DBA5101
Managerial Economics
Group Project 1

Study of Train Demand Estimates

Widya Gani Salim - A0231857Y
Hpone Myat Khine - A0125002E
Mingxuan Yang - A0231854E
Sae Jin Jang - A0231989M
Kyle Kenji Asano - A0231984X

Table of Contents

1. Executive Summary

As one of the oldest, yet persistent transportation modes, trains remain as one of the go-to public transports adopted worldwide. With established railroads, trains connect large areas and are economical transportation modes. In this project, we will estimate the demand function for trains using train ticket sales data at a particular train station.

The problem is complex by nature as the demand function can never be estimated without complication due to simultaneity of the supply and demand functions. As such, we approached the problem of estimation by adopting the Two-Stage Least Squares (2SLS) regression model. The 2SLS model allows us to eliminate endogeneity bias which would be prevalent in the Ordinary Least Squares (OLS) model of a demand function.

Based on our analysis, the train's 2SLS demand function is:

$$ln(seats) = 1.8183 - 0.2431 * \widehat{ln(price)} + 0.0012 * \; days\_in\_advance \; + \; \epsilon$$

With the instrument variable to predict ln(price):

$$\widehat{ln(price)} = 5.8739 - \; 0.0029 * \; days\_in\_advance \; - 0.7367 * \; isNormCabin \; + \; \nu$$

The average ticket price was identified as an endogenous variable, estimated using an instrument variable, cabin type. Advanced purchase variable - which estimates the number of days tickets were purchased before departure - was identified as an exogenous variable. We log-transformed the demand and price variables to measure elasticity of demand. The ln(price) coefficient was -0.2431 which means that demand is inelastic. Meanwhile, advanced purchase has a slope of 0.0012 which means that the further away the departure date, the higher the number of seats purchased per transaction.

The intuition of both variables are logical. Train demand is inelastic as other transportation modes, such as cars, planes and ships might not be readily available in certain areas. Similarly, customers are more likely to purchase more tickets further away from the departure date; groups will plan their travel early - wanting to avoid the risk of last minute bookings.

2. Business Insights

There are various degrees of 'sensitivity' of goods demand to the change in price. Suppliers - such as train companies - have to price their goods accordingly. Elasticity represents this sensitivity and products can be deemed to be elastic, unitarily elastic or inelastic. Elastic demand is where there is more than proportionate change in quantity demanded for a corresponding change in price. Inelastic demand would be the inverse - that is a less than proportionate change in quantity demanded for a corresponding change in price.

The demand function for this dataset is judged to be relatively inelastic. Given the vast distances and easy accessibility - as train stations are typically located in key cities - that trains cover, there are few viable substitutes. When train prices rise, there is less than a proportionate drop in demand. Train companies will be able to raise prices without worrying about a significant drop in demand.

Train companies can also capitalise on passengers who urgently need to travel and charge higher prices the closer the purchase date is to departure date. We included such consideration in our analysis by creating a new variable to measure purchase urgency.

3. Data & Features

The datasets used were of train ticket sales data from June 2018 to June 2019. The datasets present 209,697 entries and 14 distinct features. Each entry represents a ticket purchase within the time period.

To maintain consistency throughout the report, we have created a table of all the variables used, and how they would address it moving forward (*See Appendix 1*).

3.1 Exploratory Data Analysis (EDA)

Through our EDA, we have found that the data contained a few notable outliers and inconsistencies, including:

- **Inconsistency of the Cumulative Sales variable**. This variable was supposed to always be increasing until the end of the period but had decreasing values on multiple dates.

- **Train 'O' only had one entry.** This could be caused by data loss or that the purchases for Train 'O' were not recorded properly.

- **Outliers in the ticket price variable.** There were 2 extreme outliers in the ticket variable which cost $7,855.77 and $1,701.56. This is a huge jump from the mean price which was $230.12 (*See Appendix 7*).

3.2 Data Transformation

Natural log was applied to both the ticket price and the number of seats variables, transforming them into ln(price) and ln(seats) respectively. This was done to transform them into variables that could measure demand elasticity.

By applying log, the coefficient is now the ratio of relative change of demand to price. Simply taking the coefficient without log would result in a ratio of absolute change which would not enable us to measure elasticity.

Furthermore, to be able to explore the data further, the train number and customer category variables were transformed into dummy variables with binary input (*See Appendix 7*).

3.3 Feature Engineering

A new variable, advanced purchase, was created by subtracting departure date with purchase date. This was done to explore whether urgency affects customers' demand, as discussed in our business insights.

We also created the weekend factor variable, which indicates whether the purchase was made on a weekend, to better study the effects of the weekend on train ticket sales

3.4 Feature Selection

Based on our EDA and preliminary models, we decided to exclude the following variables:

Cumulative Sales due to its inconsistencies discussed. It comprises of sub-groups which we had no information on, which means its associations with other variables is unknown.

Train Number, Return Trip and One-way Trip variables were not included as intuitively as they should not affect customers' demand for train tickets.

Customers would not base their decision to purchase tickets based on train types, whether it's a one way trip or if it's a return ticket; they will buy a ticket only based on their need to travel.

4. Model

4.1 Assumptions

A few assumptions for the model to hold:

- Supply and demand is in equilibrium.
- No effects of inflation, foreign exchange and cyclical seasonality on any of the variables used.
- Elasticity of demand and price are constant across the time period.
- All train services remain constant across the time period.
- No external shock affecting the price and demand.

4.2 Model Formulation

In this section, we will define the variables that could best explain the demand function.

