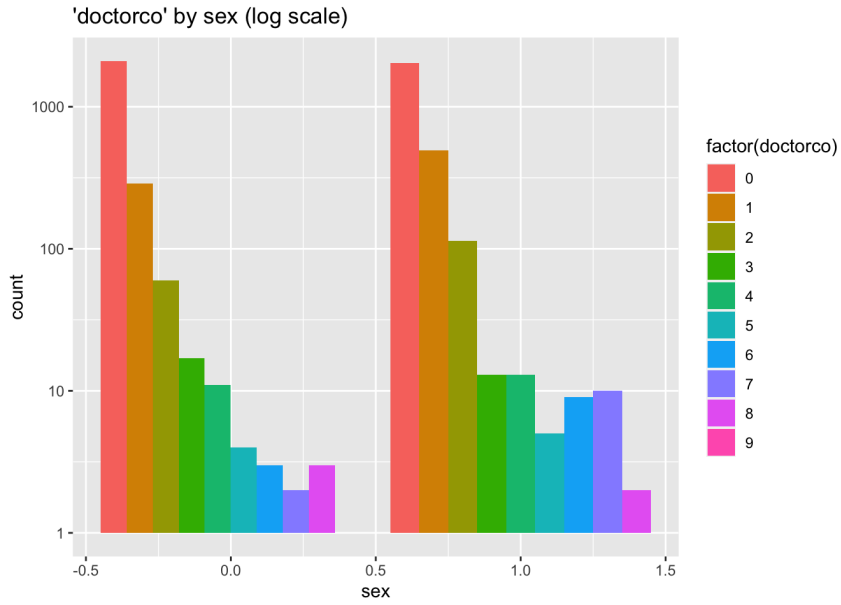
```
## Warning in chisq.test(sex_doctorco_table): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  sex_doctorco_table
## X-squared = 73.455, df = 9, p-value = 3.188e-12
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ sex, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ sex, data = df, init.theta = 0.3938035314,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.44251    0.05217 -27.652  < 2e-16 ***
## sex          0.42627    0.06844   6.229  4.7e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.3938) family taken to be 1)
##
##     Null deviance: 3017.0  on 5189  degrees of freedom
## Residual deviance: 2977.9  on 5188  degrees of freedom
## AIC: 7139.2
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.3938
##           Std. Err.:  0.0280
##
##  2 x log-likelihood:  -7133.1990
```

```
# For visualization
ggplot(df, aes(x = sex, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by sex (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



From this analysis, the Chi-squared test result (p-value = 3.188e-12) suggests a strong association between sex and the number of doctor consultations. This means that the frequency of doctor consultations is not independent of the sex of the individual. The negative binomial regression results indicate a significant relationship between sex and doctor consultations. The coefficient for 'sex' (0.42627) is positive and significant ($p < 1e-9$), which suggests that one sex (typically coded as 1) has a higher rate of doctor consultations than the other (typically coded as 0), when all other factors are held constant. From the plot, The y-axis is on a logarithmic scale, which helps to display a wide range of values for count data where some counts (like 0) are much more frequent than others. There is a visible difference in the distribution of the number of doctor consultations between the two sexes. This is consistent with the results from the negative binomial regression.

'doctorco' vs. 'age'

Bivariate analysis with variable 'age':

```
describe(df$age)
```

```
## df$age
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     5190        0       12    0.978   0.4064   0.2258     0.19     0.19
##      .25      .50      .75      .90      .95
##     0.22     0.32     0.62     0.72     0.72
##
## Value       0.19  0.22  0.27  0.32  0.37  0.42  0.47  0.52  0.57  0.62  0.67
## Frequency    752  1213   523   301   146   126   181   222   273   316   315
## Proportion 0.145 0.234 0.101 0.058 0.028 0.024 0.035 0.043 0.053 0.061 0.061
##
## Value       0.72
## Frequency    822
## Proportion 0.158
##
## For the frequency table, variable is rounded to the nearest 0
```

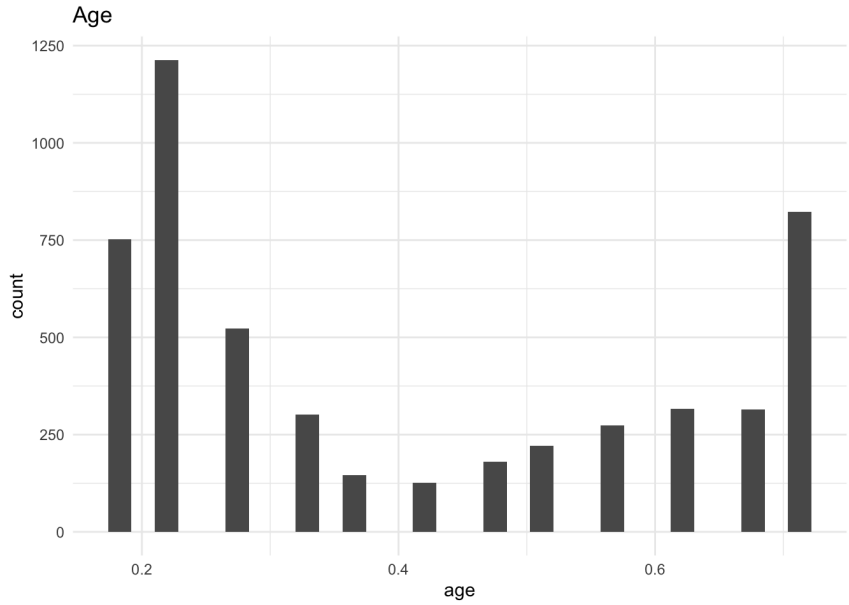
```
# value counts
df %>% count(age)
```

age <dbl>	n <int>
0.19	752
0.22	1213
0.27	523
0.32	301
0.37	146
0.42	126
0.47	181
0.52	222
0.57	273
0.62	316

1-10 of 12 rows

Previous **1** 2 Next

```
# plot of age
ggplot(df, aes(x=age)) +
  geom_histogram(position="dodge", bins=30) +
  ggtitle("Age") +
  theme_minimal()
```




```
# Create a cross-tabulation
age_doctorco_table <- table(df$age, df$doctorco)

# Chi-square test of independence
chisq.test(age_doctorco_table)
```

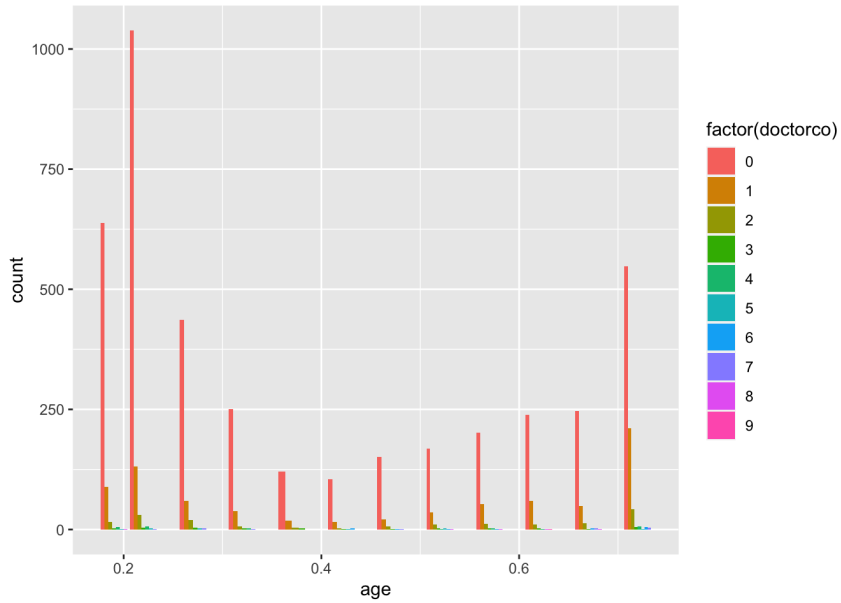
```
## Warning in chisq.test(age_doctorco_table): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  age_doctorco_table
## X-squared = 249.99, df = 99, p-value = 4.865e-15
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ age, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ age, data = df, init.theta = 0.4184902741,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.87971    0.07885  -23.84   <2e-16 ***
## age          1.54940    0.16040    9.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.4185) family taken to be 1)
##
##     Null deviance: 3084.8  on 5189  degrees of freedom
## Residual deviance: 2990.1  on 5188  degrees of freedom
## AIC: 7085.4
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.4185
##           Std. Err.:  0.0304
##
##  2 x log-likelihood:  -7079.4040
```

```
# For visualization
ggplot(df, aes(x = age, fill = factor(doctorco))) +
  #scale_y_continuous(trans = "log10") +
  geom_bar(position = "dodge", stat = "count")
```



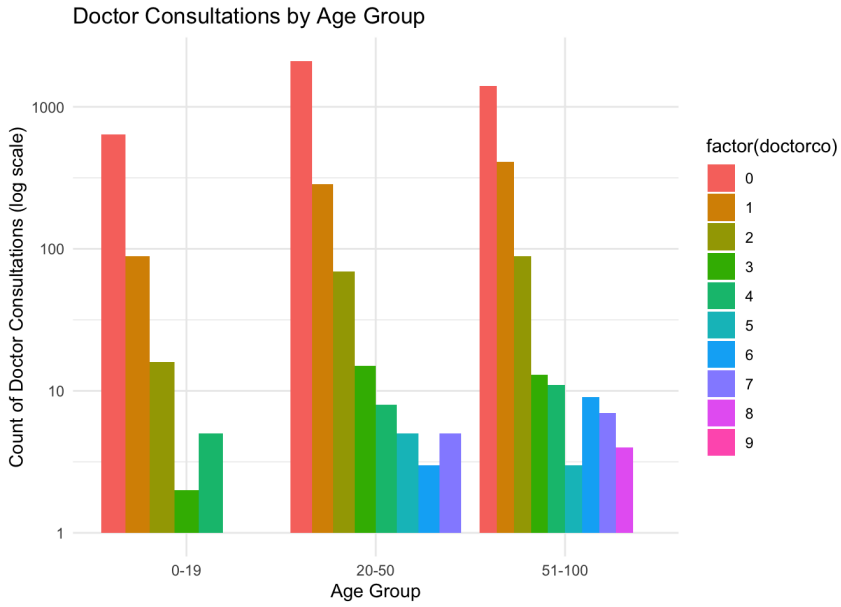
Applied Negative Binomial regression to study the interaction between response variable and ‘age’. The regression output suggests that age is a strong predictor of the number of doctor consultations. There is a significant positive relationship, meaning that as the age category increases, the expected number of doctor consultations also increases. This could be due to various factors like higher health issues with increasing age or greater health awareness.

To better analyze the variable ‘age’, we grouped it 3 categories: 15-19, 20-50 and greater than 51.

```
# First, create a new factor variable for age groups
df$age_group <- cut(df$age,
                    breaks = c(-Inf, 0.19, 0.50, 0.72),
                    labels = c("0-19", "20-50", "51-100"),
                    right = TRUE) # This includes the right endpoint in the interval

# Convert to a factor to ensure proper ordering in plots and models
df$age_group <- factor(df$age_group, levels = c("0-19", "20-50", "51-100"))


# Create a cross-tabulation table for the new age_group variable and 'doctorco'
age_group_doctorco_table <- table(df$age_group, df$doctorco)

# Chi-squared test
chisq_results <- chisq.test(age_group_doctorco_table)
```

```
## Warning in chisq.test(age_group_doctorco_table): Chi-squared approximation may
## be incorrect
```

```
# Negative binomial regression
nb_model <- glm.nb(doctorco ~ age_group, data = df)
nb_model_summary <- summary(nb_model)

# Plot the count of doctor consultations by new age_group
ggplot(df, aes(x = age_group, fill = factor(doctorco))) +
  geom_bar(position = "dodge", stat = "count") +
  scale_y_continuous(trans = "log10") +
  labs(x = "Age Group", y = "Count of Doctor Consultations (log scale)", title = "Doctor Consultations by Age Gro
up") +
  theme_minimal()
```




```
# Display results
print(chisq_results)
```

```
##
##  Pearson's Chi-squared test
##
## data:  age_group_doctorco_table
## X-squared = 132.48, df = 18, p-value < 2.2e-16
```

```
print(nb_model_summary)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ age_group, data = df, init.theta = 0.4155941835,
##     link = log)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.54756    0.09721 -15.920  < 2e-16 ***
## age_group20-50   0.08537    0.11022   0.774    0.439
## age_group51-100  0.69322    0.10905   6.357 2.06e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.4156) family taken to be 1)
##
##     Null deviance: 3077  on 5189  degrees of freedom
## Residual deviance: 2989  on 5187  degrees of freedom
## AIC: 7093.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.4156
##           Std. Err.:  0.0301
##
##  2 x log-likelihood:  -7085.7910
```

The theta value of the negative binomial distribution (0.4156) indicates overdispersion in the count data, which justifies the use of the negative binomial model over simpler Poisson regression. The analysis demonstrates that age is a predictor of the number of doctor consultations, with the “51-100” age group showing a significantly higher frequency of consultations compared to the “0-19” age group. This suggests that older individuals are more likely to have a higher number of doctor consultations. The “20-50” age group does not show a significant difference from the “0-19” age group in terms of the number of doctor consultations. Then, we provided different solutions on categorization of ‘age’ variable.

‘doctorco’ vs. ‘income’

Bivariate analysis with variable ‘income’:

```
describe(df$income)
```

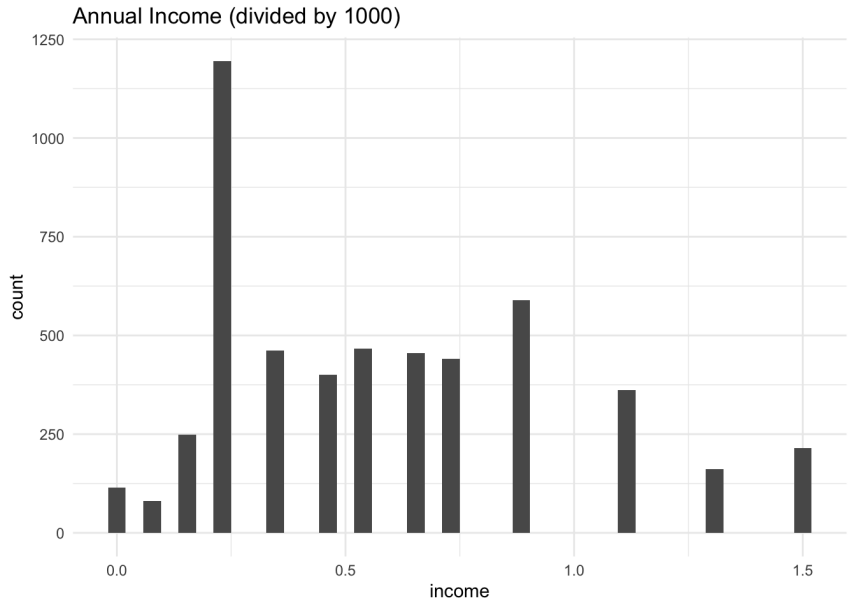
```
## df$income
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     5190        0       14    0.983   0.5832   0.4085     0.15     0.25
##      .25      .50      .75      .90      .95
##     0.25     0.55     0.90     1.10     1.30
##
## Value       0.00  0.01  0.06  0.15  0.25  0.35  0.45  0.55  0.65  0.75  0.90
## Frequency     79    35    80   249  1195   462   400   467   455   441   589
## Proportion 0.015 0.007 0.015 0.048 0.230 0.089 0.077 0.090 0.088 0.085 0.113
##
## Value       1.10  1.30  1.50
## Frequency    361   162   215
## Proportion 0.070 0.031 0.041
##
## For the frequency table, variable is rounded to the nearest 0
```

```
# value counts
df %>% count(income)
```

income <dbl>	n <int>
0.00	79
0.01	35
0.06	80
0.15	249
0.25	1195
0.35	462
0.45	400
0.55	467
0.65	455
0.75	441

1-10 of 14 rows Previous **1** 2 Next

```
# plot of age
ggplot(df, aes(x=income)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("Annual Income (divided by 1000)") +
  theme_minimal()
```




```
# Create a cross-tabulation
income_doctorco_table <- table(df$income, df$doctorco)

# Chi-square test of independence
chisq.test(income_doctorco_table)
```

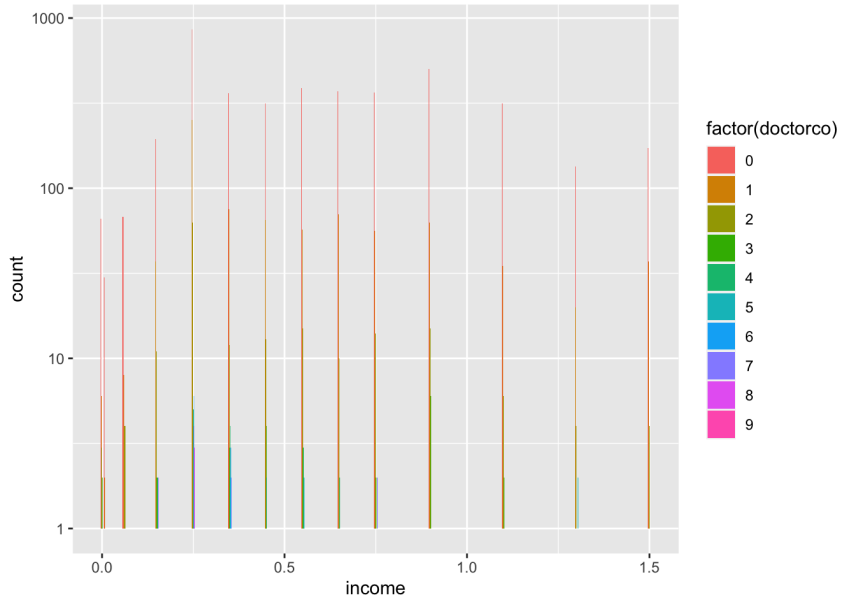
```
## Warning in chisq.test(income_doctorco_table): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  income_doctorco_table
## X-squared = 212.54, df = 117, p-value = 1.609e-07
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ income, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ income, data = df, init.theta = 0.3939394939,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.88496    0.06198  -14.28  < 2e-16 ***
## income      -0.57530    0.09636   -5.97 2.37e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.3939) family taken to be 1)
##
##     Null deviance: 3017.4  on 5189  degrees of freedom
## Residual deviance: 2979.5  on 5188  degrees of freedom
## AIC: 7140.5
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.3939
##           Std. Err.:  0.0281
##
##  2 x log-likelihood:  -7134.4660
```

```
# For visualization
ggplot(df, aes(x = income, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  geom_bar(position = "dodge", stat = "count")
```



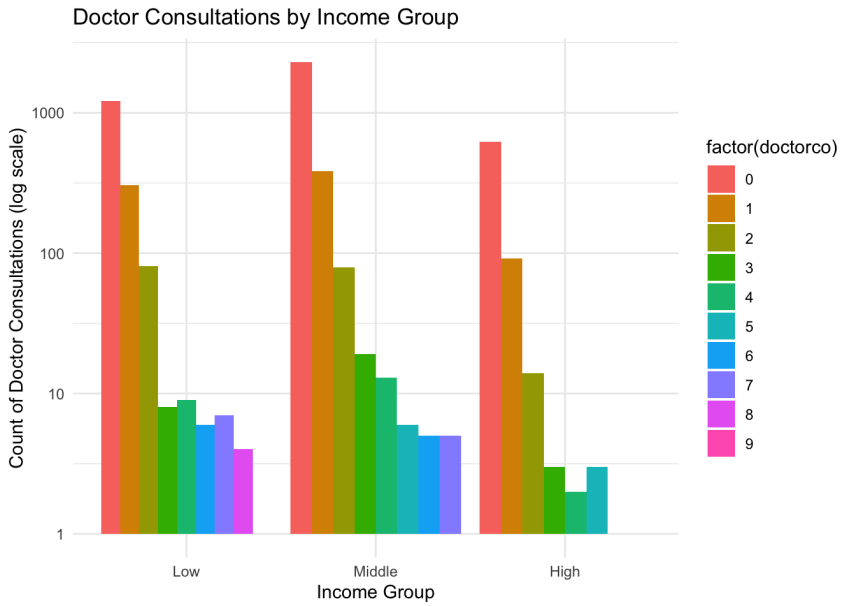
The analysis indicates that there is a significant negative relationship between income and the number of doctor consultations. Individuals with higher incomes tend to have fewer doctor consultations. This could be attributed to various factors, including but not limited to, better overall health, access to preventive care, or different health service utilization patterns among higher-income individuals.

For a better visualization and extra analysis of the variable, looking at the summary of ‘income’, the 25th, 50th (median), and 75th percentiles are given as 0.25, 0.55, and 0.90 (times 1000), respectively. These values can serve as natural dividing points for categories: - Low Income: Less than or equal to 0.25 (≤ $250) - Middle Income: Greater than 0.25 but less than or equal to 0.90 (>$250 and ≤ $900) - High Income: Greater than 0.90 (>$900)

```
df <- df %>%
  mutate(income_group = case_when(
    income <= 0.25 ~ "Low",
    income > 0.25 & income <= 0.90 ~ "Middle",
    income > 0.90 ~ "High"
  ))

df$income_group <- factor(df$income_group, levels = c("Low", "Middle", "High"))

# Create a cross-tabulation table for the new age_group variable and 'doctorco'
income_group_doctorco_table <- table(df$income_group, df$doctorco)

# Chi-squared test
chisq_results <- chisq.test(income_group_doctorco_table)
```

```
## Warning in chisq.test(income_group_doctorco_table): Chi-squared approximation
## may be incorrect
```

```
# Negative binomial regression
nb_model <- glm.nb(doctorco ~ income_group, data = df)
nb_model_summary <- summary(nb_model)

# Plot the count of doctor consultations by new age_group
ggplot(df, aes(x = income_group, fill = factor(doctorco))) +
  geom_bar(position = "dodge", stat = "count") +
  scale_y_continuous(trans = "log10") +
  labs(x = "Income Group", y = "Count of Doctor Consultations (log scale)", title = "Doctor Consultations by Inco
me Group") +
  theme_minimal()
```




```
# Display results
print(chisq_results)
```

```
##
##  Pearson's Chi-squared test
##
## data:  income_group_doctorco_table
## X-squared = 71.581, df = 18, p-value = 2.436e-08
```

```
print(nb_model_summary)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ income_group, data = df, init.theta = 0.3952392556,
##     link = log)
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.92119    0.05548 -16.603  < 2e-16 ***
## income_groupMiddle -0.40377    0.07290  -5.539 3.05e-08 ***
## income_groupHigh   -0.57077    0.11194  -5.099 3.42e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.3952) family taken to be 1)
##
##     Null deviance: 3021.1  on 5189  degrees of freedom
## Residual deviance: 2979.6  on 5187  degrees of freedom
## AIC: 7139
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.3952
##           Std. Err.:  0.0282
##
##  2 x log-likelihood:  -7130.9580
```

As stated before, the analysis and visualization provide evidence that income level is inversely associated with the number of doctor consultations. Individuals in lower income groups tend to consult doctors more frequently than those in higher income groups. This could reflect differences in health status, access to preventive care, or health service utilization patterns among income groups.

‘doctorco’ vs. ‘levyplus’

Bivariate analysis with variable ‘levyplus’:

```
describe(df$levyplus)
```

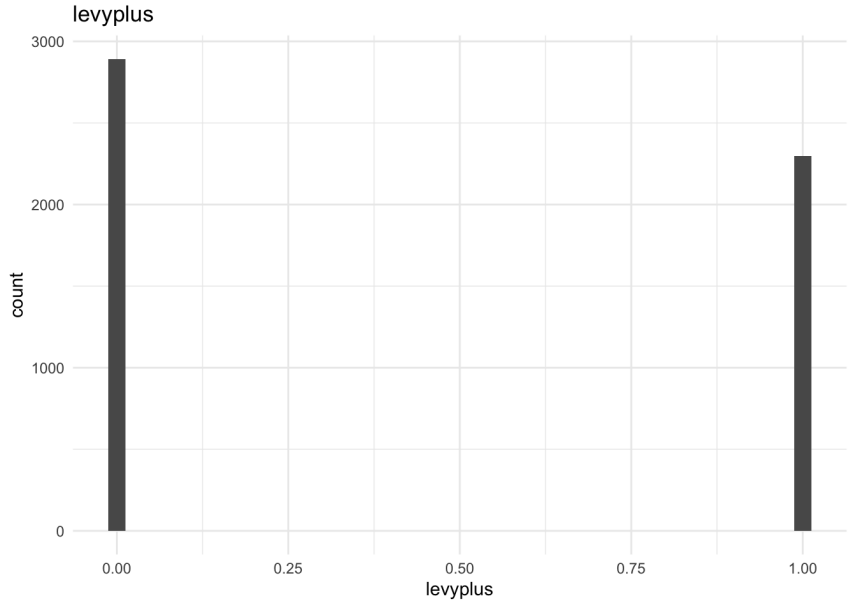
```
## df$levyplus
##        n  missing distinct     Info      Sum     Mean      Gmd
##     5190        0        2     0.74     2298   0.4428   0.4935
```

```
# value counts
df %>% count(levyplus)
```

levyplus <dbl>	n <int>
0	2892
1	2298

2 rows

```
# plot of levyplus
ggplot(df, aes(x=levyplus)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("levyplus") +
  theme_minimal()
```



```r
# Create a cross-tabulation
levyplus_doctorco_table <- table(df$levyplus, df$doctorco)

# Chi-square test of independence
chisq.test(levyplus_doctorco_table)
```

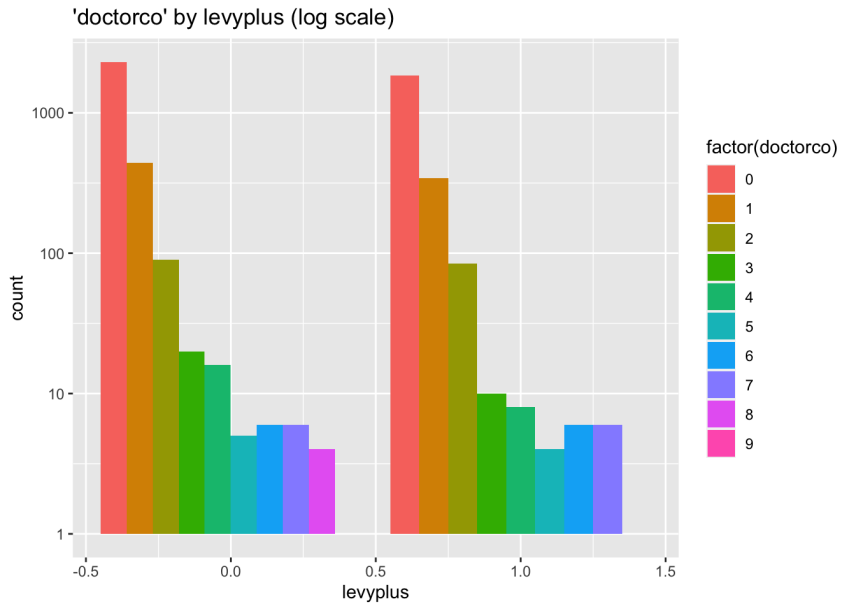
```
## Warning in chisq.test(levyplus_doctorco_table): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  levyplus_doctorco_table
## X-squared = 6.219, df = 9, p-value = 0.7178
```

```r
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ levyplus, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ levyplus, data = df, init.theta = 0.3777086009,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.17961    0.04517 -26.115   <2e-16 ***
## levyplus    -0.04252    0.06833  -0.622    0.534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.3777) family taken to be 1)
##
##     Null deviance: 2970.4  on 5189  degrees of freedom
## Residual deviance: 2970.0  on 5188  degrees of freedom
## AIC: 7177.6
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.3777
##           Std. Err.:  0.0266
##
##  2 x log-likelihood:  -7171.5960
```

```r
# For visualization
ggplot(df, aes(x = levyplus, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by levyplus (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



From Negative Binomial regression, the coefficient for 'levyplus' (-0.04252) is not statistically significant (p = 0.534), suggesting that having a private levy does not have a statistically significant effect on the number of doctor consultations. The analysis suggests that the presence or absence of a private levy does not significantly impact the frequency of doctor consultations. This could mean that other factors, such as the severity of health issues, access to public healthcare, or personal preferences, might play a more significant role in determining how often individuals seek medical advice.

