Final Project

Group Name: TheVeryCoolDuo

Group Members: Michael Rabayda and Klaire Pham

Instructor: Professor Moataz Khalifa

Teaching Assistant: Devyani Mahajan

This project provides a critical data analysis of the AirBnB market in Berlin following the CRISP-DM process, a cross-industry process for data-mining. The report consists of six phases: business understanding, data understanding, data preparation, modeling, evaluation, and deployment. Notably, the process has been adjusted to attune to the grading rubrics for the final project of the course.

Introduction and Business Understanding

AirBnB is undeniably one of the most successful startups of the centuries – the business model has introduced a novel and convenient way to connect short-term visitors to landlords. In fact, AirBnB is so successful that even Berlin, known as a city with extremely tough laws governing vacation rentals, had to overturn their 2016 law banning the model. According to a 2020 analysis by AirDNA, Berlin's vacational rental market is dominated by AirBnB listings with 97% OTA market share. Interestingly, there is a strong trend in seasonality and different hotspots in terms of activity and pricing.

We decided to examine the trends further. Looking at the dataset containing over 22,000 rental listings in Berlin as of November 2018, right after the ban lift, we decided that it would be interesting to look into the patterns determining the daily price of each listing, be it location, amenities, property type, etc. We are interested in how those variables interact with price firstly, and with each other secondary.

This data analysis is supposed to help travelers and potential AirBnB hosts in Berlin in many ways. As a potential visitor, we want to think of our trips in two ways: cost optimized and experience optimized. We would attempt to find out how the listing's factors play into its expenditure and rating scores. As a prospective host, we want to think about how the property can be upgraded for better pricing. Hence, the questions we aim to answer are:

1. What are the most high-rated room type? Most highly regarded neighborhood?
2. Does popularity always equate high ratings, thus better experience?
   (Note: In other words, knowing that number of reviews per month indicate the listing's popularity (reviews are mandatory to AirBnB users), what is the correlation between number of reviews per month and rating scores.)
3. How do variables interact with price? Which neighborhood group or room type makes price the highest?

Hypothetically, our answers to these questions are:

1. Out of the three room types, private room, entire home/ apartment, and shared room, we think that private room would be rated the highest, following by entire home, and finally shared room. These types of room offer descending level of privacy and ascending probability of sharing amenities, which is prone to create discomfort, resulting in lower rating scores.
2. High rating scores have a positive correlation with popularity because the higher the ratings of a place, the more likely people that aim for experience-optimized trip would try it out.
3. Among the neighborhood groups, the ones in the center are more expensive than the one in the outskirts. In terms of room type, entire home/ apartment would be the most expensive, following by private room and shared room.

Throughout the span of this report, we will try to confirm the accuracy of these hypotheses.

Data Understanding

The data is retrieved from Kaggle, a public site which distributes practice datasets at no cost to data science enthusiasts. It offers a rich dataset of listings, calendar entries, and reviews. The data is relatively clean: most important variables do not have issues of missing values. The numerical variables are largely skewed for understandable reasons, which then were adjusted with data cleaning steps we would be listing out below.

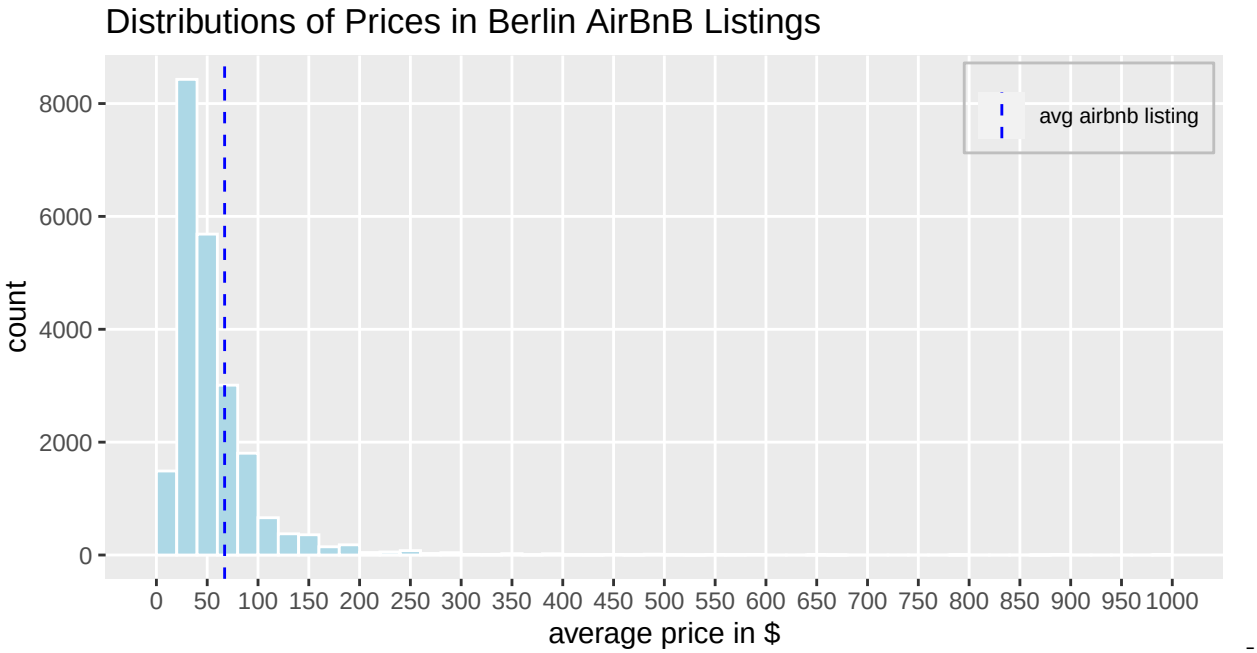
The most important variable, price, is displayed with variety in both daily, weekly, monthly, and service fee. They were not initially numerical, but were easily converted so. Overall, the dataset was useful for our research questions.

1. Data exploration

Some insights from data exploration are:
- In Berlin, there are 22,552 listings as of the time of retrieval
- Nearly 99.5% of the listings are priced under 350 per day
- About 98.4% of the listings with non-NA review scores receive a score of over 70 out of 100. Lower review scores are extremely rare.

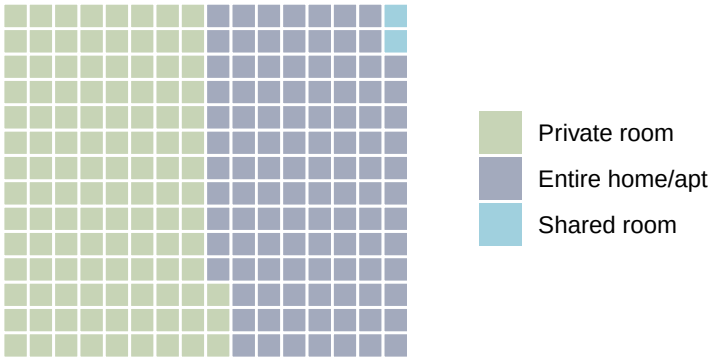
In order to confirm the rationale to data cleaning, we then plot some distribution graphs.

How are prices distributed for AirBnB listings in Berlin?



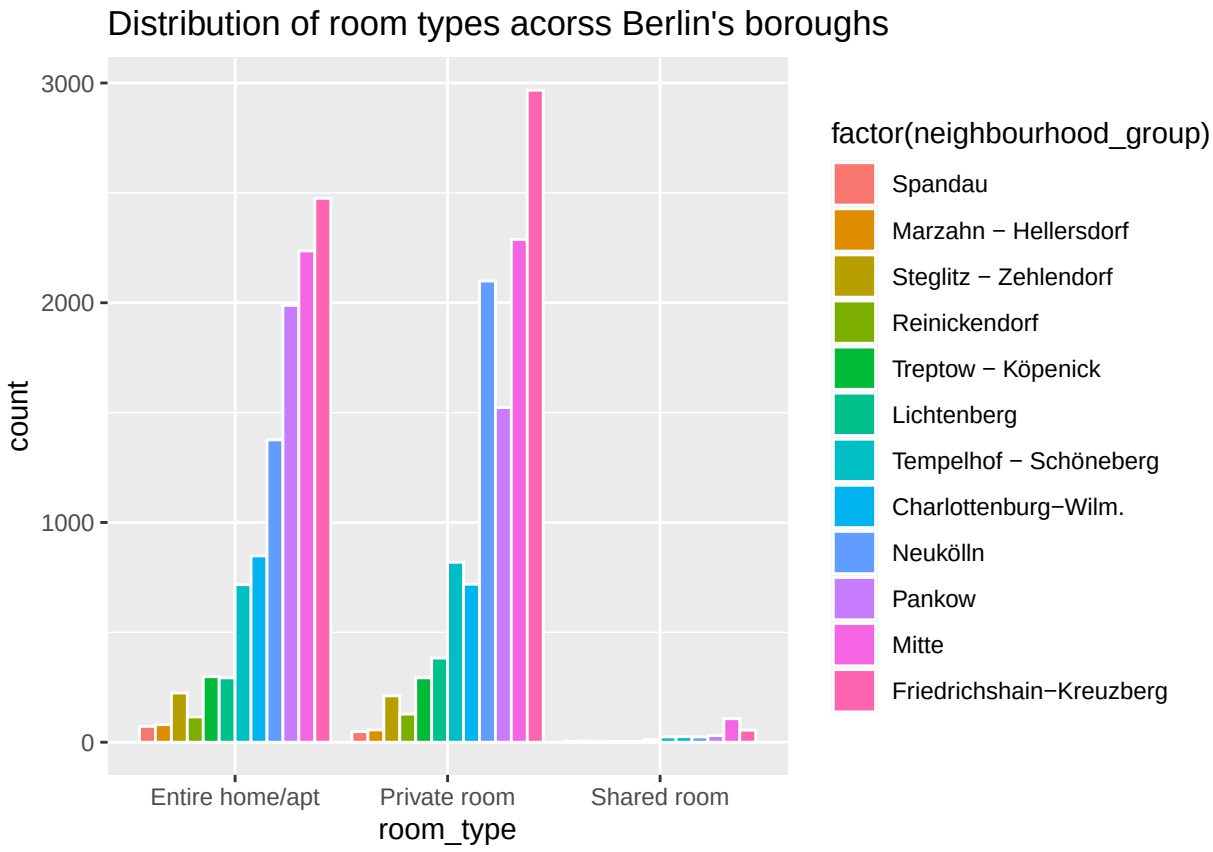
Most listings in Berlin are priced $25-75 per day
- As expected, our distribution is largely left-skewed due to big outliers caused by luxurious accommodation

