# Assignment 8

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Housekeeping

```
library(tidyr)
library(ggplot2)
library(readr)
library(tibble)
library(modelr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
hotels_features <- read_csv("hotels-europe_features.csv")
```

```
## Parsed with column specification:
## cols(
##   hotel_id = col_double(),
##   city = col_character(),
##   distance = col_double(),
##   stars = col_double(),
##   rating = col_double(),
##   country = col_character(),
##   city_actual = col_character(),
##   rating_reviewcount = col_double(),
##   center1label = col_character(),
##   center2label = col_character(),
##   neighbourhood = col_character(),
##   ratingta = col_double(),
##   ratingta_count = col_double(),
##   distance_alter = col_double(),
##   accommodation_type = col_character()
## )
```

```
hotels_price <- read_csv("hotels-europe_price.csv")
```

```
## Parsed with column specification:
## cols(
##   hotel_id = col_double(),
##   price = col_double(),
```

```
##   offer = col_double(),
##   offer_cat = col_character(),
##   year = col_double(),
##   month = col_double(),
##   weekend = col_double(),
##   holiday = col_double(),
##   nnights = col_double(),
##   scarce_room = col_double()
## )
```

```
vienna <- read_csv("hotels-vienna.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   country = col_character(),
##   city_actual = col_character(),
##   center1label = col_character(),
##   center2label = col_character(),
##   neighbourhood = col_character(),
##   city = col_character(),
##   offer_cat = col_character(),
##   accommodation_type = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

Assignment

**Exercise 1.**

Merge the two Europe datasets into one. Which hotels do not have matches in the price data? Use anti_join.

```
hotels <- hotels_features %>% inner_join(hotels_price)
```

```
## Joining, by = "hotel_id"
```

```
colSums(is.na(hotels))
```

```
##           hotel_id               city           distance
##                  0                  0                  0
##              stars             rating            country
##              26393              10523                  0
##       city_actual rating_reviewcount       center1label
##                  0              10523                  0
##       center2label      neighbourhood           ratingta
##                  0                  0              12931
##     ratingta_count     distance_alter accommodation_type
##              12931                  0                 65
##              price              offer          offer_cat
##                  0                  0                  0
##               year              month            weekend
##                  0                  0                  0
##            holiday            nnights        scarce_room
##                  0                  0                  0
```

```
vienna <- hotels %>% filter(city == "Vienna")
vienna <- na.omit(vienna)
```

```
# view(hotels)
#These hotels have no match in the price data
hotels_price %>% anti_join(hotels_features)
```

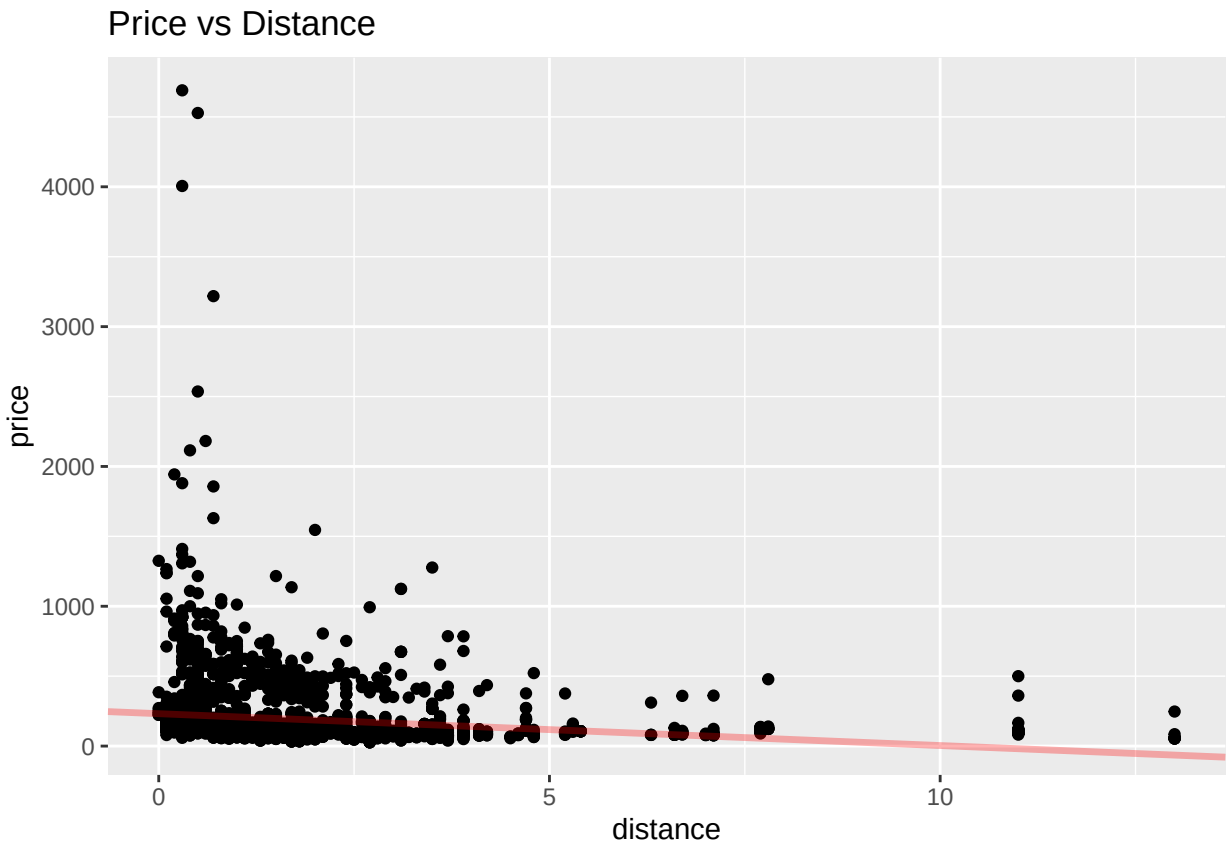
```
## Joining, by = "hotel_id"
```