‘doctorco’ vs. ‘freepoor’

Bivariate analysis with variable ‘freepoor’:

```
describe(df$freepoor)
```

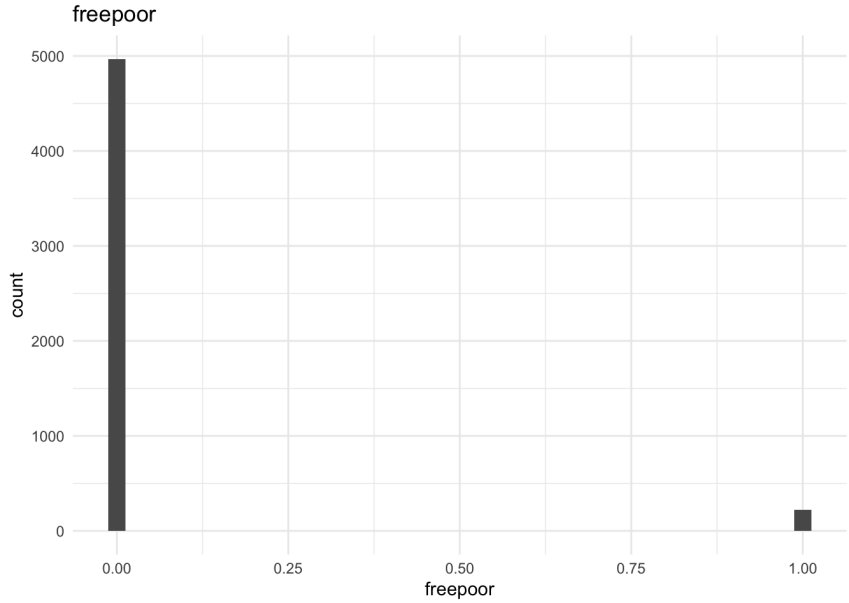
```
## df$freepoor
##        n  missing distinct     Info      Sum     Mean      Gmd
##     5190        0        2    0.123      222  0.04277  0.08191
```

```
# value counts
df %>% count(freepoor)
```

freepoor <dbl>	n <int>
0	4968
1	222

2 rows

```
# plot of freepoor
ggplot(df, aes(x=freepoor)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("freepoor") +
  theme_minimal()
```




```
# Create a cross-tabulation
freepoor_doctorco_table <- table(df$freepoor, df$doctorco)

# Chi-square test of independence
chisq.test(freepoor_doctorco_table)
```

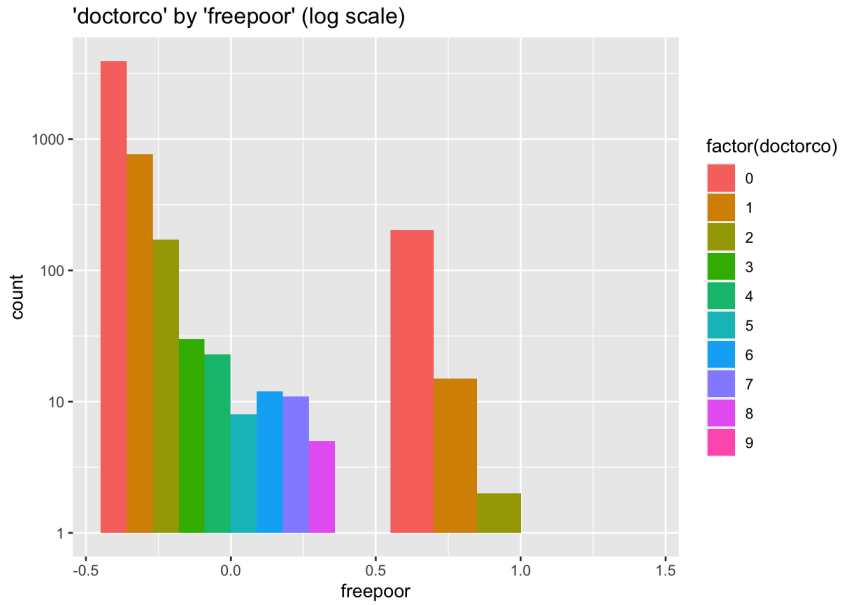
```
## Warning in chisq.test(freepoor_doctorco_table): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  freepoor_doctorco_table
## X-squared = 22.091, df = 9, p-value = 0.008594
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ freepoor, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ freepoor, data = df, init.theta = 0.3818477976,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.17710    0.03436 -34.262  < 2e-16 ***
## freepoor    -0.67023    0.20383  -3.288  0.00101 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.3818) family taken to be 1)
##
##     Null deviance: 2982.6  on 5189  degrees of freedom
## Residual deviance: 2970.9  on 5188  degrees of freedom
## AIC: 7166.4
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.3818
##           Std. Err.:  0.0269
##
##  2 x log-likelihood:  -7160.3720
```

```
# For visualization
ggplot(df, aes(x = freepoor, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'freepoor' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



Given the context that the ‘freepoor’ binary variable indicates government coverage for individuals who may have low income, are recent immigrants, or are unemployed, and that only 222 out of 5190 individuals in the dataset are covered by the government, the graph and statistical output together suggest that among the individuals in the dataset, those who are covered by the government due to their socioeconomic status tend to use fewer doctor consultations. This could be a positive indicator of the effectiveness of the coverage or, conversely, a sign that there are still unmet healthcare needs or barriers within this group.

‘doctorco’ vs. ‘freepera’

Bivariate analysis with variable ‘freepera’:

```
describe(df$freepera)
```

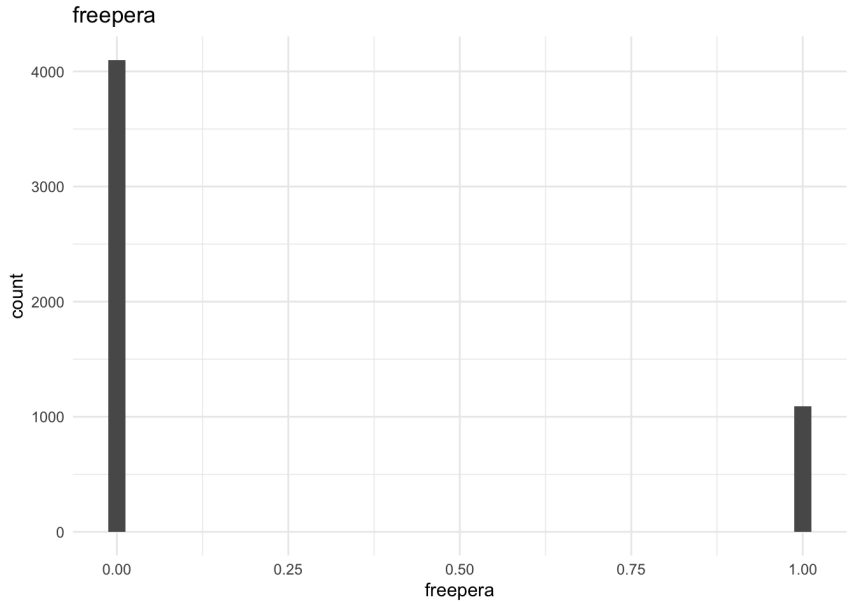
```
## df$freepera
##        n  missing distinct     Info      Sum     Mean      Gmd
##     5190        0        2    0.498     1091   0.2102   0.3321
```

```
# value counts
df %>% count(freepera)
```

freepera	n
<dbl>	<int>
0	4099
1	1091

2 rows

```
# plot of freepera
ggplot(df, aes(x=freepera)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("freepera") +
  theme_minimal()
```




```
# Create a cross-tabulation
freepera_doctorco_table <- table(df$freepera, df$doctorco)

# Chi-square test of independence
chisq.test(freepera_doctorco_table)
```

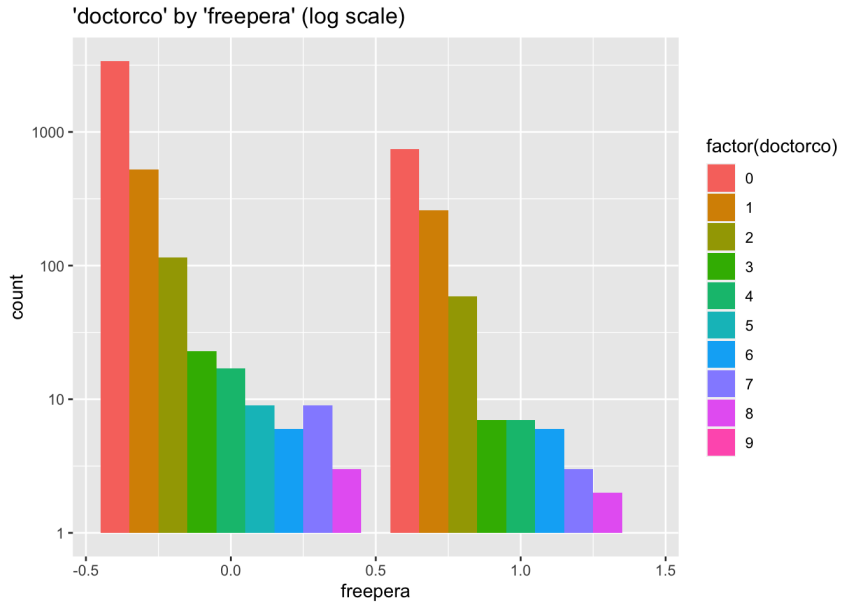
```
## Warning in chisq.test(freepera_doctorco_table): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  freepera_doctorco_table
## X-squared = 124.84, df = 9, p-value < 2.2e-16
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ freepera, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ freepera, data = df, init.theta = 0.4048503616,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.35531    0.03935 -34.440  < 2e-16 ***
## freepera     0.59291    0.07601   7.801 6.17e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.4049) family taken to be 1)
##
##     Null deviance: 3047.8  on 5189  degrees of freedom
## Residual deviance: 2986.7  on 5188  degrees of freedom
## AIC: 7117.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.4049
##           Std. Err.:  0.0291
##
##  2 x log-likelihood:  -7111.8230
```

```
# For visualization
ggplot(df, aes(x = freepera, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'freepera' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



the analysis for the binary variable ‘freepera’, which represents government health coverage due to old-age or disability pension, status as invalid veterans, or as family members of deceased veterans. Out of 5190 individuals in the dataset, 1091 are covered by the government under these criteria. The plot shows that individuals covered by ‘freepera’ tend to have more doctor consultations across all frequencies than those not covered when viewed on a logarithmic scale. This could indicate that individuals who are older, disabled, or associated with veterans may have more health concerns or a greater need for medical services. The analysis suggests that ‘freepera’ coverage is linked with increased utilization of healthcare services, as indicated by a higher number of doctor consultations. This could reflect higher healthcare needs or more robust access to healthcare services among the covered group.

‘doctorco’ vs. ‘illness’

Bivariate analysis with variable ‘illness’:

```
describe(df$illness)
```

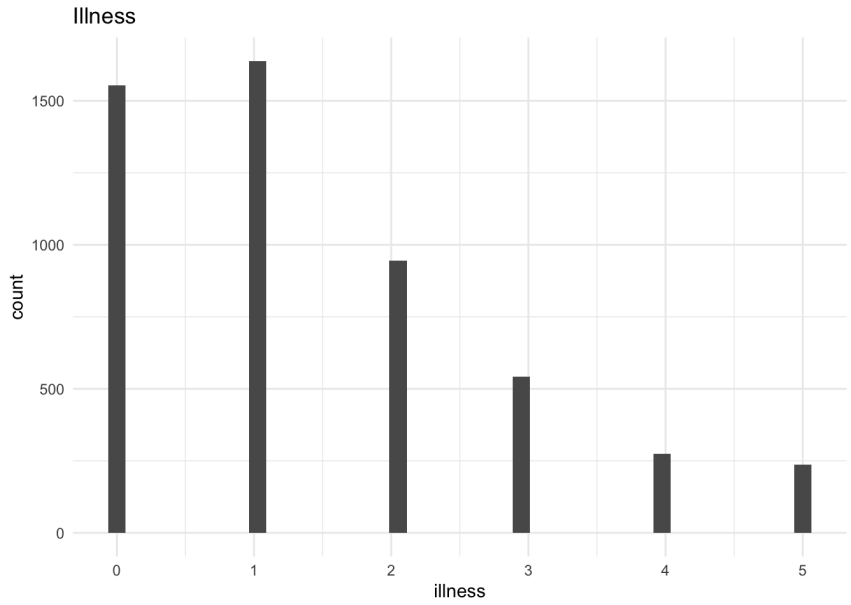
```
## df$illness
##        n  missing distinct     Info     Mean      Gmd
##     5190        0        6    0.934    1.432    1.481
##
## Value          0     1     2     3     4     5
## Frequency   1554  1638   946   542   274   236
## Proportion 0.299 0.316 0.182 0.104 0.053 0.045
##
## For the frequency table, variable is rounded to the nearest 0
```

```
# value counts
df %>% count(illness)
```

illness <dbl>	n <int>
0	1554
1	1638
2	946
3	542
4	274
5	236

6 rows

```
# plot of illness
ggplot(df, aes(x=illness)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("Illness") +
  theme_minimal()
```




```
# Create a cross-tabulation
illness_doctorco_table <- table(df$illness, df$doctorco)

# Chi-square test of independence
chisq.test(illness_doctorco_table)
```

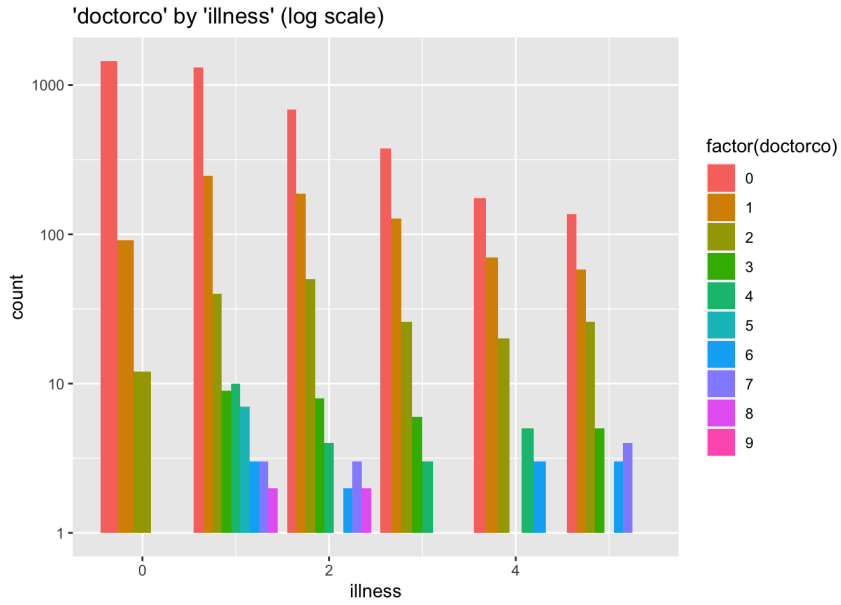
```
## Warning in chisq.test(illness_doctorco_table): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  illness_doctorco_table
## X-squared = 462.23, df = 45, p-value < 2.2e-16
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ illness, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ illness, data = df, init.theta = 0.5080141764,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.88950    0.05291  -35.71   <2e-16 ***
## illness      0.38069    0.02161   17.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.508) family taken to be 1)
##
##     Null deviance: 3299.1  on 5189  degrees of freedom
## Residual deviance: 2998.0  on 5188  degrees of freedom
## AIC: 6893.3
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.5080
##           Std. Err.:  0.0389
##
##  2 x log-likelihood:  -6887.3490
```

```
# For visualization
ggplot(df, aes(x = illness, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'illness' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



The plot demonstrates that as the number of illnesses increases, there is a corresponding increase in the count of doctor consultations across all frequencies. This suggests that individuals with more health issues are more likely to seek medical advice. The analysis strongly suggests that there is a direct relationship between the burden of illness and healthcare utilization, as measured by the number of doctor consultations. This is in line with expectations, as individuals with a greater number of health complaints are likely to require more medical attention.

'doctorco' vs. 'actdays'

Bivariate analysis with variable 'actdays':

```
describe(df$actdays)
```

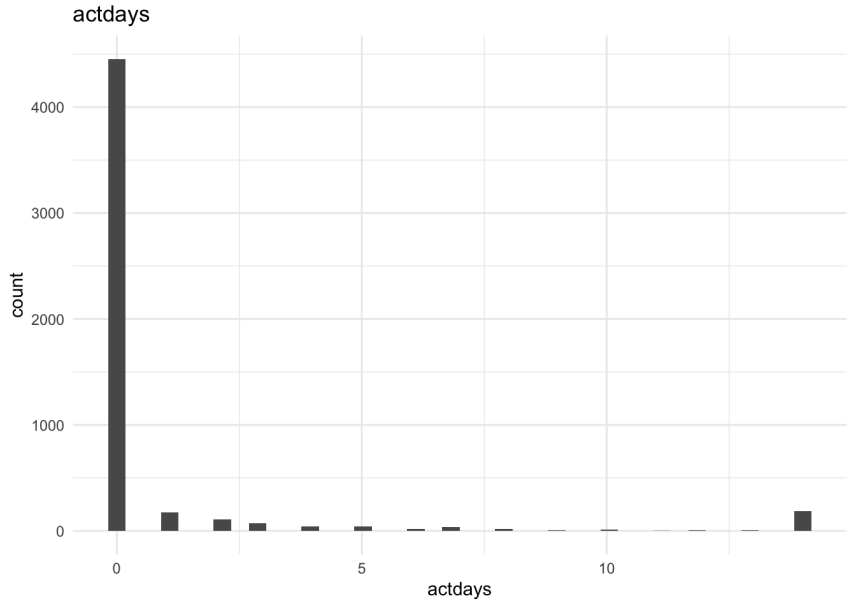
```
## df$actdays
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     5190        0       15    0.368   0.8618    1.592        0        0
##      .25      .50      .75      .90      .95
##        0        0        0        2        7
##
## Value          0     1     2     3     4     5     6     7     8     9    10
## Frequency   4454   177   108    74    45    40    17    38    17     7    12
## Proportion 0.858 0.034 0.021 0.014 0.009 0.008 0.003 0.007 0.003 0.001 0.002
##
## Value         11    12    13    14
## Frequency      2     6     5   188
## Proportion 0.000 0.001 0.001 0.036
##
## For the frequency table, variable is rounded to the nearest 0
```

```
# value counts
df %>% count(actdays)
```

actdays <dbl>	n <int>
0	4454
1	177
2	108
3	74
4	45
5	40
6	17
7	38
8	17
9	7

1-10 of 15 rows Previous **1** 2 Next

```
# plot of age
ggplot(df, aes(x=actdays)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("actdays") +
  theme_minimal()
```




```
# Create a cross-tabulation
actdays_doctorco_table <- table(df$actdays, df$doctorco)

# Chi-square test of independence
chisq.test(actdays_doctorco_table)
```

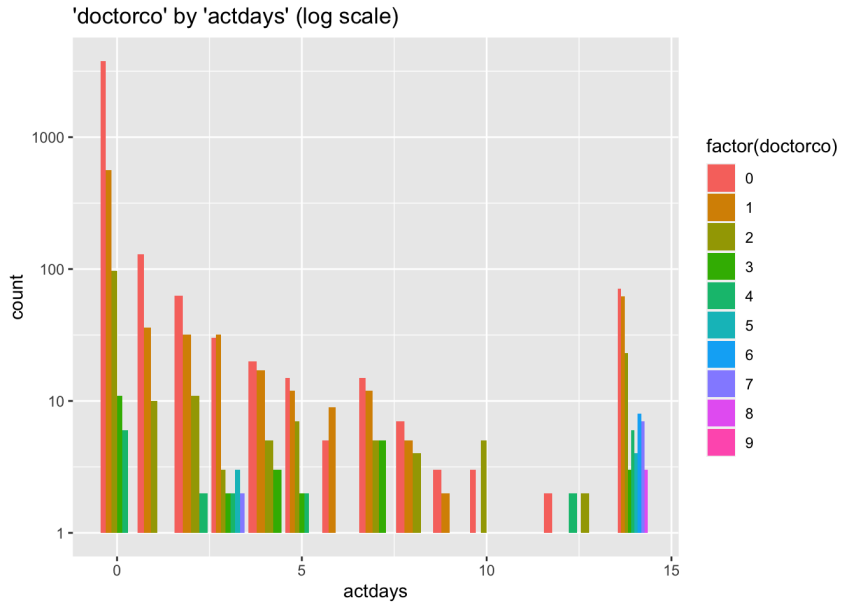
```
## Warning in chisq.test(actdays_doctorco_table): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  actdays_doctorco_table
## X-squared = 2214.8, df = 126, p-value < 2.2e-16
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ actdays, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ actdays, data = df, init.theta = 0.7958500766,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.558673   0.035051  -44.47   <2e-16 ***
## actdays      0.174052   0.006851   25.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.7959) family taken to be 1)
##
##     Null deviance: 3775.2  on 5189  degrees of freedom
## Residual deviance: 3148.5  on 5188  degrees of freedom
## AIC: 6642.3
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.7959
##           Std. Err.:  0.0741
##
##  2 x log-likelihood:  -6636.3150
```

```
# For visualization
ggplot(df, aes(x = actdays, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'actdays' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



The plot shows that individuals with more days of reduced activity have a higher frequency of doctor consultations across all consultation counts. This trend is consistent with the expectation that individuals who have been more significantly affected by illness or injury would require more medical attention. The analysis indicates a direct and significant relationship between the burden of reduced activity due to illness or injury and the utilization of healthcare services. This is a logical outcome, as one would expect that the more an illness or injury impacts daily activities, the more likely an individual is to seek medical advice.

'doctorco' vs. 'hscore'

Bivariate analysis with variable 'hscore':

```
describe(df$hscore)
```

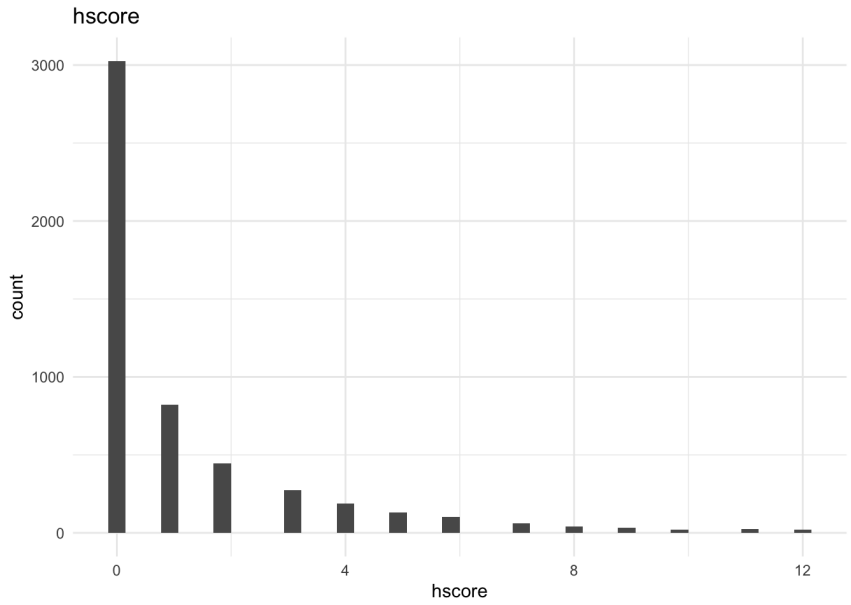
```
## df$hscore
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     5190        0       13    0.797    1.218     1.84        0        0
##      .25      .50      .75      .90      .95
##        0        0        2        4        6
##
## Value          0     1     2     3     4     5     6     7     8     9    10
## Frequency   3026   823   446   273   187   132   104    61    42    32    21
## Proportion 0.583 0.159 0.086 0.053 0.036 0.025 0.020 0.012 0.008 0.006 0.004
##
## Value         11    12
## Frequency     24    19
## Proportion 0.005 0.004
##
## For the frequency table, variable is rounded to the nearest 0
```

```
# value counts
df %>% count(hscore)
```

hscore <dbl>	n <int>
0	3026
1	823
2	446
3	273
4	187
5	132
6	104
7	61
8	42
9	32

1-10 of 13 rows Previous **1** 2 Next

```
# plot of age
ggplot(df, aes(x=hscore)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("hscore") +
  theme_minimal()
```




```
# Create a cross-tabulation
hscore_doctorco_table <- table(df$hscore, df$doctorco)

# Chi-square test of independence
chisq.test(hscore_doctorco_table)
```

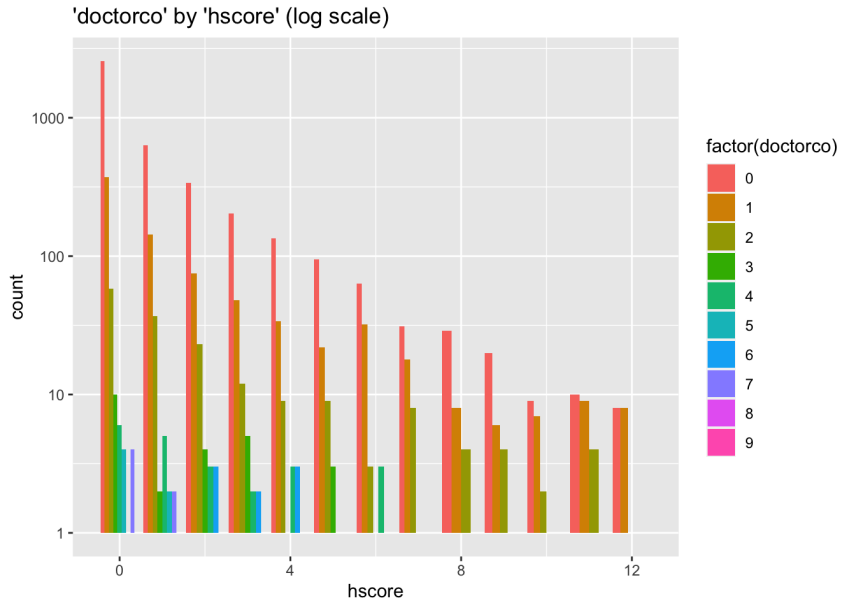
```
## Warning in chisq.test(hscore_doctorco_table): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  hscore_doctorco_table
## X-squared = 655.16, df = 108, p-value < 2.2e-16
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ hscore, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ hscore, data = df, init.theta = 0.458488221,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.48738    0.03980  -37.37   <2e-16 ***
## hscore       0.17131    0.01284   13.34   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.4585) family taken to be 1)
##
##     Null deviance: 3186.1  on 5189  degrees of freedom
## Residual deviance: 3013.9  on 5188  degrees of freedom
## AIC: 7013
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.4585
##           Std. Err.:  0.0344
##
##  2 x log-likelihood:  -7007.0450
```

```
# For visualization
ggplot(df, aes(x = hscore, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'hscore' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



The plot shows that as the health score increases, there is a corresponding increase in the count of doctor consultations across all frequencies. This suggests that individuals with higher health questionnaire scores, indicating poorer health, are more likely to seek medical advice. The analysis confirms a direct and significant relationship between the health status as measured by the 'hscore' and healthcare utilization. This is consistent with the expectation that individuals who report worse health would require more medical attention.

'doctorco' vs. 'chcond1'

Bivariate analysis with variable 'chcond1':

```
describe(df$chcond1)
```

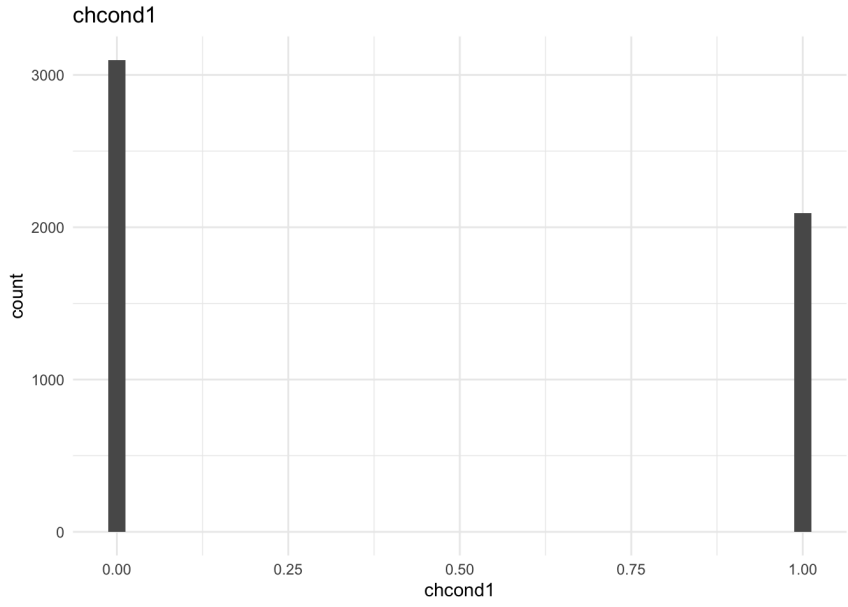
```
## df$chcond1
##        n  missing distinct     Info      Sum     Mean      Gmd
##     5190        0        2    0.722     2092   0.4031   0.4813
```

```
# value counts
df %>% count(chcond1)
```

chcond1 <dbl>	n <int>
0	3098
1	2092

2 rows

```
# plot of chcond1
ggplot(df, aes(x=chcond1)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("chcond1") +
  theme_minimal()
```



```r
# Create a cross-tabulation
chcond1_doctorco_table <- table(df$chcond1, df$doctorco)

# Chi-square test of independence
chisq.test(chcond1_doctorco_table)
```

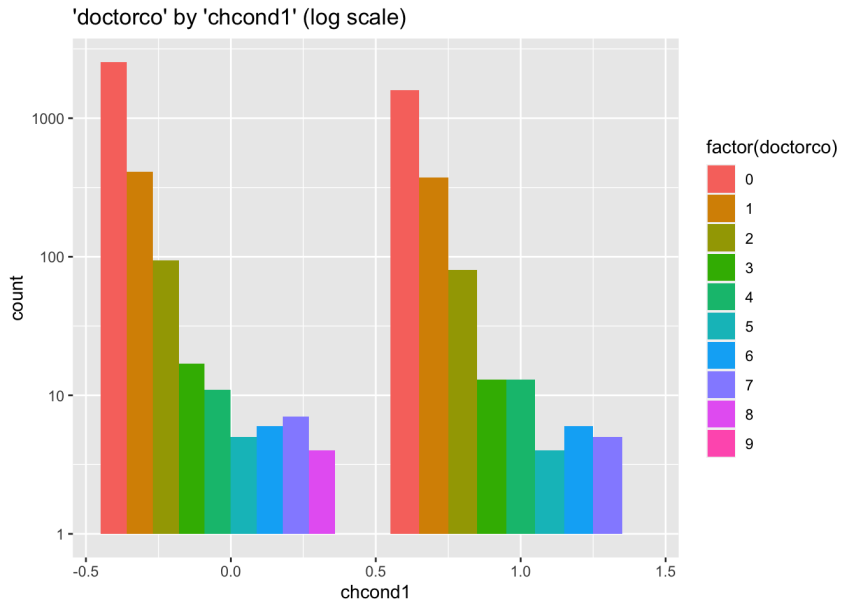
```
## Warning in chisq.test(chcond1_doctorco_table): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  chcond1_doctorco_table
## X-squared = 29.404, df = 9, p-value = 0.0005538
```

```r
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ chcond1, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ chcond1, data = df, init.theta = 0.3826860688,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.30155    0.04505 -28.890  < 2e-16 ***
## chcond1      0.23908    0.06828   3.502 0.000463 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.3827) family taken to be 1)
##
##     Null deviance: 2985.0  on 5189  degrees of freedom
## Residual deviance: 2972.8  on 5188  degrees of freedom
## AIC: 7165.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.3827
##           Std. Err.:  0.0270
##
##  2 x log-likelihood:  -7159.7520
```

```r
# For visualization
ggplot(df, aes(x = chcond1, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'chcond1' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



The positive coefficient for ‘chcond1’ (0.23908 with p = 0.000463) suggests that individuals with chronic conditions have a higher expected count of doctor consultations than those without, even if these conditions do not limit their activity. The plot shows that individuals with chronic conditions that do not limit activity still tend to have more doctor consultations across most frequencies compared to those without chronic conditions. This suggests that while these conditions may not limit daily activities, they still require medical attention. The analysis indicates a

significant relationship between the presence of non-limiting chronic conditions and healthcare utilization, as measured by the number of doctor consultations. This suggests that even when chronic conditions do not directly limit activity, they still pose health management needs that lead to increased engagement with healthcare services.