Independent Variable:
We will use ln(seats) as the total number of seats sold per day estimates the quantity of train tickets demanded.

Dependent Variables:
We chose ln(price) as the first dependent variable that will be adopted in the structural model as price is always a factor in a demand function.

Beside all the excluded variables mentioned in part 3.4, the customer category variable will also be excluded from the structure model. Customer segmentation was done by the train provider and is therefore not relevant to demand. Train riders are most likely not aware of this segmentation.

There are 3 other potential dependent variables: advanced purchase, cabin type and weekend factor. To determine the best variables, we did a backward step model selection to test whether each variable is a good demand estimator by considering its coefficient, p-values and the model's adjusted R^2.

Structural Model 1

To explore the model, we ran OLS with all the candidate dependent variables.

$$ln(seats) = 1.7024 - 0.2236 * \ln(price) + 0.0013 * days\_in\_advance + 0.0144 * isNormCabin + 0.0049 * isWeekend + \epsilon$$

Weekend factor is insignificant, with a p-value of 0.1083. We further observe that cabin type is also insignificant - when it is removed, the adjusted R^2 value remains at 0.090 (*See Appendix 3*). As such, we can remove both variables.

Intuitively, cabin type cannot be an exogenous variable for demand as it does not affect the number of tickets demanded. Instead, cabin type should affect the number of tickets supplied. The supply of a special cabin is usually less than that of a normal cabin, perhaps because a special cabin takes more space as it is bigger. Furthermore, cabin type is correlated with price as a special cabin is usually priced higher than a normal cabin.

Structural Model 2

After removing the insignificant variables, we find that advanced purchase is a viable exogenous variable. The revised model is indeed suitable with significant variables with p-value < 0.05 and correct slope. Advanced purchase has a positive slope of 0.0013 as travel groups will tend to purchase tickets in advance, while ln(price) has a negative slope of -0.2236 as an increase in price will reduce demand (*See Appendix 3*).

The chosen demand structural function can be written as below:

$$ln(seats) = 1.7541 - 0.2316 * ln(price) + 0.0013 * days\_in\_advance + \epsilon$$

The variables which were not selected in the structural model could be a potential Instrument Variable (IV).

Simultaneity Problem:

The demand curve suffers from a simultaneity problem due to its correlation with the supply curve. As such, the OLS function used to estimate the demand function will suffer from endogeneity bias.

We identify that ln(price) is the endogenous variable in the structural model. Ticket price change could be caused by hidden variables, such as supply shock. When the hidden variables change the ticket price, it will also affect the demand. A model which contains an endogenous term will be inaccurate as it suffers a bias where its error term is not completely random.

Instrument Variable (IV)

There are 3 potential IVs for ln(price): cabin type, customer category and weekend factor. The chosen instrument variable must be correlated with ln(price) but must not correlate with ln(seats). We adopted a forward step model selection to find the best IV model.

We can estimate the reduced form by expressing ln(price) in terms of the chosen IV:

$$\widehat{ln(price)} = \pi_0 + \pi_1 * feature1 + \pi_2 * feature2 + \pi_3 * feature3 + \ldots + \nu$$

IV Model 1 - Cabin Type

As shown in the structural model, cabin type is not directly linked to ln(seats). This was further validated in the correlation matrix where we found that cabin type was more correlated with ln(price), with coefficient -0.71 than with ln(seats), with coefficient 0.21 (*See Appendix 5*).

Running Model 1:

$$\widehat{ln(price)} = 5.8739 - 0.0029 * days\_in\_advance - 0.7367 * isNormCabin + \nu$$

The cabin type is significant, passed the Hausman test and gives Adjusted R^2 of 0.596, the highest among other variables. Thus, this is the best IV for price (*See Appendix 2*).

IV Model 2 - Customer Category

Customer category was explored as an IV because they might segment leisure vs business travellers. This is akin to cabin type which would have a correlation with price but not directly with seats. The results from the correlation matrix does not lend credence to it however, indicating a similarly weak correlation with ln(price) (-0.49) vs ln(seats) (0.28) (*See Appendix 5*).

Running Model 2 gives :

$$\widehat{ln(price)} = 5.8849 - 0.0041 * days\_in\_advance - 0.4885 * Customer\_Cat + \nu$$

Customer category variable is also significant and passed the Hausman test - albeit, it's Adjusted R^2 is 0.400, lower than that of cabin type (*See Appendix 2*).

IV Model 3 - Weekend Factor

For the weekend factor, the results showed that it has very little correlation not only with ln(price) and ln(seats) but with virtually every other variable (*See Appendix 5*).

Running Model 3 gives:

$$\widehat{ln(price)} = 5.5605 - 0.0052 * days\_in\_advance + 0.0576 * isWeekend + \nu$$

While the weekend variable is significant and passed the Hausman test, the R^2 of this model is the lowest among other models at 0.303. The weekend variable can thus be categorised as a weak instrument variable (*See Appendix 2*).

We also explored models where we combined the potential IVs.

IV Model 4 - Cabin Type & Customer Category:

Running Model 4 gives

$$\widehat{ln(price)} = 5.9836 - 0.0026 * days\_in\_advance - 0.6694 * isNormCabin - 0.2162 * Customer\_Cat + \nu$$

IV Model 5 - Cabin Type & Weekend Factor:

Running Model 5 gives

$$\widehat{ln(price)} = 5.8589 - 0.0029 * days\_in\_advance - 0.7367 * isNormCabin + 0.0583 * isWeekend + \nu$$

Whilst the variables in IV Model 4 and 5 are all significant and passed the Hausman test, both models did not pass the Sargan test as all p-values are < 0.05. Hence, the hypothesis that all IVs are exogenous were rejected in both models (*See Appendix 2*).

