Broadway data analysis

Julianna Szabo

12/23/2020

Executive summary

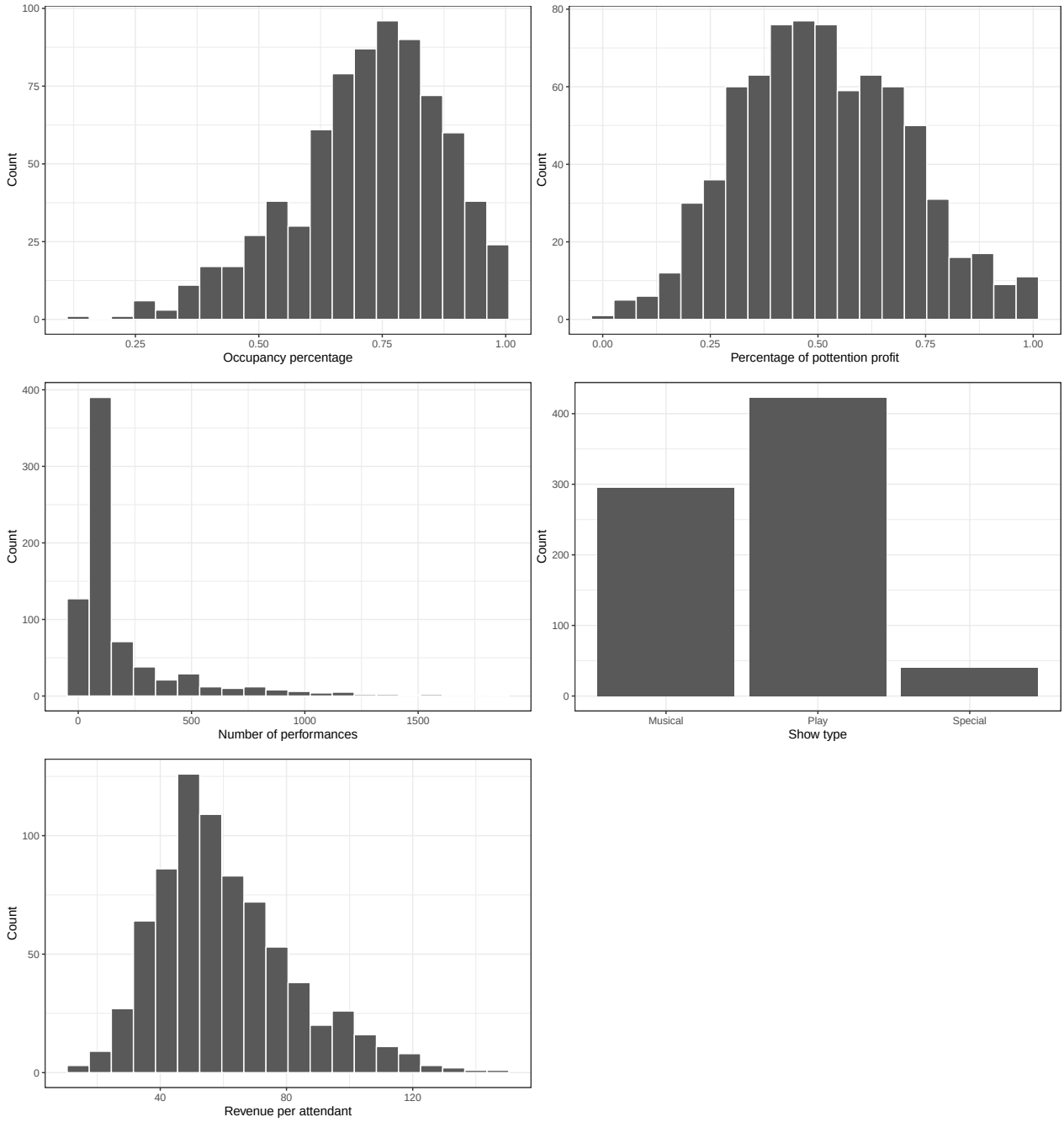
Introduction

This report aims to examine the main factors influencing the revenue per attendant of a Broadway show. Looking at the data, it is predicted that the main influencing variable would be the occupancy percentage, but have also found some other interesting variables that could affect the dependent variable. This project has great benefit for people involved in the theatre industry to see how different elements affect their revenue and possible in the end profit.

The main research question of this report will be:
What are the essential variables that affect the revenue per attendant of Broadway shows?

Data

The data comes from the CORGIS Dataset Project and was originally provided by the Broadway League. You can find a link to the original file in my GitHub folder. The data is a cross sectional time series covering all the shows that have been on Broadway from 1990 to 2016. For this project, the cross sectional aspect is the more important, and therefore, the data was aggregated based on show name. The time series aspect has been discarded but there have been a few adjustments made to control the effects it has on the data.
One of the main ones is using Revenue per attendant as the dependent variable. Since there are shows that ran for over 25 years and some for less than a year it would be unfair to compare the total revenue per show during its runtime. However, since the data includes the total revenue and the total number of attendants over the whole runtime of the show, using the ratio of these two variable gives an easy comparable relevant variable.
Overall the quality of the data is good. It included over 800 observations originally after the aggregation to cross sectional data, which, after cleaning, resulted in 758 complete observations. This shows a very representative sample, that has the potential of generalisation to other cities such as London with a similar theatre scene. There have been a few discrepancies in the data especially with the variables representing percentages. For both Occupancy percentage and Percentage of potential revenue, there were observations with values over 100%, which have been dropped.
Looking at the data more in detail, one can see that there are four explanatory variables to consider for one dependent variable.



variable	type	n	mean	median	min	max	sd
Occupancy percentage	x	758	0.72	0.74	0.15	1.00	0.15
Percentage of possible profit	x	758	0.51	0.50	0.01	1.00	0.19
Number of performances	x	758	279.96	99.00	0.00	8400.00	721.28
Revenue per Attendant	y	758	60.48	56.10	12.64	145.64	21.80

As shown in the graphs above, there are four quantitative ordered variables (including the dependent variable) and one qualitative nominal variable. All quantitative variables, with the exception of the Number of shows, are distributed somewhat normally with a left or right tail. The summary table of the variables also show the distribution of the values. Further, none of the variables are highly correlated (see Appendix 1), so the

analysis can be conducted without eliminating any variable.
After seeing the distribution of the variables, the decision was made to do transformations on some of the quantitative variables. After examining grams with possible log transformations (see Appendix 2), log transformations were applied to the Number of performances and Revenue per attendant. This was decided based on the distribution of the observations, but also due to the interpretation being more cohesive across variables. One additional transformation that was later added was to instead of a log transformation of Number of performances a dummy variable was created where 0 denotes the shows with less than one year (416) of performances while 1 denotes the ones with more than that.

Model

To create a more robust mode, the dataset has been split into train and test sets. The model exploration was done on test dataset, and then model picked was rerun and tested on the test set. I will also be working with a 95% confidence interval.
After examining the different options for models (see Appendix 3 and Model comparison file in Out folder), the best fitting model is the linear model using all four explanatory variables. The formula of this model is shown here:

$$\begin{aligned}\ln(\text{Revenue/attendant}) = \beta_0 + \beta_1 &\text{Occupancy percentage} + \beta_2 \text{ Percentage of possible profit} \\ &+\beta_3 \text{ Number of performances} + \beta_4 \text{Plays (Show type)} \\ &+\beta_5 \text{ Specials (Show type)}\end{aligned}$$

	Estimate	Std. Error	t value	Pr(>|t|)	CI Lower	CI Upper	DF
(Intercept)	3.44	0.06	57.88	0.00	3.33	3.56	600
capacity_filled	0.23	0.12	1.90	0.06	-0.01	0.46	600
percentage_of_poss_profit	0.97	0.10	9.76	0.00	0.77	1.16	600
as.factor(num_of_performances_d)1	0.02	0.03	0.78	0.44	-0.04	0.09	600
as.factor(show_type)Play	-0.11	0.02	-4.48	0.00	-0.16	-0.06	600
as.factor(show_type)Special	-0.08	0.08	-1.10	0.27	-0.23	0.07	600

Interpretation of coefficients:

Beta 0: When all explanatory variables are 0, the Revenue per attendant would be ln(3.44) - this is almost meaningless.
Beta 1: the Revenue per attendant increases by approximately 23% on average for every additional percentage of occupancy of the theatre, when all other variables are the same.
Beta 2: the Revenue per attendant increases by approximately 97% on average for every additional percentage of the possible revenue achieved when all other variables are the same.
Beta 3: For shows that run longer than one year, the Revenue per attendant increased by approximated 2% on average, when all other variables are the same.
Beta 4: If a show is a Play instead of a Musical, on average the Revenue per attendant is 11% lower on average, when all other variables are the same.
Beta 5: If a show is a Special instead of a Musical, on average the Revenue per attendant is 8% lower average, when all other variables are the same.

Even with this model, that fits the data the best, it still only explains 42% of the observations. Further, about half of the betas have a very low p value and can therefore we considered very good approximations for this dataset, while three values (Occupancy percentage, Number of performances, and Special show type) have a high p value and therefore the real slope value will fall outside of the 95% confidence interval. Looking at the previous models, it is clear that the real slope of the Occupancy percentage is closer to 1 (or 100%),

while the Number of performances is most likely closer to 0.1 (or 10%).

	Train - Estimates	Test - Estimates
Intercept	3.44	3.41
Occupancy Percentage	0.23	0.23
Percentage of pott. profit	0.97	1.00
More than one year of performances	0.02	0.02
Show type is Play	-0.11	-0.10
Show type is Special	-0.08	-0.08

The model shown is very robust based on the train and test robustness check run. As can be seen in the table above the model gives almost the same coefficients when it is rerun of the test sample. While the R squared here is only 38% but it is very close to the 41% of the original train model.