‘doctorco’ vs. ‘chcond2’

Bivariate analysis with variable ‘chcond2’:

```
describe(df$chcond2)
```

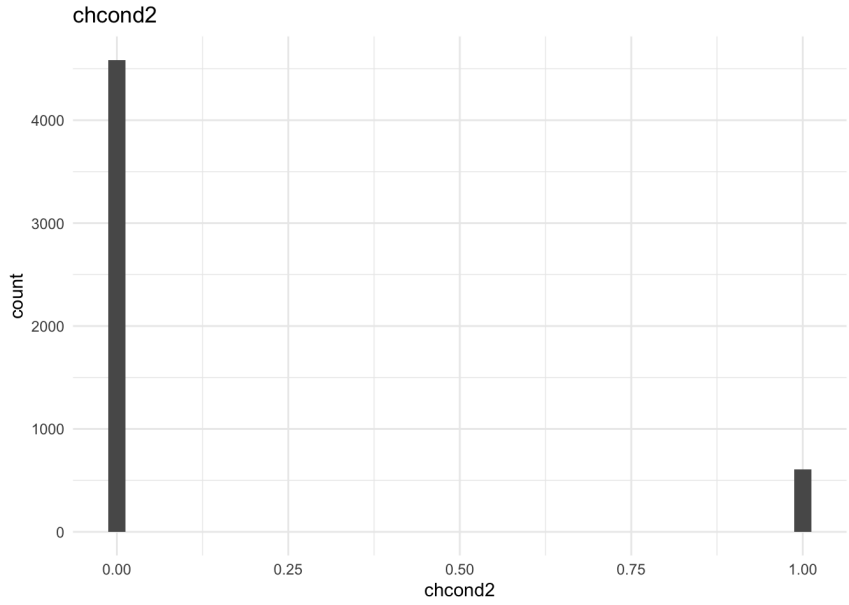
```
## df$chcond2
##        n  missing distinct     Info      Sum     Mean      Gmd
##     5190        0        2    0.309      605   0.1166    0.206
```

```
# value counts
df %>% count(chcond2)
```

chcond2	n
<dbl>	<int>
0	4585
1	605

2 rows

```
# plot of chcond2
ggplot(df, aes(x=chcond2)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("chcond2") +
  theme_minimal()
```




```
# Create a cross-tabulation
chcond2_doctorco_table <- table(df$chcond2, df$doctorco)

# Chi-square test of independence
chisq.test(chcond2_doctorco_table)
```

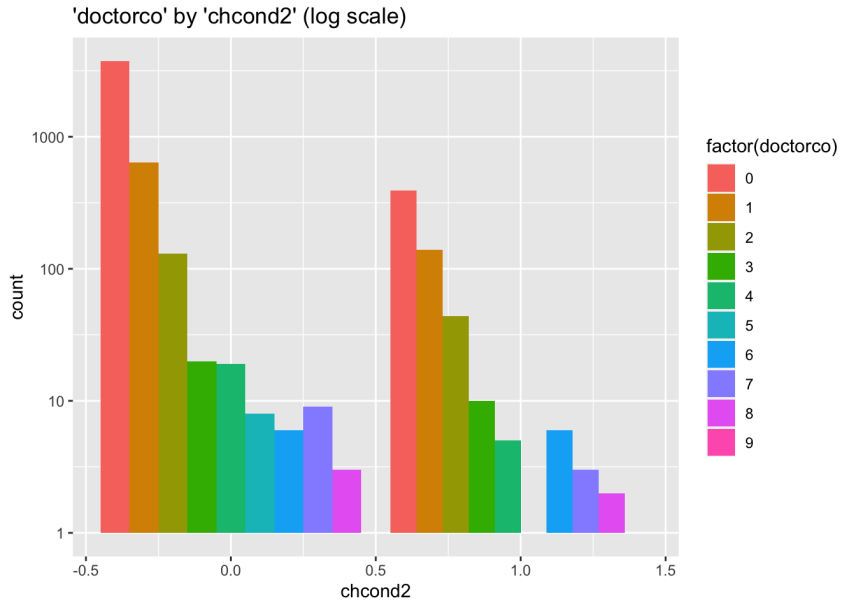
```
## Warning in chisq.test(chcond2_doctorco_table): Chi-squared approximation may be
## incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  chcond2_doctorco_table
## X-squared = 126.23, df = 9, p-value < 2.2e-16
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ chcond2, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ chcond2, data = df, init.theta = 0.419543538,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.33964    0.03678 -36.427   <2e-16 ***
## chcond2      0.83430    0.08962   9.309   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.4195) family taken to be 1)
##
##     Null deviance: 3087.6  on 5189  degrees of freedom
## Residual deviance: 2999.2  on 5188  degrees of freedom
## AIC: 7091.8
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.4195
##           Std. Err.:  0.0306
##
##  2 x log-likelihood:  -7085.7740
```

```
# For visualization
ggplot(df, aes(x = chcond2, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'chcond2' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



The positive coefficient for ‘chcond2’ (0.83430 with p < 2e-16) indicates that individuals with chronic conditions that limit their activity have a higher expected count of doctor consultations compared to those without such conditions. The plot demonstrates that individuals with chronic conditions that limit their activity have more doctor consultations across all frequencies compared to those without such conditions. This is evidenced by the higher bars for ‘1’ on the ‘chcond2’ scale. The analysis indicates a significant relationship between having limiting chronic conditions and increased healthcare utilization. Individuals with chronic conditions that impact their daily activities are likely to require more medical attention, as reflected in the number of doctor consultations.

‘doctorco’ vs. ‘nondocco’

Bivariate analysis with variable ‘nondocco’:

```
describe(df$nondocco)
```

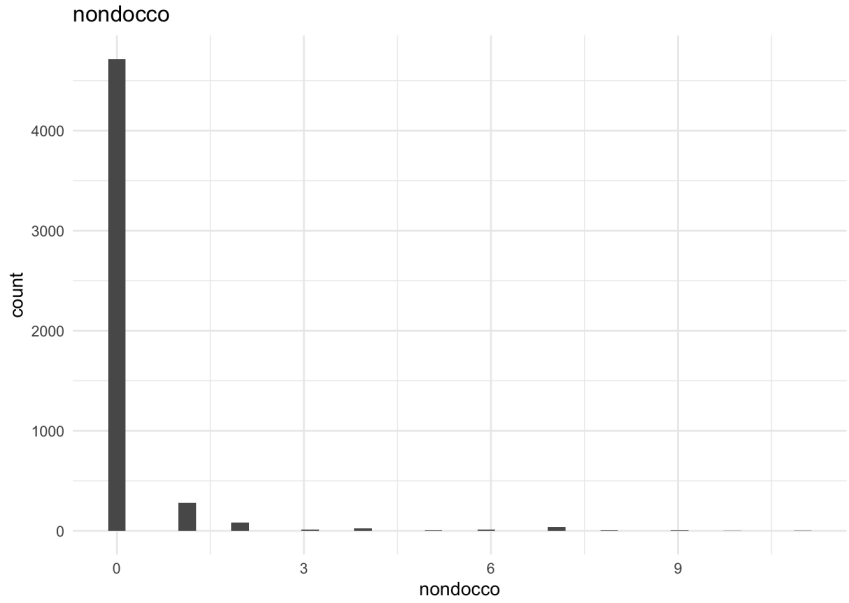
```
## df$nondocco
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     5190        0       12     0.25   0.2146   0.4072        0        0
##      .25      .50      .75      .90      .95
##        0        0        0        0        1
##
## Value          0     1     2     3     4     5     6     7     8     9    10
## Frequency   4716   278    84    14    26     6    10    37     6     8     2
## Proportion 0.909 0.054 0.016 0.003 0.005 0.001 0.002 0.007 0.001 0.002 0.000
##
## Value         11
## Frequency      3
## Proportion 0.001
##
## For the frequency table, variable is rounded to the nearest 0
```

```
# value counts
df %>% count(nondocco)
```

nondocco <dbl>	n <int>
0	4716
1	278
2	84
3	14
4	26
5	6
6	10
7	37
8	6
9	8

1-10 of 12 rows Previous **1** 2 Next

```
# plot of nondocco
ggplot(df, aes(x=nondocco)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("nondocco") +
  theme_minimal()
```




```
# Create a cross-tabulation
nondocco_doctorco_table <- table(df$nondocco, df$doctorco)

# Chi-square test of independence
chisq.test(nondocco_doctorco_table)
```

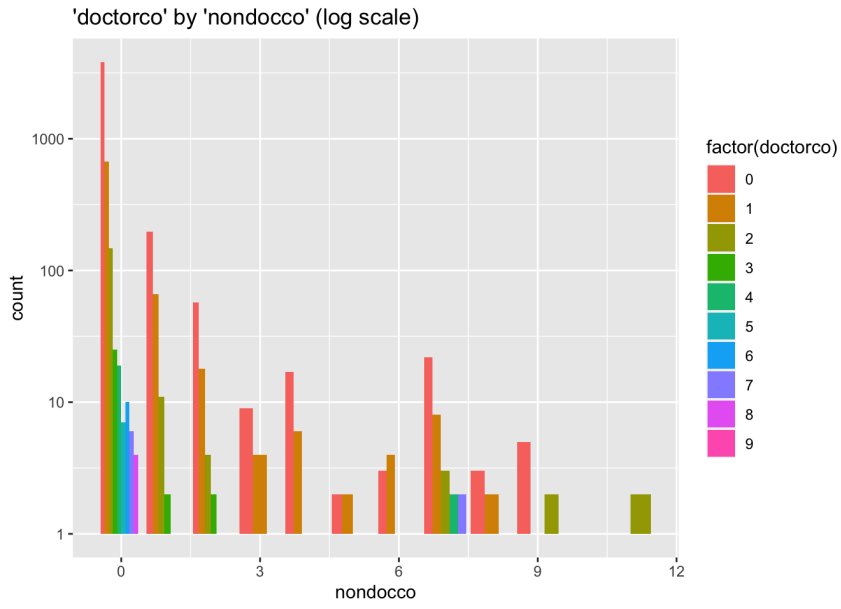
```
## Warning in chisq.test(nondocco_doctorco_table): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  nondocco_doctorco_table
## X-squared = 695.72, df = 99, p-value < 2.2e-16
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ nondocco, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ nondocco, data = df, init.theta = 0.4126326943,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.27634    0.03466 -36.822  < 2e-16 ***
## nondocco     0.21568    0.02651   8.134 4.14e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.4126) family taken to be 1)
##
##     Null deviance: 3069.1  on 5189  degrees of freedom
## Residual deviance: 2999.2  on 5188  degrees of freedom
## AIC: 7109.6
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.4126
##           Std. Err.:  0.0300
##
##  2 x log-likelihood:  -7103.6440
```

```
# For visualization
ggplot(df, aes(x = nondocco, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'nondocco' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



The plot shows that individuals with a higher number of non-doctor health professional consultations tend to have a higher frequency of doctor consultations. This pattern is visible across the logarithmic scale, indicating that engagements with various health professionals are associated with increased doctor visits. The analysis suggests that individuals who consult non-doctor health professionals also tend to have more doctor consultations. This could be due to a variety of reasons, such as the complexity of their health needs, which require multidisciplinary care involving different health professionals, or it could be an indicator of a more proactive approach to health management among these individuals.

‘doctorco’ vs. ‘hospadmi’

Bivariate analysis with variable ‘hospadmi’:

```
describe(df$hospadmi)
```

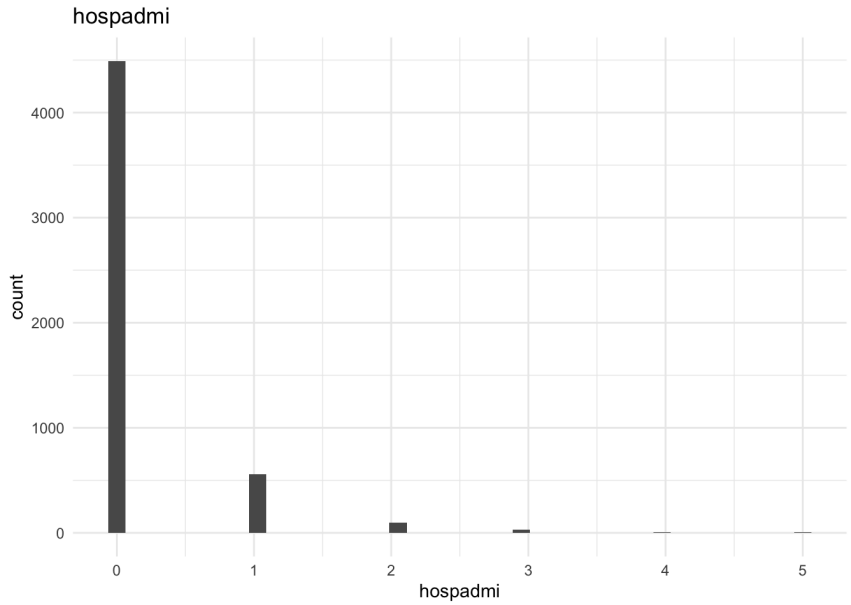
```
## df$hospadmi
##        n  missing distinct     Info     Mean      Gmd
##     5190        0        6    0.351   0.1736   0.3094
##
## Value          0     1     2     3     4     5
## Frequency   4491   561    96    28     6     8
## Proportion 0.865 0.108 0.018 0.005 0.001 0.002
##
## For the frequency table, variable is rounded to the nearest 0
```

```
# value counts
df %>% count(hospadmi)
```

hospadmi <dbl>	n <int>
0	4491
1	561
2	96
3	28
4	6
5	8

6 rows

```
# plot of hospadmi
ggplot(df, aes(x=hospadmi)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("hospadmi") +
  theme_minimal()
```




```
# Create a cross-tabulation
hospadmi_doctorco_table <- table(df$hospadmi, df$doctorco)

# Chi-square test of independence
chisq.test(hospadmi_doctorco_table)
```

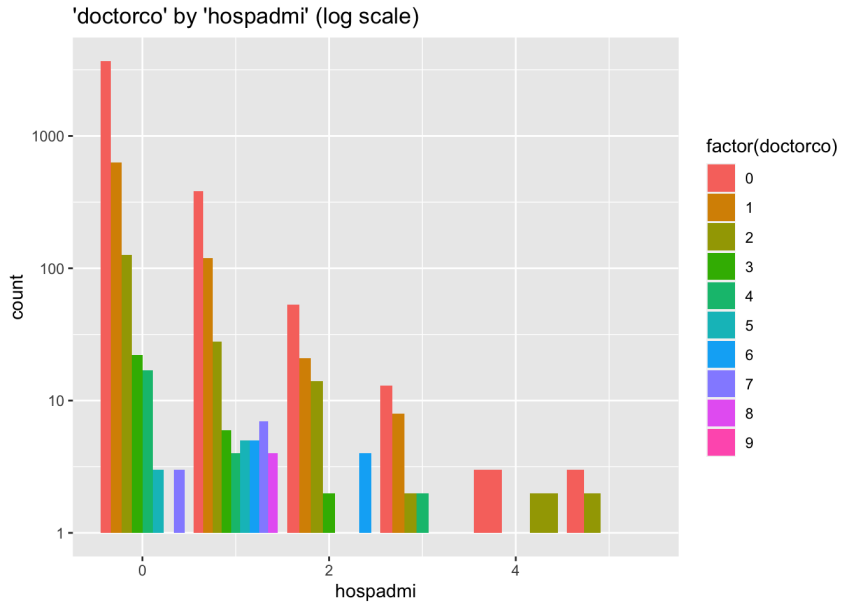
```
## Warning in chisq.test(hospadmi_doctorco_table): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  hospadmi_doctorco_table
## X-squared = 1138, df = 45, p-value < 2.2e-16
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ hospadmi, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ hospadmi, data = df, init.theta = 0.4859389294,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.40976    0.03586  -39.31   <2e-16 ***
## hospadmi     0.70440    0.04878   14.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.4859) family taken to be 1)
##
##     Null deviance: 3250.3  on 5189  degrees of freedom
## Residual deviance: 3041.0  on 5188  degrees of freedom
## AIC: 6980.7
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.4859
##           Std. Err.:  0.0375
##
##  2 x log-likelihood:  -6974.7180
```

```
# For visualization
ggplot(df, aes(x = hospadmi, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'hospadmi' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



The positive coefficient for ‘hospadmi’ (0.70440 with p < 2e-16) indicates that as the number of hospital admissions increases, there is a corresponding increase in the expected count of doctor consultations. The plot show the bars remain short at higher ‘hospadmi’ values, it might indicate that while fewer individuals have many hospital admissions, they do not necessarily have a corresponding increase in doctor consultations. The analysis indicates a significant relationship between hospital admissions and healthcare utilization, with increased admissions associated with a higher frequency of doctor consultations. This could be because hospital admissions are often for more serious health concerns, which would likely lead to more follow-up care and consultations.

‘doctorco’ vs. ‘hospdays’

Bivariate analysis with variable ‘hospdays’:

```
describe(df$hospdays)
```

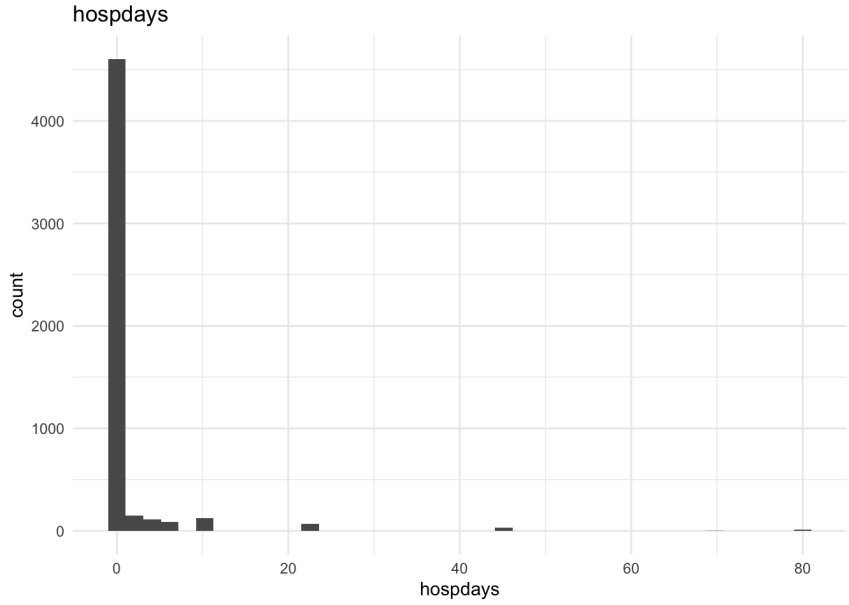
```
## df$hospdays
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     5190        0       13    0.352    1.334    2.517        0        0
##      .25      .50      .75      .90      .95
##        0        0        0        2        7
##
## Value          0     1     2     3     4     5     6     7    11    22    45
## Frequency   4491   113    89    58    56    55    30    59   125    70    30
## Proportion 0.865 0.022 0.017 0.011 0.011 0.011 0.006 0.011 0.024 0.013 0.006
##
## Value         70    80
## Frequency      2    12
## Proportion 0.000 0.002
##
## For the frequency table, variable is rounded to the nearest 0
```

```
# value counts
df %>% count(hospdays)
```

hospdays <dbl>	n <int>
0	4491
1	113
2	89
3	58
4	56
5	55
6	30
7	59
11	125
22	70

1-10 of 13 rows Previous **1** 2 Next

```
# plot of hospdays
ggplot(df, aes(x=hospdays)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("hospdays") +
  theme_minimal()
```




```
# Create a cross-tabulation
hospdays_doctorco_table <- table(df$hospdays, df$doctorco)

# Chi-square test of independence
chisq.test(hospdays_doctorco_table)
```

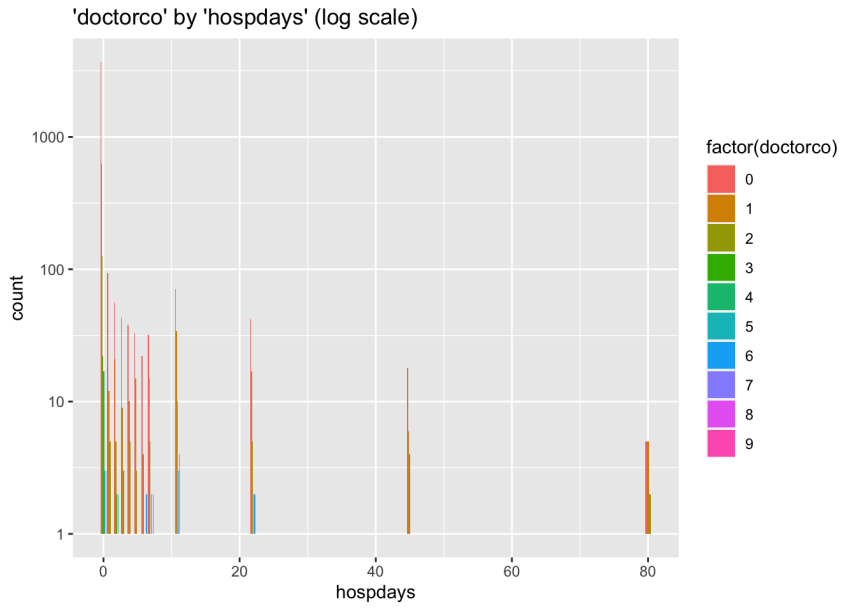
```
## Warning in chisq.test(hospdays_doctorco_table): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  hospdays_doctorco_table
## X-squared = 555.82, df = 108, p-value < 2.2e-16
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ hospdays, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ hospdays, data = df, init.theta = 0.4064760097,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.302242   0.034847  -37.37   <2e-16 ***
## hospdays     0.048617   0.003927   12.38   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.4065) family taken to be 1)
##
##     Null deviance: 3052.3  on 5189  degrees of freedom
## Residual deviance: 2965.3  on 5188  degrees of freedom
## AIC: 7092
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.4065
##           Std. Err.:  0.0286
##
##  2 x log-likelihood:  -7086.0380
```

```
# For visualization
ggplot(df, aes(x = hospdays, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'hospdays' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



The plot likely shows that most individuals have not spent a night in the hospital (indicated by the tall bar at ‘0’ on the ‘hospdays’ axis). For those who have, there is a trend where more nights in the hospital correspond to more doctor consultations, although the increase in consultations with each additional night is not as steep as might be expected. This is indicated by the positive coefficient for ‘hospdays’ in the regression model, although the effect size is relatively small (compared to the effect size for the ‘hospadmi’ variable, for example). he analysis suggests that hospital stays are associated with an increased number of doctor consultations, although the relationship is not as strong as the one between hospital admissions and consultations. This could indicate that while the occurrence of a hospital stay is a significant factor in healthcare utilization, the length of stay may not proportionately increase the number of doctor consultations afterward.