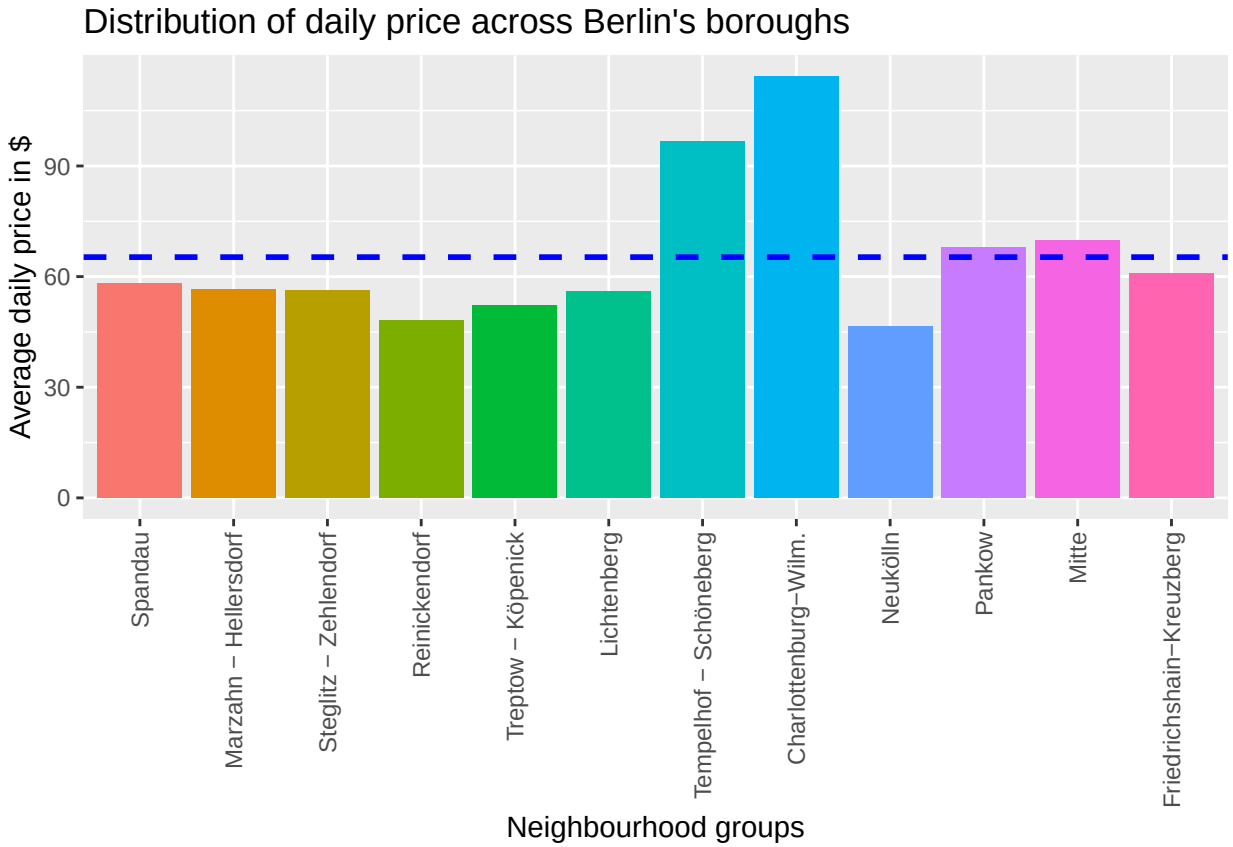
What is the distribution for room types in Berlin AirBnB listings?



- Private room and entire home have a relatively equal share of the market. They altogether make up more than 95% of the listings offered in Berlin.
  - Showing through the abundant offer of private rooms and apartments, it can be inferred that privacy seems to be highly regarded by customers. Interestingly, this is in contrary to the company's initial idea in that the customers share airbeds with locals.

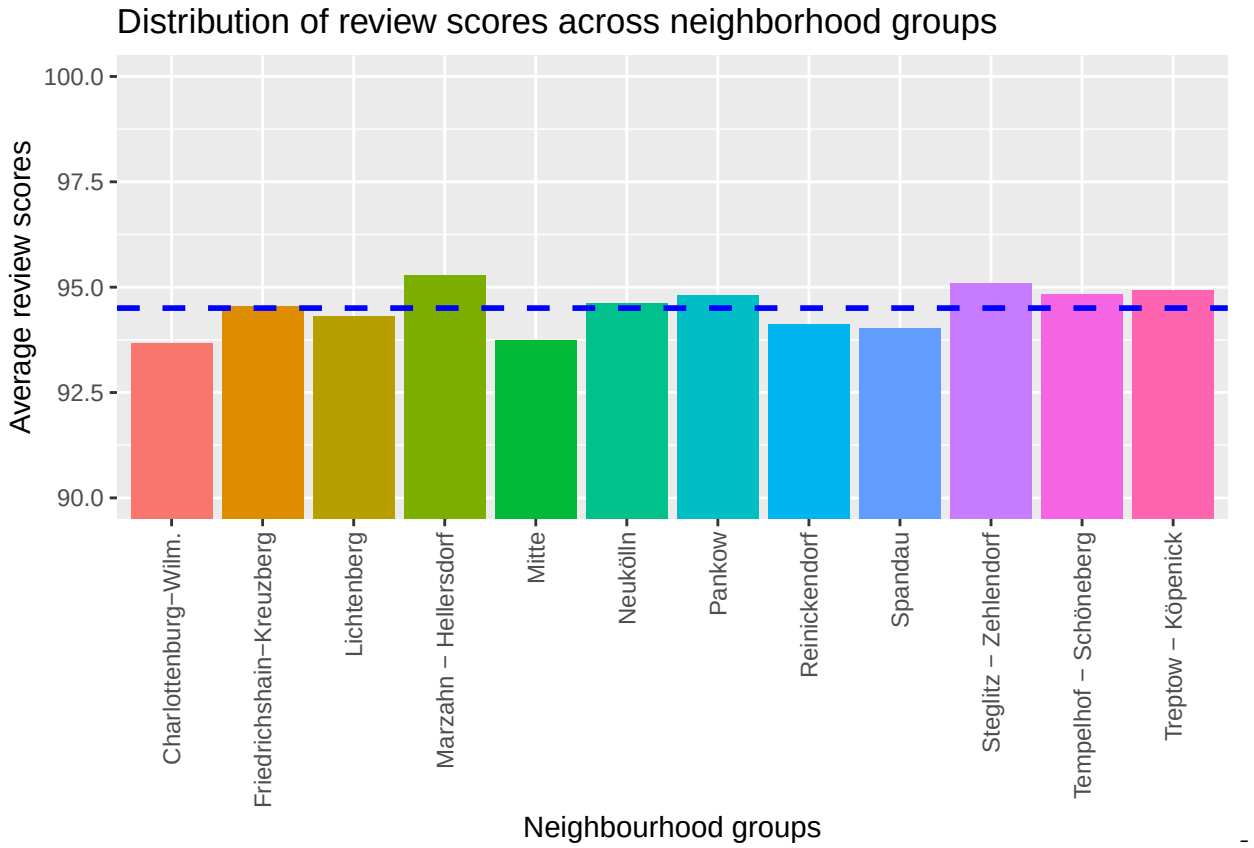
How are listings expenditure distributed across the city and around different room types?





- Although the system to measure distance to Berlin's center is not an official one, it is undeniable that boroughs nearer to the center have higher counts of listings. According to PlanetWare, the four boroughs with the highest number of listings, Mitte, Pankow, Friedrichshain-Kreuzberg, and Neukölln all are cultural hubs packed with famous sights and nightclubs.
- Charlottenburg-Wilm. and Tempelhof - Schöneberg are the two boroughs with the highest average price per day. These districts are both described as traditional and historical rather than cosmopolitan and young – they have many beautiful parks and emphasize the quality of living, which can explains the high prices in comparison with the remaining boroughs.
- The two most affordable districts are Reinickendorf and Neukölln, sharing no particular factors. Reinickendorf is sparsely populated while Neukölln is known as an emerging location for students and creative professionals.
- Pankow and Mitte are close to the average daily price of all AirBnB's in the city.

How are review scores distributed across the city?



The rating scores are relatively high – no boroughs receive average rating scores under 94 out of 100.
- Marzahn - Hellersdorf and Steglitz - Zehlendorf receives the highest average rating scores, yet they also offer extremely smaller number of listings. It might be due to the small number of listings there that the scores remain high. Similarly, districts that have high number of listings, such as Mitte and Charlottenburg-Wilm., receive lower ratings, though this relationship is rather unclear.

From the information produced in the three exploratory graphs above, we think that there is a correlation between the distance to the centers of the city to the price and number of listings (which can also indicate popularity). Interaction between neighborhood groups and price shows strong correlation as well. However, it is not the case between neighborhood and ratings where average values are similar rather than showing a detectable relationship.

2. Data quality

From the distribution graphs above, we realize the skewedness of the two variables price and review scores. Due to luxurious listings, *price* has outliers while *review_scores_rating* are largely over 70 as a result of the competitive market of AirBnB in Berlin packed with high quality listings.

Data Preparation

The data cleaning process tackle these following issues:
1) In *listings_summary*, one of our two major datasets, *price* is not a numerical variable but a character one instead, which makes plotting difficult. Thus, we removed the $ symbols and convert all price-related variables numerical.
2) For AirBnBs' price, most of the observations (over 22,000 out of 22,552) is priced under 350 euros per night. Thus, it is needed to separate the data frame into usable observations and the outliers. for both *listings* and *listings_summary*, our two main datasets, we filter out listings that are priced over $350.
3) Reviews score are mostly over 70 (over 17,000 of them). There are only a few dozens observations of

reviews_rating_scores under 70. We filter out listings with *reviews_rating_scores* under 70 and do the same thing with listings that have *number_of_reviews* over 300 (same rationale).

These cleaning steps tailors the result of our data analysis to normally priced listings with reasonable number of reviews and review scores, which would serve better in future models. While the models produced in this analysis might not apply to luxurious listings or ground-breaking places that excel in popularity or dip in quality, we believe that they fit the search for cost-optimized and experience-optimized AirBnB's that our research questions put forth.

We conclude our data cleaning process with two datasets derived from *listings* and *listings_summary*, which are respectively called *smallset* and *largeset*. We also create a training set and a test set with the ratio 70:30 to test the models. They are called *train* and *test* throughout the code that might appear in certain parts of this report.