Thus, IV Model 1 was chosen. It can be seen from Model 1 reduced form that the F-statistic - 1.544e05 - is large and p-value < 0.05 is significant. Thus we can reject the null hypothesis that cabin type is a weak instrument (*See Appendix 2*).

4.3 Final 2SLS Model

Based on the chosen IV Model, the final 2SLS was run with cabin type as the sole IV

Final Reduced Form Model:

$$\widehat{ln(price)} = 5.8739 - 0.0029 * days\_in\_advance - 0.7367 * isNormCabin + \nu$$

Reduced Form Model Interpretation

The cabin type coefficient of -0.7367 indicates that price will drop by 73.67% when the cabin type is normal. This supports our understanding that a normal cabin is always cheaper than a special cabin.

On the other hand, the coefficient of advanced purchase is -0.0029, showing that price is higher when a ticket is purchased closer to the departure date. This also aligns with our understanding because train companies usually price rushed tickets higher as they understand that travellers are willing to pay more when there is urgency.

Final 2SLS Model:

$$ln(seats) = 1.8183 - 0.2431 * \widehat{ln(price)} + 0.0012 * days\_in\_advance + \epsilon$$

2SLS Model Interpretation

The ln(price) coefficient of -0.2431 indicates a relatively inelastic demand as a 1% change in price will lead to only a 0.24% change in quantity demanded. This is in line with our understanding from business insights that the travellers will still buy train tickets if price increases given that they need to make the trip.

On the other hand, the coefficient of 0.0012 for advanced purchase is small but significant. It is observed that the customers tend to buy tickets in advance of the departure date. As such, this could have a larger effect on the quantity demanded.

5. Conclusion

By estimating ln(price) with cabin type, we have removed the hidden variables affecting price from this model. The chosen IV model passed all the endogeneity tests and we found that the original OLS model indeed suffered from endogeneity bias.

The coefficients of all the variables used in both the IV and 2SLS models are in line with our understanding and business applications - as described in part 4.3. Thus, the final 2SLS model seems to have solved the endogeneity problem posed by the original OLS model. Nonetheless, the model has a large residual as evident in the low adjusted R^2, and as such, there is room for potential improvements.

5.1 Potential Improvements

- **Better context on datasets and its sampling methods:** since the dataset was given and utilised on an 'as-is' basis, some of the features, such as cumulative sales, were dropped in the feature selection phase due their inconsistency and lack of information on their constituents.

  From a business standpoint, cumulative sales is a variable that would be a good instrumental variable, as train companies' would naturally include it into their pricing strategy. When cumulative sales are high and seats are limited, suppliers would increase prices. If more information were available, this might affect the choice of variables selected which could in turn improve the reliability of the model.

- **Better instrument variables:** The model could also be improved with potentially better candidate instrumental variables that were not available in the dataset. For example, variables like the cost of resources – diesel, electricity, oil, and coal for instance – to power trains would intuitively be included in the supply function and would probably be highly correlated with ticket prices. These variables could further reduce the endogeneity bias.

Appendix 1 - Variable Name & Source

Variable Name	Addressed in Report As	Source
num_seats_total	Number of Seats	Given in the problem set
mean_net_ticket_price	Ticket Price	
Dept_Date	Departure Date	
Purchase_Date	Purchase Date	
Train_Number_All	Train Number	
Culmulative_sales	Cumulative Sales	
isNormCabin	Cabin Type	
isReturn	Return Trip	
isOneway	One-way Trip	
Customer_Cat	Customer Category	
days_in_advance	Advanced Purchase	Created by deducting departure date with purchase date.
isWeekend	Weekend Factor	Created by identifying weekends from departure date.
ln_seats	Ln(Seats)	Log transformed number of seats.
ln_price	Ln(Price)	Log transformed ticket price.

Appendix 2 - Coefficients, p-values and endogeneity tests for all Instrument Variable (IV) Models Explored (Model 1 - 5)

	Model 1 (Adj R-2 = 0.596)		Model 2 (Adj R-2 = 0.400)		Model 3 (Adj R-2 = 0.303)	
Instrument Variables	isNormCabin		Customer_Cat		isWeekend	
Exogenous Variable	days_in_advance		days_in_advance		days_in_advance	
	Coefficient	**p-value**	**Coefficient**	**p-value**	**Coefficient**	**p-value**
Intercept	5.8739	0.0000	5.8849	0.0000	5.5600	0.0000
days_in_advance	-0.0029	0.0000	-0.0041	0.0000	-0.0052	0.0000
isNormCabin	-0.7367	0.0000				
isWeekend					0.0576	0.0000
Customer_Cat			-0.4885	0.0000		
IV Tests						
Weak Instrument	Pass		Pass		Pass	
Hausman Test	Pass		Pass		Pass	
Sargan Test	N/A		N/A		N/A	

	Model 4 (Adj R-2 = 0.613)		Model 5 (Adj R-2 = 0.597)	
Instrument Variables	isNormCabin & Customer_Cat		isNormCabin & isWeekend	
Exogenous Variable	days_in_advance		days_in_advance	
	Coefficient	**p-value**	**Coefficient**	**p-value**
Intercept	5.9836	0.0000	5.8589	0.0000
days_in_advance	-0.0026	0.0000	-0.0029	0.0000
isNormCabin	-0.6694	0.0000	-0.7367	0.0000
isWeekend			0.0583	0.0000
Customer_Cat	-0.2162	0.0000		
IV Tests				
Weak Instrument	Pass		Pass	
Hausman Test	Pass		Pass	
Sargan Test	Fail		Fail	