‘doctorco’ vs. ‘medicine’

Bivariate analysis with variable ‘medicine’:

```
describe(df$medicine)
```

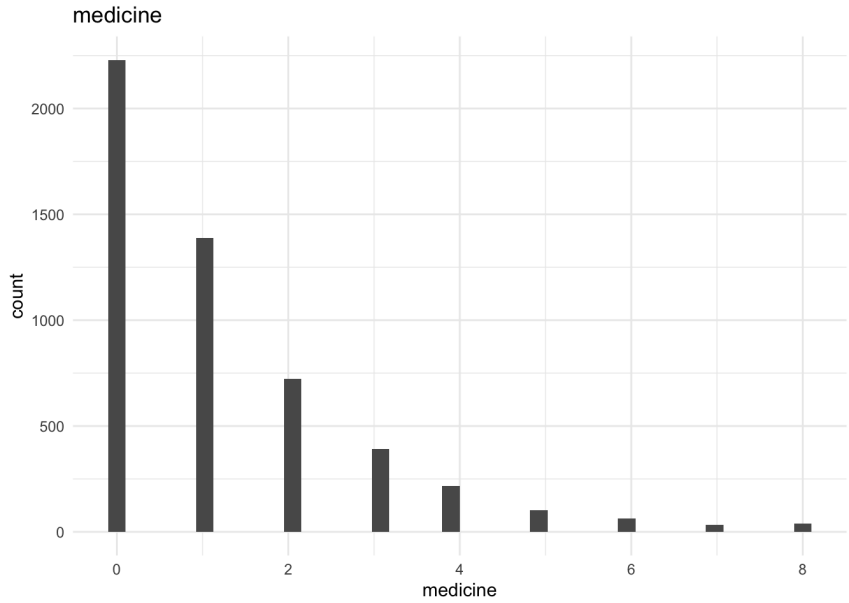
```
## df$medicine
##        n  missing distinct     Info     Mean      Gmd
##     5190        0        9    0.898    1.218    1.527
##
## Value          0     1     2     3     4     5     6     7     8
## Frequency   2229  1389   723   393   218   103    63    32    40
## Proportion 0.429 0.268 0.139 0.076 0.042 0.020 0.012 0.006 0.008
##
## For the frequency table, variable is rounded to the nearest 0
```

```
# value counts
df %>% count(medicine)
```

medicine <dbl>	n <int>
0	2229
1	1389
2	723
3	393
4	218
5	103
6	63
7	32
8	40

9 rows

```
# plot of medicine
ggplot(df, aes(x=medicine)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("medicine") +
  theme_minimal()
```




```
# Create a cross-tabulation
medicine_doctorco_table <- table(df$medicine, df$doctorco)

# Chi-square test of independence
chisq.test(medicine_doctorco_table)
```

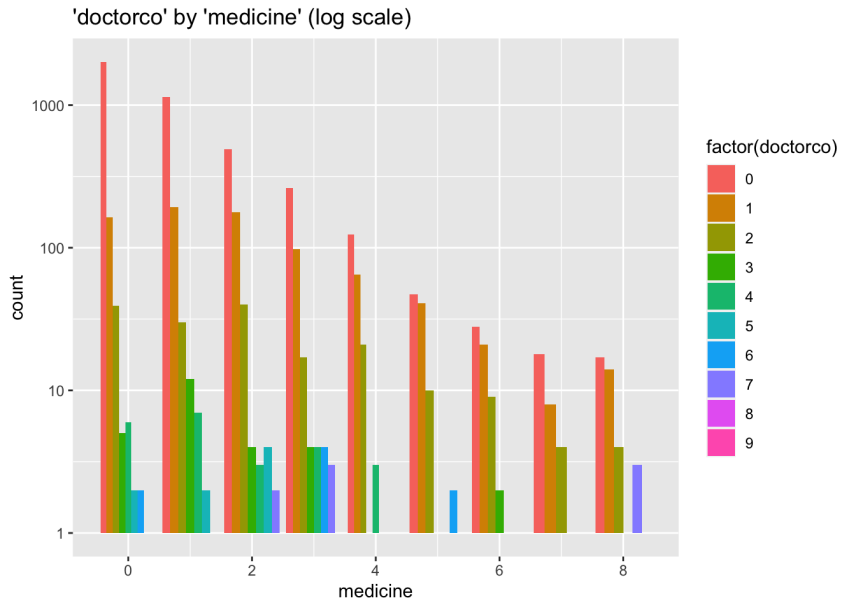
```
## Warning in chisq.test(medicine_doctorco_table): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  medicine_doctorco_table
## X-squared = 821.27, df = 72, p-value < 2.2e-16
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ medicine, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ medicine, data = df, init.theta = 0.5619353918,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.77272    0.04476  -39.61   <2e-16 ***
## medicine     0.33668    0.01706   19.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.5619) family taken to be 1)
##
##     Null deviance: 3409.1  on 5189  degrees of freedom
## Residual deviance: 3034.0  on 5188  degrees of freedom
## AIC: 6831.6
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.5619
##           Std. Err.:  0.0448
##
##  2 x log-likelihood:  -6825.6220
```

```
# For visualization
ggplot(df, aes(x = medicine, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'medicine' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



The plot likely shows that as the number of medications increases, there is a corresponding increase in doctor consultations. This trend is expected since individuals on multiple medications are often those with more complex health conditions that require frequent medical attention. The analysis suggests a significant relationship between medication use and healthcare utilization. Individuals who take more medications, whether prescribed or non-prescribed, tend to have more doctor consultations. This could reflect the need for regular monitoring and management of multiple or complex health conditions.

‘doctorco’ vs. ‘prescrib’

Bivariate analysis with variable ‘prescrib’:

```
describe(df$prescrib)
```

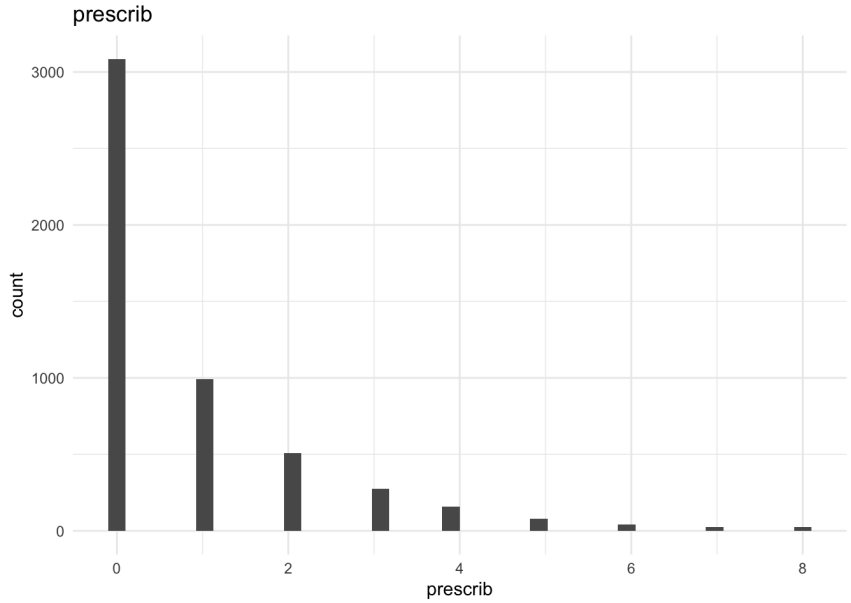
```
## df$prescrib
##        n  missing distinct     Info     Mean      Gmd
##     5190        0        9    0.782   0.8626    1.267
##
## Value          0     1     2     3     4     5     6     7     8
## Frequency   3085   994   509   276   157    80    40    23    26
## Proportion 0.594 0.192 0.098 0.053 0.030 0.015 0.008 0.004 0.005
##
## For the frequency table, variable is rounded to the nearest 0
```

```
# value counts
df %>% count(prescrib)
```

prescrib <dbl>	n <int>
0	3085
1	994
2	509
3	276
4	157
5	80
6	40
7	23
8	26

9 rows

```
# plot of prescrib
ggplot(df, aes(x=prescrib)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("prescrib") +
  theme_minimal()
```




```
# Create a cross-tabulation
prescrib_doctorco_table <- table(df$prescrib, df$doctorco)

# Chi-square test of independence
chisq.test(prescrib_doctorco_table)
```

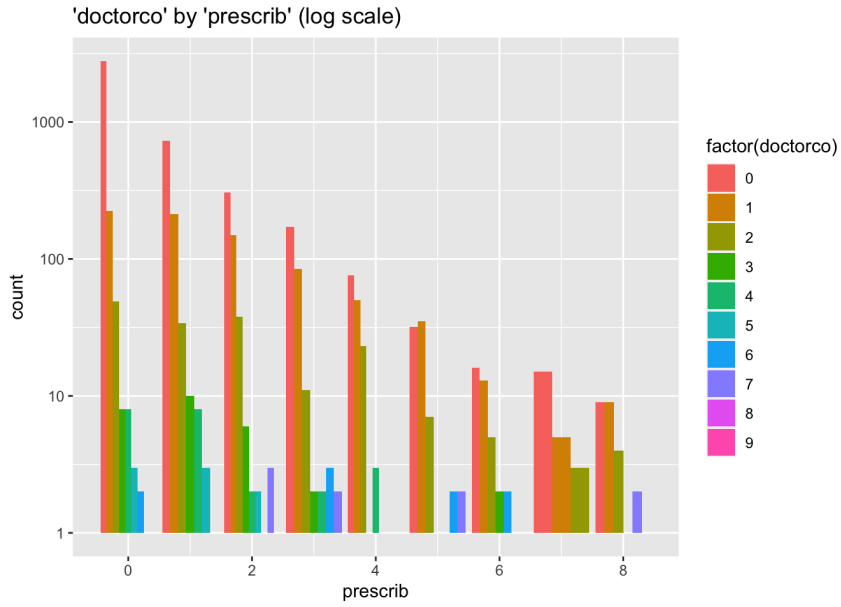
```
## Warning in chisq.test(prescrib_doctorco_table): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  prescrib_doctorco_table
## X-squared = 1104.3, df = 72, p-value < 2.2e-16
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ prescrib, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ prescrib, data = df, init.theta = 0.600523452,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.72317    0.04123  -41.79   <2e-16 ***
## prescrib     0.39106    0.01757   22.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.6005) family taken to be 1)
##
##     Null deviance: 3480.7  on 5189  degrees of freedom
## Residual deviance: 3027.6  on 5188  degrees of freedom
## AIC: 6763.3
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.6005
##           Std. Err.:  0.0483
##
##  2 x log-likelihood:  -6757.2900
```

```
# For visualization
ggplot(df, aes(x = prescrib, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'prescrib' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



The plot shows that the count of doctor consultations increases with the number of prescribed medications. This pattern is typical as individuals on multiple medications may require more medical oversight and follow-up, which would be reflected in an increased number of doctor consultations. The analysis points to a significant relationship between the use of prescribed medications and healthcare utilization. This could reflect the need for ongoing medical management of chronic conditions or the monitoring of medication effectiveness and side effects.

‘doctorco’ vs. ‘nonpresc’

Bivariate analysis with variable ‘nonpresc’:

```
describe(df$nonpresc)
```

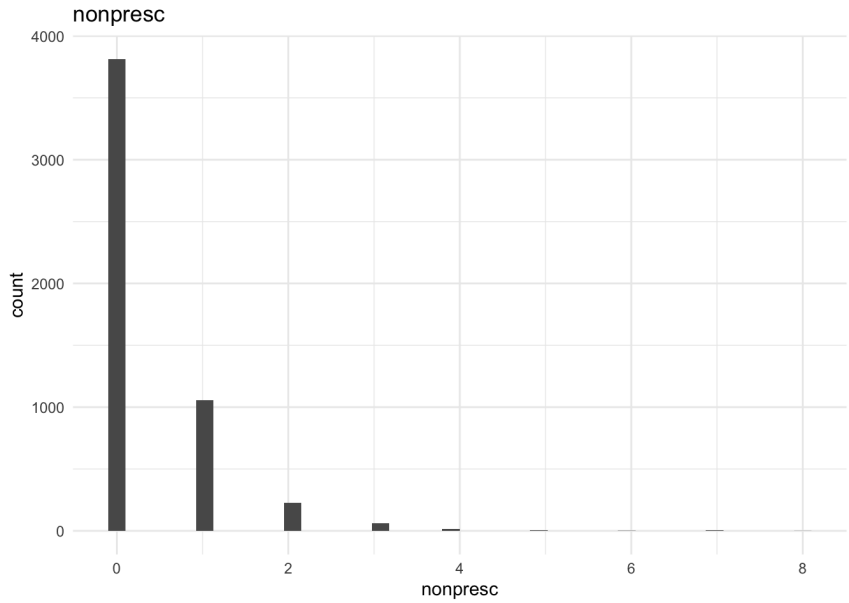
```
## df$nonpresc
##        n  missing distinct     Info     Mean      Gmd
##     5190        0        9    0.595   0.3557   0.5625
##
## Value          0     1     2     3     4     5     6     7     8
## Frequency   3814  1055   228    64    15     7     2     4     1
## Proportion 0.735 0.203 0.044 0.012 0.003 0.001 0.000 0.001 0.000
##
## For the frequency table, variable is rounded to the nearest 0
```

```
# value counts
df %>% count(nonpresc)
```

nonpresc <dbl>	n <int>
0	3814
1	1055
2	228
3	64
4	15
5	7
6	2
7	4
8	1

9 rows

```
# plot of nonpresc
ggplot(df, aes(x=nonpresc)) +
  geom_histogram(position="dodge", bins=40) +
  ggtitle("nonpresc") +
  theme_minimal()
```




```
# Create a cross-tabulation
nonpresc_doctorco_table <- table(df$nonpresc, df$doctorco)

# Chi-square test of independence
chisq.test(nonpresc_doctorco_table)
```

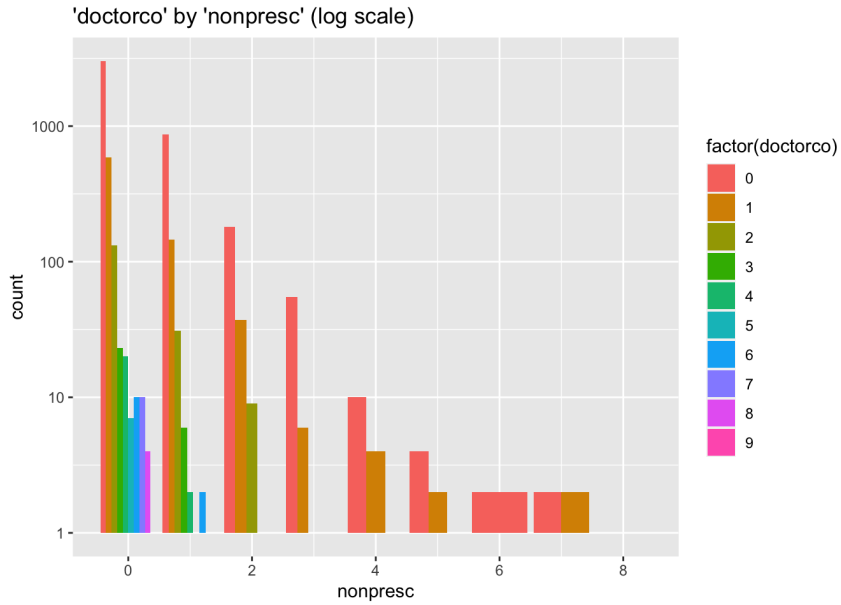
```
## Warning in chisq.test(nonpresc_doctorco_table): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  nonpresc_doctorco_table
## X-squared = 52.059, df = 72, p-value = 0.9632
```

```
# It could also fit a model like a negative binomial if doctor consultations are overdispersed

nb_model <- glm.nb(doctorco ~ nonpresc, data = df)
summary(nb_model)
```

```
##
## Call:
## glm.nb(formula = doctorco ~ nonpresc, data = df, init.theta = 0.3781419414,
##     link = log)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.17964    0.03780 -31.204   <2e-16 ***
## nonpresc    -0.05441    0.04920  -1.106    0.269
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(0.3781) family taken to be 1)
##
##     Null deviance: 2971.7  on 5189  degrees of freedom
## Residual deviance: 2970.4  on 5188  degrees of freedom
## AIC: 7176.7
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  0.3781
##           Std. Err.:  0.0266
##
##  2 x log-likelihood:  -7170.6570
```

```
# For visualization
ggplot(df, aes(x = nonpresc, fill = factor(doctorco))) +
  scale_y_continuous(trans = "log10") +
  ggtitle("'doctorco' by 'nonpresc' (log scale)") +
  geom_bar(position = "dodge", stat = "count")
```



Surprisingly, the p-value for the chi-squared test is high (p = 0.9632), indicating no significant association between the number of non-prescribed medications taken and the frequency of doctor consultations. The coefficient for 'nonpresc' is not statistically significant (-0.05441 with p = 0.269), suggesting that the number of non-prescribed medications does not have a clear effect on the number of doctor consultations. The plot show that the increase in the number of non-prescribed medications does not correspond to an increase in doctor consultations. This could be because non-prescribed medications are often used for minor ailments that do not require medical advice or because people self-medicating may not always seek professional healthcare advice. The lack of a significant relationship between the use of non-prescribed medications and doctor consultations contrasts with the findings for prescribed medications. This could reflect different health-seeking behaviors and attitudes towards healthcare utilization when it comes to self-medication versus prescribed treatment regimens.

Summary

- Age ('age'): There was a significant relationship between age categories and the number of doctor consultations. Younger age groups tended to have fewer consultations, while older age groups had more, likely reflecting the increased healthcare needs with advancing age.
- Sex ('sex') and Income ('income'): Both sex and income showed significant associations with doctor consultations. It was observed that certain sexes and income groups tend to have different frequencies of doctor visits, possibly due to differences in health status, access to healthcare resources, or health-seeking behaviors.
- Healthcare Coverage: Those covered by government healthcare due to low income, recent immigration status, or unemployment, as well as those covered due to old age or disability pensions, showed different patterns of doctor consultations, underscoring the role of healthcare accessibility and support in healthcare utilization:
    - Private Insurance ('levyplus'): No significant association was found, suggesting having private levy does not affect the frequency of doctor consultations notably.
    - Government Coverage ('freepoor', 'freepera'): There is a significant relationship, especially with 'freepera', indicating that individuals with government coverage due to old-age, disability pension, or being invalid veterans or their family members have more doctor consultations.

- Illness (‘illness’) and Activity Limitation (‘actdays’): The number of illnesses and days of reduced activity due to illness or injury were both strongly associated with the number of doctor consultations, highlighting that acute health episodes and their impact on daily life are significant drivers of medical visits.
- Health Scores (‘hscore’): General health questionnaire scores indicated that poorer perceived health status was associated with more doctor consultations. This aligns with the intuitive understanding that individuals who feel unwell are more likely to seek medical help.
- Non-doctor Healthcare Professional Consultations (‘nondocco’): Interactions with non-doctor health professionals were associated with an increase in doctor consultations, which may reflect a more integrated approach to patient care or more complex health needs that require multidisciplinary management.
- Chronic Conditions (‘chcond1’, ‘chcond2’): Both variables showed significant relationships, especially ‘chcond2’, indicating that chronic conditions, especially when limiting activity, are associated with higher doctor consultations.
- Hospital Admissions (‘hospadmi’) and Hospital Nights (‘hospdays’): Both the occurrence and duration of hospital stays were linked to an increased number of doctor consultations. This suggests that hospitalization is a significant event in a patient’s health trajectory that leads to increased follow-up care.
- Medication Use (‘medicine’, ‘prescrib’, ‘nonpresc’): There was a notable difference between prescribed and non-prescribed medication use. Prescribed medication use was strongly associated with more doctor consultations, reflecting the need for medical oversight. In contrast, the use of non-prescribed medications did not show a significant relationship, indicating that self-medication might not lead to increased healthcare utilization.

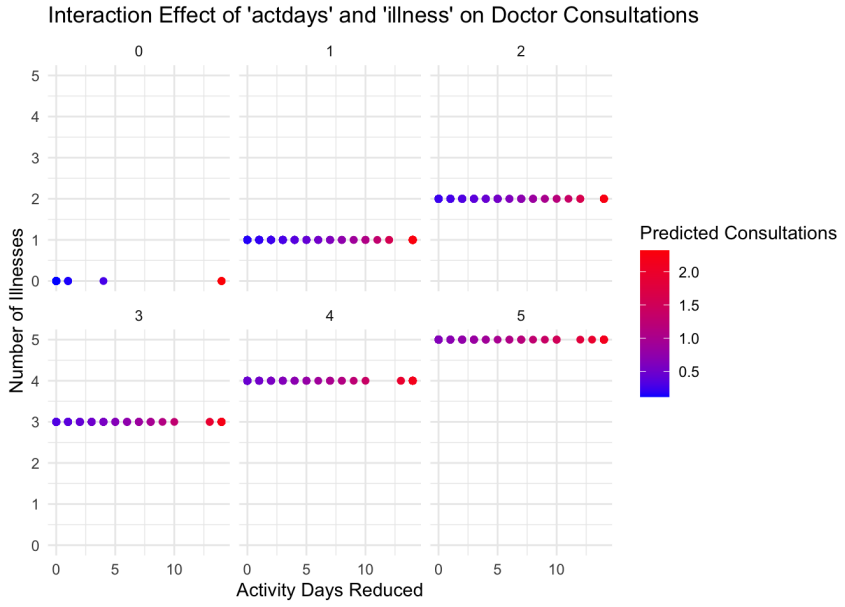
The analysis of these variables provides a multifaceted view of the factors influencing healthcare utilization, particularly doctor consultations. Age, sex, income, healthcare coverage, the presence of chronic conditions, acute health episodes, hospitalization, medication use, health status perception, and engagement with other health professionals all play a role in determining how frequently individuals seek out doctor consultations.

Variable combination - Interaction effect on response variable

In order to study and interpret the best model, considering the variable context and type, we carried out a further analysis of the combinations used.

Interaction between ‘actdays’ and ‘illness’

```
# Fit a model with main effects only
model1 <- glm.nb(doctorco ~ actdays + illness, data = df)

# Fit a model with the interaction term
model2 <- glm.nb(doctorco ~ actdays * illness, data = df)

# Compare the models
anova_result <- anova(model1, model2, test="Chisq")

# Predictions for a range of 'actdays' and 'illness'
df$pred_doctorco <- predict(model2, type = "response")

# Plotting the interaction between 'actdays' and 'illness'
ggplot(df, aes(x = actdays, y = illness, color = pred_doctorco)) +
  geom_point() +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Interaction Effect of 'actdays' and 'illness' on Doctor Consultations",
       x = "Activity Days Reduced",
       y = "Number of Illnesses",
       color = "Predicted Consultations") +
  theme_minimal() +
  facet_wrap(~ illness)
```




```
# Print the ANOVA result
print(anova_result)
```

```
## Likelihood ratio tests of Negative Binomial Models
##
## Response: doctorco
##              Model     theta Resid. df    2 x log-lik.   Test    df LR stat.
## 1 actdays + illness 0.9077012      5187       -6465.474
## 2 actdays * illness 0.9553580      5186       -6434.901 1 vs 2     1 30.57275
##        Pr(Chi)
## 1
## 2 3.215787e-08
```

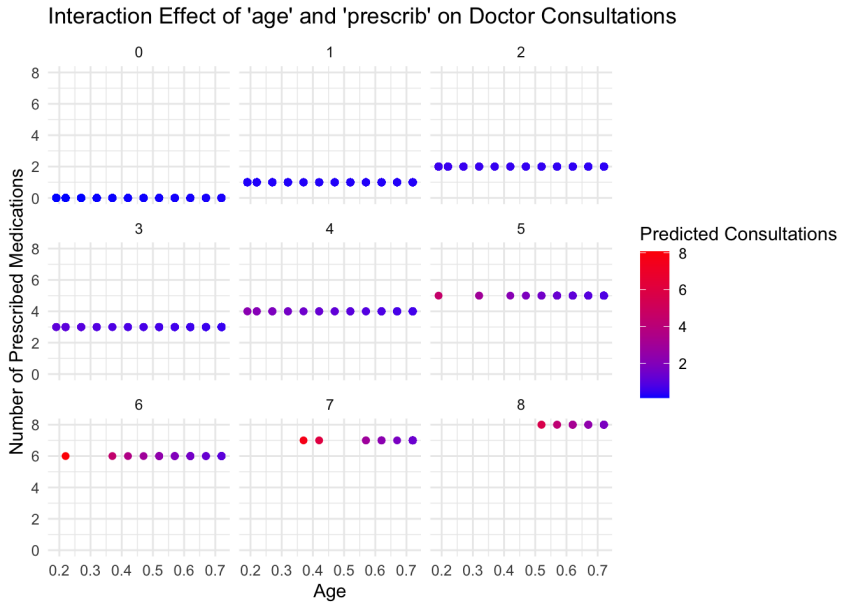
Based on the analysis and the Likelihood Ratio Test (LRT) between model1 (main effects only) and model2 (including the interaction term), there is a significant improvement in the fit of the model when the interaction term is included. The LRT is highly significant ($p < 0.001$), indicating that the interaction between 'actdays' and 'illness' has a significant effect on the number of doctor consultations ('doctorco').

The plot visualizes the interaction effect, where different colors represent different predicted numbers of doctor consultations based on 'actdays' (Activity Days Reduced) and 'illness' (Number of Illnesses). From the plot, it appears that there is a gradient effect, where the predicted number of consultations increases with the number of activity days reduced and the number of illnesses.

The model suggests that both the number of days of reduced activity due to illness or injury and the number of illnesses experienced in the last two weeks are important predictors of doctor consultations. Moreover, their combined effect is significant, which could mean that individuals who have more illnesses and more days of reduced activity are likely to have more doctor consultations.

Interaction between 'age' and 'prescrib'

```
# Fit a model with main effects only
model1 <- glm.nb(doctorco ~ age + prescrib, data = df)

# Fit a model with the interaction term
model2 <- glm.nb(doctorco ~ age * prescrib, data = df)

# Compare the models using an ANOVA
anova_result <- anova(model1, model2, test="Chisq")

# Assuming your model2 is already fit and is called 'model2'
# Predictions for a range of 'age' and 'prescrib'
df$pred_doctorco <- predict(model2, type = "response")

# Plotting the interaction between 'age' and 'prescrib'
ggplot(df, aes(x = age, y = prescrib, color = pred_doctorco)) +
  geom_point() +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Interaction Effect of 'age' and 'prescrib' on Doctor Consultations",
       x = "Age",
       y = "Number of Prescribed Medications",
       color = "Predicted Consultations") +
  theme_minimal() +
  facet_wrap(~ prescrib)
```




```
# Print the ANOVA result
print(anova_result)
```

```
## Likelihood ratio tests of Negative Binomial Models
##
## Response: doctorco
##            Model     theta Resid. df    2 x log-lik.   Test    df LR stat.
## 1 age + prescrib 0.6007208      5187       -6756.718
## 2 age * prescrib 0.6387491      5186       -6706.550 1 vs 2     1 50.16829
##        Pr(Chi)
## 1
## 2 1.411093e-12
```

The likelihood ratio test comparing a model with only main effects (age and prescribed medications) to a model with their interaction term suggests that the interaction model is significantly better. As age increases, the number of prescribed medications also tends to increase. Both of these factors individually correlate with an increased number of doctor consultations. The model predicts that, for a given age, increases in the number of prescribed medications are associated with a higher number of doctor consultations. Similarly, for a given number of medications, increases in age are associated with more doctor visits. The interaction suggests that older individuals with more prescribed medications have a disproportionately higher number of doctor consultations than what would be expected if the effects of age and medication were simply additive. This could be due to older age groups potentially having more complex medical needs, which necessitate both more medications and more frequent medical oversight.

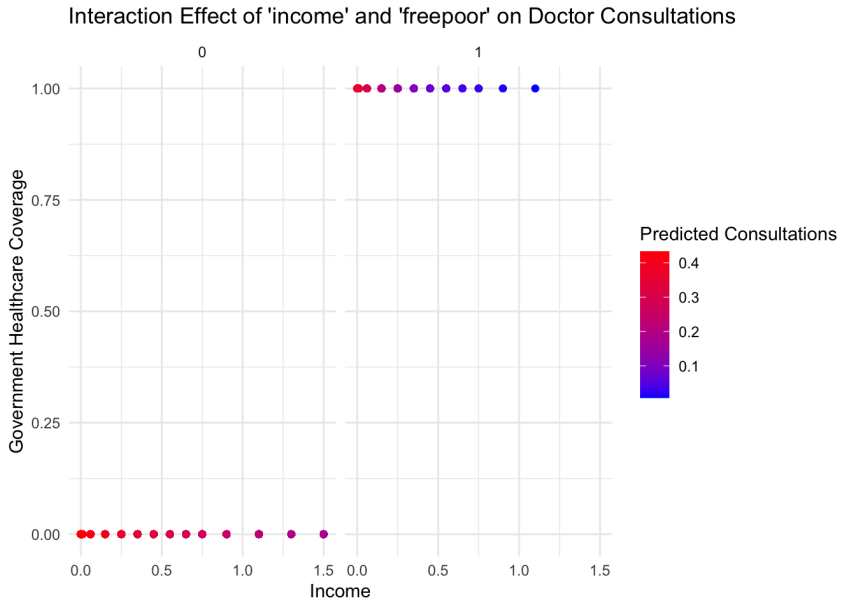
Interaction between ‘income’ and ‘freepoor’

```
# Fit a model with main effects only
model1 <- glm.nb(doctorco ~ income + freepoor, data = df)

# Fit a model with the interaction term
model2 <- glm.nb(doctorco ~ income * freepoor, data = df)

# Compare the models
anova(model1, model2, test="Chisq")
```

Model	theta	Resid. df	2 x log-lik.	Test	df	LR stat.	Pr(Chi)
<chr>	<dbl>	<int>	<dbl>	<chr>	<int>	<dbl>	<dbl>
income + freepoor	0.4015775	5187	-7115.258		*NA*	*NA*	*NA*
income * freepoor	0.4039872	5186	-7108.225	1 vs 2	1	7.032788	0.008003068

2 rows

```
# Assuming your model is already fit and is called 'model2'
# Predictions for a range of 'income' and 'freepoor'
df$pred_doctorco <- predict(model2, type = "response")

# Plotting the interaction between 'income' and 'freepoor'
ggplot(df, aes(x = income, y = freepoor, color = pred_doctorco)) +
  geom_point() +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Interaction Effect of 'income' and 'freepoor' on Doctor Consultations",
       x = "Income",
       y = "Government Healthcare Coverage",
       color = "Predicted Consultations") +
  theme_minimal() +
  facet_wrap(~ freepoor)
```



The analysis of the interaction between ‘income’ and ‘freepoor’ on the number of doctor consultations using a negative binomial model shows a significant interaction effect. While the model that includes only the main effects of ‘income’ and ‘freepoor’ has a theta parameter of 0.4016, adding the interaction term slightly adjusts this parameter to 0.4040. The likelihood ratio test comparing the model with only main effects (model1) and the model with the interaction term (model2) indicates a significant improvement in fit when the interaction term is included ($p = 0.0080$). The LR statistic is 7.033, suggesting that the interaction between income and government healthcare coverage (freepoor) significantly

affects the number of doctor consultations. The plot visualizes this interaction by showing predicted consultation numbers across different income groups, differentiated by their government healthcare coverage status. The color gradient indicates the level of predicted consultations, with deeper colors reflecting higher numbers of predicted consultations. The clear pattern in the plot would suggest that income and freepoor status combined influence the frequency of doctor consultations more than each of these variables alone.