```
## # A tibble: 3 x 10
##   hotel_id price offer offer_cat  year month weekend holiday nnights
##      <dbl> <dbl> <dbl> <chr>     <dbl> <dbl>   <dbl>   <dbl>   <dbl>
## 1        2   119     0 0% no of~  2017    11       0       0       1
## 2        2   119     0 0% no of~  2017    12       0       1       1
## 3        2   547     0 0% no of~  2017    12       0       1       4
## # ... with 1 more variable: scarce_room <dbl>
```

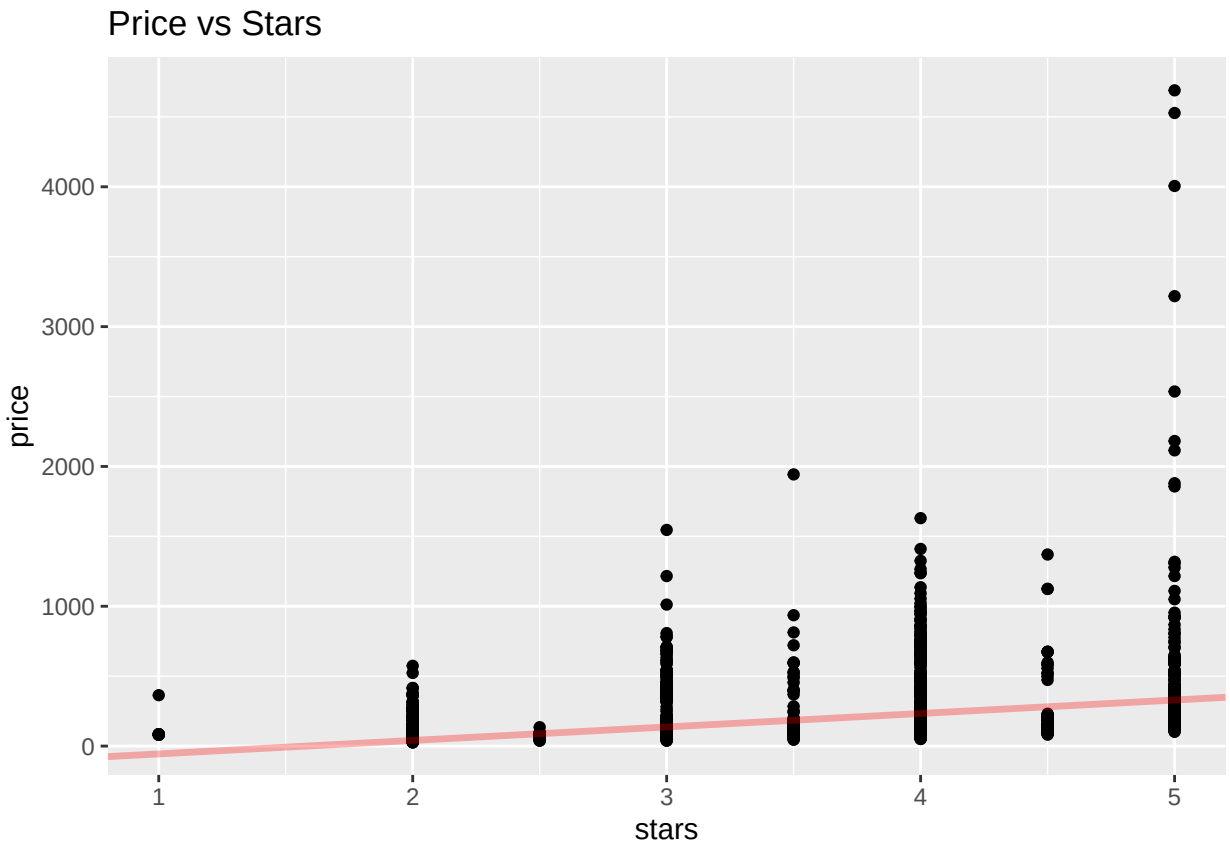
**Exercise 2.**

Come up with 3 models to explore what drives the price of hotels in Vienna (later you'll do it for another city too). Plot the price of hotels against some of the main variables that seem to have the most explanatory power.

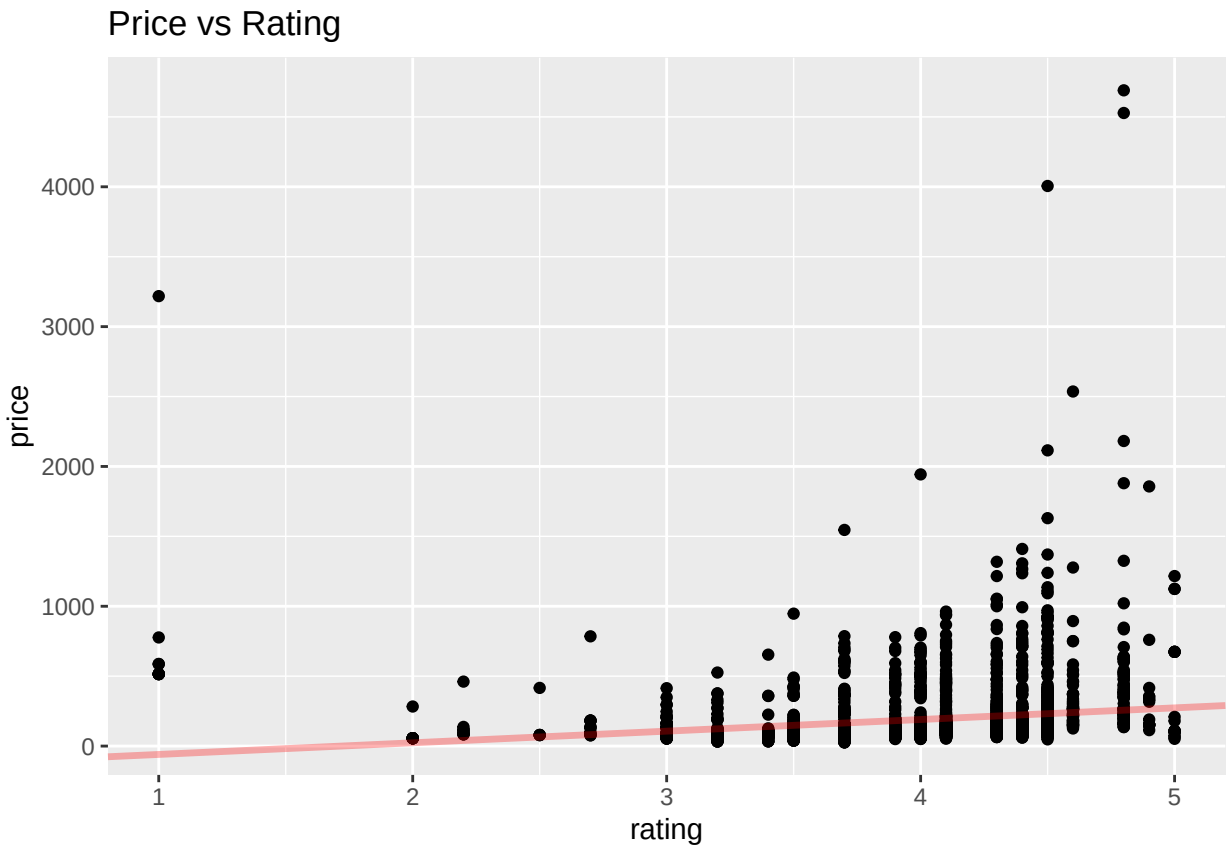
```
price_dist_model <- lm(price ~ distance, data = vienna)
price_star_model <- lm(price ~ stars, data = vienna)
price_rating_model <- lm(price ~ rating, data = vienna)

