House Rent Analysis

- Project Report -

Data Preparation

I have introduced the dataset `munichrent03` which is integrated in the R package `LinRegInteractive`, available at **README.md** file. Therefore, I can simply load this dataset to begin the subsequent steps of the analysis.

```
library(LinRegInteractive)
data(munichrent03)
data <- munichrent03
```

I began by examining the variable types to understand the structure of the dataset.

```
> str(data)
'data.frame':   2053 obs. of  12 variables:
$ rent     : num  741 716 528 554 698 ...
$ rentsqm  : num  10.9 11.01 8.38 8.52 6.98 ...
$ area     : int  68 65 63 65 100 81 55 79 52 77 ...
$ rooms    : int  2 2 3 3 4 4 2 3 1 3 ...
$ yearc    : num  1918 1995 1918 1983 1995 ...
$ bathextra: Factor w/ 2 levels "no","yes": 1 1 1 2 2 1 2 1 1 1 ...
$ bathtile : Factor w/ 2 levels "yes","no": 1 1 1 1 1 1 1 1 1 1 ...
$ cheating : Factor w/ 2 levels "yes","no": 1 1 1 1 1 1 1 1 1 1 ...
$ district : Factor w/ 25 levels "All-Umenz","Alt-Le",..: 10 10 10 17 17 17 21 21 21 21 ...
$ location : Ord.factor w/ 3 levels "normal"<"good"<..: 2 2 2 1 2 1 1 1 1 1 ...
$ upkitchen: Factor w/ 2 levels "no","yes": 1 1 1 1 2 1 1 1 1 1 ...
$ wwater   : Factor w/ 2 levels "yes","no": 1 1 1 1 1 1 1 1 1 1 ...
> names(data)
'rent''rentsqm''area''rooms''yearc''bathextra''bathtile''cheating''district''location''upkitchen''wwater'
```

After that, I reviewed the distributions, value ranges, and identified any potential missing values.

```
> summary(data)
      rent            rentsqm            area           rooms
Min.   :  77.31   Min.   : 1.470   Min.   : 17.0   Min.   :1.000
1st Qu.: 389.95   1st Qu.: 6.800   1st Qu.: 53.0   1st Qu.:2.000
Median : 534.30   Median : 8.470   Median : 67.0   Median :3.000
Mean   : 570.09   Mean   : 8.394   Mean   : 69.6   Mean   :2.598
3rd Qu.: 700.48   3rd Qu.:10.090   3rd Qu.: 83.0   3rd Qu.:3.000
Max.   :1789.55   Max.   :20.090   Max.   :185.0   Max.   :6.000

yearc      bathextra  bathtile   cheating        district      location
Min.   :1918   no :1862   yes:1673   yes:1878   Neuh-Nymp: 177   normal:1205
1st Qu.:1948   yes: 191   no : 380   no : 175   Lud-Isar : 161   good  : 803
Median :1960                                    Au-Haid  : 139   top   :  45
Mean   :1958                                    SchwWest : 137
3rd Qu.:1973                                    Maxvor   : 132
Max.   :2001                                    Laim     : 117
(Other)  :1190
upkitchen  wwater
no :1903   yes:1981
yes: 150   no :  72
```

Note that the variables `rentsqm, rent` and `area` are related by the equation: `rent` = `rentsqm` × `area`. For example, at row 100, we have:

```
> round(data$rent[100]/data$area[100],2)  #compute rentsqm
11.3
> data$rentsqm[100]
11.3
```

Since `rentsqm` is a derived variable, I chose to exclude it and instead focus on total `rent`, which may better capture the underlying relationships with other features.

```
> data$rentsqm <- NULL
```

Exploratory Data Analysis

Subsequently, I conducted an Exploratory Data Analysis (EDA) to gain initial insights into the dataset. A few key findings from this phase include:

- The 7 districts with the largest number of houses

```
Neuh-Nymp  Lud-Isar   Au-Haid  SchwWest    Maxvor      Laim    Ram-Per
177        161        139       137        132         117        115
```

