Investigation of the effects of restaurant characteristics on average ratings

Expanded Report

Felipe Campelo

November 18, 2015

Introduction

In this study I'll investigate the factors that may affect the average ratings of restaurants. The primary question of this work can be stated as: ***What are the variables that have the largest impact in the average ratings of restaurants, and what is the magnitude of their effects?***

To that end the information available in the Yelp Dataset will be employed.

Disclaimer: This work was developed for the *Data Science Specialization Capstone Project*. It is not intended as an exhaustive evaluation of all possibilities, nor as a scientific evaluation of causal relationships - its very nature as a retrospective analysis based on self-selected reporting (i.e., the reviews available in the Yelp dataset) precludes inferences of causality, and allows only (at most) conclusions about correlational effects.

Methods and Data

In this section the methods used for each step of the analysis are introduced. The analysis is divided into four main steps. The methods employed for each one are described in the subsections below, and the results of each step are detailed in the **Results** section.

Data Cleanup and Exploratory Analysis

Initially the compressed data was obtained from the Yelp Dataset Challenge downloads page. After downloading and decompressing the data, the *JSON* files containing the review, business, tip, checkin and business information were parsed using the jsonlite package, and the resulting dataframes were saved to **RDS** files, which are smaller and quicker to load.

To address the question of interest of this work, some filtering was necessary to isolate the relevant observations from the superfluous information. The following filtering steps were performed:
- Only observations of businesses categorized as *restaurant* were used;
- Only restaurants with more than 30 reviews were used (to get some level of robustness on the estimator of the average rating);
- Attributes that do not make sense for a restaurant (e.g., *Hair Types Specialized In*) were removed;
- Attributes with more than 50% missing observations were removed;
- Observations that contained attributes with extremely low occurrences (e.g., *Attire == "formal"*) were removed to prevent extremely unbalanced situations;
- Only complete cases (i.e., cases with no missing attribute values) were considered;
- For each restaurant, the output variable was calculated as the average number of stars given to that establishment.

After the initial data exploration and cleanup were performed, the relevant data for the question of interest was consolidated as a single data frame, which was also saved in the **RDS** format for faster loading.

Statistical Modeling

To investigate the most influential factors in the average ratings, an analysis of variance model (linear regression on categorical predictors) was fitted to the data. Afterwards, an AIC-based stepwise pruning routine was used to obtain a more parsimonious model, by removing the terms which do not contribute significantly to the overall explanatory power of the model.

Graphical Analysis

Effect sizes were calculated for all remaining terms of the regression model and plotted, to allow a graphical investigation of the strongest effects on the average star rating. The most important effects were then isolated and used to generate effects plots, which were used as the basis for my final considerations and discussions.

Results

Data Cleanup and Exploratory Analysis

Initially, the **JSON** files were parsed and saved as **RDS**:

```
library(jsonlite)
filenames <- c("business", "checkin", "tip", "user", "review")
files     <- paste0("../data/yelp_academic_dataset_", filenames, ".json")
outnames  <- paste0("../data/0_", filenames, ".rds")
for (i in 1:5){
  mydata <- jsonlite::stream_in(con = file(files[i ]), pagesize = 10000)
  saveRDS(mydata, outnames[i])
  assign(filenames[i], mydata)
}
rm(mydata) # free memory
```

To visualize the location of the cities, I plotted the data using the **maps** package:

```
library(maps)
map("world",
    ylim = c(20, 60),
    xlim = c(-130, 20),
    col = "gray60")
points(business$longitude,
       business$latitude,
       pch = 20,
       col = "cyan4")
```

After that the restaurants that had more than 30 reviews were detected:

```
getRestID <- function(i, V, nrev){
  if(any(c("restaurants", "restaurant") %in% tolower(V$categories[[i]]))
     && V$review_count[i] >= nrev){
    return(V$business_id[i])} else return(NA)
}

restIDs <- na.exclude(unlist(lapply(1:nrow(business), FUN  = getRestID,
                                    V = business, nrev = 30)))
```

I then proceeded to do some more data cleanup and preprocessing :