pred_vienna <- vienna %>% add_predictions(price_dist_model)
#price - distance
ggplot(vienna, aes(distance,price)) +
  geom_point() +
  geom_abline(intercept = price_dist_model$coefficients[1],
              slope = price_dist_model$coefficients[2], color = "Red", size = 1.2, alpha = 0.3) + labs(
```




```
#price - No. of stars
ggplot(pred_vienna, aes(stars,price)) +
  geom_point() +
  geom_abline(intercept = price_star_model$coefficients[1],
              slope = price_star_model$coefficients[2], color = "Red", size = 1.2, alpha = 0.3) +
  labs(title = "Price vs Stars")
```




```
#+ geom_point(data = vienna, aes(y = price_dist_model), colour = "red", size = 4)

#price - rating
ggplot(vienna, aes(rating,price)) +
  geom_point() +
  geom_abline(intercept = price_rating_model$coefficients[1],
              slope = price_rating_model$coefficients[2], color = "Red", size = 1.2, alpha = 0.3) +
  labs(title = "Price vs Rating")
```




```
# (models <- tibble(
#   a1 = runif(1000, 50, 650),
#   a2 = runif(1000, -50, 50)
# ))
#
# ggplot(vienna, aes(distance,price)) +
#   geom_abline(data = models,
#               aes(intercept = a1, slope = a2),
#               alpha = 1/4) +
#   geom_point()
```

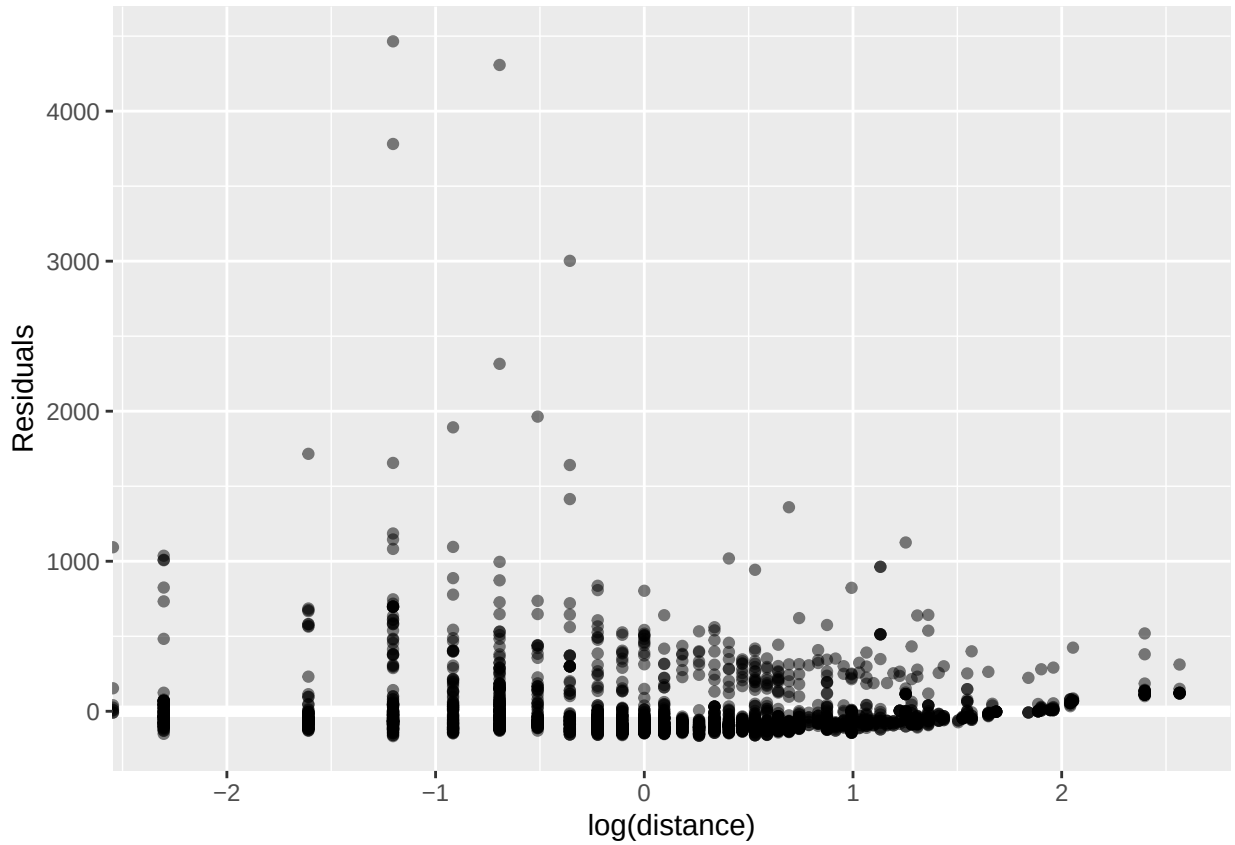
```
# model1 <- function(a, data) {
#   a[1] + data$x * a[2]
# }
#
# # MSE:Mean-squared error
# measure_distance <- function(mod, data) {
#   diff <- data$y - model1(mod, data)
#   sqrt(mean(diff^2))
# }
#
# sim1_dist <- function(a1, a2) {
#   measure_distance(c(a1, a2), sim1)
# }
#
# models <- models %>%
```