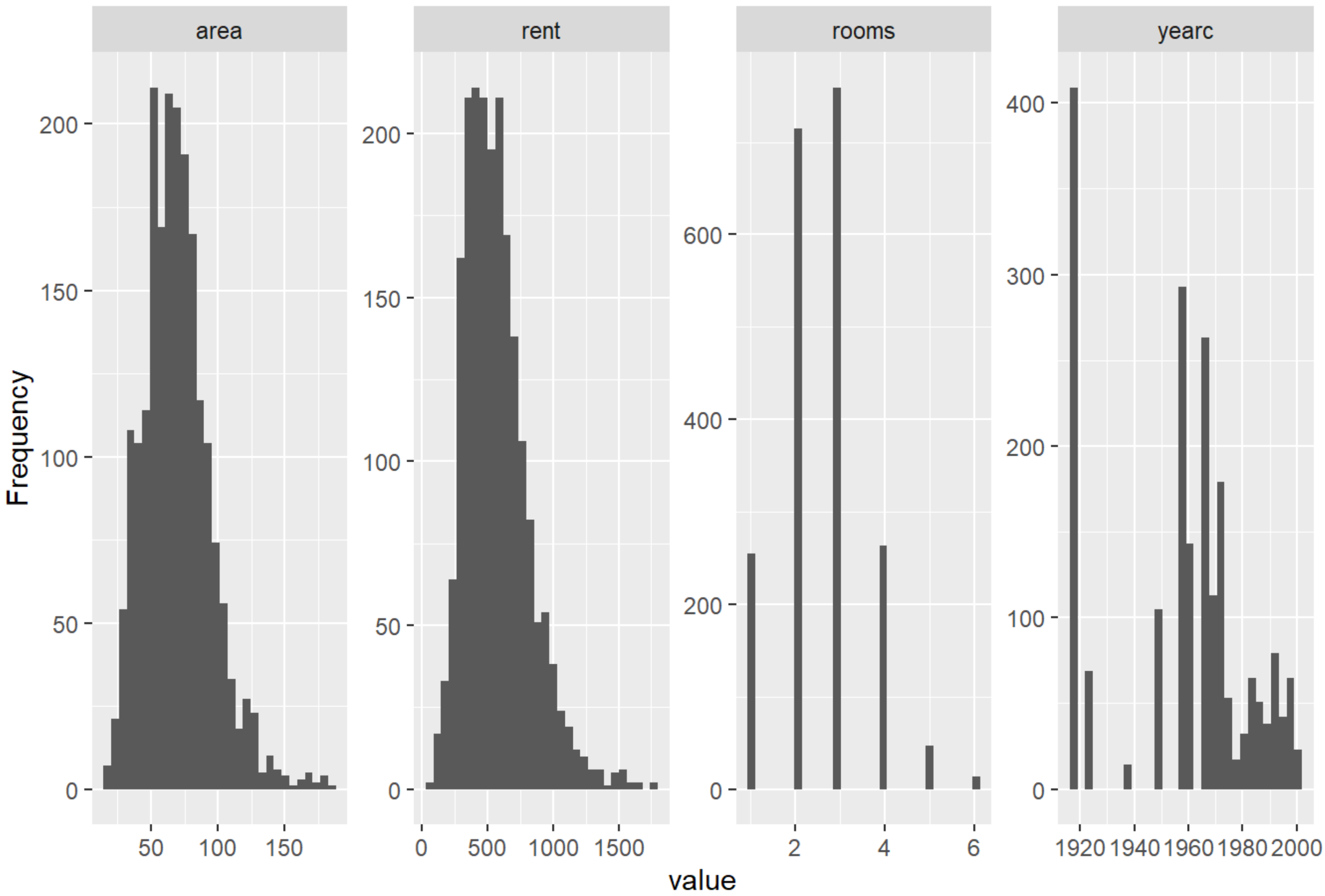
- The 7 districts with the highest number of houses where the location is classified as either “top” or “good”:

```
Maxvor    Neuh-Nymp    Lud-Isar     SchwWest      Au-Haid Schwab-Frei     Trud-Rie
118          116          97           96            81          50           37
```