Appendix 3 - Coefficients and p-values of all Structural Models Explored (Model 1 & 2)

	Model 1 (Adj R-2 = 0.090)		Model 2 (Adj R-2 = 0.090)	
Endogeneous Variables	ln_price		ln_price	
Exogenous Variables	days_in_advance, isNormCabin, isWeekend		days_in_advance	
	Coefficient	**p-value**	**Coefficient**	**p-value**
Intercept	1.7024	**0.0000**	1.7541	0.0000
ln_price	-0.2236	0.0000	-0.2316	0.0000
days_in_advance	0.0013	0.0000	0.0013	0.0000
isNormCabin	0.0144	0.0004		
isWeekend	0.0049	0.1083		

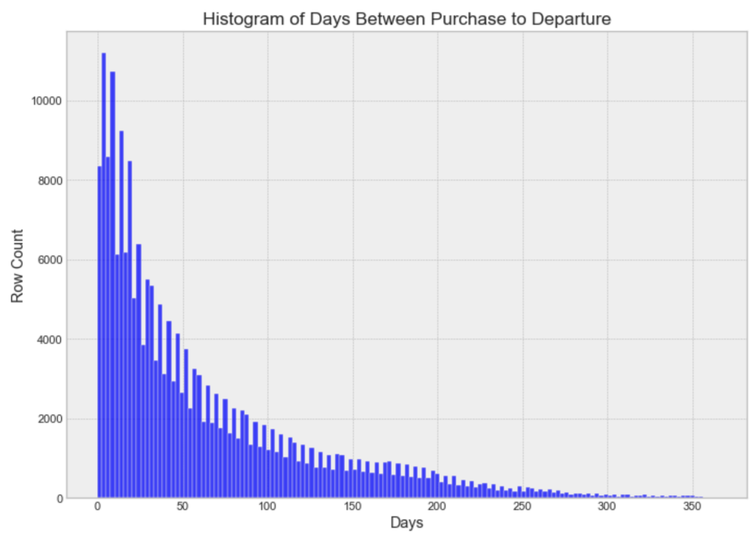
Appendix 4 - Coefficients and p-values of Final 2SLS Model

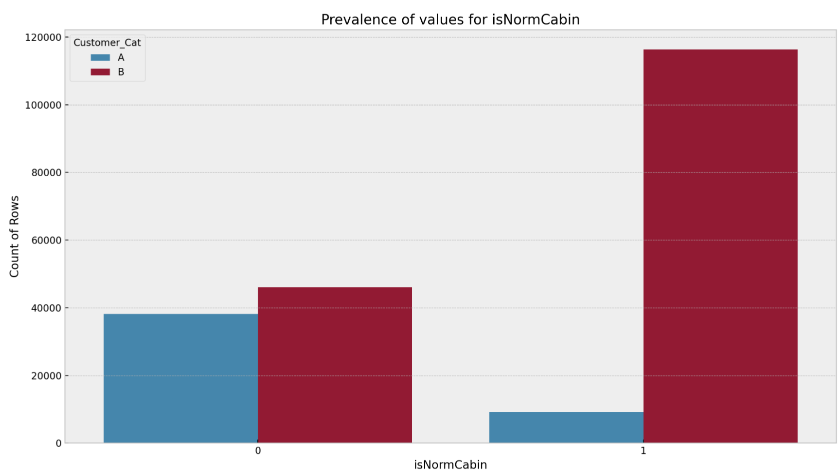
	Model 1 (Adj R-2 = 0.073)	
Variables	ln_price, days_in_advance	
	Coefficient	**p-value**
Intercept	1.8183	0.0000
ln_price	-0.2431	0.0000
days_in_advance	0.0012	0.0000

Appendix 5 - Correlation Matrix (rounded to 2 decimal places)

	ln_seat	ln_price	isNormCabin	Customer_Cat	isWeekend_Dept
ln_seat	1.00	-0.28	0.21	0.28	0.01
ln_price	-0.28	1.00	-0.72	-0.49	0.01
isNormCabin	0.21	-0.72	1.00	0.45	0.02
Customer_Cat	0.28	-0.49	0.45	1.00	0.04
isWeekend_Dept	0.01	0.01	0.02	0.04	1.00

Appendix 6 - Data profiling.





Full Pandas Profiling report: https://bit.ly/2Y2Lbou

Appendix 7 - Detailed summary of all models (IV, structural and final 2 SLS) with code written in Python.