Modeling

The modeling section is divided into two large parts:
1. Answering a prospective traveler's questions: Finding best choices for cost-optimized and experience-optimized travelers through the examination of the relationships between daily price, review scores, number of reviews per month, and neighborhood groups of the listing (Model A and B). 2. Answering a prospective host's questions: Finding the best number of guests included and the room type of a listing to secure a high price through the examination of the relationships between those variables (Model C).

1. Linear regression models of review scores, price, and neighborhood groups (A)

Independent model:

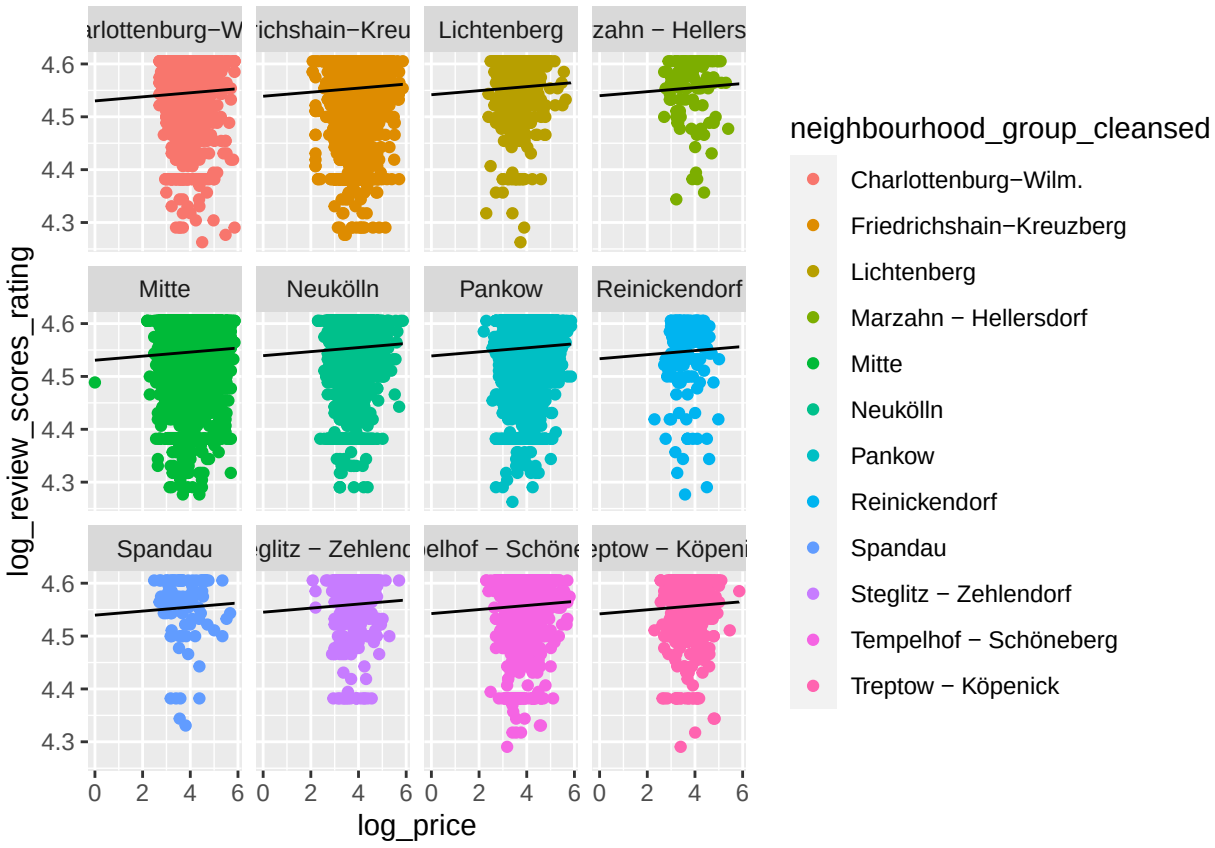
```
##
## ================================================================================
##                                                    Dependent variable:
##                                                -----------------------------
##                                                  log_review_scores_rating
## --------------------------------------------------------------------------------
## log_number_of_reviews                                -0.003*** (0.0003)
## neighbourhood_group_cleansedFriedrichshain-Kreuzberg  0.008*** (0.002)
## neighbourhood_group_cleansedLichtenberg               0.010** (0.003)
## neighbourhood_group_cleansedMarzahn - Hellersdorf      0.010 (0.006)
## neighbourhood_group_cleansedMitte                      0.001 (0.002)
## neighbourhood_group_cleansedNeukölln                  0.007** (0.002)
## neighbourhood_group_cleansedPankow                    0.009*** (0.002)
## neighbourhood_group_cleansedReinickendorf              0.003 (0.005)
## neighbourhood_group_cleansedSpandau                    0.008 (0.007)
## neighbourhood_group_cleansedSteglitz - Zehlendorf     0.014*** (0.004)
## neighbourhood_group_cleansedTempelhof - Schöneberg    0.012*** (0.003)
## neighbourhood_group_cleansedTreptow - Köpenick        0.010** (0.003)
## Constant                                             4.553*** (0.002)
## --------------------------------------------------------------------------------
## Observations                                              17,762
## R2                                                         0.009
## ================================================================================
## Note:                                          *p<0.05; **p<0.01; ***p<0.001
```

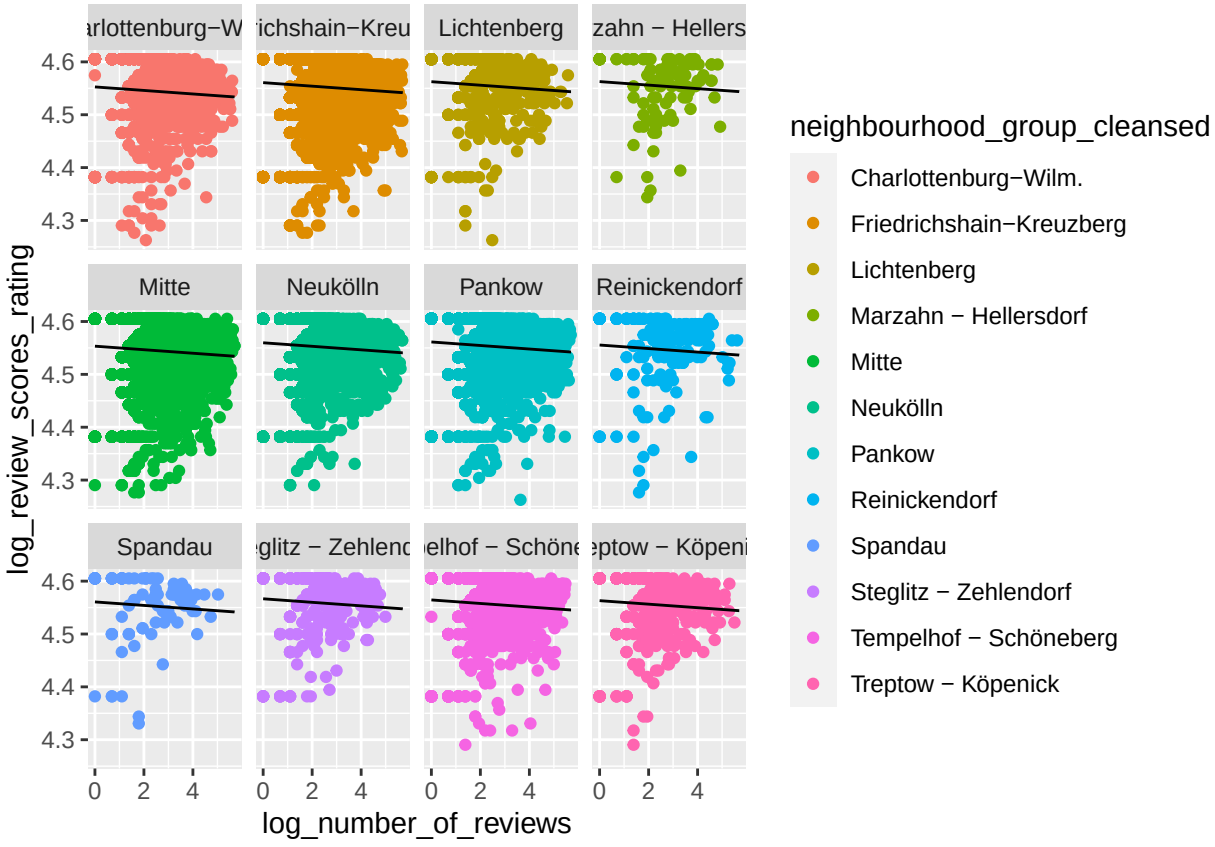
Interacting model:

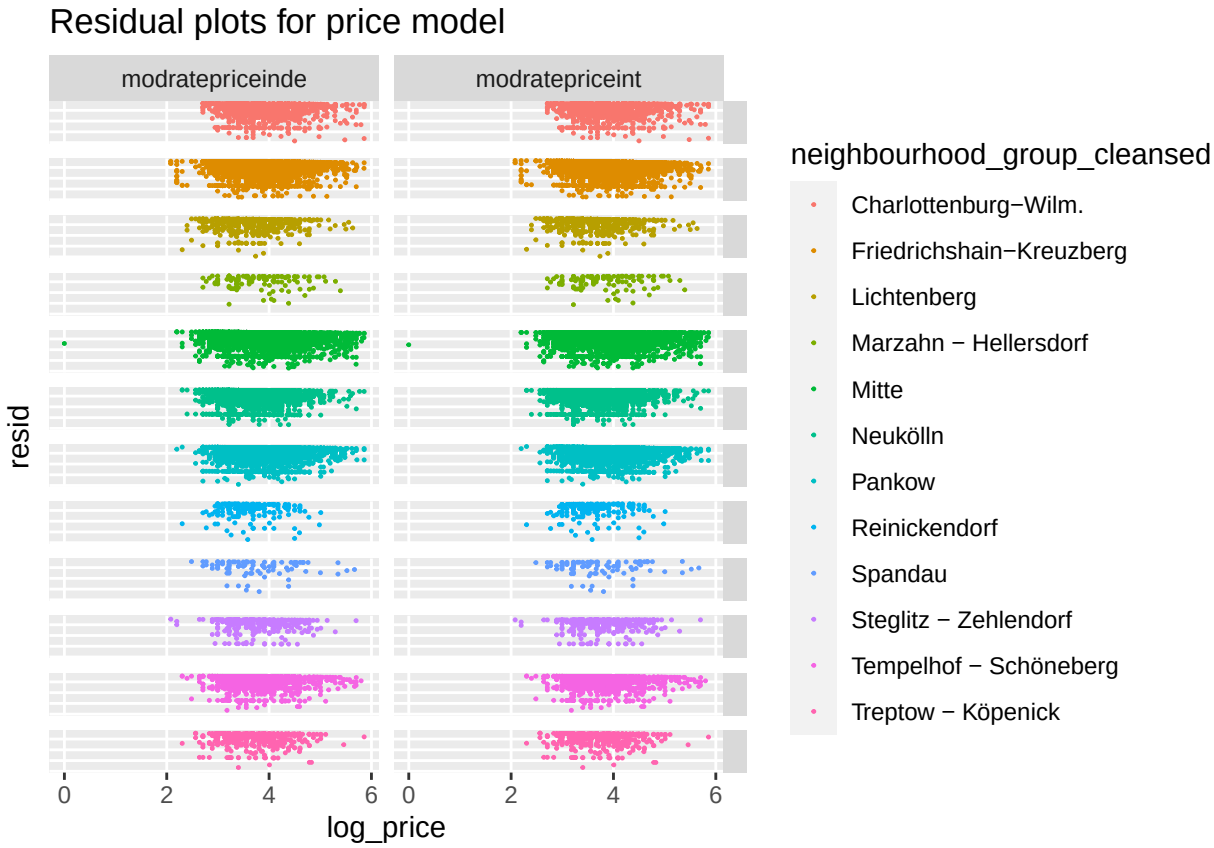
```
##
## ================================================================================================
##                                                                          Dependent variable:
```