- The proportion of houses equipped with each feature:

```
    Feature Count Percent
1 bathextra   191     9.3
2  bathtile  1673    81.5
3  cheating  1878    91.5
4 upkitchen   150     7.3
5    wwater  1981    96.5
```

- The distribution of the numerical variables in the dataset



- The variable `yearc` represents discrete individual years, and the number of rooms(`room`)ranges only from 1 to 6.

```
> table(data$room) #Count for each number of room

1   2   3   4   5   6
255 715 759 263  47  14
> table(data$yearc) #Count for each year

1918   1924   1939   1948   1957 1957.5   1960   1966   1967   1968
409      69     14    105    225     68    143    228     35     23
1969   1970   1971   1972   1973   1974   1975   1976   1977   1978
44     46     35     89     55     30     16      7      6      5
1979   1980   1981   1982   1983   1984   1985   1986   1987   1988
6     17     15      8     40     17     20     11     20     13
1989   1990   1991   1992   1993   1994   1995   1996   1997   1998
15     10     14     24     41     13      9     20     12     14
1998.5   1999   2000   2001
32      7     18      5
```

Graphical Model Learning

More Data Preparation

To learn a model from the data, some additional preprocessing steps are required. From the earlier exploration, I observed that the dataset contains many categorical variables, and only two variables, namely rent and area, are truly continuous. A Directed Graphical Model can be used to model data under two common distributional assumptions:

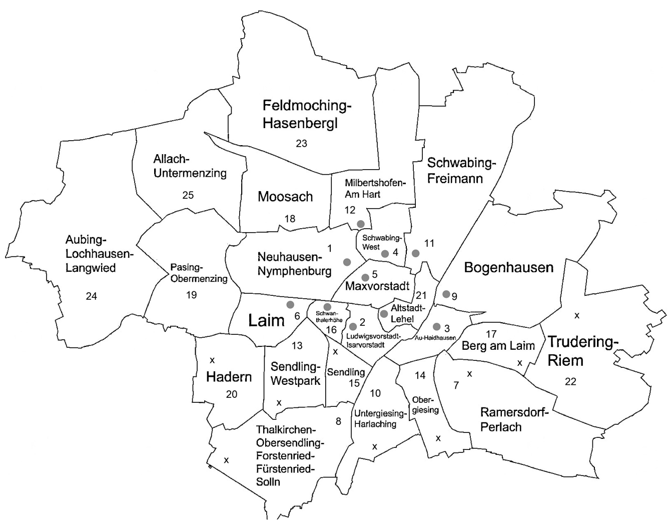
- Multinomial (each variable is categorical)
- Multivariate Normal

Since the dataset contains mostly categorical variables, I decided to model the data assuming a Multinomial distribution. Therefore, several variables needed to be transformed accordingly. Below is a summary of the preprocessing tasks (see the R code file for more details):

- For the variable `yearc`, there are two values with decimals: `1957.5` and `1998.5`. I removed the decimal part to keep only the integer year. After that, I created a new variable called `year_group` to extract information from `yearc` with fewer levels. Specifically, the years were grouped into the following periods: 1918–1959, 1960–1969, 1970–1989, and 1990–2001.

  ```
  # Count for each periods
  1918-1959 1960-1969 1970-1989 1990-2001
  890       473       471       219
  ```

- The variable `district` represents the 25 districts of Munich. If treated directly as a categorical variable, it would result in 25 levels, which would make computation extremely complex. To address this, I created a new variable called `district_group`, which clusters the 25 districts into 3 broader areas. The grouping is based on the density of houses and geographical location. The figure below illustrates this grouping: the districts are sorted in descending order based on the number of houses. Region 1 includes the districts marked with a bold dot. Region 2 includes those marked with an "x", and Region 3 consists of the remaining districts.