In [1]:

```
import pandas as pd
import numpy as np
import datetime as dt

from statsmodels.formula.api import ols
import scipy

from sklearn.metrics import r2_score

import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.ticker as ticker
from matplotlib.pyplot import figure

# Read Data

df = pd.read_csv(
  filepath_or_buffer='Data-GP1.csv',
  header='infer',
  index_col=False,
  parse_dates=['Dept_Date','Purchase_Date'],
  infer_datetime_format=True
)

# Feature Transformation
df['ln_price'] = np.log(df['mean_net_ticket_price'])
df['ln_seats'] = np.log(df['num_seats_total'])
df['Customer_Cat'].iloc[df['Customer_Cat'] == 'A'] = 0
df['Customer_Cat'].iloc[df['Customer_Cat'] == 'B'] = 1

# Feature Engineering
df['days_in_advance'] = (df['Dept_Date'] - df['Purchase_Date']) / np.timedelta64(1, "D")
df['Dept_Date'] = pd.to_datetime(df['Dept_Date'], unit='s')
df['isWeekend'] = df['Dept_Date'].dt.dayofweek
df['isWeekend'].iloc[df['isWeekend'] <= 4] = 0
df['isWeekend'].iloc[df['isWeekend'] > 4] = 1

df_departure_date_dummy = pd.get_dummies(data= df['Train_Number_All'])

df = pd.concat(objs=[df,df_departure_date_dummy],axis=1)

df.rename(
  columns={
    0: "train_A",
    1: 'train_B',
    2: 'train_C',
    3: 'train_D',
    4: 'train_E',
    5: 'train_F',
    6: 'train_G',
    7: 'train_H',
    8: 'train_I',
    9: 'train_J',
    10: 'train_K',
    11: 'train_L',
    12: 'train_M',
    13: 'train_N',
    14: 'train_O'
  },
  inplace=True
)

# Rename Columns
df.rename(columns = {'Culmulative_sales' : 'cum_sales'}, inplace = True)
```

```
/Users/salimwid/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py:1637: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#retur
ning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
/Users/salimwid/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py:1637: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#retur
ning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
/Users/salimwid/opt/anaconda3/lib/python3.8/site-packages/pandas/core/indexing.py:1637: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#retur
ning-a-view-versus-a-copy
```

```
self._setitem_single_block(indexer, value, name)
```

In [2]:
```
df.describe()
```

Out[2]:

	num_seats_total	mean_net_ticket_price	cum_sales	isNormCabin	isReturn	isOneway	ln_price	ln_seats	day
count	209697.000000	209697.000000	209697.000000	209697.000000	209697.000000	209697.000000	209697.000000	209697.000000	2
mean	2.383019	230.116900	15.875063	0.598249	0.480183	0.122873	5.250089	0.618505	
std	2.083324	147.024784	19.795677	0.490253	0.499608	0.328292	0.608971	0.661471	
min	1.000000	1.278969	1.000000	0.000000	0.000000	0.000000	0.246054	0.000000	
25%	1.000000	108.870193	3.000000	0.000000	0.000000	0.000000	4.690156	0.000000	
50%	2.000000	186.282199	8.000000	1.000000	0.000000	0.000000	5.227263	0.693147	
75%	3.000000	350.409481	21.000000	1.000000	1.000000	0.000000	5.859102	1.098612	
max	66.000000	7855.766106	187.000000	1.000000	1.000000	1.000000	8.969003	4.189655	

8 rows × 25 columns

In [3]:
```
#Demand Structural Form - Model 1 (All available variables)

num_seats_sf = ols('ln_seats ~ ln_price + days_in_advance + isNormCabin + isWeekend', df).fit()
num_seats_sf.summary2()
```

Out[3]:

Model:	OLS	Adj. R-squared:	0.090
Dependent Variable:	ln_seats	AIC:	401927.9372
Date:	2021-09-26 23:03	BIC:	401979.2043
No. Observations:	209697	Log-Likelihood:	-2.0096e+05
Df Model:	4	F-statistic:	5203.
Df Residuals:	209692	Prob (F-statistic):	0.00
R-squared:	0.090	Scale:	0.39804

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	1.7024	0.0210	80.9399	0.0000	1.6612	1.7436
ln_price	-0.2236	0.0036	-62.6993	0.0000	-0.2306	-0.2166
days_in_advance	0.0013	0.0000	49.7516	0.0000	0.0012	0.0013
isNormCabin	0.0144	0.0040	3.5554	0.0004	0.0065	0.0223
isWeekend	0.0049	0.0031	1.6059	0.1083	-0.0011	0.0110

Omnibus:	15267.397	Durbin-Watson:	1.700
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18862.003
Skew:	0.733	Prob(JB):	0.000
Kurtosis:	2.912	Condition No.:	1418

In [4]:
```
#Demand Structural Form - Model 2 (ln(Price) + Days in Advance)

num_seats_sf2 = ols('ln_seats ~ ln_price + days_in_advance', df).fit()
num_seats_sf2.summary2()
```

Out[4]:

Model:	OLS	Adj. R-squared:	0.090
Dependent Variable:	ln_seats	AIC:	401939.6982
Date:	2021-09-26 23:03	BIC:	401970.4584
No. Observations:	209697	Log-Likelihood:	-2.0097e+05
Df Model:	2	F-statistic:	1.040e+04
Df Residuals:	209694	Prob (F-statistic):	0.00
R-squared:	0.090	Scale:	0.39807

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	1.7541	0.0152	115.2879	0.0000	1.7243	1.7839
ln_price	-0.2316	0.0027	-85.5460	0.0000	-0.2369	-0.2263
days_in_advance	0.0013	0.0000	50.0564	0.0000	0.0012	0.0013

Omnibus:	15293.982	Durbin-Watson:	1.700
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18901.868
Skew:	0.734	Prob(JB):	0.000
Kurtosis:	2.912	Condition No.:	1018

In [5]:

```
#IV Models

#IV Model 1 - Predict ln_price with isNormCabin:
pred_ln_price_rf1 = ols('ln_price ~ days_in_advance + isNormCabin', df).fit()
pred_ln_price_rf1.summary2()
```

Out[5]:

Model:	OLS	Adj. R-squared:	0.596
Dependent Variable:	ln_price	AIC:	197216.8153
Date:	2021-09-26 23:03	BIC:	197247.5756
No. Observations:	209697	Log-Likelihood:	-98605.
Df Model:	2	F-statistic:	1.544e+05
Df Residuals:	209694	Prob (F-statistic):	0.00
R-squared:	0.596	Scale:	0.14996