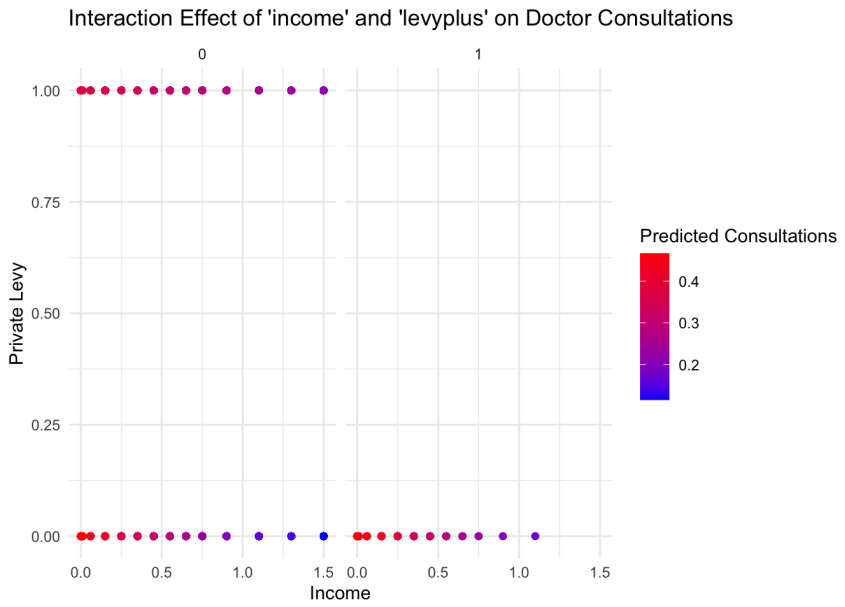
Interaction between ‘income’ and ‘levyplus’

```
# Fit a model with main effects only
model1 <- glm.nb(doctorco ~ income + levyplus, data = df)

# Fit a model with the interaction term
model2 <- glm.nb(doctorco ~ income * levyplus, data = df)

# Compare the models
anova(model1, model2, test="Chisq")
```

Model <chr>	theta <dbl>	Resid. df <int>	2 x log-lik. <dbl>	Test <chr>	df <int>	LR stat. <dbl>	Pr(Chi) <dbl>
income + levyplus	0.3943993	5187	-7132.887		*NA*	*NA*	*NA*
income * levyplus	0.3974714	5186	-7125.021	1 vs 2	1	7.865944	0.005037451

2 rows

```
# Predictions for a range of 'income' and 'levyplus'
df$pred_doctorco <- predict(model2, type = "response")

# Plotting the interaction between 'income' and 'levyplus'
ggplot(df, aes(x = income, y = levyplus, color = pred_doctorco)) +
  geom_point() +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Interaction Effect of 'income' and 'levyplus' on Doctor Consultations",
       x = "Income",
       y = "Private Levy",
       color = "Predicted Consultations") +
  theme_minimal() +
  facet_wrap(~ freepoor)
```



The analysis of the interaction between ‘income’ and ‘levyplus’ on the number of doctor consultations, using a negative binomial model, reveals a significant interaction effect. The model including only the main effects of ‘income’ and ‘levyplus’ has a theta parameter of 0.3944. Introducing the interaction term adjusts this parameter to 0.3975. The likelihood ratio test comparing the model with only main effects (model1) and the model with the interaction term (model2) indicates a significant improvement in the fit when including the interaction term ($p = 0.0050$). The LR statistic of 7.866 suggests that the combined effect of income and private levy status (‘levyplus’) significantly affects the frequency of doctor consultations. The provided plot would visualize this interaction, showing predicted consultation numbers across different income levels, with a distinction made based on whether individuals have a private levy (‘levyplus’). The color gradient indicates the level of predicted consultations, with warmer colors indicating higher numbers of predicted consultations. This pattern demonstrates that the interaction between income and private levy status has a significant effect on the predicted number of doctor consultations.

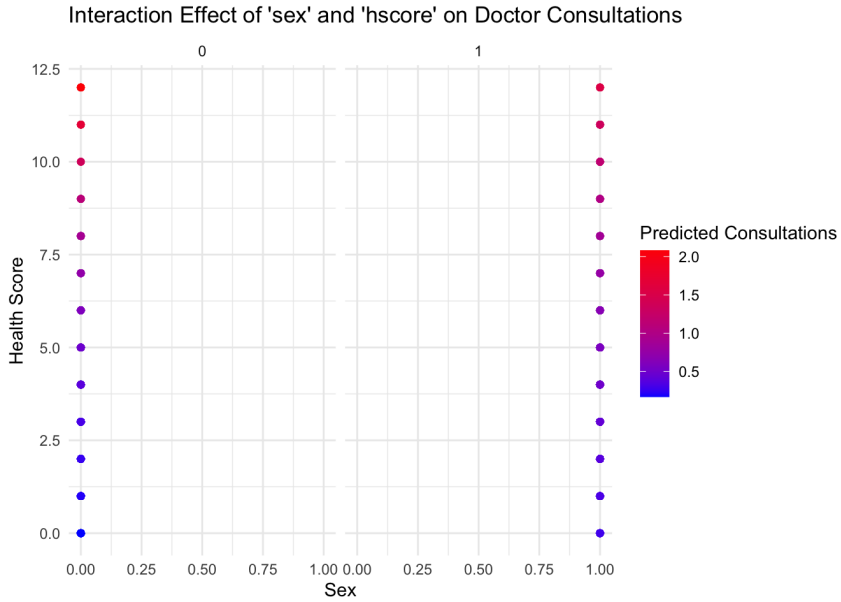
Interaction between ‘sex’ and ‘hscore’

```
# Fit a model with main effects only
model1 <- glm.nb(doctorco ~ sex + hscore, data = df)

# Fit a model with the interaction term
model2 <- glm.nb(doctorco ~ sex * hscore, data = df)

# Compare the models
anova(model1, model2, test="Chisq")
```

Model	theta	Resid. df	2 x log-lik.	Test	df	LR stat.	Pr(Chi)
<chr>	<dbl>	<int>	<dbl>	<chr>	<int>	<dbl>	<dbl>
sex + hscore	0.4734344	5187	-6971.281		*NA*	*NA*	*NA*
sex * hscore	0.4768489	5186	-6965.356	1 vs 2	1	5.92499	0.01492762

2 rows

```
# Predictions for a range of 'sex' and 'hscore'
df$pred_doctorco <- predict(model2, type = "response")

# Plotting the interaction between 'sex' and 'hscore'
ggplot(df, aes(x = sex, y = hscore, color = pred_doctorco)) +
  geom_point() +
  scale_color_gradient(low = "blue", high = "red") +
  labs(title = "Interaction Effect of 'sex' and 'hscore' on Doctor Consultations",
       x = "Sex",
       y = "Health Score",
       color = "Predicted Consultations") +
  theme_minimal() +
  facet_wrap(~ sex)
```



The analysis indicates that there is a statistically significant interaction effect between ‘sex’ and ‘hscore’ on the number of doctor consultations, as evidenced by the likelihood ratio test (p-value = 0.0149). This suggests that the relationship between health score and doctor consultations is different for different sexes. The plot visually represents this interaction, with color intensity indicating the predicted number of consultations based on the model. It appears that as the health score increases, there is a tendency for the predicted number of consultations to increase, and this pattern might vary between sexes, although the exact nature of the differences is not detailed in the output provided.To give a detailed summary, we would need to interpret the coefficients from the model output, which typically includes the estimates for the main effects and the interaction term. Since the p-value for the interaction is significant, it would be advisable to look at the coefficients to understand how sex modifies the effect of health score on the number of doctor consultations. If you have the coefficients from the model, they would tell us more about the nature of this interaction.

Cluster Analysis

Cluster analysis, that is a subset of unsupervised machine learning technique, plays a significant role in the field of healthcare dataset analysis, particularly during the Expanatory Data Analysis (EDA) phase. The importance of cluster analysis in this context can be outlined in several key aspects:

- Identifying Patient Groups: Cluster analysis helps n identifying groups or clusters of patients with similar characteristics. This can be crucial for understanding patterns in diseases, treatment responses, and patient outcomes. For instance, patients with similar symptoms or genetic profiles might be clustered together to tailor more effective treatments.
- Disease Subtyping and Precision Medicine: In diseases like cancer, where there are various subtypes with different prognoses and treatment responses, cluster analysis can help in distinguishing these subtypes. This leads to more personalized treatment approaches, enhancing the effectiveness of precision medicine.
- Resource Allocation and Management: By clustering patients based on their health status, healthcare providers can optimize resource allocation. For example, identifying groups of high-risk patients allows for the prioritization of care and resources where they are needed most.
- Predictive Analytics: Clustering can be used to identify patterns that might not be apparent through traditional statistical methods. These patterns can inform predictive models that anticipate future health trends, potential outbreaks, or the spread of diseases.
- Enhancing Healthcare Delivery: Understanding patient clusters can help in designing targeted healthcare programs and policies. It allows healthcare systems to tailor their services to meet the specific needs of different patient groups, improving overall healthcare delivery.
- Risk Stratification: Cluster analysis can assist in stratifying patients into different risk categories based on various health indicators. This stratification is crucial for preventive care and early intervention strategies.
- Discovery of New Medical Insights: Clustering can reveal previously unknown relationships and correlations within the data, leading to new medical insights and hypotheses. This can be particularly useful in areas like genomics and epidemiology.

- Cost Reduction: By enabling more efficient and targeted healthcare interventions, cluster analysis can contribute to cost reduction in healthcare systems. This is especially important in managing chronic diseases and long-term care.
- Benchmarking and Performance Improvement: Healthcare providers can use cluster analysis to benchmark performance and outcomes against similar patient groups, leading to continuous improvement in healthcare quality.

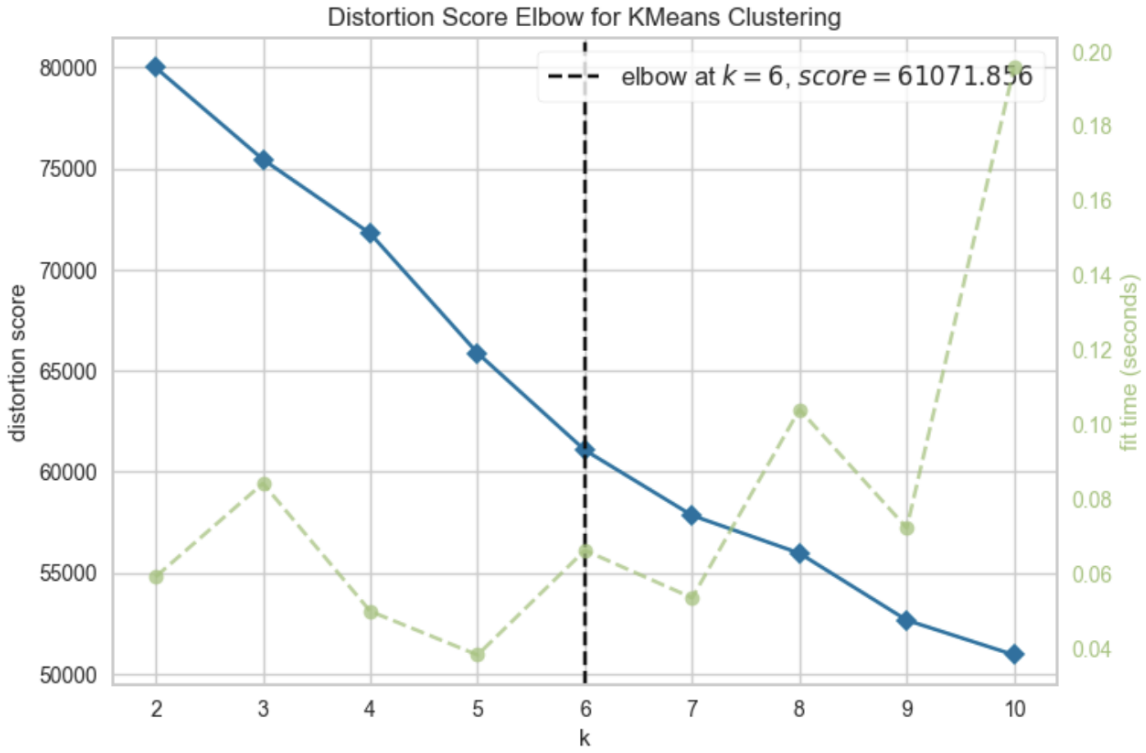
Elbow Method

The “Elbow Method” is a popular technique used in cluster analysis to determine the optimal number of clusters (k) in a dataset.

Concept of the Elbow Method

The Elbow Method involves running k-means clustering on the dataset for a range of values of k (e.g., k from 1 to 10), and then for each value of k, calculating the sum of squared distances from each point to its assigned center. When these overall dispersions are plotted against the number of clusters, the “elbow” of the curve represents an optimal value for k. This point is where the rate of decrease sharply changes, indicating that adding more clusters beyond this number does not significantly improve the fit of the model.

This part was implemented in Python. First, after doing a feature selection (excluding ‘doctorco’, ‘agesq’ and ‘constant’), the selected feature were standardized, because K-Means clustering is sensitive to the scales of the data. Then, after fixed a random state for reproducibility, a KElbowVisualizer is created, which was applied the K-Means algorithm to the data for a rang of k values (from 2 to 10). It was fitted the scaled data to the K-Means model and found the optimal k (number of clusters) by computing the distortion score for each model with different k values. It was plotted that shows the elbow curve. The plot typically shows the distortion score (within-cluster sum of squares) decreasing with the number of clusters k. The “elbow” point on this curve is where the distortion score starts to decrease at a slower rate, suggesting it as the optimal number of clusters. After that, K-Means is performed again using the optimal number of clusters determined by the Elbow Method and each record in the data set was assigned a cluster label based on the final clustering model.



Elbow Method for cluster

From this plot, it is possible to see the Eblow Plot produced by ‘KElbowVisualizer’. This plot shows the distortion score on the y-axis and the number of clusters (k) on the x-axis. The elbow is marked at k=6 with a dashed line, where the distortion score is around 61071.856, indicating that 6 is the optimal number of clusters according to the Elbow Method. The plot also shows the fit time for each k, which is less relevant for determining the number of clusters but provides insight into the computational cost.

It was also printed the centroids table that gives the average value of each feature within each cluster, which helps in understanding the profile of each clusters.

	actdays	age	chcond1	chcond2	cluster	freepera
0	0.300084	0.337259	0.097234	0.127410	6.694049	2.498002e-16
1	6.924119	0.566640	0.365854	0.552846	14.355014	4.878049e-01
2	0.479858	0.395486	0.808057	0.003555	20.893365	1.110223e-16
3	0.479238	0.267474	0.246426	0.064670	15.232131	1.633764e-02
4	0.281279	0.654110	0.677626	0.115982	8.894977	8.100457e-01
5	0.650000	0.248818	0.250000	0.109091	15.000000	2.775558e-17

	freepoor	hospadmi	hospdays	hscore	illness	income
0	0.000000e+00	0.108969	0.391450	1.010897	1.062867	0.653965
1	5.420054e-03	0.991870	11.701897	3.888889	2.926829	0.408889
2	-1.387779e-17	0.093602	0.515403	0.776066	1.321090	0.888483
3	6.938894e-18	0.115044	0.513955	1.014976	1.072158	0.643880
4	0.000000e+00	0.106849	0.672146	1.041096	1.933333	0.304831
5	1.000000e+00	0.181818	0.959091	1.781818	1.259091	0.300045

	levyplus	medicine	nondocco	nonpresc	prescrib	sex
0	9.983236e-01	0.836547	0.104778	0.398994	0.437552	0.538977
1	4.037940e-01	3.644986	1.617886	0.425474	3.219512	0.707317
2	9.454976e-01	0.909953	0.093602	0.349526	0.560427	0.482227
3	5.551115e-16	0.565691	0.072158	0.339006	0.226685	0.313138
4	1.461187e-01	2.036530	0.173516	0.311416	1.725114	0.773516
5	3.330669e-16	0.686364	0.077273	0.359091	0.327273	0.381818

Centroids

For instance, Cluster 1 seems to be characterized by a higher average ‘actdays’ (activity limitation days) and a higher ‘hscore’ (general health questionnaire score), suggesting this cluster may represent individuals with more health issues and limitations.

Cluster profiling

To profile the clusters, it was examined the centroid values for each clusters. These values gives an idea of the “typical” member of each cluster.

Cluster 0: “The Healthy Young Adults”

- Age: Younger age group.
- Health: Relatively healthy with low actdays, chcond1, chcond2, and hscore.
- Healthcare Utilization: Lower hospadmi and hospdays, indicating fewer hospital admissions and shorter stays.
- Income: Higher than average income, possibly indicating better access to health resources.
- Insurance: Almost all have private health insurance (levyplus).
- Gender: Slightly more females than males (sex).

Cluster 1: “The High-Needs Elderly”

- Age: Older age group.
- Health: Higher actdays, chcond1, chcond2, indicating more chronic conditions and health issues.
- Healthcare Utilization: Highest hospadmi and hospdays, suggesting frequent and longer hospital stays.
- Income: Lower income, which could be related to retirement.
- Insurance: Mixed insurance coverage.
- Gender: More females than males.

Cluster 2: “The Stable Middle-Aged”

- Age: Middle-aged group.
- Health: High chcond1, but low chcond2, suggesting chronic conditions without severe limitations.
- Healthcare Utilization: Low hospadmi and hospdays.
- Income: Higher income, possibly at peak career stage.
- Insurance: Mostly covered by private insurance.

Cluster 3: “The Young and Occasionally Unwell”

- Age: Young, similar to Cluster 0.
- Health: Moderate health issues, higher than Cluster 0 but less severe than other clusters.
- Healthcare Utilization: Moderate hospadmi and hospdays.
- Income: Similar to Cluster 0, relatively higher income.
- Insurance: Lacks private health insurance.
- Gender: More females than males.

Cluster 4: “The Aging with Care Needs”

- Age: Older individuals, but not as old as Cluster 1.
- Health: Many chronic conditions (chcond1 and chcond2).
- Healthcare Utilization: Moderate hospadmi and higher hospdays.
- Income: Lower income, which may impact their healthcare options.

- Insurance: Some with private health insurance.
- Gender: A higher proportion of females.

Cluster 5: “The Economically Disadvantaged”

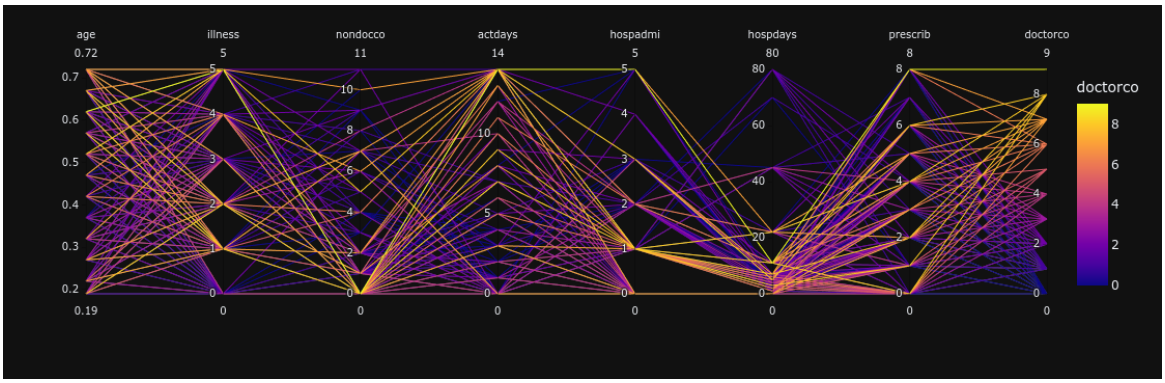
- Age: Younger, but with health issues.
- Health: Moderate actdays, some chronic conditions.
- Healthcare Utilization: Low to moderate hospadmi and hospdays.
- Income: Low income, suggesting economic challenges.
- Insurance: Lacks private health insurance, possibly relying on public assistance (freepoor).
- Gender: Balanced gender distribution.

Each of these profiles suggests different needs and characteristics. Strategic decisions can be made based on these insights. For example, preventive health measures may be prioritized for clusters with chronic conditions but not currently utilizing a lot of healthcare services (like Cluster 2). On the other hand, policy makers may focus on providing better economic support or healthcare access to clusters like Cluster 5, who might be economically disadvantaged.

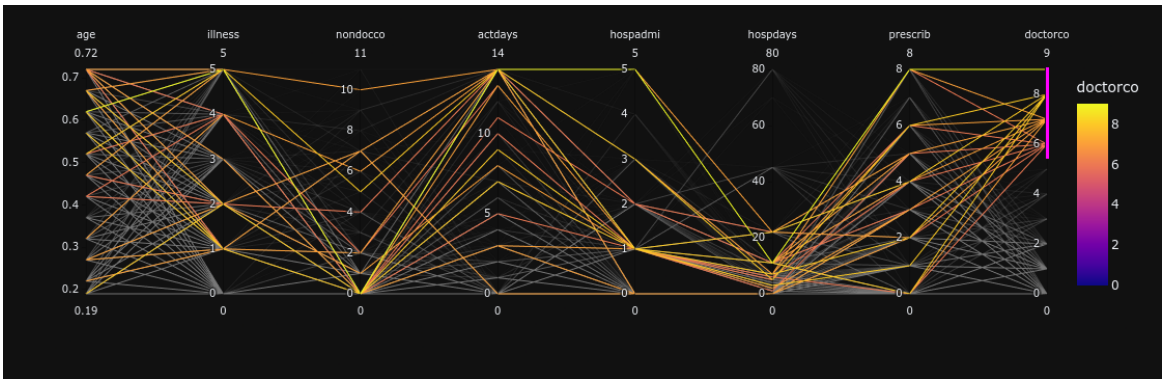
Parallel Coordinates Plot

The code for the following segment was done in Python (all code can be found in the corresponding jupyter notebook).

Let’s have a look at a parallel coordinates plot and see if there are any visible patterns in the data.



We can take a closer look at the high values of doctors visits by selecting just them:



From the plots above we can draw a few conclusions. The most surprising one is that people who visit the doctor a lot are not the same people who stay in the hospital a lot. It seems that cases with a high number of doctors visits may have a relatively normal length hospital stay, after/before which they visit the doctor a lot. Interestingly, we can also see that a lot of people with a high number of doctors visits do get admitted to hospital at least once, if not multiple times.

Another observation is that all people with a high number of doctors visits have been sick at least once in the past 2 weeks. We also don’t see any strong correlation between nondocco and doctorco which is interesting. Age also doesn’t seem to be a huge factor, although there are more old people going to the doctor.

Projecting the Data to 2 Dimensions

In order to visualize the dataset with a plot, we need to reduce the number of dimensions. It was used 2 methods for dimensionality reduction:

PCA:

PCA works very well for continuous data, but our dataset is all categorical variables, making PCA a little out of place. It should however be interesting to see how it fares.

MCA:

MCA is basically PCA but for categorical data. It should work much better for the categorical variables.

For most of this analysis, It was excluded doctors visits from the downprojection, and instead encoding it using color. This provides a way of analyzing how easy it may be to separate out high doctors visits from the rest of the data.

Explaining MCA

MCA stands for “Multiple Correspondence Analysis”, and is an extension of Correspondence Analysis.

The procedure of MCA is as follows:

1. Build an *indicator matrix* (one-hot encoding of the data)
2. Perform CA on the indicator matrix.

Correspondence Analysis

There are two explanations of how CA works in the context of MCA. One is the actual regular theory, while the other is based on PCA. The regular procedure is as follows:

1. Let N be the sum of all entries in our indicator matrix X.
2. $\mathrm{Z} = \frac{\mathrm{X}}{\mathrm{N}}$ (We're basically normalizing the data).
3. Let r be a vector containing the sums along the rows of Z, and let c be the sum along all columns of Z.
4. With this, perform the decomposition: $\mathrm{M} = \mathrm{diag(r)}^{-\frac{1}{2}}(\mathrm{Z} - \mathrm{rc}^{\mathrm{T}})\mathrm{diag(c)}^{-\frac{1}{2}}$.
5. This gives you $\mathrm{M} = \mathrm{P}\Delta\mathrm{Q}^{\mathrm{T}}$.

PCA Based Explanation

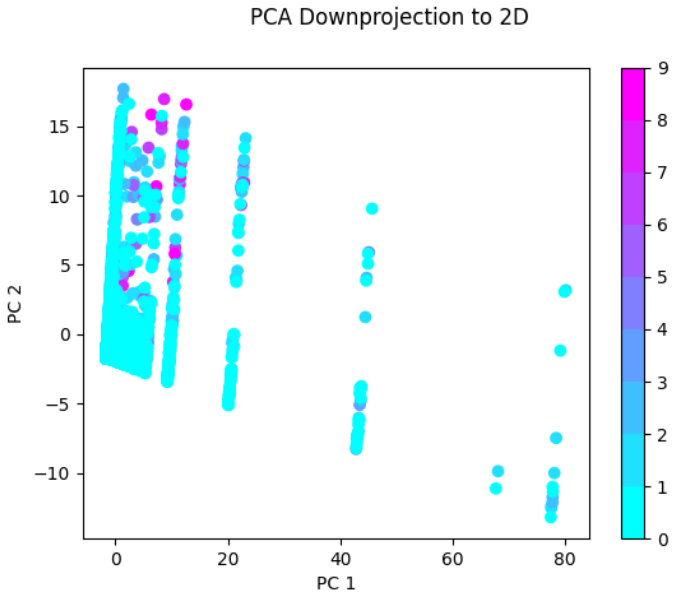
This is the PCA based explanation:

1. Let $\mathrm{y_{ik}}$ be a value in the indicator matrix and let $\mathrm{p_k}$ be the sum of row k in the indicator matrix.
2. We normalize the indicator matrix: $\mathrm{x_{ik}} = \mathrm{y_{ik}}/\mathrm{p_k} - 1$
3. Apply un-standardized PCA to this matrix.

Both of these approaches have been proven equivalent.

Down-projecting with PCA:

Simply used normalized PCA on the dataset while excluding response variable



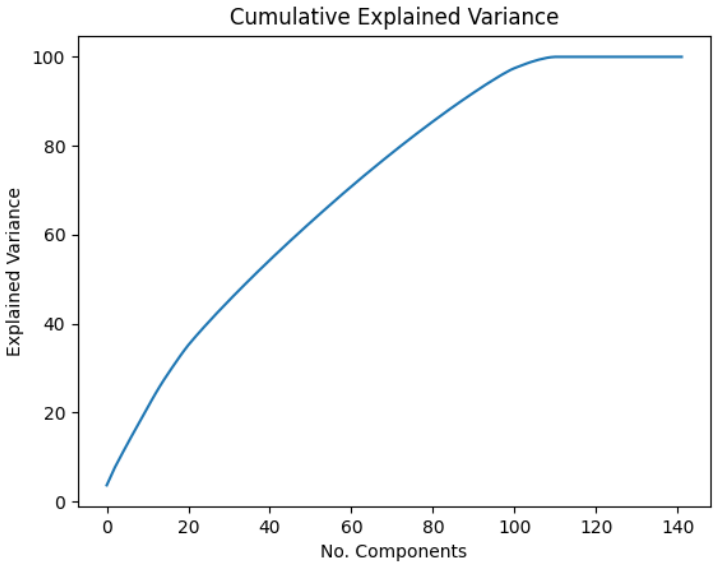
The results aren't very good-looking, but it is clearly possible to see that points with high amounts of doctors visits do stand out in some areas. It's clear however that PCA isn't really meant for categorical data.