- Merge *Good for Kids* and *Good For Kids* columns (notice the case difference);
- Convert the *Accepts Credit Cards* variable to logical;
- Removed (useless) attributes *Hair Types Specialized In* and *Accepts Insurance*;

```
# 1) Merge $`Good for Kids` and $`Good For Kids` columns
business$attributes$`Good for Kids`[is.na(business$attributes$`Good for Kids`)] <- 0
business$attributes$`Good for Kids` <- business$attributes$`Good for Kids` ||
    business$attributes$`Good For Kids`
business$attributes$`Good For Kids` <- NULL

# 2) Convert $`Accepts Credit Cards` to logical
business$attributes$`Accepts Credit Cards` <-
    lapply(business$attributes$`Accepts Credit Cards`, function(x) if(!length(x)) x[[1]] <- NA else x)
business$attributes$`Accepts Credit Cards` <-
    unlist(business$attributes$`Accepts Credit Cards`)

# Remove incoherent columns
business$attributes$`Hair Types Specialized In` <- NULL
business$attributes$`Accepts Insurance`<- NULL
```

After that, a data frame with all the relevant information was built:

```
# Function to build list of relevant review information
getRestaurant <- function(ID, Xbus, Xrev){
  xbus <- subset(Xbus, business_id == ID)
  xrev <- subset(Xrev, business_id == ID)
```

```
  # get indices of columns of class "data frame" in xbus$attributes
  dfinds <- as.integer(which(unlist(lapply(business$attributes, is.data.frame))))

  try(rm(buscols))
  for (i in dfinds){
    newcols<- as.data.frame(xbus$attributes[, i])
    names(newcols) <- paste(names(xbus$attributes)[i],
                            names(newcols),
                            sep = ".")
    if(!exists("buscols")) {
      buscols <- newcols
    } else buscols <- cbind(buscols, newcols)
  }

  buscols <- cbind(xbus$attributes[, -dfinds], buscols)
  Stars   <- mean(xrev$stars)

  data.frame(buscols, Stars)
}

RestData <- lapply(X    = restIDs,
             FUN  = getRestaurant,
             Xbus = business,
             Xrev = review)

RestDF <- data.frame(
            matrix(unlist(RestData),
                   nrow  = length(RestData),
                   byrow = TRUE))
names(RestDF) <- names(RestData[[1]])

# Remove columns that are more than 50% NA
nainds <- which(
            unlist(
              parallel::mclapply(RestDF,
                                 function(x) sum(is.na(x))/length(x) > 0.50)))
RestDF <- RestDF[, -nainds]

# Remove levels / factors that would lead to extreme unbalance
summary(RestDF)
RestDF <- RestDF[-which(RestDF$Attire=="formal"), ]
RestDF$Attire <- factor(RestDF$Attire)

# Turn all independent variables into factors
RestDF[, -ncol(RestDF)] <- lapply(RestDF[, -ncol(RestDF)], as.factor)

#Turn the dependent variable into numeric
RestDF$Stars <- as.numeric(as.character(RestDF$Stars))

# Select only complete cases
RestDF <- RestDF[complete.cases(RestDF), ]

saveRDS(RestData, "../data/RestData.rds")
```

```
saveRDS(RestDF,   "../data/RestDF.rds")
```

At the end of the process, a data frame with the following fields was generated:

Accepts.Credit.Cards	Good.For.Groups	Outdoor.Seating	Takes.Reservations	Wheelchair.Accessible
FALSE: 116	FALSE: 303	FALSE:2444	FALSE:2693	FALSE: 300
TRUE :4954	TRUE :4767	TRUE :2626	TRUE :2377	TRUE :4770

Parking.garage	Parking.street	Parking.validated	Parking.lot	Parking.valet	Good.For.latenight
FALSE:4386	FALSE:4104	FALSE:5016	FALSE:1417	FALSE:4670	FALSE:4678
TRUE : 684	TRUE : 966	TRUE : 54	TRUE :3653	TRUE : 400	TRUE : 392

Take.out	Ambience.intimate	Ambience.classy	Ambience.hipster	Ambience.divey	Ambience.touristy
FALSE: 449	FALSE:4974	FALSE:4833	FALSE:4962	FALSE:4810	FALSE:5021
TRUE :4621	TRUE : 96	TRUE : 237	TRUE : 108	TRUE : 260	TRUE : 49

Delivery	Wheelchair.Accessible	Ambience.trendy	Ambience.upscale	Ambience.casual	Good.For.dessert
FALSE:4317	FALSE: 300	FALSE:4779	FALSE:4963	FALSE:1257	FALSE:4971
TRUE : 753	TRUE :4770	TRUE : 291	TRUE : 107	TRUE :3813	TRUE : 99