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	5.8739	0.0014	4178.5832	0.0000	5.8711	5.8766
days_in_advance	-0.0029	0.0000	-203.5792	0.0000	-0.0029	-0.0029
isNormCabin	-0.7367	0.0019	-390.5586	0.0000	-0.7404	-0.7330

Omnibus:	7412.638	Durbin-Watson:	1.308
Prob(Omnibus):	0.000	Jarque-Bera (JB):	16235.415
Skew:	-0.225	Prob(JB):	0.000
Kurtosis:	4.287	Condition No.:	225

In [6]:

```
#IV Model 1 - Predict ln_price with isNormCabin
#Hausman Test

df['res1stage'] = pred_ln_price_rf1.resid
hausman_test = ols('ln_seats ~ ln_price + days_in_advance + res1stage', df).fit()
hausman_test.summary2()
```

Out[6]:

Model:	OLS	Adj. R-squared:	0.090
Dependent Variable:	ln_seats	AIC:	401928.5161
Date:	2021-09-26 23:03	BIC:	401969.5298
No. Observations:	209697	Log-Likelihood:	-2.0096e+05
Df Model:	3	F-statistic:	6937.
Df Residuals:	209693	Prob (F-statistic):	0.00
R-squared:	0.090	Scale:	0.39805

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	1.8183	0.0233	77.9163	0.0000	1.7726	1.8641
ln_price	-0.2431	0.0042	-58.2769	0.0000	-0.2513	-0.2349
days_in_advance	0.0012	0.0000	40.1242	0.0000	0.0012	0.0013
res1stage	0.0199	0.0055	3.6307	0.0003	0.0092	0.0307

Omnibus:	15262.317	Durbin-Watson:	1.700
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18854.023
Skew:	0.733	Prob(JB):	0.000
Kurtosis:	2.911	Condition No.:	1587

In [7]:

```
#IV Model 2 - Predict ln_price with Customer_Cat
pred_ln_price_rf2 = ols('ln_price ~ days_in_advance + Customer_Cat', df).fit()
pred_ln_price_rf2.summary2()
```

Out[7]:

Model:	OLS	Adj. R-squared:	0.400
Dependent Variable:	ln_price	AIC:	279819.3957

Date:	2021-09-26 23:03	BIC:	279850.1560
No. Observations:	209697	Log-Likelihood:	-1.3991e+05
Df Model:	2	F-statistic:	7.002e+04
Df Residuals:	209694	Prob (F-statistic):	0.00
R-squared:	0.400	Scale:	0.22235

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	5.8849	0.0022	2676.6526	0.0000	5.8806	5.8892
Customer_Cat[T.1]	-0.4885	0.0026	-186.0062	0.0000	-0.4936	-0.4833
days_in_advance	-0.0041	0.0000	-240.3075	0.0000	-0.0041	-0.0040

Omnibus:	4124.777	Durbin-Watson:	1.011
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3903.444
Skew:	0.296	Prob(JB):	0.000
Kurtosis:	2.690	Condition No.:	284

In [8]:

```
#IV Model 2 - Predict ln_price with Customer_Cat
#Hausman Test

df['res1stage'] = pred_ln_price_rf2.resid
hausman_test = ols('ln_seats ~ ln_price + days_in_advance + res1stage', df).fit()
hausman_test.summary2()
```

Out[8]:

Model:	OLS	Adj. R-squared:	0.114
Dependent Variable:	ln_seats	AIC:	396314.7016
Date:	2021-09-26 23:03	BIC:	396355.7152
No. Observations:	209697	Log-Likelihood:	-1.9815e+05
Df Model:	3	F-statistic:	9022.
Df Residuals:	209693	Prob (F-statistic):	0.00
R-squared:	0.114	Scale:	0.38753

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	4.5227	0.0396	114.1635	0.0000	4.4451	4.6004
ln_price	-0.7282	0.0071	-102.5954	0.0000	-0.7421	-0.7143
days_in_advance	-0.0013	0.0000	-30.5404	0.0000	-0.0014	-0.0012
res1stage	0.5785	0.0077	75.5186	0.0000	0.5635	0.5935

Omnibus:	13863.145	Durbin-Watson:	1.751
Prob(Omnibus):	0.000	Jarque-Bera (JB):	16813.582
Skew:	0.693	Prob(JB):	0.000
Kurtosis:	2.928	Condition No.:	2730

In [9]:

```
#IV Model 3 - Predict ln_price with isWeekend
pred_ln_price_rf3 = ols('ln_price ~ days_in_advance + isWeekend', df).fit()
pred_ln_price_rf3.summary2()
```

Out[9]:

Model:	OLS	Adj. R-squared:	0.303
Dependent Variable:	ln_price	AIC:	311303.9021
Date:	2021-09-26 23:03	BIC:	311334.6624
No. Observations:	209697	Log-Likelihood:	-1.5565e+05
Df Model:	2	F-statistic:	4.564e+04
Df Residuals:	209694	Prob (F-statistic):	0.00
R-squared:	0.303	Scale:	0.25837

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	5.5605	0.0017	3321.3392	0.0000	5.5572	5.5638
days_in_advance	-0.0052	0.0000	-302.0332	0.0000	-0.0052	-0.0051
isWeekend	0.0576	0.0025	23.2441	0.0000	0.0527	0.0624

Omnibus:	9497.057	Durbin-Watson:	0.865
Prob(Omnibus):	0.000	Jarque-Bera (JB):	4162.500