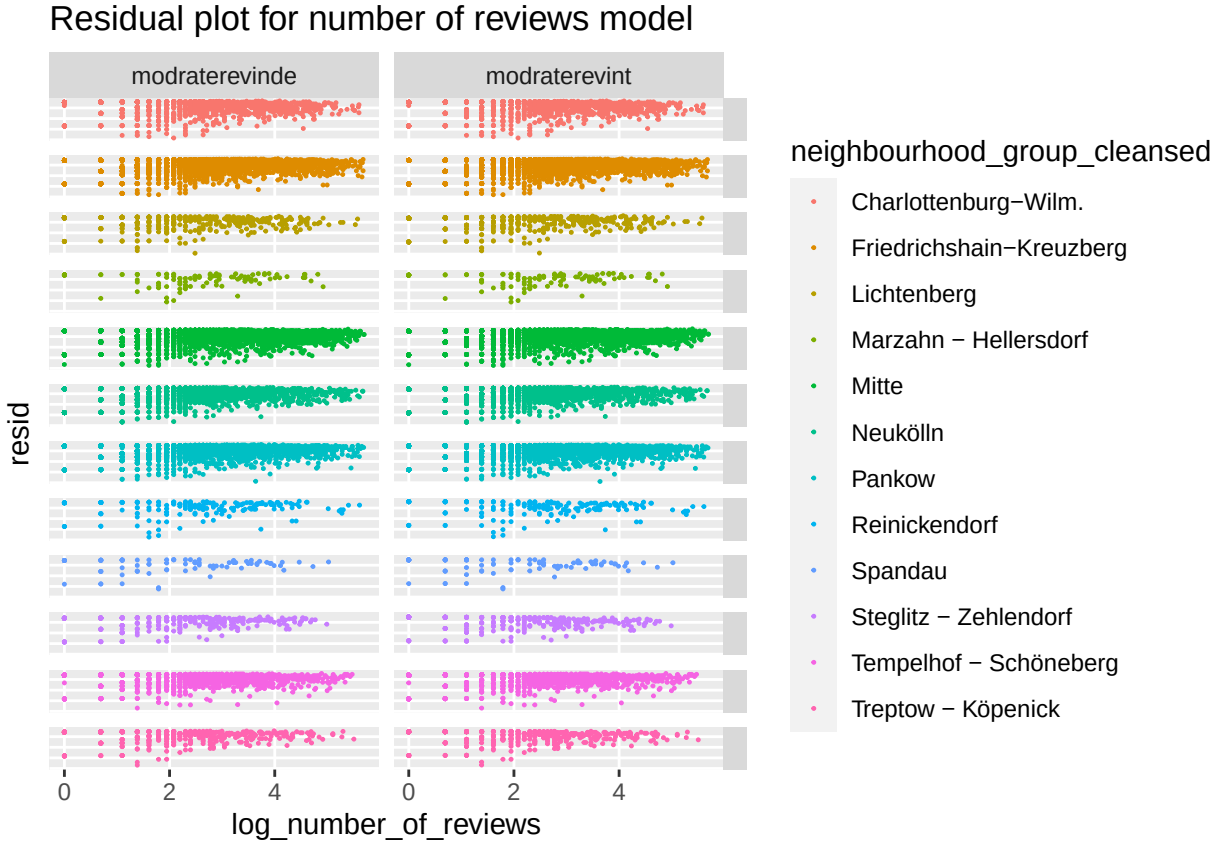
```
##                                                                                 ------------------------
##                                                                                   log_review_scores_ratin
## ---------------------------------------------------------------------------------------------------------
## log_number_of_reviews                                                                -0.003* (0.001)
## neighbourhood_group_cleansedFriedrichshain-Kreuzberg                                  0.008* (0.004)
## neighbourhood_group_cleansedLichtenberg                                               0.016* (0.006)
## neighbourhood_group_cleansedMarzahn - Hellersdorf                                      0.013 (0.013)
## neighbourhood_group_cleansedMitte                                                      0.003 (0.004)
## neighbourhood_group_cleansedNeukölln                                                  0.010* (0.004)
## neighbourhood_group_cleansedPankow                                                     0.007 (0.004)
## neighbourhood_group_cleansedReinickendorf                                             -0.007 (0.010)
## neighbourhood_group_cleansedSpandau                                                    0.004 (0.012)
## neighbourhood_group_cleansedSteglitz - Zehlendorf                                      0.004 (0.007)
## neighbourhood_group_cleansedTempelhof - Schöneberg                                    0.012* (0.005)
## neighbourhood_group_cleansedTreptow - Köpenick                                         0.007 (0.006)
## log_number_of_reviews:neighbourhood_group_cleansedFriedrichshain-Kreuzberg          0.00004 (0.001)
## log_number_of_reviews:neighbourhood_group_cleansedLichtenberg                         -0.003 (0.002)
## log_number_of_reviews:neighbourhood_group_cleansedMarzahn - Hellersdorf               -0.001 (0.005)
## log_number_of_reviews:neighbourhood_group_cleansedMitte                               -0.001 (0.001)
## log_number_of_reviews:neighbourhood_group_cleansedNeukölln                            -0.002 (0.002)
## log_number_of_reviews:neighbourhood_group_cleansedPankow                               0.001 (0.002)
## log_number_of_reviews:neighbourhood_group_cleansedReinickendorf                        0.004 (0.004)
## log_number_of_reviews:neighbourhood_group_cleansedSpandau                              0.002 (0.005)
## log_number_of_reviews:neighbourhood_group_cleansedSteglitz - Zehlendorf                0.005 (0.003)
## log_number_of_reviews:neighbourhood_group_cleansedTempelhof - Schöneberg             -0.0003 (0.002)
## log_number_of_reviews:neighbourhood_group_cleansedTreptow - Köpenick                   0.002 (0.003)
## Constant                                                                            4.552*** (0.003)
## ---------------------------------------------------------------------------------------------------------
## Observations                                                                             17,762
## =========================================================================================================
## Note:                                                                  *p<0.05; **p<0.01; ***p<0
```

Since the p-values are lower for both independent models we'll use them and check with our residuals. We then plot logged variables with regression line and the residual plots:


Lichtenberg
Mitte
Neukölln
Pankow
Reinickendorf
Spandau
log_review_scores_rating
log_price
neighbourhood_group_cleansed
Charlottenburg−Wilm.
Friedrichshain−Kreuzberg
Lichtenberg
Marzahn − Hellersdorf
Mitte
Neukölln
Pankow
Reinickendorf
Spandau
Steglitz − Zehlendorf
Tempelhof − Schöneberg
Treptow − Köpenick

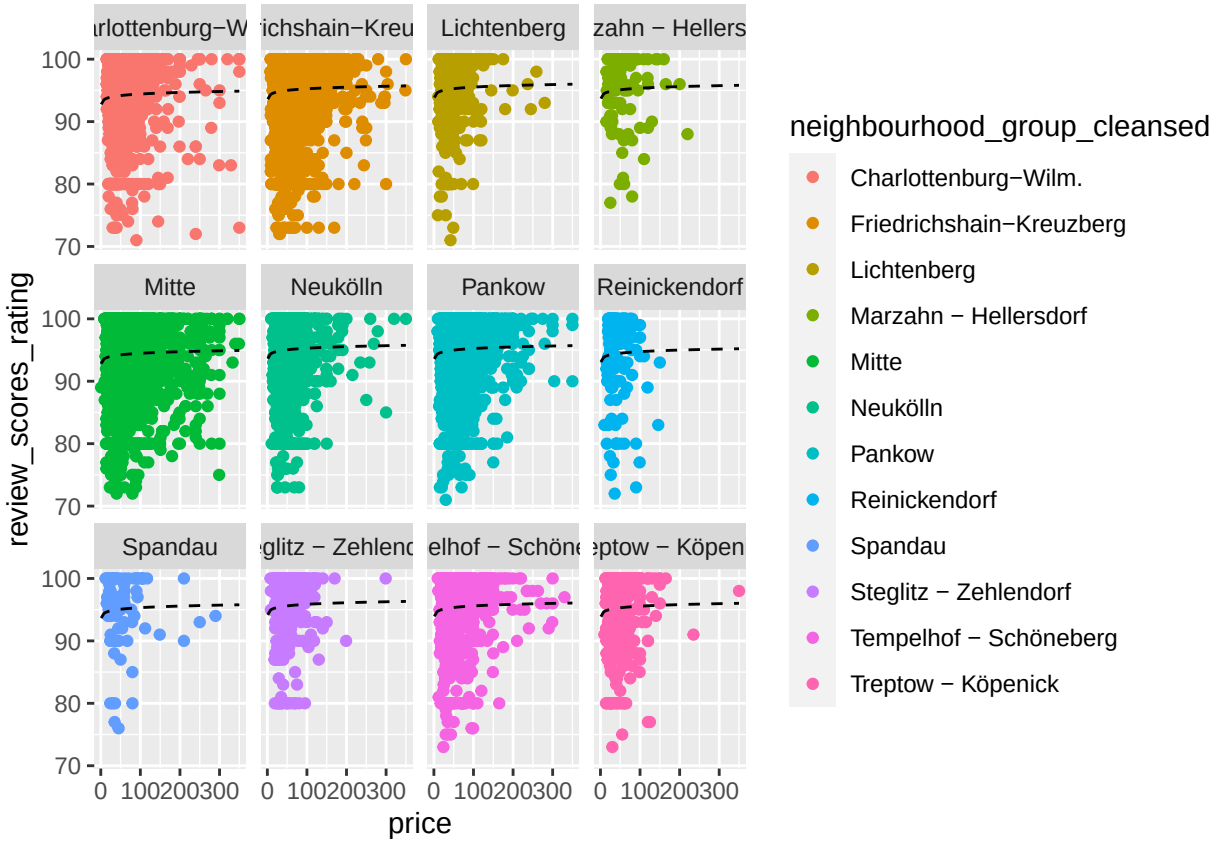

Lichtenberg
Mitte
Neukölln
Pankow
Reinickendorf
Spandau
log_review_scores_rating
log_number_of_reviews
neighbourhood_group_cleansed
Charlottenburg−Wilm.
Friedrichshain−Kreuzberg
Lichtenberg
Marzahn − Hellersdorf
Mitte
Neukölln
Pankow
Reinickendorf
Spandau
Steglitz − Zehlendorf
Tempelhof − Schöneberg
Treptow − Köpenick