Appendix

Correlation

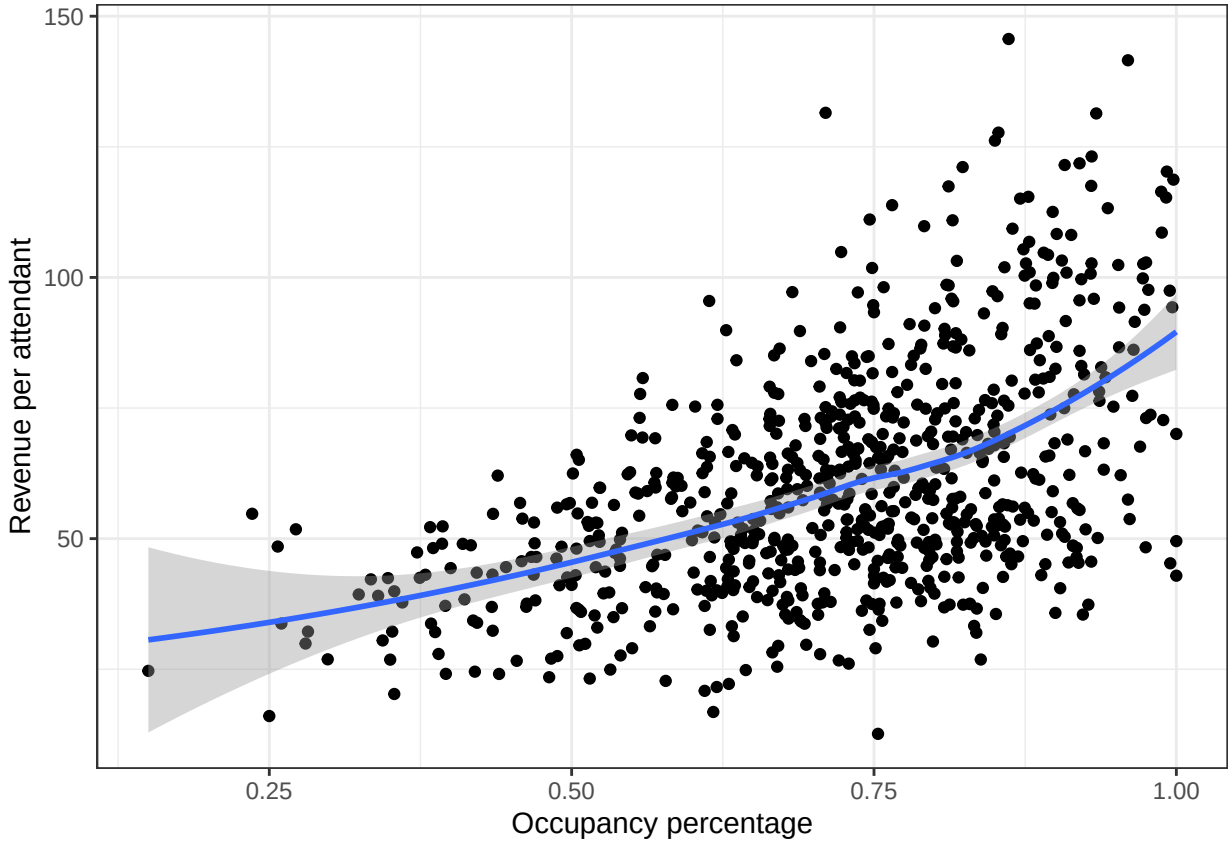
Var1	Var2	corr_val
ln_capacity_filled	capacity_filled	0.98
capacity_filled_sq	capacity_filled	0.99
ln_percentage_of_poss_profit	percentage_of_poss_profit	0.93
ln_revenue_per_att	revenue_per_att	0.97
capacity_filled	ln_capacity_filled	0.98
capacity_filled_sq	ln_capacity_filled	0.94
revenue_per_att	ln_revenue_per_att	0.97
percentage_of_poss_profit	ln_percentage_of_poss_profit	0.93
capacity_filled	capacity_filled_sq	0.99
ln_capacity_filled	capacity_filled_sq	0.94

Ln transformation

Occupancy percentage

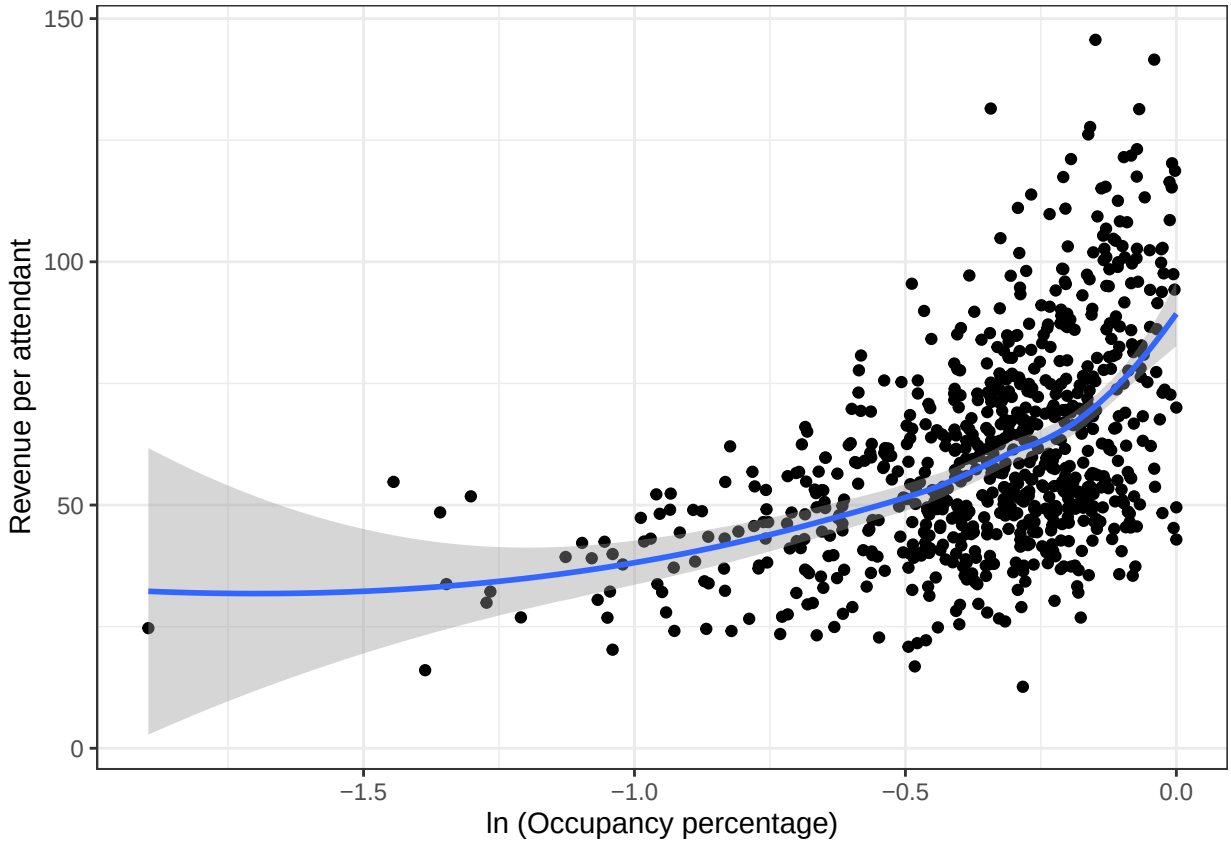
Level - level regression

```
## `geom_smooth()` using formula 'y ~ x'
```



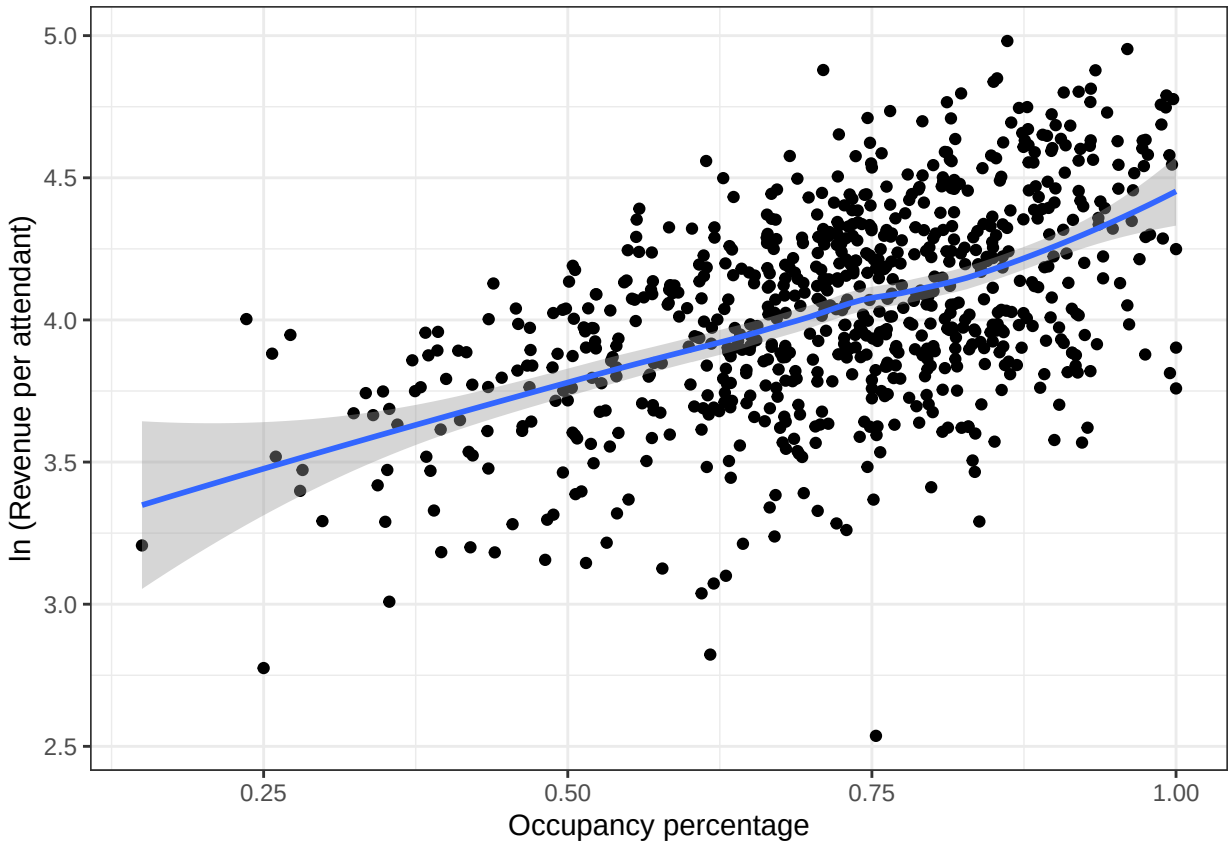
Log - level regression

```
## `geom_smooth()` using formula 'y ~ x'
```