Skew:	0.101	Prob(JB):	0.000
Kurtosis:	2.340	Condition No.:	213

In [10]:
```
#IV Model 3 - Predict ln_price with Customer_Cat
#Hausman Test

df['res1stage'] = pred_ln_price_rf3.resid
hausman_test = ols('ln_seats ~ ln_price + days_in_advance + res1stage', df).fit()
hausman_test.summary2()
```

Out[10]:

Model:	OLS	Adj. R-squared:	0.090
Dependent Variable:	ln_seats	AIC:	401938.5782
Date:	2021-09-26 23:03	BIC:	401979.5919
No. Observations:	209697	Log-Likelihood:	-2.0097e+05
Df Model:	3	F-statistic:	6933.
Df Residuals:	209693	Prob (F-statistic):	0.00
R-squared:	0.090	Scale:	0.39807

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	1.2289	0.2977	4.1275	0.0000	0.6453	1.8124
ln_price	-0.1374	0.0534	-2.5727	0.0101	-0.2420	-0.0327
days_in_advance	0.0018	0.0003	6.3666	0.0000	0.0012	0.0023
res1stage	-0.0944	0.0535	-1.7663	0.0773	-0.1992	0.0104

Omnibus:	15298.832	Durbin-Watson:	1.700
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18909.532
Skew:	0.734	Prob(JB):	0.000
Kurtosis:	2.912	Condition No.:	20244

In [11]:
```
# IV Model 4 - Predict ln_price with isNormCabin & CustomerCat
pred_ln_price_rf4 = ols('ln_price ~ days_in_advance + isNormCabin + Customer_Cat', df).fit()
pred_ln_price_rf4.summary2()
```

Out[11]:

Model:	OLS	Adj. R-squared:	0.613
Dependent Variable:	ln_price	AIC:	188253.9063
Date:	2021-09-26 23:03	BIC:	188294.9199
No. Observations:	209697	Log-Likelihood:	-94123.
Df Model:	3	F-statistic:	1.105e+05
Df Residuals:	209693	Prob (F-statistic):	0.00
R-squared:	0.613	Scale:	0.14368

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	5.9836	0.0018	3340.5153	0.0000	5.9801	5.9872
Customer_Cat[T.1]	-0.2162	0.0023	-95.7033	0.0000	-0.2206	-0.2117
days_in_advance	-0.0026	0.0000	-184.2820	0.0000	-0.0026	-0.0026
isNormCabin	-0.6694	0.0020	-338.8437	0.0000	-0.6733	-0.6655

Omnibus:	4415.336	Durbin-Watson:	1.296
Prob(Omnibus):	0.000	Jarque-Bera (JB):	9889.068
Skew:	-0.012	Prob(JB):	0.000
Kurtosis:	4.064	Condition No.:	295

In [12]:
```
#IV Model 4 - Predict ln_price with isNormCabin & CustomerCat
#Hausman Test

df['res1stage'] = pred_ln_price_rf4.resid
hausman_test = ols('ln_seats ~ ln_price + days_in_advance + res1stage', df).fit()
hausman_test.summary2()
```

Out[12]:

Model:	OLS	Adj. R-squared:	0.093
Dependent Variable:	ln_seats	AIC:	401242.9732

Date:	2021-09-26 23:03	BIC:	401283.9869
No. Observations:	209697	Log-Likelihood:	-2.0062e+05
Df Model:	3	F-statistic:	7189.
Df Residuals:	209693	Prob (F-statistic):	0.00
R-squared:	0.093	Scale:	0.39675

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	2.1989	0.0227	97.0386	0.0000	2.1545	2.2434
ln_price	-0.3114	0.0040	-76.8853	0.0000	-0.3193	-0.3034
days_in_advance	0.0009	0.0000	28.9284	0.0000	0.0008	0.0009
res1stage	0.1439	0.0054	26.4552	0.0000	0.1332	0.1545

Omnibus:	14965.500	Durbin-Watson:	1.713
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18413.925
Skew:	0.725	Prob(JB):	0.000
Kurtosis:	2.913	Condition No.:	1543

In [13]:
```
#IV Model 4 - Predict ln_price with isNormCabin & CustomerCat
#Sargan Test

ln_price_pred4 = pred_ln_price_rf4.predict(df[['days_in_advance','isNormCabin','Customer_Cat']])
df['error'] = df['ln_seats'] - ln_price_pred4

sargan_test_model = ols('error ~ isNormCabin + Customer_Cat', df).fit()
sargan_test_model.summary2()
```

Out[13]:

Model:	OLS	Adj. R-squared:	0.468
Dependent Variable:	error	AIC:	427187.9545
Date:	2021-09-26 23:03	BIC:	427218.7147
No. Observations:	209697	Log-Likelihood:	-2.1359e+05
Df Model:	2	F-statistic:	9.213e+04
Df Residuals:	209694	Prob (F-statistic):	0.00
R-squared:	0.468	Scale:	0.44900

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	-5.7020	0.0031	-1811.1925	0.0000	-5.7082	-5.6958
Customer_Cat[T.1]	0.6726	0.0039	172.0144	0.0000	0.6650	0.6803
isNormCabin	0.9185	0.0033	275.5501	0.0000	0.9120	0.9251

Omnibus:	11786.264	Durbin-Watson:	1.547
Prob(Omnibus):	0.000	Jarque-Bera (JB):	13818.479
Skew:	0.625	Prob(JB):	0.000
Kurtosis:	2.864	Condition No.:	5