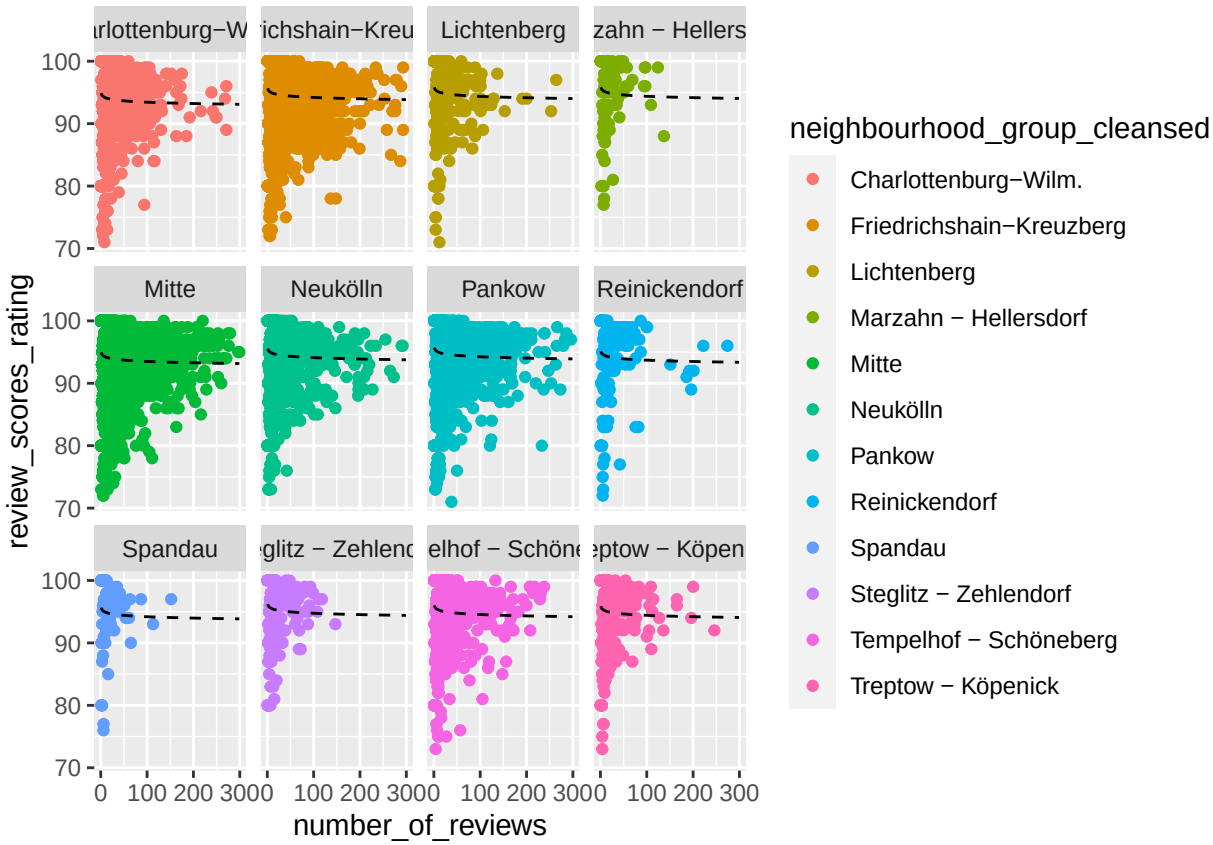




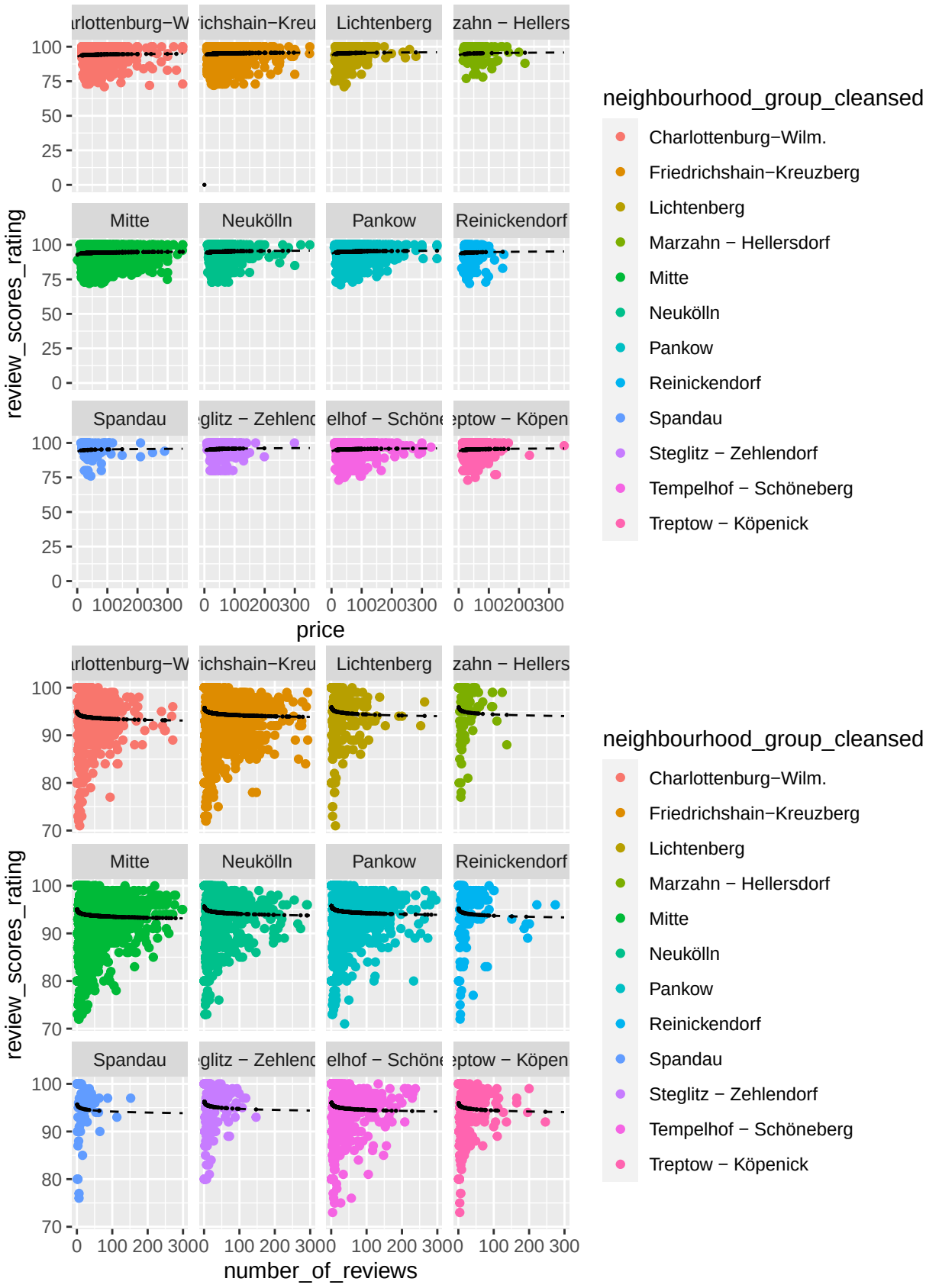
In both relationships, the independent models produce a residual plot that appears more random than the

plots the interacting models produced - this further confirms our choice to use the independent plots. We then plot original variables without log transformation with the transformed regression line using the grids created. In other words, we plot the price then number of reviews against rating scores with black dashed lines representing rating prediction using the independent models.





We then add the test set onto the plots.



Using a linear regression model of an AirBnB's *review_scores_rating* as a function of the price and the

neighbourhood group, and separately as a function of the number of reviews and the neighbourhood group. We first had to log both *review_scores_rating* and both *price* and *number_of_reviews* separately due to the skewed nature of their distributions. Through model analysis, we were able to identify the lack of interaction between the two explanatory parameters, which makes the abstract equation similar to this:

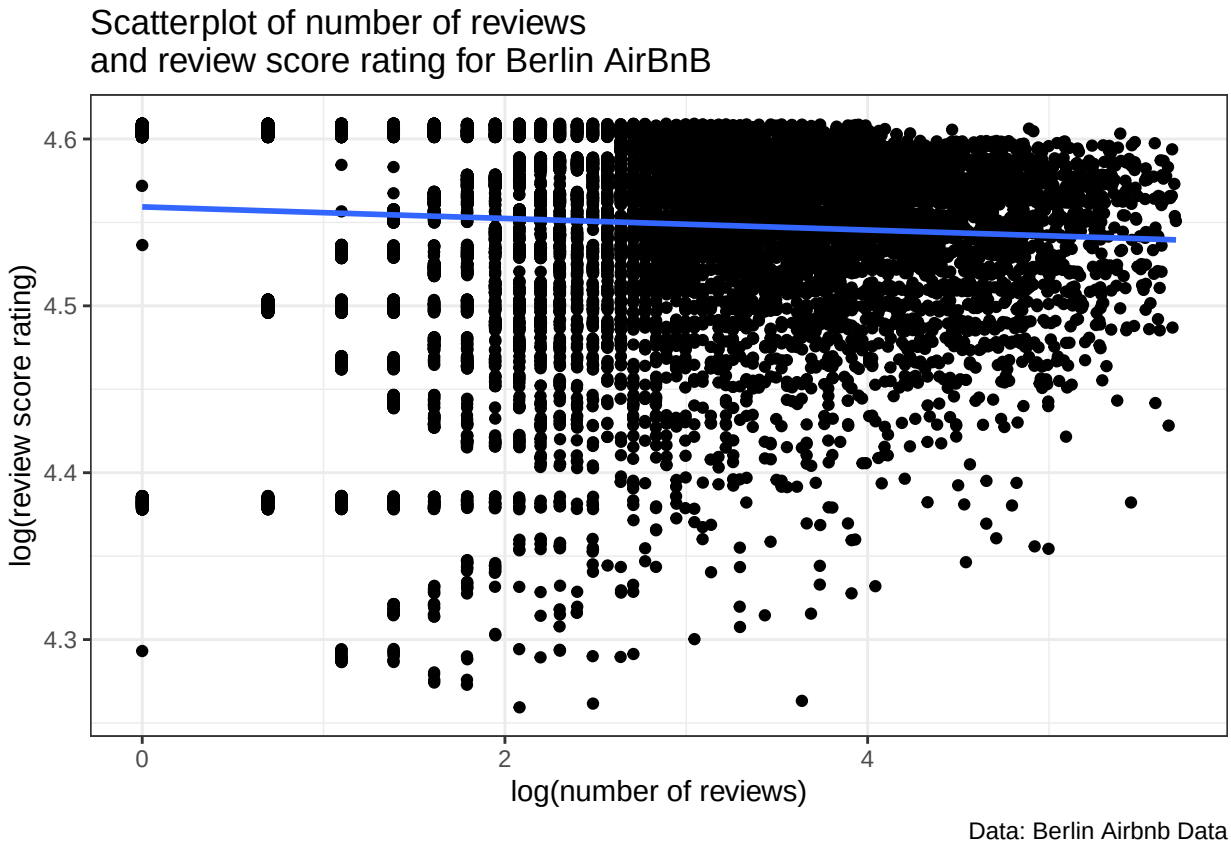
$$log(review\_scores\_rating) = Intercept + A * log(price) + B * neighbourhood\_group$$

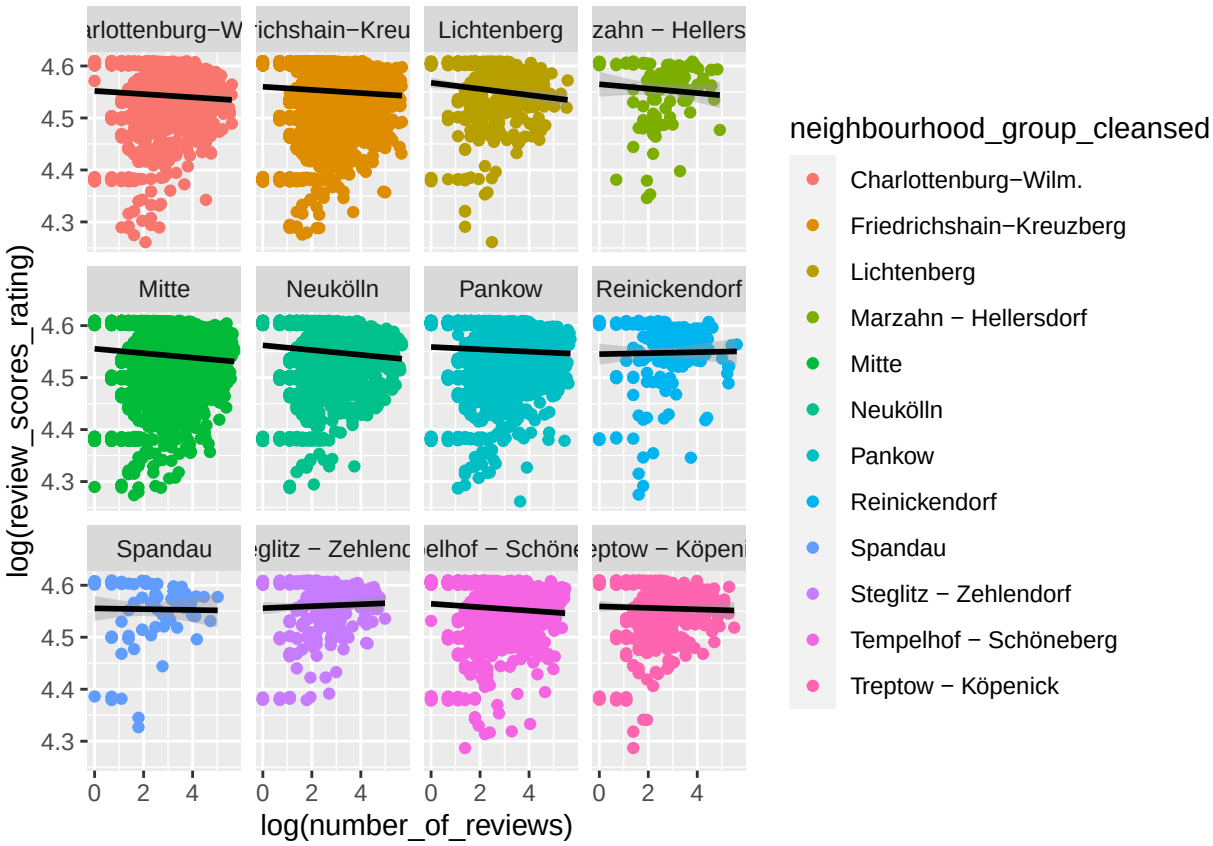
$$log(review\_scores\_rating) = Intercept + A * log(number\_of\_reviews) + B * neighbourhood\_group$$

Choosing to use a linear model was a definite challenge due to the skewed distributions of all the variables we worked with. We found that by plotting transformed histograms we were able to see which transformation to use and it ended up being a log transformation for all variables we were working with besides *neighbourhood_group*.

2. Linear regression models of review scores, number of reviews, and neighborhood groups (B)

Next, we plot additional predictor to the response variable, in this case *number_of_reviews* to *review_scores_rating*:





Repeating the steps, we create independent and interacting models, then using test set to check the validity of the models.

Independent model:

```
stargazer(modratepricerevinde,
          star.cutoffs = c(0.05, 0.01, 0.001),
          keep.stat = c("n","rsq"),
          type = 'text', single.row=TRUE)
```

```
##
## ================================================================================
##                                                        Dependent variable:
##                                                    ----------------------------
##                                                      log_review_scores_rating
## --------------------------------------------------------------------------------
## log_price                                                 0.005*** (0.001)
## log_number_of_reviews                                    -0.004*** (0.0003)
## neighbourhood_group_cleansedFriedrichshain-Kreuzberg      0.008*** (0.002)
## neighbourhood_group_cleansedLichtenberg                    0.011** (0.003)
## neighbourhood_group_cleansedMarzahn - Hellersdorf          0.010 (0.006)
## neighbourhood_group_cleansedMitte                          0.001 (0.002)
## neighbourhood_group_cleansedNeukölln                      0.008*** (0.002)
## neighbourhood_group_cleansedPankow                        0.008*** (0.002)
## neighbourhood_group_cleansedReinickendorf                  0.004 (0.005)
## neighbourhood_group_cleansedSpandau                        0.009 (0.007)
## neighbourhood_group_cleansedSteglitz - Zehlendorf         0.015*** (0.004)
## neighbourhood_group_cleansedTempelhof - Schöneberg        0.012*** (0.003)
## neighbourhood_group_cleansedTreptow - Köpenick             0.011** (0.003)
```

```
## Constant                                                        4.534*** (0.004)
## ------------------------------------------------------------------------------------
## Observations                                                        17,762
## R2                                                                   0.011
## ====================================================================================
## Note:                                                  *p<0.05; **p<0.01; ***p<0.001
```

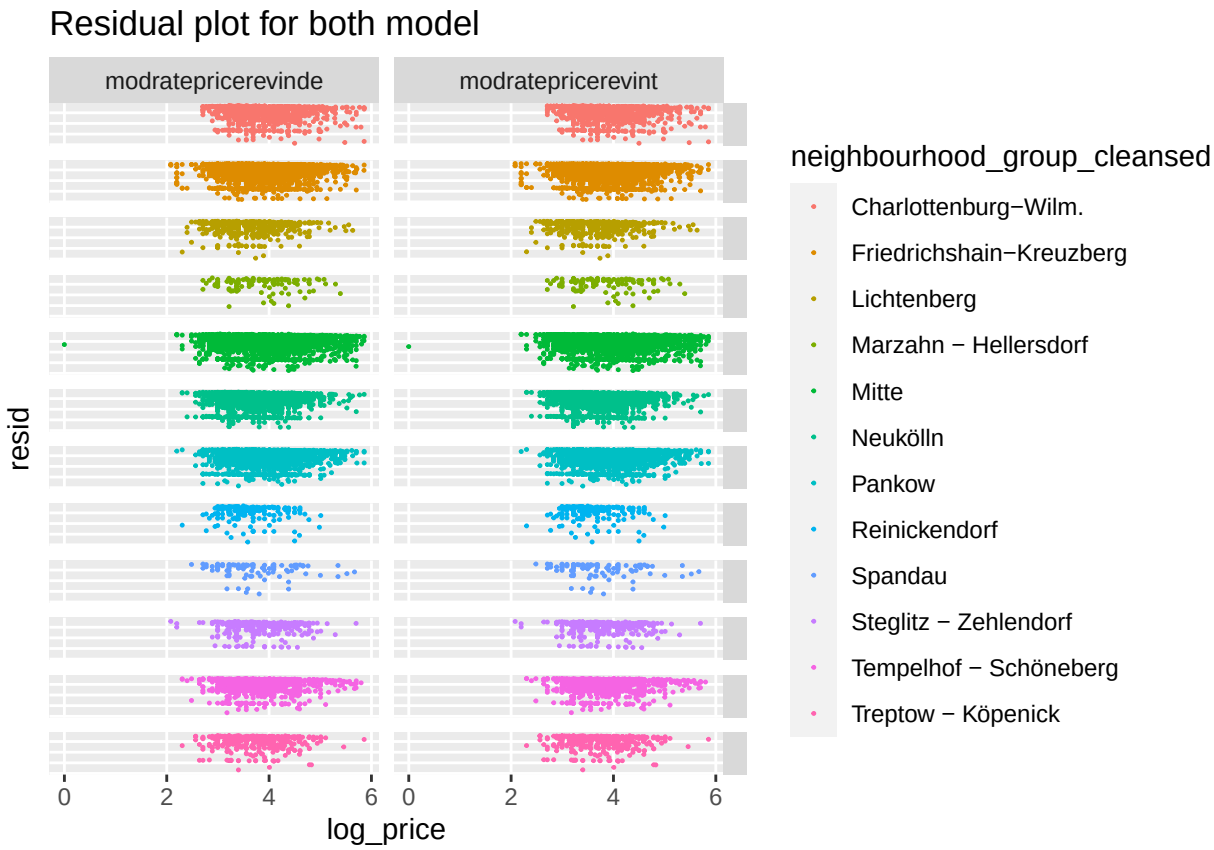
Interacting model:

```
stargazer(modratepricerevint,
          star.cutoffs = c(0.05, 0.01, 0.001),
          keep.stat = c("n","rsq"),
          type = 'text', single.row=TRUE)
```

```
##
## ====================================================================================
##                                                        Dependent variable:
##                                                    ------------------------------
##                                                      log_review_scores_rating
## ------------------------------------------------------------------------------------
## log_price                                                  0.009*** (0.002)
## log_number_of_reviews                                        0.004 (0.002)
## neighbourhood_group_cleansedFriedrichshain-Kreuzberg       0.008*** (0.002)
## neighbourhood_group_cleansedLichtenberg                    0.011*** (0.003)
## neighbourhood_group_cleansedMarzahn - Hellersdorf            0.010 (0.006)
## neighbourhood_group_cleansedMitte                            0.001 (0.002)
## neighbourhood_group_cleansedNeukölln                       0.008*** (0.002)
## neighbourhood_group_cleansedPankow                         0.008*** (0.002)
## neighbourhood_group_cleansedReinickendorf                    0.004 (0.005)
## neighbourhood_group_cleansedSpandau                          0.009 (0.007)
## neighbourhood_group_cleansedSteglitz - Zehlendorf          0.015*** (0.004)
## neighbourhood_group_cleansedTempelhof - Schöneberg         0.012*** (0.003)
## neighbourhood_group_cleansedTreptow - Köpenick              0.011** (0.003)
## log_price:log_number_of_reviews                           -0.002*** (0.001)
## Constant                                                   4.517*** (0.006)
## ------------------------------------------------------------------------------------
## Observations                                                        17,762
## R2                                                                   0.012
## ====================================================================================
## Note:                                                  *p<0.05; **p<0.01; ***p<0.001
```