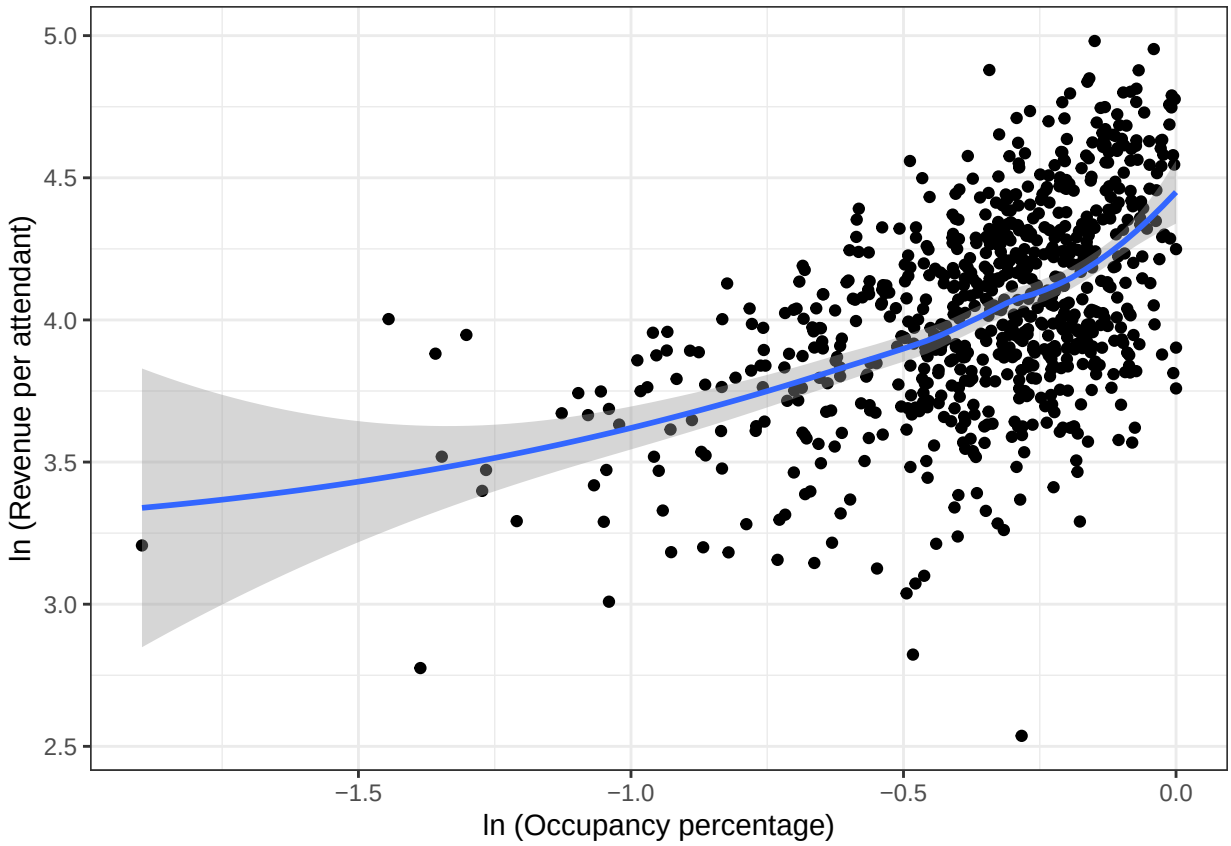
Level - log regression

```
## `geom_smooth()` using formula 'y ~ x'
```



Log - log regression

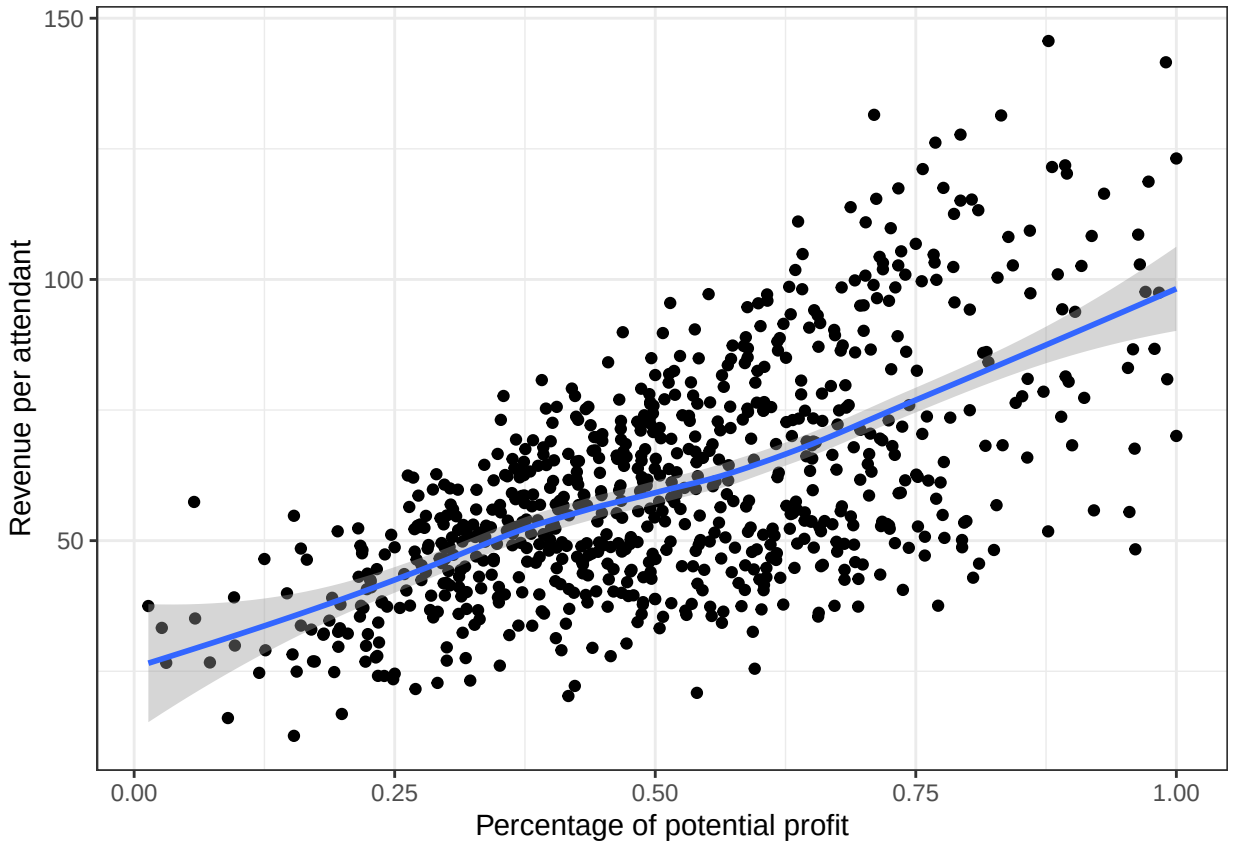
```
## `geom_smooth()` using formula 'y ~ x'
```



Percentage of potential profit

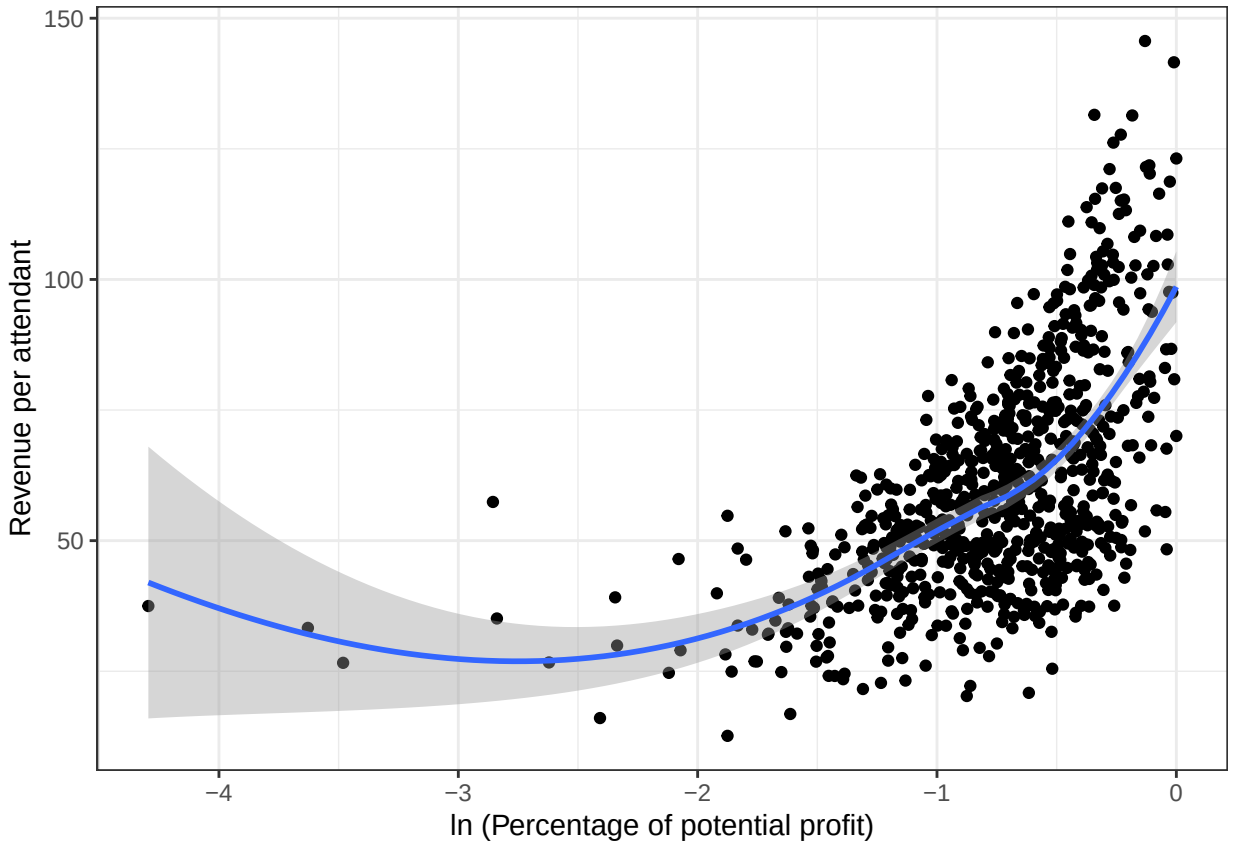
Level - level regression

```
## `geom_smooth()` using formula 'y ~ x'
```