```
#   mutate(dist = purrr::map2_dbl(a1, a2, sim1_dist))
#
# models
#
# # Get the top models
# (top_models <- models %>%
#     filter(rank(dist) <= 10))
#
# ggplot(vienna, aes(distance,price)) +
#   geom_point() +
#   geom_abline(data = top_models,
#               aes(intercept = a1, slope = a2, color = -dist))
#
# ggplot(models, aes(a1, a2, color = -dist)) +
#   geom_point(data = filter(models, rank(dist) <= 10),
#              size = 4,
#              color = 'red') +
#   geom_point()
#
# best <- optim(c(0,0), measure_distance, data = sim1)
# best$par
#
# sim1_mod <- lm(y ~ x, data = sim1)
# sim1_mod
#
# ggplot(sim1, aes(x,y)) +
#   geom_point() +
#   geom_abline(intercept = sim1_mod$coefficients[1],
#               slope = sim1_mod$coefficients[2])
```

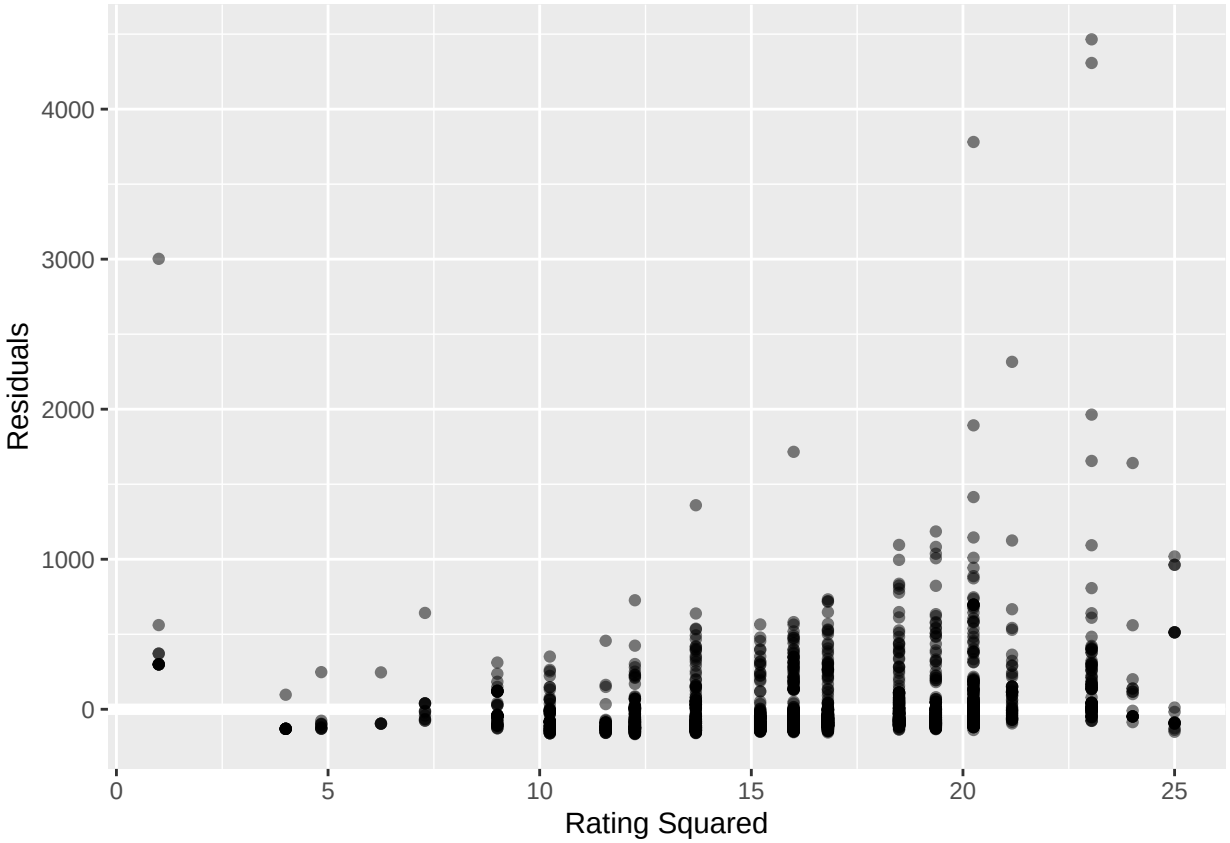
**Exercise 3.**

Look at the residuals from your favorite model. Now plot these against 3 variables that you didn't include earlier - or if you included all plot them agains some variables in a form that you did not do before (squares, logs, interactions). Do the residuals look random?

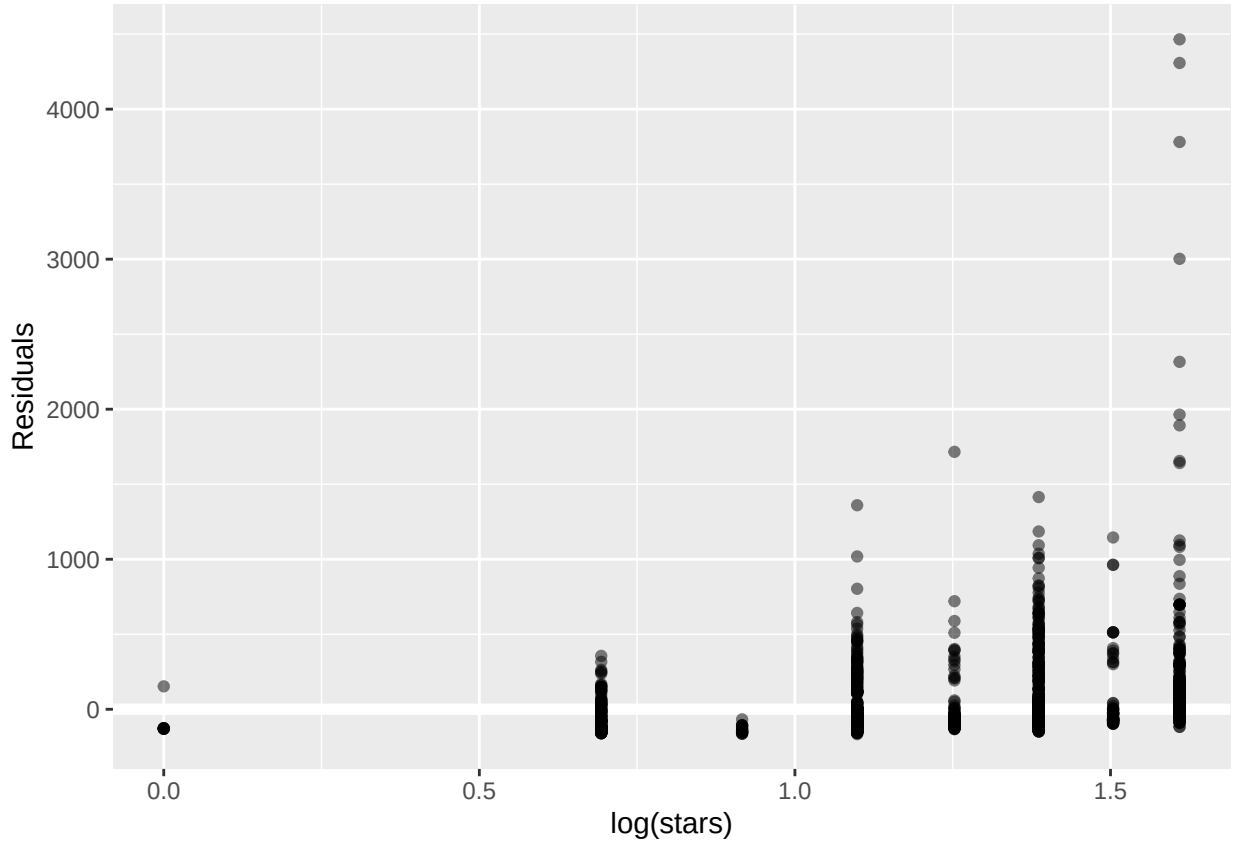
```
ggplot(vienna, aes(log(distance), price_dist_model$residuals)) + geom_ref_line(h = 0)  +
  geom_point(alpha = 0.5) + labs(Title = "Residuals vs log of Distance", y = "Residuals")
```