Yet for these models, we were only able to predict for test sets using the model, not creating any grid making or plotting due to the size of the data.

Checking the residual plots:



3. Evaluation of models (A) and (B)

We can see two separate models:
modratepriceinde:log_review_scores_rating ~ log_price + neighbourhood_group_cleansed
modratepriceint:log_review_scores_rating ~ log_price * neighbourhood_group_cleansed

After testing to see which model fit the data better, we concluded that since the independent model had much lower p-values than interactive model (whose p-value peaked at 0.9029496!) that the independent model was the better fit due to its greater statistical significance. While the residual plot of both models appeared very similar, they also both appeared random which indicates our model was accurate in capturing the relationship between dependent and explanatory variables.

modraterevinde:log_review_scores_rating ~ log_number_of_reviews + neighbourhood_group_cleansed
modraterevint:log_review_scores_rating ~ log_number_of_reviews * neighbourhood_group_cleansed

We found the same to be true about the model substituting *number_of_reviews* for *price*. The independent model for this data again showed lower p-values in comparison to the interactive model, which is why we chose to use the independent model. The residual plot of both models appeared very similar, however they also both appeared to have a slight positive trend which indicates our model could have been more accurate in capturing the relationship between dependent and explanatory variables - this led us to believe that using *number_of_reviews* as an additional predictor to the price model might capture the relationship better than just using *number_of_reviews*.

Additional Predictor Models:
modratepricereveinde:log_review_scores_rating ~ log_price + log_number_of_reviews + neighbourhood_group_cleansed
modratepricerevint:log_review_scores_rating ~ log_price * log_number_of_reviews + neighbourhood_group_cleansed

We found the same to be true about the model using *log_number_of_reviews* as the additional predictor. The independent model for this data again showed lower p-values in comparison to the interactive model, which is why we chose to use the independent model. While the residual plot of both models appeared very similar, they also both appeared very random which indicates our model was accurate in capturing the relationship between dependent and explanatory variables - this residual plot looked the most random of all the plots we had seen while fitting models with these variables. This indicated that this model probably captured the linear relationship most accurately out of all the models.

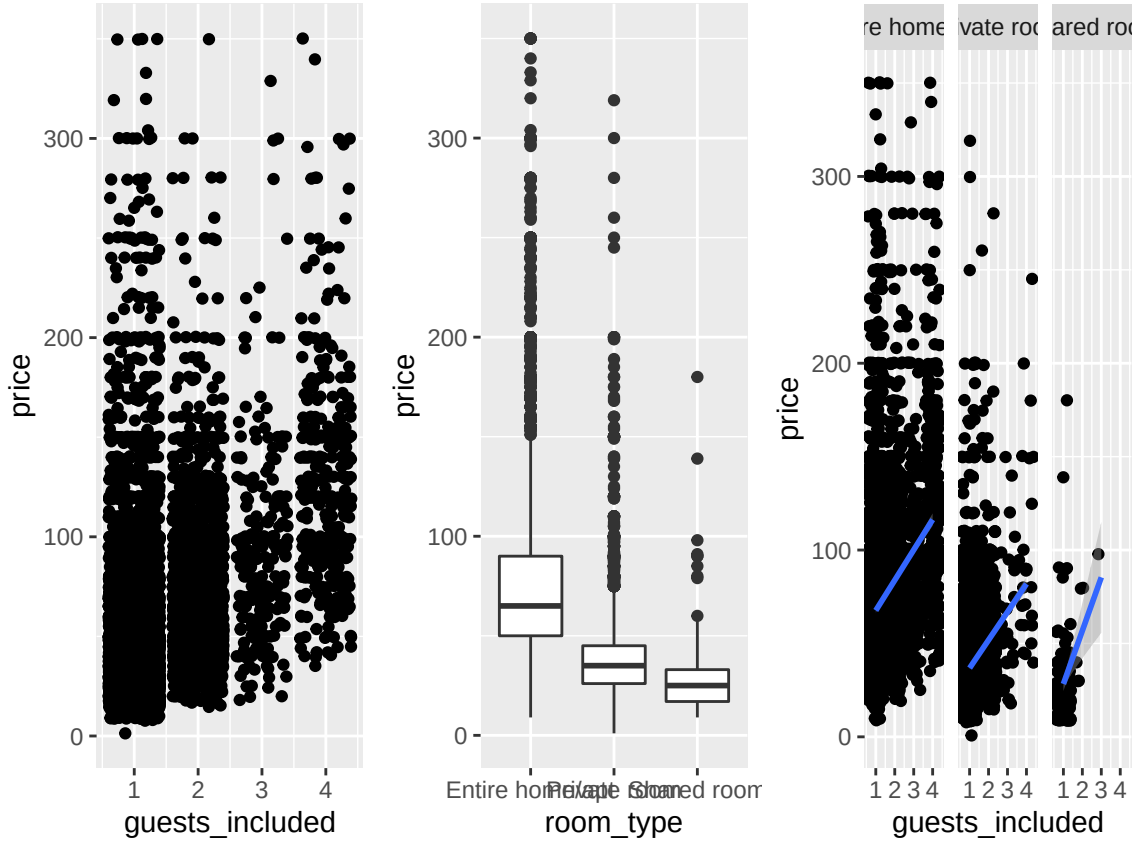
4. Linear regression models of number of guests included, room types, and price (C)

Hypothesis: The number of guests included and the room type has a positive correlation with the price. Specifically, the more guests included, the higher the price would be. Regarding room type, entire home/apt would cost more than private room, which costs more than shared room. For the number of guests included is also skewed, an additional step of filtering is done after viewing the distribution of the variables.

```
# Distribution of the relationship between various variables: guests_included, price
quantile(largeset$price)
quantile(largeset$guests_included)

# Creating test set and training set
working_set <- largeset %>%
  filter(guests_included < 5)
dt = sort(sample(nrow(working_set), nrow(working_set)*.7))
train <- working_set[dt,]
test <- working_set[-dt,]
```

Afterwards, we plot the explanatory variables against the response variable for detection of any correlation at all.



As expected, we found that price for entire home is higher than price for a single room, following by price for a shared room. The number of guests included also has a positive impact on the price. We then continue to create two model, one independent and one interacting, in which we foudn out that the independent model is much more reliable as it has lower p-values.

```
# Models
mod1 <- lm(price ~ guests_included + room_type, train)
mod2 <- lm(price ~ guests_included * room_type, train)
```

Independent model:

```
stargazer(mod1,
          star.cutoffs = c(0.05, 0.01, 0.001),
          keep.stat = c("n","rsq"),
          type = 'text', single.row=TRUE)
```

```
##
## ================================================
##                          Dependent variable:
##                      ---------------------------
##                                 price
## ------------------------------------------------
## guests_included           16.007*** (0.460)
## room_typePrivate room     -31.066*** (0.594)
## room_typeShared room      -38.828*** (2.645)
## Constant                  51.749*** (0.814)
## ------------------------------------------------
## Observations                   12,317
## R2                              0.320
## ================================================
## Note:                *p<0.05; **p<0.01; ***p<0.001
```

Interacting model:

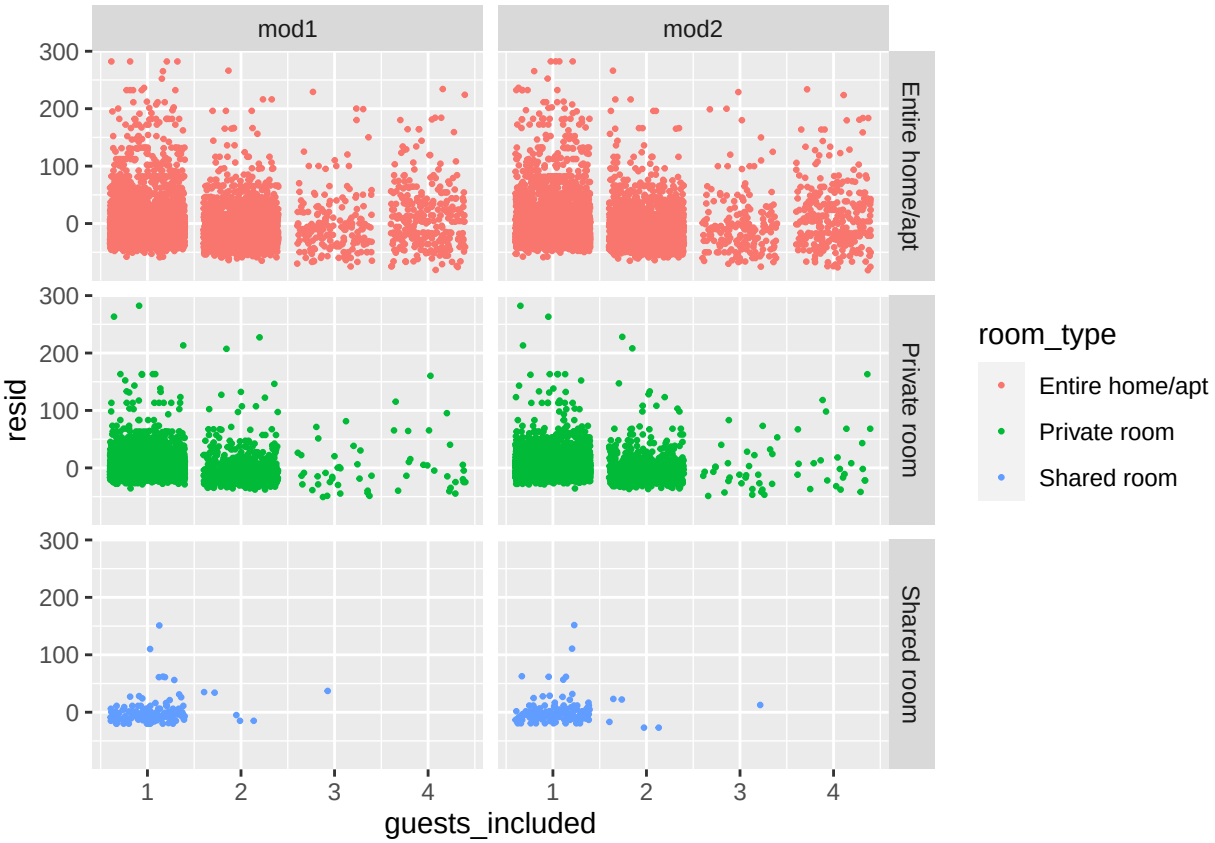
```
stargazer(mod2,
          star.cutoffs = c(0.05, 0.01, 0.001),
          keep.stat = c("n","rsq"),
          type = 'text', single.row=TRUE)
```

```
##
## ============================================================================
##                                          Dependent variable:
##                                      ---------------------------
##                                                 price
## ----------------------------------------------------------------------------
## guests_included                              16.215*** (0.513)
## room_typePrivate room                        -29.635*** (1.521)
## room_typeShared room                         -51.655*** (11.402)
## guests_included:room_typePrivate room          -1.201 (1.168)
## guests_included:room_typeShared room           12.320 (10.558)
## Constant                                     51.431*** (0.885)
## ----------------------------------------------------------------------------
## Observations                                      12,317
## R2                                                 0.320
## ============================================================================
## Note:                                  *p<0.05; **p<0.01; ***p<0.001
```

We then plot the residuals to further confirm the validity of the independent model. As shown in the graphs below, the residuals are largely random for both models given the fact that they are grouped together because of the faceted nature.

```
# Residuals plotting
train_mod <- train %>%
  gather_residuals(mod1, mod2)

ggplot() +
  geom_jitter( data = train_mod, aes(x = guests_included,
                                     y = resid, colour = room_type), size =.5) +
  facet_grid(room_type ~ model, space = "free")
```



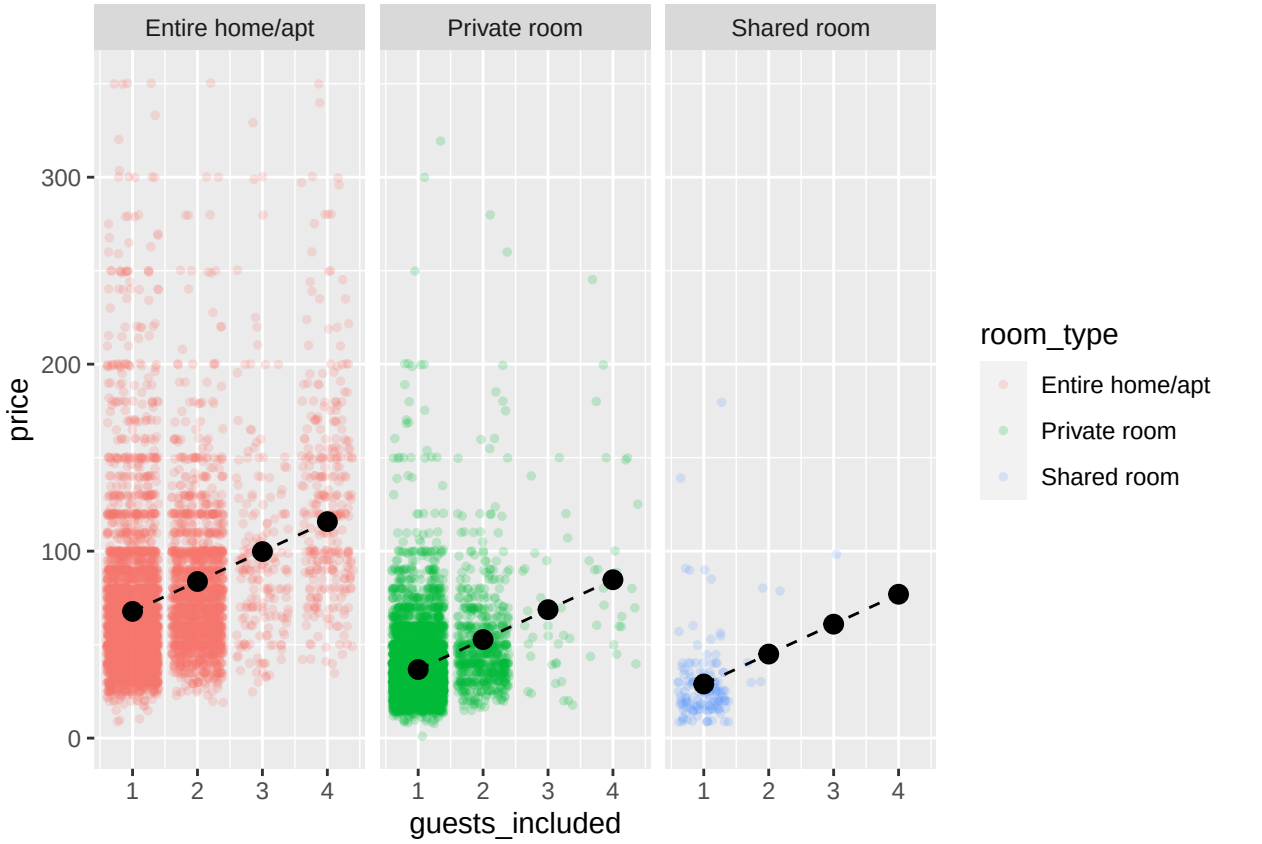
Finally, we added the predictions and the test sets to the graph to confirm the correlation using the more reliable model – the independent one.

```
# Add predictions
grid1 <- train %>%
  data_grid(guests_included, room_type) %>%
  add_predictions(mod1, "pred_price")

test <- test %>%
  data_grid(guests_included, room_type) %>%
  add_predictions(mod1, "pred_price")

ggplot()+
  geom_jitter(data = train, aes(x = guests_included, y = price, colour = room_type),
              size = 1, alpha = .2) +
```

```
geom_line ( data = grid1, aes(x = guests_included, y = pred_price),
            linetype = 'dashed') +
geom_point(data = test, aes(x = guests_included, y = pred_price) , size = 3) +
facet_wrap(~ room_type)
```



5. Evaluation of model C

We use a linear regression model of an Airbnb's daily price as a function of the number of guests included in the rent and the room's type. Through model analysis, we were able to identify the lack of interaction between the two explanatory parameters, which makes the abstract equation similar to this:

$$price = Intercept + A * guests\_included + B * room\_type$$

The challenge that interferes with regression model's assumptions is that the plot of the residuals. Considering that the data was large left-skewed, the residuals are not randomly distributed on the plot. It might be that a linear model would not describe accurately the relationship and some unseen trends might have been missed. The residuals versus fitted values plot is difficult to interpret as well.

Evaluation and Deployment

1. Evaluation

Based on the question "Can we predict the rating for a listing based on the location, price, or popularity, and what neighborhoods have the greatest influence on those ratings?" the analysis performed shows us that these factors have little to do with the rating given to the AirBnB after the stay largely due to the mostly high ratings that the AirBnBs received that skewed the data. Though there was little correlation between the

all variable combinations attempted when modeling, we did achieve the production of a reliable predictive models.

Yet, through this report, our business research questions have certainly been answered in parts. For a prospective customer looking for a cost-optimized trip, they should opt for listings in places that are faraway from the centers, have shared rooms or private rooms, and include as few guests as possible. For one that looks for places that have the highest review scores, Charlottenburg-Wilm. and Tempelhof - Schöneberg are the place to go. For potential hosts that look to optimize their earnings, using an entire home/ apartment or making the number of guests the most they can would definitely have an impact.

2. Future actions

Rabayda: I'd definitely try to go in a different direction if I was to analyze this dataset further. I learned a lot trying to predict for ratings of Airbnb reviews but if I were to do it again I'd try to use a different dataset altogether due to the nature of AirBnB reviews being very high in general for Berlin and I felt that most of the predictions couldn't answer our question well since most of the neighborhoods all had great ratings anyway.

Pham: I would look into more factors that influence the business research questions mentioned above. Though we are able to prove some of our hypotheses, they are genuinely not too helpful because of their easiness to infer from common sense. Generally, I want to continue with incorporating more explanatory variables into our models.

3. Experience documentation

Rabayda: The experience I gained during this project was great, or at least I felt that it was great. I remember during the intro weeks plotting the Airbnb data just as simple scatter plots with no transformations and the plots were practically unreadable. Data cleaning and formatting really left an impact on how I approach problems in data science and how I think about data science whenever I read about it in the news since I saw it could be so easy to filter out data in pursuit of finding a correlation/s between variables, which I'm sure has a large presence in the dark corners of data ethics. Transforming variables with skewed distributions also left an impact on my tenacity as a programmer after getting stuck with an error saying there was an NA/NaN/Inf within my dataframe when I couldn't find anything like it. 2 hours and realizing that -Inf was also a possibility to be present in said dataframe and I felt like I had run a marathon when I was finally able to run the chunk error-free.

Pham: I found the application of data science into business very interesting – the CRISP-DM method has helped me a lot in structuring this report. Plotting and adjusting the aesthetics was especially fun, and I learned the hard way that data cleaning and initial exploratory/ distribution graphs are very important to direct a project in its right way.