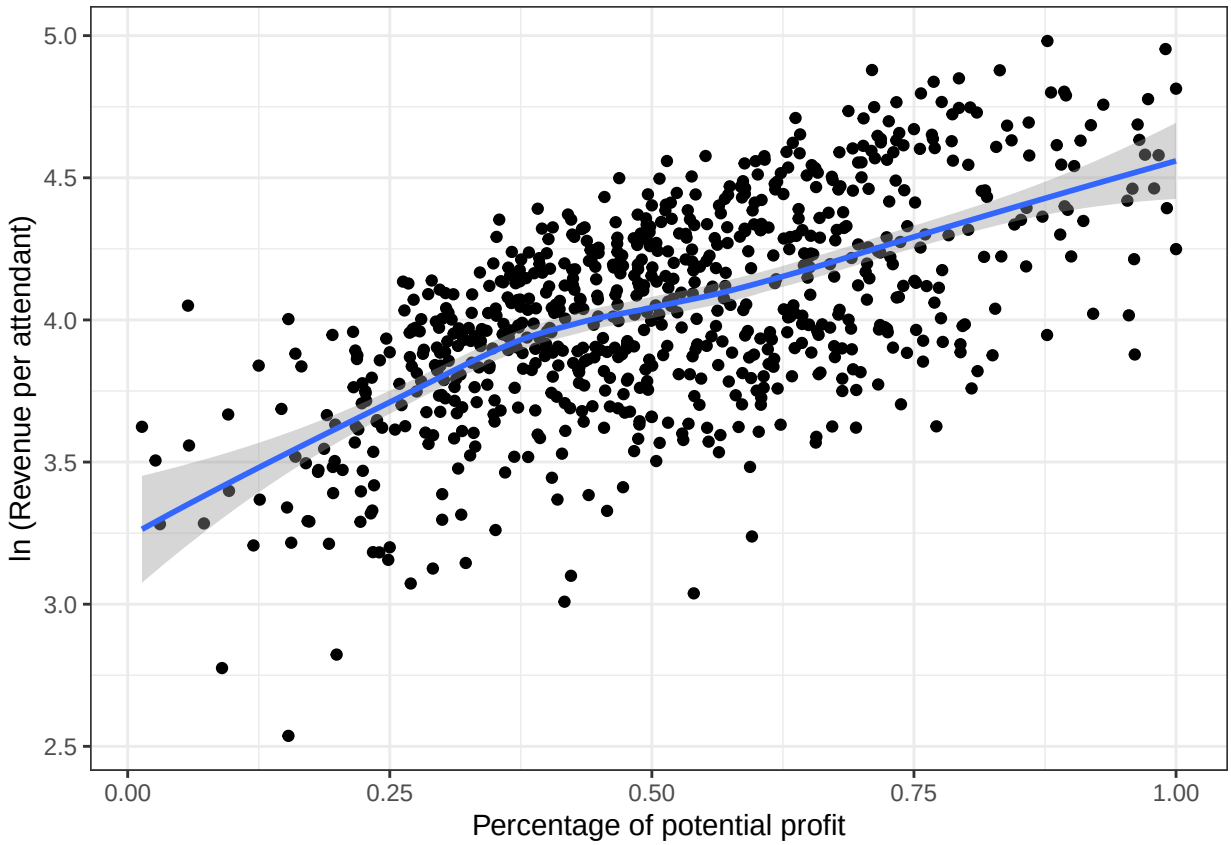
Log - level regression

```
## `geom_smooth()` using formula 'y ~ x'
```



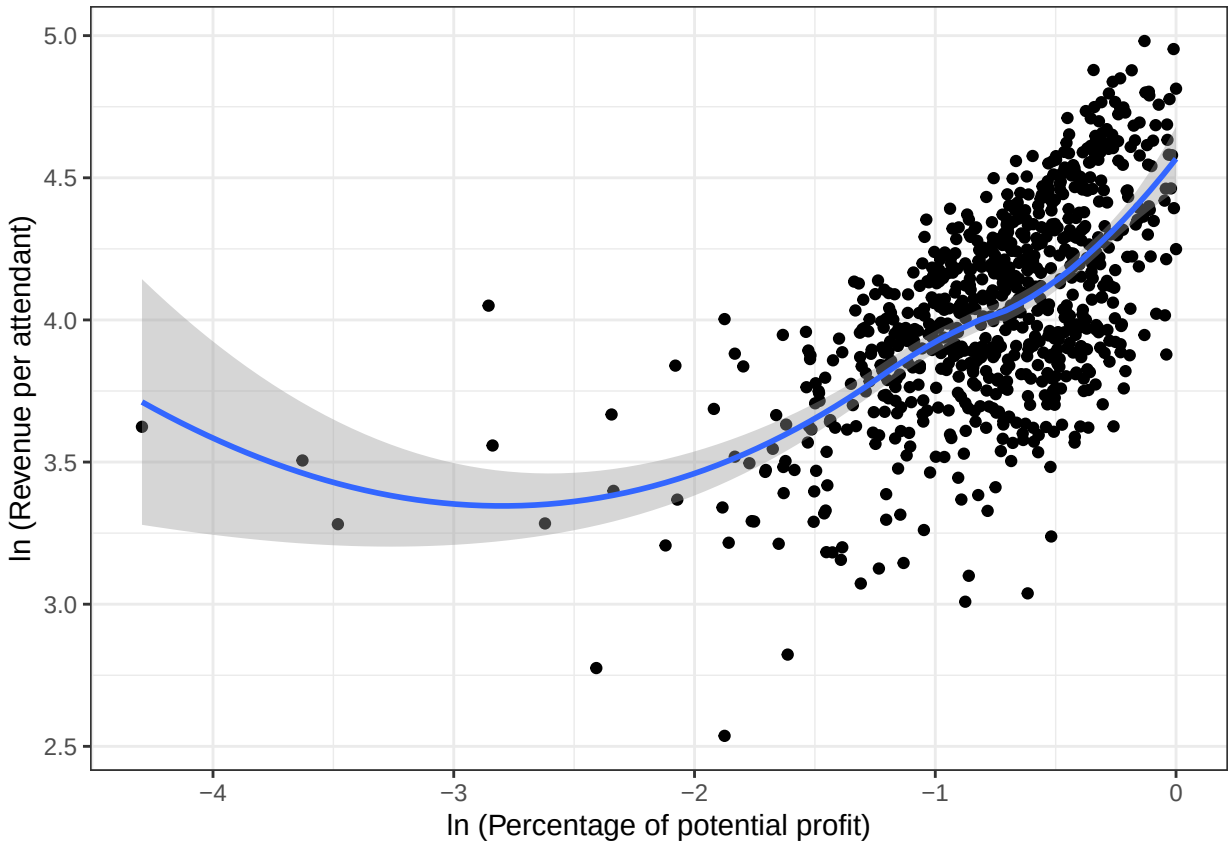
Level - log regression

```
## `geom_smooth()` using formula 'y ~ x'
```



Log - log regression

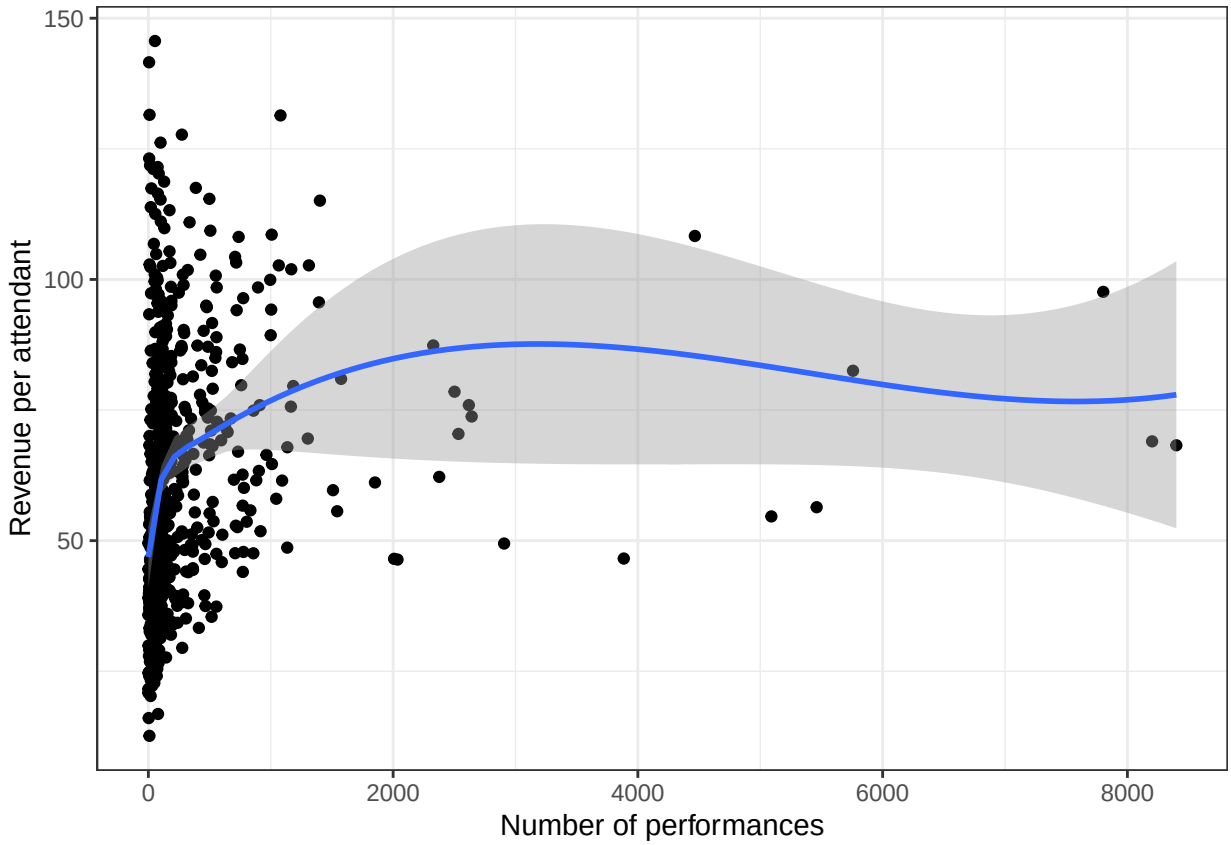
```
## `geom_smooth()` using formula 'y ~ x'
```