Down-Projecting with MCA

After applying MCA, we can first have a look at the eigenvalues and explained variance.

component	eigenvalue	% of variance	% of variance cumulative
0	0.236	3.62%	3.62%
1	0.134	2.05%	5.67%
2	0.129	1.98%	7.64%
3	0.115	1.76%	9.41%
4	0.111	1.71%	11.11%
5	0.111	1.70%	12.82%

Looking at the % of variance, this doesn't look overly promising. We can visualize this better with a graph.

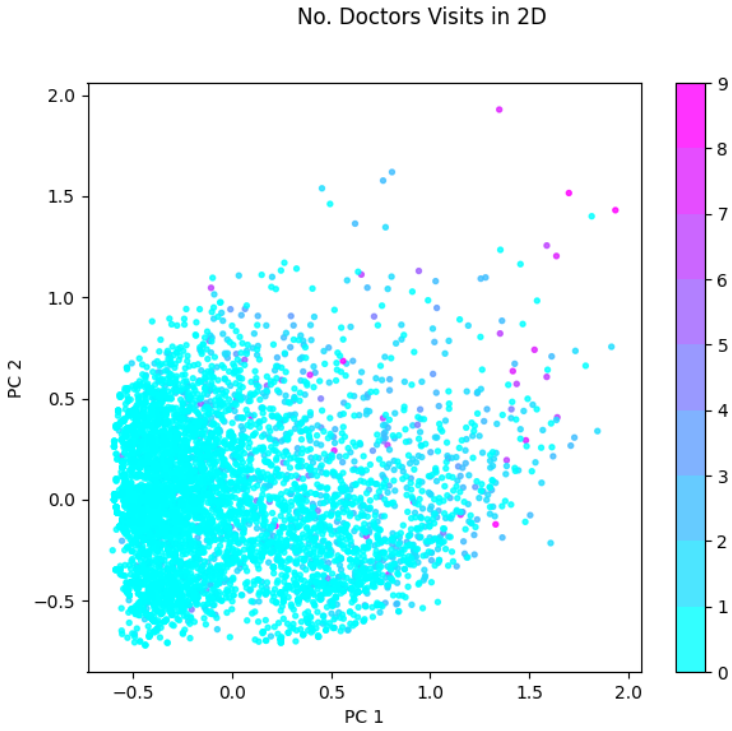


A note on the number of components in the graph, there are many more components than there are dimensions in the data because when performing MCA we have to one-hot encode the data, therefore increasing the number of dimensions.

MCA is clearly able to remove some of the obviously correlated dimensions such as age/agesq. However, there isn't any real elbow in the plot. This means that reducing the number of dimensions past the very correlated ones would lose a significant amount of information.

This is a little disappointing, however, if we were really looking to reduce the amount of dimensions, we could choose around the 20 mark where there is a slight bend in the curve.

Visualizing the Data Using MCA



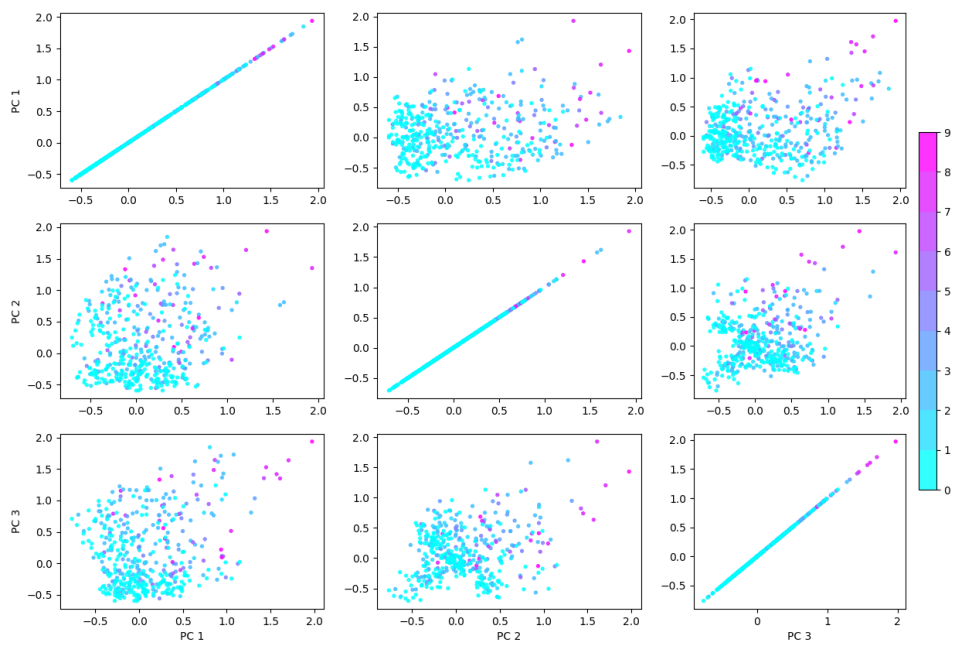
This already looks a lot better than PCA, although it's difficult to see exactly what's going on due to the overwhelming number of zero's. Despite this, we can see that again the points with a high number of doctors visit's do stand out a little.

Looking at Different Principal Components

Different PC's capture different elements of the data. We can explore this by plotting them. Here we'll have a look at all combinations of the first 3 principal components, which capture about 7.64% percent of the variance of the data.

Under-sampling the Data

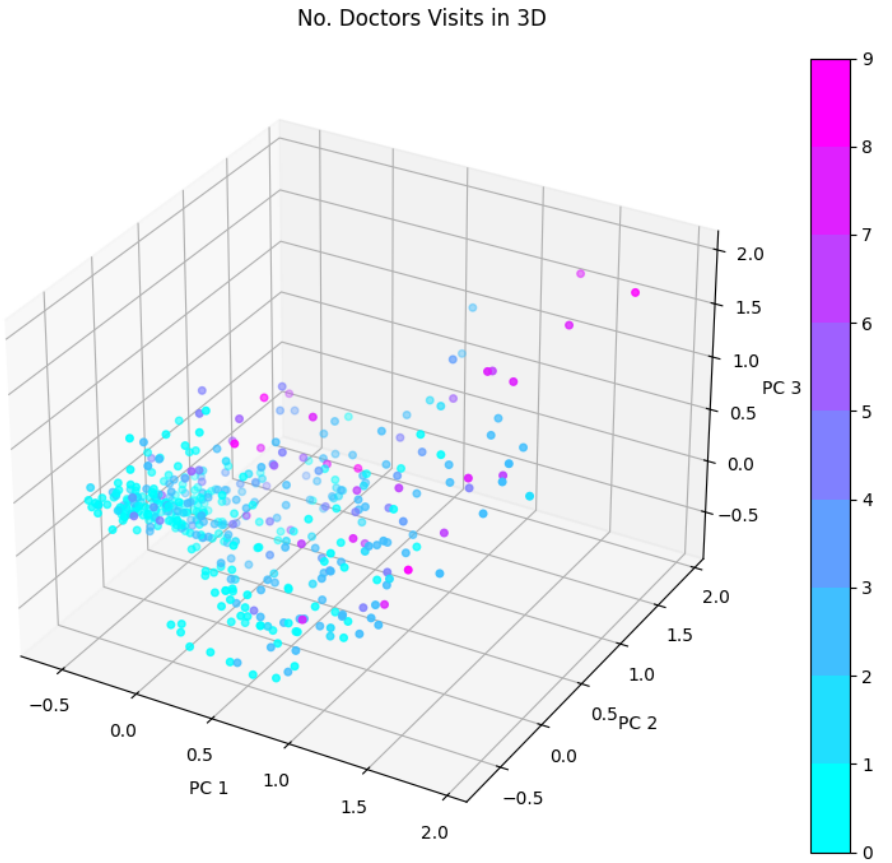
In order to aid in visualization, was under-sampled all data by a factor of 2, and additionally under-sampled 0's and 1's by a factor of 10. This makes it much easier to see points with a high number of doctors visits.



What is interesting about the plot above is that the best separation of high numbers of doctors visits is being done by the 1st and 3rd principal components. This is contrary to the fact that the 1st and 2nd capture more variance in the data.

3D Downprojection

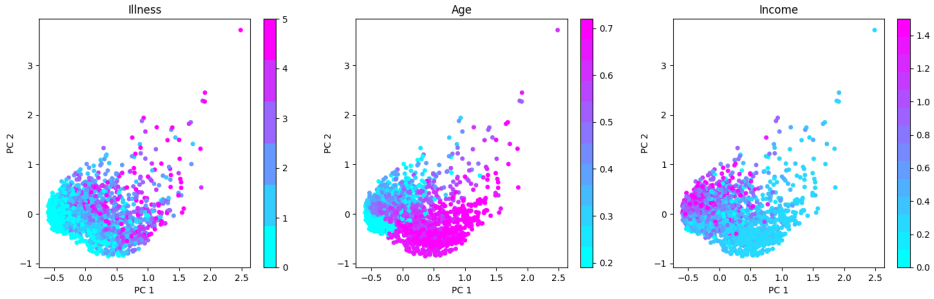
Lastly, we can also look at the projection in 3D space.



In 3D we can clearly see that there do seem to be some clusters in the data. However, they don't seem to be strictly related to doctors visits.

Additional Data Exploration

In the following down-projections was included doctors visits in the MCA. An important note is that no longer under sampling 0's and 1's here. Instead, it is undersampling the entire dataset by a factor of 2.

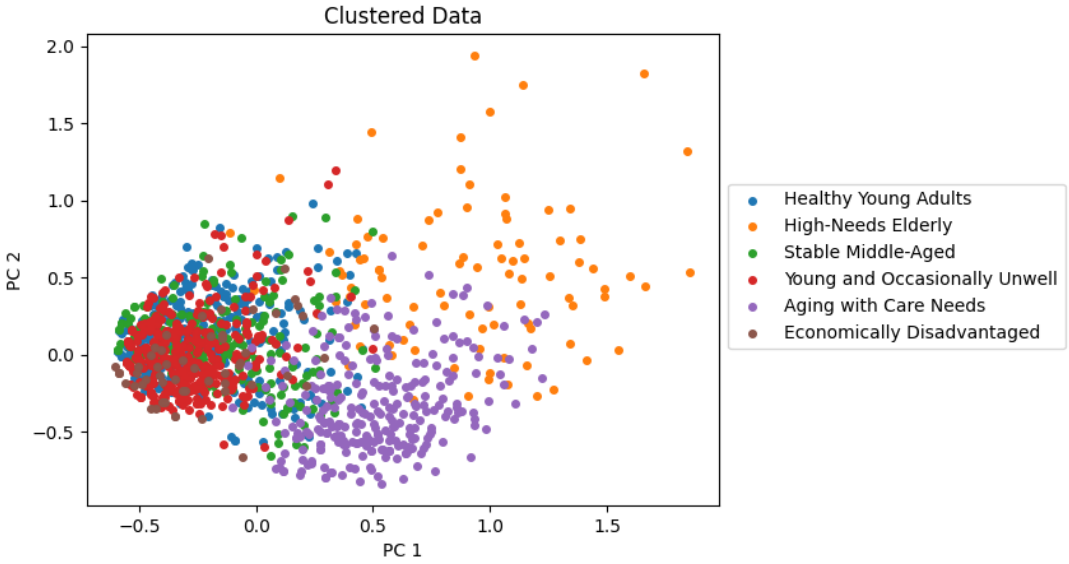


The above plot's give some interesting insight into how illness, age and income are related. We can see that the number of illnesses in the past 2 weeks is dramatically lower in young people. We can also see that Income is much lower in older people. This makes sense as they are probably in retirement and are not being paid a full time wage anymore.

The extreme points that seem like outliers are actually the ones with a high number of doctors visits.

Visualizing Higher Dimensional Clusters

The team has looked at clustering in higher dimensions, and this down-projection gives us an opportunity to visualize these clusters.



Comparing with the previous plot's, it's interesting to be able to directly see what kind of information these clusters have captured. For example, the "Aging with Care Needs" cluster is exactly on top of where the old population is in the dataset. This makes a lot of sense and may even be obvious, but it's very interesting to be able to visually see it.

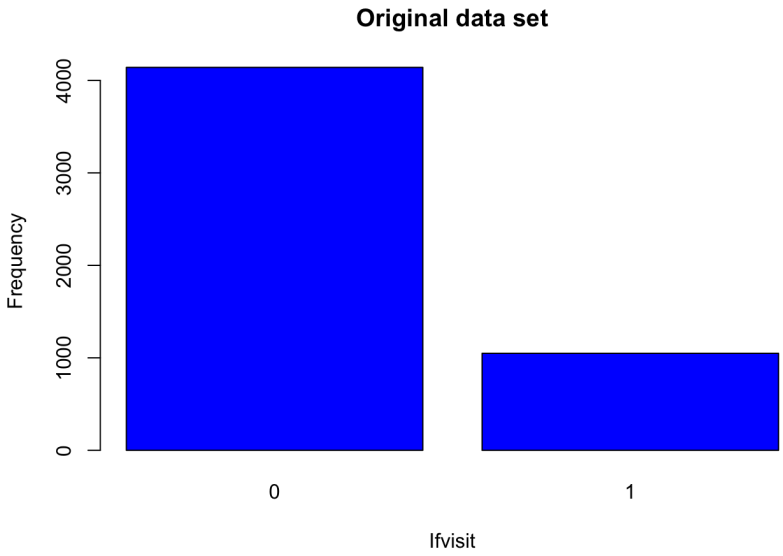
We also see that a few clusters are on top of one another. This illustrates the fact that our down-projection is inherently losing information and is called "overcrowding". We knew this was a problem based off of the explained variance presented earlier. While these clusters may make sense in higher dimensions where there is extra variance, they end up overlapping here.

Overcrowding is a common problem with dimensionality reduction, and it would be interesting to be able to try some different algorithms like t-SNE or ISOMAP, which take some extra measures over PCA to mitigate this problem.

Binary classification problem

We start by analyzing some models where the response variable "doctorco" is transformed into the binary response variable "ifvisit". This binary variable is equal to 0 if "doctorco" is 0, and is equal to 1 otherwise. By doing so, we are modelling the number of people that went to a doctor's visit at least once in the last two weeks.

To start we import the data set and create the "ifvisit" variable. From the following graph we can see the skewness of the data set regarding this variable:



We can try fitting a GAM using the obtained data set:

```
model_gam <- gam(ifvisit ~ s(hospdays) + s(actdays) + age*prescrib + freepoor + hscore + nonpresc + illness, data
= train_data, family = binomial(link = "logit"))
summary(model_gam)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## ifvisit ~ s(hospdays) + s(actdays) + age * prescrib + freepoor +
##     hscore + nonpresc + illness
##
## Parametric coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.72414    0.13763 -19.794  < 2e-16 ***
## age           1.52680    0.27834   5.485 4.13e-08 ***
## prescrib      0.91275    0.10486   8.704  < 2e-16 ***
## freepoor     -0.91922    0.30227  -3.041  0.00236 **
## hscore        0.06140    0.02004   3.064  0.00219 **
## nonpresc     -0.18891    0.06426  -2.940  0.00328 **
## illness       0.16493    0.03493   4.722 2.34e-06 ***
## age:prescrib -1.01709    0.17178  -5.921 3.20e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df Chi.sq p-value
## s(hospdays) 4.286  4.955  17.17 0.00514 **
## s(actdays)  3.259  3.931 172.31 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.205   Deviance explained = 18.8%
## UBRE = -0.16331  Scale est. = 1         n = 4152
```

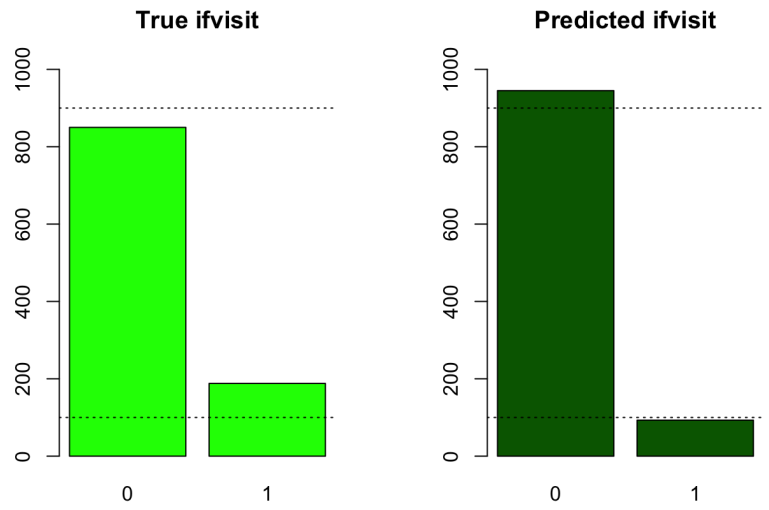
Both the variables “hospdays” and “actdays” were considered as splines after a top to bottom examination of the covariates. The summary shows that we can account for approx. 19% of explained deviance at best, which is not a great result so far.

```
## [1] "AIC (GAM): 3473.93"
```

```
## [1] "MAE (GAM): 0.1686"
```

```
## [1] "Number of true ifvisit: 188"
```

```
## [1] "Number of ifvisit predicted: 93"
```

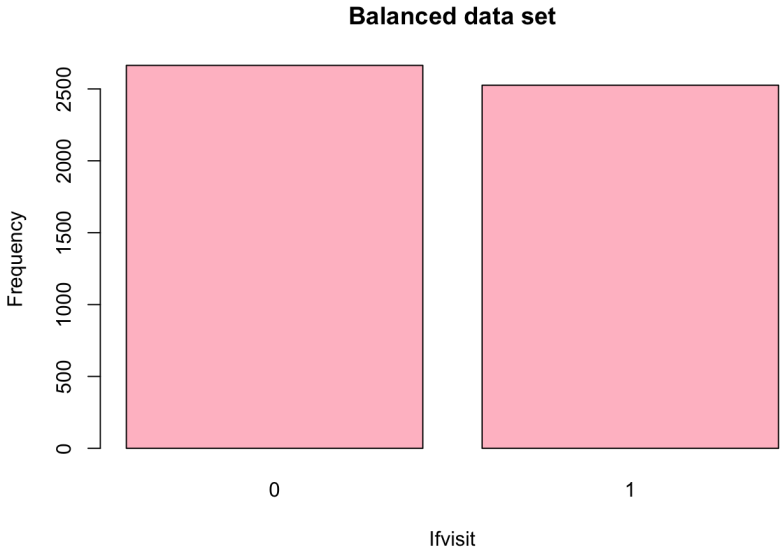


The sum of predicted “ifvisit” yealds a total of 93, against the true value of 188, indicating that GAM isn’t performing very well at predicting the minority class ‘1’. To enhance it’s abilities, we should further manipulate the data set as shown in the following section.

Balancing the data set with ROSE

Since we transformed the problem into a binary classification problem we can use techniques to balance the data set. One option is to use ROSE (Random Over-Sampling Examples), a method used for oversampling the minority class in binary classification problems to balance the data set. It involves generating synthetic examples from the existing minority class instances. This can be achieved by randomly selecting a minority class instance and introducing variations to create new synthetic instances.

```
data.rose <- ROSE(ifvisit ~ ., data = data, seed = 1, hmult.majo = 0)$data
```



The ROSE method generated some data that were not suitable for the data set, for ex. it generated negative data for variables that can only obtain positive integer values. We solved this problem by taking the absolute value of the data set after the data augmentation was done.

Here ROSE was first used to generate more data, then was considered the absolute value of the dataset since the variables can’t obtain negative values, and then rounded accordingly so that every observation has integers as values when the variable is a count.

GAM with ROSE

We can try again fitting a GAM on this new balanced data set and see if the performance improved:

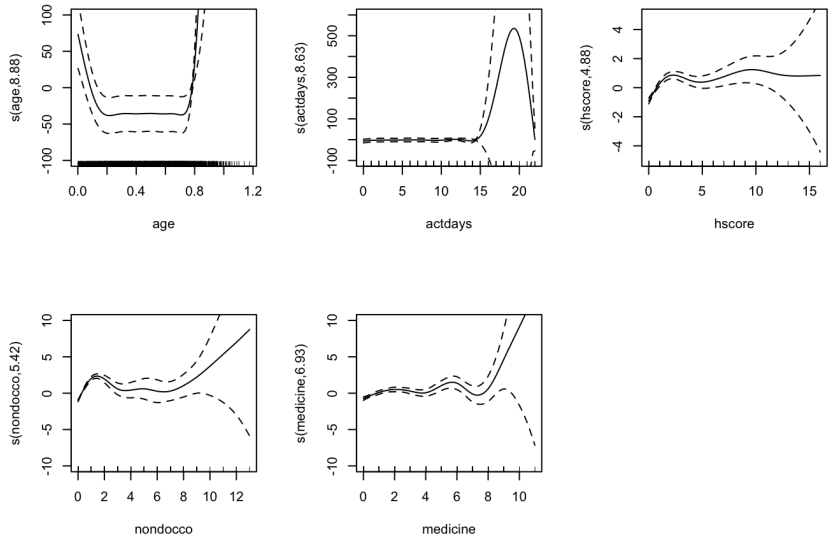
```
model_gam <- gam(ifvisit ~ s(age) + s(actdays) + s(hscore) + s(nondocco) + s(medicine), data=train_data, family =
binomial(link = "logit"))
summary(model_gam)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## ifvisit ~ s(age) + s(actdays) + s(hscore) + s(nondocco) + s(medicine)
##
## Parametric coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    40.91      13.05   3.135  0.00172 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##               edf Ref.df Chi.sq p-value
## s(age)      8.882  8.986 140.36  <2e-16 ***
## s(actdays)  8.626  8.860 596.54  <2e-16 ***
## s(hscore)   4.882  5.811  85.30  <2e-16 ***
## s(nondocco) 5.421  6.249 287.02  <2e-16 ***
## s(medicine) 6.930  7.430  49.68  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.843   Deviance explained = 79.4%
## UBRE = -0.69709  Scale est. = 1         n = 4152
```

For this model five covariates were selected in order to predict the response variable, those being “actdays”, “hscore”, “age”, “medicine” and “nondocco”, all five of them being considered as splines. The selection was done by including all the variables and sequentially cutting those not fit for the model.

We can appreciate a huge improvement over the previous model, getting to a high value of approx. 80% explained deviance.

We can see from the following plot the splines considered:



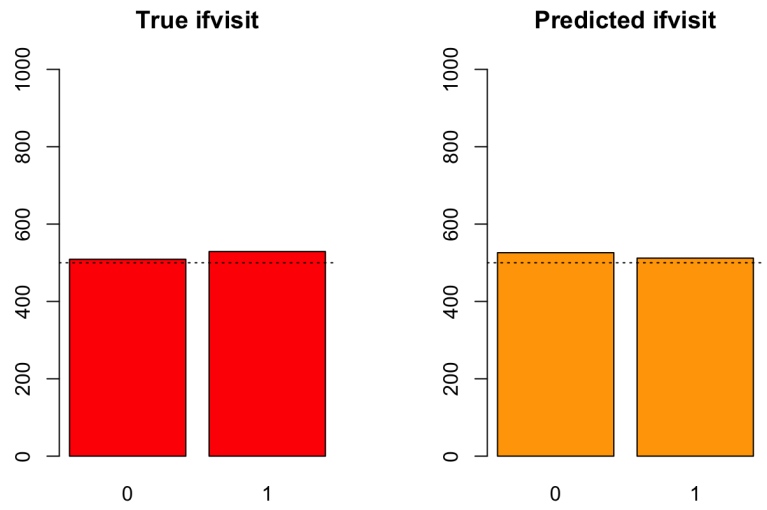
From the “age” spline we can infer that being “young” leads generally to a moderate amount of visits, being “adult” leads to less people going to the doctor, while being “old” implies a big chance of going to a doctor’s visit. This follows the common logic and experience, so it’s a good sign of the model working. Generally, high values of all the covariates implies a high chance of going to the doctor, as can be see in the previous plots.

```
## [1] "AIC (GAM): 1257.7"
```

```
## [1] "MAE (GAM): 0.05491"
```

```
## [1] "Number of true ifvisit: 529"
```

```
## [1] "Number of ifvisit predicted: 512"
```



As we can see from these results, the sum of predicted “ifvisit” gets close to the true value while also maintaining a low MAE value, meaning that the model is predicting correct values pretty consistently.

GLM with ROSE

Now let’s try fitting a GLM to the same balanced data set and see if it can compete with the GAM one:

```
model_glm <- glm(ifvisit ~  hospadmi + nondocco + illness + actdays + prescrib + nonpresc, data = train_data, fam
ily = binomial)
summary(model_glm)
```

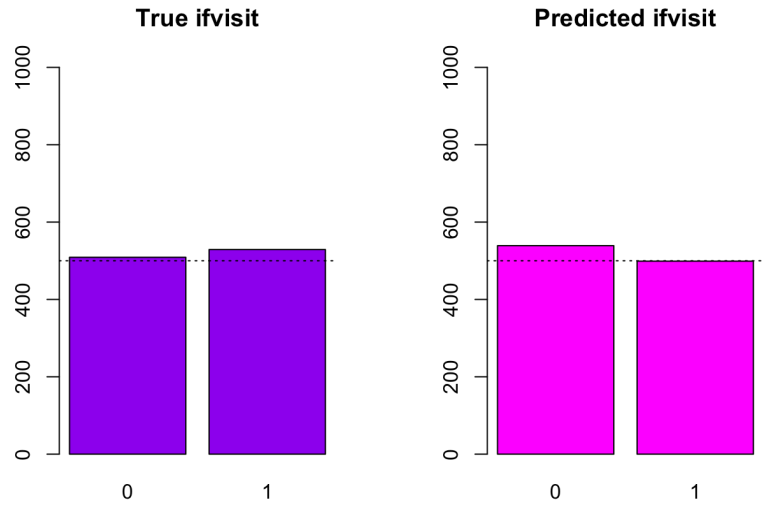
```
##
## Call:
## glm(formula = ifvisit ~ hospadmi + nondocco + illness + actdays +
##     prescrib + nonpresc, family = binomial, data = train_data)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.58171    0.09089 -28.404  < 2e-16 ***
## hospadmi     0.95475    0.09502  10.048  < 2e-16 ***
## nondocco     1.22241    0.08169  14.965  < 2e-16 ***
## illness      0.11996    0.03324   3.609 0.000307 ***
## actdays      0.53808    0.02988  18.005  < 2e-16 ***
## prescrib     0.47077    0.03747  12.563  < 2e-16 ***
## nonpresc     0.25345    0.05829   4.348 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5749.9  on 4151  degrees of freedom
## Residual deviance: 2947.5  on 4145  degrees of freedom
## AIC: 2961.5
##
## Number of Fisher Scoring iterations: 6
```

Firstly, we can see that the AIC is two times the AIC from the GAM model. Furthermore, the explained deviance is at best around 50%, which is a better result but still not as good as GAM.

```
## [1] "MAE (GLM): 0.1021"
```

```
## [1] "Number of true ifvisit: 529"
```

```
## [1] "Number of ifvisit predicted: 499"
```



From the obtained results we can see that GAM manages to find a good approximation of the total number of visits while also keeping a low value of MAE, meaning that the predictions are correct most of the times.

On the other hand, GLM is a little worse at predicting the total number of visits (sum of “ifvisit”) and it scores double the MAE from GAM meaning that the predictions are overall worse but still useful. GAM manages to understand better the variable interactions but GLM is faster and simpler to interpret.

From this analysis we can say that augmenting a skewed data set such as the one we are analyzing can improve and ease the binary classification problem, and also that the GAM model is much better, in this particular data set, at accurately predicting if a person has gone to the doctor in the past two weeks or not.

This concludes the binary classification digression, from now on all the models will try to predict the whole “doctorco” variable.

