IO ECO 7408: Problem Set 1 – Binary Logit Demand Estimation for Ride-Sharing Services

Instructor: Prof. Douglas Turner

Student: C.H.

In [25]:
```
globals().clear()
```

In [26]:
```
import sys
import numpy as np
import pandas as pd
import statsmodels.api as sm
from statsmodels.discrete.discrete_model import Logit
from scipy.optimize import minimize

import matplotlib.pyplot as plt
```

Questions

1. Data Setup and Variable Construction

(a) Load the dataset into Python using pandas. Print the first few rows to inspect the data.

In [27]:
```
df = pd.read_csv('/Users/terrylu/Desktop/UF/Courses/2025-2026/IO/IO_Co

print(df.head(5))
```

```
   wait_time_uber  wait_time_lyft  price_uber  price_lyft  age  choice_
is_uber
0       18.528105       15.619448    8.296000   26.216232   29
1
1       15.800314       13.525088   15.646098   16.062803   38
0
2       16.957476       11.926160   18.613206   21.884515   53
1
3       19.481786       13.875490   23.575972   20.009804   50
0
4       18.735116       11.800978    9.246689   24.669068   21
1
```

(b) Construct the following new variables:

- `wait_diff: Uber wait time minus Lyft wait time`
- `price_diff: Uber price minus Lyft price`

Also include choice_is_uber in your dataframe.

Hint: A more positive value of wait_diff or price_diff implies Uber is worse than lyft in that attribute.

In [28]:
```
df['wait_diff'] = df['wait_time_uber'] - df['wait_time_lyft']
df['price_diff'] = df['price_uber'] - df['price_lyft']
```

2. Estimating the Logit Model by Hand

(a) Using only numpy and scipy.optimize.minimize, estimate a binary logit model of the form:

Write out the log-likelihood function manually and optimize it using BFGS.

In [29]:
```
img = plt.imread('/Users/terrylu/Desktop/UF/Courses/2025-2026/IO/IO_Co
plt.imshow(img); plt.axis("off");
```

$$\Pr(\text{choose Uber}) = \frac{\exp(\beta_1 \cdot \text{wait\_diff} + \beta_2 \cdot \text{price\_diff})}{1 + \exp(\beta_1 \cdot \text{wait\_diff} + \beta_2 \cdot \text{price\_diff})}$$

Write out the log-likelihood function manually and optimize it using BFGS.

In [30]:
```
y = df['choice_is_uber']
X = df[['wait_diff', 'price_diff']]


def log_likelihood(beta): # define the log-likelihood function

    V = X @ beta  # systematic utility

    P = np.exp(V) / (1 + np.exp(V))

    # avoid log(0)
    esp = 1e-12
    P = np.clip(P, esp, 1-esp)

    # log-likelihood
    ll = y * np.log(P) + (1 - y) * np.log(1 - P)
```

```python
    return -np.sum(ll)  # negative log-likelihood for minimization


# initial guess for beta
beta_init = np.zeros(X.shape[1])

# optimize the log-likelihood function
result_1 = minimize(log_likelihood, beta_init, method = 'BFGS')
```

(b) Report the estimated coefficients. Do the signs of the estimates make sense? Why?

In [31]:
```python
# print the results
beta_hat_1 = result_1.x
print(f"Estimated coefficients (by hand):\n{beta_hat_1}")
```

```
Estimated coefficients (by hand):
[-0.5241069  -2.03895639]
```

Yes, those coedfficients make sence.

Both of them are negative, which inflects that, longer waiting time and higher price will decrease the probalibity of uber being chosen.

(c) Why can't age be included in the above model?

Because of the endogeneity.

Of course, age can affect utility.

But the model also has an error term, which in fact captures many unobserved or omitted variables, for example:

1. how user-friendly the platform is for older people.
2. whether the platform supports credit cards (which may matter more for adults).

Since these factors are correlated with age, including age directly in the model would lead to endogeneity problems.

3. Estimation Using statsmodels

(a) Re-estimate the same binary logit model using statsmodels.Logit. Compare the results with your manual implementation. Hint: They should be similar if done correctly. Due to numerical differences, they may not be exactly the same.

In [32]:
```python
logit_model_1 = Logit(y, X)
```

```
logit_result_1 = logit_model_1.fit(disp=1)

print("\n\nEstimation Using statsmodels\n\n\n",logit_result_1.summary(
```

```
Optimization terminated successfully.
         Current function value: 0.066951
         Iterations 12


Estimation Using statsmodels


                            Logit Regression Results
========================================================================
=======
Dep. Variable:         choice_is_uber   No. Observations:
5000
Model:                          Logit   Df Residuals:
4998
Method:                           MLE   Df Model:
1
Date:                Mon, 22 Sep 2025   Pseudo R-squ.:
0.9033
Time:                        20:47:43   Log-Likelihood:
-334.75
converged:                       True   LL-Null:
-3463.0
Covariance Type:            nonrobust   LLR p-value:
0.000
========================================================================
=======
                 coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------
-------
wait_diff     -0.5241      0.047    -11.167      0.000      -0.616
-0.432
price_diff    -2.0390      0.117    -17.451      0.000      -2.268
-1.810
========================================================================
=======

Possibly complete quasi-separation: A fraction 0.66 of observations can
be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

In [33]:

```
### With Robust Covariance Matrix

logit_result_2_robut = logit_model_1.fit(disp=1, cov_type='HC0')

print("\n\nEstimation Using Statsmodels With Robust Covariance Matrix
```

```
Optimization terminated successfully.
         Current function value: 0.066951
         Iterations 12


Estimation Using Statsmodels With Robust Covariance Matrix


                           Logit Regression Results
======================================================================
=======
Dep. Variable:         choice_is_uber   No. Observations:
5000
Model:                          Logit   Df Residuals:
4998
Method:                           MLE   Df Model:
1
Date:                Mon, 22 Sep 2025   Pseudo R-squ.:
0.9033
Time:                        20:47:43   Log-Likelihood:
-334.75
converged:                       True   LL-Null:
-3463.0
Covariance Type:                  HC0   LLR p-value:
0.000
======================================================================
=======
                 coef    std err          z      P>|z|      [0.025
0.975]
----------------------------------------------------------------------
-------
wait_diff     -0.5241      0.049    -10.595      0.000      -0.621
-0.427
price_diff    -2.0390      0.122    -16.671      0.000      -2.279
-1.799
======================================================================
=======

Possibly complete quasi-separation: A fraction 0.66 of observations can
be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

Almost the same.

(b) What is the implied willingness to wait one fewer minute (in dollars)?

Hint: Think about how much a consumer is willing to pay to save a minute of wait time.

"Willingness to wait one fewer minute (in dollars) means": keeping the overall utility unchanged, if time increases by one minute, then the price must decrease by how much?

In [34]:
```
img = plt.imread('/Users/terrylu/Desktop/UF/Courses/2025-2026/IO/IO_Co
plt.imshow(img); plt.axis("off");
```

$$U_{\text{bar}} = \beta_{\text{wait}} \cdot time + \beta_{\text{price}} \cdot price_1$$

$$U_{\text{bar}} = \beta_{\text{wait}} \cdot (time + 1) + \beta_{\text{price}} \cdot price_2$$

Difference:

$$0 = \beta_{\text{wait}} \cdot 1 + \beta_{\text{price}} \cdot \Delta price$$

$$\Delta p = -\frac{\beta_{\text{wait}}}{\beta_{\text{price}}}$$

In [35]:
```
beta_wait  = logit_result_2_robust.params['wait_diff']
beta_price = logit_result_2_robust.params['price_diff']


delta_price =  - (beta_wait / beta_price)
print(f"The implied willingness to wait one fewer minute (in dollars)
```

```
The implied willingness to wait one fewer minute (in dollars) is: -0.25
70
```

4. Re-estimate with a Product Fixed Effect

In [36]:
```
img = plt.imread('/Users/terrylu/Desktop/UF/Courses/2025-2026/IO/IO_Co
plt.figure(figsize=(10, 10))
plt.imshow(img); plt.axis("off");
```

1. Utility framework

We have two alternatives, Uber and Lyft:

$$U_{Uber} = \alpha_{Uber} + \beta_1 \cdot wait_{Uber} + \beta_2 \cdot price_{Uber}$$

$$U_{Lyft} = \alpha_{Lyft} + \beta_1 \cdot wait_{Lyft} + \beta_2 \cdot price_{Lyft}$$

---

2. Difference in utilities

For binary logit we only need:

$$V = U_{Uber} - U_{Lyft}$$

$$= (\alpha_{Uber} - \alpha_{Lyft}) + \beta_1(wait_{Uber} - wait_{Lyft}) + \beta_2(price_{Uber} - price_{Lyft})$$

Define:

$$\beta_0 = \alpha_{Uber} - \alpha_{Lyft}$$

So:

$$V = \beta_0 + \beta_1 \cdot wait\_diff + \beta_2 \cdot price\_diff$$

Here β_0 is simply the constant capturing Uber's inherent preference relative to Lyft.

---

3. Dummy-variable representation

Some textbooks prefer to write instead:

$$V = \gamma \cdot D_{Uber} + \beta_1 \cdot wait\_diff + \beta_2 \cdot price\_diff$$

- $D_{Uber} = 1$ if the alternative is Uber, 0 if Lyft.
- Since we are already working with the **difference Uber – Lyft**, the relevant case is always Uber, so $D_{Uber} = 1$.

Thus:

$$V = \gamma + \beta_1 \cdot wait\_diff + \beta_2 \cdot price\_diff$$

(a) Re-estimate the logit model, this time including a dummy variable which is 1 if the product is uber and 0 for lyft.