Number of performances

Level - level regression

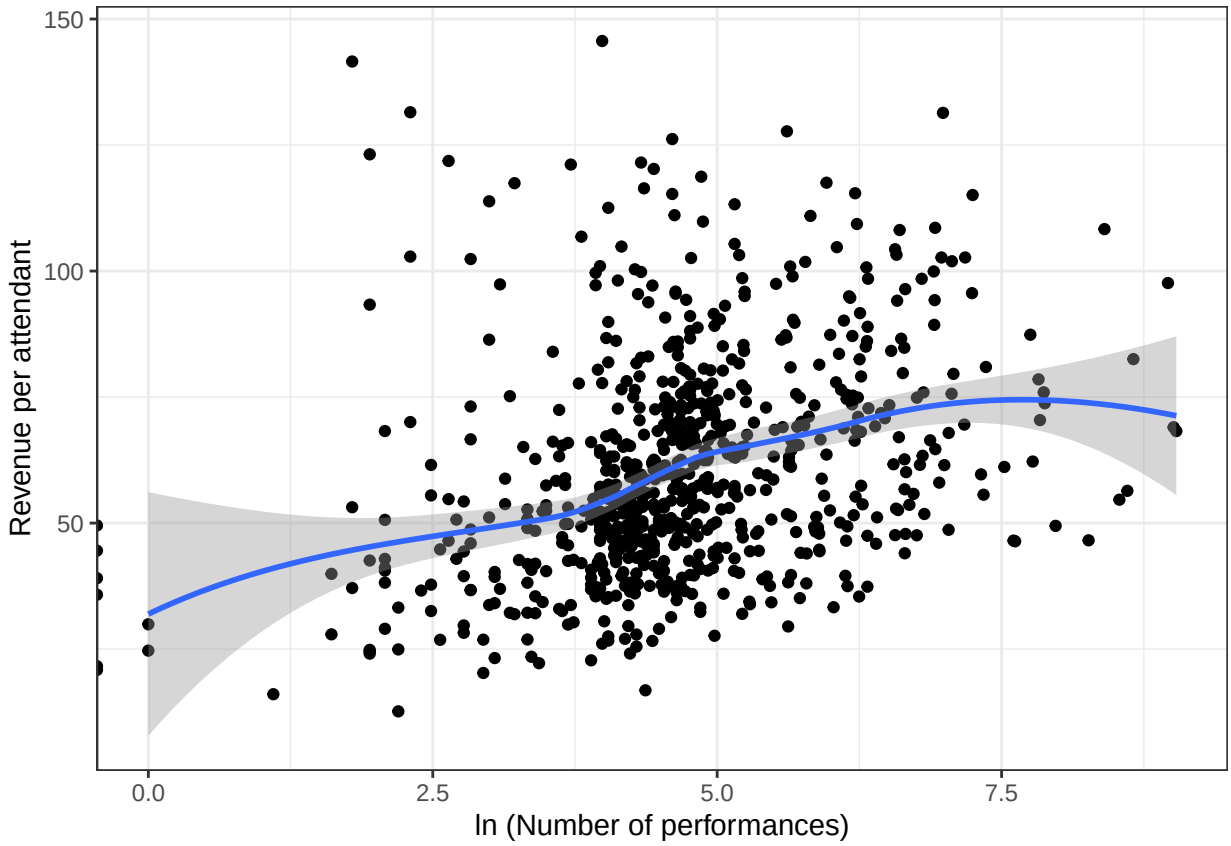
```
## `geom_smooth()` using formula 'y ~ x'
```



Log - level regression

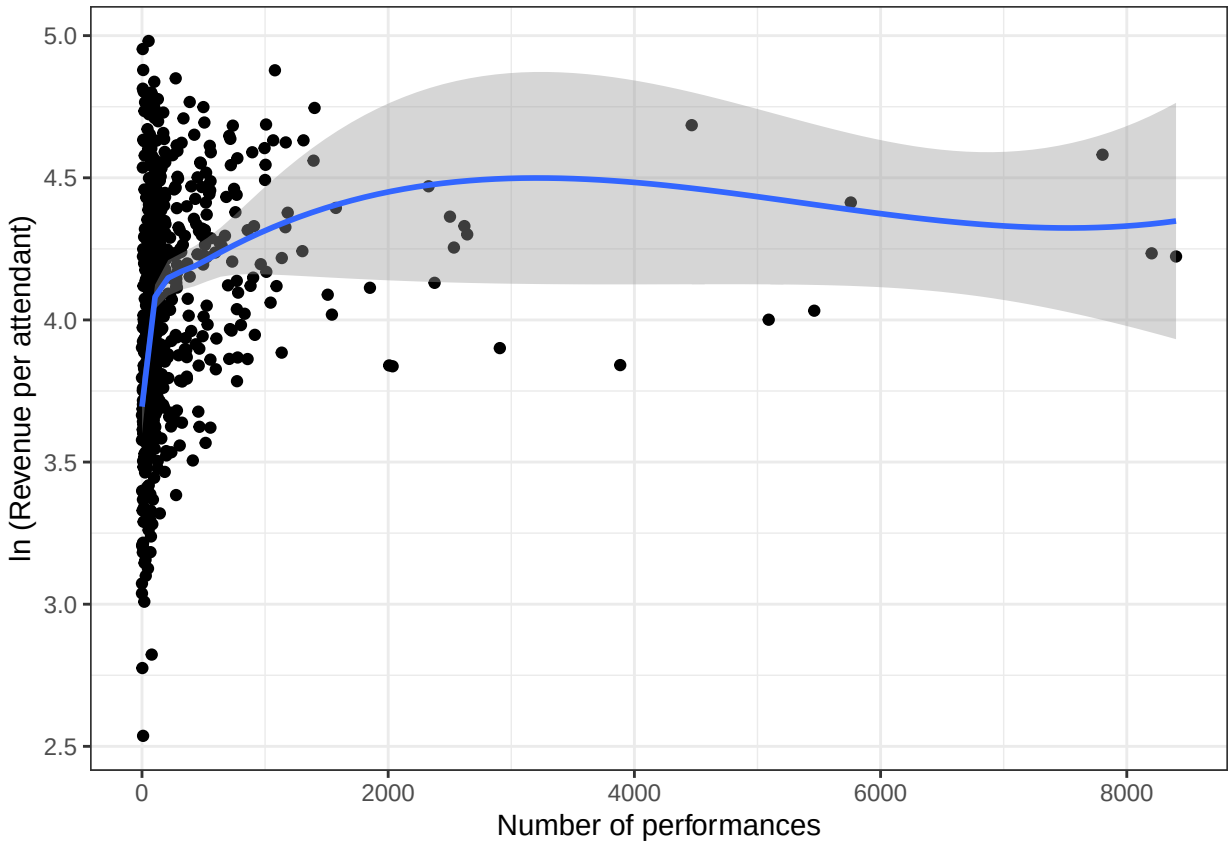
```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 6 rows containing non-finite values (stat_smooth).
```