Good.For.lunch	Good.For.dinner	Good.For.breakfast	Good.For.brunch	Waiter.Service	Has.TV
FALSE:2362	FALSE:2537	FALSE:4594	FALSE:4766	FALSE:1249	FALSE:2111
TRUE :2708	TRUE :2533	TRUE : 476	TRUE : 304	TRUE :3821	TRUE :2959

Price.Range	Alcohol	Noise.Level	Attire	Wi.Fi	Caters	Stars
1:1579	beer_and_wine: 943	average :3947	casual:4803	free:1828	FALSE:2451	Min. :1.509
2:3053	full_bar :2634	loud : 366	dressy: 267	no :3190	TRUE :2619	1st Qu.:3.368
3: 334	none :1493	quiet : 654	NA	paid: 52	NA	Median :3.714
4: 104	NA	very_loud: 103	NA	NA	NA	Mean :3.664
NA	NA	NA	NA	NA	NA	3rd Qu.:4.022
NA	NA	NA	NA	NA	NA	Max. :4.897

In this summary dataframe all regressor variables were expressed as factors, the response variable (*Stars*) was a numeric vector, and only complete cases were present.

Statistical Modeling

The regression model was fit and simplified as follows:

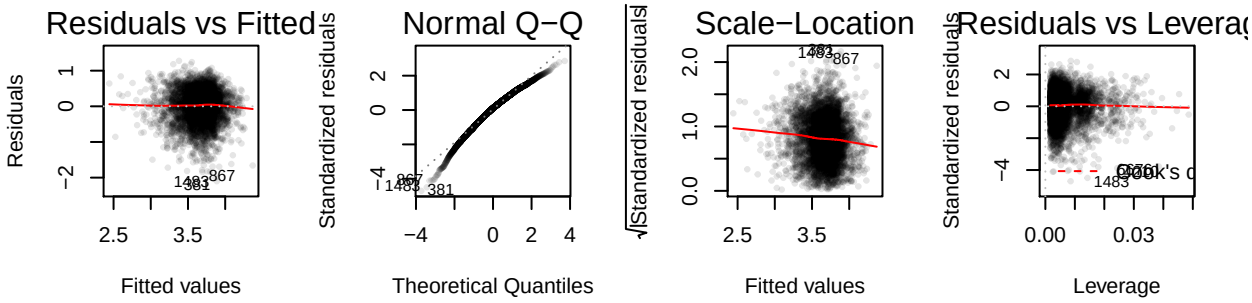
```
model1 <- lm(Stars ~ ., data = RestDF)
model2 <- step(model1, trace = 0)
summary.aov(model2)
```

```
##                          Df Sum Sq Mean Sq F value   Pr(>F)
## Accepts.Credit.Cards      1    2.9   2.862  14.225 0.000164 ***
## Good.For.Groups           1    6.2   6.246  31.048 2.65e-08 ***
## Outdoor.Seating           1    1.9   1.908   9.482 0.002086 **
## Price.Range               3   40.6  13.531  67.256  < 2e-16 ***
## Alcohol                   2   36.4  18.220  90.562  < 2e-16 ***
## Noise.Level               3   28.5   9.498  47.208  < 2e-16 ***
## Has.TV                    1    1.4   1.413   7.023 0.008071 **
## Attire                    1    1.4   1.374   6.831 0.008984 **
## Delivery                  1    0.4   0.351   1.746 0.186486
## Wi.Fi                     2   14.4   7.209  35.832 3.53e-16 ***
## Caters                    1   10.6  10.580  52.587 4.74e-13 ***
## Ambience.romantic         1    1.1   1.097   5.454 0.019559 *
## Ambience.intimate         1    3.0   3.046  15.140 0.000101 ***
## Ambience.classy           1    1.0   1.020   5.071 0.024368 *
## Ambience.hipster          1    6.3   6.346  31.546 2.05e-08 ***
## Ambience.touristy         1    7.4   7.411  36.837 1.38e-09 ***
## Ambience.trendy           1    5.7   5.665  28.156 1.17e-07 ***
## Ambience.casual           1    0.2   0.213   1.058 0.303635
## Good.For.latenight        1    3.4   3.447  17.132 3.54e-05 ***
## Good.For.lunch            1    3.6   3.569  17.739 2.58e-05 ***
## Good.For.dinner           1    0.1   0.076   0.378 0.538526
## Good.For.breakfast        1    7.8   7.755  38.546 5.78e-10 ***
## Parking.garage            1   18.4  18.393  91.422  < 2e-16 ***
## Parking.street            1    6.4   6.445  32.037 1.60e-08 ***
## Parking.lot               1    4.2   4.238  21.066 4.54e-06 ***
## Parking.valet             1    0.9   0.935   4.647 0.031162 *
## Residuals              5037 1013.4   0.201
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary.lm(model2)$r.squared
```

```
## [1] 0.1745809
```

```
par(mfrow = c(1,4)); plot(model2, pch=20, cex=.7, col = rgb(0,0,0,0.1))
```