Once the districts were grouped into three areas, the following shows the distribution of houses across the areas.

```
> table(data$district_group)

Area1 Area2 Area3
1214   658   181
```

- Next, the variable `rooms`, which indicates the number of rooms in each house and only takes values from 1 to 6, was converted into a factor and treated as a categorical variable. The variable `location` is an ordered factor by default, so I converted it into a regular (unordered) factor to ensure compatibility with the modeling functions.

```
> data$rooms <- as.factor(data$rooms)
> data$location <- as.factor(as.character(data$location))
```

- Now, only two continuous variables remain: `rent` and `area`. To model the data under the Multinomial distribution, I applied a discretization method to convert these variables into discrete ones, aiming to preserve as much information as possible. Specifically, the discretization technique used is called information-preserving discretization, also known as the `Hartemink` method. The details of this transformation are presented in the code file. For the sake of brevity, the method itself will not be discussed in this report.

  The function that performs this task from the package `bnlearn` directly transforms the two mentioned continuous variables into discrete ones, without creating new variables. The `rent` and `area` variables have now been transformed into discrete variables, each divided into a set of intervals.

```
> table(data_disc$rent)

[77.31,406.132] (406.132,687.072] (687.072,1789.55]
582               923               548
> table(data_disc$area)

[17,44]  (44,78] (78,185]
307     1086      660
```

  The dataset after applying all the transformations described above is named `data_disc`, and its structure is as follows:

```
> str(data_disc)
'data.frame':   2053 obs. of  11 variables:
$ rent          : Factor w/ 3 levels "[77.31,406.132]",..: 3 3 2 2 3 3 1 2 2 1 ...
$ area          : Factor w/ 3 levels "[17,44]","(44,78]",..: 2 2 2 2 3 3 2 3 2 2 ...
$ rooms         : Factor w/ 6 levels "1","2","3","4",..: 2 2 3 3 4 4 2 3 1 3 ...
$ bathextra     : Factor w/ 2 levels "no","yes": 1 1 1 2 2 1 2 1 1 1 ...
$ bathtile      : Factor w/ 2 levels "yes","no": 1 1 1 1 1 1 1 1 1 1 ...
$ cheating      : Factor w/ 2 levels "yes","no": 1 1 1 1 1 1 1 1 1 1 ...
$ location      : Factor w/ 3 levels "good","normal",..: 1 1 1 2 1 2 2 2 2 2 ...
$ upkitchen     : Factor w/ 2 levels "no","yes": 1 1 1 1 2 1 1 1 1 1 ...
$ wwater        : Factor w/ 2 levels "yes","no": 1 1 1 1 1 1 1 1 1 1 ...
$ year_group    : Factor w/ 4 levels "1918-1959","1960-1969",..: 1 4 1 3 4 3 1 1 1 1 ...
$ district_group: Factor w/ 3 levels "Area1","Area2",..: 1 1 1 2 2 2 2 2 2 2 ...
```

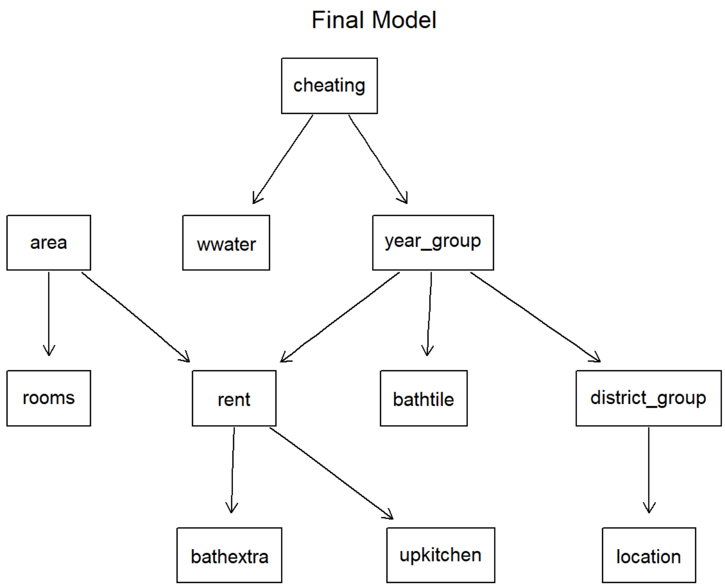
Model Learning

The detailed implementation for learning models from the data and selecting the most appropriate one can be found in the accompanying R code file. Additionally, understanding the modeling process requires background knowledge in topics related to Directed Graphical Models, such as learning methods, model learning algorithms, and more. A full discussion of these topics is beyond the scope of this report. Therefore, I will only present the final result. For a deeper understanding of Directed Graphical Models, one may refer to the book: **Bayesian Networks With Examples in R**.