Level - log regression

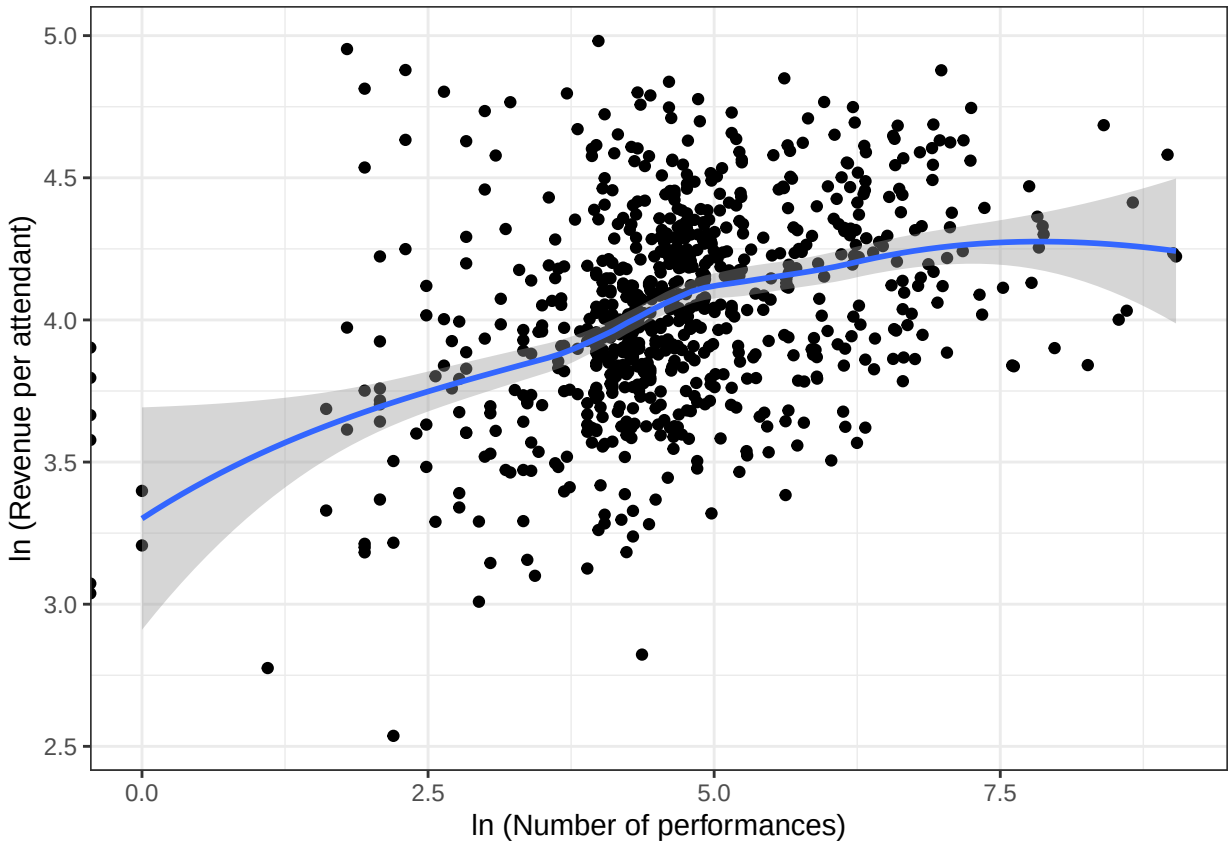
```
## `geom_smooth()` using formula 'y ~ x'
```



Log - log regression

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 6 rows containing non-finite values (stat_smooth).
```

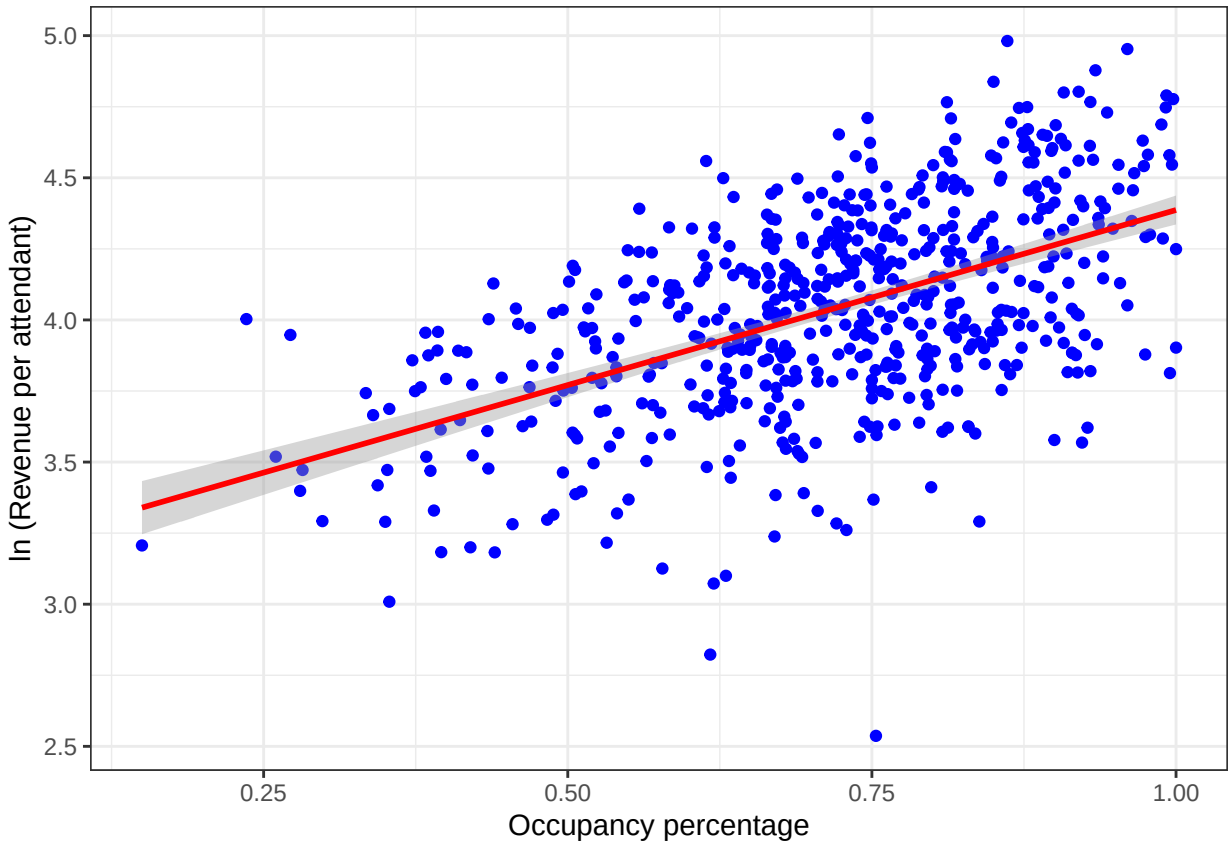


Regression modes

To make my regression model more robust, I created a train and test data set

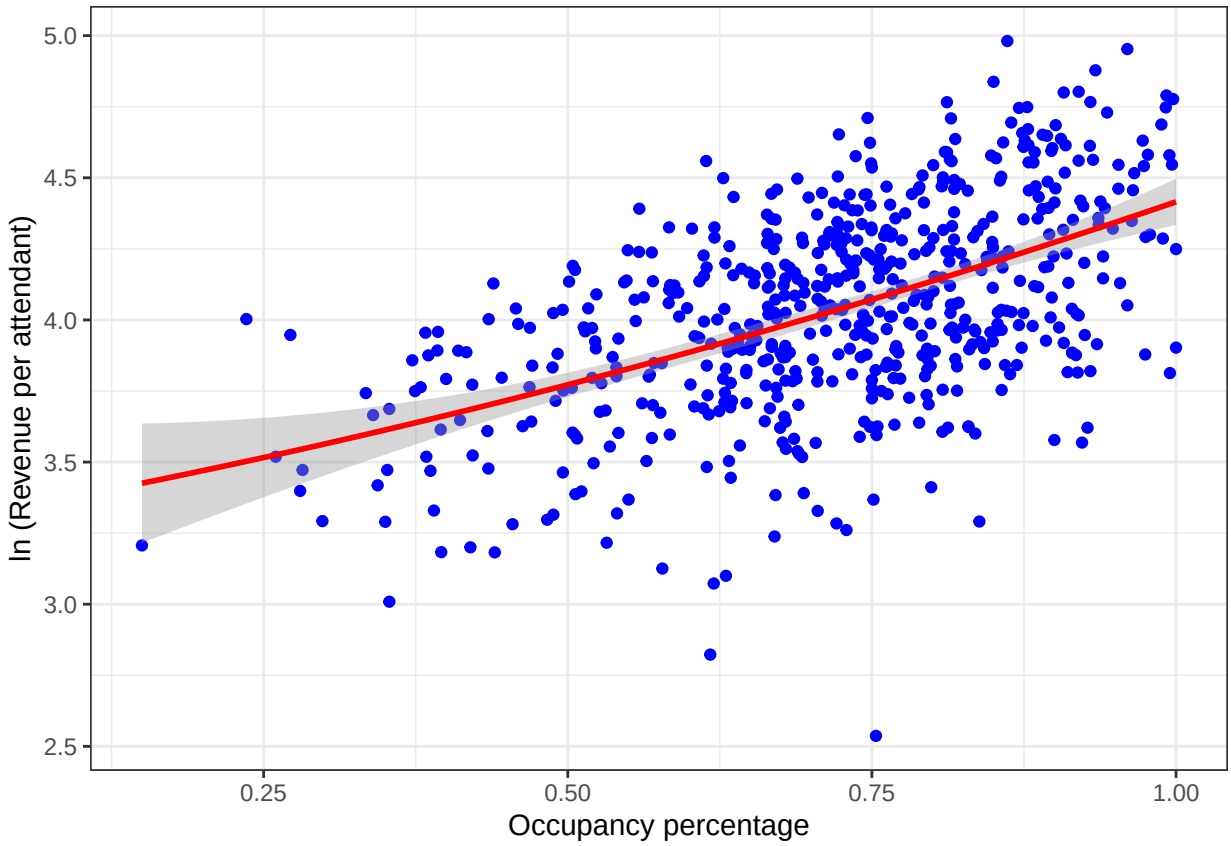
Regression 1 - Simple linear regression

```
##
## Call:
## lm_robust(formula = ln_revenue_per_att ~ capacity_filled, data = train,
##     se_type = "HC2")
##
## Standard error type:  HC2
##
## Coefficients:
##                 Estimate Std. Error t value   Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)        3.155    0.05645   55.89 7.315e-241    3.044    3.266 604
## capacity_filled    1.232    0.07822   15.75  4.626e-47    1.078    1.385 604
##
## Multiple R-squared:  0.278 , Adjusted R-squared:  0.2768
## F-statistic:   248 on 1 and 604 DF,  p-value: < 2.2e-16