ZERO INFLATED NEGATIVE BINOMIAL

Traditional Negative Binomial regression extends Poisson regression to manage overdispersion in count data, but it fails when an unusually high number of zero counts is present. The ZINB model is combining the principles of NB regression with a mechanism to account for excess zeros. Specifically, it differentiates between two sources of zeros: those arising from the data’s natural variability “sampling zeros” and those that are structurally inherent or “excess zeros.”

$$P(Y_i = y_i) = \begin{cases} \pi_i + (1-\pi_i)\frac{\Gamma(r+y_i)}{\Gamma(r)y_i!}\left(\frac{r}{r+\mu_i}\right)^r\left(\frac{\mu_i}{r+\mu_i}\right)^{y_i} & \text{if } y_i = 0, \\ (1-\pi_i)\frac{\Gamma(r+y_i)}{\Gamma(r)y_i!}\left(\frac{r}{r+\mu_i}\right)^r\left(\frac{\mu_i}{r+\mu_i}\right)^{y_i} & \text{if } y_i > 0. \end{cases}$$

The ZINB model accounts for the excess of zeros through the component π_i, which represents the probability that an observation will have a count of zero not due to the process described by the Negative Binomial distribution but due to some other, external process. Another key feature is the dispersion parameter r of the Negative Binomial distribution, which is used to model overdispersion. Smaller values of r indicate greater overdispersion relative to the Poisson distribution.

We use the model ‘zeroinfl’ that has two parts:

```
ZINB_model <- zeroinfl(doctorco ~illness * actdays + hscore + chcond1 + age: chcond2
+ hospadmi  + prescrib + nonpresc|levyplus + age:income:freepoor + freepera
+ illness * actdays + prescrib, data = train_data, dist = "negbin")

AIC(ZINB_model)
```

```
## [1] 4908.415
```

```
summary(ZINB_model)
```

```
##
## Call:
## zeroinfl(formula = doctorco ~ illness * actdays + hscore + chcond1 +
##     age:chcond2 + hospadmi + prescrib + nonpresc | levyplus + age:income:freepoor +
##     freepera + illness * actdays + prescrib, data = train_data, dist = "negbin")
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.2568 -0.4370 -0.2498 -0.1729 11.1233
##
## Count model coefficients (negbin with log link):
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.958567   0.132317  -7.244 4.34e-13 ***
## illness          0.097356   0.035625   2.733 0.006280 **
## actdays          0.101887   0.013337   7.640 2.18e-14 ***
## hscore           0.024368   0.013309   1.831 0.067112 .
## chcond11        -0.149808   0.088641  -1.690 0.091017 .
## hospadmi         0.185576   0.044157   4.203 2.64e-05 ***
## prescrib         0.082195   0.024279   3.385 0.000711 ***
## nonpresc        -0.151691   0.048728  -3.113 0.001852 **
## illness:actdays -0.007148   0.004526  -1.579 0.114245
## age:chcond20    -0.209779   0.209290  -1.002 0.316183
## age:chcond21    -0.624787   0.240798  -2.595 0.009469 **
## Log(theta)       0.893182   0.165991   5.381 7.41e-08 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            1.8827     0.2604   7.231 4.79e-13 ***
## levyplus1             -0.4330     0.2189  -1.978 0.047921 *
## freepera1             -1.5382     0.3749  -4.103 4.07e-05 ***
## illness               -0.4317     0.1119  -3.856 0.000115 ***
## actdays               -2.3676     0.8194  -2.890 0.003857 **
## prescrib              -1.6775     0.2515  -6.669 2.57e-11 ***
## illness:actdays        0.4764     0.1719   2.772 0.005567 **
## age:income:freepoor0   0.2498     0.6597   0.379 0.704958
## age:income:freepoor1  20.5385     8.8031   2.333 0.019643 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 2.4429
## Number of iterations in BFGS optimization: 72
## Log-likelihood: -2433 on 21 Df
```

There are significant variables that influence the number of doctor visits. Notably, income factors (both middle and high income showing lower visit rates compared to low-income counterparts), levyplus, health-related variables like illness severity, active days, and hospital admissions directly correlate with increased doctor visits, emphasizing the link between health needs and healthcare demand. Prescription medication requirements further elevate visit frequencies, reflecting ongoing health management needs. Conversely, the use of non-prescription medications is associated with fewer visits, hinting at self-care practices for minor health concerns.

The interaction term age:income:freepoor1 and its significant positive coefficient suggest that older individuals with higher income who qualify for free healthcare are less likely to visit the doctor. This pattern may arise from various factors such as improved health status, access to alternative health resources, or specific policies that affect their healthcare utilization differently. Additionally, the interaction between illness and actdays demonstrates a significant positive effect, indicating that individuals who are ill and experience more days of activity restriction are more likely to seek medical attention, which aligns with expectations.

In the zero-inflation part, variables like levyplus, freepera, illness, actdays, and prescrib are significant, pointing to specific factors that influence the propensity to have zero visits.

Theta is a parameter of the Negative Binomial distribution part of the model it is inversely related to the variance; a smaller θ indicates more dispersion (more variability in count data than what a Poisson model would suggest). Theta of 2.4 suggests some level of overdispersion in the data, but not extremely high.

```
predicted_counts_zinb <- round(predict(ZINB_model, newdata = test_data, type = "response"))
predicted_category_zinb <- ifelse(predicted_counts_zinb < 1, 0,predicted_counts_zinb)

true_counts <- test_data$doctorco
mae_zinb <- mean(abs(predicted_counts_zinb - true_counts))
cat("MAE:", mae_zinb, "\n")
```

```
## MAE: 0.2649326
```

```
rmse_zinb <- sqrt(mean((predicted_counts_zinb - true_counts)^2))
cat("RMSE:", rmse_zinb, "\n")
```

```
## RMSE: 0.6988843
```

The modelsts Mean Absolute Error (MAE) is indicating a relatively precise prediction capability given the context of count data; but still has room for improvement, particularly in accurately predicting higher counts of visits as seen from the Root Mean Squared Error (RMSE).

This section explores the examination of binary outcomes—specifically, the presence or absence of doctor visits. By utilizing confusion matrices and metrics such as balanced accuracy and AUC-ROC, we want to evaluate the model's ability to accurately predict actual visits against the backdrop of a skewed distribution.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 778 104
##          1  72  84
##
##                Accuracy : 0.8304
##                  95% CI : (0.8062, 0.8528)
##     No Information Rate : 0.8189
##     P-Value [Acc > NIR] : 0.17730
##
##                   Kappa : 0.3878
##
##  Mcnemar's Test P-Value : 0.01945
##
##             Sensitivity : 0.9153
##             Specificity : 0.4468
##          Pos Pred Value : 0.8821
##          Neg Pred Value : 0.5385
##              Prevalence : 0.8189
##          Detection Rate : 0.7495
##    Detection Prevalence : 0.8497
##       Balanced Accuracy : 0.6811
##
##        'Positive' Class : 0
##
```

```
## Balanced Accuracy: 0.5384615
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

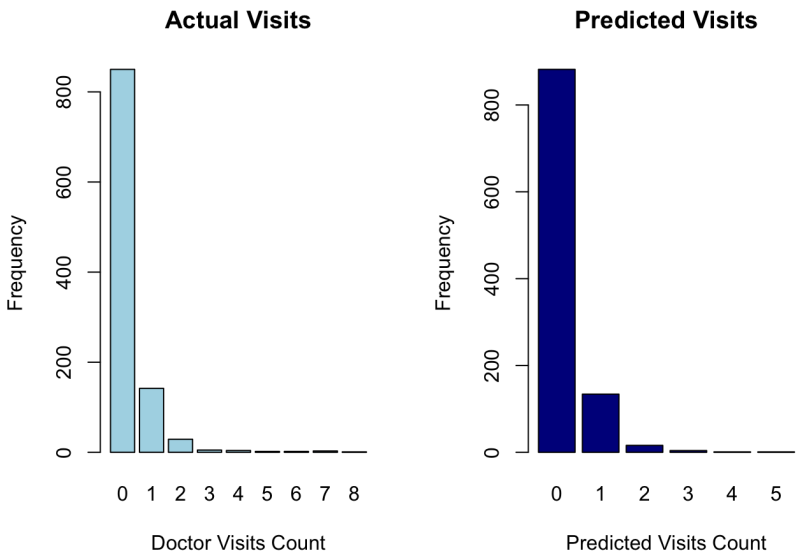
```
## AUC-ROC: 0.6810513
```

The high accuracy indicates that almost all predictions made by the model are correct. This is a relatively high overall accuracy rate. The sensitivity of reflects the model's strong performance in predicting non-visits accurately, a result of the data's inherent imbalance towards this outcome.

However, the model's balanced accuracy and an AUC-ROC score suggest that while the model is better than a dummy classifier, there is room for improvement, particularly in correctly identifying actual visits.

By plotting distribution of both actual and predicted visits, we gain insights into the model's performance in capturing the true distribution of healthcare utilization. Additionally, examining the specific actual versus predicted counts across visit frequencies enables us to identify where the model performs well.

```
actual_freq <- table(true_counts)
predicted_freq_zinb <- table(predicted_counts_zinb)

par(mfrow=c(1,2))

# Bar plot for Actual Counts
barplot(actual_freq, main="Actual Visits", xlab="Doctor Visits Count",
        ylab="Frequency", col="lightblue")

# Bar plot for Predicted Counts
barplot(predicted_freq_zinb, main="Predicted Visits", xlab="Predicted Visits Count",
        ylab="Frequency", col="darkblue")
```




```
par(mfrow=c(1,1))


for (i in 0:9) {
  actual_ <- true_counts == i
  predicted_ <- predicted_counts_zinb == i
  actual_count <- sum(actual_)
  predicted_count<- sum(predicted_)
  cat("Actual count for", i, "Visits:", actual_count, "\n")
  cat("Predicted count for", i, "Visits:", predicted_count, "\n\n")
}
```

```
## Actual count for 0 Visits: 850
## Predicted count for 0 Visits: 882
##
## Actual count for 1 Visits: 142
## Predicted count for 1 Visits: 134
##
## Actual count for 2 Visits: 29
## Predicted count for 2 Visits: 16
##
## Actual count for 3 Visits: 5
## Predicted count for 3 Visits: 4
##
## Actual count for 4 Visits: 4
## Predicted count for 4 Visits: 1
##
## Actual count for 5 Visits: 2
## Predicted count for 5 Visits: 1
##
## Actual count for 6 Visits: 2
## Predicted count for 6 Visits: 0
##
## Actual count for 7 Visits: 3
## Predicted count for 7 Visits: 0
##
## Actual count for 8 Visits: 1
## Predicted count for 8 Visits: 0
##
## Actual count for 9 Visits: 0
## Predicted count for 9 Visits: 0
```

The model does well at predicting when there are no doctor visits, although it predicts slightly more zeros than there actually are. However, as the number of visits goes up, the model does not do as well. It is close when predicting one visit but starts to fall short with two visits and struggles more as the visit numbers increase, not predicting any visits of five or more at all. This gap between the actual and predicted numbers, especially for higher counts of visits, suggests that the model might need some improvements or additional data to better predict these less common situations.

HURDLE NEGATIVE BINOMIAL

The hurdle model offers a distinct approach to modeling count data, unlike Zero-Inflated models, hurdle models decompose the prediction process into two sequential components: a binary process for distinguishing between zero and non-zero counts, followed by a truncated count distribution model exclusively for the positive counts. This structure creates a “hurdle” that separates zero predictions from positive ones, meaning observations must first cross this hurdle before they are considered for positive count predictions.

In the hurdle model framework, the probability of observing a zero count ($y_i = 0$) is denoted by p_i, while the distribution of positive counts ($y_i > 0$) follows a truncated distribution, adjusted to exclude the probability of zero counts. The mathematical representation of the HNB model is as follows:

$$P(Y_i = y_i) = \begin{cases} p_i & \text{if } y_i = 0, \\ (1 - p_i) \frac{p(y_i; \mu_i)}{1 - p(y_i = 0; \mu_i)} & \text{if } y_i > 0, \end{cases}$$

Here, p_i delineates the probability that an observation falls into the zero count category, while $p(y_i; \mu_i)$ denotes the probability mass function (PMF) for positive counts, parameterized by μ_i, within a Negative Binomial distribution managing the observed overdispersion in the data.

The HNB model separates the prediction of no visits from the prediction of one or more visits to better understand healthcare usage. It first decides if a visit happens at all and then predicts how many visits will happen if it does. The model uses different factors to predict both the chance of no visits and the expected number of visits, helping us understand what influences these outcomes.

The hurdle model implemented with the 'pscl' library in R is designed dividing the modeling process into two distinct parts separated by '|'.

```
hurdle_model <- hurdle(doctorco ~ illness + actdays + hospadmi|income:freepoor +
                         actdays *illness + sex*hscore + hospadmi + prescrib + nonpresc,
                       data = train_data, dist ="negbin")

AIC(hurdle_model)
```

```
## [1] 4980.611
```

```
#summary(hurdle_model)

#hurdle with factors
hurdle_model2 <- hurdle(doctorco ~ income_factor+illness +  actdays+ hospadmi|
                          income:freepoor + actdays *illness + sex*hscore + hospadmi +
                          age_factor*prescrib + nonpresc,
                        data = train_data, dist = "negbin")

AIC(hurdle_model2)
```

```
## [1] 4921.703
```

Key findings from the model coefficients suggest that factors such as income level, illness severity, the number of activity days, hospital admissions, and prescription medication usage significantly influence both the likelihood of making any doctor visit and the frequency of those visits among patients who do.

Notably, the interaction terms, such sex with health score, underscore how the combined effect of these variables can either increase or decrease the likelihood of seeking medical care. For instance, the significant negative coefficient for the interaction between income and freepoor1 suggests that patients from lower-income brackets with access to free poor services are less likely to have zero visits, indicating targeted healthcare access among vulnerable populations.

The theta value, reported as 0.1933 in the count model, is indicative of the degree of overdispersion relative to what a Poisson distribution would predict. A theta value significantly lower than 1 points towards high overdispersion, validating the choice of a negative binomial distribution over a Poisson.

```
predicted_counts_hurdle <- round(predict(hurdle_model2, newdata=test_data, type = "response"))
predicted_category_hnb <- ifelse(predicted_counts_hurdle< 1, 0, predicted_counts_hurdle)


true_counts <- test_data$doctorco
mae_hurdle <- mean(abs(predicted_counts_hurdle- true_counts))
cat("MAE:", mae_hurdle, "\n")
```

```
## MAE: 0.283237
```

```
rmse_hurdle <- sqrt(mean((predicted_counts_hurdle - true_counts)^2))
cat("RMSE:", rmse_hurdle, "\n")
```

```
## RMSE: 0.7602859
```

The MAE indicates a relatively small deviation, suggesting that the model's predictions are, on average, close to the true number of visits. The RMSE, which penalizes larger errors more heavily, is higher, suggesting that there are some instances of larger prediction errors, but overall, the model demonstrates a decent level of accuracy.

Also for the hurdle model, we evaluate alternative metrics for the binary outcome, focusing on the dichotomy of having or not having doctor visits. Employing confusion matrices, balanced accuracy, and AUC-ROC, this analysis aims to assess the model's precision in distinguishing actual visits in a dataset significantly skewed towards non-visits.

```
actual_binary <- ifelse(true_counts > 0, 1, 0)
predicted_binary <- ifelse(predicted_counts_hurdle > 0, 1, 0)
conf_matrix <- table(Actual = actual_binary, Predicted = predicted_binary)
confusionMatrix(as.factor(predicted_binary), as.factor(actual_binary))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 788 124
##          1  62  64
##
##                Accuracy : 0.8208
##                  95% CI : (0.7961, 0.8437)
##     No Information Rate : 0.8189
##     P-Value [Acc > NIR] : 0.4552
##
##                   Kappa : 0.3069
##
##  Mcnemar's Test P-Value : 7.722e-06
##
##             Sensitivity : 0.9271
##             Specificity : 0.3404
##          Pos Pred Value : 0.8640
##          Neg Pred Value : 0.5079
##              Prevalence : 0.8189
##          Detection Rate : 0.7592
##    Detection Prevalence : 0.8786
##       Balanced Accuracy : 0.6337
##
##        'Positive' Class : 0
##
```

```
# Balanced accuracy
balanced_accuracy <- (sensitivity(conf_matrix, positive = "1") +
                        specificity(conf_matrix, positive = "1")) / 2
cat("Balanced Accuracy:", balanced_accuracy, "\n")
```

```
## Balanced Accuracy: 0.5079365
```

```
# AUC-ROC
roc_result <- roc(actual_binary, as.numeric(predicted_binary) - 1)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc_roc <- auc(roc_result)
cat("AUC-ROC:", auc_roc, "\n")
```

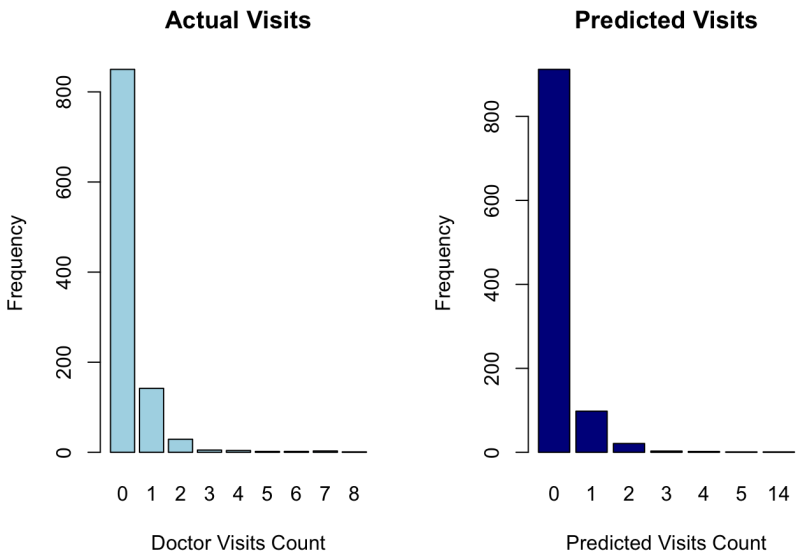
```
## AUC-ROC: 0.6337422
```

The confusion matrix generated from this analysis revealed that the model correctly predicted 788 instances with no visits and 66 instances where visits occurred, against 62 false positives and 122 false negatives. This resulted in a high accuracy rate, a figure slightly higher than the no information rate, indicating the model's predictive capability beyond random chance, albeit with room for improvement, particularly in correctly identifying positive instances.

The model demonstrated a high sensitivity, indicating a strong ability to correctly identify true negatives, but a lower specificity, reflecting challenges in accurately predicting true positives. The balanced accuracy, an average of sensitivity and specificity is suggesting a need to enhance the model's ability to balance both types of correct predictions. The Area Under the Receiver Operating Characteristic curve (AUC-ROC) is showcasing the model's fair discrimination ability between zero and non-zero visit

The model is adept at identifying a significant portion of the non-visits (as evidenced by high sensitivity), it struggles more with accurately predicting actual visits (reflected in lower specificity and NPV). This can happen if the model better captures the zero-inflation aspect but less so the count distribution among the positive outcomes. in the following part we will take a look at how the model predicts the count part of the model.

```
actual_freq <- table(true_counts)
predicted_freq <- table(predicted_counts_hurdle)

par(mfrow=c(1,2))

# Bar plot for Actual Counts
barplot(actual_freq, main="Actual Visits", xlab="Doctor Visits Count",
        ylab="Frequency", col="lightblue")

# Bar plot for Predicted Counts
barplot(predicted_freq, main="Predicted Visits", xlab="Predicted Visits Count",
        ylab="Frequency", col="darkblue")
```




```
par(mfrow=c(1,1))

for (i in 0:9) {
  actual_ <- true_counts == i
  predicted_ <- predicted_counts_hurdle == i
  actual_count <- sum(actual_)
  predicted_count <- sum(predicted_)
  cat("Actual count for", i, "Visits:", actual_count, "\n")
  cat("Predicted count for", i, "Visits:", predicted_count, "\n\n")
}
```

```
## Actual count for 0 Visits: 850
## Predicted count for 0 Visits: 912
##
## Actual count for 1 Visits: 142
## Predicted count for 1 Visits: 98
##
## Actual count for 2 Visits: 29
## Predicted count for 2 Visits: 21
##
## Actual count for 3 Visits: 5
## Predicted count for 3 Visits: 3
##
## Actual count for 4 Visits: 4
## Predicted count for 4 Visits: 2
##
## Actual count for 5 Visits: 2
## Predicted count for 5 Visits: 1
##
## Actual count for 6 Visits: 2
## Predicted count for 6 Visits: 0
##
## Actual count for 7 Visits: 3
## Predicted count for 7 Visits: 0
##
## Actual count for 8 Visits: 1
## Predicted count for 8 Visits: 0
##
## Actual count for 9 Visits: 0
## Predicted count for 9 Visits: 0
```

The hurdle model shows a good ability to predict no doctor visits, with a prediction slightly higher than the actual numbers. For one visit, the model underpredicts, indicating some difficulty in accurately forecasting lower visit counts. This trend of underprediction continues for two visits and becomes more noticeable for higher visit counts, with the model predicting fewer visits than actually occurred, and failing to predict any instances of five or more visits, except for a single prediction for seven visits. This pattern suggests that while the hurdle model can effectively identify cases with no visits, its performance in predicting actual visit counts, especially for rarer higher visit counts, is limited.

Zero Inflated VS Hurdle

Both models demonstrate strength in predicting no visits, with the hurdle model predicting slightly more no-visit cases than the zero-inflated model. However, when it comes to predicting actual visits, both models struggle with higher counts, underestimating the actual occurrences. The zero-inflated model appears to provide a closer approximation for one visit but also fails to predict visits of five or more. In contrast, the hurdle model underpredicts across most visit counts more significantly, including one visit, and barely predicts higher visit counts.

Zero-inflated and hurdle models address excess zeros in count data differently. The ZINB model is particularly useful when the data include both ‘structural zeros’—instances where no visits occur due to lack of necessity or access—and ‘sampling zeros,’ where visits could have occurred but did not. This model separates the data into two processes: one that models the probability of excess zeros and another that models the count of visits among those expected to have them, using a negative binomial distribution to account for overdispersion.

The HNB model treats all zeros as coming from a single process but separates the analysis into two stages: a binary outcome predicting the occurrence of any visits and a truncated count model for the number of visits among those who have at least one. This approach is effective when the focus is on distinguishing between non-use and use of healthcare services. The interpretation of a hurdle model is more straightforward, focusing on the hurdle of initiating healthcare service use before addressing the frequency of use among those who cross that hurdle.

The fact that the ZINB model has a lower AIC indicates that it provides a better fit to the data, suggesting that the additional complexity of separating the zero observations into those that are structurally zero and those that are zeros due to sampling is justified by the data. The lower MAE suggests that the ZINB model is more accurate in predicting the actual number of doctor visits, including accurately predicting the absence of visits. It indicates that a significant portion of the zero visits can be attributed to individuals who are not just non-users of healthcare services by chance but are systematically different from those who do visit doctors.

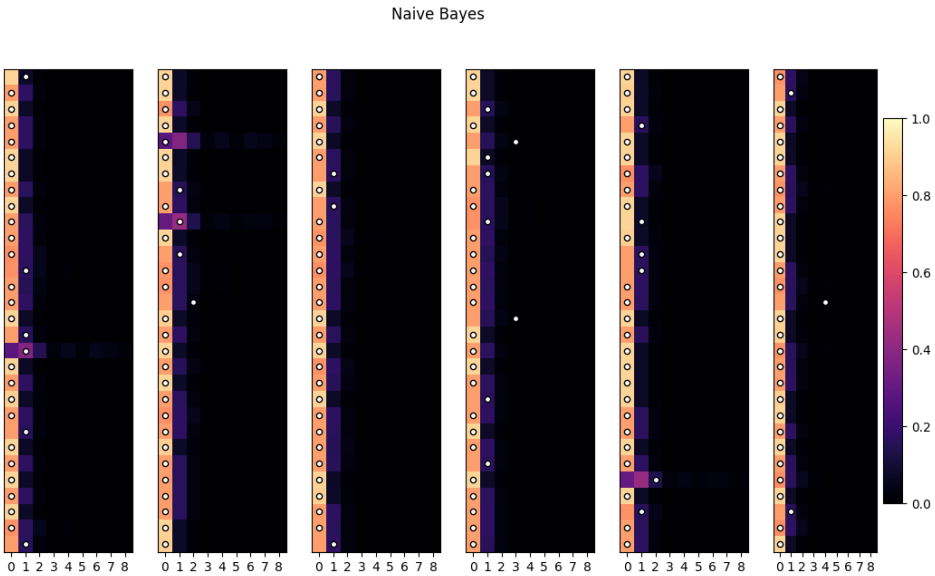
The choice of the Negative Binomial distribution over the Poisson distribution for both models is crucial due to the observed overdispersion in the data—where. The Negative Binomial distribution introduces an additional parameter to model the variance, providing a more flexible and accurate fit for count data that cannot be adequately modeled by the Poisson distribution's equal mean and variance assumption.

#Naive Bayes

The Naive Bayes model is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions. It is a popular method for text classification, but it can be used for any type of data. The model is trained by estimating the probability of each class given the input data, and then it uses these probabilities to predict the class of new data points. The Naive Bayes model is known for its simplicity and speed, and it is often used as a baseline for comparison with more complex models.

The model is based on the assumption that the features are independent. In our case, as shown by our analysis, this assumption is not realistic, but we are still interested in the performance of the Naive Bayes model as a baseline for comparison with more complex models.

```
def training(verbose=False, plot=False, random_state=None, title='Naive Bayes'):
    data = Data()
    data.x_to_one_hot()
    data.y_to_one_hot()

    # keep only some columns
    names = ['actdays_0', 'actdays_10', 'actdays_14', 'age_0.27',
             'hscore_0', 'illness_0', 'income_0.55']

    data.keep_cols(names)

    x_train, x_test, y_train, y_test = data.train_test_split(random_state=random_state)

    y_train_values = np.argmax(y_train.values, axis=1)

    model = MultinomialNB()
    model.fit(x_train, y_train_values)
    y_pred = model.predict_proba(x_test)

    return evaluate(y_train, y_test, y_pred, verbose=verbose, plot=plot, title=title)
```




```
Performance

avg_rmse           0.190
avg_mabse          0.292
std_rmse           0.004
std_mabse          0.026
avg_dummy_rmse     0.194
avg_dummy_mabse    0.303
avg_training_time  0.079 s
std_training_time  0.004 s
```

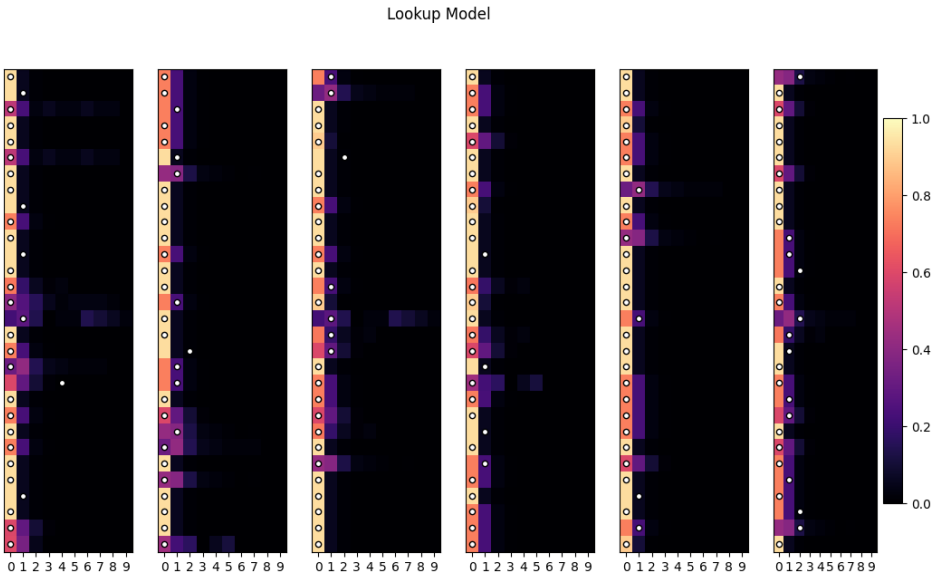
These results have been obtained on a subsample of features, carefully selected to maximize the performance of the model.

Lookup model

This very simple model makes prediction by averaging the values of the target variables that match the features of the input data-point. Despite it's simplicity, it can be a good baseline to compare with more complex models.