```
ggplot(vienna, aes(rating^2, price_dist_model$residuals)) + geom_ref_line(h = 0) +
  geom_point(alpha = 0.5) + labs(Title = "Residuals vs log of Distance", x = "Rating Squared", y = "Resi
```




```
ggplot(vienna, aes(log(stars), price_dist_model$residuals)) + geom_ref_line(h = 0) + geom_point(alpha =
```



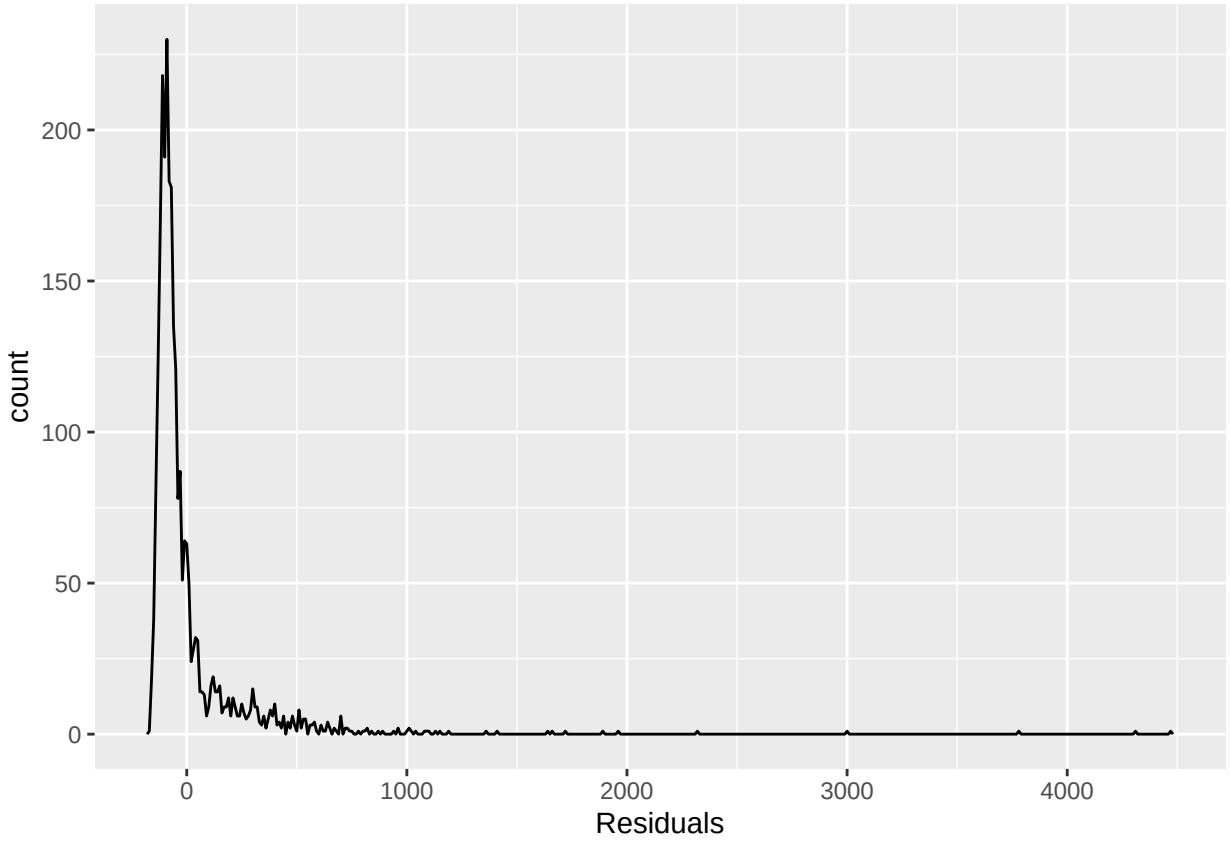

**Ans:** The residuals look only slightly random when potted against the log of the the distance,stars and rating. We will need to explore more to give a definite verdict.

**Exercise 4.**

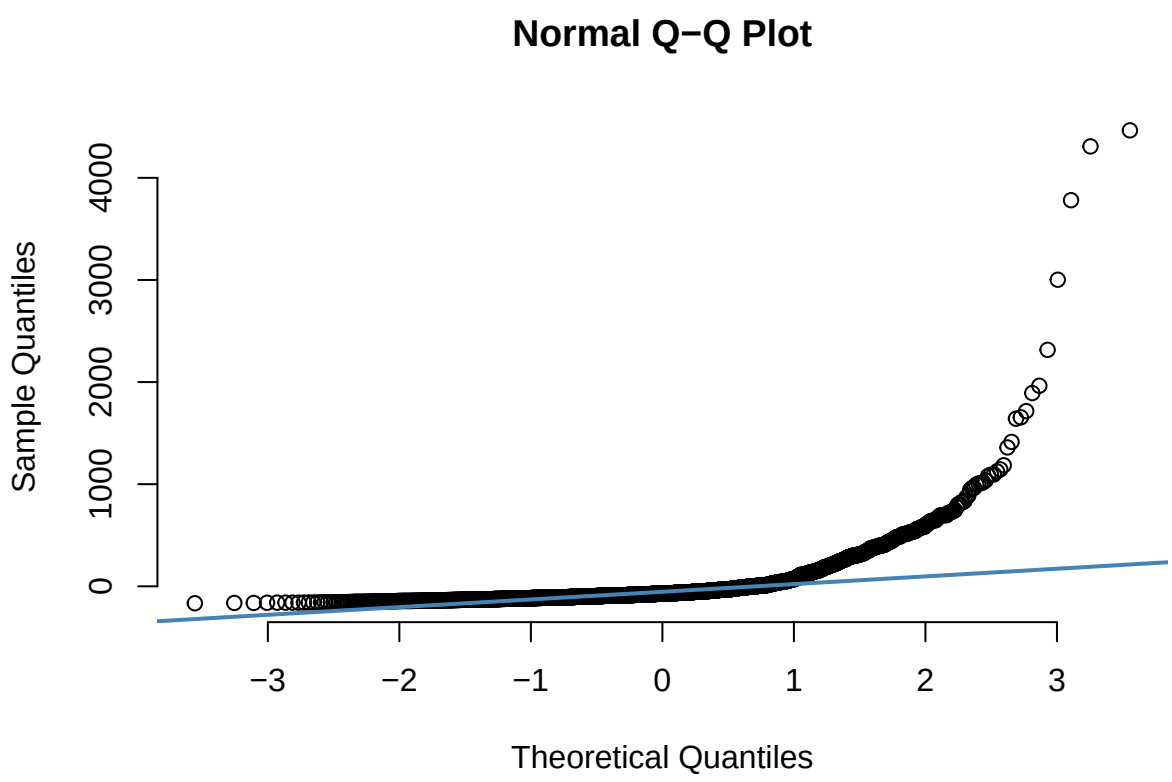
Try to find out some methods to check if residuals are random and how to code this in R. Find at least two ways.

**Ans:** I have used frequence polygon and QQplot to check for randomness.