## `geom_smooth()` using formula 'y ~ x'
```



Regression 2 - Quadratic (linear) regression

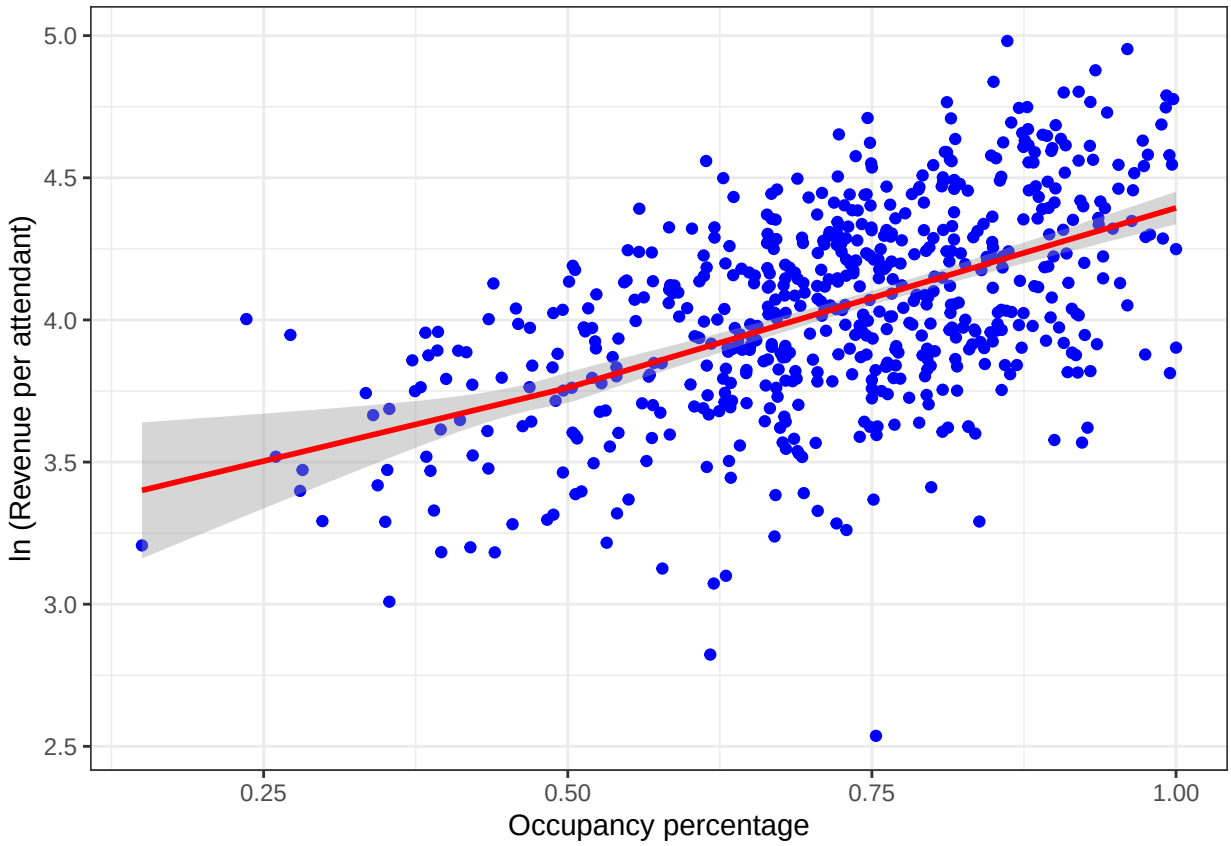
```
##
## Call:
## lm_robust(formula = ln_revenue_per_att ~ capacity_filled + capacity_filled_sq,
##     data = train)
##
## Standard error type:  HC2
##
## Coefficients:
##                    Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)          3.3033     0.1670 19.7798 1.719e-67   2.9753    3.631 603
## capacity_filled      0.7625     0.5132  1.4858 1.379e-01  -0.2454    1.770 603
## capacity_filled_sq   0.3497     0.3837  0.9113 3.625e-01  -0.4038    1.103 603
##
## Multiple R-squared:  0.2789 ,    Adjusted R-squared:  0.2765
## F-statistic: 123.9 on 2 and 603 DF,  p-value: < 2.2e-16
```



Regressipn 3 - Piecewise linear spline regression

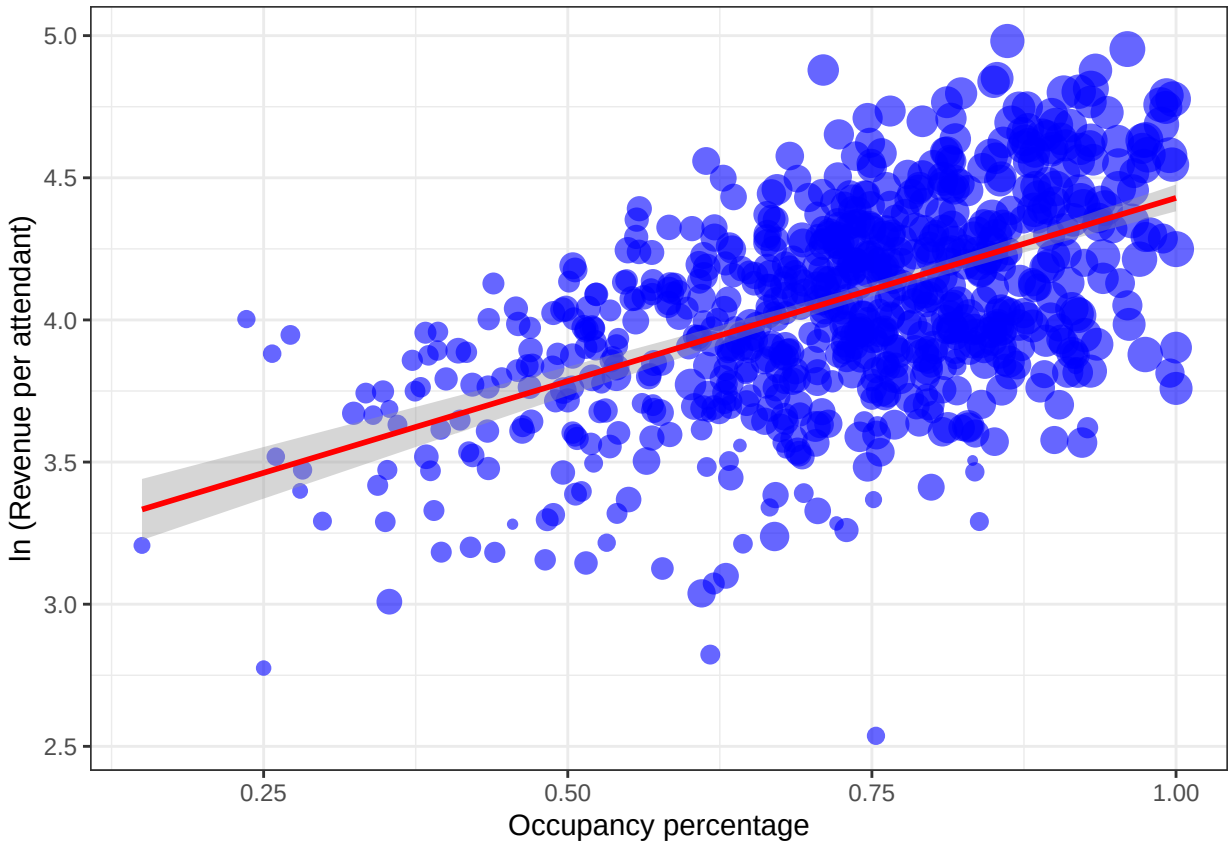
Using 0.5 as a cutof point

```
##
## Call:
## lm_robust(formula = ln_revenue_per_att ~ lspline(capacity_filled,
##     cutoff), data = train)
##
## Standard error type:  HC2
##
## Coefficients:
##                                  Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                         3.246     0.1660  19.548 2.794e-66
## lspline(capacity_filled, cutoff)1   1.032     0.3562   2.896 3.913e-03
## lspline(capacity_filled, cutoff)2   1.265     0.1010  12.522 3.872e-32
##                                  CI Lower CI Upper  DF
## (Intercept)                        2.9195    3.572 603
## lspline(capacity_filled, cutoff)1  0.3321    1.731 603
## lspline(capacity_filled, cutoff)2  1.0667    1.463 603
##
## Multiple R-squared:  0.2783 ,    Adjusted R-squared:  0.2759
## F-statistic: 123.4 on 2 and 603 DF,  p-value: < 2.2e-16
```



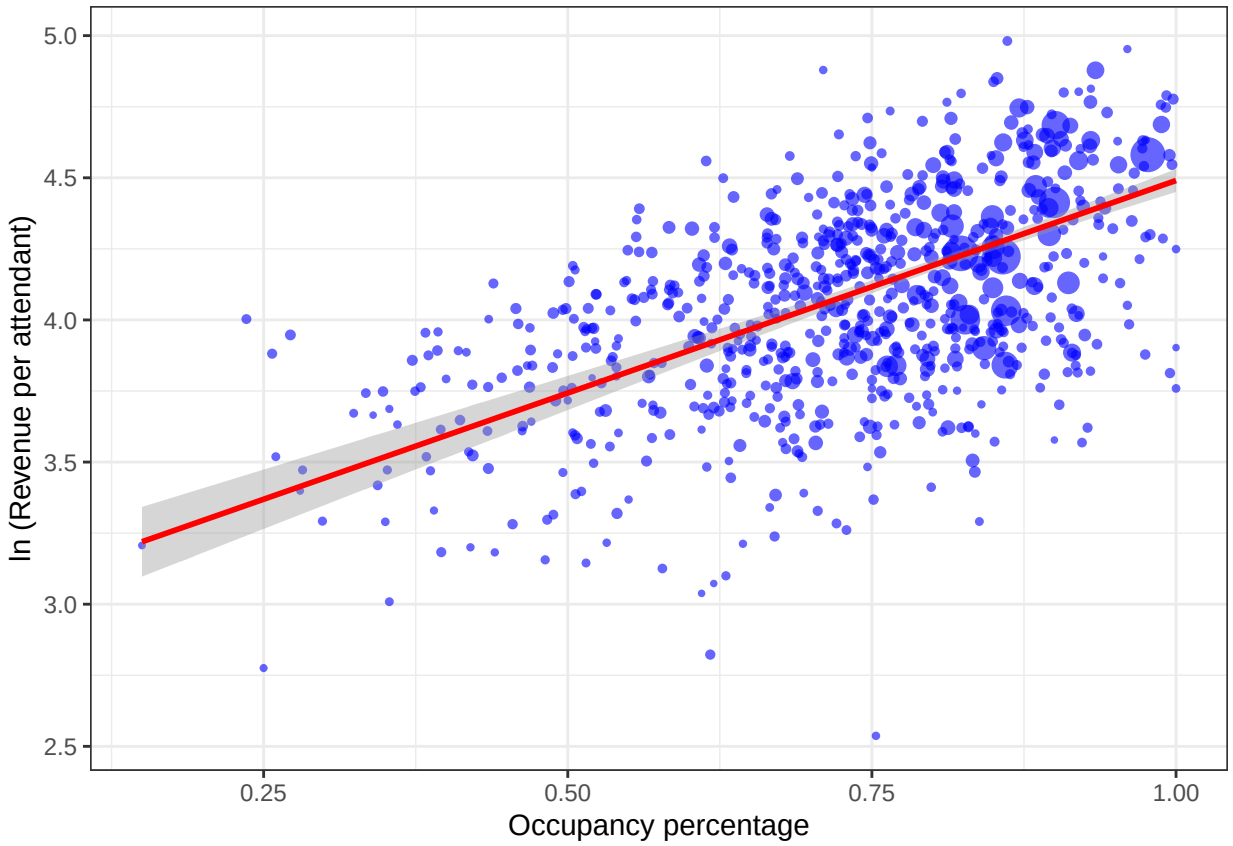
Regression 4 - Weighted linear regression, where weights = percentage of total revenue