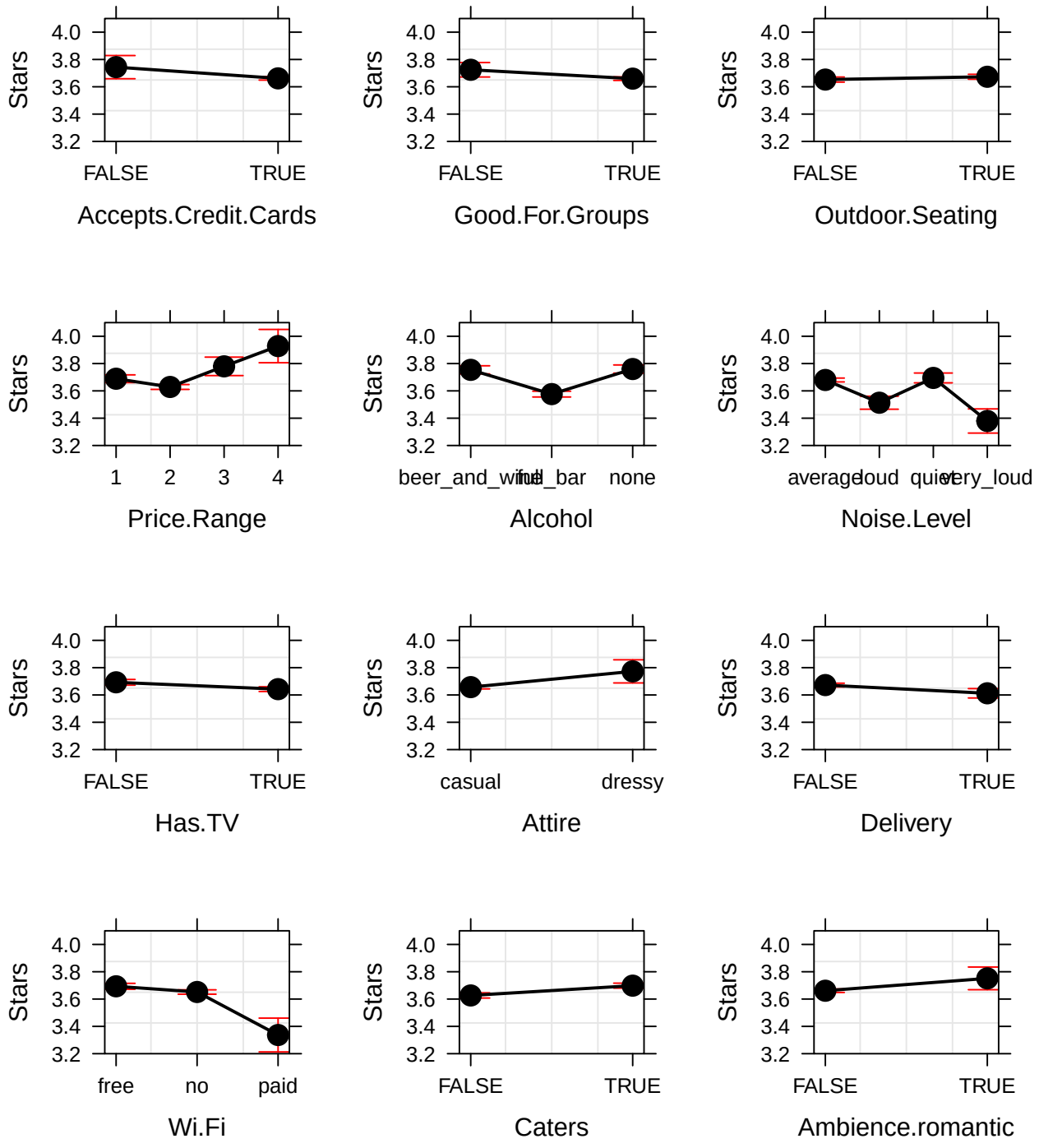


Notice that only the main effects were used as regressors, which makes this model rather simple but adequate for an initial investigation of the factor effects. The proportion of explained variability in the final model is relatively low (0.17458), which may be due to the effect of interactions being absorbed by the residual term in the model, as well as the effect of undocumented variables. The residual plot of *model2* suggests that no large violations of the verifiable assumptions of the linear model (normality, homoscedasticity) are likely to be present.