```
ggplot(price_dist_model, aes(price_dist_model$residuals)) +
  geom_freqpoly(binwidth = 10) + labs(Title = "Frequency Polygon of Residuals", x = "Residuals")
```




```
qqnorm(price_dist_model$residuals, pch = 1, frame = FALSE)
qqline(price_dist_model$residuals, col = "steelblue", lwd = 2)
```



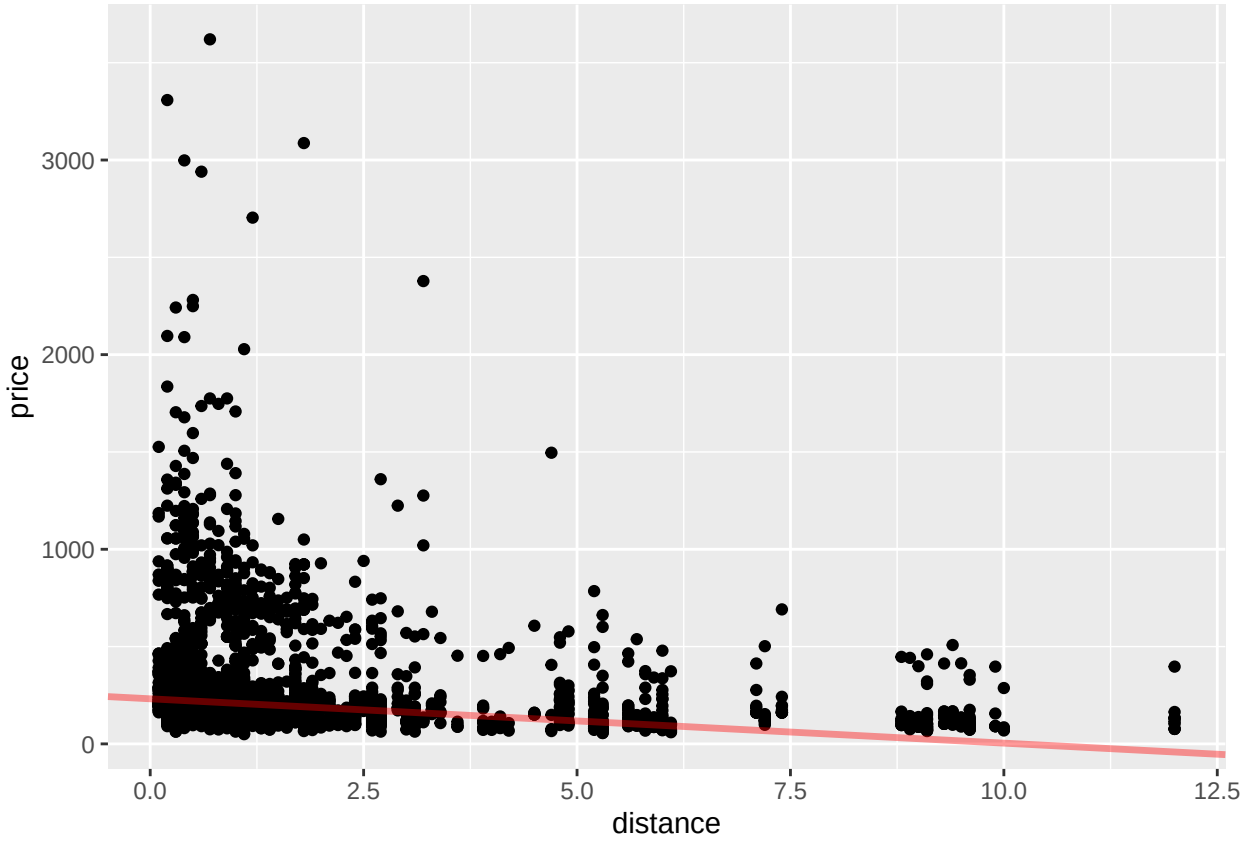

**Exercise 5.**

Use your model estimated on Vienna to predict prices for another city. How well does your model do, if you use the parameter values estimated on Vienna? Example: In part 3 you will have run some model similar to vienna_model <- lm(y ~ x, data = my_vienna_data) for some such. Use the vienna_model, with the parameters estimated on the Vienna dataset to see how well it does (and what it predicts) for some other city. the notes on lecture8-pre-class.R contain code that computes predictions. You pick how you measure 'how well it does'.

```
amsterdam <- hotels %>% filter(city == "Amsterdam")
amsterdam_pred <- hotels %>% filter(city == "Amsterdam") %>% add_predictions(price_dist_model)

price_new_model <- lm(pred ~ distance, data = amsterdam_pred)

n <- amsterdam_pred %>% nrow()
n
```

```
## [1] 2055
```

```
ggplot(amsterdam_pred, aes(distance, price)) +
  geom_point() + geom_abline(intercept = price_new_model$coefficients[1], slope = price_new_model$coeff
```




```r
# 2. R2 Score components

# 2.1. Average of actual data
avr_y_actual <- mean(amsterdam_pred$price)

avr_y_pred <- mean(amsterdam_pred$pred)