```
##
## Call:
## lm_robust(formula = ln_revenue_per_att ~ capacity_filled, data = train,
##     weights = percentage_of_poss_profit)
##
## Weighted, Standard error type:  HC2
##
## Coefficients:
##                 Estimate Std. Error t value   Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)        3.140    0.06324   49.65 2.222e-215    3.016    3.264 604
## capacity_filled    1.289    0.08717   14.79  1.787e-42    1.118    1.461 604
##
## Multiple R-squared:  0.2669 ,    Adjusted R-squared:  0.2657
## F-statistic: 218.8 on 1 and 604 DF,  p-value: < 2.2e-16

## `geom_smooth()` using formula 'y ~ x'
```



Regression 5 - Weighted linear regression, where weights = number of performances

```
##
## Call:
## lm_robust(formula = ln_revenue_per_att ~ capacity_filled, data = train,
##     weights = num_of_performances)
##
## Weighted, Standard error type:  HC2
##
## Coefficients:
##                 Estimate Std. Error t value   Pr(>|t|) CI Lower CI Upper  DF
## (Intercept)        2.995     0.1097  27.317 1.353e-107     2.78    3.211 604
## capacity_filled    1.495     0.1504   9.938  1.179e-21     1.20    1.790 604
##
## Multiple R-squared:  0.2995 ,    Adjusted R-squared:  0.2984
## F-statistic: 98.76 on 1 and 604 DF,  p-value: < 2.2e-16

## `geom_smooth()` using formula 'y ~ x'
```



Model Comparison

```
## The table was written to the file '/Users/Terez/OneDrive - Central European University/Data_Analysis_
```

Looks like these models with mainly one variable are not a great fit for the data. Therefore, I will include additional variables to try and get a better fit. Further, it looks like the original use of "Number of Performances" has no impact so I will try and create a dummy variable and use that instead. I will use 0 for any show that had less than one year of performances so less than 8*52 (416) and one for those that have had more.

Additional models

Check if it becomes better if one of the weights are included as variables

```
##
## Call:
## lm_robust(formula = ln_revenue_per_att ~ capacity_filled + percentage_of_poss_profit,
##     data = train, se_type = "HC2")
##
## Standard error type:  HC2
##
## Coefficients:
##                          Estimate Std. Error t value   Pr(>|t|) CI Lower
## (Intercept)                3.3269    0.05699  58.374 2.790e-250  3.21501
```

```
## capacity_filled             0.2981    0.12044   2.476  1.358e-02  0.06162
## percentage_of_poss_profit   0.9705    0.10052   9.654  1.329e-20  0.77304
##                           CI Upper  DF
## (Intercept)                 3.4389 603
## capacity_filled             0.5347 603
## percentage_of_poss_profit   1.1679 603
##
## Multiple R-squared:  0.3893 ,    Adjusted R-squared:  0.3873
## F-statistic: 180.9 on 2 and 603 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm_robust(formula = ln_revenue_per_att ~ capacity_filled + as.factor(num_of_performances_d),
##     data = train, se_type = "HC2")
##
## Standard error type:  HC2
##
## Coefficients:
##                                  Estimate Std. Error t value   Pr(>|t|)
## (Intercept)                        3.1937    0.05735   55.69 7.288e-240
## capacity_filled                    1.1505    0.08179   14.07  4.530e-39
## as.factor(num_of_performances_d)1  0.1226    0.03217    3.81  1.530e-04
##                                  CI Lower CI Upper  DF
## (Intercept)                       3.08108   3.3064 603
## capacity_filled                   0.98987   1.3111 603
## as.factor(num_of_performances_d)1 0.05939   0.1857 603
##
## Multiple R-squared:  0.2924 ,    Adjusted R-squared:   0.29
## F-statistic:   141 on 2 and 603 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm_robust(formula = ln_revenue_per_att ~ capacity_filled + percentage_of_poss_profit +
##     as.factor(num_of_performances_d), data = train, se_type = "HC2")
##
## Standard error type:  HC2
##
## Coefficients:
##                                  Estimate Std. Error t value   Pr(>|t|)
## (Intercept)                       3.34931    0.05768  58.063 7.273e-249
## capacity_filled                   0.26727    0.12078   2.213  2.728e-02
## percentage_of_poss_profit         0.94336    0.10028   9.408  1.054e-19
## as.factor(num_of_performances_d)1 0.08596    0.02906   2.958  3.219e-03
##                                  CI Lower CI Upper  DF
## (Intercept)                       3.23602   3.4626 602
## capacity_filled                   0.03007   0.5045 602
## percentage_of_poss_profit         0.74643   1.1403 602
## as.factor(num_of_performances_d)1 0.02889   0.1430 602
##
## Multiple R-squared:  0.3964 ,    Adjusted R-squared:  0.3934
## F-statistic: 130.6 on 3 and 602 DF,  p-value: < 2.2e-16
```

```
##
```

```
## Call:
## lm_robust(formula = ln_revenue_per_att ~ capacity_filled + percentage_of_poss_profit +
##     as.factor(num_of_performances_d) + as.factor(show_type),
##     data = train, se_type = "HC2")
##
## Standard error type:  HC2
##
## Coefficients:
##                                   Estimate Std. Error t value   Pr(>|t|)
## (Intercept)                        3.44289    0.05949 57.8765 1.059e-247
## capacity_filled                    0.22564    0.11848  1.9045  5.733e-02
## percentage_of_poss_profit          0.96774    0.09920  9.7550  5.753e-21
## as.factor(num_of_performances_d)1  0.02461    0.03160  0.7789  4.363e-01
## as.factor(show_type)Play          -0.11155    0.02489 -4.4813  8.888e-06
## as.factor(show_type)Special       -0.08299    0.07541 -1.1004  2.716e-01
##                                   CI Lower CI Upper  DF
## (Intercept)                        3.326059  3.55971 600
## capacity_filled                   -0.007046  0.45833 600
## percentage_of_poss_profit          0.772912  1.16257 600
## as.factor(num_of_performances_d)1 -0.037445  0.08667 600
## as.factor(show_type)Play          -0.160434 -0.06266 600
## as.factor(show_type)Special       -0.231094  0.06512 600
##
## Multiple R-squared:  0.4145 ,    Adjusted R-squared:  0.4096
## F-statistic: 85.85 on 5 and 600 DF,  p-value: < 2.2e-16

## The table was written to the file '/Users/Terez/OneDrive - Central European University/Data_Analysis
```

Check model on test data

```
reg_test <-  lm_robust( ln_revenue_per_att ~ capacity_filled + percentage_of_poss_profit + as.factor(nu
summary( reg_test , digit = 2)
```

```
##
## Call:
## lm_robust(formula = ln_revenue_per_att ~ capacity_filled + percentage_of_poss_profit +
##     as.factor(num_of_performances_d) + as.factor(show_type),
##     data = test, se_type = "HC2")
##
## Standard error type:  HC2
##
## Coefficients:
##                                   Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                        3.41297    0.13217 25.8223 2.696e-56
## capacity_filled                    0.23469    0.20515  1.1440 2.545e-01
## percentage_of_poss_profit          1.00004    0.16366  6.1105 8.592e-09
## as.factor(num_of_performances_d)1  0.02234    0.06911  0.3233 7.469e-01
## as.factor(show_type)Play          -0.09988    0.05126 -1.9487 5.325e-02
## as.factor(show_type)Special       -0.07637    0.15738 -0.4852 6.282e-01
##                                   CI Lower CI Upper  DF
## (Intercept)                         3.1517 3.674182 146
```

```
## capacity_filled                    -0.1708 0.640132 146
## percentage_of_poss_profit           0.6766 1.323486 146
## as.factor(num_of_performances_d)1  -0.1142 0.158925 146
## as.factor(show_type)Play           -0.2012 0.001417 146
## as.factor(show_type)Special        -0.3874 0.234662 146
##
## Multiple R-squared:   0.38 , Adjusted R-squared:  0.3588
## F-statistic: 22.08 on 5 and 146 DF,  p-value: < 2.2e-16
```

```
data_out <- "/Users/Terez/OneDrive - Central European University/Data_Analysis_02/DA2_Assignment_2/out/
htmlreg( list(reg9, reg_test),
         type = 'html',
         custom.model.names = c("Train model", "Test model"),
         caption = "Modelling Revenue per attendant for different shows",
         file = paste0( data_out ,'model_comparison_train_test.html'), include.ci = FALSE)
```

```
## The table was written to the file '/Users/Terez/OneDrive - Central European University/Data_Analysis_
```