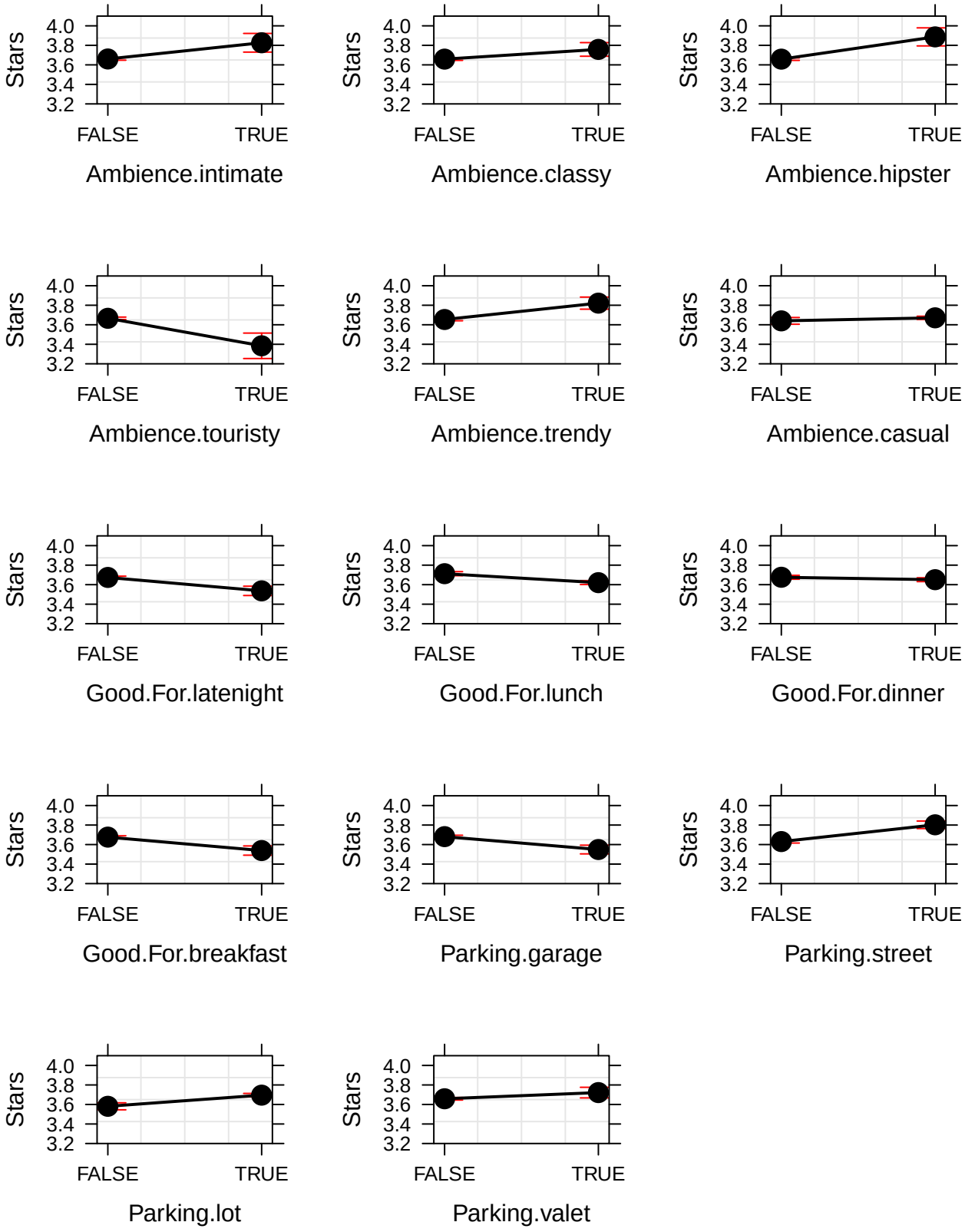
Graphical Analysis

The effect sizes and confidence intervals were calculated and plotted using the **effects** package.

```
library(effects)
Effs <- allEffects(model2)
plot(Effs[1:12], main = effect, ylim = c(3.2, 4.1),
     grid = TRUE, col = 3, row = 4)
```