After the model learning process and the selection of the most appropriate model, we arrive at the following result:

Model Directed Acylic Graph

As mentioned earlier, in a Directed Graphical Model, the joint probability distribution of all variables factorizes according to a directed acyclic graph (DAG). The Final Model has the following DAG structure:



The joint distribution is:

$$\boxed{\begin{array}{l} p(\text{cheating}).p(\text{wwater}|\text{cheating}).p(\text{year\_group}|\text{cheating}).p(\text{district\_group}|\text{year\_group}).p(\text{location}|\text{district\_group}) \\ \times\, p(\text{bathtile}).p(\textbf{rent}|\text{year\_group},\ \text{area}).p(\text{bathextra}|\text{rent}).p(\text{upkitchen}|\text{rent}).p(\text{rooms}|\text{rent}). \end{array}}$$

Model Parameters

The parameters are conditional probabilities. I will present only a few conditional probability tables for selected variables, rather than displaying all of them, in order to keep the report concise and readable. For example, conditional probability tables of `rooms` and `location` are:

```
> fit$rooms

Parameters of node rooms (multinomial distribution)

Conditional probability table:

area
rooms     [17,44]     (44,78]    (78,185]
1 0.736156352 0.025782689 0.001515152
2 0.263843648 0.565377532 0.030303030
3 0.000000000 0.390423573 0.507575758
4 0.000000000 0.018416206 0.368181818
5 0.000000000 0.000000000 0.071212121
6 0.000000000 0.000000000 0.021212121
> fit$location

Parameters of node location (multinomial distribution)

Conditional probability table:

district_group
location       Area1       Area2       Area3
good   0.514827018 0.224924012 0.165745856
normal 0.464579901 0.746200608 0.828729282
top    0.020593081 0.028875380 0.005524862
```

Model Applications

The `rent` is directly related to the property's area (`area`), year of construction (`year_group`), and the presence of high-quality bathroom and upscale kitchen equipment (`bathextra` and `upkitchen`). The remaining factors are indirectly related to the rental price through these direct factors. Given information about the direct factors, the indirect factors become conditionally independent of the rental price.

Use the Model

Once the model has been learned from the data, it is used for several purposes, including the following:

Analyzing Conditional Independencies

It can be simply understood that when variables X and Y are conditionally independent given Z, this means that once the information about Z is known, knowing X provides no additional information about Y, and vice versa.

Directed Graphical Models can help analyze such relationships. Once again, the precise theoretical details can be quite complex to present in full. Fortunately, the `R` package `bnlearn` provides a function `dsep()` that can be used to test the conditional independence between any three variables.

For example, once the `area` of a house is known, the number of rooms (`rooms`) and the `rent` become conditionally independent.

```
> dsep(fit,x="rooms",y="rent",z="area")
[1] TRUE
```

Approximating Conditional Probabilities

The `bnlearn` library also supports efficient approximation of conditional probabilities. As an example, I estimated the likelihood that a house located in `Area1` is equipped with High quality equipment in the bathroom:

$$\mathbb{P}(\text{bathextra} = \text{yes} \mid \text{district\_group} = \text{Area1})$$

```
> cpquery(fit,event = (bathextra=="yes"),evidence = (district_group == "Area1"),n=10^6)
[1] 0.09121026
> cpquery(fit,event = (bathextra=="yes"),evidence = (district_group == "Area1"),n=10^6)
[1] 0.09108432
```

Note that this probability approximation function is based on a randomized algorithm, so the result may vary with each run. The argument `n` controls the number of samples: the larger its value, the more accurate the result will be, but at the cost of increased computational time.

Querying and Predicting

Based on the above applications, I now perform queries to answer specific questions and make some predictions. The results will be presented here, while the corresponding code can be found in the accompanying R script.