This model's weakness is that it can't predict unseen data-points, as it requires the exact same features to be present. Therefore, it is unable to generalize, and it can handle only few features at the time.

```
class Model(dict):

    def train(self, x_train, y_train,
              names=('actdays_0', 'actdays_14',
                     'prescrib_0', 'hospdays_0', 'hospadmi_0')):
        self['names'] = names
        self['y_size'] = y_train.shape[1]
        self['x_train'] = x_train
        self['y_train'] = y_train

    def predict(self, x_test):
        """
        The model outputs the average y_train of the x_train data-points
        with the correct values for the features
        """
        out = np.zeros((len(x_test), self['y_size']))
        names = self['names']
        for i, (index, x) in enumerate(x_test.iterrows()):
            indices = np.ones(len(self['x_train']), dtype=bool)
            for name in names:
                indices *= self['x_train'][name] == x[name]
            out[i] = self['y_train'][indices].mean(axis=0)
        return out


def training(verbose=False, plot=False, random_state=None, title='Lookup Model'):
    data = Data()
    data.x_to_one_hot()
    data.y_to_one_hot()

    x_train, x_test, y_train, y_test = data.train_test_split(random_state=random_state)

    model = Model()
    model.train(x_train, y_train)
    y_pred = model.predict(x_test)

    return evaluate(y_train, y_test, y_pred,
                    verbose=verbose, plot=plot, title=title)
```



Lookup model examples from the test set

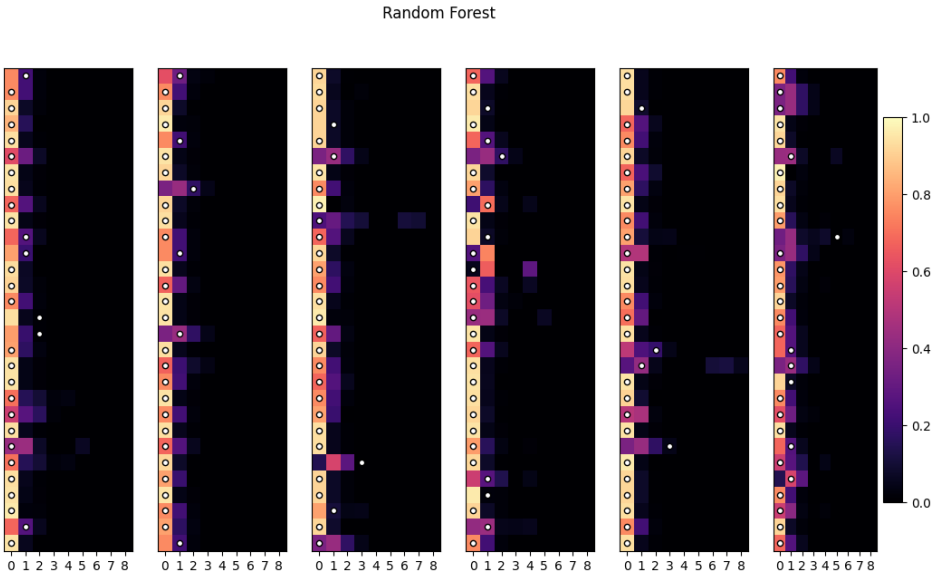
```
Performance

avg_rmse           0.173
avg_mabse          0.288
std_rmse           0.004
std_mabse          0.021
avg_dummy_rmse     0.184
avg_dummy_mabse    0.296
avg_training_time  0.905 s
std_training_time  0.039 s
```

An extensive search for the best features shows that the following dummy features yield a model with comparatively decent performance: *actdays==0, actdays==14, prescrib==0, hospdays==0, hospadmi==0*.

Random Forest

The random forest is an ensemble method that aggregates the predictions of several individual decision trees. Each tree is trained on a random subset of the data. Random forest uses a technique called bagging, that trains independently multiple decision trees by randomly sampling the training data with replacement.

```
def training(verbose=False, plot=False, random_state=None, title='Random Forest'):
    data = Data()
    data.x_to_one_hot()
    data.y_to_one_hot()

    # keep only some columns
    names = ['actdays_0', 'actdays_14', 'actdays_6', 'age_0.22',
             'age_0.32', 'age_0.42', 'age_0.52', 'hospadmi_0',
             'hospadmi_1', 'hscore_0', 'hscore_8', 'income_0.06',
             'prescrib_0']
    data.x = data.x[names]

    x_train, x_test, y_train, y_test = data.train_test_split(random_state=random_state)

    y_train_values = np.argmax(y_train.values, axis=1)

    model = RandomForestClassifier(random_state=1)
    model.fit(x_train, y_train_values)

    y_pred = model.predict_proba(x_test)

    return evaluate(y_train, y_test, y_pred, verbose=verbose, plot=plot, title=title)
```



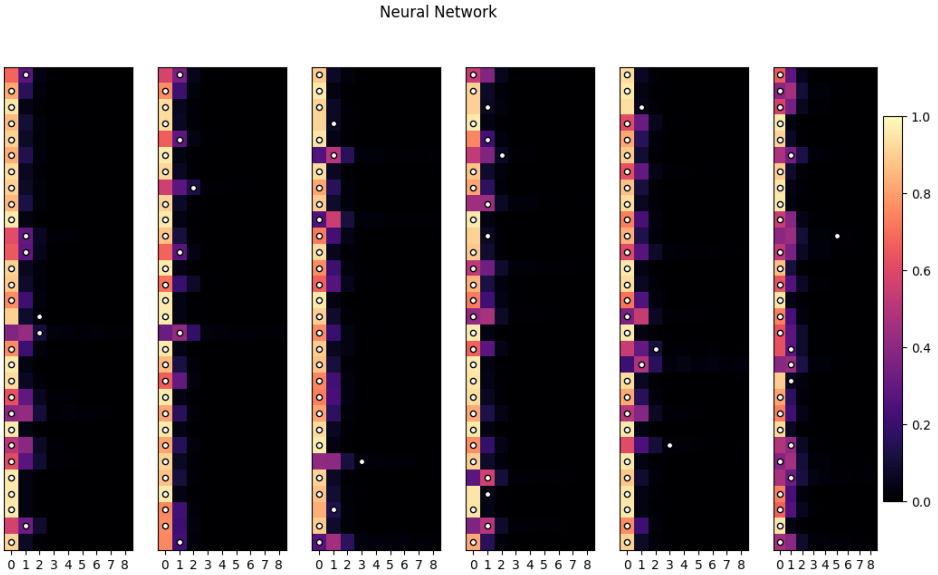
Random forest examples from the test set

```
Performance

avg_rmse           0.186
avg_mabse          0.298
std_rmse           0.003
std_mabse          0.019
avg_dummy_rmse     0.194
avg_dummy_mabse    0.303
avg_training_time  0.187 s
std_training_time  0.006 s
```

Also for this model, features were carefully selected, as the model tends to learn spurious patterns from the limited data provided.

Neural Network

```
class Model(dict):
    """naive bayes"""
    def fit(self, x_train, y_train, random_state=42, verbose=True):
        y_train_values = np.argmax(y_train, axis=1)
        self['model'] = MLPClassifier(hidden_layer_sizes=(200, 100),
                                      activation="relu",
                                      alpha=0.001,
                                      learning_rate_init=1e-5,
                                      max_iter=500,
                                      random_state=random_state,
                                      tol=1e-4,
                                      verbose=verbose,
                                      n_iter_no_change=20,
                                      beta_1=0.99,
                                      beta_2=0.9)
        self['model'].fit(x_train, y_train_values)

    def predict(self, x_test):
        return self['model'].predict_proba(x_test)


def training(verbose=False, plot=False, random_state=None, title='Neural Network'):
    data = Data()
    data.keep_cols(['actdays', 'prescrib', 'hospadmi', 'illness', 'hscore', 'nondocco'])
    data.x_to_one_hot()
    data.y_to_one_hot()

    x_train, x_test, y_train, y_test = data.train_test_split(random_state=random_state)

    model = Model()
    model.fit(x_train, y_train, random_state=random_state, verbose=verbose)

    y_pred = model.predict(x_test)

    return evaluate(y_train, y_test, y_pred, verbose=verbose, plot=plot, title=title)
```



Neural network examples from the test set

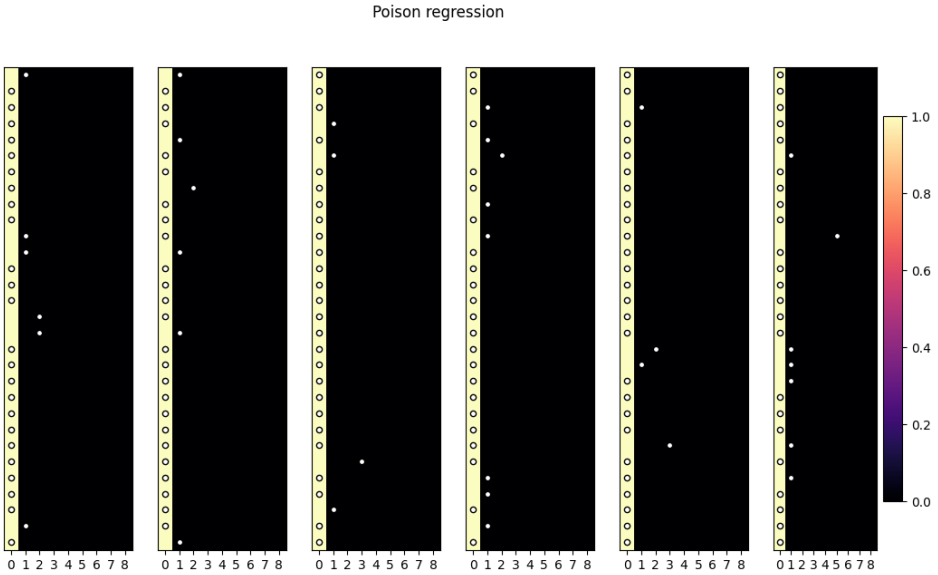
```
Performance

avg_rmse           0.182
avg_mabse          0.287
std_rmse           0.003
std_mabse          0.019
avg_dummy_rmse     0.195
avg_dummy_mabse    0.305
avg_training_time  44.66 s
std_training_time  9.656 s
```

Neural networks were able to reasonably minimize the mean absolute error of the prediction, even without a careful feature selection. In fact, the features used by this model are simply the ones with highest correlation with the target variable.

Poisson regression

Poisson regression is a statistical method used to model response variables that involve count data. It helps revealing which explanatory variables impact the target variable, making it particularly useful. It has been extensively researched and refined over time to address various practical scenarios.

```
class Model(dict):
    """naive bayes"""
    def fit(self, x_train, y_train, random_state=42, verbose=True):
        y_train_values = np.argmax(y_train, axis=1)
        self['model'] = PoissonRegressor(max_iter=1000)
        self['model'].fit(x_train, y_train_values)

    def predict(self, x_test):
        y_pred = self['model'].predict(x_test)
        y_pred = np.clip(y_pred, 0, 8)

        # convert y_pred to one-hot
        y_pred = np.eye(9)[y_pred.astype(int)]

        return y_pred


def training(verbose=False, plot=False, random_state=None, title='Poisson regression'):
    data = Data()
    data.keep_cols(['actdays', 'prescrib', 'hospadmi',
                    'illness', 'hscore', 'nondocco'])
    data.x_to_one_hot()
    data.y_to_one_hot()

    x_train, x_test, y_train, y_test = data.train_test_split(random_state=random_state)

    model = Model()
    model.fit(x_train, y_train, random_state=random_state, verbose=verbose)

    y_pred = model.predict(x_test)

    return evaluate(y_train, y_test, y_pred,
                    verbose=verbose, plot=plot, title=title)
```



Poisson examples from the test set

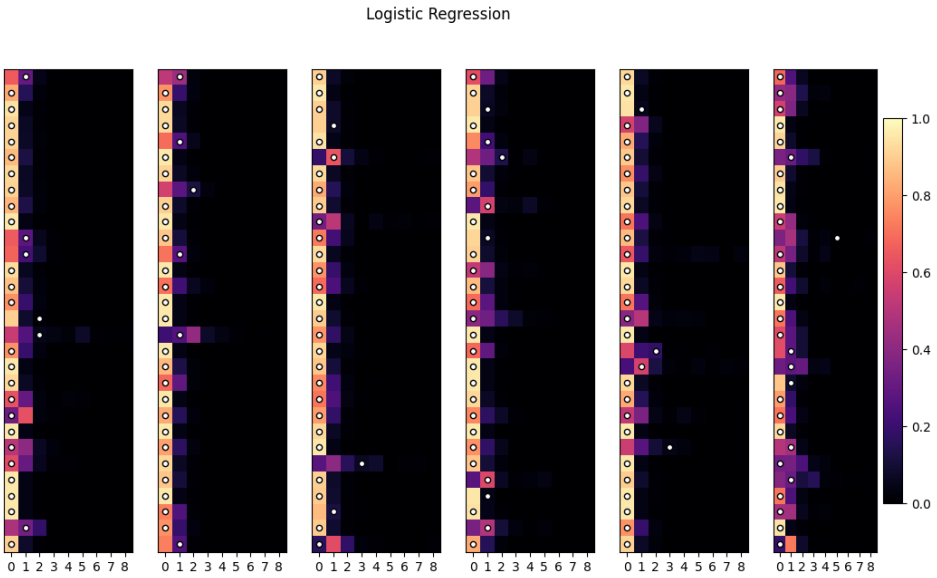
```
Performance

avg_rmse           0.212
avg_mabse          0.302
std_rmse           0.006
std_mabse          0.029
avg_dummy_rmse     0.194
avg_dummy_mabse    0.302
avg_training_time  0.038 s
std_training_time  0.025 s
```

The Poisson regression on its own fails to make useful predictions due to the unbalanced nature of the data.

Logistic regression

Logistic regression is a statistical model commonly used for classification and predictive analytics. It estimates the probability of an event occurring based on a given dataset of independent variables. Logistic regression focuses on predicting binary outcomes, but can be extended to handle multiclass classification.

```
class Model(dict):
    """naive bayes"""
    def fit(self, x_train, y_train, random_state=42, verbose=True):
        y_train_values = np.argmax(y_train, axis=1)
        self['model'] = LogisticRegression(random_state=random_state, max_iter=1000)
        self['model'].fit(x_train, y_train_values)

    def predict(self, x_test):
        return self['model'].predict_proba(x_test)


def training(verbose=False, plot=False, random_state=None, title='Logistic Regression'):
    data = Data()
    data.keep_cols(['actdays', 'prescrib', 'hospadmi', 'illness', 'hscore', 'nondocco'])
    data.x_to_one_hot()
    data.y_to_one_hot()

    x_train, x_test, y_train, y_test = data.train_test_split(random_state=random_state)

    model = Model()
    model.fit(x_train, y_train, random_state=random_state, verbose=verbose)

    y_pred = model.predict(x_test)

    return evaluate(y_train, y_test, y_pred, verbose=verbose, plot=plot, title=title)
```



Logistic examples from the test set

```
Performance

avg_rmse           0.181
avg_mabse          0.289
std_rmse           0.005
std_mabse          0.026
avg_dummy_rmse     0.195
avg_dummy_mabse    0.305
avg_training_time  0.172 s
std_training_time  0.018 s
```

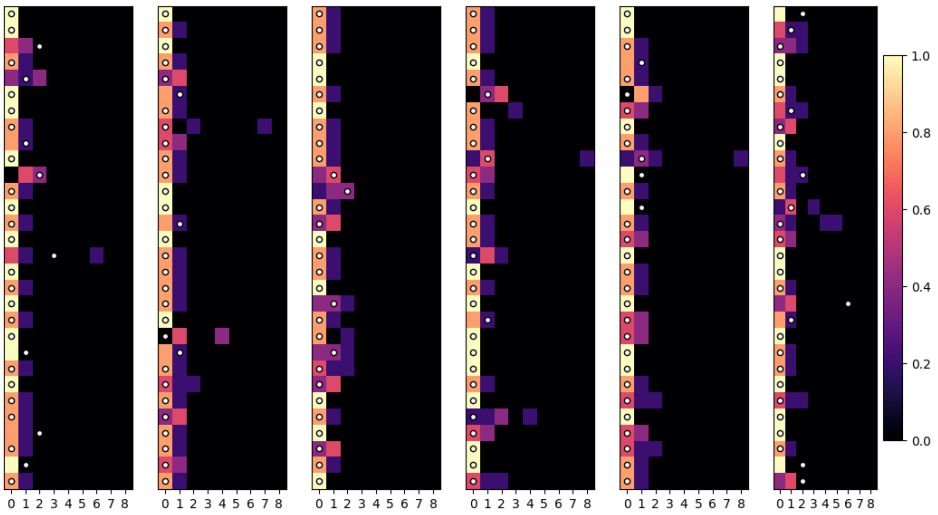
MCA-KNN

The MCA-KNN model is a combination of two methods: Multiple Correspondence Analysis (MCA) and K-Nearest Neighbors (KNN).

The purpose of MCA is to reduce the dimensionality the data. It is analogous to PCA, but specifically designed for categorical data. The output of MCA is a latent representation of the data, which is then used as input to the KNN model. In this form, the data points belonging to different classes are expected to be more separable.

The KNN model is a non-parametric method used for classification. It is based on the idea that similar data points are likely to belong to the same class. The model is trained by storing the training data, and then it classifies new data points based on their similarity to the training data. The model is simple and intuitive, but it can be sensitive to the choice of the number of neighbors and the distance metric. A big number of neighbors “k” can lead to underfitting, especially for categories with a low number of samples. On the other hand, a small number of neighbors can lead to overfitting.

```
class Model(dict):
    """naive bayes"""
    def fit(self, x_train, y_train, n_neighbors=10,
            n_components=5, random_state=42, verbose=True):

        # learn a function to reduce dimensionality
        self['mca'] = MCA(n_components=n_components,
                          copy=True,
                          check_input=True,
                          engine='sklearn',
                          random_state=random_state,
                          one_hot=False
                          )

        self['mca'] = self['mca'].fit(x_train)
        reduced_x = self['mca'].row_coordinates(x_train).to_numpy()

        self['model'] = KNeighborsClassifier(n_neighbors=n_neighbors,
                                             weights="uniform",
                                             algorithm="auto",
                                             leaf_size=30,
                                             p=2,
                                             metric="minkowski")

        y_train_values = np.argmax(y_train, axis=1)

        self['model'].fit(reduced_x, y_train_values)

    def predict(self, x_test):
        reduced_data = self['mca'].row_coordinates(x_test).to_numpy()
        y_pred = self['model'].predict_proba(reduced_data)
        return y_pred


def training(verbose=False, plot=False, random_state=None, title='MCA-KNN',
             n_neighbors=5, n_components=10, n_ones_min=5):
    data = Data()
    data.keep_cols(['actdays', 'prescrib', 'hospadmi', 'illness', 'nondocco'])
    data.x_to_one_hot()
    data.y_to_one_hot()

    # remove columns with not enough 1s
    remove_cols = data.x.columns[(data.x.sum() <= n_ones_min)]
    data.remove_cols(remove_cols)

    if verbose:
        print(f'Columns with less than {n_ones_min} ones removed:')
        print(remove_cols)

    x_train, x_test, y_train, y_test = data.train_test_split(random_state=random_state)

    model = Model()
    model.fit(x_train, y_train,
              n_neighbors=n_neighbors, n_components=n_components,
              random_state=random_state, verbose=verbose)

    y_pred = model.predict(x_test)

    return evaluate(y_train, y_test, y_pred, verbose=verbose, plot=plot, title=title)
```



MCA-KNN examples from the test set

```
Performance

avg_rmse           0.189
avg_mabse          0.297
std_rmse           0.005
std_mabse          0.028
avg_dummy_rmse     0.194
avg_dummy_mabse    0.303
avg_training_time  0.102 s
std_training_time  0.038 s
```

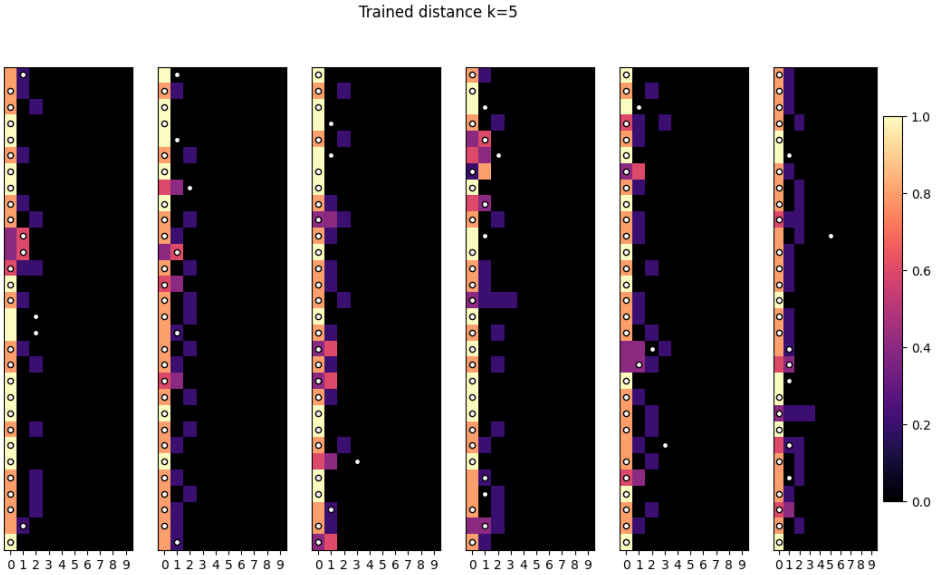
This method could benefit from a better latent representation of the data. The main limitation in our case is that not many options exist when dealing with mostly categorical data. That is exactly what the next method tries to implement.

Trained distance

This method consists on using neural networks to learn a distance between two data points, based on the difference between the values of the respective target features. During prediction, the new data point is compared to all the training data points, and the k closest data points are used to calculate the average of the target features.

```
class Model(dict):
    """Trained proximity knn model
    the model learns a distance between two data points
    """

    def fit(self, x_train, y_train,
            random_state=42, data_size=20_000, epsilon=0.1):
        print('\nBuilding dataset\n')
        self['x_train'] = x_train
        self['y_train_onehot'] = y_train
        y_train_values = np.argmax(y_train, axis=1)
        self['y_train_values'] = y_train_values

        # create an empty matrix with the same number of columns as x_train
        x_diff = np.zeros((0, x_train.shape[1]))
        y_diff = []

        for _ in range(data_size):
            if _ % 10_000 == 0:
                print(f'building dataset {_}/{data_size}')
            i = np.random.randint(0, x_train.shape[0])
            j = np.random.randint(0, x_train.shape[0])

            if _ % 10 == 0:
                i = j

            x_i, x_j = x_train.iloc[i], x_train.iloc[j]
            x_abs_diff = np.abs(x_i - x_j)
            y_abs_diff = np.abs(y_train_values[i] - y_train_values[j])

            x_diff = np.vstack((x_diff, x_abs_diff))
            y_diff.append(y_abs_diff)

        y_diff = np.array(y_diff)
        y_diff = np.log(y_diff + epsilon)

        x_diff_train = pd.DataFrame(x_diff, columns=x_train.columns)
        y_diff_train = pd.DataFrame(y_diff, columns=['y_diff'])
        y_diff_train = np.ravel(y_diff_train)

        print(f'\nTraining model with {data_size} samples\n')
        # the model learns the distance between two data points
        self['model'] = MLPRegressor(hidden_layer_sizes=(200, 100),
                                     activation="relu",
                                     batch_size="auto",
                                     learning_rate="constant",
                                     max_iter=200,
                                     random_state=random_state,
                                     tol=1e-3,
                                     verbose=True
                                     )
        self['model'].fit(x_diff_train, y_diff_train)

    def predict(self, x_test, k=5):
        """
        check each row in x_test against each row in x_train, and
        return the average of y_train of the closest data points
        """
        y_pred = []
        y_train_onehot = self['y_train_onehot'].to_numpy()

        for i in range(x_test.shape[0]):

            if i % 100 == 0:
                print(f'predicting {i}/{x_test.shape[0]}')

            # compute the distance (y_proba) between x_test[i]
            # and each x_train[j]
            x_i = x_test.iloc[i]
            x_diff = np.abs(self['x_train'] - x_i)
            x_diff = pd.DataFrame(x_diff, columns=x_test.columns)
            y_proba = self['model'].predict(x_diff)

            # join the distance with the y_train_values, then sort by distance,
            # take the k closest data points
            v = list(zip(y_proba, y_train_onehot))
            v.sort(key=lambda x: x[0])
            v = v[:k]

            # take the average of the k closest data points
            new_y = np.mean([x[1] for x in v], axis=0)
            y_pred.append(new_y)

        return np.array(y_pred)


def training(verbose=False, plot=False, title='Trained distance', k=5):
```

```
    data = Data()

    # keep only some columns
    data.keep_cols(['actdays', 'prescrib', 'hospadmi', 'illness', 'hscore'])

    # normalize data
    for name in data.x.columns:
        data.x[name] /= data.x[name].max()

    data.x_to_one_hot()
    data.y_to_one_hot()

    x_train, x_test, y_train, y_test = data.train_test_split()

    model = Model()
    model.fit(x_train, y_train)

    y_pred = model.predict(x_test, k=k)

    return evaluate(y_train, y_test, y_pred, verbose=verbose,
                    plot=plot, title=title + ' k=' + str(k))
```



examples from the test set

```
Performance

avg_rmse           0.197
avg_mabse          0.350
std_rmse           0.003
std_mabse          0.004
avg_dummy_rmse     0.193
avg_dummy_mabse    0.350
avg_training_time  84.11 s
std_training_time  15.50 s
```

The poor performance of this method can be attributed to the fact that the noise in the data makes it really hard for the neural network to learn a meaningful distance among data points.

References:

- https://www.health.gov.au/medicare-turns-40/history# (https://www.health.gov.au/medicare-turns-40/history#):~:text=The%20Australian%20Government%2C%20under%20Prime,Australian%20Labor%20Party%20formed%20government
- https://www.nma.gov.au/defining-moments/resources/medicare# (https://www.nma.gov.au/defining-moments/resources/medicare#):~:text=The%20incoming%20Fraser%20government%20modified,had%20blocked%20while%20in%20opposition
- Young DS, Roemmele ES, Yeh P. Zero-inflated modeling part I: Traditional zero-inflated count regression models, their applications, and computational tools. WIREs Comput Stat. 2022; 14:e1541. https://doi.org/10.1002/wics.1541 (https://doi.org/10.1002/wics.1541)
- Lambert, Diane. 1992. “Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing.” Technometrics 34 (1).
- Fletcher, David & MacKenzie, Darryl & Villouta Stengl, Eduardo. (2005). Modelling skewed data with many zeros: A simple approach combining ordinary and logistic regression. Environmental and Ecological Statistics. 12. 45-54. 10.1007/s10651-005-6817-1 (https://www.researchgate.net/publication/226071827_Modelling_skewed_data_with_many_zeros_A_simple_approach_combining_ordinary_and_logistic_reg (https://www.researchgate.net/publication/226071827_Modelling_skewed_data_with_many_zeros_A_simple_approach_combining_ordinary_and_logistic_reg
- Farbmacher, Helmut. “Estimation of hurdle models for overdispersed count data.” The Stata Journal 11.1 (2011): 82-94.
- Ana Gonzalez-Blanks, Jessie M. Bridgewater & Tuppett M. Yates (2020) Statistical Approaches for Highly Skewed Data: Evaluating Relations between Maltreatment and Young Adults’ Non-Suicidal Self-injury, Journal of Clinical Child & Adolescent Psychology, 49:2, 147-161, DOI: 10.1080/15374416.2020.1724543

- Dixit SK, Sambasivan M. A review of the Australian healthcare system: A policy perspective. SAGE Open Medicine. 2018;6. doi:10.1177/2050312118769211 (doi:10.1177/2050312118769211)
- C. Ford, 2016. “Getting started with Hurdle Models.” UVA Library, StatLab. https://library.virginia.edu/data/articles/getting-started-with-hurdle-models (https://library.virginia.edu/data/articles/getting-started-with-hurdle-models)
- Feng, C.X. A comparison of zero-inflated and hurdle models for modeling zero-inflated count data. J Stat Distrib App 8, 8 (2021). https://doi.org/10.1186/s40488-021-00121-4 (https://doi.org/10.1186/s40488-021-00121-4)
- Packages used: ggplot2, Hmisc, mgcv, pROC, PRROC, pscl, reticulate, ROSE, tidyverse, base, datasets, dplyr, forcats, graphics, grDevices, lattice, lubridate, MASS, methods, nlme, purrr, readr, rlang, stats, stringr, tibble, tidyr, utils.