# 2.2. Total sum of squares
ss_actual <- sum((amsterdam_pred$price - avr_y_actual)^2)/n

# 2.3. Regression sum of squares
ss_pred <- sum((amsterdam_pred$pred - avr_y_pred)^2)/n

# 2.4. Residual sum of squares
ss_residuals <- sum((amsterdam_pred$price - amsterdam_pred$pred)^2)

# 3. R2 Score
r2 <- ss_pred / ss_actual
```

**Ans:** Just 2.7% of the variation in the model is explained by the variation in the data. This means that our model is not a good predictor of hotel prices in Amsterdam.

**Exercise 6.**

Reestimate your favorite model (the same as from exercise 1) for another city. How much do the parameter values change? How well does the model do now? If you had to summarise the difference between the 2 cities in one number based on your model, what would it be? Example: In this exercise, you should run the same

formula as in exercise 3, but run it on the data for another city (the same one as in exercise 5). Thus if you ran lm(y ~ x, data = my_vienna_data), you now run lm(y ~ x, data = my_other_city_data)

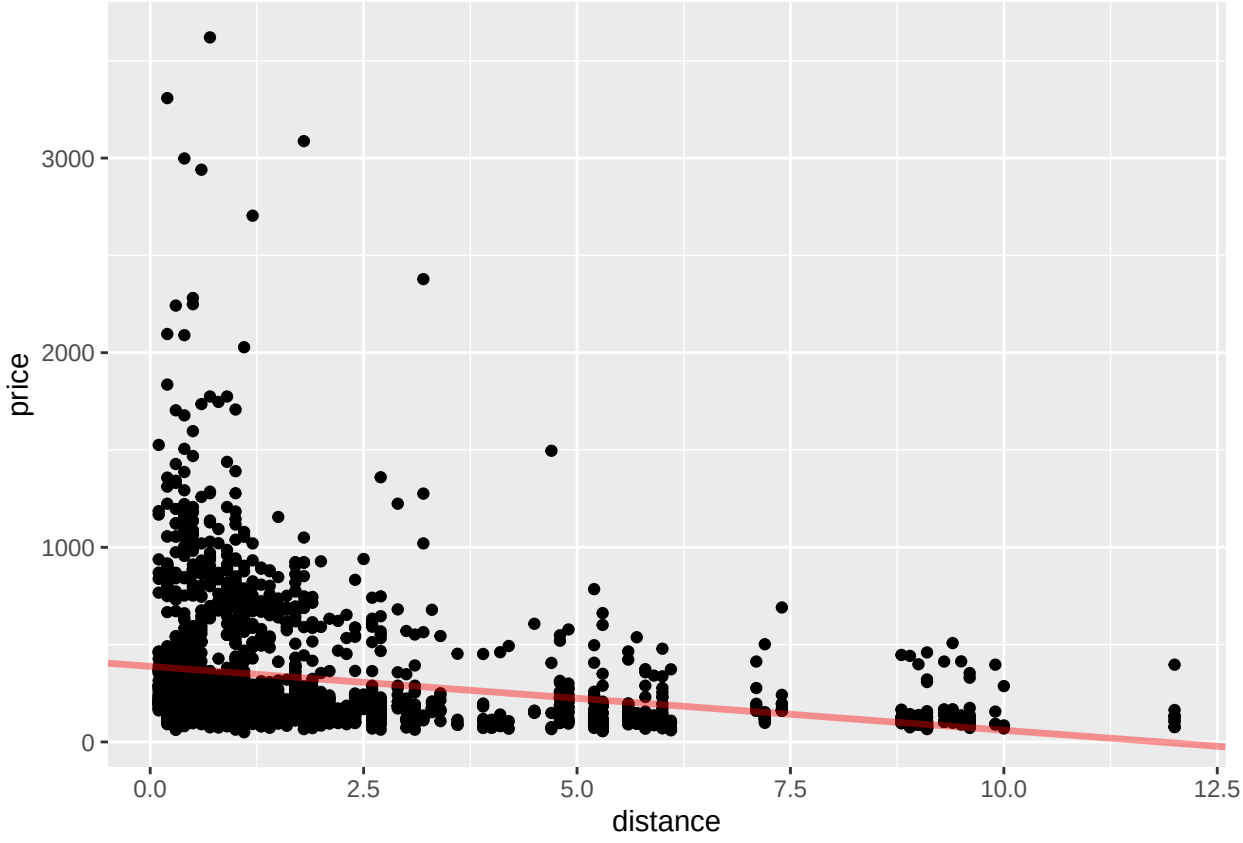
```
price_dist_ams <- lm(price ~ distance, data = amsterdam)

amsterdam_pred <- amsterdam %>% add_predictions(price_dist_ams)

price_new_model$coefficients
```

```
## (Intercept)    distance
##   231.69896   -22.78177
```

```
price_dist_ams$coefficients
```

```
## (Intercept)    distance
##   388.31396   -32.82638
```

```
ggplot(price_dist_ams, aes(distance,price)) +
  geom_point() +
  geom_abline(intercept = price_dist_ams$coefficients[1],
              slope = price_dist_ams$coefficients[2], color = "Red", alpha = 0.4, size = 1.2)
```




```
R2 <- (cor(amsterdam_pred$price, price_dist_ams$fitted.values))^2

avr_y_actual <- mean(amsterdam_pred$price)

avr_y_pred <- mean(amsterdam_pred$pred)
```

```r
# 2.2. Total sum of squares
ss_actual <- sum((amsterdam_pred$price - avr_y_actual)^2)/n

# 2.3. Regression sum of squares
ss_pred <- sum((amsterdam_pred$pred - avr_y_pred)^2)/n

# 2.4. Residual sum of squares
ss_residuals <- sum((amsterdam_pred$price - amsterdam_pred$pred)^2)

# 3. R2 Score
r2 <- ss_pred / ss_actual
```

**Ans:** Parameter values have changed to: Slope has increase to -32 from -12 which means that for every km increase in distance the price drops by $32. Intercept has increased to 338 from 148 which means that hotels at the city centre are on average over $100 more expensive in Amsterdam

R2 value has doubled to 0.06, which means that around 6% of the variance in the regression is explained by the data which means this model is doing better then the one we made using the vienna dataset.

If I have to summarise and compare both cities I would compare by the slope of our linear regression which shows how much price changes for every km change in distance-