In [37]:
```
X_inter = sm.add_constant(df[['wait_diff', 'price_diff']])
```

```
logit_model_2 = Logit(y, X_inter)

logit_result_2 = logit_model_2.fit(disp=1)

print("\n\nEstimation with Interception\n\n\n",logit_result_2.summary(
```

```
Optimization terminated successfully.
         Current function value: 0.066753
         Iterations 12


Estimation with Interception


                           Logit Regression Results
==============================================================================
=======
Dep. Variable:           choice_is_uber   No. Observations:
5000
Model:                            Logit   Df Residuals:
4997
Method:                             MLE   Df Model:
2
Date:                  Mon, 22 Sep 2025   Pseudo R-squ.:
0.9036
Time:                          20:47:44   Log-Likelihood:
-333.77
converged:                         True   LL-Null:
-3463.0
Covariance Type:              nonrobust   LLR p-value:
0.000
==============================================================================
=======
                 coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------------
-------
const          0.1384      0.099      1.401      0.161      -0.055
0.332
wait_diff     -0.5251      0.047    -11.151      0.000      -0.617
-0.433
price_diff    -2.0450      0.117    -17.433      0.000      -2.275
-1.815
==============================================================================
=======

Possibly complete quasi-separation: A fraction 0.66 of observations can
be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

(b) Why might it be important to include this dummy in the model? What does the intercept capture?

Without the intercept, the model forces the probability of choosing Uber to be 0.5 when wait_diff=0 and price_diff=0.

With the intercept, the model allows for this baseline preference to differ from 0.5.

For example, the intercept here is 0.1384, which means that when wait_diff=0 and price_diff=0, the probability of choosing Uber is 0.5345, which is greater than 0.5. Which indicates that, Uber has a slight baseline preference advantage over Lyft.

5. Prediction

Suppose an individual faces the following options:

- Uber: 22-minute wait, $7
- Lyft: 14-minute wait, $10

(a) Construct the appropriate values for wait_diff and price_diff.

In [38]:
```
wait_diff_pre = 22 - 14

price_diff_pre = 7 - 10
```

(b) Use your most recent model (with the intercept) to predict the probability that the individual chooses Lyft.

In [39]:
```
x_pre = [[1,wait_diff_pre, price_diff_pre]]

y_pre = logit_result_2.predict(x_pre)

y_pre_lyft = 1 - y_pre

print(f"predictd probability of choosing Uber:\n {y_pre}")

print(f"predictd probability of choosing Lyft:\n {y_pre_lyft}")
```

```
predictd probability of choosing Uber:
 [0.88823384]
predictd probability of choosing Lyft:
 [0.11176616]
```

6. Heterogeneity by Age

(a) Add a new interaction term age_wait_diff equal to wait_diff * age.

In [40]:
```
df['age_wait_diff'] = df['wait_diff'] * df['age']
```

(b) Re-estimate the model using wait_diff, price_diff, and age_wait_diff (plus a constant).

In [41]:
```
x_6 = df[['wait_diff', 'price_diff', 'age_wait_diff']]
x_6 = sm.add_constant(x_6)

logit_model_6 = Logit(y, x_6)
logit_result_6 = logit_model_6.fit(disp=0)
print("Estimation with Age & Interaction\n\n\n",logit_result_6.summary
```

```
Estimation with Age & Interaction


                            Logit Regression Results
========================================================================
=======
Dep. Variable:         choice_is_uber   No. Observations:
5000
Model:                          Logit   Df Residuals:
4996
Method:                           MLE   Df Model:
3
Date:                Mon, 22 Sep 2025   Pseudo R-squ.:
0.9036
Time:                        20:47:52   Log-Likelihood:
-333.76
converged:                       True   LL-Null:
-3463.0
Covariance Type:            nonrobust   LLR p-value:
0.000
========================================================================
==========
                    coef    std err          z      P>|z|      [0.025
0.975]
------------------------------------------------------------------------
----------
const             0.1386      0.099      1.402      0.161      -0.055
0.332
wait_diff        -0.5323      0.115     -4.649      0.000      -0.757
-0.308
price_diff       -2.0449      0.117    -17.431      0.000      -2.275
-1.815
age_wait_diff     0.0002      0.003      0.070      0.944      -0.005
0.005
========================================================================
==========

Possibly complete quasi-separation: A fraction 0.66 of observations can
be
perfectly predicted. This might indicate that there is complete
quasi-separation. In this case some parameters will not be identified.
```

The coefficient on age_wait_diff is close to zero and statistically insignificant.

This implies that in this dataset, the effect of wait time on the probability of choosing Uber does not vary systematically with age.

7. Reflection

(a) Suppose a third option was added to the dataset. Specifically, suppose consumers also had the option of choosing uber black (a premium version of uber). Briefly explain one limitation of the logit model in this context.

One limitation of the logit model is the IIA (Independence of Irrelevant Alternatives) assumption: the relative choice probability between any two options does not change when a third option is added or removed.

In other words, in the logit model, the choice share between Uber and Lyft would remain the same regardless of whether Uber Black exists.

However, In reality, this assumption clearly does not hold, because Uber Black is much closer a substitute to Uber than to Lyft (they are available within the same app). As a result, many consumers who originally chose Uber would likely switch to Uber Black, rather than randomly switching away from both Uber and Lyft. However, the logit model would incorrectly predict that the introduction of Uber Black takes market share proportionally from both Uber and Lyft, which is unrealistic.