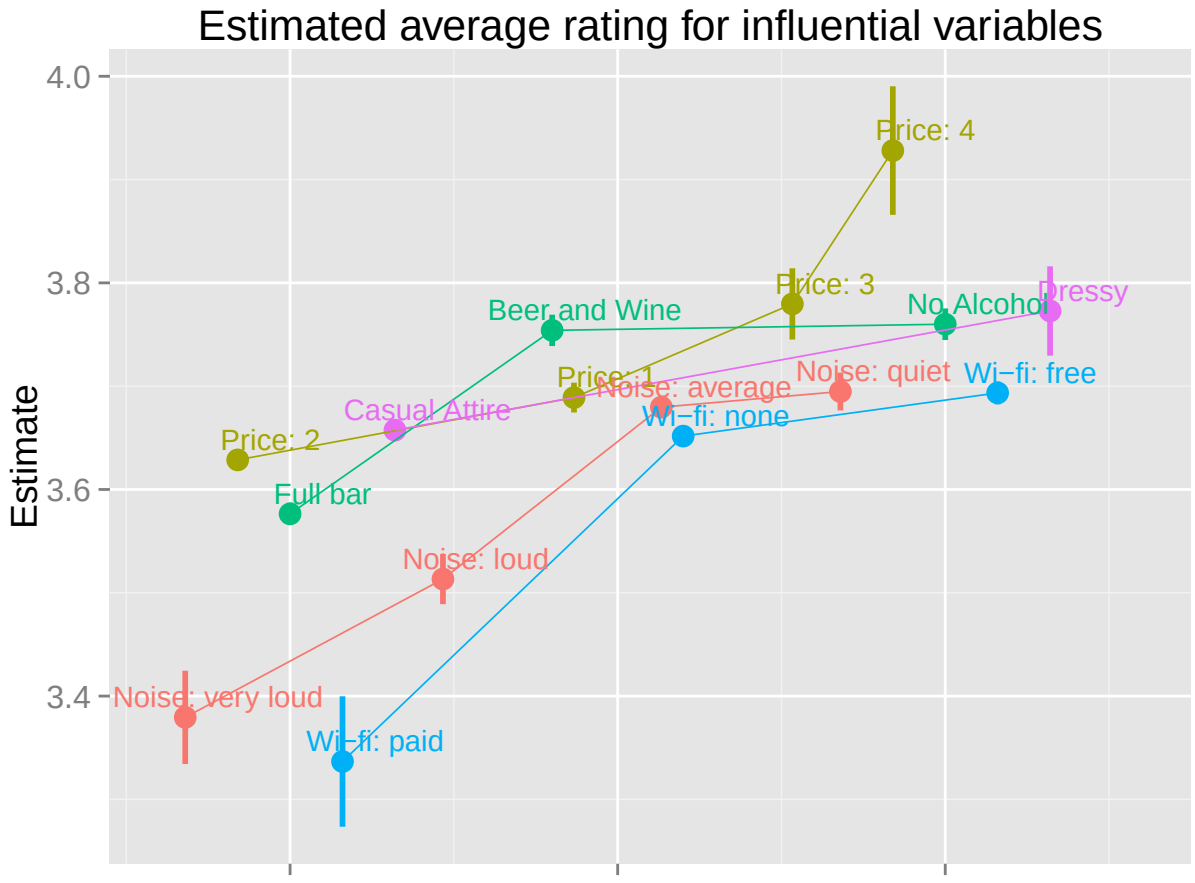
```
plot(Effs[13:26], main = effect, ylim = c(3.2, 4.1),
     grid = TRUE, col = 3, row = 5)
```



By examining this plot, two groups of variables were clearly shown to have interesting effects: The first were *Noise Level*, *Price Range*, *Alcohol*, *Wi-fi* and *Attire*, while the second were the ones related to the *Ambience* of the restaurant. To explore these effects in more detail, two plots were generated using the **ggplot2** package:

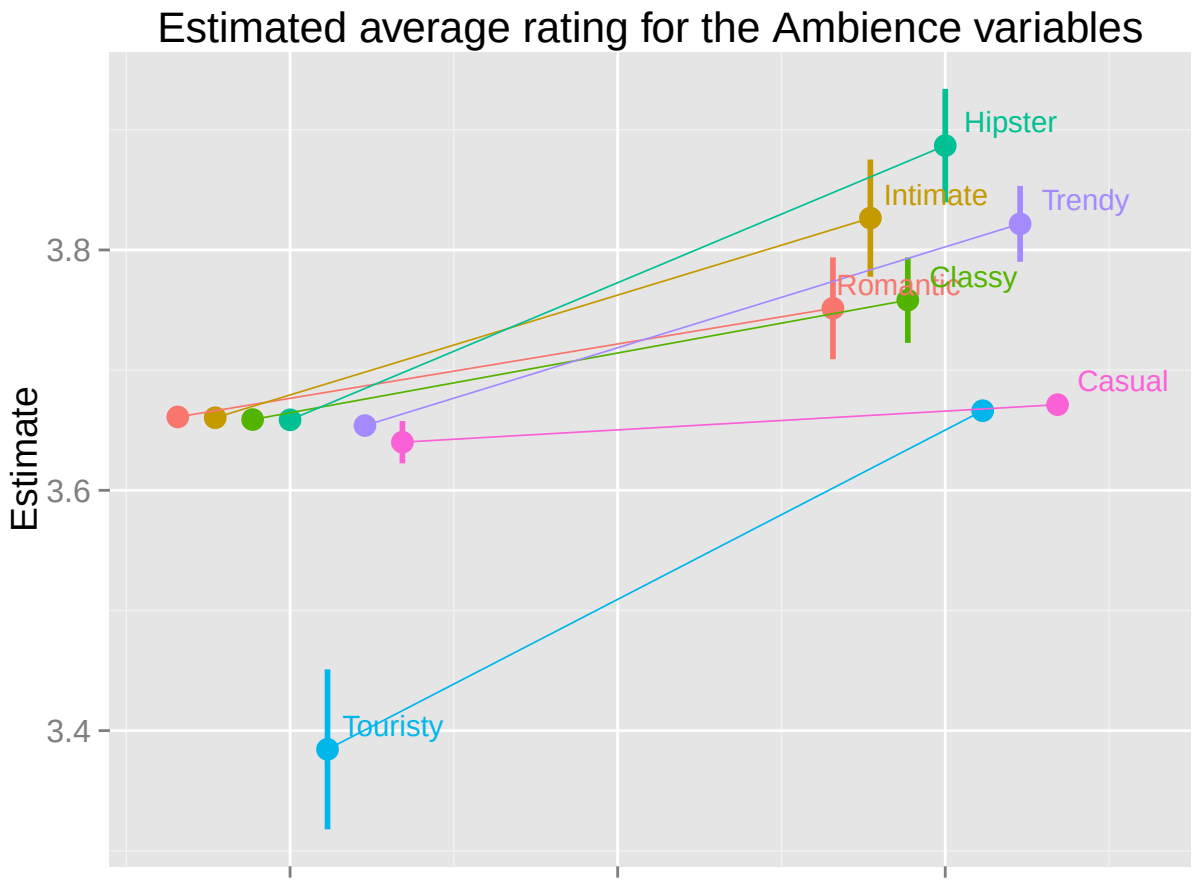
```
par(mfrow = c(1,1))

get_effect_data <- function(effname, Efflist){
  myEff <- Efflist[[effname]]
  xbar  <- myEff[["fit"]]
  xse   <- myEff[["se"]]
  lvls  <- myEff[["variables"]][[effname]][["levels"]]
  ind   <- order(xbar)
  xbar  <- xbar[ind]
  xse   <- xse[ind]
  lvls  <- lvls[ind]
  lvnm  <- (seq_along(lvls) - 1) / (length(lvls) - 1)

  return(data.frame(Effect   = rep(effname, length(xse)),
                    Level    = as.character(lvls),
                    LevelNum = lvnm,
                    Estimate = xbar,
                    StErr    = xse))
}

effnames <- c("Noise.Level", "Price.Range", "Alcohol", "Wi.Fi", "Attire")
ambnames <- names(Effs)[grep("Ambience", names(Effs))]

mainEffs <- lapply(X = effnames, FUN = get_effect_data, Efflist = Effs)
AmbEffs  <- lapply(X = ambnames, FUN = get_effect_data, Efflist = Effs)

mainEffs <- do.call(what = rbind, args = mainEffs)
AmbEffs  <- do.call(what = rbind, args = AmbEffs)

rownames(mainEffs) <- seq_along(mainEffs[,1])
rownames(AmbEffs)  <- seq_along(AmbEffs[,1])

mainEffs$Level <-
    c("Noise: very loud", "Noise: loud", "Noise: average", "Noise: quiet",
      "Price: 2", "Price: 1", "Price: 3", "Price: 4",
      "Full bar", "Beer and Wine", "No Alcohol",
      "Wi-fi: paid", "Wi-fi: none", "Wi-fi: free",
      "Casual Attire", "Dressy")


AmbEffs$Level <- c(" ", "Romantic",
                   " ", "Intimate",
                   " ", "Classy",
                   " ", "Hipster",
                   "Touristy", " ",
                   " ", "Trendy",
                   " ", "Casual")

library(ggplot2)
p1 <- ggplot(data = mainEffs,
```

```
            mapping = aes(group = Effect, x = LevelNum, y = Estimate, color = Effect,
                          ymax  = Estimate + StErr, ymin  = Estimate - StErr))

p1 +
    geom_pointrange(size = 1, position = position_dodge(width  = 0.4)) +
    geom_line(size = .3, linetype = 1, position = position_dodge(width  = 0.4)) +
    geom_text(aes(x = LevelNum + 0.05, y = Estimate + 0.02, label = Level),
            position = position_dodge(width  = 0.4), size = 4) +
    ggtitle("Estimated average rating for influential variables") +
    theme(legend.position = "none",
        axis.title.x = element_blank(), axis.text.x  = element_blank(),
        axis.title.y = element_text(size = 14), axis.text.y  = element_text(size = 12),
        plot.title   = element_text(size = 16)) +
    scale_x_continuous(limits = c(-0.2, 1.3))
```




```
p3 <- ggplot(data = AmbEffs,
            mapping = aes(group = Effect, x = LevelNum, y = Estimate, color = Effect,
                          ymax  = Estimate + StErr, ymin  = Estimate - StErr))

p3 +
    geom_pointrange(size = 1, position = position_dodge(width  = 0.4)) +
    geom_line(size = .3, linetype = 1, position = position_dodge(width  = 0.4)) +
    geom_text(aes(x = LevelNum + 0.1, y = Estimate + 0.02, label = Level),
            position = position_dodge(width  = 0.4), size = 4) +
  ggtitle("Estimated average rating for the Ambience variables") +
```

```
theme(legend.position = "none",
      axis.title.x = element_blank(), axis.text.x  = element_blank(),
      axis.title.y = element_text(size = 14), axis.text.y  = element_text(size = 12),
      plot.title   = element_text(size = 16)) +
  scale_x_continuous(limits = c(-0.2, 1.3))
```