In [14]:
```
#IV Model 5 - Predict ln_price with isNormCabin & isWeekend
pred_ln_price_rf5 = ols('ln_price ~ days_in_advance + isNormCabin + isWeekend', df).fit()
pred_ln_price_rf5.summary2()
```

Out[14]:

Model:	OLS	Adj. R-squared:	0.597
Dependent Variable:	ln_price	AIC:	196260.9980
Date:	2021-09-26 23:03	BIC:	196302.0117
No. Observations:	209697	Log-Likelihood:	-98126.
Df Model:	3	F-statistic:	1.038e+05
Df Residuals:	209693	Prob (F-statistic):	0.00
R-squared:	0.597	Scale:	0.14927

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	5.8589	0.0015	3949.6813	0.0000	5.8560	5.8618
days_in_advance	-0.0029	0.0000	-205.2626	0.0000	-0.0029	-0.0029
isNormCabin	-0.7367	0.0019	-391.4823	0.0000	-0.7404	-0.7330
isWeekend	0.0583	0.0019	30.9837	0.0000	0.0546	0.0620

Omnibus:	7611.071	Durbin-Watson:	1.307
Prob(Omnibus):	0.000	Jarque-Bera (JB):	17026.856
Skew:	-0.225	Prob(JB):	0.000
Kurtosis:	4.321	Condition No.:	233

In [15]:

```
#IV Model 5 - Predict ln_price with isNormCabin & isWeekend
#Hausman Test

df['res1stage'] = pred_ln_price_rf5.resid
hausman_test = ols('ln_seats ~ ln_price + days_in_advance + res1stage', df).fit()
hausman_test.summary2()
```

Out[15]:

Model:	OLS	Adj. R-squared:	0.090
Dependent Variable:	ln_seats	AIC:	401929.8329
Date:	2021-09-26 23:03	BIC:	401970.8466
No. Observations:	209697	Log-Likelihood:	-2.0096e+05
Df Model:	3	F-statistic:	6936.
Df Residuals:	209693	Prob (F-statistic):	0.00
R-squared:	0.090	Scale:	0.39805

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	1.8147	0.0233	78.0024	0.0000	1.7691	1.8603
ln_price	-0.2425	0.0042	-58.3028	0.0000	-0.2506	-0.2343
days_in_advance	0.0012	0.0000	40.2984	0.0000	0.0012	0.0013
res1stage	0.0189	0.0055	3.4446	0.0006	0.0081	0.0296

Omnibus:	15262.980	Durbin-Watson:	1.700
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18854.956
Skew:	0.733	Prob(JB):	0.000
Kurtosis:	2.911	Condition No.:	1582

In [16]:

```
#IV Model 5 - Predict ln_price with isNormCabin & isWeekend
#Sargan Test

ln_price_pred5 = pred_ln_price_rf5.predict(df[['days_in_advance','isNormCabin','isWeekend']])
df['error'] = df['ln_seats'] - ln_price_pred5

sargan_test_model = ols('error ~ isNormCabin + isWeekend', df).fit()
sargan_test_model.summary2()
```

Out[16]:

Model:	OLS	Adj. R-squared:	0.405
Dependent Variable:	error	AIC:	444602.7686
Date:	2021-09-26 23:03	BIC:	444633.5289
No. Observations:	209697	Log-Likelihood:	-2.2230e+05
Df Model:	2	F-statistic:	7.123e+04
Df Residuals:	209694	Prob (F-statistic):	0.00
R-squared:	0.405	Scale:	0.48788

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	-5.3240	0.0026	-2069.4663	0.0000	-5.3290	-5.3190
isNormCabin	1.1745	0.0031	377.4206	0.0000	1.1684	1.1806
isWeekend	-0.0367	0.0034	-10.7916	0.0000	-0.0434	-0.0300

Omnibus:	13684.573	Durbin-Watson:	1.479
Prob(Omnibus):	0.000	Jarque-Bera (JB):	16377.043
Skew:	0.679	Prob(JB):	0.000
Kurtosis:	2.822	Condition No.:	3

In [18]:

```
#Final 2SLS Model - with isNormCabin as IV to predict ln_price

num_seats_sf_final = ols('ln_seats ~ ln_price + days_in_advance', df).fit()
```

```python
iv_final = ols('ln_price ~ days_in_advance + isNormCabin', df).fit()
pred_ln_price = iv_final.predict(df[['days_in_advance', 'isNormCabin']])

sls_final = ols('ln_seats ~ pred_ln_price + days_in_advance', df).fit()
sls_final.summary2()
```

Out[18]:

Model:	OLS	Adj. R-squared:	0.073
Dependent Variable:	ln_seats	AIC:	405825.6369
Date:	2021-09-26 23:03	BIC:	405856.3972
No. Observations:	209697	Log-Likelihood:	-2.0291e+05
Df Model:	2	F-statistic:	8282.
Df Residuals:	209694	Prob (F-statistic):	0.00
R-squared:	0.073	Scale:	0.40552

	Coef.	Std.Err.	t	P>|t|	[0.025	0.975]
Intercept	1.8183	0.0236	77.1955	0.0000	1.7722	1.8645
pred_ln_price	-0.2431	0.0042	-57.7378	0.0000	-0.2514	-0.2349
days_in_advance	0.0012	0.0000	39.7530	0.0000	0.0011	0.0013

Omnibus:	14984.644	Durbin-Watson:	1.701
Prob(Omnibus):	0.000	Jarque-Bera (JB):	18368.375
Skew:	0.722	Prob(JB):	0.000
Kurtosis:	2.864	Condition No.:	1562

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js