Discussion

The results obtained in the previous section allow the inference of some interesting effects, that may help answering my question of interest (*what variables present the strongest effects on the average ratings of restaurants*):

1. The rating tends to go up as the price increases, which is relatively intuitive. However, *Price Range = 2* had a lower average rating than *Price Range = 1*, suggesting that very cheap places may benefit from some sort of "low expectations effect";

2. The level of noise is also a strong predictor of the ratings, with *quiet* places gaining on average 0.3 stars more than *very loud* ones. The difference between *Noise: average* and *Noise: quiet* is, however, quite low;

3. People seems to get quite angry at places that charge for Wi-fi, much more than at places that do not provide it. The message to restaurants seems to be clear: if you provide Wi-fi, do it for free. Otherwise, don't even bother, as *paid Wi-fi* tends to drop mean ratings by an average 0.3 stars compared to no *No Wi-fi*;

4. Places with *full bar* tend to be ranked lower than those with *Beer and Wine* bars and those without Alcohol (maybe due to lots of drunk people bothering other customers?);

5. Hipster, Trendy and Intimate Ambiences tend to generate palpable increases in the average rating of restaurants. It is possible that these variables present some high level of collinearity (experience suggests that trendy and hipster are possibly very correlated, but I did not investigate it in this report.);

6. Touristy restaurants tend to be ranked much lower than non-touristy. Again, it is very possible that *touristy* correlates negatively with *hipster* or *intimate*, but this effect was not investigated here;